



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

**PROTOTIPO DE UN SISTEMA DE RESERVACIÓN DE ENTRADAS PARA
CINE Y EVENTOS USANDO TECNOLOGÍA HTML5 Y COMPONENTES JAVA**

**Trabajo de Titulación en conformidad con los requisitos establecidos para optar
por el título de Ingeniero en Sistemas de Computación e Informática**

Profesor Guía

Juan José León

Autor

Esteban Raul Capello Cabrera

Año

2013

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el/la estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

.....

Juan José León

Ing. Msc. Sistemas

1707506760

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

.....
Esteban Raul, Capello Cabrera

1715920490

AGRADECIMIENTOS

Agradezco a todas las personas que me apoyaron en la elaboración de este Trabajo de Titulación ya sea con su tiempo o palabras de aliento, y en especial a mi familia y a Juan José León por todo el tiempo y dedicación que utilizó para guiarme en este camino.

RESUMEN

La tendencia de la virtualización de recursos y ventas se presenta con mayor fuerza en el mundo actual, razón por la cual Ecuador no puede quedar atrás en esta evolución tecnológica, gracias a herramientas como las tarjetas de crédito es posible llevar a la comodidad de un computador personal la compra o reservación de los eventos ocurridos en una ciudad en la cual uno se encuentra.

Por este motivo se desarrolló un prototipo de un sistema de administración de eventos, centrado en la venta de entradas de manera anticipada, que mediante el uso de la última tendencia de herramientas tecnológicas como son HTML5 y diferentes componentes JAVA, el cual promete ser una ventana para aumentar las ventas en el área de mercado que aplica el presente trabajo.

El presente prototipo se lo concibió para demostrar la capacidad que pueden presentar este tipo software en mercado ecuatoriano y así impulsar a las organizaciones interesadas a realizar un desarrollo completo de la aplicación e incursionar en un nuevo modo de marketing.

ABSTRACT

There is a global tendency to virtualize almost every resource, including the pay and sale areas, becoming this an excellent reason for not letting Ecuador fall behind in this technological evolution, with tools like credit cards it has become usual to bring, to the comfort of a personal computer, the ease of purchasing or booking almost all kind of events in a city in which one finds oneself.

Because of this, it has been develop an event management prototype system, that focuses on purchase and booking of tickets beforehand, that by using the latest trend of technological tools such as HTML5 and different JAVA components, which promises to be a window to increase sales in the field of action of this work.

This prototype was conceived to demonstrate the capability of this type of software in the Ecuadorian market and thus encourage interested organizations to develop a full version of it, and so venture into a new way of marketing.

ÍNDICE

Introducción.....	1
Antecedentes.....	2
Justificación.....	3
Objetivos.....	4
Objetivo General.....	4
Objetivos Específicos.....	4
Alcance.....	5
Áreas de Aplicación.....	5
1. Capítulo I. Metodologías.....	8
1.1. Metodologías de Desarrollo de Software Tradicionales...8	
1.2. Metodologías de Desarrollo de Software Ágiles.....9	
1.3. Cuadro comparativo entre metodologías tradicionales y ágiles.....	10
1.4. Metodología Ágil Scrum.....	11
1.4.1. Iteración (Sprint).....	12
1.4.2. Actividades.....	12
1.4.3. Roles.....	14
1.4.4. Herramientas.....	16

1.5. Cuadro comparativo entre las metodologías Scrum, XP y Crystal Clear.....	17
1.6. Ventajas y Desventajas de la Metodología Scrum.....	18
1.7. Herramientas de Apoyo.....	19
1.7.1. Wireframes.....	19
1.7.2. Diagrama de Base de Datos (Entidad - Relación).....	21
1.7.3. Diagrama de Clases.....	24
1.7.4. Diagrama de Casos de Uso.....	28
1.7.5. Modelos de Navegación.....	31
2. Capítulo II. Tecnologías.....	33
2.1. HTML5.....	33
2.1.1. Independencia y Rendimiento.....	34
2.1.2. Almacenamiento.....	34
2.1.3. Gráficos.....	35
2.2. Spring 3 Framework.....	36
2.2.1. Contenedor de Control de Inversión e Inyección de Dependencias.....	36
2.2.2. Acceso a Datos.....	36
2.2.3. Administración de Transacciones.....	37
2.2.4. Spring 3 MVC.....	37
2.3. PrimeFaces.....	38

2.4. Hibernate.....	40
2.5. Servidor de Aplicaciones JBoss.....	41
2.6. JavaScript.....	41
3. Capítulo III. Diseño y Desarrollo.....	43
3.1. Scrum.....	43
3.1.1. Datos Generales.....	43
3.1.2. Product Backlog.....	44
3.1.3. Iteraciones.....	45
3.1.4. Scrum Daily Meeting.....	46
3.1.5. Sprint Review y Sprint Retrospective.....	47
3.1.6. Burn Down Chart.....	47
3.2. Modelos de Navegación.....	48
3.2.1. Roles.....	48
3.2.2. Navegación – Usuario sin cuenta.....	48
3.2.3. Navegación – Usuario Administrador.....	49
3.2.4. Navegación – Usuario Vendedor.....	49
3.2.5. Navegación – Usuario Cliente.....	50
3.3. WireFrames.....	51
3.3.1. Rol: Usuario sin Cuenta.....	51
3.3.2. Rol: Usuario Administrador.....	52
3.3.3. Rol: Usuario Cliente.....	58

3.3.4. Rol: Usuario Vendedor.....	60
3.4. Diagrama de Base de Datos (Entidad - Relación).....	61
3.5. Diagrama de Clases.....	62
3.6. Diagrama de Casos de Uso.....	63
3.6.1. Especificación de Caso de Uso: Registro.....	63
3.6.2. Especificación de Caso de Uso: Administración de localidades.....	64
3.7. Pruebas.....	70
4. Capítulo IV. Conclusiones y Recomendaciones.....	73
4.1. Conclusiones.....	73
4.2. Recomendaciones.....	74
5. Referencias.....	75

ÍNDICE DE ANEXOS

Anexo 1: Crystal Clear.....	81
Anexo 2: Extreme Programming (XP).....	86
Anexo 3: Funcionalidad Adicional de HTML5.....	92
Anexo 4: Ejemplo de HTML5.....	98
Anexo 5: Ejemplo en Spring 3 MVC.....	108
Anexo 6: Ejemplo en PrimeFaces.....	112
Anexo 7: Ejemplo usando Spring, PrimeFaces y Base de Datos.....	125
Anexo 8: Modelo Conceptual de la Base de Datos.....	141
Anexo 9: Manual de Instalación.....	142
Anexo 10: Product Backlog.....	146
Anexo 11: Sprints.....	152
Anexo 12: Daily Meetings.....	171
Anexo 13: WireFrames.....	203
Anexo 14: Casos de Uso.....	237
Anexo 15: Análisis Económico.....	277

Introducción

El desarrollo de software ha sido siempre uno de los principales aspectos de la Informática, al presentarse de una manera versátil y adaptable a cualquier campo de aplicación ha resultado fundamental para el desarrollo tecnológico en el mundo, sin importar su complejidad o la cantidad de lenguajes de programación que existen.

Este trabajo de titulación utilizó esta área de la informática, enfocada en la organización, venta y promoción de eventos, para demostrar un avance en el mercado ecuatoriano que por el momento se ha visto rezagado e, incluso de cierta forma, olvidado debido a la desconfianza existente en el medio de compras por internet.

Para la demostración de este potencial existente en el mercado, de eventos y entretenimiento, se planteará un prototipo de un sistema que rente su funcionalidad a cualquier empresa de eventos y que, con su información específica, pueda configurar y levantar su negocio para que el medio de los usuarios pueda encontrarlo e incluso adquirir su producto de una manera fácil, confiable y segura.

La adquisición de bienes y/o servicios por internet siempre ha tenido un puesto importante en el desarrollo económico de los países de primer mundo, sin embargo en Ecuador ha sido visto como un medio por el cual es posible que se estafe o robe al consumidor final. Esta idea ha ido cambiando conforme pasa el tiempo y los mismos clientes ecuatorianos ahora compran productos a empresas extranjeras sin el menor riesgo, pero todavía reside cierto miedo al momento de hacerlo con empresas nacionales.

El mercado del entretenimiento presenta una excelente oportunidad para cambiar esta manera de pensar del consumidor y empezar un proceso de fortalecimiento a las compras online que pueden abrir muchos campos de trabajo y desarrollo para el país.

Antecedentes

Durante mucho tiempo los eventos en el Ecuador han continuado funcionando de la misma manera, sin avanzar al mismo ritmo que la tecnología, por este motivo la sociedad actual, poco a poco, ha ido presentando quejas por la calidad del servicios, un ejemplo de esto se hizo público el 10 de junio del 2012, en el diario El Comercio:

“Para el cotejo ante Colombia, el pasado 10 de junio, salieron a la venta 37 353 localidades, que dejaron una recaudación de USD 977 348,75. Pero, en las gradas hubo cerca de 40 000, incluso ubicados debajo del marcador electrónico. Esto desató reclamos airados de hinchas que se quedaron afuera con boleto en mano. Y ese fue el detonante para la medida tomada por el Municipio” (El Comercio, 2012).

Esta manera de manejar la organización de un evento ha desembocado en una gran pérdida del mercado, no solo en ámbito económico sino en la acogida de cierto tipo de actividades.

El ámbito de los cines no es indiferente a esta realidad, donde se puede visualizar una evidente falta de progreso. Sin embargo no todo esto es culpa de la falta de organización y planificación sino también del medio que envuelve el crecimiento tecnológico.

“Hemos identificado dos problemas: la ausencia de sistemas de pago, como mencioné antes, para que los comercios puedan transaccionar en línea y recibir los pagos de la gente, y la otra es una especie de temor de los organismos financieros a los fraudes en línea” (Béjar, 2012).

Justificación

En la mayoría de locales de eventos en el país, se maneja de forma manual la venta y reservación de asientos para sus eventos. Esto ha funcionado, sin embargo también ha causado incomodidad en sus clientes. Esta incomodidad se puede traducir en una pérdida en ventas.

El Ecuador se ha visto desactualizado en la forma de manejar este tipo de eventos, en otros países como Italia existe la opción de realizar compras anticipadas (hasta por meses) por internet, de boletos a funciones de teatro, boletos a museos, conciertos, etc.

Todos los aspectos aquí planteados brindan la opción de desarrollar un manejo integral del negocio de los eventos de una manera centralizada, ordenada y segura. Una aplicación web capaz de ofrecer a cada empresa el poder de administrar sus espectáculos de manera sencilla y que la misma presente a sus potenciales clientes la opción de anticipar sus compras.

