

Universidad Autónoma del Estado de México

Centro Universitario UAEM Texcoco



Aplicación del Modelo Incremental Para el Desarrollo del
Sistema de Información Docente

Que Para Obtener el Título de
Ingeniería en Computación

Presenta

José David Sánchez Vázquez

Directora

M. en C. C. Dolores Arévalo Centeno

Revisores

M. En C. C. Leticia Arévalo Cedillo

M. En C. C. Juan Carlos Moreno Sánchez

Texcoco, Estado de México [Fecha de publicación]



Universidad Autónoma del Estado de México
Centro Universitario UAEM Texcoco

Texcoco, México a 8 de Junio de 2018

Asunto: Etapa de digitalización

M. EN C. E. VIRIDIANA BANDA ARZATE
SUBDIRECTORA ACADÉMICA DEL
CENTRO UNIVERSITARIO UAEM TEXCOCO
PRESENTE:

AT'N: M. en C. Leticia Arévalo Cedillo
RESPONSABLE DEL DEPARTAMENTO DE TITULACIÓN

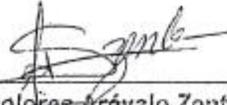
Con base en las revisiones efectuadas al trabajo escrito titulado "Aplicación del Modelo Incremental para el Sistema de Información Docente", que para obtener el título de Licenciado en Ingeniería en Computación presenta el sustentante José David Sánchez Vázquez con número de cuenta 1028533 se concluye que cumple con los requisitos teórico-metodológicos por lo que se otorga el voto aprobatorio para su sustentación, pudiendo continuar con la etapa de digitalización del trabajo escrito.

2018

M. en C. C. Juan Carlos Moreno Sánchez

ATENTAMENTE

M. en C. Leticia Arévalo Cedillo


M. en C. C. Dolores Arévalo Zenteno

c.c.p José David Sánchez Vázquez
c.c.p M. en C. C. Dolores Arévalo Centeno
c.c.p Titulación/ Lic. En D. Marco Rodrigo López González

**TITULACION
RECIBIDO**
Por L. Arévalo
Texcoco, Méx., a 08 de Junio del 18

Centro Universitario UAEM Texcoco
Av. Jardín de Zumpango s/n Fr. cc. El Tejocote, C. P. 56259,
Texcoco, Estado de México

DEDICATORIAS

Este presente trabajo quiero dedicárselo primeramente a mi madre, que, gracias a ella, deposito en mi toda su confianza a lo largo de toda mi trayectoria académica para salir adelante, el cual siempre estaré infinitamente agradecido con ella.

También quiero dedicárselo a mi hermana, que, gracias a sus mil maneras de motivarme, logre concluir unos de los proyectos más importantes en mi vida, ojalá y esto te sirva como ejemplo de que todo lo que te propongas lo podrás conseguir, con esfuerzo y dedicación hermana.

Además, quiero dedicárselo a mi abuelita, tíos y primos que me han apoyado en todo momento y que están cerca de mí. Quiero también dedicárselo a mi abuelito y mi tía Amada, que en donde quiera que se encuentren, sé que estarían orgullosos de mí.

A todos ustedes.

¡Gracias!

AGRADECIMIENTOS

Quiero agradecer primeramente a la Universidad Autónoma del Estado de México por aceptarme en su honorable plantilla de estudiantes y que fui formado en el Centro Universitario UAEM Texcoco.

A todos los profesores, les agradezco por todo lo que me enseñaron durante mi trayectoria académica dentro del Centro Universitario, sin ustedes y sus enseñanzas, no hubiera logrado mis metas.

A mis amigos con los que compartí buenas y malas experiencias durante la carrera.

Quiero agradecer a la profesora Dolores Arévalo por brindarme la confianza y la oportunidad para realizar este proyecto. Gracias por todos los consejos y enseñanzas que me brindo, este trabajo también es de usted.

Agradezco a mis revisores, la profesora Leticia y el profesor Juan Carlos por su dedicación, comentarios, observaciones tiempo y compromiso realizados durante el desarrollo del presente trabajo.

A todos ustedes.

¡Gracias!



Contenido

INTRODUCCIÓN	1
DEFINICIÓN DEL PROBLEMA.....	2
METODOLOGÍA.....	2
JUSTIFICACIÓN.....	3
OBJETIVO.....	4
Objetivos Específicos.....	4
CAPÍTULO 1 ESTADO DEL ARTE	5
1.1 Evolución de la Web cómo Plataforma.....	6
1.1.1 De la web 1.0 a la 3.0	6
1.1.2 Web 4.0.....	6
1.2 Lenguajes de Programación Web	7
1.2.1 Lenguajes de Etiquetas	8
1.3 SPA (Single Page Application).....	8
1.4 Desarrollo Back-End	8
1.5 Desarrollo Front-End	9
1.6 Responsive Web Design	9
1.7 Web Components.....	10
1.7.1 Custom Elements.....	10
1.7.2 Templates	10
1.7.3 Shadow DOM.....	10
1.7.4 HTML Imports	11
1.7.5 Librerías Web Components	11
1.8 Cloud-Computing	12
CAPÍTULO 2 MODELOS DE DESARROLLO DE SOFTWARE	13
2.1 Introducción.....	13
2.2 Modelo de Desarrollo en Cascada	14
2.2.1 Antecedentes.....	14
2.2.2 Modelo de Royce.....	14
2.3 Modelo de Desarrollo en Espiral	18
2.3.1 Antecedentes.....	18
2.3.2 Características del Modelo en Espiral.....	18
2.3.3 Plan de Gestión de Riesgos	19
2.4 Modelo de Desarrollo en V.....	20
2.4.1 Antecedentes.....	20
2.4.2 Modelo en V-Típico.....	20
2.4.3 Modelo en V-Avanzado.....	20

2.4.4	Pruebas de Gestión	21
2.5	Modelo de Desarrollo Incremental.....	23
2.5.1	Antecedentes.....	23
2.5.2	Descripción del Modelo.....	23
2.5.3	Ingeniería de Integración	23
CAPÍTULO 3 PATRONES DE DISEÑO		25
3.1	Introducción.....	25
3.2	Patrones de Creación.....	26
3.2.1	Abstract Factory.....	26
3.2.2	Factory Method.....	27
3.2.3	Prototype	28
3.3	Patrones de Estructura.....	30
3.3.1	Adapter	30
3.3.2	Composite.....	31
3.3.3	Decorator	32
3.4	Patrones de Comportamiento.....	34
3.4.1	Chain of Responsibility	34
3.4.2	Mediator.....	35
3.4.3	Template Method	36
3.5	Modelo-Vista-Controlador.....	38
3.5.1	Ejemplo De Implementación	39
CAPÍTULO 4 TENDENCIAS TECNOLÓGICAS.....		40
4.1	Introducción.....	40
4.2	Realidad Virtual	41
4.2.1	Tipos de Realidades	41
4.2.2	Aplicaciones.....	42
4.3	Realidad Aumentada	43
4.4	Inteligencia Artificial.....	45
4.4.1	Tendencias de IA	46
4.5	Big Data y Análisis de Datos	47
4.6	Internet of Thing	48
CAPÍTULO 5 CASO PRACTICO.....		50
5.1	Introducción.....	50
5.2	Incremento I Roles de Usuario y Captura de Catálogos	51
5.2.1	Análisis de Requerimientos	51
5.2.2	Diseño.....	53
5.2.3	Codificación	59
5.2.4	Implementación.....	60
5.3	Incremento II Administración de Usuarios	63

5.3.1	Análisis de Requerimientos	63
5.3.2	Diseño.....	64
5.3.3	Codificación	67
5.3.4	Implementación.....	68
5.4	Incremento III Solicitud de Requerimientos.....	72
5.4.1	Análisis de Requerimientos	72
5.4.2	Diseño.....	73
5.4.3	Codificación	76
5.4.4	Implementación.....	77
5.5	Incremento IV Responder Solicitud	80
5.5.1	Análisis de Requerimientos	80
5.5.2	Diseño.....	80
5.5.3	Codificación	83
5.5.4	Implementación.....	83
5.6	Diseño de Base de Datos.....	86
CAPÍTULO 6 CONCLUSIÓN.....		87
CAPÍTULO 7 ANEXO		89
BIBLIOGRAFÍA		90

Lista de Figuras

<i>Figura 1</i> Navegación Web (Kemp, 2017)	5
<i>Figura 2</i> Web 4.0 (Wix, 2013)	6
<i>Figura 3</i> Web Cognitiva (Piper, 2015)	7
<i>Figura 4</i> Desarrollo Back-End	9
<i>Figura 5</i> Desarrollo Front-End (Bhatti, 2016)	9
<i>Figura 6</i> Web Components (Unziplogic System, 2014).....	10
<i>Figura 7</i> Navegadores que Soportan Web Components (www.webcomponents.org, 2017) 11	
<i>Figura 8</i> Cloud Computing (Takala, 2017).....	12
<i>Figura 9</i> Modelo SAGE (Benington, 1983).....	14
<i>Figura 10</i> Modelo Para Pequeños Desarrollos (Royce, 1970)	15
<i>Figura 11</i> Modelo en Cascada Secuencial e Iterativo (Royce, 1970).....	15
<i>Figura 12</i> Retroalimentación (Royce, 1970).....	16
<i>Figura 13</i> Diseño Previo y Aspectos Técnicos (Royce, 1970).....	16
<i>Figura 14</i> Creación de Prototipos (Royce, 1970)	17
<i>Figura 15</i> Modelo en Espiral de Boehm (Boehm B. W., 1988).....	18
<i>Figura 16</i> Modelo en V-Típico (Mathur & Malik, 2010).....	20
<i>Figura 17</i> Modelo en V-Avanzado (Mathur & Malik, 2010).....	21
<i>Figura 18</i> Esquema del Modelo Incremental (Pressman, 2001)	23
<i>Figura 19</i> Integración de Incrementos (Mills, 1980)	24
<i>Figura 20</i> Estructura de Abstract Factory (Gamma, Helm, Johnson, & Vlissides, 1994).....	26
<i>Figura 21</i> Ejemplo de Implementación (Tutorials Point , 2015).....	27
<i>Figura 22</i> Estructura de Factory Method (Gamma, Helm, Johnson, & Vlissides, 1994).....	28
<i>Figura 23</i> Ejemplo de Implementación (Tutorials Point , 2015).....	28
<i>Figura 24</i> Estructura de Prototype (Gamma, Helm, Johnson, & Vlissides, 1994)	29
<i>Figura 25</i> Ejemplo de Implementación (Tutorials Point , 2015).....	29
<i>Figura 26</i> Estructura de Adapter (Gamma, Helm, Johnson, & Vlissides, 1994)	30
<i>Figura 27</i> Ejemplo de Implementación (Tutorials Point , 2015).....	31
<i>Figura 28</i> Estructura de Composite (Gamma, Helm, Johnson, & Vlissides, 1994)	32
<i>Figura 29</i> Ejemplo de Implementación (Tutorials Point , 2015).....	32
<i>Figura 30</i> Estructura de Decorator (Gamma, Helm, Johnson, & Vlissides, 1994).....	33
<i>Figura 31</i> Ejemplo de Implementación (Tutorials Point , 2015).....	33
<i>Figura 32</i> Estructura de Chain of Responsibility (Gamma, Helm, Johnson, & Vlissides, 1994)	34
<i>Figura 33</i> Ejemplo de Implementación (Tutorials Point , 2015).....	35
<i>Figura 34</i> Estructura de Mediator (Gamma, Helm, Johnson, & Vlissides, 1994)	36
<i>Figura 35</i> Ejemplo de Implementación (Tutorials Point , 2015).....	36
<i>Figura 36</i> Estructura de Template Method (Gamma, Helm, Johnson, & Vlissides, 1994)	37
<i>Figura 37</i> Ejemplo de Implementación (Tutorials Point , 2015).....	37
<i>Figura 38</i> Estructura de MVC (Sweat, 2006)	38

<i>Figura 39</i> Funcionamiento de MVC (Sommerville, 2005).....	38
<i>Figura 40</i> Ejemplo de Implementación (Sweat, 2006).....	39
<i>Figura 41</i> Diagrama de Clases de MVC (Sweat, 2006)	39
<i>Figura 42</i> Uso de Internet Por Continente (Kemp, 2017)	40
<i>Figura 43</i> Realidad Virtual (Rasulury, 2015).....	41
<i>Figura 44</i> Esquema de Realidades	43
<i>Figura 45</i> Realidad Aumentada (Krause, 2016)	44
<i>Figura 46</i> Inteligencia Artificial (Ansari, 2017)	45
<i>Figura 47</i> Casos de Uso y Áreas de Desarrollo de IA (Kaul & Wheelock, 2016)	46
<i>Figura 48</i> Futuro e Impacto de IA (Mobinius, 2017)	47
<i>Figura 49</i> Áreas de Análisis de Big Data (Rayo, 2016)	47
<i>Figura 50</i> Uso e Impacto de Big Data (Schroeck, Shockley, Smart, Romero-Morales, & Tufano, 2014).....	48
<i>Figura 51</i> Sectores que Implementan IoT (Verizon, 2016).....	49
<i>Figura 52</i> Diagrama de Casos de Uso (Incremento I Catálogos).....	54
<i>Figura 53</i> Diagrama de Actividades (Incremento I Catálogos)	54
<i>Figura 54</i> Diagrama de Clases (Incremento I Catálogos)	55
<i>Figura 55</i> Formulario de Inicio de Sesión.....	56
<i>Figura 56</i> Formulario Recuperar Contraseña.....	56
<i>Figura 57</i> Ingreso al Sistema (Administrador).....	57
<i>Figura 58</i> Vista de los Catálogos (Sección Licenciaturas)	57
<i>Figura 59</i> Vista de los Catálogos (Sección Profesores)	57
<i>Figura 60</i> Vista de los Catálogos (Sección Requerimientos)	58
<i>Figura 61</i> Subsecciones de Administración de Usuarios	58
<i>Figura 62</i> Subsecciones de Solicitudes de Requerimientos	59
<i>Figura 63</i> Subsecciones de Perfil	59
<i>Figura 64</i> Formulario de Cambio de Contraseña	59
<i>Figura 65</i> Datos de Acceso al Sistema	60
<i>Figura 66</i> Correo de Recuperación de Contraseña.....	60
<i>Figura 67</i> Formulario para Cambiar Contraseña.....	61
<i>Figura 68</i> Mensaje de Error de Recuperación de Contraseña	61
<i>Figura 69</i> Implementación Para Cargar Catálogo de Licenciaturas.....	61
<i>Figura 70</i> Implementación Para Cargar Catálogo de Profesores.....	62
<i>Figura 71</i> Implementación Para Cargar Catálogo de Requerimientos	62
<i>Figura 72</i> Diagrama de Casos de Uso (Incremento II Administración de Usuarios).....	64
<i>Figura 73</i> Diagrama de Actividades de Administración de Usuarios (Incremento II Administración de Usuarios).....	65
<i>Figura 74</i> Diagrama de Clases (Incremento II Administración de Usuarios)	66
<i>Figura 75</i> Subsecciones de Administración de Usuarios	66
<i>Figura 76</i> Formulario de Administrador	67

<i>Figura 77</i> Formulario de Profesor	67
<i>Figura 78</i> Lista de Usuarios	67
<i>Figura 79</i> Registro de Administrador.....	68
<i>Figura 80</i> Lista de Administradores.....	68
<i>Figura 81</i> Email de Activación de Usuario.....	68
<i>Figura 82</i> Mensaje de Activación de Cuenta.....	69
<i>Figura 83</i> Activación de Cuenta	69
<i>Figura 84</i> Mensaje de Cuenta ya Activada.....	69
<i>Figura 85</i> Registro de Profesor	70
<i>Figura 86</i> Mensaje de Eliminación de Usuario.....	70
<i>Figura 87</i> Lista de Profesores Actualizados	70
<i>Figura 88</i> Mensaje de Cambio de Contraseña.....	71
<i>Figura 89</i> Diagrama de Casos de Uso (Incremento III Crear Solicitud de Requerimiento) ..	73
<i>Figura 90</i> Diagrama de Actividades (Incremento III Crear Solicitud de Requerimiento).....	74
<i>Figura 91</i> Diagrama de Clases (Incremento III Crear Solicitud de Requerimientos).....	74
<i>Figura 92</i> Subsecciones Para Solicitudes de Requerimiento.....	75
<i>Figura 93</i> Interfaz de Solicitud Todas las Licenciaturas	75
<i>Figura 94</i> Interfaz Solicitud por Licenciatura	76
<i>Figura 95</i> Interfaz Solicitud Por Profesor	76
<i>Figura 96</i> Interfaz Historial de Solicitudes.....	76
<i>Figura 97</i> Crear Solicitud De Requerimiento Para Todas Las Licenciaturas	77
<i>Figura 98</i> Creación de Solicitud de Requerimiento por Licenciatura (Ejemplo: Lic. en Economía)	77
<i>Figura 99</i> Creación de Solicitud por Profesor (Selección de Profesores)	78
<i>Figura 100</i> Email de Notificación de Solicitud de Requerimiento.....	78
<i>Figura 101</i> Solicitudes de Requerimientos para Todas las Licenciaturas (Bibliografía)	79
<i>Figura 102</i> Solicitud de Requerimiento por Licenciatura (Marcado Con Amarillo)	79
<i>Figura 103</i> Creación de Solicitudes por Profesor (Cuadro Azul).....	79
<i>Figura 104</i> Diagrama de Casos de Uso (Incremento IV Responder Solicitud de Requerimiento)	81
<i>Figura 105</i> Diagrama de Actividades (Incremento IV Responder Solicitud de Requerimiento)82	
<i>Figura 106</i> Diagrama de Clases (Incremento IV Responder Solicitud de Requerimiento)...	82
<i>Figura 107</i> Inicio de Sesión de un Profesor	83
<i>Figura 108</i> Lista de Solicitudes de Requerimientos	83
<i>Figura 109</i> Formulario para Subir Archivo de Requerimiento	83
<i>Figura 110</i> Lista de Requerimientos a Responder	84
<i>Figura 111</i> Cargar Archivo para Requerimiento	84
<i>Figura 112</i> Mensaje de Archivo Guardado	84
<i>Figura 113</i> Archivo (Evidencia) Almacenada.....	85
<i>Figura 114</i> Evidencia Vista Desde el Navegador (Usuario).....	85

<i>Figura 115</i> Historial Actualizado de Solicitudes (Administrador)	85
<i>Figura 116</i> Modelo Relacional de la Base de Datos del Sistema	86
<i>Figura 117</i> Repositorio del Sistema	89

Lista de Tabla

<i>Tabla 1</i> Lenguajes de Programación Backend y Frontend.....	7
<i>Tabla 2</i> Lenguajes de Etiquetas y Diseño	8
<i>Tabla 3</i> Librerías de Web Components	11
<i>Tabla 4</i> Plan de Documentación	16
<i>Tabla 5</i> Sectores y Tareas del Modelo	19
<i>Tabla 6</i> Ciclo (round) de la Espiral	19
<i>Tabla 7</i> Acciones para un Plan de Gestión de riesgos	19
<i>Tabla 8</i> Fases del Modelo en V-Avanzado	20
<i>Tabla 9</i> Fases de Pruebas de Software	22
<i>Tabla 10</i> Fases de Gestión de Mantenimiento	22
<i>Tabla 11</i> Plan de Integración de Incrementos	24
<i>Tabla 12</i> Clases de Abstract Factory	26
<i>Tabla 13</i> Clases de Factory Method.....	27
<i>Tabla 14</i> Clases de Prototype	29
<i>Tabla 15</i> Clases de Adapter	30
<i>Tabla 16</i> Clases de Composite	31
<i>Tabla 17</i> Clases de Decorator.....	32
<i>Tabla 18</i> Clases de Chain Of Responsibility	34
<i>Tabla 19</i> Clases de Mediator.....	35
<i>Tabla 20</i> Clases de Template Method	36
<i>Tabla 21</i> Tipos de Realidad Virtual.....	41
<i>Tabla 22</i> Compañías de Desarrollo de Realidad Virtual.....	42
<i>Tabla 23</i> Aplicación de la Realidad Virtual	42
<i>Tabla 24</i> Áreas de Aplicación de AR	44
<i>Tabla 25</i> Menú del Administrador.....	51
<i>Tabla 26</i> Menú del Profesor	52
<i>Tabla 27</i> Tabla Para Licenciaturas	53
<i>Tabla 28</i> Tabla para Profesores	53
<i>Tabla 29</i> Tabla para Requerimientos.....	53
<i>Tabla 30</i> Estructura del Menú y Submenú (Administrador).....	56
<i>Tabla 31</i> Estructura de los Formularios de Usuarios.....	63

INTRODUCCIÓN

Los cambios tecnológicos a los que estamos expuestos, permiten que se desarrollen y se solucionen tareas que suelen ser repetitivas o en otros casos, difícil de seguir manteniendo debido al impacto que puede causar. El uso de las tecnologías de la información y comunicación (TIC'S) pueden mejorar el nivel de trabajo y ser más eficientes para el desarrollo de actividades.

En este trabajo de investigación, se pretende desarrollar una aplicación web con las tecnologías y herramientas de vanguardia, la cual permita la toma de decisiones. Una aplicación web es definida como como una aplicación que es accedida vía web, es decir, a través de internet.

La subdirección académica es un órgano de control dentro del Centro Universitario UAEM Texcoco, el cual realiza peticiones de requerimientos a todo el personal docente de las licenciaturas con las que cuenta el centro universitario. La característica principal de este proceso, es que todo se hace de manera física y puede traer consigo el mal manejo o pérdida de dicha solicitud. En ese sentido, la aplicación web ayudara al área de subdirección académica el evitar los posibles riesgos durante la solicitud del requerimiento.

El interés por resolver esta problemática, se centra principalmente en que este proceso puede ser solucionado con herramientas y la tecnología adecuada que se adapten a la infraestructura del centro universitario y en general, de la propia universidad.

Para el desarrollo de la aplicación web, se centrará en el uso de un modelo de desarrollo que permita el analizar, diseñar, desarrollar e implementar la aplicación web de una manera eficaz que desde el primer momento sea utilizado. El modelo de desarrollo incremental, es un modelo que permite el desarrollo de software a base de incrementos, lo cual permite que sea un componente metodológico ideal para el desarrollo de la aplicación web (o sistema).

El uso del paradigma orientado a objetos (OOP) es un paradigma que contempla el desarrollo de software de tal manera que sea fácil de entender y crear. En ese sentido, el uso del paradigma se centrará en el análisis, diseño y codificación del sistema (o aplicación web) contemplando aspectos importantes, tales como el uso del lenguaje modelado unificado (UML) para el diseño de diagramas y la programación orientada a objetos.

La estructura del trabajo se compone de la siguiente manera:

- Capítulo 1 Estado del Arte.
- Capítulo 2 Modelos de Desarrollo de Software.
- Patrones de Diseño.
- Capítulo 4 Tendencias Tecnológicas.
- Capítulo 5 Caso Practico.
- Capítulo 6 Conclusión.
- Capítulo 7 **Anexo**.

- Bibliografía.

DEFINICIÓN DEL PROBLEMA

La subdirección académica del Centro Universitario (CU) UAEM Texcoco, hace una serie de solicitudes de requerimientos a los profesores de todas las licenciaturas. Estas solicitudes, tratan de abordar la manera y estilo de trabajo de cada profesor durante el semestre.

Actualmente, cada solicitud se hace de manera personal, a través de una serie de formatos para su entrega. La respuesta a esta solicitud se responde mediante el documento físico, complicando de manera considerable el proceso de solicitud-entrega de requerimientos.

La manera de solicitud-entrega atrae riesgos. Por ejemplo,

- El extravió de los requisitos entregados.
- No llevar un control de entrega.
- No tener un seguimiento de las solicitudes.
- Respuesta tardía a la solicitud.

La presente tesis requiere comprobar si existe un método automatizado que integre la información solicitada por la subdirección académica del CU UAEM Texcoco, a través de una plataforma disponible para la web, facilitando el acceso y la correcta utilización de los requisitos solicitados de los maestros para la toma de decisiones.

METODOLOGÍA

Para la realización de este trabajo de investigación, fue necesario recurrir a diferentes técnicas de investigación, las cuales son:

- ✓ Investigación descriptiva.
- ✓ Investigación documental.
- ✓ Investigación experimental.

El uso de la investigación documental, nos dará la capacidad de realizar un compendio de referencias específicas para la obtención de la información que servirá de apoyo para la descripción de los capítulos a presentar.

El uso de la investigación descriptiva, nos permite realizar la descripción de los aspectos importantes de cada tema presente en este trabajo, es decir, analizar y describir de manera concreta y breve, cada uno de los temas que permita la toma de decisiones en el presente trabajo.

La investigación experimental será usada cuando se desarrolle el tema del caso práctico, es decir, desarrollando el sistema (o web app) con toda la información recabada.

JUSTIFICACIÓN

La tecnología ha permitido que muchos procesos que generen conflictos dentro y fuera de una organización sean automatizados. La automatización de procesos y en general el software, ha logrado que las personas sean más productivas y sus problemas sean solucionados de manera sencilla y eficaz.

El CU si bien es cierto que cuenta con la licenciatura de ingeniería en computación (ICO), en general, la deficiencia dentro del área que brinda soporte en aspectos de TI, suelen ser los mismos profesores. Esto da pauta a que algunos de los procesos de control y tramites que maneja el CU y las áreas encargadas, sean presenciales y en algunos casos, son archivadas de manera física sin llevar un seguimiento del mismo.

Ante estos retos, los alumnos de ICO atraigan estos retos y que sean solucionados mediante la inclusión de tecnologías que sean capaces de cumplir y satisfacer, las necesidades que suelen presentarse dentro del CU.

Por tal motivo, el haber sido parte de ICO, se realiza el presente trabajo. El reto durante este trabajo es desarrollar un sistema (o web app) que permita llevar un seguimiento y concentración de la información recabada para solicitud de requerimientos donde se facilite el proceso de solicitud y respuesta entre personal de subdirección académica y el profesor.

Es necesario mencionar que, de acuerdo a este problema detectado, se requiere que el sistema sea desarrollado con herramientas tecnológicas que se adapten a la infraestructura tecnológica con la que cuenta el CU. Para lograrlo, el uso de metodologías es un factor importante. En ese sentido, el uso del modelo de desarrollo incremental es una metodología para desarrollar software que, desde el inicio de su desarrollo, los usuarios involucrados tengan la capacidad de interactuar con el sistema hasta su terminación.

OBJETIVO

Desarrollar un sistema de información web que permita a la subdirección académica del CU UAEM Texcoco, solicitar y administrar requerimientos hacia los profesores de todas las licenciaturas académicas que constituyen el CU, mediante el uso de herramientas tecnológicas que se adapten a la infraestructura tecnológica del CU y de la universidad en general.

Objetivos Específicos

- ✓ Cumplir con los requerimientos obtenidos de la subdirección académica para el desarrollo del sistema.
- ✓ Determinar el número de incrementos a utilizar para el desarrollo del sistema, con base en el uso del modelo incremental.
- ✓ Usar las herramientas para el desarrollo del sistema adecuadas y que se adapten a la infraestructura del CU
- ✓ Desarrollar un sistema que sea fácil de usar y que permita la toma de decisiones.
- ✓ Seleccionar un paradigma de desarrollo adecuado para el desarrollo del software
- ✓ Implementar el paradigma orientado a objetos (OOP siglas en inglés) para el análisis, diseño, desarrollo e implementación del sistema.

CAPÍTULO 1 ESTADO DEL ARTE

La tecnología ha evolucionado de tal manera que los usuarios estén siempre conectados, todo gracias al internet. Desde la primera página de internet en la década de los 90 hasta las aplicaciones para dispositivos móviles, que simplifiquen las tareas de los usuarios de su día a día. Empresas tecnológicas como Microsoft, Twitter, Facebook, Apple y Startups, se han encargado de crear herramientas idóneas con base en las necesidades de los usuarios (Alvarez, Innovación en el Desarrollo, 2016).

El termino startup se relaciona con empresas emergentes que tienen una vinculación con la tecnología con ideas innovadoras y sobresalientes. La página Shopify define a la startup como (Pierce, 2016):

“Una Startup es una organización humana con gran capacidad de cambio, que desarrolla productos o servicios, de gran innovación, altamente deseados o requeridos por el mercado, donde su diseño y comercialización están orientados completamente al cliente”.

La evolución de la tecnología, adopto un estilo de vida donde los usuarios, están conectados y navegando a través de internet. Como consecuencia, incremento la publicidad en línea, streaming, consulta de redes sociales, geolocalización, transacciones bancarias, por mencionar alguno de los tantos procesos que se hacen en línea. Un estudio realizado por Simon Kemp llamado **“Digital in 2017”**, muestra los dispositivos más usados para navegar en redes sociales y el internet en general (**Figura 1**).



Figura 1 Navegación Web (Kemp, 2017)

1.1 Evolución de la Web cómo Plataforma

La evolución de internet, permitió a los consumidores y compañías, expandirse de manera rápida. Esto trajo como consecuencia, la comunicación, interacción y colaboración entre todos los países del mundo que tienen acceso a internet. Para saber más a detalle como evoluciono, se describirán las principales etapas de la evolución de la web como una plataforma.

1.1.1 De la web 1.0 a la 3.0

La evolución de internet y las páginas web, fueron de gran importancia. A través de ellas, los usuarios podrían interactuar con la computadora y consultar información de cualquier tipo. La primera generación (web 1.0), se creó en 1989, por el fundador de la W3C (World Wide Web Consortium) Tim Berners.Lee, con paginas estáticas y difícil interacción (Patel, 2013).

Presentada por Tim O'Reilly, la segunda generación (web 2.0) se centra en el desarrollo de aplicaciones web para los usuarios, pensado en que la interacción sea más fácil. A partir de ese momento, se siguió trabajando en mejorar la interactividad usuario-página web (Aghaei, Nematbakhsh, & Farsani, 2012).

Fue hasta el 2006 cuando John Markof presentó la tercera generación (web 3.0), teniendo como objetivo, analizar y generar grandes cantidades de datos, mejorando la calidad de la información mostrada hacia los usuarios. La web 3.0 es conocida también como “web ejecutable”, “web semántica” o “web de datos” (Choudhury, 2014).

1.1.2 Web 4.0

La cuarta generación crea nuevos conceptos y la forma de interactuar a través de internet, tomando el concepto de hombre-máquina para facilitar las tareas y procesos de los usuarios. La web 4.0 es mencionada como una “web simbiótica”, web ubicua” o “agente electrónico ultra-inteligente” (mencionado en el tema **¡Error! No se encuentra el origen de la referencia.**). Ayudará al desarrollo en otras áreas, tales como telecomunicaciones, nanotecnología y control de interfaces (Choudhury, 2014).



Figura 2 Web 4.0 (Wix, 2013)

Algunos autores mencionan la transición de la web 4.0 a la web 5.0, esta última tendría como características muy importantes, por ejemplo: que los dispositivos identificaran el lenguaje humano y sus emociones, llamándola como “Web Sensorial y Emotiva” o “Web Cognitiva” (Patel, 2013).



Figura 3 Web Cognitiva (Piper, 2015)

1.2 Lenguajes de Programación Web

Los lenguajes de programación para crear aplicaciones web permiten a los usuarios finales, interactuar con contenidos de las páginas. Existen una gran variedad de lenguajes de programación para desarrollar aplicaciones web. A continuación, se describirán algunos de los lenguajes más utilizados para desarrollar aplicaciones web. Estos lenguajes, son usados para desarrollar la parte del backend y frontend (**Tabla 1**).

Tabla 1 Lenguajes de Programación Backend y Frontend

Nombre	Acrónimo	Definición
ASP.Net	Active Server Page	Desarrollado por Microsoft, forma parte de la plataforma .NET, combinando el lenguaje HTML. Utiliza otros lenguajes como C#, Visual Basic.Net, JScript y J# (Giardina, 2011).
Perl		Lenguaje de programación multiplataforma que permite la manipulación de texto y desarrollo rápido. Cuenta con frameworks, módulos, encriptación de datos, manipulación de base de datos, que permite la creación de aplicaciones web muy completas (Point, 2015).
PHP	Hypertext Pre-processor	Lenguaje de programación que se ejecuta en la parte del servidores cuando el usuario hace una petición y es mostrada en el navegador en HTML. Soporta programación orientada a objetos y bases de datos (Nixon, 2015).
Ruby on Rails		Permite desarrollar aplicaciones web sencillas, robustas y de manera rápida (Rails, 2016). Ruby. Lenguaje de programación Script con sintaxis simple para fácil entendimiento, soportando orientado a objetos. Rails. Es el entorno de desarrollo de aplicaciones web de Ruby

Java Script	JS		Lenguaje de programación para la creación de páginas web más dinámicas interactuando con HTML y CSS (Duckett, 2014).
JSON	Java Objetc Notation	Script	Basado en JS, es una alternativa a XML para el intercambio y almacenamiento de datos ligeros (JSON.ORG, 2017).
JAVA			Pertenece al compendio empresarial Java EE7. Algunos de sus componentes para el desarrollo web son (Jendrock, Cervera-Navarro, Evans, Haase, & Markito, 2014): <ul style="list-style-type: none"> • Java Servlet • Java Server Faces JSP (Java Server Pages)
Python			Necesita de un módulo CGI (Common Gateway Interface) y Frameworks para desarrollar aplicaciones web. Los frameworks más usados son (Asadi, 2015): <ul style="list-style-type: none"> • Django • Turbo Gear • Web2py

1.2.1 Lenguajes de Etiquetas

Tabla 2 Lenguajes de Etiquetas y Diseño

Nombre	Acrónimo	Definición
HTML	HyperText Markup Language	Estándar de W3C, para documentos y páginas web (W3Schools, 2017).
XML	eXtensible Markup Language	Lenguaje de etiquetas diseñado para almacenar, transportar y generar disponibilidad de grandes cantidades de datos, similar a HTML (W3Schools, 2017).
CSS	Cascading Style Sheets	Conjunto de reglas para diseñar una página web (W3Schools, 2017)

1.3 SPA (Single Page Application)

Un SPA es una técnica de desarrollo web donde muestra el contenido de la página en una sola, dividida en diferentes secciones o vistas (Alvarez, Innovación en el Desarrollo, 2016).

1.4 Desarrollo Back-End

El desarrollo Back-End tiene como objetivo, diseñar, crear e implementar la funcionalidad de un software que no es visible para el usuario. Se ejecuta del lado del servidor y es implementado a través del Front-End, es decir del lado del usuario. El desarrollo Back-End ayudará a establecer una conexión entre las diferentes plataformas en que sea ejecutado (**Figura 4**) (Alvarez, Innovación en el Desarrollo, 2016).

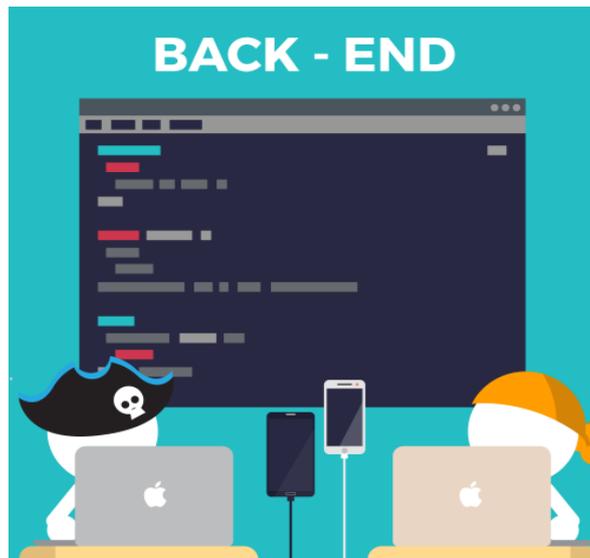


Figura 4 Desarrollo Back-End

1.5 Desarrollo Front-End

El desarrollo Front-End, es el encargado de implementar el Back-End a través del usuario, con un diseño agradable y eficaz. Front-End trabaja con los diferentes lenguajes de programación donde se crea el Back-End, dependiendo de la plataforma, usando elementos de HTML, CSS y JS principalmente (**Figura 5**) (Alvarez, Innovación en el Desarrollo, 2016).



