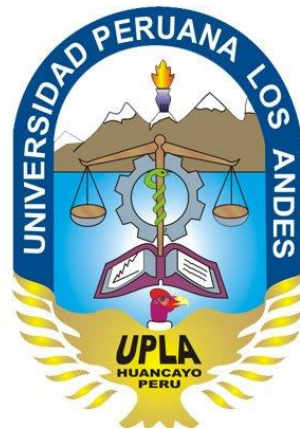


“Año de la Diversificación Productiva y del Fortalecimiento de la Educación”

UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

**ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS Y COMPUTACIÓN**



TESIS

**“CREACIÓN Y APLICACIÓN DE UN FRAMEWORK PARA EL
DESARROLLO WEB ÁGIL, COOPERATIVO, SEGURO Y ESCALABLE,
ORIENTADO A LA OFICINA UNIVERSITARIA DE INFORMÁTICA Y
SISTEMAS DE LA UNIVERSIDAD PERUANA LOS ANDES”.**

PRESENTADO POR:

Bach. Emily Rossy Salinas Chamorro

Bach. José Antonio Rios Monterrey

PARA OPTAR POR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS Y COMPUTACIÓN

HUANCAYO – PERÚ

2015

DR. RUBEN DARIO TAPIA SILGUERA
PRESIDENTE

MG. JORGE VLADIMIR PACHAS HUAYTÁN
JURADO

MG. WAGNER ENOC VICENTE RAMOS
JURADO

MG. JAIME HUMBERTO ORTIZ FERNÁNDEZ
JURADO

MG. MIGUEL ÁNGEL CARLOS CANALES
SECRETARIO DOCENTE

ASESOR:
ING. JOWEL CABRERA PADILLA

DEDICATORIA

El presente trabajo quiero dedicarlo a mis dos hermanos, Abel y Lindaura, por haberme demostrado que una caída no demuestra que estés perdido sino que es un medio para superar las dificultades. Además, por demostrarme la importancia de la perseverancia, responsabilidad y ayudarme a cumplir mis objetivos como persona y como estudiante a través de sus consejos. También quiero dedicarlo a mi hermana Elsa quien desde pequeña me enseñó a que la rectitud y responsabilidad te ayudan a ser una mejor persona cada día.

Emily Rossy

Todo el esfuerzo y sacrificio colocado en cada una de las páginas de este trabajo, se las dedico en especial a mi madre querida, cuyo sueño dorado es disfrutar del éxito de cada uno de sus hijos. A ella le obsequio este pequeño paso que doy en el largo camino que me queda por recorrer en la vida. También quiero dedicar este trabajo a mi padre, quien me enseñó a estudiar y trabajar siempre, aún frente a las adversidades que el mundo te da. Por otro lado, también dedico este trabajo a mi tía Haydé, quien siempre ha estado cerca de mí, cuidándome y celebrando cada uno de mis logros como una segunda madre. Finalmente, esta tesis se la dedico a mis hermanos Pablo, Liliana y Eduardo, quienes siempre están presentes en mi mente como fuente de inspiración para crecer y superarme.

José Antonio

AGRADECIMIENTOS

En primera instancia, quiero agradecer a Dios, quien a través de su palabra me ha ayudado a salir vencedora ante las adversidades, a ser mejor persona cada día, y por bendecirme en cada acción que realizo. Además, quiero agradecer a mis hermanos por cuidarme y por enseñarme el camino de la responsabilidad, del estudio; por el amor que me han brindado y por los sabios consejos que me dan y por tomarme como ejemplo de mis sobrinos pues esto me da fuerzas para seguir adelante y ser mejor persona cada día. Quiero agradecer también a mi enamorado, José, quién me ha demostrado que somos un gran equipo, pues si alguno de los dos tiene alguna una idea simple, los dos la mejoramos y convertimos en una idea superior.

Emily Rossy

Quiero agradecer en primer lugar, a Dios, mi Padre Celestial, por bendecirme cada día, en cada segundo de mi vida, por hacer de mí una persona sincera y humana. En segundo lugar, quiero agradecer a mi hermano Pablo, por sus consejos firmes y sus pensamientos profundos, por creer siempre en mí y admirarme. Quiero agradecer también a mi gran compañera de tantos años, a mi enamorada Emily, con quien no sólo compartimos el ámbito académico que incluye este trabajo, sino también muchas venturas y desventuras que la vida nos ha deparado.

José Antonio

INDICE DE CONTENIDOS

RESUMEN	XII
ABSTRACT	XIII
INTRODUCCIÓN	XIV
PARTE I: GENERALIDADES	17
CAPÍTULO I: ASPECTOS GENERALES	18
1.1. Descripción de la Organización	18
1.1.1. La Universidad Peruana Los Andes	18
1.1.2. La Oficina Universitaria de Informática y Sistemas	18
1.2. Planteamiento del Problema	19
1.2.1. Situación problemática	19
1.2.2. Definición del problema	20
1.3. Objetivos	20
1.3.1. Objetivo General	20
1.3.2. Objetivos Específicos	20
1.4. Limitaciones	20
1.5. Factibilidad	21
1.5.1. Factibilidad Técnica	21
1.5.2. Factibilidad Económica	24
1.5.3. Factibilidad Operativa	26
1.6. Justificación	26
1.6.1. Justificación Práctica	26
1.6.2. Justificación Metodológica	27
1.6.3. Justificación Económica	27
CAPÍTULO II: MARCO TEÓRICO	28
2.1. Antecedentes	28
2.1.1. Antecedentes Nacionales	28
2.1.2. Antecedentes Internacionales	30
2.2. Bases Teóricas	31
2.2.1. Framework Web	31
2.2.2. Escalabilidad en la Web	38
2.2.3. Lenguajes y Tecnologías de Programación	38
2.2.4. Seguridad en el Desarrollo Web	48
2.3. Metodologías	68

2.3.1.	Metodología Bussiness Process Management (BPM).....	69
2.3.2.	Metodología SCRUM.....	71
PARTE II: PRESENTACION DE RESULTADOS.....		77
CAPÍTULO III: ANÁLISIS DE REQUERIMIENTOS.....		78
3.1.	Recopilación de Requerimientos:	78
3.1.1.	Entrevistas.....	78
3.2.	Representación de Requerimientos:.....	83
3.2.1.	Mapa Jerárquico de la Oficina Universitaria de Informática y Sistemas.....	83
3.2.2.	Casos de Uso del negocio.....	84
3.2.3.	Mapa de Procesos de la Oficina Universitaria de Informática y Sistemas mediante BPM.....	89
3.2.4.	Diagrama de Procesos de la Oficina Universitaria de Informática y Sistemas mediante BPM.....	91
3.3.	Diagrama de Casos de Uso del Sistema	93
3.4.	Fase “Pre-Game” de SCRUM	94
3.4.1.	Backlog del Producto: Requisitos del cliente	94
CAPÍTULO IV: DISEÑO DEL FRAMEWORK.....		96
4.1.	Fase “Game” de SCRUM.....	96
4.1.1.	Capa de Acceso a Datos	96
4.1.2.	Base de Datos de Seguridad (Microsoft SQL Server)	102
4.1.3.	Capa de Seguridad.....	109
4.1.4.	Controlador del Framework.....	114
4.1.5.	Capa de Presentación.....	122
4.1.6.	Arquitectura Final.....	129
PARTE III: DISCUSIÓN DE RESULTADOS		134
CAPÍTULO V: PRUEBAS Y APLICACIÓN DEL FRAMEWORK (Fase “Post – Game” de SCRUM) ...		135
5.1.	Pruebas Unitarias	135
5.1.1.	Prueba Unitaria N°1	136
5.1.2.	Prueba Unitaria N°2	139
5.1.3.	Prueba Unitaria N°3	141
5.1.4.	Prueba Unitaria N°4	143
5.1.5.	Prueba Unitaria N°5	145
5.1.6.	Prueba Unitaria N°6	149
5.1.7.	Prueba Unitaria N°7	152
5.1.8.	Prueba Unitaria N°8	153
5.2.	Pruebas de Integración	155

5.2.1.	Prueba de “Agilidad en el desarrollo”	156
5.2.2.	Prueba de “Cooperatividad en el desarrollo”	160
5.2.3.	Prueba de la “Seguridad en la aplicación”	162
5.2.4.	Prueba de la “Escalabilidad en la aplicación”	172
5.3.	Pruebas de Aceptación.....	173
5.3.1.	Prueba de Aceptación – “Oficina Universitaria de Informática y Sistemas”	173
5.3.2.	Prueba de Aceptación – “Unidad de Grados y Títulos”	173
CONCLUSIONES.....		175
RECOMENDACIONES		176
BIBLIOGRAFÍA.....		177
ANEXOS.....		179
ANEXO A: Entrevista a los trabajadores de la Oficina Universitaria de Informática y Sistemas		180
ANEXO B: Documentación del Framework BunnyPHP		197
ANEXO C: Documentación del Sistema de Grados y Títulos (Desarrollado con el Framework BunnyPHP).....		231
ANEXO D: Actas de Aceptación de Proyecto.....		256
ACTA DE ACEPTACIÓN DEL PROYECTO (OFICINA UNIVERSITARIA DE INFORMÁTICA Y SISTEMAS)		257
ACTA DE ACEPTACIÓN DEL PROYECTO (OFICINA UNIVERSITARIA DE INFORMÁTICA Y SISTEMAS)		258

INDICE DE ILUSTRACIONES

Ilustración 2.1: Diagrama de impacto de los riesgos de seguridad al negocio.....	49
Ilustración 2.2: Representación de la Criptografía Simétrica.....	62
Ilustración 2.3: Representación de la Criptografía Asimétrica.....	63
Ilustración 2.4: Ejemplo de un Mapa de Procesos.....	70
Ilustración 3.1: Mapa Jerárquico de la "Oficina Universitaria de Informática y Sistemas"	84
Ilustración 3.2: Diagrama de Casos de Uso - Administrador de Base de Datos.....	85
Ilustración 3.3: Diagrama de Casos de Uso - Administrador de Redes y Servidores	86
Ilustración 3.4: Diagrama de Casos de Uso - Analista de Sistemas.....	87
Ilustración 3.5: Diagrama de Casos de Uso - Encargado de Mesa de Ayuda.....	88
Ilustración 3.6: Diagrama de Casos de Uso - Programador de Sistemas	89
Ilustración 3.7: Mapa de Procesos de la "Oficina Universitaria de Informática y Sistemas"	90
Ilustración 3.8: Diagrama de Procesos de " Implementación de Proyectos tecnológicos e informáticos"	91
Ilustración 3.9: Diagrama de Procesos de " Desarrollo e Implementación del Proyecto"	92
Ilustración 3.10: "Diagrama de Casos de Uso del Sistema"	93
Ilustración 4.1: Burn down chart del Sprint 1 - "Capa de Acceso a Datos"	98
Ilustración 4.2: Casos de Uso – “Capa de Acceso a Datos”	98
Ilustración 4.3: Código fuente de "Conexion.php"	100
Ilustración 4.4: Código fuente de "variables.php"	100
Ilustración 4.5: Código fuente de "accesobd.php"	102
Ilustración 4.6: Burn down chart del Sprint 2: "Base de Datos de Seguridad"	104
Ilustración 4.7: Casos de Uso - "Base de Datos de Seguridad"	104
Ilustración 4.8: Diagrama físico de la Base de Datos	106
Ilustración 4.9: Burn down chart del Sprint 3 - "Capa de Seguridad"	110
Ilustración 4.10: Caso de Uso - "Capa de Seguridad"	110
Ilustración 4.11: Diagrama de "Sistema de Encriptación - Registro de Nuevo Usuario"	113
Ilustración 4.12: Diagrama de "Sistema de Encriptación - Validación de Acceso de Usuarios"	113
Ilustración 4.13: Burn down chart del Sprint 4 - "Controlador"	116
Ilustración 4.14: Caso de Uso - "Controlador"	116
Ilustración 4.15: Código fuente de "uriS_General.php"	118
Ilustración 4.16: Código fuente de "uriD_Pregrado.php"	120
Ilustración 4.17: Código fuente de ".htaccess"	121
Ilustración 4.18: Código fuente de "enrutamiento.php"	122
Ilustración 4.19: Burn down chart del Sprint 5 - "Capa de Presentación"	123
Ilustración 4.20: Casos de Uso - "Capa de Presentación"	123
Ilustración 4.21: Código fuente de "estiloS_login.css"	125
Ilustración 4.22: Código fuente de "estiloS_login.php"	125
Ilustración 4.23: Código fuente de "funcS_registUsuario.js"	126
Ilustración 4.24: Código fuente de "modS_login.js"	128
Ilustración 4.25: Código fuente de "plantS_menu.php"	129
Ilustración 4.26: Burn down chart del Sprint 6 - "Arquitectura Final"	131
Ilustración 4.27: Diagrama de Despliegue del framework "BunnyPHP"	132
Ilustración 4.28: Árbol de Directorios del Framework.....	133
Ilustración 5.1: Entrada de "Caso de Prueba N°1"	138
Ilustración 5.2: Salida de "Caso de Prueba N°1"	139

Ilustración 5.3: Entrada y Salida del "Caso de Prueba N°2"	140
Ilustración 5.4: Entrada del "Caso de Prueba N°3"	142
Ilustración 5.5: Intermedio del "Caso de Prueba N°3"	142
Ilustración 5.6: Salida del "Caso de Prueba N°3"	143
Ilustración 5.7: Entrada en "Caso de Prueba N°4"	144
Ilustración 5.8: Salida en "Caso de Prueba N°4"	145
Ilustración 5.9: Entrada N°1 de "Caso de Prueba N°5"	147
Ilustración 5.10: Salida N°1 de "Caso de Prueba N°5"	147
Ilustración 5.11: Entrada N°2 de "Caso de Prueba N°5"	148
Ilustración 5.12: Salida N°2 de "Caso de Prueba N°5"	148
Ilustración 5.13: Entrada de "Caso de Prueba N°6"	150
Ilustración 5.14: Salida N°1 de "Caso de Prueba N°6"	151
Ilustración 5.15: Salida N°2 de "Caso de Prueba N°6"	151
Ilustración 5.16: Entrada de "Caso de Prueba N°7"	152
Ilustración 5.17: Salida de "Caso de Prueba N°7"	153
Ilustración 5.18: Entrada de "Caso de Prueba N°8"	154
Ilustración 5.19: Salida de "Caso de Prueba N°8"	155
Ilustración 5.20: Código fuente de "uriD_pregrado.php"	158
Ilustración 5.21: Código fuente de "uriD_pregrado.php"	159
Ilustración 5.22: Interfaz del Módulo de "Pregrado"	161
Ilustración 5.23: Interfaz del Módulo de "Administración de Resoluciones"	161
Ilustración 5.24: Captura de Pantalla "RapidSVN en funcionamiento"	161
Ilustración 5.25: Prueba de Inyección "Envío normal de un dato"	163
Ilustración 5.26: Modelo de estructura XML enviada a la Base de Datos sin "Ataque de Inyección"	163
Ilustración 5.27: Prueba de Inyección "Envío de un dato con código SQL"	164
Ilustración 5.28: Modelo de estructura XML enviada a la Base de Datos con "Ataque de Inyección" Fuente: Elaboración propia	164
Ilustración 5.29: Diagrama del manejo de "Códigos de Acceso" en el framework	166
Ilustración 5.30: Prueba ataque XSS "Envío de código malicioso por URL"	167
Ilustración 5.31: Captura de Pantalla "La URL no muestra información de archivos PHP"	168
Ilustración 5.32: Captura de Pantalla "No se puede acceder a archivo protegido"	169
Ilustración 5.33: Prueba Exposición de Datos Sensibles "Creación de usuario con una contraseña básica"	170
Ilustración 5.34: Prueba Exposición Datos Sensibles "Almacenamiento de contraseña básica en la Base de Datos"	170

INDICE DE CUADROS

Cuadro 1.1: Servidor de Aplicaciones (Software Libre)	24
Cuadro 1.2: Servidor de Aplicaciones (Windows)	24
Cuadro 1.3: Servidor de Base de Datos (SQL Server)	24
Cuadro 1.4: Gastos para Desarrollo	25
Cuadro 1.5: Gastos para Implementación	25
Cuadro 1.6: Recursos Operativos - Oficina Universitaria de Informática y Sistemas	26
Cuadro 2.1: Probabilidad e Impacto Técnico de Amenazas.....	50
Cuadro 3.1: Síntesis de Entrevista al Bachiller Irwin Lizandro Yauri Orihuela	79
Cuadro 3.2: Síntesis de Entrevista al Ingeniero Arturo Huber Martínez Bravo	81
Cuadro 3.3: Síntesis de Entrevista al Ingeniero Jowel Cabrera Padilla	83
Cuadro 3.4: Backlog del Producto	95
Cuadro 4.1: Backlog del Sprint 1 - "Capa de Acceso a Datos"	97
Cuadro 4.2: Backlog del Sprint 2 - "Base de Datos de Seguridad (Microsoft SQL Server)"	103
Cuadro 4.3: Diccionario de Datos de la "Base de Datos de Seguridad"	108
Cuadro 4.4: Backlog del Sprint 3 - "Capa de Seguridad"	109
Cuadro 4.5: Backlog del Sprint 4 - "Controlador del Framework"	115
Cuadro 4.6: Backlog del Sprint 5 - "Capa de Presentación" Fuente:.....	122
Cuadro 4.7: "Arquitectura"	130
Cuadro 5.1: Caso de Prueba N°1	138
Cuadro 5.2: Caso de Prueba N°2	140
Cuadro 5.3: Caso de Prueba N°3	141
Cuadro 5.4: Caso de Prueba N°4	144
Cuadro 5.5: Caso de Prueba N°5	146
Cuadro 5.6: Caso de Prueba N°6	150
Cuadro 5.7: Caso de Prueba N°7	152
Cuadro 5.8: Caso de Prueba N°8	154
Cuadro 5.9: Pruebas de "Pérdida de Autenticación y Gestión de Sesiones"	166
Cuadro 5.10: Pruebas de "Referencia Directa Insegura a Objetos"	168

RESUMEN

En la presente tesis, se abordó como problemática la falta de una arquitectura de software definida para el desarrollo de sistemas informáticos en la Oficina Universitaria de Informática y Sistema de la Universidad Peruana Los Andes, lo cual perjudica al desarrollo de aplicaciones al no tener una estructura estándar para llevarlo a cabo, y al mantenimiento al tener una gran variedad de sistemas cuyas características son diferentes, complicando y demorando la comprensión de estos. Es por ello que se planteó como objetivo la construcción de un Framework que permita el desarrollo web ágil, cooperativo, seguro y escalable, el cual pueda ser utilizado por la Oficina Universitaria de Informática y Sistemas para sus próximos proyectos. Para lograrlo se realizaron entrevistas al personal de dicha unidad organizacional, incluyendo al jefe de la oficina, se recopiló información del estatuto de la universidad y documentación acerca de los sistemas administrados por la oficina, etc. Toda esta información fue modelada y diagramada utilizando la Metodología BPM (Business Process Management). Luego, se procedió a desarrollar el framework utilizando la Metodología Ágil de Desarrollo de Software SCRUM, mediante la cual se fueron construyendo cada una de las funcionalidades del framework, avanzando en cada sprint, obteniendo finalmente una estructura de programación que cumplía los objetivos planteados. Este Framework fue llamado “BunnyPHP”. Para verificar la correcta funcionalidad de dicho framework, se procedió a aplicarlo en el desarrollo de un nuevo “Sistema de Grados y Títulos” para la Universidad Peruana Los Andes, comprobando así que “BunnyPHP” permite un desarrollo web ágil y escalable, permitiendo la cooperación de trabajadores con diferentes especialidades sobre un mismo proyecto, y sobre todo dando como resultado aplicaciones seguras.

Palabras clave: Sistema informático, framework web, arquitectura de software, desarrollo web, desarrollo ágil, desarrollo cooperativo, seguridad de aplicaciones, escalabilidad de aplicaciones, sistema gestor de base de datos, lenguaje de programación.

ABSTRACT

In this project, we took as problematic the lack of a defined software architecture for the development of computer systems in the “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, which against the development of applications. Because of the lack of structure standard for software implementation, maintenance and the variety of systems with different characteristics, complicating and delaying the understanding of these, we set out to build a Framework to enable agile, collaborative, secure and scalable web development, which can be used by the “Oficina Universitaria de Informática y Sistemas” for their upcoming projects. To achieve the project, we carried out interviews by staff of the organizational unit, including the head of the office, information university status and documentation was gathered about the office managed systems, etc. All this information was modeled and plotted using the Methodology BPM (Business Process Management). Then we proceeded to develop the framework using the Agile Methodology Software Development SCRUM, by building each of the features of the framework, progressing in each sprint, finally getting a programming structure to achieve the objectives. This Framework was called "BunnyPHP". To verify the correct functionality of the framework, we proceeded to apply it to the development of a new "Sistema de Grados y Títulos" for “Universidad Peruana Los Andes”. Thus, "BunnyPHP" allows a flexible and scalable web development, allowing cooperation workers with different specialties on the same project, especially resulting in secure applications.

Keywords: Computer system, web framework, software architecture, web development, agile development, collaborative development, application security, application scalability, database management system, programming language.

INTRODUCCIÓN

Hoy en día, toda organización, desde una botica o clínica, una ferretería o supermercado, hasta una institución educativa como un colegio o universidad, manejan volúmenes considerables de información, la cual se vuelve cada vez más compleja y se hace necesaria una administración eficiente de esta, para obtener así datos estadísticos confiables que den soporte a la toma de decisiones y faciliten la gestión de cualquiera de estas organizaciones. Es aquí donde nace la necesidad de poseer “Sistemas Informáticos” que automaticen la gestión de datos y generen rápidamente reportes estadísticos que satisfagan las necesidades de toda empresa.

Existen diversas formas de obtener un “Sistema Informático”, todas estas podemos clasificarlas en dos grandes grupos, el primero consistiría en la “Adaptación de un sistema desarrollado previamente, al negocio” y el segundo sería el “Desarrollo de un sistema a medida, tomando en cuenta las características y necesidades peculiares de la empresa”. Actualmente, en el mercado aún no se determina cuál de las dos maneras sea la más conveniente, aunque la respuesta podría ser relativa, dependiendo del negocio, de las características del sistema desarrollado en el primer caso, y de la capacidad del equipo de desarrollo para crear un sistema a medida en el segundo. Los autores de la presente tesis optamos mayormente por la segunda manera, es decir por el “Desarrollo de sistemas a medida”.

Con esta premisa, es importante tener en cuenta que muchas de las organizaciones que manejan estas amplias cantidades de información tienen una “Unidad Organizativa” encargada del desarrollo, mantenimiento y monitoreo de los sistemas. Esta “Unidad Organizativa” podría encontrar obstáculos en su labor, los cuales podrían estar relacionados con la estandarización del desarrollo de software, la determinación de la arquitectura de programación a utilizar, los lenguajes, técnicas y normas a considerar, y hasta la forma en la que cooperan los trabajadores que componen esta unidad.

Es por ello que, en la presente tesis, nos hemos centrado en esta problemática, tomando como organización a la “Universidad Peruana Los Andes”, y como unidad organizativa a la “Oficina Universitaria de Informática y Sistemas”. En dicha oficina se desarrollan los Sistemas Informáticos utilizados por la universidad. La “Oficina Universitaria de Informática y Sistemas” ha ido encontrando, como se venía describiendo, algunas dificultades en su labor. Dificultades como la falta de una “Arquitectura de Software definida”, la deficiente “Escalabilidad” de los sistemas, la falta de “Seguridad” por el lado de la aplicación, la falta de “Interacción dinámica” entre diferentes sistemas, etc.

En razón de lo anteriormente expuesto, los objetivos de la tesis se orientan a la solución de la problemática hallada en la “Oficina Universitaria de Informática y Sistemas” para beneficio general de la universidad. Buscando también desplazar el “Desarrollo de Software” a la nube (internet), lo cual facilitaría el mantenimiento y la implementación de sistemas, optimizaría recursos y diversificaría los tipos de dispositivos desde los cuales

se puede acceder a la aplicación. Es entonces que, los objetivos de la presente tesis se orientan sobre todo a lograr un Desarrollo Web “Ágil”, “Cooperativo”, “Seguro” y “Escalable”.

1. El desarrollo web debe ser “Ágil”, al permitir programar rápidamente y sin mayores complicaciones los requerimientos obtenidos, plasmando rápidamente la lógica del negocio sin necesidad de programar de manera reiterada cuestiones que no tienen que ver con los requerimientos funcionales. Al hablar de un desarrollo web Ágil, nos podemos referir también a la utilización de librerías que optimicen la programación al permitirnos utilizar objetos ya programados previamente por un tercero.
2. Al hablar de un desarrollo web “Cooperativo”, nos referimos a la necesidad de facilitar la interacción de diferentes especialistas en el desarrollo de un mismo sistema. En la “Oficina Universitaria de Informática y Sistemas” se ha encontrado a trabajadores con diferentes especialidades, y entre los que están sobre todo comprometidos con el desarrollo de proyectos tenemos al “Administrador de Base de Datos”, a los “Analistas de Sistemas”, a los “Programadores de Sistemas” y al “Administrador de Redes y Servidores”.
3. Tomando ahora, la tercera característica que se busca en el Desarrollo web, tenemos a la “Seguridad”. Esta sea tal vez la característica más importante que se busca en desarrollo web, pues no afecta solamente al desarrollo, sino también a los datos que maneja la aplicación. Por lo tanto, en el desarrollo web es importante tener en cuenta los riesgos en aplicaciones web más comunes y prevenirlos.
4. La cuarta característica que busca cumplir la presente tesis es la “Escalabilidad”. Esto es muy importante cuando hablamos de la “Universidad Peruana Los Andes”, pues esta institución tienen políticas y procedimientos bastante cambiantes, hablando tanto por la gestión interna (Administración de la misma Universidad) como por la gestión externa (Administración de entidades ajenas a la universidad). Es por ello que los sistemas deben tener una estructura dinámica, que soporte los cambios en la lógica del negocio y que dependa lo menos posible de los datos, es decir, que el funcionamiento del sistema no varíe ante el cambio de datos almacenados en la Base de Datos.

La presente tesis se organiza en 5 capítulos:

El Capítulo I recoge los aspectos generales de la investigación, la cual describe a la “Universidad Peruana Los Andes” y a la “Oficina Universitaria de Informática y Sistemas”, así como a la problemática planteada, los objetivos de la tesis, los alcances y limitaciones, el estudio de factibilidad y la justificación de la investigación. Mediante este capítulo es posible conocer en términos generales las características y circunstancias de la organización en estudio, así como las intenciones de la presente investigación.

En el Capítulo II se desarrolla el “Marco Teórico” de la investigación. En este capítulo se describen los antecedentes y diferentes trabajos cuya temática haya estado

relacionada a la presente tesis, para hallar de esta manera un punto de partida en la investigación, o en todo caso un sustento temático para cuando se requiera. También se incluye en este capítulo la base teórica de la tesis, describiendo todos los términos necesarios para facilitar el entendimiento del lenguaje técnico utilizado en este trabajo. Para finalizar este capítulo, se describen las metodologías útiles para la investigación.

En el Capítulo III se comienza la parte práctica de la tesis mediante el “Análisis de Requerimientos”. En este capítulo se muestran los requerimientos recopilados y su análisis mediante diferentes metodologías.

De la misma manera, en el Capítulo IV se continúa con el “Diseño del Framework” en el cual se demuestran las características de su desarrollo mediante la metodología SCRUM.

Luego de haber logrado el desarrollo del framework, en el Capítulo V se expone al detalle sus características mediante su documentación.

Finalmente se concluye con la “Prueba y aplicación del Framework”. El Capítulo VI describe el desarrollo de un sistema, basándose en los requerimientos de algún proceso de la universidad, utilizando el Framework desarrollado.

En consecuencia, pensamos que la tesis aporta al desarrollo de aplicaciones en la “Universidad Peruana Los Andes”, en específico a la labores de la “Oficina Universitaria de Informática y Sistemas”, pues se espera que a partir de esto, su labor consista, únicamente en la aplicación de la lógica del negocio al framework, el desarrollo cooperativo de las aplicaciones y el mantenimiento de manera escalable de estos sistemas, evitando perder tiempo en el establecimiento de una arquitectura de software, y en la comprensión de un sistema al no tener una arquitectura única definida. Se espera también, el mejoramiento futuro de la presente propuesta a cargo de la misma “Oficina Universitaria de Informática”.

PARTE I: GENERALIDADES

CAPÍTULO I: ASPECTOS GENERALES

1.1. Descripción de la Organización

1.1.1. La Universidad Peruana Los Andes

La Universidad Peruana Los Andes, es una universidad ubicada en la ciudad de Huancayo, Perú. Fue creada el 30 de diciembre de 1983. Reconocida como una de las universidades más prósperas de la Región Junín, la Universidad Peruana Los Andes cuenta con una filial en la ciudad de Lima, dos sedes en “La Merced” y en “Satipo”, y cinco Centros de Asistencia Tutorial (CAT) en Ayacucho, Huancavelica, Tarma, Pasco y Huánuco. La Universidad Peruana Los Andes es una universidad autónoma, esto implica que tiene potestad académica, económica, normativa y administrativa dentro de la Ley. El Gobierno de la Universidad es ejercido por la Asamblea Universitaria, el Consejo Universitario, Rector, el Consejo de Facultad y el Decano. Actualmente las máximas autoridades de la universidad son, el Dr. José Manuel Castillo Custodio, rector de la universidad, el Dr. Jesús Armando Cavero Carrasco, vicerrector académico y el Dr. Casio Aurelio Torres López, vicerrector Administrativo.

1.1.2. La Oficina Universitaria de Informática y Sistemas

La Oficina Universitaria de Informática y Sistemas es una unidad organizacional perteneciente al rectorado de la universidad, la cual se encarga de los sistemas de información y tecnologías, la implementación de hardware y software, normas para el uso adecuado y legal de hardware y software, administración de la red informática y servicios de comunicación, y de la administración del portal de transparencia de información de sus actividades. Actualmente, el Jefe de la unidad es el Ing. Jowel Cabrera Padilla.

1.2. Planteamiento del Problema

1.2.1. Situación problemática

En la actualidad, toda empresa en crecimiento que tenga una cantidad considerable de datos para procesar y convertirlos en información útil, tanto para la toma de decisiones como para sus múltiples actividades, necesita tener acceso a “sistemas informáticos”, los cuales deben haber sido adecuadamente desarrollados, no sólo para lograr un óptimo funcionamiento inmediato, sino para perdurar en el tiempo y adaptarse a los constantes cambios directivos.

Para ello, muchas de estas empresas cuentan con una “Unidad Organizacional” encargada del desarrollo y mantenimiento de estos “Sistemas Informáticos”, en la cual, el personal que la compone puede tener diferentes especialidades, las cuales van desde “Administración de Servidores”, “Administración de Base de Datos” o “Análisis y Desarrollo de Sistemas”, hasta “Mantenimiento y Reparación de Computadoras”. También se da que la permanencia de este personal está sujeta a renovación de contratos, y esto pone en riesgo sobre todo al mantenimiento de los sistemas desarrollados por la unidad, pues la carencia de estándares de programación dificulta, y hasta impide el mantenimiento de aquellos “sistemas informáticos” desarrollados por alguien más. Esto es algo que no le es ajeno a la “Oficina Universitaria de Informática y Sistemas” de la Universidad Peruana Los Andes.

Por otro lado, el desarrollo de software se ve ralentizado debido a que no existe una modalidad de programación, la cual permita una óptima colaboración de los diferentes especialistas que interactúan para culminar el proyecto en el tiempo establecido, es decir, se necesita un desarrollo de software “ágil” y “cooperativo”.

También es importante considerar que algunas de las aplicaciones web que administra actualmente la “Oficina Universitaria de Informática y Sistemas” han sufrido anteriormente ataques informáticos, y aunque se ha mejorado la seguridad en los sistemas, esto se ha hecho únicamente desde la parte de infraestructura y base de datos, quedando pendiente la “seguridad” en las aplicaciones. A todo esto se añade que dichas aplicaciones no han soportado cambios en la lógica de su programación, ni el considerable incremento en la cantidad de datos, necesitando por tanto tomar en cuenta la “escalabilidad” en las futuras aplicaciones a desarrollar.

Finalmente, con respecto al tema del personal contratado que labora en la Oficina Universitaria de Informática y Sistemas de la Universidad Peruana

Los Andes, es importante optimizar la curva de aprendizaje de los trabajadores que laboran en dicha oficina, ya que los requerimientos y cambios en la lógica del negocio son frecuentes, y el personal con el que se cuenta no es estable.

1.2.2. Definición del problema

En la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes” existen problemas al desarrollar y dar soporte a los sistemas informáticos, esto debido a la falta de una arquitectura de software definida, que permita un desarrollo ágil y cooperativo de aplicaciones web seguras y escalables.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar y aplicar un Framework para lograr un desarrollo ágil y cooperativo de aplicaciones web seguras y escalables, orientado a la “Oficina Universitaria de Informática y Sistemas” de la Universidad Peruana Los Andes.

1.3.2. Objetivos Específicos

- Representar los “procesos” y los “casos de uso” de desarrollo de software de la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, para definir sus requerimientos en el desarrollo del framework, mediante la metodología BPM (Business Process Management) y la técnica de entrevistas dirigido al personal de la oficina.
- Construir un “Framework” acorde los requerimientos definidos, para el desarrollo de software en la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, utilizando la Metodología de Desarrollo de Software SCRUM.
- Desarrollar una Aplicación Web orientada a la automatización de algún proceso de la Universidad Peruana Los Andes, para probar y depurar el framework construido, utilizando este mismo en el desarrollo de la aplicación.

1.4. Limitaciones

De acuerdo a la problemática descrita en la presente tesis, el desarrollo del framework abarcará los siguientes puntos:

Con respecto a la aplicación, se considerará la arquitectura de software, el lenguaje de programación, las tecnologías y técnicas de desarrollo, seguridad y protección contra ataques, compatibilidad con sistemas operativos y navegadores, optimización de algoritmos, diseño amigable e intuitivo de interfaz gráfica, y finalmente, optimización y seguridad en la interacción con la base de datos.

Con respecto a la base de datos, el proyecto abarcará procedimientos almacenados, funciones, triggers y tablas propias de la interacción con el framework, mas no con los que estén relacionados con la función del sistema. Por otro lado, con respecto a la Infraestructura de Redes y Servidores, sólo se recopilará información necesaria para la aplicación del framework. Además, se tendrán en cuenta algunos procesos de la Universidad Peruana los Andes y serán modelados como ejemplo de aplicación del framework.

El presente estudio y desarrollo del proyecto no tomará en cuenta la selección y comparación del motor de base de datos, depuración y corrección de los datos almacenados, ni administración de estos, pues el desarrollo del framework se adaptará a lo que la universidad utiliza actualmente.

Finalmente, el presente proyecto de tesis abarcará las reglas, instrucciones y recomendaciones para utilización de framework.

1.5. Factibilidad

Luego de haber definido la problemática y haber sustentado las causas que ameritan la creación de un framework para el desarrollo web en la “Oficina Universitaria Informática y Sistemas”, es acertado realizar un estudio de factibilidad para evaluar la infraestructura tecnológica y la capacidad técnica que implica la creación del framework en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la universidad. Esta evaluación ha permitido determinar las posibilidades de diseñar el framework propuesto y su aplicación en la universidad, los aspectos tomados en cuenta para este estudio fueron clasificados en las siguientes áreas:

1.5.1. Factibilidad Técnica

La Factibilidad Técnica consistió en realizar una evaluación y análisis de la tecnología que la universidad y sobre todo, la Oficina Universitaria de Informática y Sistemas posee; este estudio estuvo destinado a recolectar información sobre las actividades, recursos de hardware y software, métodos y funciones propias de la oficina, la posibilidad de hacer uso de los mismos en el desarrollo del framework propuesto, y si fuera necesario, los requerimientos tecnológicos que deberían ser adquiridos.

Se ha tomado en cuenta que la Oficina Universitaria de Informática y Sistemas maneja como principal Sistema Gestor de Base de Datos al “Microsoft SQLServer”, mientras que las aplicaciones están desarrolladas

en diferentes lenguajes de programación con diferentes tecnologías. Por tanto, es importante que el desarrollo del framework considere, tanto en la parte física como lógica, la interacción con “Microsoft SQLServer”, sin dejar de lado otros sistemas gestores de bases de datos. Pero hablando de la parte de las aplicaciones, el estado actual de la gestión informática de la universidad no precisa lenguajes ni tecnologías, por lo tanto esto quedará libre de establecer, según las necesidades de la oficina.

Para evaluar la tecnología necesaria, se tomaron en cuenta dos enfoques: **Hardware y Software.**

Hardware

Con respecto al Hardware, se evaluó la necesidad de contar con un “Servidor de Aplicación” y un “Servidor de Base de Datos”. A continuación se detallan los requerimientos mínimos del proyecto:

SERVIDOR DE APLICACIÓN

- Procesador de 2 GHz o superior.
- Placa Madre.
- 2 GB de Memoria RAM. La cantidad de Memoria RAM requerida varía según la cantidad de usuarios Web, así como de la cantidad de tareas extras que ejecute el servidor.
- Disco Duro con 80 GB libres.
- Tarjeta de Red.
- Tarjeta de Video
- Monitor.
- Teclado
- Mouse.
- Sistema de Alimentación Ininterrumpida (UPS).

SERVIDOR DE BASE DE DATOS

- Procesador multinúcleo de 2 GHz o superior.
- Placa Madre.
- 4 GB de Memoria RAM. La cantidad de Memoria RAM requerida varía según la cantidad de usuarios Web, así como de la cantidad de tareas extras que ejecute el servidor.
- Disco Duro con 100 GB libres.
- Tarjeta de Red.
- Tarjeta de Video
- Monitor.
- Teclado

- Mouse.
- Sistema de Alimentación Ininterrumpida (UPS).

Luego de haber evaluado los requerimientos mínimos, se determinó que la universidad no necesita hacer una inversión para la adquisición de nuevos equipos, ni repotenciación o actualización de los equipos que posee, ya que estos satisfacen ampliamente los requerimientos mínimos del framework.

Software

Con respecto al Software, se evaluó, al igual que con el Hardware, la necesidad de contar con un “Servidor de Aplicación” y un “Servidor de Base de Datos”, pero en este caso se toma en cuenta el software requerido. A continuación se detallan los requerimientos mínimos del proyecto:

SERVIDOR DE APLICACIÓN

- Sistema Operativo Linux (CentOS, Fedora, Debian).
- Servidor HTTP Apache.
- Servidor Apache Tomcat.
- Lenguaje de Programación PHP.
- Lenguaje de Programación JAVA. (Para reportes .jasper).
- Driver de SQLServer para PHP.
- JavaBridge.

SERVIDOR DE BASE DE DATOS

- Sistema Operativo Windows Server 2008 R2 o superior.
- Microsoft SQLServer 2008 o superior (y todos los requerimientos que este supone).

Luego de haber evaluado los requerimientos mínimos de software, se determinó que la universidad no necesita hacer una inversión para la adquisición de software, ya que los requerimientos por parte del Servidor de Aplicaciones son todos “software libre”, y con respecto al Servidor de Base de Datos, la universidad cuenta con licencia para la utilización del software requerido.

A continuación, en el siguiente cuadro, se hace una descripción del hardware y software disponible en la Universidad Peruana Los Andes y que es utilizado actualmente por la Oficina Universitaria de Informática y Sistemas para brindar el soporte tecnológico necesario.

SERVIDOR DE APLICACIONES (SOFTWARE LIBRE):	
MARCA:	IBM
MODELO	SYSTEM X3850 X5
PROCESADOR:	XEON E7-4820 2.00 Ghz (32 Cores)
MEMORIA RAM:	100GB
DISCO DURO:	1.1 TB
ENERGÍA	1975 W
SISTEMA OPERATIVO Y APLICACIONES:	LINUX CENTOS, APACHE SERVER, PHP 5, ENTRE OTROS.

Cuadro 1.1: Servidor de Aplicaciones (Software Libre)
Fuente: Documentación Servidores OUIS – UPLA 2013

SERVIDOR DE APLICACIONES (WINDOWS):	
MARCA:	DELL
MODELO	POWEREDGE T610
PROCESADOR:	XEON 5500 2.0 Ghz (4 Cores)
MEMORIA RAM:	16GB
DISCO DURO:	1.1 TB
ENERGÍA	570 W
SISTEMA OPERATIVO Y APLICACIONES:	WINDOWS SERVER 2008 R2, NET FRAMEWORK, C#.

Cuadro 1.2: Servidor de Aplicaciones (Windows)
Fuente: Documentación Servidores OUIS – UPLA 2013

SERVIDOR DE BASE DE DATOS (SQL SERVER)	
MARCA:	IBM
MODELO	SYSTEM X3850 X5
PROCESADOR:	XEON E7-4820 2.00 Ghz (32 Cores)
MEMORIA RAM:	100GB
DISCO DURO:	1.1 TB
ENERGÍA	1972 W
SISTEMA OPERATIVO Y APLICACIONES:	WINDOWS SERVER 2008 R2, NET FRAMEWORK, SQLSERVER.

Cuadro 1.3: Servidor de Base de Datos (SQL Server)
Fuente: Documentación Servidores OUIS – UPLA 2013

Por lo tanto, se concluye que el presente proyecto es técnicamente factible.

1.5.2. Factibilidad Económica

La Factibilidad Económica consistió en evaluar las posibilidades que tiene la Universidad Peruana Los Andes para invertir en la adquisición y contrato de hardware y software para dar soporte a la utilización del framework planteado (Gastos para la Implementación). Por otro lado, los gastos que impliquen el diseño y desarrollo del framework, estarán a cargo de los tesisistas (Gastos para el Desarrollo). En los siguientes cuadros se detallan los montos:

GASTOS PARA DESARROLLO

Concepto	Descripción	Monto	Observaciones
Equipos de Cómputo	Laptop Core i5	S/. 2300	----
Fedora v.21	Distribución de Linux	S/. 0	Software Libre
Komodo	IDE de Desarrollo	S/. 0	Software Libre
MS SQL Server	SGBD	S/. 0	Versión Trial
Bizagi Modeler	Software para modelamiento BPM	S/. 0	Descarga Gratuita
Artículos de Escritorio	Pizarra, plumones, post-it, papeles, etc.	S/. 100	----
TOTAL		S/. 2400	

Cuadro 1.4: Gastos para Desarrollo
Fuente: Elaboración Propia

GASTOS PARA IMPLEMENTACIÓN

Concepto	Descripción	Monto	Observaciones
Servidor de Aplicaciones	Hardware	S/. 0	La Oficina Universitaria de Informática y Sistemas ya cuenta con todas estas características.
	Software	S/. 0	
Servidor de Base de Datos	Hardware	S/. 0	
	Software	S/. 0	
Equipos de Cómputo para Desarrollo	IDE de Desarrollo	S/. 0	
TOTAL		S/. 0	

Cuadro 1.5: Gastos para Implementación
Fuente: Elaboración Propia

Como se puede observar, los únicos gastos a ser evaluados en la Factibilidad Económica son aquellos realizados para el desarrollo, los cuales serán asumidos por los tesistas.

Por lo tanto, se concluye que el proyecto es económicamente factible.

1.5.3. Factibilidad Operativa

La Factibilidad Operativa consistió en evaluar, sobre todo a la Oficina Universitaria de Informática y Sistema, para determinar si esta cuenta con el personal necesario para utilizar el framework planteado.

Teniendo el framework la característica de permitir el “Desarrollo Cooperativo” de software, se necesitará el trabajo en equipo de diferentes especialistas en el campo de la informática. A continuación se detallan los requerimientos operativos mínimos:

- 1 Analista de Sistemas.
- 1 Programador Web.
- 1 Administrador de Base de Datos.
- 1 Administrador de Servidores y Redes Informáticas.
- 1 Encargado de Soporte Técnico.

Si fuera necesario, un solo trabajador puede asumir dos o hasta tres de los roles descritos, pero cabe considerar que esto retardaría el desarrollo.

Luego de haber evaluado los requerimientos operativos mínimos, se determinó que la universidad cuenta con el personal necesario para utilizar el framework.

A continuación, en el siguiente cuadro, se hace una descripción de los recursos operativos de la Universidad Peruana Los Andes y en específico de la Oficina Universitaria de Informática y Sistemas:

RECURSOS OPERATIVOS OFICINA UNIV. DE INFORMÁTICA Y SISTEMAS
1 ANALISTA DE SISTEMAS
3 ANALISTAS – PROGRAMADORES DE SISTEMAS
2 ADMINISTRADORES DE BASE DE DATOS
1 ADMINISTRADOR DE REDES Y SERVIDORES
4 ENCARGADOS DE SOPORTE TÉCNICO
1 PRACTICANTE

Cuadro 1.6: Recursos Operativos - Oficina Universitaria de Informática y Sistemas
Fuente: Entrevista Jefe de OUIS - UPLA

1.6. Justificación

1.6.1. Justificación Práctica

La creación de un Framework para el desarrollo de Sistemas Web basándose en los procesos de la “Oficina Universitaria de Informática y Sistemas” de la Universidad Peruana Los Andes, será útil, tanto para el desarrollo como para el mantenimiento de los sistemas en esta oficina, pues estandarizará la

estructura de estos. Por otro lado, el Framework se adaptará a las características de Hardware (Servidores, estructuras de red, etc) y Software (Sistema Gestor de Base de Datos, Sistemas Operativos de servidores, etc), lo cual garantizará “seguridad de datos”, “velocidad de funcionamiento”, “utilización óptima de los recursos de hardware”, “desarrollo cooperativo”, “aplicaciones escalables”, etc.

1.6.2. Justificación Metodológica

En el desarrollo del framework como solución frente a la problemática descrita en la presente tesis, se llevaron a cabo diferentes actividades típicas del desarrollo de software, tales como la recopilación y análisis de requerimientos, diseño del framework, la construcción y finalmente pruebas técnicas y de aceptación. Para lograrlo, se utilizaron algunas metodologías y herramientas, las cuales fueron combinadas y utilizadas de tal manera que resultaron en una metodología específica para la presente investigación tecnológica, la misma que servirá como base para futuros trabajos cuya temática, objetivos y necesidades estén relacionadas al presente trabajo de investigación.

1.6.3. Justificación Económica

El desarrollo de la presente solución, brindará a la Oficina Universitaria de Informática y Sistemas un producto que servirá como un marco de trabajo para el desarrollo de software, del cual, una de sus más notables características será la mayor facilidad de aprendizaje frente a otros productos equivalentes existentes en el mercado, razón por la cual no habrá la necesidad de contratar personal con altos conocimientos en determinados frameworks, sino más bien personal con la habilidad y lógica suficiente para aprender la utilización de esta herramienta. Todo esto tendrá un impacto positivo en los recursos económicos utilizados en personal por la Universidad Peruana Los Andes.

Por otro lado, al agilizar el desarrollo y el mantenimiento del software desarrollado con este framework, se reducirá el tiempo utilizado desde la recopilación de requerimientos hasta la puesta en marcha, obteniendo pronto software útil que optimizará los procesos de la universidad, traduciéndose esto en un ahorro importante para la institución.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes

Para dar sustento al presente trabajo de Investigación Tecnológica, se han analizado algunos trabajos cuya temática se relaciona a la presente:

2.1.1. Antecedentes Nacionales

- En la Tesis “Desarrollo de un Framework Web para el envío remoto de tareas, monitoreo y recuperación de resultados para desktop Grids usando una Arquitectura orientada a servicios: caso BOINC”¹ se plantea la problemática que tienen los científicos al momento de realizar tareas de alta demanda de recursos computacionales, los cuales cuando no pueden ser satisfechos por la adquisición de “supercomputadores”, se solucionan con la utilización de “Infraestructuras para la computación distribuida”, en este caso BOINC. El problema es que la utilización de esta herramienta demanda altos conocimientos técnicos, dificultando la labor de los científicos. La solución que se plantea frente a ello es el desarrollo de un “Framework Web” que permita la fácil interacción entre el usuario final (científicos) y el BOINC, sirviendo como middleware de configuración con una interfaz útil e intuitiva. Como resultado se obtuvo un framework llamado “Sistema Legión”, el cual está comprendido por dos capas, una llamada “Legión Web Interface” cuya utilidad es servir como interfaz de usuario, siendo intuitiva y grafica para el usuario final, y la otra llamada “Legión Web Services for BOINC”, que es la que finalmente ejecutará las operaciones sobre el servidor BOINC y proveerá la posibilidad de ejecutar cálculos en este. El aporte que brinda dicho trabajo de investigación tecnológica a la presente tesis es la importancia y la forma de separar completamente la

¹ Fonseca, P. (2014, Enero) – Pontificia Universidad Católica del Perú. Lima, Perú. - *Desarrollo de un Framework Web para el envío remoto de tareas, monitoreo y recuperación de resultados para desktop Grids usando una Arquitectura orientada a servicios: caso BOINC* [Archivo PDF]. Extraído de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5337/FONSECA_PABLO_FRAMEWORK_WEB_DESKTOP_GRID_ARQUITECTURA_ORIENTADA_SERVICIOS_CASO_BOINC.pdf?sequence=1

interfaz de usuario (UI) de la parte lógica que se conecta con los datos e interactúa con los controladores de software.

- En la Tesis “Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial”² se describe la necesidad de hacer posible la administración y atención de los planes curriculares funcionales y terapéuticos para personas con necesidades especiales, así como consolidar el conocimiento de trastornos. La solución que se plantea en este proyecto es el desarrollo de un sistema que satisfaga los requerimientos anteriormente expuestos, con una arquitectura de cuatro capas con funciones específicas e independientes entre sí: manteniendo las capas de Presentación y Acceso a Datos separadas, mientras que la capa de Lógica de negocio fue dividida en la capa de Seguridad y la capa de Aplicación (navegabilidad entre páginas). Finalmente se consiguió implementar una solución automatizada capaz de administrar los programas educativos, planes de tareas, actividades, entre otros, con una arquitectura en capas que ofrece una mejor escalabilidad para futuras integraciones con nuevas herramientas y servicios aplicando la reutilización de componentes. La tesis descrita nos aporta información con respecto a los beneficios de trabajar con una arquitectura de n capas, incluyendo la utilización de una capa de seguridad a parte de la capa de acceso a datos, la capa lógica y la de presentación.
- En la Tesis “Framework Integral de Seguridad”³ se expone la necesidad de incluir la “seguridad lógica” dentro de un sistema informático, a lo cual la gran mayoría de instituciones u organizaciones dan gran importancia ya que desean que sus clientes se encuentren seguros y confiados de utilizar sus productos. Es por ello que como solución plantean el desarrollo de un framework que pueda integrarse fácilmente con cualquier aplicación, y que brinde seguridad y confiabilidad en todo el proceso que estas abarquen. Como resultado se ha obtenido el XSF (Extreme Security Framework), una solución integral para cualquier institución u organización que desee integrar seguridad a sus sistemas, y posee como objetivo brindar el manejo completo de la seguridad en los procesos de las aplicaciones. Este trabajo brinda un gran aporte, ya que nos expone la importancia que tiene la seguridad en la arquitectura de una aplicación, y de esta manera nos sugiere un camino a seguir en la necesidad de implementar esta característica en el framework que se ha planteado como solución.

² Romero, R. (2012, Setiembre) – Pontificia Universidad Católica del Perú. Lima, Perú. - *Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial* [Archivo PDF]. Extraído de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1562/ROMERO_GALINDO_RAUL_SISTEMA_INFORMACION_EDUCACION_ESPECIAL.pdf?sequence=1

³ Arpasí, L., & Torres, B. (2007) – Universidad Peruana de Ciencias Aplicadas. Lima, Perú. – *Framework de Seguridad* [Archivo Web]. Extraído de <http://tesislatinoamericanas.info/index.php/record/view/34091>

2.1.2. Antecedentes Internacionales

- En la Tesina “Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST”⁴, se aborda la problemática de una empresa dedicada a la distribución de contenidos audiovisuales a través de portales de internet o directamente a terceros a través de interfaces. Dicha empresa cuenta con un framework cuyo mantenimiento se hace cada vez más complicado debido a problemas de diseño. Es por ello que plantean como posibles soluciones la “modificación del que poseen”, la “utilización de alguno existente” o la “creación de uno nuevo”. Como resultado utilizaron y adaptaron el Zend Framework a las necesidades propias de la empresa, demostrando cómo es posible separar la “lógica de negocios”, el “acceso a datos” y la “capa de presentación”, evitando que una afecte a la otra. De esta manera se logra una curva de aprendizaje corta, logrando implementar módulos de forma ordenada y no restrictiva, lo cual permitirá a la empresa afrontar futuros desafíos. Este trabajo nos aporta información acerca de la “Programación por Capas”, describiendo las bondades que tiene la separación de la parte visual de la parte lógica, y estas a su vez del acceso a datos. Del mismo modo, sustenta la importancia que tiene un framework con respecto a su curva de aprendizaje.

- En el trabajo de investigación “Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE”⁵, se plantea la necesidad de desarrollar aplicaciones empresariales de una forma óptima, estándar y simplificada, dando especial importancia al aspecto gráfico de la aplicación. Como solución plantean desarrollar un “Framework de Presentación” personalizado a partir del análisis y comprensión del funcionamiento de la plataforma de programación “J2EE” de Java. Como resultado se obtuvo el “Framework Juoc” y su aplicación en el desarrollo de una “Aplicación de Test”, la cual consiste en una lista de artículos contenidos en un folleto que los presenta y promociona describiendo sus atributos, con pantalla de buscadores, menús, etc. Si bien, el framework planteado en la presente investigación no se orienta solamente a la parte visual, sino a todo lo que engloba una aplicación web, este trabajo nos brinda un enfoque acerca de la importancia que tiene la interfaz de un sistema, y una idea de cómo trabajarlo.

⁴ Roberto, E. (2010, Febrero) – Universidad de Belgrano. Buenos Aires, Argentina. - *Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST* [Archivo PDF]. Extraído de http://www.ub.edu.ar/investigaciones/tesinas/354_Tesina_Zulian.pdf

⁵ Galindo, J., & Camps, J. (2008, Enero) – Universitat Oberta de Catalunya. Cataluña, España. - *Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE* [Archivo PDF]. Extraído de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf>

- En la Tesis “Estudio e implementación de un framework de desarrollo de aplicaciones con funciones de seguridad y privacidad para móviles”⁶, exponen los beneficios de utilizar “aplicaciones móviles corporativas”, como la alta flexibilidad para realizar operaciones de todo tipo desde cualquier lugar. En base a esto, plantean como problemática la carencia de aplicaciones orientadas a la seguridad y a la infraestructura de clave pública (PKI) que ofrezcan de manera cómoda y sencilla servicios como la firma electrónica o el cifrado. Sustentan que, las principales plataformas para dispositivos móviles iOS y Android, disponen de métodos para realizar las principales acciones criptográficas, sin embargo resultan complicados de aplicar si no se tiene un alto entendimiento en la materia. Como solución plantean seleccionar un framework de desarrollo y plataformas que permitan el máximo despliegue en los móviles y realizar un demostrador que valide el uso de dicho framework. Como resultado seleccionaron el framework PhoneGap y se crearon plugins dedicados a operaciones criptográficas de firma para las plataformas iOS y Android. La Tesis descrita nos aporta información acerca de aplicaciones web para móviles, siendo de vital importancia para el desarrollo del Framework planteado, ya que actualmente la tecnología móvil está en crecimiento.

2.2.Bases Teóricas

2.2.1. Framework Web

Es una estructura conceptual de soporte definido con módulos de software concretos, el cual puede servir de base para una organización en el desarrollo de software. Estos módulos incluyen programas, bibliotecas y un lenguaje interpretado para poder desarrollar y unir los diferentes componentes de un proyecto. Un framework es diseñado para facilitar el desarrollo de software, permitiendo a los diseñadores y programadores mayor tiempo para identificar los requerimientos de software. En simples palabras, el framework es el esqueleto de todo desarrollo de software, pues permite la integración de diversos módulos basados en los requerimientos del software.⁷

⁶ Navarro, A. (2013, Abril) – Universitat Politècnica de Catalunya. Cataluña, España. - *Estudio e implementación de un framework de desarrollo de aplicaciones con funciones de seguridad y privacidad para móviles* [Archivo PDF]. Extraído de <http://upcommons.upc.edu/pfc/bitstream/2099.1/18253/1/85737.pdf>

⁷ Comunicad Wikipedia (2014). Framework [Artículo Web]. Extraído de <http://es.wikipedia.org/wiki/Framework>

2.2.1.1. Tipos de Framework

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Cocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.⁸

A continuación los Frameworks Libres más importantes:

2.2.1.1.1. CodeIgniter

CodeIgniter es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. “Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.”

También hay que destacar que CodeIgniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba CodeIgniter “porque es rápido, ligero y parece poco un entorno”.⁹

Características: Algunos de los puntos más interesantes sobre este framework, sobre todo en comparación con otros productos similares, son los siguientes:

- **Versatilidad:** CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.

⁸ Gutiérrez, J. (s.f.) *¿Qué es un framework web?* [Archivo PDF].

Extraído de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

⁹ Comunicad Wikipedia (2014). *EllisLab* [Artículo Web]. Extraído de <http://es.wikipedia.org/wiki/EllisLab>

- **Compatibilidad:** CodeIgniter, es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Por supuesto, funciona correctamente también en PHP 5.
- **Actualizado:** Desde la versión 2 de CodeIgniter ya solo es compatible con la versión 5 de PHP. Para los que todavía usen PHP 4 pueden descargar una versión antigua del framework, como CodeIgniter V 1.7.3, que todavía era compatible. Estas versiones están en la página de descargas de CodeIgniter.
- **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.
- **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.
- **Documentación tutorializada:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial.¹⁰

2.2.1.1.2. CakePHP

Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web. Su principal objetivo es que el programador pueda trabajar de forma estructurada y rápida, sin pérdida de flexibilidad.

¹⁰ Álvarez, M. (2009, noviembre). *CodeIgniter*. [Artículo Web]. Extraído de <http://www.desarrolloweb.com/articulos/codeigniter.html>

Con CakePHP el desarrollo web ya no es monótono porque ofrece las herramientas para que empiece a escribir el código que realmente necesitas: la lógica específica de tu aplicación.

Características:

- Comunidad activa y amistosa.
- Licencia flexible.
- Compatible con PHP4 y PHP5.
- CRUD integrado para la interacción con la base de datos.
- Soporte de aplicación [scaffolding].
- Generación de código.
- Arquitectura Modelo Vista Controlador (MVC).
- Despachador de peticiones [dispatcher], con URLs y rutas personalizadas y limpias.
- Validación integrada.
- Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes [helpers]).
- Ayudantes para AJAX, Javascript, formularios HTML y más.
- Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes.
- Listas de control de acceso flexibles.
- Limpieza de datos.
- Caché flexible.
- Localización
- Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración de Apache¹¹.

2.2.1.1.3. Zend Framework

Es una implementación que usa código 100% orientado a objetos. En la estructura de los componentes de ZF; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un framework de aplicaciones web al combinarse. ZF ofrece una implementación MVC, una abstracción de base de datos, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para

¹¹ Cake Software Foundation (2012). *¿Qué es CakePHP y por qué hay que utilizarlo?* [Artículo Web]. Extraído de <http://book.cakephp.org/1.3/es/The-Manual/Beginning-With-CakePHP/What-is-CakePHP-Why-Use-it.html>

que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como Zend_Auth y Zend_Acl, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los web services más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir el tiempo de desarrollo.¹²

Requerimientos:

Zend Framework tiene los siguientes requerimientos:

- PHP 5.2.4 (o mayor)
- Un servidor web que tenga habilitada la extensión mod_rewrite o similar.

También se debe asegurar de que Apache está configurado para soportar archivos .htaccess. Esto normalmente se hace cambiando la configuración:

AllowOverride None a AllowOverride All en el archivo httpd.conf. Mirar la documentación de la distribución para detalles más exactos. Si no se tiene configurado correctamente mod_rewrite, y el archivo “.htaccess”, no se va a poder navegar a ninguna página aparte de la página de inicio.¹³

2.2.1.1.4. Yii

Yii es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo. El nombre Yii (pronunciado /i:/) es por fácil (en inglés: easy), eficiente (en inglés: efficient) y extensible (en inglés: extensible).

Requerimientos

Para correr una aplicación Web Yii, se necesita tener un servidor Web con soporte PHP 5.1.0 o superior.

¹² Comunicad Wikipedia (2014). *Zend Framework* [Artículo Web]. Extraído de http://es.wikipedia.org/wiki/Zend_Framework

¹³ Borda, W. (2014). *Manual Zend Framework* [Archivo PDF]. Extraído de <https://es.scribd.com/doc/235176349/Manual-Zend-Framework>

Para desarrolladores que deseen utilizar Yii, el entendimiento de Programación Orientada a Objetos (OOP) será de gran ayuda ya que Yii es un framework totalmente basado en OOP.

Características

- Patrón de diseño Modelo Vista Controlador (MVC).
- Database Access Objects (DAO), query builder, Active Record y migración de base de datos.
- Integración con jQuery.
- Entradas de Formulario y validación.
- Widgets de Ajax, como autocompletado de campos de texto y demás.
- Soporte de Autenticación incorporado. Además soporta autorización via role-based access control (RBAC) jerárquico.
- Personalización de aspectos y temas.
- Generación compleja automática de WSDL, especificaciones y administración de peticiones Web service.
- Internacionalización y localización (I18N and L10N). Soporta traducciones, formato de fecha y hora, formato de números, y localización de la vista.
- Esquema de caching por capas. Soporta el caché de datos, caché de páginas, caché por fragmentos y contenido dinámico. El medio de almacenamiento del caché puede ser cambiado.
- El manejo de errores y logging. Los errores son manejados y personalizados, y los log de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
- Las medidas de seguridad incluyen la prevención cross-site scripting (XSS), prevención cross-site request forgery (CSRF), prevención de la manipulación de cookies, etc.
- Herramientas para pruebas unitarias y funcionales basados en PHPUnit y Selenium.
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.
- Generación de código por componentes de Yii y la herramienta por línea de comandos cumple con los estándares de XHTML.
- Cuidadosamente diseñado para trabajar bien con código de terceros. Por ejemplo, es posible usar el código de PHP o Zend Framework en una aplicación Yii.¹⁴

¹⁴ Comunicad Wikipedia (2014). *Yii* [Artículo Web]. Extraído de <http://es.wikipedia.org/wiki/Yii>

2.2.1.1.5. Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características.