Esto ayudará a que las organizaciones interesadas incursionar en un nuevo modo de marketing, para de esta manera aumentar sus ventas y calidad de atención al cliente, además de posibles ahorros en materiales para impresión de boletos.

Objetivos

Objetivo General

Realizar un prototipo de un sistema de administración de eventos, centrado en la venta de entradas de manera anticipada para poder proponer una idea y mejorar así el nivel de ventas y la aceptación de empresa ante el público.

Objetivos Específicos

- Comprender las nuevas tecnologías mediante la investigación de las mismas, para que incorporadas a JAVA puedan asegurar una aplicación eficiente y moderna.
- Implementar un prototipo de un sistema de administración de eventos que este enfocado plenamente a la venta anticipada de entradas, a la promoción de la empresa y a la satisfacción del cliente.
- Realizar una aplicación amigable para que cualquier usuario de la misma pueda comenzar a utilizar el sistema de forma fácil y cómoda.

Alcance

El prototipo está enfocado al área de entretenimiento que comprende eventos y cines, da la facilidad de que una empresa dé sus datos y se le proporcione el servicio según las especificaciones que tenga su negocio, además provee la capacidad de actualizar los productos disponibles según sea necesario, para que los posibles clientes puedan acceder a toda la información necesaria en el momento oportuno.

El sistema proporciona la siguiente forma de pago:

- **Reservación de Boletos:** Presenta la opción de reservar los tickets de la función que se desea ingresar para poder pagar lo adeudado 60 minutos antes de ingresar al evento, caso contrario la reservación caducará.

Para poder realizar el prototipo se utilizará la metodología de desarrollo ágil Scrum. La implementación está dirigida para operar en un entorno web mediante el uso de Spring 3, prime faces y HTML5. Al estar basada en componentes JEE (Estándar para aplicaciones Empresariales Java) permite la separación en múltiples capas de programación. La información será guardada en el motor de base de datos PostgreSQL. Estos datos podrán ser accedidos en la aplicación por medio de una capa de persistencia que se encuentra en un servidor de aplicaciones. El uso de un servidor de aplicaciones da la facilidad al usuario de poder acceder a la información utilizando un navegador web.

Áreas de Aplicación

Mediante el enfoque provisto a este trabajo de titulación es posible abarcar muchos segmentos del mercado de eventos, donde la organización y despliegue de los asistentes sea uno de los factores primordiales, a continuación se detallarán algunos de estos y sus posibles usos:

Cines

En la venta de entradas de forma anticipada, dando la opción al consumidor de optar por una función y un horario que le complazca sin el riesgo de no poder encontrar una entrada para esa película.

Teatros

El acoplamiento a la tecnología en esta área es ideal tanto para mejorar en publicidad y oportunidad en mercado como para utilizar este nuevo tipo de publicidad para ofrecer al público un lugar a su gusto para ver la función y al mismo tiempo realizar preventas de eventos por venir.

Centros deportivos

Después del penoso incidente ocurrido en el estadio Atahualpa (Ver Antecedentes), el municipio será en encargado de regular las asistencias, aprovechando este cambio es posible entrar de igual manera al mercado de las ventas en línea para promocionar los partidos de fútbol que van a ocurrir en esta sede y vender a los usuarios el asiento numerado que deseen, evitándoles el problema de encontrar asientos contiguos cuando se desea ingresar con un grupo de personas grande.

Conciertos

Al igual que en los teatros, es posible realizar una preventa de un concierto en la localidad que el usuario desee, dando la comodidad de poder ir a canjear la entrada en un tiempo oportuno, sin molestias de colas y el riesgo de no poder ingresar al concierto que se desea atender por no poder acceder a la compra de entradas con anticipación.

Tours Guiados en medios de transporte

Los eventos turísticos también se pueden ver beneficiados en este campo, tal se da el ejemplo de los tours guiados en medios de transporte.

“Quito Tour Bus ofrece un recorrido interesante, cómodo y seguro en el podrás escuchar a través de un sistema de audio guía en español e inglés, las explicaciones de los sitios emblemáticos de esta ciudad milenaria, patrimonial y cosmopolita” (Empresa Pública Metropolitana Quito Turismo, 2012).

El sector turístico al estar en contacto con potenciales clientes extranjeros pueden presentar una especial ventaja al permitir la adquisición de sus boletos y tours guiados desde internet, permitiendo a los turistas una planificación segura, confiable y mejores opiniones sobre el turismo Ecuatoriano.

1. Metodologías

1.1 Metodologías de Desarrollo de Software Tradicionales

El desarrollo de software siempre ha acarreado constantes problemas al momento de realizarlo, ya sea por tiempos mal planteados, como por los medios en los que se debería ejecutar el mismo. Por esta razón siempre han existido distintas metodologías de desarrollo de software que intentan abarcar distintos ámbitos del mismo.

Algunas tienen por objetivo el focalizar el esfuerzo en el planteamiento del problema y estructuración del proyecto para su posterior ejecución, estas son las conocidas como metodología tradicionales. Su principal desventaja radica en la firmeza que existe al momento de no poder cambiar el plan de ejecución por más de que el proyecto lo requiera o el alto costo que esto significa.

Esto no significa que todas las metodologías tradicionales no “sirvan” para desarrollar un producto de software de manera exitosa, sino que deben ser analizadas a profundidad para poder evaluar qué proyecto se puede llevar a cabo con que metodología. Cada una de estas técnicas puede ser aplicada dependiendo del tamaño del proyecto así como de la dimensión del equipo de trabajo para el mismo, un ejemplo de esto nos presenta la metodología RUP, donde se recomienda el uso en proyectos de gran tamaño y equipos de trabajo conformados por más de 20 empleados.

Algunos ejemplos de este tipo de metodologías de desarrollo son:

- MSF
- RUP
- En Cascada
- En Espiral
- Etc.

1.2 Metodologías de Desarrollo de Software Ágiles

Dado a que el desarrollo de una aplicación esta, casi siempre, envuelto en un medio que constantemente cambia, también se han planteado diversas metodologías de desarrollo llamadas ágiles.

Estas estrategias nacen bajo la necesidad de que el software debe por cambiar conforme sea necesario, sin importar en la etapa de desarrollo en la cual se encuentre. Todas estas metodologías presentan sus fundamentos en el llamado "Manifiesto Ágil"

"Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas
Software funcionando sobre documentación extensiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda" (Beck et al., 2001).

Existen diversas metodologías de carácter ágil, sin embargo entre sus principales podemos destacar:

- Scrum
- XP (Xtreme Programming)
- Crystal Clear
- Agile MSF
- Agile Unified Process
- Entre otras

El presente proyecto de titulación utiliza, tanto para el desarrollo del mismo como de la aplicación propuesta, la metodología ágil Scrum, que será detallada más adelante (Ver punto 1.4 Metodología Ágil Scrum).

1.3 Cuadro comparativo entre las metodologías tradicionales y ágiles

A continuación se muestra una tabla realizada en base a las investigaciones sobre los diferentes tipos de metodologías, donde se resumen las principales diferencias que poseen entre dichos tipos.

Tabla 1: Comparación entre metodologías tradicionales y ágiles

	Tradicionales	Ágiles
Basado en estándares	Sigue protocolos y procesos definidos por buenas practicas	Su enfoque en la calidad, obliga el seguimiento de estándares para asegurarla
Control de Procesos	Por medio de modelos y regulaciones de cada metodología	Por medio de herramientas flexibles y roles para los ejecutores
Contrato Establecido	Rígido establecido con el cliente	Flexible, ya que el mismo cliente puede modificar ciertos aspectos
Cliente	Interactúa en Reuniones	Parte del Equipo
Adaptación al cambio	Una vez estructurada el modelo a seguir, es muy costos el cambiarlo	Que el cliente sea parte del equipo de trabajo ayuda a mantener el enfoque en la funcionalidad más que en un modelo
Equipo de Desarrollo	Recomendable para grupos grandes de personas	Desde Equipos de trabajo de 3 personas en adelante
Planificación anticipada	Por medio del uso de procesos de levantamiento de requerimientos y herramientas para establecer procesos	Se puede establecer tareas con muy poca anticipación, debido a los posibles cambios que pueden existir
Arquitectura de Software	Con ayuda de modelos se convierte en el tronco de desarrollo	Tiene menor importancia, para acceder a mayor flexibilidad
Principal Finalidad	El desarrollo completo del software, basado en los requerimientos del cliente	La completa satisfacción del cliente por medio de la funcionalidad

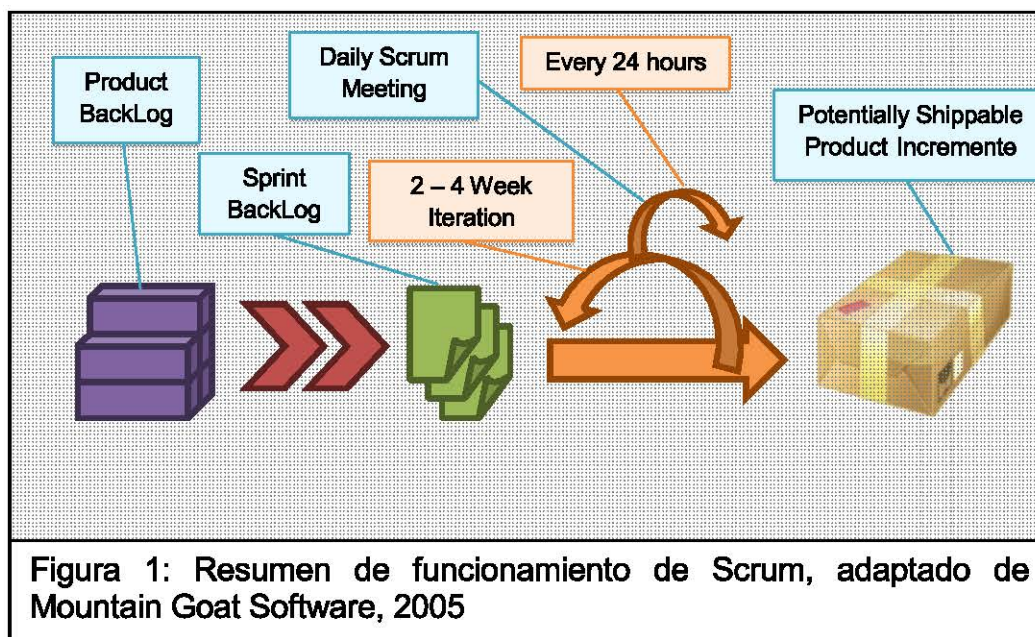
1.4 Metodología Ágil Scrum

Esta metodología fue definida en 1993 y normalizada en 1995 después de un análisis exhaustivo a los procesos para el desarrollo de productos que empresas como Canon, Xerox, Honda, HP, entre otras, empezaron a implementar alrededor de 1986. Estas compañías se emprendían en un entorno donde sus productos debían salir al mercado en un tiempo menor al de productos anteriores, así como el partir de ideas novedosas y que podían estar sujetas a cambios constantes en su fabricación.