Figura 5 Desarrollo Front-End (Bhatti, 2016)

1.6 Responsive Web Design

El termino responsive web design (diseño web responsivo), se enfoca en el diseño Front-End de las páginas web en cualquier dispositivo, sin cambiar su diseño y resolución de pantalla para conservar el mismo contenido, siendo esta, una solución única (**Figura 6**).

Los colaboradores de desarrollosweb.com definen responsive web design como (Alvarez & Martínez, 2016):

“ES UNA TÉCNICA QUE PERMITE HACER WEBS ADAPTABLES A LAS CONDICIONES DEL ORDENADOR O DISPOSITIVO QUE LAS ESTÁ ACCEDIENDO. TAMBIÉN LAS CONOCEMOS COMO TÉCNICAS ADAPTATIVAS, FOCALIZADAS PRINCIPALMENTE A LAS DIMENSIONES DE LA PANTALLA.”



Figura 6 Web Components (Unziplogic System, 2014)

1.7 Web Components

El termino Web components (WC) fue utilizados por desarrolladores de los navegadores Chrome y Mozilla para la colección e implementación de elementos HTML. Estos elementos permiten encapsular código HTML, JS y CSS, para ser utilizado en cualquier proyecto, creando etiquetas más simples para los proyectos de desarrollo y diseño web (acensTechnologies, 2016).

Para el desarrollo de WC, es necesario utilizar 4 elementos complementarios independientes entre sí: 1) Custom Elements, 2) Templates, 3) HTML Imports y 4) Shadow DOM. También, es necesario contar con un navegador que soporte web components (**Figura 7**).

1.7.1 Custom Elements

Permite al desarrollador crear sus propias etiquetas HTML para la reutilización de estas etiquetas creadas en cualquier proyecto (acensTechnologies, 2016).

1.7.2 Templates

Contiene código HTML y CSS que inicialmente no se muestran al inicio de la página y que a través de JS se acceda a su código, haciendo que cada tarea que se termine, sea preparada para ser usada y reutilizada sin volver a hacerla otra vez (acensTechnologies, 2016).

1.7.3 Shadow DOM

Shadow DOM (Modelo de Objetos del Documento) permite la encapsulación de elementos que se aíslan en el DOM para proteger estilos scripts, etc. DOM proporciona una representación estructural del documento para la modificación de su contenido y

visualización, comunicando las páginas web con scripts u otros lenguajes de programación (Kitamura, 2014).

1.7.4 HTML Imports

Permite la interacción con otros archivos que contengan código HTML CSS y JS (Kitamura, 2015).

1.7.5 Librerías Web Components

Las bibliotecas para WC (**Nota:** Información tomada de .

), nos ayuda a utilizar al máximo las herramientas y elementos para el desarrollo de sitios y aplicaciones web, predominando su sencillez y calidad, además de contar con un catálogo para su uso.

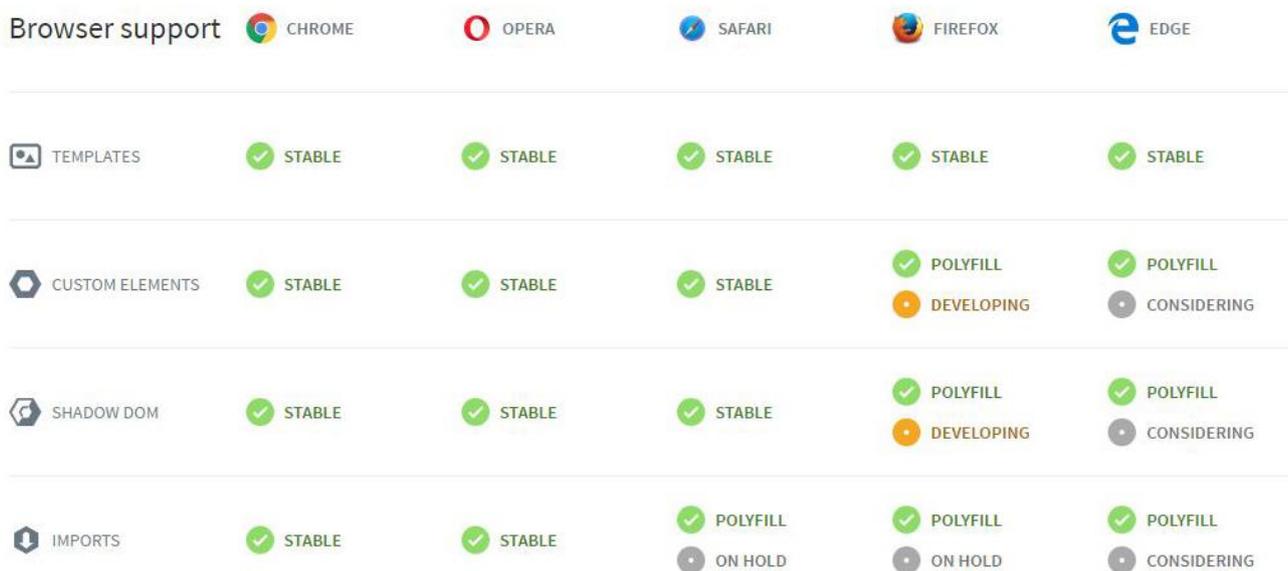


Figura 7 Navegadores que Soportan Web Components (www.webcomponents.org, 2017)

Tabla 3 Librerías de Web Components

Librería	Definición
Polymer	Es una biblioteca web, construida encima de la WC, diseñado para aprovechar la plataforma web junto con los navegadores web modernos.
X-Tag.	Proporcionar una interfaz compacta, rica en características para el desarrollo de componentes rápida
Skate.js	Permite escribir componentes web funcionales y performance con una huella muy pequeña
Slim.js	Se concentra para proporcionar al desarrollador la capacidad de escribir componentes web sólidas y nativas sin la molestia de dependencias y una sobrecarga de un marco..
Polyfills	Es un componente script que permite que la WC sean soportados en navegadores que carecen de soporte nativamente.

Nota: Información tomada de (www.webcomponents.org, 2017).

1.8 Cloud-Computing

La National Institute of Standards of Technology, define cloud-computing como (Mell & Grance, 2011):

“Cómputo en la nube es un modelo que permite el acceso adecuado y bajo demanda a un conjunto de recursos de cómputo configurable (por ejemplo: redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente provistos y puestos a disposición del cliente con un mínimo esfuerzo de gestión y de interacción con el proveedor de servicio”.

Con el reciente crecimiento de la nube, muchos administradores de TI, serán capaces de hacer la transición más rápida. Hitachi Ltd. Comenta (Rey, 2017: El Año de la Transformación Digital, 2017):

“LOS PORTALES DE VIRTUALIZACIÓN, CONVERGENCIA, ALMACENAMIENTO DE OBJETOS Y GESTIÓN DE NUBES, FACILITARÁN EL MOVIMIENTO HACIA LA CLOUD-COMPUTING”.

De tal manera que, el desarrollo de software se centrará hacia la nube y permitirá el análisis de otras tendencias como Big Data, IA e IoT .



Figura 8 Cloud Computing (Takala, 2017)

CAPÍTULO 2 MODELOS DE DESARROLLO DE SOFTWARE

2.1 Introducción

Durante la primera y segunda generación de computadoras, se usaban para programar grandes cálculos matemáticos y algunos procesos para automatizar. Sin embargo, al ir evolucionando la computadora, cada vez se desarrollaban sistemas más complejos que requerían un seguir una serie de pasos para crear el software tal como lo querían, pero no existía una metodología para hacerlo. Aunque hubo algunos que desarrollaron sus propios pasos para hacer el software, muchos fracasaban. En una conferencia por la OTAN sobre el software, se usó el término “crisis del software”, debido a la problemática que surgía en el desarrollo del software. También, en esa época se empezó a manejar el concepto de “ingeniería de software”. La ingeniería de software es definida como aspectos de la producción de un software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste, después de que se utiliza (Sommerville, 2005).

En ese sentido, el capítulo 2 muestra los antecedentes y descripción de los modelos de desarrollo que ayudaron a sacar proyectos de software adelante. Cabe mencionar que cada modelo de desarrollo presente, tiene diferentes maneras y enfoques para su uso. Cada modelo se creó en diferente año, pero también, comparten similitudes entre ellos. Los siguientes modelos de desarrollo descritos a continuación son:

- Modelo de desarrollo en cascada.
- Modelo de desarrollo en espiral.
- Modelo de desarrollo en V.
- Modelo de desarrollo incremental.

2.2 Modelo de Desarrollo en Cascada

2.2.1 Antecedentes

El modelo de desarrollo en cascada, es uno de los más viejos para el desarrollo de software. Sus orígenes datan de junio de 1959, cuando Herbert Benington, describe una técnica para el proyecto SAGE (Semi-Automatic Ground Environment), tomando como fases para el desarrollo las siguientes: **1) Plan operacional, 2) Especificación de operaciones, 3) Especificación de código, 4) Codificación, 5) Parámetro de pruebas, 6) Pruebas de ensamblaje y 7) Evaluación del sistema** (Benington, 1983) (**Figura 9**).

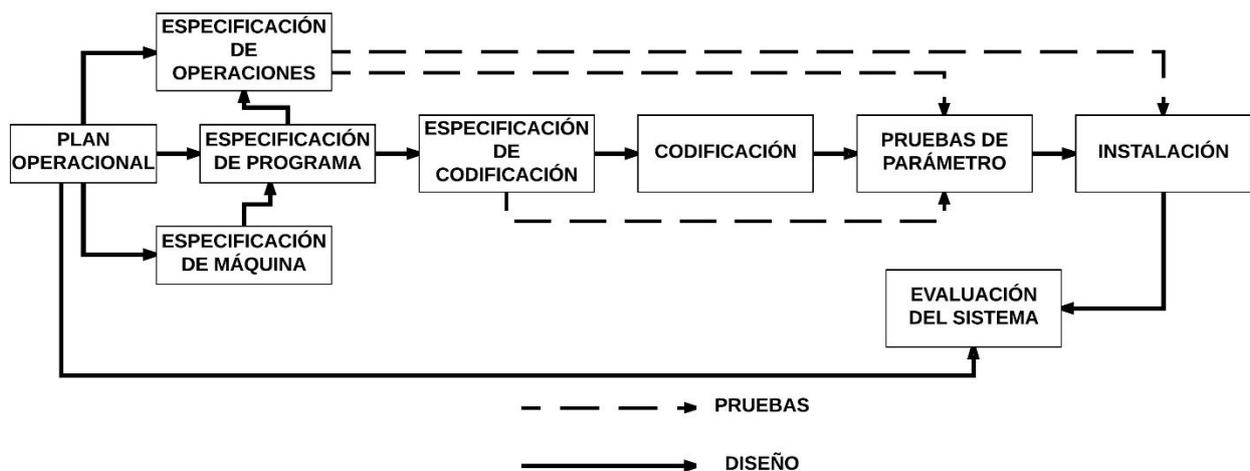


Figura 9 Modelo SAGE (Benington, 1983)

Fue hasta 1970 cuando Winston Royce retoma el modelo de Herbert en su artículo “**Administración de Grandes Desarrollos de Sistemas de Software**”, creando lo que hoy se conoce como modelo de desarrollo en cascada (Royce, 1970).

2.2.2 Modelo de Royce

El modelo de Royce consistía en crear un modelo de desarrollo secuencial combinado con la parte iterativa, para influir en todas las fases del desarrollo. Para Barry Boehm (creador del modelo en espiral), le da 2 virtudes al modelo de Royce: retroalimentación y creación de prototipos (Boehm B. W., 1988). Royce detectó que cada desarrollo de software era completamente diferente, por lo que presenta 2 modelos para pequeños y grandes desarrollos de software (Royce, 1970).

2.2.2.1 Pequeños Desarrollos De Software

El modelo de desarrollo para software pequeño (**Figura 10**), únicamente constaba de 2 fases en el modelo: análisis y codificación. El uso de este sistema desarrollado, sería de manera interna en las compañías.

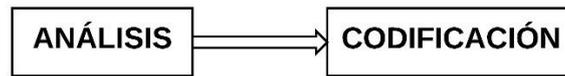


Figura 10 Modelo Para Pequeños Desarrollos (Royce, 1970)

2.2.2.2 Grandes Desarrollos De Software

Para crear software de grande y de larga duración, propone de manera general un modelo de desarrollo de 7 fases (**Figura 11**; **Error! No se encuentra el origen de la referencia.**). El uso del modelo puede ser usado de manera secuencial o iterativa, dependiendo del software que se desarrolle, enfocándose a la idea general de Royce que se propuso al implementar el modelo (Royce, 1970):

“A MEDIDA QUE CADA FASE PROGRESA Y EL DISEÑO ES MÁS DETALLADO, HAY UNA ITERACIÓN CON LAS FASES ANTERIORES Y POSTERIORES”.

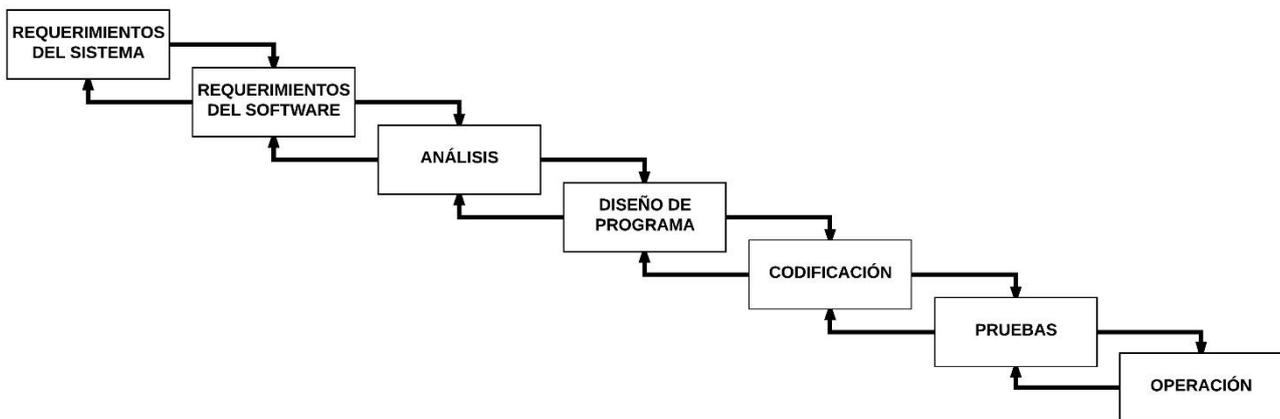


Figura 11 Modelo en Cascada Secuencial e Iterativo (Royce, 1970)

Cuando se implementa un modelo de desarrollo, existen ventajas y desventajas del mismo. En ese sentido, Boehm critica este modelo con 3 puntos que considera importantes cuando es usado de manera iterativa el modelo en cascada (Boehm B. W., 1988):

- 1 Codificación costosa.
- 2 Corrección mal estructurada.
- 3 Rechazo, aunque cumple con todos los requerimientos.

La retroalimentación del modelo en cascada de manera iterativa es muy importante para la detección y mitigación de riesgos a través de estas fases que Royce detecto son:: Requerimientos de Software, Diseño de Programa y Pruebas (**Figura 12**).

Los grandes desarrollos de software implican un mayor grado de riesgo y podría llevar al fracaso del proyecto. Para Royce la mitigación de riesgos y en general para el software es importante. En el caso del modelo en cascada, Royce añade 5 fases para lograr el éxito: diseño previo, documentación, prototipos, plan de control y monitoreo de pruebas y la participación del cliente (Royce, 1970).

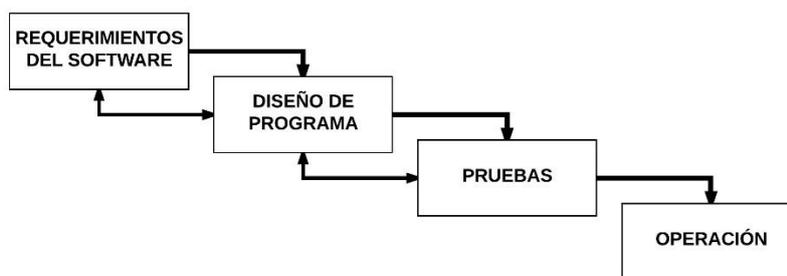


Figura 12 Retroalimentación (Royce, 1970)

2.2.2.2.1 Diseño Previo

Un diseño previo ayuda a prevenir y detectar los posibles riesgos y la identificación previa de cómo sería el software en general, con base en los requerimientos centrándose en el inicio del diseño del programa, la manera en cómo se procesarán los datos y generar la documentación necesaria del diseño (Royce, 1970) (**Figura 13**).

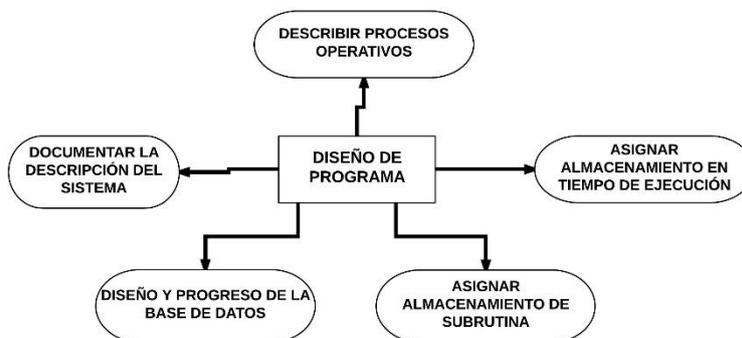


Figura 13 Diseño Previo y Aspectos Técnicos (Royce, 1970)

2.2.2.2.2 Documentación

La documentación ayudará a mejorar, detectar y mitigar riesgos del sistema en fases específicas (**Tabla 4**). Documentar todo lo relacionado al sistema, permite tener un respaldo de todo lo que se necesita para ser desarrollado.

Tabla 4 Plan de Documentación

#	Nombre del Documento	Fases
1	Requerimiento de Software	Requerimiento de Software
2	Diseño Previo (detalles)	Diseño Preliminar
3	Diseño de Interfaz	Diseño de Programa
4	Diseño Final (detalles y construcción)	<ul style="list-style-type: none"> • Diseño de Programa • Codificación • Pruebas
5	<ul style="list-style-type: none"> • Pruebas de Plan • Pruebas de Resultados 	<ul style="list-style-type: none"> • Diseño de Programa • Pruebas
6	Instrucciones de Operación	Operación

Nota: Tabla tomada de (Royce, 1970).

2.2.2.2.3 Prototipos

El uso de prototipos en el desarrollo de software ayuda en gran medida a identificar, mejorar y crear una versión pequeña del sistema, a través del uso de algunas fases del modelo, siendo implementadas de manera más concreta y probablemente ser usada en la versión final del software desarrollado, a partir de la fase de desarrollo preliminar del programa (Royce, 1970) (**Figura 14**).

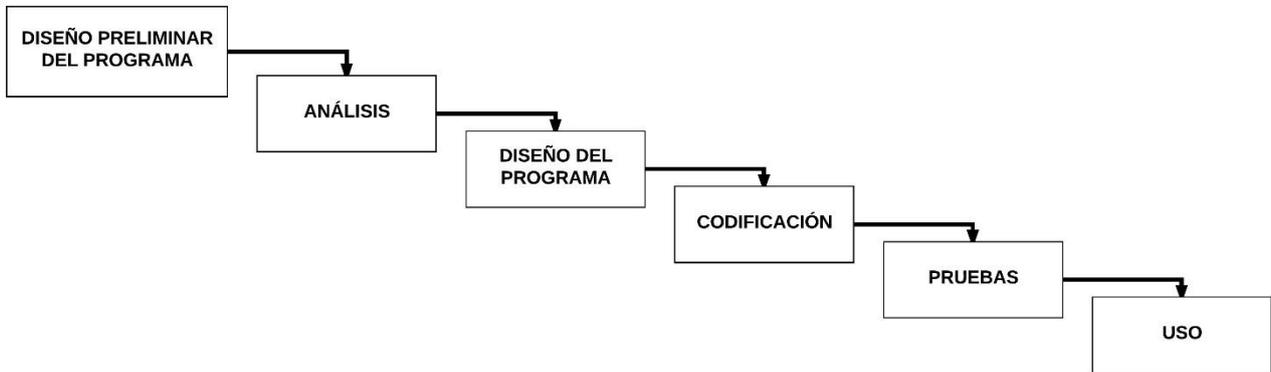


Figura 14 Creación de Prototipos (Royce, 1970)

2.2.2.2.4 Participación del Cliente

El incluir al cliente, ayuda a la aclaración de dudas en los requerimientos y llevar una excelente relación empresa-cliente.

2.3 Modelo de Desarrollo en Espiral

2.3.1 Antecedentes

A través de la compañía TRW Inc., Barry Boehm presenta el modelo de desarrollo en cascada en 1988, durante el desarrollo de SPS (Software Productivity Systems), tomando como base el modelo en cascada de Royce (Boehm B. W., 1988).

El modelo en espiral es definido como (Pressman & Maxim, 2015):

“EL MODELO EN ESPIRAL, ES UN MODELO GENERADOR DE PROCESOS CONDUCTOS POR LOS RIESGOS, UTILIZADO PARA GUIAR A LOS PARTICIPANTES EN INGENIERÍA CONCURRENTE DE SISTEMAS INTENSIVOS DE SOFTWARE. SE DISTINGUE POR SU ENFOQUE CÍCLICO DE INCREMENTOS CON UN MAYOR GRADO DE DEFINICIÓN Y EJECUCIÓN PARA LA DISMINUCIÓN DEL RIESGO. ADEMÁS, CUENTA CON UN CONJUNTO DE HITOS PARA ASEGURAR EL COMPROMISO DE LOS PARTICIPANTES PARA SOLUCIONES VIABLES Y MUTUAMENTE SATISFACTORIAS”.

2.3.2 Características del Modelo en Espiral

El modelo en espiral se basa en el desarrollo de software con un alto grado de riesgo que pueda afectar el proyecto y conduzca al fracaso. Un riesgo, es una situación posible que hace que el proyecto no cumpla con los objetivos establecidos desde un principio. Por eso, es necesario priorizar y mitigar la mayor cantidad de riesgo que afecte el desarrollo del proyecto (Boehm & Hansen, 2000).

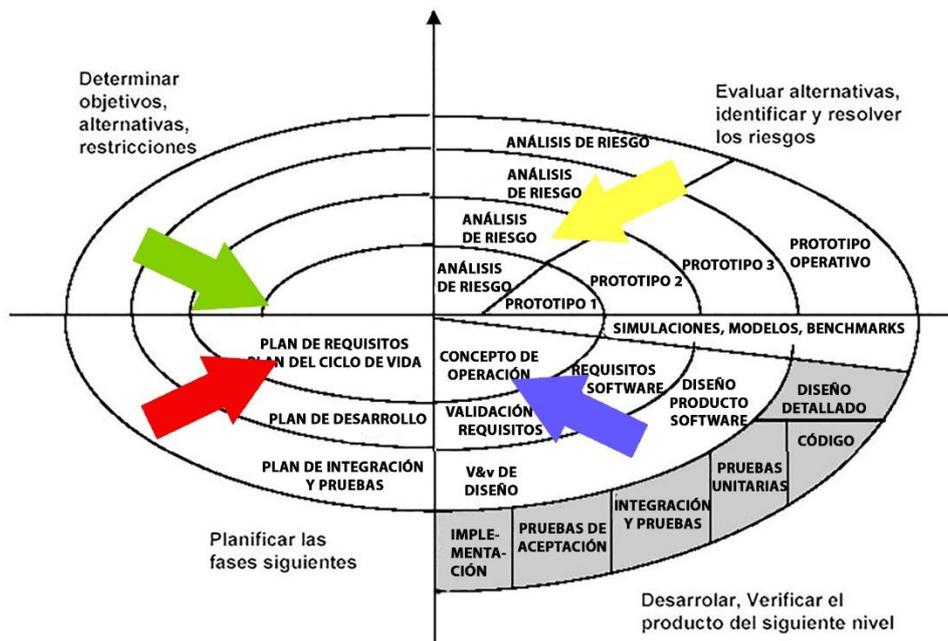


Figura 15 Modelo en Espiral de Boehm (Boehm B. W., 1988)

El modelo en espiral (**Figura 15**) se divide en 4 sectores con tareas específicas (**Tabla 5**) para que los desarrolladores sean guiados y puedan concluir un ciclo de la espiral, conocidos también como rounds (**Tabla 6**).

Tabla 5 Sectores y Tareas del Modelo

Sectores	Tareas
1. Determinar objetivos, alternativas y restricciones.	✓ Concepto de Operaciones
2. Evaluar alternativas, identificar y resolver riesgos.	✓ Requerimientos de Software
3. Desarrollar y verificar el siguiente nivel del producto.	✓ Diseño del Producto Software
4. Planear la siguiente fase	✓ Detalles del Diseño
	✓ Codificación
	✓ Prueba de Unidad
	✓ Integración y pruebas
	✓ Aceptación de pruebas
	✓ Implementación

Nota: Información Tomada de (Bruegge, 2006).

Tabla 6 Ciclo (round) de la Espiral

Ciclo	Rutina	Sector	Descripción
1	Concepto de Operaciones	Determinar objetivos, alternativas y restricciones	En este punto, se inicia el proyecto y la forma de trabajo (flecha verde).
		Evaluar alternativas	Se evalúan alternativas y restricciones del punto anterior para generar estrategias para mitigar riesgos (flecha amarilla).
		Identificar y resolver riesgos	Se determinan actividades para la evolución del proyecto, niveles de prioridad y resolución de riesgo (flecha azul).
		Desarrollar y verificar	Determinar requerimientos y el plan del ciclo de vida para el proyecto, una vez concluido inicia el round 2 (flecha roja).
		Planificar la siguiente fase	

Nota: Información Tomada de (Bruegge, 2006).

2.3.3 Plan de Gestión de Riesgos

El plan de gestión consiste en una serie de criterios de detección y mitigación de riesgos que afectan el desarrollo del proyecto, con la creación de estrategias que se deben adaptar al ciclo de vida del proyecto, que puede ser adaptable. De acuerdo con Bohem, es necesario tomar en cuenta ciertas recomendaciones las cuales sugiere (Boehm B. W., 1988) (**Tabla 7**):

Tabla 7 Acciones para un Plan de Gestión de riesgos

#	Recomendación
1	Identificar principales riesgos durante el desarrollo.
2	Presentar un plan para la solución de cada riesgo.
3	Listar y actualizar la prioridad de riesgo y revisar sus resultados.
4	Resaltar el estatus de los elementos de riesgo revisados del proyecto y compararlos con resultados anteriores.
5	Iniciar acciones correctivas apropiadas

Nota: Tomado de (Boehm B. W., 1988).

2.4 Modelo de Desarrollo en V

2.4.1 Antecedentes

Presentado por Jensen y Tonies en 1979, el modelo en V es similar al modelo en cascada, destacando el alto grado de detalle en cada fase durante el desarrollo. Para 1980, Paul Rook propone una versión del modelo en V. Hay dos versiones del modelo, v-típico y v-avanzado (Tatikonda & Lorence, 2002).

2.4.2 Modelo en V-Típico

El modelo en V-típico (**Figura 16**), funciona de arriba-abajo, de izquierda-derecha, donde presenta las fases de desarrollo que dependen entre sí, para seguir avanzando durante el desarrollo del software.

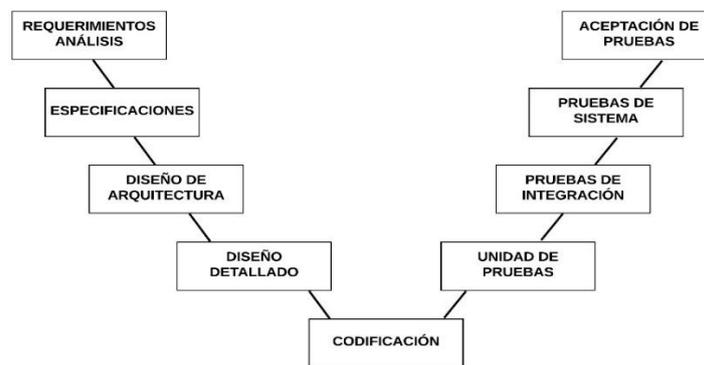


Figura 16 Modelo en V-Típico (Mathur & Malik, 2010)

2.4.3 Modelo en V-Avanzado

El modelo en V-avanzado cambia de manera significativa del modelo típico, ya que en este todas las fases de desarrollo están relacionadas entre sí, para la realización de pruebas de cada a través de mecanismos tales como: comunicación, modelado, construcción y mantenimiento de actividades (**Tabla 8**) (**Figura 17**) (Mathur & Malik, 2010).

Tabla 8 Fases del Modelo en V-Avanzado

Fase	Descripción
Análisis y Especificación de Requerimientos	Se solicitan los requerimientos al cliente para ser analizados y entendidos para el desarrollo del software
Especificación de Diseño.	Se hace el diseño del sistema cuando son entendidos los requerimientos, abordando aspectos de configuración y comunicación en el desarrollo
Especificación de Programa	Es la continuación de la fase de diseño. Se observan aspectos de funcionalidad, selección de la arquitectura, transferencia de datos y comunicación entre módulos hacia otros sistemas
Codificación	Selección de herramientas y tecnologías para desarrollar el sistema usando la información recabada en fases anteriores. Es necesario

la consideración de pruebas en todas las fases a través de la gestión de las pruebas con base en la arquitectura del software.

Nota: Información Tomada de (Mathur & Malik, 2010).

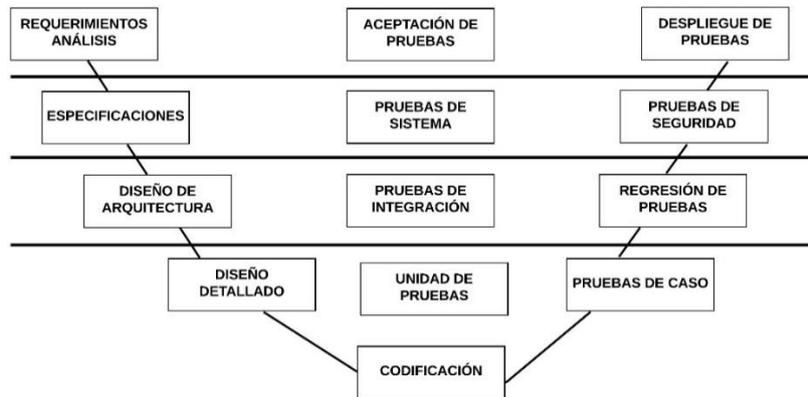


Figura 17 Modelo en V-Avanzado (Mathur & Malik, 2010)

En el caso del modelo en V-avanzado, su base principal es el diseño arquitectónico del sistema. De acuerdo con Jacobson, una arquitectura se define como (Yadav, 2012):

“LA ARQUITECTURA DE SOFTWARE ES EL CONJUNTO DE MODELOS PARA SER CONSTRUIDOS, DONDE CADA MODELO TIENE UN CONJUNTO DE ELEMENTOS Y NOTACIONES DE MODELADO, CON VISTAS CONCEPTUALES DEL PROCESO ADAPTADO PARA EL DESARROLLO DEL SOFTWARE”.

Además de la definición de Jacobson, otra fuente citada define la arquitectura como (Mathur & Malik, 2010):

“LA ARQUITECTURA DE SOFTWARE, SON AQUELLAS PROPIEDADES EXTERNAMENTE VISIBLES DE ESOS COMPONENTES Y LA RELACIÓN ENTRE ELLOS, ES DECIR, LA ESTRUCTURA(S) QUE CONFORMA EL PROGRAMA”.

2.4.4 Pruebas de Gestión

Las pruebas de gestión que se muestran en el modelo, se toman de la ingeniería de software de IEEE, las cuales la divide en dos secciones: de software y mantenimiento.

Pressman define las pruebas de software (Yadav, 2012):

“Proceso de ejecución de un programa o sistema para localizar errores”.

Así mismo, tomada de otra fuente, las pruebas de software se define (Mathur & Malik, 2010):

“Actividad para evaluar los atributos o capacidades de un programa o sistema para que cumpla con los resultados requeridos”.

2.4.4.1 Pruebas De Gestión De Software

Esta categoría de pruebas se basa en 4 fases (*Tabla 9*), las cuales se muestra en el modelo V-avanzado.

Tabla 9 Fases de Pruebas de Software

#	Nombre de Fase	Descripción
1	Unidad de pruebas	Comprueba cada función del programa de manera individual para que cumpla con su función de manera adecuada.
2	Pruebas de integración	Realiza el montaje y la integración de los componentes a partir de un paquete de software completo.
3	Pruebas de sistema	Se prueba el sistema para saber si se cumple con los requisitos establecidos, incluyendo requisitos funcionales y no funcionales.
4	Aceptación de pruebas	El sistema cumple con los requerimientos solicitados. El cliente siempre debe hacer la prueba de aceptación, ya que es la única persona cualificada y sabe lo que solicito.

Nota: Tomada de (Yadav, 2012).

2.4.4.2 Plan De Gestión De Mantenimiento

El mantenimiento de software permite que se corrija algunos defectos y que se pueda mejorar también, después de ser entregado el producto. El plan de mantenimiento consta de 5 fases que permite a los desarrolladores, mejorar el software creado (*Tabla 10*).

Tabla 10 Fases de Gestión de Mantenimiento

#	Nombre Fase	Descripción
1	Prueba de Casos	Conjunto de condiciones o variables en las que una persona que ponga a prueba el software, determinara si cumple con las especificaciones a nivel de unidad.
2	Pruebas de Regresión	Pruebas anteriores podrían repetirse en otros niveles con el fin de entregar el valor final del producto.
3	Prueba de Seguridad	Evalúa el funcionamiento adecuado de la seguridad para garantizar integridad y confidencialidad de datos.
4	Prueba de Despliegue (Simulación)	El sistema se comprueba, pero la diferencia importante es el cambio en el enfoque: (1) Pruebas de sistemas, comprueba que el sistema que se ha especificado ha sido entregado. (2) Aceptación de prueba, comprueba que el sistema cumple con lo solicitado.
5	Prueba de Liberación	Hacer pruebas internas del sistema completo para observar su funcionamiento y con la ayuda del modelo de negocio de la empresa comprobar la adaptabilidad y operatividad del sistema.

Nota: Tomado de (Yadav, 2012).

2.5 Modelo de Desarrollo Incremental

2.5.1 Antecedentes

En 1970, Harlan Mills durante su trabajo en IBM, diseño un modelo iterativo. Pero fue hasta 1980, cuando se conoce como modelo incremental tomando algunas fases del modelo en cascada (Larman & Basili, 2003).

“Es posible generar secuencia de sistemas intermedios de código y sub-especificaciones funcionales que, a cada paso, cada (intermedio) sistema puede ser verificado para ser correcto”.

2.5.2 Descripción del Modelo

El modelo incremental combina la forma secuencial e iterativo a través de prototipos funcionales, es decir, cada evolución del proyecto se considera un incremento. El primer incremento que sale (primera versión del software), contiene elementos básicos (Core) del proyecto hasta seguir con los siguientes incrementos para mejorar su funcionalidad, priorizando los requerimientos más importantes. Es importante definir la cantidad de incrementos que se requieren para crear el software (Figura 18) (Pressman, 2001).