Características

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.¹⁵

2.2.2. Escalabilidad en la Web

Escalabilidad es aumentar la capacidad de atender usuarios o volumen de datos de manera lineal con la capacidad de cómputo añadida.

Para que un sistema sea escalable tiene que cumplir tres condiciones:

- Tiene que poder adaptarse a un incremento en el número de usuarios.
- Tiene que poder adaptarse a un incremento en el tamaño de los datos que maneja.
- Tiene que ser mantenible.

Cuanto más fácil sea cumplir estas tres condiciones, más escalable será el sistema.

Y ahora, llegamos al punto que más confusión crea al hablar de escalabilidad: la escalabilidad no es lo mismo que el rendimiento del sistema. El rendimiento (performance) de la aplicación se define como la velocidad a la que procesa las peticiones. Pues bien, en general se puede mejorar fácilmente el rendimiento de las aplicaciones que son escalables, pero lo contrario no es cierto: una aplicación puede rendir muy bien pero no ser escalable.¹⁶

2.2.3. Lenguajes y Tecnologías de Programación

2.2.3.1. HTTP

Es un protocolo de comunicaciones que permite la transferencia de documentos de lenguaje de marcas de hipertexto (HTML) desde servidores web a navegadores web. HTML es un lenguaje de

¹⁵ Comunidad LibrosWeb (2014). *Symfony en pocas palabras*. [Artículo Web]. Extraído de http://librosweb.es/symfony_1_1/capitulo_1/symfony_en_pocas_palabras.html

¹⁶ Fernández-Capel, A. (2010). *Mitos sobre la escalabilidad de aplicaciones web*. [Artículo Web]. Extraído de <http://codigocomestible.com/2010/03/28/mitos-escalabilidad-aplicaciones-web>

identificadores para la creación de documentos que contienen enlaces a información relacionada. Se puede pulsar con el ratón sobre un enlace para acceder a otros documentos, imágenes u objetos multimedia, así como obtener información adicional acerca del elemento enlazado.

Para enviar y recibir documentos HTML e interactuar con la World Wide Web, tanto el servidor como el cliente deben soportar HTTP.¹⁷

Métodos de petición

HTTP define 8 métodos (algunas veces referido como "verbos") que indica la acción que desea que se efectúe sobre el recurso identificado. Lo que este recurso representa, si los datos pre-existentes o datos que se generan de forma dinámica, depende de la aplicación del servidor. A menudo, el recurso corresponde a un archivo o la salida de un ejecutable que reside en el servidor.

HEAD

Pide una respuesta idéntica a la que correspondería a una petición GET, pero sin el cuerpo de la respuesta. Esto es útil para la recuperación de meta-información escrita en los encabezados de respuesta, sin tener que transportar todo el contenido.

GET

Pide una representación del recurso especificado. Por seguridad no debería ser usado por aplicaciones que causen efectos ya que transmite información a través del URI agregando parámetros a la URL.

Ejemplo:

GET /images/logo.png HTTP/1.1 obtiene un recurso llamado logo.png

Ejemplo con parámetros:

/index.php?page=main&lang=es

POST

Envía los datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes o ambas cosas.

PUT

Sube, carga o realiza un upload de un recurso especificado (archivo), es el camino más eficiente para subir archivos a un servidor, esto es porque

¹⁷ IBM AS/400 Information Center (2000). *Protocolo de transferencia de hipertexto (HTTP)*. [Artículo Web]. Extraído de <http://publib.boulder.ibm.com/html/as400/v4r5/ic2931/info/RZAI1RZAHU070HTTPCO.HTM>

en POST utiliza un mensaje multiparte y el mensaje es decodificado por el servidor. En contraste, el método PUT te permite escribir un archivo en una conexión socket establecida con el servidor.

La desventaja del método PUT es que los servidores de hosting compartido no lo tienen habilitado.

Ejemplo:

```
PUT /path/filename.html HTTP/1.1
```

DELETE

Borra el recurso especificado.

TRACE

Este método solicita al servidor que envíe de vuelta en un mensaje de respuesta, en la sección del cuerpo de entidad, toda la data que reciba del mensaje de solicitud. Se utiliza con fines de comprobación y diagnóstico.

OPTIONS

Devuelve los métodos HTTP que el servidor soporta para un URL específico. Esto puede ser utilizado para comprobar la funcionalidad de un servidor web mediante petición en lugar de un recurso específico.

CONNECT

Se utiliza para saber si se tiene acceso a un host, no necesariamente la petición llega al servidor, este método se utiliza principalmente para saber si un proxy nos da acceso a un host bajo condiciones especiales, como por ejemplo "corrientes" de datos bidireccionales encriptadas (como lo requiere SSL).¹⁸

2.2.3.2. PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes.

¹⁸ Comunicad Wikipedia (2000). *Hypertext Transfer Protocol*. [Artículo Web]. Extraído de http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.¹⁹

Características:

Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.

El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).

Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Permite aplicar técnicas de programación orientada a objetos. Incluso aplicaciones como Zend framework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.

No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

Tiene manejo de excepciones (desde PHP5).

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aún haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de

¹⁹ Comunicad Wikipedia (2014). *PHP*. [Artículo Web]. Extraído de <http://es.wikipedia.org/wiki/PHP>

desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

2.2.3.3. HTML

HTML es una implementación del standard SGML (Standard Generalized Markup Language), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

Metalinguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones.

Características:

- Permite crear lenguajes de codificación descriptivos.
- Define una estructura de documentos jerárquica, con elementos y componentes interconectados.
- Proporciona una especificación formal completa del documento.
- No tiene un conjunto implícito de convenciones de señalización. Soporta, por tanto, un conjunto flexible de juegos de etiquetas.
- Los documentos generados por él son legibles.²⁰

2.2.3.4. CSS

En la recopilación del libro virtual “Introducción a CSS”, Javier Eguiluz señala que CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos, su presentación y es imprescindible para crear páginas web complejas.

²⁰ Pedro Rufo Martín (s.f.). *HTML*. [Archivo PDF]. Extraído de <http://www.asptutor.com/zip/cbhtml.pdf>

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.²¹

2.2.3.5. HTTP Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

Ventajas:

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte).²²

2.2.3.6. AJAX

Acrónimo de “Asynchronous JavaScript And XML” (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el

²¹ Comunicad LibrosWeb (s.f.). *Introducción a CSS*. [Artículo Web]. Extraído de http://librosweb.es/css/capitulo_1.html

²² Comunicad Wikipedia (2014). *Servidor HTTP Apache*. [Artículo Web]. Extraído de http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objetoiframe en lugar del XMLHttpRequest para realizar dichos intercambios. PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.²³

²³ Comunicad Wikipedia (s.f.). AJAX. [Artículo Web]. Extraído de <https://es.wikipedia.org/wiki/AJAX>

2.2.3.7. Java

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.²⁴

2.2.3.8. JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.²⁵

²⁴ Comunicad Wikipedia (2014). *Java (lenguaje de programación)*. [Artículo Web]. Extraído de [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programación\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programación))

²⁵ Comunicad Wikipedia (2014). *JavaScript*. [Artículo Web]. Extraído de <http://es.wikipedia.org/wiki/JavaScript>

2.2.3.9. XML

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

XSL funciona como un lenguaje avanzado para crear hojas de estilos. Es capaz de transformar, ordenar y filtrar datos XML, y darles formato basándolo en sus valores. XPath identifica partes de un documento XML concreto, como pueden ser sus atributos, elementos, etc. XLink por su lado, describe un camino estándar para añadir hiperenlaces en un archivo XML. Es decir, es un mecanismo de vinculación a otros documentos XML. Funciona de forma similar a un enlace en una página Web, es decir, funciona como lo haría ``, sólo que a href es un enlace unidireccional. Sin embargo, XLink permite crear vínculos bidireccionales, lo que implica la posibilidad de moverse en dos direcciones. Esto facilita la obtención de información remota como recursos en lugar de simplemente como páginas Web. XPointer funciona como una sintaxis que apunta a ciertas partes de un documento XML, es como una extensión de XPath. Se utiliza para llegar a ciertas partes de un documento XML. Primero, XLink permite establecer el enlace con el recurso XML y luego es XPointer el que va a un punto específico del documento. Su funcionamiento es muy similar al de los identificadores de fragmentos en un documento HTML ya que se añade al final de una URI y después lo que hace es encontrar el lugar especificado en el documento XML. Al ser XPointer una extensión de XPath, XPointer tiene todas las ventajas de XPath y además permite establecer un rango en un documento XML, es decir, con XPointer es posible establecer un punto final y un punto de inicio, lo que incluye todos los elementos XML dentro de esos dos puntos. Finalmente, XQL, lenguaje de consultas, se basa en operadores de búsqueda de un modelo de datos para documentos XML que puede realizar consultas en infinidad de tipos de documentos como son documentos estructurados,

colecciones de documentos, bases de datos, estructuras DOM, catálogos, etc.²⁶

Ejemplo de XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<regiones>
  <region id="0">Costa</ region >
  < region id="1">Sierra</ region >
  < region id="2">Selva</ region >
</ regiones >
```

2.2.3.10.JSON

Es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.²⁷

Ejemplo de XML:

```
{"regiones":[
  {" region ": { "@id": "0", "#text": "Costa" }}
,
  {" region ": { "@id": "1", "#text": "Sierra" }}
,

```

²⁶ Comunicad W3C (s.f.). *Guía Breve de Tecnologías XML*. [Artículo Web]. Extraído de <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>

²⁷ Comunicad JSON (s.f.). *Introducción a JSON*. [Artículo Web]. Extraído de <http://json.org/json-es.html>