“Jeff Sutherland creó el primer equipo de Scrum en 1993 y trabajó con Ken Schwaber para formalizar Scrum en OOPSLA'95 (Object-Oriented Programming, Systems, Languages & Applications). Juntos, expandieron y mejoraron Scrum en muchas compañías de software, ayudaron a escribir el Manifiesto Ágil en 2001, y fueron los autores la guía de Scrum” (Sutherland, s.f.).

Scrum, a diferencia de otras metodologías ágiles, es a veces considerada como una estructura de procesos más que un proceso completo o metodología para el desarrollo de software. Su principal fortaleza radica en su amplia predisposición al cambio así como su orientación principal a nivel de proyectos más que al desarrollo de la aplicación en sí.

El funcionamiento de Scrum viene dado por el establecimiento bidimensional de actividades y roles, además de tres herramientas principales, todo este conjunto de características más la continua repetición de las mismas es lo que conforman el marco de trabajo Scrum.



1.4.1 Iteración (Sprint)

Cada Sprint se lo debe realizar de manera continua con iteraciones de al menos 2 semanas y máximo 4 semanas, éste será el tiempo donde se ejecutarán todas las tareas establecidas para cada equipo (ver Sprint BackLog). Mediante un consenso se establecerá el tiempo de dicha ejecución en base a la experiencia del equipo y puede irse modificando de acuerdo a la complejidad del proyecto que se está realizando.

1.4.2 Actividades

Las actividades también son llamadas reuniones, estas son donde se debe establecer el camino a seguir en base a las decisiones que han sido tomadas con los resultados de cada iteración o de cada día.

Planificación de la liberación (Release Planning): Representa una o varias guías donde se identifica cuáles son las características del producto que se implementarán y cuando se las completará, y así saber cuándo se pueden presentar posibles entregables o la finalización del proyecto.

Para realizar un plan de liberación es necesario poseer un Product Backlog optimizado y con las estimaciones de tiempo de cada tarea para así seleccionar los objetivos deseados de cada liberación. Esta planificación suele envolver desde tres hasta doce iteraciones, esta cifra depende del proyecto, los objetivos, y de los entregables que se deseen dar del mismo.

Este no es un plan estático, siempre es posible ir acomodándolo conforme se haya modificado o re-calculado tareas de del Product Backlog, por lo que se sugiere que se lo revise después de cada sprint.

Planificación de la iteración (Sprint Planning Meeting): Es la primera reunión de cada iteración, donde se deben definir las tareas y tácticas que se tomarán para realizar las mismas, estas actividades deben poder ser realizadas para que finalizado el Sprint el Cliente pueda pedir una demostración (Entregable) y que esta no representa una gran inversión de tiempo ponerla en marcha.

Para la selección de tareas el cliente suele presentar una lista de priorizada (Product Backlog), donde cada miembro del equipo se auto-assigna las actividades que puede realizar para ir completando el Sprint Backlog de la iteración correspondiente. El cliente tiene la obligación de poder responder a las dudas que el equipo vaya teniendo durante la duración de esta reunión que no superará las 8 horas.

Reunión diaria del equipo (Scrum Daily Meeting): Es una actividad que se la debe realizar cada día después de que haya sido planificado el Sprint, donde se debe facilitar la información necesaria para que el equipo conozca el estado de las tareas de todos y cada uno de sus miembros, para así poder conocer las tareas que afectan de manera directa o indirecta a otro miembro del equipo. Esencialmente se deben responder las siguientes preguntas:

- ¿Qué se ha realizado desde ayer?
- ¿Qué es lo que se va a realizar el día de hoy?
- ¿Existe algún inconveniente para realizar las tareas?

Estas respuestas ayudarán al Scrum Master (Ver punto 1.4.3 Roles) a planificar y ayudar a resolver cualquier inconveniente de manera oportuna y precisa.

Scrum de Scrum: Esta reunión se la realizará en caso de que el proyecto sea realizado por más de un equipo de trabajo, donde en primera instancia cada equipo deberá designar a uno de sus integrantes para que asista a la actividad después de cada Scrum Daily Meeting.

La integración es el objetivo principal que deberán tener en cuenta todos los participantes ya que deberán contestar las siguientes preguntas:

- ¿Qué ha realizado tu equipo desde ayer?
- ¿Qué actividades realizará tu equipo hasta la siguiente reunión?
- ¿Existen impedimentos para realizar las tareas?
- ¿Puede que alguna actividad interfiera en el desarrollo de otro equipo?

Reunión de revisión de la iteración (Sprint Review Meeting): Actividad que debe realizarse en un tiempo aproximado de 4 horas, este tiempo puede variar dependiendo de la duración de cada sprint, donde se demostrarán todos los avances que se ha realizado en forma de un “demo” a la finalización de cada Sprint. Además es importante mencionar las tareas que no se han logrado completar para poder realizar adaptaciones al proyecto así como para ayudar a la siguiente Sprint Planning Meeting.

Retrospectiva de la iteración (Sprint Retrospective): Al final de cada Sprint es necesario llevar a cabo un análisis retrospectivo del mismo donde se deberán dejar plasmadas ideas y opiniones para una continua superación de cada Sprint. Al igual que en el Sprint Review Meeting el tiempo de esta reunión depende de la duración de la iteración que se está revisando, sin embargo es aconsejable utilizar un tiempo aproximado de 4 horas.

1.4.3 Roles

Se subdividen en dos grupos importantes donde cada responsabilidad debe ser realizada a cabalidad para la exitosa ejecución del proyecto.

Roles Principales

Corresponde a las representaciones que tiene un contacto directo con la elaboración del proyecto y sus tareas:

Product Owner: Es la persona encargada de dirigirse al equipo de trabajo en nombre de todos los interesados en el proyecto. Debe de poseer la suficiente autoridad para poder tomar decisiones y para plantear al equipo las tareas enfocándose en la perspectiva que tiene el negocio. Es el responsable de elaborar el Product Backlog y de priorizar las tareas del mismo para cada Sprint.

Scrum Master: No representa a un líder del proyecto, sino a un facilitador que hará todo lo posible por eliminar los obstáculos que se vayan presentando en el equipo de trabajo, su intermediación hace posible que la ejecución de la metodología de Scrum se la realice siguiendo las reglas establecidas por este proceso, además de proteger al equipo de cualquier tipo de interrupción que se pueda presentar en medio de cada Sprint.

Scrum Team: Está conformado por todas las personas (máximo 9) que poseen las habilidades necesarias para llevar a cabo el diseño, desarrollo, pruebas y documentación del proyecto. Este grupo auto-organizado de gente son los responsables de cuantificar tanto las tareas como el tiempo y complejidad que representa cada una de las mismas.

Roles Secundarios

Son todas aquellas personas que influyen mas no tienen una relación directa con la ejecución del proyecto:

Stakeholders: Son los principales beneficiarios de la culminación del proyecto y quienes justifican la producción del mismo. Este grupo de personas solo intervendrán directamente en las reuniones de revisión de cada Sprint

Administradores: Son los encargados de proporcionar un ambiente favorable para el desarrollo del producto.

1.4.4 Herramientas

Historias de Usuario: Son frases cortas que abarcan lo que el usuario hace o parte de lo que hace para realizar una tarea.

Product Backlog: Es un documento que abarca todo el proyecto, donde se encuentran plasmadas las expectativas que posee el cliente sobre el proyecto, consta de una lista priorizada según las necesidades y funcionalidades más deseadas para ser desarrolladas, en una primera instancia. Esta lista solo puede ser modificada por el Product Owner. Debe cumplir con las siguientes características:

- Debe ser elaborada por el cliente, el mismo debe colocar valores estimados según el deseo de ver completada alguna funcionalidad o segmento del software.
- Al ser el cliente el principal responsable de esta lista de tareas, estas deben ser plasmadas a manera de historias de usuario.
- El Scrum Master, junto con el equipo, deben proporcionar de igual manera una estimación del costo que significa realizar un requerimiento así como las dependencias de las tareas planteadas.

Es posible colocar posibles sprints armados con las tareas del product backlog, donde el cliente incluso pueda evidenciar cuales serán, tentativamente, los entregables que puede recibir después de cada iteración.

Sprint Backlog: Corresponde a una lista de tareas que han sido tomadas del product backlog, para conformar todas las actividades que se realizarán en la presente iteración. Representa un documento inmodificable (a más de que el proyecto depende de un cambio) que sirve como guía al equipo para desarrollar las tareas así como al cliente para exigir un “demo” que demuestre lo trabajado.

Usualmente posee una estructura donde se especifica la tarea, quien la va a desarrollar, el estado en la que se encuentra, las horas pendientes que posee y la fecha de realización de la misma.

Burn Down Chart: Es un gráfico donde es posible visualizar el trabajo pendiente para la finalización del proyecto, así como también la manera en la que el producto está evolucionando respecto a tareas y tiempos de desarrollo. Da la facilidad de reconocer cuando una tarea se está demorando o atrasando para poder tomar las medidas respectivas.

Este cuadro es alimentado tanto por el product backlog como por el Sprint backlog, que conforman la línea base y la línea de desarrollo real, respectivamente.

1.5 Cuadro comparativo entre las metodologías Scrum, XP y Crystal Clear

A continuación se muestra una tabla, donde se resumen las principales diferencias entre tres metodologías ágiles.