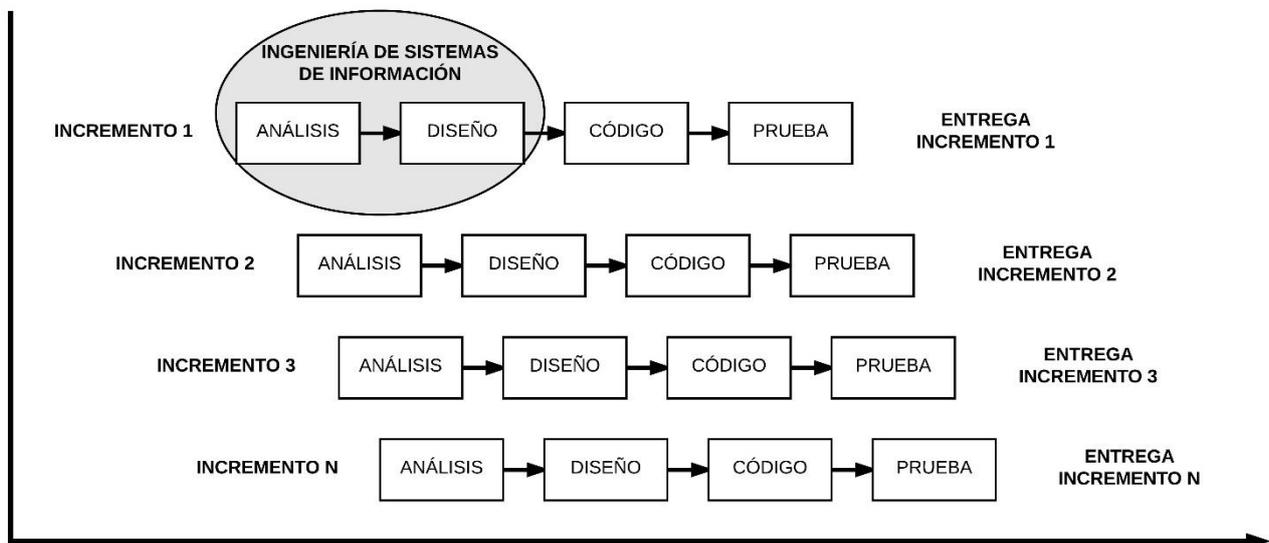


Figura 18 Esquema del Modelo Incremental (Pressman, 2001)

2.5.3 Ingeniería de Integración

Una vez definidos los incrementos, es necesario que estén organizados, donde se debe separar las funciones de otras funciones de los incrementos, es decir, identificar los requerimientos más importantes y acomodarlos de manera adecuada en cada incremento, minimizando la cantidad de cambios sin afectar la funcionalidad del incremento anterior. En

ese sentido, la ingeniería de integración crea un ambiente adecuado para integrar los incrementos, con estrategias y una planeación para su unificación (Tabla 11) (Mills, 1980).

Tabla 11 Plan de Integración de Incrementos

Planeación	Estrategia
Desarrollo incremental del software	Ejercicios de incremento inicial
Gestión de especificación de interfaz de usuario	Ejercicio de incremento intermedio
Metodología de integración del software	Ejercicios de incremento provisional
Simulación de software	Ejercicios de incremento final

Nota: Tomado de (Mills, 1980).

A través de estas estrategias, las integraciones de los incrementos deben ser funcionales o con flujo de procesamiento. La integración funcional, organiza los incrementos con las principales capacidades del sistema de cualquier fase, mientras que, mediante la integración por flujo de procesamiento, abarca un subconjunto de interfaces totales durante la fase de integración dada (Figura 19) (Mills, 1980).

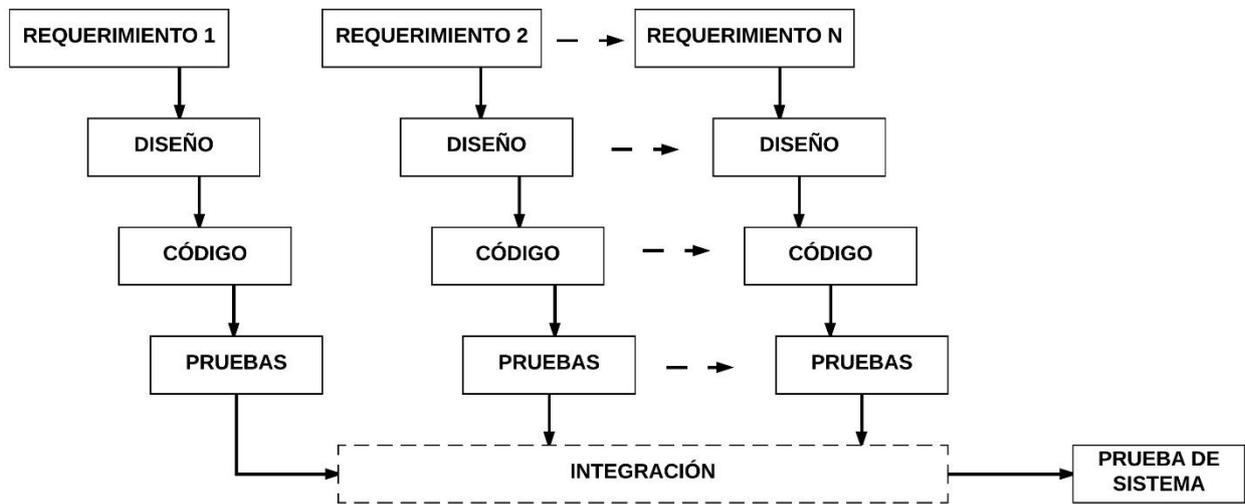


Figura 19 Integración de Incrementos (Mills, 1980)

CAPÍTULO 3 PATRONES DE DISEÑO

3.1 Introducción

Anteriormente, cuando se realizaba un desarrollo de software no se planteaba una serie de técnicas que permitan el desarrollo de mejor manera, propiciando que muchos de estos proyectos desarrollados, llegaran al fracaso o surja un retraso en su despliegue, generando un impacto muy alto, principalmente en costos.

Existen muchos desarrollos de software que fueron creados mediante el uso del paradigma orientado a objetos (OO). Ante la posible amenaza que puedan generar el desarrollo correcto de los proyectos de software, los desarrolladores de software requieren de directrices y herramientas que les permitan crear el software de una manera adecuada y que sean guiados mediante estas directrices.

El uso del paradigma orientado a objetos en conjunto con directrices y conceptos adecuados ayudará al desarrollo correcto del software. Esto dio origen a la creación de los denominados “Patrones de Diseño”. Los patrones de diseño, son directrices que los desarrolladores necesitan para ser guiados durante el análisis, diseño y desarrollo del software. De acuerdo con Christopher Alexander, un patrón de diseño se define (Gamma, Helm, Johnson, & Vlissides, 1994).

“Cada patrón describe un problema encima de nuestro entorno y entonces describirá la solución básica del problema, de tal forma que se puede usar esta solución millones de veces, sin nunca hacer el mismo camino dos veces”.

El uso de un patrón de diseño durante el desarrollo de software, será acorde a la problemática que tenga el desarrollador a través de patrones de creación, de estructura y de comportamiento, que le faciliten el desarrollo de su proyecto. En este capítulo se abordará de manera concreta, algunos patrones de diseño que contiene cada clasificación, con su estructura demostrada en un ejemplo de manera teórica, para saber cómo funciona cada patrón de diseño.

3.2 Patrones de Creación

Divididos en patrones de objetos y clases, esta categoría de patrones se centra en crear instancias que sean independientes de los sistemas. El primero crea instancias con comportamientos particulares mientras que el segundo usa el concepto de herencia para varear la clase creada de una instancia. También, maneja otros conceptos como ocultamiento y encapsulación. La categoría patrones de creación, está conformada por 5 patrones (Gamma, Helm, Johnson, & Vlissides, 1994).

3.2.1 Abstract Factory

Este patrón trabaja a través de una interfaz para crear objetos relacionados o dependientes sin ser especificados, coleccionando métodos abstractos y propiedades que devuelven varios objetos diferentes a petición Su estructura está conformada por 5 clases para poder ser implementada (Tabla 12) (Figura 20) (Gamma, Helm, Johnson, & Vlissides, 1994)).

Tabla 12 Clases de Abstract Factory

Clase	Clase-Ejemplo	Definición
AbstractFactory	AbstractFactory	Declara una interfaz para operaciones de productos que crean objetos abstractos
ConcreteFactory	<ul style="list-style-type: none"> ShapeFactory ShapeColor 	Implementa las operaciones de productos que crean objetos concretos.
AbstractProduct	<ul style="list-style-type: none"> Shape Color 	Declara una interfaz de producto para un tipo de objeto.
ConcreteProduct	FactoryProducer	Define un objeto producto a ser creado correspondiente de ConcreteFactory e implementa la interfaz AbstractFactory
Client	AbstractFactoryPatternDemo	Usa la interfaz declarada por las clases AbstractFactory y AbstractProduct.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015).

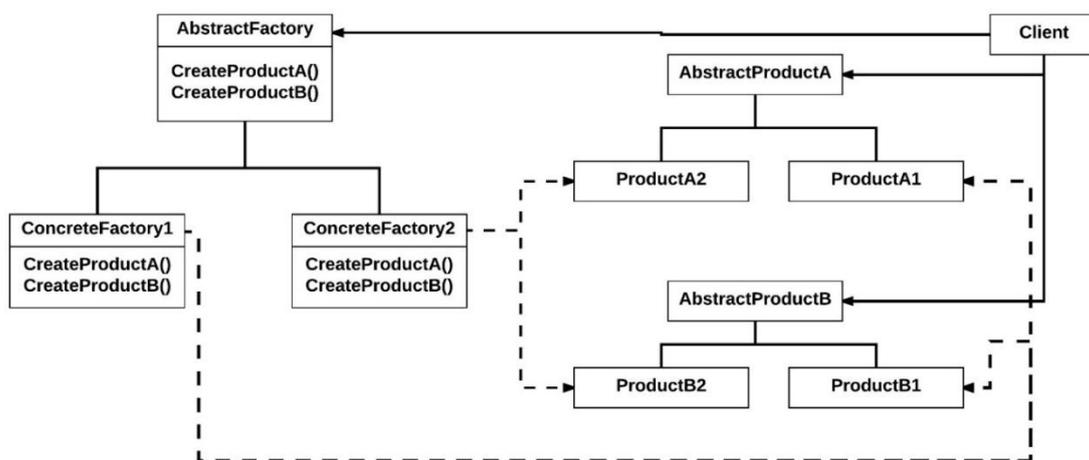


Figura 20 Estructura de Abstract Factory (Gamma, Helm, Johnson, & Vlissides, 1994)

3.2.1.1 Ejemplo De Implementación

El ejemplo se basa en figuras geométricas donde se les asignara un color (**Figura 21**). Creando dos interfaces (Shape y Color), se asignan el nombre del color y figura. La clase ShapeFactory y ColorFactory, cada uno de manera individual, obtiene el color y la figura previamente declaradas en las interfaces.

La clase AbstractFactory unifica la figura y su color para ser obtenido pro FactoryProducer y obtener los resultados de AbstractFactory y la clase cliente será representada por AbstractFactoryPatternDemo, quien visualizará la figura y el color.

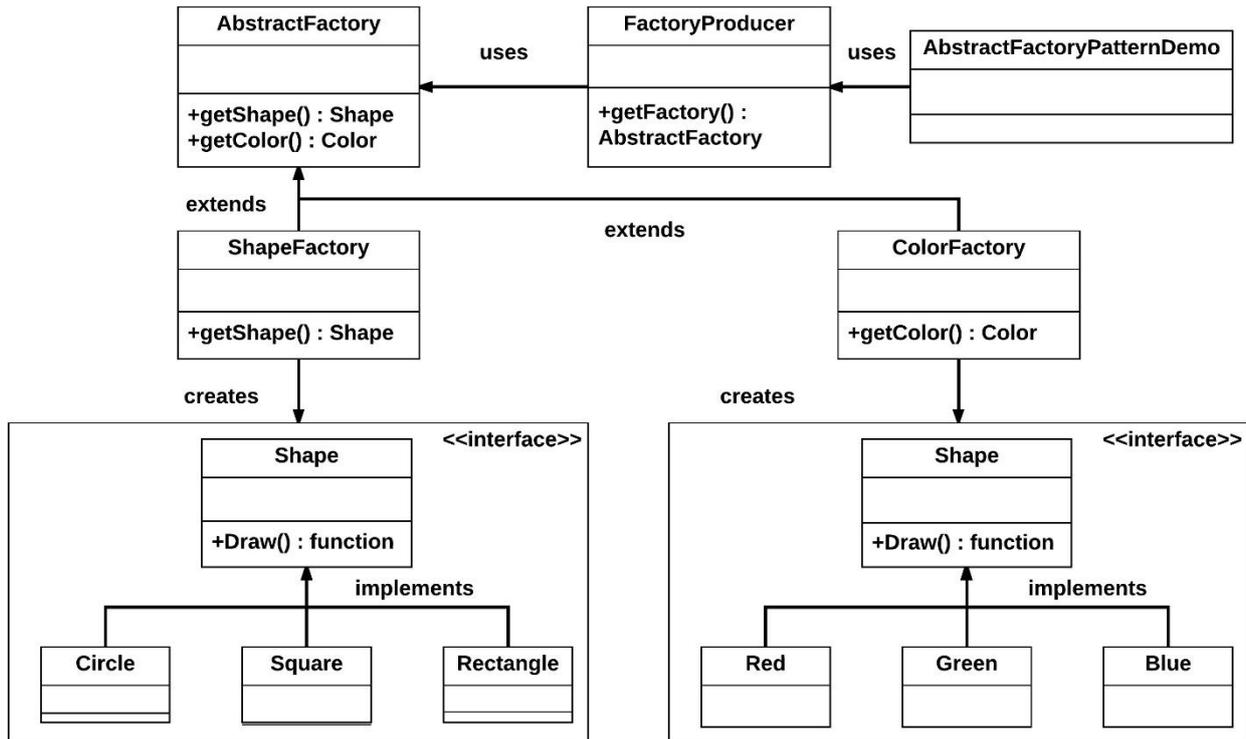


Figura 21 Ejemplo de Implementación (Tutorials Point , 2015)

3.2.2 Factory Method

Este patrón ayuda a crear objetos sin exponer su lógica de creación a través de una interfaz común, donde devuelve una de varias clases posibles, teniendo una clase padre con sus métodos que cumplen con tareas y datos diferentes. Su estructura se compone de 4 clases (Tabla 13) (Figura 22) (Gamma, Helm, Johnson, & Vlissides, 1994).

Tabla 13 Clases de Factory Method

Clase	Clase-Ejemplo	Descripción
Product	Shape	Define la interfaz del objeto del Factory Method creado.
ConcreteProduct	<ul style="list-style-type: none"> • Circle • Square • Rectangle 	Implementa la interfaz Product.

Creator	FactoryPatternDemo	Declara el Factory Method, que retorna un objeto de tipo ProductCreator también puede definir una implementación default del Factory Method que retorna un objeto ConcreteProduct y llamarlo para crear un objeto Product.
ConcreteCreato	ShapeFactory	Anula el Factory Method que retorna una instancia de ConcreteProduct.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015) .

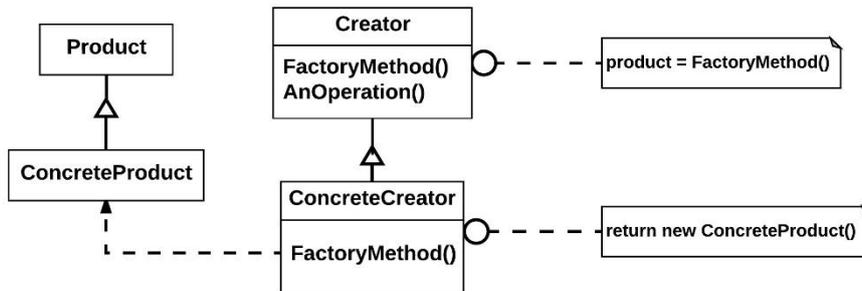


Figura 22 Estructura de Factory Method (Gamma, Helm, Johnson, & Vlissides, 1994)

3.2.2.1 Ejemplo De Implementación

Usando las figuras para este ejemplo que son las 3 clases (Circle, Square, Rectangle), implementaran la interfaz creada Shape para dibujar las figuras. La clase ShapeFactory obtiene los objetos que salen de Shape a través de la clase FactoryPatternDemo (Figura 23).

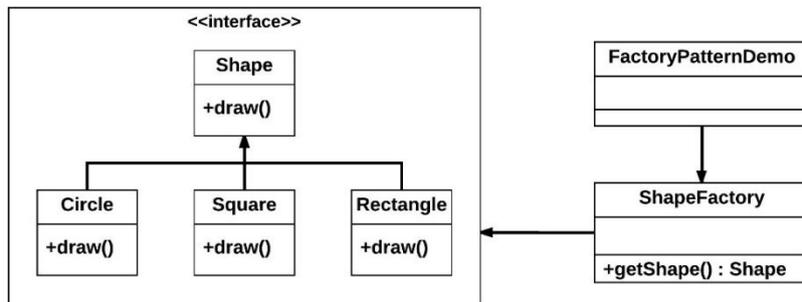


Figura 23 Ejemplo de Implementación (Tutorials Point , 2015)

3.2.3 Prototype

Prototype es un patrón que permite la creación de objetos duplicados manteniendo su rendimiento. Suele ser utilizado cuando un sistema debe ser independiente en su creación, estructura y representación. Su estructura se basa en 3 clases (Tabla 14) (Figura 24) (Tutorials Point , 2015).

Tabla 14 Clases de Prototype

Clase	Clase-Ejemplo	Descripción
Prototype	Shape	Declara la interfaz para clonarse a sí mismo.
ConcretePrototype	ShapeCache	Implementa una operación para clonarse a sí mismo.
Client	PrototypePatternDemo	Crea un nuevo objeto preguntando a un prototipo para clonarse a sí mismo.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015).

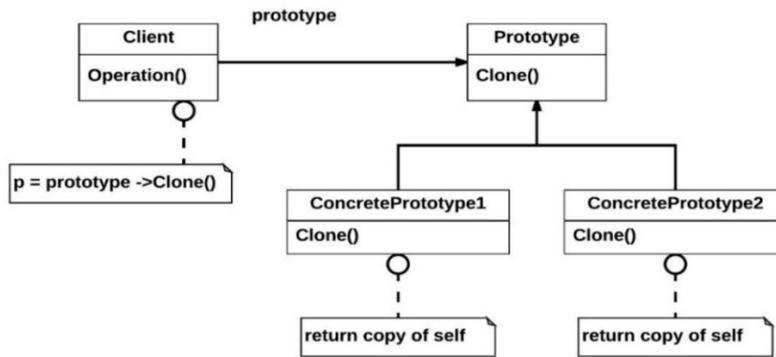


Figura 24 Estructura de Prototype (Gamma, Helm, Johnson, & Vlissides, 1994)

3.2.3.1 Ejemplo De Implementación

Con base en las figuras, se crean 3 clases de que representan cada figura (Circle, Rectangle, Sqaure), las cuales, se extienden de la clase abstracta Shape. La clase ShapeCache, almacena los objetos es un array para que devuelva un clon cuando sea solicitado por la clase PrototypePatternDemo, obteniendo así, los objetos de la clase Shape (Figura 25).

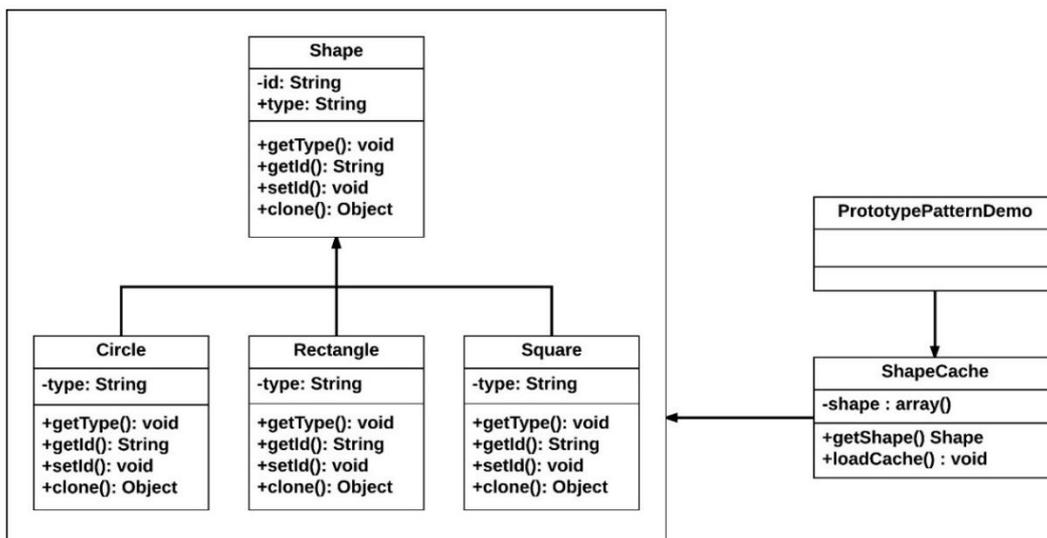


Figura 25 Ejemplo de Implementación (Tutorials Point , 2015)

3.3 Patrones de Estructura

Los patrones de estructura funcionan a través de la composición de objetos y clases. Se toma el concepto por la composición de interfaces y objetos para nuevas funcionalidades. Los patrones enfocados a las clases, describen el uso de la herencia. Los patrones enfocados a objetos, describen la composición de su estructura, es decir, objetos dentro de objetos (Gamma, Helm, Johnson, & Vlissides, 1994).

3.3.1 Adapter

Adapter es un patrón que une dos interfaces diferentes con una única clase, a través de la herencia. Su estructura se compone de 4 clases (Tabla 15) (Figura 26) (Gamma, Helm, Johnson, & Vlissides, 1994).

Tabla 15 Clases de Adapter

Clase	Clase-Ejemplo	Definición
Target	<ul style="list-style-type: none"> MediaPlayer AudioPlayer 	Define el dominio específico de la interfaz que usa el cliente.
Client	AdapterPatternDemo	Colabora con los objetos conformados de la interfaz Target.
Adaptee	AdvancedMediaPlayer	Define una interfaz existente que se necesita adaptar.
Adapter	MediaAdapter	Adapta la interfaz de Adaptee de la interfaz Target.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point, 2015).

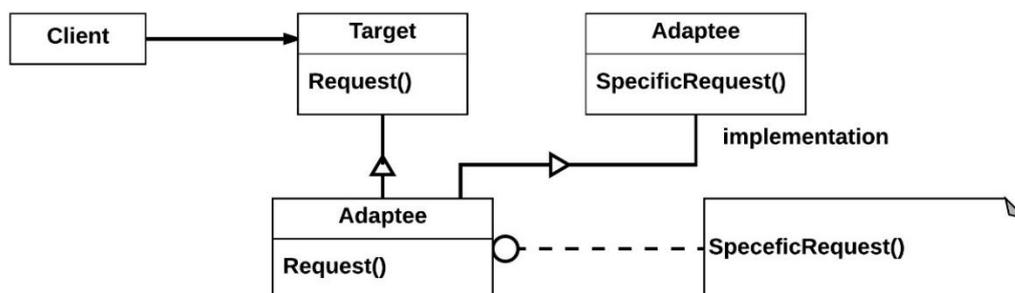


Figura 26 Estructura de Adapter (Gamma, Helm, Johnson, & Vlissides, 1994)

3.3.1.1 Ejemplo De Implementación

La representación de este patrón, se basará en un reproductor de música, al tratar de reproducir cualquier formato de música. Se crean dos clases (MediaAdapter y AudioPlayer) que contienen el tipo de archivo a reproducir implementado la interfaz MediaPlayer. Se crean dos clases (VlcPlayer y Mp4Player) que implementen la interfaz AdvancedMediaPlayer para crear los objetos y ser utilizados por la clase MediaAdapter.

AudioPlayer usa la clase MediaAdapter que pasa un tipo de audio deseado sin saber la clase actual que puede reproducir el formato. AdapterPatternDemo, usara la clase AudioPlayer para reproducir varios formatos (Figura 27).

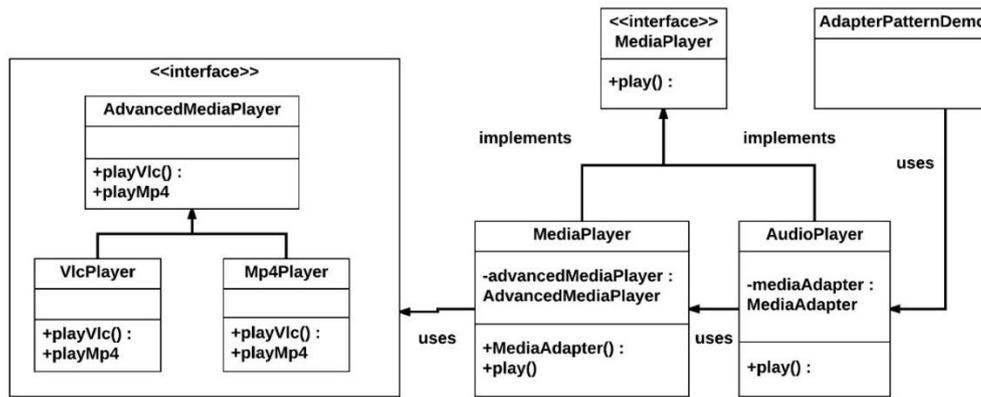


Figura 27 Ejemplo de Implementación (Tutorials Point , 2015)

3.3.2 Composite

Representa la jerarquía parcial o total del objeto que contiene componentes individuales con un objeto o una colección de objetos en su estructura de árbol para ser manipulados y sean ignorados por el cliente. La estructura de Composite la componen 4 clases y además como debe estar compuesto un objeto (Tabla 16) (Figura 28) (Gamma, Helm, Johnson, & Vlissides, 1994).

Tabla 16 Clases de Composite

Clase	Clase-Ejemplo	Definición
Component		Posee varias características entre las que se destacan: <ul style="list-style-type: none"> • Declarar una interfaz para los objetos en la composición. • Implementa un comportamiento por default para la interfaz común de todas las clases. • Declara una clase para acceder y manejar pequeños componentes.
Leaf		Representa el objeto hoja en su composición y no tiene hijos y define el comportamiento para objetos primitivos en su composición.
Composite	Employee	Define el comportamiento de componentes que tienen hijos, los almacena e implementa las operaciones relacionadas con los hijos en la interfaz de Component.
Client	CompositePatterDemo	Manipula los objetos de su composición mediante la interfaz de Component.

Nota: Información Tomada de(Gamma, Helm, Johnson, & Vlissides, 1994)

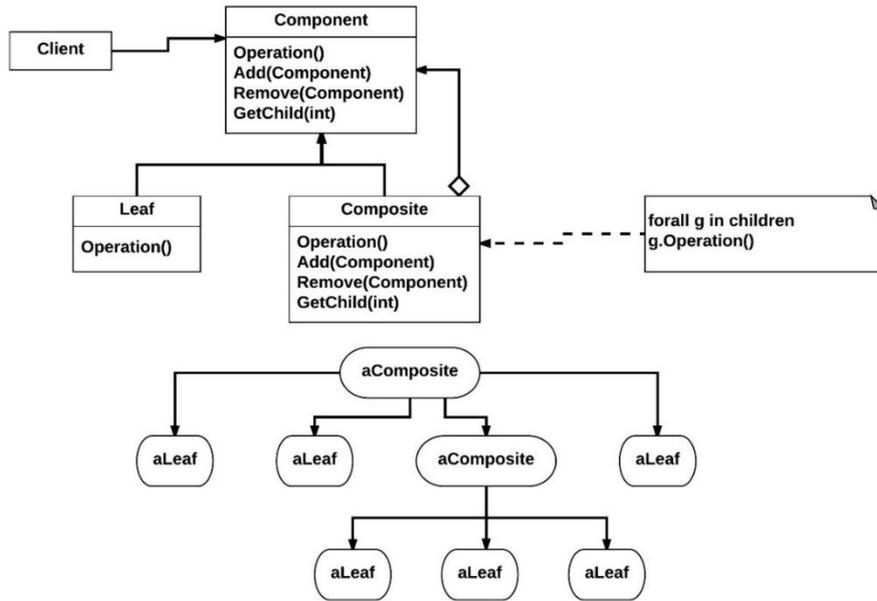


Figura 28 Estructura de Composite (Gamma, Helm, Johnson, & Vlissides, 1994)

3.3.2.1 Ejemplo De Implementación

La representación de composite será a través de la jerarquía de puestos de los empleados de una empresa. Se crea una clase Employee que actúa como una clase de agente del patrón Composite. CompositePatternDemo es una clase que usara Employee para añadir la jerarquía de nivel departamental y mostrara todos los empleados (**Figura 29**).

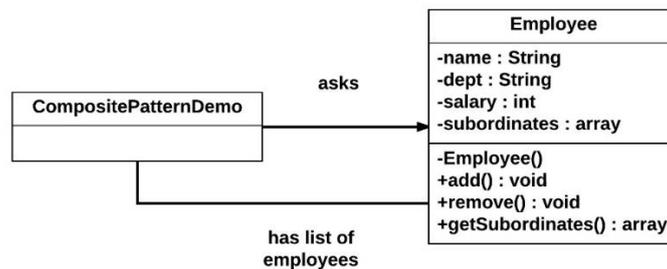


Figura 29 Ejemplo de Implementación (Tutorials Point , 2015)

3.3.3 Decorator

Decorator es un patrón que añade funcionalidad a un objeto de manera flexible y dinámica sin alterar su estructura, funcionando como un contenedor de clase existente sin la necesidad de crear clases derivadas. Su estructura se compone de 4 clases (*Tabla 17*) (*Figura 30*) (Gamma, Helm, Johnson, & Vlissides, 1994).

Tabla 17 Clases de Decorator

Clase	Clase-Ejemplo	Definición
Component	Shape	Define la interfaz para objetos agregados dinámicamente que puedan tener responsabilidades.

ConcreteComponent	Circle Rectangle	Define un objeto para que le agreguen responsabilidades que puedan ser atacadas.
Decorator	ShapeDecorator	Mantiene una referencia del objeto Component y define una interfaz que conforma Components.
ConcreteDecorator	RedShapeDecorator	Da responsabilidades al componente..

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015).

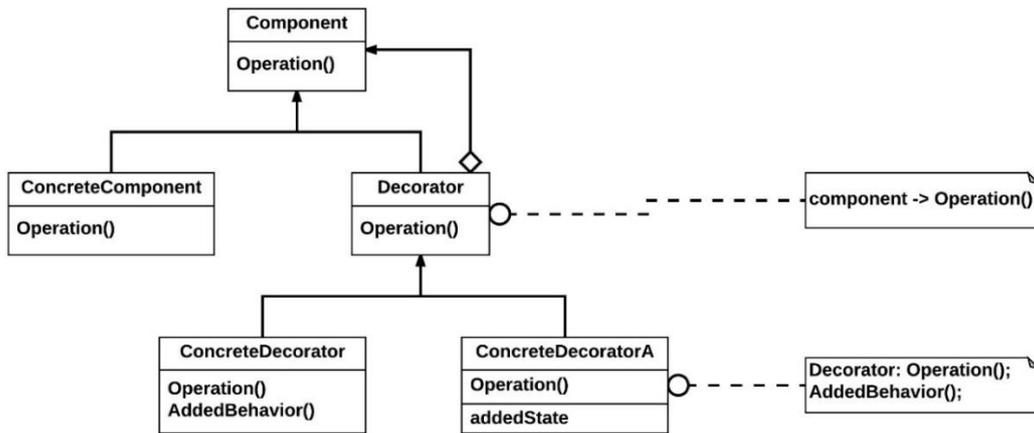


Figura 30 Estructura de Decorator (Gamma, Helm, Johnson, & Vlissides, 1994)

3.3.3.1 Ejemplo De Implementación

Se representa el círculo y rectángulo para ser decoradas de un color su contorno. Se crea una interfaz llamada Shape y dos clases que representan las figuras e implementan la interfaz. (Circle y Rectangle).

Se crea una clase abstracta llamada ShapeDecorator implementando la interfaz Shape. RedShapeDecorator que implementa la interfaz ShapeDecorator donde obtendrá el color del borde de las figuras. DecoratorPatternDemo, es una clase que usara RedShapeDecorator para decorar el objeto Shape ().

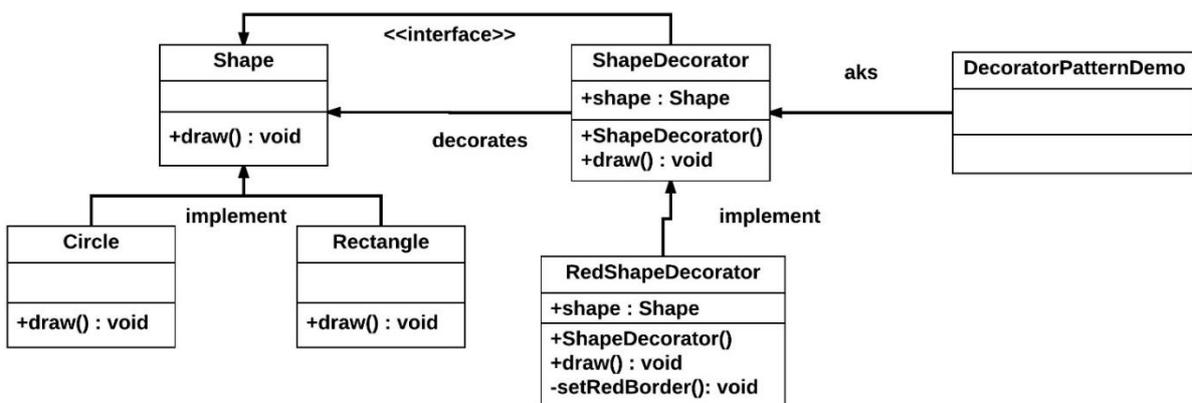


Figura 31 Ejemplo de Implementación (Tutorials Point , 2015)

3.4 Patrones de Comportamiento

Los patrones de comportamiento se enfocan en los algoritmos y la asignación de responsabilidades entre objetos. Se describe patrones de clases, objetos y de comunicación, caracterizados por el flujo de control complejo y la dificultad de seguimiento en los tiempos de ejecución, centrándose principalmente en la comunicación (Gamma, Helm, Johnson, & Vlissides, 1994).

3.4.1 Chain of Responsibility

Permite a un conjunto de objetos controlar solicitudes específicas basadas en su tipo, a través del desacoplamiento del remitente y destinatario. Para recibir la solicitud, los objetos recibidos se encadenan para ser manejados, en caso contrario, pasa al siguiente. Su estructura se compone de 3 clases (**Tabla 18**) (*Figura 32*) (Gamma, Helm, Johnson, & Vlissides, 1994).

Tabla 18 Clases de Chain Of Responsibility

Clase	Clase-Ejemplo	Definición
Handler	AbstractLogger	Define una interfaz para manejar solicitudes y puede implementar una liga sucesora.
ConcreteHandler	<ul style="list-style-type: none"> • ConsoleLogger • ErrorLogger • FileLogger 	Maneja la solicitud responsable para tener acceso al sucesor. Si ConcreteHandler puede manejar una petición, lo hace; de lo contrario reenvía la petición a su sucesor.
Client	ChainPatternDemo	Inicia la solicitud al objeto de cadena ConcreteHandler.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015).