```

{"region": { "@id": "2", "#text": "Selva" }}
}]

```

2.2.4. Seguridad en el Desarrollo Web

2.2.4.1. Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP)

El proyecto abierto de seguridad en aplicaciones Web (OWASP por sus siglas en inglés) es una comunidad abierta dedicada a facultar a las organizaciones a desarrollar, adquirir y mantener aplicaciones que pueden ser confiables.²⁸

Los proyectos OWASP se dividen en dos categorías principales: proyectos de documentación y proyectos de desarrollo.

Los proyectos de documentación actuales son:

- **Guía OWASP:** Un enorme documento que proporciona una guía detallada sobre la seguridad de las aplicaciones web.
- **OWASP Top 10:** Documento de alto nivel que se centra sobre las vulnerabilidades más críticas de las aplicaciones web.
- **Métricas:** Un proyecto para definir métricas aplicables de seguridad de aplicaciones web.
- **Legal:** Un proyecto para ayudar a los vendedores y compradores de software a negociar adecuadamente los aspectos de seguridad en sus contratos.
- **Guía de pruebas:** Una guía centrada en la prueba efectiva de la seguridad de aplicaciones web.
- **ISO 17799:** Documentos de apoyo para organizaciones que realicen revisiones ISO 17799.
- **AppSec FAQ:** Preguntas y respuestas frecuentes sobre seguridad de aplicaciones web.

Los proyectos de desarrollo incluyen:

- **WebScarab:** Una aplicación de chequeo de vulnerabilidades de aplicaciones web incluyendo herramientas proxy.

²⁸ Comunicad OWASP (2014). *Sobre OWASP*. [Artículo Web]. Extraído de https://www.owasp.org/index.php/Sobre_OWASP

- **Filtros de validación:** (Stinger para J2EE, filters para PHP):
 - Filtros genéricos de seguridad perimetral que los desarrolladores pueden usar en sus propias aplicaciones.
- **WebGoat:** Una herramienta interactiva de formación y benchmarking para que los usuarios aprendan sobre seguridad de aplicaciones web de forma segura y legal.
- **DotNet:** Un conjunto de herramientas para securizar los entornos .NET.²⁹

2.2.4.1.1. Riesgos de Seguridad en Aplicaciones

Los atacantes pueden potencialmente usar rutas diferentes a través de la aplicación para hacer daño a su negocio u organización. Cada una de estas rutas representa un riesgo que puede, o no, ser lo suficientemente grave como para justificar la atención.

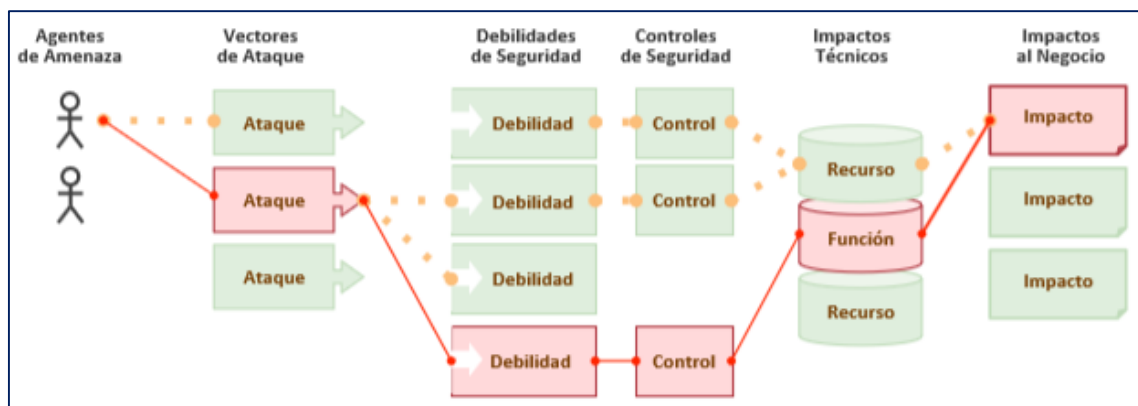


Ilustración 2.1: Diagrama de impacto de los riesgos de seguridad al negocio
Fuente: OWASP Top 10 - 2013

OWASP Top 10

Se enfoca en la identificación de los riesgos más serios para una amplia gama de organizaciones. Para cada uno de estos riesgos, proporciona información genérica sobre la probabilidad y el impacto técnico a través del siguiente esquema de calificaciones.

²⁹ Comunicad Wikipedia (2014). *Open Web Application Security Project*. [Artículo Web]. Extraído de http://es.wikipedia.org/wiki/Open_Web_Application_Security_Project

Agente de amenaza	Vectores de Ataque	Prevalencia de Debilidades	Detectabilidad de Debilidades	Impacto Técnico	Impacto al Negocio
Específico de la aplicación	Fácil	Difundido	Fácil	Severo	Específico de la aplicación / negocio
	Promedio	Común	Promedio	Moderado	
	Difícil	Poco común	Difícil	Menor	

Cuadro 2.1: Probabilidad e Impacto Técnico de Amenazas
Fuente: OWASP Top 10 - 2013

A1 – Inyección

Las fallas de inyección, tales como SQL, OS y LDAP, ocurren cuando los datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder a datos no autorizados.

Vulnerabilidad

Para averiguar si una aplicación es vulnerable a una inyección es verificar que en todo uso de intérpretes se separa la información no confiable del comando o consulta. Para llamados SQL, esto significa usar variables parametrizadas en todas las sentencias preparadas (prepared statements) y procedimientos almacenados, evitando las consultas dinámicas.

Verificar el código es una manera rápida y precisa para ver si la aplicación usa intérpretes de manera segura. Herramientas de análisis de código pueden ayudar al analista de seguridad a ver como se utilizan los intérpretes y seguir el flujo de datos a través de la aplicación. Los testadores pueden validar estos problemas al crear pruebas que confirmen la vulnerabilidad. El análisis dinámico automatizado, el cual ejercita la aplicación puede proveer una idea de si existe alguna inyección explotable. Los analizadores automatizados no siempre pueden alcanzar a los intérpretes y se les dificulta detectar si el ataque fue exitoso. Un manejo pobre de errores hace a las inyecciones fáciles de descubrir.

Prevención

Para evitar una inyección requiere mantener los datos no confiables separados de los comandos y consultas.

1. La opción preferida es usar una API segura la cual evite el uso de intérpretes por completo o provea una interface parametrizada. Sea cuidadoso con las APIs, como los procedimientos almacenados, que son parametrizados, pero que aún pueden introducir inyecciones en el motor del intérprete.
2. Si una API parametrizada no está disponible, debe codificar cuidadosamente los caracteres especiales, usando la sintaxis de escape específica del intérprete.
3. La validación de entradas positiva o de “lista blanca” también se recomienda, pero no es una defensa integral dado que muchas aplicaciones requieren caracteres especiales en sus entradas. Si se requieren caracteres especiales, solo las soluciones anteriores 1 y 2 harían su uso seguro.

A2 - Pérdida de Autenticación y Gestión de Sesiones

Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, claves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.

Vulnerabilidad

Puedes ser vulnerable si:

1. Las credenciales de los usuarios no están protegidas cuando se almacenan utilizando un hash o cifrado.
2. Se pueden adivinar o sobrescribir las credenciales a través de funciones débiles de gestión de la sesión. Por ejemplo: creación de usuarios, cambio de contraseñas, recuperación de contraseñas, ID de sesión débiles).
3. Los ID de sesión son expuestos en la URL. Por ejemplo, reescritura de URL.
4. Los ID de sesión son vulnerables a ataques de fijación de la sesión.
5. Los ID de sesión no expiran, o las sesiones de usuario o los tokens de autenticación. En particular, los tokens de inicio de sesión único (SSO), no son invalidados durante el cierre de sesión.
6. Los ID de sesiones no son rotados luego de una autenticación exitosa.

7. Las contraseñas, ID de sesión y otras credenciales son transmitidas a través de canales no cifrados.

Prevención

1. Un único conjunto de autenticación y gestión de sesiones fuerte. Dichos controles deberán conseguir:
 - a. Cumplir con todos los requisitos de autenticación y gestión de sesiones definidos en el Application Security Verification Standard (ASVS) de OWASP, secciones V2 (Autenticación) y V3 (Gestión de sesiones).
 - b. Tener un interfaz simple para los desarrolladores. Considerar el uso de ESAPI Authenticator y las APIs de usuario como buenos ejemplos a seguir, utilizar o sobre los que construir.
2. Se debe realizar un gran esfuerzo en evitar vulnerabilidades de XSS que podrían ser utilizadas para robar ID de sesión.

A3 - Secuencia de Comandos en Sitios Cruzados (XSS)

Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario. Destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

Vulnerabilidad

Es vulnerable si no asegura que todas las entradas de datos ingresadas por los usuarios son codificadas adecuadamente; o si no se verifica en el momento de ingreso que los datos sean seguros antes de ser incluidos en la página de salida. Sin la codificación o validación debida, dicha entrada será tratada como contenido activo en el navegador. De utilizarse Ajax para actualizar dinámicamente la página, ¿utiliza una API de Javascript segura? De utilizar una API de Javascript insegura, se deben realizar la codificación o validación de las entradas. Mediante el uso de herramientas automatizadas se pueden identificar ciertas vulnerabilidades de XSS. Sin embargo, cada

aplicación construye las páginas de salida de forma diferente y utiliza distintos intérpretes en el navegador como Javascript, ActiveX, Flash o Silverlight, dificultando la detección automática. Una cobertura completa requiere además de enfoques automáticos, una combinación de técnicas como la revisión manual de código y de pruebas de penetración.

Las tecnologías Web 2.0. como Ajax, hacen que XSS sea mucho más difícil de detectar mediante herramientas automatizadas.

Prevención

Prevenir XSS requiere mantener los datos no confiables separados del contenido activo del navegador.

1. La opción preferida es codificar los datos no confiables basados en el contexto HTML (cuerpo, atributo, Javascript, CSS o URL) donde serán ubicados.
2. También se recomienda la validación de entradas positiva o de “lista blanca”, considerando que esta técnica no es una defensa completa ya que muchas aplicaciones requieren aceptar caracteres especiales como parte de las entradas válidas. Dicha validación debe, en la medida de lo posible, validar el largo, los caracteres, el formato y reglas de negocio que debe cumplir el dato antes de aceptarlo como entrada.
3. Para contenido en formato enriquecido, considere utilizar bibliotecas de auto sanitización como AntiSamv de OWASP o el proyecto sanitizador de HTML en Java.
4. Considere utilizar políticas de seguridad de contenido (CSP)

A4 - Referencia directa insegura a objetos

Una referencia directa a objetos ocurre cuando un desarrollador expone una referencia a un objeto de implementación interno, tal como un fichero, directorio, o base de datos. Sin un chequeo de control de acceso u otra protección, los atacantes pueden manipular estas referencias para acceder datos no autorizados.

Vulnerabilidad

La mejor manera de poder comprobar si una aplicación es vulnerable a referencias inseguras a objetos es verificar que

todas las referencias a objetos tienen las protecciones apropiadas. Para conseguir esto, considera:

1. Para referencias directas a recursos restringidos, la aplicación necesitaría verificar si el usuario está autorizado para acceder al recurso en concreto.

Prevención

Requiere seleccionar una forma de proteger los objetos accesibles por cada usuario (identificadores de objeto, nombres de fichero):

1. Utilizar referencias indirectas por usuario o sesión. Esto evitaría que los atacantes accedieran directamente a recursos no autorizados. Por ejemplo, en vez de utilizar la clave del recurso de base de datos, se podría utilizar una lista de 6 recursos que utilizase los números del 1 al 6 para indicar cuál es el valor elegido por el usuario. La aplicación tendría que realizar la correlación entre la referencia indirecta con la clave de la base de datos correspondiente en el servidor.
2. Comprobar el acceso. Cada uso de una referencia directa a un objeto de una fuente que no es de confianza debe incluir una comprobación de control de acceso para asegurar que el usuario está autorizado a acceder al objeto solicitado.

A5 - Configuración de Seguridad Incorrecta

Una buena seguridad requiere tener una configuración segura, definida e implementada para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de datos, y plataforma. Esto incluye mantener todo el software actualizado, incluidas las librerías de código utilizadas por la aplicación.

Vulnerabilidad

¿Cuenta su aplicación con el apropiado fortalecimiento en seguridad a través de todas las capas que la componen?

Incluyendo:

1. ¿Tiene algún software sin actualizar? Esto incluye el SO, Servidor Web/Aplicación, DBMS, aplicaciones, y todas las librerías de código.
2. ¿Están habilitadas o instaladas alguna característica innecesaria (puertos, servicios, páginas, cuentas, privilegios)?

3. ¿Están las cuentas por defecto y sus contraseñas aún habilitadas y sin cambiar?
4. ¿Su manejo de errores revela rastros de las capas de aplicación u otros mensajes de error demasiado informativos a los usuarios?
5. ¿Están las configuraciones de seguridad en su framework de desarrollo (Struts, Spring, ASP.NET) y librerías sin configurar a valores seguros?

Sin un proceso repetible y concertado de configuración de seguridad para las aplicaciones, los sistemas están en alto riesgo.

Prevención

Las recomendaciones primarias son el establecimiento de todo lo siguiente:

1. Un proceso rápido, fácil y repetible de fortalecimiento para obtener un entorno apropiadamente asegurado. Los entornos de Desarrollo, QA y Producción deben ser configurados idénticamente (con diferentes contraseñas usadas en cada entorno). Este proceso puede ser automático para minimizar el esfuerzo de configurar un nuevo entorno seguro.
2. Un proceso para mantener y desplegar las nuevas actualizaciones y parches de software de una manera oportuna para cada entorno.
3. Una fuerte arquitectura de aplicación que proporcione una separación segura y efectiva entre los componentes.
4. Considere ejecutar, escanear y realizar auditorías periódicamente para ayudar a detectar fallos de configuración o parches omitidos.

A6 - Exposición de datos sensibles

Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.

Vulnerabilidad

Lo primero que debe determinar es el conjunto de datos sensibles que requerirán protección extra. Por ejemplo, contraseñas, números de tarjetas de crédito, registros médicos, e información personal deberían protegerse. Para estos datos:

1. ¿Se almacenan en texto claro a largo plazo, incluyendo sus respaldos?
2. ¿Se transmite en texto claro, interna o externamente? El tráfico por Internet es especialmente peligroso.
3. ¿Se utiliza algún algoritmo criptográfico débil/antiguo?
4. ¿Se generan claves criptográficas débiles, o falta una adecuada rotación o gestión de claves?
5. ¿Se utilizan tanto cabezales como directivas de seguridad del mismo navegador cuando son enviados o provistos por el mismo?

Prevención

Los riesgos completos de utilizar cifrado de forma no segura, uso de SSL, y protección de datos escapan al alcance del Top 10. Dicho esto, para los datos sensibles, se deben realizar como mínimo lo siguiente:

1. Considere las amenazas de las cuáles protegerá los datos (por ejemplo: atacante interno, usuario externo), asegúrese de cifrar los datos sensibles almacenados o en tráfico de manera de defenderse de estas amenazas.
2. No almacene datos sensibles innecesariamente, descártelos apenas sea posible. Datos que no se poseen no pueden ser robados.
3. Asegúrese de aplicar algoritmos de cifrado fuertes y estándar así como claves fuertes y gestiónelas de forma segura.
4. Asegúrese que las claves se almacenan con un algoritmo especialmente diseñado para protegerlas, como podrían ser “bcrypt”, “PBKDF2” o “scrypt”.
5. Deshabilite el autocompletar en los formularios que recolectan datos sensibles. Deshabilite también el cacheado de páginas que contengan datos sensibles.

A7 - Ausencia de Control de Acceso a Funciones

Mayoría de aplicaciones web verifican los derechos de acceso a nivel de función antes de hacer visible en la misma interfaz

de usuario. A pesar de esto, las aplicaciones necesitan verificar el control de acceso en el servidor cuando se accede a cada función. Si las solicitudes de acceso no se verifican, los atacantes podrán realizar peticiones sin la autorización apropiada.

Vulnerabilidad

La mejor manera de determinar si una aplicación falla en restringir adecuadamente el acceso a nivel de funcionalidades es verificar cada funcionalidad de la aplicación.

1. ¿La interfaz de usuario (UI) muestra la navegación hacia funcionalidades no autorizadas?
2. ¿Existe autenticación del lado del servidor, o se han perdido las comprobaciones de autorización?
3. ¿Los controles del lado del servidor se basan exclusivamente en la información proporcionada por el atacante?

Usando un proxy, navegue su aplicación con un rol privilegiado. Luego visite reiteradamente paginas restringidas usando un rol con menos privilegios. Si el servidor responde a ambos por igual, probablemente es vulnerable. Algunas pruebas de proxies apoyan directamente este tipo de análisis. También puede revisar la implementación del control de acceso en el código. Intente seguir una solicitud unitaria y con privilegios a través del código y verifique el patrón de autorización. Luego busque en el código para detectar donde no se está siguiendo ese patrón.

Las herramientas automatizadas no suelen encontrar estos problemas.

Prevención

La aplicación debería tener un módulo de autorización consistente y fácil de analizar, invocado desde todas las funciones de negocio. Frecuentemente, esa protección es provista por uno o más componentes externos al código de la aplicación.

1. El proceso de gestión de accesos y permisos debería ser actualizable y auditable fácilmente. No lo implemente directamente en el código sin utilizar parametrizaciones.
2. La implementación del mecanismo debería negar todo acceso por defecto, requiriendo el establecimiento

explícito de permisos a roles específicos para acceder a cada funcionalidad.

3. Si la funcionalidad forma parte de un workflow, verifique y asegúrese que las condiciones del flujo se encuentren en el estado apropiado para permitir el acceso.

A8 - Falsificación de Peticiones en Sitios Cruzados

Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima.

Vulnerabilidad

Para conocer si una aplicación es vulnerable, verifique la ausencia de un token impredecible en cada enlace y formulario. En dicho caso, un atacante puede falsificar peticiones maliciosas. Una defensa alternativa puede ser la de requerir que el usuario demuestre su intención de enviar la solicitud, ya sea a través de la re-autenticación, o mediante cualquier otra prueba que demuestre que se trata de un usuario real (por ejemplo, un CAPTCHA).

Centrese en los enlaces y formularios que invoque funciones que permitan cambios de estados, ya que éstos son los objetivos más importantes del CSRF.

Deben verificarse las operaciones de múltiples pasos, ya que no son inmunes a este tipo de ataque. Los atacantes pueden falsificar fácilmente una serie de solicitudes mediante el uso de etiquetas o incluso de código Javascript.

Tenga en cuenta que las cookies de sesión. Direcciones IP de origen, así como otras información enviada automáticamente por el navegador no proveen ninguna defensa ya que esta información también se incluye en las solicitudes falsificadas. La herramienta de pruebas CSRF (CSRF Tester) de OWASP puede ayudar a generar casos de prueba que ayuden a demostrar los daños y peligros de los fallos de tipo CSRF.

Prevención

La prevención CSRF por lo general requiere la inclusión de un token no predecible en cada solicitud HTTP. Estos tokens deben ser, como mínimo, únicos por cada sesión del usuario.

1. La opción preferida es incluir el token único en un campo oculto. Esto hace que el valor de dicho campo se envíe en el cuerpo de la solicitud HTTP, evitando su inclusión en la URL, sujeta a mayor exposición.
2. El token único también puede ser incluido en la propia URL, o un parámetro de la misma. Sin embargo, esta práctica presenta el riesgo e inconveniente de que la URL sea expuesta a un atacante, y por lo tanto, pueda comprometer el token secreto.

CSRF Guard de OWASP puede incluir automáticamente los tokens secretos en Java EE, .NET, aplicaciones PHP. Por otro lado, ESAPI de OWASP incluye también métodos para que los desarrolladores puedan utilizar con tal de evitar este tipo de vulnerabilidades.

3. Requiera que el usuario vuelva a autenticarse, o pruebas que se trata de un usuario legítimo (por ejemplo mediante el uso de CAPTCHA) pueden también proteger frente a ataques de tipo CSRF.

A9 - Utilización de componentes con vulnerabilidades conocidas

Algunos componentes tales como las librerías, los frameworks y otros módulos de software casi siempre funcionan con todos los privilegios. Si se ataca un componente vulnerable esto podría facilitar la intrusión en el servidor o una pérdida seria de datos. Las aplicaciones que utilicen componentes con vulnerabilidades conocidas debilitan las defensas de la aplicación y permiten ampliar el rango de posibles ataques e impactos.

Vulnerabilidad

En teoría, debería ser fácil distinguir si estas usando un componente o biblioteca vulnerable. Desafortunadamente, los reportes de vulnerabilidades para software comercial o de código abierto no siempre especifican exactamente que

versión de un componente es vulnerable en un estándar, de forma accesible. Más aún, no todas las bibliotecas usan un sistema numérico de versiones entendible. Y lo peor de todo, no todas las vulnerabilidades son reportadas a un centro de intercambio fácil de buscar, Sitios como CVE y NVD se están volviendo fáciles de buscar.

Para determinar si es vulnerable necesita buscar en estas bases de datos, así como también mantenerse al tanto de la lista de correos del proyecto y anuncios de cualquier cosa que pueda ser una vulnerabilidad, si uno de sus componentes tiene una vulnerabilidad, debe evaluar cuidadosamente si es o no vulnerable revisando si su código utiliza parte del componente vulnerable y si el fallo puede resultar en un impacto del cual cuidarse.

Prevención

Una opción es no usar componentes que no ha codificado. Pero eso no es realista. La mayoría de los proyectos de componentes no crean parches de vulnerabilidades de las versiones más antiguas. A cambio, la mayoría sencillamente corrige el problema en la versión siguiente. Por lo tanto, actualizar a esta nueva versión es crítico.

Proyectos de software deberían tener un proceso para:

1. Identificar todos los componentes y la versión que están ocupando, incluyendo dependencias (ej: El plugin de versión).
2. Revisar la seguridad del componente en bases de datos públicas, lista de correos del proyecto, y lista de correos de seguridad, y mantenerlos actualizados.
3. Establecer políticas de seguridad que regulen el uso de componentes, como requerir ciertas prácticas en el desarrollo de software, pasar test de seguridad, y licencias aceptables.
4. Sería apropiado, considerar agregar capas de seguridad alrededor del componente para deshabilitar funcionalidades no utilizadas y/o asegurar aspectos débiles o vulnerables del componente.

A10 - Redirecciones y reenvíos no validados

Las aplicaciones web frecuentemente redirigen y reenvían a los usuarios hacia otras páginas o sitios web, y utilizan datos

no confiables para determinar la página de destino. Sin una validación apropiada, los atacantes pueden redirigir a las víctimas hacia sitios de phishing o malware, o utilizar reenvíos para acceder páginas no autorizadas.

Vulnerabilidad

La mejor forma de determinar si una aplicación dispone de redirección y reenvíos no validados, es:

1. Revisar el código para detectar el uso de redirecciones o reenvíos (llamados transferencias en .NET). Para cada uso, identificar si la URL objetivo se incluye en el valor de algún parámetro. Si es así, si la URL objetivo no es validada con una lista blanca, usted es vulnerable.
2. Además, recorrer la aplicación para observar si genera cualquier redirección (códigos de respuesta de HTTP 300-307, típicamente 302). Analizar los parámetros facilitados antes de la redirección para ver si parecen ser una URL de destino o un recurso de dicha URL. Si es así, modificar la URL de destino y observar si la aplicación redirige al nuevo destino.
3. Si el código no se encuentra disponible, se deben analizar todos los parámetros para ver si forman parte de una redirección o reenvío de una URL de destino y probar lo que hacen estos.

Prevención

El uso seguro de reenvíos y redirecciones puede realizarse de varias maneras:

1. Simplemente evitando el uso de redirecciones y reenvíos.
2. Si se utiliza no involucrar parámetros manipulables por el usuario para definir el destino. Generalmente, esto puede realizarse.
3. Si los parámetros de destino no pueden ser evitados, asegúrese que el valor suministrado sea válido y autorizado para el usuario.

Se recomienda que el valor de cualquier parámetro de destino sea un valor de mapeo, el lugar de la dirección URL real o una porción de esta y en el código del servidor traducir dicho valor a la dirección URL de destino. Las aplicaciones pueden utilizar

ESAPI para sobrescribir el método `sendRedirect()` y asegúrese que todos los destinos redirigidos son seguros.³⁰

2.2.4.2. Criptografía Simétrica

Utiliza una clave para la encriptación y descryptación del mensaje. Esta clave se debe intercambiar entre los equipos por medio de un canal seguro. Ambos extremos deben tener la misma clave para cumplir con el proceso.

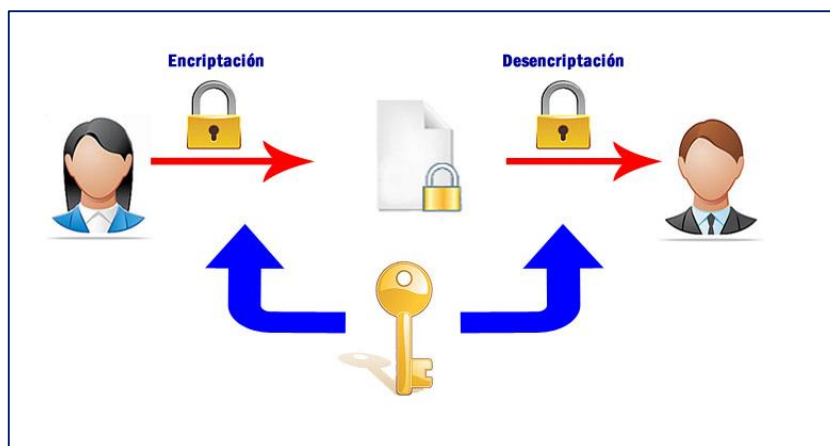


Ilustración 2.2: Representación de la Criptografía Simétrica
Fuente: Elaboración Propia

2.2.4.3. Criptografía Asimétrica

Se basa en el uso de dos claves diferentes, claves que poseen una propiedad fundamental: una clave puede descryptar lo que la otra ha encriptado. Una de las claves de la pareja, llamada clave privada, es usada por el propietario para encriptar los mensajes, mientras que la otra, llamada clave pública, es usada para descryptar el mensaje.³¹

³⁰ Comunicad OWASP (2013). *OWASP Top10 - 2013*. [Archivo PDF]. Extraído de https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf

³¹ Universidad Nacional Abierta y a Distancia (s.f Lección 25: Firma Digital y Certificado Digital. [Artículo Web]. Extraído de http://datateca.unad.edu.co/contenidos/233015/233015Exe/leccin_25_firma_digital_y_certificado_digital.html

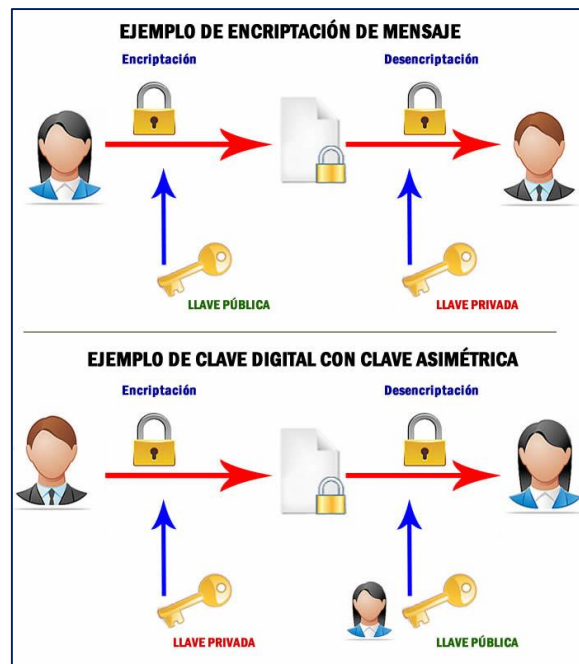


Ilustración 2.3: Representación de la Criptografía Asimétrica
Fuente: Elaboración Propia

2.2.4.4. Algoritmos HASH

2.2.4.4.1. Definición

Una función hash es un método para generar claves o llaves que representen de manera casi unívoca a un documento o conjunto de datos. Es una operación matemática que se realiza sobre un conjunto de datos de cualquier longitud, y su salida es una huella digital, de tamaño fijo e independiente de la dimensión del documento original. El contenido es ilegible.

Es posible que existan huellas digitales iguales para objetos diferentes, porque una función hash, en el caso del SHA-1 tiene 160bits, y los posibles objetos a resumir no tienen un tamaño límite.

A partir de un hash o huella digital, no podemos recuperar el conjunto de datos originales. Los más conocidos son el MD5 y el SHA-1. Cifrar una huella digital se conoce como firma digital.

Requisitos que deben cumplir las funciones hash:

- Imposibilidad de obtener el texto original a partir de la huella digital.
- Imposibilidad de encontrar un conjunto de datos diferentes que tengan la misma huella digital (aunque

como hemos visto anteriormente es posible que este requisito no se cumpla).

- Poder transformar un texto de longitud variable en una huella de tamaño fijo (como el SHA-1 que es de 160bits).
- Facilidad de empleo e implementación.³²

2.2.4.4.2. MD5

En criptografía, MD5 (Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991.

El algoritmo MD5 es una función de cifrado tipo hash que acepta una cadena de texto como entrada, y devuelve un número de 128 bits. Las ventajas de este tipo de algoritmos son la imposibilidad (computacional) de reconstruir la cadena original a partir del resultado, y también la imposibilidad de encontrar dos cadenas de texto que generen el mismo resultado.

Esto nos permite usar el algoritmo para transmitir contraseñas a través de un medio inseguro. Simplemente se cifra la contraseña, y se envía de forma cifrada. En el punto de destino, para comprobar si el password es correcto, se cifra de la misma manera y se comparan las formas cifradas.

Codificación y Algoritmos MD5

La codificación del MD5 funciona de la siguiente manera; su codificación de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal.

Un ejemplo a continuación de una codificación de MD5:

En MD5 ("esto si es una prueba de MD5")

Ahora vemos su Hash de salida=
e99008846853ff3b725c27315e469fbc

³² Comunicad Wikipedia (2014). *Hashing*. [Artículo Web]. Extraído de <https://es.wikipedia.org/wiki/Hashing>

Pero veamos lo complejo que puede llegar a ser, con tan solo cambiar una letra del mensaje original arroja un Hash de salida muy diferente;

En MD5 ("esto no es una prueba de MD5")

Ahora vemos su Hash de salida=
dd21d99a468f3bb52a136ef5beef5034

El simple hecho de codificar un espacio vacío también nos da como resultado un Hash de salida complejo al igual que los antes vistos;

En MD5 (" ")

Ahora vemos su Hash de salida=
d41d8cd98f00b204e9800998ecf8427e

Si en nuestra página web tenemos un sistema de usuarios y queremos proteger las contraseñas para prevenir posibles vulnerabilidades en nuestro servidor, es una medida eficaz encriptar las contraseñas, como se hace con el MD5, de manera que si alguien puede acceder a ellas no pueda ver la contraseña.

Aplicaciones

Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado.

Un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores comparando una suma MD5 publicada con la suma de comprobación del archivo descargado. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una descarga corrupta o incompleta. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software.

Para comprobar la integridad de un archivo descargado de Internet se puede utilizar una herramienta MD5 para comparar la suma MD5 de dicho archivo con un archivo MD5SUM con el resumen MD5 del primer archivo.

El MD5 también se puede usar para comprobar que los correos electrónicos no han sido alterados usando claves públicas y privadas.

Algoritmo MD5

Se comienza suponiendo que se tiene un mensaje de n bits de longitud, escritos $m_0, m_1, \dots, m_{(b-1)}$.

El algoritmo tiene cinco pasos.

1. Adición de bits de relleno.

El mensaje es rellenado con n bits, de tal manera que le falte a su longitud 64 bits para ser múltiplo de 512. El primero de los n bits es 1 y el resto son 0.

2. Adición de la longitud.

La nueva longitud es una representación de 64 bits y es añadida en forma de dos palabras de 32 bits, en primer lugar se muestran los bits menos significativos. Si la longitud del mensaje es mayor que 264, se usan los 64 bits menos significativos.

3. Inicializar los cuatro bufferes, A,B,C y D, que son registros de 32 bits.

Inicializados con los sigs. valores

A: 01 23 45 67

B: 89 ab cd ef

C: fe dc ba 98

D: 76 54 32 10

4. Procesar el mensaje en bloques de 16 bits (se tendrá una entrada y salida de 32 bits).

$$F(X,Y,Z) = (X \text{ AND } Y) \text{ OR } ((\text{NOT}(X)) \text{ AND } Z)$$

$$G(X,Y,Z) = (X \text{ AND } Z) \text{ OR } (Y \text{ AND } (\text{NOT}(Z)))$$

$$H(X,Y,Z) = X \text{ XOR } Y \text{ XOR } Z$$

$$I(X,Y,Z) = Y \text{ XOR } (X \text{ OR } (\text{NOT}(Z)))$$

Se usa una tabla de 64 elementos $T[1 \dots 64]$ construida con la función seno, siendo T_i la parte entera de $294967296 * \text{abs}(\text{sen}(i))$ (i en radianes).

5. Salida.

Mensaje producido por A, B, C, D, empezando con los bits menos significativos de A y terminando con los más significativos de D. Independientemente de la longitud del mensaje, su tamaño será de 128 bits.³³

2.2.4.4.3. SHA

El SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es una familia de funciones hash de cifrado publicadas por el Instituto Nacional de Estándares y Tecnología (NIST). La primera versión del algoritmo fue creada en 1993 con el nombre de SHA, aunque en la actualidad se la conoce como SHA-0 para evitar confusiones con las versiones posteriores. La segunda versión del sistema, publicada con el nombre de SHA-1, fue publicada dos años más tarde. Posteriormente se han publicado SHA-2 en 2001 (formada por diversas funciones: SHA-224, SHA-256, SHA-384, y SHA-512) y la más reciente, SHA-3, que fue seleccionada en una competición de funciones hash celebrada por el NIST en 2012. Esta última versión se caracteriza por ser la que más difiere de sus predecesoras.

A lo largo de su historia, se conocen algunos ataques a esta familia de algoritmos:

En 1998 se encontró una vulnerabilidad para SHA-0, aunque esta no se podía extender a SHA-1. En cualquier caso, la NSA aumentó en ese momento la seguridad del SHA-1.

En 2004 se encontró una debilidad matemática en SHA-1, que permitiría encontrar colisiones de hash más rápido. Sin embargo, este hallazgo resulta poco relevante, pues la complejidad de búsqueda de colisiones pasaría de 2^{80} a 2^{69} , algo que aún es computacionalmente inviable, requiriendo incluso más trabajo que MD5 (2^{64}).

³³ Comunicad Wikipedia (2014). MD5. [Artículo Web]. Extraído de <https://es.wikipedia.org/wiki/MD5>

SHA-1

SHA-1 ha sido examinado muy de cerca por la comunidad criptográfica pública, y no se ha encontrado ningún ataque efectivo. No obstante, en el año 2004, un número de ataques significativos fueron divulgados sobre funciones criptográficas de hash con una estructura similar a SHA-1; lo que ha planteado dudas sobre la seguridad a largo plazo de SHA-1.

SHA-0 y SHA-1 producen una salida resumen de 160 bits (20 bytes) de un mensaje que puede tener un tamaño máximo de 264 bits, y se basa en principios similares a los usados por el profesor Ronald L. Rivest del MIT en el diseño de los algoritmos de resumen de mensaje MD4 y MD5.³⁴

2.2.4.4. Técnica de Encriptación con SALT

Un salt criptográfico es un dato que se utiliza durante el proceso de hash para eliminar la posibilidad de que el resultado pueda buscarse a partir de una lista de pares precalculados de hash y sus entradas originales, conocidas como tablas rainbow.

Es decir, un salt es un pequeño dato añadido que hace que los hash sean significativamente más difíciles de crackear. Existe un gran número de servicios online que ofrecen grandes listas de códigos hash precalculados, junto con sus datos de entrada originales. El uso de salt hace muy difícil o imposible encontrar el hash resultante en cualquiera de estas listas.

2.3. Metodologías

A continuación se describe las metodologías que fueron utilizadas para el desarrollo del presente trabajo de investigación tecnológica:

- Metodología Business Process Management (BPM)

Esta metodología nos permitirá diagramar los procesos que se llevan a cabo en la Oficina Universitaria de Informática y Sistemas de la Universidad Peruana Los Andes, y gracias a ello, representar las actividades y la forma de cómo interactúan los trabajadores en el Desarrollo de Sistemas Informáticos. Teniendo como base esta información gráfica y detallada, será posible tener una idea global de las características que debe tener el Framework planteado.

³⁴ Comunicad Wikipedia (2014). *Secure Hash Algorithm*. [Artículo Web]. Extraído de https://es.wikipedia.org/wiki/Secure_Hash_Algorithm

- **Metodología SCRUM**

Siendo, SCRUM una metodología ágil, se decidió utilizarla puesto que da mayor prioridad a los tiempos de entrega, parametriza las reuniones del equipo y la forma de trabajo optimizando la productividad.

Es preciso aclarar que, si bien estas metodologías pueden interactuar de manera sinérgica para lograr el objetivo del presente trabajo de investigación, en muchos casos se pueden contradecir y obstaculizar el desarrollo por diferencias en sus principios. Es por ello que de estas metodologías se utilizan los elementos y recomendaciones que se crean convenientes, dejando de lado otras que no serán útiles. A continuación se define cada una de las metodologías seleccionadas:

2.3.1. Metodología Business Process Management (BPM)

2.3.1.1. Definición de BPM

Es una metodología empresarial utilizada para la gestión de procesos de negocio mediante su automatización. El objetivo de esta metodología es mejorar el desempeño (Eficiencia, Eficacia y Productividad) de la empresa a través de la “Gestión de Procesos”, los cuales se deben diseñar, modelar, organizar, documentar y optimizar de forma continua.

2.3.1.2. Conceptos relacionados a BPM

2.3.1.2.1. Análisis de Procesos de Negocio (BPA)

BPA es la documentación de modo sistemático de todas las operaciones de la empresa en cada una de las diferentes áreas funcionales. Para ello, se requiere un conjunto de métodos, herramientas y tecnologías utilizadas en el análisis y mejora de los procesos del negocio. En esta fase se procede a modelar los procesos de negocio, lo cual ya supone una mejora importante.³⁵

Mapa de procesos

Es la representación gráfica de todos los procesos que constituyen las actividades de la empresa y sus interrelaciones, tanto entre sí, como con el exterior. Se aconseja recoger los macroprocesos y procesos operativos de la organización, así

³⁵ Cybertec (s.f.). *BPM Business Process Management* (p. 11). Lima, Perú.

como, prescindir de los detalles (procesos de orden inferior o subproceso). La tarea de identificación y diseño del mapa de procesos está asociada al establecimiento de la visión y planificación estratégica de la organización.

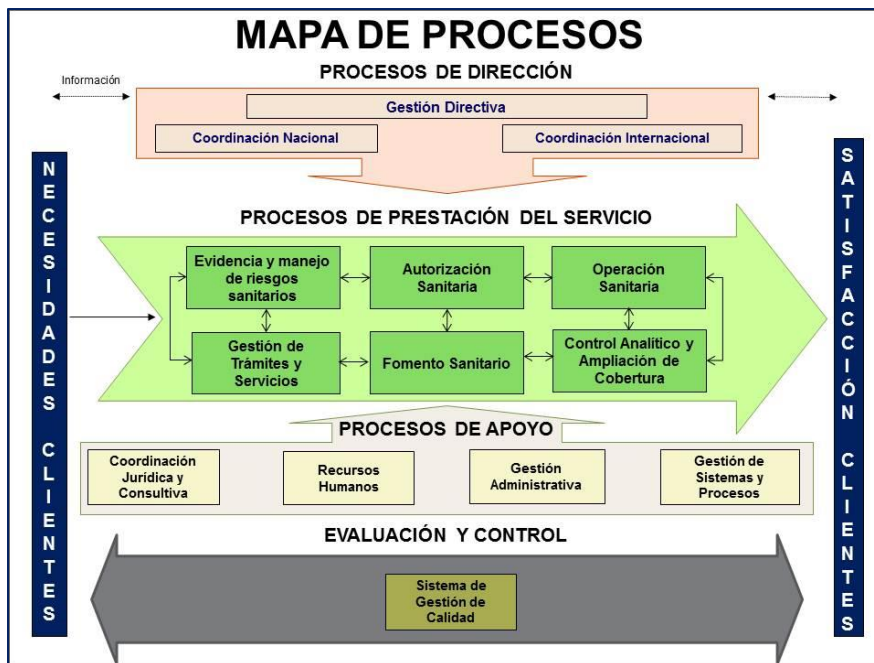


Ilustración 2.4: Ejemplo de un Mapa de Procesos
Fuente: COFEPRIS - Mexico

2.3.1.2.2. Lenguaje de Ejecución de Procesos de Negocio (BPEL)

BPEL es un lenguaje de orquestación: un lenguaje de alto nivel que lleva el concepto de servicio un paso adelante, al proporcionar métodos de definición y soporte para flujos de trabajo y procesos de negocio. Además, permite proveer funciones de manipulación simple de datos, requeridas para definir datos de procesos y flujos de control, así como, usar servicios Web como modelos para la descomposición y ensamblaje de procesos.³⁶

2.3.1.2.3. Arquitectura Orientada a Servicios (SOA)

SOA es un criterio de diseño de software que “define la utilización de servicios para dar soporte a los requisitos de negocio”.³⁷

SOA es una forma de diseñar aplicaciones, que consiste en juntar servicios según las necesidades. El objetivo no es

³⁶ Cybertec (s.f.). *BPM Business Process Management* (p. 12). Lima, Perú.

³⁷ Comunicad Wikipedia (2014). *Arquitectura orientada a servicios*. [Artículo Web]. Extraído de http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios

programar servicios, si no “componer” nuevos servicios basándose en otros básicos ya programados.

2.3.1.2.4. Notación para el Modelado de Procesos de negocio (BPMN)

Es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow). BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG (Object Management Group), después de la fusión de las dos organizaciones en el año 2005. Su versión actual, a abril de 2011, es la 2.0.³⁸

Diagrama de Procesos: Son diagramas de flujo extendido con suficiente información para que el proceso pueda ser analizado, simulado y/o ejecutado. Los diagramas de proceso utilizan el estándar BPMN.

2.3.2. Metodología SCRUM

2.3.2.1. Definición

Scrum es una metodología de desarrollo de software ágil y flexible, esta es utilizada para gestionar el desarrollo teniendo como principal objetivo maximizar el retorno de la inversión de la empresa (ROI). Scrum se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación.

La metodología Scrum promueve que el cliente se comprometa con el proyecto, lo cual permite que este lo vea crecer iteración a iteración, lo cual genera en el cliente un mayor compromiso con el desarrollo. Asimismo le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.

Esta metodología de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades.

³⁸ Comunicad Wikipedia (2014). *Business Process Model and Notation*. [Artículo Web]. Extraído de http://es.wikipedia.org/wiki/Business_Process_Model_and_Notation

2.3.2.2. ¿Qué es Sprint?

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea potencialmente entregable, de manera que cuando el cliente (Product Owner) lo solicite sólo sea necesario un esfuerzo mínimo para que el producto esté disponible para ser utilizado. Para ello, durante la iteración el equipo colabora estrechamente y se llevan a cabo las siguientes dinámicas:

- Cada día el equipo realiza una reunión de sincronización, donde cada miembro inspecciona el trabajo de los otros para poder hacer las adaptaciones necesarias, comunica cuales son los impedimentos con que se encuentra, actualiza el estado de la lista de tareas de la iteración (Sprint Backlog) y los gráficos de trabajo pendiente (Burndown charts).
- El Facilitador (Scrum Master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.
- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

2.3.2.3. El Proceso y actividades

En Scrum un proyecto se ejecuta en iteraciones (bloques temporales cortos y fijos). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la replanificación de objetivos del producto, que realiza durante la iteración con vista a las siguientes iteraciones.

Las actividades que se llevan a cabo en Scrum son las siguientes:

2.3.2.3.1. Planificación de la iteración (Pre - Game)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

Selección de requisitos (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

Planificación de la iteración (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.

2.3.2.3.2. Ejecución de la Iteración (Game)

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

2.3.2.3.3. Cierre (Post – Game)

Preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la versión. El producto queda ahora en la etapa de cierre del desarrollo y posterior distribución.

2.3.2.4. Roles de Scrum

2.3.2.4.1. Roles Principales

Product Owner

El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

ScrumMaster

El *Scrum* es facilitado por un *ScrumMaster*, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *ScrumMaster* no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

Equipo de desarrollo

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

2.3.2.4.2. Roles Auxiliares

Los roles auxiliares en los "equipos Scrum" son aquellos que no tienen un rol formal y no se involucran frecuentemente en el "proceso Scrum", sin embargo deben ser tomados en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (stakeholders). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

Stakeholders (Clientes, Proveedores, Vendedores, etc)

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

Administradores (Managers)

Es la gente que establece el ambiente para el desarrollo del producto.

2.3.2.5. Beneficios de Scrum

Cumplimento de expectativas: El cliente establece sus expectativas indicando el valor que le aporta cada requisito / historia del proyecto, el equipo los estima y con esta información el Product Owner establece su prioridad. De manera regular, en las demos de Sprint el Product Owner comprueba que efectivamente los requisitos se han cumplido y transmite se feedback al equipo.

Flexibilidad a cambios: Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.

Reducción del Time to Market: El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.

Mayor calidad del software: La metódica de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.

Mayor productividad: Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.

Maximiza el retorno de la inversión (ROI): Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.

Predicciones de tiempos: Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia),

con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está en el Backlog.

Reducción de riesgos: El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.³⁹

³⁹ IBM (Noviembre, 2010). Metodología Ágil: Scrum. [Artículo Web]. Extraído de <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational%20Team%20Concert%20for%20Scrum%20Projects/page/SCRUM%20como%20metodolog%C3%ADa>

PARTE II: PRESENTACION DE RESULTADOS

CAPÍTULO III: ANÁLISIS DE REQUERIMIENTOS

3.1. Recopilación de Requerimientos:

3.1.1. Entrevistas

Las entrevistas completas están incluidas en el “ANEXO A: Entrevista a los trabajadores de la Oficina Universitaria de Informática y Sistemas”. A continuación se expone la síntesis de las entrevistas realizadas:

3.1.1.1. Síntesis de Entrevista al Bachiller Irwin Lizandro Yauri Orihuela

ENTREVISTADO	Bach/Ing. Irwin Lizandro Yauri Orihuela
ENTREVISTADORES	Bach/Ing. Emily Rossy Salinas Chamorro Bach/Ing. José Antonio Rios Monterrey
FECHA/LUGAR	10 de Abril del 2014, a las 6:15 p.m. Oficina Universitaria de Informática y Sistemas – Unidad de Desarrollo de Proyectos.
RESUMEN	El área está dividida en dos: Unidad de Desarrollo Unidad de Soporte y Mantenimiento El sistema con más dificultades es el sistema académico <ul style="list-style-type: none">- Lado de la aplicación: No existió análisis de procesos, no se utilizó ninguna metodología, es inseguro.- Lado de la base de datos: está mal estructurado.

	<p>Planificación de la jefatura:</p> <ul style="list-style-type: none"> - La reformulación del sistema académico. - Desarrollo de sistema de caja y policlínico. - Módulo para admisión. <p>Requisitos importantes que el framework debe cumplir:</p> <ul style="list-style-type: none"> - Debe ser ágil. - Fácil de implementar. - Sin código innecesario. - Se debe implementar librerías. - Debe utilizar códigos de acceso. - Debe tener una sesión segura. - Debe tener llaves encriptadas. - URLs limpias. - Seguro contra ataques Get, Post. - Debe ser seguro contra ataques de inyección. <p>Utilización de software libre en el desarrollo web.</p> <ul style="list-style-type: none"> - Antes: Se utilizaba .Net Administrador de base de datos SQL Server. - Ahora: Se utiliza PHP. Servidores: CentOS <p>Tecnologías a utilizar para el desarrollo web</p> <ul style="list-style-type: none"> - Librería jquery para manejar javascript. - Ajax. - PHP.
--	--

Cuadro 3.1: Síntesis de Entrevista al Bachiller Irwin Lizandro Yauri Orihuela
Fuente: Entrevista a trabajadores de OUIS (Anexo A)

3.1.1.2. Síntesis de Entrevista al Ingeniero Arturo Martínez Bravo

ENTREVISTADO	Ing. Arturo Martínez Bravo
ENTREVISTADORES	Bach/Ing. Emily Rossy Salinas Chamorro Bach/Ing. José Antonio Rios Monterrey
FECHA/LUGAR	16 de Abril del 2014, a las 7:30p.m. Oficina Universitaria de Informática y Sistemas – Unidad de Desarrollo de Proyectos.
RESUMEN	Funciones de Unidad de Desarrollo de Proyectos

Debe enfocarse principalmente a crear los sistemas que necesita la universidad.

Limitaciones de la Unidad de Desarrollo de Proyectos

Falta de personal como analistas de sistemas, programadores, analista programadores, administrador de base de datos con conocimientos en manejo de servidores, jefe de proyectos, tester.

Funciones como Administrador de Base de Datos

- Resguardar la información.
- Optimización en base al funcionamiento del sistema.
- Seguridad virtual y física de la base de datos.

Seguridad a la base de datos

Debido a que hemos tenido ataques hacia la base de datos:

- Colocamos un filtro cortafuegos para nuestro servidor.
- Solo el administrador de base de datos y el jefe de proyectos manejan el password para restringir el acceso al total del personal de informática.

Dificultades encontradas al comenzar su labor en la oficina de informática

- Como no existía el rol de administrador de base de datos, tuve que trabajar cumpliendo el rol de dar soporte técnico a la parte de programación, y el tema de seguridad de la base de datos era importante, es así como hace dos años se crea también el puesto de administrador.

Dificultades encontradas en el desarrollo de sistemas

- Los sistemas con mayores problemas son el sistema académico y el sistema financiero por la forma como fueron construidos.
- No existió la correcta planificación para la construcción de la arquitectura del sistema; por ello actualmente se hace difícil el mantenimiento del mismo.

Con respecto a la estructura de una base de datos para el desarrollo de sistemas:

- Trabajar en equipo para tomar las consideraciones necesarias en todos los ámbitos.

Con respecto al sistema gestor de base de datos:

- Microsoft SQL Server

	<ul style="list-style-type: none"> - Ventaja: Al ser una institución educativa, el costo de licenciamiento con respecto a un banco es cinco veces menor. - POSTGRE SQL - Ventaja: Siendo POSTGRE, software libre, el costo beneficio es mayor por ser gratis. - Desventaja: Con respecto en tema seguridad el costo es adicional, y mayor a Microsoft SQLServer. <p>Con respecto al framework a implementar:</p> <ul style="list-style-type: none"> - Se debe estructurar un diagrama de base de datos para ser aplicado únicamente a la Universidad. - Debe tener una arquitectura fija para brindar mantenimiento a los sistemas desarrollados. - Debe manifestar seguridad entre el desarrollador y administrador.
--	---

Cuadro 3.2: Síntesis de Entrevista al Ingeniero Arturo Huber Martínez Bravo
Fuente: Entrevista a trabajadores de OUIS (Anexo A)

3.1.1.3. Síntesis de Entrevista al Ingeniero Jowel Cabrera Padilla

ENTREVISTADO	Ing. Jowel Cabrera Padilla
ENTREVISTADORES	Bach/Ing. Emily Rossy Salinas Chamorro Bach/Ing. José Antonio Rios Monterrey
FECHA/LUGAR	16 de Abril del 2014, a las 7:30p.m. Oficina Universitaria de Informática y Sistemas – Unidad de Desarrollo de Proyectos.
RESUMEN	<p>Funciones de la Oficina Universitaria de Informática y Sistemas</p> <ul style="list-style-type: none"> - Planificar, plantear y desarrollar los sistemas de información, o en caso de que no se tenga la disponibilidad de recursos humanos, mandarlo a desarrollar por terceros. - Velar por la normatividad de uso y manejo de tecnologías de información dentro de la universidad. - Tener operativas todas las tecnologías de información implementadas en la universidad. <p>Problemas encontrados en la Oficina Universitaria de Informática</p> <ul style="list-style-type: none"> - La gestión, especialmente de lo que son los sistemas de información, no está hecha bajo ninguna norma.

- En la parte de software, encontramos las aplicaciones desarrolladas sin ninguna documentación, algunos de los aplicativos sólo tenían el código ejecutable.
- En la parte de hardware, no existía una sala de servidores,

Soluciones planteadas en la Oficina Universitaria de Informática

- La última gestión que se está haciendo es implementar el sistema de enfriamiento, y luego dotarla de la seguridad necesaria para que sólo personal autorizado pueda acceder a la sala de servidores.
- Para los códigos de los ejecutables, como estaban en fox, se encontró una herramienta para descompilarlo y se obtuvo el código fuente, y bueno, se siguió descifrando los códigos, y de cierta forma se ha ido documentando.
- Hemos establecido algunos estándares, algunos formatos para poder implementar software desarrollado, se está optando por metodologías de trabajo, etc.

Dificultades encontradas en el desarrollo de sistemas

- Sistema financiero
La mayor dificultad que nos ha traído.
El crecimiento de ingresantes.
El crecimiento de filiales.
Los procesos de SUNAT han cambiado y las modalidades de los pagos han cambiado.
La implementación en Fox Pro, un sistema desfasado.
El desconocimiento de ciertos procesos por parte de la dirección de Economía.
- Sistema Académico
Superó al Sistema Financiero en cuanto al uso de tecnología; sin embargo, no se tuvo el orden ni la visión de la programación ordenada, el desarrollo, el diseño.
Con respecto a la documentación sólo contenía tres gráficos y esto impide el mantenimiento de este sistema.
La arquitectura que tiene es monolítica, no es modular ni por capas.

Con respecto a los sistemas desarrollados bajo su gestión

- Los sistemas desarrollados bajo la gestión son, el Sistema de Trámite Documentario, el Sistema de Grados y Títulos y el Sistema de Control de Asistencia de Docentes.

	<ul style="list-style-type: none"> - Esos tres sistemas se han desarrollado bajo algunos enfoques metodológicos y algunos estándares que ya se han establecido, incluso la utilización de herramientas libres. <p>Con respecto a los sistemas que se desarrollarán:</p> <ul style="list-style-type: none"> - Se ha planteado la utilización del framework como herramienta, para el desarrollo del nuevo sistema académico incluyendo la adaptación de los nuevos procesos. - Para el próximo año corregir el Sistema financiero. <p>Con respecto al sistema gestor de base de datos:</p> <ul style="list-style-type: none"> - MySQL se orienta sobre todo a Páginas Web, es por ello que no soporta gran cantidad de registros. - POSTGRE SQL es un sistema mucho más seguro, preparado para trabajar con miles de miles de registros. Sin embargo, no existe el presupuesto necesario para contratar a un especialista en PostGre SQL. <p>Con respecto al framework a implementar:</p> <ul style="list-style-type: none"> - Debe ser de fácil manejo para darle un uso intuitivo, debe ser manejable por los diversos especialistas que se presenten. - Debe permitir hacer un trabajo cooperativo e incluso para trabajar en equipo, como para aplicar SCRUM. - Debe ser un producto que esté en software libre.
--	--

Cuadro 3.3: Síntesis de Entrevista al Ingeniero Jowel Cabrera Padilla
Fuente: Entrevista a trabajadores de OUIS (Anexo A)

3.2. Representación de Requerimientos:

3.2.1. Mapa Jerárquico de la Oficina Universitaria de Informática y Sistemas

A continuación, en la *Ilustración 3.1*: Mapa Jerárquico de la "Oficina Universitaria de Informática y Sistemas" se expone el mapa jerárquico de la Oficina Universitaria de Informática y Sistemas.

Es importante tener en cuenta la estructura orgánica de esta unidad organizacional puesto que esta nos dará un alcance respecto al entorno en el cuál será utilizado el framework planteado.

Como se puede observar, esta unidad organizativa está encabezada por el "Jefe de la Oficina de Informática". Luego, como apoyo directo, el jefe tiene a la "Secretaria" encargada de la emisión y recepción de documentos, así como a la "Mesa de Ayuda" que se encarga de la atención inmediata a los requerimientos de la oficina. Luego, teniendo en cuenta que la Oficina

Universitaria de Informática y Sistemas está compuesta por dos unidades, la Unidad de Proyecto y la Unidad de Soporte y Redes, la primera se encarga de ejecutar todos los proyectos de la oficina y está encabezada por el “Coordinador de Proyectos”, el cual coordina la labor del “Administrador de Base de Datos” con el “Analista – Programador de Sistemas”. Es importante considerar que por cada uno de estos cargos puede haber más de un empleado. Por otro lado, la otra Unidad es encabezada por el “Coordinador de Soporte y Redes”, el mismo que tiene a su cargo la coordinación de la labor del “Administrador de Servidores”, el “Administrador de Redes Informáticas” y los “Técnicos de Soporte”.

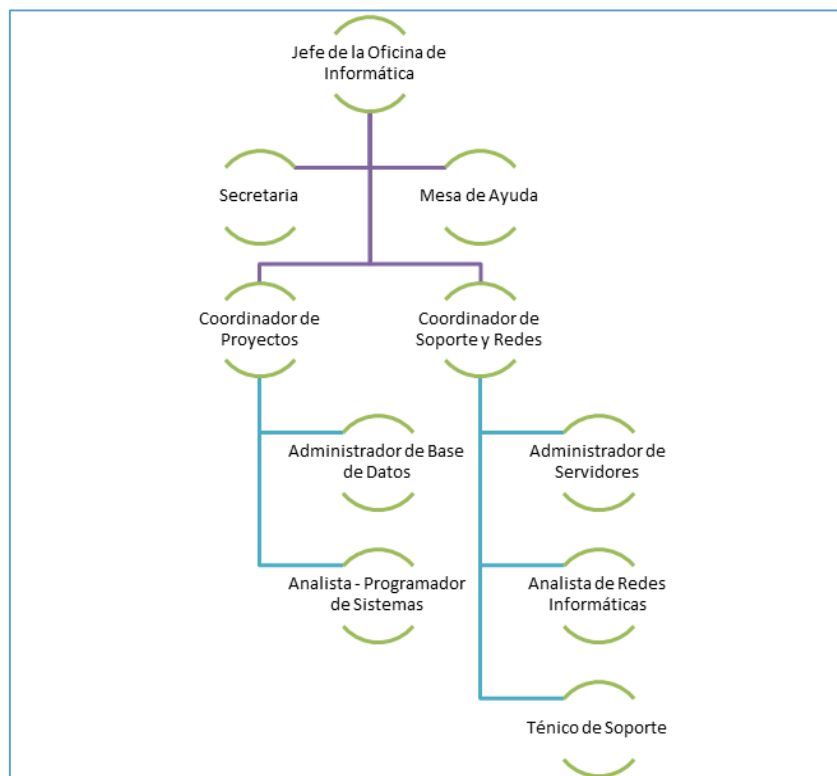


Ilustración 3.1: Mapa Jerárquico de la "Oficina Universitaria de Informática y Sistemas"
Fuente: Entrevista a Jefe de OUIS

3.2.2. Casos de Uso del negocio

Como parte de la representación de toda la información obtenida en la recopilación de requerimientos, se presenta a continuación los diagramas de casos de uso de cada uno de los actores (trabajadores) de la Oficina Universitaria de Informática y Sistemas.

3.2.2.1. Casos de Uso del Administrador de Base de Datos

El Administrador de Base de Datos (DBA) es el especialista que administra las tecnologías de información y la comunicación, siendo responsable de los aspectos técnicos, tecnológicos, científicos, inteligencia de negocios y aspectos legales de bases de datos.

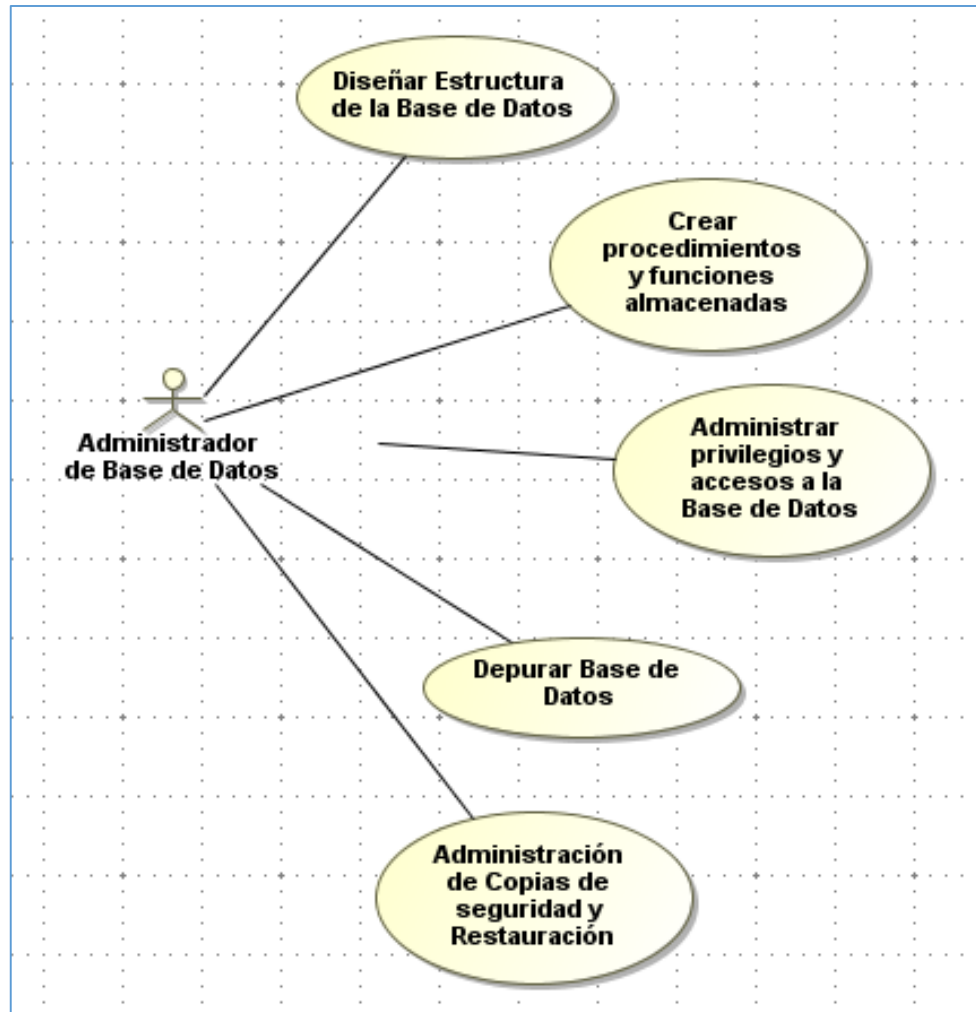


Ilustración 3.2: Diagrama de Casos de Uso - Administrador de Base de Datos
Fuente: Elaboración propia

3.2.2.2. Casos de Uso del Administrador de Redes y Servidores

El Administrador de Redes y Servidores es el especialista encargado de la administración, configuración y mantenimiento de los servidores y las redes informáticas. Se encarga también de los proyectos relacionados a este rubro.

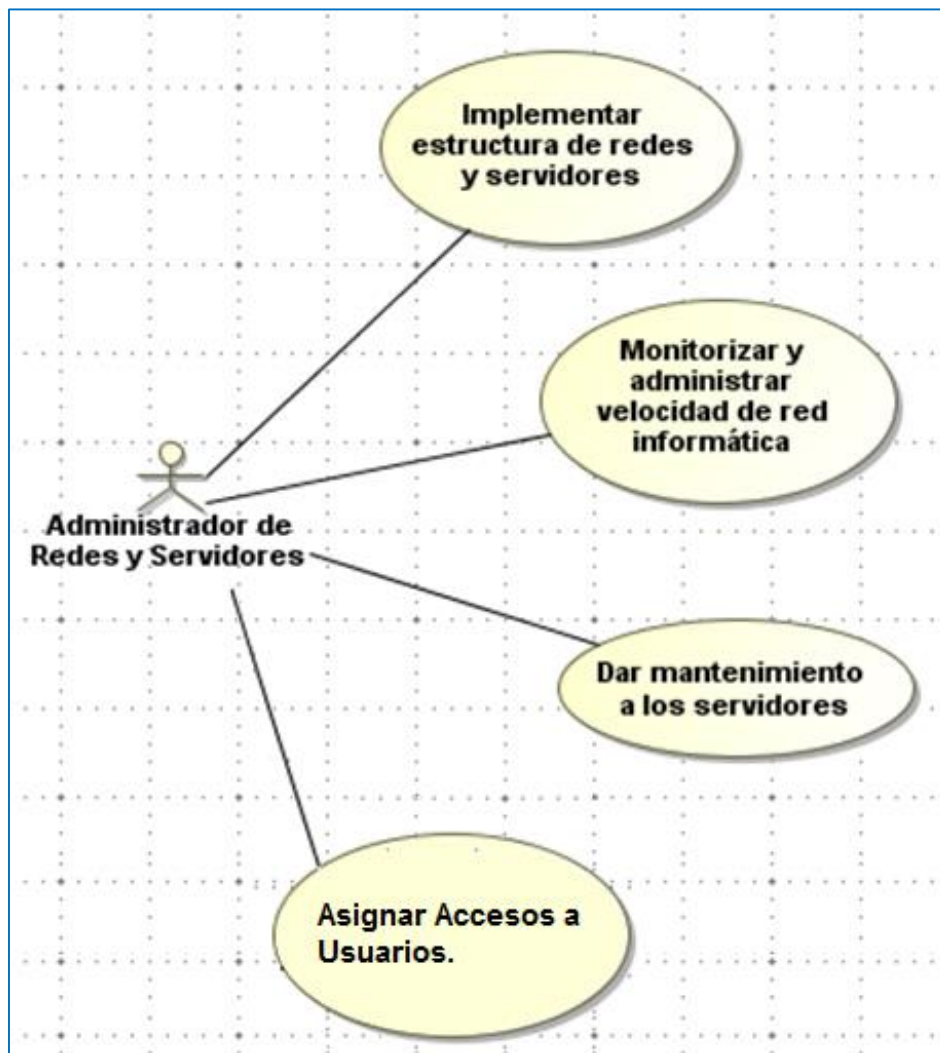


Ilustración 3.3: Diagrama de Casos de Uso - Administrador de Redes y Servidores
Fuente: Elaboración propia

3.2.2.1. Casos de Uso del Analista de Sistemas

El Analista de Sistemas es el especialista encargado del desarrollo de aplicaciones en lo que respecta a su diseño y obtención de los algoritmos, así como de analizar las posibles utilidades y modificaciones necesarias de los sistemas operativos para una mayor eficacia de un sistema informático.

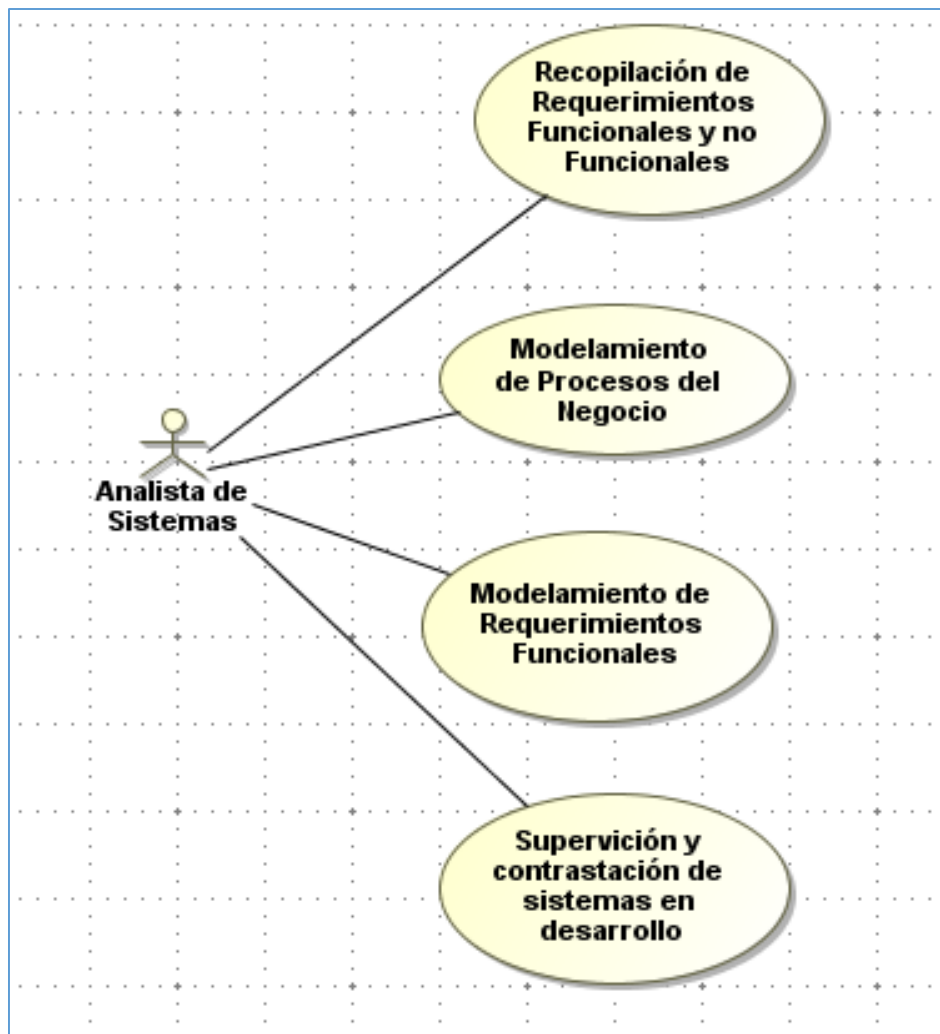


Ilustración 3.4: Diagrama de Casos de Uso - Analista de Sistemas
Fuente: Elaboración propia

3.2.2.1. Casos de Uso del Encargado de Mesa de Ayuda (Help Desk)

Es aquel trabajador encargado de prestar servicios con la posibilidad de gestionar y solucionar todas las posibles incidencias de manera integral, junto con la atención de requerimientos relacionados a las Tecnologías de la Información y la Comunicación (TIC).

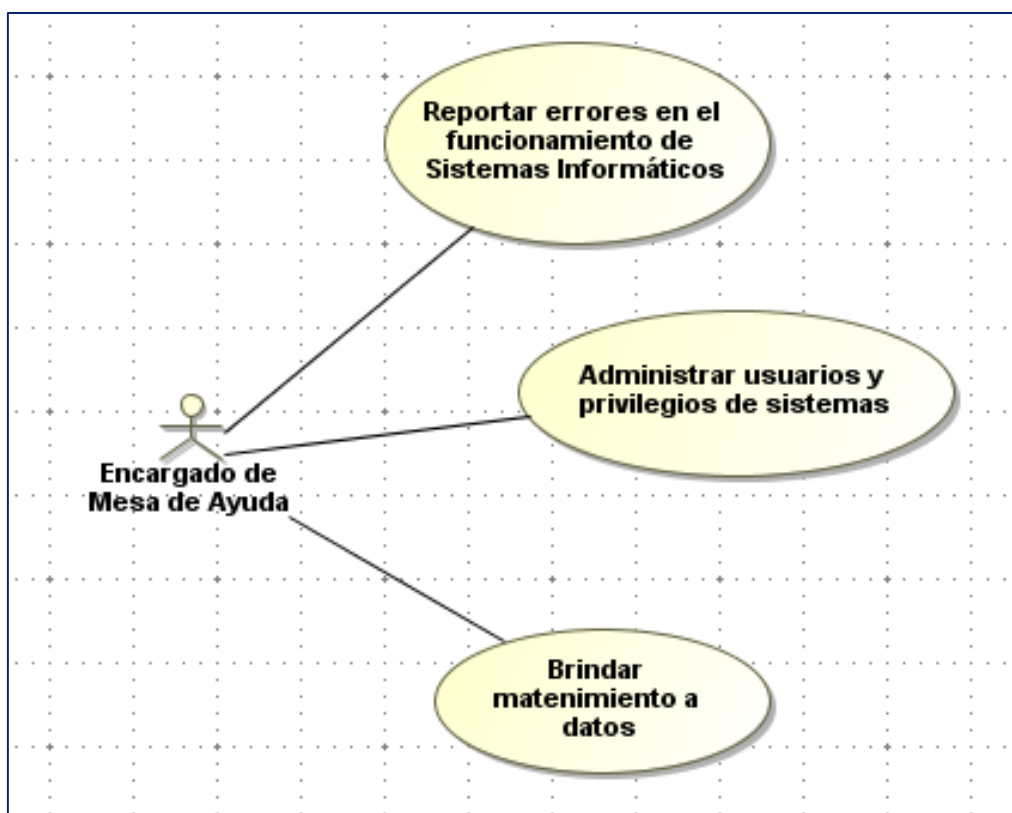


Ilustración 3.5: Diagrama de Casos de Uso - Encargado de Mesa de Ayuda
Fuente: Elaboración propia

3.2.2.1. Casos de Uso del Programador de Sistemas

Es aquel especialista encargado de escribir, depurar y mantener el código fuente de un programa informático, es decir, del conjunto de instrucciones que ejecuta el hardware de una computadora para realizar una tarea determinada. También conocido como “Desarrollador de software”, forman parte de un equipo de personas de distintas especialidades informáticas.

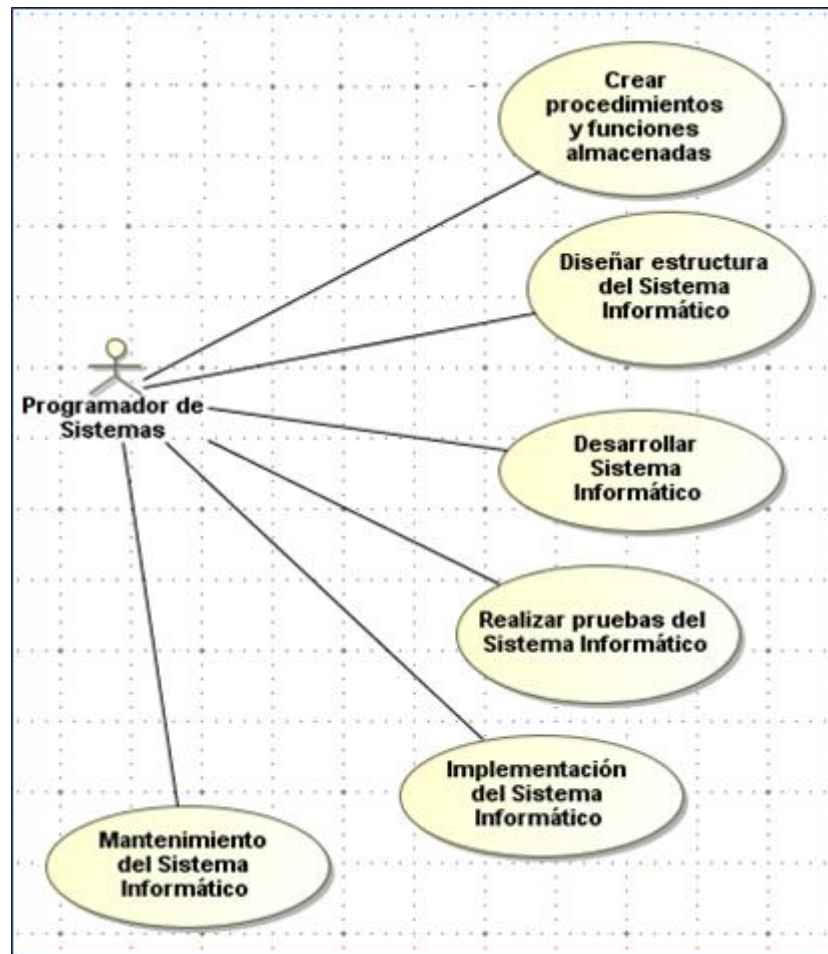


Ilustración 3.6: Diagrama de Casos de Uso - Programador de Sistemas
Fuente: Elaboración propia

3.2.3. Mapa de Procesos de la Oficina Universitaria de Informática y Sistemas mediante BPM

Como se podrá observar, en la *Ilustración 3.7*: Mapa de Procesos de la "Oficina Universitaria de Informática y Sistemas" se expone el “Mapa de Procesos” de la Oficina Universitaria de Informática y Sistemas, el cual ha sido desarrollado en base a la información recopilada del personal que trabaja en dicha oficina.

Los procesos que se llevan a cabo en esta oficina están divididos en “Estratégicos”, “Operativos” y de “Apoyo”. Los procesos “Estratégicos” son aquellos que engloban la planificación, la toma de decisiones y el despliegue de planes y políticas. Los procesos “Operativos” son aquellos que generan valor, los que tiene que ver de manera más directa con la misión, en este caso, de la oficina. Y los procesos de “Apoyo”, como bien lo describe su nombre, son procesos que brindan apoyo y dotan de los recursos necesarios a los procesos “Operativos”.

En este caso, tratándose el presente trabajo del “Desarrollo de un Framework”, tomaremos en cuenta solo el proceso de “Implementación de Proyectos tecnológicos e informáticos”, ya que en este se utilizaría el framework para desarrollar una aplicación, es por ello que, posterior al Mapa de Procesos, encontraremos un “Diagrama de Procesos” representando dicho proceso.

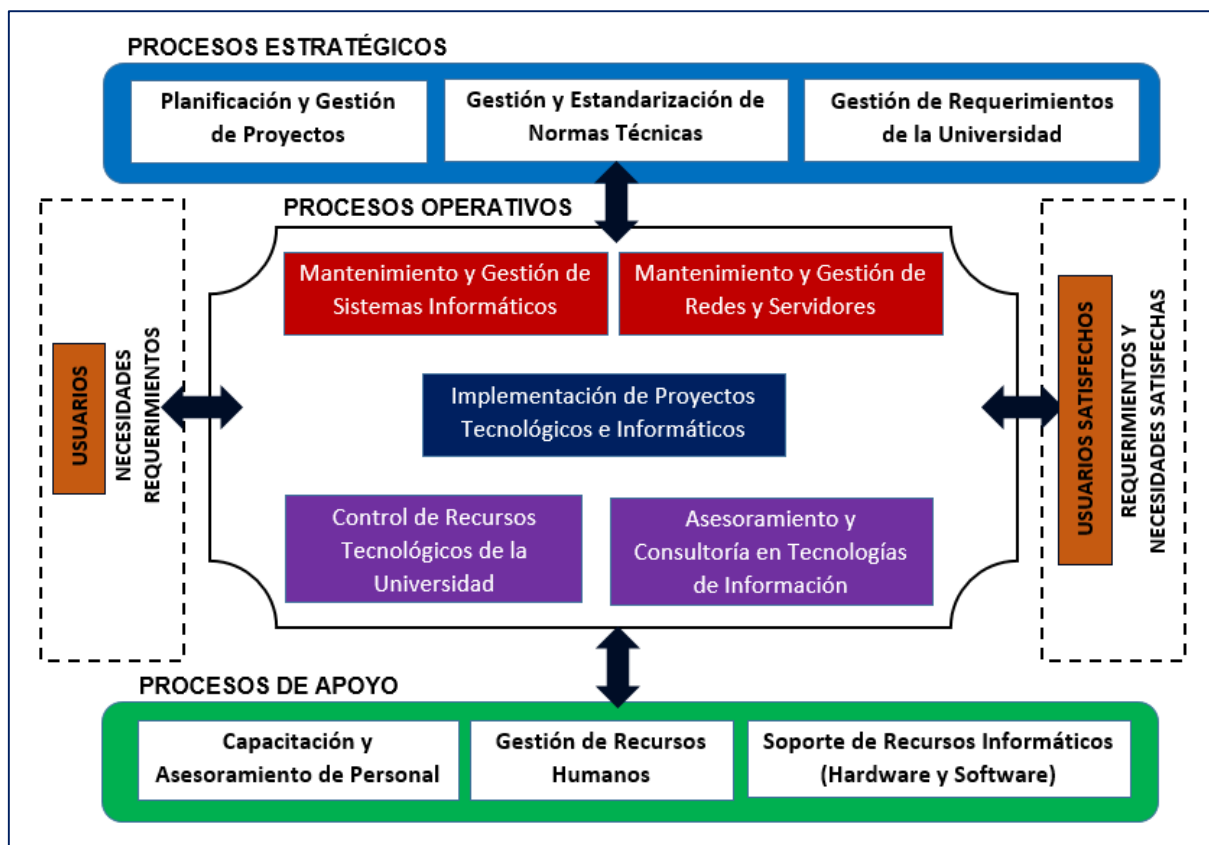


Ilustración 3.7: Mapa de Procesos de la "Oficina Universitaria de Informática y Sistemas"

Fuente: Elaboración propia

3.2.4. Diagrama de Procesos de la Oficina Universitaria de Informática y Sistemas mediante BPM

Proceso de “Implementación de Proyectos tecnológicos e informáticos”

En la *Ilustración 3.8*: Diagrama de Procesos de " Implementación de Proyectos tecnológicos e informáticos" ubicada a continuación, se presenta el diagrama del proceso descrito, en el cual se detalla la manera de cómo se llevan a cabo las actividades y cómo interactúan los tres elementos principales.

Si observamos el diagrama, nos daremos cuenta de que este proceso contiene “subprocesos”, pero tratándose esta investigación del desarrollo de un Framework para la creación de aplicaciones, sólo se detallará, posterior al primer diagrama, un diagrama del Subproceso “Desarrollo e Implementación del proyecto”.

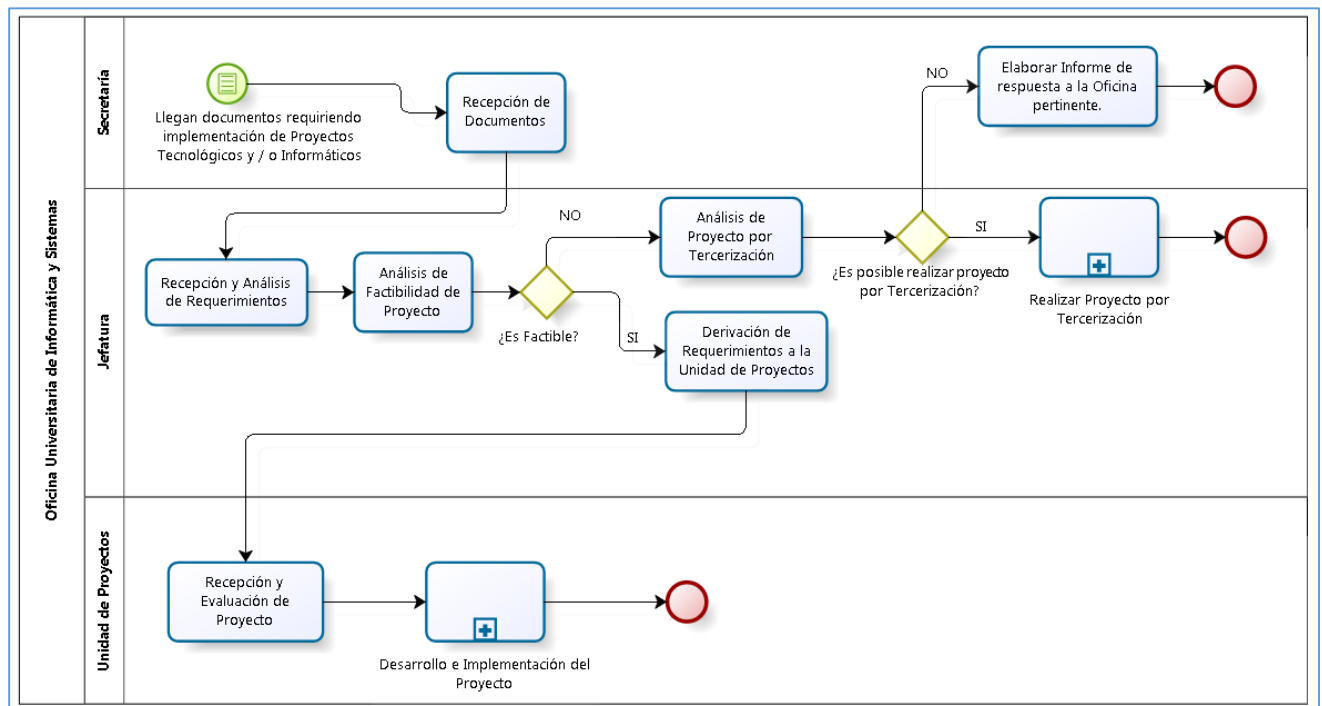


Ilustración 3.8: Diagrama de Procesos de " Implementación de Proyectos tecnológicos e informáticos"
Fuente: Elaboración propia

Como se puede observar en el diagrama de la *Ilustración 3.8*: Diagrama de Procesos de " Implementación de Proyectos tecnológicos e informáticos", para describir el proceso tendremos en cuenta 3 elementos de la Oficina Universitaria de Informática y Sistemas, los cuales son la “Unidad de Proyectos”, la “Jefatura” de la unidad y la “Secretaría”. El proceso comienza en la “Secretaría” cuando llegan los documentos requiriendo la implementación de nuevos proyectos, los documentos son recepcionados

y enviados a la Jefatura de la unidad, en donde se analiza la factibilidad del proyecto, evaluando si la Oficina Universitaria de Informática y Sistemas cuenta con las posibilidades de llevar a cabo dicho proyecto, considerando la disponibilidad del personal encargado del desarrollo y de los recursos necesarios. Si el proyecto no es factible, se evalúa su realización por “Tercerización”, si es posible realizarlo se lleva cabo por “terceros”, en caso contrario, se elabora un informe respondiendo a la Oficina pertinente las razones por las cuales no será posible llevar a cabo dicho proyecto. En caso de que dicho proyecto sea factible y puede ser realizado por la misma Oficina Universitaria de Informática y Sistemas, se derivan los requerimientos a la Unidad de Proyectos, en donde se evalúa (tiempo que tomará el proyecto, personal que se encargará del desarrollo, recursos a utilizar, etc.) y se comienza con el “Desarrollo e Implementación del Proyecto”.

Sub Proceso de “Desarrollo e Implementación del Proyecto”

En este diagrama se detalla (*Ilustración 3.9: Diagrama de Procesos de " Desarrollo e Implementación del Proyecto"*), de manera más precisa, la forma de cómo la “Unidad de Proyectos” de la “Oficina Universitaria de Informática y Sistemas” desarrolla e implementa los proyectos informáticos que tiene a su cargo.

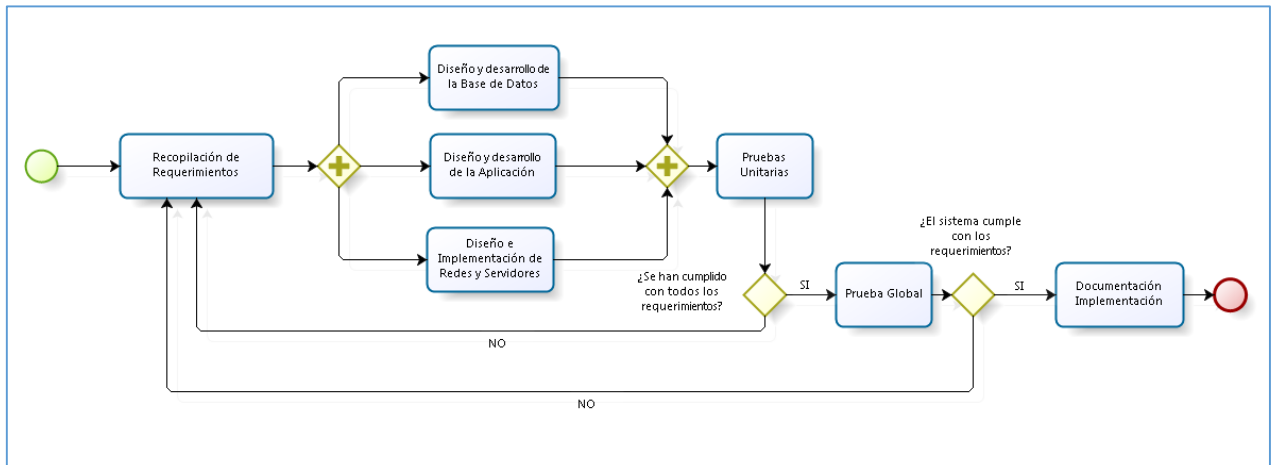


Ilustración 3.9: Diagrama de Procesos de " Desarrollo e Implementación del Proyecto"
Fuente: Elaboración propia

Como se puede observar en la *Ilustración 3.9: Diagrama de Procesos de " Desarrollo e Implementación del Proyecto"*, este subproceso comienza con la “Recopilación de Requerimientos”, acto seguido y paralelamente se llevan a cabo 3 tareas, las cuales son “Diseño y desarrollo de la Base de Datos”, “Diseño y desarrollo de la Aplicación” y “Diseño e Implementación de Redes y Servidores”, dependiendo siempre del proyecto y del alcance que este suponga. Luego de esto, se llevan a cabo “Pruebas unitarias”, y si los

resultados aún no son los esperados, se vuelve a la “Recopilación de Requerimientos”, generando así iteraciones hasta lograr resultados satisfactorios en las “Pruebas Unitarias”, posteriormente se realizan las “Pruebas Globales”, generando también las iteraciones necesarias hasta cumplir con todos los requerimientos, Posterior a ello se realiza la Documentación y la Implementación del proyecto, finalizando así este subproceso.

3.3. Diagrama de Casos de Uso del Sistema

En base a los requerimientos recopilados y diagramados, es posible determinar los “casos de uso del sistema”, en donde el principal actor es el “Programador”, quien será el principal usuario del framework. Por otro lado, también se incluye como actor al “Administrador de Base de Datos”, sobre todo en la administración de seguridad (usuarios y privilegios de acceso). A continuación se muestra la *Ilustración 3.10: "Diagrama de Casos de Uso del Sistema"* :

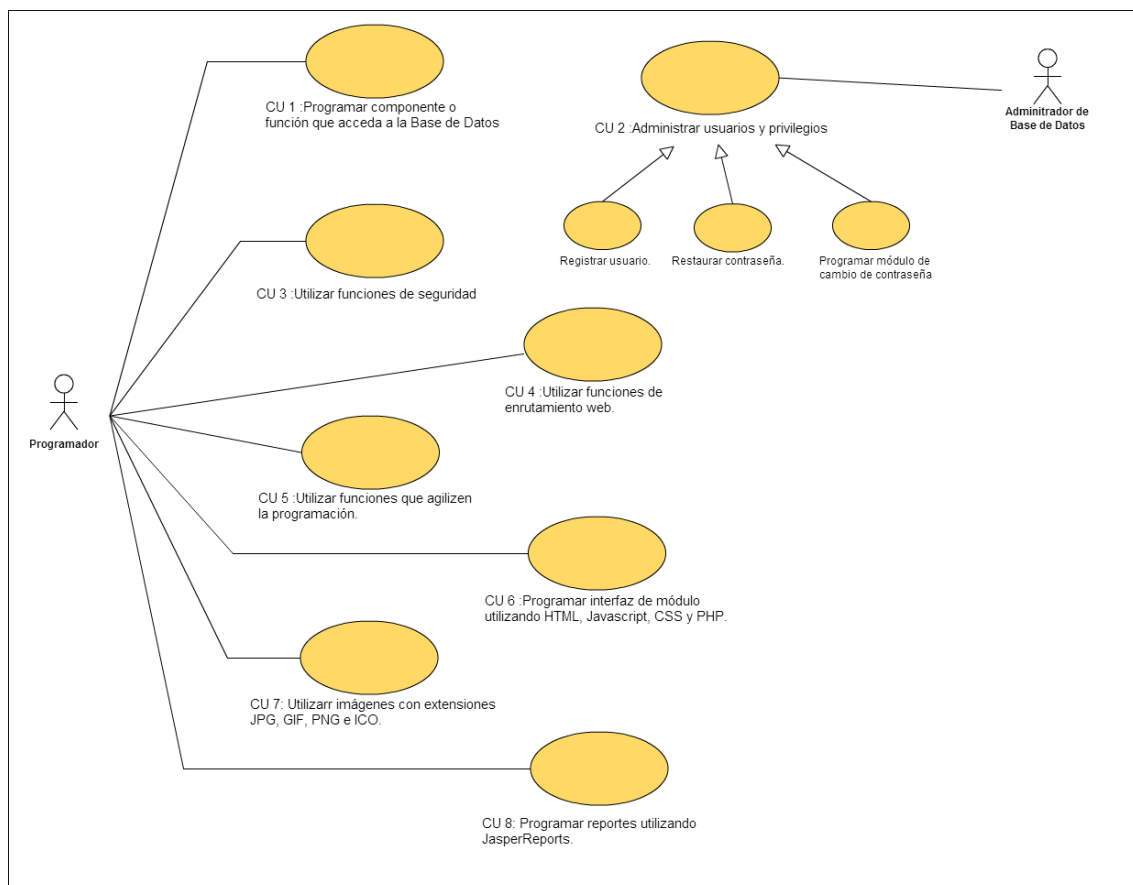


Ilustración 3.10: "Diagrama de Casos de Uso del Sistema"
Fuente: Elaboración propia

3.4. Fase “Pre-Game” de SCRUM

3.4.1. Backlog del Producto: Requisitos del cliente

Según se especifica en la fase de Pre - Game, el backlog del producto detalla los requerimientos iniciales del cliente, en este caso de la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”; dicha lista dará inicio a la planificación de cada sprint para el desarrollo y culminación del proyecto.

A continuación se presenta el Backlog del Producto, el cual fue generado a partir de las entrevistas realizadas a los trabajadores de la Unidad Organizativa mencionada.

Id	Estimación	Descripción	Criterio de Validación	Observación
1	50 días	Capa de Acceso a Datos	Es posible el acceso a datos almacenados en la Base de Datos desde la aplicación.	El Sistema Gestor de Base de Datos debe ser el Microsoft SQL Server, ya que es el indicado por la Oficina Universitaria de Informática y Sistemas
2	30 días	Base de Datos de Seguridad (MS SQLServer)	Es posible realizar un control de accesos según “Sistema Informático”, “Módulo de Sistema” y “Dependencia”, desde la Base de Datos.	Esta funcionalidad, por ahora, se reducirá únicamente a la Base de Datos. Más adelante será integrado con la parte de la aplicación del Framework.
3	80 días	Capa de Seguridad	Es posible acceder a la Base de Datos de Seguridad desde la aplicación, y esta obedece los privilegios de Acceso almacenados en la Base de Datos.	En esta funcionalidad se desarrollan también los módulos necesarios para la gestión de usuarios y privilegios desde la aplicación.
4	80 días	Controlador del Framework	Se puede realizar un enrutamiento seguro, utilizando URLs limpias, gestionando, estados y verbos HTTP.	----
5	30 días	Capa de Presentación	Es posible desarrollar un nuevo módulo de manera ágil, pudiendo	En esta capa se incluirá la parte básica de la página web (código

			utilizar la “Capa de Acceso a Datos” para conectarse al Microsoft SQLServer, la “Capa de Seguridad” para integrar sus funcionalidades y el “Controlador del Framework”.	HTML), hojas de estilos (CSS) y scripts (Javascript).
6	20 días	Arquitectura Final del Framework	La arquitectura final debe ser una estructura organizada y entendible, que permita el desarrollo rápido e intuitivo.	---

Cuadro 3.4: Backlog del Producto

Fuente: Elaboración propia

CAPÍTULO IV: DISEÑO DEL FRAMEWORK

4.1. Fase “Game” de SCRUM

En la fase “Game o Ejecución” de SCRUM se crearon los “Backlogs del sprint “, en los cuales se establecieron los objetivos o “Sprint goals” a realizarse en cada Sprint; además de la duración que tuvo cada uno. De la misma forma, se incluye el Burn down chart (Gráficos de trabajo pendiente), herramienta que permite representar el avance de las tareas de cada sprint, comparándolo con un estimado, logrando así que los implicados en el proyecto podamos medir nuestra velocidad de desarrollo y controlar sus avances. Posterior a ello, por cada uno de los sprints, se expone el resumen de los requerimientos en base al análisis realizado, para luego mostrar detalles del diseño y desarrollo de cada uno de los componentes.

4.1.1. Capa de Acceso a Datos

En el siguiente “Backlog del Sprint” (*Cuadro 4.1: Backlog del Sprint 1 - “Capa de Acceso a Datos”*.) se presenta el primer sprint, cuyo objetivo o “sprint goal” fue desarrollar la “Capa de Acceso a Datos”, la cual sirve para permitir la conexión entre la aplicación y la Base de Datos.

<i>Objetivo: Desarrollar la "Capa de Acceso a Datos"</i>		<i>N° Sprint: 1</i>	<i>Día de Inicio: 17 de Febrero del 2014</i>		<i>Duración: 21 días</i>
Id	Tarea	Tipo	Días	Estado	Responsable
1	Realizar la planificación de la iteración	Planificación	2	Terminado	Emily / José
2	Analizar los requerimientos	Análisis	2	Terminado	Emily / José
3	Crear "Base de Datos" básica en SQL Server para probar conexión.	Desarrollo	5	Terminado	Emily
4	Descargar, instalar y configurar controlador PHP – MSSQL.	Desarrollo	5	Terminado	José
5	Programar funciones en PHP para utilizar PHP-MSSQL en el framework.	Desarrollo	7	Terminado	José
6	Probar y depurar conexión de PHP con Base de Datos en Microsoft SQL Server.	Prueba	7	Terminado	Emily
7	Documentar	Documentación	5	Terminado	Emily / José

Cuadro 4.1: Backlog del Sprint 1 - "Capa de Acceso a Datos".
Fuente: Elaboración propia

Acto seguido, en la *Ilustración 4.1: Burn down chart del Sprint 1 - "Capa de Acceso a Datos"* se observa el Burn Down chart de este sprint, el cual demuestra que la presente iteración fue completada cumpliendo la cantidad de días estimados.

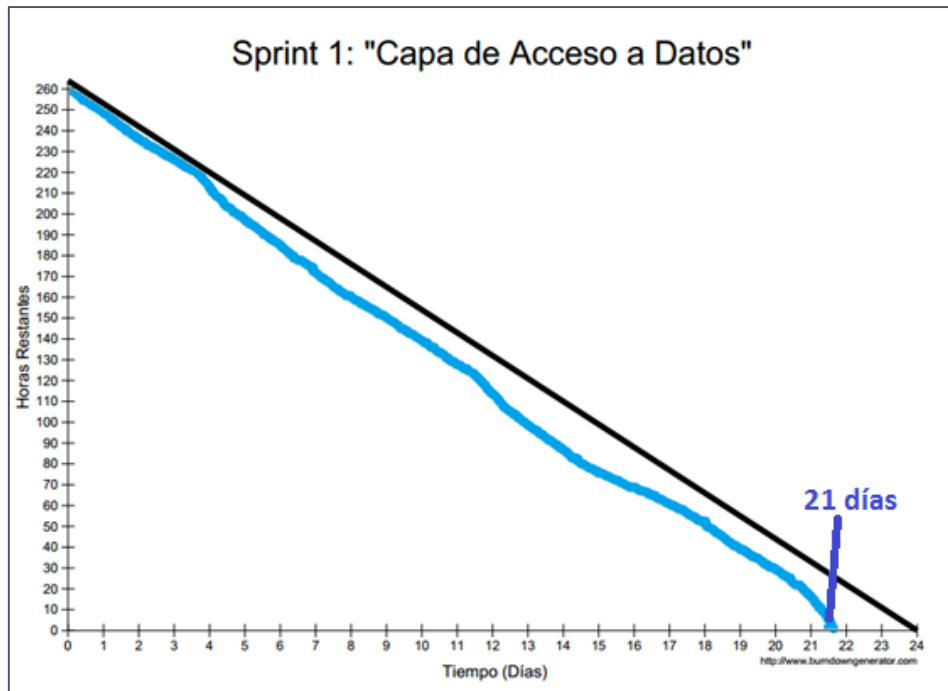


Ilustración 4.1: Burn down chart del Sprint 1 - "Capa de Acceso a Datos"
Fuente: Elaboración propia

4.1.1.1. Resumen de los requerimientos de la “Capa de Acceso a Datos”

A continuación, en la *Ilustración 4.2: Casos de Uso – “Capa de Acceso a Datos”* podemos observar el “Caso de Uso” correspondiente a este componente:

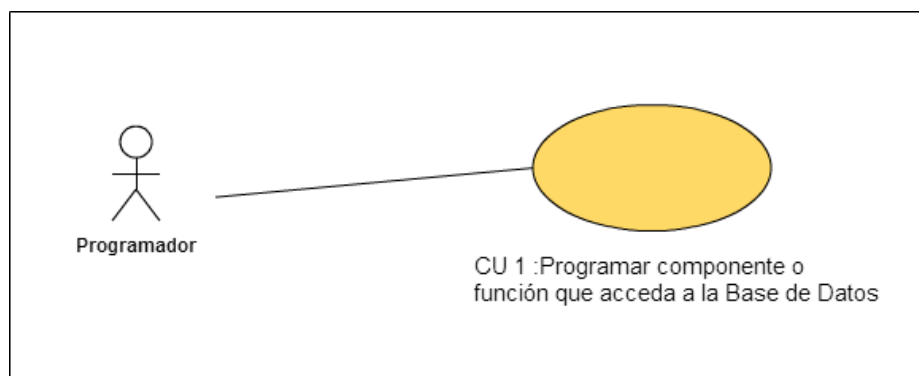


Ilustración 4.2: Casos de Uso – “Capa de Acceso a Datos”
Fuente: Elaboración propia

En base al análisis, se puede resumir que los requerimientos de la “Capa de Acceso a Datos” son:

- Utilización y acceso rápido. Si bien este es un requerimiento general del framework, al hablar de esta capa en específico, es

importante que el acceso a los datos almacenados en la “Base de Datos” sea rápido e intuitivo para el programador.

- Seguro. Si bien, como se expone en el requerimiento anterior, la capa de acceso a los datos debe ser rápida e intuitiva, esta también debe ser “segura”, evitando así el acceso no autorizado y malintencionado al servidor de la base de datos.
- El Sistema Gestor de la Base de Datos que debe soportar el framework de manera necesaria es el “Microsoft SQL Server”.

4.1.1.2. Diseño y desarrollo de la “Capa de Acceso a Datos”

En base a los requerimientos se determinó que la “Capa de Acceso a Datos” debe tener una estructura que permita acceder a los datos de manera rápida y eficiente, pero que la información de acceso (cadena de conexión, usuario, contraseña, etc.) deberá estar en un directorio aparte y privado, al cual sólo tendrá acceso el personal encargado, mientras que los programadores podrán desarrollar el proyecto sin conocer ni la cadena de conexión, ni datos relacionados.

4.1.1.2.1. Privado

Este es un directorio que guarda información del acceso a la “Base de Datos”. Si bien, el directorio “Proyecto” debería colocarse en el directorio raíz de aplicaciones del servidor web (/www/Proyecto), “Privado” debería colocarse fuera de este directorio, quedando en paralelo con el directorio raíz.

- Subdirectorio “Conexion_proyecto”:

Este directorio contiene un archivo llamado “Conexion.php”, el cual contiene todos los datos que permiten la conexión a la Base de Datos.

```

<?php
//CLASE PARA CONEXION CON LA BASE DE DATOS
//-----
class Conexion{
    var $con;
    function Conexion()
    {
        $connection['server']="192.168.248.128:1433";//host
        $connection['user'] ="desarrollo";// usuario
        $connection['pass'] ="xxxxxxx";//password

        // crea la conexión pasándole el servidor , usuario y clave
        $conect= mssql_connect($connection['server'],$connection['user'],$connection['pass']);
        if ($conect) // si la conexion fue exitosa , selecciona la base
        {
            mssql_select_db(BD);
            $this->con=$conect;
        }
    }

    function getConexion(){// devuelve la conexion
        return $this->con;
    }

    function Close(){ // cierra la conexion
        mssql_close($this->con);
    }
}
?>

```

Ilustración 4.3: Código fuente de "Conexion.php"
Fuente: Elaboración propia

- Archivo “variables.php”:

Este archivo contiene variables privadas de seguridad que deben permanecer seguras. Estas son creadas con la función “define()” de PHP, la cual define constantes en tiempo de ejecución.