Tabla 2: Comparación entre Scrum, XP y Crystal Clear

	SCRUM	XP	Crystal Clear
Área de Principal atención	La entrega del proyecto de acuerdo al gusto del cliente y los tiempos acordados	La satisfacción del desarrollador dentro del equipo y la entrega de un producto que el cliente esté de acuerdo	La entrega del producto final intentando minimizar los riesgos al máximo
Control de Riesgos	Se lo da mediante la constante charla con el cliente y la alta disponibilidad al cambio	Se minimizan los riesgos manteniendo la simplicidad del producto y la baja susceptibilidad a los cambios	Se lo controla gracias a la variedad de metodologías que existen en esta familia así como con la estrategia de victoria temprana
Nivel de conocimiento para usarlo	Es necesaria la existencia de un Scrum Master, quien dará una guía para seguir, mas no un liderazgo. Sin embargo es un modelo bastante	El entrenador es el experto en la metodología que debe involucrarse en el proyecto para liderar el mismo. Es bastante concreto y fácil de seguir sin	Crystal al ser una metodología creada en base a otras ya existentes y que utiliza herramientas de otras metodologías tiene un nivel de dificultad

	simple de seguir	embargo se necesita de alguien con conocimientos para enseñar el uso correcto de la programación en parejas	mayor al momento de utilizarla, se requiere de una persona que lidere con amplia experiencia en metodologías ágiles
Documentación Generada	Scrum puede utilizar una gran variedad de herramientas al momento de documentar, por lo que dependiendo del proyecto y de los objetivos del mismo puede tener: Diagramas de Clase, diagrama de base de datos, diagramas de casos de uso, etc.	XP se centra en la programación, mas no en la documentación. Sin embargo como documentos se pueden tener las historias de usuarios, las tarjetas CRC y algunos esquemas que se pueden generar en la etapa de diseño como herramienta mas no como documentos estrictamente dicho	Puede generar historias de usuarios, tarjetas CRC, diagramas de casos de uso, diagrama de base de datos, etc. Ya que es una metodología tan flexible se obtendrá tanta documentación como se desee
Equipo de Desarrollo	Se recomienda usarlo en equipo desde 3 personas hasta 9.	Se utiliza en grupos medianos desde 16 hasta 30 personas	Depende de la cantidad de personas la metodología que se vaya a utilizar (ver anexo 1)

Para revisar con mayor profundidad sobre Crystal Clear y XP, es posible evidenciar un resumen en: [Anexo 1](#) y [Anexo 2](#).

1.6 Ventajas y Desventajas de la Metodología Scrum

Ventajas

- La constante adaptabilidad al cambio que puede ser introducida en cada sprint, permite versatilidad al proyecto dejando que este se acople a lo que realmente se desea.

- Al tener un representante del Cliente involucrado dentro del proyecto permite acercarse de una manera rápida y sencilla a la expectativa que tiene el cliente sobre su aplicativo.
- Presentar entregables continuamente ayuda a mantener en mente el verdadero objetivo del proyecto así como mantener al cliente satisfecho enseñándole los avances esperados y los resultados propuestos al principio de cada sprint. Además esta metodología se acopla perfectamente al uso de un versionador.
- Al no centrarse directamente en la documentación da tiempo para focalizar esfuerzos en el desarrollo del proyecto.

Desventajas

- Normalmente se requiere de un experto en la metodología cuando se lo desea utilizar en un proyecto grande.
- No es una metodología que se recomienda usar en equipos grandes de trabajo.
- Se establece como premisa que el equipo está completamente capacitado y motivado (No presenta herramientas ni formas de motivar al equipo de trabajo)
- Que el cliente esté presente en el equipo de trabajo todo el tiempo es irreal, ya que muy pocas empresas están dispuestas a prestar un recurso de conocimientos tan amplios en su área negocio durante un largo tiempo.

1.7 Herramientas de Apoyo

1.7.1 Wireframes

Es una guía visual que representa la esencia del contenido de una pantalla, es también conocido como el esquema de una página o el esqueleto de la misma. Su objetivo principal radica en poder representar con rasgos sencillos (en 2 -D) el diseño de una página web, la pantalla de un aplicación, etc.



Figura 2. Ejemplo de WireFrame

Para desarrolladores esta es una forma de vincular el aspecto visual de una pantalla con su funcionalidad explícita, ya que solo se enfocará en lo que la pantalla realmente hace, de igual manera para los diseñadores los wireFrames proporcionan un gran apoyo al momento de definir y empezar a realizar las UI (Interfaces de usuario). En su elaboración se puede utilizar distintos instrumentos desde bocetos en papel hasta programas dedicados exclusivamente a este objetivo.

1.7.1.1 Beneficios

Al ser dibujos sin mucha complejidad son elaborados rápidamente, estos pueden ser hechos junto con el cliente para que de una manera gráfica pueda demostrar lo que realmente necesita y como desea que funcione. Cabe recalcar que no es objetivo de los wireFrames el color de las paginas, el tipo de fuente a ser usando, etc. sino su forma y funcionalidad.

Además al momento de definir funcionalidad es mucho más comprensible un gráfico que frases escritas y menos interpretables, por lo que ayuda a causar menos confusión al momento de retomar un tema posteriormente.

1.7.1.2 Tipos

Se pueden clasificar en dos tipos dependiendo de su nivel de detalle:

- **Baja Fidelidad:** Son rápidos de hacer pero poseen pocos detalles, ayudan a que el equipo del proyecto vea la funcionalidad y dirección que van a tomar los componentes de esas pantallas, utilizan etiquetas para definir cada sección o texto de relleno para representar los lugares donde va la información.
- **Alta Fidelidad:** Normalmente son usados para fortalecer la documentación del proyecto ya que sus detalles son de mayor semejanza a las pantallas reales. Estos son desarrollados con una aplicación especializadas y pueden contener desde JavaScript hasta animaciones para fortalecer su enfoque.

1.7.2 Diagrama de Base de datos (Entidad - Relación)

Es un modelo conceptual creado en 1976 por Peter Chen para simplificar el diseño de las bases de datos, consiste en una percepción del mundo real que va a ser representada usando descripciones textuales de requerimientos y objetos denominados entidades, atributos y relaciones.

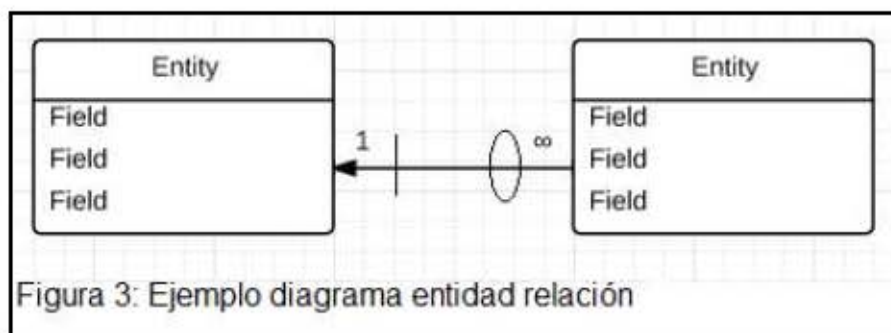


Figura 3: Ejemplo diagrama entidad relación

1.7.2.1 Entidad

Corresponde a un objeto concreto o abstracto, distinguible de otros que existe dentro de un contexto y del cual deseamos guardar información

Ejemplo:

Los profesores son una entidad dentro del contexto escuela que se pueden diferenciar claramente de otras entidades como estudiantes, personal administrativo, etc.

1.7.2.2 Atributos

Cuando, en entidades, nos referimos a que deseamos guardar información, esta concierne a los atributos. Estos son propiedades o características innatas que pueden tomar un valor dependiendo de las instancias de cada entidad.

Ejemplo:

Para la entidad Persona existen los atributos Nombres, Apellidos, Documento de Identidad, etc.

Clave Primaria

Para poder identificar de forma única a cada instancia de una entidad, es necesario remarcar una o más atributos que posean esta característica, ya que estas serán utilizadas para referenciar cada instancia por separado.

Clave Foránea

Cuando existe una relación entre dos entidades, dependiendo del tipo de relación, la clave primaria de una de ellas es colocada en la segunda bajo el nombre clave externa o foránea.

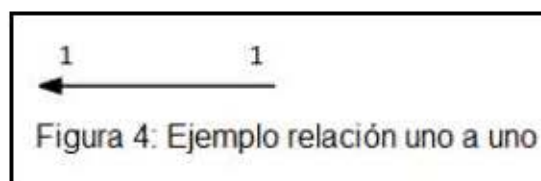
1.7.2.3 Relación

Son las formas en las que interactúan dos o más entidades, se lo representa con una flecha, se dividen en los siguientes tipos:

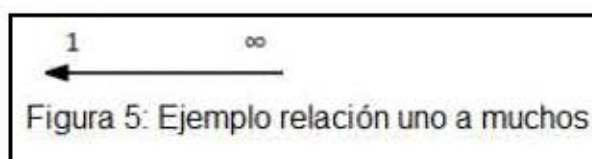
Por Cardinalidad

Para la siguiente clasificación se debe tomar en cuenta que el análisis debe hacerse en ambos sentidos de la relación.

Relación uno a uno: Para que exista este tipo de relación solo una instancia de una entidad puede corresponder a una sola instancia de la otra entidad, para este tipo la clave foránea debe estar presente en una de las dos entidades. Su forma de representación es la siguiente:



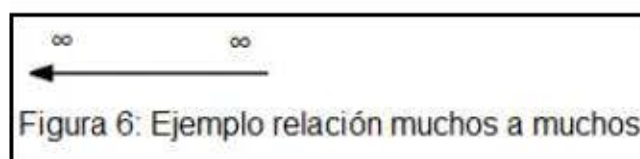
Relación uno a muchos: Se presenta cuando un registro de la tabla principal puede tener varios registros de la tabla secundaria, pero una sola instancia de la tabla secundaria puede tener uno y solo un registro de la tabla principal. Su forma de representación es la siguiente:



Relación muchos a muchos: Siempre que dos tablas puedan relacionarse de tal manera que cualquiera de ellas pueda referirse con múltiples registros de la otra, y viceversa, cumple con esta relación.

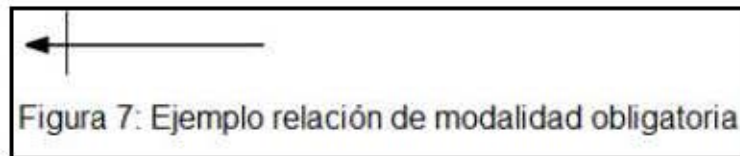
La existencia de este tipo de relación no puede darse solo con las dos tablas principales, es necesario crear una tabla de vinculación que posea como clave primaria las dos claves foráneas de las tablas principales.

