

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación



Desarrollo de una Aplicación Web para la Administración de Conferencias Académicas

Trabajo Especial de Grado
Presentado ante la Ilustre
Universidad Central de Venezuela
Por los Bachilleres:

Ilio Gabriel Espinel Contreras
C.I.: 18.353.899
gabriel.espinel.c@gmail.com

Antonio Alejandro Alarcón Guerra
C.I.: 18.039.050
antonio.alarcon@outlook.com

Para optar al título de Licenciado en Computación

Tutores:

Prof. Eric Gamess
Prof. Dedaniel Urribarri

Enero 2019



Universidad Central de Venezuela
 Facultad de Ciencias
 Escuela de Computación
 Laboratorio de Comunicación y Redes



ACTA DE VEREDICTO

Quienes suscriben, miembros del jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por los Bachilleres Gabriel Espinel (C.I. V-18.353.899) y Antonio Alarcon (C.I. V-18.039.050), con el título **“Desarrollo de una Aplicación Web para la Administración de Conferencias Académicas”**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el nombrado trabajo por cada uno de los miembros del jurado, éste fijó el día jueves 31 de enero de 2019, para que sus autores lo defendieran en forma pública, en la Mezzanina de la Biblioteca Central de la UCV, mediante una exposición oral de su contenido, luego de la cual respondieron satisfactoriamente a las preguntas que les fueron formuladas por el jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela.

Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

Es de aclarar que Gabriel Espinel y Antonio Alarcon hicieron su presentación a través de una vídeo-conferencia, ya que se encuentran viviendo en Chile, con la aprobación de todos los miembros del jurado. Adicionalmente, el Profesor Eric Gamess se encuentra fuera del país. Por esta razón, el Profesor Robinson Rivas, director de la Escuela de Computación, firma el presente documento en su lugar.

En fe de lo cual se levanta la presente acta en Caracas, a los 31 días del mes de enero del año 2019, dejando constancia que el Profesor Dedaniel Urribarri actuó como coordinador del jurado.

 <hr/> por el Prof. Eric Gamess Tutor		 <hr/> Prof. Dedaniel Urribarri Tutor
 <hr/> Prof. Yosly Hernández Jurado Principal		 <hr/> Prof. Andrés Sanoja Jurado Principal

Resumen

TÍTULO

Desarrollo de una Aplicación Web para la Administración de Conferencias Académicas.

AUTORES

Ilio G. Espinel C. y Antonio A. Alarcón G.

TUTOR

Prof. Eric Gamess

Prof. Dedaniel Urribari

RESUMEN

En este Trabajo Especial de Grado se propone la creación de una aplicación Web capaz de administrar y automatizar los procesos de gestión de conferencias académicas pagas como la *Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa)*, y conferencias gratuitas como el *Simposio Científico y Tecnológico en Computación (SCTC)*.

La solución busca cubrir las necesidades particulares que no son cubiertas por las aplicaciones de administraciones de conferencias usadas actualmente, como es el caso de EasyChair u Open Conference System (OCS).

La aplicación fue desarrollada bajo los lineamientos de la metodología de desarrollo AgilUs. Asimismo, se implementa con una variedad de tecnologías modernas, entre las cuales podemos mencionar: Java, Angular, Spring Boot, Hibernate, entre otras. Como resultado se obtuvo un sistema que automatiza procesos administrativos de las conferencias académicas, simplificando el trabajo del personal y mejorando el rendimiento de estos en función del tiempo y costo invertido en dicha administración.

También, es de aclarar que este sistema fue usado exitosamente para la administración de CoNCISa 2017 y CoNCISa 2018.

Palabras clave: Aplicación Web, Desarrollo de Software, Conferencias Académicas, CoNCISa, SCTC.

Agradecimientos

Este Trabajo Especial de Grado y todo el esfuerzo que implicó su desarrollo se lo quiero dedicar a todas aquellas personas que contribuyeron a su realización.

Principalmente, agradezco a mi madre Thamara por su apoyo, guía, preocupaciones, regaños, y buenos deseos durante todos los años de mi vida. Por servir como un inquebrantable ejemplo de que no importa las adversidades que la vida nos depare, siempre hay que seguir luchando. A ella y a mi hermano Gilmar, agradezco infinitamente su apoyo siempre que lo necesitaba. A mi padre Jorge, quién sirvió como ejemplo a seguir durante todos estos años, en los distintos ámbitos de la vida. ¡Los amo inmensamente!, ¡Gracias!

A mi esposa Francis y mi hija Alana, muchas gracias por su apoyo y comprensión en la realización de esta etapa de mi vida. No habría podido lograrlo sin ustedes apoyándome. ¡Las amo profundamente!, ¡Gracias!

A mi tío Evangelista, por su apoyo incondicional durante todos estos años. Por considerarme como un hijo para él, tanto como yo lo considero como un padre. ¡Gracias, tío!

A mis tíos Sonia y Orlando, quienes han servido como ejemplo durante mi vida, y por sus buenos consejos. Junto con mis primos Miguel y Nana, quienes siempre estuvieron disponibles y me ayudaron. ¡Muchas gracias!

A mis tías Chile, Betty, y Amparo, que me apoyaron en momentos claves de mi vida, indicándome el camino correcto. ¡Muchas gracias!

A mi tutor y profesor Eric Gamess, quien nunca se rindió y me impulsó a finalizar mi carrera. ¡Gracias por no rendirse!

A mi compañero Antonio, por acompañarme en esta aventura y ayudarme a cerrar el capítulo de esta etapa. ¡Gracias!

Agradezco a todos aquellos que me apoyaron durante todos los años para lograr esta meta y que esta página no logra mencionar explícitamente. ¡Gracias por su apoyo y buenos deseos!

“Si he visto más lejos es porque he subido sobre los hombros de gigantes”, Isaac Newton.

Gabriel Espinel

Agradecimientos

A la vida. Estoy agradecido de tener la oportunidad de llegar a este momento tan importante, la dicha de conocer a tantas personas maravillosas, las experiencias vividas, todo fue fundamental para forjar el carácter y las convicciones que dieron pie para que este Trabajo Especial de Grado fuera posible.

A mi padre Eduardo. Por siempre estar presente, enseñarme valores, a dar lo mejor para ser una persona de bien. Tus consejos y el trabajo duro fueron el mejor ejemplo para lograr cualquier meta que quisiera cumplir, este trabajo representa años de apoyo incondicional de tu parte.

A mi madre Morela. Eres mi ejemplo de perseverancia y dedicación, me diste cariño, me enseñaste la compasión, ser integro, humilde y ser fuerte ante las adversidades, sin ti estas palabras no hubieran sido posible.

A mi hermano Eduardo. Me acompañas en todo momento, la vida te presento una gran responsabilidad y la cumpliste, representa tu firmeza, tu compasión, tu compromiso, siempre estaré agradecido por tu apoyo.

A mi pareja y amiga Catherine. Tu Amor, apoyo y esfuerzo han sido claves durante todos estos años, me acompañas en todo momento con una gran sonrisa, eres la responsable de mis alegrías y ganas de seguir cumpliendo las metas.

A mi tutor. Al profesor Eric Gamess por su gran calidad humana, siempre atento para lograr los objetivos y apoyarnos durante todo este proceso. Felicito su compromiso con la educación y la Universidad Central de Venezuela, gracias a eso este trabajo es posible.

A mi compañero de tesis Gabriel. Tu calidad como persona y compañero es excepcional, gracias por invitarme a formar parte y compartir este trabajo contigo, espero que la vida te de todo lo que sueñas, te lo mereces.

A mi familia, amigos y profesores que no pude mencionar, les agradezco haber estado presentes, por su apoyo y cariño, son parte importante de mi vida y de mis logros.

Antonio Alarcón

Tabla de Contenido

ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	11
1. INTRODUCCIÓN.....	13
2. EL PROBLEMA.....	15
2.1. PLANTEAMIENTO DEL PROBLEMA	15
2.2. JUSTIFICACIÓN	16
2.3. OBJETIVOS.....	16
2.3.1. Objetivo General.....	16
2.3.2. Objetivos Específicos.....	17
2.4. ALCANCE.....	17
3. CONFERENCIAS ACADÉMICAS.....	19
3.1. SIMPOSIO CIENTÍFICO Y TECNOLÓGICO EN COMPUTACIÓN.....	20
3.2. CONFERENCIA NACIONAL DE COMPUTACIÓN, INFORMÁTICA Y SISTEMAS.....	20
4. ESTILOS DE ARQUITECTURAS DE SISTEMAS DE SOFTWARE.....	23
4.1. ARQUITECTURA CLIENTE-SERVIDOR.....	24
4.1.1. Servidor de Archivos	25
4.1.2. Servidor de Correos Electrónicos	25
4.1.3. Servidor Web	25
4.1.4. Servidor de Base de Datos	25
4.2. CLIENTE-SERVIDOR POR CARGA DE PROCESAMIENTO	25
4.3. CLIENTE-SERVIDOR POR MANEJO DE ESTADO DE LA APLICACIÓN.....	26
4.4. APLICACIÓN WEB COMO IMPLEMENTACIÓN DE LA ARQUITECTURA CLIENTE-SERVIDOR	27
5. TRABAJOS RELACIONADOS	30
5.1. OPEN CONFERENCE SYSTEM	30
5.2. CONFTOOL	31
5.3. CONFERENCE MANAGEMENT SYSTEM (COMS)	33
5.4. OPENCONF	33
5.5. EASYCHAIR.....	35
5.6. COMPARATIVA DE CARACTERÍSTICAS.....	37
6. MARCO METODOLÓGICO	39
6.1. ADAPTACIÓN DE LA METODOLOGÍA DE DESARROLLO AGILUS	39
6.2. CICLO DE VIDA.....	39
6.2.1. Requisitos.....	40
6.2.2. Análisis	40
6.2.3. Prototipaje	40
6.2.4. Entrega	40
7. MARCO TECNOLÓGICO	41
7.1. TECNOLOGÍAS DE FRONT-END	41
7.1.1. HTML.....	41
7.1.2. CSS	42
7.1.3. Bootstrap	43
7.1.4. JavaScript.....	44
7.1.5. Angular	45

7.1.6. TypeScript.....	46
7.2. TECNOLOGÍAS DE BACK-END.....	46
7.2.1. Java.....	46
7.2.2. Spring.....	50
7.2.3. H2.....	52
7.2.4. Hibernate.....	52
7.2.5. iText.....	53
7.2.6. JasperReports.....	54
7.3. HERRAMIENTAS DE DESARROLLO.....	54
7.3.1. JHipster.....	55
7.3.2. Node Package Manager.....	55
7.3.3. Webpack.....	56
7.3.4. Git.....	56
7.3.5. Apache Maven.....	57
8. MARCO APLICATIVO.....	59
8.1. ANÁLISIS GENERAL.....	59
8.2. PLAN DE ENTREGA.....	59
8.2.1. Primera Fase.....	60
8.2.2. Segunda Fase.....	61
8.3. DESARROLLO DE LA APLICACIÓN.....	61
8.3.1. Arquitectura General.....	61
8.3.2. Modelo Final de la Base de Datos.....	63
8.3.3. Desarrollo de la Primera Fase.....	64
8.3.4. Desarrollo de la Segunda Fase.....	85
8.3.5. Pruebas y Resultados.....	92
9. CONCLUSIONES.....	94
9.1. LIMITACIONES.....	94
9.2. TRABAJOS FUTUROS.....	95

Índice de Figuras

Figura 4.1: Arquitectura Cliente-Servidor.....	24
Figura 4.2: Modelo Petición-Respuesta Utilizando el Protocolo HTTP.....	27
Figura 5.1: Demostración en Línea de Open Conference System.....	30
Figura 5.2: Demostración en Línea de ConfTool.....	32
Figura 5.3: Demostración en Línea de OpenConf.....	34
Figura 5.4: Demostración en Línea de EasyChair en el Módulo Smart CFP	36
Figura 7.1: Diagrama Sintaxis CSS.....	43
Figura 7.2: Diagrama de Independencia de Plataforma de una Aplicación Java.....	48
Figura 7.3: Composición de Java Platform Standard Edition.....	50
Figura 7.4: Arquitectura de Spring Framework.....	51
Figura 8.1: Diagrama de Contexto del Sistema.....	62
Figura 8.2: Diagrama de Contenedores del Sistema.....	63
Figura 8.3: Modelo Final de la Base de Datos.....	64
Figura 8.4: Prototipo de Registro de Usuario	66
Figura 8.5: Prototipo de Inicio de Sesión de Usuario.....	67
Figura 8.6: Prototipo de Página de Inicio de Usuario Autenticado.....	68
Figura 8.7: Prototipo de Página de Inicio de Administrador Autenticado	68
Figura 8.8: Prototipo de Recuperación de Contraseña	69
Figura 8.9: Prototipo de Error Recuperación de Contraseña. Correo Inválido	69
Figura 8.10: Prototipo de Restablecimiento de Contraseña	70
Figura 8.11: Prototipo de Cambio de Contraseña	71
Figura 8.12: Prototipo de Reservación de Entradas a Conferencia como Asistente a Conferencia Paga.....	72
Figura 8.13: Prototipo de Reservación de Entradas a Conferencia como Autor a Conferencia Paga.....	73
Figura 8.14: Prototipo de Reservación de Entrada a Tutorial como Asistente a Conferencia Paga.....	74
Figura 8.15: Prototipo de Reservación con Recibo de Pago.....	75
Figura 8.16: Prototipo de Reservación con Factura	76
Figura 8.17: Prototipo de Lista de Reservas.....	77
Figura 8.18: Prototipo de Detalle de Reserva	77
Figura 8.19: Prototipo de Envío de Información de Pago con Depósito.....	78
Figura 8.20: Prototipo de Envío de Información de Pago con Transferencia	78
Figura 8.21: Prototipo de Envío de Información de Pago Exonerado	79
Figura 8.22: Prototipo de Cancelación de Reserva	79
Figura 8.23: Prototipo de Visualización de Historial de Reservas.....	80
Figura 8.24: Prototipo de Administración de Usuarios	81
Figura 8.25: Prototipo de Administración de Ordenes	81
Figura 8.26: Prototipo de Administración de Certificados de Asistencia	83

Figura 8.27: Prototipo de Administración de Distintivos	84
Figura 8.28: Prototipo de Edición de Perfil de Usuario como Usuario Administrador	85
Figura 8.29: Prototipo de Edición de Perfil de Usuario como Usuario Regular	86
Figura 8.30: Prototipo de Creación y Edición de Conferencia.....	87
Figura 8.31: Prototipo de Edición de Costos de Asistencia y Descuentos de Conferencias	88
Figura 8.32: Prototipo de Edición de Costos de Autores y Descuentos de Conferencia	88
Figura 8.33: Prototipo de Administración de Contribuciones	89
Figura 8.34: Prototipo de Creación y Edición de Contribución	90
Figura 8.35: Prototipo de Administración de Tutoriales	91
Figura 8.36: Prototipo de Creación y Edición de Tutoriales	91
Figura 8.37: Resultado de Ejecución de Pruebas Automatizadas	92

Índice de Tablas

Tabla 4.1: Estilos de Arquitecturas más Comunes.....	23
Tabla 5.1: Requisitos Mínimos del Sistema Open Conference System.....	31
Tabla 5.2: Requisitos Mínimos del Sistema ConfTool.....	33
Tabla 5.3: Requisitos Mínimos del Sistema OpenConf	35
Tabla 5.4: Tabla Comparativa entre las Herramientas Disponibles en el Mercado	37
Tabla 8.1: Etapas de Registro de Usuario	67
Tabla 8.2: Etapas de Autenticación de Usuario	68
Tabla 8.3: Etapas de Autorización de Usuarios a Secciones de la Aplicación.....	69
Tabla 8.4: Etapas de Recuperación de Contraseña	70
Tabla 8.5: Etapas de Cambio de Contraseña	71
Tabla 8.6: Etapas de Reservación de Entrada a Conferencia como Asistente a Conferencia Paga.....	72
Tabla 8.7: Etapas de Reservación de Entrada a Conferencia como Autor a Conferencia Paga	73
Tabla 8.8: Etapas de Reservación de Entrada a Tutorial como Asistente a Conferencia Paga	75
Tabla 8.9: Etapas de Reservación con Recibo de Pago o Factura.....	77
Tabla 8.10: Etapas de Lista de Reservaciones	77
Tabla 8.11: Etapas de Envío de Información de Pago.....	79
Tabla 8.12: Etapas de Cancelación de Reserva.....	80
Tabla 8.13: Etapas de Visualizar Historial de Reservas	80
Tabla 8.14: Etapas de Administración de Usuario	81
Tabla 8.15: Etapas de Administración de Reservas.....	82
Tabla 8.16: Etapas de Administración de Certificados de Asistencia	84
Tabla 8.17: Etapas de Administración de Distintivos	85
Tabla 8.18: Etapas de Edición de Perfil de Usuario	87
Tabla 8.19: Etapas de Administración de Conferencias.....	87
Tabla 8.20: Etapas de Edición de Costos y Descuentos de Conferencia	89
Tabla 8.21: Etapas de Administración de Contribuciones	90
Tabla 8.22: Etapas de Administración de Tutoriales	92

1. Introducción

Las Tecnologías de Información y Comunicación (TIC) se han abierto camino en nuestra sociedad, permitiendo nuevas formas de colaboración, de automatización y optimización de procesos, de recopilación y compartición de información, e incluso nuevas formas de pensar y analizar la información para tomar mejores decisiones.

La automatización de procesos mediante computadores intenta dejar la intervención del hombre en segundo plano, de esta forma facilitando los procesos, liberando así a las personas de funciones rutinarias, obteniendo ahorro significativo de tiempo, dinero y esfuerzo, ofreciendo menor margen de error y un esquema de trabajo más controlado.

Internet, específicamente la *World Wide Web* (WWW), dio un gran auge a las tecnologías de información, ya que, al mantener un gran número de nodos interconectados, hizo posible el acceso a recursos remoto de una manera simple a través de la red. Este hecho hizo que las aplicaciones web hayan aumentado su popularidad de manera exponencial, logrando que muchas organizaciones prefieran apalancarse en estas tecnologías.

El comité organizador de conferencias en Venezuela como el Simposio Científico y Tecnológico en Computación (SCTC), y la Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa), tiene la necesidad de automatizar la organización y administración de estos eventos, basándose en las tecnologías de información y comunicación, para hacer estos eventos más accesibles desde un punto de vista comunicacional, y más rentable desde un punto de vista organizacional.

El objetivo de este proyecto de investigación es el desarrollo de una aplicación web, la cual se encargará de gestionar y automatizar varios de los procesos involucrados en la creación y administración de conferencias académicas. Este desarrollo tecnológico beneficiará no sólo a la comunidad que atiende a estas conferencias, sino potencialmente a otras comunidades que quieran hacer uso de esta herramienta para organizar conferencias académicas en su propia industria.

El presente documento se encuentra estructurado en los siguientes capítulos:

Capítulo 2 - El Problema: se realiza un planteamiento formal del problema que será estudiado, además de introducir los objetivos que se desean alcanzar, y la propuesta del Trabajo Especial de Grado según la base teórica desarrollada en los capítulos anteriores.

Capítulo 3 - Conferencias Académicas: en este capítulo se presentará los conceptos relacionados a los eventos y conferencias académicas. Se examinarán algunos ejemplos, entre ellos, una conferencia que se llevan a cabo por la Escuela de Computación de la

Universidad Central de Venezuela y una conferencia organizada por la Sociedad Venezolana de Computación.

Capítulo 4 – Estilos de Arquitecturas de Sistemas de Software: se describirán los fundamentos de las arquitecturas de software y se entrará en detalle acerca de la arquitectura Cliente-Servidor, elegida como arquitectura de sistema de software para este proyecto.

Capítulo 5 – Trabajos Relacionados: se presenta una comparación entre las aplicaciones presentes actualmente en el mercado, junto a sus características más relevantes para los objetivos de este proyecto.

Capítulo 6 – Marco Metodológico: en este capítulo se analiza la metodología de desarrollo de software AgilUs.

Capítulo 7 – Marco Tecnológico: se presenta una visión acerca de las tecnologías a ser usadas en el proyecto. Se hace una clara distinción de las tecnologías usadas para la aplicación del lado del servidor, como las tecnologías que se ejecutan del lado del cliente.

Capítulo 8 – Marco Aplicativo: se describe el Marco Aplicativo del proyecto.

Capítulo 9 – Conclusiones: este capítulo presenta las conclusiones de la investigación. También se presentan las recomendaciones para trabajos futuros.

2. El Problema

En este capítulo se realiza un planteamiento formal del problema que será estudiado. Se exponen los argumentos que justifican la creación de objetivos y alcances para obtener una solución eficaz.

2.1. Planteamiento del Problema

La administración de eventos o conferencias académicas se puede definir como el conjunto de procesos que se ejecutan para el diseño, planificación, y producción de dichos eventos. Para que un evento se lleve a cabo exitosamente, es necesario coordinar todas las actividades y aspectos técnicos como el presupuesto, selección de ubicación, transporte, equipos de comunicación, etc. Particularmente, para un evento como las conferencias académicas, adicionalmente es necesario gestionar aquellos procesos específicos como la gestión del proceso de las presentaciones, envíos de artículos, selección, registros, emisión de certificados de participación, verificación de pagos, etc.

En la actualidad, el comité organizador de conferencias como SCTC o CoNCISa, no cuenta con una aplicación que cubra sus necesidades de una manera automatizada, viéndose en la necesidad de ejecutar muchas de estas tareas de una manera manual, lo cual requiere un esfuerzo mucho mayor por parte de los organizadores de conferencias para poder llevar a cabo este tipo de eventos.

Los organizadores de estas conferencias, conformaron un Comité de Inscripción con el único propósito de gestionar el registro de los asistentes y autores a las conferencias. Este Comité de Inscripción se encuentra compuesto por 8 personas, aproximadamente, lo que genera un alto esfuerzo de personas dedicadas a esta sola tarea.

Al iniciar las conferencias, y luego de finalizar, los organizadores deben generar una serie de documentos digitales, tales como: los distintivos que se entregan a los participantes; los certificados de asistencia y participación; los recibos de pago, en el caso de conferencias de tipo CoNCISa. Estas tareas se llevan a cabo de manera manual, lo cual impone un alto esfuerzo por parte del personal para generar estos documentos. También tienen un alto costo monetario, por los insumos usado en las impresiones de estos documentos.

Asimismo, para las conferencias de tipo CoNCISa, donde se realizan pagos por la asistencia a las conferencias, los datos de los pagos se capturan por medios como el correo electrónico o llamadas telefónicas. Adicionalmente, los usuarios deben hacer el cálculo del monto que deben pagar, tomando en cuenta los eventos que desean asistir. Este proceso es propenso a errores debido a un mal cálculo por parte de los usuarios

que impacta los tiempos de los organizadores al tener que resolver estos inconvenientes.

Todas estas tareas se pueden automatizar para facilitar el proceso de organización de las conferencias académicas. Las herramientas consideradas en este proyecto, no permiten cumplir con las necesidades de los organizadores para una gestión eficiente y eficaz de estas tareas administrativas.

Tomando en cuenta esta situación que se presenta, se desarrollará una aplicación web moderna, funcional, usable, que facilite a los organizadores de eventos disminuir el esfuerzo y al mismo tiempo ofrecer un servicio de calidad en cuanto a la preparación y desarrollo de eventos académicos se refiere.

2.2. Justificación

Si bien existen, hoy en día, aplicaciones que pueden ser usadas para la organización de estas conferencias, estas gestionan el ciclo de vida de los artículos y no se ajustan a las necesidades de los organizadores de las conferencias académicas en relación a los procesos administrativos.

Asimismo, ofrecen limitaciones de funcionalidades, o limitaciones por Licencias de Uso. También, varias requieren realizar pagos para su uso y debido a la situación económica que atraviesa Venezuela, los organizadores no quieren incurrir en ningún tipo de gasto por el uso de estas aplicaciones.

Bajo esta situación, surge la necesidad de una aplicación que no sólo se ajuste a las necesidades particulares de los organizadores de las conferencias, sino que también sea amigable al usuario, basada en tecnologías mundialmente reconocidas que facilite su mantenimiento, y que sea de fácil instalación, configuración, y ejecución.