```

<?php
define("captcha_private_key",'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
define("salt", "#o5o34+-o/g+)d1)2");//OJO: No cambiar cuando el sistema ya ha sido puesto en funcionamiento

define("servidor","192.168.248.128");
define("usuario","desarrollo");
define("pass","xxxxxxx");
?>

```

Ilustración 4.4: Código fuente de "variables.php"
Fuente: Elaboración propia

La variable “captcha_private_key” es la llave privada que permite el acceso al Servicio Web de Google “Re Captcha”, el cual deberá ser generado para cada proyecto, según el dominio sobre el cuál será publicada la aplicación, para hacerlo basta acceder a la página de este servicio. La variable “salt” es un código salado que permite la encriptación de contraseñas. Los tres siguientes, “servidor”, “usuario” y “pass” son datos que permiten el acceso a la Base de Datos.

4.1.1.2.2. Subdirectorío “Capa_AccesoDatos”

Esta es la capa que se encarga de acceder a la Base de Datos, realizar consultas y ejecutar operaciones, mediante la estructura establecida por el framework. Para lograrlo, la Capa de Acceso a Datos posee 2 archivos:

- **“accesobd.php”**

Contiene las funciones que permiten la realización de consultas y ejecución de operaciones. En esta primera versión del framework se tiene una sola función llamada “ejecutar_pa”, la cual se encarga, como dice su nombre, de ejecutar los procedimientos almacenados.

A continuación se muestra la codificación del archivo:

```

<?php
public static function ejecutar_pa($pa,$formato="",$xml="",$sal=false)
{
    $obj=new sQuery();
    switch($formato)
    {
        case 'array':
            $result=$obj->array_pa_datos($pa,$xml,$sal);
            return ($sal)?$result:$obj->fetchAll();
        case 'json':
            $obj->json_pa_datos($pa,$xml,$sal);
            break;
        case 'xml':
            $result=$obj->array_pa_datos($pa,$xml,$sal);
            return ($sal)?array2xml($result):array2xml($obj->fetchAll());
        case "":
            $result=$obj->array_pa_datos($pa,$xml,$sal);
            return ($sal)?$result["rpta"]:true;
    }
}
?>

```

Ilustración 4.5: Código fuente de "acesobd.php"

Los parámetros que recibe la función son:

\$pa: Nombre del procedimiento;
 \$formato: Formato de la respuesta;
 \$xml: Variable XML que envía datos al procedimiento;
 \$sal: Variable booleana para indicar si retorna respuesta "rpta";

Como se puede observar, esta función puede retornar datos de respuesta en el formato que se requiera.

- sQuery.php

Este archivo contiene todas las funciones que se encargan de realizar las consultas y ejecutar las operaciones en la Base de Datos, utilizando el "PHP-MSSQL".

4.1.2. Base de Datos de Seguridad (Microsoft SQL Server)

A continuación, en el *Cuadro 4.2: Backlog del Sprint 2 - "Base de Datos de Seguridad (Microsoft SQL Server)"* se puede observar el segundo

sprint, cuyo objetivo fue el desarrollo de la “Base de Datos de Seguridad” en Microsoft SQL Server, la cual permite el control de accesos y privilegios del framework y de las aplicaciones que sean desarrolladas mediante este.

Objetivo: Desarrollar la “Base de Datos de Seguridad”		N° Sprint: 2	Día de Inicio: 12 de Marzo del 2014		Duración: 42 días
Id	Tarea	Tipo	Días	Estado	Responsable
1	Realizar la planificación de la iteración.	Planificación	2	Terminado	Emily / José
2	Analizar los requerimientos.	Análisis	5	Terminado	Emily / José
3	Diseñar la estructura y las técnicas de seguridad.	Análisis	7	Terminado	Emily
4	Diseñar la Base de Datos y los procedimientos almacenados.	Análisis	15	Terminado	Emily
5	Programar la Base de Datos y los procedimientos almacenados en Microsoft SQL Server.	Desarrollo	20	Terminado	José
6	Probar y Depurar el funcionamiento de la Base de Datos.	Prueba	8	Terminado	Emily / José
7	Documentar	Documentación	5	Terminado	Emily / José

Cuadro 4.2: Backlog del Sprint 2 - "Base de Datos de Seguridad (Microsoft SQL Server)"

Fuente: Elaboración propia

Luego, en la *Ilustración 4.6: Burn down chart del Sprint 2: "Base de Datos de Seguridad"* se puede observar el Burn down chart del Sprint 2, el mismo que demuestra que se incumplió levemente el estimado de días planeados, ya que se pensaron en 34 días, siendo finalmente 42 los que se utilizaron para llevar a cabo esta iteración.

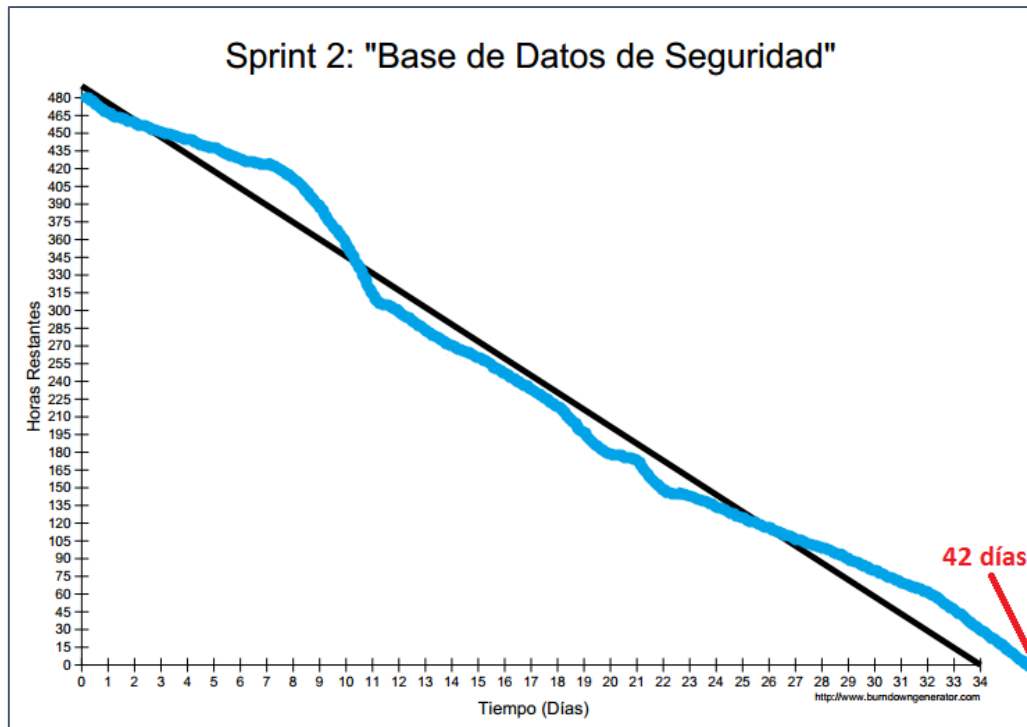


Ilustración 4.6: Burn down chart del Sprint 2: "Base de Datos de Seguridad"
Fuente: Elaboración propia

4.1.2.1. Resumen de los requerimientos de la "Base de Datos de Seguridad"

En la *Ilustración 4.7: Casos de Uso - "Base de Datos de Seguridad"* mostrada a continuación, es posible observar el "Caso de Uso" correspondiente a este componente, el cual contiene otro caso de uso como herencia:

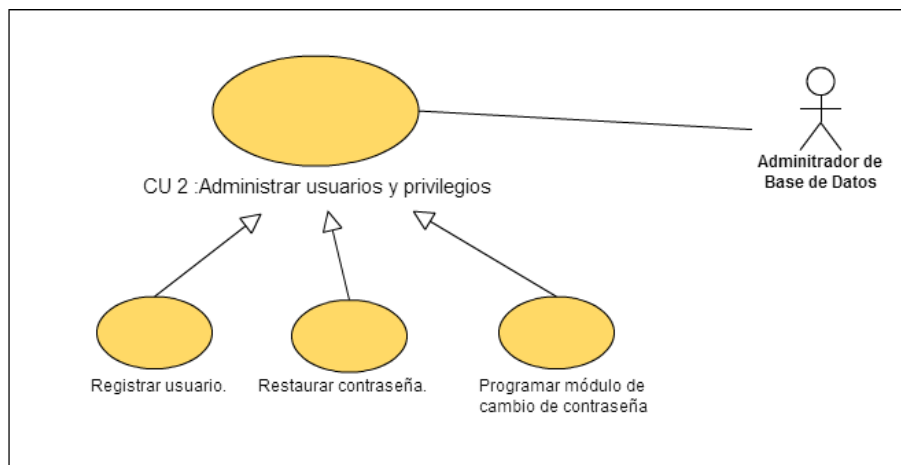


Ilustración 4.7: Casos de Uso - "Base de Datos de Seguridad"
Fuente: Elaboración propia

La Base de Datos de seguridad es aquella que estará configurada de manera predeterminada en el framework, y que según el análisis, el resumen de sus requerimientos son:

- La Base de Datos de seguridad debe estar orientada a la lógica de la Universidad Peruana Los Andes, donde se manejan diversos “Sistemas Informáticos” en las diferentes dependencias o facultades, y son accedidos por administrativos, docentes y estudiantes. Cada uno de ellos debe tener diferentes tipos de accesos.
- Con respecto a la gestión de usuarios y privilegios, la base de datos debe ser única para todos los sistemas.
- Es importante guardar datos personales de los usuarios.
- El Sistema Gestor de Base de Datos debe ser el Microsoft SQL Server.

4.1.2.2. Diseño y desarrollo de la “Capa de Acceso a Datos”

A continuación se puede observar la *Ilustración 4.8: Diagrama físico de la Base de Datos* diseñado para el framework. Posteriormente observaremos la documentación de cada una de las tablas que lo componen.

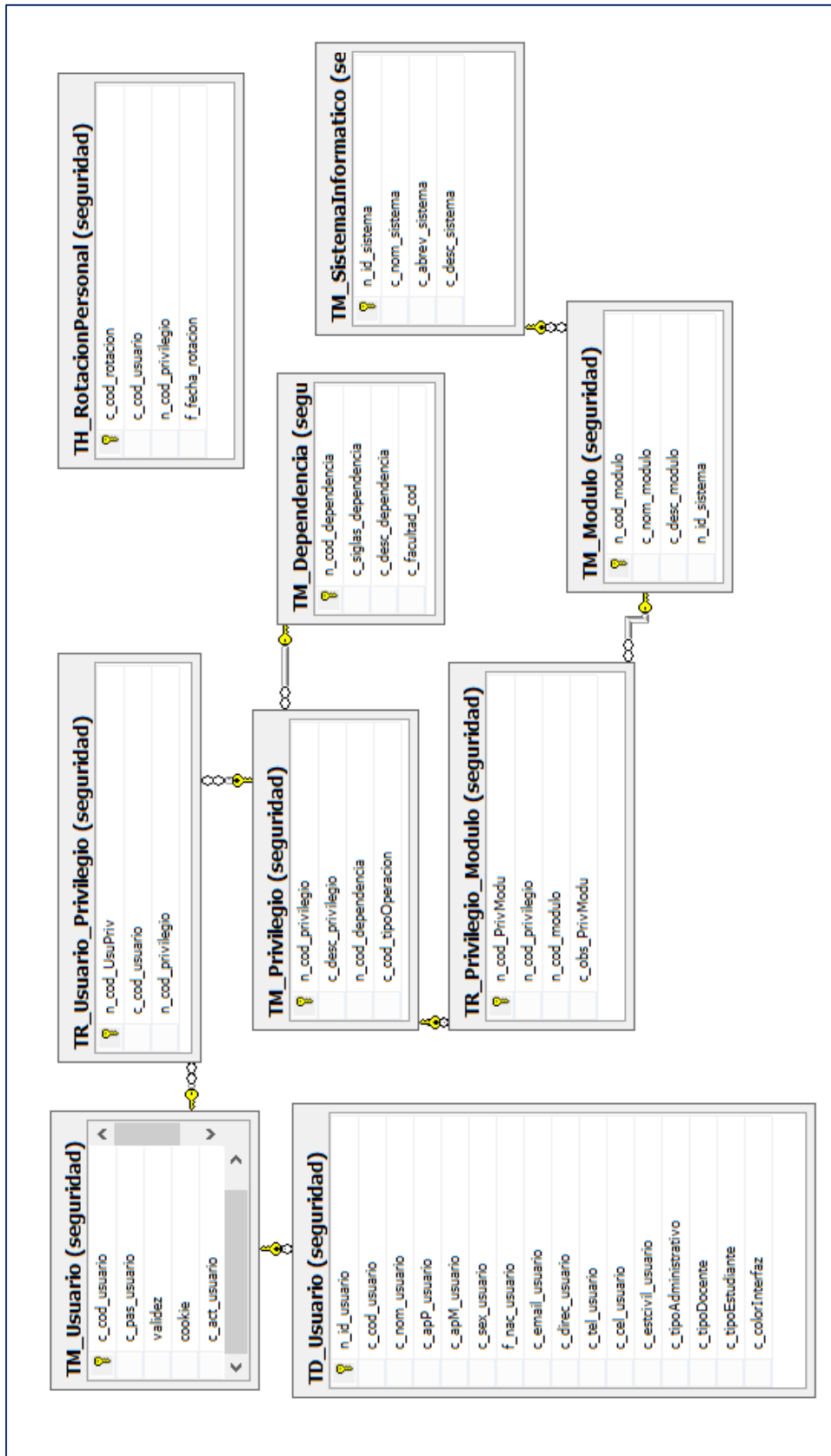


Ilustración 4.8: Diagrama físico de la Base de Datos
Fuente: Elaboración propia

4.1.2.2.1. Diccionario de Datos

A continuación, en el *Cuadro 4.3: Diccionario de Datos de la "Base de Datos de Seguridad"* podemos observar la documentación de cada una de las tablas que componen esta base de datos.

TM_Usuario			
Campo	Tipo de Dato	Permite Nulos	Descripción
c_cod_usuario	varchar	NO	Código del Usuario (DNI) e Identificador único de la Tabla Maestra de Usuarios.
c_pas_usuario	Varbinar y	NO	Contraseña del usuario.
validez	datetime	SI	Tiempo de Validez del Cookie Lógico.
cookie	varchar	SI	Código del Cookie Lógico.
c_act_usuario	Bit	SI	Dato booleano que determina si el usuario está activo o no.
TD_Usuario			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_id_usuario	Int	NO	Identificador autogenerado de la Tabla Detalle de Usuario.
c_cod_usuario	varchar	NO	Código de Identificación del Usuario (DNI).
c_nom_usuario	varchar	NO	Nombre del Usuario.
c_apP_usuario	varchar	NO	Apellido Paterno del Usuario.
c_apM_usuario	varchar	NO	Apellido Materno del Usuario.
c_sex_usuario	Char	NO	Sexo o Género del Usuario.
f_nac_usuario	datetime	SI	Fecha de nacimiento del Usuario.
c_email_usuario	varchar	SI	Correo Electrónico del Usuario.
c_direc_usuario	varchar	SI	Dirección Domiciliaria del Usuario.
c_tel_usuario	varchar	SI	Teléfono Fijo del Usuario.
c_cel_usuario	varchar	SI	Teléfono móvil o celular del usuario.
c_estcivil_usuario	Char	SI	Estado civil del usuario.
c_tipoAdministrativo	Bit	SI	Dato booleano que registra si el usuario es Administrativo o no.
c_tipoDocente	Bit	SI	Dato booleano que registra si el usuario es Docente o no.
c_tipoEstudiante	Bit	SI	Dato booleano que registra si el usuario es Estudiante o no.
TM_Privilegio			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_cod_privilegio	Int	NO	Identificador autogenerado de la Tabla Maestra de Privilegios.
c_desc_privilegio	nvarchar	SI	Nombre del Privilegio.
c_desc_privilegio	varchar	SI	Nombre del Privilegio.

n_cod_dependencia	Int	SI	Código de la Dependencia a la que pertenece el privilegio.
TM_Dependencia			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_cod_dependencia	Int	NO	Identificador auto generado de la Tabla Maestra de Dependencias.
c_siglas_dependencia	nvarchar	SI	Siglas o abreviación de la Dependencia.
c_desc_dependencia	nvarchar	SI	Nombre de la Dependencia.
c_facultad_cod	nchar	SI	Código de Facultad que maneja la Universidad.
TM_Modulo			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_cod_modulo	Int	NO	Identificador autogenerado de la Tabla Maestra de Módulos.
c_nom_modulo	nvarchar	NO	Nombre del Módulo
c_desc_modulo	Text	SI	Breve descripción del módulo.
n_id_sistema	Int	SI	Código del Sistema Informático al cual pertenece el módulo.
TM_SistemaInformatico			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_id_sistema	Int	NO	Identificador autogenerado de la Tabla Maestra de Sistemas Informáticos
c_nom_sistema	nvarchar	NO	Nombre del Sistema Informático
c_abrev_sistema	nvarchar	SI	Abreviatura del Sistema Informático
c_desc_sistema	Text	SI	Descripción del Sistema Informático
TR_Usuario_Privilegio			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_cod_UsuPriv	Int	NO	Identificador autogenerado de la Tabla Relación de Usuarios y Privilegios.
c_cod_usuario	varchar	NO	Código del usuario a relacionar.
n_cod_privilegio	Int	NO	Código del privilegio a relacionar.
TR_Privilegio_Modulo			
Campo	Tipo de Dato	Permite Nulos	Descripción
n_cod_PrivModu	Int	NO	Identificador autogenerado de la Tabla Relación de Privilegios y Módulos.
n_cod_privilegio	Int	SI	Código de Privilegio a relacionar.
n_cod_modulo	Int	SI	Código de Módulo a relacionar.
c_obs_PrivModu	Text	SI	Comentarios u observaciones.

Cuadro 4.3: Diccionario de Datos de la "Base de Datos de Seguridad"
Fuente: Elaboración propia

4.1.3. Capa de Seguridad

En el Cuadro 4.4: Backlog del Sprint 3 - "Capa de Seguridad", se puede observar el Tercer sprint, cuyo objetivo fue el desarrollo de la "Capa de Seguridad", la cual incluye funciones y técnicas que permiten la protección de datos.

<i>Objetivo: Desarrollar la "Capa de Seguridad"</i>		<i>N° Sprint: 3</i>	<i>Día de Inicio: 25 de Abril del 2014</i>		<i>Duración: 42 días</i>
Id	Tarea	Tipo	Días	Estado	Responsable
1	Realizar la planificación de la iteración.	Planificación	2	Terminado	Emily / José
2	Analizar los requerimientos.	Análisis	2	Terminado	Emily / José
3	Diseñar la estructura de la Capa de Seguridad.	Análisis	5	Terminado	Emily
4	Programar funciones de seguridad.	Desarrollo	15	Terminado	José
5	Implementar técnica de "Protección de Cadena de Conexión"	Desarrollo	10	Terminado	Emily
6	Programar la función: - Registrar usuarios	Desarrollo	5	Terminado	Emily
7	Programar las funciones: - Cambiar contraseña - Restaurar contraseña	Desarrollo	5	Terminado	José
8	Probar y Depurar la "Capa de Seguridad"	Prueba	10	Terminado	Emily / José
9	Documentar	Documentación	8	Terminado	Emily / José

Cuadro 4.4: Backlog del Sprint 3 - "Capa de Seguridad"

Fuente: Elaboración propia

Como se puede observar en la *Ilustración 4.9: Burn down chart del Sprint 3 - "Capa de Seguridad"*, no se pudo cumplir el estimado de 39 días que se propuso para la presente iteración.

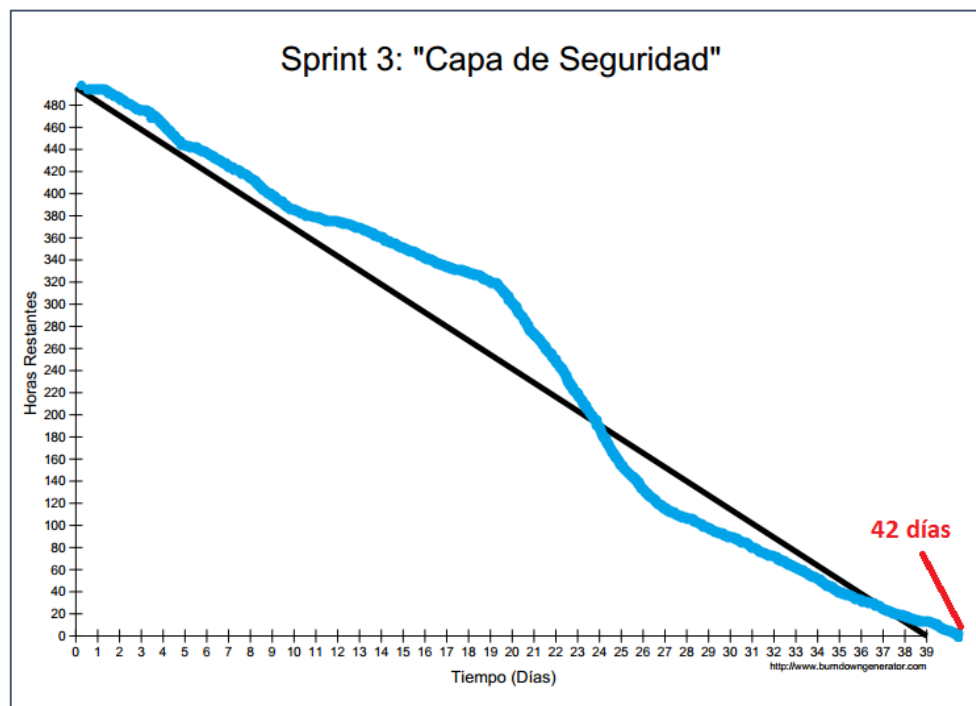


Ilustración 4.9: Burn down chart del Sprint 3 - "Capa de Seguridad"
Fuente: Elaboración propia

4.1.3.1. Resumen de los requerimientos de la "Capa de Seguridad"

A continuación podemos observar en la *Ilustración 4.10: Caso de Uso - "Capa de Seguridad"*, el "Caso de Uso" correspondiente a este componente:

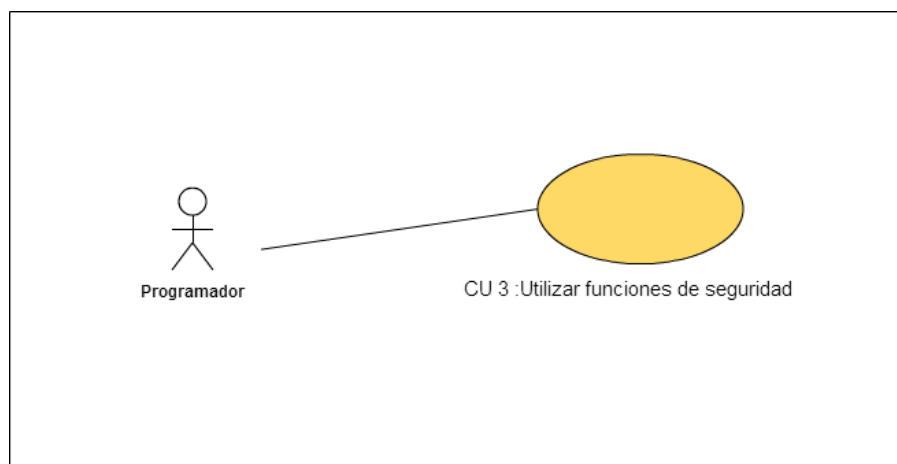


Ilustración 4.10: Caso de Uso - "Capa de Seguridad"
Fuente: Elaboración propia

Entre los requerimientos recopilados de la “Oficina Universitaria de Informática y Sistemas”, no se especifica la necesidad de una “Capa de Seguridad”, pero si existen ciertos requerimientos al respecto. Por lo tanto, se planteó como solución desarrollar un componente orientado a cumplir con este requerimiento. La “Capa de Seguridad” protege los datos basándose principalmente en el “Top 10 de Amenazas de Seguridad” de OWASP. Por tanto, los requerimientos están basados en ello.

4.1.3.2. Diseño y desarrollo de la “Capa de Seguridad”

Esta es la “Capa de Seguridad” del framework, la cual contiene todos los componentes de seguridad que vienen implementados de manera predeterminada en el framework, y los que el programador vaya a implementar como parte de su desarrollo. A continuación se describen los elementos predeterminados de framework:

- **seguridadPHP.php**

En este archivo se incluyen todas las funciones de seguridad utilizadas por el framework, las cuales se encuentran programadas en lenguaje PHP, y por ende son ejecutadas en el Servidor.

- **seguridadJS.js**

En este archivo se incluyen todas las funciones de seguridad utilizadas por el framework, las cuales se encuentran programadas en JAVASCRIPT, y por ende son ejecutadas por el lado del cliente, es decir en el navegador del usuario.

- **variablesInt_seguridad.php**

En este archivo se incluyen todas aquellas variables relacionadas a la seguridad del framework que no pueden estar incluidas en la parte externa del proyecto (Directorio “Privado”), sino que deben estar ubicadas en la parte interna de la arquitectura y que deben poder ser modificadas por el mismo programador.

4.1.3.2.1. Sistema de Protección de Accesos (Uso alternativo)

BunnyPHP es un framework que incluye una estructura de Base de Datos en SQLServer, la misma que permite incluir en el framework un Sistema de Protección de Accesos. Esto es posible gracias a que BunnyPHP ha sido desarrollado orientándose a las necesidades de la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, y por ende es posible tener un modelo de “lógica del negocio”.

El uso de este “Sistema de Protección de Accesos” es alternativo, ya que, al estar BunnyPHP apto para ser utilizado en cualquier otra aplicación no relacionada con la Universidad Peruana Los Andes, es posible que la arquitectura de Base de Datos que se incluye no sea válida para un negocio con otra lógica.

A continuación se diagrama la técnica utilizada para encriptar y proteger las contraseñas, de esta manera permitir el “registro” de nuevos usuarios, y la comprobación segura de la contraseña al “validar” el acceso de un usuario al sistema.

Sistema de Encriptación y protección de contraseña

A continuación, en la *Ilustración 4.11: Diagrama de "Sistema de Encriptación - Registro de Nuevo Usuario"* se muestra la forma de cómo se aplica la técnica de encriptación de contraseñas de BunnyPHP al registrar un nuevo usuario. Luego de que el usuario ingrese la nueva contraseña a la aplicación, esta cadena de texto es encriptada mediante el algoritmo MD5, luego, al resultado se le agrega un “código salado” para finalmente volver a encriptar el resultado mediante el algoritmo SHA1. Luego de haber llevado a cabo todo este proceso en la aplicación, el resultado es enviado al “Servidor de Base de Datos”, donde es encriptado nuevamente mediante un procedimiento utilizando el algoritmo “SHA” de Microsoft SQL Server, el cual devuelve el valor en “binario”. Finalmente se almacena este valor binario en la Base de Datos.

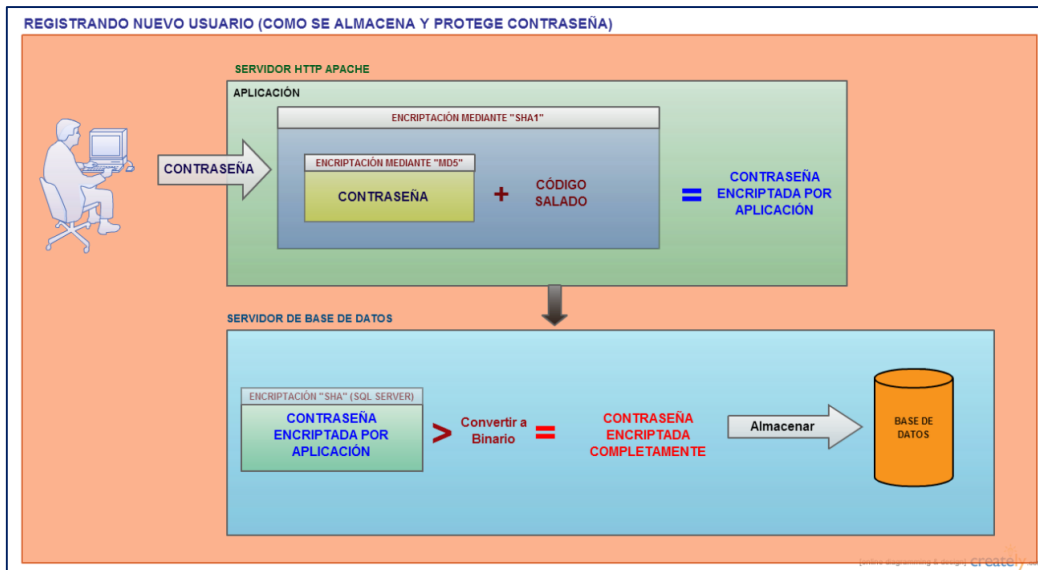


Ilustración 4.11: Diagrama de "Sistema de Encriptación - Registro de Nuevo Usuario"
Fuente: Elaboración propia

Como se observará a continuación en el *Ilustración 4.12: Diagrama de "Sistema de Encriptación - Validación de Acceso de Usuarios"* se muestra la forma de cómo se valida el acceso de un usuario ya registrado al sistema, comprobando la contraseña que ingresa en el formulario de logueo con la contraseña encriptada y almacenada en la base de datos. El proceso por el que pasa la contraseña es el mismo, hasta obtener el valor final encriptado, es en este momento en el que se compara con el valor almacenado en la base de datos, y se valida si la contraseña ingresada ha sido correcta o no.

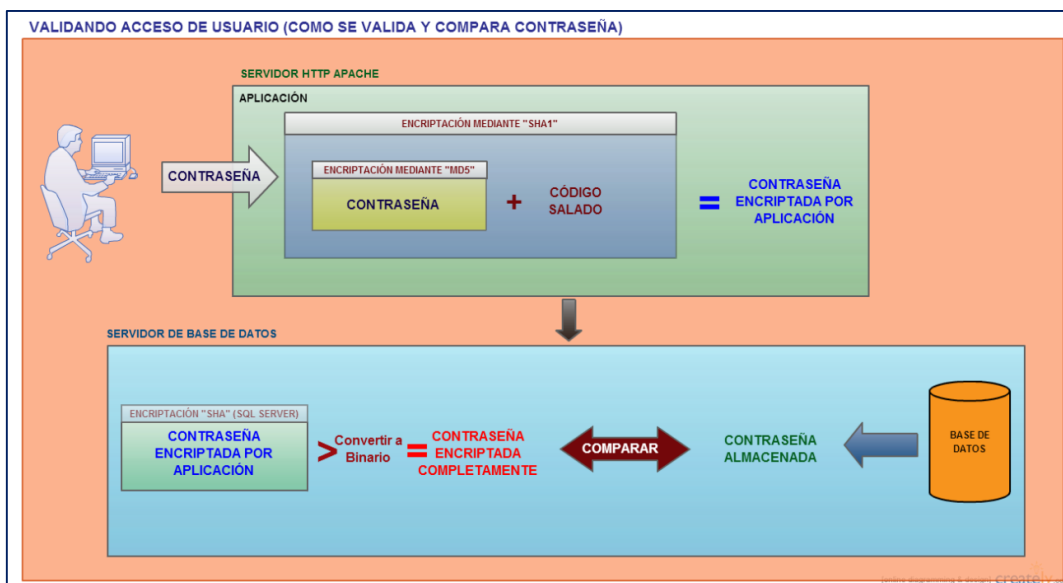


Ilustración 4.12: Diagrama de "Sistema de Encriptación - Validación de Acceso de Usuarios"
Fuente: Elaboración propia

4.1.3.2.2. Módulos del Sistema de Protección de Accesos

El framework BunnyPHP viene, de manera predeterminada, con 3 módulos programados que forman parte del Sistema de Protección de Accesos, los cuales podrán ser utilizados de manera alternativa por el programador. Los módulos son los siguientes:

Registrar Usuario

Este módulo es utilizado para registrar usuarios nuevos en el sistema, utilizando la técnica descrita en la parte 4.1.3.2.1. de la presente tesis para encriptar y almacenar la contraseña en la Base de Datos. El programador podrá modificar este módulo, tanto la apariencia como la parte funcional, según lo requiera.

Cambiar Contraseña

Este módulo es utilizado para que el usuario pueda cambiar su propia contraseña, al igual que el módulo “Registrar Usuario”, este utiliza la técnica descrita en la parte 4.1.3.2.1. de la presente tesis para encriptar y almacenar la contraseña en la Base de Datos. De la misma manera, el programador podrá modificar este módulo, tanto la apariencia como la parte funcional, según lo requiera.

Restaurar Contraseña

Este módulo es utilizado por un “administrador” que tenga acceso y pueda restaurar la contraseña de otro usuario que la haya olvidado. De la misma manera, este módulo utiliza la técnica de BunnyPHP para encriptar la contraseña. El módulo “Restaurar Contraseña” puede ser modificado por el programador según este lo requiera.

4.1.4. Controlador del Framework

A continuación se puede observar el *Cuadro 4.5: Backlog del Sprint 4 - "Controlador del Framework"* que representa el cuarto sprint, cuyo objetivo fue el desarrollo del “Controlador del Framework”, la parte más importante de este, ya que se encargará del control de peticiones, enrutamiento, utilización de verbos HTTP, etc.

<i>Objetivo: Desarrollar el "Controlador del Framework"</i>		<i>N° Sprint: 4</i>	<i>Día de Inicio: 09 de Junio del 2014</i>		<i>Duración: 40 días</i>
Id	Tarea	Tipo	Días	Estado	Responsable
1	Realizar la planificación de la iteración.	Planificación	2	Terminado	Emily / José
2	Analizar los requerimientos.	Análisis	2	Terminado	Emily / José
3	Diseñar el "Controlador del Framework".	Análisis	8	Terminado	Emily / José
4	Programar el "Sistema de Enrutamiento".	Desarrollo	10	Terminado	José
5	Programar las funciones del Controlador.	Desarrollo	10	Terminado	Emily
6	Probar y depurar el "Controlador del Framework".	Prueba	8	Terminado	Emily / José
7	Documentar	Documentación	10	Terminado	Emily / José

Cuadro 4.5: Backlog del Sprint 4 - "Controlador del Framework"
Fuente: Elaboración propia

De igual manera, en la *Ilustración 4.13: Burn down chart del Sprint 4 - "Controlador"* se presenta el avance que se tuvo con respecto al tiempo estimado. Esta vez en el Sprint 4, la cantidad de días empleados estuvo dentro del límite estimado.

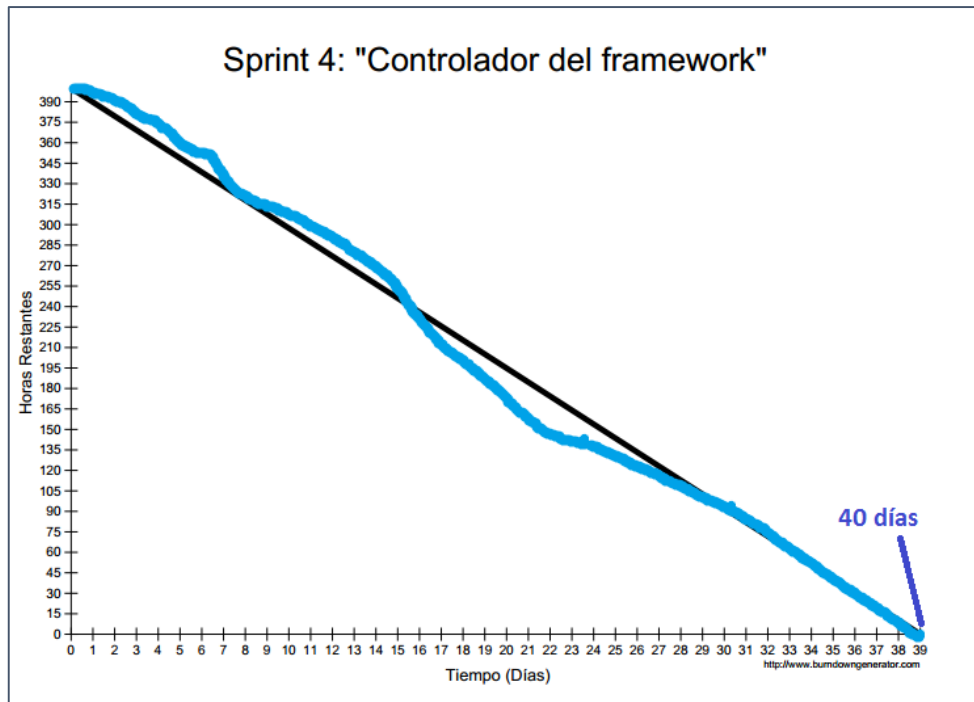


Ilustración 4.13: Burn down chart del Sprint 4 - "Controlador"
Fuente: Elaboración propia

4.1.4.1. Resumen de los requerimientos del "Controlador"

En la *Ilustración 4.14: Caso de Uso - "Controlador"* ubicada a continuación, podemos observar los "casos de uso" correspondientes:

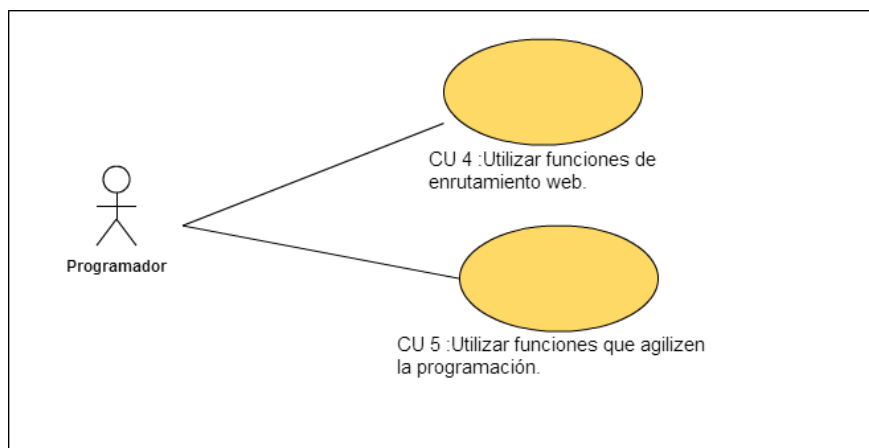


Ilustración 4.14: Caso de Uso - "Controlador"
Fuente: Elaboración propia

En base al análisis realizado a los requerimientos del framework, se vio por conveniente desarrollar un componente especial para cumplir con lo siguiente:

- Sistema de Enrutamiento.
- Utilización de verbos y códigos de estado HTTP.
- Utilización de URIs.
- Funciones que permitan agilizar el desarrollo de aplicaciones.
- Funciones que permitan un enlace rápido y seguro con la “Capa de Acceso a Datos”.

4.1.4.2. Diseño y desarrollo del “Controlador”

Este subdirectorio es la parte central de BunnyPHP, es como un núcleo encargado de administrar la aplicación desarrollada en este framework, administra las solicitudes, las respuestas, contiene las funciones que agilizan la programación y hacen que el sistema sea más escalable. El subdirectorio “Controlador” está compuesto por 2 archivos predeterminados y algunos más según necesidad del programador, los cuáles se describen a continuación:

- **cerebro.php**

Este es el “cerebro del conejo”, es decir el núcleo del framework “BunnyPHP”, pues las principales funcionalidades del framework se encuentran en este archivo. El Archivo “cerebro.php” gestiona las solicitudes (request) de la aplicación, permite el uso de URLs limpias, gestiona el uso de diferentes verbos HTTP, códigos de estado HTTP, funciones que agilizan el desarrollo y facilitan la programación de interfaz, permiten el llamado de reportes JASPER, etc.

Este fue desarrollado a partir del modelo de mini-framework llamado “stradlin” de Antonio Ognio. A partir de este modelo se implementaron las diferentes necesidades encontradas para hacer altamente funcional al framework.

- **uriS_General.php**

Siendo “BunnyPHP” un framework que trabaja con URLs limpias, mediante la función “ruta” se especifican las URIs

de cada módulo, operación y consulta, enviando los datos necesarios y el tipo de verbo correspondiente, permitiendo un enrutamiento web mediante URIs.

En este archivo se codifican las URIs correspondientes a los módulos y operaciones predeterminadas del framework, más no los desarrollados por el programador.

Ejemplo:

A continuación un ejemplo de archivo, al cual llamaremos “uriS_General.php”

```
<?php
ruta('/sesion','GET',function($params)
{
    set_status_code(200);
    seguridad_sesion($params[0]);
    conf_pre();

    echo "<html>";
    echo "<head>";
    set_link("estiloS_login.php");
    script('$( document ).ready(function(){$("#usuario").focus();});');
    //-----
    if(isset($_GET["error"]))
    {
        script('$( document ).ready(function()
        {
            aviso("Contraseña equivocada","white","#B40404");
        });');
    };
}
//-----
echo "</head>";
echo "<body>";

set_aviso();
incluir_modulo("modS_login.php");

echo "</body>";
echo "</html>";
});
?>
```

Ilustración 4.15: Código fuente de "uriS_General.php"
Fuente: Elaboración propia

- uriD_(NombreModulo).php

Estos archivos cumplen una función similar a “uriS_General.php”, sólo que en este archivo se codifican las URIs correspondientes a cada módulo y operación desarrollada por el mismo programador. BunnyPHP recomienda que el nombre de estos archivos tenga como prefijo “uriD_” más el nombre del módulo al que pertenece.

Ejemplo:

A continuación un ejemplo de archivo, al cual llamaremos “uriD_Pregrado.php”

```

<?php
ruta('/pregrado','GET',function($params)
{
    set_status_code(200);
    seguridad_sesion($params[0]);
    conf_pre();
    echo "<html>";
    echo "<head>";
    //set_link("pie.css");

    set_link("estiloS_menu.php");
    set_link("menu.js");
    set_link("funcD_pregrado.js");
    set_link("estiloS_general.php");
    set_link("estiloD_pregrado.php");

    script('$( document ).ready(function(){$("#txt_estudiante").focus();
        minimizar("capa1");
        minimizar("capa2");
        minimizar("capa3");
        menu_lateral();
    });');

    set_link_ext("//code.jquery.com/ui/1.10.4/themes/black-tie/jquery-ui.css","css");
    set_link_ext("//code.jquery.com/jquery-1.10.2.js","js");
    set_link_ext("//code.jquery.com/ui/1.10.4/jquery-ui.js","js");

    echo "</head>";
    echo "<body>";
    echo '<div id="popupbox_reporte"></div>';

    set_avisos();
    set_cargando();
    //incluir_plantilla("cabecera.php");
    incluir_plantilla("plantS_menu.php");
    incluir_modulo("modD_pregrado.php");
    echo "</body>";
    echo "</html>";

});

ruta('/estado_listoEntregar','POST',function($params)
{
    $resol=isset($_POST['resolucion']) ? $_POST['resolucion'] : null ;

    $datos_xml='
    <resoluciones>
        <nro_resol>'.$resol.'</nro_resol>
    </resoluciones>';
    $datos=logica::ejecutar_pa('GradosTitulos.[paAct_listoEntregar_resol]','json',$datos_xml);
});

```

Ilustración 4.16: Código fuente de "uriD_Pregrado.php"
Fuente: Elaboración propia

- **Subdirectorío “Librerías”**

En este subdirectorío se incluyen todas las librerías utilizadas por el framework, y también las que el programador decida adjuntar según las necesidades de la aplicación en desarrollo. BunnyPHP trae, de manera predeterminada las siguientes librerías:

- **Subdirectorío “php-jru”**

En este subdirectorío encontramos 2 archivos. El primero, “JdbcConnection.php”, que contiene las funciones para permitir la conexión del framework al servidor “Tomcat”. El segundo archivo llamado “php-jru.php” contiene funciones que permiten la ejecución de reportes JASPER para que sean mostrados en la interfaz de la aplicación.

- **Archivo “.htaccess”**

También conocido como “archivo de configuración distribuida”, permite definir diferentes directivas de configuración para cada directorío en el cual se incluya, sobrescribiendo la configuración global del servidor “HTTP Apache”. En BunnyPHP, el archivo “.htaccess” está predefinido para permitir el funcionamiento de su arquitectura. A continuación se muestra la codificación del archivo “.htaccess” del framework:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.+)$ enrutamiento.php/$1
```

Ilustración 4.17: Código fuente de “.htaccess”

Fuente: Elaboración propia

- **Archivo “enrutamiento.php”**

En este archivo, mediante la sentencia de PHP “require_once” se incluye el archivo “uriS_General.php” y los diferentes archivos “uriD_(NombreModulo).php” para que puedan ser reconocidos por el framework al momento de ser ejecutados.

Ejemplo:

```
<?php
require_once('Controlador/uriS_General.php');
require_once('Controlador/uriD_pregrado.php');
require_once('Controlador/uriD_postgrado.php');
require_once('Controlador/uriD_resoluciones.php');
?>
```

Ilustración 4.18: Código fuente de "enrutamiento.php"
Fuente: Elaboración propia

4.1.5. Capa de Presentación

En el Cuadro 4.6: *Backlog del Sprint 5 - "Capa de Presentación"* presentado a continuación, se puede observar el quinto sprint, cuyo objetivo fue el desarrollo de la "Capa de Presentación", la parte del framework con la cual interactuará constantemente el desarrollador y que contendrá las diferentes funciones que permitirán un desarrollo ágil e intuitivo.

<i>Objetivo: Desarrollar la "Capa de Presentación"</i>		<i>N° Sprint: 5</i>	<i>Día de Inicio: 21 de Julio del 2014</i>		<i>Duración: 23 días</i>
Id	Tarea	Tipo	Días	Estado	Responsable
1	Realizar la planificación de la iteración.	Planificación	2	Terminado	Emily / José
2	Analizar los requerimientos.	Análisis	2	Terminado	Emily / José
3	Diseñar la "Capa de Presentación".	Análisis	4	Terminado	Emily
4	Programar la "Capa de Presentación".	Desarrollo	5	Terminado	José
5	Programar funciones para la "Capa de Presentación".	Desarrollo	5	Terminado	Emily
6	Probar y depurar la "Capa de Presentación".	Prueba	2	Terminado	Emily
7	Adaptar módulos de Seguridad a la "Capa de Presentación".	Desarrollo	4	Terminado	Emily / José
8	Documentar	Documentación	4	Terminado	Emily / José

Cuadro 4.6: Backlog del Sprint 5 - "Capa de Presentación" Fuente: Elaboración propia

Por otro lado, en la *Ilustración 4.19: Burn down chart del Sprint 5 - "Capa de Presentación"* se presenta el Burn down chart del Sprint 5, el mismo que demuestra que en esta iteración se pudo cumplir con el tiempo estimado para su desarrollo.

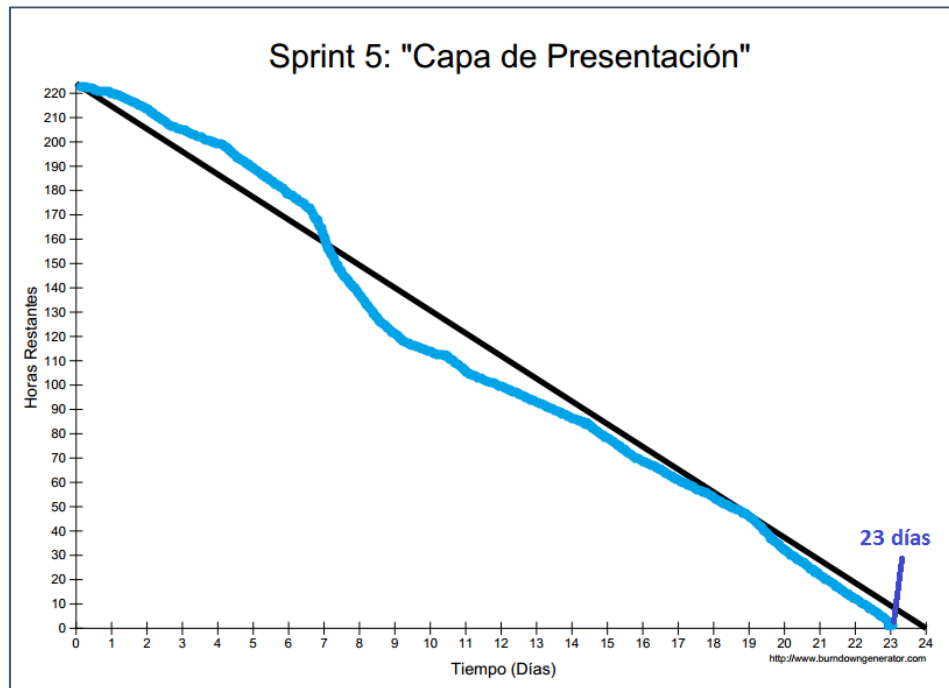


Ilustración 4.19: Burn down chart del Sprint 5 - "Capa de Presentación"
Fuente: Elaboración propia

4.1.5.1. Resumen de los requerimientos de la "Capa de Presentación"

En la *Ilustración 4.20: Casos de Uso - "Capa de Presentación"* ubicada a continuación, podemos observar los "casos de uso" correspondientes:

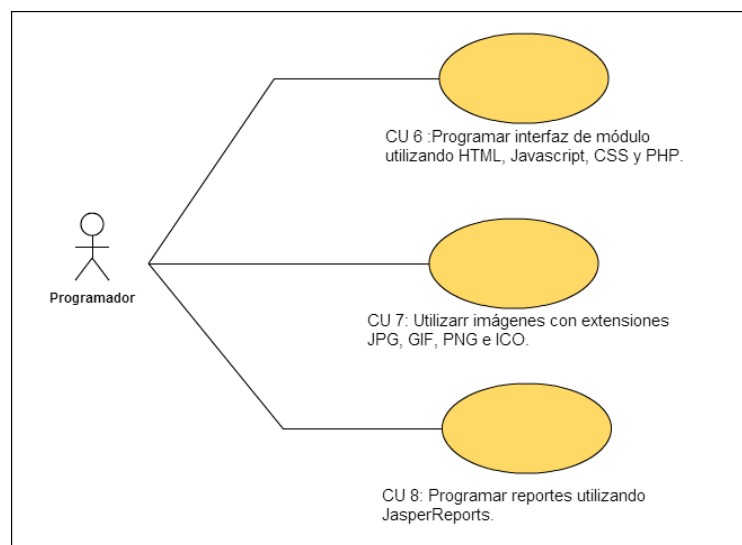


Ilustración 4.20: Casos de Uso - "Capa de Presentación"
Fuente: Elaboración propia

La “Capa de Presentación” es el componente que contiene todo el código y los recursos destinados a conformar la interfaz de los sistemas, el mismo que interactuará con el usuario. Los requerimientos de este componente, en forma resumida y en base al análisis realizado son:

- El HTML debe interactuar, de manera organizada y óptima con JavaScript y PHP.
- Los estilos (CSS) no deben estar programados en el mismo archivo, sino deben estar organizados en directorios independientes.
- Se deberán organizar, dentro de la “Capa de Presentación”, los recursos, que incluirá imágenes con extensiones como JPG, GIF, PNG, ICO, etc.
- El framework utilizará JasperReports para la creación de reportes.
- Siendo la “Capa de Presentación” el componente de mayor interacción con el programador, este deberá ser intuitivo y organizado.

4.1.5.2. Diseño y desarrollo de la “Capa de Presentación”

Esta es la “Capa de Presentación” del framework, la cual contiene todos los componentes necesarios de la interfaz, entre los cuales se incluyen las “Hojas de Estilos”, las “Funciones en Javascript”, los “Módulos en PHP”, las “Plantillas en PHP”, los “Reportes de JASPER” y los recursos.

- **Estilos**

Estas son las hojas de estilos que definen el diseño de los diferentes módulos que componen la aplicación que se desarrolla. Estos son archivos que mayormente tienen extensión CSS, aunque en otros casos serán utilizados como archivos PHP cuando se necesita utilizar estas mismas hojas de estilos, pero con variables PHP que definen características de interfaz como colores, tipo y tamaño de fuentes, color de fondo o imagen de fondo, etc.

El prefijo a utilizar para el nombre de estos archivos es “estiloD_NombreDelMódulo.(php o css)”, en el caso de pertenecer a módulos desarrollados por el programador, o “estilosS_NombreDelMódulo.(php o css)” en caso de pertenecer a módulos predefinidos del sistema.

Ejemplo:

A continuación un ejemplo de archivo, al cual llamaremos “estiloS_login.css”.

```
/* ##### CONTENEDOR ##### */

.sub_titulo
{
  background-color: rgb(46, 43, 72);
  width: 100%;
  color: #FFF;
  font-weight: bolder;
  height: 29px;
  font-family: Times;
  padding-top: 9px;
  text-align: center;
  vertical-align: middle;
  font-size: 20px;
}
```

Ilustración 4.21: Código fuente de "estiloS_login.css"
Fuente: Elaboración propia

Y otro ejemplo de un archivo de estilos con variables, al cual llamaremos “estiloS_login.php”:

```
<?php
header('content-type:text/css');
$colorPrincipal = "rgb(46, 43, 72);";

echo <<<FINCSS
/* ##### CONTENEDOR ##### */

.sub_titulo
{
  background-color: $colorPrincipal;
  width: 100%;
  color: #FFF;
  font-weight: bolder;
  height: 29px;
  font-family: Times;
  padding-top: 9px;
  text-align: center;
  vertical-align: middle;
  font-size: 20px;
}

FINCSS;
?>
```

Ilustración 4.22: Código fuente de "estiloS_login.php"
Fuente: Elaboración propia

- Funciones

Estas son funciones escritas en lenguaje “Javascript”, las cuales dan soporte a la funcionalidad de los módulos programados en PHP. Estas funciones estarán contenidas en archivos con extensión JS, y solamente se incluirán en esta parte de arquitectura aquellas que son de uso único de módulos. Pero en el caso de detectar funciones de uso reiterado, como la validación de un tipo de dato, la utilización de una función de cálculo constante en el proyecto, entre otros, estas funciones podrán colocarse en “Librerías”, el cual es un directorio paralelo a la “Capa de Presentación”.

El prefijo a utilizar para el nombre de estos archivos es “funcD_NombreDelMódulo.js”, en el caso de pertenecer a módulos desarrollados por el programador, o “funcS_NombreDelMódulo.js” en caso de pertenecer a módulos predefinidos del sistema.

Ejemplo:

A continuación un ejemplo de archivo, al cual llamaremos “funcS_registUsuario.js”.

```
function reg_usu() {
  if(v_reg_usu())
  {
    params={};
    params.dni=$("#codigo_reg_usu").val();
    params.nombre=$("#nombre_reg_usu").val();
    params.apP=$("#apP_reg_usu").val();
    params.apM=$("#apM_reg_usu").val();
    $.ajax({
      data : params,
      type: "POST",
      url: "registrarUsuario",
      dataType: "json",
      success : function(data)
      {
        aviso(data["rpta"],"white","green");
      }
      ,error: function()
      { aviso("No se pudo registrar usuario!","white","#B40404"); }
    });
  }
}
```

Ilustración 4.23: Código fuente de "funcS_registUsuario.js"

Fuente: Elaboración propia

- Módulos

En este directorio se incluyen todos los archivos PHP correspondientes a los diferentes módulos de la aplicación, los cuales incluirán código PHP y HTML según sea necesario, pero únicamente el código que deba estar incluido dentro de las etiquetas `<body>` `</body>`, ya que mediante la arquitectura de BunnyPHP, estas serán completadas según corresponda. Al utilizar por primera vez una copia limpia del framework, encontraremos archivos PHP ya incluidos, los cuales son módulos predeterminados que, junto con la estructura definida de la Base de Datos, forman parte de Control de Accesos a la aplicación que trae el framework para el uso recomendado pero no obligatorio del desarrollador.

Al tratarse de módulos predeterminados que vienen con el sistema, tiene el prefijo “modS_”, pero si se tratase de módulos desarrollados por el programador, BunnyPHP recomienda que utilice como prefijo “modD_” que significa “módulo desarrollado”.

Todos estos módulos tendrán soporte, tanto en el directorio “Estilo”, hablando del diseño de interfaz, como en “Funciones”, hablando de programación de funcionalidad.

Ejemplo:

A continuación un ejemplo de archivo, al cual llamaremos “modS_login.js”

```

<div class="contenedor_general_login">
  <script type="text/javascript"> var RecaptchaOptions = {theme : 'white'}; </script> <!--TEMA DEL CAPTCHA-->
  <div class="sub_titulo" id="sub_titulo" name="sub_titulo">SISTEMA INFORMÁTICO DE GRADOS Y TÍTULOS</div>
  <div class="login_contenedor">
    <div class="signin-box" style="color:white;font-size:21px;font-weight: bolder;font-family: sans-serif">
      CONTROL DE ACCESOS
      <form novalidate id="gaia_loginform" action="login" method="GET">
        <?php
          #####
          //variables de sesion
          add_vS("color1","rgb(2, 31, 55)");
          add_vS("color2","rgb(0,0,0)");
          #####
          #####
          //crea variables de Session, para datos de usuario
          $val_vS="dni, nomape, sexo, dependencia, codFac,codPriv, modulos";
          setDatosUsuarios($val_vS);
          #####
        ?>
        <input type="usuario" id="usuario" onFocus="" onKeyUp="return ValNumero(this);" value="" />
        <input type="password" id="contrasena" name="contrasena" placeholder="Ingresar Constraseña"/>
        <?php
          $error=null;
          if ($_SESSION[vS_ContadorErrados]>=n_intentos_captcha)
          {
            echo '<br>';
            echo '<br>';
            echo recaptcha_get_html(captcha_public_key, $error);
          }
        ?>
        <!--input type="checkbox" id="recordarme" name="recordarme" value="">Recordar usuario...-->
        <input type="submit" id="nuevoEstudiante_FaCarEsp" onclick="asignarCaptcha();" value="Ingresar">
        <div align="right"><label id="aviso" name="aviso"></label></div>
      </form>
    </div>
  </div>
</div>

```

Ilustración 4.24: Código fuente de "modS_login.js"

Fuente: Elaboración propia

- Plantillas

En este directorio se incluyen todos los archivos PHP correspondientes a los componentes de las plantillas utilizadas de manera reiterada en la programación de la aplicación. Por ejemplo, una cabecera conteniendo el título de la aplicación, un pie de la página conteniendo datos del desarrollador, un menú de navegación, etc.

Ejemplo:

A continuación un ejemplo de archivo, al cual llamaremos “plantS_menu.php”

```
<div id='cssmenu'>
<ul>
<li><a href='inicio'><div id="m_inicio"></div></a></li>
<li><a href='pregrado'><div id="m_busqueda"></div></a></li>
<li><a href='postgrado'><div id="m_estadistica"></div></a></li>
<li><a href='resoluciones'><div id="m_operaciones"></div></a></li>
<li><a href='#'><div id="m_administracion"></div></a>
<ul>
<li class='submenu'><a href='cambiarContrasena'><div class='submenu'>Cambiar Contraseña</div></a></li>
<li class='submenu'><a href='regUsuario'><div class='submenu'>Registrar Usuario</div></a></li>
<li class='submenu'><a href='restpass'><div class='submenu'>Restaurar Contraseña</div></a></li>
</ul>
</li>
<li><a href='salir'><div id="m_cerrar"></div></a></li>
</ul>
</div>
```

Ilustración 4.25: Código fuente de "plantS_menu.php"
Fuente: Elaboración propia

- **Recursos**

Este es el directorio que contiene todos los elementos agregados de la aplicación, específicamente de los diferentes módulos que la componen. Estos son sobre todo archivos como imágenes, documentos de formato portátil (pdf), videos, audios, etc. BunnyPHP recomienda que estos recursos sean agrupados en subdirectorios cuyos nombres serán el de los módulos a los que pertenecen, en caso de pertenecer a varios módulos se puede colocar en el subdirectorio “Recursos_general”.

- **Reportes**

En este directorio se almacenan los archivos JASPER de los diferentes reportes que utilice la aplicación, los cuales son ejecutados por el Servidor Tomcat.

4.1.6. Arquitectura Final

A continuación se presenta, en el *Cuadro 4.7: "Arquitectura"*, el sexto sprint, cuyo objetivo fue el diseño y desarrollo de la “Arquitectura Final”,

que permitirá la interacción ordenada y segura de todos los componentes de framework desarrollados hasta el momento.

Objetivo: Diseño y desarrollo de la "Arquitectura Final"		N° Sprint: 6	Día de Inicio: 15 de Agosto del 2014		Duración: 33 días
Id	Tarea	Tipo	Días	Estado	Responsable
1	Realizar la planificación de la iteración.	Planificación	2	Terminado	Emily / José
2	Analizar los requerimientos.	Análisis	2	Terminado	Emily / José
3	Diseñar la "Arquitectura del Framework".	Análisis	5	Terminado	Emily
4	Programar la "Arquitectura del Framework".	Desarrollo	10	Terminado	Emily
5	Adaptar todos los componentes programados a la Arquitectura.	Desarrollo	15	Terminado	José
6	Probar y Depurar el Framework.	Prueba	8	Terminado	Emily / José
7	Documentar	Documentación	5	Terminado	Emily / José

Cuadro 4.7: "Arquitectura"
Fuente: Elaboración propia

Finalmente, en la *Ilustración 4.26*: Burn down chart del Sprint 6 - "Arquitectura Final" se presenta el Burn down chart del Sprint 6, el mismo en el que podemos observar que se pudo cumplir con el tiempo estimado.

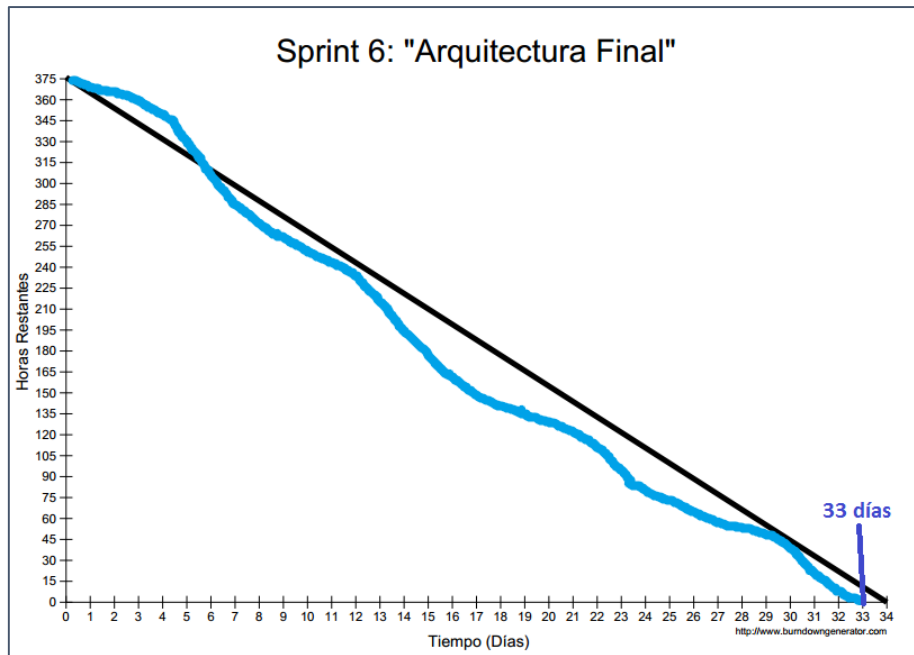


Ilustración 4.26: Burn down chart del Sprint 6 - "Arquitectura Final"
Fuente: Elaboración propia

4.1.6.1. Resumen de los requerimientos de la "Arquitectura"

La Arquitectura final del Framework ha sido determinada en base a los requerimientos recopilados de la "Oficina Universitaria de Informática y Sistemas" y lo que se ha determinado por conveniente según el desarrollo que se ha venido haciendo hasta el presente sprint.

Estos son:

- Básicamente, la arquitectura debe ser "Cliente – Servidor", ya que todos los datos están centralizados en los servidores ubicados en el local central.
- La cadena de conexión y la información necesaria que es utilizada para acceder a la Base de Datos debe estar protegida.
- Procurar utilizar la "Programación por capas" para separar la parte de diseño de interfaz de la lógica del negocio.

4.1.6.2. Diseño y desarrollo de la "Arquitectura Final"

El framework BunnyPHP posee una arquitectura "Cliente – Servidor" de "N Capas". Si bien posee varios elementos en su arquitectura, los principales son justamente las 3 capas que lo componen: la "Capa de Presentación", la "Capa de Seguridad" y la "Capa de Acceso a Datos". A diferencia de lo tradicional en la arquitectura de las aplicaciones que utilizan "N Capas", este framework no posee la "Capa de la Lógica del Negocio", pues uno de sus principios es desplazar toda la lógica a la Base de Datos. Esto permitirá a la aplicación ser más escalable ya que

no contiene datos en su programación, sino que procesa los datos que obtiene de la Base de Datos. Es por ello que, al momento de plasmar en la aplicación un cambio en la “lógica del negocio”, este cambio o mantenimiento se realizará únicamente sobre la Base de Datos y no sobre la programación.

A continuación observaremos la *Ilustración 4.27: Diagrama de Despliegue del framework "BunnyPHP"*

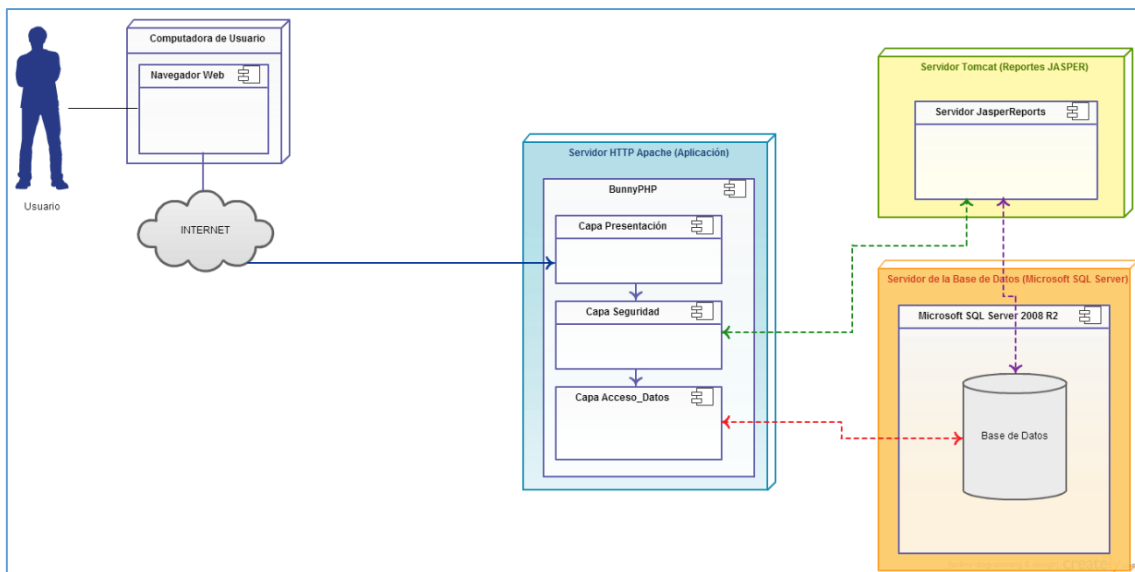


Ilustración 4.27: Diagrama de Despliegue del framework "BunnyPHP"
Fuente: Elaboración propia

4.1.6.3. Árbol de directorios del Framework y su funcionamiento

Si bien, en el “Diagrama de Despliegue” del framework se expuso que este está compuesto básicamente por tres capas; la “Capa de Presentación”, la “Capa de Seguridad” y la “Capa de Acceso a Datos”, la arquitectura de BunnyPHP es mucho más compleja que ello, por lo tanto es necesario exponerla mediante su “Árbol de Directorios”. Como se podrá observar, el framework está básicamente compuesto por un directorio llamado “Privado”, donde se incluyen datos vulnerables relacionados con el acceso a los servidores de la base de datos, y el directorio llamado “Proyecto”, donde se encuentran las tres capas del framework, el controlador y las librerías. Para finalizar, luego de realizar el desarrollo de todos los componentes descritos en los sprints anteriores, se estableció la Arquitectura final del framework. En la *Ilustración 4.28: Árbol de Directorios del Framework* se muestra su estructura:

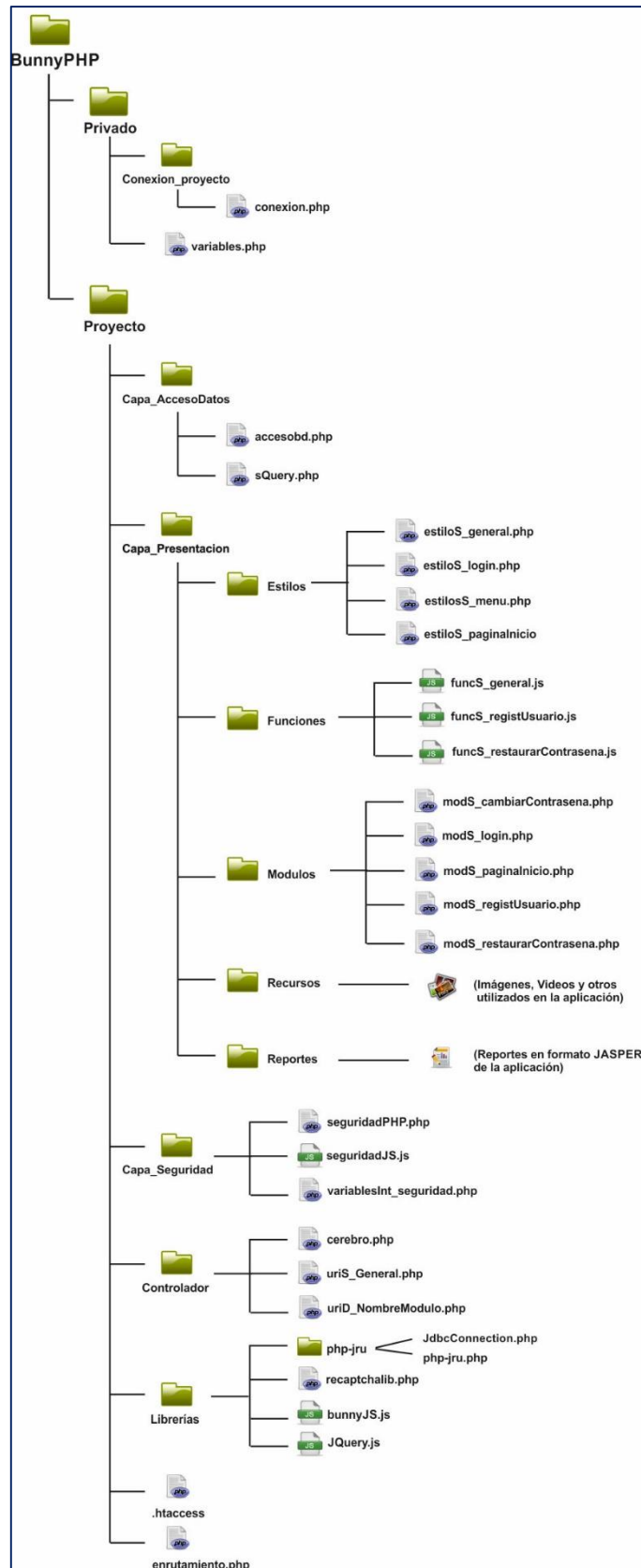


Ilustración 4.28: Árbol de Directorios del Framework
Fuente: Elaboración propia

PARTE III: DISCUSIÓN DE RESULTADOS

CAPÍTULO V: PRUEBAS Y APLICACIÓN DEL FRAMEWORK (Fase “Post – Game” de SCRUM)

Luego de haber realizado el “análisis, diseño y desarrollo” de cada uno de los componentes del framework, en este capítulo se procederá a realizar todas las pruebas necesarias para determinar si el resultado obtenido hasta el momento cumple con los requerimientos plasmados en los “Casos de Uso”, y con los objetivos planteados en la primera parte de la tesis.