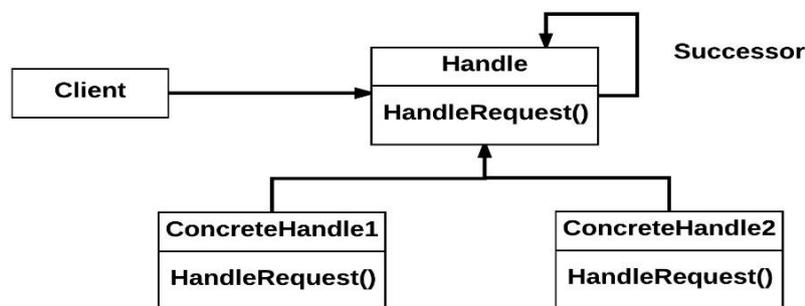


Figura 32 Estructura de Chain of Responsibility (Gamma, Helm, Johnson, & Vlissides, 1994)

3.4.1.1 Ejemplo De Implementación

La forma en cómo funciona el patrón, se representará mediante la generación de tres tipos de mensaje. Se crea una clase abstracta llamada AbstractLogger, para el registro de los mensajes los cuales sera recibido por las tres clases creadas, es decir los tipos de mensaje que recibirá la clase abstracta (ConsoleLogger, ErrorLogger y FileLogger), donde imprime mensajes diferentes. ChainPatternDemo, será el encargado de hacer funcionar el programa (*Figura 33*).

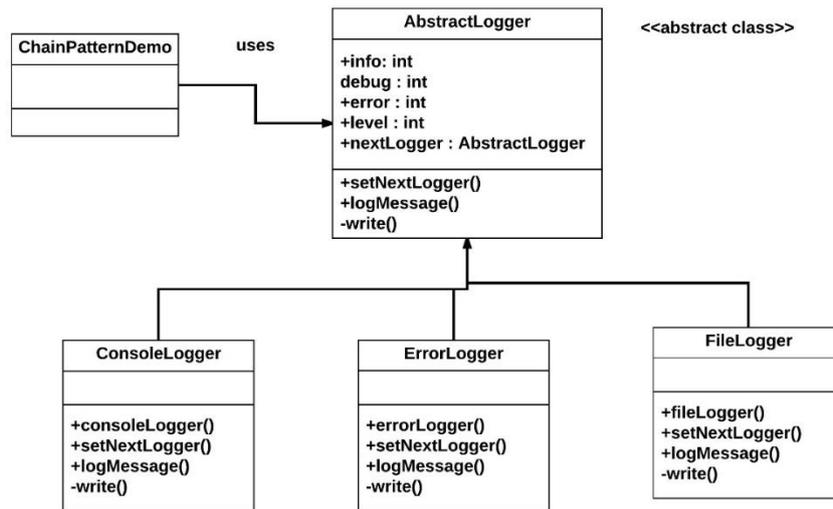


Figura 33 Ejemplo de Implementación (Tutorials Point , 2015)

3.4.2 Mediator

Mediator es un patrón que reduce la complejidad de comunicación entre objetos o clases, a través de la unión de una única clase, capaz de manejar todas las comunicaciones de las diferentes clases. Su estructura se compone de 3 clases (**Tabla 19**) (Figura 34) (Gamma, Helm, Johnson, & Vlissides, 1994).

Tabla 19 Clases de Mediator

Clase	Clase-Ejemplo	Definición
Mediator		Define una interfaz para comunicarse con los objetos de tipo Colleague.
ConcreteMediator	ChatRoom	Implementa el comportamiento cooperativo mediante la coordinación de los objetos Colleague, los conoce y mantiene.
ColleagueClasses	User	Cada clase Colleague conoce al objeto Mediator, además de comunicarse con el otro mediator.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015).

3.4.2.1 Ejemplo De Implementación

Se describe como ejemplo la sala de un chat donde múltiples usuarios interactúan a través del envío de mensajes. Se crean dos clases, ChatRoom y User. El objeto User usara el método de ChatRoom para compartir los mensajes entre los usuarios. MediatorPatternDemos, es la clase que usar la clase User para mostrar la comunicación entre ellos (Figura 35).

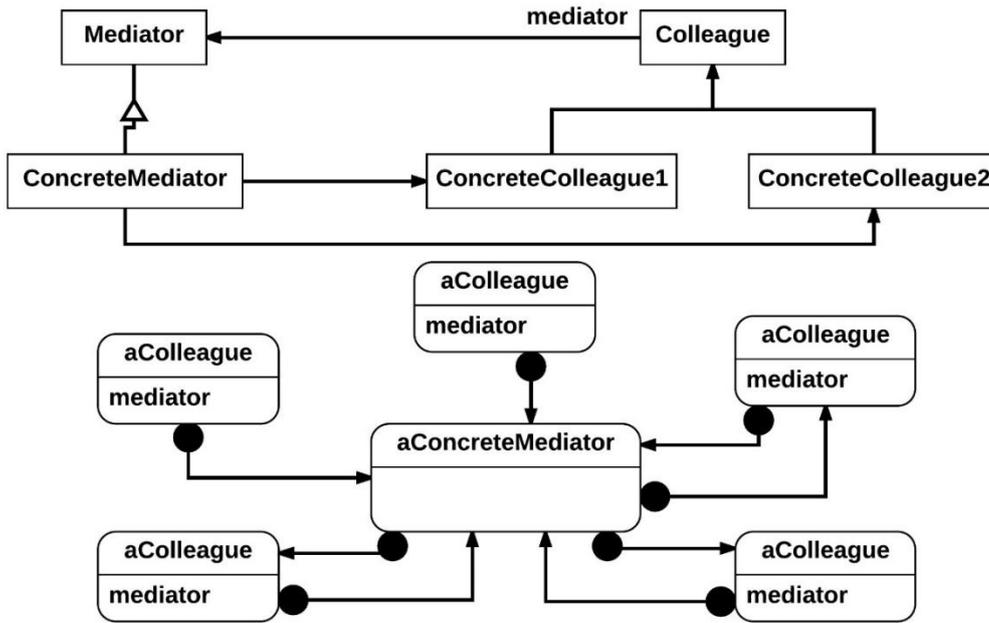


Figura 34 Estructura de Mediator (Gamma, Helm, Johnson, & Vlissides, 1994)

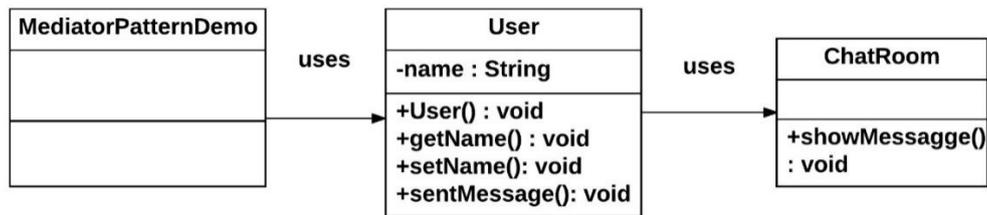


Figura 35 Ejemplo de Implementación (Tutorials Point , 2015)

3.4.3 Template Method

Define una plantilla de algún algoritmo en una operación, permitiendo que las subclases redefinan ciertos pasos de algún algoritmo sin cambiar la estructura del algoritmo. Su estructura cuenta con 2 clases (Tabla 20) (Figura 36) (Cooper, 2000).

Tabla 20 Clases de Template Method

Clase	Clase-Ejemplo	Definición
AbstractClass	Game	Define las abstracciones primitivas que ConcreteClass define para implementar los pasos del algoritmo y el Template Method con un esqueleto definido de un algoritmo para llamar las operaciones primitivas definidas en AbstractClass
ConcreteClass	<ul style="list-style-type: none"> • Cricket • Football 	Implementa las operaciones primitivas que lleva a cabo las etapas de cada subclase, especifica del algoritmo.

Nota: Información Tomada de (Gamma, Helm, Johnson, & Vlissides, 1994) y (Tutorials Point , 2015).

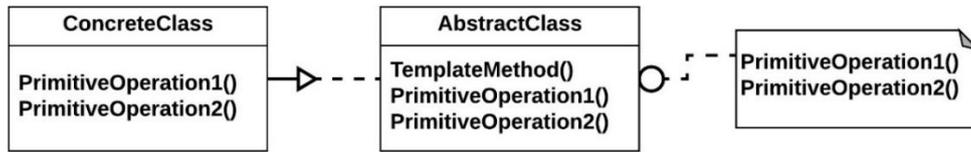


Figura 36 Estructura de Template Method (Gamma, Helm, Johnson, & Vlissides, 1994)

3.4.3.1 Ejemplo De Implementación

Se hace una descripción de dos deportes: cricket y fútbol. Se crea una clase abstracta llamada Game, que define un conjunto de operaciones como plantillas, de manera que no se anulan. Cricket y Football son dos clases creadas que heredan de Game cuyos métodos se invalidan por ser similares a los de Game. TemplatePatternDemo, es una clase que usará Game para demostrar el uso del patrón (Figura 37).

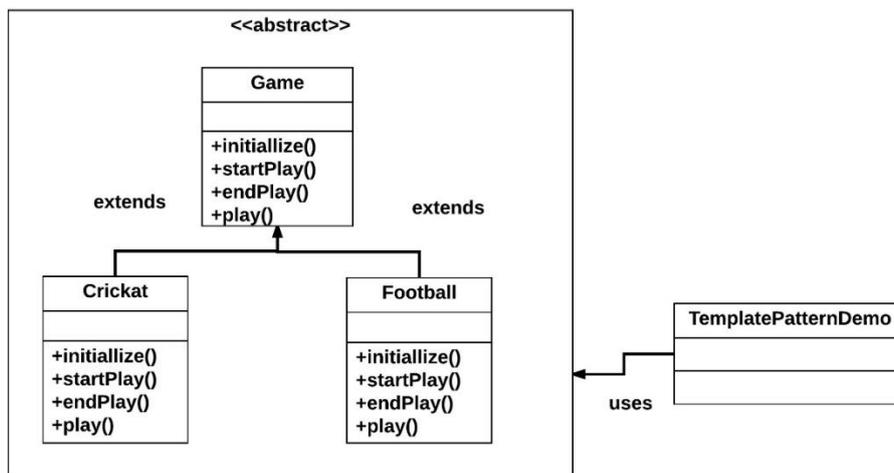


Figura 37 Ejemplo de Implementación (Tutorials Point , 2015)

3.5 Modelo-Vista-Controlador

El modelo vista-controlador (MVC) organiza y separa el software en 3 distintos roles requeridos en una interfaz gráfica de usuario (GUI) (Figura 38) (Sweat, 2006).

- 1- Modelo. Encapsula los datos de la aplicación, flujo y lógica.
- 2- Vista. Extrae los datos del modelo y formatos para presentación.
- 3- Controlador. Controla los flujos directos de la aplicación y recibe las entradas que traslada al modelo y vista.

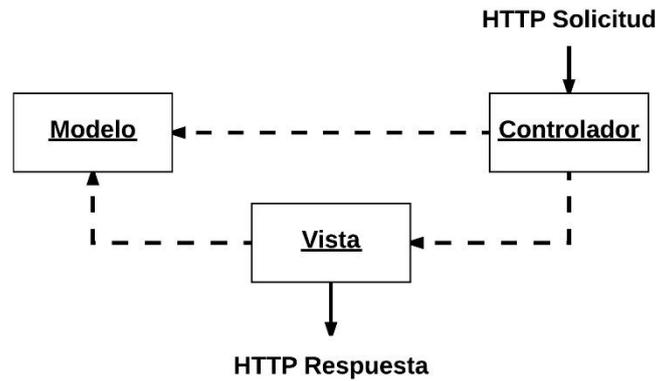


Figura 38 Estructura de MVC (Sweat, 2006)

El modelo será responsable de los llamados de datos negocio o lógica de aplicación (Figura 39). Es por eso que MVC nunca tiene clases individuales, sino, se implementan como un grupo de objetos relacionados donde cada objeto de MVC realiza una tarea específica.

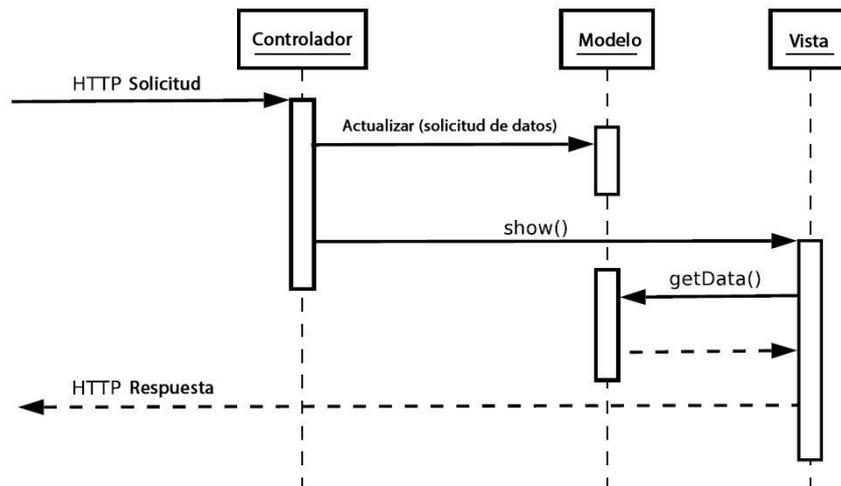


Figura 39 Funcionamiento de MVC (Sommerville, 2005)

3.5.1 Ejemplo De Implementación

En un ejemplo sencillo para representar MVC, se toma un termostato, el cual contiene los 3 elementos de MVC, además de su diagrama de clases (*Figura 40, Figura 41*).

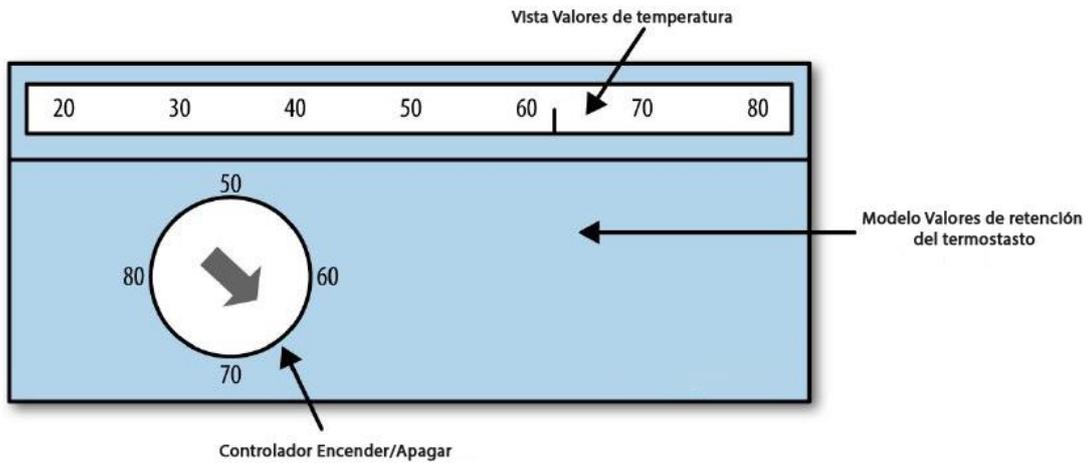


Figura 40 Ejemplo de Implementación (Sweat, 2006)

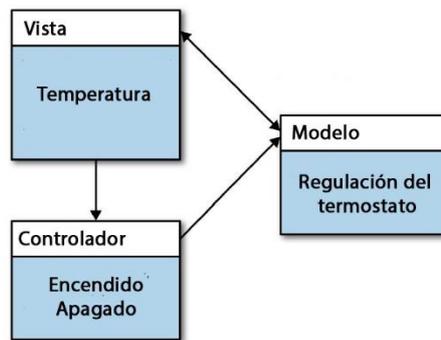


Figura 41 Diagrama de Clases de MVC (Sweat, 2006)

CAPÍTULO 4 TENDENCIAS TECNOLÓGICAS

4.1 Introducción

La evolución de la tecnología ha generado un impacto en el corto plazo en todos los usuarios que, a través de internet, se conectan desde cualquier parte del mundo por consultar, informar, compartir e interactuar con otros usuarios que, sin el internet y la creación de nuevas tecnologías, sería imposible.

El internet, siendo un factor de cambio global ha contribuido a que las compañías desarrollen sus proyectos de software enfocados a ofrecer algún tipo de servicio o control interno a través de diferentes dispositivos, desde paginas estáticas hasta la web semántica o cognitiva, capaces de predecir nuestras preferencias.

Un estudio realizado por Simon Kemp para “We Are Social”, revela que los usuarios de todo el mundo, van creciendo en cuanto al uso del internet y la manera en ¿cómo? y ¿para qué? es usada la navegación de internet, distribuida regionalmente (*Figura 42*) (Kemp, 2017).

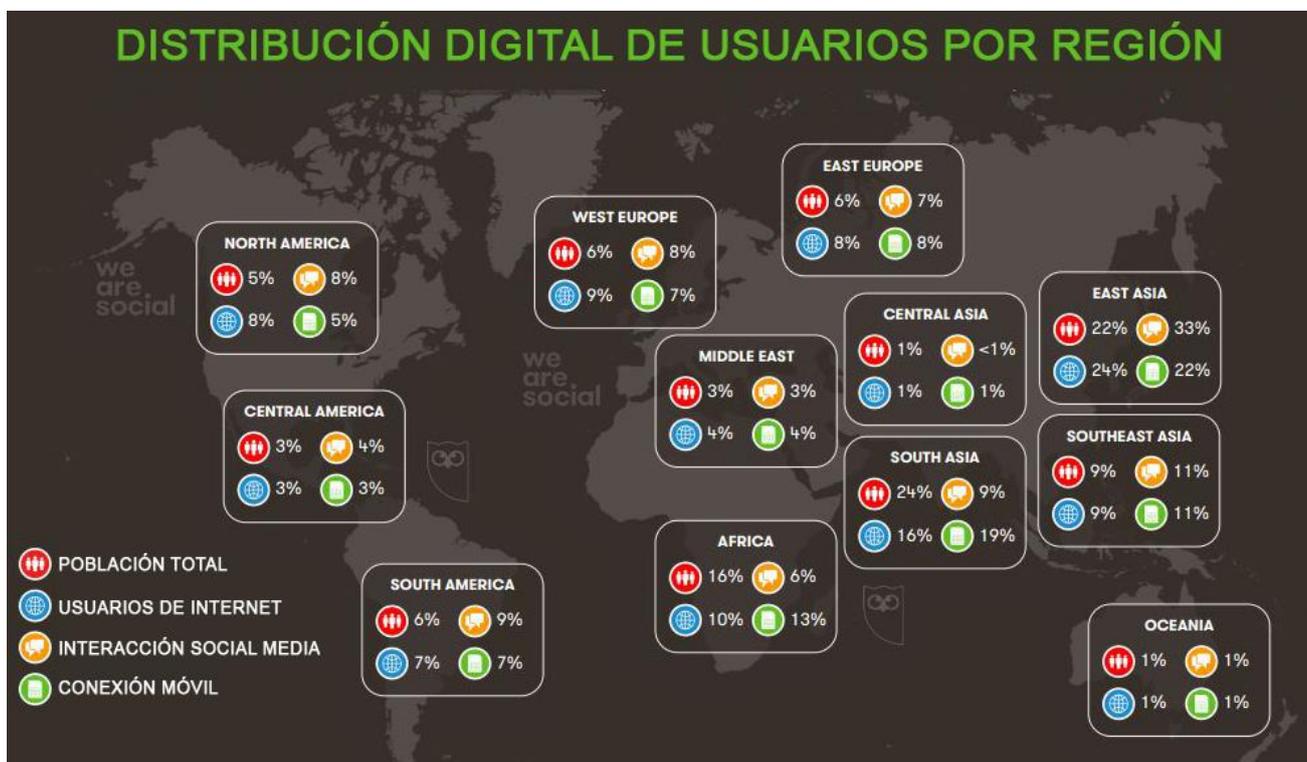


Figura 42 Uso de Internet Por Continente (Kemp, 2017)

A través de estudios que miden la manera en como interactuamos en internet, permite a las compañías determinar las preferencias de los usuarios para la mejora de servicios, contenidos, u otro tipo de consulta que pueda ser solicitada. También, es importante mencionar la manera en cómo se tiene acceso a internet a través de diferentes dispositivos determinando así, la manera de crear software y facilitar la problemática de los usuarios.

A partir de las nuevas necesidades de los usuarios, la competencia entre las grandes compañías tecnológicas por atraer nuevos usuarios, ha abierto nuevas áreas que están marcando tendencias. Con base en las principales tendencias tecnológicas, describiré algunas de ellas que considero, han sido importantes y que podrán marcar la manera de cómo vamos a interactuar entre personas y maquinas.

4.2 Realidad Virtual

El termino realidad virtual (virtual reality, VR), fue adoptado por Jaron Lainer, fundador de VPL Research en 1980, el cual es definido como un entorno 3D creado en computadora que permite a los usuarios interactuar con la realidad alterna, simulando algunas formas de realidad. Otros términos como se conocen a VR son: realidad artificial, ciberespacio, mundo virtual y ambiente virtual (Giraldi, Silva, & Oliveira).

La VR debe satisfacer 3 necesidades que hagan la diferencia: tiempo real, interacción e inmersión. Para lograrlo, es necesario contar con dispositivos de entrada que permita al usuario obtener una respuesta del mundo virtual a través de sus sentidos (Martínez, 2011).



Figura 43 Realidad Virtual (Rasulury, 2015)

4.2.1 Tipos de Realidades

Existen diferentes tipos de VR para desarrollar y crear a través de la computadora, donde cada una posee características que se levemente se diferencian (Tabla 21).

Tabla 21 Tipos de Realidad Virtual

Tipo	Definición
Sistema Desktop de Realidad Virtual	El usuario ve la imagen en primera persona. Muestran una imagen 2D o 3D en una pantalla de computadora en lugar de proyectarla a un HMD. El usuario viaja en cualquier dirección dentro del mundo tridimensional que se muestra en un monitor, casco, gafas o pantalla de proyección (videojuegos).
	“Ver para creer”. El usuario se ve a sí mismo dentro de la escena. Es un integrante “visible” del mundo virtual porque ve la proyección de su

VR en Segunda Persona	imagen en un fondo o ambiente. Este sistema involucra percepciones y respuestas en tiempo real a las acciones de los humanos involucrados, que no llevan cascos, guantes, HMD's, gafas o cualquier otro tipo de interfaz.
Telepresencia	Los sistemas de telepresencia forman el grupo de aplicaciones de realidad virtual. Cámaras, dispositivos táctiles y de retroalimentación, ligados a elementos de control remoto que permiten manipular robots o dispositivos ubicados a distancia mientras se experimentan en forma virtual (Telemedicina, Telerrobótica).
Sistemas de Inmersión de Realidad Virtual	Sumergen al usuario en el mundo virtual, utilizando sistemas visuales tipo CAVE, con sensores de posición y movimiento. El usuario en el mundo virtual responde a los movimientos de la cabeza de manera similar a como ocurre en el mundo real. Los mundos de inmersión existen en 3 dimensiones. A través del envío de imágenes ligeramente diferentes a cada ojo se habilita la sensación de profundidad, perspectiva y dimensión.

Nota: Información Tomada de (BBVA Innovation Center, 2015).

A partir del año 2014, la VR retoma otra vez un repunte en cuanto al desarrollo de esta tecnología, permitiendo que grandes compañías como Facebook, Google, Samsung, HTC, entre otras, inviertan en el desarrollo de esta tecnología. Estas compañías han creado sus dispositivos para que los usuarios tengan acceso a la VR (*Tabla 22*).

Tabla 22 Compañías de Desarrollo de Realidad Virtual

Compañía	Dispositivo
Oculus Rift-Facebook	Oculus Rift
Sony	Project Morpheus
Samsung	Gear VR
HTC	HTC Vive
Google	Cardboard

Nota: Información Tomada de (BBVA Innovation Center, 2015).

4.2.2 Aplicaciones

Sin duda, el uso de la VR ha contribuido al desarrollo de muchas áreas que permitan a las personas interactuar con los entornos creados a través de computadoras, haciendo que la inmersión sea cada vez más fuerte (*Tabla 23*).

Tabla 23 Aplicación de la Realidad Virtual

Área	Aplicación
Ingeniería	1- Manipulación remota. 2- Manipulación de robots. 3- Prototipos virtuales. 4- Automatización y mejora de procesos.
Medicina	1- Capacitación, preparación y enseñanza de procedimientos complicados. 2- Mejoras terapéuticas.
Psicología	1- Atención a enfermedades psicológicas.
Arquitectura	1- Modelado virtual de sus diseños.

	2- Recorridos virtuales.
Social	1- Visitar lugares que no conoces.
	2- Visitar otros países y sus ciudades importantes.
	3- Recorrer museos, paisajes, etc.

Nota: Información Tomada de (BBVA Innovation Center, 2015).

Steven LaValle de la Universidad de Cambridge, define VR como (LaValle, 2017):

“Inducir comportamiento dirigido en un organismo usando la estimulación sensorial artificial, mientras que el organismo tiene poca o ninguna conciencia de la interferencia”.

Mark Zuckerberg, define la VR (BBVA Innovation Center, 2015):

“Estamos ante una nueva plataforma de comunicación. Lo increíble de esta tecnología es que sientes que estás realmente en otro lugar con otras personas. Imagina disfrutar de un partido de básquet en un asiento a pie de pista, estudiar en una clase llena de alumnos y profesores de todo el mundo o tener una cita médica, cara a cara, con tu doctor con tan solo ponerte unas gafas en casa”.

4.3 Realidad Aumentada

La realidad aumentada (AR o RA en inglés) se define como una combinación de un entorno real físico con objetos virtuales hechos en computadora con gráficos 3D mostrados en dispositivos de salida (Carmigniani & Furht, 2011).



Figura 44 Esquema de Realidades

AR trata de proporcionar información a través de los componentes gráficos hechos por computadora de los objetos del mundo real, aumentando la experiencia del usuario. En ese sentido, AR ha ido creciendo de manera importante en los últimos tiempos, donde su aplicación se ha diversificado en diferentes áreas permitiendo a las compañías, invertir en esta tecnología que está en pleno desarrollo (Tabla 24).

Tabla 24 Áreas de Aplicación de AR

Área	Aplicación
Medicina	<ul style="list-style-type: none"> • Análisis de imágenes biomédicas. • Simulación de sistemas fisiológicos. • Entrenamiento médico. • Visualización de procesos quirúrgicos. • Creación de mapas tridimensionales.
Educación	<ul style="list-style-type: none"> • Experiencia interactiva sobre diferentes temas abordados durante el aprendizaje.
Entretenimiento	<ul style="list-style-type: none"> • Juegos de consola (PlayStation, XBOX). • Apps Móvil. • Información adicional en deportes. • Cine.
Militar	<ul style="list-style-type: none"> • Información del piloto. • Simulaciones de vuelo.
Comercial	<ul style="list-style-type: none"> • Creación y presentación de prototipos. • Mercados de consumo.

Nota: Información Tomada de (Rojas & Echeverri, 2012).

AR aún está en fase de ser desarrollado para maximizar su uso entre los usuarios. Por ejemplo, el juego de Pokemon Go, creado para los dispositivos Android y IOS con características de AR (Figura 45). Se sigue investigando en crear nuevas herramientas que permitan el uso de AR sin requerir otros dispositivos, aunque existen algunos como: lentes, apps móviles, cascos virtuales, pantallas virtuales que se podrán construir y controlar a través de sensores (Carmigniani, y otros, 2010)



Figura 45 Realidad Aumentada (Krause, 2016)

4.4 Inteligencia Artificial

El estudio de la inteligencia artificial (IA), engloba una gran cantidad de conceptos que pueden ser definidos, de acuerdo a las diferentes perspectivas de cada persona. A través de estas definiciones, es necesario entender de manera clara IA. Con base en diferentes autores, la definición que considero adecuada para IA es:

“IA ES UNA CIENCIA QUE SE ENCARGA DE EMULAR EL COMPORTAMIENTO Y LA MANERA DE PENSAR DE LOS HUMANOS A TRAVÉS DE COMPUTADORAS Y CON AYUDA DE ALGORITMOS PARA LA TOMA DE DECISIONES”.

Empresas como Google, Apple, Microsoft, Intel y Samsung, han desarrollado IA en sus principales productos que incluyen machine-learning, data science, lenguaje natural, entre otros, para la solución de problemas reales (GP. Bullhound, 2016).

A través de la evolución tecnológica, IA trata de entender y solucionar la problemática real de las personas de una manera simple, tratando de predecir ciertos parámetros que tiene la persona, entre otras aplicaciones que tiene IA. IA ha permitido a las grandes empresas a entender nuestro comportamiento y generar datos que les ayuden a mejorar su servicio hacia el usuario, por ejemplo, Spotify, en las preferencias musicales o Google, tratando de adivinar que buscamos, haciendo más cognitiva la interacción a internet (*Figura 46*).

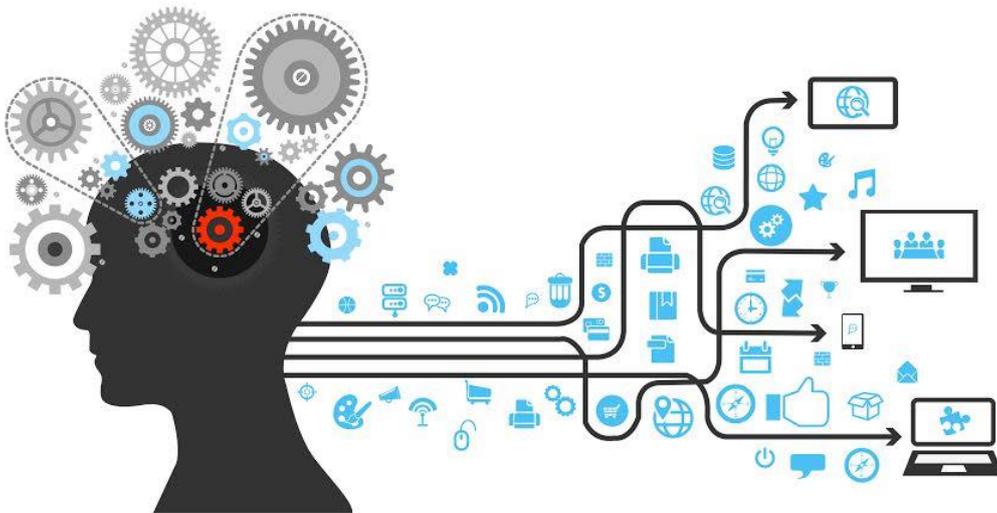


Figura 46 Inteligencia Artificial (Ansari, 2017)

Las grandes compañías de tecnología, han invertido mucho en IA para ofrecer muchos servicios y generar competencia entre ellos. Prueba de ello, es la realización de diferentes pruebas de casos de uso, implementan una gran cantidad de algoritmos necesarios para IA en conjunto de Big Data para el entrenamiento de los mismos, presentando una gran oportunidad de desarrollo profesional y generar nuevas áreas de oportunidad (*Figura 47*).

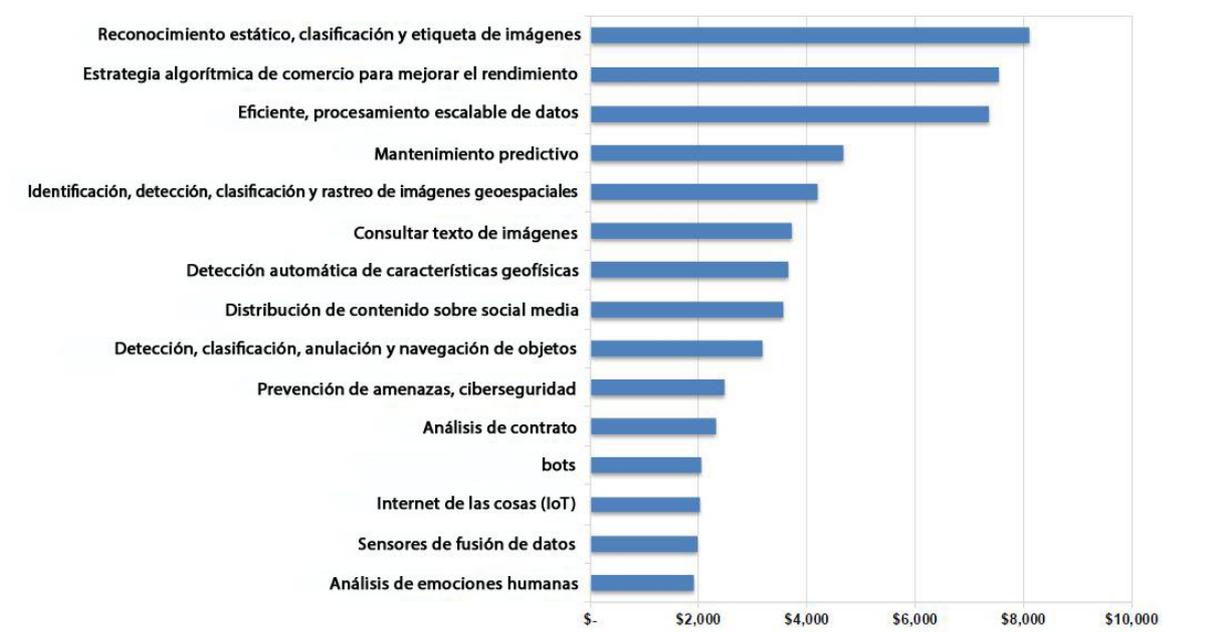


Figura 47 Casos de Uso y Áreas de Desarrollo de IA (Kaul & Wheelock, 2016)

4.4.1 Tendencias de IA

La evolución de IA ha permitido el desarrollo de áreas que van relacionadas con la capacidad de entendimiento de las personas, haciendo y simplificando aún más sus tareas cotidianas, en ese sentido y con base en diferentes artículos previamente consultados, describiré algunas tendencias que ha impactado la IA.

El desarrollo de apps inteligentes ayuda a estudiar el comportamiento de usuarios a través del uso de la app, con el fin de determinar y obtener respuestas para el mejoramiento de la misma para la satisfacción del usuario (Mobinius, 2017)..

Los asistentes virtuales se han convertido en una herramienta que facilite la interacción hombre-máquina. Los asistentes virtuales más usados son: Siri de Apple, Cortana de Microsoft, Google Now y Alexa de Amazon (Cobos, 2016).

Existen plataformas y herramientas open source que permite la colaboración entre pequeñas y grandes empresas a través de machine learning para la atracción y competencia entre ellas. Algunas de estas herramientas y plataformas son: TensorFlow, H2O, MLlib y Caffe (Mobinius, 2017). Actualmente, Google, Amazon, Microsoft, SAS, H2O.ai, utilizan IA para sus necesidades empresariales (Press, 2017).

Otra de las áreas que donde se usa la IA son en robots y la automatización de procesos, con la finalidad de reducir costos y tiempo de ejecución para dicha actividad que puede ser repetitiva (Press, 2017).

Algunas industrias han obtenido un impacto positivo por el uso de IA. Un estudio revelado por tratica, muestra el impacto que tendrá la IA durante los próximos años hasta el 2025,

mostrando el nivel del uso y la dependencia de IA para estos sectores (**Figura 48**) (Mobinius, 2017).