2.3. Objetivos

A continuación, se presentan los objetivos trazados para este proyecto.

2.3.1. Objetivo General

Diseñar y desarrollar una aplicación web para la administración de conferencias académicas.

2.3.2. Objetivos Específicos

1. Identificar y estudiar los requerimientos de la Organización de Eventos Académicos llevados a cabo por los organizadores de las conferencias.
2. Definir la arquitectura de la solución.
3. Diseñar el esquema de la base de datos.
4. Modelar e implementar prototipos de la solución basada en los requerimientos.
5. Desarrollar los prototipos de manera incremental e iterativa para generar el producto final de acuerdo a las especificaciones.
6. Creación, configuración, y ejecución de pruebas automatizadas.
7. Diseño y ejecución de pruebas funcionales de aceptación.

2.4. Alcance

A continuación, se define el alcance del proyecto:

1. Automatizar los procesos administrativos de las dos conferencias académicas:
 - a. Simposio Científico y Tecnológico en Computación (SCTC)
 - b. Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa)
2. Crear una aplicación web multiplataforma, de fácil instalación y despliegue.
3. Desplegar la aplicación en ambientes de producción.

3. Conferencias Académicas

Las conferencias son eventos sociales cuyo objetivo es juntar a las personas, bien sea cara-a-cara o virtualmente a través de la red, para intercambiar ideas e información; para discutir, y en algunos casos, negociar; para construir relaciones personales y profesionales; para fomentar un mejor desempeño de los individuos y organizaciones. El término usado para identificar estos eventos puede variar (Cumbre, Reunión, Conferencia, Asamblea, Convención, Congreso, etc.), y cada uno de esos eventos puede tener diferentes formatos y énfasis, pero los ingredientes esenciales y los objetivos son los mismos [1].

Las conferencias académicas son conferencias para presentar y discutir trabajos de investigación. Estas conferencias proveen un canal importante para el intercambio de información entre investigadores y expertos. Las presentaciones en estas conferencias son seguidas, generalmente, por discusiones donde los participantes intercambian sus reacciones y críticas. Estas interacciones distinguen a las conferencias académicas de otros tipos de presentaciones escolares donde no existe interacción.

En general, un trabajo es comunicado en forma de una presentación corta y concisa con una duración entre diez (10) y treinta (30) minutos. Luego de la presentación, inicia la fase de discusión acerca del tema expuesto. El trabajo puede ser revisado y publicado en las Memorias de la Conferencia (*Conference Proceedings*). Paneles de discusión, discusiones de mesa redonda, o talleres pueden ser parte de la conferencia. Los candidatos a presentadores en una conferencia usualmente deben enviar un resumen (*Abstract*) del trabajo o el trabajo en extenso, que será revisado por el Comité del Programa antes de aceptar o no la presentación en la conferencia.

Las conferencias pueden ser de presentación Simple o Múltiple, donde una Simple tiene una única presentación ejecutándose a la vez, mientras que en una Múltiple existen varias presentaciones en paralelo con presentadores en diferentes salas, al mismo tiempo. Las conferencias académicas se pueden categorizar de la siguiente manera [2]:

- Conferencias organizadas y basadas en un tema particular.
- Conferencias generales con sesiones o presentaciones de una amplia variedad de temas. Este tipo de conferencias suelen ser organizadas por sociedades regionales, nacionales, o internacionales, y se llevan a cabo de manera regular.
- Conferencias profesionales que no son limitadas a la sociedad académica, pero si relacionada con ella.

3.1. Simposio Científico y Tecnológico en Computación

El Simposio Científico y Tecnológico en Computación (SCTC) es una conferencia académica general organizado por la Escuela de Computación de la Universidad Central de Venezuela. Esta se lleva a cabo de manera bienal, y tiene como finalidad consolidar el intercambio sobre experiencias investigativas, académicas, y tecnológicas, para crear lazos estrechos de cooperación a nivel nacional e internacional [3]. El SCTC se ha caracterizado por una filosofía de gratuidad para el público, donde todas sus actividades se ofrecen sin costo alguno. Esta filosofía está motivada por el pensamiento de que es primordial que la transferencia del conocimiento sea libre y se haga sin costo. Este beneficio no es sólo para el público en general, sino que también se extiende a los autores de las contribuciones aceptadas donde no tienen costo de registro.

Adicionalmente a las presentaciones de las contribuciones investigativas, el SCTC ofrece la posibilidad de impartir Conferencias Plenarias, Talleres, Tutoriales, Cursos Cortos, Feria de Empleo, entre otros.

El actual trabajo de investigación busca satisfacer las necesidades de administración de los eventos como el SCTC, a manera de simplificar la organización y gestión de este tipo de eventos tan beneficiosos para la comunidad.

3.2. Conferencia Nacional de Computación, Informática y Sistemas

La Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa) es uno de los principales eventos anuales de la Sociedad Venezolana de Computación (SVC). Esta conferencia cuenta con el respaldo académico del Centro Latinoamericano de Estudios en Informática (CLEI) y tiene como finalidad consolidar el intercambio de experiencias de investigación, académicas, y tecnológicas, para impulsar el desarrollo del área de computación y crear lazos estrechos de cooperación a nivel nacional [4].

En el año 2017, CoNCISa se llevó a cabo conjuntamente con la Escuela Venezolana de Informática (EVI). En este último evento se dictaron cursos avanzados en diferentes dominios de las ciencias computacionales, por expertos nacionales e internacionales. Son cursos que cubren temas de interés para la disciplina Informática, y que son complementarios a los temas que figuran en los currículos de las carreras de Informática en el país. Son divididos en cursos tutoriales de medio día (4 horas) o de días completo (8 horas) [4].

CoNCISa se sustenta bajo un modelo de cobro por registro tanto para autores como para asistentes. Ofrecen un rango de tarifas para los diferentes tipos de asistentes y autores.

Asimismo, ofrecen tarifas con descuento para aquellos que se inscriban temprano en el proceso.

El actual trabajo de investigación, también tomará en cuenta las necesidades de administración de CoNCISa y EVI a la hora del desarrollo de la aplicación.

4. Estilos de Arquitecturas de Sistemas de Software

Un Estilo de Arquitectura, a veces llamado Patrón Arquitectónico, es un conjunto de principios que proporciona un marco de trabajo abstracto para una familia de sistemas [5]. Más específicamente, un estilo arquitectónico determina el vocabulario de componentes y conectores que pueden ser usados en las instancias de ese estilo, junto a un conjunto de restricciones acerca de cómo pueden combinarse. Estas pueden incluir restricciones topológicas en la descripción de la arquitectura (*e.g.*, sin ciclos). Otras restricciones, digamos relacionadas con la semántica de la ejecución, pueden también ser parte de la definición del estilo [6]. La Tabla 4.1 (tomada de [5]), muestra una breve descripción de varios estilos de arquitecturas.

Estilo	Descripción
Cliente-Servidor	Segrega el sistema en dos aplicaciones, donde el cliente hace peticiones al servidor.
Basada en componentes	Descompone el diseño de la aplicación en componentes funcionales o lógicos reusables.
Diseño orientado por el Dominio (DDD)	Estilo enfocado en modelar el dominio de negocio y definir objetos de negocios basados en entidades dentro del dominio.
Arquitectura en Capas	Segrega las responsabilidades de la aplicación en grupos apilados (capas).
Bus de Mensajes	Estilo que especifica el uso de un sistema de software que reciba y envíe mensajes usando uno o más canales de comunicación, permitiendo a la aplicación interactuar sin saber detalles específicos acerca de los otros sistemas.
N-Niveles	Segrega la funcionalidad en segmentos separados similar a la Arquitectura en Capas, pero cada segmento estará ubicado físicamente en nodos separados.
Orientado a Objetos	Un paradigma de diseño basado en la división de responsabilidades de una aplicación o sistema en objetos autosuficientes y reusables.

Tabla 4.1: Estilos de Arquitecturas más Comunes

En las secciones siguientes, se entrará en más detalle acerca del estilo de arquitectura cliente-servidor.

4.1. Arquitectura Cliente-Servidor

La arquitectura cliente-servidor se puede definir como una arquitectura distribuida que permite a usuarios finales obtener acceso a un conjunto de información mediante peticiones entre procesos.

El término cliente-servidor es usado para describir un modelo computacional basado en la distribución de funciones entre dos tipos de procesos independientes y autónomos: un servidor y un cliente. El cliente es cualquier proceso que solicita servicios específicos del proceso servidor. El servidor es un proceso que provee servicios solicitados por los clientes. Cliente y servidor pueden residir en el mismo computador o en diferentes computadores conectados a través de la red [7].

La Figura 4.1, muestra la clásica configuración de los clientes y servidores en esta arquitectura.

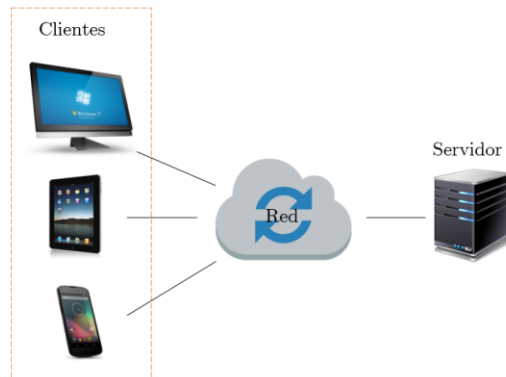


Figura 4.1: Arquitectura Cliente-Servidor

La principal ventaja de esta arquitectura es que facilita la separación de las funciones según su servicio, permitiendo situar cada función en la plataforma más adecuada para su ejecución. Asimismo, presenta las siguientes ventajas [8]:

- Las redes de computadores permiten que múltiples procesadores puedan ejecutar partes distribuidas de una misma aplicación, logrando concurrencia de procesos.
- Existe la posibilidad de migrar aplicaciones de un procesador a otro con modificaciones mínimas en los programas.
- Posibilita el acceso a los datos independientemente de donde se encuentre el usuario.
- Se obtiene escalabilidad de las aplicaciones tanto de manera horizontal como vertical. La **escalabilidad horizontal** es la capacidad de añadir o suprimir estaciones de trabajo o clientes que hagan uso de la aplicación, sin que afecte sustancialmente el rendimiento general. La **escalabilidad vertical** se refiere a la capacidad de migrar hacia servidores de mayor capacidad o velocidad, o de un tipo distinto de arquitectura sin que afecte a los clientes.

Normalmente el proceso servidor es responsable de proveer a múltiples clientes un único tipo de servicio definido. Por tal razón, existen múltiples servidores para diferentes necesidades de los clientes.

4.1.1. Servidor de Archivos

Un servidor de archivos provee a los clientes acceso a los archivos que residen en el servidor. Este tipo de servidores son útiles para compartir archivos a través de la red.

4.1.2. Servidor de Correos Electrónicos

Proceso que gestiona el flujo de correos electrónicos, mensajería, y comunicaciones entre sistemas o usuarios finales.

4.1.3. Servidor Web

Servidor que ofrece información como documentos, imágenes, datos, etc. a sus clientes. Los clientes usualmente son procesos livianos como un navegador web, que puede comunicarse con el servidor usando el protocolo más comúnmente usado en la Web: HTTP¹.

4.1.4. Servidor de Base de Datos

Datos que residen en el servidor, almacenados de manera estructurada o no estructurada son expuestos a los clientes a través de servicios. Estos servicios pueden ser accedidos a través de interfaces particulares de cada servidor, frecuentemente usando SQL para las bases de datos relacionales, o algún otro mecanismo, posiblemente propietario, en el caso de bases de datos no relacionales (*e.g., NoSQL*).

4.2. Cliente-Servidor por Carga de Procesamiento

Un cliente o un servidor es categorizado dependiendo por la cantidad de procesamiento que debe ejecutar en comparación a su contraparte (servidor o cliente). Estas categorías son denominadas *Liviana* o *Pesada*.

Un cliente liviano es aquél que lleva a cabo un procesamiento mínimo, mientras que el servidor pesado es el que realiza la mayor carga de procesamiento. Este concepto de liviano vs. pesado es determinado por un criterio importante dentro de la arquitectura

¹ Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*, por sus siglas en inglés)

de la aplicación: ¿Qué tanta funcionalidad está ubicada del lado del cliente vs. del lado del servidor?

Clientes Pesados: dentro de esta arquitectura, el cliente tiene asignada más trabajo o funcionalidades. Eso disminuye la carga sobre el servidor, pero puede ser difícil el mantenimiento de los clientes cuando el número es alto [7].

Servidores Pesados: en este tipo de arquitectura se les coloca mayor número de funcionalidades al computador que ofrece el servicio. Esta arquitectura requiere un uso de recursos más eficiente y mayor capacidad de procesamiento, ya que el mismo servidor puede servir potencialmente a un número alto de clientes. Al tener gran parte de la aplicación centralizada, facilita el mantenimiento de la misma, a costo de necesitar un hardware con mayores capacidades para servir a la misma cantidad de clientes que podría servir un servidor liviano.

4.3. Cliente-Servidor por Manejo de Estado de la Aplicación

Un servidor “Sin Estado” procesa todas sus peticiones como transacciones independientes que no guardan relación alguna con cualquier petición previa. Una ventaja de un servidor “Sin Estado” es la simplificación de la implementación de los servidores al no tener que mantener registro de las conversaciones llevadas a cabo.

Un ejemplo claro de un servidor “Sin Estado” es un servidor Web (HTTP). El protocolo HTTP es, por diseño, un protocolo “Sin Estado” [9], el cual debe ser procesado independientemente de otras peticiones HTTP.

Un servidor “Con Estado” es aquél que mantiene información del cliente y de la conversación (serie de peticiones y respuestas realizadas). Este comportamiento permite manejar las peticiones de una manera más eficiente al sólo tener que intercambiar datos de menor tamaño, pues el servidor puede almacenar datos acerca de las peticiones que el cliente ha realizado anteriormente. Entre las desventajas se puede describir la complejidad de invalidar estas sesiones con el cliente. Adicionalmente, el aumento en el uso de recursos del servidor para poder almacenar la información adicional de cada cliente con quien se inicia una conversación.

Un servidor del protocolo FTP es un ejemplo de un servidor “Con Estado”. Las solicitudes de archivos a este servidor, asume que hubo una previa autenticación del usuario para acceder a dicho recurso.

4.4. Aplicación Web como Implementación de la Arquitectura Cliente-Servidor

Una aplicación web es un software que se codifica en un lenguaje soportado por los navegadores web, en la que se confía la ejecución al navegador y son un tipo especial de aplicaciones cliente-servidor, donde el cliente (generalmente un navegador o explorador²) realiza peticiones a un servidor de aplicaciones utilizando el Protocolo de Transferencia de Hipertexto (HTTP), el cual es usado en cada transacción del *World Wide Web* (WWW). El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son empleados en Internet. Este protocolo desarrollado por la IETF³ está orientado a transacciones y sigue el esquema petición respuesta entre un cliente y un servidor. La Figura 4.2, muestra el modelo lógico de petición respuesta utilizando el protocolo HTTP.

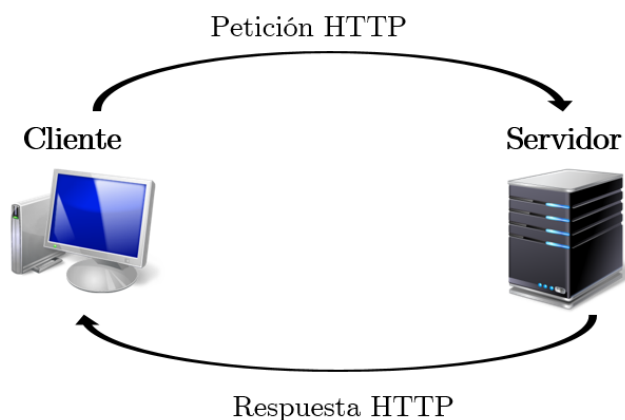


Figura 4.2: Modelo Petición-Respuesta Utilizando el Protocolo HTTP

Las aplicaciones web son independientes del sistema operativo del cliente. Una vez desplegadas, procesan las peticiones del cliente en el servidor y envían las respuestas de una manera consistente, sin importar donde éstas sean ejecutadas, mientras cumplan con las especificaciones estándares de los navegadores web.

Las aplicaciones web suelen estar estructuradas en 3 capas o niveles: el nivel superior que interactúa con el usuario (normalmente un navegador web), el nivel inferior que proporciona los datos (generalmente una base de datos), y un nivel intermedio que procesa los datos y utiliza alguna tecnología web dinámica, como por ejemplo: Java, ASP, PHP, Python, o Ruby on Rails, entre muchos otros lenguajes [8]. De acuerdo a los

² El navegador es un software que permite el acceso a Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos.

³ El *Internet Engineering Task Force* (Grupo de trabajo de Ingeniería de Internet) es una organización internacional abierta de normalización, que tiene como objetivos el contribuir con la ingeniería de Internet.

niveles de una aplicación web, y en base a la arquitectura cliente-servidor, un usuario realiza una petición mediante un cliente web especificando la dirección de la página que desea consultar. Este recurso se identifica mediante un URL⁴.

Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta de base de datos, entre muchos otros. Una vez el servidor procese la información, él mismo se encargará de enviar dicha página o dato al cliente. Si se trata de una página HTML, el cliente se encarga de interpretar el código y mostrarlo al usuario. Si el cliente web encuentra instrucciones que hacen referencia a otros objetos (imágenes, sonidos, animaciones, etc.), establece automáticamente comunicación con el servidor web para solicitar dicha información.

Una vez finalizado este proceso, se cierra la conexión hasta que el cliente realice alguna otra petición. Durante este proceso, el usuario tendrá la sensación de mantener una conexión permanente debido a que normalmente las aplicaciones web son aplicaciones “Con Estado”. Esto contrasta con el servidor que sirve dicha aplicación, que, siguiendo la naturaleza del protocolo HTTP, es en servidor “Sin Estado”.

En conclusión, la capa de aplicación debe ejecutar lógica adicional para mejorar la experiencia del usuario al mantener un estado.

⁴ *Uniform Resource Locator* (Localizador Uniforme de Recursos)

5. Trabajos Relacionados

En este capítulo se presentan una serie de trabajos relacionados con aplicaciones de administración de conferencias. También, se presentan comparaciones entre algunas de las distintas soluciones presentes en el mercado en la actualidad.

5.1. Open Conference System

Esta aplicación fue publicada en el año 2000 como una aplicación de código abierto [10]. Permite administrar conferencias y artículos a través de la web.

Se distribuye de manera gratuita para ser usada en instalaciones locales. También ofrecen la aplicación bajo el modelo *SaaS*⁵, alojada directamente en los servidores de la compañía. En la Figura 5.1, se aprecia una captura de pantalla de la demostración en línea que ofrece *Open Conference System*.

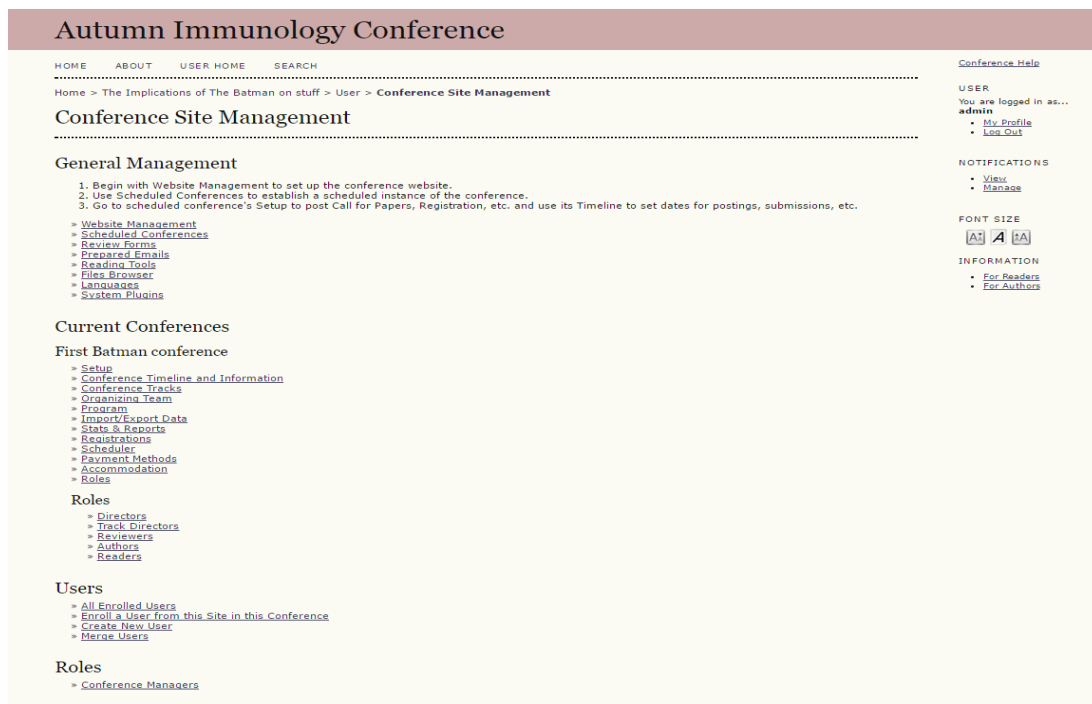


Figura 5.1: Demostración en Línea de Open Conference System

Características Principales

Sus principales características son:

⁵ *Software as a Service*

- Generación de un Sitio Web para conferencias
- Composición y envío de CFPs (*Call For Papers*)
- Recepción, evaluación, y aceptación de artículos.
- Registro de participantes

Permisos y Licencias

Open Conference System se distribuye de manera gratuita bajo la licencia *GNU General Public License* (GPL) versión 2.

Requerimientos Mínimos de Sistema

El único requerimiento que se debe cumplir para la versión alojada en la nube, es un navegador compatible con la aplicación.

Para la versión descargable, la Tabla 5.1 describe los requisitos mínimos:

Requisito	Descripción
Servidor HTTP	<ul style="list-style-type: none"> • Apache versión 1.3.2x o superior • Apache versión 2.0.4x o superior • Microsoft IIS 6 o superior.
Plataforma de aplicación	Requiere PHP: <ul style="list-style-type: none"> • versión 4.2.o superior para servidor Apache • versión 5.x o superior para Microsoft IIS
Capa de persistencia	<ul style="list-style-type: none"> • MySQL versión 3.23.23 o superior. • PostgreSQL 7.1 o superior.

Tabla 5.1: Requisitos Mínimos del Sistema Open Conference System

5.2. ConfTool

ConfTool es una herramienta multi-idioma basada en la web para la administración de eventos enfocados a conferencias académicas, talleres, congresos, y seminarios. Esta herramienta se presenta en dos versiones para suplir diferentes necesidades:

VSIS ConfTool

Es la versión gratuita con funciones básicas o limitadas, orientada a eventos pequeños, no comerciales, como los eventos académicos. Esta se distribuye sin ningún soporte y para instalación local en un servidor web.

ConfTool Pro

Viene con funcionalidades adicionales y sin limitaciones. Se ofrece bajo modalidad alojada directamente en los servidores de la compañía, bajo un modelo SaaS. En la

Figura 5.2, se observa una captura de pantalla de la demostración en línea que ofrece *ConfTool*.