Para ello, se han realizado dos tipos de pruebas, las “unitarias” y las de “integración”. Con las unitarias se procederá a comprobar el cumplimiento de cada uno de los “casos de uso del sistema”, realizando pruebas de “caja negra” por cada uno de ellos. Mediante las pruebas de integración se buscará contrastar el cumplimiento de los objetivos principales del framework, los cuales consisten en la obtención de características específicas de este en el desarrollo web, como son: “ágil”, “cooperativo”, “seguro” y “escalable”. Por cada una de estas características se diseñaron y ejecutaron algunas pruebas, logrando así evaluar la eficiencia del framework.

5.1. Pruebas Unitarias

En esta parte, las pruebas unitarias fueron diseñadas y ejecutadas en base a los “casos de uso del sistema”, los mismos que fueron representados en la *Ilustración 3.10: "Diagrama de Casos de Uso del Sistema"*. Estos “Casos de Uso” ya fueron expuestos en los capítulos anteriores, sin embargo serán nuevamente listados a continuación:

- CU1: Programar componente o función que acceda a la Base de Datos.
- CU2: Administrar usuarios y privilegios.
- CU 3: Utilizar funciones de seguridad.

- CU 4: Utilizar funciones de Enrutamiento Web.
- CU 5: Utilizar funciones que agilizen la programación.
- CU 6: Programar interfaz de módulos utilizando HTML, CSS, Javascript y PHP.
- CU 7: Utilizar imágenes con extensiones JPG, GIF, PNG, ICO.
- CU 8: Programar reportes utilizando JasperReports.

Luego de esto, también es importante mencionar cuales son los “componentes” obtenidos en base a los requerimientos descritos en los “Casos de Uso del Sistema”. Estos son:

- Capa de Acceso a Datos
- Base de Datos de Seguridad
- Capa de Seguridad
- Controlador del Framework
- Capa de Presentación
- Arquitectura Final

Con todo esta información ya expuesta, es importante ahora describir la estructura de las pruebas realizadas. Estas pruebas serán plasmadas en cuadros llamados “Casos de Prueba”, describiendo posterior a estos detalles e imágenes de las pruebas realizadas. Los “Casos de Prueba” incluirán:

- Caso de Prueba, que muestra el número de prueba que se está realizando.
- Caso de Uso, que muestra el número del “Caso de Uso” del cual se está comprobando su cumplimiento mediante la prueba.
- Objetivo, que describe en términos generales el objetivo de la prueba.
- Componente relacionado, que señala el componente del framework comprometido con la prueba.
- Descripción de la prueba, que describe de manera específica la forma de cómo se llevará a cabo la prueba.
- Entradas, que describe el parámetro o código que se ingresará al componente para probar su funcionamiento.
- Salidas, que describe el parámetro, código o resultado de la ejecución de la prueba.
- Resultado, que mostrará si la prueba cumple o no con los requerimientos.

A continuación se describirán las pruebas:

5.1.1. Prueba Unitaria N°1

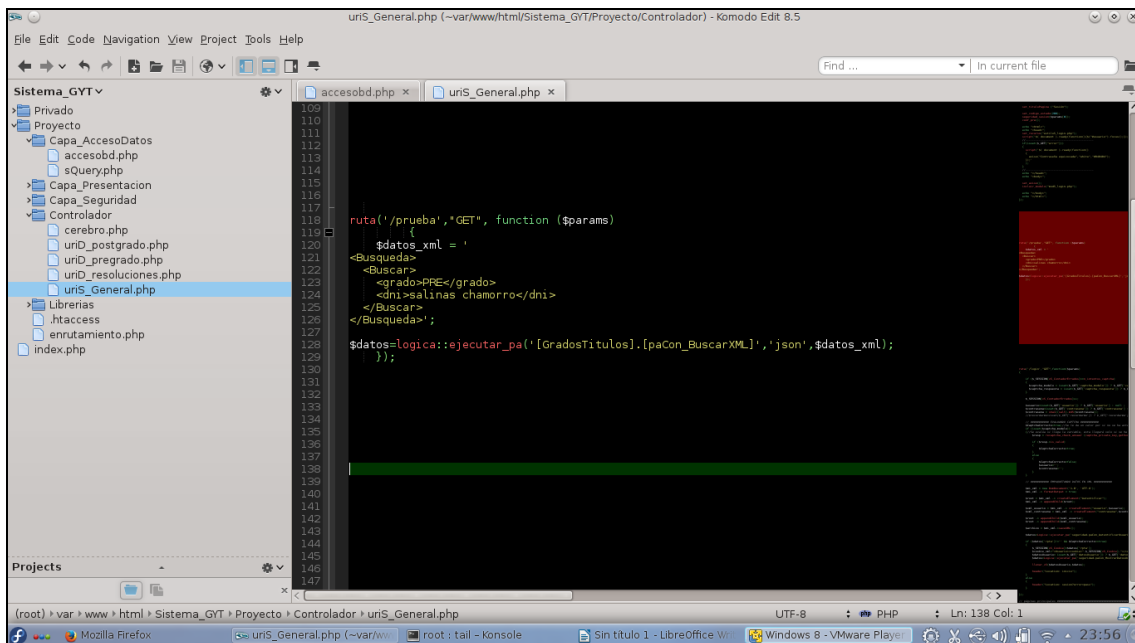
A continuación, en el *Cuadro 5.1: Caso de Prueba N°1* se exponen los detalles de esta prueba.

Prueba Unitaria	
Caso de Prueba	1
Caso de Uso	CU 1: Programar componente o función que acceda a la Base de Datos.
Objetivo	Probar el acceso a la “Base de Datos” desde la “Capa de Acceso a Datos”.
Componente relacionado	Capa de Acceso a Datos
Descripción de la Prueba	Se realizará una prueba del archivo “accesobd.php”, sobre la función encargada de realizar las consulta a la base de datos “ejecutar_pa”.
Entradas	<p>Este es el “XML” que se envía mediante la función de la “Capa de “Acceso a Datos” a un procedimiento de la Base de Datos.</p> <pre>\$datos_xml = ' <Busqueda> <Buscar> <grado>PRE</grado> <dni>rios monterrey</dni> </Buscar> </Busqueda>';</pre> <p>Y esta es la línea de código que ejecuta la función de la “Capa de Acceso a Datos”.</p> <pre>\$datos=logica::ejecutar_pa('[GradosTitulos].[paCon_BuscarXML]','json',\$datos_xml);</pre>
Salidas	<p>La salida (lo que se muestra en la pantalla del navegador) son los datos obtenidos de la Base de Datos en formato JSON, pues es lo que se especificaba en la línea de código de la “Entrada”.</p> <pre>{ "1": ["70312129", "EMILY ROSSY SALINAS CHAMORRO", "INGENIER\u00cdA", "INGENIER\u00cdA DE SISTEMAS Y COMPUTACI\u00d3N", "SIN ESPECIALIDAD", "SIN MENCIÓN", "AUTOMÁTICO", "BACHILLER", "IS", "1", "A80074I", "01", "3", "Entregado"], "2": ["20073089", "LINDAURA SALINAS CHAMORRO", "CIENCIAS DE LA SALUD", "FARMACIA Y BIOQU\u00cdMICA", "SIN ESPECIALIDAD", "SIN MENCIÓN", "AUTOMÁTICO", "BACHILLER", "FB", "1", "952016B", "05", "3", "Entregado"], "3": ["00000000", "RAUL SALINAS CHAMORRO", "EDUCACI\u00d3N Y CIENCIAS HUMANAS", "EDUCACI\u00d3N SECUNDARIA", "COMPUTACI\u00d3N E INFORM\u00c1TICA - MATEM\u00c1TICA", "SIN"] }</pre>

	MENCION", "NO TIENE", "NO TIENE", "ED", "0", "851601G", "04", "0", "Sin Diploma"]}
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.1: Caso de Prueba N°1
Fuente: Elaboración propia

A continuación, en la *Ilustración 5.1: Entrada de "Caso de Prueba N°1"* se puede observar, en las líneas de código, el XML con los datos de consulta y la utilización de la función de la “Capa de Acceso a Datos” que conecta con la Base de Datos.



```

109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147

ruta('/prueba','GET', function ($params)
{
    $datos_xml = '
<Busqueda>
<Buscar>
<grado>PRE</grado>
<dni>salinas chamorro</dni>
</Buscar>
</Busqueda>';
    $datos=logica::ejecutar_pa('GradosTítulos',[paCon_BuscarXML'],'json',$datos_xml);
});

```

Ilustración 5.1: Entrada de "Caso de Prueba N°1"
Fuente: Elaboración propia

Luego, en la *Ilustración 5.2: Salida de "Caso de Prueba N°1"* podemos observar la salida, en la cual se muestran los datos que retorna la Base de Datos en formato JSON.

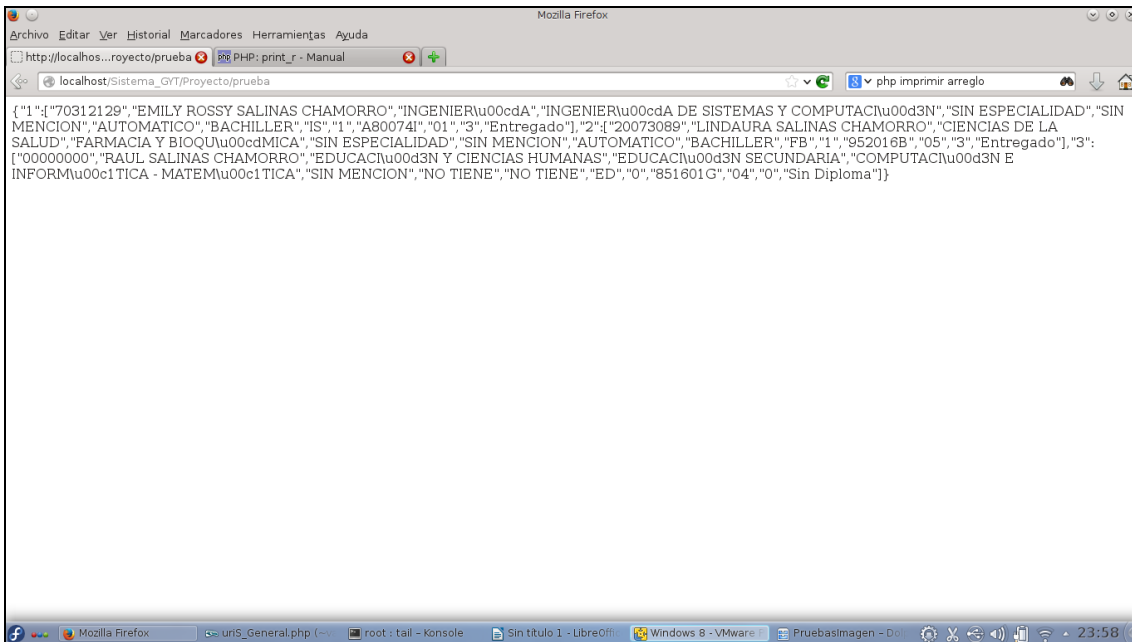


Ilustración 5.2: Salida de "Caso de Prueba N°1"
Fuente: Elaboración propia

5.1.2. Prueba Unitaria N°2

En el *Cuadro 5.2: Caso de Prueba N°2* que se muestra a continuación, es posible observar los detalles de esta prueba:

Prueba Unitaria	
Caso de Prueba	2
Caso de Uso	CU 2: Administrar usuarios y privilegios.
Objetivo	Probar el funcionamiento de la "Base de Datos de Seguridad" y "Procedimientos Almacenados" relacionados.
Componente relacionado	Base de Datos de Seguridad.
Descripción de la Prueba	Se realizará una prueba sobre la Base de Datos de seguridad, ejecutando el procedimiento almacenado "seguridad.paIns_RegistrarUsuario" y realizando una consulta a la tabla de usuarios.
Entradas	Este es el "XML" que se envía directamente al procedimiento almacenado desde el "Microsoft SQL Server Management" para realizar la prueba: <pre>declare @xml xml = ', <reg_usu> <datosExt> <cod>70312129</cod> <nombre>Emily Rossy</nombre> <apP>Salinas</apP> <apM>Chamorro</apM> <cmb_sexo>F</cmb_sexo></pre>

	<pre> <cmb_priv>01</cmb_priv> <cla_reg_usu>xxxxxxx</cla_reg_usu> </datosExt> </reg_usu> </pre> <p>declare @rpta varchar(25) exec seguridad.paIns_RegistrarUsuario @xml,@rpta output print @rpta</p>
Salidas	Esta es la salida que se muestra en la consola del “Microsoft SQLServer Management”: Grabado correctamente
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.2: Caso de Prueba N°2

Fuente: Elaboración propia

En la *Ilustración 5.3: Entrada y Salida del "Caso de Prueba N°2"*, se muestra una captura de pantalla del “Microsoft SQL Server Management”. En la sección central (sección de sripts de consultas) se observa la creación de una variable de tipo “xml” y la asignación de su valor, siendo ejecutado posteriormente el procedimiento almacenado “seguridad.paIns_RegistrarUsuario”. Luego, en la parte inferior (sección de Mensajes) se puede observar la “salida o resultado” de la “ejecución” de este procedimiento almacenado, el cual es “Grabado Correctamente”.

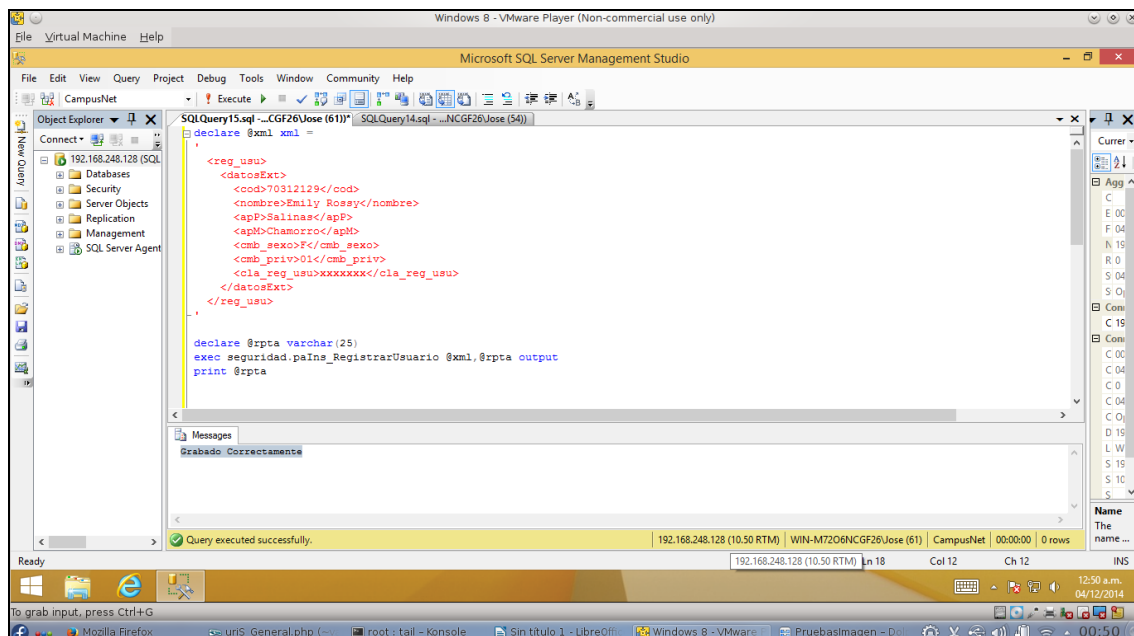


Ilustración 5.3: Entrada y Salida del "Caso de Prueba N°2"

Fuente: Elaboración propia

5.1.3. Prueba Unitaria N°3

Prueba Unitaria	
Caso de Prueba	3
Caso de Uso	CU 3: Utilizar funciones de seguridad.
Objetivo	Probar el funcionamiento de una de las funciones de seguridad de la “Capa de Seguridad”.
Componente relacionado	Capa de Seguridad
Descripción de la Prueba	El framework incluye un módulo de “login” (ingreso al sistema). Por seguridad una sesión dura 10 minutos. Se realizará una prueba ingresando al sistema y esperando 10 minutos, actualizando la página luego de ello.
Entradas	En este caso no se puede mostrar de manera explícita la entrada, sino que se hará una breve descripción: <ul style="list-style-type: none"> - Se ingresa, en el módulo de “Login”, los datos de acceso (usuario + contraseña). - Se deja el sistema en “inactividad” por un tiempo mayor a 10 minutos.
Salidas	Al igual que las entradas, en este caso se realizará sólo una breve descripción de la salida. <ul style="list-style-type: none"> - Al actualizar la página luego de un periodo de inactividad de 12 minutos, el sistema se cierra y se vuelve a mostrar la pantalla de “login”.
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.3: Caso de Prueba N°3

Fuente: Elaboración propia

En la captura de imagen mostrada en la *Ilustración 5.4: Entrada del "Caso de Prueba N°3"*, se observa que el usuario está ingresando al sistema mediante sus datos de acceso (usuario y contraseña). Esto lo está realizando a las “7:00” como se puede observar en la parte inferior derecha de la pantalla. Luego de esto, se ingresará y se dejará al sistema en “inactividad” por más de 10 minutos (tiempo establecido en el framework).

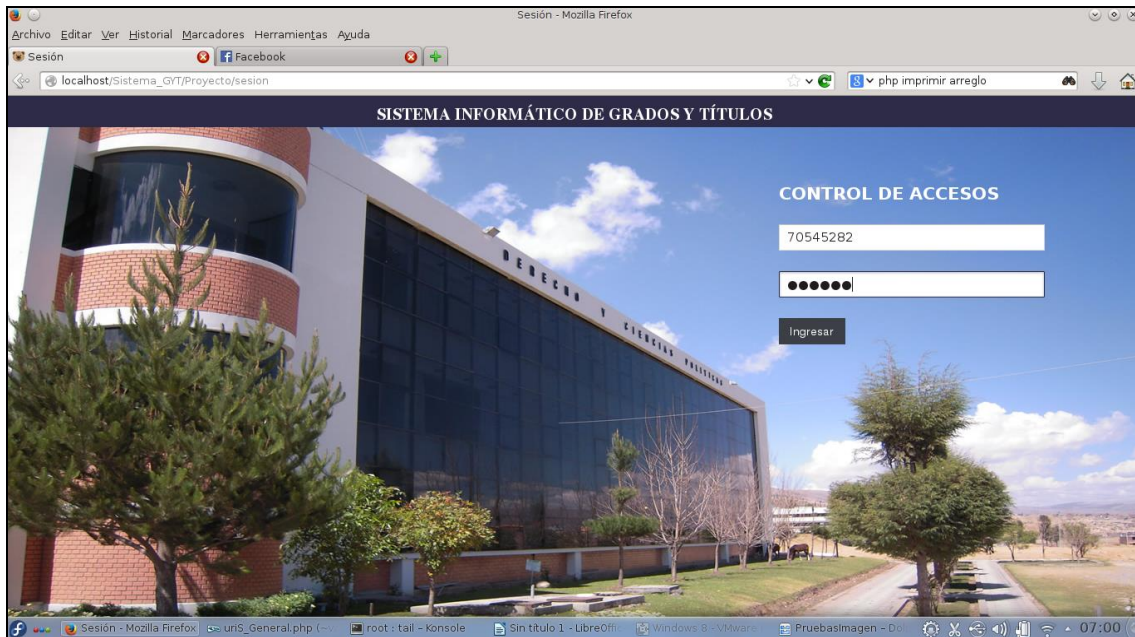


Ilustración 5.4: Entrada del "Caso de Prueba N°3"
Fuente: Elaboración propia

La Ilustración 5.5: Intermedio del "Caso de Prueba N°3" muestra que el sistema ha sido accedido, y siendo las "07:07" según el reloj ubicado en la parte inferior derecha, permanece en "inactividad".

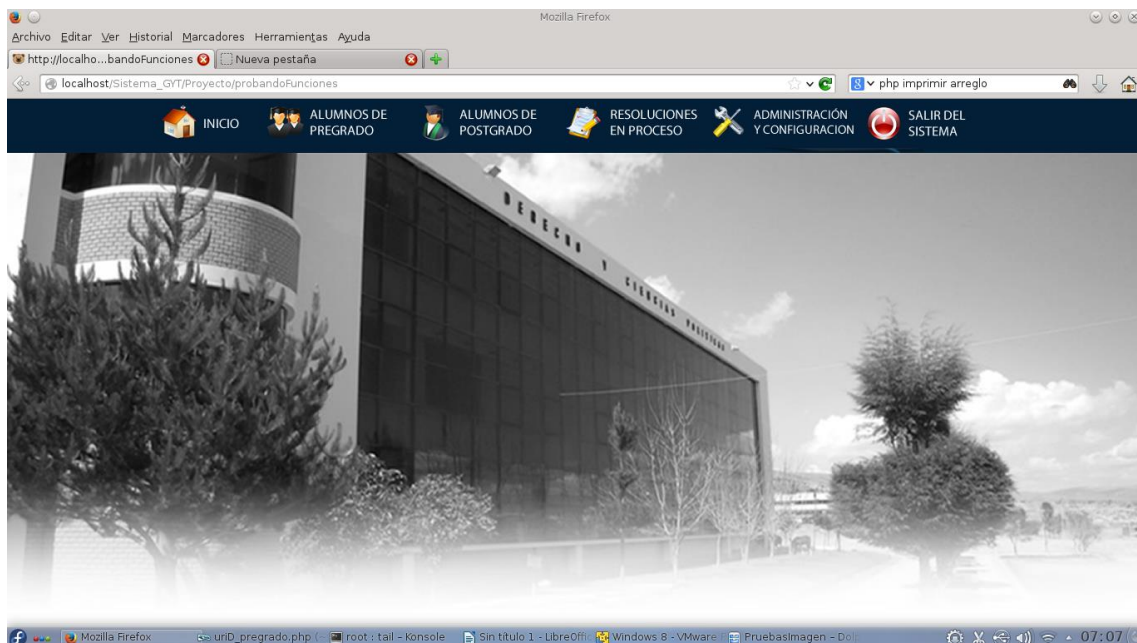


Ilustración 5.5: Intermedio del "Caso de Prueba N°3"
Fuente: Elaboración propia

Finalmente, en la *Ilustración 5.6: Salida del "Caso de Prueba N°3"*, siendo las "07:11" se hizo un refresco de la página, y habiéndose vencido el tiempo máximo de 10 minutos, el sistema terminó con la sesión y volvió a la pantalla de login (control de accesos).

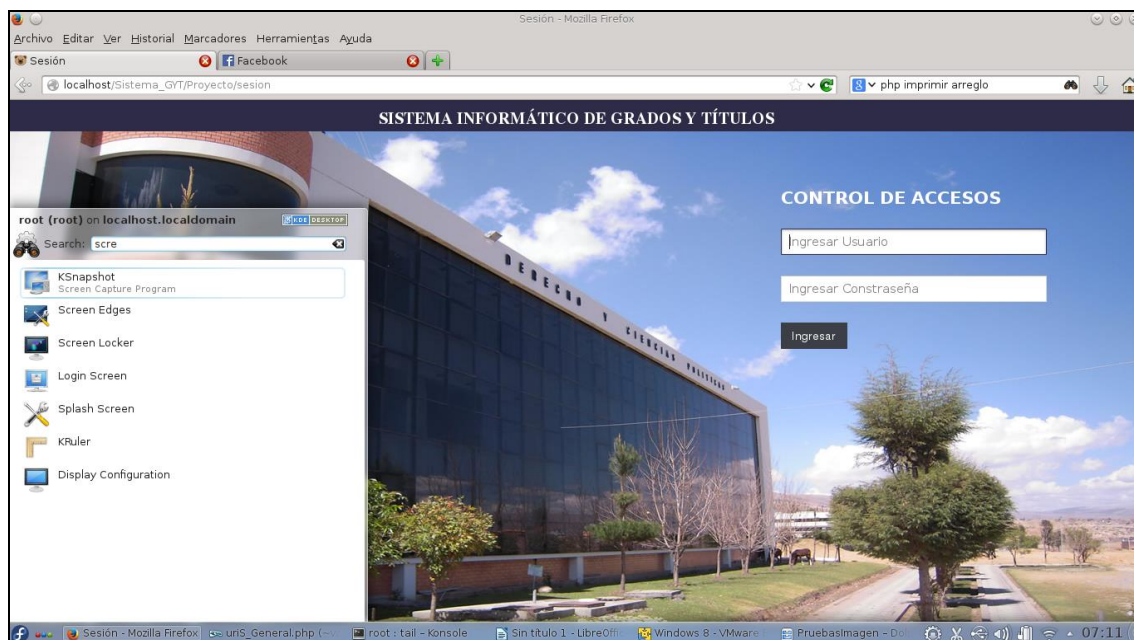


Ilustración 5.6: Salida del "Caso de Prueba N°3"
Fuente: Elaboración propia

5.1.4. Prueba Unitaria N°4

Prueba Unitaria	
Caso de Prueba	4
Caso de Uso	CU 4: Utilizar funciones de enrutamiento web.
Objetivo	Probar el sistema de enrutamiento web del framework.
Componente relacionado	Controlador
Descripción de la Prueba	Se realizará una prueba directa sobre el archivo "uriS_General.php" del Controlador, programando HTML básico, relacionándolo a un URI (identificador de recursos uniforme). Luego, se realizará la prueba accediendo a este, mediante el ingreso del URI en la barra de direcciones.
Entradas	Este es el código que se escribe sobre el archivo "uriS_General.php". Como se podrá observar, el uri asignado es "probando_uri". <pre> ruta("/probando_uri","GET",function(\$params) { echo "<html>"; echo "<head>"; echo "<title>Probando enrutamiento y URI</title>"; echo "</head>"; </pre>

	<pre> echo "<body>"; echo "<p>Esta es una prueba del Sistema de Enrutamiento web mediante URIs</p>"; echo "</body>"; echo "</html>"; }); </pre>
Salidas	<p>Luego de haber ingresado en la “barra de direcciones” del navegador, además de la ruta del proyecto, la uri “probando_uri”, se pudo obtener en la pantalla del navegador el siguiente mensaje:</p> <p>Esta es una prueba del Sistema de Enrutamiento web mediante URIs.</p>
Resultado	<p>Esta Prueba Unitaria tuvo resultados satisfactorios.</p>

Cuadro 5.4: Caso de Prueba N°4
Fuente: Elaboración propia

Para realizar la prueba del “Sistema de enrutamiento web del framework”, se escribió el código que aparece en la *Ilustración 5.7: Entrada en "Caso de Prueba N°4"*, utilizando como URI “probando_uri” y como verbo HTTP el “GET”.

```

358     Logica::ejecutar_pa(["seguridad].[paAct_RestaurarContrasena]", "json", $cod, true);
359 });
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380 ruta("/probando_uri", "GET", function($params)
381 {
382     echo "<html>";
383     echo "<head>";
384     echo "<title>Probando enrutamiento y URI</title>";
385     echo "</head>";
386
387     echo "<body>";
388     echo "<p>Esta es una prueba del Sistema de Enrutamiento web mediante URIs</p>";
389     echo "</body>";
390     echo "</html>";
391 });
392
393
394

```

Ilustración 5.7: Entrada en "Caso de Prueba N°4"
Fuente: Elaboración propia

Luego de ello, se pone en la “barra de direcciones”, posterior al url correspondiente al proyecto, el URI “probando_uri”, logrando de esta manera leer el código programado. Todo esto se puede observar en la *Ilustración 5.8: Salida en "Caso de Prueba N°4"*:

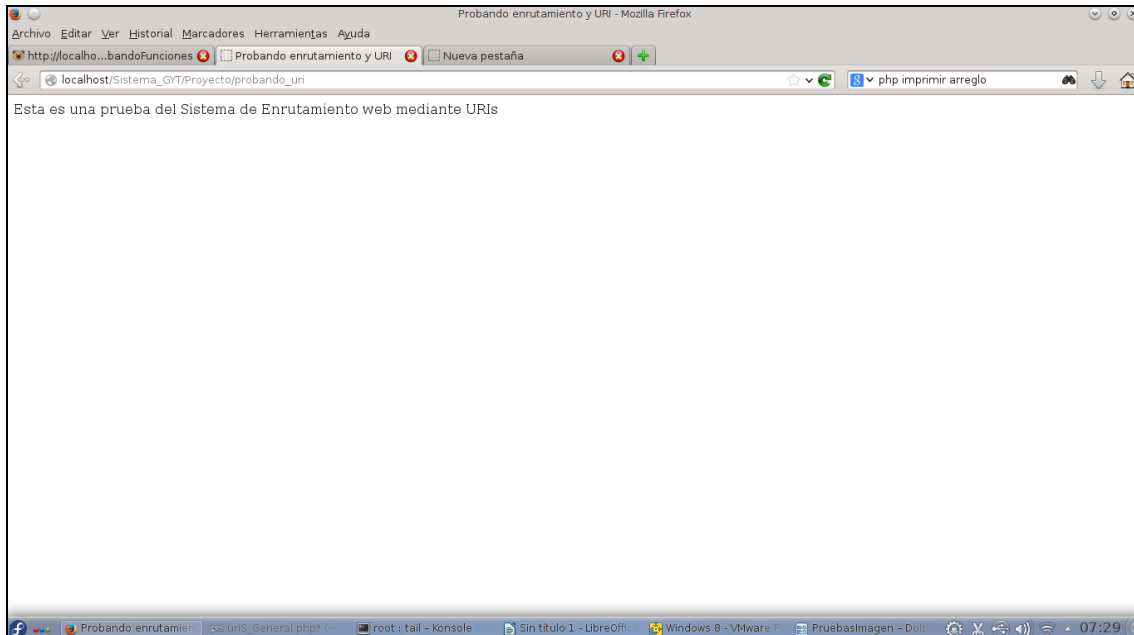


Ilustración 5.8: Salida en "Caso de Prueba N°4"
Fuente: Elaboración propia

5.1.5. Prueba Unitaria N°5

En el *Cuadro 5.5: Caso de Prueba N°5* ubicado a continuación, se detallan las características de esta prueba:

Prueba Unitaria	
Caso de Prueba	5
Caso de Uso	CU 5: Utilizar funciones que agilicen la programación.
Objetivo	Probar funciones que agilizan la programación.
Componente relacionado	Controlador
Descripción de la Prueba	Se realizará una prueba de las funciones “incluir_plantilla()” e “incluir_modulo()”, comprobando en el navegador que estas funcionan eficientemente.
Entradas	A continuación se mostrará código, en el cual se incluye la utilización de algunas funciones, pero en este caso lo que nos interesa es la utilización de las funciones “incluir_plantilla()” e “incluir_modulo()”: <pre> ruta('/probandoFunciones',"GET",function(\$params) { conf_pre(); echo "<html>"; </pre>

	<pre> echo "<head>"; set_recurso("estiloS_menu.php"); set_recurso("menu.js"); set_recurso("funcD_pregrado.js"); set_recurso("estiloS_general.php"); set_recurso("estiloD_pregrado.php"); echo "</head>"; echo "<body>"; echo '<div id="popupbox_reporte"></div>'; echo '<div id="popupbox_minimizado">Asignaturas Matriculables</div>'; <u>incluir_plantilla("plantS_menu.php");</u> <u>incluir_modulo("modD_pregrado.php");</u> echo "</body>"; echo "</html>"; }); </pre>
Salidas	Luego, al probar dicho código, en la pantalla del navegador se muestra la “plantilla” y el “módulo”.
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.5: Caso de Prueba N°5

Fuente: Elaboración propia

A continuación, en la *Ilustración 5.9: Entrada N°1 de "Caso de Prueba N°5"* se puede observar que se está utilizando la función “incluir_plantilla()”, ingresando como dato el nombre de la plantilla “plantS_menu.php”. En este caso, aún no se está utilizando “incluir_modulo()”.

```

373
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
}

ruta('/probandoFunciones', "GET", function($params)
{
    conf_pre();
    echo "<html>";
    echo "<head>";
    //set_recurso("pie.css");

    set_recurso("estilos_menu.php");
    set_recurso("menu.js");
    set_recurso("funcD_pregrado.js");
    set_recurso("estilos_general.php");
    set_recurso("estiloD_pregrado.php");

    echo "</head>";
    echo "<body>";
    echo "<div id='popupbox_reporte'></div>";
    echo "<a href='javascript:mostrar_popupAsignaturas()'><div id='popupbox_minimizado'>Asignaturas Matriculables";

    incluir_plantilla("plantS_menu.php");
    //incluir_modulo("modD_pregrado.php");

    echo "</body>";
    echo "</html>";
});
}

```

Ilustración 5.9: Entrada N°1 de "Caso de Prueba N°5"
Fuente: Elaboración propia

En la *Ilustración 5.10: Salida N°1 de "Caso de Prueba N°5"*, se observa que se ha incluido la plantilla “plantS_menu.php”, pero aún no se ha incluido ningún módulo.

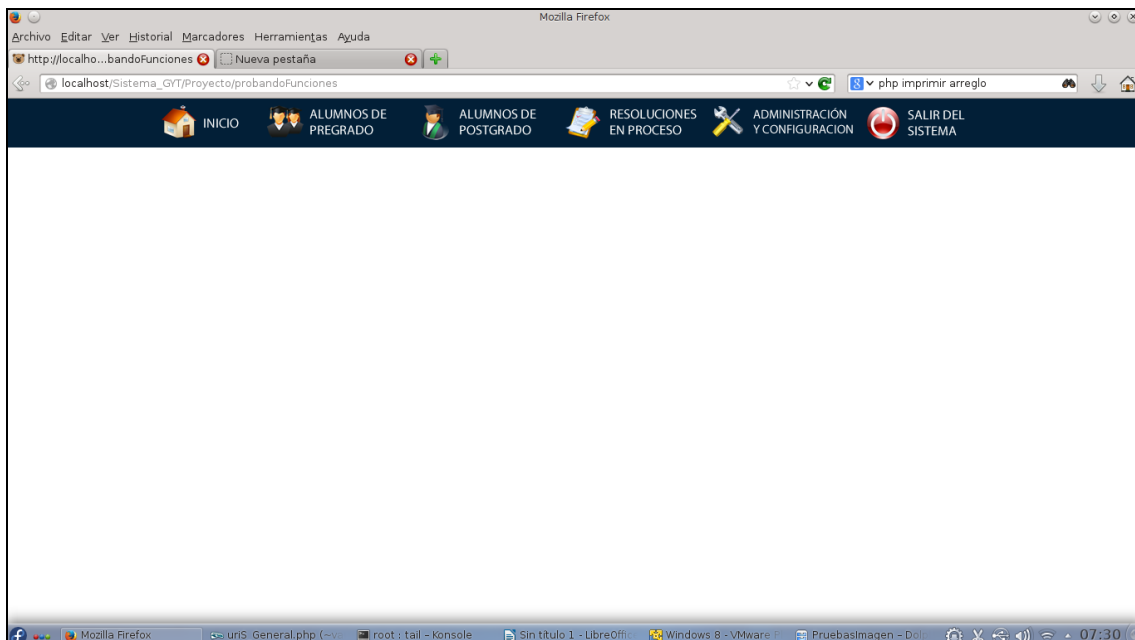
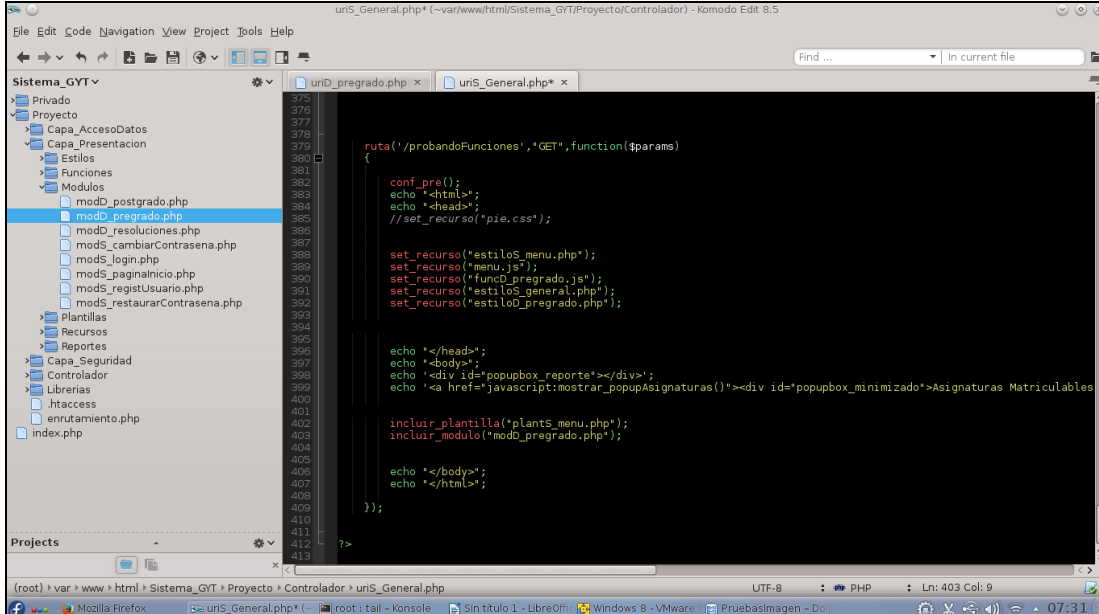


Ilustración 5.10: Salida N°1 de "Caso de Prueba N°5"
Fuente: Elaboración propia

Continuando con este “caso de pruebas”, el código que se puede visualizar en la *Ilustración 5.11: Entrada N°2 de "Caso de Prueba N°5"* es como el de la *Ilustración 5.9: Entrada N°1 de "Caso de Prueba N°5"*, sólo que en este caso se está utilizando la función “`incluir_modulo()`” para utilizar módulos que hayan sido programados previamente.



```

375
376
377
378
379 ruta('/probandoFunciones', 'GET', function($params)
380 {
381
382     conf_pre();
383     echo "<html>";
384     echo "<head>";
385     //set_recurso("pie.css");
386
387     set_recurso("estiloS_menu.php");
388     set_recurso("menu.js");
389     set_recurso("funcD_pregrado.js");
390     set_recurso("estiloS_general.php");
391     set_recurso("estiloS_pregrado.php");
392
393     echo "</head>";
394     echo "<body>";
395     echo "<div id='popupbox_reporte'></div>";
396     echo "<a href='javascript:mostrar_popupAsignaturas()'><div id='popupbox_minimizado'>Asignaturas Matriculables";
397
398     incluir_plantilla("plantS_menu.php");
399     incluir_modulo("modD_pregrado.php");
400
401     echo "</body>";
402     echo "</html>";
403
404 });
405
406
407
408
409
410
411
412
413

```

Ilustración 5.11: Entrada N°2 de "Caso de Prueba N°5"

Fuente: Elaboración propia

Finalmente, al haber agregado la función “`incluir_modulo()`” sobre el código, podemos observar que en la *Ilustración 5.12: Salida N°2 de "Caso de Prueba N°5"* se visualiza, además de la plantilla, el módulo agregado.

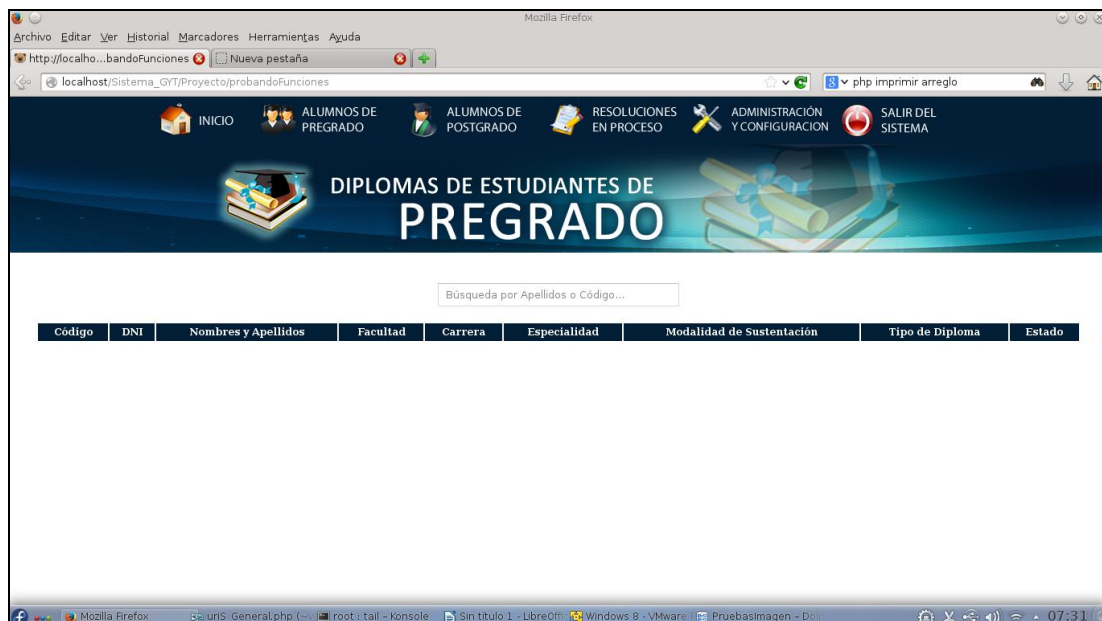


Ilustración 5.12: Salida N°2 de "Caso de Prueba N°5"

Fuente: Elaboración propia

5.1.6. Prueba Unitaria N°6

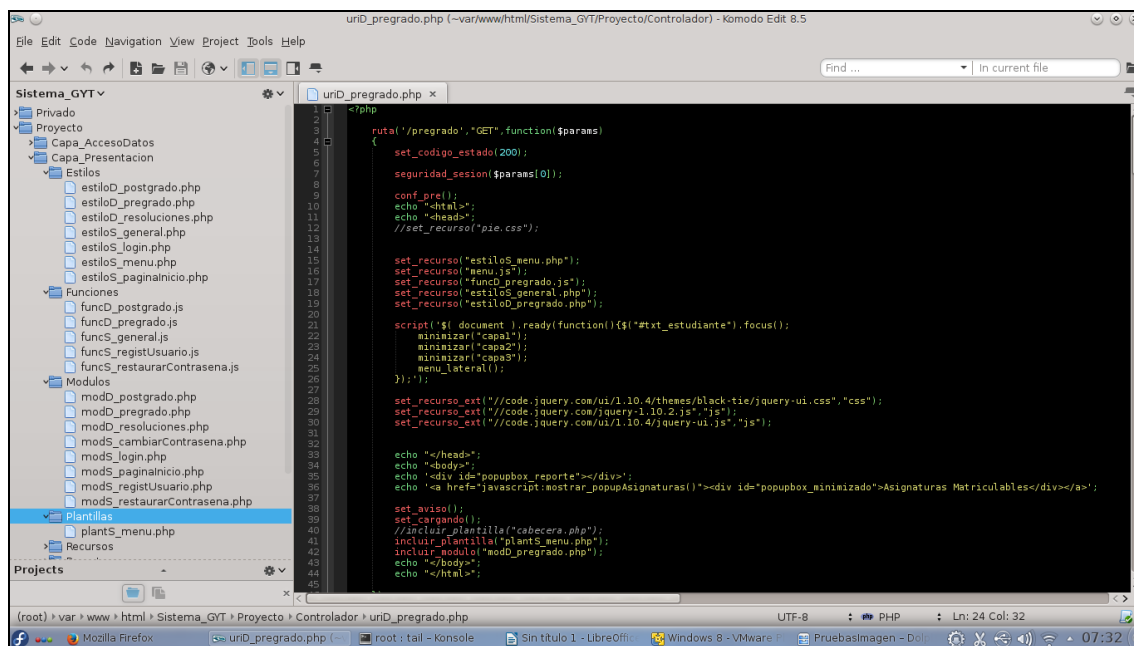
Prueba Unitaria	
Caso de Prueba	6
Caso de Uso	CU 6: Programar interfaz de módulo utilizando HTML, CSS, Javascript y PHP.
Objetivo	Probar la posibilidad de utilizar código HTML, CSS, Javascript y PHP de forma organizada.
Componente relacionado	Capa de Presentación
Descripción de la Prueba	Se programará un módulo de ejemplo, en el cual se utilizarán recursos HTML, CSS, Javascript y PHP de manera organizada, cumpliendo la estructura del framework.
Entradas	<p>A continuación se muestra como ejemplo un módulo que hace uso de los diferentes recursos:</p> <pre> ruta('/pregrado','GET',function(\$params) { set_codigo_estado(200); seguridad_sesion(\$params[0]); conf_pre(); echo "<html>"; echo "<head>"; set_recurso("estiloS_menu.php"); set_recurso("menu.js"); set_recurso("funcD_pregrado.js"); set_recurso("estiloS_general.php"); set_recurso("estiloD_pregrado.php"); script('\$ (document).ready(function(){\$("#txt_estudiante").focus(); minimizar("capa1"); minimizar("capa2"); minimizar("capa3"); menu_lateral(); });'); set_recurso_ext("//code.jquery.com/ui/1.10.4/themes/black-tie/jquery-ui.css","css"); set_recurso_ext("//code.jquery.com/jquery-1.10.2.js","js"); set_recurso_ext("//code.jquery.com/ui/1.10.4/jquery-ui.js","js"); </pre>

	<pre> echo "</head>"; echo "<body>"; echo '<div id="popupbox_reporte"></div>'; echo '<div id="popupbox_minimizado">Asignaturas Matriculables</div>'; set_avis(); set_cargando(); incluir_plantilla("plantS_menu.php"); incluir_modulo("modD_pregrado.php"); echo "</body>"; echo "</html>"; }); </pre>
Salidas	Luego, al probar dicho código, en la pantalla del navegador se muestra el módulo de “pregrado”.
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.6: Caso de Prueba N°6

Fuente: Elaboración propia

En la *Ilustración 5.13: Entrada de "Caso de Prueba N°6"* se puede observar la programación de un módulo bajo el uri “pregrado”. En este módulo se puede observar la utilización de diferentes “recursos internos” y “recursos externos”, los cuales son incluidos mediante las funciones “set_recursos” y “set_recursos_ext” respectivamente. Estas funciones pertenecen al framework y permiten utilizar HTML, CSS, Javascript, PHP de manera eficiente y organizada.



```

uriD_pregrado.php (-var/www/html/Sistema_GYT/Proyecto/Controlador) - Komodo Edit 8.5
1  *?php
2
3  ruta('/pregrado', 'GET', function($params)
4  {
5      set_codigo_estado(200);
6      seguridad_sesion($params[0]);
7
8
9      conf_pre();
10     echo "<html>";
11     echo "<head>";
12     //set_recurso("pie.css");
13
14
15     set_recurso("estilos_menu.php");
16     set_recurso("menu.js");
17     set_recurso("funcD_pregrado.js");
18     set_recurso("estilos_general.php");
19     set_recurso("estiloD_pregrado.php");
20
21     script($[ document ].ready(function){$("#txt_estudiante").focus();
22         minimizar("capa1");
23         minimizar("capa2");
24         minimizar("capa3");
25         menu_lateral();
26     });});
27
28
29     set_recurso_ext("//code.jquery.com/ui/1.10.4/themes/black-tie/jquery-ui.css", "css");
30     set_recurso_ext("//code.jquery.com/jquery-1.10.2.js", "js");
31     set_recurso_ext("//code.jquery.com/ui/1.10.4/jquery-ui.js", "js");
32
33
34     echo "</head>";
35     echo "<body>";
36     echo '<div id="popupbox_reporte"></div>';
37     echo '<a href="javascript:mostrar_popupAsignaturas()"><div id="popupbox_minimizado">Asignaturas Matriculables</div></a>';
38
39     set_avis();
40     set_cargando();
41     //incluir_plantilla("cabecera.php");
42     incluir_plantilla("plantS_menu.php");
43     incluir_modulo("modD_pregrado.php");
44     echo "</body>";
45     echo "</html>";

```

Ilustración 5.13: Entrada de "Caso de Prueba N°6"

Fuente: Elaboración propia

Posteriormente, se realiza la prueba del módulo programado ingresando el uri correspondiente en la "barra de direcciones". En la *Ilustración 5.14: Salida N°1 de "Caso de Prueba N°6"* podemos observar que el módulo se carga satisfactoriamente, y en la *Ilustración 5.15: Salida N°2 de "Caso de Prueba N°6"* se comprueba que los estilos CSS y las funciones Javascript se cargan satisfactoriamente.



Ilustración 5.14: Salida N°1 de "Caso de Prueba N°6"
Fuente: Elaboración propia

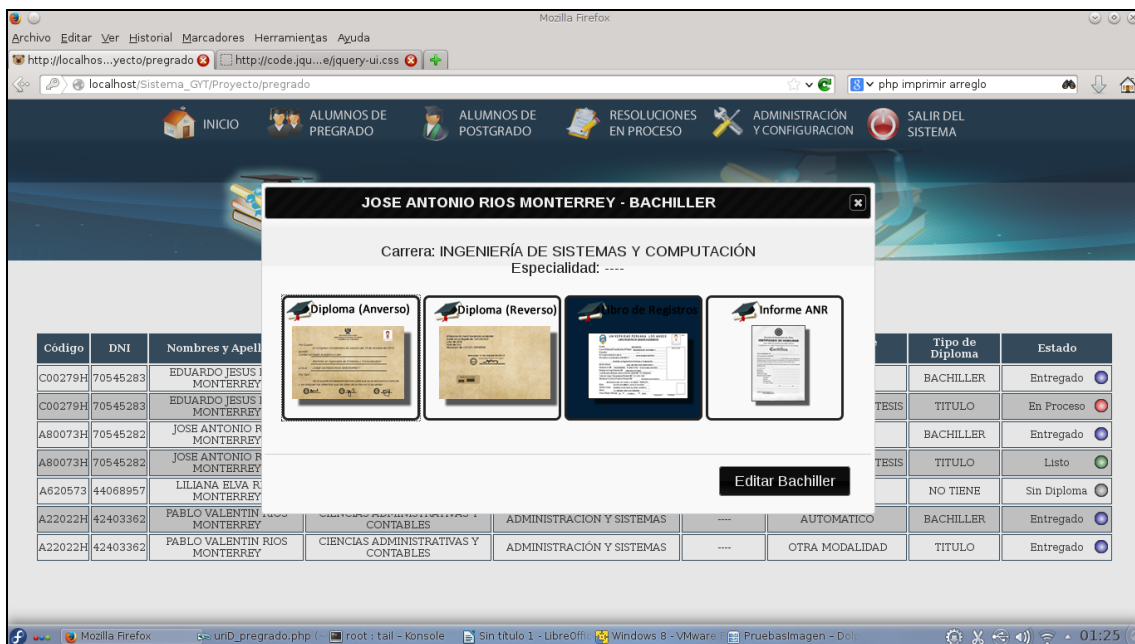


Ilustración 5.15: Salida N°2 de "Caso de Prueba N°6"
Fuente: Elaboración propia

5.1.7. Prueba Unitaria N°7

En el *Cuadro 5.7: Caso de Prueba N°7* se pueden observar los detalles de esta prueba unitaria:

Prueba Unitaria	
Caso de Prueba	7
Caso de Uso	CU 7: Utilizar imágenes con extensión JPG, GIF, PNG e ICO.
Objetivo	Probar el uso de imágenes con extensión JPG, GIF, PNG e ICO.
Componente relacionado	Capa de Presentación
Descripción de la Prueba	Se realizará una prueba mediante la función “set_imagen()” del Controlador, ingresando el nombre de la imagen a mostrar.
Entradas	A continuación se muestra código en el cual se utiliza únicamente la función set_imagen(), relacionándolo con la URI “probandoImágenes”.
	<pre> ruta("/probandoImágenes","GET",function(\$params) { set_imagen("cargando.gif"); }); </pre>
Salidas	Luego, al probar dicho código, en la pantalla del navegador se muestra una imagen GIF.
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.7: Caso de Prueba N°7

Fuente: Elaboración propia

En la *Ilustración 5.16: Entrada de "Caso de Prueba N°7"*, se puede observar el uso de la función “set_imagen()”.

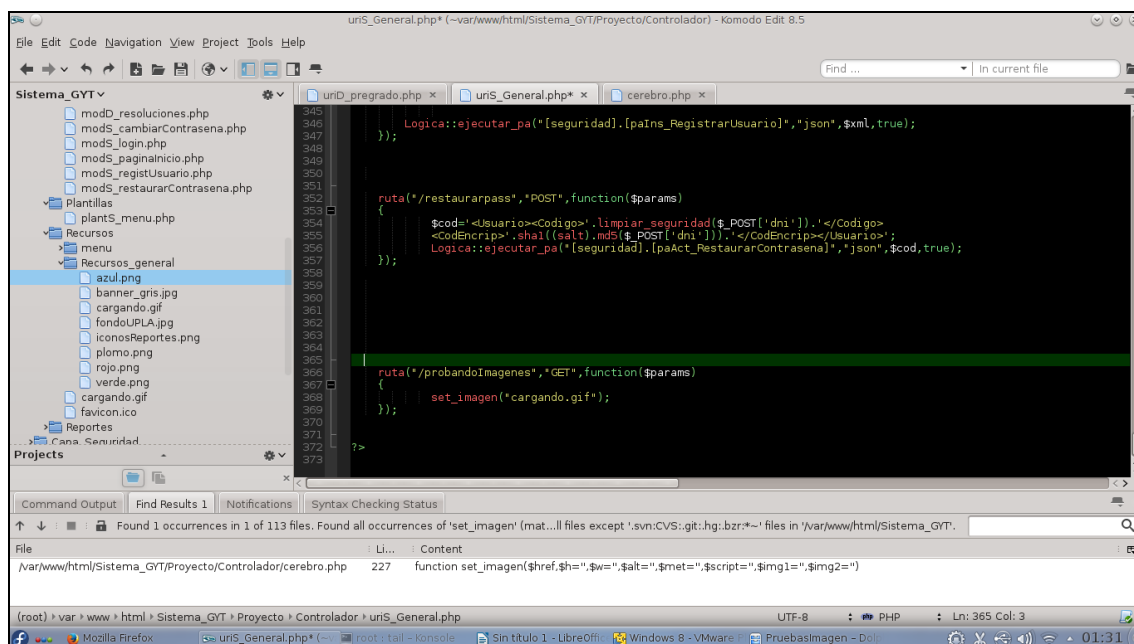


Ilustración 5.16: Entrada de "Caso de Prueba N°7"

Fuente: Elaboración propia

Luego, al realizar la prueba de este módulo en el navegador, comprobaremos que se muestra satisfactoriamente. Esto lo podemos observar en la *Ilustración 5.17: Salida de "Caso de Prueba N°7"*, a continuación:

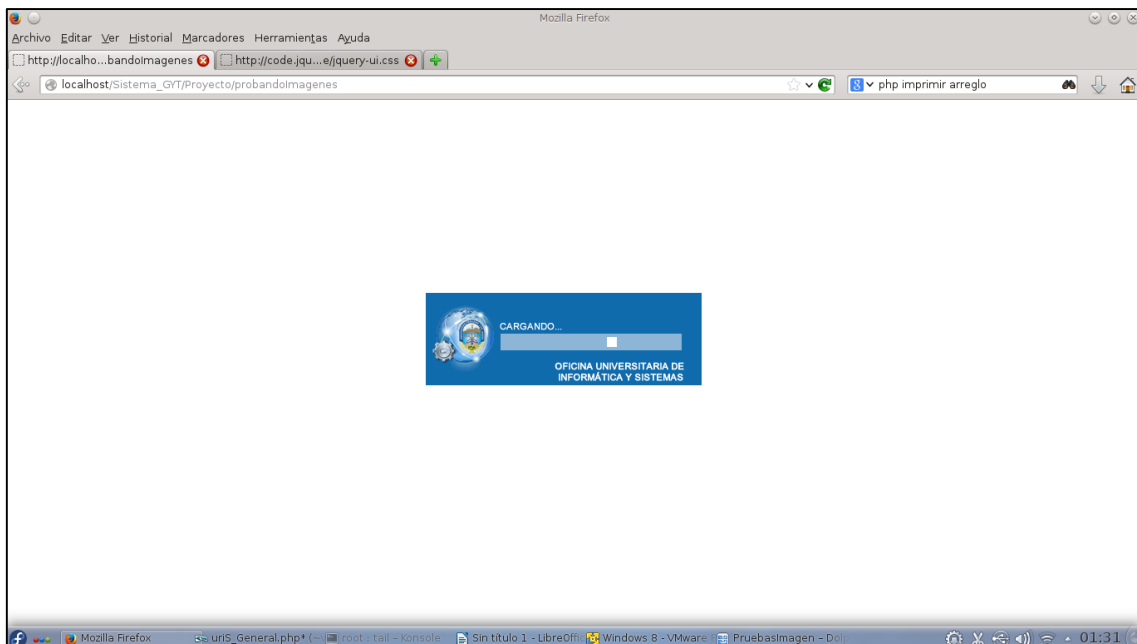


Ilustración 5.17: Salida de "Caso de Prueba N°7"
Fuente: Elaboración propia

5.1.8. Prueba Unitaria N°8

En el *Cuadro 5.8: Caso de Prueba N°8* se detalla esta prueba unitaria:

Prueba Unitaria	
Caso de Prueba	8
Caso de Uso	CU 6: Programar reportes utilizando JasperReports.
Objetivo	Probar la utilización de reportes JasperReports mediante las funciones del framework.
Componente relacionado	Capa de Presentación
Descripción de la Prueba	Sobre el módulo programado en el Caso de Pruebas N°6, se añadirá una función para generar un Reporte. En el navegador se comprobará que este es mostrado correctamente.
Entradas	A continuación, se muestra como ejemplo, un módulo que hace uso de las diferentes funciones, entre ellas "set_reporte_jasper()" que sirve para generar un reporte: <pre> ruta('/diploma',"GET",function(\$params) { \$Alumno=isset(\$_GET['Alumno']) ? \$_GET['Alumno'] : null ; </pre>

	<pre> \$car_esp=isset(\$_GET['car_esp']) ? \$_GET['car_esp'] : null ; \$n_tdi=isset(\$_GET['n_tdi']) ? \$_GET['n_tdi'] : null ; \$n_tdi=intval(\$n_tdi); if (\$n_tdi==1) \$nomarchivo='diplo_bac_fte26'; else if (\$n_tdi==2) \$nomarchivo='diplo_tit_fte26'; else if (\$n_tdi==3) \$nomarchivo='diplo_Mae_fte26'; else if (\$n_tdi==4) \$nomarchivo='diplo_Doct_fte26'; else if (\$n_tdi==5) \$nomarchivo='diplo_Seg_fte26'; \$parametros=array (array("Alumno",\$Alumno), array("car_esp",\$car_esp), array("n_tdi",\$n_tdi)); set_reporte_jasper(\$nomarchivo,\$parametros); script('ocultarCargando()'); }); </pre>
Salidas	Se visualiza en la pantalla de navegador el reporte generado.
Resultado	Esta Prueba Unitaria tuvo resultados satisfactorios.

Cuadro 5.8: Caso de Prueba N°8

Fuente: Elaboración propia

En la *Ilustración 5.18: Entrada de "Caso de Prueba N°8"*, podemos observar la programación de un módulo con uri “diploma” que utiliza la función “set_reporte_jasper”, el mismo que se encarga de acceder a los archivos JASPER, convertirlos a PDF y mostrarlos en pantalla.