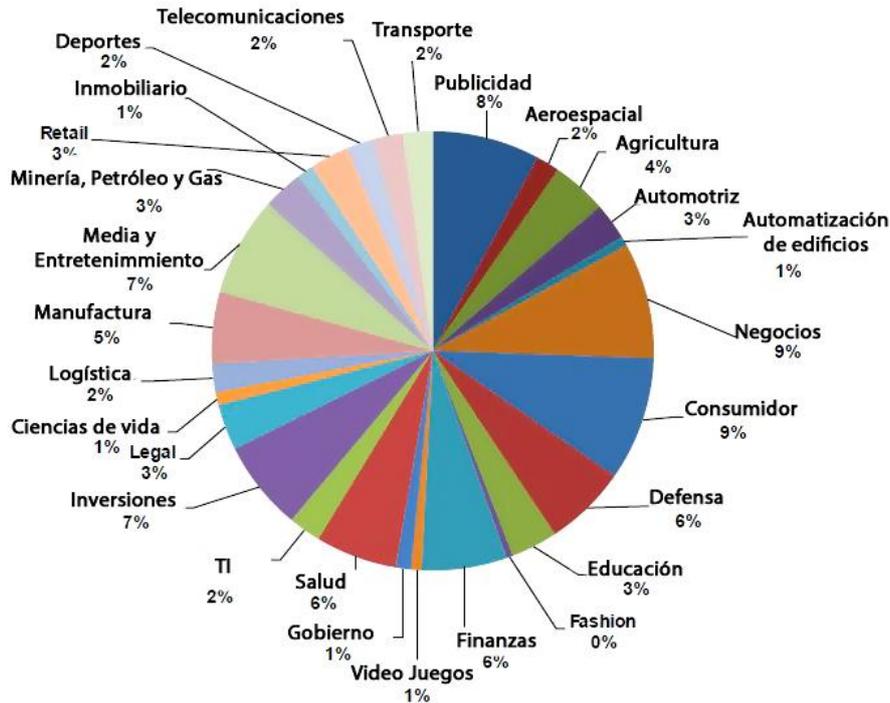


Figura 48 Futuro e Impacto de IA (Mobinius, 2017)

4.5 Big Data y Análisis de Datos

La conectividad a internet desde cualquier dispositivo, genera una gran cantidad de información y datos, que permite la toma de decisiones y el entendimiento de los mismos, las cuales son descritos (estructurados, no estructurados y semi-estructurados) y que tendrían ser costosos en términos de tiempo y de carga a una base de datos relacional para su análisis. El objetivo principal de Big Data, generar conocimiento (Fragoso, 2012).

Big data analiza toneladas de información que está clasificada en diferentes tipos de datos para ser analizados (**Figura 49**).



Figura 49 Áreas de Análisis de Big Data (Rayo, 2016)

El impacto de big data para la generación de conocimiento, ha permitido desarrollar tecnologías y modelos de negocio para el entendimiento y la toma de decisiones de muchos sectores orientados al usuario. Un estudio realizado por IBM como se implementa big data y el impacto que ha generado en diferentes áreas (**¡Error! No se encuentra el origen de la referencia.**).

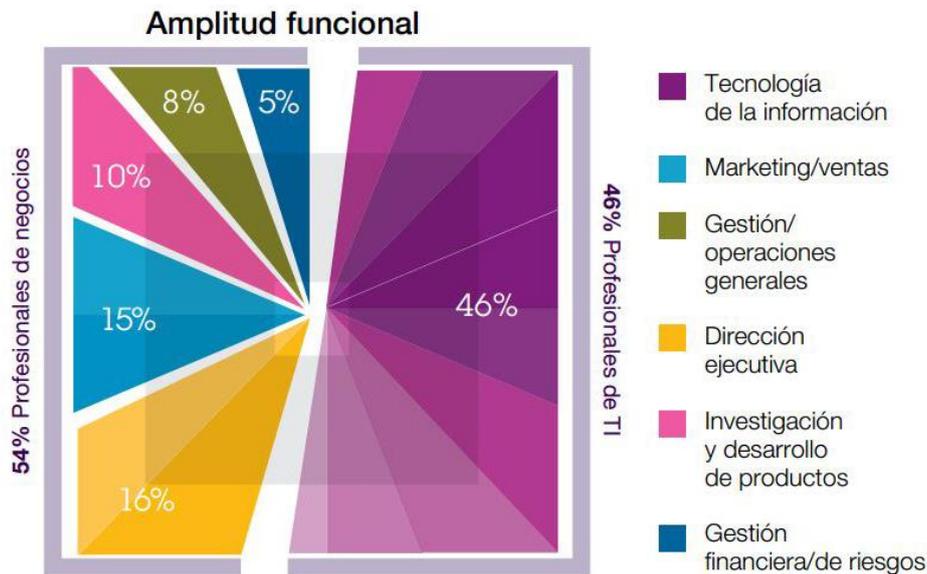


Figura 50 Uso e Impacto de Big Data (Schroeck, Shockley, Smart, Romero-Morales, & Tufano, 2014)

Un artículo publicado por el centro de innovación de BBVA cita puntualmente como visualizan Big Data del área de Data & Analytics (BBVA Innovation Center, 2015):

“Data y Big Data están cambiando muchas cosas, ya no solo a la hora de tomar decisiones comerciales, deportivas, políticas o de otra índole, sino también a la hora de crear productos, de innovar, almacenar datos, desarrollar, visualizar cosas... Es una tendencia que se ha generalizado y que parece que va a quedarse entre nosotros mucho tiempo.”

4.6 Internet of Thing

Internet de las cosas (internet of things IoT en inglés) es un término recurrente del cual se está siendo desarrollado para que cada dispositivo cotidiano sea conectado a internet y obtener información de estos dispositivos. Prueba de esto es que algunos dispositivos tales como TV, coches, casas inteligentes, entre otros, están siendo conectados a internet para facilitar la manera de interactuar con ellos.

El desarrollo de IoT a través de la innovación, debe generar la transformación en diferentes sectores como negocios, gobierno, salud y servicios, cambiarán la forma de experiencia del usuario (Lund, Turner, MacGillivray, & Morales, 2014)

Un reporte hecho por Verizon muestra que entre 2014 y 2015, diferentes sectores han apostado por desarrollar IoT para tener que incrementar la innovación y el consumo hacia sus clientes (**Figura 51**).

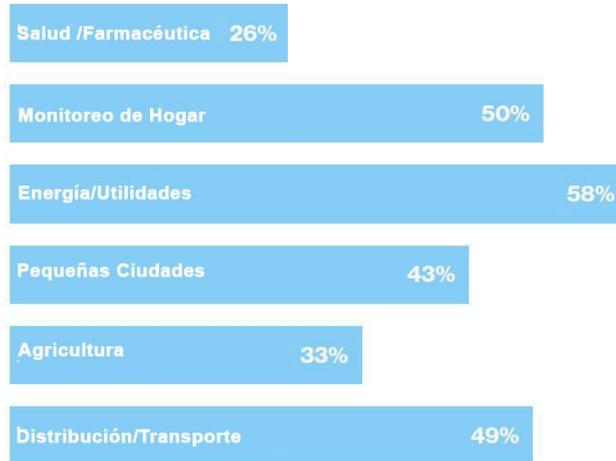


Figura 51 Sectores que Implementan IoT (Verizon, 2016)

Para el año 2020, se espera que existan más de 200 millones de dispositivos basados en IoT. Un artículo publicado por la unión europea (UE), visualiza 4 pilares fundamentales para el desarrollo de IoT hacia el futuro: estándares, conectividad, seguridad y protección y privacidad de datos personales (Davies, 2015).

CAPÍTULO 5 CASO PRACTICO

5.1 Introducción

A lo largo de este trabajo, se han plasmados una serie de temas que van acorde al proyecto a desarrollar, retomando algunos de ellos para su elaboración. En este capítulo se mostrará paso a paso el desarrollo del sistema de información docente.

Parte fundamental para el caso práctico del sistema, es la inclusión de una metodología de desarrollo. El modelo de desarrollo incremental, fue seleccionado para la realización del sistema. A través del modelo y las fases que lo integran (análisis de requerimientos, diseño, codificación, implementación), podremos desarrollar el sistema de tal manera que, los usuarios tengan acceso e interactúen con él.

Para el desarrollo del sistema, se han planteado 4 incrementos los cuales, serán descritos cada uno que completan un módulo funcional del sistema. Cada incremento está estructurado con base en sus fases.

El uso del modelo incremental va acompañado del paradigma orientado a objetos que contempla la programación orientada a objetos, análisis y diseño mediante el uso de Unified Model Language (UML), los cuales incluyen los siguientes diagramas:

- 1- Diagrama de casos de uso.
- 2- Diagrama de actividades.
- 3- Diagrama de clases.

Además, se presentará en cada incremento descrito la interfaz diseñada que corresponde a cada incremento funcional, para obtener de manera secuencial el desarrollo del sistema.

5.2 Incremento I Roles de Usuario y Captura de Catálogos

5.2.1 Análisis de Requerimientos

El CU UAEM Texcoco a través de la oficina de subdirección académica, requieren una aplicación web, es decir, un sistema en donde le permitan recopilar cierta información que solicitan a los maestros, con la finalidad de saber su plan de apoyo para cada semestre de la asignatura y licenciatura que le corresponde.

Como primer punto, es necesario que el sistema cuente con el acceso tanto de usuarios de la oficina de subdirección y anexas, como de profesores que respondan a cada petición, tomando cada uno diferentes roles y privilegios, divididos en dos tipos de usuarios: administradores y profesores.

Para el acceso al sistema, es necesario que cada tipo de usuario, tenga una cuenta de acceso y sea autenticado por motivos de seguridad, es decir, que exista un formulario para iniciar sesión y tener acceso al sistema. Los tipos de usuarios tendrán diferentes privilegios, por lo que cada uno tendrá diferentes opciones de visualización.

Además, en caso de que olviden su contraseña de acceso, el formulario de inicio de sesión debe contar con un link para recuperar su contraseña mediante el uso de su email previamente registrado en el sistema.

5.2.1.1 Roles y Acceso de Funciones del Administrador

Los usuarios de tipo administrador tendrán las siguientes acciones una vez dentro del sistema:

- 1- Se encargarán de hacer la carga de catálogos que se presenta en el sistema (Titulo 5.2.1.3).
- 2- Se encargarán de la administración de usuarios, es decir, tendrán la capacidad de añadir, modificar y eliminar usuarios administradores y usuarios profesores.
- 3- Serán capaces de crear y revisar las solicitudes de requerimientos hechas para los profesores (Titulo 5.4).

Las funciones del administrador, es decir, el menú de privilegios, debe de contar con las siguientes secciones (**Tabla 25**):

Tabla 25 Menú del Administrador

Sección	Descripción
Catálogos	Dentro de la sección de catálogos, se debe contar con las opciones para cargar cada uno de los catálogos de manera individual y posteriormente sean mostrados
Administración de Usuarios	Se debe contar con dos secciones, una para agregar un nuevo usuario, la cual nos permita seleccionar si es un administrador o profesor. La otra sección será para modificar o eliminar a los usuarios

Solicitud de Requerimientos	<p>Se debe contar con dos secciones; crear solicitud e historial de solicitudes.</p> <p>La primera sección debe de mostrar las 3 diferentes maneras de solicitudes: todas las licenciaturas, por licenciatura y por profesor.</p> <p>La sección de historial, debe mostrar todas las solicitudes hechas por el usuario</p>
Perfil	<p>La sección de perfil debe mostrar dos secciones. La primera será para modificar su contraseña. Mientras que la segunda opción, para cerrar sesión. Esta opción también estará disponible para los profesores</p>

5.2.1.2 Roles y Acceso de Funciones del Profesor

Los usuarios de tipo profesor, tendrán las siguientes acciones una vez dentro del sistema (Tabla 26):

- 1- Podrán actualizar su información personal por cualquier circunstancia.
- 2- Revisaran sus solicitudes de requerimientos que les fueron hechas por los administradores
- 3- Responderán esas solicitudes a través de subir un archivo en formato PDF para cada requerimiento.
- 4- El menú a desplegar del profesor consta de dos secciones, solicitudes de requerimientos donde solo podrá revisarlos y responderlos y la sección de perfil para cambiar su contraseña.

Tabla 26 Menú del Profesor

Sección	Descripción
Solicitud de Requerimientos	En esta parte, se muestra la sección de las solicitudes que fueron hechas hacia el profesor y responderlas.
Perfil	La sección de perfil debe mostrar dos secciones. La primera será para modificar su contraseña. Mientras que la segunda opción, para cerrar sesión. Esta opción también estará disponible para los profesores

5.2.1.3 Captura de Catálogos

Parte importante para llenar la base de datos y en general, observar la información para generar la documentación adecuada en el sistema, es necesario que los administradores hagan la carga de los catálogos de manera rápida. Para ello, el sistema contara con 3 catálogos los cuales serán obtenidos a través de un archivo en Excel (xlsx). Los 3 catálogos son:

1. Licenciaturas.
2. Profesores.
3. Requerimientos.

Para cada catálogo debe existir una función independiente que permita hacer la selección del archivo en Excel y después que se suban al sistema, serán mostrados como señal de que se llenó la base de datos de manera exitosa.

Para el catálogo de licenciaturas, se debe de mostrar las siglas y el nombre de la licenciatura. Para el catálogo de profesores, deberá contener el grado de estudios, nombre completo, email contraseña de acceso, licenciatura a la que pertenece y teléfono de contacto. Cabe mencionar que subir el catálogo de profesores, también genera su cuenta de usuario para el acceso al sistema. El catálogo de requerimientos, deberá contener el nombre y descripción del requerimiento.

5.2.2 Diseño

Para el diseño de este primer incremento se mostrará cada diagrama elaborado en UML y la interfaz gráfica del sistema.

5.2.2.1 Estructura De Los Archivos

El archivo de Excel para cada catálogo está estructurado de la siguiente manera (*Tabla 27, Tabla 28, Tabla 29*):

Tabla 27 Tabla Para Licenciaturas

Licenciaturas	
Siglas	Nombre de Licenciatura
ICO	Ingeniería en Computación

Tabla 28 Tabla para Profesores

Profesor					
Licenciatura	Nombre	Apellidos	Email	Contraseña	Teléfono
ICO	David	Sánchez	email@email.com	1234	5531182

Tabla 29 Tabla para Requerimientos

Requerimientos	
Nombre	Descripción
Bibliografía	Los profesores...

5.2.2.2 Diagramas De Incremento

5.2.2.2.1 Diagrama de Casos de Uso

El diagrama de casos de uso para el primer incremento, muestra la manera en como los usuarios administradores tendrán la capacidad de subir cada catálogo y ser mostrados, a partir del inicio de sesión del mismo. El usuario administrador inicia sesión y mediante la sección de catálogos cargara cada uno de ellos mediante la selección de un archivo de Excel. Cada archivo de Excel tendrá una estructura con base en la vista de las tablas de cada catálogo que mostrará la información una vez cargado el archivo (*Figura 52*).

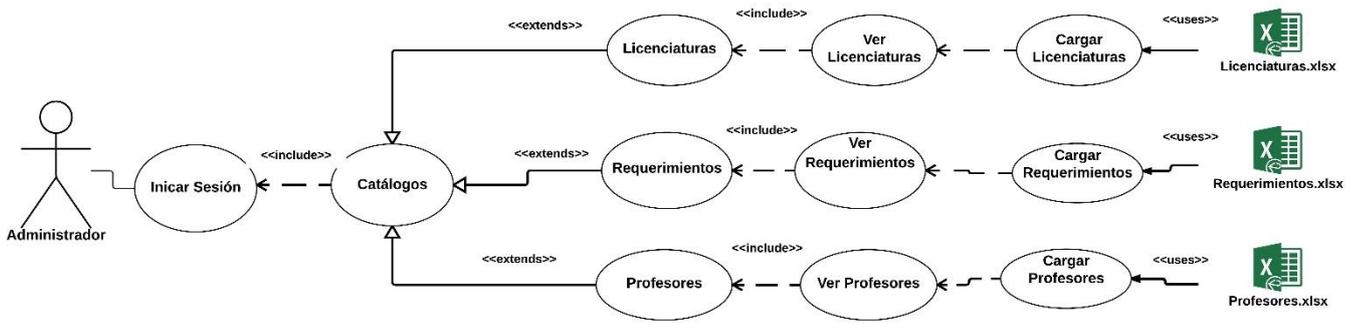


Figura 52 Diagrama de Casos de Uso (Incremento I Catálogos)

5.2.2.2.2 Diagrama de Actividades

El diagrama de actividades muestra el proceso en que cada catálogo es llenado y visualizado una vez extraída la información del archivo de Excel, terminando así el proceso del primer incremento (Figura 53).

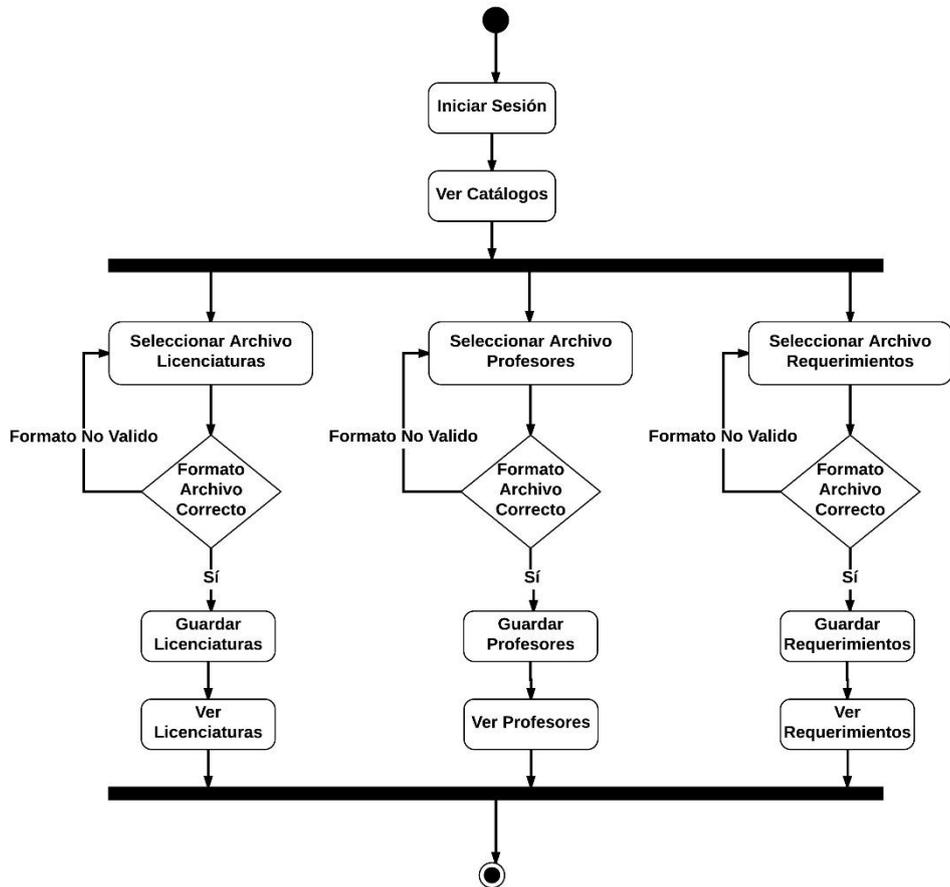


Figura 53 Diagrama de Actividades (Incremento I Catálogos)

5.2.2.2.3 Diagrama De Clases

El diagrama de clases se centra en los 3 catálogos principales: licenciaturas, requerimientos y profesores. Todos los catálogos cuentan con sus respectivas clases. Además, se crean

clases controladoras que sirven para dar funcionalidad a las clases de los catálogos respectivamente.

La clase PHPExcel_IOFactory, es una clase que nos dará la capacidad de poder manejar los datos de los archivos en Excel y obteniéndolos mediante las clases de los catálogos y almacenarlos en la base de datos (Figura 54).

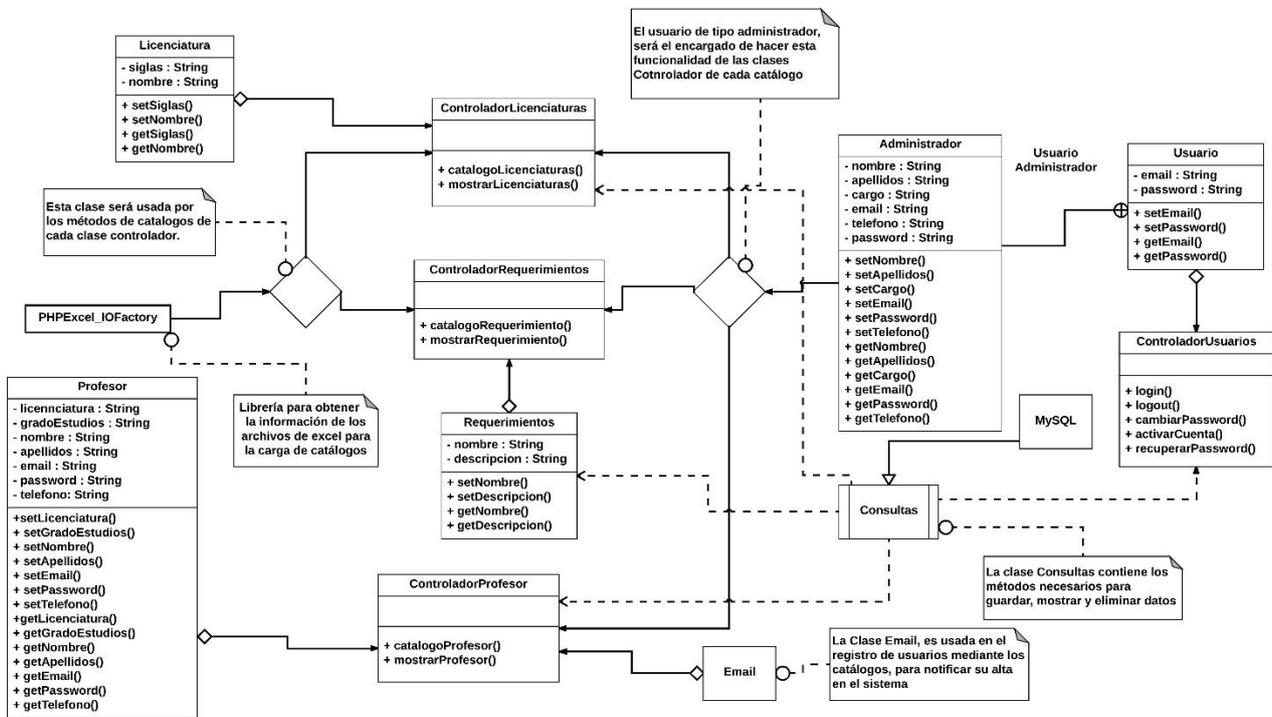


Figura 54 Diagrama de Clases (Incremento I Catálogos)

5.2.2.3 Interfaz De Usuario

5.2.2.3.1 Formulario de Acceso al Sistema

El formulario de acceso al sistema se conforma de dos campos y dos botones. Los campos sirven para capturar el email y la contraseña del usuario. Mientras que los botones, el botón azul permite el acceso al sistema y el otro botón, a un formulario para recuperar la contraseña (Figura 55).

5.2.2.3.2 Recuperar Contraseña

En el formulario de inicio de sesión, existe la opción de recuperar la contraseña del usuario ingresando únicamente su email. Se le notificara vía email, los pasos y el link para reestablecer su contraseña e ingresar nuevamente al sistema, ya con la nueva contraseña (Figura 56).



Figura 55 Formulario de Inicio de Sesión

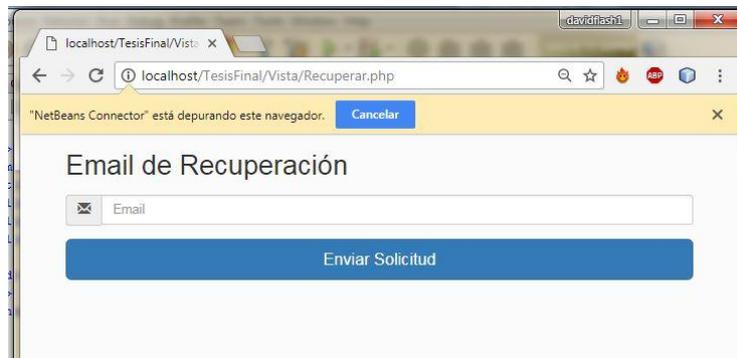


Figura 56 Formulario Recuperar Contraseña

5.2.2.3.3 Interfaz de Inicio de Sesión

Una vez que se ingrese al sistema, el administrador observara el menú que le corresponde, es decir, las funciones que podrá desempeñar en el sistema, el cual tienen las siguientes secciones (*Tabla 30, Figura 57*).

Tabla 30 Estructura del Menú y Submenú (Administrador)

Sección	Subsección
Inicio	N/A
Catálogos	<ul style="list-style-type: none"> • Cargar licenciaturas. • Cargar profesores. • Cargar requerimientos.
Administración de Usuarios	<ul style="list-style-type: none"> • Nuevo usuario <ul style="list-style-type: none"> ○ Registrar administrador. ○ Registrar profesor. • Lista de usuario.
Solicitud de Requerimientos	<ul style="list-style-type: none"> • Nueva solicitud. <ul style="list-style-type: none"> ○ Solicitud a todas las licenciaturas. ○ Solicitud por licenciatura. ○ Solicitud por profesor. • Historial de solicitudes.
Perfil	<ul style="list-style-type: none"> • Modificar perfil. • Cerrar sesión.

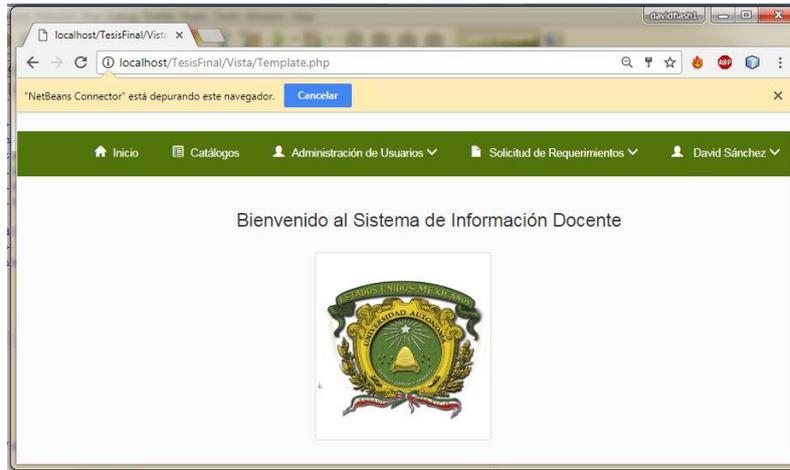


Figura 57 Ingreso al Sistema (Administrador)

5.2.2.3.4 Catálogos

La sección de catálogos, muestra las 3 secciones que corresponden cada uno para ser llenados. También, una vez cargado el catálogo, se muestra la tabla con los datos previamente guardados en cada catálogo, mostrándose de manera individual (Figura 58, Figura 59, Figura 60).

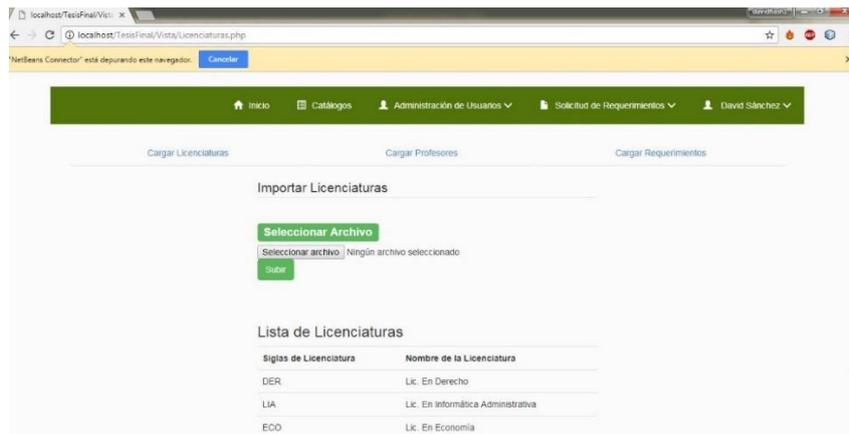


Figura 58 Vista de los Catálogos (Sección Licenciaturas)

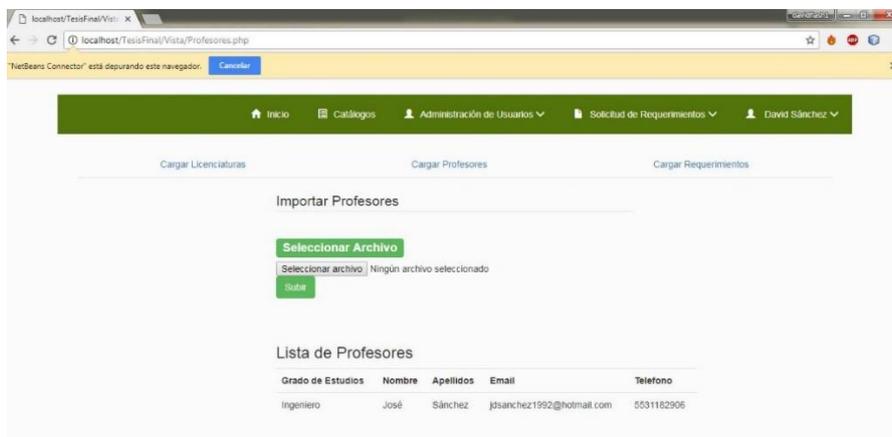


Figura 59 Vista de los Catálogos (Sección Profesores)

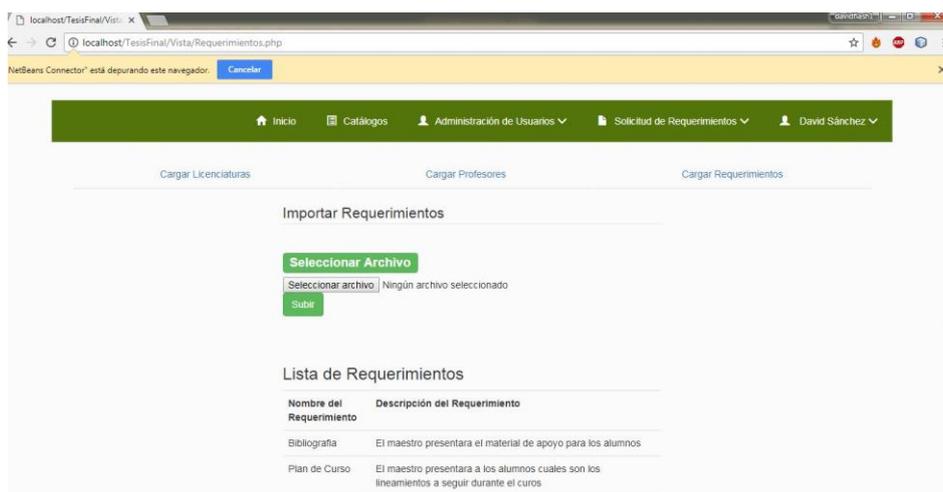


Figura 60 Vista de los Catálogos (Sección Requerimientos)

5.2.2.3.5 Administración de Usuarios

La sección de administración de usuarios, cuenta con dos subsecciones, las cuales son dar de alta un nuevo usuario y modificar usuarios (*Figura 61*).

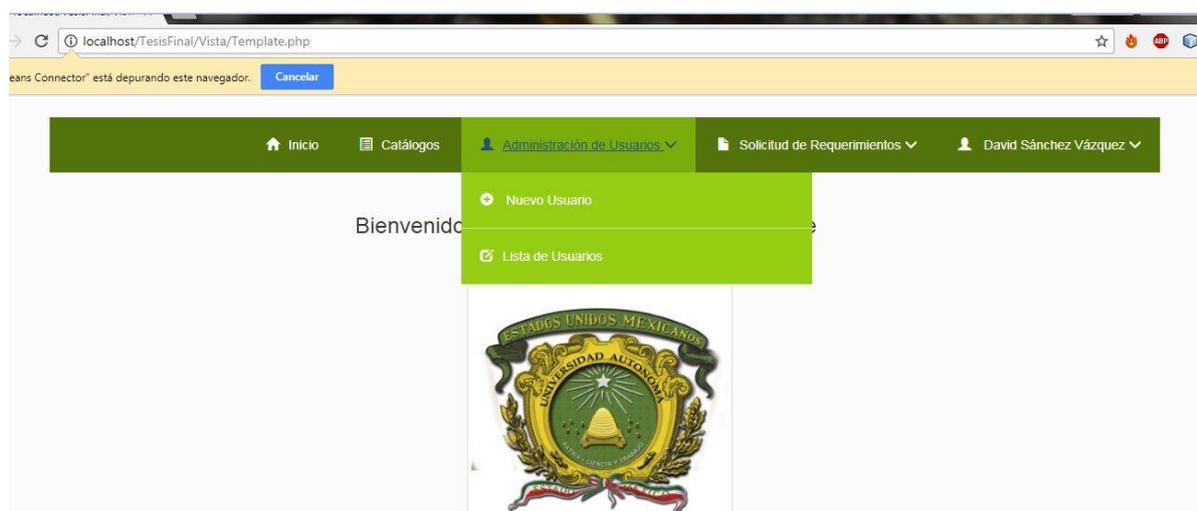


Figura 61 Subsecciones de Administración de Usuarios

Al ingresar a la sección de nuevos usuarios, nos dirige a los formularios para registrar a los dos tipos de usuarios que interactúan en el sistema: formulario de usuario administrador y formulario de profesor.

5.2.2.3.6 Solicitud de Requerimientos

La sección de solicitudes de requerimientos nos permite hacer dos cosas (*Figura 62*):

- 1- Crear solicitudes de requerimientos, los cuales nos dirige a la pantalla para realizar las solicitudes de las 3 maneras posibles: todas las licenciaturas, por licenciatura, por profesor.

- 2- El historial de solicitudes nos brindara la posibilidad de revisar las solicitudes de requerimientos hechas a cada profesor, además de poder visualizar la respuesta del profesor.



Figura 62 Subsecciones de Solicitudes de Requerimientos

5.2.2.3.7 Perfil

La sección de perfil, nos dirige a dos opciones; modificar perfil, para actualizar nuestra información y la opción para cerrar sesión del sistema y nos re-direcciona al formulario de inicio de sesión (Figura 63).



Figura 63 Subsecciones de Perfil

5.2.2.3.7.1 Modificar Perfil

Muestra el formulario para cambiar la contraseña de acceso al sistema (Figura 64).

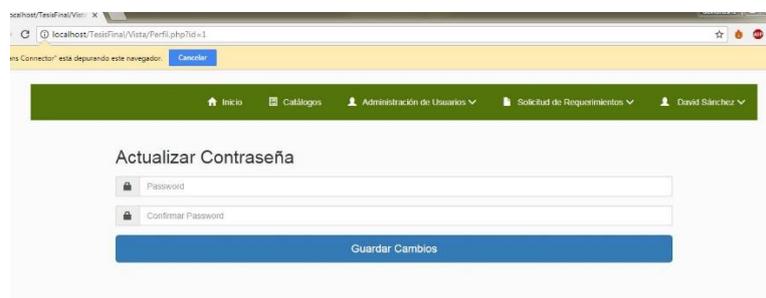


Figura 64 Formulario de Cambio de Contraseña

5.2.3 Codificación

Ir a capítulo 7 Anexo

5.2.4 Implementación

5.2.4.1 Inicio De Sesión Y Recuperar Contraseña

Como se mencionó anteriormente (Títulos 5.2.2.3.1, 5.2.2.3.2), el ingresar su email y contraseña permite el acceso al sistema, dependiendo del tipo de usuario, será su nivel de privilegios (*Figura 65*).