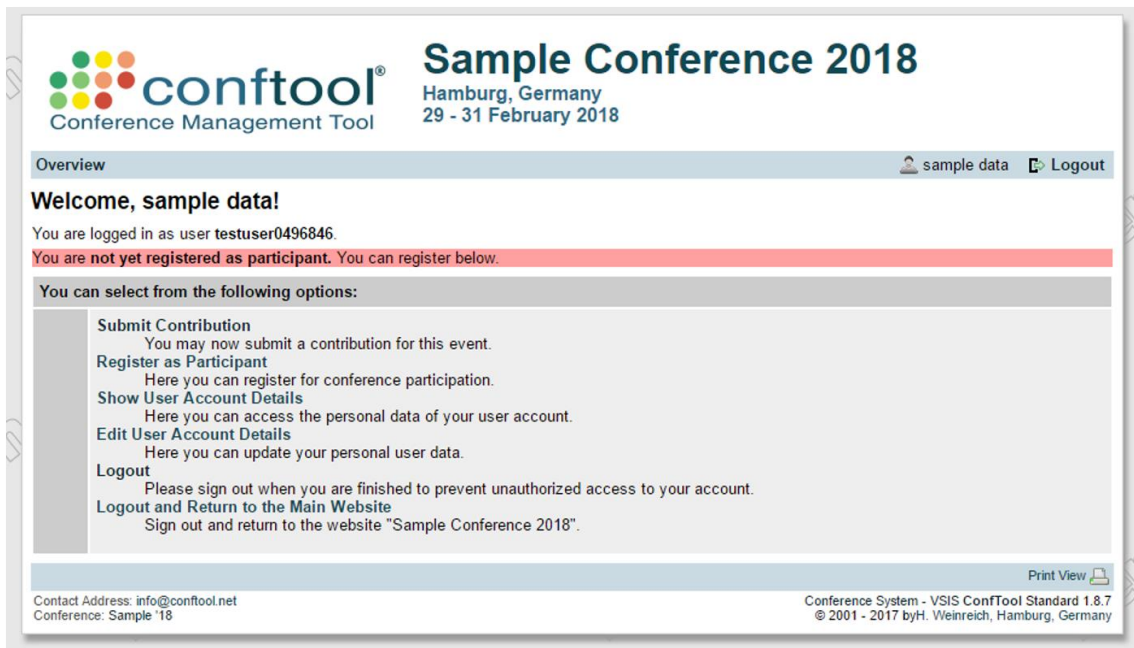


Figura 5.2: Demostración en Línea de ConfTool

Características Principales

Las características más destacadas de ConfTool, independientemente de la versión, son:

- Interfaz multi-idioma.
- Registro en línea de participantes.
- Foro en línea.
- Envío y revisión de artículos.
- Múltiples métodos de pago.
- Integración con pasarelas de pagos como PayPal.

Permisos y Licencias

ConfTool se distribuye como un paquete binario que contiene la aplicación. Este se distribuye bajo dos diferentes licencias:

- Licencia *Freeware*, para la versión VSIS ConfTool.
- Licencia Comercial, para la versión ConfTool Pro.

Requerimientos Mínimos del Sistema

La Tabla 5.2 describe los requisitos mínimos:

Requisito	Descripción
Servidor HTTP	Apache versión 2.0 o superior
Plataforma de aplicación	PHP versión 5.3 o superior, con módulos <i>curl</i> , y <i>mbstring</i> habilitados.
Capa de persistencia	MySQL versión 5.1 o superior
Otros	Acceso a servidor SMTP y DNS

Tabla 5.2: Requisitos Mínimos del Sistema ConfTool

5.3. Conference Management System (COMS)

Solución que ofrece una plataforma web para la administración de eventos a lo largo de su ciclo de vida. Se ofrece como producto propietario basado en los servidores de la compañía, bajo un modelo de pago por cuentas de usuarios. Asimismo, ofrecen funcionalidades opcionales que pueden irse agregando de ser necesario.

Características Principales

Es una herramienta sumamente completa en cuanto a funcionalidades se refiere. Entre sus principales características se encuentran:

- Interfaz multi-idioma.
- Registros de usuarios.
- Registro y seguimiento de pagos.
- Selección de artículos.
- Administración de eventos

Permisos y Licencias

Esta aplicación se ofrece bajo licencia de pago, basada en pago-por-conferencia, alojada directamente en los servidores de la compañía.

Requerimientos Mínimos del Sistema

El único requerimiento que se debe cumplir para la versión alojada en la nube, es tener un navegador compatible con la aplicación.

5.4. OpenConf

Es una aplicación web que ofrece tres ediciones diferentes para cubrir diferentes necesidades. Fue nombrada como *partner* preferido por el IEEE [11].

OpenConf Community Edition

Esta edición ofrece de manera gratuita las funcionalidades más básicas: envío online, revisiones, aceptaciones, notificaciones, entre otros.

OpenConf Plus Edition

Incluye todas las funcionalidades de la edición *Community* más otras adicionales como: soporte técnico, carga de archivos, discusiones de comité, entre otros. Esta edición se ofrece bajo previa compra de licencia y se instala en los servidores propios del comprador.

OpenConf Professional Edition

Es la versión más avanzada y completa, donde se ofrecen docenas de módulos que extienden las funcionalidades. Además de incluir las funcionalidades de la edición Plus, la edición Professional incluye: alojamiento en servidores bajo modalidad SaaS, personalización de formularios, aceptación de pagos por envíos de artículos, entre otros.

En la Figura 5.3, se observa una captura de pantalla de la demostración en línea que ofrece *OpenConf*.

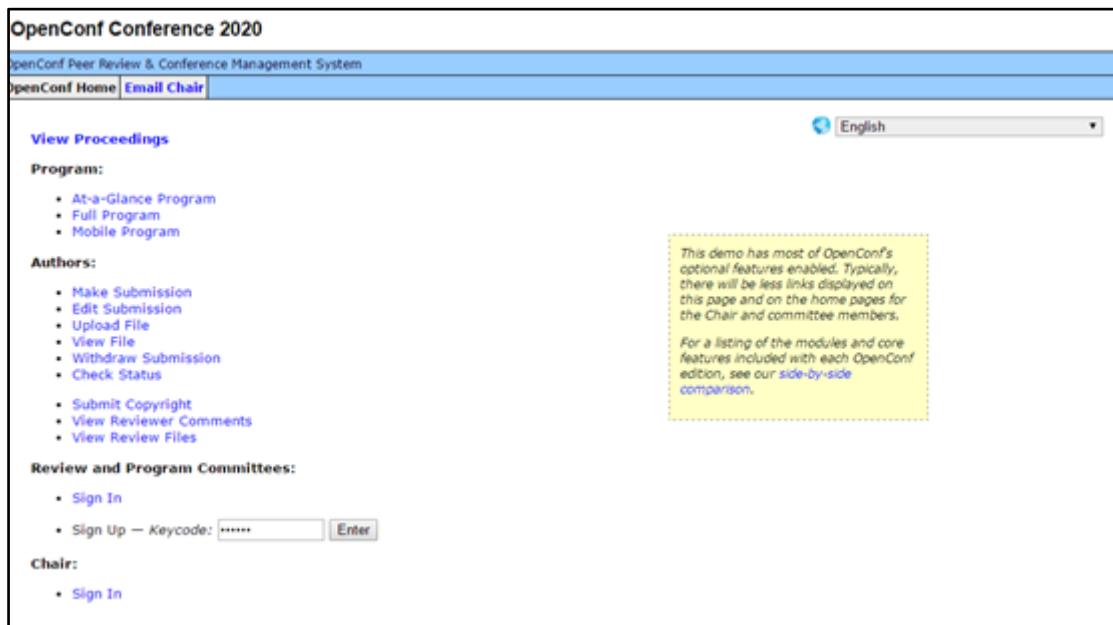


Figura 5.3: Demostración en Línea de OpenConf

Características Principales

Incluyendo todas las ediciones de OpenConf, a continuación, se presentan las características más relevantes:

- Envíos de artículos online.
- Carga de archivos.
- Revisiones de envíos.
- Notificaciones vía email.
- Exportación de datos a formatos CSV, XML, Excel, SQL.
- Chequeo de plagio.

Permisos y Licencias

OpenConf se ofrece bajo dos tipos de licencias para sus tres ediciones:

Community Edition: sólo tiene permiso de ser usado en los servidores propios del usuario, mientras no se generen o reciban pagos relacionados con la aplicación.

Plus Edition y Professional Edition: su uso se hace bajo previa compra de la licencia que permite una única instalación. Cada instalación adicional requiere la compra de una nueva licencia.

Requerimientos Mínimos del Sistema

La Tabla 5.3 describe los requisitos mínimos:

Requisito	Descripción
Servidor HTTP	Cualquiera con soporte PHP
Plataforma de aplicación	PHP versión 5.3.7 o superior, con módulos <i>json</i> , <i>mysqli</i> , <i>mycrypt</i> , y <i>zip</i> , habilitados.
Capa de persistencia	<ul style="list-style-type: none"> • MySQL versión 5.0 o superior • MariaDB 5.0 o superior

Tabla 5.3: Requisitos Mínimos del Sistema OpenConf

5.5. EasyChair

EasyChair es un sistema de gestión de conferencias flexible, fácil de usar, y tiene muchas características que lo hacen idóneo para varios modelos de conferencias. Es probablemente el sistema de gestión de conferencias más usado [12]. Se ofrece bajo el modelo SaaS con características dependientes de la licencia. En la Figura 5.4, se observa una captura de pantalla de la demostración en línea que ofrece EasyChair.

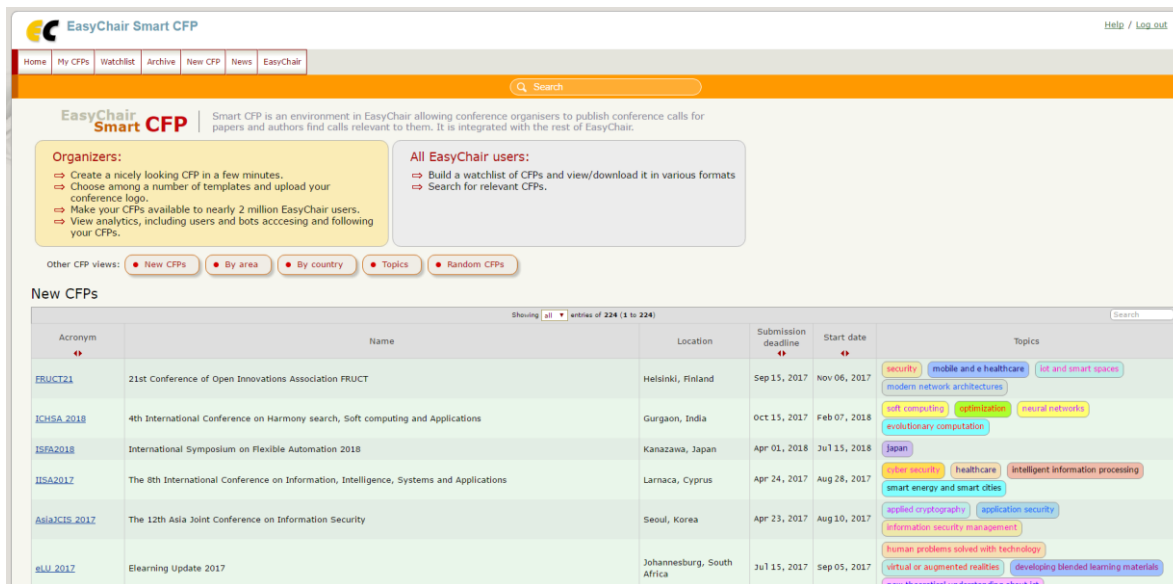


Figura 5.4: Demostración en Línea de EasyChair en el Módulo Smart CFP

Actualmente, EasyChair soporta dos modelos:

- 1) El modelo estándar destinado a conferencias que tienen un solo Comité de Programa. Los artículos se distribuyen entre los miembros del comité, usualmente basado en sus preferencias.
- 2) La versión *multi-tracks* destinada a multi-conferencias, es decir, un evento donde se desarrollan en paralelo varias conferencias. Cada conferencia en tiene su propio Comité de Programa. En otras palabras, hay una cierta independencia entre una conferencia y otra. Se requiere hacer un pago para tener el soporte *multi-tracks*.

Características Principales

Entre las características más relevantes de esta solución se pueden mencionar las siguientes:

- Gestión y monitoreo del Comité de Programa.
- Gestión de los accesos de los miembros del comité y conflictos de interés.
- Envío y revisión de artículos.
- Asignación de artículos basados en las preferencias de los miembros del comité
- Lista de los últimos eventos.
- Preparación de las memorias de la conferencia.
- Generación y publicación del programa de la conferencia.
- Generación y publicación de folletos con el programa.
- Discusión en línea de los artículos.

Permisos y Licencias

EasyChair soporta tres tipos diferentes de licencias: *Free*, *Professional*, y *Executive*. Cada una de estas licencias tienen diferentes costos, limitaciones, y características disponibles⁶. Por ejemplo, la versión *Free* es gratuita, y no tiene soporte para multi-conferencias. Sin embargo y a pesar de sus limitaciones, la versión *Free* es, en general, suficiente para conferencias pequeñas.

Requerimientos Mínimos del Sistema

El único requerimiento que se necesita es un navegador compatible con la aplicación, ya que el sistema en sí está mantenido por el Departamento de Ciencias de la Computación de la Universidad de Manchester.

5.6. Comparativa de Características

Habiendo evaluado varias de las soluciones disponibles en el mercado relacionadas con el actual trabajo de investigación, a continuación, se presenta la Tabla 5.4, donde se muestra una comparación directa entre estas herramientas tomando en cuenta las características relevantes para el trabajo de investigación.

Característica	OCS	ConfTool	COMS	OpenConf	EasyChair
Licencia gratuita	✓	✓ ⁷	✓ ⁷	✓	✓ ⁷
Fácil ejecución		SaaS	SaaS	SaaS	
Registro de usuario en línea	✓	✓	✓	✓	✓
Emisión de llamado a artículos	✓	✓	✓	✓	✓
Recepción y revisión de artículos en línea	✓	✓	✓	✓	✓
Discusiones en línea de las presentaciones				✓	✓
Generación de certificados digitales de asistencia					
Generación de distintivos					
Administración de reservas a conferencias tipo SCTC					
Captura de datos de pago					
Administración de descuentos		✓			

Tabla 5.4: Tabla Comparativa entre las Herramientas Disponibles en el Mercado

⁶ <http://easychair.org/licenses.cgi>

⁷ Con limitaciones.

6. Marco Metodológico

Para alcanzar los objetivos planteados en el Capítulo 2, es necesario definir un esquema o metodología de trabajo que permita el desarrollo coherente de cada uno de los requerimientos de la aplicación. A continuación, se presenta la especificación de la metodología usada y otros detalles importantes que fueron tomados en cuenta para el desarrollo e implementación de la aplicación.

6.1. Adaptación de la Metodología de Desarrollo AgilUs

El método AgilUs es un método de desarrollo ágil creado en el Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación, de la Universidad Central de Venezuela. Se basa principalmente en el concepto de usabilidad y en la necesidad de desarrollar software usable. Este método de desarrollo de software pone mayor peso en la construcción de interfaces de usuario, la cual no debe ser una adición meramente estética al final del desarrollo, sino por el contrario, el desarrollo de las interfaces de usuario debe guiar las decisiones en la Ingeniería del Software [13].

En AgilUs, son los usuarios, no el cliente ni los programadores, quienes guían el desarrollo del proyecto. Este método busca proporcionar un conjunto de actividades organizadas para construir la usabilidad en el diseño de interfaces de usuario durante el desarrollo de un producto de software.

El enfoque principal de AgilUs es la usabilidad. El proceso de desarrollo de software engloba todas las actividades del proyecto, así como los requisitos, análisis, prototipaje, y entrega del mismo. Asimismo, incluye las evaluaciones de la usabilidad en cada etapa del proceso. Cualquier proyecto que se diseñe debe estar adaptado para ser funcional de una manera eficiente y así, de esta manera, se cubren los requerimientos de los usuarios.

6.2. Ciclo de Vida

Es importante destacar que toda metodología presenta un ciclo de vida. Este ciclo va a estar marcado por la usabilidad y debe estar incluido en cada etapa del desarrollo del proyecto. Estas etapas pueden mejorarse con las observaciones realizadas por los usuarios. Por ello, es importante que se mantenga un prototipo y cada evaluación de usabilidad sea medida por cada desarrollador para generar la menor cantidad de errores en el desarrollo del software.

6.2.1. Requisitos

Se lleva a cabo el levantamiento de requerimientos, generando un perfil de usuario. Cada requerimiento definido se deberá desarrollar e implementar en las etapas subsecuentes, para cumplir con las exigencias de los usuarios.

6.2.2. Análisis

Una vez estipulados los requisitos que persigue el sistema, se debe realizar un análisis sobre los mismos para determinar de qué manera serán implementados apoyándose en el uso de diagramas.

6.2.3. Prototipaje

Son elaborados diversos prototipos, incluyendo un prototipo de Interfaz Gráfica de Usuario o GUI (*Graphical User Interface*), facilitado por los diagramas de interacción. En esta etapa, el usuario lleva a cabo la evaluación de los prototipos a fin de conocer si los mismos cumplen con los principios de usabilidad. Para realizar estas inspecciones de usabilidad, se lleva a cabo la elaboración de listas de comprobación, las cuales se utilizan para juzgar los atributos y los métodos de interacción de la interfaz de usuario; también se elaboran guías de estilo, con las cuales se determinan las normativas en relación al aspecto de la interfaz; por último, se lleva a cabo una evaluación heurística, que consiste en un análisis de los aspectos de usabilidad para determinar si se encuentran incluidos en cada elemento de la interfaz de usuario.

6.2.4. Entrega

Consiste en la aplicación de pruebas finales para confirmar que la aplicación cumple con los principios de usabilidad. Posteriormente, la aplicación es pasada a producción.

Con el cumplimiento de estas etapas se estará desarrollando un software de calidad cuyo enfoque principal es la usabilidad. Teniendo en consideración que, si se aplican de manera correcta cada uno de los procesos establecidos, se puede lograr satisfacer las necesidades del cliente y lo más importante tener un software útil con resultados de calidad.

7. Marco Tecnológico

Según la Real Academia Española, el término *Tecnología* se define como: “Conjunto de teorías y técnicas que permiten el aprovechamiento práctico del conocimiento científico”. Tomando esta definición como base, este capítulo define todo el conjunto de técnicas, herramientas, programas, y librerías de software que apoyan al desarrollador a lo largo del ciclo de vida de la construcción de software.

7.1. Tecnologías de Front-End

En una aplicación web, el *Front-End* se define como todas aquellas tecnologías que interactúan con el usuario final, ejecutándose del lado del cliente, normalmente en un navegador web.

A continuación, se entrará en más detalle acerca de estas tecnologías, que serán usadas como parte de este proyecto de investigación.

7.1.1. HTML

HTML es el acrónimo de *Hypertext Markup Language* (Lenguaje de Marcas de Hipertexto). Es un estándar para la elaboración de páginas web y su desarrollo está a cargo del Consorcio W3C [14].

Las páginas web son archivos escritos en lenguaje HTML. Este lenguaje se emplea para dar formato y organizar los documentos mediante etiquetas, las cuales deben ser interpretadas por los navegadores para presentar el contenido al usuario. Al ser un estándar, HTML busca ser un lenguaje que permita la compatibilidad entre las versiones, logran así, que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma por cualquier navegador web.

El lenguaje HTML consta de un conjunto de etiquetas o marcas (*Tags*), encerrados entre corchetes angulares (<,>) y por parejas, siendo una de inicio (apertura) y otra de fin (cierre). HTML también puede describir, hasta cierto punto la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML [8].

HTML5 representa un paso importante en el diseño web, maquetado, y usabilidad. Provee una manera simple de manipular gráficos en el navegador web sin necesidad de usar extensiones para entornos de ejecución de Aplicaciones de Internet Enriquecidas (RIA por sus siglas en inglés) como Flash, Silverlight, o Java Applets. Adicionalmente,

ofrece mecanismos para insertar audio y video en las páginas web sin necesidad de ningún tipo de componente externo [15].

Algunas de estas adiciones más notables según su función son las siguiente [16]:

- Semántica: lo que le permite describir con mayor precisión el contenido. Entre ellas mejoras a etiquetas para estructurar una página web, formularios más avanzados que permiten atributos y validaciones, nuevas etiquetas de audio y video etc.
- Conectividad: lo que le permite comunicarse con el servidor de formas nuevas e innovadoras mediante *web sockets*, eventos, etc.
- Desconectado y Almacenamiento: permite a páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.
- Multimedia: permite utilizar etiquetas de video y audio y utilizar el API de la cámara del computador.
- Gráficos y Efectos 2D/3D: permite una gama mucho más amplia de opciones de presentación mediante *Canvas*, *SVG*, y *WebGL*.
- Rendimiento e Integración: proporciona una mayor optimización de la velocidad y un mejor uso del hardware del equipo.
- Dispositivo de Acceso: admite el uso de varios dispositivos de entrada y salida como la cámara, eventos táctiles, geo localización, orientación del dispositivo, etc.

Una vez que se utiliza HTML para estructurar el contenido de una aplicación web, se puede combinar con hojas de estilo en cascada (CSS, por sus siglas en inglés) para dar formato al contenido previamente estructurado, y *JavaScript* para mejorar la interactividad del lado del cliente.

7.1.2. CSS

Las hojas de estilo en cascada o CSS⁸ es un lenguaje utilizado para describir la presentación de documentos HTML o XML [17]. Estas buscan separar el contenido de la presentación lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, solo modificando unos parámetros de CSS. Las hojas de estilo pueden controlar el formato de múltiples páginas web al mismo tiempo, cualquier cambio en el estilo marcado para un elemento afectará a todas las páginas vinculadas a él mismo.

El *World Wide Web Consortium* (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Las hojas de estilo tienen una sintaxis muy sencilla y se componen en base a reglas. Una

⁸ Cascading Style Sheet

hoja de estilo puede estar compuesta por una o más reglas aplicadas a un documento HTML o XML. Una regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne. La Figura 7.1 es un ejemplo de una regla CSS, describiendo sus partes.

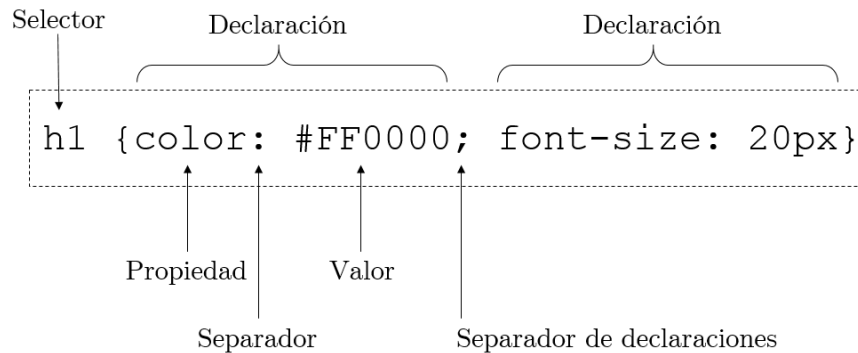


Figura 7.1: Diagrama Sintaxis CSS

Actualmente, CSS3 es la última versión del estándar para hojas de estilo en cascada, dividida en varios documentos separados, llamados módulos los cuales contiene las especificaciones de versiones previas de CSS y las nuevas funcionalidades, siempre en busca de mantener la compatibilidad con las versiones anteriores. Algunos módulos de CSS3 más importantes son:

- Selectores
- Fondos, colores y bordes
- Efectos de texto
- Transformaciones 2D y 3D
- Animaciones
- Diseños de multi-columna

7.1.3. Bootstrap

Bootstrap es un *framework* de código abierto creado por Twitter para el diseño de *Front-End* Web (sitios y aplicaciones web), basado en HTML y CSS. Contiene plantillas de diseño con fuentes, formularios, botones, cuadros, menús de navegación y otros elementos. Adicionalmente, ofrece extensiones en JavaScript para lograr mayor interactividad con el usuario.

Con el advenimiento del desarrollo web móvil, junto al hecho de que los dispositivos móviles se están convirtiendo en el estándar de facto para navegar en el Internet, es esencial que los sitios web sean desarrollados desde una perspectiva *mobile-first*, bajo la técnica de diseño web *Responsive*, para adaptar el diseño a diferentes tamaños de pantallas. Bootstrap ofrece soluciones y atributos que ayudan a los desarrolladores con las tareas difíciles de una manera sencilla [18].

Desde el año 2013, Bootstrap se ha convertido en uno de los proyectos más populares en GitHub. Es apoyado por una gran comunidad y un vasto ecosistema que logran aportar recursos adicionales cuando se trata de plantillas complejas, incluyendo extensiones construidas alrededor de este *framework* [18]. Con un enfoque modular, Bootstrap le permite ahorrar al desarrollador un monto considerable de tiempo y esfuerzo, permitiéndole enfocarse en las partes importantes del proyecto de desarrollo web.

7.1.4. JavaScript

JavaScript es un lenguaje de programación interpretado, creado para darle mayor dinamismo a las páginas web, y permitir un mayor nivel de interactividad con el usuario. Inicialmente fue diseñado para ser un lenguaje de elaboración de scripts del lado del cliente que pudiera incrustarse en archivos HTML permitiendo mejoras en la interfaz de usuario y páginas webs [19]. Dado que JavaScript es interpretado por el navegador, podrá ser ejecutado en cualquier plataforma. Este solo se verá limitado por la versión y el navegador en el que se ejecute.