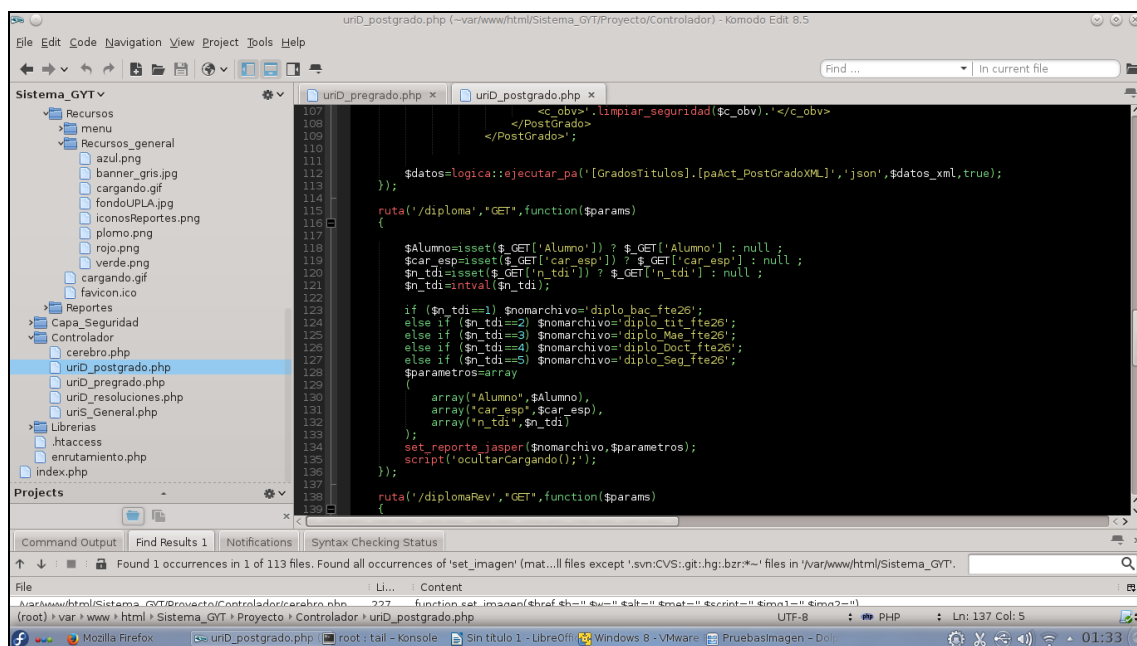


Ilustración 5.18: Entrada de "Caso de Prueba N°8"

Fuente: Elaboración propia

Como prueba del código mostrado en la *Ilustración 5.18: Entrada de "Caso de Prueba N°8"*, en la *Ilustración 5.19: Salida de "Caso de Prueba N°8"* se puede observar la generación de un reporte.

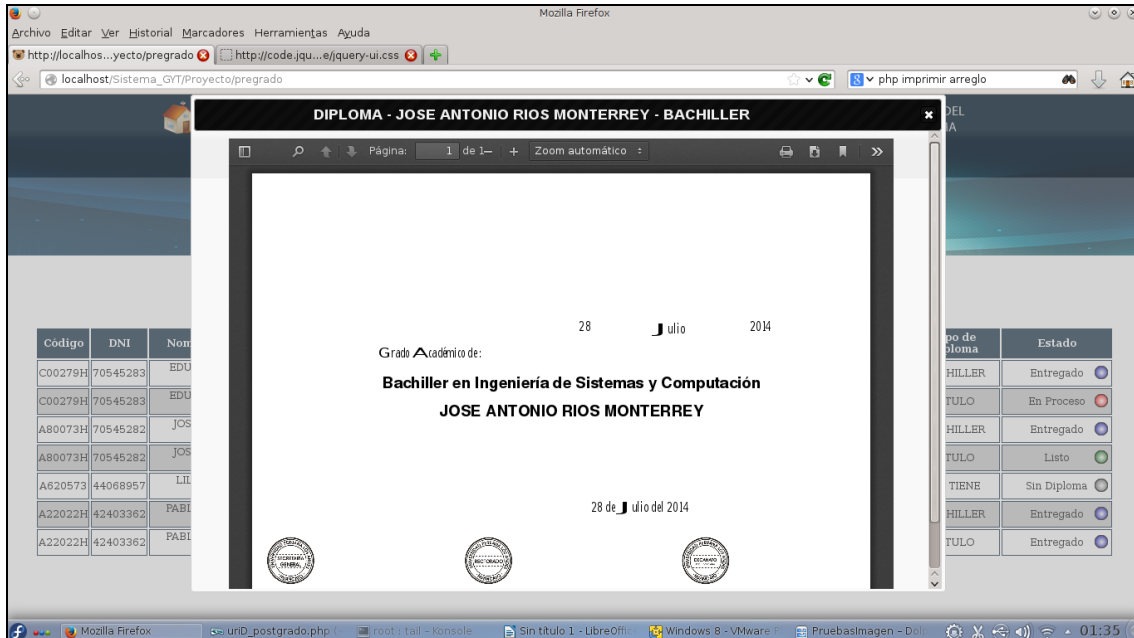


Ilustración 5.19: Salida de "Caso de Prueba N°8"
Fuente: Elaboración propia

5.2. Pruebas de Integración

Para realizar las “pruebas de integración” del framework y llevar a cabo una depuración de su funcionamiento, se realizó su aplicación en el desarrollo de un sistema, basándose en un proceso de la Universidad Peruana Los Andes. En este caso se seleccionó el Proceso de “Asignación de Grados y Títulos” de la Universidad Peruana Los Andes, llevada cabo por la Oficina de “Secretaría General”, mediante su sección de “Grados y Títulos”. Para lograrlo, se revisó documentación al respecto y conversó con el responsable de dicho proceso, el Lic. Ángel Velazco Gonzáles, a partir de lo cual se obtuvieron los requerimientos necesarios para el desarrollo del sistema.

Todas las características, desde la recopilación de requerimientos para este nuevo “Sistema de Grados y Títulos”, hasta el desarrollo final de la aplicación, han sido documentadas e incluidas en el *ANEXO C: Documentación del Sistema de Grados y Títulos (Desarrollado con el Framework BunnyPHP)* de la presente tesis.

A continuación se describen las pruebas realizadas orientadas a probar el cumplimiento de los objetivos. La presente tesis se ha basado en la obtención de un framework, el cual permita el Desarrollo Web con las siguientes cualidades:

- Ágil
- Cooperativo
- Seguro
- Escalable

Por lo tanto, las pruebas fueron las siguientes:

5.2.1. Prueba de “Agilidad en el desarrollo”

El desarrollo web “Ágil” se refiere a la cualidad del framework para permitir la programación rápida y sin mayores complicaciones. Esto implica la utilización de librerías programadas previamente por terceros, las cuáles deben poder ser incluidas en la aplicación de manera rápida y organizada, evitando la reiterada inclusión de estas cuando deben ser utilizadas constantemente.

Al aplicar el framework en el desarrollo del nuevo “Sistema de Grados y Títulos”, este brindó las siguientes bondades:

El framework BunnyPHP posee una arquitectura en la cual, cada tipo de elemento que compone una página HTML (estilos, funciones scripts y la misma página) debe ser colocado en un directorio específico, y a partir de ello ser incluido en el módulo de manera ágil y efectiva.

Por otro lado, la conexión a la base de datos debe estar segura, pero también debe ser accedida fácilmente mediante funciones, para permitir el desarrollo ágil de la aplicación mediante el framework.

Ejemplo 1:

Para implementar el módulo “Pregrado”, se crearon los siguientes elementos:

- “estiloD_pregrado.php” – Archivo de estilos incluido en el directorio “Estilos” de la “Capa de Presentación”.
- “funcD_pregrado.js” – Archivo de scripts que contiene funciones en lenguaje JavaScript que dan soporte al funcionamiento del framework, y está incluido en el directorio “Funciones” de la “Capa de Presentación”.
- modD_pregrado.php” – Archivo HTML que contiene código PHP y la parte básica del módulo, y está incluido en el directorio “Módulos” de la “Capa de Presentación”.

De la misma manera, el framework permite la utilización de plantillas, que son bloques de códigos que serán utilizados reiteradamente en todos los módulos:

- “plantS_menu.php” – Archivo HTML que contiene código PHP y está incluido en el directorio “Plantillas” de la “Capa de Presentación”.

Estos módulos funcionan teniendo como soporte imágenes, las cuáles están incluidas en el directorio “Recursos”, de la “Capa de Presentación”.

Posteriormente se creó el archivo de enrutamiento “uriD_pregrado.php” que contiene el siguiente código:

```

ruta('/pregrado',"GET",function($params)
{
    set_codigo_estado(200);

    seguridad_sesion($params[0]);

    conf_pre();
    echo "<html>";
    echo "<head>";
    //set_recurso("pie.css");

    set_recurso("estiloS_menu.php");
    set_recurso("menu.js");
    set_recurso("funcD_pregrado.js");
    set_recurso("estiloS_general.php");
    set_recurso("estiloD_pregrado.php");

    script('$ ( document
).ready(function() {$("#txt_estudiante").focus();
    minimizar("capa1");
    minimizar("capa2");
    minimizar("capa3");
    menu_lateral();
});');

set_recurso_ext("//code.jquery.com/ui/1.10.4/themes/black-tie/jquery-ui.css","css");

set_recurso_ext("//code.jquery.com/jquery-1.10.2.js","js");

set_recurso_ext("//code.jquery.com/ui/1.10.4/jquery-ui.js","js");

    echo "</head>";
    echo "<body>";
    echo '<div id="popupbox_reporte"></div>';
    echo '<a
href="javascript:mostrar_popupAsignaturas()"><div
id="popupbox_minimizado">Asignaturas
Matriculables</div></a>';

    set_aviso();
    set_cargando();
    incluir_plantilla("plantS_menu.php");
    incluir_modulo("modD_pregrado.php");
    echo "</body>";
    echo "</html>";

});

```

Ilustración 5.20: Código fuente de "uriD_pregrado.php"
Fuente: Elaboración propia

Como se puede observar en el código anterior, existen funciones que permiten la programación ágil. Entre las funciones tenemos:

- **set_recurso()**, la cual es utilizada para incluir, tanto estilos (css o php) como funciones de javascript (js) de manera rápida.
- **set_recurso_ext()**, el cual es similar a “set_recurso()”, con la diferencia de que esta función permite incluir recursos exteriores al proyecto, ingresando como parámetro la URL del recurso.
- **set_avisos()**, utilizada para habilitar el uso de la capa de aviso, la cual es una franja ubicada en la parte superior de la pantalla que permite mostrar avisos en la aplicación.
- **set_cargando()**, utilizada para habilitar el uso de la capa “Cargando” que bloquea toda la interfaz de la aplicación, mientras el framework se encuentra ejecutando algún procedimiento de la Base de Datos.
- **incluir_plantilla()**, utilizada para incluir en el código base del archivo de enrutamiento una plantilla, únicamente con colocar el nombre del archivo.
- **incluir_modulo()**, utilizada para incluir en el código base del archivo de enrutamiento el módulo, únicamente con colocar el nombre del archivo.

Ejemplo 2:

Este ejemplo muestra una parte de uno de los archivos de enrutamiento llamado “uriD_pregrado.php”, el cual no sirve ya para mostrar un módulo como en el ejemplo anterior, sino es utilizado para ejecutar un “procedimiento almacenado” de la Base de Datos.

```

ruta('/entregar_Diploma',"POST",function($params)
{
$dni=isset($_POST['dni']) ? $_POST['dni'] : null ;;
$t_diploma=isset($_POST['t_diploma']) ?
$_POST['t_diploma'] : null ;;
    $car_esp=isset($_POST['car_esp']) ?
$_POST['car_esp'] : null ;;

    $datos_xml='
        <diploma>
            <dni>'.$dni.'</dni>
            <t_diploma>'.$t_diploma.'</t_diploma>
            <car_esp>'.$car_esp.'</car_esp>
        </diploma>';

    $datos=logica::ejecutar_pa('GradosTitulos.[paAct_DiplomaEn
tregado]','json',$datos_xml); });

```

Ilustración 5.21: Código fuente de "uriD_pregrado.php"
Fuente: Elaboración propia

Como se puede observar, inicialmente se obtienen datos mediante “POST”, luego se crea una variable de tipo cadena que almacena una estructura “XML”, y finalmente se ejecuta la función “ejecutar_pa” que permite a la aplicación conectarse con la base de datos y ejecutar un procedimiento almacenado.

5.2.2. Prueba de “Cooperatividad en el desarrollo”

Para dar paso a la exposición de los resultados de esta prueba, es importante recordar que BunnyPHP fue desarrollado orientándose a satisfacer las necesidades de la “Oficina Universitaria de Informáticas y Sistemas” de la “Universidad Peruana Los Andes”. Es por ello que la “Cooperatividad” como una cualidad de desarrollo de este framework es sumamente importante, ya que la Unidad Organizativa mencionada está compuestas por diferentes especialistas, los cuáles deben poder unir sus conocimientos de manera organizada y sinérgica, para desarrollar las aplicaciones de manera óptima y eficiente.

BunnyPHP, para lograr la “Cooperatividad” en el desarrollo de sistema, posee una arquitectura organizada de tal manera que los diferentes elementos que componen una página HTML (estilos, scripts, código PHP) y propiedades propias del framework (archivos de enrutamiento, librerías) están distribuidas cada una en un directorio específico, incluso con un formato de nombre propuesto en la documentación de BunnyPHP, esto junto con la utilización de una herramienta de “Control de Versiones” como “RapidSVN” permite que diferentes programadores puedan estar trabajando sobre la misma aplicación, y lograr un desarrollo cooperativo. A la par, gracias a que BunnyPHP posee una función específica para ejecutar procedimientos almacenados, es posible que mientras un grupo de programadores está trabajando en el desarrollo de la aplicación, en la base de datos se encuentre otro desarrollando los procedimientos almacenados.

Al aplicar este framework en el desarrollo del nuevo sistema de “Grados y Titulos”, se pudo comprobar su “Cooperatividad” al lograr desarrollar a la par dos módulos que componen este sistema. Mientras uno de los programadores desarrollaba el Módulo de “Pregrado” junto con los recursos que le corresponden (estilos y scripts), el otro programador hacía lo propio con el Módulo de “Administración de Resoluciones”. Todo esto se logró utilizando la herramienta “RapidSVN”.



Ilustración 5.22: Interfaz del Módulo de "Pregrado"
Fuente: Elaboración propia

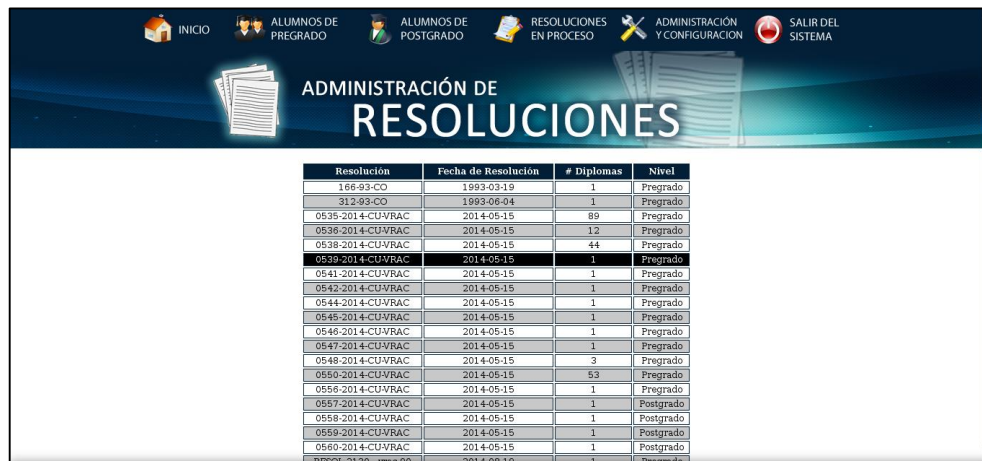


Ilustración 5.23: Interfaz del Módulo de "Administración de Resoluciones"
Fuente: Elaboración propia

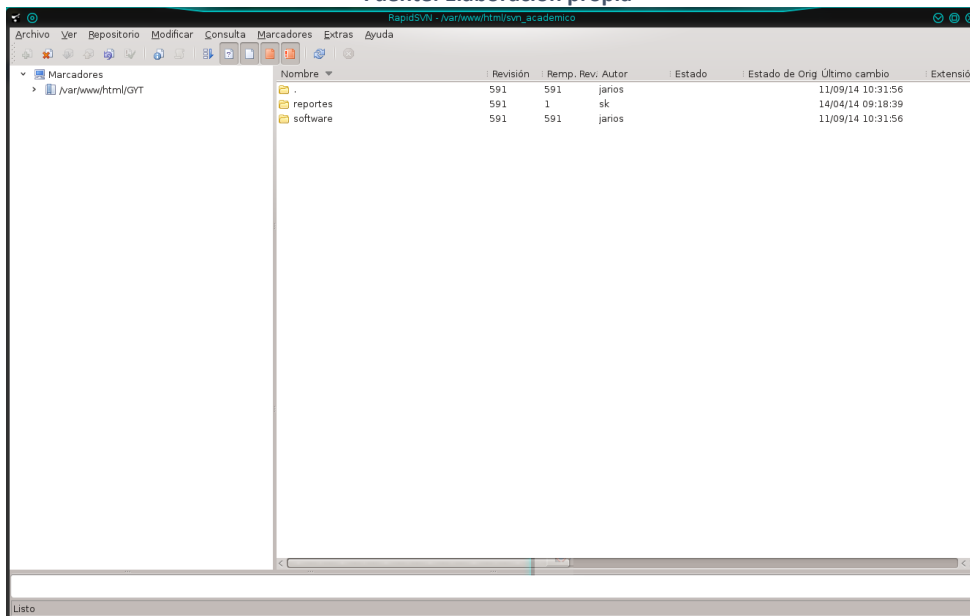


Ilustración 5.24: Captura de Pantalla "RapidSVN en funcionamiento"
Fuente: Elaboración propia

5.2.3. Prueba de la “Seguridad en la aplicación”

Para llevar a cabo las pruebas de “Seguridad” del framework, se utilizó como guía el Top 10 de riesgos de seguridad de OWASP 2013, una referencia confiable de los errores de seguridad más importantes a nivel de aplicaciones.

5.2.3.1. Inyección

Según el Top 10 de OWASP, pueden existir fallas de inyección SQL, OS y LDAP. En lo que respecta al framework, BunnyPHP se encargará únicamente de la inyección SQL ya que abarca únicamente el desarrollo de aplicaciones web. Para poder hacer una “inyección SQL” a una aplicación web y ejecutar de esta manera código malicioso en la Base de Datos, es necesario enviar, en lugar de lo que debería ser un simple dato, código SQL. Para evitar que esto suceda, BunnyPHP tiene una función encargada de borrar todo aquel carácter correspondiente al lenguaje SQL, evitando de esta manera la llegada de código malicioso al Servidor de Base de Datos.

Prueba:

Para interactuar con la Base de Datos, BunnyPHP envía los datos en un XML. La prueba para evaluar si es posible hacer una Inyección a una aplicación desarrollada con BunnyPHP, consiste en habilitar la aplicación para que pueda generar un archivo que contenga el XML que envía datos a la Base de Datos, y evaluar si esos datos pueden contener código SQL si este es enviado desde el formulario.

- Para comenzar, se envía por el formulario un dato normal: “ríos monterrey”, como se muestra a continuación:

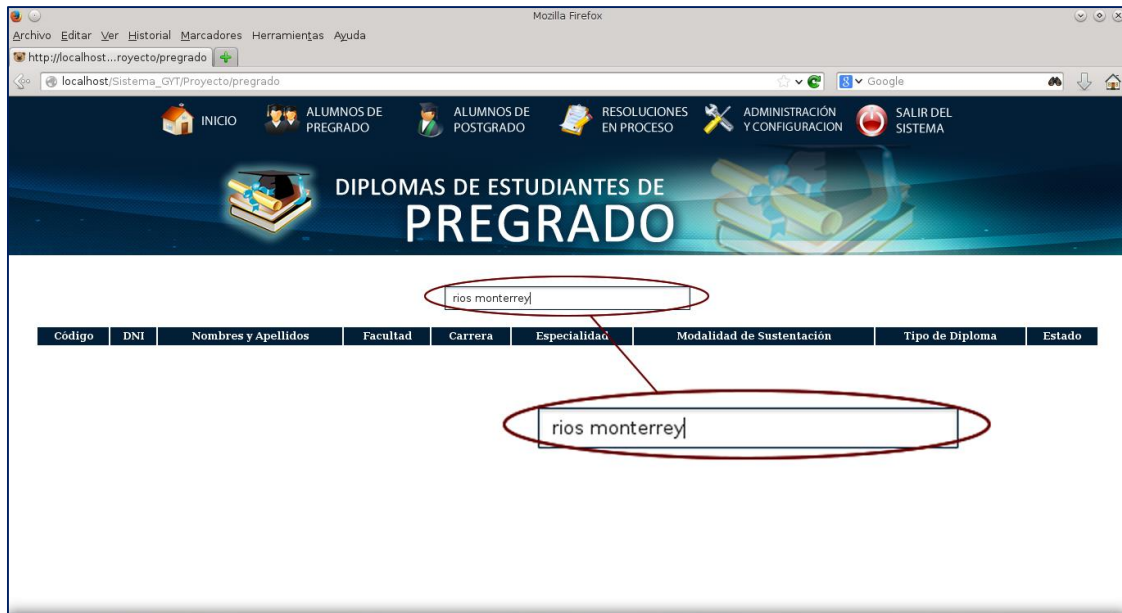


Ilustración 5.25: Prueba de Inyección "Envío normal de un dato"
Fuente: Elaboración propia

- A continuación, se verifica la estructura XML que se obtuvo. Como se puede observar, en el XML se encuentra el dato de manera íntegra, lo cual garantiza que el procedimiento almacenado será ejecutado satisfactoriamente de acuerdo a la petición del usuario.

```

<Busqueda>
  <Buscar>
    <grado>PRE</grado>
    <dni>rios monterrey</dni>
  </Buscar>
</Busqueda>

```

Ilustración 5.26: Modelo de estructura XML enviada a la Base de Datos sin "Ataque de Inyección"
Fuente: Elaboración propia

- Para continuar con la prueba, se envía esta vez un dato que contiene código SQL, el cual simulará el intento de un ataque de inyección SQL. El dato a enviar será: "select into tablita".

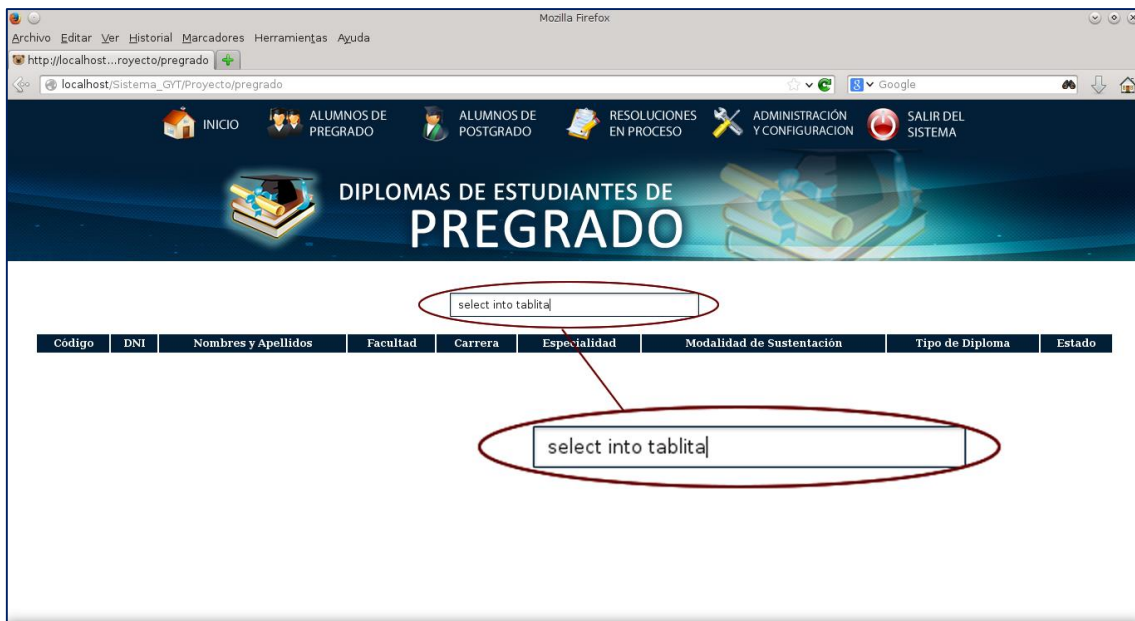


Ilustración 5.27: Prueba de Inyección "Envío de un dato con código SQL"

Fuente: Elaboración propia

- A continuación se verifica la estructura XML que se obtuvo. Como se puede observar, en el XML se encuentra el dato que se envió menos los códigos correspondientes a SQL, lo cual demuestra la imposibilidad de hacer una inyección sobre una aplicación desarrollada con el framework BunnyPHP.

```

<Busqueda>
  <Buscar>
    <grado>PRE</grado>
    <dni>tablita</dni>
  </Buscar>
</Busqueda>

```

Ilustración 5.28: Modelo de estructura XML enviada a la Base de Datos con "Ataque de Inyección" Fuente: Elaboración propia

5.2.3.2. Pérdida de Autenticación y Gestión de Sesiones

A continuación se presenta un cuadro que detalla las pruebas que se le hicieron al framework con respecto a la “Autenticación” y “Gestión de Sesiones”.

DESCRIPCIÓN DE PRUEBA	RESULTADOS	EXPLICACIÓN
Ingresar a algún módulo mediante URL, sin antes haberse logueado (acceder mediante un usuario y contraseña al sistema).	A pesar de haber ingresado el URL correspondiente al módulo, el sistema re direcciona nuevamente a la pantalla de logeo.	El sistema verifica si existe algún “código de acceso” almacenado en el servidor de aplicación correspondiente al navegador del usuario para poder acceder a un módulo diferente del “Login”, si no lo encuentra redirecciona hacia la pantalla de logeo.
Cuando ya se ha logueado, ingresar a un módulo no correspondiente al privilegio de usuario mediante interfaz.	El sistema envía un mensaje de error informando que no se tiene acceso a dicho módulo.	Antes de responder, el sistema verifica si el “código de acceso” corresponde a un usuario con el privilegio necesario y si aún no ha caducado.
Cuando ya se ha logueado, ingresar a un módulo no correspondiente al privilegio de usuario mediante URL.	El sistema envía un mensaje de error informando que no se tiene acceso a dicho módulo.	Antes de responder, el sistema verifica si el “código de acceso” corresponde a un usuario con el privilegio necesario y si aún no ha caducado.
Verificar si es posible ingresar al sistema desde diferentes computadoras con el mismo usuario.	Cuando se intentó ingresar al sistema en otra computadora con el mismo usuario, este ingresó normalmente pero el otro se cerró.	El sistema verifica que el código de acceso almacenado en el servidor correspondiente al navegador sea el mismo que está almacenado en la Base de Datos, si no es así lo interpreta como caducado y genera uno nuevo,

		dejando inválida la sesión en el otro navegador.
--	--	--

Cuadro 5.9: Pruebas de "Pérdida de Autenticación y Gestión de Sesiones"
Fuente: Elaboración propia

Todo esto se debe a que el Framework BunnyPHP gestiona las sesiones creando un código de acceso con tiempo de caducidad en la base de datos, y lo almacena en el servidor de aplicación enlazándolo con el navegador del usuario, toda vez que por cada operación que se realiza, se hace una comparación entre el código almacenado en el servidor de aplicación y en la base de datos, comprobando también si aún no ha caducado, si todo es correcto, el sistema continúa funcionando correctamente y amplía el tiempo de caducidad, en caso contrario elimina el código de la base de datos y del navegador, cerrando finalmente la sesión.

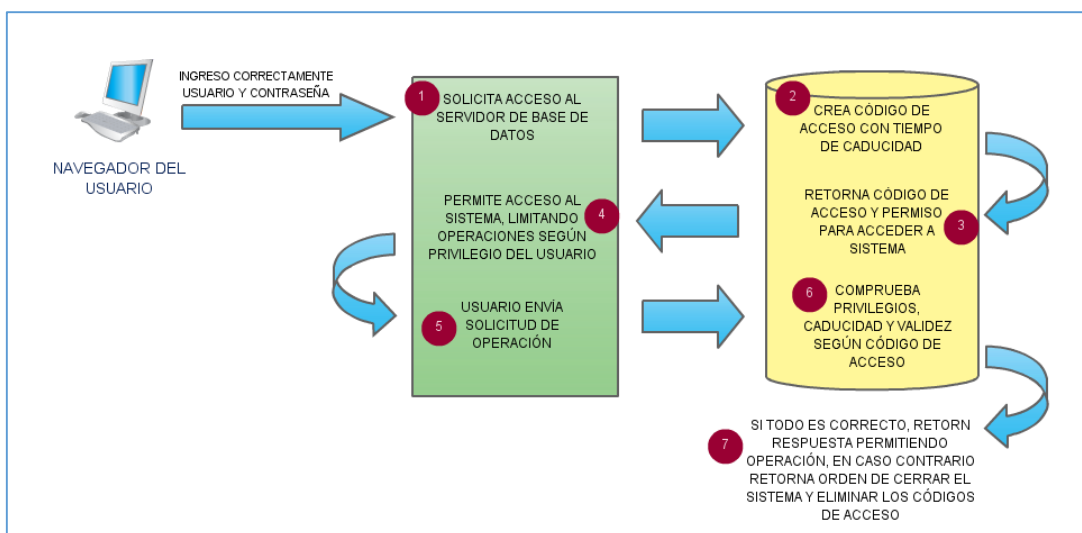


Ilustración 5.29: Diagrama del manejo de "Códigos de Acceso" en el framework
Fuente: Elaboración propia

5.2.3.3. Secuencia de Comandos en Sitios Cruzados

La "Secuencia de Comandos en Sitios Cruzados" (XSS), como ya se explicó en el marco teórico, sucede cuando un usuario mal intencionado envía código malicioso a la aplicación web y se coloca en forma de un hipervínculo para conducir al usuario a otro sitio web, mensajería instantánea o un correo electrónico. Para lograrlo, el usuario debe poder enviar por la URL código malicioso que incluye etiquetas "<script>", "<frame>", entre otras. Para proteger las aplicaciones, BunnyPHP contiene una función que limpia las variables GET y POST de este tipo de código, impidiendo así el XSS.

PRUEBA:

Para evaluar la seguridad del framework contra ataques XSS, se intentará enviar código malicioso por la URL. Esto será lo que se enviará:

“...Sistema_GYT/Proyecto/inicio?smsPermiso=<script>Hola</script>”

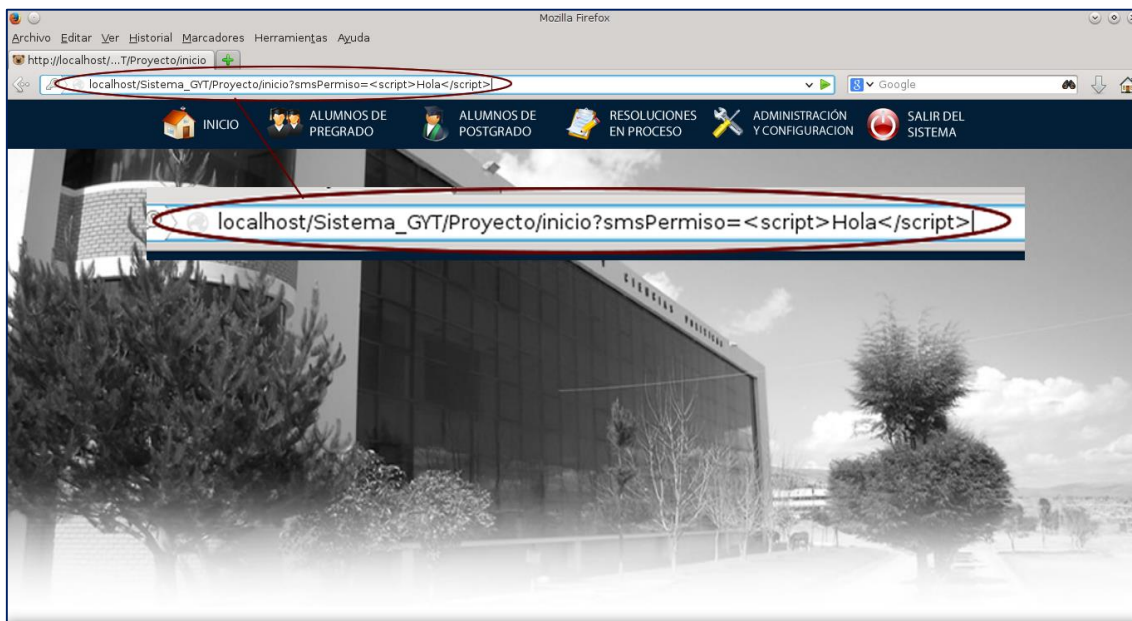


Ilustración 5.30: Prueba ataque XSS "Envío de código malicioso por URL".
Fuente: Elaboración propia

RESULTADO:

El resultado de ejecutar dicho URL, hizo que automáticamente el framework elimine la sesión, regresando a la pantalla de Acceso.

5.2.3.4. Referencia Directa Insegura a Objetos

Este tipo de inseguridad hace referencia a la vulnerabilidad de una aplicación cuando el desarrollador expone una referencia a un objeto de implementación interno como un archivo PHP.

Para comprobar el funcionamiento del framework al respecto, se hicieron las siguientes pruebas:

DESCRIPCIÓN DE PRUEBA	RESULTADOS	EXPLICACIÓN
Ingresar un módulo mediante la interfaz del sistema, y averiguar con ello el nombre de algún	En la URL de la aplicación aparece únicamente el URI (identificador de recursos uniforme)	El framework BunnyPHP, mediante su controlador central (cerebro.php) envía código HTML al

recurso PHP, su ubicación en el servidor o la arquitectura de la aplicación.	del módulo, más no el nombre del archivo, ni mucho menos su distribución o ubicación en el servidor.	navegador bajo URIs que son gestionadas por su sistema de enrutamiento.
Acceder al archivo de conexión que almacena información de acceso a la Base de Datos ingresando su ruta en el URL.	El sistema envía un mensaje de error informando que el archivo no existe.	El framework BunnyPHP está programado de tal manera que devolverá un mensaje de error cuando se intente acceder a archivos protegidos.

Cuadro 5.10: Pruebas de "Referencia Directa Insegura a Objetos"
Fuente: Elaboración propia

A continuación se muestran algunas imágenes que representan las pruebas descritas:

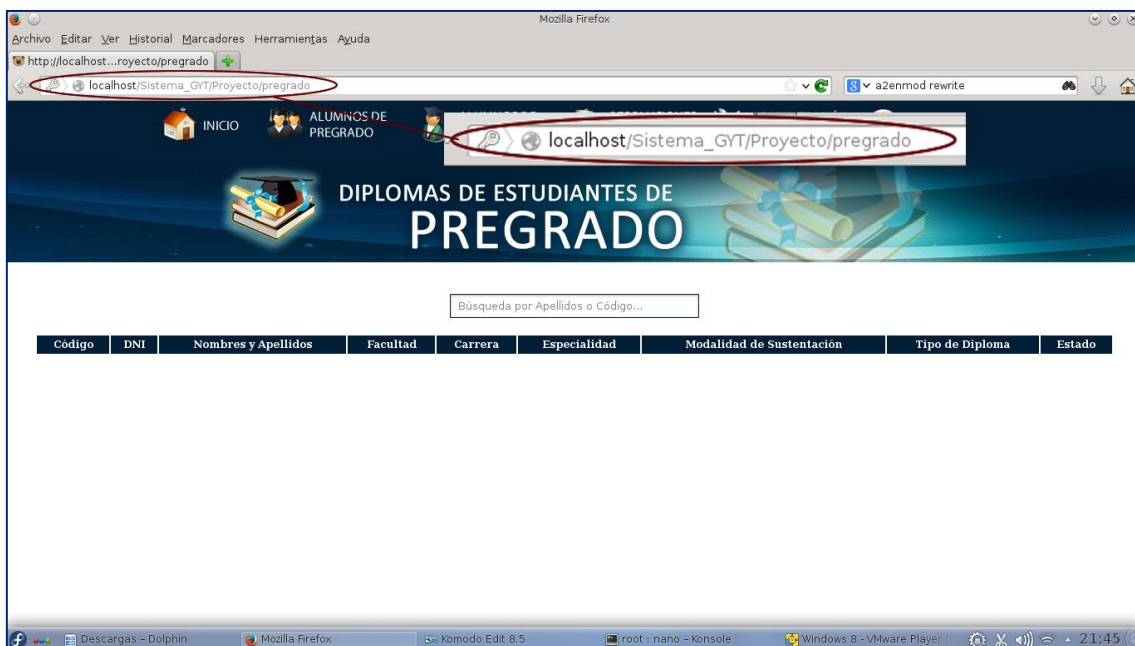


Ilustración 5.31: Captura de Pantalla "La URL no muestra información de archivos PHP"
Fuente: Elaboración propia

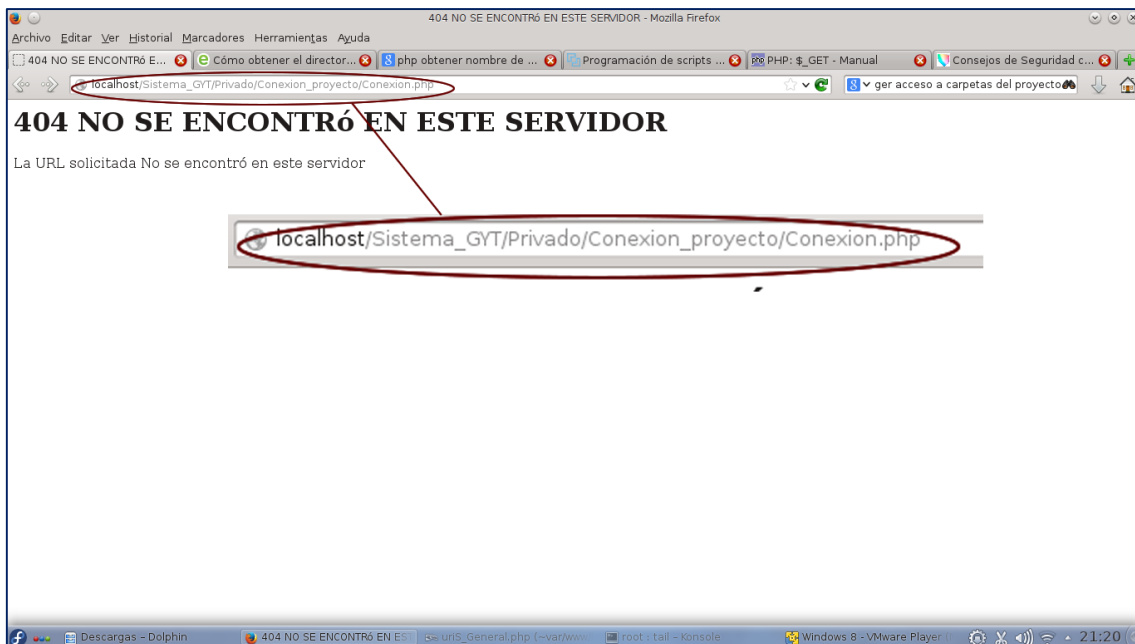


Ilustración 5.32: Captura de Pantalla “No se puede acceder a archivo protegido”
Fuente: Elaboración propia

5.2.3.5. Configuración de Seguridad Incorrecta

El Framework BunnyPHP abarca todo lo que corresponde a la Aplicación misma y la Base de Datos. Como ya se demostró con las pruebas anteriores, en estos puntos el Framework está correctamente configurado. En lo que concierne a la configuración de servidores, dependerá del servicio que contrate y/o configure el desarrollador, pues esto no abarca el framework.

5.2.3.6. Exposición de datos sensibles

Toda aplicación web maneja las contraseñas del usuario, las cuáles son datos sensibles que deben ser protegidas correctamente. BunnyPHP las protege mediante diferentes técnicas que permiten la encriptación completa del dato. Las técnicas que utiliza el framework han sido representadas en dos diagramas expuestos anteriormente. (Ver *Ilustración 4.11: Diagrama de "Sistema de Encriptación - Registro de Nuevo Usuario"* e *Ilustración 4.12: Diagrama de "Sistema de Encriptación - Validación de Acceso de Usuarios"*)

PRUEBA:

Para realizar la prueba correspondiente a este tipo de inseguridad, se procederá a crear un usuario cuya contraseña sea “123456” y se observará, luego, como es almacenado en la base de Datos.

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://localhost...yecto/regUsuario

localhost/Sistema_GYT/Proyecto/regUsuario

INICIO ALUMNOS DE PREGRADO ALUMNOS DE POSTGRADO RESOLUCIONES EN PROCESO ADMINISTRACIÓN Y CONFIGURACIÓN SALIR DEL SISTEMA

ADMINISTRACIÓN Y CONFIGURACIONES

Registrar Usuario

Código o DNI: 20202020

Nombre: Usuario

Ap. Paterno: de Prueba

Ap. Materno: de Framework

Sexo: Masculino

Privilegio: PRACTICANTE DE GRADOS Y TÍTULOS

Contraseña: 123456

Registrar

Ilustración 5.33: Prueba Exposición de Datos Sensibles "Creación de usuario con una contraseña básica"
Fuente: Elaboración propia

RESULTADO:

A continuación se muestra como es almacenada la contraseña en la base de datos.

Windows 8 - VMware Player (Non-commercial use only)

Microsoft SQL Server Management Studio

SQLQuery11.sql - CGF26Jose (54)

```
select * from seguridad.TM_Usuario
```

c_cod_usuario	c_pas_usuario	validez	cookie	c_act_usuario
12345678	0x211E48BE6A380375F5C0F4567EAD2675E3F9469CF	NULL	0	1
20202020	0x4AB9B28BEACC9D2E9CC1BA754E6223F34E6AE6E0	NULL	0	1
44444444	0x7E3478C9F73E028E49B230801F124D7F6AD9A3	NULL	0	1
70312129	0xA1DF581E31613CD4C27AC78C818D1315151682	NULL	0	1
70545282	0x37462318F35816EC462416194E2CA3082100E6	NULL	0	1

Query executed successfully. | 192.168.248.128 (10.50 RTM) | WIN-M7206NCGF26Jose (54) | CampusNet | 00:00:00 | 6 rows

Ilustración 5.34: Prueba Exposición Datos Sensibles "Almacenamiento de contraseña básica en la Base de Datos"
Fuente: Elaboración propia

5.2.3.7. Ausencia de Control de Acceso a Funciones

En el framework BunnyPHP se controla el acceso a los diferentes módulos según el privilegio que tenga asignado el usuario. Las pruebas y la explicación de esto se encuentran en el apartado 5.2.3.2. Pérdida de Autenticación y Gestión de Sesiones.

5.2.3.8. Falsificación de Peticiones en Sitios Cruzados (CSRF)

Como ya se describió en el Marco Teórico, CSRF es un tipo de vulnerabilidad en que la víctima, sin darse cuenta, ejecuta un script en su navegador que permite aprovecharse de la sesión que hubiese iniciado en algún sitio particular.

Para prevenirlo, el framework BunnyPHP utiliza tres mecanismos:

- El uso de URIs, como ya se explicó en el punto 5.2.3.4, impide el uso directo de los archivos PHP que se conectan con la base de datos y ejecutan alguna operación.
- El uso de XML en los procedimientos almacenados obliga que para ejecutar alguna operación en la base de datos, se debe enviar los parámetros en una estructura XML de manera obligatoria. Para lograrlo, se debería conocer de manera precisa las etiquetas y la estructura del XML. En el caso de que se conocieran estas características del XML, las funciones “DefensaHacker_get()” y “DefensaHacker_post()” limpian los caracteres especiales, por lo cual sería imposible enviar una estructura XML mediante GET o POST.
- Por otro lado, BunnyPHP utiliza los “códigos de acceso” descritos en el apartado 5.2.3.2, los mismos que son enviados de manera oculta a la Base de Datos, en la cual se comprueba la autenticidad y la “no caducidad” de estos, siendo imposible para el atacante falsificar una petición.

Como prueba de demostración, se puede considerar lo realizada en el apartado 5.2.3.4.

5.2.3.9. Utilización de Componentes con vulnerabilidades conocidas

Los componentes utilizados son:

- MSSQL (Controlador que permite conexión de PHP con Microsoft SQLServer).
- PHP/Java Bridge (Potente herramienta que nos permite ejecutar e instanciar clases java y ejecutar su código desde PHP, y viceversa).
- jQuery (Librerías programadas en Javascript que facilitan la programación de la interfaz gráfica de las aplicaciones).
- reCaptcha (Librería de Google que permite utilizar su Web Service que implementa un Sistema Captcha).

No se ha reportado ningún tipo de inseguridad para los componentes mencionados.

5.2.3.10.Redirecciones y reenvíos no válidos

Las redirecciones y reenvíos no válidos se refieren a la utilización de esta característica HTTP para redirigir a las víctima a sitios que podrían perjudicarlas. BunnyPHP no permite que el usuario pueda especificar el destino de la redirección mediante el URL, puesto que, como ya se explicó en el punto 5.2.3.3, este framework utiliza funciones que limpian la URL de caracteres especiales, permitiendo únicamente el paso de parámetros válidos. Por otro lado, la inclusión de redirecciones y reenvíos en la programación dependerá únicamente del programador, quien tendrá cuidado de no incluir destinos maliciosos en su código.

5.2.4. Prueba de la “Escalabilidad en la aplicación”

Cuando se habla de “escalabilidad” de aplicaciones, nos referimos a la capacidad en general de estas para adaptarse a los cambios en la “Lógica del Negocio”. Uno de los objetivos de este framework fue lograr el desarrollo de aplicaciones escalables, las cuales puedan soportar el cambio en los procesos propios de la institución, sobre todo tratándose de la Universidad Peruana Los Andes.

Por lo tanto, se planteó como solución que toda la lógica del negocio se desarrolle en la base de datos, utilizando procedimientos almacenados que serán accedidos desde la aplicación de una manera segura, ágil y estandarizada.

A continuación se mencionan las características del framework que permiten que las aplicaciones desarrolladas sean escalables:

- El uso de un Sistema de Enrutamiento, el cual utiliza URIs que protegen la identidad real de los objetos que acceden a los datos de la Base de Datos, separando y organizando los diferentes módulos que se vayan creando, permitiendo la fácil modificación y crecimiento de la aplicación .

- Una arquitectura que separa las funciones script (JavaScript), estilos (CSS) y el código PHP, logrando que interactúen de una manera ágil y escalable. Esta arquitectura permite también la reutilización de librerías que contienen funciones de uso común y constante en los diferentes módulos de la aplicación.

En este caso, hablando de escalabilidad, no es necesario hacer pruebas ya que el cumplimiento de estas características del framework depende del desarrollador, quien se encargará de desarrollar los Procedimientos Almacenados, en los cuales se plasmará la lógica del negocio, y siempre y cuando se encuentren desarrollados de manera escalable, el Framework proporcionará una forma ágil y óptima de utilizar estos procedimientos y cumplir con esta propiedad.

5.3. Pruebas de Aceptación

Para concluir con las pruebas del Framework, se llevan a cabo las “Pruebas de Aceptación” las cuales consisten en verificar la conformidad del cliente. En este caso, los clientes serán los usuarios del framework (Oficina Universitaria de Informática y Sistemas”, y los usuarios finales que utilizarán la aplicación desarrollada mediante este framework (Unidad de Grados y Títulos).

5.3.1. Prueba de Aceptación – “Oficina Universitaria de Informática y Sistemas”

Para comprobar que efectivamente para el cliente, el “Framework” permite un desarrollo “ágil” y “cooperativo” de aplicaciones “seguras” y “escalables”, es necesario que el cliente tenga conocimientos técnicos del tema, por tanto esta “Prueba de Aceptación” es llevada a cabo en la “Oficina Universitaria de Informática y Sistemas”, específicamente a las siguientes personas:

- Ing. Jowel Cabrera Padilla (Jefe de la Oficina de Informática)
- Bach. Rolando Paucar Camargo (Analista - Programador)
- Ing. Arturo Martínez Bravo (Adm. de Base de Datos)

Dichos trabajadores dieron finalmente su conformidad mediante un “Acta de Conformidad de Proyecto”. (Adjunta en los Anexos)

5.3.2. Prueba de Aceptación – “Unidad de Grados y Títulos”

Tomando en cuenta que el “Framework” busca satisfacer las necesidades de su cliente inmediato (trabajadores de la Oficina Universitaria de Informática y Sistemas), los clientes finales serán los usuarios de las aplicaciones web desarrolladas mediante el framework, por tanto es

importante recabar su conformidad mediante la “Prueba de Aceptación” correspondiente.

En este caso se desarrolló una aplicación web llamada “Sistema de Grados y Títulos”, la cual fue desarrollada teniendo como base los requerimientos de la “Unidad de Grados y Títulos”. Para verificar la conformidad se aplicó la “Prueba de Aceptación” a las siguientes personas:

- Lic. Ángel Velazco Gonzales (Responsable de Grados y Títulos)
- S.E. Daysi Veliz Kaque (Secretaria de Grados y Títulos)

Dichos trabajadores dieron finalmente su conformidad mediante un “Acta de Conformidad de Proyecto”. (Adjunta en los Anexos)

Por tanto, el Framework supera satisfactoriamente la “Prueba de Aceptación”.

CONCLUSIONES

- La representación de los “procesos” y los “casos de uso” de la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, mediante la metodología BPM (Business Process Management) y la técnica de entrevistas dirigido al personal de la oficina, nos permitió determinar los requerimientos para el desarrollo del framework.
- El uso de la Metodología de Desarrollo de Software SCRUM, permitió la construcción de un “Framework” para el desarrollo de software en la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, acorde a los requerimientos definidos.
- El Desarrollo de una Aplicación Web mediante el “Framework”, permitió la automatización del “Proceso de Emisión de Diplomas de Grados y Títulos”, y a la vez posibilitó la depuración del “Framework”.
- El Sistema de “Grados y Títulos” desarrollado mediante el “Framework” es seguro y escalable, por tanto su mantenimiento y desarrollo de nuevas funcionalidades será óptimo y eficiente.
- El “Framework” creado permite el desarrollo ágil y cooperativo de aplicaciones web seguras y escalables.

RECOMENDACIONES

- Estandarizar y normar el uso obligatorio del Framework “BunnyPHP” para todas las aplicaciones web que se vayan a desarrollar en la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes” a partir de la fecha, ya que este cumple con los requerimientos establecidos.
- Asociar el uso del Framework “BunnyPHP” con la Metodología Ágil de Desarrollo de Software SCRUM, para el desarrollo de Aplicaciones Web en la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, ya que esto incrementa las características de “agilidad” y “cooperatividad” del framework en el desarrollo de software.
- Asociar el uso del Framework “BunnyPHP” con la Metodología Ágil de Desarrollo de Software SCRUM, para el desarrollo de Aplicaciones Web en la “Oficina Universitaria de Informática y Sistemas” de la “Universidad Peruana Los Andes”, ya que esto incrementa las características de “agilidad” y “cooperatividad” del framework en el desarrollo de software.
- Utilizar las características del Framework “BunnyPHP” al momento de realizar un mantenimiento o desarrollo de nuevas funcionalidades del “Sistema de Grados y Títulos”, de esta manera se conservarán sus propiedades de escalabilidad y seguridad.

BIBLIOGRAFÍA

- Consejo Universitario de la Universidad Peruana Los Andes (2013). *Estatuto 2013*, Huancayo – Perú. Universidad Peruana Los Andes.
- Fraga, M. (s.f.). Metodología de Trabajo Ágil y Eficiente: El método Scrum aplicado a la Gestión de Proyectos en general: Tema 3-Scrum: *Roles y Procesos*, Madrid – España. Fundación Antonio de Nebrija.
- Garimella, K; Lees, M; Williams, B. (2008). *Introducción a BPM para Dummies*, USA. Wiley Publishing Inc.
- Hundermark, P (2009). *Measuring for Results: Metrics and Myths*. Consultado el 23 de Abril del 2014. Disponible en: <http://www.scrumsense.com/wp-content/uploads/2009/10/Measuring-for-Results-2-small.pdf>
- Palacio, J (2014). *Gestión de proyectos Scrum Manager (Scrum Manager I y II)* v. 2.5. Consultado el 20 de Abril. Disponible en: http://www.scrummanager.net/files/sm_proyecto.pdf
- Schwaber, K; Sutherland, J (2013). *La guía definitiva de Scrum: Las reglas del juego*. Consultado el 20 de Abril del 2014. Disponible en https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide%202011%20-%20ES.pdf
- The Open Web Application Security Project OWASP (2013). *OWASP Top 10 – 2013 “Los diez riesgos más críticos en Aplicaciones Web”*. Consultado el 15 de Marzo. Disponible en: https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf
- White S.A.; Miers, D. (2009). *BPMN – Guía de Referencia y Modelado*, Florida – USA. Future Strategies Inc.
- Fonseca, P. (2014, Enero) *Desarrollo de un Framework Web para el envío remoto de tareas, monitoreo y recuperación de resultados para desktop Grids usando una*

Arquitectura orientada a servicios: caso BOINC. Consultado el 21 de Marzo del 2014. Disponible en:

http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5337/FONSECA_PABLO_FRAMEWORK_WEB_DESKTOP_GRIDS_ARQUITECTURA_ORIENTADA_SERVICIOS_CASO_BOINC.pdf?sequence=1

Romero, R. (2012) *Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial*. Consultado el 21 de Marzo del 2014. Disponible en:

http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1562/ROMERO_GALINDO_RAUL_SISTEMA_INFORMACION_EDUCACION_ESPECIAL.pdf?sequence=1

Arpasi, L., & Torres, B. (2007) *Framework de Seguridad*. Consultado el 22 de Marzo del 2014. Disponible en:

<http://tesislatioamericanas.info/index.php/record/view/34091>

Roberto, E. (2010, Febrero) *Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST*. Consultado el 24 de Marzo del 2014. Disponible en http://www.ub.edu.ar/investigaciones/tesinas/354_Tesina_Zulian.pdf

Galindo, J., & Camps, J. (2008, Enero) *Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE*. Consultado el 25 de marzo del 2014. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf>

Navarro, A. (2013, Abril) *Estudio e implementación de un framework de desarrollo de aplicaciones con funciones de seguridad y privacidad para móviles*. Extraído de <http://upcommons.upc.edu/pfc/bitstream/2099.1/18253/1/85737.pdf>

ANEXOS

ANEXO A: Entrevista a los trabajadores de la Oficina Universitaria de Informática y Sistemas

Entrevista a los trabajadores de la Oficina Universitaria de Informática y Sistemas

Se realizaron entrevistas a trabajadores específicos y de manera estratégica para obtener información acerca de la forma de desarrollo de software de la oficina. Los trabajadores entrevistados fueron:

- Bach. Irwin Lizandro Yauri Orihuela (Analista – Programador de Sistemas).
- Ing. Arturo Martínez Bravo (Administrador de Base de Datos).
- Ing. Jowel Cabrera Padilla (Jefe de la Oficina Universitaria de Informática y Sistemas).

A continuación se incluyen las entrevistas realizadas:

ENTREVISTA N°1:



ENTREVISTADO:

Bach/Ing. Irwin Lizandro Yauri Orihuela – Analista Programador de Sistemas de la Oficina Universitaria de Informática y Sistemas

ENTREVISTADORES:

- Emily Rossy Salinas Chamorro
- José Antonio Rios Monterrey

LUGAR: Oficina Universitaria de Informática y Sistemas – Unidad de Desarrollo de Proyectos.

DÍA Y HORA: El 10 de Abril del 2014, a las 6:15 p.m.

MODALIDAD: Se cumplió con la reunión pactada en la Oficina Universitaria de Informática y Sistemas, luego de un cordial saludo se prosiguió a realizar la entrevista dirigida al Bach. Irwin Yauri. La entrevista se registró en un archivo de audio, y posteriormente se transcribió en el presente documento.

CUERPO DE LA ENTREVISTA:

Buenas noches, en este momento nos encontramos en las instalaciones de la Oficina Universitaria de Informática y Sistemas, en la unidad de Desarrollo de Proyectos, en este momento se encuentra con nosotros el Bach/Ing. Irwin Lizandro Yauri Orihuela, Analista – Programador de Sistemas. Los entrevistadores somos Emily Rossy Salinas Chamorro y José Antonio Ríos Monterrey. La presente reunión tiene como objetivo realizar una entrevista al Bach/Ing. Irwin Yauri para recopilar recomendaciones y requerimientos con respecto a la tesis que estamos desarrollando, la cual consiste en la “Creación y Aplicación de un Framework para el desarrollo web ágil, cooperativo, seguro y escalable, orientado a la Oficina Universitaria de Informática y Sistemas de la Universidad Peruana los Andes”.

Pregunta: ¿Cuáles son las funciones más importantes que debe cumplir la Unidad de Desarrollo de Proyectos de la Oficina Universitaria de Informática y Sistemas?

Según el estatuto 2012 de la Universidad Peruana los Andes, los servicios principales son:

El servicio de implementación de sistemas informáticos, servicio de soporte y mantenimiento, servicio de Implementación de hardware y software, servicio de internet y comunicaciones, servicio de administración del portal web de la universidad, servicio de disposición de normas y uso correcto de las TIC's.

Pero exclusivamente el área está dividida en dos, una es la “Unidad de Desarrollo” y la otra de “Soporte y Mantenimiento”. En la unidad de desarrollo nos ocupamos de dar mantenimiento y soporte de datos, también de satisfacer requerimientos de software, además del desarrollo de proyectos informáticos.

Pregunta: Y específicamente cuando hablamos de “Soporte de Datos”, ¿A qué se refiere?

Se refiere al análisis de la información que se encuentra almacenada en nuestra base de datos, con el fin de depurar la información que tenemos, sobre todo si hablamos de los dos grandes sistemas que son el Sistema Académico y el Sistema Financiero, con el fin de integrarlos ya que estos sistemas se han desarrollado de una forma empírica, sin ninguna planificación previa, en consecuencia, se tiene esta función de dar soporte y mantenimiento a los datos.

Pregunta: En todo caso, ¿Cuál hubiera sido la manera correcta de desarrollar esos sistemas?

Bueno, los sistemas, tanto el Académico como el Financiero, debieron ser planificados previamente, analizados, haber hecho por lo menos un “Diagrama de Procesos” o haberse basado en alguna metodología de desarrollo, pero al parecer los anteriores trabajadores no vieron por conveniente eso.

Pregunta: ¿Desde qué fecha aproximadamente cumple usted el cargo de Analista Programador de Sistemas en la oficina, y cuáles son las dificultades que halló al comenzar su labor, hablando sobre todo de los sistemas informáticos?

Bueno, empecé a trabajar en septiembre del 2012, al inicio no se contaba con una estructura de datos definida, no habían procesos definidos, existía un desinterés para con uno a la hora de solicitar un apoyo para interpretar la estructura de base de datos; muy parte, no se trabajaba con software libre de ninguna índole, tanto en los servidores, como en el desarrollo de aplicaciones y en el mantenimiento de datos, esto más la falta de documentación y por último, que gran parte de los sistemas estaban hechos en Fox. Hecho que nos desfavorece mucho porque en la actualidad se ha dejado de lado el Fox, por su seguridad, y por su difícil implementación, pues es tedioso e inseguro.

Pregunta: ¿Cuál es el sistema que más dificultades ha traído a la Unidad de Desarrollo de Proyectos, y cuáles son las características que hacen que usted lo defina como tal?

En mi opinión, el sistema con más dificultad vendría a ser el Sistema Académico, pues este sistema se ha desarrollado empíricamente, sin ninguna metodología, ni análisis de procesos bajo ningún estándar. Está hecho en .Net, lo cual no está mal, pero tiene código duro. Es inseguro y no solamente por el lado de la aplicación, también por el lado de la Base de Datos. Está mal estructurado y se ha tenido que hacer muchos cambios.

Pregunta: ¿Y a qué se refiere cuando Ud. dice código duro?

Hay código que debería ser dinámico, debería poder ser cambiado de una manera ágil, debería ser escalable, pero el sistema académico tiene en su código “contraseñas”, “permisos, si uno desea hacer un cambio tiene que crear un formulario nuevo, bueno, de esta manera se ha estado dando mantenimiento casi siempre; por ello el Jefe de Informática está tomando la decisión de poder reformular y reestructurar el sistema académico.

Pregunta: ¿Lo que Ud. quiere decir es que, para que el Sistema Académico funcione, se tiene que hacer una serie de “parches” en el mismo código?

Sí.

Pregunta: Según la planificación de la jefatura, ¿Cuáles son los sistemas que se han previsto desarrollar o están en desarrollo a partir de la fecha?

Los sistemas que se han previsto desarrollar son, en primer lugar la reestructuración del sistema académico, en segundo lugar están algunos sistemas más pequeños como el sistema de caja y policlínico, también se tiene previsto desarrollar un módulo para el sistema de admisión, para ver los resultados del examen de admisión.

Pregunta: Sr. Irwin, nosotros, como parte del desarrollo de nuestra tesis para optar el título profesional de “Ingeniero de Sistemas y Computación”, hemos planteado la creación de un framework, el cual debe ser ágil, cooperativo, seguro y escalable. A

partir de su experiencia y de sus conocimientos, ¿Cuáles deberían ser las características del framework?

Claro, está bien, debe ser ágil, la forma de desarrollo debe ser correctamente diagramado, fácil de implementar, no debe tener código innecesario, debe ser de código abierto ya que si yo deseo utilizar una librería que no está en la arquitectura del framework, pueda yo implementarlo y utilizarlo, además que pueda combinar el conocimiento que Uds. hayan puesto en su framework con la librería que yo desee, debe ser seguro, debe tener códigos de acceso, una sesión segura, llaves encriptadas, debe tener URLs limpias y ser seguro contra ataques, tanto de índole de Get, Post, código script, código PHP, HTML, inyección, debe ser escalable.

Pregunta: ¿Qué opina Ud. de la utilización de software libre en el desarrollo web?

Cuando ingresé, como les comenté, estaba en la oficina de informática y se usaba .Net y como Sistema Gestor de Base de Datos se usaba el SQL Server, pero el jefe de la oficina vio por conveniente migrar a software libre para el desarrollo de aplicaciones, ahora se usa PHP ya que es más ligero, escalable y fácil de administrar; en tema de servidores, por ejemplo, se optó por usar CentOS, ya que nos permite aprovechar de manera óptima los recursos del servidor, no es tan sencillo de dar mantenimiento pero con una buena documentación nos ha funcionado bien la parte de seguridad, es súper seguro, más seguro que Windows y con la experiencia hemos desarrollado un sistema de Grados y Títulos que hasta el momento nos está funcionando bien, es escalable, salvo unas cosas pero el resto es seguro. Y en el tema de Sistema Gestor de Base de Datos, no hemos cambiado por MySql o PostgreSQL porque la licencia ya ha sido comprada.