Su forma de representación es la siguiente:

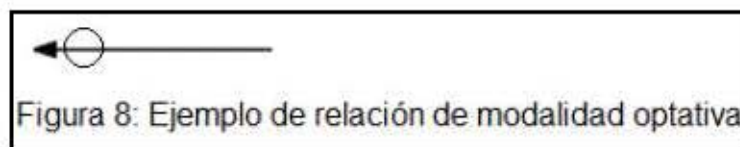


Por Modalidad

Relación de modalidad Obligatoria: Es cuando todos los registros de la tabla principal deben poseer al menos una instancia en la tabla secundaria. Su forma de representación es la siguiente:

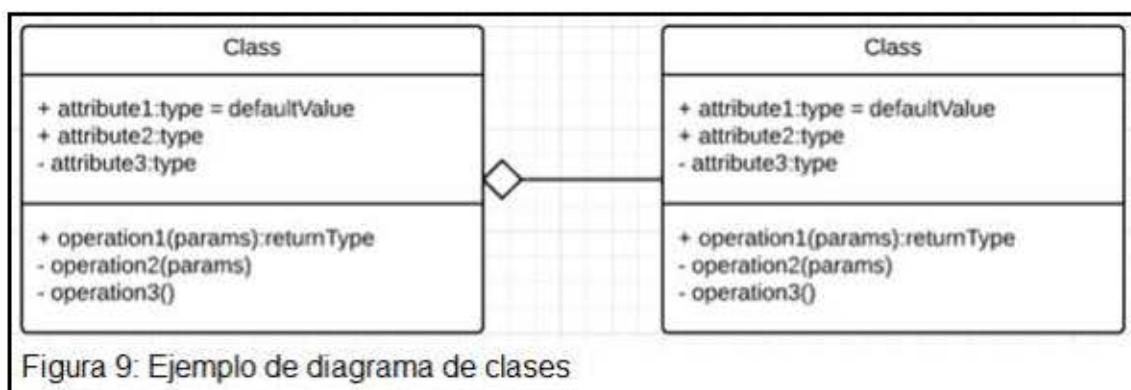


Relación de modalidad Optativa: Para que se presente este tipo de relación es necesario que para todos los registros de la tabla principal haya o no instancias relacionadas de la tabla secundaria. Su forma de representación es la siguiente:



1.7.3 Diagrama de Clases

Es el tipo de diagrama UML más utilizado, su función radica en representar la orientación a objetos que posee un sistema. Sus elementos principales son las relaciones y las clases.



1.7.3.1 Clases

Como dice la teoría de orientación a objetos, una clase es una estructura que define las características y el comportamiento de un conjunto de objetos que comparten el mismo patrón. En el diagrama se lo representa como un rectángulo con 3 secciones, donde el primero representa el nombre que posee o va a poseer una clase, las siguientes se explicarán a continuación.

Atributos

Los atributos son las características de cada clase que definen los valores que una instancia de la misma va a poseer. Estos pueden tener distinta visibilidad, las cuales son:

- Public: Puede utilizarse tanto dentro como fuera de una clase
- Protected: Solo bajo herencia o si se encuentran en un mismo paquete, se puede acceder a estos atributos
- Private: El atributo solo puede ser usado dentro de su misma clase

Métodos

Es el comportamiento que posee una clase, es decir su funcionalidad al momento de interactuar con otras clases. Estos también pueden tener distintos tipos de visibilidad como public, protected y private.

1.7.3.2 Relaciones

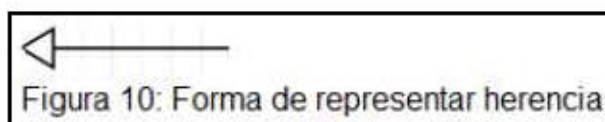
Es la forma en que se afectan las clases entre ellas en el sistema. Para que estas existan es necesario especificar la cardinalidad que poseen, estos pueden ser:

- De uno a muchos (1...n)
- De cero a muchos (0...n)
- O con números fijos (m, donde m es cualquier número)

Pueden existir de distintos tipos:

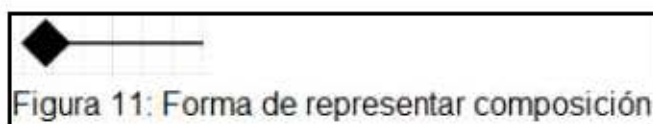
Herencia

Se presenta cuando una clase denominada hija posee los atributos y métodos visibles de una clase padre, además de los propios. Se lo representa usando una flecha sin relleno que va en dirección de la clase padre.



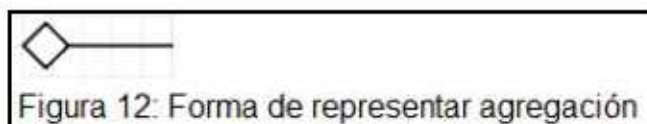
Composición

Es también llamada relación por valor, y como su nombre indica aparece cuando un objeto que se encuentra dentro de una clase solo puede existir mientras el tiempo de vida de la instancia de la clase que lo contiene perdure. Se lo representa con un rombo con relleno que va hacia donde se encuentra la clase principal.



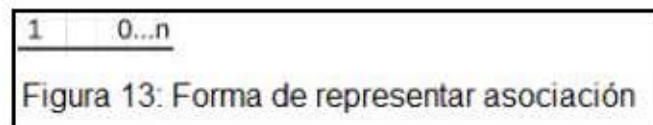
Agregación

Se la conoce como relación por referencia, ya que la existencia del objeto relacionado a la clase es independiente del tiempo de vida de la instancia de la misma. Se lo grafica con un rombo sin relleno que va en dirección de la clase principal.



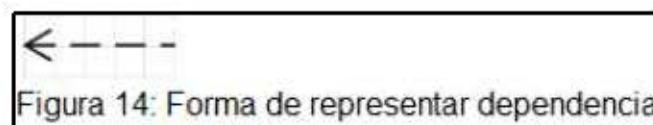
Asociación

Existe para demostrar como dos clases pueden trabajar en conjunto para un objetivo, los tiempos de vida de las instancias son independientes en cada clase a pesar de que estas posean este tipo de relación. Se lo representa como una línea donde en los extremos se debe especificar la cardinalidad que poseen.



Dependencia

Cuando una clase para instanciarse depende de la existencia de una instancia de otra clase existe una relación de dependencia entre la primera de la segunda. La forma de representarse es con una flecha en dirección de la clase dependiente y líneas entrecortadas.



1.7.4 Diagrama de Casos de Uso

Es la forma que plantea UML para documentar el comportamiento del sistema desde la óptica del usuario, siempre tomando en cuenta que para elaborarlos es necesario poseer los requerimientos funcionales de un sistema.

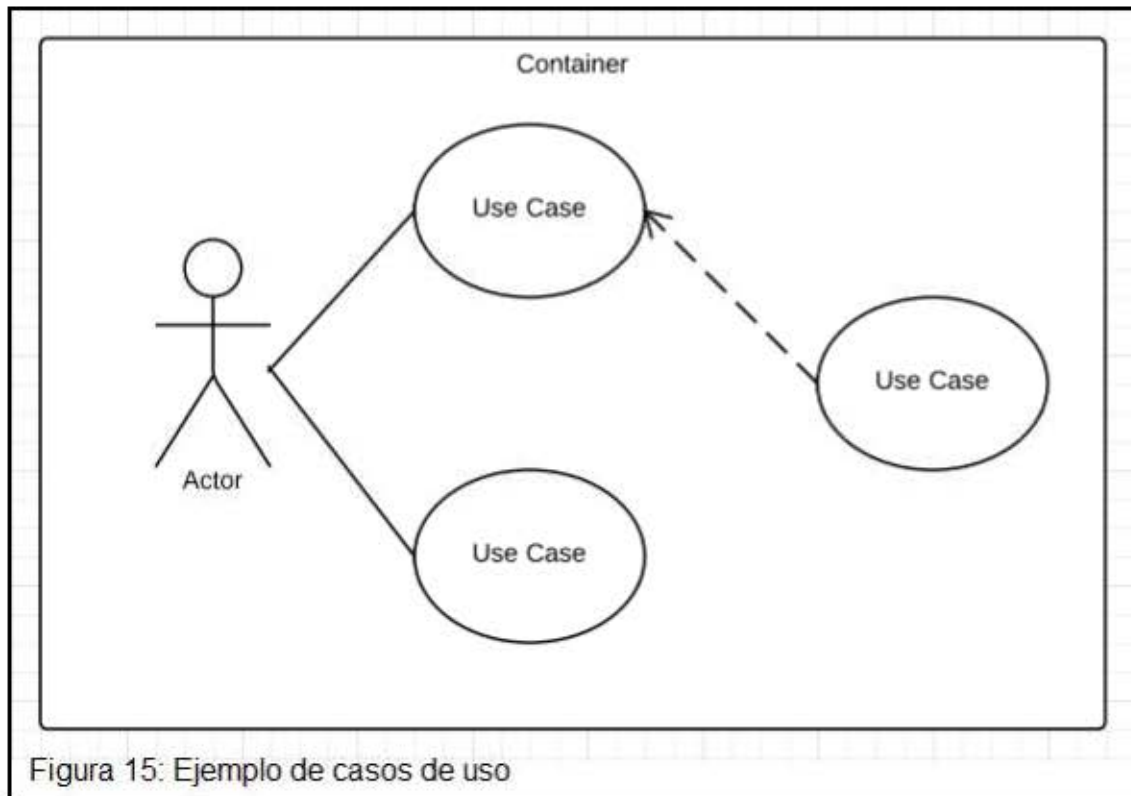


Figura 15: Ejemplo de casos de uso

Para representarlo se utilizan los siguientes elementos.

1.7.4.1 Actor

Es cualquier agente externo al sistema que, por medio de un rol, va a interactuar con el mismo, estos pueden ser personas, otros sistemas, etc. Hay que tomar en cuenta que para el diagrama el verdadero actor es siempre el rol. Para representarlo se utiliza un dibujo simple de una persona.



Figura 16: Representación de un actor

1.7.4.2 Caso de Uso

Es una acción puntual que puede realizar el sistema bajo alguna orden actor o con la activación de otro caso de uso. Se lo representa con una elipse que contiene la acción.

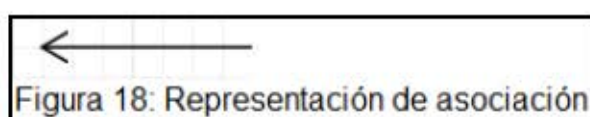


1.7.4.3 Relaciones

Es la forma en la que trabajan juntos un actor y un caso de uso o varios casos de uso entre sí. Existen de los siguientes tipos:

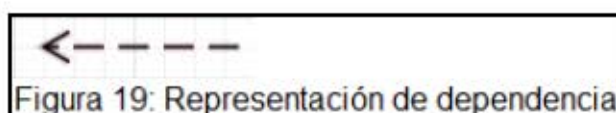
Asociación

Esta relación se la utiliza para indicar la activación de un caso de uso por parte de un actor o de otro caso de uso. Su símbolo es una flecha.



Dependencia

Demuestra que para que se ejecute un caso de uso, necesariamente se debe ejecutar primero el que lo invocó, este tipo de relación solo puede ser utilizada entre casos de uso. Se lo representa con una flecha con una línea entrecortada.

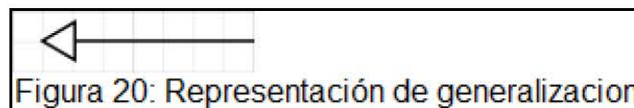


Se subdivide en dos tipos:

1. Incluye (<<include>>): Se lo utiliza para detallar si un caso de uso es común entre dos o más casos de uso. Su flecha tiene una dirección desde el caso de uso particular hacia el común.
2. Herencia (<<extends>>): Cuando un caso de uso tiene un comportamiento similar a otro, pero que dependiendo de ciertas circunstancias puede cambiar su comportamiento base se lo conoce como herencia. Su flecha tiene una dirección desde el caso de uso modificado hacia el original.