En una aplicación web, las sentencias JavaScript contenidas en un documento HTML pueden reconocer y responder a eventos generados por el usuario. Una de las mayores ventajas del lenguaje es que permite la interacción con cualquier elemento del Modelo de Objeto de Documento (DOM⁹). No sólo permite alterar sus propiedades e invocar sus métodos, sino también suprimir o crear nuevos elementos y colocarlos en el documento.

A lo largo de los últimos años, gracias a la gran competencia entre los desarrolladores de navegadores web, se han desarrollado motores de JavaScript que han permitido aprovechar al máximo las características del lenguaje. Esto ha desencadenado la aparición de una gran cantidad de librerías y *frameworks*, destacando principalmente la librería jQuery, una de las más usadas por ser de código abierto. jQuery provee fácil manejo de los objetos del DOM, capacidad de realizar animaciones y efectos, manejo de consultas AJAX¹⁰, además de ser compatible con los navegadores más comunes del mercado como Mozilla Firefox, Internet Explorer, Safari, Opera, Google Chrome, y navegadores en dispositivos móviles.

En la actualidad JavaScript ha evolucionado pasando de ser un lenguaje del lado del cliente, a poder ser ejecutado del lado del servidor, independientemente de la

⁹ El *Document Object Model* es la estructura de objetos que genera el navegador cuando se carga un documento y se encarga de modelar todos los elementos dentro de una página web.

¹⁰ JavaScript asíncrono y XML (AJAX, por sus siglas en inglés), es una técnica de desarrollo web para crear aplicaciones interactivas al permitir comunicación con el servidor sin tener que recargar toda la página.

plataforma, ya que solamente se necesita un intérprete JavaScript para su funcionamiento.

7.1.5. Angular

En sus inicios, JavaScript era usado principalmente para dar cierta interactividad entre el usuario y la página web; validaciones de formularios y animaciones de controles web. A medida que JavaScript se hizo más popular, jQuery apareció en la escena del desarrollo Front-End, llevando al límite las capacidades del lenguaje y los navegadores.

El desarrollo de aplicaciones en JavaScript se ha convertido en un desafío debido a su naturaleza maleable y por ser un lenguaje débilmente tipado. Además, JavaScript es usado para todo tipo de procesamiento del lado del cliente como manipulación de la Interfaz de Usuario, interacción entre el cliente y el servidor, ejecución de lógica de negocio, validaciones, entre otros. Como resultado, se tiene un código difícil de mantener y probar. Librerías como jQuery ayudan a ser más productivos y disminuir, en general, el número de líneas de código. Sin embargo, estas librerías carecen de una guía estructural que ayude al desarrollador a medida que crece la cantidad de código en el proyecto. Es aquí donde el Patrón Arquitectónico Modelo Vista Controlador (MVC) juega un papel crucial.

Angular es un *framework* para TypeScript (basado en JavaScript) que permite la creación de *Single Page Application*¹¹(SPA, por sus siglas en inglés) de una manera organizada, siguiendo el patrón MVC. Creado por Google, este *framework* ha ganado bastante interés debido a su diseño modular, potentes características, y gran desempeño. Se ha convertido en una gran elección para construir aplicaciones de negocios sobre la plataforma JavaScript [20].

Por otro lado, Angular presenta ciertas ventajas sobre otras tecnologías con bastante auge como es ReactJS, de Facebook. Entre estas ventajas consideramos que Angular tiene una clara separación entre la vista y el modelo, al requerir que la vista sea codificada en formato HTML, lo cual reduce el tiempo de desarrollo al no tener que aprender ningún lenguaje específico para la capa de presentación, permitiendo usar un estándar definido. Asimismo, Angular ofrece una amplia gama de componentes a través de sus diferentes capas, mientras ReactJS se enfoca únicamente en la capa de presentación, obligando a los desarrollares a crear sus propios componentes para las otras capas, o buscar componentes disponibles en la comunidad. Finalmente, Angular (desde la versión 2) soporta el uso de TypeScript, que es un super-conjunto de JavaScript, lo cual lo hace más potente que otros *frameworks* de desarrollo en Front-End al proveer más características que facilitan el desarrollo y fomentan la organización del código y las buenas prácticas.

¹¹ SPA son aplicaciones enriquecidas (RIA), donde no hay recarga/redirección de la página, manteniendo la posibilidad de marcar la página como favorita. Normalmente involucran un fuerte diseño UX y uso de patrones de arquitectónicos como MVC, MVP, MVVM, etc.

7.1.6. TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado por y mantenido por Microsoft con la finalidad de facilitar la creación de aplicaciones JavaScript a gran escala [21].

TypeScript fue diseñado con las siguientes metas en mente [21]:

- Realiza chequeos de tipos estáticos en tiempo de compilación para identificar problemas en el código.
- Altamente compatible con JavaScript. Cualquier código escrito en JavaScript que sea válido, también es un código TypeScript válido.
- Proveer mecanismo de estructuración de grandes piezas de código haciendo uso de conceptos como Clases, Interfaces, y Módulos.
- No impone ninguna carga de desempeño al ejecutarse. TypeScript es compilado a JavaScript. Así que, lo que se ejecuta finalmente es, de hecho, JavaScript.

7.2. Tecnologías de Back-End

Si el *Front-End* corresponde a las tecnologías del lado del cliente, el *Back-End* es su contraparte del lado del servidor. Comprende a aquellas tecnologías que apoyan a la aplicación web, para servir a las peticiones de los clientes. Estas son tecnologías muy diversas correspondientes a las diferentes capas de una aplicación. Por ejemplo, en la capa de aplicación se puede desarrollar en diferentes lenguajes y plataformas como Java, PHP, Python, Ruby, .Net, etc. En la capa de Persistencia de Datos se pueden tener bases de datos relacionales como MySQL, PostgreSQL, Oracle; como bases de datos no relacionales como MongoDB, CouchDB, Cassandra, entre otros.

A continuación, se verá en más detalle las tecnologías propuestas para el desarrollo de la aplicación web.

7.2.1. Java

Java es una plataforma de desarrollo creada por Sun Microsystems en 1995, inicialmente como plataforma de transferencia de contenidos multiplataforma, que luego incursionó en el área del Internet como parte del navegador Netscape [22], para mejorar la experiencia del usuario y su interactividad a través de su tecnología *Applets*.

Uno de los aspectos más llamativos de la plataforma Java es su característica de multiplataforma, bajo el lema “*Write once, run anywhere*”¹². Esto permite que una vez

¹² Escribe una vez, ejecuta donde sea.

que una aplicación sea desarrollada y compilada, puede ser ejecutada en cualquier sistema operativo que soporte la plataforma Java.

La plataforma Java se compone de varias tecnologías, entre las cuales se encuentran dos de sus principales componentes: Lenguaje de programación Java, y Entorno de ejecución.

Lenguaje de Programación Java

El lenguaje de programación Java es un lenguaje de propósito general, concurrente, y orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El objetivo era crear un lenguaje de programación parecido a C++ en estructura y sintaxis, pero que tuviera la menor dependencia posible de la plataforma donde se ejecutaría.

Java es un lenguaje fuertemente tipado, con claras distinciones entre errores en tiempo de compilación que pueden y deben ser detectados por el compilador, y aquellos errores que ocurren en tiempo de ejecución. El proceso de compilación consiste en la traducción del código fuente Java a una representación *Byte Code* independiente de la plataforma de hardware. Los procesos en tiempo de ejecución incluyen la carga y el enlazado de las clases necesarias para ejecutar un programa, la generación de código de máquina, la optimización dinámica del programa, y la ejecución del programa en sí.

Es un lenguaje de alto nivel, en el sentido que oculta los detalles de la representación de la máquina que no están disponibles a través del lenguaje. Entre sus ventajas principales, se encuentran:

- Está basado en el paradigma orientado a objetos.
- Un programa Java es portable. Permite la ejecución de un mismo programa en diferentes sistemas operativos.
- Provee todas las librerías necesarias para el manejo de recursos del sistema como el acceso a recursos de disco, red, I/O. Todo sin necesidad de instalar librerías o componentes adicionales.

El lenguaje en sí mismo está especificado en la *Java Language Specification (JLS)*. Desde la versión 1.4 de Java, la evolución del lenguaje ha sido regulada por el *Java Community Process*, que usa *Java Specification Request (JSR)* para proponer y especificar cambios en la plataforma Java.

Entorno de Ejecución de Java

Cualquier aplicación que esté destinada para la ejecución en la plataforma Java necesita de al menos dos componentes presentes en el sistema donde se va a ejecutar: una máquina virtual de Java (JVM), y un conjunto de librerías que le proporcionen los servicios que la aplicación requiera. A la implementación de estos dos componentes se le conoce como el *Java Runtime Environment (JRE)* y es lo mínimo que requiere un

sistema para poder ejecutar aplicaciones Java. Existen diferentes implementaciones de estos componentes, entre los cuales se pueden mencionar a Oracle JRE y OpenJRE.

Máquina Virtual de Java

Es una máquina virtual de proceso nativo capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (conocido como el *Byte Code* de Java), el cual es generado por el compilador de Java a partir del código fuente.

Es este el componente que hace portable las aplicaciones Java al procesar el *Byte Code* (instrucciones de bajo nivel independientes de la plataforma) y generar instrucciones nativas de la plataforma donde se está ejecutando la aplicación. Debe ser compilado en función de cada plataforma, de manera que la máquina virtual de Java no es independiente de la plataforma.

La Figura 7.2 muestra el flujo de una aplicación Java desde su creación como código fuente independiente de la plataforma, hasta su ejecución en una plataforma en particular.

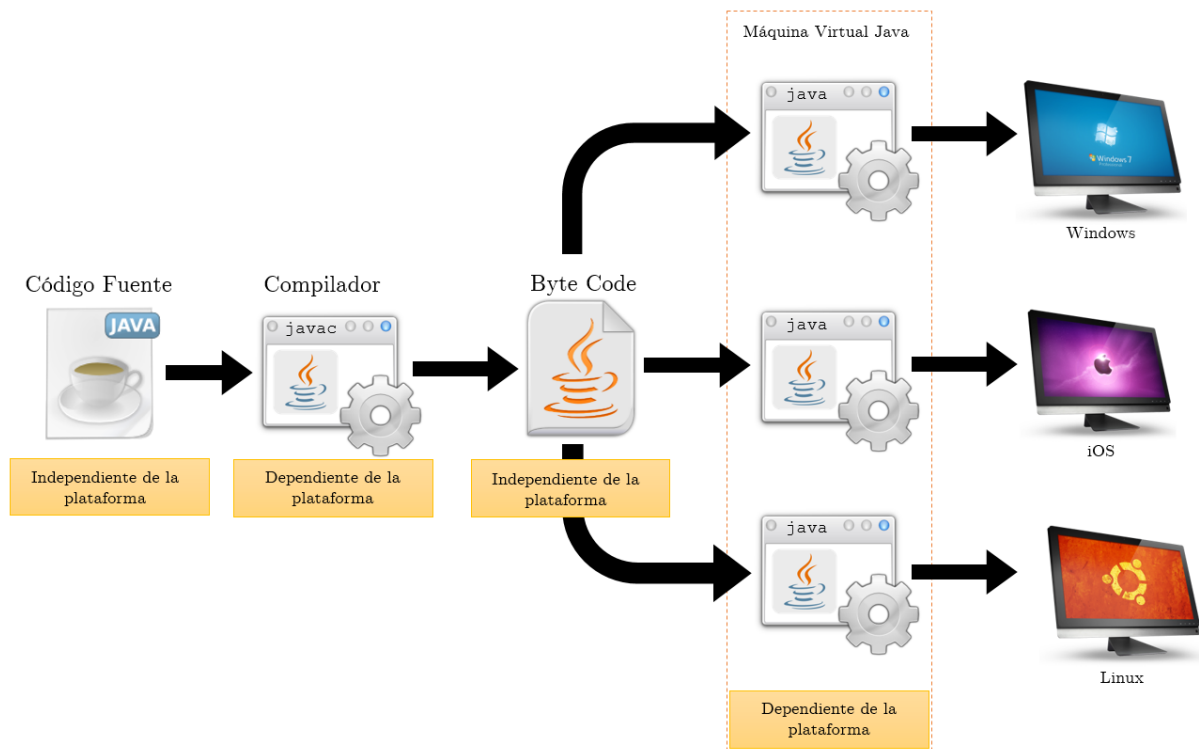


Figura 7.2: Diagrama de Independencia de Plataforma de una Aplicación Java

Librería Estándar de Java

Los sistemas operativos actuales ofrecen un API (*Application Program Interface*) para permitir a las aplicaciones ejecutar ciertas tareas. Este API normalmente se ofrece en forma de librerías que pueden ser enlazadas dinámicamente durante la ejecución de la

aplicación. El problema se encuentra precisamente en que estas librerías son dependientes de la plataforma en ejecución, mientras que se pretende que las aplicaciones Java sean independientes de la misma. Por esta razón, las aplicaciones Java no pueden apoyarse sobre la API que provee el sistema operativo subyacente, sino en lugar de eso, la Plataforma Java define una serie de librerías estándar que contienen muchas de las funcionalidades reutilizables disponibles en los sistemas operativos.

La librería estándar de Java tiene dos propósitos dentro de la plataforma:

- Ofrecer al desarrollador un conjunto bien definido de funciones para realizar tareas comunes.
- Proporcionar una interfaz abstracta para las tareas que son altamente dependientes del hardware de la plataforma destino.

Estas librerías han sido definidas, en un intento por cubrir todas las plataformas, en tres categorías:

Java ME (Java Platform, Micro Edition)

Orientada a entornos de limitados recursos, como teléfonos móviles, PDA, impresoras y más.

Java SE (Java Platform, Standard Edition)

Orientada a entornos de gama media y estaciones de trabajo. Por ejemplo, el usuario promedio con PC se sitúa en esta categoría. Esta plataforma ofrece todos los componentes necesarios para el desarrollo de aplicaciones Java. En la Figura 7.3 se puede apreciar con detalle todas las especificaciones que componen esta plataforma.

Java EE (Java Platform, Enterprise Edition)

Orientada a entornos distribuidos empresariales o Internet. Esta plataforma extiende la plataforma Java SE para ofrecer una serie de especificaciones orientadas al aumento de productividad en el desarrollo de aplicaciones empresariales.

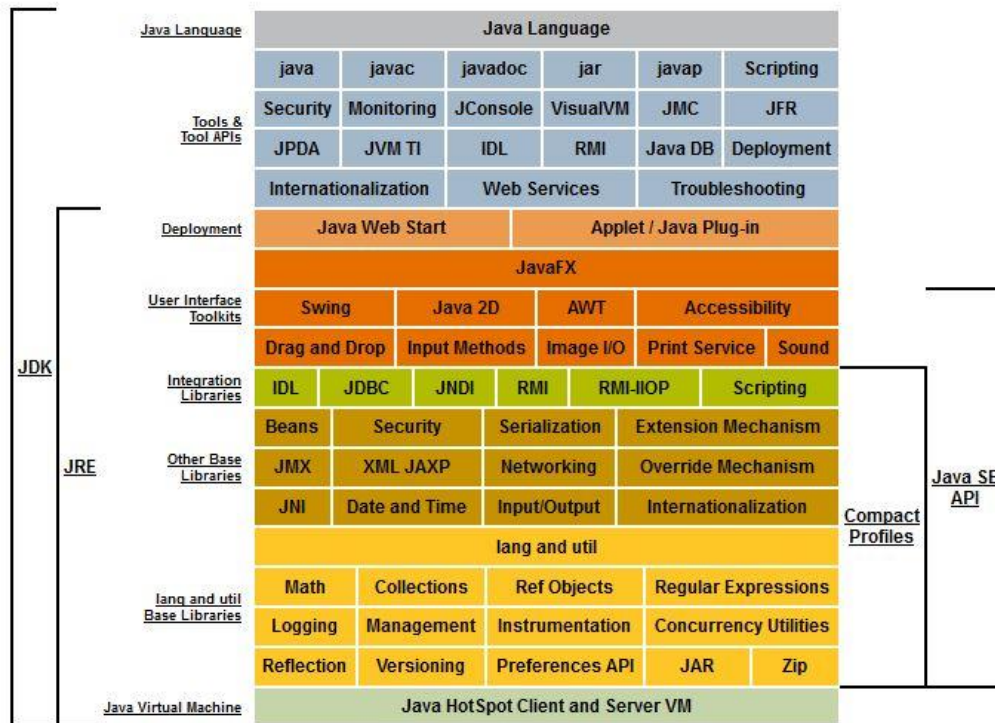


Figura 7.3: Composición de Java Platform Standard Edition

7.2.2. Spring

Spring es un *framework* ligero de código abierto para la construcción de aplicaciones Java. Puede ser usado para construir cualquier tipo de aplicación en Java como aplicaciones *stand-alone*, web, o aplicación Java Enterprise Edition. Esta flexibilidad lo pone por encima de otros *frameworks* similares que están limitados a la construcción de aplicaciones web (e.g. Apache Struts).

Spring Framework está compuesto por múltiples tecnologías y herramientas de código abierto, que una vez unificados bajo el estandarte de Spring, facilitan al desarrollador hacer uso de las funcionalidades de dichas tecnologías sin tener la necesidad de acoplar la aplicación muy de cerca con alguna herramienta en particular.

El núcleo de Spring Framework (Spring Core) se basa en el principio o patrón Inversión de Control (IoC, por sus siglas en inglés). IoC es una técnica que externaliza la creación y administración de las dependencias de los componentes. Esto lo logra aplicando una técnica llamada Inyección de Dependencias (DI, por sus siglas en inglés), término acuñado por Martin Fowler que resulta mucho más descriptivo [23].

El concepto de DI establece que un componente no debe crear o generar los otros componentes de los cuales depende, sino obtener dichas dependencias a través de un proceso externo. Traducido al mundo de Java, esto significa que una clase (componente) no debe instanciar un objeto de otra clase que necesita para su lógica

interna. En su lugar, deberá recibir la instancia de dicho componente a través del constructor o alguna función setter.

La implementación de DI en Spring se basa en dos conceptos básicos de Java: JavaBeans e interfaces [23]. Al mantener el código orientado a interfaces, es decir, las dependencias en el código se crean con interfaces Java (objetos abstractos) en lugar de clases concretas, se evita el acoplamiento del código a estos componentes, permitiendo el intercambio entre implementaciones de las interfaces con poco esfuerzo. En el contexto de DI, Spring actúa más como un contenedor que como un *framework*; provee instancias de las clases de la aplicación con todas las dependencias que necesita, pero lo hace de una manera poco intrusiva.

A lo largo de sus versiones, Spring, además de las funcionalidades ofrecidas por el Core, ha incorporado más componentes a su repertorio, ofreciendo así mayores funcionalidades a los desarrolladores. Entre ellas, se pueden mencionar las siguientes [23]:

- Spring Context: *ApplicationContext*, UI, Validaciones, JNDI, Enterprise JavaBeans (EJB), remoting, and soporte para e-mail.
- Spring DAO: Infraestructura transaccional, *Java Database Connectivity* (JDBC) y soporte para Objetos de Acceso a Datos (DAO, por sus siglas en inglés).
- Spring ORM: Hibernate, iBATIS, y soporte para *Java Data Objects* (JDO).
- Spring AOP: una implementación de Programación Orientada a Aspectos (AOP, por sus siglas en inglés) que cumple con el AOP Alliance.
- Spring Web MVC: *framework* Modelo-Vista-Controlador (MVC) basado en la web
- WebSocket, SockJS, y mensajería STOMP.

En la Figura 7.4, se puede observar un diagrama de la arquitectura de Spring con sus principales componentes.

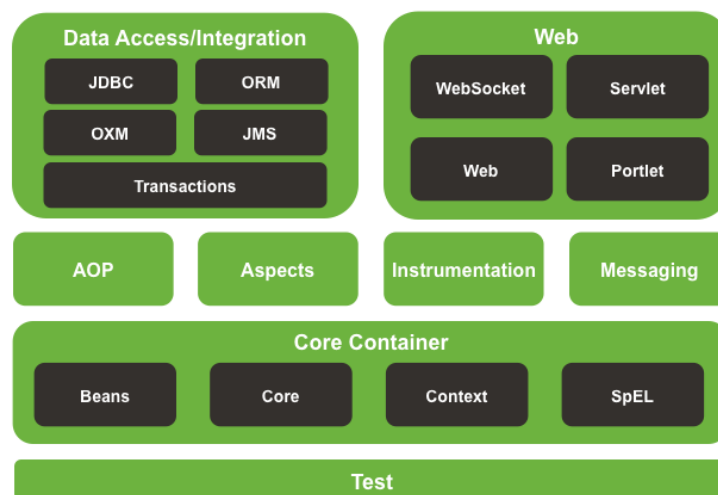


Figura 7.4: Arquitectura de Spring Framework

De igual forma, adicionalmente a los componentes principales, muchos proyectos se han creado alrededor de este eco sistema, entre ellos se pueden mencionar: Spring Security, Spring Integration, Spring Batch, Spring Data, Spring Web Flow, Spring Boot, Spring Social, entre otros.

7.2.3. H2

H2 es un sistema manejador de bases de datos relacional de código abierto escrito totalmente en Java. Entre sus principales características se pueden destacar la siguientes [24]:

- Puede ser embebido en aplicaciones Java o ejecutado en modo Cliente-Servidor.
- Puede ser usado como base de datos en memoria (*in-memory DB*)
- Ofrece implementación del API estándar de Java de conexión de bases de datos JDBC.
- Soporta la mayoría de las operaciones SQL.
- Soporta las características ACID.
- El tamaño total no supera los 2MB.
- Ofrece consola de administración basado en la web, embebido.

Este sistema ofrece simplicidad de uso sobre otros sistemas más grandes al minimizar los esfuerzos de operaciones e instalación. Asimismo, la consola de administración web facilita la ejecución de operaciones directamente en la base de datos sin requerir de aplicaciones adicionales.

Aunque es un sistema que presenta ciertas limitaciones de tamaño y cantidad de datos [25], se ajusta totalmente a aplicaciones de corto alcance con baja o mediana carga de trabajo.

7.2.4. Hibernate

Hibernate es una herramienta de Mapeo Objeto/Relación (ORM, por sus siglas en inglés), disponible para la plataforma Java, que permite a los desarrolladores escribir aplicaciones que requieren persistencia de datos en una base de datos relacional. En esencia, ORM funciona transformando datos de una representación (objetos Java) a otra (sentencias SQL). Hibernate abstrae al desarrollador, en la mayoría de los casos, de tener que lidiar con sentencias SQL, que pueden resultar enredadas dependiendo del nivel de complejidad de las relaciones y la consulta que se requiera.

Entre los beneficios que tiene el uso de Hibernate [26]:

- Productividad: elimina la mayoría del trabajo repetitivo y permite concentrarse en los problemas del negocio. En resumen, disminuye el tiempo de desarrollo.

- **Mantenibilidad:** reduce la cantidad de líneas de código necesarias para lograr las tareas de persistencia de datos, resultando en un código más entendible y más fácil de refactorizar.
- **Desempeño:** aunque Hibernate introduce cierto nivel de carga al ser usado, en general, aplica varias optimizaciones en todo momento que permiten un aumento general del desempeño. Por ejemplo, el uso de niveles de caché para los objetos consultados.
- **Independencia de Proveedor:** puede ayudar a mitigar los riesgos asociados a estar estrechamente acoplado con un proveedor o producto (e.g. MySQL). Aun si no se piensa cambiar de proveedor a lo largo del proyecto, Hibernate soporta diferentes productos RDBMS, lo que permite el uso de diferentes productos en diferentes ambientes de desarrollo. Por ejemplo, usar H2 para el ambiente de desarrollo y MySQL para el ambiente de producción sin necesidad de cambiar la lógica de la aplicación.