Pregunta: Entonces, ¿La recomendación implícita que Ud. tiene es la utilización del software libre en el desarrollo de este framework?

Claro, pues un framework debe ser de “código abierto” para poderle implementar más características, y aparte, no contener código innecesario ya que hace pesada la aplicación, algo que sucede con el .NET que con su framework ha querido dar mayor facilidad al programador, pero al final genera código innecesario.

Pregunta: ¿En todo caso que tecnologías recomendaría Ud. para el desarrollo web?

Para el desarrollo web, recomiendo utilizar básicamente la librería jQuery, para manejar el JavaScript, recomiendo usar en consecuencia también Ajax, que se utilice PHP de base como motor web pues es sencillo de aprender, es sencillo de implementar y tiene soporte para hacer conexión con SQL.

Sr. Irwin, estamos muy agradecidos por su aporte a nuestra investigación, esperamos contar con su apoyo en lo subsiguiente del desarrollo de la presente tesis y le deseamos éxitos tanto en su gestión, su vida profesional y su vida personal. Muchas gracias.

ENTREVISTA N°2:**ENTREVISTADO:**

Ing. Arturo Martínez Bravo – Administrador de Base de Datos de la Oficina Universitaria de Informática y Sistemas

ENTREVISTADORES:

- Emily Rossy Salinas Chamorro
- José Antonio Rios Monterrey

LUGAR: Oficina Universitaria de Informática y Sistemas – Unidad de Desarrollo de Proyectos.

DÍA Y HORA: El 16 de Abril del 2014, a las 7:30 p.m.

MODALIDAD: Se cumplió con la reunión pactada en la Oficina Universitaria de Informática y Sistemas, luego de un cordial saludo se prosiguió a realizar la entrevista dirigida al Ingeniero Arturo. La entrevista se registró en un archivo de audio, y posteriormente se transcribió en el presente documento.

CUERPO DE LA ENTREVISTA:

Buenas noches, en este momento nos encontramos en las instalaciones de la Oficina Universitaria de Informática y Sistemas, en la unidad de Desarrollo de Proyectos. En este momento se encuentra con nosotros el Ing. Arturo Martínez Bravo, Administrador de Base de Datos. Los entrevistadores somos Emily Rossy Salinas Chamorro y José Antonio Rios Monterrey. La presente reunión tiene como objetivo realizar una entrevista al Ing. Arturo Martínez para recopilar recomendaciones y requerimientos con respecto a la tesis

que estamos desarrollando, la cual consiste en la “Creación y Aplicación de un Framework para el desarrollo web ágil, cooperativo, seguro y escalable, orientado a la Oficina Universitaria de Informática y Sistemas de la Universidad Peruana los Andes”.

Pregunta: ¿Cuáles son las funciones más importantes que debe cumplir la Unidad de Desarrollo de Proyectos de la Oficina Universitaria de Informática y Sistemas?

Como su nombre lo indica, la unidad de desarrollo de proyectos debe enfocarse principalmente a crear los sistemas que necesita la universidad para que pueda funcionar adecuadamente.

Pregunta: Esta unidad de desarrollo, ¿Desde cuándo existe o siempre ha existido?

La unidad de desarrollo ya tiene funcionando aproximadamente 1 año; digamos que el área de informática ha visto por conveniente tener 2 áreas, una unidad de desarrollo de proyectos y la otra unidad es de mantenimiento.

Pregunta: ¿En realidad, actualmente la unidad de desarrollo está cumpliendo la función que se le ha asignado o hay algún problema por la cual no puedan cumplir?

Están cumpliendo con parte de los requerimientos del área de usuario, no en total debido a la falta de personal que la unidad de desarrollo de proyectos necesita; como por ejemplo un analista de sistemas, programadores, analistas programadores, necesitan también de lo que es netamente un administrador de base de datos, como la función que estoy cumpliendo, especialistas en programas, también se necesita que tenga un jefe de proyecto y testing para el control de calidad.

Pregunta: ¿Cuáles son las funciones que cumple usted como Administrador de Base de Datos?

Las funciones principales que se está cumpliendo son de resguardar la información, el otro es hacer la optimización en base al funcionamiento del sistema que se llama optimus y una de ellas es básicamente la lógica de programación que está en la base de datos. Otra de las funciones también que tengo encomendada es la seguridad, que incluye seguridad de usuarios, mayormente tener una política de resguardar la información ya que estamos expuestos a que cualquiera pueda, ingresar, modificar y nadie se entera, entonces es una de las tareas principales, otro es el resguardo de la información en el ámbito físico. Se plantea como objetivo en este año realizar ese tipo de proyectos ya que se necesita también que el administrador de base de datos conozca el manejo de servidores, pues en la oficina adolecemos de eso.

Pregunta: ¿Existe algún tipo de política ya planteada y que se esté utilizando actualmente?

Todavía no se tiene plasmado en documentos ya que estamos cumpliendo múltiples funciones en el área, por ejemplo, yo estoy como administrador de base de datos, pero a la vez estoy haciendo o me dedico a hacer mantenimiento y a cumplir los requerimientos de los usuarios, también estoy como programador, como analista, a veces como jefe de

proyectos, hago múltiples roles, por ello no hago al 100% las funciones del administrador de la base de datos.

Pregunta: ¿Ingeniero, tenemos alguna información de que anteriormente han habido ataques a la base de datos e incluso hubo modificación de datos, Ud. está enterado de algo y si así fue, que es lo que se hizo para evitar eso?

La información no la tenemos segura aún al 100%. Si hemos tenido un caso en el que habían ingresado maliciosamente a la base de datos, y en coordinación con el área de redes se colocó un filtro cortafuegos para nuestro servidor, para evitar ataques, también se han superado políticas dentro de la misma oficina, se controla el acceso de las personas pues anteriormente no habían políticas y se daba indistintamente el acceso a cualquier persona que ingresaba a trabajar a la oficina, prácticamente era como confiar en la buena fe de las personas y eran varios que modificaban los datos, entonces fue por ello que para tener ese control se ha tomado la política de que solo el administrador de base de datos y en este caso, un jefe de proyectos puedan manejar el password para restringir el acceso al total del personal de informática.

Pregunta: En todo caso, ¿Eso quiere decir que actualmente se cuenta con mucha mayor seguridad?

Se ha avanzado en tema de seguridad y como le comentaba el tema de la falta de personal hace que no nos aboquemos al 100% todavía.

Pregunta: ¿Desde qué fecha aproximadamente cumple usted el cargo de Administrador de Base de Datos en la oficina, y cuáles son las dificultades que halló al comenzar su labor, hablando sobre todo de los sistemas informáticos?

Inicie aproximadamente hace 2 años y el tema es que en la institución no había ese rol, dado a la forma como se trabajaba en la oficina, le hice el planteamiento al jefe de cómo se podría trabajar, ya que yo mismo estaba cumpliendo el rol de dar soporte técnico a la programación de la base de datos y el tema de seguridad que era también propio del administrador de base de datos, entonces fue así como se forma ese rol y a partir de entonces se crea el puesto de DBA.

Pregunta: ¿Cuál es el sistema que más dificultades ha traído a la Unidad de Desarrollo de Proyectos, y cuáles son las características que hacen que usted lo defina como tal?

La cantidad de personal es uno de ellos y que no se cuenta con especialistas.

Pregunta: ¿Y qué sistema es el que más dificultades ha traído?

Son el sistema académico y el sistema financiero.

Pregunta: ¿Y cuáles son las características de estos sistemas para que se haya tenido esas dificultades?

La forma en la que fueron construidos, no ha habido un arquitecto de software que indique la manera correcta de desarrollar, entonces lo construyeron de forma empírica y el sistema

fue creciendo. Luego, recién se dieron cuenta de que a la hora de hacer el mantenimiento se necesitaba una cantidad considerable de personal, lo que hasta hoy está incurriendo en costos.

Pregunta: ¿Según usted, cuál sería la manera correcta de estructurar una base de datos para el desarrollo de un sistema para la Universidad Peruana Los Andes?

En principio, que los usuarios conozcan de “Gestión por Procesos”, por ejemplo la metodología BPM. De esta manera puedan precisar de una manera más eficiente sus requerimientos y lograr que estos no sean tan cambiantes.

Pregunta: Sabemos que actualmente la Oficina Universitaria de Informática y Sistemas utiliza como principal Sistema Gestor de Base de Datos al “Microsoft SQL Server”, sin embargo existe una idea de utilizar sobre todo Software Libre. ¿Cuál es la razón por la cual se utiliza este motor, y cual son sus beneficios sobre motores como MYSQL o POSTGRE SQL que son libres?

Se escogió el motor de SQL Server dado a que organizaciones, bancos, actualmente utilizan este Sistema Gestor de Base de Datos, en el mercado tiene un porcentaje alto y en nuestro medio que es la región Junín, son muchas las instituciones como el Banco de Crédito, la Caja Municipal de Huancayo.

Las ventajas que tiene en comparación al software libre es que como somos una institución educativa, el costo de licenciamiento frente a un banco es cinco veces menor, entonces es importante aprovechar eso, pues si nosotros elegimos software libre; este es hasta cierto punto conveniente a nivel de en costo beneficio, pero al implementarse teniendo en cuenta el tema de seguridad, ya tiene otro costo, entonces haciendo la comparación con SQL Server nos saldría más caro utilizar PostgreSQL. El tema de MySQL no lo utilizamos porque al ser la UPLA una institución grande, tenemos bastante información que MySQL no podría soportar.

Pregunta: Ing. Arturo, nosotros, como parte del desarrollo de nuestra tesis para optar el título profesional de “Ingeniero de Sistemas y Computación”, hemos planteado la creación de un framework, el cual debe ser ágil, cooperativo, seguro y escalable. ¿Cree usted que, si bien se está creando un framework para el desarrollo web de sistemas orientados a la Universidad Peruana Los Andes, este debería incluir una estructura de base de datos, al menos con respecto a la administración de usuarios y privilegios de acceso?

En el tema de armar una arquitectura, tener un framework es esencial para que se pueda hacer cualquier sistema, para poder dar mantenimiento y como institución. Es necesario porque a través de una arquitectura estándar, prácticamente vamos a poder hacer mantenimiento al sistema y de acuerdo a cada uno de sus roles, por ejemplo, el tema de tener la seguridad prácticamente entre desarrollador y administrador de base de datos, ya se va a poder hablar en un mismo lenguaje al solucionar los inconvenientes que tenemos.

Pregunta: En todo caso, como se trata de un framework, no para el uso general sino específicamente para la institución, en este caso la Universidad Peruana Los Andes, ¿Se puede estructurar un diagrama de base de datos para que pueda ser aplicado únicamente a la universidad?

Sí, para el área educativa de la universidad, tener la estructura bien plasmada desde el principio es importante.

Pregunta: Ing. Arturo, estamos muy agradecidos por la información brindada de su parte, ¿Hay algo más que quiera agregar a la entrevista?

Bueno, indicarles, básicamente que el rol de administrador de la base de datos tiene mucho más por aportar, el tema de que a veces miramos que en desarrollar un sistema, a veces vemos que necesitamos de analistas, de programadores y no tenemos básicamente una forma de hacer las cosas. En el rol de administrador de base de datos existen herramientas que ayudan también, existen herramientas para hacer pruebas, también existen para tener más profesionalismo en manejar la información. Una de las cosas que se desea plasmar ahora en el área de desarrolladores es tener nuestro servidor de prueba, para el administrador de base de datos es esencial tener eso para que puedan ellos controlar la información que envían al servidor de producción, entonces tendríamos que implementar un servidor de desarrollo, un servidor de pruebas y un servidor de producción.

Ing. Arturo, estamos muy agradecidos por su aporte a nuestra investigación, esperamos contar con su apoyo en lo subsiguiente del desarrollo de la presente tesis y le deseamos éxitos tanto en su labor, su vida profesional y su vida personal. Muchas gracias.

ENTREVISTA N°3:

ENTREVISTADO: Ing. Jowel Cabrera Padilla – Jefe de la Oficina Universitaria de Informática y Sistemas

ENTREVISTADORES:

- Emily Rossy Salinas Chamorro
- José Antonio Rios Monterrey

LUGAR: Oficina Universitaria de Informática y Sistemas – Unidad de Desarrollo de Proyectos.

DÍA Y HORA: El 16 de Abril del 2014, a las 6:30 p.m.

MODALIDAD: Se cumplió con la reunión pactada en la Oficina Universitaria de Informática y Sistemas, luego de un cordial saludo se prosiguió a realizar la entrevista dirigida al Jefe de la oficina. La entrevista se registró en un archivo de audio, y posteriormente se transcribió en el presente documento.

CUERPO DE LA ENTREVISTA:

Buenas tardes, en este momento nos encontramos en las instalaciones de la Oficina Universitaria de Informática y Sistemas, en este momento se encuentra con nosotros el Ing. Jowel Cabrera Padilla, jefe de la oficina. Los entrevistadores somos Emily Rossy Salinas Chamorro y José Antonio Rios Monterrey. La presente reunión tiene como objetivo realizar una entrevista al Ing. Jowel para recopilar recomendaciones y requerimientos con respecto a la tesis que estamos desarrollando, la cual consiste en la “Creación y Aplicación de un Framework para el desarrollo web ágil, cooperativo, seguro

y escalable, orientado a la Oficina Universitaria de Informática y Sistemas de la Universidad Peruana los Andes”.

Pregunta: ¿Cuáles son las funciones más importantes que debe cumplir la Oficina Universitaria de Informática y Sistemas?

Las funciones establecidas de la oficina son las siguientes, en primer lugar planificar, plantear, hacer que se implementen tecnologías de información en la universidad para dar soporte a todos los procesos que se realizan. En segundo lugar es: proponer y desarrollar los sistemas de información, o en caso de que no se tenga la disponibilidad de recursos humanos, mandarlo a desarrollar por terceros. Otra de las funciones es velar por la normatividad de uso y maneje de tecnologías de información dentro de la universidad, luego, también es tener operativo todas las tecnologías de información implementadas en la universidad.

Pregunta: ¿Desde qué año se encuentra usted a cargo de la oficina, y cuáles son las dificultades que halló al comenzar su gestión, hablando sobre todo de los sistemas informáticos?

Yo me hice cargo en Agosto del 2010. Bueno, la situación en que encontré la oficina es que la gestión, especialmente de lo que son los sistemas de información, no está hecha bajo ninguna norma, entonces, comenzando desde que nosotros encontramos los aplicativos desarrollados sin ninguna documentación, algunos de los aplicativos sólo tenían el código ejecutable, no el código fuente. Eso es en la parte de software, ahora, en la parte de hardware, la sala de servidores, prácticamente no estaba correlacionado con ese concepto, encontramos los servidores, algunos hasta en el piso o sobre alguna mesa o escritorio, entonces poco a poco tuvimos que ir ordenando, y todavía está en proceso de ordenar todo para que de pronto pueda caber en la dimensión de esa palabra “Sala de Servidores”, pues la última gestión que se está haciendo es implementar el sistema de enfriamiento, y luego dotarla de la seguridad necesaria para que sólo personal autorizado pueda acceder a la sala de servidores.

Pregunta: Ingeniero, ¿Y cómo hicieron frente, el personal que estaba a su cargo y usted para poner operativos los sistemas que no tenían documentación y solamente eran ejecutables, e incluso para hacer el soporte respectivo?

Bien, inicialmente mantuve al personal que estaba trabajando, que por estar trabajando ya tiempo conocían como funcionaba, y entonces podían ellos, de alguna manera darle el soporte necesario, pero mantenían también las viejas costumbres de cómo se manejaba, donde realmente no se hacían las funciones que debería tener una unidad de TI. Prácticamente, informática era un tramitador de las oficinas como de Asuntos Académico o, en este caso la Oficina de Desarrollo Académico, o lo que es ahora la Dirección Universitaria de Desarrollo Académico, de Economía, de Personal, entonces, se les hacía el trabajo, ingresando los datos, a veces directamente a la Base de Datos porque no existía la interface adecuada para que ellos puedan manipular, y entonces eso era lo que se hacía. Entonces cada uno de los trabajadores tenía una bruma de expediente, en la cual se dedicaban solamente a hacer eso. Por ejemplo llegaba “Rectificar tal nota porque el

profesor está rectificando”, había que buscar la entrada directamente en la Base de Datos, modificar ahí y ya está, uno tras otro, la parte económica también, de pronto que a tal alumno le cobraban demás, nos olvidamos de tal pensión, entonces prácticamente no se hacía desarrollo, entonces, cuando entramos se mantuvo ese personal y se trajo nuevo persona para que aprendiera de ellos y bueno, no siempre se tuvo la suerte de que pueda haber personal adecuado que hiciera esa función, porque habían muchos otros que se apeaban a la función anterior y seguían el mismo círculo vicioso, bueno se logró romper eso conjuntamente con un personal que pudo aprender del negocio y en algún momento pudimos prescindir de ese personal que estuvo trabajando largo tiempo, y respecto a los códigos, prácticamente se tuvo que esperar a leer, ha descifrar que estaba haciendo, y habían fechas en que ya nos ponían contra la espada y la pared y, como sea teníamos que resolver con trabajo bajo presión, con horas extras, etc., porque habían épocas en las que los muchachos se iban a las diez, once de la noche. Para los códigos de los ejecutables, menos mal que estaban en fox, se encontró una herramienta para decompilarlo y se obtuvo el código fuente, y bueno, se siguió descifrando los códigos, y de cierta forma se ha ido documentando, pero hay cosas incluso, que todavía ni sabemos para que sirven, hay Bases de Datos que hasta ahora sigue depurando Arturo, entonces, ese es el trabajo, y recién podemos decir, hace dos años, o desde el año pasado, recién se ha empezado a “desarrollar”, osea recién estamos cumpliendo en sí, la función de una oficina de TI. Entonces, se ha empezado a desarrollar y hacer nuestras funciones, eso también debido a que muchas de esas funciones que antes hacíamos, de trabajo que deberían hacer las otras oficinas lo hemos ido automatizando y entregando para que ellos mismos lo hagan. Entonces, eso nos ha ido liberando. Claro que ahora todavía se hace aún pero ya un pequeño porcentaje, de lo que antes se hacía habremos bajado siquiera un 80%, hay un 20% que todavía nos llega y tenemos que ayudarlo, porque no nos ha alcanzado el tiempo para poder hacer los respectivos módulos de automatización, y darles su trabajo, pero ya vamos en camino. Es así que ya últimamente incluso hemos establecido algunos estándares, algunos formatos para poder implementar software desarrollado, se está optando por metodologías de trabajo, etc.

Pregunta: Y las demás oficinas, ¿Cómo han tomado la automatización de algunos módulos o algunos trabajos que ustedes realizaban en forma manual?

Bueno, también eso es un cambio de cultura organizacional, entonces en un principio hubo rechazo, pero bueno, menos mal hubo la colaboración de los jefes de las oficinas que disusieron que se hiciera así hasta que se vayan acostumbrando, pero como también habían otras oficinas que pusieron resistencia como por ejemplo la de Secretaría General, pues persistían en mandarnos a nosotros, cuan le decíamos que nosotros ya no lo hacemos, incluso hubo una discusión hasta que parece que ya se resignaron.

Pregunta: ¿Cuál es el sistema que más dificultades ha traído a la oficina que usted dirige, y cuáles son las características que hacen que usted lo defina como tal?

En ranking, podríamos decir que la que mayor dificultad nos ha traído es la parte financiera, porque ese sistema en primer lugar ya es bastante antiguo, fue hecho cuando las circunstancias que se manejaban eran distintas a las de ahora, se manejaba ese sistema

cuando, por ejemplo la universidad tenía siete mil u ocho mil estudiantes, ahora llegamos a triplicarlo o cuadriplicarlos inclusive, entonces ese es uno de los aspectos. Lo otro es que también las sedes han crecido, la filial ha crecido tremendamente, ahora han cambiado muchos procesos que se manejaban, después la parte económica ha ido cambiando con esto de la SUNAT, entre otras cosas han ido cambiando los formatos, ese sistema no fue preparado, por ejemplo para hacer pagos adelantados, y así por el estilo existen operaciones para las cuales se tenían que hacer malabares para ajustarlo, y lo es que se ha implementado en un software que es bastante inseguro y ya desfasado que es el Fox Pro, entonces ese es el problema que tenemos hasta la actualidad. Entonces no hemos tenido tiempo para implementa un nuevo sistema, y todavía tenemos que hacer malabares para que funcione para que podamos responder a todo lo que nos van diciendo. Y lo otro que no había hablado también es que los que están a cargo de la gestión de la parte de economía, a veces no conocen sus procesos, no conocen a detalle, entonces se guían más bien por lo que el sistema dice, pero cuando nosotros queremos que nos digan cómo se hace de manera manual no se puede. Yo recuerdo que una vez hubo una discusión entre la contadora y Davy, porque Davy le decía: “Explíqueme manualmente como se hace ese proceso para yo corregirlo”, porque decían que estaba mal, y no le supo explicar, y más bien nos achacó que nosotros no sabíamos de contabilidad y por eso no podíamos hacerlo. Entonces la contadora debió explicar el procedimiento. Lógicamente, para nosotros poder automatizar una cosa nos debe explicar el proceso y el procedimiento, y de ahí se puede determinar que partes se pueden automatizar y que partes se tiene que seguir haciendo manualmente, entonces eso también no nos ayudó.

Y en segundo lugar, también en el ranking, el Sistema Académico, que de alguna manera superaba al anterior que ya fue hecho en un lenguaje más moderno, una plataforma moderna como es el Visual .Net, incluso la base de datos está en Microsoft SQL Server que ya es un manejador de base de datos más seguro, por eso incluso lo superaba. Pero tampoco no se tuvo el orden ni la visión de la programación ordenada, el desarrollo, el diseño, porque al final encontramos una documentación del sistema que son unos cuatro gráficos, algunos formularios y listo, eso es toda la documentación. Entonces, la persona que lo mantenía, realmente no tenía ninguna metodología de trabajo, era todo empíricamente, incluso los cambios lo hacía empíricamente, incluso ahora último me estoy enterando que, sin mi consentimiento hacía cambios, para tapar lo que no se había hecho en forma ordenada. Entonces, había cambios en la base de datos, entre otros. Eso ha hecho que sea difícil su mantenimiento, porque si hubiera habido un orden, una metodología no hubiera sido difícil, incluso la arquitectura que tiene es monolítica, no es modular ni por capas.

Pregunta: Bueno Ingeniero, en todo caso, las herramientas utilizadas tal vez fueron las correctas, pero no la forma como se utilizaba.

Claro, faltó metodología.

Pregunta: ¿Cuáles son los sistemas desarrollados bajo su gestión, y cuál es su opinión con respecto a ellos?

Los sistemas desarrollados bajo mi gestión, ya son el Sistema de Trámite Documentario, y el Sistema de Grados y Títulos y el Sistema de Control de Asistencia de Docentes. Esos tres sistemas se han desarrollado bajo algunos enfoques metodológicos y algunos estándares que ya se han establecido, incluso también hemos tenido como política el de tratar de utilizar herramientas libres, salvo la base de datos que lo mantenemos en SQL Server, el resto procuramos que sean libres, para no depender básicamente de las licencias porque, dicho sea de paso, la licencia del Microsoft SQL Server ya se nos venció y tenemos que renovar.

Pregunta: En todo caso Ingeniero, ¿Por qué mantienen la utilización del SQL Server, o han pensado en algún momento utilizar otro motor?

Si, al menos mi visión cuando yo entré, era pasarlo todo a Software Libre, incluso el manejador de Base de Datos, por eso que, dentro de las que se presentaban había dos opciones, una era MySQL o Postgres. Entonces, ¿Cuál de ellos era más conveniente? Según la investigación que hice es que el MySQL es más que nada orientado para dar soporte, más que nada, a Páginas Web, y entonces no soportaba gran cantidad de registros, en cambio Postgres tiene muchas de las características del Oracle, que es un sistema mucho más seguro, preparado para trabajar con miles de miles de registros. Es así que yo me atreví a consultar con un gurú de software libre, no recuerdo ahora su nombre pero es un argentino que ha desarrollado una distribución de Linux que se llama Ututo, entonces yo le consulté, y me contestó que a ojos cerrados “Postgres”, el problema de Postgres es que se necesita gente que conozca, y el que conocía era Davy, pero como todavía nuestras aplicaciones estaban en SQL Server, y todavía estábamos con esa carga de los otros trabajos y no había tiempo para desarrollar, prácticamente no aprovechamos esos conocimientos de Davy, ahora se fue y nadie sabe Postgres, y ni vamos a encontrar en el mercado persona que sepan Postgres, y si conseguimos por ahí es uno muy especializado que nos va a querer cobrar y la universidad no va a estar al alcance para que pague eso.

Pregunta: En todo caso, Ingeniero, ¿Por ahora los planes a futuro es seguir utilizando el SQL Server?

Sí, es lo que tenemos a la mano, más que nada porque no hay alguien que conozca. Yo incluso una vez le dije a Davy, “Haz una base de datos paralela y vamos haciendo el espejo”, pero no hubo tiempo, porque tú le veías recontra ocupado. No había tiempo.

Pregunta: Según la planificación de su oficina, ¿Cuáles son los sistemas que se han previsto desarrollar o están en desarrollo a partir de la fecha?

Bueno para este año, se ha proyectado reescribir o rehacer el sistema académico que ya estamos en proceso y justamente una de las herramientas que nos va a ayudar es el framework para rehacer este sistema y vamos a integrarlo rehaciendo, adaptando a los nuevos procesos que se están dando en la universidad y que sea acorde a la realidad actual entonces esperamos que hasta fin de año tengamos el sistema académico completo como se debe bajo los estándares gestionar arquitectura por capas todo documentado para que sea lo más fácil de hacer mantenimiento etc queremos lograrlo y para el próximo año

debemos de enfrentar el monstruo que es el sistema financiero bueno eso son los proyectos que estamos planteando para estos dos años.

Ingeniero, nosotros, como parte del desarrollo de nuestra tesis para optar el título profesional de “Ingeniero de Sistemas y Computación”, hemos planteado la creación de un framework, el cual debe ser ágil, cooperativo, seguro y escalable.

Pregunta: A partir de su experiencia, ¿cuáles deberían ser las características del framework?

Bueno las características las que se han indicado son adecuadas pero lo principal debe ser de fácil manejo en primer lugar para darle un manejo intuitivo para poder ser manejable por diversos especialistas que se presenten porque si es demasiado complicado que va a pasar va a generar rechazo en el usuario entonces va a decir no mejor yo trabajo por mi lado. Yo creo que son las características fundamentales que debe tener. Por lo demás si me parece interesante de que puedes hacer un trabajo cooperativo e incluso sirve para trabajar en equipo como para aplicar SCRUM por ejemplo es una característica fundamental que exige las metodologías actuales. Lo otro es un producto que este en software libre entonces está dentro de las políticas.

De todos modos mayormente cuando se habla de un framework como ya existe Symphony, Zend Framework que trabajan con php lo que se plantea es el lado de la aplicación creo que hasta ahora no existe un framework que incluya la estructura de la base de datos; en todo caso lo que nosotros hemos planteado, ya que nosotros no estamos utilizando un framework para cualquier desarrollo, sino orientado al desarrollo de la Universidad, entonces lo que se ha planteado es también incluir una estructura de base de datos, en coordinación con el Ingeniero Arturo, con quien hemos tenido una entrevista, que nos pueda dar las pautas de las necesidades. En todo caso, ¿Qué opina usted con respecto a que el framework también incluya una estructura de Base de Datos?

Si es que es orientado específicamente para una organización, entonces, sí se puede hacer. Ahora, si quisiéramos hacer uno general, como cada una de las organizaciones tiene sus propias características, no se podría hacer. Pero si es específicamente para una organización, entonces estamos particularizando.

Pregunta: Ingeniero, estamos muy agradecidos por la información brindada de su parte, ¿Hay algo más que quiera agregar a la entrevista?

Bueno, lo principal, para poder desarrollar cualquier aplicación es guiarse de normas o estándares que se dan, tener una mitología de trabajo y de desarrollo, creo que siguiendo esas pautas y la documentación correspondiente va a ser exitoso el trabajo. Y bueno, poco a poco nos estamos también disciplinando en eso, porque a veces, las malas prácticas hacen que uno adquiera también esos hábitos, entonces lo que tenemos que hacer es romper esos paradigmas desfasados, de pronto, antes el manejo era de pequeñas organizaciones con poca complejidad, pero cuando va creciendo ahí es cuando te ves abrumado por la complejidad y luego ya no lo puedes manejar, y es eso lo que nos está

pasando con los sistemas, por eso es fundamental la disciplina y el orden de las metodologías y los estándares, por eso existen, porque justamente esas recomendaciones de los estándares son fruto de las buenas prácticas que se han hecho en ciertas organizaciones, pero hay muchos que todavía han hecho cosas pequeñas, entonces, les ha funcionado esa informalidad. Pero cuando ya crece la organización, ahí bien el problema. Ahí es donde se ve si tu diseño de la Base de Datos estuvo bien, si tu arquitectura elegida era la correcta, pero si no has hecho una selección correcta o no lo hiciste bien, va a empezar a fallar, entonces eso se ve en los sistemas grandes. Eso serían todas mis recomendaciones.

Ingeniero, estamos muy agradecidos por su aporte a nuestra investigación, esperamos contar con su apoyo en lo subsiguiente del desarrollo de la presente tesis y le deseamos éxitos tanto en su gestión, su vida profesional y su vida personal. Muchas gracias.

De igual manera, el agradecimiento a ustedes porque, de alguna manera, con su trabajo están aportando también a la gestión, y creo que, de por sí, ya el apoyo es mutuo.

ANEXO B: Documentación del Framework BunnyPHP

BUNNYPHP



DOCUMENTACION DEL FRAMEWORK

1. Descripción general

BunnyPHP es un framework de código abierto diseñado para optimizar el desarrollo de aplicaciones web con PHP 5, JavaScript, AJAX, XML y JSON. BunnyPHP posee una estructura de programación por capas, donde se tiene principalmente una capa destinada a la conexión con la Base de Datos, una capa de Presentación para el diseño de la interfaz y una capa de Seguridad. La capa de Lógica del negocio, muy común en el desarrollo de software, no es utilizada en BunnyPHP, ya que uno de sus principios de escalabilidad es trasladar toda la lógica a procedimientos almacenados y funciones de la Base de Datos.

BunnyPHP separa todo el “proyecto de desarrollo” de los “datos de acceso a la base de datos” en un directorio externo llamado “Privado”, lo cual aumenta el nivel de seguridad en el desarrollo, ya que en las buenas prácticas no es correcto que un “Programador” tenga los datos de acceso a la Base de Datos, ya que estos deben ser de uso únicamente de los DBA (Administradores de Base de Datos) y de manera algo más restringida los DBD (Desarrolladores de Base de Datos).

Una característica peculiar de BunnyPHP es que posee una estructura de Base de Datos desarrollada en “Microsoft SQL Server”, orientada sobre todo al control de accesos y seguridad. Si bien BunnyPHP puede soportar cualquier “Gestor de Base de Datos”, la razón por la cual posee una estructura en este Gestor Privativo es que este framework está orientado sobre todo al desarrollo de “Aplicaciones Web” en la Universidad Peruana Los Andes, cuya Oficina Universitaria de Informática y Sistemas gestiona los datos principalmente con el “Microsoft SQL Server”.

Este framework fue desarrollado por Emily Rossy Salinas Chamorro y José Antonio Ríos Monterrey, utilizando inicialmente una pequeño mini – framework

de prueba, desarrollado y publicado por el Desarrollador de Software Antonio Ognio en GitHub, un sistema controlador de versiones. Para lograrlo, el equipo de desarrollo conformado por Emily Salinas y José Ríos recibió constante apoyo y asesoramiento por parte de la Oficina Universitaria de Informática y Sistema de la UPLA, sobresaliendo sobre todo la participación del Analista – Programador de sistemas Irwin Yauri Orihuela.

1.1. Acerca del nombre del Framework “BunnyPHP”

Como es clásico en el mundo del Software Libre, las diferentes aplicaciones y distribuciones de Linux son relacionadas muchas veces con algún animal. En este caso, “Bunny” quiere decir en español “Conejito”, y este, en principio, es un animal del agrado de los creadores, y por otro lado, este es símbolo de agilidad, inteligencia, sabiduría y cooperatividad.

1.2. Mascota – Logo representante del framework



BunnyPHP está representado por un “conejo” famoso entre los videojuegos, su nombre es Jazz. Este fue el personaje principal del videojuego “Jazz Jackrabbit” y es un conejo verde muy inteligente, ágil y fuerte, los rasgos exactos para representar a este framework.

2. Librerías y Archivos de Terceros utilizados

Antes de exponer detalles más técnicos de este framework, es importante dar a conocer las diferentes librerías desarrolladas por terceros que fueron adjuntadas a su arquitectura, respetando siempre el derecho de autor y su licencia de uso.

2.1.Stradlin

Esta librería fue desarrollada por Antonio Ognio, un “Arquitecto de Sistemas” peruano con amplia experiencia en el lenguaje PHP. Stradlin es el modelo de un mini-framework que permite el manejo de los diferentes verbos, el enrutamiento web, códigos de estado HTTP, manejo de cabeceras, entre otros. Dicha librería fue adaptada a la arquitectura del framework, principalmente para manejar el enrutamiento web, y tiene el nombre de “cerebro.php”.

2.2.jQuery

jQuery es una librería de JavaScript creada por John Resig. Esta librería permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

2.3.PHP-JRU

PHP-JRU (PHP Jasper Report Utils) es una librería pensada para generar reportes diseñados bajo la herramienta iReport desde una simple aplicación escrita en PHP. Para ello esta librería se comunica con la JVM a través de Java/Bridge, y de esta manera poder cargar las librerías proporcionadas por JasperReport he indicarle las tareas que deben realizarse para generar los reportes tal cual como si se estuviera haciendo desde una aplicación escrita en java. En esta versión PHP-JRU provee de una serie de rutinas para que sea una tarea muy fácil.

2.4.Recptchalib

Es una librería desarrollada por Google, que al ser incluida en un proyecto, permite utilizar su Web Service “Recaptcha”, una extensión de la prueba Captcha que se utiliza para reconocer texto presente en imágenes. Emplea por tanto la prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano para, a su vez, mejorar la digitalización de textos.

3. Instalación

A continuación se describen los diferentes pasos que se deben seguir para lograr una instalación correcta del framework BunnyPHP y poder utilizarlo en el desarrollo de una aplicación web.

3.1. Obtener una copia estable de BunnyPHP

Antes de comenzar con la instalación, propiamente dicha, se debe obtener una copia limpia del framework.

3.2. Instalar Servidor HTTP Apache y PHP 5 a superior.

Para comenzar, es importante realizar la instalación del Servidor Web que dará soporte a tu aplicación desarrollada mediante el framework BunnyPHP. En este caso, BunnyPHP ha sido desarrollado y testeado con el servidor HTTP Apache, el mismo que puede ser instalado tanto en una plataforma Linux como Windows.

Luego, también es necesario instalar el lenguaje PHP, en una versión no inferior a la 5 para que sea completamente compatible con el framework.

3.3. Configuración del Servidor HTTP Apache para dar soporte a BunnyPHP

BunnyPHP utiliza el archivo “.htaccess” ubicado en la raíz del proyecto. Este es un fichero especial, popularizado por el Servidor HTTP Apache que permite definir diferentes directivas de configuración para cada directorio (con sus respectivos subdirectorios) sin necesidad de editar el archivo de configuración principal de Apache.

Para que el servidor HTTP Apache permita la utilización de “.htaccess” es necesario modificar una línea en su archivo de configuración “httpd.conf”. Buscar la línea que dice “AllowOverride None” y cambiarlo por “AllowOverride All”, luego de guardar los cambios, el servidor HTTP Apache estará correctamente configurado.

3.4. Instalar Servidor Tomcat

El Servidor Apache Tomcat, Jakarta Tomcat o simplemente Tomcat, permite la ejecución de código Java. Este servidor deberá ser instalado sólo si se desea la utilización de las funciones de BunnyPHP que permiten la ejecución de reportes “Jasper” desarrollados con el iReport, en caso contrario, no sería necesaria la instalación de este servidor.

3.5. Configurar Servidor Tomcat y extensión JavaBridge

En caso de haber instalado el Servidor Tomcat para ejecución de reportes Jasper, es necesaria también la instalación del PHP JavaBridge, el cual es utilizado como puente de interacción entre PHP y JAVA.

En la copia estable que tenemos del framework, deberemos editar el archivo “php-jru.php”, el cual está ubicado en “RaizProyecto/Librerias/php-jru/ php-jru.php”. En la línea 33 del archivo, deberemos cambiar el código de conexión al Servidor Tomcat según como haya sido la instalación.

3.6. Colocar copia estable de BunnyPHP en el directorio web y realizar pruebas de funcionamiento

Finalmente, antes de comenzar a utilizar nuestro framework, será importante colocar nuestra copia estable de BunnyPHP en la raíz del servidor web HTTP Apache y cambiarle el nombre del proyecto. Posteriormente, podremos realizar las pruebas necesarias para comprobar que nuestro framework esté funcionando correctamente. Si se quiere utilizar la estructura de la base de datos en SQLserver que trae el framework, será necesario cumplir con el paso 3.7. Que se expone a continuación.

3.7. Paso alternativo – instalación del Esquema de Seguridad desarrollado en Microsoft SQL Server

Si se ha optado por la utilización de la estructura de la base de datos en SQLserver que trae el framework, será necesario tener instalado en un servidor cuya plataforma debe ser necesariamente Windows, el Microsoft SQL Server. Posteriormente se ejecutarán los scripts que vienen adjuntos al paquete del framework sobre la base de datos deseada.

Los datos de conexión al servidor de base de datos se encuentran en el archivo “Conexión.php”, el cual está ubicado en “Privado/Conexion/Conexion.php”.

Los demás datos, como el nombre de la Base de Datos se encuentra en el archivo “variablesInt_seguridad.php” ubicado en “Proyecto/variablesInt_seguridad.php”.

3.8. ¡Disfruta!

Luego de haber concluido la instalación del framework, todo está listo para comenzar con el desarrollo de nuestra aplicación web.

4. Documentación de funciones útiles para el desarrollador:

El framework BunnyPHP, a lo largo de toda su arquitectura, posee una serie de funciones cuya utilidad varía, desde permitir la conexión con la “Base de Datos”, hasta formas de mostrar mensajes en la interfaz del sistema. De todas estas funciones, hay algunas que son utilizadas únicamente por el framework para lograr su funcionamiento, mientras otras han sido desarrolladas para uso exclusivo del programador. Es por ello que, a continuación, se documentan aquellas funciones útiles para el desarrollador, mientras que las funciones utilizadas únicamente por el framework, están documentadas en el mismo código fuente de BunnyPHP.

4.1. Funciones de “Proyecto/Capa_AccesoDatos/accesobd.php”

```
ejecutar_pa($param1,$param2,$param3,$param4)
```

Esta función es utilizada por la “Capa de Acceso a Datos” para realizar consultas y ejecutar operaciones desde la aplicación sobre la Base de Datos.

Parámetros:

- **\$param1:** Nombre del procedimiento almacenado.
- **\$param2:** Formato del retorno. Este puede ser ‘array’, el cual devolverá los datos en un arreglo PHP. Puede ser también “json”, devolviendo los datos en esta estructura. Finalmente, puede ser también “xml”, devolviendo los datos en esta estructura. Si no se especifica este parámetro, se devolverá un “dato unitario” con el índice ‘rpta’ en un arreglo PHP.
- **\$param3:** Es el parámetro de entrada del procedimiento almacenado, este debe estar en formato XML conteniendo todos los datos que necesite el procedimiento- En caso de que el procedimiento no necesite parámetros de entrada, no se envía este dato.
- **\$param4:** Este es un dato booleano, el cual puede ser “true” o “false”. En caso de ser “false” (falso), el procedimiento devolverá su retorno en un matriz bidimensional, según el formato que se haya especificado (array, json o xml). En caso de ser “true” (verdadero), el retorno será en una matriz unidimensional de un solo dato, con el índice “rpta”. Por tanto, se utiliza “true” cuando la respuesta del “Procedimiento almacenado” posee como parámetros de entrada @xml (contiene todos los datos) y @rpta output (que almacena el valor único de retorno), y se utiliza “false” cuando el procedimiento posee como único dato de entrada @xml, devolviendo por consiguiente una consulta, la cual será capturada por el framework como una matriz bidimensional.

Retorno:

- En caso de que “\$param4” sea “true”, el retorno será una matriz unidimensional de un solo dato, cuya respuesta podrá ser capturada con el índice “rpta”. Esta matriz será devuelta en el formato especificado en el “\$param2”.
- En caso de que “\$param4” sea “false”, el retorno será una matriz bidimensional conteniendo toda la consulta de retorno del procedimiento almacenado, cuyos datos podrán ser capturados haciendo un recorrido a la matriz data[x][y]. Esta matriz será devuelta en el formato especificado en el “\$param2”.

4.2.Funciones de “Controlador/cerebro.php”

4.2.1. Funciones para enrutamiento, URLs, estados HTTP, cabeceras HTTP

Función “ruta(\$regexp, \$methods, \$callback)”

Función de enrutamiento del framework.

```
function ruta($regexp, $methods, $callback) {
    set_content_type('Content-Type', 'text/html');

    if (!preg_match('/^\^(.)+$/ ', $regexp)) {
        $regexp = sprintf("%s", $regexp);
    }
    if (!preg_match('/^(.)+\$\$/ ', $regexp)) {
        $regexp = sprintf("%s$", $regexp);
    }
    $regexp = str_replace("/", "\/", $regexp);
    $regexp = sprintf("/%s/", $regexp);

    /* Create array of accepted HTTP methods */
    $methods = explode(",", $methods) ;
    foreach($methods as $k=>$v) {
        $methods[$k] = trim(strtoupper($v));
    }

    $uri = get_uriSolicitada();
    if (comprobar_metodosPeticion($methods) &&
    preg_match($regexp, $uri, $params)) {
        $callback($params);
        $existe_pagina=true;
        exit();
    }
}
```

Función “codigo_estado_mapa_ENG ()”

Función que retorna una matriz con todos los códigos de estado en inglés.

```
function codigo_estado_mapa_ENG() {
    return array(
        200 => 'OK',
        201 => 'Created',
        202 => 'Accepted',
        203 => 'Non',
        204 => 'No Content',
        205 => 'Reset Content',
        206 => 'Partial Content',
        300 => 'Multiple Choices',
        301 => 'Moved Permanently',
        302 => 'Found',
        303 => 'See Other',
        304 => 'Not Modified',
        305 => 'Use Proxy',
        307 => 'Temporary Redirect',
        400 => 'Bad Request',
        401 => 'Unauthorized',
        402 => 'Payment Required',
        403 => 'Forbidden',
        404 => 'Not Found',
        405 => 'Method Not Allowed',
        406 => 'Not Acceptable',
        407 => 'Proxy Authentication Required',
        408 => 'Request Timeout',
        409 => 'Conflict',
        410 => 'Gone',
        411 => 'Length Required',
        412 => 'Precondition Failed',
        413 => 'Request Entity Too Large',
        414 => 'Request',
        415 => 'Unsupported Media Type',
        416 => 'Requested Range Not Satisfiable',
        417 => 'Expectation Failed',
        500 => 'Internal Server Error',
        501 => 'Not Implemented',
        502 => 'Bad Gateway',
        503 => 'Service Unavailable',
        504 => 'Gateway Timeout',
        505 => 'HTTP Version Not Supported',
    );
}
```

Función “codigo_estado_mapa_ESP()”

Función que retorna una matriz con todos los códigos de estado en Español.

```
function codigo_estado_mapa_ESP() {
    return array(
        200 => 'Bien',
        201 => 'Creado Correctamente.',
        202 => 'Esta Aceptado',
        203 => 'Non',
        204 => 'No existe Contenido Alguno.',
        205 => 'No se restablece el contenido.',
        206 => 'Es un contenido parcial.',
        300 => 'Tiene multiples opciones.',
        301 => 'Fue movido permanentemente.',
        302 => 'Encontrado.',
        303 => 'No se encuentra veo tras opciones.',
        304 => 'No tiene cambios.',
        305 => 'Debe usar proxy.',
        307 => 'Se redirecciono temporalmente.',
        400 => 'Es una solicitud incorrecta.',
        401 => 'No esta autorizado.',
        402 => 'Necesita de un pago, para poder ingresar.',
        403 => 'Es prohibida',
        404 => 'No se encontró en este servidor',
        405 => 'No permite este método.',
        406 => 'No Aceptado',
        407 => 'Necesita autenticarse en el proxy.',
        408 => 'Necesita un tiempo de espera.',
        409 => 'Tiene un conflicto.',
        410 => 'Gone',
        411 => 'Requiere más espacio.',
        412 => 'Necesita una precondition correcta.',
        413 => 'Solicitud demasiado grande.',
        414 => 'Solicitud',
        415 => 'No soporta tipo de archivo.',
        416 => 'Necesita el rango solicitado.',
        417 => 'Fallo de exception.',
        500 => 'Tiene un error interno del Servidor',
        501 => 'No esta implementado',
        502 => 'Tiene un error en la puerta de enlace',
        503 => 'Este servicio no está Disponible',
        504 => 'Gateway Timeout',
        505 => 'No soporta la versión del HTTP',
    );
}
```

Función “set_tipo_contenido (\$_type,\$content_type)”

Función que da valor a las cabeceras HTTP

```
function
set_tipo_contenido($_tipo,$tipo_contenido)
{
    header("$_tipo: $tipo_contenido");
}
```

Función “set_meta (\$_tipo,\$_dato)”

Función que establece metadatos HTTP que son utilizados por los buscadores.

```
function set_meta($_tipo,$_dato)
{
    echo '<meta name="'.$_tipo.'"
        content="'.$_dato.'">';
}
```

Función “function get_uriSolicitada()”

Función que devuelve el URI de la página.

```
function get_uriSolicitada() {
    if (array_key_exists('PATH_INFO', $_SERVER)) {
        return $_SERVER['PATH_INFO'];
    } elseif (array_key_exists('REDIRECT_URL',
$_SERVER)) {
        return $_SERVER['REDIRECT_URL'];
    } else {
        return '/';
    }
}
```

Función “function comprobar_metodosPeticon(\$metodos)”

Función que comprueba si el método HTTP (verbo) utilizado para acceder a la página se encuentra en el arreglo que obtiene como variable de entrada.

```
function comprobar_metodosPetición($metodos) {
    return in_array($_SERVER['REQUEST_METHOD'],
    $metodos);
}
```

Función “set_codigo_estado (\$codigo_estado)”

Función que permite establecer el código de estado HTTP de la página.

```
function set_codigo_estado($codigo_estado) {
    $map = codigo_estado_mapa_ENG();
    header(sprintf("HTTP/1.1 %d %s",
    $codigo_estado,
    strtoupper($map[$codigo_estado])
    ));
}
```

Función “matriz_todosVerbos()”

Función que retorna una matriz con todos los verbos permitidos.

```
function matriz_todosVerbos(){
    return array(
    "OPTIONS",
    "GET",
    "HEAD",
    "POST",
    "PUT",
    "DELETE",
    "TRACE",
    "CONNECT"
    );
}
```

Función “todosVerbos()”

Función que retorna todos los verbos según la matriz de la función all_verbs_array.

```
function todosVerbos(){
    return implode(", ", all_verbs_array());
}
```


Función “respuesta_estado(\$codigo_estado)”

Función que escribe en pantalla la descripción del código de estado HTTP de la página.

```
function respuesta_estado($codigo_estado)
{
    $map = codigo_estado_mapa_ENG();
    $map_explicacion=codigo_estado_mapa_ESP();

    header(sprintf("HTTP/1.1 %d %s",
    $codigo_estado,
    strtoupper($map[$codigo_estado])));

    echo "<!DOCTYPE HTML PUBLIC>";
    echo "<html><head>";
    set_title($codigo_estado."
    ".strtoupper($map[$codigo_estado]));
    echo "</head><body>";
    echo "<h1>".$codigo_estado."
    ".strtoupper($map[$codigo_estado])."</h1>". "\n";
    echo "<p>The requested URL
    ".$_SERVER['REDIRECT_URL']."
    ".$map_explicacion[$codigo_estado]."</p>". "\n";
    echo "</body></html>";
}
```

4.2.2. Funciones básicas para programación

Función “incluir_plantilla(\$template, \$template_dir'Capa_Presentacion/Plantillas/')”

Función que permite incluir una plantilla (archivo contenido en el directorio Plantillas).

```
function incluir_plantilla($template,
$template_dir'Capa_Presentacion/Plantillas/') {
    $path = str_replace("//", "/",
    $template_dir.'/'.$template);
    include($path);
}
```

Función “incluir_modulo(\$template, \$template_dir='Capa_Presentacion/Modulos/')”

Función que permite incluir un módulo (archivo contenido en el directorio Módulos).

```
function incluir_modulo($template,
$template_dir='Capa_Presentacion/Modulos/') {
    $path = str_replace("//", "/",
$template_dir.'/'.$template);
    include($path);
}
```

Función “set_favicon (\$href)”

Función que permite establecer el ícono de la aplicación.

```
function set_favicon($href)
{
    echo '<link rel="icon"
href="Capa_Presentacion/Recursos/'.$href.'"
type="image/x-icon" />';
}
```

Función

“set_imagen(\$href,\$h=',\$sw=',\$salt=',\$met=',\$script=',\$img1=',\$img2=')”

Función que permite insertar una imagen en el código HTML.

```

function
set_imagen($href,$h='', $w='', $alt='', $met='', $sc
ript='', $img1='', $img2='')
{
    if($img1=='')
    {
        if($met=='')
        {
            echo '';
        }
        else
        {
            echo '';
        }
    }
    else
    {
        if($met=='')
        {
            echo '';
        }
        else
        {
            echo '';
        }
    }
}
}

```

Función “imagen(\$href)”

Función que permite identificar una imagen (contenida en los recursos de la "Capa de Presentación).

```
function imagen($href)
{
    echo "Capa_Presentacion/Recursos/$href";
}
```

Función “set_tituloPagina(\$nombre)”

Función que permite establecer el título de una página.

```
function set_tituloPagina($nombre) ($nombre)
{
    echo '<title>'.$nombre.'</title>';
}
```

Función “set_recurso(\$_ruta)”

Función que permite agregar un recurso, ya sea estilo (css o php) o función de javascript (js).

```
set_recurso($_ruta)
{
    $_tipo=substr($_ruta,strpos($_ruta,".")+1);

    switch($_tipo)
    {
        case 'css':case 'php':
            echo '<link rel="stylesheet"
type="text/css"
href="Capa_Presentacion/Estilos/'.$_ruta.'" />';
            break;

        case 'js':
            echo '<script type="text/javascript"
src="Capa_Presentacion/Funciones/'.$_ruta.'"></s
cript>';
            break;
    }
}
```

Función “set_recurso_ext(\$_ruta,\$_tipo)”

Función que permite agregar un recurso EXTERNO, ya sea estilo (css o php) o función de javascript (js).

```

set_recurso_ext($_ruta,$_tipo)
{
    switch($_tipo)
    {
        case 'css':case 'php':
            echo '<link rel="stylesheet"
type="text/css" href="'. $_ruta.'" />';
            break;

        case 'js':
            echo '<script type="text/javascript"
src="'. $_ruta.'"></script>';
            break;
    }
}

```

Función “set_librerias(\$_ruta)”

Función que permite incluir en el módulo una librería, la cual debe estar contenida en el directorio "Librerias".

```

function set_librerias($_ruta)
{
    echo '<script type="text/javascript"
src="Librerias/'. $_ruta.'"></script>';
}

```

Función “script (\$cuerpo=”)”

Función que permite escribir código de Javascript rápidamente dentro de PHP.

```

function script($cuerpo=”)
{
    echo "<script>";
    echo $cuerpo;
    echo "</script>";
}

```

Función “combo(\$data,\$val,\$opcion,\$name,\$class='', \$style='')”

Función que permite incluir un combo (etiqueta select en html).

```
function
combo($data,$val,$opcion,$name,$class='', $style=
'')
{

//$data=Logica::ejecutar_pa($pa,'array',$xml,$sa
1);
    echo '<select name="'. $name.'" id="'. $name.'"
class="'. $class.'" style="'. $style.'" >';
    echo '<option value="0">elegir...</option>';
    foreach ($data as $fila):
        echo '<option
value="'. utf8_encode($fila[$val]).'">'. utf8_enco
de($fila[$opcion]).'</option>';
    endforeach;
    echo '</select>';
}
```

Función “conf_pre ()”

Función que permite incluir la configuración predeterminada para los módulos.

```
function conf_pre()
{
    set_meta("keywords","Grados y Títulos");
    set_meta("Description","Universidad Peruana
Los Andes");
    set_content_type("author","Emily Salinas -
José Rios");

    set_favicon("favicon.ico");//asignar icono
estatico Favicon
    set_librerias("jquery-1.10.2.js");
    set_librerias("jquery-ui.js");
    set_librerias("bunnyJS.js");
    set_link("funciones_reportes.js");
}
```

Función “function agregar_vS(\$nombre,\$valor=null)”

Función utilizada para crear variables de sesión.

```
function agregar_vS($nombre,$valor=null)
{
    $_SESSION[$nombre]=$valor;
}
```

Función “set_href(\$href,\$nombre_href,\$clase=’’)”

Función utilizada para crear enlaces, especificando su nombre y clase.

```
function set_href($href,$nombre_href,$clase='')
{
    echo '<a href =''.$href.'''
        class=""'.$clase.'''>'.$nombre_href.'</a>';
}
```

Función “set_avisó ()”

Función para habilitar el uso del "Aviso" en la parte superior.

```
function set_avisó()
{
    echo '<div name="avisó" id="avisó"
style="background:rgb(218, 218, 218);" ></div>';
    return;
}
```

Función “set_cargando ()”

Función para habilitar el uso de la capa "Cargando" que bloquea la página.

```
function set_cargando()
{
    echo '<div name="cargando"
        id="cargando"></div>';
}
```

Función “set_reporte_jasper(\$nomarchivo,\$sarreglo)”

Función para ejecutar un reporte Jasper y mostrarlo en pantalla.

```

function
set_reporte_jasper($nomarchivo,$arreglo){

    $fichero= explode("/",$_SERVER['PHP_SELF']);
    $ruta= $fichero[1];
    $raiz= $_SERVER['DOCUMENT_ROOT'];
    //Ruta a donde se muestra el archivo de salida
Pdf

$salida="../../../../../../../../tmp/pdf/".$nomarchivo.'.
pdf';
    //Ruta a donde deseo Guardar Mi archivo de
salida Pdf
    $nombrepdf=
$raiz.'/tmp/pdf/'.$nomarchivo.'.pdf';
    //Llamando las librerias
    require('Librerias/php-jru/php-jru.php');
    //Llamando la funcion JRU de la libreria php-
jru
    $jru=new JRU();
    //Ruta del reporte compilado Jasper generado
por IReports
    $Reporte= $raiz .
'/'.$ruta.'/Proyecto/Capa_Presentacion/Reportes/
'. $nomarchivo.'.jasper';

    //Parametro en caso de que el reporte no este
parametrizado
    $Parametro=new java('java.util.HashMap');

    for ($i=0; $i<sizeof($arreglo);$i++)
    {
        $Parametro-
>put($arreglo[$i][0],$arreglo[$i][1]);
    }

    $Conexion= new
JdbcConnection("com.microsoft.sqlserver.jdbc.SQL
ServerDriver","jdbc:sqlserver://".servidor.";dat
abaseName=".BD,usuario,pass);
    //$Conexion= new
JdbcConnection("com.microsoft.sqlserver.jdbc.SQL
ServerDriver","jdbc:sqlserver://".servidor.";dat
abaseName=DBCampusNet",usuario,pass);
    //Generamos la Exportacion del reporte
    $jru-
>runReportToPdfFile($Reporte,$nombrepdf,$Paramet
ro,$Conexion->getConnection());
    echo "<embed src= ".$salida." width= 800
height= 600 ></embed> ";
}

```


4.2.3. Funciones de utilidad variada

Función “`serve_json($doc, $options = 0, $cbname = 'callback')`”

Función que transforma a una estructura JSON un arreglo PHP.

```
function serve_json($doc, $options = 0, $cbname = 'callback') {
    set_content_type('Content-Type', 'application/json');
    $doc = str_replace("\\", "", json_encode($doc, $options));
    $callback = null;
    if (array_key_exists('HTTP_JSONP_CALLBACK', $_SERVER) &&
        strlen($_SERVER['HTTP_JSONP_CALLBACK']) > 0) {
        $callback = $_SERVER['HTTP_JSONP_CALLBACK'];
    } elseif (array_key_exists($cbname, $_REQUEST) &&
        strlen($_REQUEST[$cbname]) > 0) {
        $callback = $_REQUEST[$cbname];
    }
    if (strlen($callback) > 0) {
        printf("%s(%s);\n", $callback, $doc);
    } else {
        echo $doc;
    }
}
```

Función “`phpinfo_to_file ($path)`”

Función que almacena la información de la instalación de PHP en la ruta especificada.

```
function phpinfo_to_file($path) {
    ob_start();//Activa el almacenamiento en búfer de la salida
    phpinfo();//Muestra información sobre la configuración de PHP
    $contents = ob_get_contents();//Devolver el contenido del búfer de salida
    ob_end_clean();//Limpiar (eliminar) el búfer de salida y deshabilitar el almacenamiento en el mismo
    $fp = @fopen($path, 'w+');
    @fputs($fp, $contents);
    @fclose($fp);
}
```

4.3. Funciones de “Librerías/bunnyJS.js”

4.3.1. Funciones útiles para la interfaz

Función “mostrarCargando()”

Función que bloquea toda la interfaz y muestra una capa que indica que el sistema está cargando.

```
function mostrarCargando()  
{  
    $("#cargando").css("display", "block");  
    return;  
}
```

Función “ocultarCargando ()”

Función que oculta la capa "Cargando" desbloqueando toda la pantalla de la aplicación.

```
function ocultarCargando(){  
    $("#cargando").css("display", "none");  
    return  
}
```

Función “alerta (mensaje, título)”

Función que muestra un "Cuadro de Diálogo" parecido al alert() de Javascript, pero con interfaz personalizada.

```
function alerta(mensaje,titulo){
    var div = $('<div />').appendTo('body');
    div.attr('id', 'alerta');
    $( "#alerta" ).empty();
    $("#alerta").append(mensaje);
    $( "#alerta" ).dialog({
        title:titulo,
        resizable: true,
        modal: true,
        minWidth : 500,
        position: ["center",170],
        buttons: {
            OK: function() {
                $( this ).dialog( "close" );
            }
        }
    });
}
```

Función “aviso(mensaje,color,back)”

Muestra un mensaje en el label "aviso" del color indicado.

```
function aviso(mensaje,color,back)
{
    $("#aviso").text(mensaje);
    $("#aviso").css("color", color);
    $("#aviso").css("background", back);
}
```

Función “cajaColor(cajaErrada,color)”

Encierra de rojo la caja de texto errada y muestra en el label "aviso" el mensaje.

```
function cajaColor(cajaErrada,color)
{
    document.getElementById(cajaErrada).style.border=
    "2px solid "+color;
    return false;
}
```

Función “avisoLimpiar (cajaErrada)”

Limpia la caja de texto encerrada.

```
function avisoLimpiar(cajaErrada)
{
    document.getElementById(cajaErrada).
    style.border="";
}
```

4.3.2. Funciones útiles para la validación de datos

Función “campoVacio(nombreCampo)”

Función que comprueba si el valor está vacío.

```
function campoVacio(nombreCampo){
    var campo=$("#"+nombreCampo).val();
    if (campo.length == 0 || /^s+$/ .test(campo))
    {
        return true;
    }
    return false;
}
```

Función “ValidaNumero_Fijo (tel)”

Función que comprueba si el valor es un número de teléfono fijo válido.

```
function ValidaNumero_Fijo(tel)
{
    if((/^d{6}$/ .test(tel)) ||
        (^d{7}$/ .test(tel)))
    {
        return true;
    }
    return false;
}
```

Función “ValidaNumero_Cel (cel)”

Función que comprueba si el valor es un número de celular válido.

```
function ValidaNumero_Cel(cel)
{
  if((/^\d{9}$/.test(cel)) ||
    (/^\d{10}$/.test(cel))) {
    return true;
  }
  return false;
}
```

Función “ValidaMail (mail)”

Función que comprueba si el valor es un correo electrónico válido.

```
function ValidaMail(mail)
{
  var er = /^[0-9a-z_\-\.\.]+@([a-z0-9\-\
]+\.\.?)*[a-z0-9]+\.[a-z]{2,4}|travel)$/i;
  return er.test(mail);
}
```

Función “ValidaDni (dni_usuario)”

Función que comprueba si el valor es un número de DNI válido.

```
function ValidaDni(dni_usuario)
{
  if(dni_usuario=="1234567" ||
    dni_usuario== "87654321" ||
    dni_usuario== "00000000" ||
    dni_usuario== "12345678" ||
    dni_usuario== "11111111" ||
    dni_usuario== "22222222" ||
    dni_usuario== "33333333" ||
    dni_usuario== "44444444" ||
    dni_usuario== "55555555" ||
    dni_usuario== "66666666" ||
    dni_usuario== "77777777" ||
    dni_usuario== "88888888" ||
    dni_usuario== "99999999" ||
    dni_usuario.length!=8)
  {
    return false;
  }
  return true;
}
```

Función “generalCase (palabra)”

Función para hacer indiferente las mayúsculas y las tildes.

```
function generalCase(palabra)
{
    return palabra.toUpperCase()
        .replace(/Á/g,"A")
        .replace(/É/g,"E")
        .replace(/Í/g,"I")
        .replace(/Ó/g,"O")
        .replace(/Ú/g,"U");
}
```

Función “ceros_izq (variable)”

Función que permite 0 a la izquierda del número.

```
function ceros_izq(variable){
    cad='';
    for(i=0;i<variable.length;i++)
    {
        v=variable.substr(i,1);

        switch(v){
            case '0':cad+=v;break;
            case '1':cad+=v;break;
            case '2':cad+=v;break;
            case '3':cad+=v;break;
            case '4':cad+=v;break;
            case '5':cad+=v;break;
            case '6':cad+=v;break;
            case '7':cad+=v;break;
            case '8':cad+=v;break;
            case '9':cad+=v;break;

        }

    }
    return cad;
}
```

Función “Solo_Numerico (variable)”

Función que elimina las letras de la cadena de texto y devuelve solo los números.

```
function Solo_Numerico(variable)
{
    Numer=parseInt(variable);
    //esto valida el inicio
    if (isNaN(Numer))
    {
        return "";
    }
    return ceros_izq(variable);
}
```

Función “ValNumero (Control)”

Función que impide escribir letras en un control.

```
function ValNumero(Control)
{
    Control.value=Solo_Numerico(Control.value);
}
```

Función “trim (cadena)”

Función que corta los espacios en blanco de los extremos de una cadena.

```
function trim (cadena)
{
    return cadena.replace(/^s+/g, '').
        replace(/\s+$/g, '');
}
```

Función “str_replace (busca, repla, orig)”

Función que reemplaza un caracter por otro en una cadena texto, como el “str_replace” de PHP.

```
function str_replace(busca, repla, orig)
{
    str      = new String(orig);
    rExp     = "/" + busca + "/g";
    rExp     = eval(rExp);
    newS     = String(repla);
    str = new String(str.replace(rExp, newS));
    return str;
}
```

4.3.3. Funciones útiles para codificación y decodificación UTF8

Función “utf8_decode (str_data)”

Función que convierte una cadena con los caracteres codificados con UTF-8 a un sencillo byte.


```

function utf8_decode (str_data)
{
    var tmp_arr = [],
        i = 0,
        ac = 0,
        c1 = 0,
        c2 = 0,
        c3 = 0,
        c4 = 0;

    str_data += '';

    while (i < str_data.length) {
        c1 = str_data.charCodeAt(i);
        if (c1 <= 191) {
            tmp_arr[ac++] = String.fromCharCode(c1);
            i++;
        } else if (c1 <= 223) {
            c2 = str_data.charCodeAt(i + 1);
            tmp_arr[ac++] = String.fromCharCode(((c1 &
                31) << 6) | (c2 & 63));
            i += 2;
        } else if (c1 <= 239) {
            c2 = str_data.charCodeAt(i + 1);
            c3 = str_data.charCodeAt(i + 2);
            tmp_arr[ac++] = String.fromCharCode(((c1 &
                15) << 12) | ((c2 & 63) << 6) | (c3 & 63));
            i += 3;
        } else {
            c2 = str_data.charCodeAt(i + 1);
            c3 = str_data.charCodeAt(i + 2);
            c4 = str_data.charCodeAt(i + 3);
            c1 = ((c1 & 7) << 18) | ((c2 & 63) << 12) |
                ((c3 & 63) << 6) | (c4 & 63);
            c1 -= 0x10000;
            tmp_arr[ac++] = String.fromCharCode(0xD800 |
                ((c1 >> 10) & 0x3FF));
            tmp_arr[ac++] = String.fromCharCode(0xDC00 |
                (c1 & 0x3FF));
            i += 4;
        }
    }
    return tmp_arr.join('');
}

```

Función “utf8_encode (argString)”

Función que codifica un string ISO-8859-1 a UTF-8.

```

function utf8_encode(argString)
{
    if (argString === null || typeof argString
        === 'undefined') { return '';}
    var string = (argString + ''); //
    .replace(/\r\n/g, "\n").replace(/\r/g, "\n");
    var utftext = '',
        start, end, stringl = 0;

    start = end = 0;
    stringl = string.length;
    for (var n = 0; n < stringl; n++) {
        var c1 = string.charCodeAt(n);
        var enc = null;

        if (c1 < 128) {
            end++;
        } else if (c1 > 127 && c1 < 2048) {
            enc = String.fromCharCode(
                (c1 >> 6) | 192, (c1 & 63) | 128
            );
        } else if ((c1 & 0xF800) !== 0xD800) {
            enc = String.fromCharCode(
                (c1 >> 12) | 224, ((c1 >> 6) & 63) |
                128, (c1 & 63) | 128
            );
        } else { // surrogate pairs
            if ((c1 & 0xFC00) !== 0xD800) {
                throw new RangeError('Unmatched trail
                    surrogate at ' + n);
            }
            var c2 = string.charCodeAt(++n);
            if ((c2 & 0xFC00) !== 0xDC00) {
                throw new RangeError('Unmatched lead
                    surrogate at ' + (n - 1));
            }
            c1 = ((c1 & 0x3FFF) << 10) + (c2 & 0x3FFF)
                + 0x10000;
            enc = String.fromCharCode(
                (c1 >> 18) | 240, ((c1 >> 12) & 63) |
                128, ((c1 >> 6) & 63) | 128, (c1 & 63)
                | 128);
        }
        if (enc !== null) {
            if (end > start) {
                utftext += string.slice(start, end);
            }
            utftext += enc;
            start = end = n + 1;
        }
    }
    if (end > start) {
        utftext += string.slice(start, stringl);
    }
    return utftext;
}

```

4.3.4. Funciones de enrutamiento

Función “enviarForm(form)”

Función que envía (submit) un formulario desde Javascript.

```
function enviarForm(form)
{
    form.submit();
}
```

Función “url_vars (vr)”

Función que te permite capturar una variable desde la URL.

```
function url_vars (vr)
{
    try{var src = String( window.location.href )
        .split('?')[1];
        var vrs = src.split('&');
        for (var x = 0, c = vrs.length; x < c; x++)
        {
            if (vrs[x].indexOf(vr) != -1)
            {
                return decodeURI( vrs[x].split('=')[1] );
                break;
            };
        };
        catch(Exception){
            return 0;
        }
    }
}
```

Función “detectaNavegador ()”

Función que detecta navegador.

```
function str_replace(busca, repla, orig)
{
    str      = new String(orig);
    rExp     = "/" + busca + "/g";
    rExp     = eval(rExp);
    newS     = String(repla);
    str = new String(str.replace(rExp, newS));
    return str;
}
```

4.3.5. Funciones de utilidad variada

Función “act_check(check_x,objeto_x)

Función que determina si se habilita o deshabilita un input según el check.

```
function act_check(check_x,objeto_x)
{
    if($("#"+check_x).is(':checked')){
        $("#"+objeto_x).attr("disabled",false);
    }
    else{
        $("#"+objeto_x).attr("disabled",true);
    }
}
```

5. Recomendaciones de uso

El framework BunnyPHP recomienda cumplir los siguientes puntos para lograr un uso óptimo y eficiente.