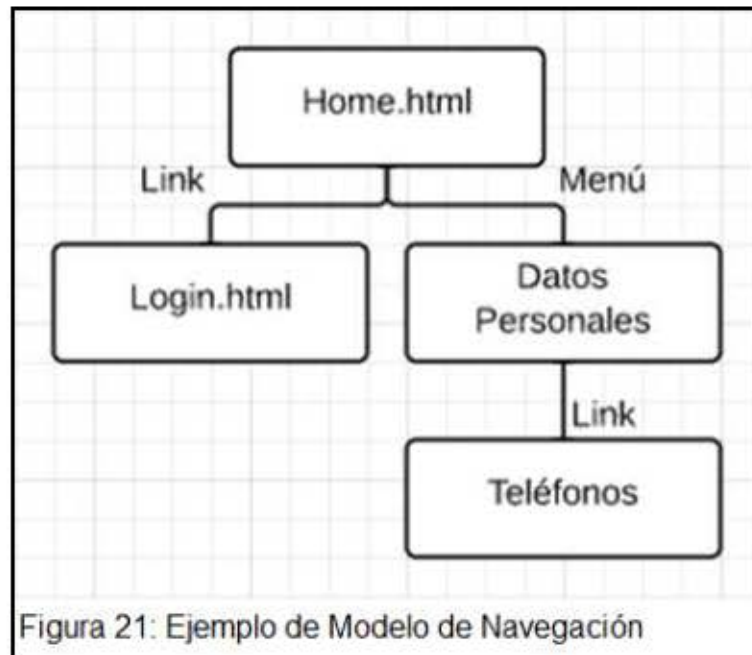
Generalización

Es otro tipo de herencia, sin embargo este es un tipo de especialización que indica que, en cuanto a reglas del negocio, tanto padre como hijo poseen las mismas características, por lo que en un momento dado del sistema es posible decir que se cambia el lugar entre los dos y el sistema no se ve alterado. Su forma de representación es una flecha sin relleno que va desde el caso de uso especializado hacia el original.



1.7.5 Modelos de Navegación

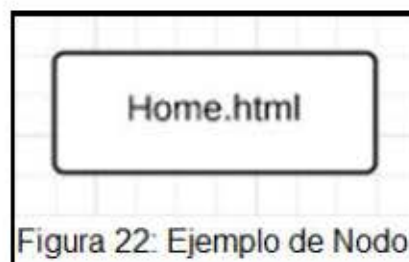
Tipo de diagrama comúnmente usado en desarrollos orientados a web, donde se puede identificar la forma en que las páginas se encuentran entrelazadas entre sí, en un momento dado o bajo condiciones definidas.



Para la elaboración de esta herramienta se utilizan nodos y relaciones.

1.7.5.1 Nodos

Es la unidad de navegación, utilizada para representar una o varias páginas en un espacio de tiempo. Se lo representa mediante un rectángulo con el nombre de la página dentro del mismo.

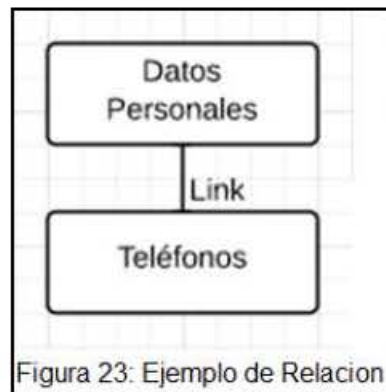


1.7.5.2 Relaciones

Para poder identificar las formas en las que interactúan los nodos, se utilizan relaciones, las cuales pueden ser de diferentes tipos, aquellas que fueron utilizadas en este trabajo de titulación se encuentran definidas a continuación:

- **Enlaces:** Forma de enlace que representa el vínculo por cual se debe pasar para acceder desde un nodo a otro.
- **Menús:** Indica un contenedor organizado de enlaces, en cual es posible visualizar una lista de nodos.
- **Tabs:** Corresponde a un conjunto de enlaces dentro de una barra de navegación horizontal.

La representación de las relaciones se lo hace mediante una línea que junta dos o más nodos, e indica a que tipo corresponde.



2. Tecnologías

El objetivo de este capítulo es explicar de manera sencilla y general las tecnologías que van a ser utilizadas en el desarrollo del prototipo planteado. Su intención no es de profundizar completamente en estos temas ni de explicarlos en su totalidad.

2.1 HTML5

HTML o lenguaje de marcado de hipertexto fue diseñado por Tim Berners-Lee, con el objetivo de reducir la complejidad de desarrollar la documentación necesaria para un documento web. Al principio ideado para describir semánticamente documentos científicos, terminó convirtiéndose en un estándar para el desarrollo de todo tipo de páginas web.

Conforme al pasar de los años, los nuevos dispositivos, las necesidades y nuevos requerimientos, el lenguaje se ha visto obligado a ir evolucionando para soportar estos constantes cambios creando así la quinta versión de HTML, donde se plantea embeber el uso y documentación de las especificaciones:

DOM para HTML: Modelo de objeto de documento, es una interfaz de plataforma, que tiene como propósito el permitir a programas y scripts a acceder y manipular el contenido de una página web, sin necesidad de modificar el fuente propio del recurso.

XHTML: Es una reformulación de HTML como un XML, que da tanto las facilidades de usar lenguaje HTML como de serializarlo y proveerlo de flexibilidad usando el estándar XML. De esta manera es posible re-utilizar las herramientas ya existentes para XML así como escoger entre usar DOM para HTML o DOM para XML.

A pesar de que esta versión todavía se encuentra en un nivel de un borrador liberado, una gran cantidad de desarrolladores web se encuentran capacitándose e incluso programando ya en HTML5, un ejemplo de esto se puede visualizar en la página de videos YouTube, donde ya es utilizada esta nueva versión del estándar. Todo esto se debe a las mejoras y nuevas

funcionalidades que se presentan para su uso al público además de su completa compatibilidad hacia sus propias versiones anteriores. De todas estas funcionalidades, los elementos que serán utilizados en este Trabajo de Titulación, se explicarán a continuación:

2.1.1 Independencia y Rendimiento

Una de las principales ventajas que ofrece esta nueva tecnología es la independencia de plug-ins o aplicaciones que antes eran necesarias tener instaladas en el navegador de internet para poder visualizar el contenido, como por ejemplo las etiquetas de video y audio (ver punto [Anexo 3: 3.1](#)) evitan la obligación de tener instalado los productos QuickTime o Adobe Flash para poder reproducir esta clase de elementos.

De igual manera otro objetivo de este lanzamiento es el empezar a rivalizar a nivel de rendimiento con las aplicaciones de escritos, donde gracias a elementos ahora embebidos dentro de la funcionalidad de HTML, como la funcionalidad para gráficos (ver punto [3.1.3](#)), los API's para trabajar en fuera de línea (ver punto [Anexo 3: 3.2](#)), etc., pueden significar una mejora a nivel de velocidad de respuesta al momento de presentar o procesar contenido, antes considerado "pesado", en el navegador.

2.1.2 Almacenamiento

El almacenamiento con *Cookies* fue la primera impresión que se pudo proporcionar a las aplicaciones web para poder almacenar pares de datos (clave, valor). Sin embargo al tener la necesidad de querer almacenar algo más allá de esto o de un tamaño superior, resultaba imposible hasta que HTML5 logró proporcionar la manera de que el navegador, nativamente sea capaz de almacenar la información.

Esta capacidad de se la puede conocer con distintos nombres como:

- Especificación *Web Storage*
- *Local Storage*
- *DOM Storage*

Sin embargo su funcionalidad es la misma y corresponde a guardar pares de datos de tipo cadena de caracteres que pueden ser almacenados de hasta 5MB cada uno, para luego por medio de JavaScript convertir los datos al formato que necesitemos a para operar.

Existe una manera adicional para poder tanto leer como escribir archivos en el PC del cliente, y es por medio del uso del *File System API* que puede incluso llegar a permitir al desarrollador web crear aplicaciones de edición musical o grafica para que el usuario pueda guardar directamente sus archivos en su computador en vez de descargar el producto final desde el servidor donde se encuentra la aplicación.

2.1.3 Gráficos

La etiqueta <canvas> es añadida en esta versión para poder realizar cualquier gráfico que se desee desplegar en una página web, como su traducción en español lo indica el “lienzo” es un espacio reservado dentro de la página que permitirá tanto importar imágenes como crearlas.

Por el momento solo existe, nativamente, el contexto para gráficos en 2D, sin embargo es posible utilizar API's de terceros para utilizar un contexto de imágenes en 3D, y en un futuro se espera acoplar esta funcionalidad dentro de la especificación de HTML5.

Para proyectar una imagen dentro de <canvas> se debe tomar en cuenta que está formado por los ejes de coordenadas x-y, donde el origen es el extremo superior izquierdo del lienzo. Esto permite realizar cualquier tipo de imagen e incluso animación que se desee realizar, ya que por medio del uso de JavaScript se puede capturar los diferentes eventos de la etiqueta y personalizar las acciones según como el desarrollador web lo desee.

Para mayor información sobre funcionalidad nueva de HTML5, revisar el [Anexo 3](#). Es posible visualizar un ejemplo de uso en el [Anexo 4](#).

2.2 Spring 3 Framework

Es un conjunto de funcionalidades que ayudan a manejar la infraestructura de una aplicación, facilitando al desarrollador el enfocarse en el comportamiento del sistema a desarrollarse. Casi todas las características centrales pueden ser aplicadas en cualquier tipo de aplicación java, sin embargo existen algunas que solo podrán ser utilizadas en modelos construidos bajo el estándar de aplicaciones empresariales de java (JEE).

Entre los distintos módulos que posee los siguientes serán utilizados en el presente Trabajo de Titulación:

2.2.1 Contenedor de Control de Inversión e Inyección de Dependencias

Es el comportamiento central que posee *Spring*, que por medio del uso de programación por reflexión le es posible configurar y administrar los objetos creados por la aplicación en el momento de la ejecución (*runtime*). El contenedor IoC (*Inversion of control container*) es el responsable del manejo del ciclo de vida de los objetos (*beans*) creados y configurados dentro del *framework*.

Estos *Beans* normalmente dependen de la instanciación de otros por lo que por medio de una búsqueda de los constructores, propiedades o métodos *Factory* son construidos automáticamente y colocados en los objetos que dependen de ellos.