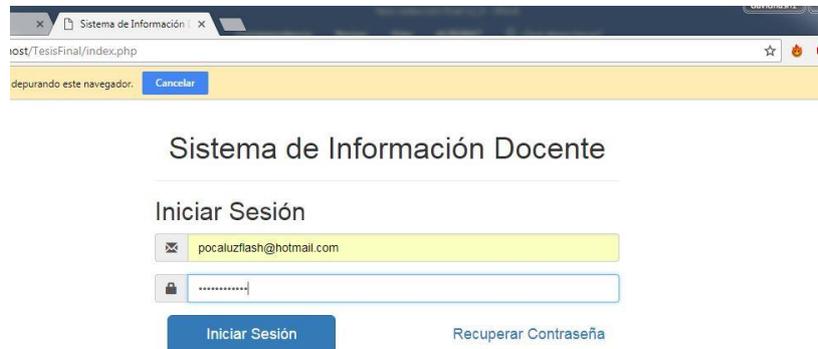


Figura 65 Datos de Acceso al Sistema

Cuando un usuario es registrado, se envía un email para activar su cuenta, de lo contrario, no tendrá acceso al sistema. Es importante mencionar que el inicio de sesión cuenta con 3 validaciones por motivos de seguridad:

- 1- Si es nuevo el usuario, tendrá que activar su cuenta, de lo contrario aparecerá el mensaje que aún no se activa su cuenta y no tiene acceso al sistema.
- 2- Si se ingresa un usuario y contraseña que no está registrado en el sistema, le mostrara un mensaje indicando que el usuario no existe.
- 3- Si el usuario se equivoca en algún dato, ya sea email o contraseña, el sistema le mostrara dicho mensaje.

Cuando el usuario no recuerda su contraseña, este tendrá que dar clic en recuperar contraseña y lo dirige al formulario donde escribe únicamente su correo y se confirmara si se envía o existe error (*Figura 66*).



Figura 66 Correo de Recuperación de Contraseña

Una vez recibido el correo, se muestra un link el cual tiene que dirigirse para agregar la nueva contraseña. Cuando se agrega aparecerá el mensaje que fue exitoso el cambio (Figura 67).



Figura 67 Formulario para Cambiar Contraseña

En caso de que se cambie un dato en el link para recuperar la contraseña, muestra un error describiendo que los datos son erróneos (Figura 68).

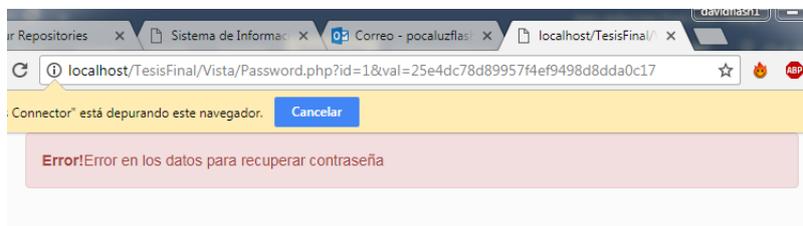


Figura 68 Mensaje de Error de Recuperación de Contraseña

5.2.4.2 Cargar Catálogos

Cuando el administrador inicia sesión, se recomienda cambiar su contraseña de acceso para evitar posibles alteraciones en su cuenta.

Cuando se requiere hacer la carga de catálogos se muestra a continuación, el proceso de selección de un archivo, haciendo referencia a la plantilla previamente descrita en el análisis de requerimientos. Cada carga de catálogo funciona de la misma manera, es decir, se obtienen los datos y a su vez, son mostrados de manera instantánea (Figura 69, Figura 70, Figura 71).

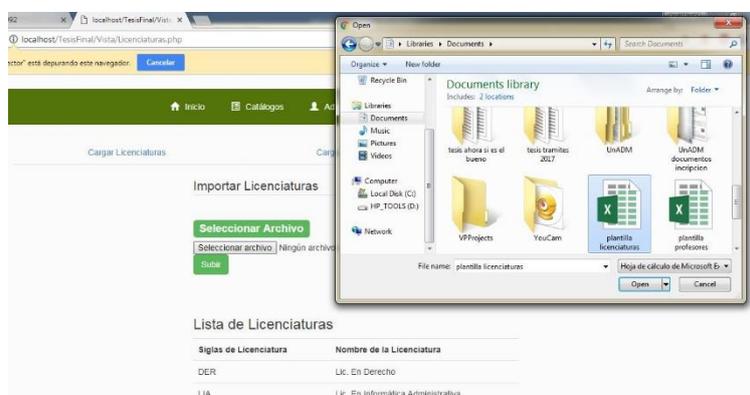


Figura 69 Implementación Para Cargar Catálogo de Licenciaturas

Aplicación del Modelo Incremental Para el Desarrollo del Sistema de Información Docente

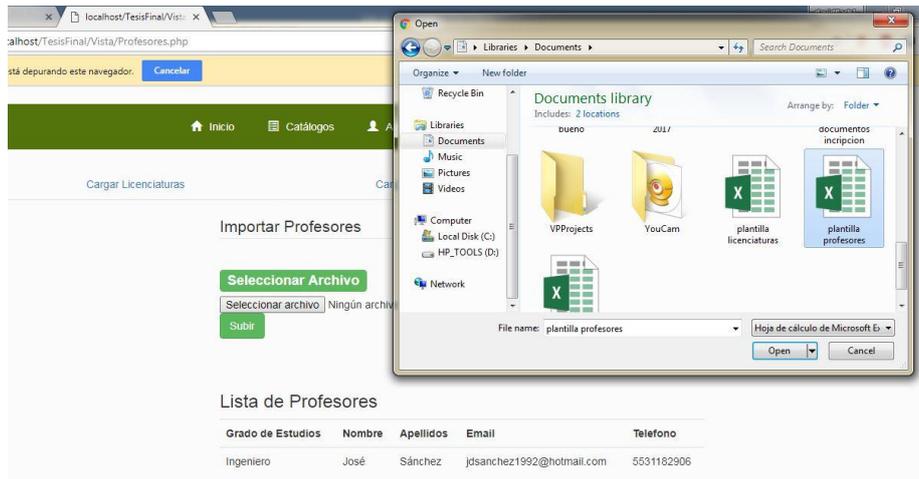


Figura 70 Implementación Para Cargar Catálogo de Profesores

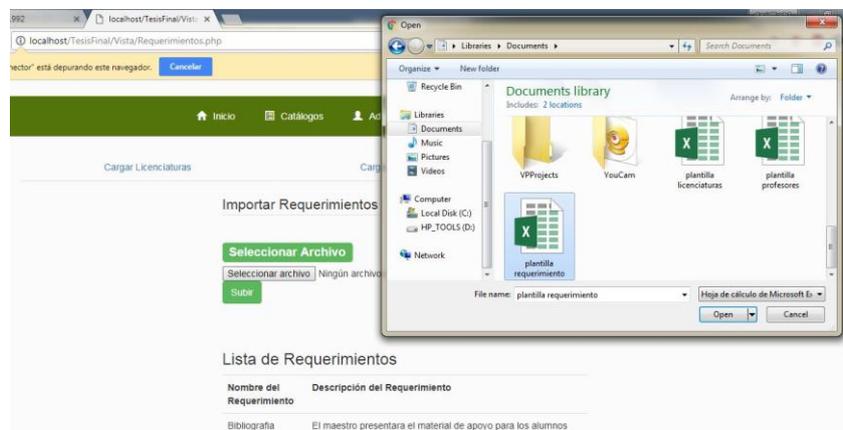


Figura 71 Implementación Para Cargar Catálogo de Requerimientos

5.3 Incremento II Administración de Usuarios

5.3.1 Análisis de Requerimientos

El segundo incremento para el sistema, se centrará en la funcionalidad de poder administrar a los usuarios que sean dados de alta en el sistema. Esto abarca a profesores y administradores.

La administración de los usuarios debe de contener 4 funcionalidades básicas, las cuales son:

- ✓ Crear.
- ✓ Mostrar/Leer.
- ✓ Actualizar.
- ✓ Eliminar.

La administración de usuarios debe tener la capacidad para crear, es decir, registrar nuevos usuarios de ambos tipos. Al ser registrados, deben de ser mostrados una vez registrados. Si hay que corregir cierta información, debe tener la posibilidad de actualizar su información y al final, la capacidad de dar de baja a un usuario mediante su eliminación de acceso al sistema. Este término suele denominarse en inglés como CRUD (create, read, update, delete). Cuando un usuario es registrado, se le envía un email con sus datos de acceso.

5.3.1.1 Formularios de Registro

Debido a la existencia de dos tipos de usuarios, el registro de ambos debe de ser de manera independiente, es decir, que cada uno tenga su propio formulario. Para el formulario de los profesores, debe de estar conformado de la siguiente manera (*Tabla 31*):

Tabla 31 Estructura de los Formularios de Usuarios

Formulario de Administrador	Formulario de Profesor
✓ Nombre completo.	✓ Licenciatura.
✓ Apellidos.	✓ Grado de estudios.
✓ Puesto de trabajo.	✓ Nombre completo.
✓ Email	✓ Apellidos.
✓ Teléfono.	✓ Email
✓ Contraseña	✓ Teléfono.
	✓ Contraseña

Se añade un campo más para confirmar que la contraseña y validar si coincide con el campo anterior.

5.3.1.2 Lista De Usuarios

La subsección lista de usuarios, muestra todos los usuarios registrados en el sistema. Los administradores tendrán el privilegio de eliminar al usuario. Cuando se añaden profesores mediante archivos de Excel, serán mostrados en la lista, además, si se elimina un profesor, también se eliminará del catálogo general de profesores.

5.3.1.3 Activación de Cuenta para Acceso al Sistema

Cuando un usuario ya sea administrador o profesor son registrados, se les envía una notificación vía email la cual consiste en que requiere activar su cuenta de acceso al sistema, mientras no lo hagan, no podrán ingresar, además de validaciones importantes tales como, si existe o no el usuario, error en el email o contraseña y la activación. Esta función también aplica cuando los profesores son registrados mediante la sección de catálogos de profesor.

5.3.2 Diseño

5.3.2.1 Diagramas De Incremento

5.3.2.1.1 Diagrama De Casos De Uso

El diagrama de casos de uso, muestra la funcionalidad para registrar usuarios administradores y profesores, siendo los administradores, los que poseen las 4 funciones requeridas planteadas en el análisis de requerimientos (CRUD). También, muestra la manera en cómo se ingresarán los datos y la manera en que se enviara el email, una vez que sean dados de alta.

Los usuarios serán visualizados en sus respectivas secciones y en el caso del profesor, también será dado de alta en el catálogo con todos sus datos (Figura 72).

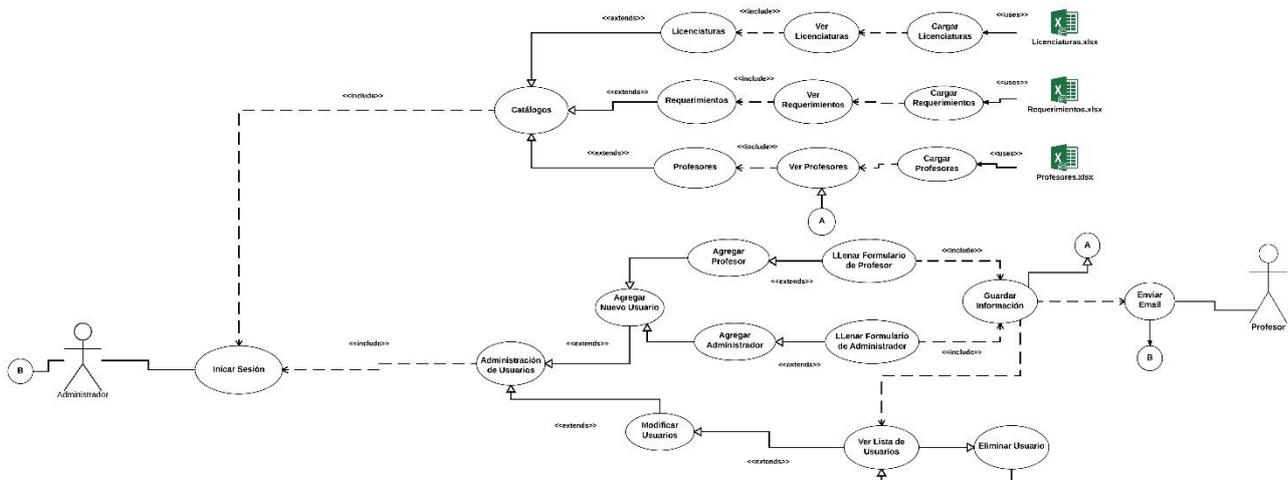


Figura 72 Diagrama de Casos de Uso (Incremento II Administración de Usuarios)

5.3.2.1.2 Diagrama de Actividades

El diagrama de actividades muestra el proceso de cómo se usa la funcionalidad de administración de usuarios, observando las funciones previamente señaladas: (crear, leer/mostrar, actualizar, eliminar) (Figura 73).

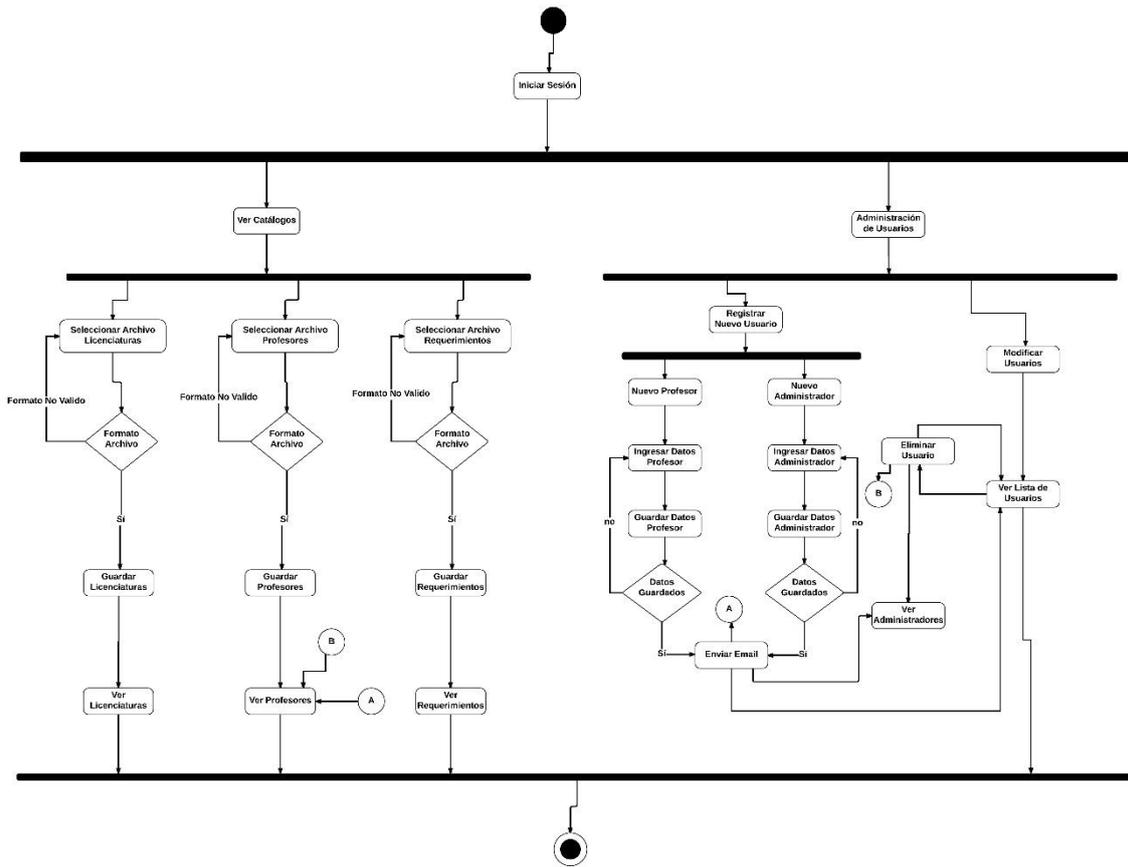


Figura 73 Diagrama de Actividades de Administración de Usuarios (Incremento II Administración de Usuarios)

5.3.2.1.3 Diagrama de Clases

En este segundo incremento, el diagrama de clases resalta el anexar dos clases para la administración de los usuarios. La clase ControladorAdministrador, permite a un usuario de tipo administrador crear y mostrar usuarios de tipo administrador. También, el mismo administrador usa la clase de ControladorProfesor para registrar un profesor mediante el uso del método registrarProfesor. Este método se usa cuando se registra un profesor a través del formulario dentro de la opción de administración de usuarios.

La clase ControladorAdministracionUsuarios, es la clase que permite mostrar a los usuarios que están registrados en el sistema, sin importar el tipo de usuario que sea, permitiendo también que se pueda eliminar el mismo (Figura 74; **Error! No se encuentra el origen de la referencia.**).

5.3.2.2 Interfaz De Usuario

Dentro de la opción de administración de usuarios del menú del administrador, cuenta con dos opciones de navegación, agregar nuevo usuario y ver la lista de usuarios (Figura 75).

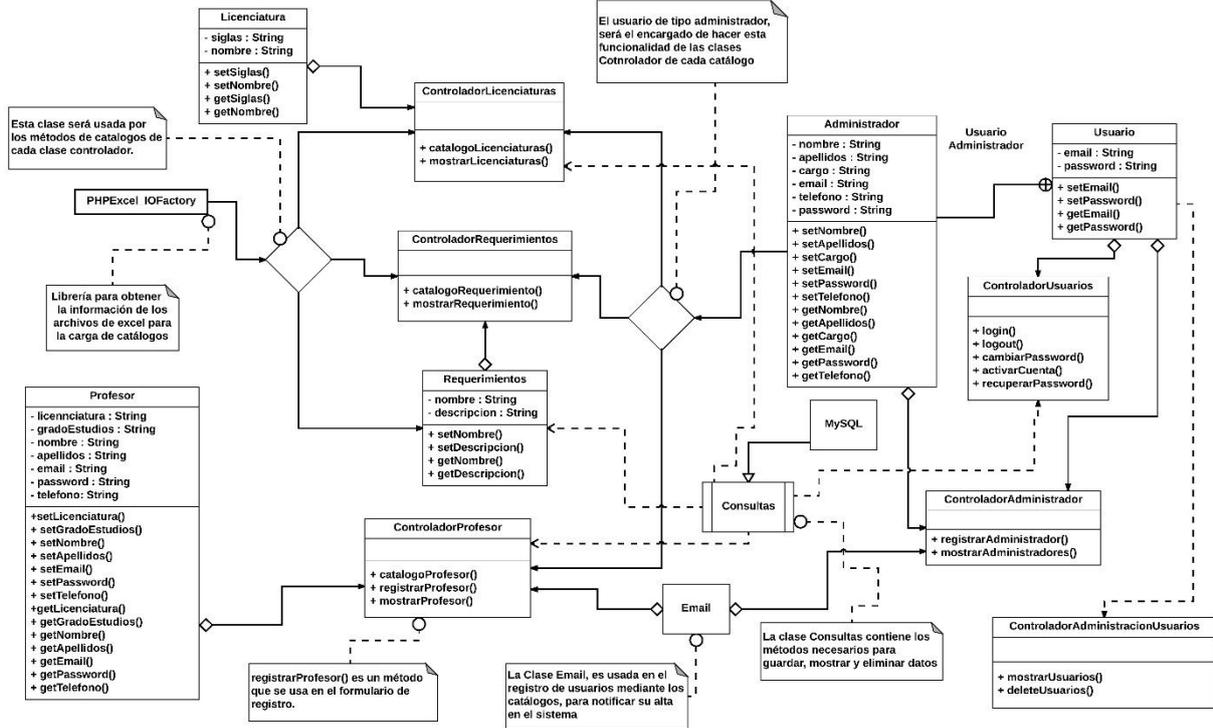


Figura 74 Diagrama de Clases (Incremento II Administración de Usuarios)

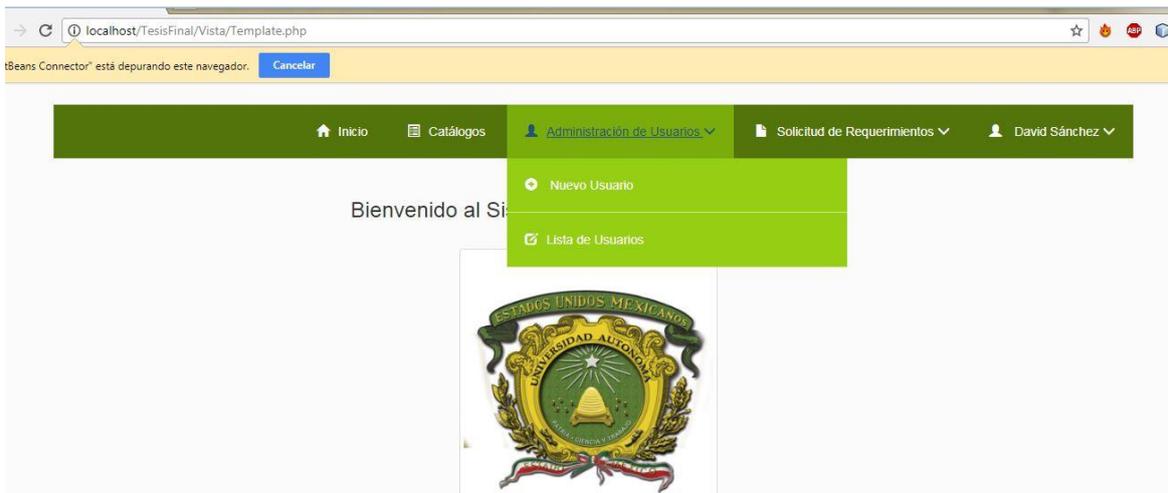


Figura 75 Subsecciones de Administración de Usuarios

5.3.2.2.1 Formularios de Registro de Usuarios

Cuando se quiere agregar un nuevo usuario muestra las opciones para ingresar a los formularios de un usuario administrador y profesor, respectivamente. Cuando un usuario es registrado, aparece debajo del formulario (Figura 76, Figura 77).

5.3.2.2.2 Lista de Usuarios

De manera general, al navegar a la lista de usuarios, muestra a los usuarios y su tipo, es decir, si son administradores o profesores. En esta sección solo se podrá eliminar a los usuarios. Cuando un profesor es eliminado, también se elimina del catálogo de los profesores y no podrá ingresar o recibir información proveniente del sistema ().

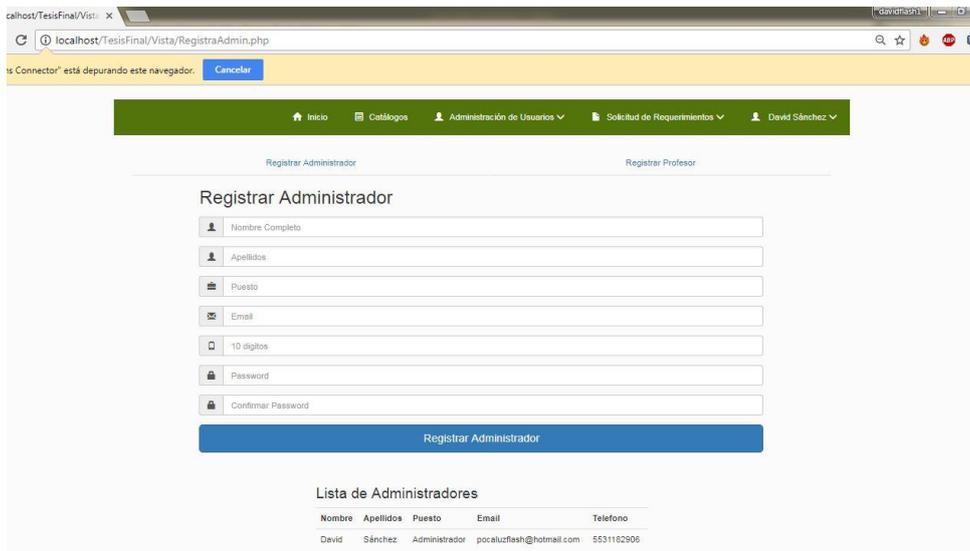


Figura 76 Formulario de Administrador

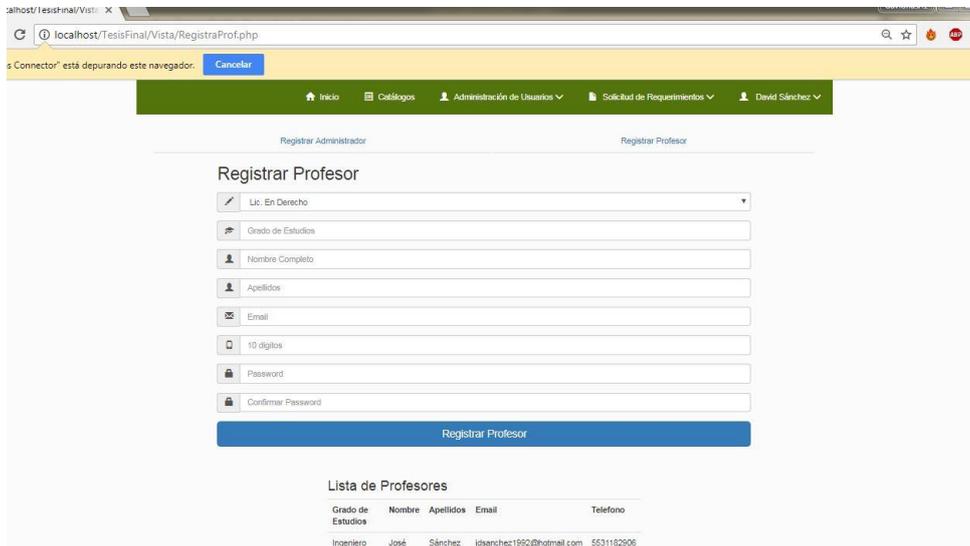


Figura 77 Formulario de Profesor

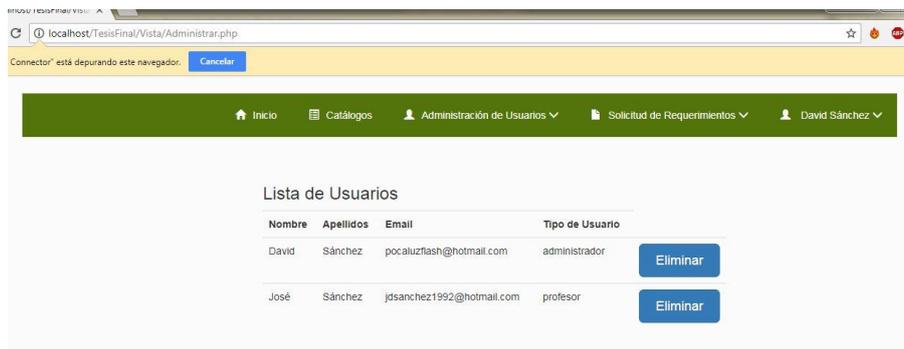


Figura 78 Lista de Usuarios

5.3.3 Codificación

Ir al capítulo 7 Anexo

5.3.4 Implementación

5.3.4.1 Registro De Administrador

Únicamente un usuario de tipo administrador podrá registrar a otro administrador. Cuando el administrador es registrado, se mostrara en la lista de administradores y de todos los usuarios (*Figura 79, Figura 80*).

Figura 79 Registro de Administrador

Lista de Administradores				
Nombre	Apellidos	Puesto	Email	Telefono
David	Sánchez	Administrador	pocaluzflash@hotmail.com	5531182906
Juan	Perez	Admin	administrador@gustr.com	55555555

Figura 80 Lista de Administradores

Cuando el usuario es registrado, recibirá su email para que pueda activar su cuenta, de lo contrario no tendrá acceso al sistema (*Figura 81, Figura 82*).

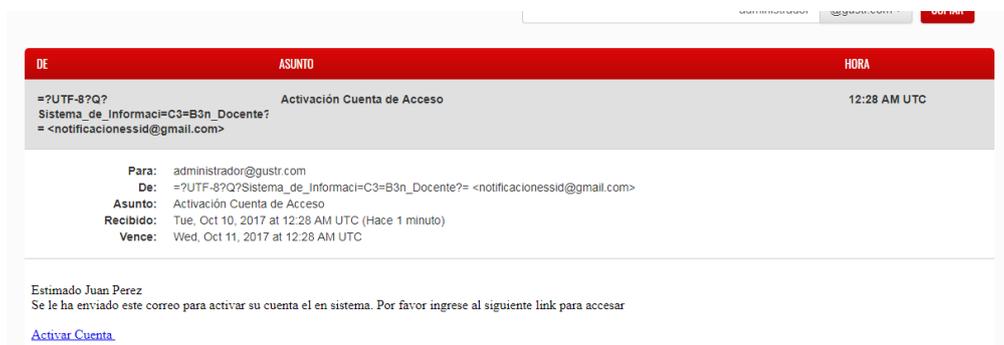


Figura 81 Email de Activación de Usuario

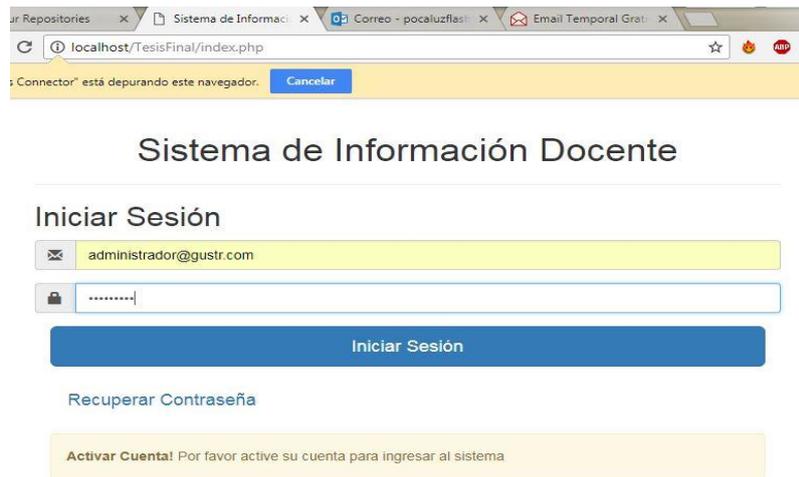


Figura 82 Mensaje de Activación de Cuenta

Cuando el usuario activa su cuenta, tendrá acceso al sistema (Figura 83). Sin embargo, si el usuario quisiera volver a activar su cuenta, mostrará un mensaje notificando que ya se activó su cuenta, por lo que tendrá acceso al sistema (Figura 83, Figura 84).

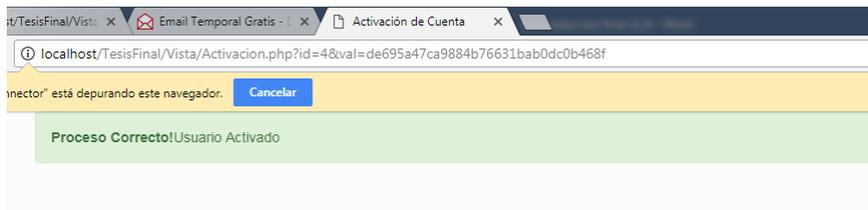


Figura 83 Activación de Cuenta

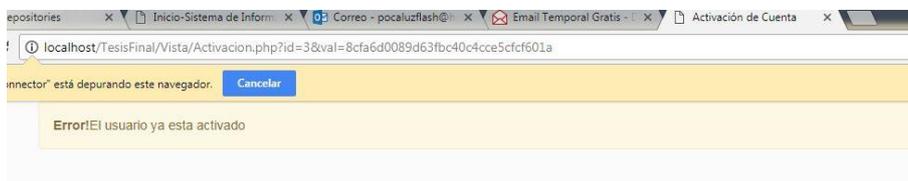


Figura 84 Mensaje de Cuenta ya Activada

Este mismo procedimiento aplica también para los profesores (Figura 85).

5.3.4.2 Lista De Usuarios

Al registrar los usuarios, pasan a esta lista donde se muestra sus datos de acceso y el tipo de usuario que es. Cuando un usuario es eliminado, este ya no será visible tanto en la lista de usuarios, como en la lista al tipo de usuario que corresponde. En este caso, se eliminó el usuario prueba profesor (Figura 86, Figura 87).

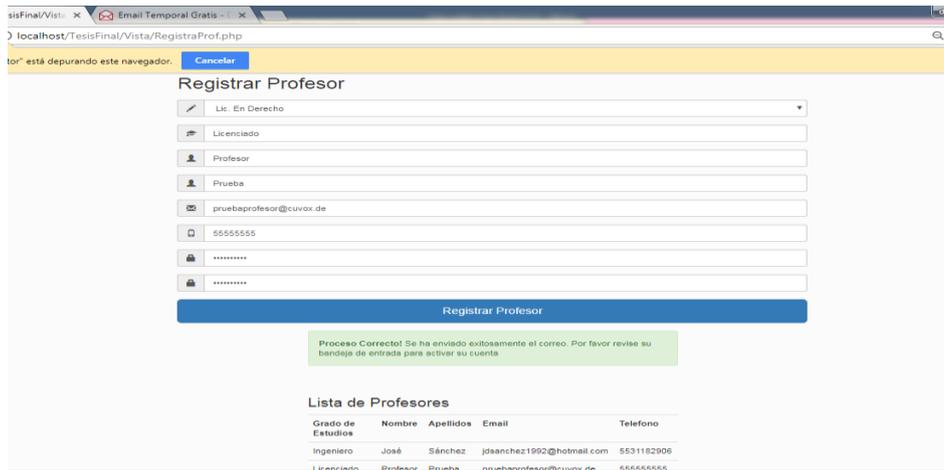


Figura 85 Registro de Profesor

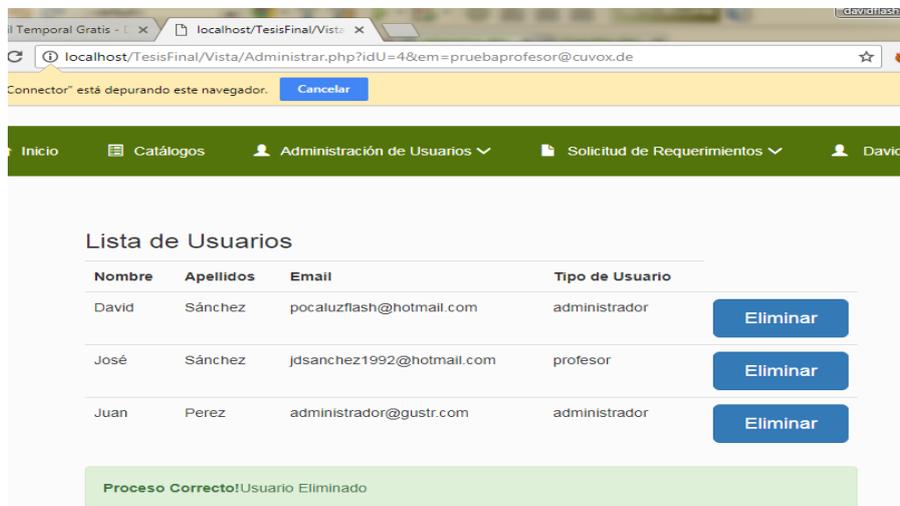


Figura 86 Mensaje de Eliminación de Usuario

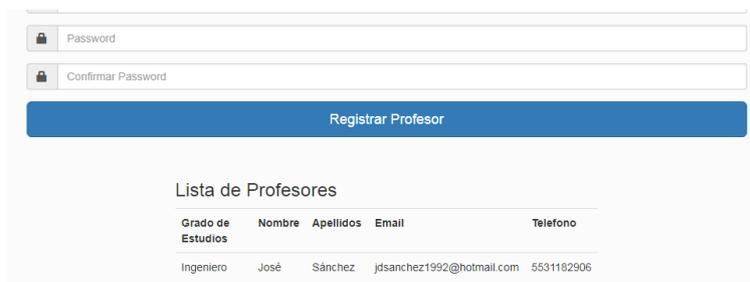


Figura 87 Lista de Profesores Actualizados

5.3.4.3 Cambiar Contraseña De Acceso

En la sección de perfil (donde aparece el nombre del usuario) podrá modificar únicamente su contraseña de acceso en modificar perfil, mostrando el mensaje en caso de exitoso o de error el proceso de cambio (Figura 88).