7.2.5. iText

iText es una librería creada originalmente para Java, luego portada a C# bajo el nombre iTextSharp. Estas librerías hacen posible el manejo de archivos PDF de manera dinámica. Se puede integrar con aplicaciones para generar documentos PDF como una alternativa a la impresión en papel. Puede firmar digitalmente un documento, dividir o concatenar diferentes documentos, entre otras funcionalidades. Esto lo logra al exponer APIs que deben ser llamadas por la aplicación, que abstrae de los detalles de la implementación o el formato de los archivos PDF. Posibilita la generación de documentos digitales como facturas de compras realizadas por clientes; producir boletos de entradas a un evento; crear certificados de asistencia digital [27].

Formato de Archivos PDF

El formato PDF (*Portable Document Format*) es un formato de archivo para representar documentos de una manera independiente de la aplicación, *hardware*, y sistema operativo que se usó para crearlo y del dispositivo de salida en el que será mostrado o impreso.

Un documento PDF consiste en una colección de objetos que, en conjunto, describen la aparición de una o más páginas, posiblemente acompañado de elementos interactivos adicionales y datos de aplicación de alto nivel. Un archivo PDF contiene los objetos que componen un documento PDF, junto con la información estructural asociada, todo representado como una única secuencia de Bytes auto-contenida [28].

Las páginas de un documento PDF (y otros elementos visuales) pueden contener cualquier combinación de texto, gráficos, e imágenes. Adicionalmente, las páginas pueden contener elementos interactivos que sólo pueden ser posibles en su representación electrónica (i.e. no impresa). Soporta anotaciones de muchos tipos como notas de texto, enlaces web, archivos adjuntos, sonidos, y videos. Un documento PDF puede definir su propia interfaz de usuario; las entradas de teclado y ratón pueden

disparar acciones especificadas por los objetos PDF. También puede contener formularios interactivos para ser llenado por el usuario, y exportar esos datos a otras aplicaciones. El formato de archivos PDF combina tres tecnologías:

- Un subconjunto del lenguaje de programación de descripción de página PostScript, para generar las plantillas y gráficas.
- Un sistema de reemplazo/incrustación de fuentes para permitir a las fuentes viajar con el documento.
- Un sistema de almacenamiento estructurado que empaca todos los elementos y cualquier contenido asociado en un archivo único, con compresión de datos cuando sea apropiado.

7.2.6. JasperReports

JasperReports es una librería potente y flexible para generar reportes en pantalla, impresos, o almacenados como archivos en formato PDF, HTML, XML, entre otros.

Esta librería está escrito completamente en Java y puede ser usada en una variedad de aplicaciones basadas en Java, incluyendo las plataformas Java EE, o aplicaciones web, generando reportes usando contenido dinámico [29].

Similar a otras herramientas de reportes, JasperReports usa plantillas de reportes estructuradas en múltiples secciones, como Título, Resumen, Detalle, y Encabezados de Grupos y Pie de Página. Cada sección tiene un diseño flexible donde se pueden configurar varios tipos de elementos, incluyendo imágenes, campos estáticos y dinámicos, líneas, rectángulos, entre otros. Las plantillas son almacenadas en formato XML (JRXML) [29], y pueden ser creadas con cualquier editor de texto, o usando una aplicación de diseño llamada *Jaspersoft Studio* que provee una interfaz gráfica para la creación y edición de las plantillas.

El motor de reportes usa estas plantillas para rellenarlas con los datos aportados por la aplicación que usa la librería. Una vez generado el reporte, se puede exportar a cualquiera de los formatos soportados.

7.3. Herramientas de Desarrollo

Habiendo visto las tecnologías a usar tanto en el Front-End como en el Back-End, se continúa a presentar aquellas herramientas que permiten un incremento en la productividad del desarrollo e implementación de las soluciones que se plantean. Estas herramientas facilitan muchas de las tareas que deben ser realizadas por los desarrolladores, y en muchos casos se consideran repetitivas; o ayudan a preservar la integridad y seguridad del proyecto; pueden facilitar la colaboración entre múltiples participantes o el uso de componentes externos a la aplicación.

A continuación, se presentarán las herramientas más relevantes a ser usadas en este proyecto, tanto del lado del Front-End como del lado del Back-End. En varios casos se basan o usan las tecnologías mencionadas en el Capítulo 3.

7.3.1. JHipster

JHipster es un generador de proyectos web de código abierto que combina tres *frameworks* muy exitosos en el mundo del desarrollo web: Bootstrap, Angular, y Spring Boot. Esta herramienta facilita el inicio de un proyecto de desarrollo web, al proveer o generar un proyecto totalmente nuevo, que contiene estos *frameworks* y otras herramientas integradas, ahorrando el tiempo y esfuerzo que estas tareas conllevan, permitiendo así, enfocarse principalmente en la lógica de negocio de la aplicación que se está desarrollando.

JHipster usa como núcleo el generador Yeoman. Yeoman es un generador de código que, a través de la ejecución de comandos, genera aplicaciones completas o piezas útiles de una aplicación [30]. Esta herramienta se encarga de proveer todo lo necesario para empezar el desarrollo sin el esfuerzo asociado a una configuración manual.

Yeoman promueve y sigue un flujo de trabajo (*Workflow*) que genera un proyecto basado en valores pre-definidos que pueden ser actualizados luego durante el desarrollo. Este flujo de trabajo se conforma de tres tipos de herramientas que mejoran la productividad cuando se desarrolla una aplicación web [30]:

- La herramienta de *scaffolding* (e.g. herramienta “yo”)
- La herramienta de construcción (*Build tool*, e.g. Grunt, Gulp)
- El administrador de paquetes (e.g. Bower, NPM)

7.3.2. Node Package Manager

Node Package Manager (NPM, por sus siglas en inglés) es un administrador de paquetes, usado por el entorno de ejecución JavaScript llamado *Node.js*.

Para los desarrolladores JavaScript, NPM simplifica compartir código creado por ellos para resolver problemas particulares. De igual forma, agiliza el uso de código o componentes creados por otros desarrolladores. Una vez se crea una dependencia a uno de estos componentes creados por otros desarrolladores, NPM facilita la verificación de actualizaciones del componente, y descargar dichas actualizaciones si existen [31].

7.3.3. Webpack

Webpack es un empaquetador y procesador de activos de proyectos de Front-End. Este procesador se encarga de traducir, compilar, y desplegar el código del proyecto [32]. Integra servidores web embebidos que apoyan la fase de desarrollo. También se integra fácilmente con NPM, siendo capaz de obtener las dependencias definidas por este para ser procesadas.

Webpack se adecua a proyectos grandes porque permite diferentes modos de procesamientos como *Desarrollo* o *Producción*. En modo *Desarrollo*, utiliza las versiones no minimizados de archivos JavaScript o CSS, permitiendo depurar la aplicación más fácilmente. En el modo *Producción*, puede minimizar todos los archivos y empaquetarlos con el fin de reducir su tamaño y hacerlos más eficientes de descargar.

7.3.4. Git

Git es un Sistema de Control de Versión (VCS, por sus siglas en inglés) que permite a un sistema registrar cambios hechos a un archivo o conjunto de archivos a través del tiempo, con la finalidad de poder recuperar versiones específicas luego. Estos sistemas de control permiten revertir cambios hechos a un archivo y llevarlo a un estado anterior, revertir el estado de un proyecto completo, comparar cambios a través del tiempo, ver quién modificó algún archivo que causó algún problema, entre otras funcionalidades [33].

Git es considerado como un VCS distribuido, es decir, los clientes no sólo se descargan la última versión de los archivos desde el servidor, sino que tienen una réplica completa de ellos localmente. De esta manera, si el servidor es inaccesible o el repositorio se ve afectado, cualquiera de los repositorios en los clientes puede ser copiado y restaurado en el servidor.

El proveedor seleccionado del repositorio de este proyecto en la nube es GitLab. GitLab es una aplicación en la nube que ofrece el manejo del ciclo de vida de aplicaciones. Desde la planificación hasta el monitoreo, GitLab cubre todas las etapas y permite la gestión segura de ellas. Ofrece repositorios Git privados gratuitos, módulos de documentación tipo Wiki e incluye un motor de Integración Continua.

Entre las características principales de Git, se pueden mencionar las siguientes [33]:

- Diseño Simple: gran soporte para desarrollo no-lineal (miles de ramas de desarrollo en paralelo).
- Totalmente Distribuido: capaz de manejar proyectos grandes como el *Kernel* de Linux de manera eficiente.
- Operaciones Locales en su Mayoría: la mayoría de las operaciones en Git sólo necesita recursos locales para poder operar (i.e. no se requiere información de

otro computador de la red). Esto conduce a un mayor desempeño en el tiempo de las operaciones al no tener que pasar por la red para ejecutarlas.

- Integridad: todo en Git es verificado a través de un checksum antes de ser almacenado y luego es referenciado usando ese checksum. Esto hace virtualmente imposible cambiar el contenido de cualquier archivo sin que Git se entere.

7.3.5. Apache Maven

Apache Maven (o sólo Maven) es un administrador de proyectos de software. Basado en el concepto de *Project Object Model* (POM) como modelo de estructura del proyecto, Maven puede manejar la construcción del proyecto, reportes y documentación desde una pieza central de información [34]. Esta herramienta de construcción (*Build Tool*) es usada principalmente para proyectos Java, aunque no está limitada únicamente para esta plataforma. Maven, a través del uso de *plugins*, automatiza muchas de las tareas que deben ejecutarse durante la construcción de un proyecto, definiendo interfaces para cada paso del ciclo de vida de la construcción de un proyecto que permiten a los *plugins* ejecutar las acciones que requieran. Adicionalmente, Maven gestiona las dependencias del proyecto con otros componentes o librerías, siendo capaz de descargar dichas dependencias o las dependencias transitivas (dependencias de las dependencias) desde un repositorio central. Maven se basa en los siguientes principios y conceptos:

- Convención Sobre Configuración: concepto simple donde los sistemas, librerías, y *frameworks* deben asumir valores razonables por defecto sin requerir configuración innecesaria para que el sistema pueda funcionar [35].
- Interfaz Común: antes de que Maven introdujera una interfaz común, cada proyecto podía tener su propia estructura, y su propia manera de generar al entregable final. Esto introducía la necesidad de un rol dentro del proyecto, donde alguien debía hacerse cargo de todas estas tareas relacionada con la construcción del proyecto. Asimismo, esos eran estándares que podían aplicar únicamente a un proyecto, con lo que, si se trabajaba en un proyecto de un equipo o compañía diferente, podía tener una estructura totalmente distinta, desperdiciando una gran cantidad de tiempo sólo para entender y aprender cómo generar y construir el entregable.
- Reutilización Universal Usando *Plugins*: el núcleo de Maven no es muy potente en sí mismo. De hecho, el núcleo no hace mucho más que procesar algunos documentos XML y gestionar los pasos del ciclo de vida. Maven ha sido diseñado para delegar la mayoría de la responsabilidad a un conjunto de *Plugins* que pueden afectar el ciclo de vida de Maven [35].
- Modelo Conceptual de "Proyecto": Maven mantiene el modelo de un proyecto: no sólo se está compilando código fuente, también define respuestas a interrogantes como ¿Cuál es la licencia del proyecto?, ¿Cuál es la versión?, ¿Quién lo desarrolla y contribuye?, ¿De qué otros proyectos depende? Este

modelo provee de una nueva semántica relacionada a proyectos de software y desarrollo de software. Este modelo ofrece características como:

- Administración de dependencias.
- Repositorios remotos.
- Reutilización universal de lógica de construcción.
- Integración y portabilidad de Maven.

8. Marco Aplicativo

En este capítulo se explica el diseño de la solución en base a los lineamientos de la metodología AgilUs, descrita en el Capítulo 6.

8.1. Análisis General

Siguiendo los lineamientos propuestos por la metodología en el Capítulo 6, antes de iniciar el desarrollo de la aplicación, se llevó a cabo una fase de captación de requisitos, donde se determinó de manera global los principales requerimientos para satisfacer la necesidad del usuario y cubrir los objetivos estipulados en el Capítulo 2 para lograr construir la solución a la problemática planteada.

Para lograr este análisis de alto nivel se realizaron llamadas telefónicas con el cliente, resultando en la siguiente lista de requerimientos generales de las funcionalidades deseadas para la aplicación:

- Debe permitir creación, edición, y eliminación de conferencias pagas y gratuitas.
- Debe permitir creación, edición, y eliminación Contribuciones y Tutoriales.
- Los Tutoriales pueden tener un costo asociado independiente del costo de asistencia a la conferencia para conferencias pagas.
- Debe poder aplicar descuentos en función del perfil del usuario para conferencias pagas.
- Debe manejar diferentes costos dependiendo de la fecha de reservación (temprano/tardío) para conferencias pagas.
- Debe restringir la asistencia a los Tutoriales, i.e., los Tutoriales tienen una capacidad máxima de asistencia.
- Debe generar certificados digitales (PDF) para autores, asistentes a la conferencia, y asistentes a los tutoriales.
- Debe generar distintivos para autores, asistentes a la conferencia, y asistentes a los tutoriales.
- Debe poder aceptar información de pagos para ser validadas por el usuario.
- Debe permitir el registro de usuarios para asistir como autor o como asistente.
- Debe restringir funcionalidades de administración a usuarios con permisos.
- Debe poder administrar datos del usuario.
- Debe exportar información de pago y facturación.

8.2. Plan de Entrega

Considerando que una nueva entrega de la Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa) estaba por realizar en Octubre de 2017 y, se

determinó que la fecha tentativa para la realización de la próxima conferencia Simposio Científico y Tecnológico en Computación (SCTC) sería Mayo de 2018. Es necesario definir y priorizar las funcionalidades mínimas necesarias para desarrollar un Producto Mínimo Viable¹³ para cumpla con esas fechas.

Se definió un plan de entrega dividido en dos fases. La primera fase contendría aquellas funcionalidades requeridas para una ejecución exitosa del CoNCISa 2017. La segunda fase contendría aquellos elementos del CoNCISa que no fueron incluidos en la primera fase porque se consideraron con menor prioridad. Adicionalmente, incluye los elementos necesarios para la ejecución del SCTC 2018.

8.2.1. Primera Fase

La primera fase de entrega se define como aquella donde se desarrollan los componentes mínimos necesarios teniendo como objetivo la ejecución de la Conferencia Nacional de Computación, Informática y Sistemas 2017 (CoNCISa 2017).

A continuación, se presenta la lista de requerimientos mínimos necesarios para esta primera fase:

- Registro de usuarios (ver Tabla 8.1).
- Autenticación de usuario (ver Tabla 8.2).
- Autorización de usuario (ver Tabla 8.3).
- Recuperación de contraseña (ver Tabla 8.4).
- Cambio de contraseña (ver Tabla 8.5).
- Reservación de entrada a conferencia como Asistente a conferencia paga (ver Tabla 8.6).
- Reservación de entrada a conferencia como Autor a conferencia paga (ver Tabla 8.7).
- Reservación de entrada a Tutorial como Asistente a conferencia paga (ver Tabla 8.8).
- Reservación con Recibo de Pago o Factura (ver Tabla 8.9).
- Lista de Reservaciones (ver Tabla 8.10).
- Envío de información de Pago (ver Tabla 8.11).
- Cancelación de reserva (ver Tabla 8.12).
- Visualizar historial de reservas (ver Tabla 8.13).
- Administración de usuarios (ver Tabla 8.14).
- Administración de reservas (ver Tabla 8.15).
- Administración de certificados de asistencia (ver Tabla 8.16).
- Administración de distintivos (ver Tabla 8.17).

¹³ Un *Producto Mínimo Viable* es aquel que tiene la cantidad mínima de funcionalidades para satisfacer las necesidades de mayor prioridad, y obtener resultados tempranos para futuros desarrollos.

8.2.2. Segunda Fase

En esta fase queda acordado la entrega del resto de los requerimientos definidos inicialmente, comprendiendo todas aquellas funcionalidades consideradas de valor y que, en su conjunto, forman la aplicación completa. Esta última entrega coincide con la fecha esperada de la conferencia Simposio Científico y Tecnológico en Computación 2018 (SCTC 2018) en Mayo de 2018.

A continuación, se presenta la lista de requerimientos restantes a entregar en esta fase:

- Edición de perfil de usuarios (ver Tabla 8.18).
- Administración de conferencias (ver Tabla 8.19, Tabla 8.20, y Tabla 8.21).
- Soporte para conferencias gratuitas.

8.3. Desarrollo de la Aplicación

La primera etapa en iniciar fue la etapa de Requisitos donde se recopilaban los requerimientos de alto nivel. Estos requerimientos sirvieron de base para crear la visión general de lo que sería la aplicación, y pasar a la etapa siguiente de Análisis, donde se crearon los elementos fundamentales para soportar el desarrollo iterativo definido por la metodología de desarrollo.

8.3.1. Arquitectura General

De esta etapa se continuó a la etapa de Análisis, donde se construyó la arquitectura general de la aplicación a ser desarrollada.

En la Figura 8.1, se presenta la arquitectura de la aplicación web, basada en el Modelo de bosquejo C4¹⁴. En esta, se aprecia el contexto del sistema: El Sistema y las Dependencias Externas del Sistema.

¹⁴ Software Architecture and the C4 Model, <http://static.codingthearchitecture.com/c4.pdf>

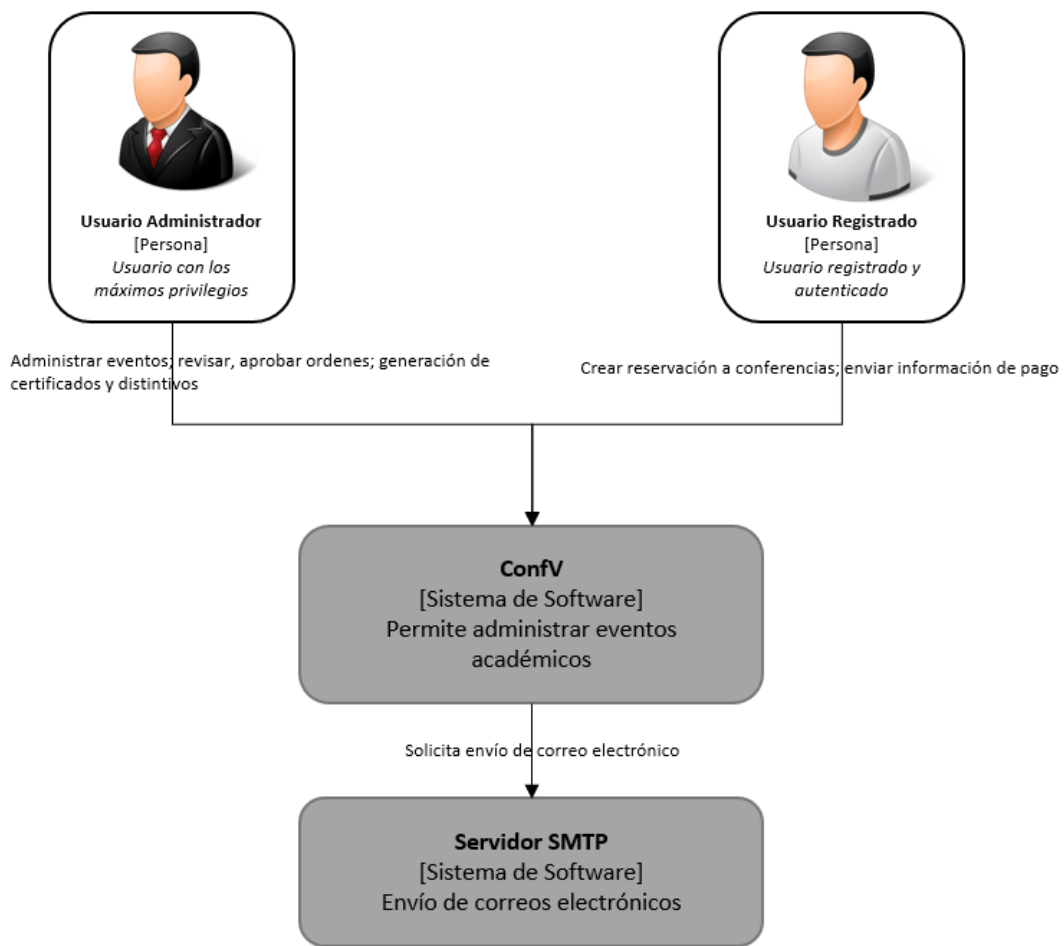


Figura 8.1: Diagrama de Contexto del Sistema

En la Figura 8.2, se presenta el segundo modelo C4 de la aplicación donde se aprecia la forma general de la arquitectura del sistema y las tecnologías elegidas.

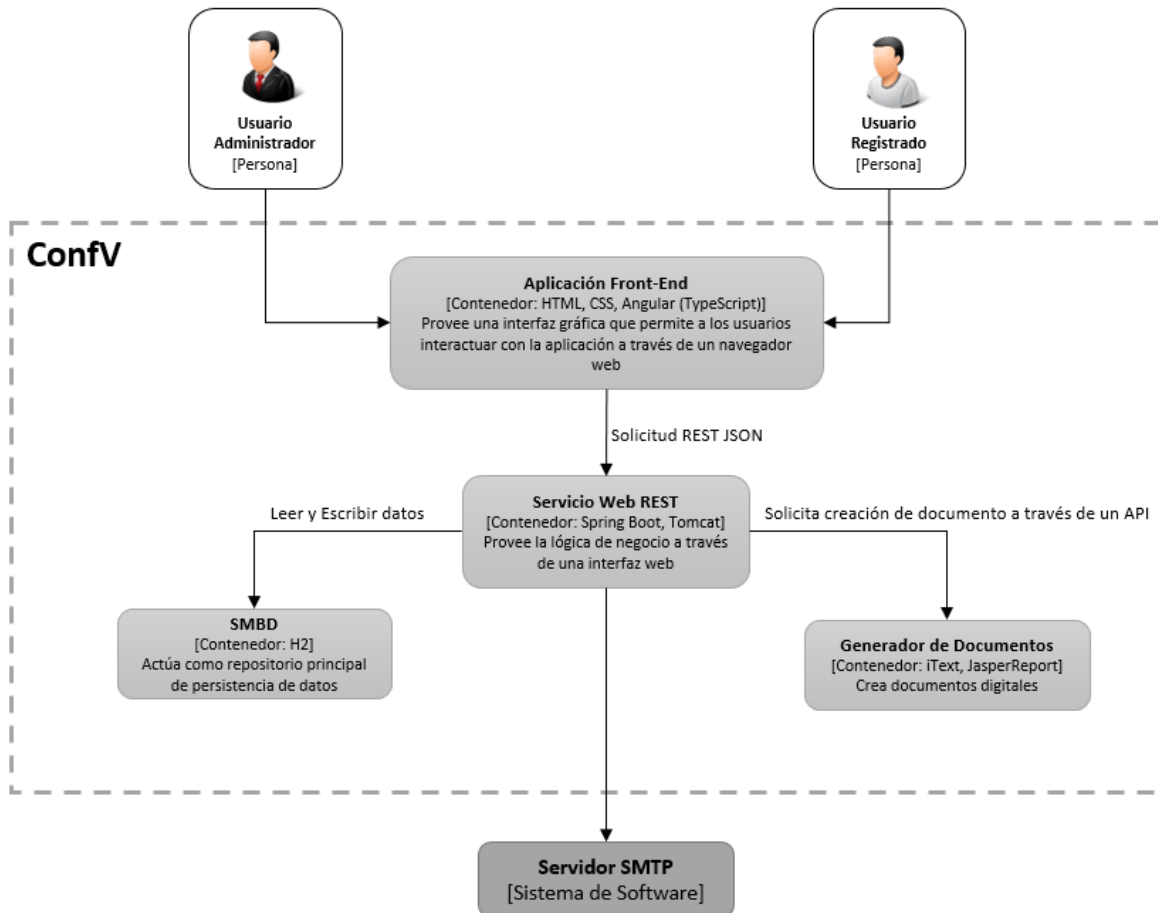


Figura 8.2: Diagrama de Contenedores del Sistema

8.3.2. Modelo Final de la Base de Datos

Adicionalmente a la arquitectura presentada en la Sección 8.3.1 (Arquitectura General), es necesario crear el modelo de datos encargado de dar forma y estructurar los datos que van a ser almacenados en la aplicación.