En el prototipo es posible evidenciar esta configuración de uso en el archivo "application-context.xml" donde se encuentran configuradas las dependencias a ser usadas en la capa de los controladores de vistas.

2.2.2 Acceso a datos

Para la interacción de la aplicación con la base de datos, *Spring* presenta su módulo de acceso a datos que puede comunicarse con distintos tipos de *API's* tales como *JPA*, etc.

Su principal funcionalidades son:

Administración de recursos: Donde se encargan de abrir y cerrar las conexiones necesarias a la base de datos.

Manejo de excepciones: Traduce cualquier tipo de error que puede presentar la base de datos a excepciones dentro de la jerarquía.

Participación en las transacciones: Mediante el uso del módulo de transacciones es posible que el *framework* ayude en la administración de las mismas de una manera transparente.

Desenvoltura de recursos: Obtiene los objetos de la base de datos desde un *pool* de conexiones.

Abstracción: Para el manejo de tipos de datos como BLOB y CLOB.

2.2.3 Administración de Transacciones

Este componente se encuentra integrado dentro del módulo de acceso a datos, sin embargo puede ser ocupado de manera independiente. Es el encargado de brindar la abstracción necesaria para el manejo de transacciones tanto de manera global como local, que mediante la creación de clases a manera de *proxys* permite administrar tanto succiones como componentes enteros de la aplicación.

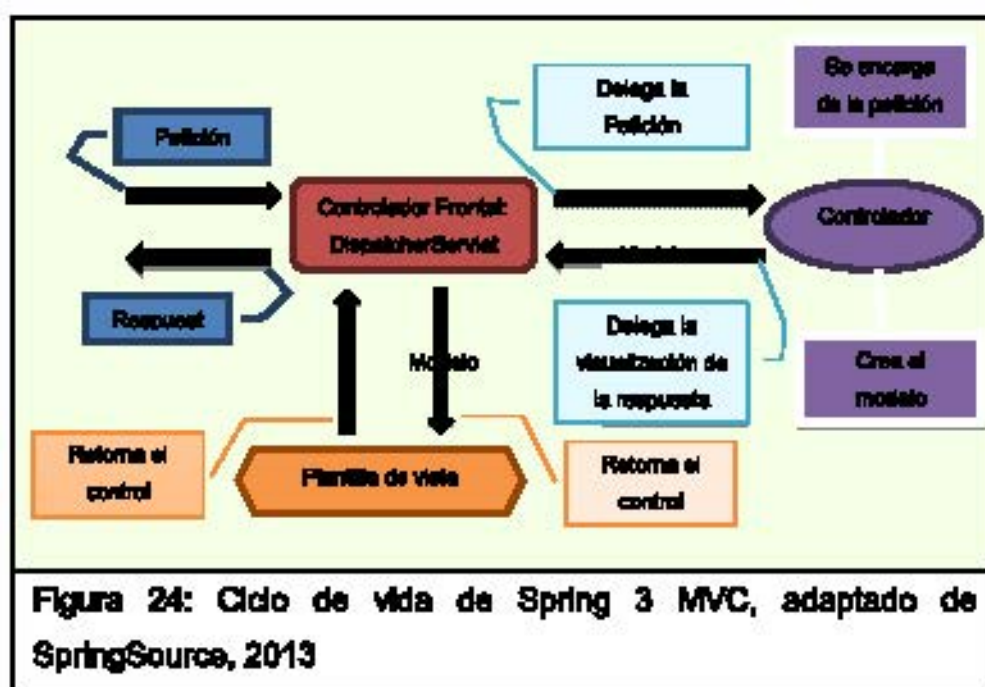
2.2.4 Spring 3 MVC

Tiene la propiedad de proveer una alta funcionalidad para desarrollar aplicaciones web. Todos los componentes de *Spring* han sido desarrollados para poder ser configurables en casi cualquier aspecto así como permitir el acoplamiento de esta tecnología con muchas otras disponibles en el mercado, que juntándoles con los otros componentes del *framework* pueden llegar a ser aplicaciones empresariales muy sólidas.

Como casi todos los *frameworks* orientados a arquitectura MVC (Modelo-Vista-Controlador) *Spring* es regido por peticiones que son re direccionadas hacia un

serviet para enviarlas a sus respectivos controladores. En el caso de este componente se lo conoce como `DispatcherServlet`.

En el siguiente gráfico se puede evidenciar el funcionamiento y ciclo de vida, del componente *Spring MVC*.



Para el desarrollo del prototipo, se utilizará este componente para el manejo de las peticiones y respuestas desde la vista hasta la capa de negocio o *back-end*. El patrón de diseño principal de la aplicación será MVC. Toda la arquitectura de las clases utilizarán las anotaciones, API's e implementaciones disponibles en *Spring Framework* para su funcionamiento.

Para revisar un ejemplo simple del uso del *framework* dirigirse al [Anexo 5](#).

2.3 PrimeFaces

Para entender de mejor manera que es PrimeFaces, es necesario conocer primero que significa el estándar JSF 2.

JavaServer Faces es una especificación java, presentada a manera de *framework* que tiene por finalidad el facilitar al desarrollador la integración de una aplicación a la parte de las interfaces de usuario de la misma. Este es un

framework basado en solicitudes que se encuentra impulsado por un modelo de interfaz de usuario, que mediante el procesamiento de peticiones a través de su *FacesServlet* carga archivos XML denominados plantillas de vista o *Facelets*, maneja eventos, genera un árbol de componentes y redirige las respuestas HTML al cliente.

Este comportamiento es posible evidenciarlo en el siguiente gráfico donde se visualiza la interacción de JSF con páginas HTML, los controladores de las mismas y finalmente con la siguiente capa del software donde es utilizado

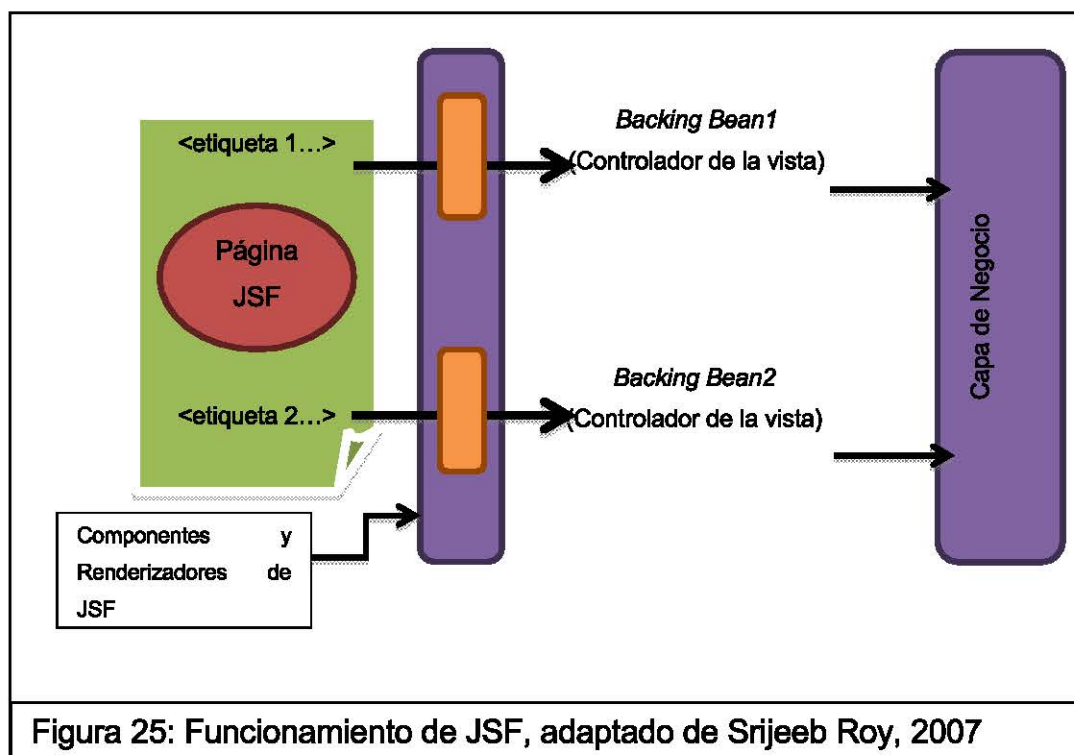


Figura 25: Funcionamiento de JSF, adaptado de Srijeeb Roy, 2007

Este conjunto de API's puede ser reforzado mediante el uso de implementaciones especializadas como es el caso de *PrimeFaces* que posee más de 100 componentes para desarrollar interfaces de usuarios además de contar con una implementación embebida de Ajax y más de 25 temas diseñados para personalización de aplicaciones.

Todas las interfaces de usuario desarrolladas para el prototipo de este Trabajo de Titulación, van a utilizar el *framework* de *PrimeFaces* que integrado con los componentes HTML5 (ver punto [3.1](#)) forman el *front-end* de la aplicación.

Además en la capa del sistema que posee los controladores de las páginas se utilizará el *framework* de JSF para la declaración y manejo de dichos *beans*.

Una guía de cómo implementar y utilizar el *framework* se encuentra en el [Anexo 6](#) del presente Trabajo.

2.4 Hibernate

Es una implementación de la especificación JPA, que se enfoca a resolver los problemas de persistencia de una aplicación por medio del mapeo objeto-relacional. Para la comunicación entre los dos sistemas (aplicación y base de datos) *Hibernate* cumple la función de traductor, ya que transforma automáticamente los tipos de datos entre java y SQL.

Para los diferentes tipos de relaciones que pueden presentar los esquemas de base de datos, este *framework* provee las siguientes estrategias para solventarlas:

Lazy Initialization: Se realiza la consulta de los datos relacionados solo cuando el sistema lo requiere.

Eager inicialización: Al momento de consultar el objeto principal, también alimenta los objetos que están relacionados a este.

Listas: En las relaciones muchos a muchos o uno a muchos, los objetos son representados por medio de listas para indicar la dependencia interna que tiene estas clases mapeadas.

En el presente trabajo de Titulación se utilizará esta librería para la conexión con una base de datos desarrollada en PostgreSQL.

2.5 Servidor de Aplicaciones JBoss

Jboss AS (Application Server) es uno de los productos pertenecientes a *RedHat* dentro de su rama de productos empresariales de *Middleware*. Este producto entra en la categoría de *open source*, ya que su código, descarga y uso se los presenta mediante el uso de la licencia de *GNU Lesser General Public license*

El uso de este servidor es especializado para el desarrollo y despliegue de aplicaciones java que siguen el estándar J2EE, que son servicios o portales. Puede ser ejecutado en cualquier plataforma sin importar el sistema operativo que esta posea.