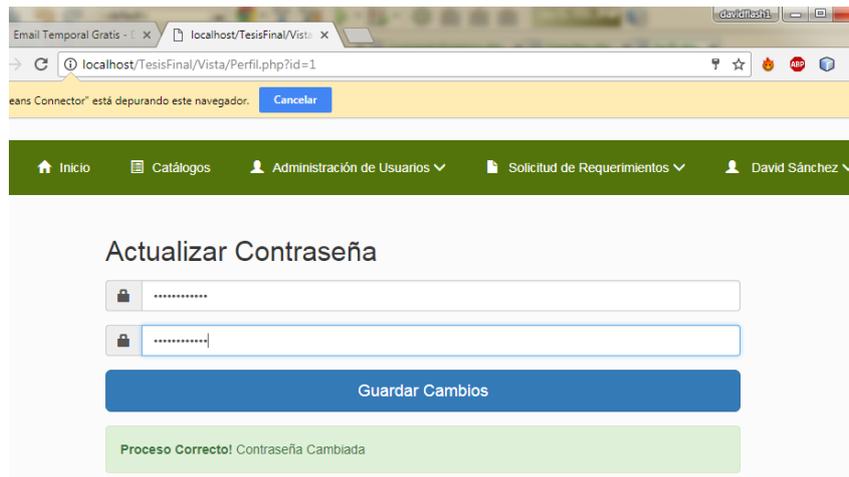


Figura 88 Mensaje de Cambio de Contraseña

5.4 Incremento III Solicitud de Requerimientos

5.4.1 Análisis de Requerimientos

Para este tercer incremento, se requiere desarrollar la funcionalidad de crear peticiones o solicitudes de requerimientos a los profesores de 3 maneras posibles:

- Solicitud para todas las licenciaturas.
- Solicitud por licenciatura.
- Solicitud por profesor.

Este tipo de solicitudes que se pretenden desarrollar hace que se facilite y optimice el proceso de crear las solicitudes, debido a la gran cantidad de profesores que hay y esto provocaría un retraso o mal manejo de las solicitudes.

5.4.1.1 Formulación De Requerimientos

- a) La primera opción para la crear solicitudes será a partir de todas las licenciaturas, es decir, únicamente habrá dos opciones para la creación de solicitudes, seleccionar todas las licenciaturas y posteriormente, seleccionar el requisito. El mismo requisito será enviado a todas las licenciaturas de los profesores que la componen.
- b) La segunda opción será mediante la selección de una licenciatura en específico y posteriormente seleccionar el requisito a solicitar, para que sea enviado a los profesores.
- c) La tercera opción será a través de la visualización de todos los profesores de cada licenciatura seleccionada y solo se seleccionará a una serie de maestros para enviarles la solicitud, escogiendo el requerimiento deseado.

Toda solicitud creada, será enviada vía email a los profesores para ser notificados. Una vez creada esta solicitud será mostrada en forma de tabla en la sección de historial de solicitudes, resaltando los siguientes campos:

- Fecha de creación.
- Nombre del requerimiento.
- Emisor.
- Receptor.
- Fecha de respuesta.
- Nombre del archivo (link de descarga).

Cuando se crea la solicitud de requerimiento y es mostrada en el historial, los campos de fecha de respuesta y nombre del archivo, permanecerán vacíos mientras el profesor lo responde. Al ser respondido en automático se actualizará la información.

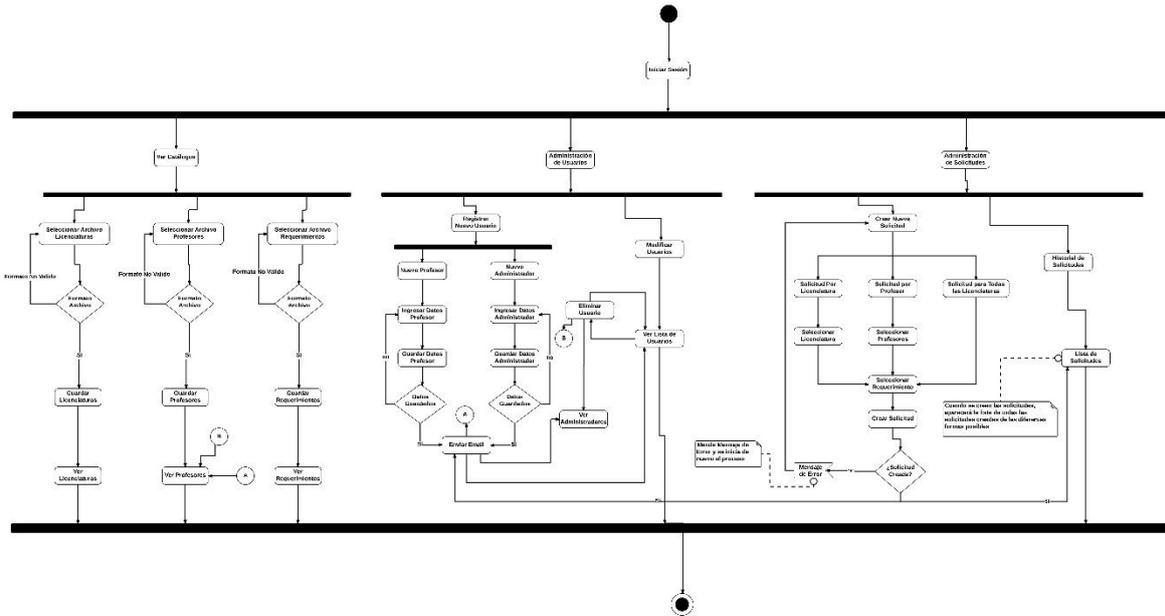


Figura 90 Diagrama de Actividades (Incremento III Crear Solicitud de Requerimiento)

5.4.2.1.3 Diagrama de Clases

En el diagrama de clases de este incremento, se añade la clase Solicitudes y ControladorSolicitudes. La clase Solicitudes contiene los datos que se requieren para crear la solicitud. Mientras que la clase ControladorSolicitudes, es la clase que contiene la funcionalidad para hacer las solicitudes de las 3 maneras que se requieren, además de que serán mostradas respectivamente cada requerimiento, dependiendo de la persona que las creo (emisor) y de las personas que recibió la solicitud (receptor) (Figura 91) .

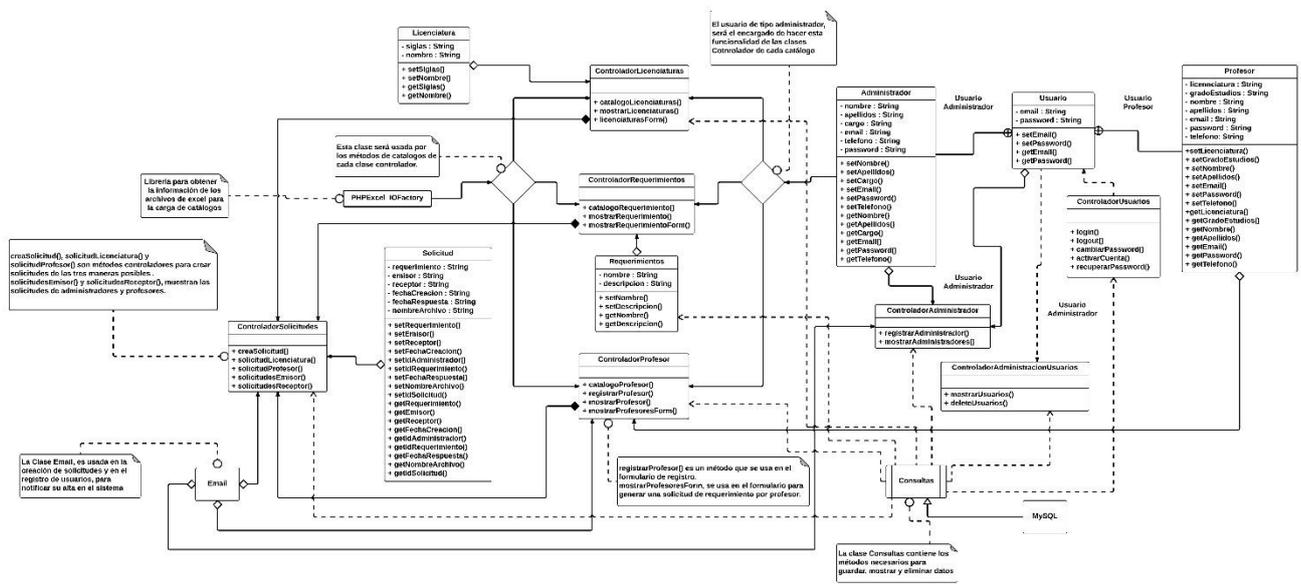


Figura 91 Diagrama de Clases (Incremento III Crear Solicitud de Requerimientos)

5.4.2.2 Interfaz de Usuario

La sección de solicitud de requerimientos muestra dos opciones: crear nueva solicitud e historial de solicitudes. La primera permite ingresar a los formularios para crear una solicitud. Es necesario mencionar que se muestran 3 pestañas que representan la manera en cómo se crea dicha solicitud (*Figura 92*).

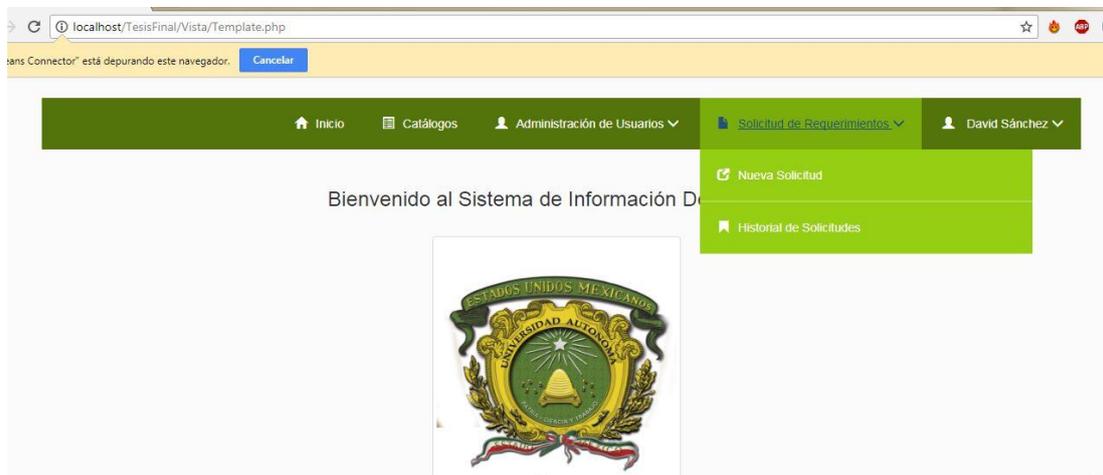


Figura 92 Subsecciones Para Solicitudes de Requerimiento

5.4.2.2.1 Solicitud para todas las Licenciaturas

La solicitud para todas las licenciaturas consta de solo seleccionar el requerimiento y será enviada a todos los profesores de todas las licenciaturas (*Figura 93*).



Figura 93 Interfaz de Solicitud Todas las Licenciaturas

5.4.2.2.2 Solicitud por Licenciatura

La solicitud por licenciatura, funciona a partir de la selección del requerimiento y la licenciatura, la cual será enviada a los profesores que pertenezcan a dicha licenciatura (*Figura 94*).

5.4.2.2.3 Solicitud por Profesor

La solicitud por profesor, funciona mediante la selección del requerimiento y muestra la lista de profesores en general, sin importar la licenciatura a la que pertenece. (*Figura 95*).

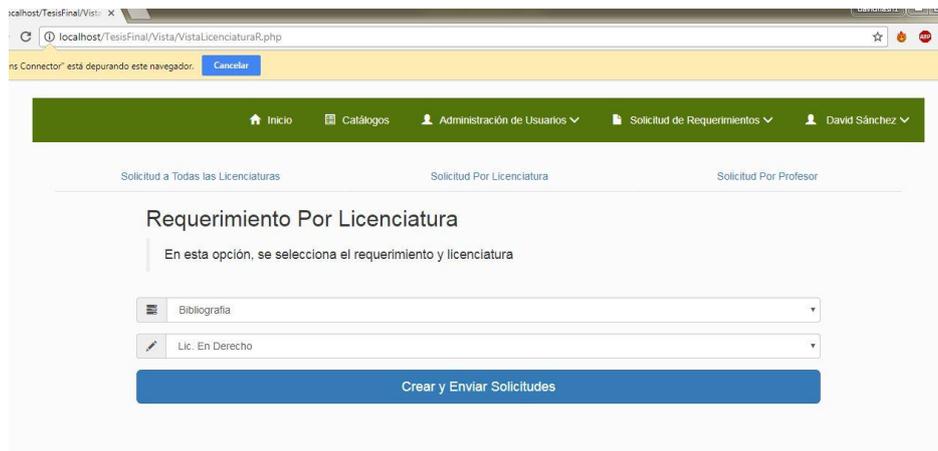


Figura 94 Interfaz Solicitud por Licenciatura



Figura 95 Interfaz Solicitud Por Profesor

5.4.2.2.4 Historial de Solicitudes

Sin importar la forma en cómo se crea la solicitud, serán mostradas todas las solicitudes con los campos previamente establecidos en el análisis de requerimientos. Mientras el profesor no responda el requerimiento, mostrara el mensaje de “sin respuesta” (*Figura 96*).

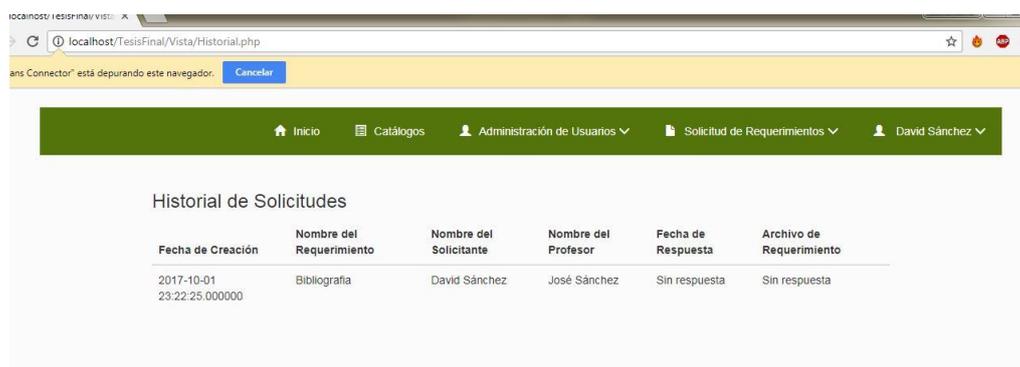


Figura 96 Interfaz Historial de Solicitudes

5.4.3 Codificación

Ir al capítulo 7 Anexo

5.4.4 Implementación

5.4.4.1 Crear Solicitudes Para Todas las Licenciaturas

En esta primera opción para crear una solicitud de requerimiento únicamente se seleccionará el requerimiento. Cuando se selecciona el requerimiento y se envía la solicitud, los profesores de todas las licenciaturas, reciben un email describiendo la solicitud de requerimiento que debe de atender (*Figura 97*).

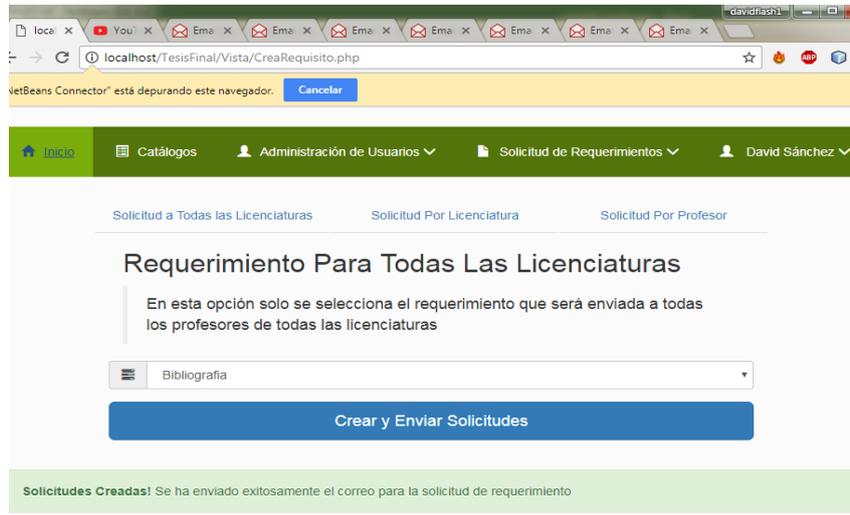


Figura 97 Crear Solicitud De Requerimiento Para Todas Las Licenciaturas

5.4.4.2 Crear Solicitud Por Licenciatura

En esta segunda opción para crear solicitudes de requerimientos, se debe seleccionar el requerimiento y la licenciatura. Cuando son seleccionadas ambas partes, el profesor que pertenece a la licenciatura, recibirá el email de notificación describiendo la solicitud de requerimiento que debe atender (*Figura 98*).

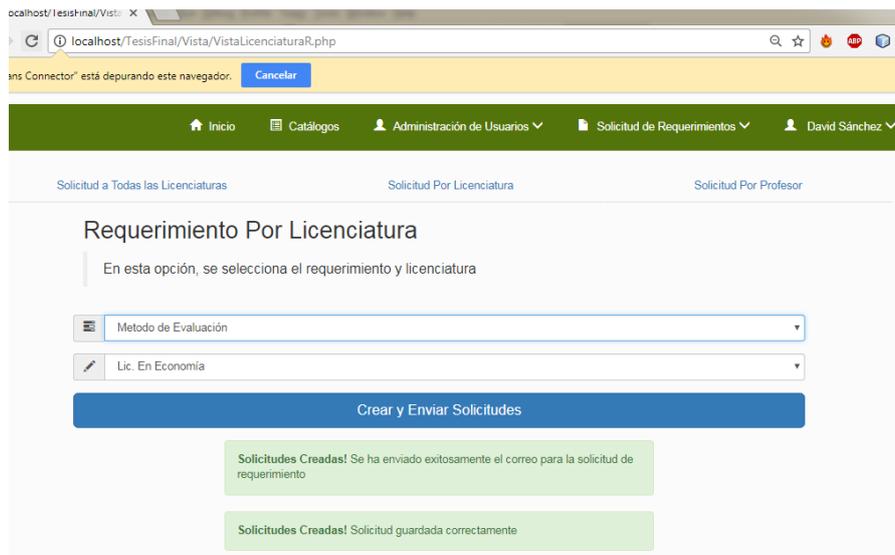


Figura 98 Creación de Solicitud de Requerimiento por Licenciatura (Ejemplo: Lic. en Economía)

5.4.4.3 Crear Solicitudes Por Profesor

En esta tercera opción para crear solicitud de requerimientos, seleccionamos el requerimiento y se despliega todos los profesores, sin importar la licenciatura a la cual pertenece. Es importante mencionar que se puede escoger los profesores que sean necesarios para enviar la solicitud. De igual manera, cuando están seleccionados el requerimiento y los profesores, se enviará una notificación a cada uno de ellos vía email sobre la solicitud de requerimiento a atender (*Figura 99*).

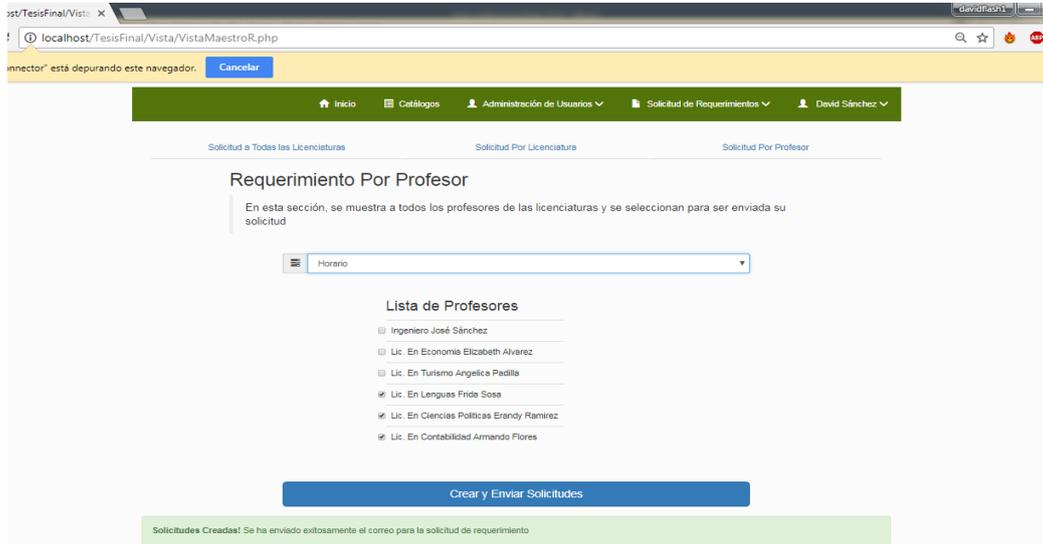


Figura 99 Creación de Solicitud por Profesor (Selección de Profesores)

Sin importar la manera en cómo se crean las solicitudes, los profesores recibirán un correo de notificación sobre el requerimiento que necesitan atender (*Figura 100*).

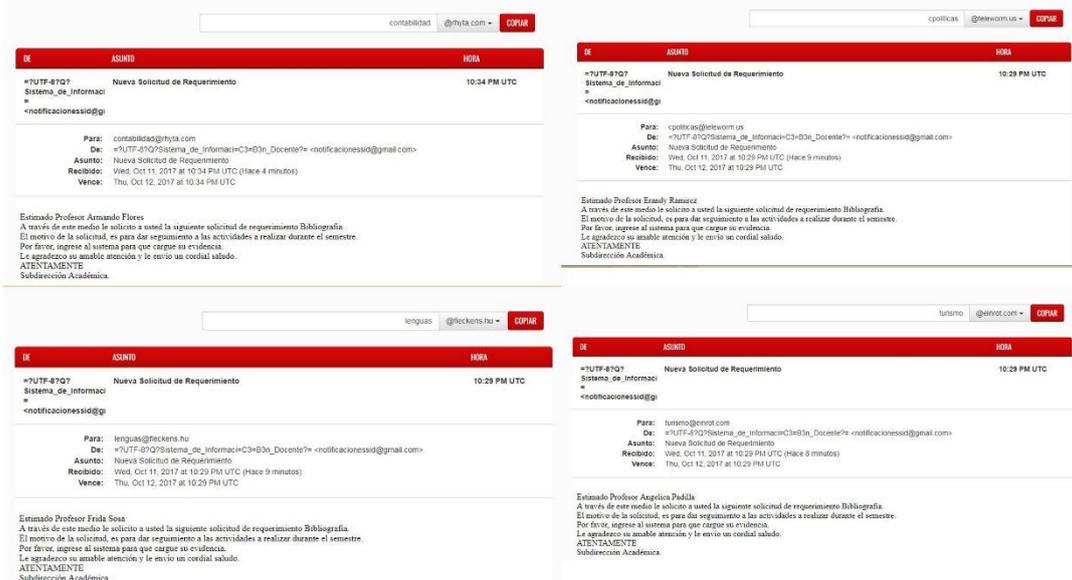


Figura 100 Email de Notificación de Solicitud de Requerimiento

5.4.4.4 Historial De Solicitudes

Toda solicitud creada sin importar la opción la cual fue hecha, será mostrada en forma de lista, esperando que sea respondida por el profesor (*Figura 101, Figura 102, Figura 103*).

Fecha de Creación	Nombre del Requerimiento	Nombre del Solicitante	Nombre del Profesor	Fecha de Respuesta	Archivo de Requerimiento
2017-10-01 23:22:25.000000	Bibliografía	David Sánchez	José Sánchez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	José Sánchez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Elizabeth Alvarez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Angelica Padilla	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Frida Sosa	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Erandy Ramirez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Armando Flores	Sin respuesta	Sin respuesta

Figura 101 Solicitudes de Requerimientos para Todas las Licenciaturas (Bibliografía)

Fecha de Creación	Nombre del Requerimiento	Nombre del Solicitante	Nombre del Profesor	Fecha de Respuesta	Archivo de Requerimiento
2017-10-01 23:22:25.000000	Bibliografía	David Sánchez	José Sánchez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	José Sánchez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Elizabeth Alvarez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Angelica Padilla	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Frida Sosa	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Erandy Ramirez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Armando Flores	Sin respuesta	Sin respuesta
2017-10-11 18:21:44.000000	Metodo de Evaluación	David Sánchez	Elizabeth Alvarez	Sin respuesta	Sin respuesta

Figura 102 Solicitud de Requerimiento por Licenciatura (Marcado Con Amarillo)

Fecha de Creación	Nombre del Requerimiento	Nombre del Solicitante	Nombre del Profesor	Fecha de Respuesta	Archivo de Requerimiento
2017-10-01 23:22:25.000000	Bibliografía	David Sánchez	José Sánchez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	José Sánchez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Elizabeth Alvarez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Angelica Padilla	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Frida Sosa	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Erandy Ramirez	Sin respuesta	Sin respuesta
2017-10-11 17:27:52.000000	Bibliografía	David Sánchez	Armando Flores	Sin respuesta	Sin respuesta
2017-10-11 18:21:44.000000	Metodo de Evaluación	David Sánchez	Elizabeth Alvarez	Sin respuesta	Sin respuesta
2017-10-11 18:38:16.000000	Horario	David Sánchez	Frida Sosa	Sin respuesta	Sin respuesta
2017-10-11 18:38:16.000000	Horario	David Sánchez	Erandy Ramirez	Sin respuesta	Sin respuesta
2017-10-11 18:38:16.000000	Horario	David Sánchez	Armando Flores	Sin respuesta	Sin respuesta

Figura 103 Creación de Solicitudes por Profesor (Cuadro Azul)

5.5 Incremento IV Responder Solicitud

5.5.1 Análisis de Requerimientos

Para concluir con el desarrollo del sistema, se solicita que se cree la funcionalidad de responder las solicitudes de requerimientos creadas por los administradores. Cuando el administrador crea la solicitud, el profesor debe de responderla mediante el acceso al sistema. El usuario profesor, únicamente podrá observar la lista de requerimientos, además de modificar su contraseña y cerrar sesión en la parte del Perfil (el cual muestra su nombre).

El profesor debe de contar con la siguiente vista una vez que acceso a la lista de requerimientos solicitados que el corresponde:

- Fecha de creación.
- Nombre del requerimiento.
- Nombre del emisor
- Fecha de respuesta.
- Nombre del archivo.

En el caso de la sección del archivo, el profesor debe de contar con la opción de subir el archivo. Esta función dará acceso a una ventana la cual, podrá hacer la carga de la evidencia (archivo). El formato de la evidencia debe ser en formato PDF, por lo cual, será restringida en caso de que seleccione otro formato que no sea válido.

Cuando el archivo ha sido almacenado, se oculta la función para subir el archivo y se mostrara el nombre del archivo para que pueda ser visualizado y descargado por ambos usuarios. También, se actualizará el campo de fecha de respuesta, mostrando el día y hora en que fue guardada la evidencia.

Además, cuando el administrador inicie sesión, y vaya a la lista de requerimientos, pueda observar de igual manera la actualización de los campos, comprobando que se le dio seguimiento y respuesta a su petición.

5.5.2 Diseño

5.5.2.1 Diagrama De Incrementos

En este incremento, se muestran los diagramas completos, es decir, la manera en cómo fue evolucionando el sistema hasta llegar a este último punto, concluyendo con el desarrollo del sistema.

5.5.2.1.1 Diagrama de Casos de Uso

El diagrama de casos de uso muestra la manera en que el profesor realizara la respuesta de la solicitud previamente hecha por el administrador. El diagrama muestra como el profesor hará la selección y almacenamiento del archivo en formato PDF para ser revisado posteriormente por el administrador (*Figura 104*).

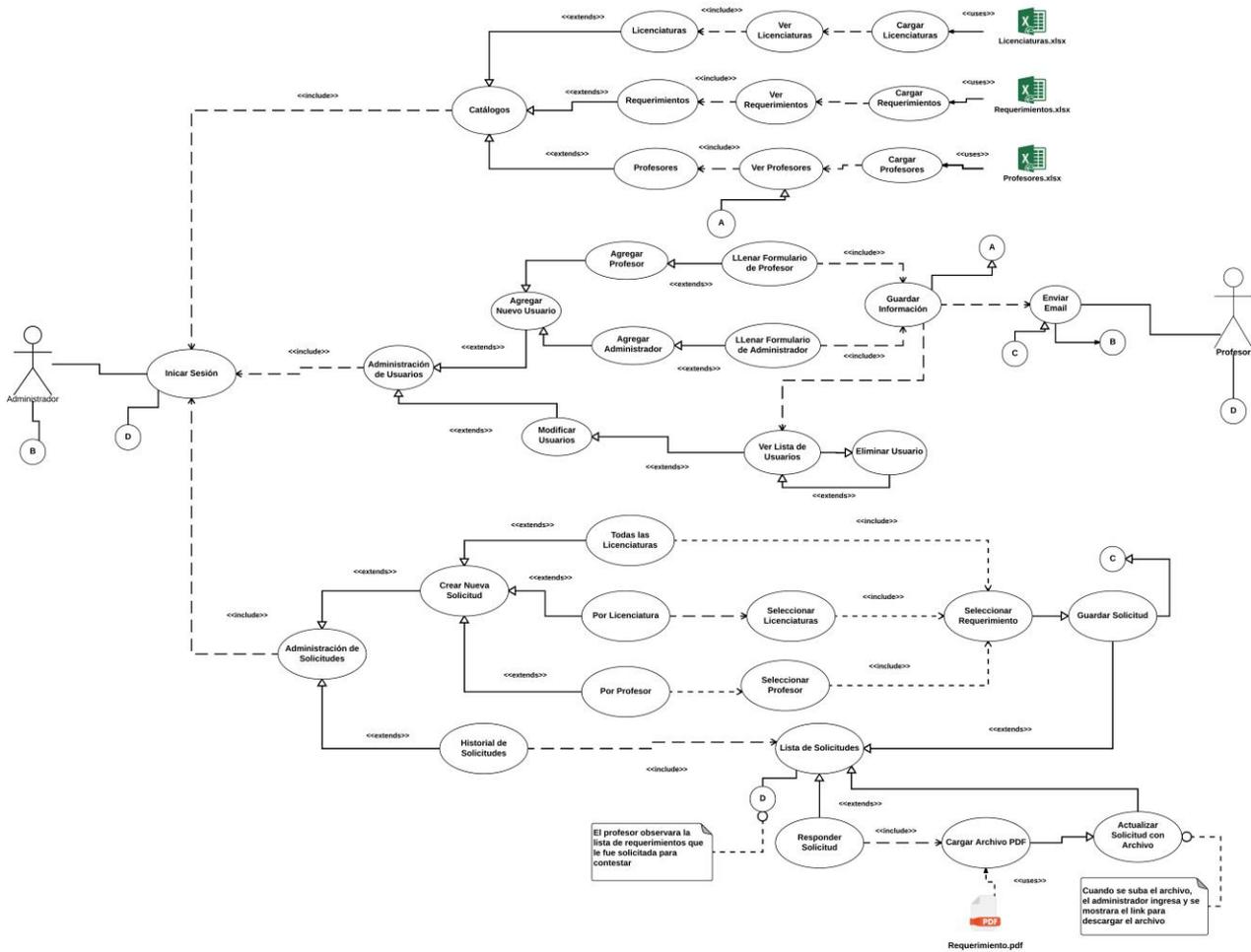


Figura 104 Diagrama de Casos de Uso (Incremento IV Responder Solicitud de Requerimiento)

En este caso, se añaden pocos estados para este caso de uso, ya que la mayoría están planteados desde el tercer incremento. Solo se añadió el proceso para la carga del archivo, y como se actualiza el historial de requerimientos con el archivo incluido.

5.5.2.1.2 Diagrama de Actividades

El diagrama de actividades especifica más a detalle la manera en cargar el archivo en formato PDF hecha por el profesor. Desde su ingreso al sistema, el profesor observará la tabla, es decir, su lista de requerimientos, con los campos similares al del administrador, predominando la función de selección y carga del archivo. Se añade la actividad de la selección y carga del archivo (Figura 105).

5.5.2.1.3 Diagrama de Clases

El diagrama de clases se centra en el uso de la clase ControladorArchivo, el cual permite que se pueda almacenar el archivo en la base de datos y ser visualizado por ambos usuarios. El uso de esta clase en conjunto con los métodos de solicitudEmisor y solicitudReceptor mostrará el archivo para ser descargado (Figura 106).