Cabe destacar que, aunque se presenta la versión final de la base de datos, esta fue construida iterativamente a medida que se iba avanzando en el desarrollo. Siendo fieles a la metodología de desarrollo AgilUs, donde el desarrollo es guiado por la usabilidad, este modelo es la representación final de cada una de las etapas ejecutadas de esta metodología. En otras palabras, en lugar de iniciar el desarrollo con un modelo de datos estricto, a medida que se analizaban los requerimientos, el modelo se iba adaptando según fuera pertinente.

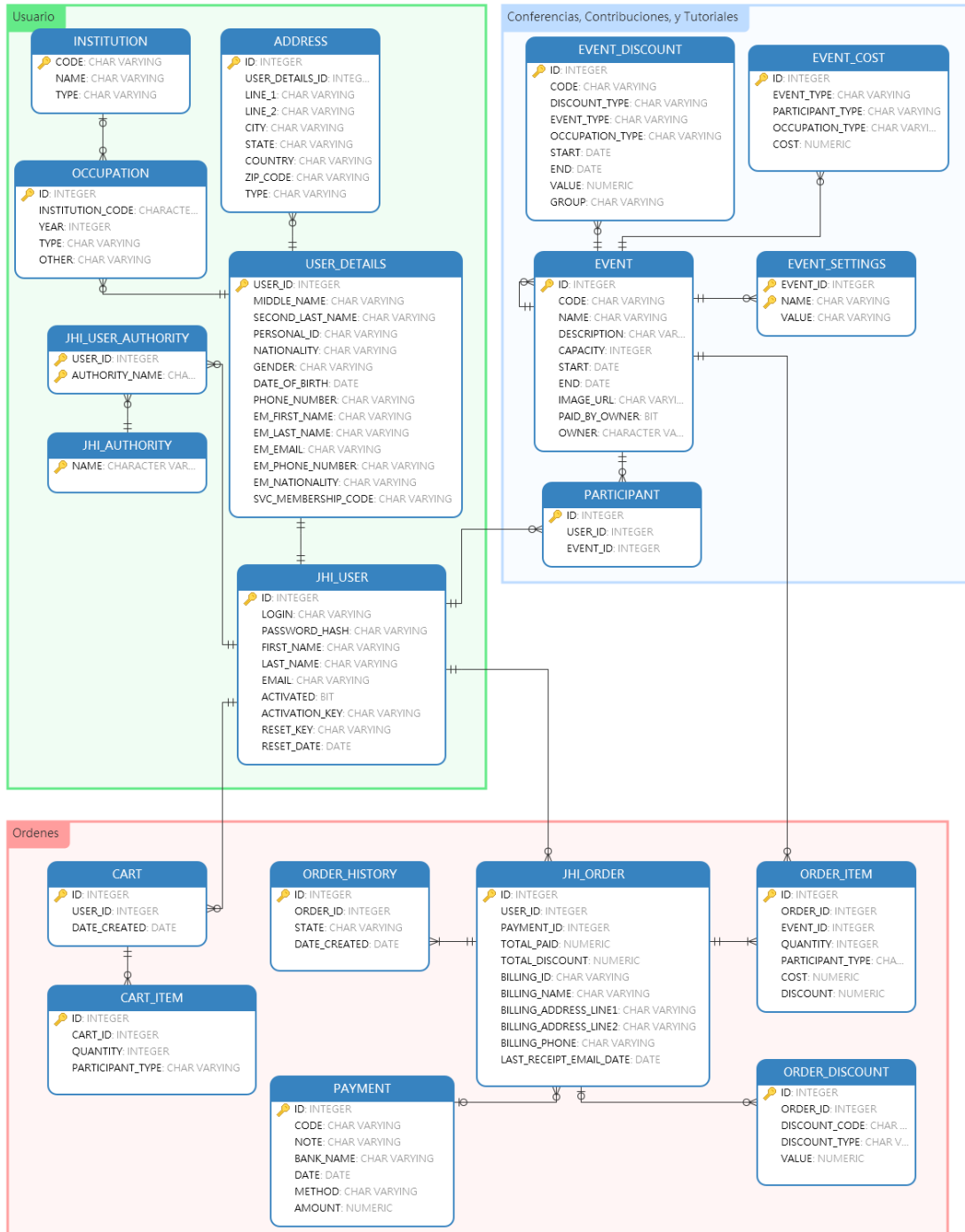


Figura 8.3: Modelo Final de la Base de Datos

8.3.3. Desarrollo de la Primera Fase

Siguiendo los lineamientos de la metodología, a continuación, se presenta una lista de los elementos desarrollados en la aplicación, dirigidos por las etapas establecidas en la metodología: Requisitos, Análisis, Prototipaje, y Entrega.

Requisito y Análisis

Título	Registro de usuarios
Descripción	Funcionalidad que permitirá a los nuevos usuarios registrarse para hacer uso de la aplicación, capturando los datos necesarios para la operación de las conferencias.

Prototipos

Registro

Por favor rellena todos los campos requeridos para crear tu cuenta

Datos de la Cuenta

<input type="text" value="Usuario"/>	<input type="text" value="Nueva contraseña"/>
<input type="text" value="Correo electrónico"/>	<input type="text" value="Confirme Contraseña"/>

Datos Personales

<input type="text" value="Primer Nombre"/>	<input type="text" value="Primer Apellido"/>
<input type="text" value="Segundo Nombre"/>	<input type="text" value="Segundo Apellido"/>
<input type="text" value="Fecha de Nacimiento (AAAA-MM)"/>	<input type="button" value="Hombre"/> <input type="button" value="Mujer"/>
<input type="text" value="Venezuela"/> ▼	<input type="text" value="V-00000000"/>

Datos de Contacto

<input type="text" value="Dirección Línea 1"/>	<input type="text" value="Dirección Línea 2"/>
<input type="text" value="Ciudad"/>	<input type="text" value="Estado"/>
<input type="text" value="Venezuela"/> ▼	<input type="text" value="Codigo Postal"/>
<input type="text" value="Teléfono"/>	

Afiliación

Estudiante de Pregrado

Seleccione una Institución Año de ingreso

Estudiante de Postgrado

Otro Año de ingreso

Nombre de la Institución

Profesor Universitario

Seleccione una Institución

Profesional de la Computación

Nombre de la Institución

Soy miembro de la Sociedad Venezolana de Computación (SVC)

Código de membresía

Contacto de Emergencia

Primer Nombre Primer Apellido

Teléfono Correo electrónico

Venezuela

Registrar

Figura 8.4: Prototipo de Registro de Usuario

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el formulario de registro (ver Figura 8.4) está vacío o alguno de los campos requeridos no está completado.</p> <p><i>Cuando</i> trata de hacer clic en el botón “Registrar”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El botón “Registrar” no está disponible y, 2) No realiza ninguna acción.
Prueba de Aceptación 2	<p><i>Dado que</i> únicamente los campos requeridos están completados o todos los campos del formulario están completados.</p> <p><i>Cuando</i> trata de hacer clic en el botón “Registrar”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) Se visualiza un mensaje indicando el registro exitoso del usuario y, 2) El usuario es redireccionado a la página de inicio de sesión y, 3) Un correo electrónico es enviado para activar la cuenta.
Prueba de Aceptación 3	<p><i>Dado que:</i></p>

	<p>1) Únicamente los campos requeridos están completados o todos los campos del formulario están completados y,</p> <p>2) El nombre de usuario o el correo electrónico ya existe en el sistema.</p> <p><i>Quando</i> trata de hacer clic en el botón “Registrar”.</p> <p><i>Entonces</i> se visualiza un mensaje de error indicando que el usuario ya existe.</p>
--	---

Tabla 8.1: Etapas de Registro de Usuario

Requisito y Análisis	
Título	Autenticación de usuario
Descripción	Funcionalidad que permite verificar la identidad del usuario, para el uso de la aplicación.

Prototipos

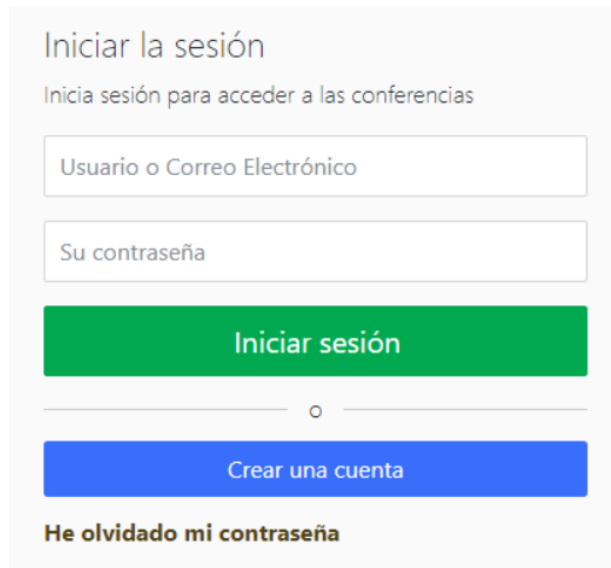


Figura 8.5: Prototipo de Inicio de Sesión de Usuario

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el formulario de inicio de sesión (ver Figura 8.5) está vacío o alguno de los campos requeridos no está completado.</p> <p><i>Quando</i> trata de hacer clic en el botón “Iniciar sesión”.</p> <p><i>Entonces</i> El botón “Iniciar sesión” no está disponible y no realiza ninguna acción.</p>
Prueba de Aceptación 2	<p><i>Dado que</i> el formulario de registro está completado.</p> <p><i>Quando</i> trata de hacer clic en el botón “Iniciar sesión”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El botón “Iniciar sesión” está disponible y, 2) El usuario es redireccionado a la página de selección de conferencia.
Prueba de Aceptación 3	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El formulario de registro está completado y,

	<p>2) El usuario no ha sido activado. <i>Cuando</i> trata de hacer clic en el botón “Iniciar sesión”. <i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El botón “Iniciar sesión” está disponible y, 2) Se visualiza un mensaje de error indicando que el usuario no está activo.
--	--

Tabla 8.2: Etapas de Autenticación de Usuario

Requisito y Análisis	
Título	Autorización de usuario
Descripción	Funcionalidad que garantiza el acceso a usuarios a otras funcionalidades de la aplicación de acuerdo a los permisos asignados.

Prototipos

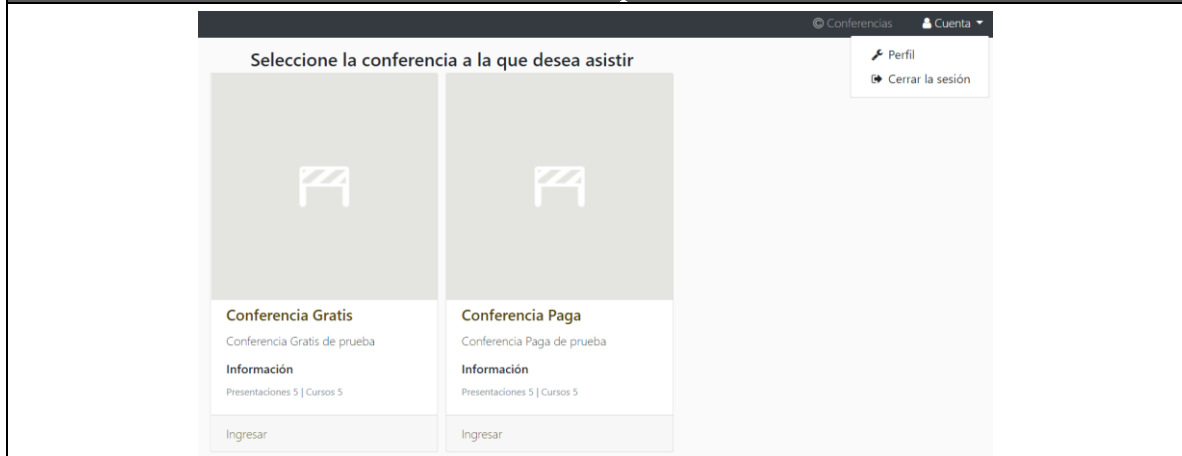


Figura 8.6: Prototipo de Página de Inicio de Usuario Autenticado

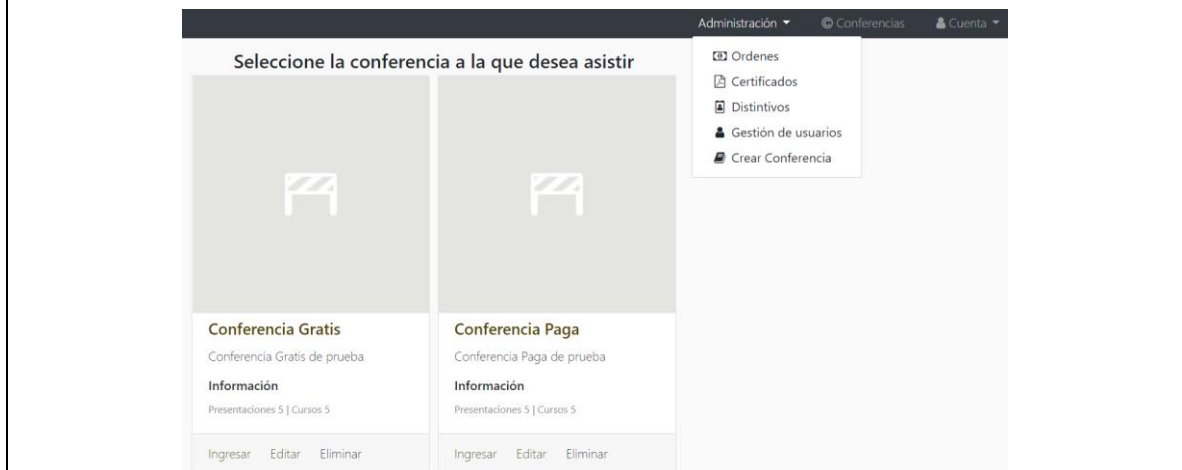


Figura 8.7: Prototipo de Página de Inicio de Administrador Autenticado

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario ha iniciado sesión con el rol Usuario. <i>Cuando</i> visualiza las acciones disponibles. <i>Entonces:</i></p>

	<ol style="list-style-type: none"> 1) No está disponible la sección de administración de conferencias (ver Figura 8.6). 2) No está disponible la sección de administración de usuarios. 3) Se listan las conferencias existentes en el sistema.
Prueba de Aceptación 2	<p><i>Dado que</i> el usuario ha iniciado sesión con el rol Administrador. <i>Cuando</i> visualiza las acciones disponibles. <i>Entonces:</i></p> <ol style="list-style-type: none"> 1) Está disponible la sección de administración de conferencias (ver Figura 8.7). 2) Está disponible la sección de administración de usuarios. 3) Se listan las conferencias existentes en el sistema.

Tabla 8.3: Etapas de Autorización de Usuarios a Secciones de la Aplicación


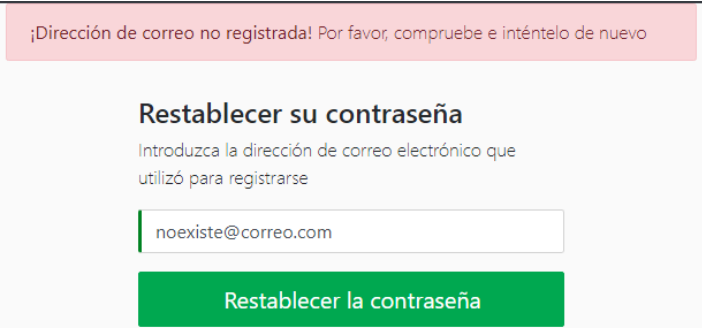
Requisito y Análisis	
Título	Recuperación de contraseña
Descripción	Funcionalidad que permite a un usuario recuperar y restablecer su contraseña
Prototipos	
 <p>The image shows a web form titled "Restablecer su contraseña". It contains a text input field with the placeholder "Correo electrónico" and a green button labeled "Restablecer la contraseña".</p>	
<p>Figura 8.8: Prototipo de Recuperación de Contraseña</p>	
 <p>The image shows a red error message banner: "¡Dirección de correo no registrada! Por favor, compruebe e inténtelo de nuevo". Below it is the same "Restablecer su contraseña" form, but the input field contains the text "noexiste@correo.com".</p>	
<p>Figura 8.9: Prototipo de Error Recuperación de Contraseña. Correo Inválido</p>	

Figura 8.10: Prototipo de Restablecimiento de Contraseña

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario ha hecho clic en el enlace “He olvidado mi contraseña”.</p> <p><i>Cuando</i> ingresa un correo electrónico válido.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El botón “Restablecer la contraseña” está disponible (ver Figura 8.8). 2) Al hacer clic sobre el botón “Restablecer la contraseña”, se envía un correo electrónico a la dirección de correo indicada incluyendo un enlace a la aplicación.
Prueba de Aceptación 2	<p><i>Dado que</i> el usuario ha hecho clic en el enlace “He olvidado mi contraseña”.</p> <p><i>Cuando</i> ingresa un correo electrónico inválido.</p> <p><i>Entonces</i> se muestra un mensaje de error al usuario (ver Figura 8.9).</p>
Prueba de Aceptación 3	<p><i>Dado que</i> el usuario ha iniciado el proceso de recuperación de contraseña.</p> <p><i>Cuando</i> hace clic en el enlace incluido en el correo de recuperación de contraseña.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El usuario es dirigido a la página de Restablecimiento de Contraseña (ver Figura 8.10). 2) El botón “Guardar” no está disponible hasta que los campos requeridos tengan los valores adecuados. 3) Una vez botón “Guardar” esté disponible y se hace clic sobre él, el registro del usuario es actualizado con la nueva contraseña indicada.
Prueba de Aceptación 4	<p><i>Dado que</i> la contraseña ha sido restablecida.</p> <p><i>Cuando</i> se intenta iniciar sesión usando la nueva contraseña.</p> <p><i>Entonces</i> el usuario es autenticado exitosamente.</p>
Prueba de Aceptación 5	<p><i>Dado que</i> has restablecida la contraseña.</p> <p><i>Cuando</i> se intenta iniciar sesión usando la antigua contraseña.</p> <p><i>Entonces</i> el usuario recibe un mensaje de error.</p>

Tabla 8.4: Etapas de Recuperación de Contraseña

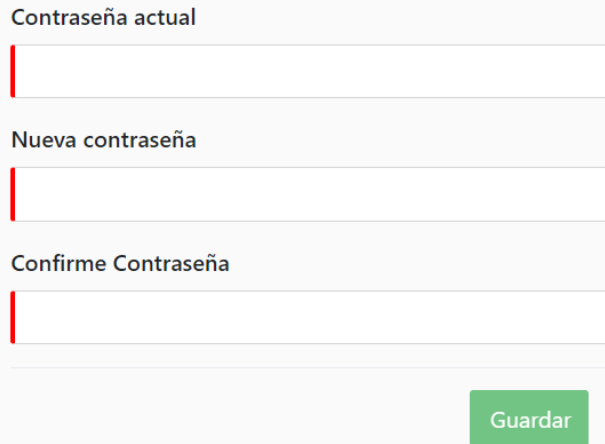
Requisito y Análisis	
Título	Cambio de contraseña
Descripción	Funcionalidad que permite a un usuario autenticado cambiar la contraseña.
Prototipos	
 <p>El prototipo muestra un formulario con tres campos de texto: 'Contraseña actual', 'Nueva contraseña' y 'Confirme Contraseña'. Cada campo tiene un ícono de ojo para alternar visibilidad. Debajo de los campos hay un botón verde que dice 'Guardar'.</p>	
<p>Figura 8.11: Prototipo de Cambio de Contraseña</p>	
Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario ha iniciado</p> <p><i>Cuando</i> el usuario hace clic en la opción de cambio de contraseña en el perfil.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El usuario es dirigido a la página de Restablecimiento de Contraseña (ver Figura 8.11). 2) El botón “Guardar” no está disponible hasta que los campos requeridos tengan los valores adecuados.
Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario ha iniciado, y 2) Se encuentra en la pantalla de cambio de contraseña, y 3) Todos los datos requeridos han sido ingresados correctamente <p><i>Cuando</i> el usuario hace clic en el botón “Guardar”.</p> <p><i>Entonces</i> el registro del usuario es actualizado con la nueva contraseña indicada.</p>

Tabla 8.5: Etapas de Cambio de Contraseña

Requisito y Análisis	
Título	Reservación de entrada a Conferencia como Asistente a conferencia paga
Descripción	Funcionalidad que permite al usuario autenticado realizar reservas como Asistente a la conferencia.
Prototipos	

Tipo de Participación	Costo
<input checked="" type="checkbox"/> Asistente a la Conferencia	1.000,00
<input type="checkbox"/> Autor de la Contribución	0,00
Descuento SVC (Particular, Tardío) 50%	-500,00
Sub-total	500,00

Figura 8.12: Prototipo de Reservación de Entradas a Conferencia como Asistente a Conferencia Paga

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario ha iniciado. <i>Cuando</i> el usuario ha seleccionado a una conferencia de la lista de conferencias (ver Figura 8.6) <i>Entonces</i> el usuario es redirigido a la página de reservación.</p>
Prueba de Aceptación 2	<p><i>Dado que</i> el usuario se encuentra en la página de reservación. <i>Cuando</i> el usuario selecciona la opción “Asistente a la Conferencia” (ver Figura 8.12). <i>Entonces</i>:</p> <ol style="list-style-type: none"> 1) El monto Sub-Total es actualizado con el costo de la asistencia a la conferencia, y 2) Si el usuario aplica a un descuento, se muestra en la pantalla como se muestra en la Figura 8.12, y 3) Al hacer clic sobre el botón “Reservar”, se genera una reserva del usuario a la conferencia como Asistente.

Tabla 8.6: Etapas de Reservación de Entrada a Conferencia como Asistente a Conferencia Paga

Requisito y Análisis	
Título	Reservación de entrada a Conferencia como Autor a conferencia paga
Descripción	Funcionalidad que permite al usuario autenticado realizar reservas como Autor de la conferencia.
Prototipos	

Tipo de Participación	Costo
<input type="checkbox"/> Asistente a la Conferencia	1.000,00
<input checked="" type="checkbox"/> Autor de la Contribución	1.350,00
<input checked="" type="checkbox"/> Artículo 1	1.000,00
<input checked="" type="checkbox"/> Artículo 2	350,00
<input type="checkbox"/> Artículo 3	350,00
<input type="checkbox"/> Artículo 4	350,00
<input type="checkbox"/> Artículo 5	350,00
Descuento SVC (Autor, Tardío) 50%	-675,00
Sub-total	675,00

Figura 8.13: Prototipo de Reservación de Entradas a Conferencia como Autor a Conferencia Paga

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario se encuentra en la página de reservación. <i>Cuando</i> el usuario selecciona la opción “Autor de la Conferencia” <i>Entonces</i> se despliega el listado de contribuciones registradas para la conferencia (ver Figura 8.13).</p>
Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario se encuentra en la página de reservación, y 2) Ha seleccionado la opción “Autor de la Conferencia”, y 3) Se ha desplegado la lista de contribuciones. <p><i>Cuando</i> el usuario selecciona una o múltiples contribuciones de la lista. <i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El monto Sub-Total es actualizado con la suma de los costos de las contribuciones seleccionadas, y 2) Si el usuario aplica a un descuento, se muestra en la pantalla como se muestra en la Figura 8.13, y 3) Al hacer clic sobre el botón “Reservar”, se genera una reserva del usuario a la conferencia como Asistente.