- Todas las consultas u operaciones sobre la Base de Datos deben ser programadas en procedimientos almacenados.
- Con respecto a los parámetros del procedimiento almacenado:
 - o Parámetro @xml (obligatorio): Este parámetro contiene todos los datos que necesita el procedimiento almacenado para ser ejecutado, todos empaquetados en xml. Este parámetro es obligatorio, en caso de que el procedimiento no necesite datos de entrada, @xml será una cadena vacía.
 - o Parámetro de retorno @rpta (opcional): Cuando el procedimiento devuelve una matriz de resultados (consulta SQL), la variable @rpta no será utilizada, por otro lado, en caso de que el retorno del procedimiento almacenado sea simplemente una variable, bastará utilizar el parámetro @rpta.
- BunnyPHP recomienda utilizar los siguientes prefijos al nombrar los procedimientos almacenados:
 - o Para consultas:

Esquema.paCon_(Acción+Objeto)

Ejemplo: Sistema.paCon_listarProductos

- Para inserciones:

Esquema.paIns_(Acción+Objeto)

Ejemplo: Sistema.paIns_agregarProducto

- Para actualizaciones:

Esquema.paAct_(Acción+Objeto)

Ejemplo: Sistema.paAct_deshabilitarAcceso

- Para eliminaciones:

Esquema.paEli_(Acción+Objeto)

Ejemplo: Sistema.paEli_limpiarUsuarios

- BunnyPHP recomienda utilizar AJAX cuando pueda ser aplicable, para optimizar la carga de páginas y agregar dinamismo a las mismas.
- Para utilizar de manera correcta el framework, BunnyPHP recomienda los siguientes pasos:
 - Comenzar a programar el nuevo módulo con su respectivo URI en un archivo de nombre “uriD_(NombreMódulo)”, dentro del directorio “Controlador”.
 - Continuar programando la interfaz con características propias del módulo en el directorio “Capa_Presentacion/Modulos”.
 - Utilizar, al momento de desarrollar la interfaz del módulo, las plantillas necesarias, las cuales deben estar incluidas en el directorio “Capa_Presentacion/Plantillas”.
 - Todos los estilos que compongan la interfaz independiente de cada módulo deben estar escritos en un archivo, dentro del directorio “Capa_Presentacion/Estilos”.
 - Con respecto a la funcionalidad del nuevo módulo, cuando se necesite utilizar el lenguaje de programación JavaScript, el código debe ser incluido en un archivo de extensión JS dentro del directorio “Capa_Presentacion/Funciones”.
 - Todas las nuevas funciones JavaScript programadas, que se consideren de necesidad y utilidad constante, pueden ser

incluidas dentro del archivo “bunnyJS.js”, ubicado en el directorio “Controlador”.

- Finalmente, es importante considerar que BunnyPHP se encuentra en su primera versión, por lo cual sus funcionalidades no son muchas en comparación con otros frameworks del mercado, pero cuenta con características peculiares y una dificultad de uso intermedia. Se espera que este proyecto siga creciendo y mejorando a partir de la experiencia de los usuarios que se atrevan a desarrollar aplicaciones utilizando este framework.

**ANEXO C: Documentación del Sistema de Grados y Títulos
(Desarrollado con el Framework BunnyPHP)**

CAPITULO 1

REQUERMIENTOS DEL SISTEMA

1.1. DESCRIPCIÓN GENERAL

Actualmente, la Universidad Peruana Los Andes cuenta con un Sistema Web de Grados y Títulos, el cual fue implementado en Octubre del año 2013. Este sistema cumple con los requerimientos iniciales que la “Oficina Universitaria de Informática y Sistemas” recopiló en un principio, para cumplir con el desarrollo de este sistema. Pero, en los días y meses siguientes a la implementación, sucedió, como es común en la mayoría de sistemas, que muchos de los requerimientos cambiaron, así como surgieron otros nuevos. Frente a esto, el personal de la “Oficina Universitaria de Informática y Sistemas” fue implementando dichos cambios, los cuales en un comienzo fueron simples, pues el sistema es medianamente escalable. Pero cada cambio en el sistema era, en cierta medida tedioso, y con el pasar del tiempo y el aumento de nuevos requerimientos, se ha ido volviendo cada vez más complejo, aunque su funcionalidad por la parte del usuario se mantiene consistente.

Por otro lado, de la modificación y el mantenimiento del sistema de grados y títulos que fue implementado en el año 2013, se encarga únicamente el personal que lo desarrolló, generando así una dependencia que podría llegar a ser problemática para la “Oficina Universitaria de Informática y Sistemas”, y por ende, para la Universidad Peruana Los Andes. Es por ello que se plantea el nuevo desarrollo del Sistema de Grados y Títulos, utilizando el framework “BunnyPHP”, un marco de trabajo que podría integrar a todos los sistemas y funcionar como única arquitectura, lo cual permitiría que cualquiera que conforme el personal de la “Oficina Universitaria de Informática y Sistemas” pueda dar mantenimiento a los sistemas y desarrolle nuevos módulos, con tan sólo conocer el funcionamiento de este framework.

1.2. ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES

1.2.1. REQUERIMIENTOS FUNCIONALES GENERALES

- En la “Sección de Grados y Títulos” de la “Oficina de Secretaría General” del Local Central, se podrá observar, realizar operaciones y generar los reportes requeridos con todos los estudiantes de la universidad, tanto los pertenecientes a Pregrado como los de Postgrado.
- La búsqueda podrá ser realizada introduciendo el DNI o Apellidos del estudiante. Al ejecutar la búsqueda se mostrarán las características

requeridas del estudiante o estudiantes (en caso de haber sido buscado por Apellidos y los estudiantes sean hermanos, o simplemente coincidan en los apellidos).

- Las características que arrojará la búsqueda por cada estudiante serán:

PREGRADO	POSTGRADO
Código del Estudiante	Código del Estudiante
DNI	DNI
Nombres y Apellidos	Nombres y Apellidos
Facultad	Escuela
Carrera	Especialidad
Especialidad	Mención
Modalidad de Sustentación	Modalidad de Sustentación
Tipo de Diploma (Bachiller y/o Título)	Tipo de Diploma (Maestría, Doctorado y/o Segunda Especialidad).

- Al mostrar los resultados, se podrá disponer de la opción de asignarle el grado o título que le corresponda según resolución.

	<i>En caso de...</i>	Se le puede otorgar...
Pregrado	No tener ni Grado ni Título Profesional.	GRADO ACADÉMICO DE BACHILLER
	Tener Grado Académico de BACHILLER	TÍTULO PROFESIONAL
Postgrado	Estar matriculado en MAESTRÍA	GRADO ACADÉMICO DE MAGISTER
	Estar matriculado en DOCTORADO	GRADO ACADÉMICO DE DOCTOR
	Estar matriculado en SEGUNDA ESPECIALIZACIÓN	GRADO ACADÉMICO EN SEGUNDA ESPECIALIZACIÓN

- Para otorgar un nuevo grado o título, se tendrá que rellenar una serie de datos según el diploma correspondiente. Dichos datos se detallan a continuación:

OTORGAR GRADO DE BACHILLER	
DNI	<i>Rellenado automáticamente</i>
Nombres y Apellidos	<i>Rellenado automáticamente</i>
Facultad	<i>Rellenado automáticamente</i>
Carrera	<i>Rellenado automáticamente</i>
Especialidad	<i>Rellenado automáticamente</i>
Modalidad de Sustentación	<i>Rellenado automáticamente</i>
Tipo de Diploma	<i>Rellenado automáticamente</i>

Fecha de Acto	<i>Se debe rellenar una fecha.</i>
Fecha de Sesión Universitaria	<i>Se debe rellenar una fecha.</i>
Resolución de Consejo Univ.	Rellenar manualmente
Fecha de Res. Consejo Univ.	<i>Se debe rellenar una fecha.</i>
Resolución de Cons. de Facultad	Rellenar manualmente
Código de Diploma	Rellenar manualmente
Fecha de Diploma	<i>Se debe rellenar una fecha.</i>
Autoridad: RECTOR	<i>Rellenado automáticamente</i>
Autoridad: SECRETARIO	<i>Rellenado automáticamente</i>
Autoridad: DECANO	<i>Rellenado automáticamente</i>
Observaciones	Rellenar manualmente (No debe ser obligatorio)

OTORGAR TÍTULO PROFESIONAL	
DNI	<i>Rellenado automáticamente</i>
Nombres y Apellidos	<i>Rellenado automáticamente</i>
Facultad	<i>Rellenado automáticamente</i>
Carrera	<i>Rellenado automáticamente</i>
Especialidad	<i>Rellenado automáticamente</i>
Modalidad de Sustentación	Elegir entre: <ul style="list-style-type: none"> - Sust. De Tesis - Exámen Profesional - Inf. Trab. Profesional - Otra Modalidad
Tipo de Diploma	<i>Rellenado automáticamente</i>
Fecha de Acto	<i>Se debe rellenar una fecha.</i>
Fecha de Sesión Universitaria	<i>Se debe rellenar una fecha.</i>
Resolución de Consejo Univ.	Rellenar manualmente
Fecha de Res. Consejo Univ.	<i>Se debe rellenar una fecha.</i>
Resolución de Cons. de Facultad	Rellenar manualmente
Código de Diploma	Rellenar manualmente
Fecha de Diploma	<i>Se debe rellenar una fecha.</i>
Autoridad: RECTOR	<i>Rellenado automáticamente</i>
Autoridad: SECRETARIO	<i>Rellenado automáticamente</i>
Autoridad: DECANO	<i>Rellenado automáticamente</i>
Observaciones	Rellenar manualmente (No debe ser obligatorio)

OTORGAR POSTGRADO (Maestría y Doctorado)	
DNI	<i>Rellenado automáticamente</i>
Nombres y Apellidos	<i>Rellenado automáticamente</i>
Escuela	<i>Rellenado automáticamente</i>
Especialidad	<i>Rellenado automáticamente</i>
Mención	<i>Rellenado automáticamente</i>
Modalidad de Sustentación	Elegir entre: <ul style="list-style-type: none"> - Sust. De Tesis - Exámen Profesional - Informe. Trab. Prof. - Otra Modalidad

Tipo de Diploma	<i>Rellenado automáticamente</i>
Fecha de Acto	<i>Se debe rellenar una fecha.</i>
Fecha de Sesión Universitaria	<i>Se debe rellenar una fecha.</i>
Resolución de Consejo Univ.	<i>Rellenar manualmente</i>
Fecha de Res. Consejo Univ.	<i>Se debe rellenar una fecha.</i>
Resolución de Cons. de Facultad	<i>Rellenar manualmente</i>
Código de Diploma	<i>Rellenar manualmente</i>
Fecha de Diploma	<i>Se debe rellenar una fecha.</i>
Autoridad: RECTOR	<i>Rellenado automáticamente</i>
Autoridad: SECRETARIO	<i>Rellenado automáticamente</i>
Autoridad: DIRECTOR DE GRADOS Y TÍTULOS	<i>Rellenado automáticamente</i>
Observaciones	<i>Rellenar manualmente (No debe ser obligatorio)</i>

OTORGAR POSTGRADO (Segunda Especialidad)	
DNI	<i>Rellenado automáticamente</i>
Nombres y Apellidos	<i>Rellenado automáticamente</i>
Escuela	<i>Rellenado automáticamente</i>
Especialidad	<i>Rellenado automáticamente</i>
Mención	<i>Rellenado automáticamente</i>
Modalidad de Sustentación	Elegir entre: <ul style="list-style-type: none"> - <i>Sust. De Tesis</i> - <i>Exámen Profesional</i> - <i>Informe. Trab. Prof.</i> - <i>Otra Modalidad</i>
Tipo de Diploma	<i>Rellenado automáticamente</i>
Fecha de Acto	<i>Se debe rellenar una fecha.</i>
Fecha de Sesión Universitaria	<i>Se debe rellenar una fecha.</i>
Resolución de Consejo Univ.	<i>Rellenar manualmente</i>
Fecha de Res. Consejo Univ.	<i>Se debe rellenar una fecha.</i>
Resolución de Cons. de Facultad	<i>Rellenar manualmente</i>
Código de Diploma	<i>Rellenar manualmente</i>
Fecha de Diploma	<i>Se debe rellenar una fecha.</i>
Autoridad: RECTOR	<i>Rellenado automáticamente</i>
Autoridad: SECRETARIO	<i>Rellenado automáticamente</i>
Autoridad: DECANO	<i>Rellenado automáticamente</i>
Observaciones	<i>Rellenar manualmente (No debe ser obligatorio)</i>

- Después de haber asignado el diploma correspondiente, se podrán editar datos como: Fecha de Acto, Fecha de Sesión Universitaria, Resolución de Consejo Universitario, Fecha de Resolución de Consejo Universitario, Resolución de Consejo de Facultad, Fecha de Diploma y Observaciones.

1.2.2. REQUERIMIENTOS FUNCIONALES DE REPORTES

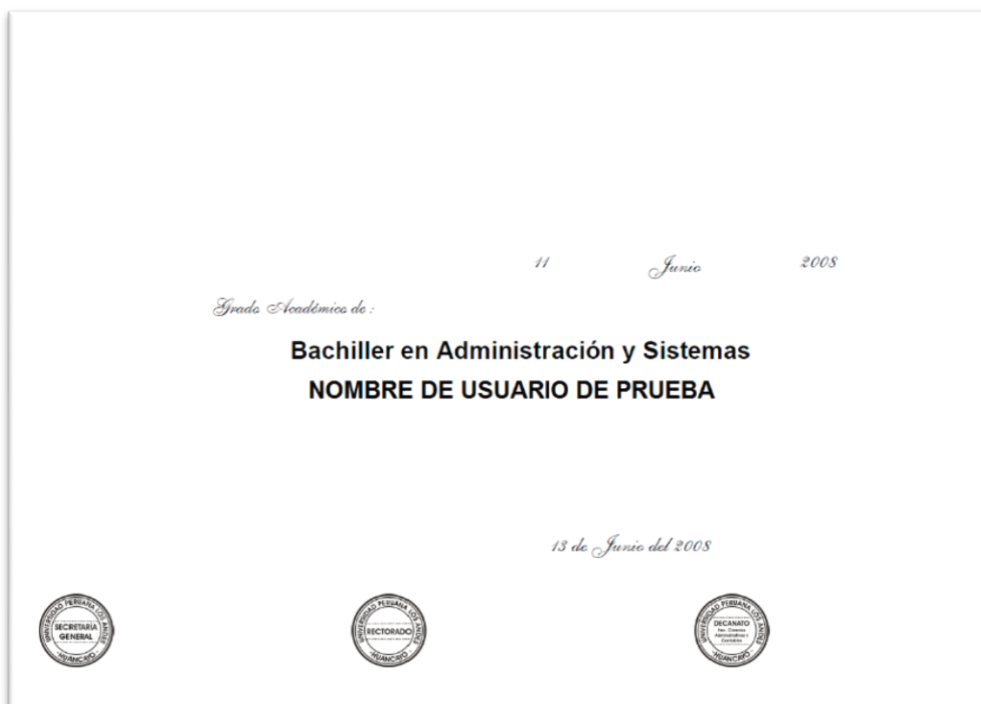
Entre los reportes que el Sistema debe generar, se encuentran los diplomas, una serie de reportes estadísticos e informes para la Asamblea General de Rectores.

Los reportes son los siguientes:

- Diploma
- Diploma (Reverso)
- Libro de Registros
- Informe ANR

A. DIPLOMA Y DIPLOMA REVERSO



El Diploma es aquel reporte que se entrega al estudiante, en el cual figura su nombre, su grado o título, características de este, además de la firma y sello de las autoridades. Este diploma es impreso sobre un formato de material especial, el cual ya tiene impreso el formato base del diploma, por tanto el reporte consistirá solamente en rellenar este formato. Por tanto los reportes deben ser de la siguiente manera:





B. LIBRO DE REGISTROS

El “Libro de Registros” es el reporte que se almacena en forma de libros en las Oficinas de Grados y Títulos. Por cada diploma que se emite, se guarda uno de estos reportes que contiene características del diploma, así como la firma del estudiante. Este reporte no necesita ningún formato especial, pues es impreso en una hoja A4 en blanco. El “Libro de Registros” es como el que se tiene a continuación:

UNIVERSIDAD PERUANA LOS ANDES LIBRO REGISTRO DE GRADOS ACADÉMICOS		00106
		
<i>Facultad:</i> _____	Ciencias Administrativas y Contables	OBSERVACIONES
<i>Carrera Profesional / Escuela Académica Profesional:</i> _____	ADMINISTRACIÓN Y SISTEMAS	
<i>Especialidad:</i> _____	SIN ESPECIALIDAD	
<i>El Consejo Universitario en sesión de:</i> _____	11 de Junio del 2008	
<i>Ha conferido el Grado Académico de Bachiller en:</i> _____	Bachiller en Administración y Sistemas	
<i>A Don (Doña):</i> _____	NOMBRE DE USUARIO DE PRUEBA	
<i>Registrado con el N°</i> 210100000364 <i>Expedido con fecha:</i> 13 de Junio del 2008		
<i>Resolución de Consejo Universitario N°</i> 587-2008-CU		
<i>Grado Académico Automático conforme al Decreto Legislativo N° 739 y Reglamento General de Grados y Títulos aprobado por Resolución N° 312-2001-CU</i>		
<i>Aprobado por el Consejo de Facultad con Resolución N°</i> 054-2008/CF-FCAC		
AUTORIDADES QUE SUSCRIBEN EL DIPLOMA N° A957173		
<i>Rector</i>	DOCTOR DIMAS FERNANDEZ BARRANTES	
<i>Secretario General</i>	ABOGADO VICTOR JULIO VILLAREAL SIFUENTES	
<i>Decano</i>	MAGISTER WILBER GONZALO VASQUEZ VASQUEZ	
<i>Fecha de Registro, Huancayo</i> 13 <i>de</i> Junio <i>de</i> 2008	REGISTRADOR	INTERESADO

C. FORMATO PARA EL REGISTRO NACIONAL DE GRADOS Y TÍTULOS (ANR)

Este es un reporte que se imprime en un formato proporcionado por la Asamblea Nacional de Rectores (ANR) y es emitido por la misma universidad, en este caso la “Universidad Peruana Los Andes”, el cual vale como documento garantizador de un grado o título determinado hacia la institución que lo requiera.

1.3. ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES

1.3.1. REQUERIMIENTOS DEL PRODUCTO

1.3.1.1. REQUERIMIENTOS DE USABILIDAD

Considerando la numerosa cantidad de datos que serán procesados por el sistema, es importante que este posea un funcionamiento veloz, para lograrlo, se hará uso del framework “BunnyPHP”, y de sus recomendaciones para la creación de procedimientos almacenados.

El sistema deberá tener una interfaz gráfica agradable e intuitiva, para que el usuario pueda familiarizarse rápidamente con la aplicación y obtener en poco tiempo un buen nivel de uso.

Del mismo modo, el sistema deberá ser escalable para soportar cambios en su estructura y programación, sin tener que ser modificado significativamente.

1.3.1.2. REQUERIMIENTOS DE EFICIENCIA

La cantidad de datos que se procesarán en este sistema es considerablemente grande, puesto que no sólo se procesan aquellos estudiantes que ya tienen un grado o título, ya sea en Pregrado o Postgrado, sino también se tienen en cuenta a todos los alumnos que se encuentran estudiando en la universidad sin ningún tipo de diploma, pues de ellos se genera la mayoría de Bachilleres y Titulados.

A pesar de interactuar con una gran cantidad de datos, este sistema debe ser veloz y utilizar la mínima cantidad posible de recursos de redes y servidor.

Se debe también tener en cuenta que esta aplicación no funcionará solamente en computadoras del local central, sino también se tendrá acceso a esta desde las diferentes facultades.

1.3.1.3. REQUERIMIENTOS DE FIABILIDAD

El sistema deberá validar los datos en el ingreso, de tal manera que la base de datos no tenga almacenada información inválida o duplicada.

Del mismo modo, deberá brindar al usuario la seguridad de que los datos almacenados no sufrirán pérdidas ni modificaciones indebidas.

Por otro lado, el sistema debe ofrecer al usuario, la confianza de que los datos no serán accedidos ni manipulados por personas sin autorización.

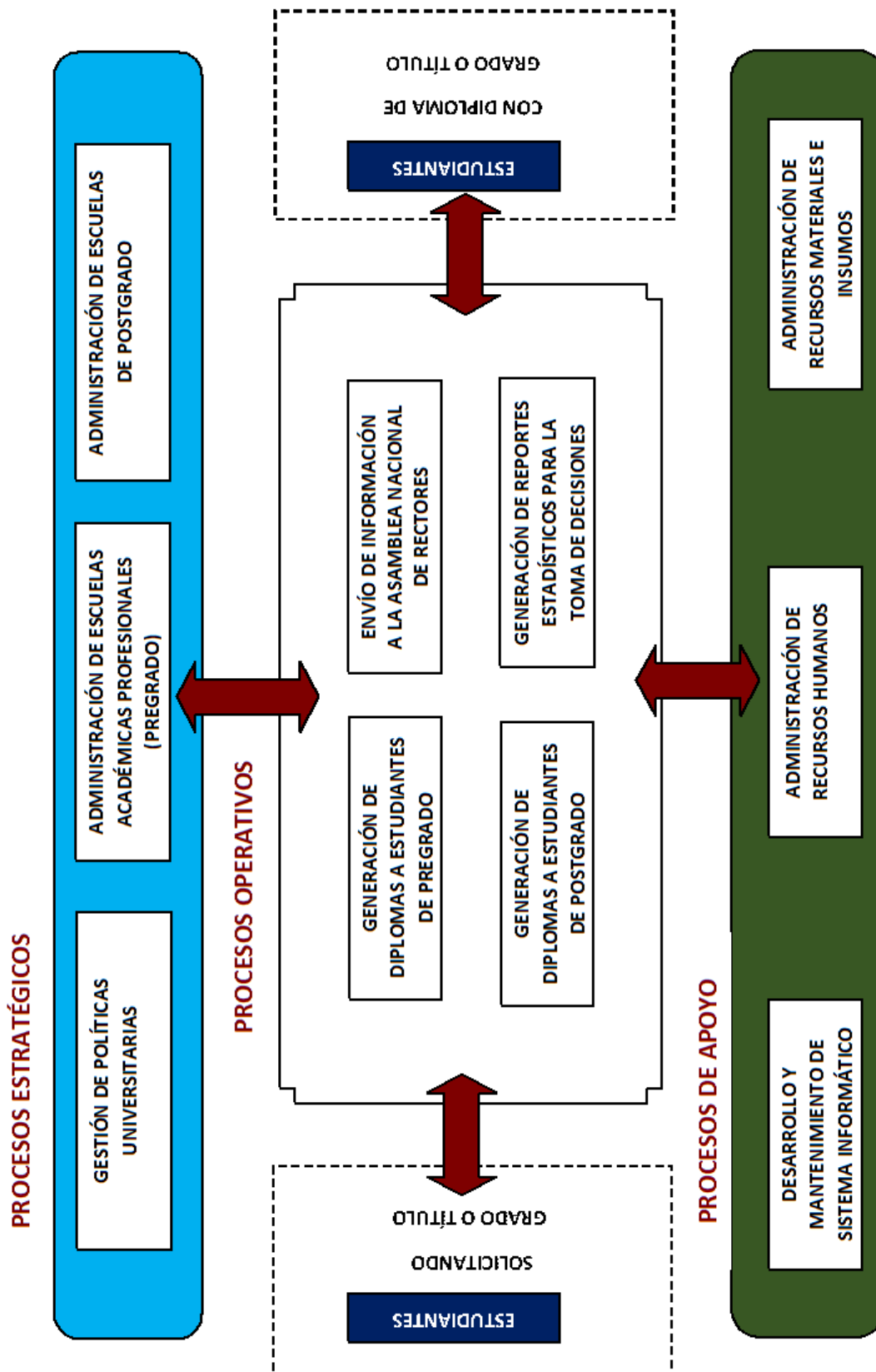
1.3.1.4. REQUERIMIENTOS DE PORTABILIDAD

El sistema deberá poder ser accedido desde cualquier computadora sin ningún requisito previo de software además del Navegador Web. Esto quiere decir que el sistema podrá correr sobre cualquier Sistema Operativo (Multiplataforma).

Del mismo modo, considerando que el sistema será ejecutado desde el Navegador Web del cliente, este deberá correr en cualquiera de ellos, sin importa que sea Mozilla Firefox, Google Chrome, Opera, Internet Explorer, etc.

1.4. MAPA DE PROCESOS DE LA SECCIÓN DE GRADOS Y TÍTULOS

(El Mapa de Procesos se encuentra en la siguiente página).



1.5. GLOSARIO DE TÉRMINOS

- **Pregrado:** Son todos los estudios superiores hasta el Título Profesional. Estos son necesarios para poder acceder a los estudios de Postgrado. Sirven para preparar al estudiante para el desempeño de ocupaciones, para el ejercicio de una profesión o disciplina determinada, de naturaleza tecnológica o científica o en el área de humanidades, las artes y la filosofía.
- **Postgrado:** Son todos los estudios universitarios posteriores al Título Profesional y comprenden los estudios de Maestría (también denominados máster o magíster), Doctorado y Segunda Especialidad.
- **Facultad:** Sección de la universidad que corresponde a una rama del saber, organiza los estudios de varias carreras y expide los títulos. En la Universidad Peruana Los Andes, hasta la presentación de esta documentación se cuenta con 6 Facultades:
 - Facultad de Ingeniería
 - Facultad de Ciencias Administrativas y Contables
 - Facultad de Derecho y Ciencias Políticas
 - Facultad de Educación y Ciencias Humanas
 - Facultad de Ciencias de la Salud
 - Facultad de Medicina Humana
- **Carrera:** Es un conjunto de cursos académicos que una persona debe completar para poder obtener un título profesional. Por ejemplo, la carrera de Ingeniería de Sistemas y Computación.
- **Especialidad:** Es un conjunto de cursos académicos que una persona debe completar para poder obtener un título profesional y que divide una “Carrera” cuando esta es lo suficientemente grande. Por ejemplo, la carrera de Educación Secundaria, la cual se divide en especialidades como “Lengua y Literatura”.
- **Mención:** Es la descripción del grado obtenido. Las menciones están contenidas en todas las Maestrías, y en algunos casos en Segunda Especialidad.
- **Grado Académico:** Es una distinción dada por la universidad, generalmente después de la terminación exitosa de algún programa de estudios. Esta denominación suele utilizarse para denominar más concretamente a las distinciones de rango universitario. En la universidad se tienen los grados académicos de Bachiller, Maestría, Doctorado y Segunda Especialidad.
 - **Bachiller:** Es un grado académico que se emite automáticamente al egresado luego de haber concluido satisfactoriamente su carrera

profesional. Este grado académico permite al egresado postular para obtener el Título Profesional y/o optar por un Programa de Maestría.

- **Maestría:** Es un grado académico que se emite luego de haber concluido satisfactoriamente un Programa de Maestría y haber sustentado el informe respectivo.
 - **Doctorado:** Es un grado académico que se emite luego de haber concluido satisfactoriamente un Programa de Doctorado y haber sustentado el informe respectivo.
 - **Segunda Especialidad:** Es un grado académico que se emite luego de haber concluido satisfactoriamente un Programa de Segunda Especialización y haber sustentado el informe respectivo.
-
- **Título Profesional:** Reconocimiento de validez oficial de estudios en alguna carrera, a los estudiantes egresados que cumplen con los requisitos académicos y administrativos necesarios, entre los cuales se encuentra el grado académico de Bachiller.
 - **Diploma:** Documento oficial expedido por la universidad que representa un grado académico o título profesional.
 - **Consejo de Facultad:** Es el órgano de gobierno máximo de cada Facultad, encargado de establecer las políticas y normas de gestión académica. El Consejo de Facultad sesiona obligatoriamente, cada treinta días y extraordinariamente cuando lo convoque el Decano o a solicitud escrita de la mitad más uno de sus miembros, precisando su objetivo.
 - **Consejo Universitario:** El Consejo Universitario es la máxima autoridad de la Universidad. Sus funciones son elaborar las normas y disposiciones que ayuden a una mejor organización y funcionamiento de la Universidad en sus aspectos académicos y técnicos.
 - **Asamblea Nacional de Rectores (ANR):** La Asamblea Nacional de Rectores es un organismo público autónomo constituido por los Rectores de las Universidades Públicas y Privadas, como ente rector de estudio, coordinación y orientación de las actividades universitarias del país.

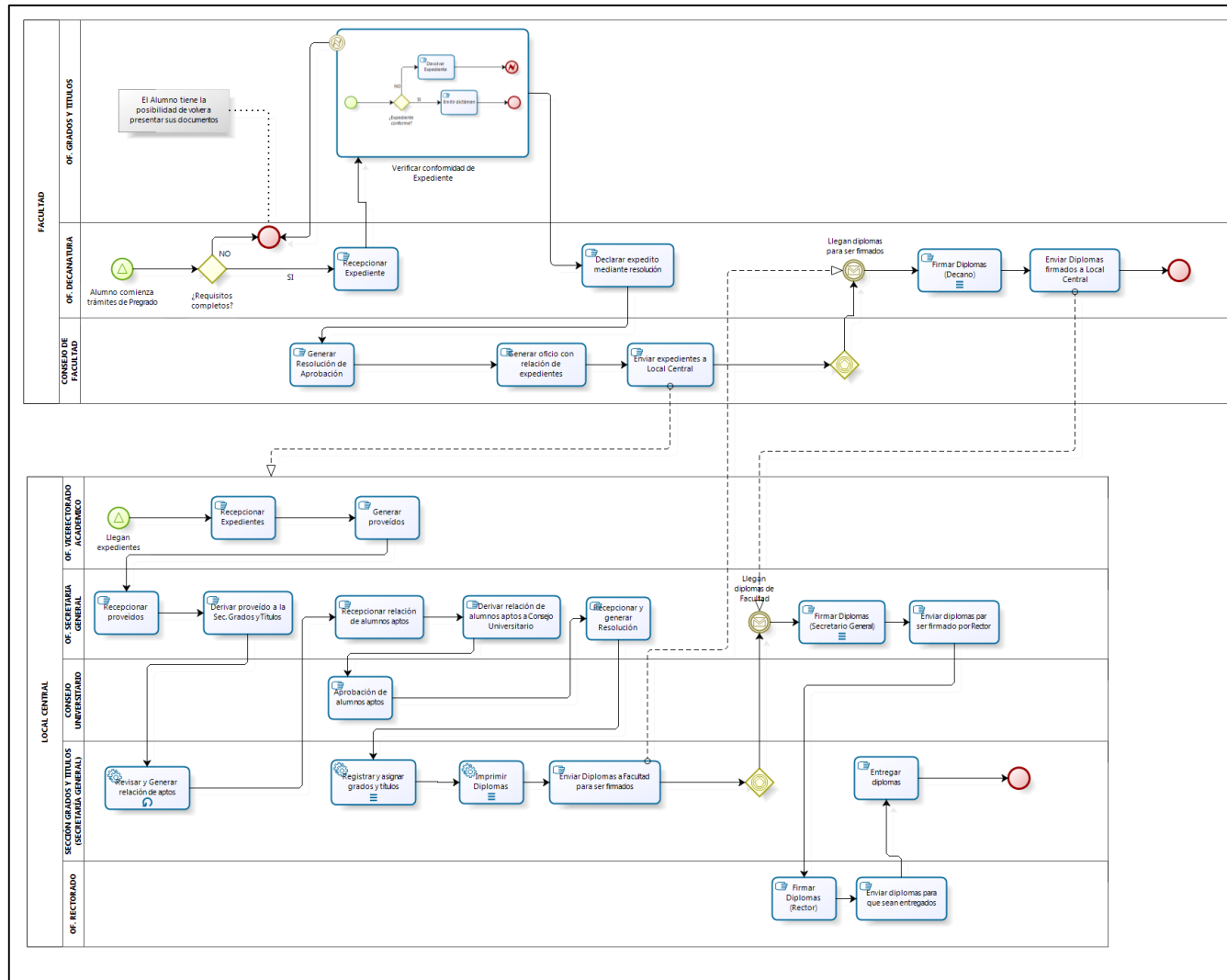
CAPITULO 2

DISEÑO DEL SISTEMA

2.1. DIAGRAMAS DE PROCESO DEL NEGOCIO

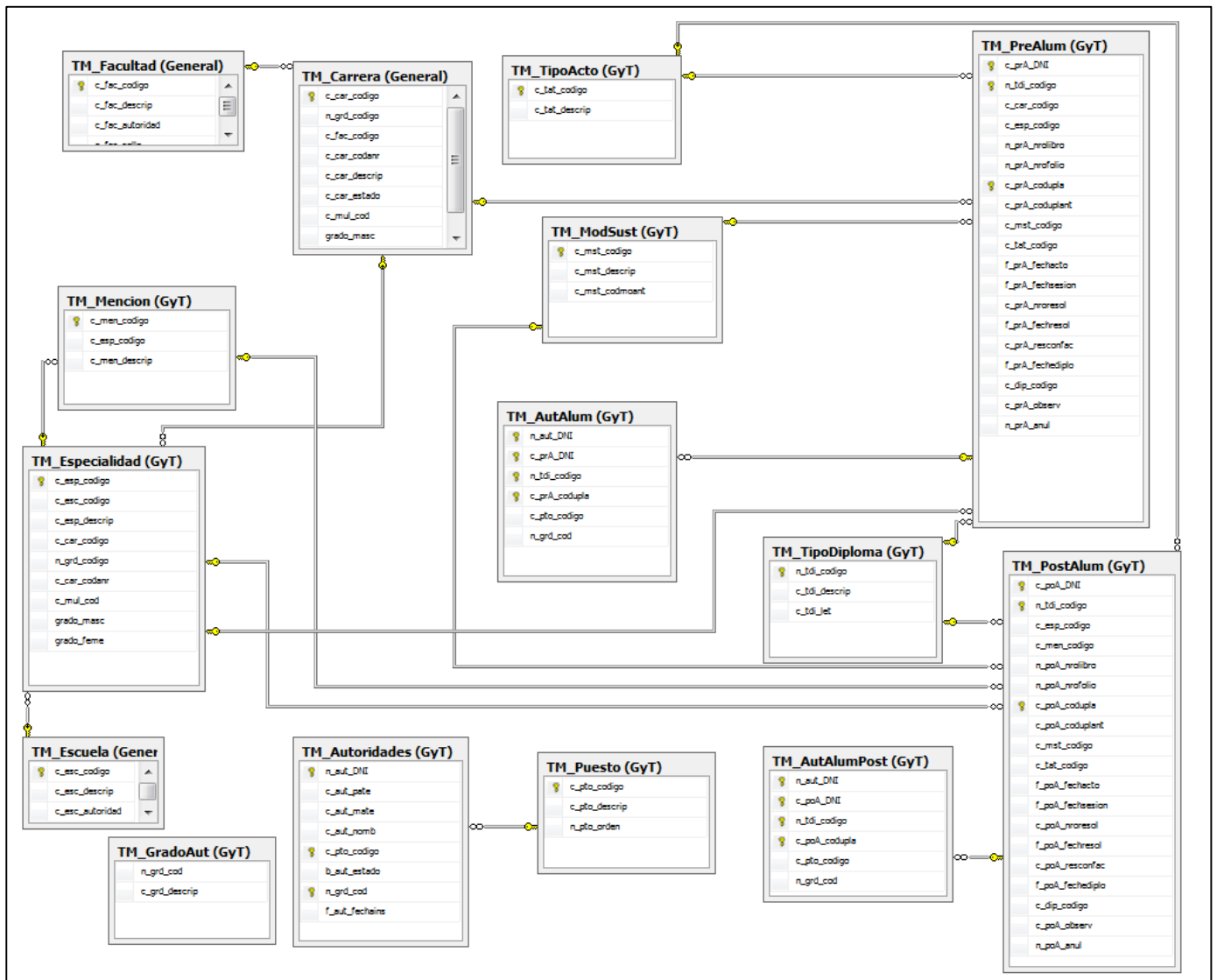
Partiendo del “Mapa de Procesos” diagramado en la siguiente imagen, son diagramados los procesos operativos con la “Notación para el Modelado de Procesos del negocio” (BPMN). Los procesos son:

- Generación de Diplomas a Estudiantes de Pregrado.
- Generación de Diplomas a Estudiantes de Postgrado.
- Envío de Información a la Asamblea Nacional de Rectores.
- Generación de Reportes Estadísticos para la toma de Decisiones.



Proceso: Generación de diplomas de estudiantes de pregrado (Bachiller y Título)

2.2. DIAGRAMA DE LA BASE DE DATOS



2.3. DISEÑO DE LA INTERFÁZ GRÁFICA

2.3.1. INTERFAZ GRÁFICA DEL MÓDULO DE ACCESO



RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Módulo de Acceso de Usuarios
TIPO DE MÓDULO	Página completa
USUARIOS PERMITIDOS	Todos
DESCRIPCIÓN	Este módulo comprueba el DNI y la contraseña del usuario. Si el usuario es válido, obtiene todos los datos de este, entre los cuáles sobresalen el “Privilegio de Acceso” y la “Dependencia”.

2.3.2. INTERFAZ GRÁFICA DE LA PÁGINA DE INICIO



RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Módulo de Inicio (Página Principal)
TIPO DE MÓDULO	Página completa
DESCRIPCIÓN	Esta es la página que se carga automáticamente luego de ingresar al sistema. Para comenzar, esta página contiene el menú principal y un banner de presentación.

2.3.3. INTERFAZ GRÁFICA DE PREGRADO

Código	DNI	Nombres y Apellidos	Facultad	Carrera	Especialidad	Modalidad de Sustentación	Tipo de Diploma	Estado
C00279H	70545283	EDUARDO JESUS RIOS MONTERREY	INGENIERÍA	ARQUITECTURA	----	AUTOMATICO	BACHILLER	Entregado
C00279H	70545283	EDUARDO JESUS RIOS MONTERREY	INGENIERÍA	ARQUITECTURA	----	SUSTENTACION DE TESIS	TITULO	En Proceso
A80073H	70545282	JOSE ANTONIO RIOS MONTERREY	INGENIERÍA	INGENIERÍA DE SISTEMAS Y COMPUTACIÓN	----	AUTOMATICO	BACHILLER	Listo
A80073H	70545282	JOSE ANTONIO RIOS MONTERREY	INGENIERÍA	INGENIERÍA DE SISTEMAS Y COMPUTACION	----	SUSTENTACION DE TESIS	TITULO	Listo
A620573	44068957	LILIANA ELVA RIOS MONTERREY	MEDICINA HUMANA	MEDICINA HUMANA	----	NO TIENE	NO TIENE	Sin Diploma
A22022H	42403362	PABLO VALENTIN RIOS MONTERREY	CIENCIAS ADMINISTRATIVAS Y CONTABLES	ADMINISTRACIÓN Y SISTEMAS	----	AUTOMATICO	BACHILLER	Entregado
A22022H	42403362	PABLO VALENTIN RIOS MONTERREY	CIENCIAS ADMINISTRATIVAS Y CONTABLES	ADMINISTRACIÓN Y SISTEMAS	----	OTRA MODALIDAD	TITULO	Entregado

RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Estudiantes de Pregrado
TIPO DE MÓDULO	Página completa
DESCRIPCIÓN	Este módulo contiene el Menú Principal, un banner, una caja de texto para introducir búsqueda y finalmente una grilla que lista a los alumnos de “Pregrado”. Luego de listar a los alumnos, permite opciones de asignación de nuevos diplomas y generación de reportes individuales.

2.3.4. INTERFAZ GRÁFICA DE POSTGRADO



RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Estudiantes de Postgrado
TIPO DE MÓDULO	Página completa
DESCRIPCIÓN	Este módulo contiene el Menú Principal, un banner, una caja de texto para introducir búsqueda y finalmente una grilla que lista a los alumnos de “Postgrado”. Luego de listar a los alumnos, permite opciones de asignación de nuevos diplomas y generación de reportes individuales.

2.3.5. INTERFAZ GRÁFICA DE RESOLUCIONES EN PROCESO

Resolución	Fecha de Resolución	# Diplomas	Nivel
166-93-CO	1993-03-19	1	Pregrado
312-93-CO	1993-06-04	1	Pregrado
0535-2014-CUVRAC	2014-05-15	89	Pregrado
0536-2014-CUVRAC	2014-05-15	12	Pregrado
0538-2014-CUVRAC	2014-05-15	44	Pregrado
0539-2014-CUVRAC	2014-05-15	1	Pregrado
0541-2014-CUVRAC	2014-05-15	1	Pregrado
0542-2014-CUVRAC	2014-05-15	1	Pregrado
0544-2014-CUVRAC	2014-05-15	1	Pregrado
0545-2014-CUVRAC	2014-05-15	1	Pregrado
0546-2014-CUVRAC	2014-05-15	1	Pregrado
0547-2014-CUVRAC	2014-05-15	1	Pregrado
0548-2014-CUVRAC	2014-05-15	3	Pregrado
0550-2014-CUVRAC	2014-05-15	53	Pregrado
0556-2014-CUVRAC	2014-05-15	1	Pregrado
0557-2014-CUVRAC	2014-05-15	1	Postgrado
0558-2014-CUVRAC	2014-05-15	1	Postgrado
0559-2014-CUVRAC	2014-05-15	1	Postgrado
0560-2014-CUVRAC	2014-05-15	1	Postgrado
RESOL-2130-2014-08-19	2014-08-19	1	Pregrado

RESUMEN DEL MÓDULO

NOMBRE DEL MÓDULO	Administración de Resoluciones
TIPO DE MÓDULO	Página completa con Reporte
DESCRIPCIÓN	Este módulo contiene el Menú Principal, un banner y la lista de Resoluciones “En Proceso”. Al seleccionar una de las resoluciones y cambiar su estado a “Lista para Entregar”, permitirá que todos los diplomas que conforman esta resolución cambien su estado a “Listo para Entregar”.

2.3.6. INTERFAZ GRÁFICA DE CAMBIAR CONTRASEÑA

The screenshot shows a web application interface with a dark blue header. The header contains a navigation menu with icons and labels: 'INICIO', 'ALUMNOS DE PREGRADO', 'ALUMNOS DE POSTGRADO', 'RESOLUCIONES EN PROCESO', 'ADMINISTRACIÓN Y CONFIGURACION', and 'SALIR DEL SISTEMA'. Below the menu is a large banner with the text 'ADMINISTRACIÓN Y CONFIGURACIONES' and an icon of a computer monitor. The main content area is titled 'Cambiar Contraseña' and contains three input fields: 'Contraseña Actual' (with placeholder 'Contraseña actual'), 'Nueva Contraseña' (with placeholder 'Nueva contraseña'), and 'Comprobación Contraseña' (with placeholder 'Confirme nueva contraseña'). A 'Cambiar' button is located below the fields.

RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Cambiar contraseña
TIPO DE MÓDULO	Página completa
DESCRIPCIÓN	Este módulo contiene el Menú Principal, un banner y un formulario que, luego de ingresar a forma de comprobación la contraseña actual del usuario, y dos veces la nueva contraseña, permite actualizar la contraseña de acceso al sistema.

2.3.7. INTERFAZ GRÁFICA DE REGISTRAR USUARIO

The screenshot shows a web application interface with a dark blue header. The header contains a navigation menu with icons and labels: 'INICIO', 'ALUMNOS DE PREGRADO', 'ALUMNOS DE POSTGRADO', 'RESOLUCIONES EN PROCESO', 'ADMINISTRACIÓN Y CONFIGURACIONES', and 'SALIR DEL SISTEMA'. Below the navigation is a large banner with the text 'ADMINISTRACIÓN Y CONFIGURACIONES'. The main content area is titled 'Registrar Usuario' and contains a form with the following fields:

- Código o DNI:** Ingrese Código o DNI
- Nombre:** Ingrese Nombre
- Ap. Paterno:** Ingrese Ap. Paterno
- Ap. Materno:** Ingrese Ap. Materno
- Sexo:** ...
- Privilegio:** elegir...
- Contraseña:** Contraseña <--> Contraseña

A 'Registrar' button is located at the bottom right of the form.

RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Registrar Usuario
TIPO DE MÓDULO	Página completa
DESCRIPCIÓN	Este módulo contiene el Menú Principal, un banner y un formulario que permite ingresar los datos del nuevo usuario a crear, en este formulario se puede seleccionar también el privilegio de acceso que se le asignará al usuario.

2.3.8. INTERFAZ GRÁFICA DE RESTAURAR CONTRASEÑA

INICIO ALUMNOS DE PREGRADO ALUMNOS DE POSTGRADO RESOLUCIONES EN PROCESO ADMINISTRACIÓN Y CONFIGURACIÓN SALIR DEL SISTEMA

ADMINISTRACIÓN Y CONFIGURACIONES

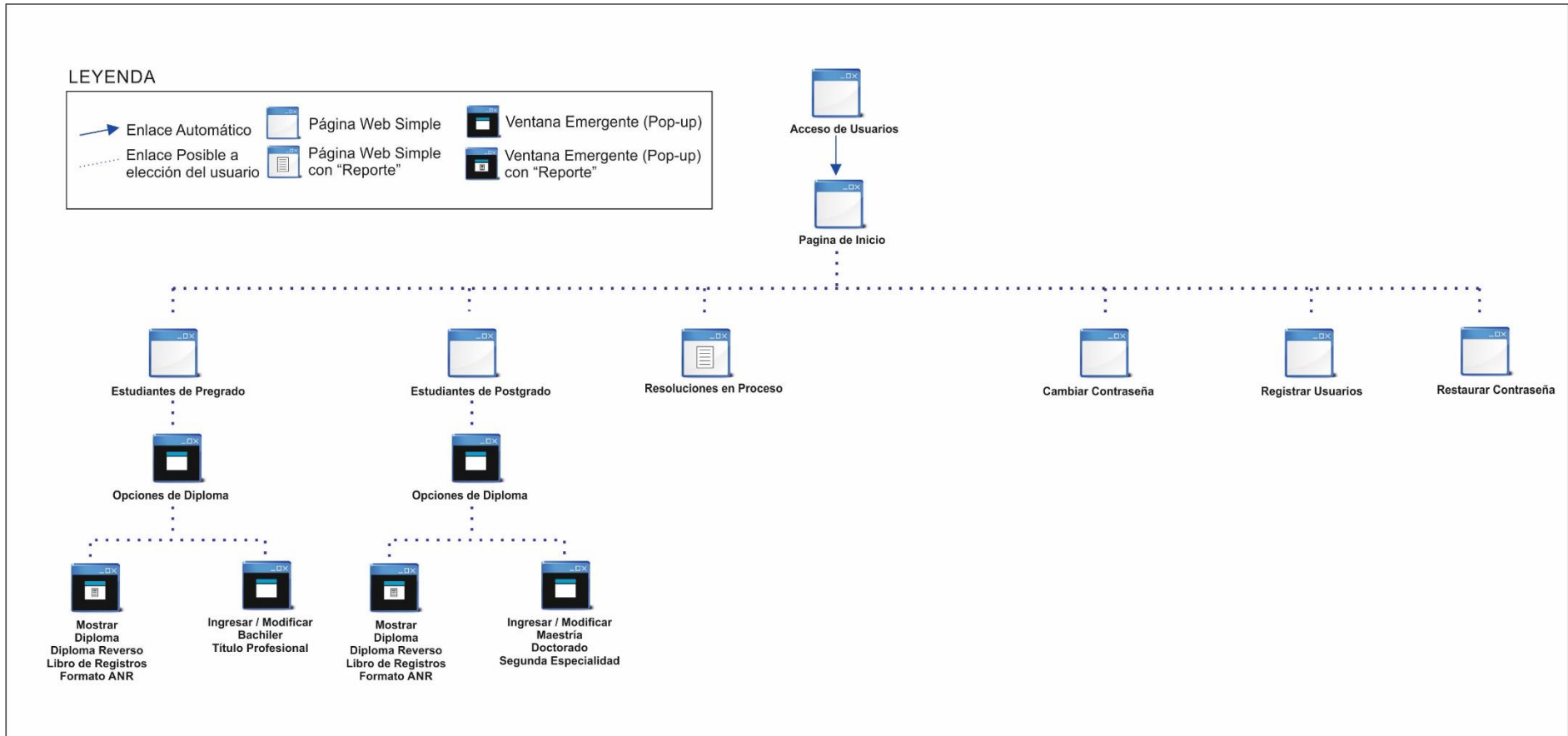
Restaurar contraseña de usuarios

Código o DNI:

[Restablecer](#)

RESUMEN DEL MÓDULO	
NOMBRE DEL MÓDULO	Restaurar contraseña de usuarios
TIPO DE MÓDULO	Página completa
DESCRIPCIÓN	Este módulo contiene el Menú Principal, un banner y un campo de texto que permite ingresar el DNI de un usuario, lo cual permitirá restaurar su contraseña cambiándola a su número de DNI.

3.1. DIAGRAMA DE MÓDULOS



ANEXO D: Actas de Aceptación de Proyecto

ACTA DE ACEPTACIÓN DEL PROYECTO (OFICINA UNIVERSITARIA DE INFORMÁTICA Y SISTEMAS)

ACTA DE ACEPTACIÓN DE PROYECTO

NOMBRE DEL PROYECTO	SIGLAS DEL PROYECTO
FRAMEWORK PARA EL DESARROLLO DE APLICACIONES WEB	FRAMEWORK
NOMBRE DEL CLIENTE O SPONSOR	
CLIENTE: OFICINA UNIVERSITARIA DE INFORMÁTICA Y SISTEMAS (UNIVERSIDAD PERUANA LOS ANDES)	

DECLARACIÓN DE LA ACEPTACIÓN FORMAL

Por la presente se deja constancia de que el Proyecto "FRAMEWORK PARA EL DESARROLLO DE APLICACIONES WEB" a cargo de los bachilleres **JOSÉ ANTONIO RIOS MONTERREY** y **EMILY ROSSY SALINAS CHAMORRO**, ha sido aceptado y aprobado por el Ing. Jowel Cabrera Padilla y personal de la "Oficina Universitaria de Informática y Sistemas". Damos constancia por la presente que ha sido desarrollado exitosamente.

El proyecto comprendía la entrega de los siguientes entregables:

- 1. Copia limpia del Framework (En DVD de datos)**
 - 1.1. Parte de la Aplicación.
 - 1.2. Tablas de Seguridad (Base de Datos).
 - 1.3. Procedimientos y Funciones Almacenadas de Seguridad (Base de Datos).
- 2. Documentación del Framework (En DVD de datos)**
 - 2.1. Descripción General.
 - 2.2. Librerías utilizadas.
 - 2.3. Instalación
 - 2.4. Arquitectura
 - 2.5. Sistema de Protección de Accesos.
 - 2.6. Utilización de un "Control de Versiones" para el desarrollo "Cooperativo".
 - 2.7. Documentación de funciones útiles para el desarrollador.
 - 2.8. Recomendaciones de uso.
- 3. Copia del Nuevo Sistema de Grados y Títulos (En DVD de datos)**
 - 3.1. Aplicación.
 - 3.2. Tablas (Base de Datos).
 - 3.3. Procedimientos y Funciones Almacenadas (Base de Datos).
- 4. Documentación del Nuevo Sistema de Grados y Títulos (En DVD de datos)**
 - 4.1. Requerimientos del Sistema.
 - 4.2. Diseño del Sistema.
 - 4.3. Diagrama de Módulos.

El proyecto fue iniciado el 01 de Julio del 2014, y terminó el 12 de Diciembre del 2014.

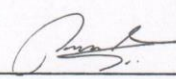
OBSERVACIONES ADICIONALES

ACEPTADO POR


NOMBRE DEL CLIENTE, SPONSOR U OTRO FUNCIONARIO	CARGO	FECHA
ING. JOWEL CABRERA PADILLA	Jefe de la Oficina de Informática	30-06-2015
BACH. ROLANDO PACUAR CAMARGO	Analista - Programador de Sistemas	30-06-2015
ING. ARTURO MARTINEZ BRAVO	Administrador de Base de Datos	30-06-2015



ING. JOWEL CABRERA PADILLA



BACH. ROLANDO PAUCAR CAMARGO



ING. ARTURO MARTINEZ BRAVO

ACTA DE ACEPTACIÓN DEL PROYECTO (OFICINA UNIVERSITARIA DE INFORMÁTICA Y SISTEMAS)

ACTA DE ACEPTACIÓN DE PROYECTO

NOMBRE DEL PROYECTO NUEVO SISTEMA DE GRADOS Y TÍTULOS (UNIVERSIDAD PERUANA LOS ANDES)	SIGLAS DEL PROYECTO GYT
NOMBRE DEL CLIENTE O SPONSOR CLIENTE: UNIDAD DE GRADOS Y TÍTULOS (UNIVERSIDAD PERUANA LOS ANDES)	

DECLARACIÓN DE LA ACEPTACIÓN FORMAL

Por la presente se deja constancia de que el Proyecto "NUEVO SISTEMA DE GRADOS Y TÍTULOS" a cargo de los bachilleres **JOSÉ ANTONIO RÍOS MONTERREY** y **EMILY ROSSY SALINAS CHAMORRO**, ha sido aceptado y aprobado para gestionar su próxima implementación por el Lic. Ángel Salvador Velazco Gonzales, damos constancia por la presente ha sido desarrollado exitosamente.

El proyecto comprendía la entrega de los siguientes entregables:

1. Copia del Nuevo Sistema de Grados y Títulos (En DVD de datos)

- 1.1. Aplicación.
- 1.2. Tablas (Base de Datos).
- 1.3. Procedimientos y Funciones Almacenadas (Base de Datos).

2. Documentación del Nuevo Sistema de Grados y Títulos (En DVD de datos)

- 2.1. Requerimientos del Sistema.
- 2.2. Diseño del Sistema.
- 2.3. Diagrama de Módulos.

El proyecto fue iniciado el 15 de Enero del 2015, y terminó el 15 de mayo del 2015.

OBSERVACIONES ADICIONALES

ACEPTADO POR

NOMBRE DEL CLIENTE, SPONSOR U OTRO FUNCIONARIO	CARGO	FECHA
LIC. ÁNGEL SALVADOR VELAZCO GONZALES	Responsable de la Unidad de Grados y Títulos	30-06-2015
S.E. DAYSI CINTIA VELIZ KAQUE	Secretaria de Grados y Títulos	30-06-2015



[Handwritten signature of Ángel Salvador Velazco Gonzales]

LIC. ÁNGEL SALVADOR VELAZCO GONZALES

[Handwritten signature of Daysi Cintia Veliz Kaque]

S.E. DAYSI CINTIA VELIZ KAQUE