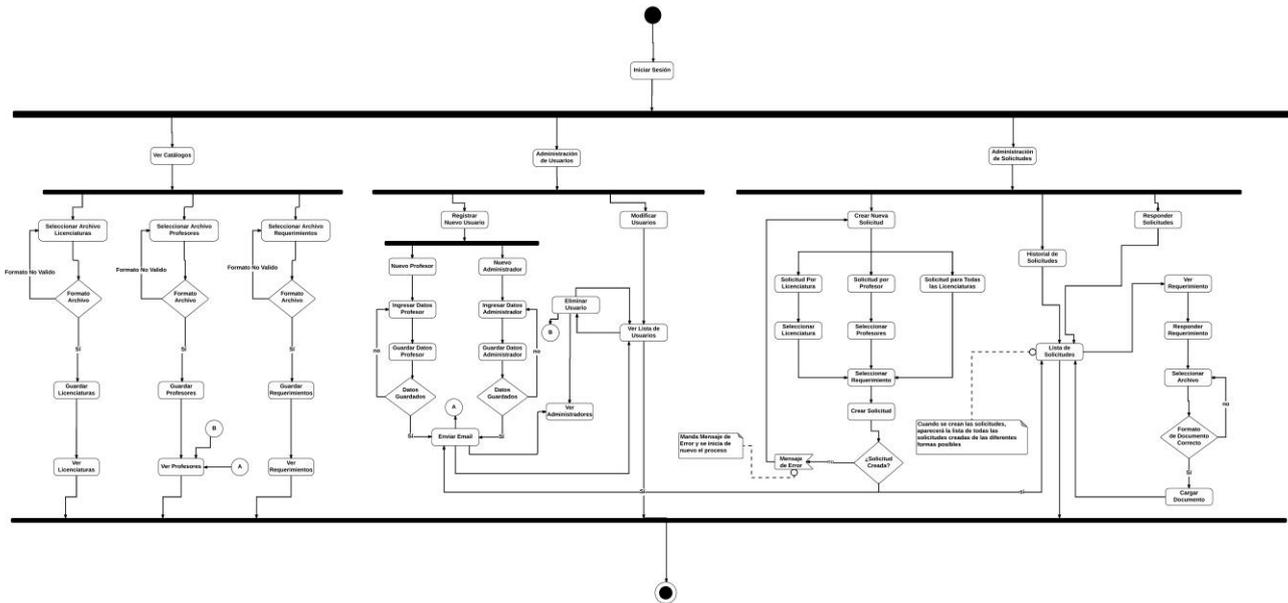


Figura 105 Diagrama de Actividades (Incremento IV Responder Solicitud de Requerimiento)

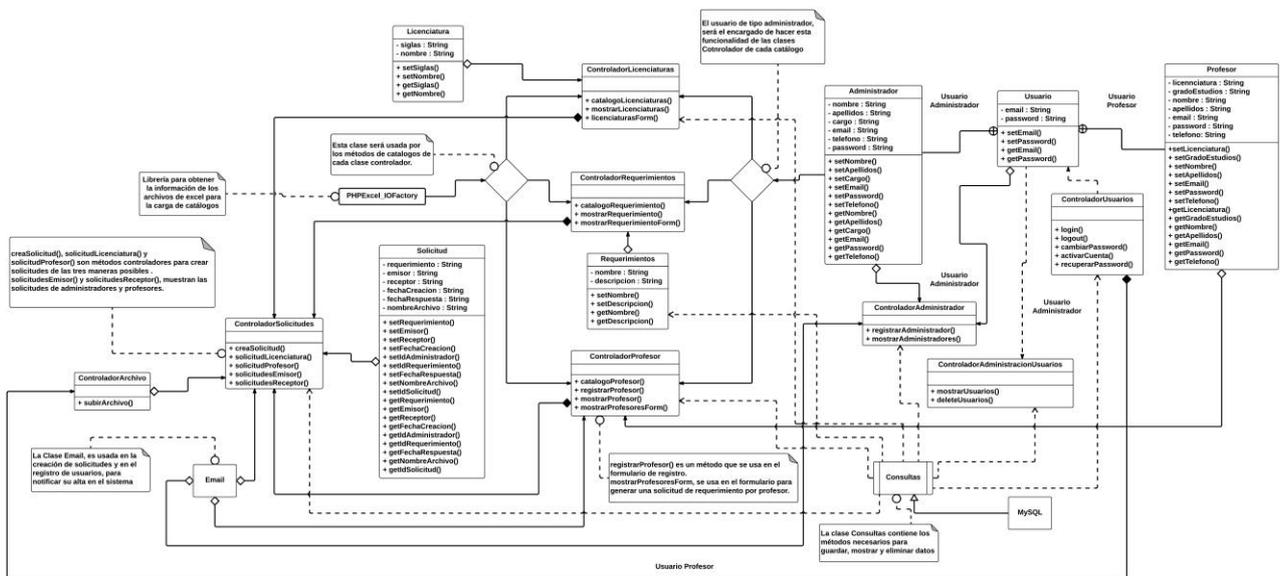


Figura 106 Diagrama de Clases (Incremento IV Responder Solicitud de Requerimiento)

5.5.2.2 Interfaz de Usuario

El profesor solo contará con tres opciones del menú correspondiente a sus privilegios, destacando que solo tendrá acceso a sus solicitudes de requerimiento y perfil (Figura 107).

Cuando navega en la sección de ver requerimientos, mostrara la lista de solicitudes de requerimientos para que el usuario responda mediante la carga de su archivo en formato PDF (Figura 108).



Figura 107 Inicio de Sesión de un Profesor

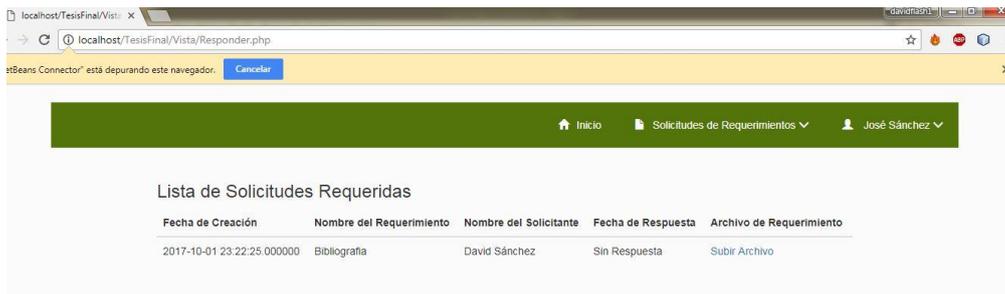


Figura 108 Lista de Solicitudes de Requerimientos

5.5.2.2.1 Subir Archivo

Cuando se pretende subir el archivo, este nos dirige al formulario para hacerlo, y posteriormente para regresar al historial de requerimientos para comprobar que ya fue guardado y así, ocultar esta funcionalidad (**Figura 109**).



Figura 109 Formulario para Subir Archivo de Requerimiento

5.5.3 Codificación

Ir a capítulo 7 Anexo

5.5.4 Implementación

Cuando el profesor ingresa al sistema, este podrá ver la lista de solicitudes de requerimientos para que pueda subir su evidencia como respuesta. Solo podrá subir un archivo (**Figura 110**).

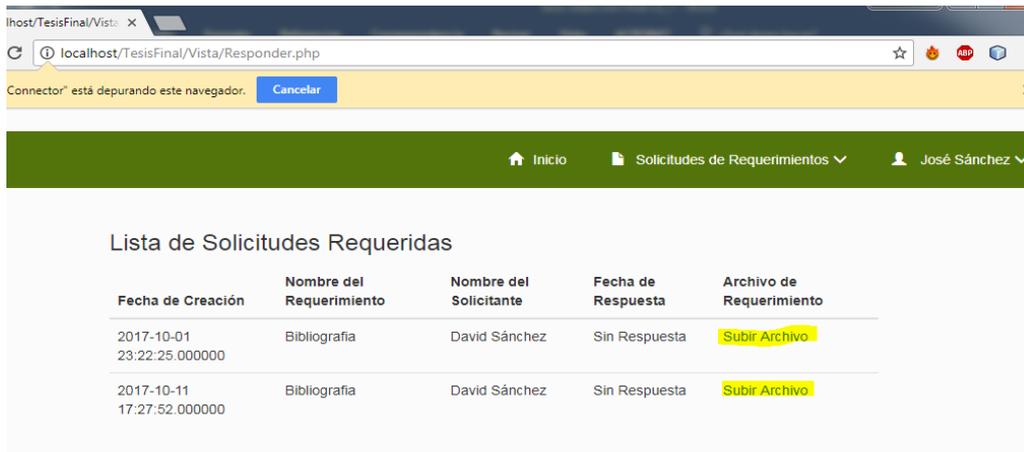


Figura 110 Lista de Requerimientos a Responder

Al dar clic en subir archivo (marcado en amarillo), nos dirige al formulario para cargar la evidencia (archivo). El formato para el archivo será en PDF, tal como se muestra en la imagen (Figura 111). Cuando se guarda el archivo, aparece un mensaje que se ha guardado con éxito (Figura 112).

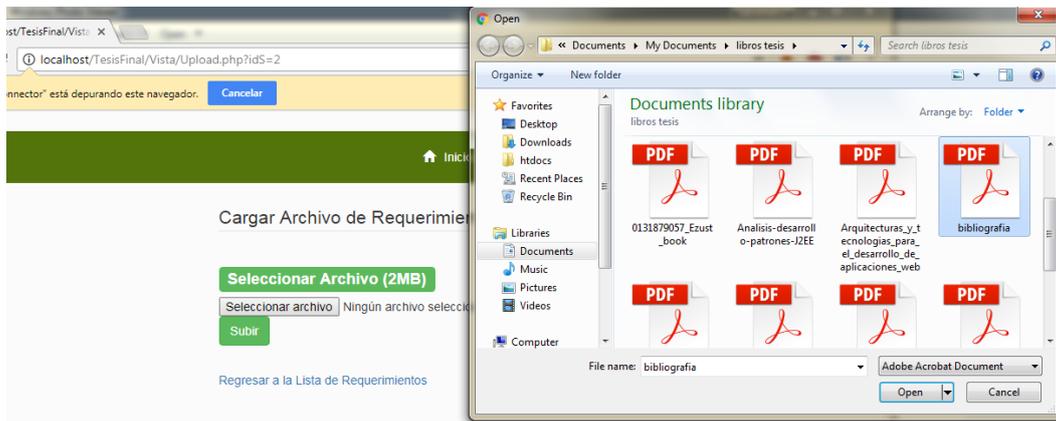


Figura 111 Cargar Archivo para Requerimiento

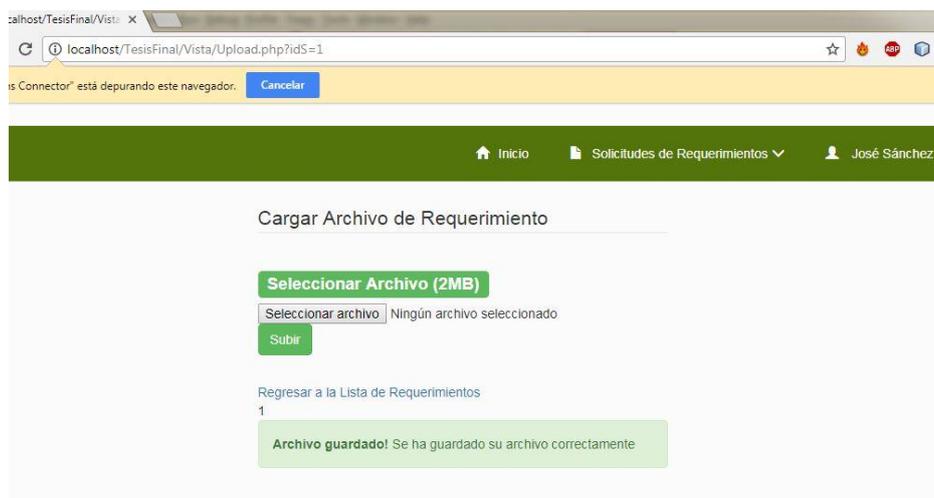


Figura 112 Mensaje de Archivo Guardado

Cuando el archivo es almacenado, se oculta la función de subir archivo y se pone el nombre del archivo para ser descargado (*Figura 113*).

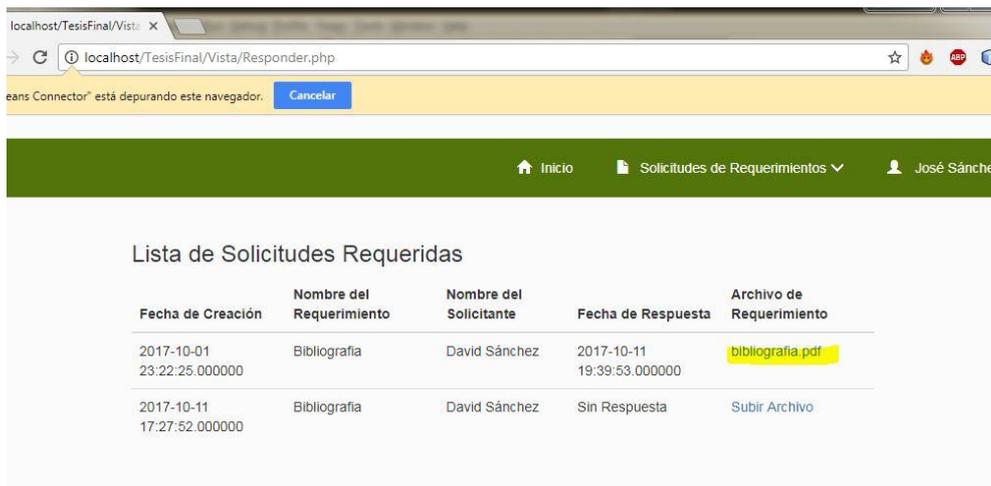


Figura 113 Archivo (Evidencia) Almacenada



Figura 114 Evidencia Vista Desde el Navegador (Usuario)

De igual manera, cuando el administrador inicie sesión, podrá ver el archivo descargarlo, además se actualiza las columnas de fecha de respuesta y nombre del archivo (*Figura 115*).

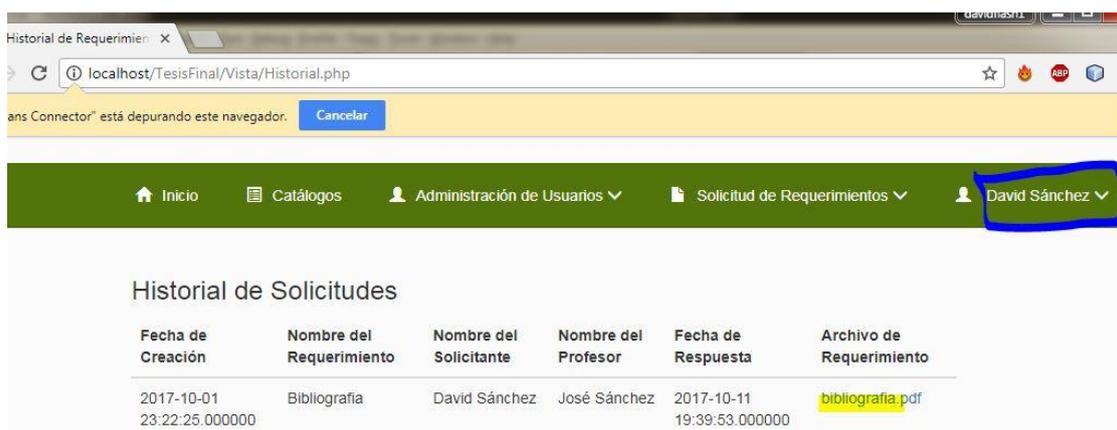


Figura 115 Historial Actualizado de Solicitudes (Administrador)

5.6 Diseño de Base de Datos

La integridad y almacenamiento de los datos, es de vital importancia para cada software a desarrollar. En ese sentido y de manera ajena a los incrementos, se muestra el modelo relacional de la base de datos. Cada tabla está relacionada para cumplir con las necesidades de almacenamiento que se llevan cabo en cada incremento descrito y desarrollado.

Las tablas administrador y profesor, son las tablas que almacenaran la información personal de los usuarios. La tabla usuarios, será la encargada de dar el acceso al sistema, tomando algunos campos de las tablas anteriores. Para llevar el control de las relaciones entre la tabla administrador-usuarios y profesor-usuario, cuentan con una tabla para verificar la cuenta a la cual están asociadas.

La tabla solicitud, requiere del administrador, profesor y requerimientos para almacenar la información y mediante la tabla que relaciona entre la solicitud y el profesor, se llevara el control de las solicitudes hechas con su identificador de tablas anteriores (id).

La tabla, cat_requerimientos y licenciaturas, son usadas cuando se insertan los datos a través de la función de catálogos. Para el desarrollo del sistema se usó el gestor de base de datos MySQL (Figura 116).

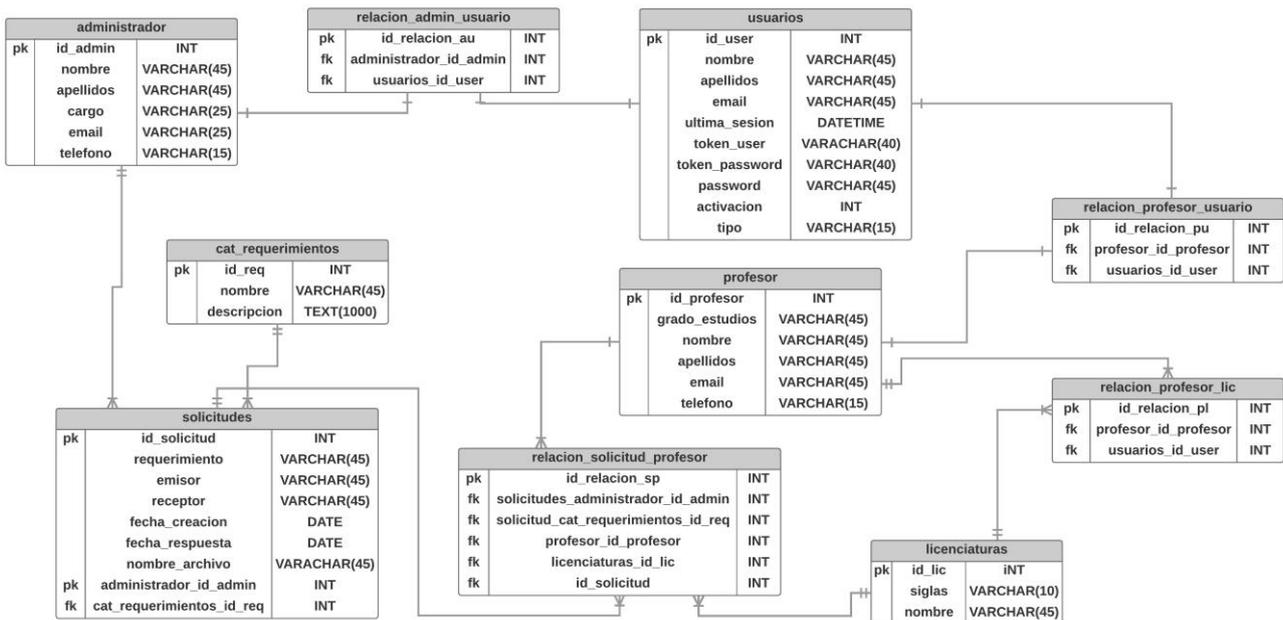


Figura 116 Modelo Relacional de la Base de Datos del Sistema

CAPÍTULO 6 CONCLUSIÓN

La recolección de requerimientos es muy importante para saber y comprender las necesidades del cliente y ofrecerles una alternativa a dichos requerimientos. Tal como se demuestra en el desarrollo de este sistema. A partir de los requerimientos de la subdirección académica, estos fueron recabados con un alto grado de explicación para desarrollar el sistema. Al obtener todos los requerimientos, cada requerimiento fue analizado y comprendido a su máxima elección permitiendo la toma de decisiones para la selección de metodologías y herramientas para el desarrollo del sistema.

La selección de las herramientas y tecnologías para el desarrollo del sistema, se basó en la infraestructura con la que cuenta el centro universitario y de la universidad en general. El uso del lenguaje de programación PHP (que soporta programación orientada a objetos) y el motor de base de datos MySQL, permite que la universidad tenga los mecanismos necesarios para que el sistema pueda operar sin ningún contratiempo. El uso de frameworks como Bootstrap, por ejemplo, ayuda a que el sistema sea fácil en términos de interacción.

Parte de la toma de decisiones que se tomó durante el análisis de requerimientos, fue la selección del modelo de desarrollo incremental. Una de las ventajas que encontré en este modelo fue la manera de entregar el proyecto, es decir, determinar el número de incrementos para su creación y la retroalimentación. Los requerimientos recabados se priorizaron de mayor a menor importancia determinando 4 incrementos para el desarrollo del sistema.

Se determinaron 4 incrementos para desarrollar el sistema debido a que, durante el análisis de requerimientos, se detectó primeramente 4 funcionalidades que son importantes para el desarrollo y funcionalidad del sistema. El primer incremento se basó en los catálogos y los privilegios de los tipos de usuario. El segundo se centró en la administración de usuarios. El tercero se basó en crear solicitudes de requerimientos y el cuarto se basó en responder los requerimientos. En ese sentido, la elección del modelo incremental permitió que existiera siempre una constante retroalimentación entre usuario y desarrollador desde el primer incremento, donde se tomaron las observaciones para ser mejorado el sistema durante los siguientes incrementos, sin descuidar los requerimientos prioritarios.

La selección del paradigma orientado a objetos se determinó con base en la manera que puede ayudar para el análisis, diseño, desarrollo e implementación del sistema, adaptándolo al modelo incremental en cada una de las fases donde fue necesario. Desde la manera de analizar los requerimientos y diseñar los diagramas UML (casos de uso, actividades y clases) que permitieran explicar de manera sencilla, la funcionalidad del sistema.

El desarrollo del sistema desde la parte del backend, fue desarrollado con la programación orientada a objetos, en conjunto con el patrón de diseño Modelo-Vista-Controlador (MVC). El patrón MVC fue útil para el acomodo del backend y el frontend, como parte de las buenas prácticas para un correcto desarrollo de software. Parte de las clases detectadas durante el análisis y diseño del sistema mediante el uso de UML, las clases controladoras se almacenaron en la sección de Controlador donde se contiene la funcionalidad de las clases,

mientras que, en cuestiones de obtención y almacenamiento de datos, se almacenaron en el Modelo. La parte del frontend se almaceno en la carpeta Vista, conteniendo toda la interfaz gráfica del sistema.

CAPÍTULO 7 ANEXO

El código fuente del sistema se encuentra en un repositorio en github. En ese sentido, copio el link de acceso al código de la aplicación web (*Figura 117*).

<https://github.com/yodavid1992/sid.git>

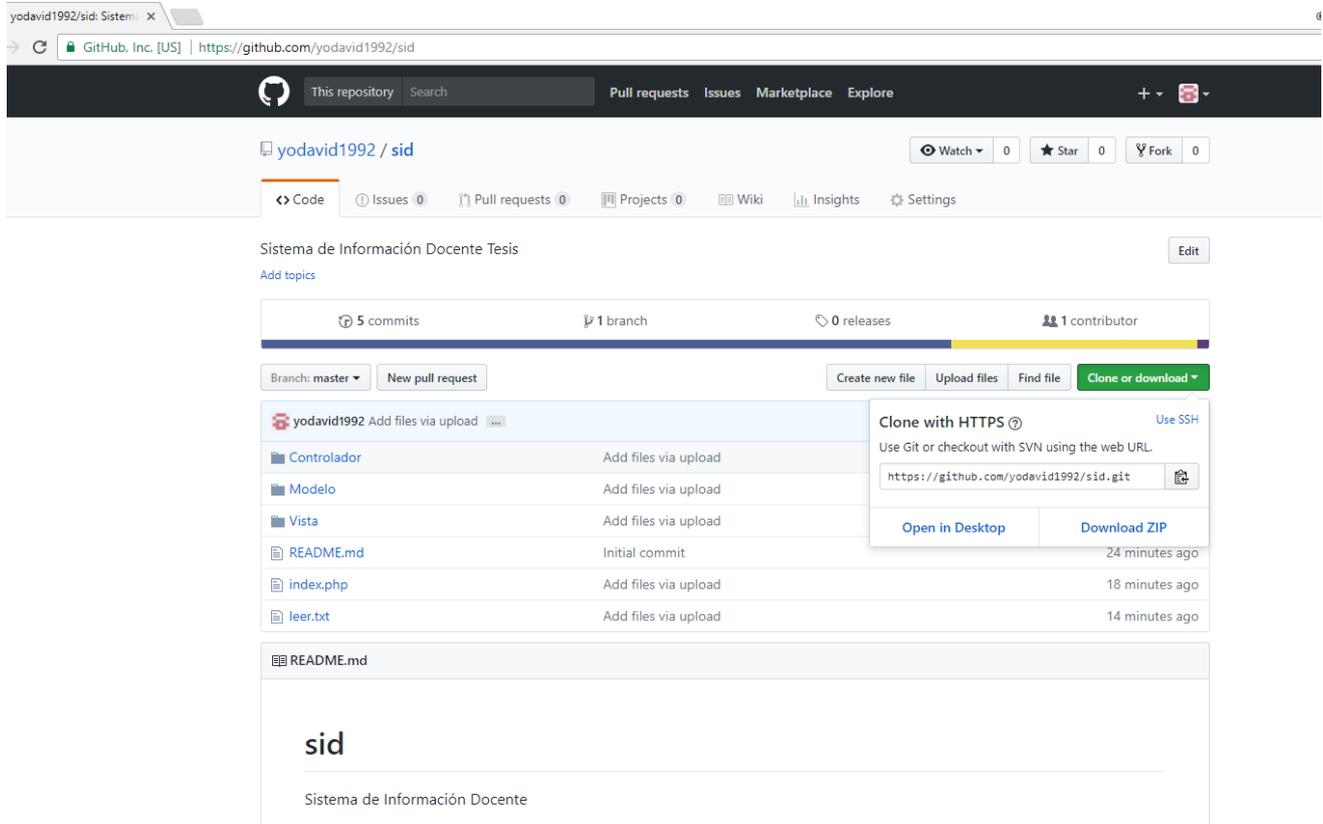


Figura 117 Repositorio del Sistema

BIBLIOGRAFÍA

- acensTechnologies. (Mayo de 2016). *acensTechnologies*. Obtenido de <https://www.acens.com/wp-content/images/2016/05/web-components-wp-acens.pdf>
- Aghaei, S., Nematbakhsh, M. A., & Farsani, H. K. (2012). Evolution of the World Wide Web: From Web 1.0 To Web 4.0. *International Journal of Web & Semantic Technology*, 1-10.
- Alvarez, M. A. (2016). Innovación en el Desarrollo. *DesarrolloWeb.com*, 1-18.
- Alvarez, M. A., & Martínez, D. (2016). Manual de Responsive Web Design. *desarrolloweb.com*, 1-52.
- Ansari, D. (8 de Marzo de 2017). *Webizar*. Obtenido de Webizar: <https://webizar.com/artificial-intelligence-transformed-seo/>
- Asadi, A. (2015). *The Python Book*. Imagine Publishing Ltd.
- BBVA Innovation Center. (Mayo de 2015). *BBVA Innovation Center*. Obtenido de <https://www.centrodeinnovacionbbva.com/sites/default/files/ebook-cibbva-realidad-virtual.pdf>
- BBVA Innovation Center. (2015). *BBVA Innovation Center*. Obtenido de <http://www.centrodeinnovacionbbva.com/sites/default/files/ebook-data-innovation-trends-cibbva-esp.pdf>
- Benington, H. D. (1983). *Production of Large Computer Programs* .
- Bhatti, K. (16 de Diciembre de 2016). *Linkedin*. Obtenido de <https://www.linkedin.com/pulse/what-front-end-website-brief-tools-used-development-bhatti->
- Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE*, 61-72.
- Boehm, B. W., Egyed, A., Kwan, J., & Madachy, R. (1997). Developing Multimedia Applications with the WinWin Spiral Model. *ESEC/FSE*, 1-20.
- Boehm, B., & Hansen, W. J. (2000). Spiral Development: Spiral Development: and Refinements. *Carnegie Mellon University Software Engineering Institute*, 1-37.
- Boehm, B., Bose, P., Horowitz, E., & Lee, M.-J. (1994). Software Requirements As Negotiated Win Conditions. *IEEE*, 74-83.
- Boehm, B., Bose, P., Horowitz, E., Scacchi, W., USC, Bendifallah, S., & Madni, A. (1993). Next-Generation Software Processes and Their Environment Support. *USC Center for Software Engineering*, 1-24.

- Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., & Madachy, R. (1998). Using the WinWin Spiral Model: A Case Study. *IEEE*, 33-44.
- Bruegge, B. (2006). *TUM Informatik Lehrstuhl 1*. Obtenido de TUM Informatik Lehrstuhl 1: https://www1.in.tum.de/lehrstuhl_1/files/teaching/ws0607/Software%20Engineering%20I/L17_SoftwareLifecycleModels.pdf
- Carmigniani, J., & Furht, B. (2011). *Department of Computer and Electrical Engineering and Computer Sciences*. Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.460.8192&rep=rep1&type=pdf>
- Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (20 de Diciembre de 2010). *ResearchGate*. Obtenido de https://www.researchgate.net/publication/216813812_Augmented_reality_technologies_systems_and_applications?enrichId=rgreq-f37e821bcba16d058001e139e97b732c-XXX&enrichSource=Y292ZXJQYWdlOzlxNjgxMzgxMjtBUzoxMzlyMTgzMzY4NDU4MjJRAMTQwODUzNDcxNjI3NQ%3D%3D&el=1_x
- Choudhury, N. (2014). World Wide Web and Its Journey from Web 1.0 to Web 4.0. *International Journal of Computer Science and Information Technologies*, 8096-8100.
- Cobos, G. E. (2016). 10 Tendencias Tecnológicas para el 2017. *Tecno*, 18-19.
- Cooper, J. W. (2000). *Java™ Design Patterns: A Tutorial*. Addison Wesley.
- Davies, R. (2015). The Internet of Things: Opportunities & Challenges. *EPRS European Parliamentary Research Service*, 2-8.
- Duckett, J. (2014). *JAVASCRIPT & JQUERY Interactive Front-End Web Development*. Indiana: John Wiley & Sons, Inc.
- Fragoso, R. B. (18 de Junio de 2012). *IBM Software Group México*. Obtenido de IBM Software Group México: <https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Giardina, F. (2011). Guía de ASP.NET: Desarrollo de sitios y aplicaciones web dinámicas. *Maestro de la Web*, 1-73.
- Giraldi, G., Silva, R., & Oliveira, J. C. (s.f.). *Introduction to Virtual Reality*. COMCIDIS Research Group.
- GP. Bullhound. (2016). 2017 Technology Predictions. *GP. Bullhound*, 1-52.
- Jendrock, E., Cervera-Navarro, R., Evans, I., Haase, K., & Markito, W. (2014). *Java Platform, Enterprise Edition The Java EE Tutorial, Release 7*. Oracle.

- JSON.ORG. (2017). *JSON.ORG*. Obtenido de <http://www.json.org/json-es.html>
- Kaul, A., & Wheelock, C. (2016). Artificial Intelligence: 10 Trends to Watch in 2017 and Beyond. *Tractica LLC*, 1-23.
- Kemp, S. (2017). *We Are Social's DIGITAL IN 2017*. We Are Social.
- Kitamura, E. (29 de Octubre de 2014). *www.webcomponents.org*. Obtenido de <https://www.webcomponents.org/community/articles/introduction-to-shadow-dom>
- Kitamura, E. (4 de Enero de 2015). *www.webcomponents.org*. Obtenido de <https://www.webcomponents.org/community/articles/introduction-to-html-imports>
- Krause, J. (22 de Mayo de 2016). *Renegade Tribune*. Obtenido de <http://www.renegadetribune.com/witness-nightmarish-future-augmented-reality/>
- Larman, C., & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. *IEEE Computer Society*, 2-11.
- Lund, D., Turner, V., MacGillivray, C., & Morales, M. (2014). Worldwide and Regional Internet of Things (IoT) 2014–2020 Forecast: A Virtuous Circle of Proven Value and Demand. *IDC*, 1-28.
- Martínez, F. J. (2011). Presente y Futuro de la Tecnología de la Realidad Virtual. *Creatividad y Sociedad*, 1-39.
- Mathur, S., & Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications*, 29-34.
- Mell, P., & Grance, T. (Septiembre de 2011). *National Institute of Standards and Technology*. Obtenido de <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Mills, H. D. (1980). Management of Software Engineering, The - Part I: Principles of Software Engineering. *IBM Systems Journal*, 414-477.
- Mobinius. (7 de Abril de 2017). *Mobinius*. Obtenido de <https://www.mobinius.com/7-ai-trends-look-2017/>
- Nixon, R. (2015). *PHP: 20 Lessons to Successful Web Development*. McGraw-Hill Education.
- Patel, K. (2013). Incremental Journey for World Wide Web: Introduced with Web 1.0 to Recent Web 5.0 – A Survey Paper. *International Journal of Advanced Research in Computer Science and Software Engineering*, 410-417.
- Pénaire, C., Edwards, M., Fernandes, A., Mancin, E., & Carroll, K. (2007). *The IBM Rational Unified Process for System z*. International Business Machines Corporation.

- Pierce, J. (2016). *Shopify Inc.* Obtenido de Shopify Inc.: <https://es.shopify.com/blog/14934065-hablemos-sobre-startups-que-son-sus-caracteristicas-e-importancia>
- Piper, K. (21 de Marzo de 2015). *Piper Report.* Obtenido de <http://piperreport.com/blog/2015/03/21/behavioral-healthcare-and-technology-science-based-web-and-mobile-health-tools-to-improve-care/>
- Pogue, D. (19 de Noviembre de 2015). *Yahoo Tech.* Obtenido de <https://www.yahoo.com/tech/battle-of-the-voice-assistants-siri-cortana-211625975.html>
- Point, T. (2015). *Tutorials Point (I) Pvt. Ltd.* Obtenido de Tutorials Point (I) Pvt. Ltd.: http://www.tutorialspoint.com/perl/perl_tutorial.pdf
- Press, G. (23 de Enero de 2017). *Forbes.* Obtenido de <https://www.forbes.com/sites/gilpress/2017/01/23/top-10-hot-artificial-intelligence-ai-technologies/#1c89fa151928>
- Pressman, R. S. (2001). *Software Engineering: a Practitioner's Approach.* New York: McGraw-Hill Higher Education.
- Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering A Practitioner's Approach.* NY: Mc Graw Hill Education.
- Rails, R. o. (2016). *Ruby on Rails.* Obtenido de Ruby on Rails: <http://rubyonrails.org/>
- Rasulury, P. (Enero de 2015). *Whatsupgeek.* Obtenido de <http://www.whatsupgeek.com/2015/01/virtual-reality-future-of-gaming.html>
- Rayo, Á. (17 de Mayo de 2016). *Bit .* Obtenido de <http://www.bit.es/knowledge-center/tipos-de-datos-en-big-data/>
- Rey, G. G. (2016). 2017: El Año de la Transformación Digital. *InnovationWeek Magazine*, 1-16.
- Rojas, J. A., & Echeverri, C. G. (2012). Obtenido de http://www.udi.edu.co/congreso/historial/congreso_2012/ponencias/comunicacion/PE_RSPECTIVAS_DE_LA_REALIDAD_AUMENTADA_Y_SU_USO_EN_PROCESOS_COMUNICATIVOS.pdf
- Rowen, R. B. (1990). Software Project Management Under Incomplete. *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, 10-21}.
- Royce, W. W. (1970). Managing the Development of Large Software Systems. *IEEE by TRW*, 328-338.
- Schroek, M., Shockley, R., Smart, J., Romero-Morales, D., & Tufano, P. (2014). Analytics: el uso de big data en el mundo real. *IBM Global Business Services*, 1-20.

- Sommerville, I. (2005). *Ingeniería de Software*. Madrid: Pearson Educación .
- Sweat, J. E. (2006). *php|architect's Guide to PHP Design Patterns*. Toronto: Marco Tabini & Associates, Inc.
- Takala, H. (19 de Enero de 2017). *Net Man*. Obtenido de Net Man: <https://netman.fi/fi/it-blogi/tietohallintometeorologiaa-ja-pilviennuste/>
- Tatikonda, M. V., & Lorence, M. (2002). Towards Effective Software Development: A Conceptual Framework of Software Project Types, Development Processes, and Functional Outcomes. En *New Directions in Supply-Chain Management: Technology, Strategy and Implementation*. New York: Boone and Ganeshan American Management Association,.
- Tutorials Point . (2015). *Tutorials Point*. Obtenido de Tutorials Point: https://www.tutorialspoint.com/design_pattern/design_pattern_tutorial.pdf
- Unziplogic System. (2014). *Unziplogic System*. Obtenido de Unziplogic System: <http://unziplogic.com/node/11>
- Vera, A. (Enero de 2017). *steemit*. Obtenido de steemit: <https://steemit.com/spanish/@javf1016/big-data>
- Verizon. (2016). *Verizon*. Obtenido de <https://www.verizon.com/about/sites/default/files/state-of-the-internet-of-things-market-report-2016.pdf>
- W3Schools. (2017). *W3Schools*. Obtenido de W3Schools: <https://www.w3schools.com/html/default.asp>
- W3Schools. (2017). *W3Schools*. Obtenido de <https://www.w3schools.com/xml/>
- W3Schools. (2017). *W3Schools*. Obtenido de W3Schools: <https://www.w3schools.com/css/default.asp>
- Wix. (18 de Abril de 2013). *Wix*. Obtenido de Wix: <http://www.wix.com/blog/2013/04/building-a-website/>
- www.webcomponents.org. (26 de Mayo de 2017). *www.webcomponents.org*. Obtenido de <https://www.webcomponents.org/libraries>
- Yadav, R. S. (2012). Improvement in the V-Model. *International Journal of Scientific & Engineering Research*, 1-8.