Tabla 8.7: Etapas de Reservación de Entrada a Conferencia como Autor a Conferencia Paga

Requisito y Análisis	
Título	Reservación de entrada a Tutorial como asistente a conferencia paga
Descripción	Funcionalidad que permite al usuario autenticado realizar reservaciones a tutoriales dictados en la conferencia.
Prototipos	

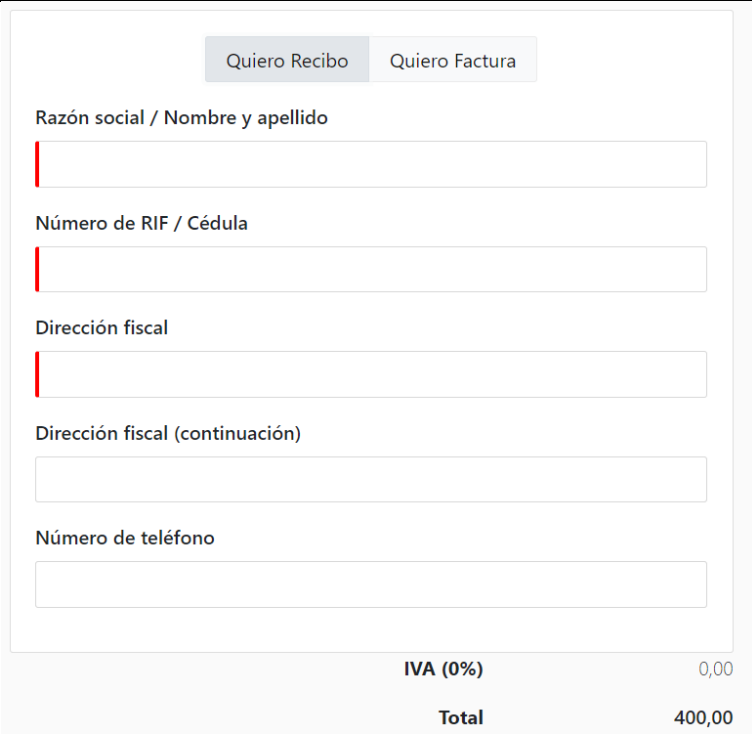
Tutoriales	
Información Tutorial	Costo
<input checked="" type="checkbox"/> COURSE-PAY-11: Tutorial 1 16 Oct, 2017. 10:30 AM - Disponible	800,00
<input type="checkbox"/> COURSE-PAY-12: Tutorial 1 16 Oct, 2017. 09:30 AM - Disponible Conflicto	800,00
<input type="checkbox"/> COURSE-PAY-21: Tutorial 2 17 Oct, 2017. 04:00 AM - 01:00 PM Disponible Conflicto	400,00
<input type="checkbox"/> COURSE-PAY-22: Tutorial 2 17 Oct, 2017. 04:00 AM - 01:00 PM Disponible Conflicto	400,00
<input type="checkbox"/> COURSE-PAY-31: Tutorial 3 18 Oct, 2017. 04:00 AM - 07:30 AM Disponible Conflicto	800,00
<input type="checkbox"/> COURSE-PAY-32: Tutorial 3 18 Oct, 2017. 04:00 AM - 07:30 AM Disponible Conflicto	800,00
<input type="checkbox"/> COURSE-SOLD-3: Tutorial 4 18 Oct, 2017. 04:00 AM - 07:30 AM Agotado	800,00
Descuento SVC (Particular, Tardío) 50%	-400,00
Sub-total	400,00

Figura 8.14: Prototipo de Reservación de Entrada a Tutorial como Asistente a Conferencia Paga

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario se encuentra en la página de reservación. <i>Cuando</i> el usuario selecciona una o más opciones de reservaciones a tutoriales (ver Figura 8.14). <i>Entonces</i> el monto Sub-Total es actualizado con el costo de la asistencia a la conferencia.</p>
Prueba de Aceptación 2	<p><i>Dado que</i> el usuario se encuentra en la página de reservación. <i>Cuando</i> el usuario selecciona una opción de reservación a tutoriales <i>Entonces</i>:</p> <ol style="list-style-type: none"> 1) Aquellas opciones cuyos horarios se solapan con la opción seleccionada estarán deshabilitadas para su selección, y 2) Las opciones en conflictos serán marcadas con una etiqueta con el mensaje "Conflicto".
Prueba de Aceptación 3	<p><i>Dado que</i>:</p> <ol style="list-style-type: none"> 1) El usuario se encuentra en la página de reservación, y 2) Existe un tutorial cuya capacidad ha sido agotada.

	<p><i>Quando se visualiza la opción de selección del tutorial con capacidad agotada.</i></p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) La opción con capacidad agotada estará deshabilitada para su selección, y 2) Las opciones con capacidad agotada será marcada con una etiqueta con el mensaje “Agotado”.
--	--

Tabla 8.8: Etapas de Reservación de Entrada a Tutorial como Asistente a Conferencia Paga

Requisito y Análisis					
Título	Reservación con Recibo de Pago o Factura				
Descripción	Funcionalidad que permite al usuario indicar si desea Factura o Recibo de Pago.				
Prototipos					
 <p>The image shows a web form for reservation with payment receipt. At the top, there are two buttons: 'Quiero Recibo' (selected) and 'Quiero Factura'. Below are five input fields with red error bars on the left: 'Razón social / Nombre y apellido', 'Número de RIF / Cédula', 'Dirección fiscal', 'Dirección fiscal (continuación)', and 'Número de teléfono'. At the bottom right, there is a summary table:</p> <table style="margin-left: auto; margin-right: 0;"> <tr> <td style="text-align: right;">IVA (0%)</td> <td style="text-align: right;">0,00</td> </tr> <tr> <td style="text-align: right;">Total</td> <td style="text-align: right;">400,00</td> </tr> </table>		IVA (0%)	0,00	Total	400,00
IVA (0%)	0,00				
Total	400,00				
<p>Figura 8.15: Prototipo de Reservación con Recibo de Pago</p>					

Quiero Recibo Quiero Factura

Razón social / Nombre y apellido

Número de RIF / Cédula

Dirección fiscal

Dirección fiscal (continuación)

Número de teléfono

Cargo por emisión de factura	2.500,00
IVA (9%)	261,00
Total	3.161,00

Figura 8.16: Prototipo de Reservación con Factura

Entrega	
Prueba de Aceptación 1	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario se encuentra en la página de reservación, y 2) Ha seleccionado las opciones de asistencia. <p><i>Cuando</i> el usuario visualiza el formulario.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El botón con la opción “Quiero Recibo” está presente (ver Figura 8.15), y 2) El botón con la opción “Quiero Factura” está presente (ver Figura 8.16). 3) Los campos de recibo o facturación están presentes.
Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario se encuentra en la página de reservación, y 2) Ha seleccionado las opciones de asistencia, y 3) Ha seleccionado la opción “Quiero Recibo”, y 4) Los campos requeridos de Recibo han sido completados. <p><i>Cuando</i> el usuario hace clic sobre el botón “Reservar”.</p> <p><i>Entonces</i> se genera una reserva exitosa con la configuración seleccionada.</p>
Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario se encuentra en la página de reservación, y 2) Ha seleccionado las opciones de asistencia, y 3) Ha seleccionado la opción “Quiero Factura”, y 4) Los campos requeridos de Factura han sido completados.

	<p><i>Cuando</i> el usuario hace clic sobre el botón “Reservar”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) Se visualiza el costo fijo por facturación, y 2) Se visualiza el porcentaje y monto del IVA aplicado al monto acumulado, y 3) Se genera una reserva exitosa con la configuración seleccionada.
--	--

Tabla 8.9: Etapas de Reservación con Recibo de Pago o Factura

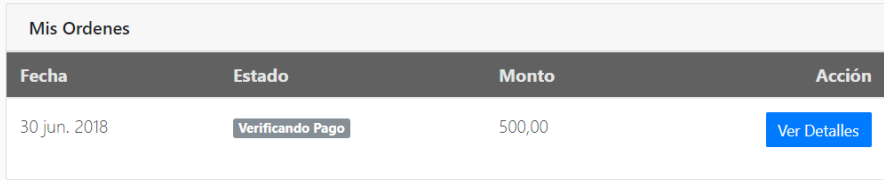
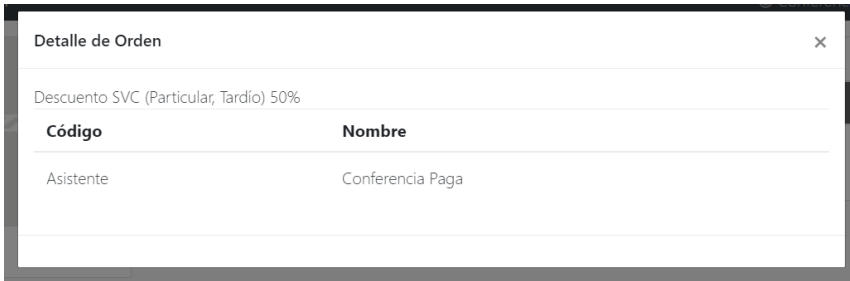
Requisito y Análisis	
Título	Lista de Reservaciones
Descripción	Funcionalidad que permite al usuario hacer seguimiento a las reservaciones realizadas en la conferencia.
Prototipos	
 <p>The image shows a web interface titled 'Mis Ordenes'. It contains a table with columns: Fecha, Estado, Monto, and Acción. One row is visible with the date '30 jun. 2018', the status 'Verificando Pago', and the amount '500,00'. There is a blue button labeled 'Ver Detalles' next to the row.</p>	
<p>Figura 8.17: Prototipo de Lista de Reservas</p>	
 <p>The image shows a 'Detalle de Orden' pop-up window. It displays a discount: 'Descuento SVC (Particular, Tardío) 50%'. Below this is a table with two columns: 'Código' and 'Nombre'. One row is visible with 'Asistente' in the code column and 'Conferencia Paga' in the name column.</p>	
<p>Figura 8.18: Prototipo de Detalle de Reserva</p>	
Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario ha realizado una reserva.</p> <p><i>Cuando</i> el usuario visualiza la página.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El usuario puede visualizar el listado de reservas con sus respectivos estatus (ver Figura 8.17). 2) Cada entrada de la lista de reservas tiene asociado una acción para ver el detalle de la reserva.
Prueba de Aceptación 2	<p><i>Dado que</i> el usuario ha realizado una reserva.</p> <p><i>Cuando</i> el usuario hace clic en el botón “Ver Detalles”.</p> <p><i>Entonces</i> una ventana “pop up” aparece mostrando el detalle relevante de la reserva (ver Figura 8.18).</p>

Tabla 8.10: Etapas de Lista de Reservaciones

Requisito y Análisis	
Título	Envío de información de Pago
Descripción	Funcionalidad que permite al usuario enviar información referente al pago de la reservación.

Prototipos

Información de pago

Realizar los pagos a la siguiente cuenta

Nombre: Fundación CLEI Venezuela
RIF: J-40156380-5
Banco: Mercantil
Tipo de cuenta: Corriente
Número de cuenta: 0105-0039-73-8039029821
Email: eventos@svc.net.ve

Número de voucher

Fecha de pago

Monto Pagado

Figura 8.19: Prototipo de Envío de Información de Pago con Depósito

Número de transferencia

Banco origen

Fecha de pago

Monto Pagado

Figura 8.20: Prototipo de Envío de Información de Pago con Transferencia

Figura 8.21: Prototipo de Envío de Información de Pago Exonerado

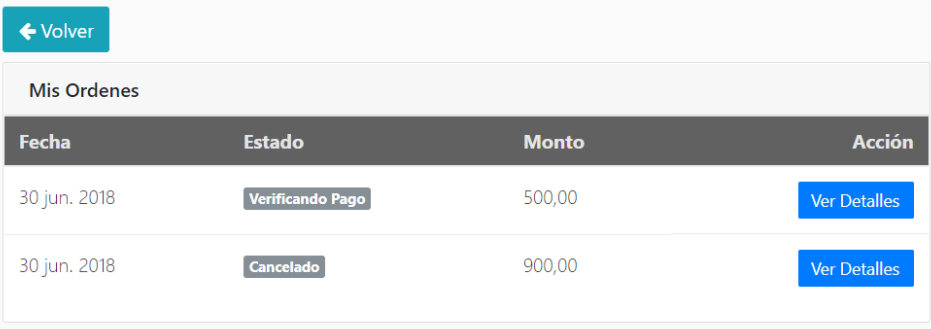
Entrega	
Prueba de Aceptación 1	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario ha realizado una reserva, y 2) El usuario ha ingresado a la página de la conferencia. <p><i>Cuando</i> el usuario visualiza el formulario.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El botón “Enviar Pago” está presente (ver Figura 8.19), y 2) El botón “Enviar Pago” no está disponible para hacer clic hasta que los campos requeridos tengan valor.
Prueba de Aceptación 2	<p><i>Dado que</i> llena todos los campos con datos correctos (ver Figura 8.19, Figura 8.20, y Figura 8.21).</p> <p><i>Cuando</i> hace clic en el botón “Enviar Pago”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El Estado de la reserva es actualizada a “Verificando Pago”. 2) El formulario de Envío de Pago no es visible.

Tabla 8.11: Etapas de Envío de Información de Pago

Requisito y Análisis	
Título	Cancelación de Reserva
Descripción	Funcionalidad que permite a un usuario cancelar la reserva.
Prototipos	
Figura 8.22: Prototipo de Cancelación de Reserva	
Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario envía los datos de pago de la reservación.</p> <p><i>Cuando</i> visualiza la Lista de reservaciones.</p> <p><i>Entonces</i> el botón “Cancelar” se encuentra disponible para cancelar la reserva (ver Figura 8.22).</p>
Prueba de Aceptación 2	<p><i>Dado que</i> el usuario envía los datos de pago de la reservación.</p>

	<p><i>Cuando</i> el usuario hace clic en el botón “Cancelar”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El Estado de la reserva es actualizada a “Cancelada”, y 2) El usuario está habilitado para hacer nuevas reservas en la conferencia.
--	---

Tabla 8.12: Etapas de Cancelación de Reserva

Requisito y Análisis	
Título	Visualizar Historial de Reservas
Descripción	Funcionalidad que permite al usuario ver todas las ordenes realizadas de la conferencia.
Prototipos	
	
<p>Figura 8.23: Prototipo de Visualización de Historial de Reservas</p>	

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario ingresa a una conferencia paga.</p> <p><i>Cuando</i> el usuario hace clic en la opción “Mis Órdenes” (ver Figura 8.23).</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El usuario es dirigido a la página de historial de reservas, y 2) Puede visualizar la lista de las ordenes hechas por el usuario en la conferencia.

Tabla 8.13: Etapas de Visualizar Historial de Reservas

Requisito y Análisis	
Título	Administración de Usuario
Descripción	Funcionalidad que permite a un usuario administrador visualizar usuarios registrados, eliminar usuarios, activar/desactivar usuarios, asignar/remover roles.
Prototipos	





















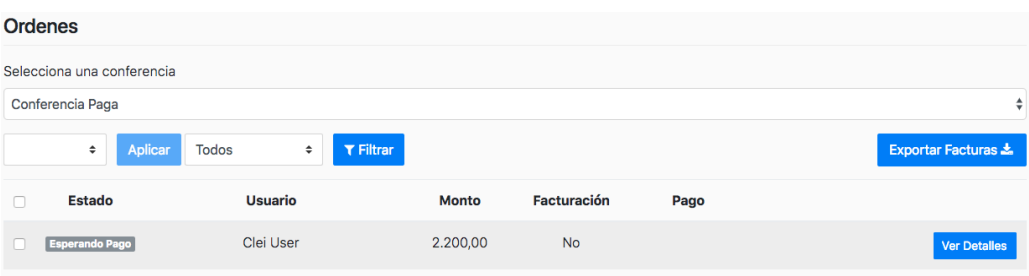
Usuarios								
ID	Nombre	Apellidos	Login	Email	Perfiles	Fecha de creación		
1	Superuser	Superuser	su	su@confv	Activado	ROLE_SU ROLE_USER ROLE_ADMIN	12/05/18 10:32	  
2	System	System	system	system@confv	Activado		12/05/18 10:32	  
3	Administrator	Administrator	admin	admin@confv	Activado	ROLE_USER ROLE_ADMIN	12/05/18 10:32	  
4	Svc	User	svc-undergrad	user1@confv	Activado	ROLE_USER	31/12/17 21:00	  
5	Svc	User	svc-postgrad	user2@confv	Activado	ROLE_USER	31/12/17 21:00	  
6	Svc	User	svc-professor	user3@confv	Activado	ROLE_USER	31/12/17 21:00	  
7	Svc	User	svc	user4@confv	Activado	ROLE_USER	31/12/17 21:00	  

Figura 8.24: Prototipo de Administración de Usuarios

Entrega	
Prueba de Aceptación	<p><i>Dado que</i> un usuario administrador inicia sesión. <i>Cuando</i> el usuario hace clic en la opción “Gestión de Usuarios” en el menú “Administración”.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) El usuario es redirigido a la página de administración de usuarios (ver Figura 8.24), y 2) Puede visualizar los botones de acciones sobre los registros, “Visualizar Perfil”, “Editar”, y “Eliminar”.

Tabla 8.14: Etapas de Administración de Usuario

Requisito y Análisis	
Título	Administración de Reservas
Descripción	Funcionalidad que permite al usuario administrador visualizar todas las reservas de una conferencia, aceptar/rechazar reservas, filtrar lista de reservas.
Prototipos	
	
Figura 8.25: Prototipo de Administración de Ordenes	
Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> un usuario administrador ha iniciado sesión. <i>Cuando</i> el usuario hace clic en la opción “Ordenes” del menú “Administración”.</p>

	<i>Entonces</i> el usuario es dirigido a la página de Ordenes (ver Figura 8.25).
Prueba de Aceptación 2	<i>Dado</i> que un usuario administrador se encuentra en la página de Ordenes. <i>Cuando</i> selecciona una conferencia. <i>Entonces</i> : <ol style="list-style-type: none"> 1) Se despliega el listado de las reservas realizadas por los usuarios para la conferencia, y 2) Puede visualizar filtros y acciones sobre el listado de órdenes.
Prueba de Aceptación 3	<i>Dado que</i> : <ol style="list-style-type: none"> 1) Un usuario administrador se encuentra en la página de Ordenes, y 2) Selecciona una conferencia, y 3) Selecciona elementos del listado de órdenes. <i>Cuando</i> : <ol style="list-style-type: none"> 1) Selecciona un estado para actualizar, y 2) Hace clic en el botón “Aplicar”. <i>Entonces</i> todos los registros seleccionados son actualizados al estado seleccionado.
Prueba de Aceptación 4	<i>Dado que</i> : <ol style="list-style-type: none"> 1) un usuario administrador se encuentra en la página de Ordenes, y 2) Selecciona una conferencia, y 3) Selecciona un Estado para filtrar el listado. <i>Cuando</i> hace clic en el botón “Filtrar”. <i>Entonces</i> el listado solamente muestra aquellos registros cuyo Estado sea igual al seleccionado en el filtro.
Prueba de Aceptación 5	<i>Dado que</i> <ol style="list-style-type: none"> 1) Un usuario administrador se encuentra en la página de Ordenes, y 2) Selecciona una conferencia. <i>Cuando</i> hace clic en el botón “Exportar Facturas”. <i>Entonces</i> un archivo con formato CSV es descargado automáticamente con la información relevante de facturación.

Tabla 8.15: Etapas de Administración de Reservas

Requisito y Análisis	
Título	Administración de Certificados de Asistencia
Descripción	Funcionalidad que permite al usuario administrador generar, descargar, enviar por correo electrónico los certificados de asistentes por tutorial; generar, descargar, enviar por correo electrónico los certificados de autor por contribución.
Prototipos	

Certificados

Selecciona una conferencia

Conferencia Paga

Asistentes

Asistentes

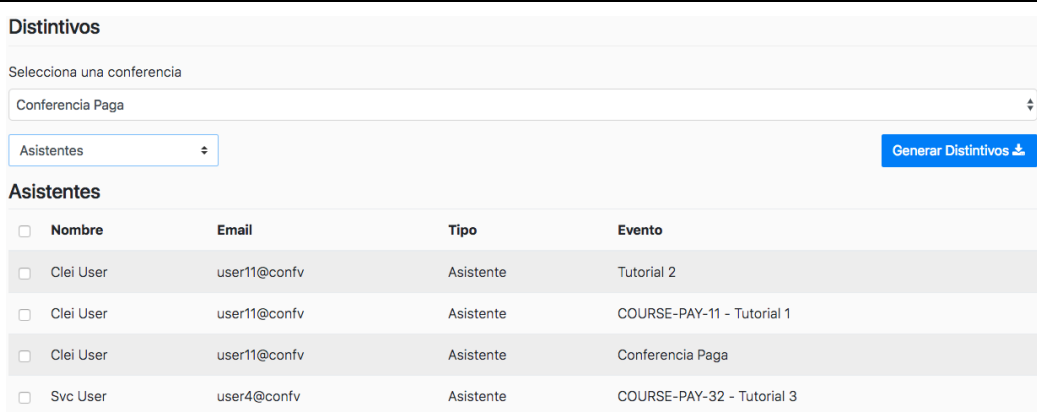
<input type="checkbox"/>	Nombre	Email	Usuario	Accion
<input type="checkbox"/>	User Clei	user11@confv	public	<input type="button" value="Descargar"/>
<input type="checkbox"/>	User Svc	user4@confv	svc	<input type="button" value="Descargar"/>

Figura 8.26: Prototipo de Administración de Certificados de Asistencia

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> un usuario administrador ha iniciado sesión.</p> <p><i>Cuando</i> hace clic en la opción “Certificados” del menú “Administración”.</p> <p><i>Entonces:</i> el usuario es dirigido a la página de Administración de Certificados (ver Figura 8.26).</p>
Prueba de Aceptación 2	<p><i>Dado que</i> un usuario administrador se encuentra en la página de Certificados.</p> <p><i>Cuando</i> selecciona una conferencia.</p> <p><i>Entonces:</i></p> <ol style="list-style-type: none"> 1) Se despliega el listado de los asistentes a la conferencia, y 2) Puede visualizar filtros y acciones sobre el listado de asistentes.
Prueba de Aceptación 3	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador se encuentra en la página de Certificados, y 2) Selecciona una conferencia, y 3) Selecciona una o más entradas del listado. <p><i>Cuando</i> hace clic en el botón “Exportar CSV”.</p> <p><i>Entonces</i> un archivo con formato CSV es descargado automáticamente con la información relevante de las entradas seleccionadas.</p>
Prueba de Aceptación 4	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador se encuentra en la página de Certificados, y 2) Selecciona una conferencia, y 3) Selecciona una o más entradas del listado. <p><i>Cuando</i> hace clic en el botón “Enviar Certificados”.</p> <p><i>Entonces</i> un correo electrónico será enviado a cada usuario seleccionado en la lista con el certificado en formato PDF adjunto con la información del certificado de asistencia.</p>
Prueba de Aceptación 5	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador se encuentra en la página de Certificados, y

	<p>2) Selecciona una conferencia. <i>Cuando</i> hace clic en el botón “Descargar” de una entrada del listado. <i>Entonces</i> un archivo con formato PDF es descargado automáticamente con la información del certificado de asistencia.</p>
--	--

Tabla 8.16: Etapas de Administración de Certificados de Asistencia

Requisito y Análisis	
Título	Administración de Distintivos
Descripción	Funcionalidad que permite al usuario administrador generar y descargar distintivos de asistentes, autor de contribución, y asistente a tutorial de la Conferencia.
Prototipos	
	
<p>Figura 8.27: Prototipo de Administración de Distintivos</p>	
Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> un usuario administrador ha iniciado sesión. <i>Cuando</i> hace clic en la opción “Distintivos” del menú “Administración”. <i>Entonces</i> el usuario es dirigido a la página de Administración de Distintivos (ver Figura 8.27).</p>
Prueba de Aceptación 2	<p><i>Dado que</i> un usuario administrador se encuentra en la página de Distintivos. <i>Cuando</i> selecciona una conferencia. <i>Entonces:</i></p> <ol style="list-style-type: none"> 1) Se despliega el listado de los asistentes a la conferencia, y 2) Puede visualizar filtros por tipo de participación: Autor de contribución o Asistente.
Prueba de Aceptación 3	<p><i>Dado que</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador se encuentra en la página de Distintivos, y 2) Selecciona una conferencia, y 3) Selecciona una o más entradas del listado. <p><i>Cuando</i> hace clic en el botón “Generar Distintivos”.</p>

	<i>Entonces</i> un archivo con formato PDF es descargado automáticamente con los distintivos de participación de los usuarios seleccionados.
--	--

Tabla 8.17: Etapas de Administración de Distintivos

8.3.4. Desarrollo de la Segunda Fase

A continuación, se presenta una lista de los elementos desarrollados en la aplicación, establecido en la Sección 8.2.2.


Requisito y Análisis	
Título	Edición de perfil de usuario
Descripción	Funcionalidad que permite a los usuarios regulares editar su perfil de usuario y a los usuarios administradores editar el perfil de cualquier usuario.
Prototipos	
	

Figura 8.28: Prototipo de Edición de Perfil de Usuario como Usuario Administrador

Figura 8.29: Prototipo de Edición de Perfil de Usuario como Usuario Regular

Entrega

<p>Prueba de Aceptación 1</p>	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) El usuario administrador ha iniciado sesión, y 2) Ha hecho clic en la opción “Gestión de usuarios” del menú “Administración”. <p><i>Cuando</i> hace clic en el botón “Visualizar” de un registro de usuario.</p> <p><i>Entonces</i> es redirigido al perfil del usuario (ver Figura 8.28).</p>
<p>Prueba de Aceptación 2</p>	<p><i>Dado que</i> el usuario administrador ha ingresado al perfil de usuario de un usuario.</p> <p><i>Cuando:</i></p> <ol style="list-style-type: none"> 1) Edita los datos, y 2) Hace clic en el botón “Guardar”. <p><i>Entonces</i> los datos del perfil de usuario son actualizados con los nuevos datos.</p>
<p>Prueba de Aceptación 3</p>	<p><i>Dado que</i> el usuario administrador ha ingresado al perfil de usuario de un usuario.</p> <p><i>Cuando</i> edita los datos con valores incorrectos o incompletos.</p> <p><i>Entonces</i> el botón “Guardar” no estará disponible para hacer clic.</p>
<p>Prueba de Aceptación 4</p>	<p><i>Dado que</i> el usuario regular ha iniciado sesión.</p> <p><i>Cuando</i> hace clic en la opción “Perfil” el menú “Cuenta”.</p> <p><i>Entonces</i> es redirigido al perfil del usuario (ver Figura 8.29)</p>
<p>Prueba de Aceptación 5</p>	<p><i>Dado que</i> el usuario regular ha ingresado a su perfil de usuario de usuario.</p> <p><i>Cuando</i> edita los datos y hace clic en el botón “Guardar”.</p> <p><i>Entonces</i> los datos del perfil de usuario son actualizados con los nuevos datos.</p>

Prueba de Aceptación 6	<p><i>Dado que</i> el usuario regular ha ingresado a su perfil de usuario de usuario.</p> <p><i>Cuando</i> edita los datos con valores incorrectos o incompletos y hace clic en el botón “Guardar”.</p> <p><i>Entonces</i> el botón “Guardar” no estará disponible para hacer clic.</p>
------------------------	---

Tabla 8.18: Etapas de Edición de Perfil de Usuario

Requisito y Análisis	
Título	Administración de Conferencias
Descripción	Funcionalidad que permite al usuario administrador crear, editar, y eliminar conferencias.

Prototipos

Figura 8.30: Prototipo de Creación y Edición de Conferencia

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> el usuario administrador ha iniciado sesión.</p> <p><i>Cuando</i> hace clic en la opción “Crear” el menú “Administración”.</p> <p><i>Entonces</i> el usuario es dirigido a la página de Crear Conferencia (ver Figura 8.30).</p>

Tabla 8.19: Etapas de Administración de Conferencias

Requisito y Análisis	
Título	Edición de Costos y Descuentos de Conferencia
Descripción	Funcionalidad que permite al usuario administrador editar los costos y descuentos de una conferencia.

Prototipos

Conference Manager | Administración | Conferencias

Conferencias / Costo asistente

Costos
Complete el siguiente formulario para configurar los costos para los asistentes.

General | Profesor | Estudiante

Costo sin descuento
1000

Descuento inscripción temprana

General
% 25.5 | Costo: 745

CLEI
% 50 | Costo: 500

SVC/Local
% 75 | Costo: 250

Descuento inscripción tardía

Figura 8.31: Prototipo de Edición de Costos de Asistencia y Descuentos de Conferencias

Costos
Complete el siguiente formulario para configurar los costos de los autores.

Costo sin descuento
1000

Artículo adicional
650

Descuento inscripción temprana

General
% 25 | Costo: 750 | Adicional: 262.5

CLEI
% 50 | Costo: 500 | Adicional: 175

SVC/Local
% 75 | Costo: 250 | Adicional: 87.5

Descuento inscripción tardía

General

Figura 8.32: Prototipo de Edición de Costos de Autores y Descuentos de Conferencia

Entrega

Prueba de Aceptación 1

Dado que:

- 1) Un usuario administrador crea una nueva conferencia o edita una conferencia existente, y

	<p>2) Selecciona modificar los costos de la conferencia para Asistentes (ver Figura 8.31) o Autores (ver Figura 8.32), y <i>Cuando</i> hace clic en las pestañas “General”, “Profesor”, o “Estudiante”.</p> <p><i>Entonces</i> muestra el formulario de edición de acuerdo con cada opción.</p>
Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador crea una nueva conferencia o edita una conferencia existente y 2) Selecciona modificar los costos de la conferencia para asistentes (ver Figura 8.31), y <p><i>Cuando:</i></p> <ol style="list-style-type: none"> 1) Llena el formulario, y 2) Hace clic en el botón “Guardar”. <p><i>Entonces</i> los datos de costo de los asistentes son actualizados.</p>

Tabla 8.20: Etapas de Edición de Costos y Descuentos de Conferencia

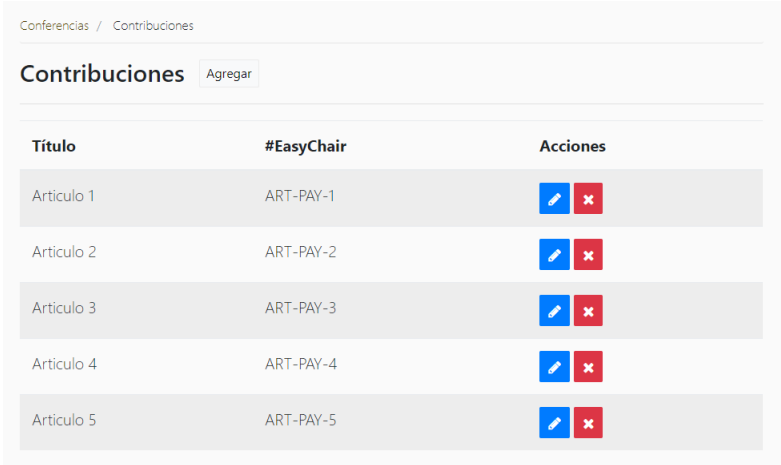
Requisito y Análisis																			
Título	Administración de Contribuciones																		
Descripción	Funcionalidad que permite al usuario administrador crear, editar, y eliminar contribuciones de la conferencia.																		
Prototipos																			
 <p>The screenshot shows a web interface for managing contributions. At the top, there is a breadcrumb "Conferencias / Contribuciones" and a section header "Contribuciones" with an "Agregar" button. Below this is a table with three columns: "Título", "#EasyChair", and "Acciones". The table contains five rows, each representing an article. Each row has a blue pencil icon for editing and a red 'x' icon for deleting.</p> <table border="1"> <thead> <tr> <th>Título</th> <th>#EasyChair</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Artículo 1</td> <td>ART-PAY-1</td> <td>[Pencil] [X]</td> </tr> <tr> <td>Artículo 2</td> <td>ART-PAY-2</td> <td>[Pencil] [X]</td> </tr> <tr> <td>Artículo 3</td> <td>ART-PAY-3</td> <td>[Pencil] [X]</td> </tr> <tr> <td>Artículo 4</td> <td>ART-PAY-4</td> <td>[Pencil] [X]</td> </tr> <tr> <td>Artículo 5</td> <td>ART-PAY-5</td> <td>[Pencil] [X]</td> </tr> </tbody> </table>		Título	#EasyChair	Acciones	Artículo 1	ART-PAY-1	[Pencil] [X]	Artículo 2	ART-PAY-2	[Pencil] [X]	Artículo 3	ART-PAY-3	[Pencil] [X]	Artículo 4	ART-PAY-4	[Pencil] [X]	Artículo 5	ART-PAY-5	[Pencil] [X]
Título	#EasyChair	Acciones																	
Artículo 1	ART-PAY-1	[Pencil] [X]																	
Artículo 2	ART-PAY-2	[Pencil] [X]																	
Artículo 3	ART-PAY-3	[Pencil] [X]																	
Artículo 4	ART-PAY-4	[Pencil] [X]																	
Artículo 5	ART-PAY-5	[Pencil] [X]																	
<p>Figura 8.33: Prototipo de Administración de Contribuciones</p>																			



Figura 8.34: Prototipo de Creación y Edición de Contribución

Entrega	
Prueba de Aceptación 1	<p><i>Dado que</i> un usuario administrador crea una nueva conferencia o edita una conferencia existente.</p> <p><i>Cuando</i> selecciona el menú de contribuciones.</p> <p><i>Entonces</i> se muestra el listado de contribuciones existentes (ver Figura 8.33)</p>
Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador crea una nueva conferencia o edita una conferencia existente, y 2) Se encuentra en la página de contribuciones. <p><i>Cuando:</i></p> <ol style="list-style-type: none"> 1) Hace clic en el botón “Agregar” <p><i>Entonces</i> se levanta una ventana con el formulario para agregar una nueva contribución (ver Figura 8.34)</p>

Tabla 8.21: Etapas de Administración de Contribuciones

Requisito y Análisis	
Título	Administración de Tutoriales
Descripción	Funcionalidad que permite al usuario administrador crear, editar, y eliminar tutoriales de la conferencia.
Prototipos	

Conferencias / Tutoriales

Tutoriales Agregar















Nombre	Código	Cupo	Fecha	Hora	Duración	Acciones
Tutorial 1	COURSE-PAY-11	23	16 oct. 2017	10:30	8 horas	 
Tutorial 1	COURSE-PAY-12	23	16 oct. 2017	9:30	8 horas	 
Tutorial 2	COURSE-PAY-21	24	17 oct. 2017	4:0	4 horas	 
Tutorial 2	COURSE-PAY-22	24	17 oct. 2017	4:0	4 horas	 
Tutorial 3	COURSE-PAY-31	23	18 oct. 2017	4:0	8 horas	 
Tutorial 3	COURSE-PAY-32	23	18 oct. 2017	4:0	8 horas	 
Tutorial 4	COURSE-SOLD-3	0	18 oct. 2017	4:0	8 horas	 


Figura 8.35: Prototipo de Administración de Tutoriales

Crear tutorial ×

Código

Nombre

Capacidad

Fecha inicio  **Hora Inicio** HH : MM

Duración

Figura 8.36: Prototipo de Creación y Edición de Tutoriales

Entrega

Prueba de Aceptación 1

Dado que un usuario administrador crea una nueva conferencia o edita una conferencia existente.
Cuando selecciona el menú de tutoriales.
Entonces se muestra el listado de tutoriales existentes (ver Figura 8.35).

Prueba de Aceptación 2	<p><i>Dado que:</i></p> <ol style="list-style-type: none"> 1) Un usuario administrador crea una nueva conferencia o edita una conferencia existente, y 2) Se encuentra en la página de tutoriales. <p><i>Cuando</i> hace clic en el botón “Agregar”.</p> <p><i>Entonces</i> se levanta una ventana con el formulario para agregar un nuevo tutorial (ver Figura 8.36).</p>
------------------------	--

Tabla 8.22: Etapas de Administración de Tutoriales

8.3.5. Pruebas y Resultados

Las pruebas de entrega o aceptación definidas en las Secciones 8.2.1 y 8.2.2, definen el comportamiento esperado del sistema bajo ciertos escenarios de negocio mínimos. Estas pruebas han sido ejecutadas satisfactoriamente, confirmando la efectividad de la aplicación.

Adicional a estas pruebas de aceptación, la aplicación también implementa una serie de pruebas unitarias automatizadas que garantizan el funcionamiento de componentes claves de la aplicación (e.g., el motor de precios encargado de calcular los costos de las reservas). En la Figura 8.37 se puede ver que todas las pruebas automatizadas fueron pasadas de manera exitosa y pueden ser ejecutadas en cualquier momento desde el proyecto, permitiendo la verificación de la salud del sistema.

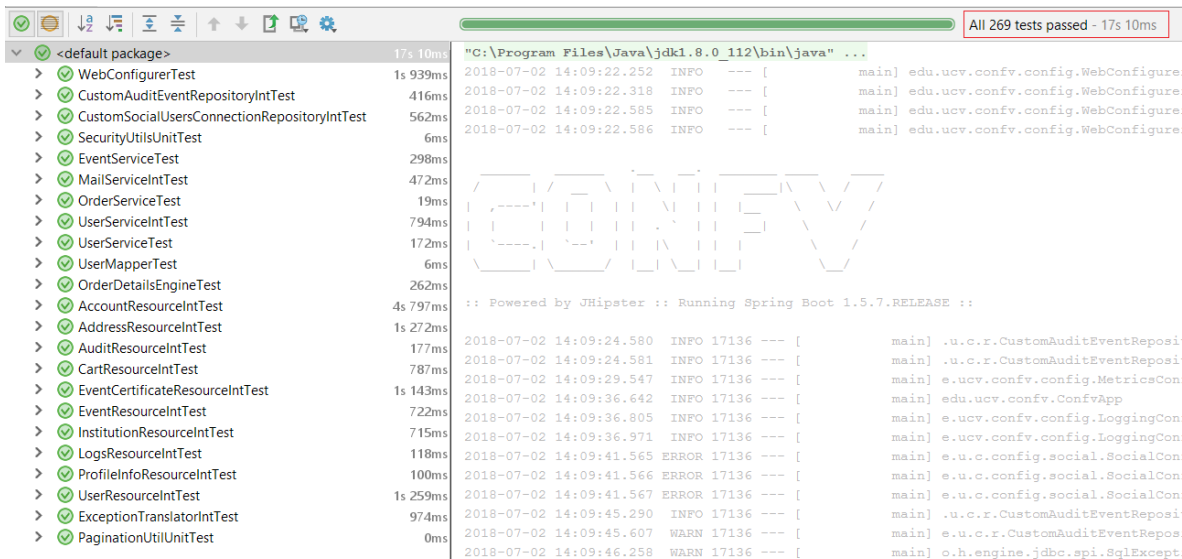


Figura 8.37: Resultado de Ejecución de Pruebas Automatizadas

La aplicación se desplegó exitosamente en las entregas de CoNCISa 2017. En esta primera instancia, la aplicación cumplió su objetivo de facilitar la organización de la conferencia al automatizar los procesos administrativos. Cabe mencionar que durante

esta entrega surgieron defectos que tuvieron que ser solventados a través de actualizaciones a la aplicación durante la conferencia.

Como se mencionó en el Capítulo 8.2, esta primera entrega contiene solo una parte de las funcionalidades implementadas. Estas funcionalidades no incluían mecanismos para editar o modificar datos de la conferencia. Como solución temporal, cualquier actualización requerida a los datos, debía implementarse como *script* SQL y se ejecutaba en producción luego de pasar por el ambiente de pruebas. Con este enfoque, se tuvo una solución al problema de no tener la capacidad de actualizar datos de la aplicación y, aun siendo un enfoque manual, permitió a los organizadores usar la aplicación en esta entrega de una manera exitosa y a tiempo.

9. Conclusiones

Las conferencias o eventos académicos se presentan como uno de los métodos de distribución de conocimientos más efectivos y usados a nivel mundial. La organización de estos eventos puede consumir mucho tiempo en tareas repetitivas que pueden ser automatizadas con las tecnologías existentes en la actualidad.

Se captaron los requerimientos del comité organizador del CoNCISa y SCTC, durante el Análisis General de la aplicación, y de manera iterativa a lo largo del desarrollo de la aplicación, a través de llamadas telefónicas e intercambio de correos electrónicos con el usuario.

De igual forma, durante el Análisis General, se diseñó la arquitectura general y el modelo inicial de la base de datos.

A través de iteraciones de desarrollo, siguiendo la metodología AgilUs, se implementaron y probaron las funcionalidades de la aplicación. Esto se logró a través de pruebas de aceptación y pruebas unitarias automatizadas.

La aplicación se ejecutó de manera exitosa en el CoNCISa 2017 y 2018. Generando un aporte de valor significativo para los organizadores de las conferencias, en relación a la optimización de sus procesos.

Esta aplicación generó un ahorro de tiempo y dinero a la organización de las conferencias académicas. Por ejemplo, ahorro en dinero por disminución del uso de insumos de impresión, al poder enviar los documentos digitales a través de correo electrónico a los participantes, de una manera sencilla. Estos documentos, en particular los certificados de asistencia y participación, tomaban aproximadamente 5 minutos para poder generarlos de forma manual. La aplicación genera todos estos certificados de forma automática, lo cual resulta en ahorro significativo de tiempo y esfuerzo. De igual forma, el Comité de Inscripción, que inicialmente estaba conformado por un aproximado de 8 personas, ahora sólo son necesarias 2 personas dedicadas a esta tarea.

9.1. Limitaciones

Durante el desarrollo de la aplicación, un problema recurrente era la compatibilidad de todos los componentes usados para la aplicación, en particular, los componentes del lado del Front-End. Estos requerían un cuidado particular ya que, al actualizar alguno de los componentes a una versión no compatible con el resto, rompía la estabilidad de la aplicación. Para esto, se recurrió a hacer uso de un versionamiento más estricto en el Front-End con el uso de NPM. De esta manera, una vez comprobada la compatibilidad

entre componentes, estos no debían actualizarse sin realizar las pruebas pertinentes previamente.

Un tema recurrente cuando se usan componentes de código abierto desarrollados por la comunidad, es que algunas veces la documentación es inexistente, incompleta, desactualizada, o insuficiente. Esto requiere esfuerzo adicional para su uso más allá de los casos básicos. De igual forma, cuando se presentan errores inesperados y contrarios a lo especificados por los autores de los componentes, un esfuerzo adicional es requerido para investigar dichos errores y solventarlos adecuadamente.

Otra limitación encontrada durante el proyecto, fue el corto tiempo entre el inicio del desarrollo y la primera entrega de la aplicación. Esto influyo considerablemente la etapa de análisis para considerar las mejores tecnologías y enfoques a usar para desarrollar la aplicación.

9.2. Trabajos Futuros

Si bien la aplicación desarrollada cumple con los objetivos planteados, también puede servir como base para futuros desarrollos, extensiones, o mejoras a la aplicación.

Como trabajos futuros, se pueden recomendar los siguientes:

- Evaluar los beneficios del uso *offline* o sin conexión de la aplicación.
- Creación de un módulo de reportes y estadísticas. Estos, ayudaran en la toma de decisiones para futuras entregas de las conferencias.
- Creación de módulo de administración de recursos como salas de conferencias y horarios de presentaciones y tutoriales.
- Creación de módulo de Generación de las Memorias de la Conferencia.
- Ampliar el grado de cobertura de pruebas automatizadas tanto en el Back-End como en el Front-End.
 - Pruebas automatizadas *End-to-End* (e.g. Selenium).
 - Pruebas de carga y desempeño (e.g. JMeter).
 - Pruebas unitarias en Front-End (e.g. Jasmin, Protractor).
 - Pruebas unitarias y de integración en Back-End (e.g. JUnit).
- Uso de componentes en el Front-End, que aumenten las capacidades de la aplicación, facilitando la interacción con el usuario. Por ejemplo, hacer uso de tablas interactivas que admitan filtro para todas las columnas y paginación.

- Hacer uso de contenedores (e.g. Docker) para permitir una evolución del stack tecnológico del proyecto.
- Configurar procesos de Integración Continua (*Continuous Integration*) y Entrega Continua (*Continuous Delivery*).
- Implementar configuraciones a nivel de infraestructura, para garantizar alta disponibilidad.

Referencias Bibliográficas

- [1] T. Rogers, “*Conferences and Conventions: A Global Industry*”, Routledge, Enero 2013.
- [2] New World Encyclopedia, “*Academic Conference*”, http://www.newworldencyclopedia.org/entry/Academic_conference
- [3] SCTC, “*Simposio Científico y Tecnológico en Computación*”, <http://www.sctc.org.ve>
- [4] CoNCISa, “*Conferencia Nacional de Computación, Informática y Sistemas*”, <http://www.concisa.net.ve>
- [5] D. Hill, “*Architectural Patterns and Styles*”, Microsoft Developer Network, <https://msdn.microsoft.com/en-us/library/ee658117.aspx>
- [6] D. Garlan y M. Shaw, “*An Introduction to Software Architecture*”, CMU-CS-94-166, New Jersey, Enero 1994.
- [7] S. K. Singh y S. C. Yadav, “*An Introduction to Client/Server Computing*”, New Delhi: New Age International, Diciembre 2009.
- [8] S. Luján, “*Programación de Aplicaciones Web: Historia, Principios Básicos y Cliente Web*”, San Vicente: Editorial Club Universitario, Noviembre 2002.
- [9] R. Fielding y J. Reschke, “*Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*”, RFC 7230, Junio 2014.
- [10] PKP, “*Open Conference Systems*”, <https://pkp.sfu.ca/ocs>
- [11] OpenConf, “*OpenConf News*”, <https://www.openconf.com/news/#20140613>
- [12] EasyChair, “*EasyChair Home*”, <http://easychair.org>
- [13] A. E. Acosta, “*Agilus: Construcción Ágil de la Usabilidad*”, Caracas, 2011.
- [14] CERN, “*The Birth of the Web*”, <http://cds.cern.ch/record/1998446>
- [15] R. Nixon, “*Learning PHP, MySQL, JavaScript, CSS & HTML.*”, Tercera ed., O’Reilly, Mayo 2014.
- [16] E. Shepherd, “*HTML5*”, Mozilla Developer Network, <https://developer.mozilla.org/es/docs/HTML/HTML5>
- [17] Mozilla Developer Network, “*CSS*”, <https://developer.mozilla.org/es/docs/Web/CSS>
- [18] A. Shenoy y U. Sossou, “*Learning Bootstrap*”, Packt Publishing, Diciembre 2014.
- [19] M. Á. Sánchez, “*JavaScript*”, Málaga: Innovación y Cualificación S.L., Enero 2012.
- [20] Chandermani, “*AngularJS by Example*”, Packt Publishing, Marzo 2015.
- [21] R. Jensen, “*Learning TypeScript*”, Birmingham: Packt Publishing, 2015.
- [22] J. Byous, “*Java Technology: The Early Years*”, http://srjstaff.santarosa.edu/~dpearson/mirrored_pages/java.sun.com/Java_Technology_-_An_early_history.pdf
- [23] C. Schaefer, C. Ho y R. Harrop, “*Pro Spring*”, Apress, Septiembre 2014.

- [24] H2, “*H2 Database*”, <http://www.h2database.com/html/main.html>
- [25] H2, “*H2 Database*”, http://www.h2database.com/html/advanced.html#limits_limitations
- [26] C. Baver, G. King y G. Gregory, “*Java Persistence with Hibernate*”, Manning Publications, 2016.
- [27] B. Lowagie, “*iText in Action*”, Manning Publications, Octubre 2010.
- [28] Adobe Systems Incorporated, “*PDF Reference*”, Adobe Systems Incorporated, Noviembre 2006.
- [29] JasperReports, “*The Definitive Guide to JasperReports*”, Apress, 2007.
- [30] M. Raible, “*The JHipster Mini-Book*”, C4Media, 2016.
- [31] npm, “*What is npm?*”, <https://docs.npmjs.com/getting-started/what-is-npm>
- [32] M. Clow, “*Angular 5 Projects*”, Apress, 2018.
- [33] S. Chacon y B. Straub, “*Pro Git*”, Apress, Marzo 2015.
- [34] Apache Maven Project, “*Welcome to Apache Maven Project*”, <https://maven.apache.org>
- [35] Sonatype, “*Maven: The Definitive Guide*”, O'Reilly, Agosto 2008.
- [36] R. S. Pressman, “*Software Engineering: A Practitioner’s Approach*”, 1997.
- [37] S. Priolo, “*Métodos Ágiles*”, Users, 2009.
- [38] K. E. Kendall y J. E. Kendall, “*Análisis y Diseño de Sistemas*”, Pearson, 2005.
- [39] P. Letelier, “*Proceso de Desarrollo de Software*”, DSIC, 2003.
- [40] F. Toro López, “*Administración de Proyectos de Informática*”, Ecoe Ediciones, 2013.
- [41] K. Beck, “*Manifiesto por el Desarrollo Ágil de Software*”, <http://agilemanifesto.org/iso/es/manifesto.html>
- [42] M. Fowler, “*The New Methodology*”, <https://martinfowler.com/articles/newMethodology.html>
- [43] A. V. Méndez, “*Metodologías de Desarrollo de Software*”, Instituto Tecnológico Superior de Apatzigán, 2010.