En su nueva versión dispone de dos modos de operación:

- **StandAlone:** Se lo conoce como un servidor de un solo nodo donde normalmente se colocan las aplicaciones en fase de desarrollo mientras son probadas y corregidas
- **Domain:** Utilizada comúnmente en ambientes de producción presenta un modelo de alta disponibilidad donde es posible poseer varios nodos, y también manejar el concepto de balanceo de carga.

Todos los componentes del servidor se presentan en forma de módulos de servicios, donde cualquier servicio puede tanto iniciarse como detenerse.

Este servidor de aplicaciones será usado para la elaboración del prototipo permitiendo el despliegue y visualización de la aplicación web dentro de un navegador de internet.

2.6 JavaScript

Es un lenguaje de programación, de alto nivel, orientado a realizar actividades repetitivas dentro de páginas web, este tipo de tareas se las conoce como *scripting*.

Normalmente se lo utiliza para realizar actividades o colocar contenido dinámico a las páginas web que se presentan del lado de cliente de la aplicación, ya que permite cambiar el contenido de los párrafos, cambiar

imágenes, abrir ventanas nuevas, hacer que objetos de la pantalla aparezcan o desaparezcan, etc. Estos *scripts* se ejecutan junto con la navegación de la página, si el usuario decide cerrar el navegador, entonces todos los *scripts*, detendrán su accionar.

En el contexto del prototipo a desarrollarse, se utilizará este lenguaje como un apoyo para realizar tareas definidas del lado del front-end. Se lo empleará cuando sea necesario junto con las otras dos tecnologías (ver puntos [3.1](#) y [3.3](#)) explicadas anteriormente.

3. Diseño y Desarrollo

3.1 Scrum

El propósito de este segmento es para colocar las herramientas y datos utilizados durante el desarrollo de este Trabajo de Titulación y del prototipo siguiendo la metodología ágil Scrum

3.1.1 Datos Generales

Tabla 3: Datos las iteraciones del proyecto

Datos de Sprints	
Número de Sprints	41
Número real de Sprints	36
Duración de cada Sprint	7 días
Horas de cada Sprint	35 Horas

Tabla 4: Roles

Asignación de Roles	
Rol	Encargado
Patrocinadores	UDLA
Administradores	Esteban Capello
Product Owner	Esteban Capello
Scrum Master	Juan José León
Desarrollador	Esteban Capello

3.1.2 Product Backlog

A continuación se coloca un fragmento del product backlog, para visualizarlo completo se puede revisar el Anexo 10.

Tabla 5: Muestra de 15 tareas dentro del product backlog

ID Historia	Nombre de la Historia	Fecha de Culminación	Tamaño (días)	Sprint	Comentarios
1	Planteamiento de ideas para el capítulo 1	6/8/2012	1	1	
2	Escritura de antecedentes, justificación y objetivos	7/8/2012	0,5	1	
3	Planeación de alcance y áreas de aplicación	7/8/2012	0,5	1	
4	Escritura de alcance y áreas de aplicación	8/8/2012	1	1	
5	Escritura de la introducción	9/8/2012	1	1	
6	Revisión del capítulo 1	10/8/2012	1	1	
7	Correcciones al capítulo 1	12/8/2012	1,5	1	
8	Planteamiento de ideas para el capítulo 2	13/8/2012	1	2	
9	Investigación de metodologías de desarrollo de software tradicionales	13/8/2012	0,5	2	
10	Escritura de lo investigado sobre metodologías de desarrollo tradicionales	14/8/2012	0,5	2	
11	Investigación de metodologías de desarrollo de software ágiles	15/8/2012	1	2	
12	Escritura de lo investigado sobre metodologías de desarrollo ágiles	15/8/2012	0,5	2	
13	Desarrollo de un cuadro comparativo sobre los tipos de metodologías	16/8/2012	1	2	
14	Investigación sobre Scrum	18/8/2012	1,5	2	
15	Escritura de metodología Scrum (Partes y análisis de ventajas y desventajas)	20/8/2012	2	3	

3.1.3 Iteraciones

Dentro de este Trabajo de Titulación se realizaron un total de 36 sprints, sin embargo en esta sección se colocó solo el segundo. Las restantes 35 iteraciones se las pueden encontrar en el Anexo 11

Tabla 6: Segunda iteración

Sprint 2

Fecha Inicio	12/ago/2012
Fecha Fin	18/ago/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de un cuadro comparativo sobre los tipos de metodologías	Esteban Capello	Hecho	5	2	16/8/2012	13/8/2012
Investigación sobre Scrum	Esteban Capello	Hecho	7.5	5	18/8/2012	14/8/2012
Escritura de metodología Scrum (Partes y análisis de ventajas y desventajas)	Esteban Capello	Hecho	10	5	20/8/2012	15/8/2012
Investigación de metodologías XP y Cristal Clear	Esteban Capello	Hecho	15	12	23/8/2012	18/8/2012
Desarrollo de un cuadro comparativo sobre metodologías ágiles	Esteban Capello	Hecho	5	3	24/8/2012	18/8/2012

3.1.4 Scrum Daily Meeting

Durante el desarrollo del proyecto no era posible el reunirse diariamente con el Scrum Master (Profesor Guía). Sin embargo a través de un archivo compartido con la siguiente tabla era posible estar en contacto con él todos los días.

Tabla 7: Resumen de Reuniones del segundo Sprint

Sprint 2

Fecha Inicio	12/ago/2012
Fecha Fin	18/ago/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
13	13/8/2012	Investigar y escribir sobre metodologías ágiles	Cuadro comparativo de tipos de metodologías	No	Listo	OK!
14	14/8/2012	Cuadro comparativo de tipos de metodologías	Investigar Scrum	No	Listo	Chévere
15	15/8/2012	Investigar Scrum	Escribir la sección de scrum	No	Listo	OK!
16	16/8/2012	Escribir la sección de scrum	Investigar XP y Cristal Clear	No	Listo	OK!
16	17/8/2012	Investigar XP y Cristal Clear	Escribir los anexos de XP y Cristal Clear	No	Listo	OK!
16 - 17	18/8/2012	Escribir los anexos de XP y Cristal Clear	Cuadro comparativo de las metodologías ágiles	No	Listo	Perfecto

Para revisar todas las reuniones realizadas revisar el Anexo 12

3.1.5 Sprint Review y Sprint Retrospective

Ambas tareas de Scrum fueron llevadas a cabo de manera presencial con el Scrum Master donde de manera oral se fueron demostrando las metas alcanzadas así como también planteando las mejoras y cambios necesarios para los siguientes Sprints.

Ya que objetivo planteado en el presente Trabajo de Titulación es la implementación de un prototipo, se decidió no plantear varios entregables o "demos" sino entregar un prototipo final.

3.1.6 Burn Down Chart

En la siguiente figura es posible evidenciar tanto la planificación del proyecto así como el camino por el cual avanzó hasta completarse.

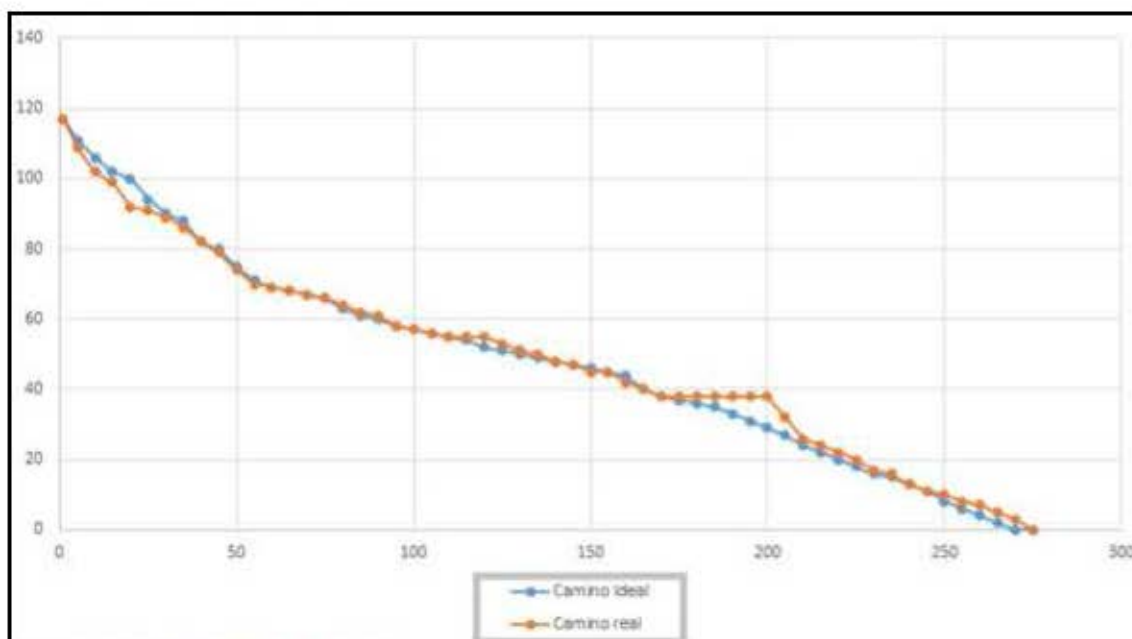


Figura 26: Burn Down Chart

3.2 Modelos de Navegación

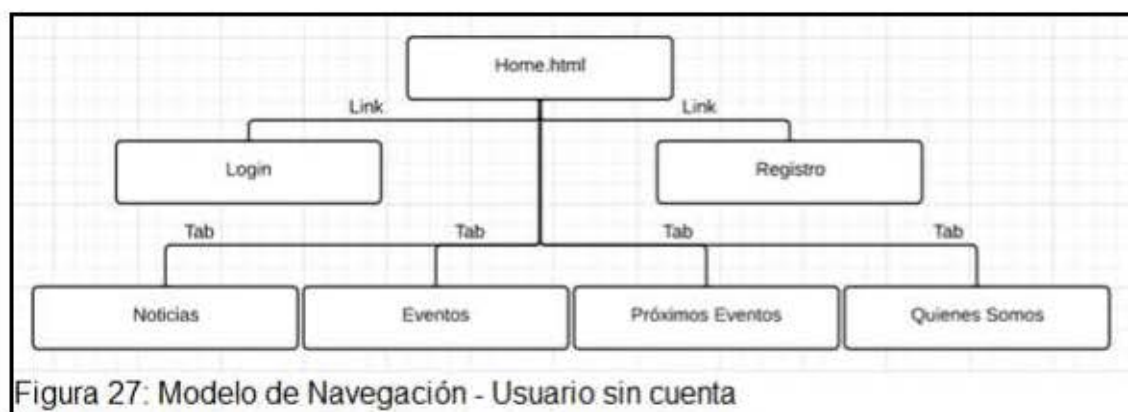
Para proporcionar un mejor entendimiento se han planteado los siguientes modelos de navegación a partir de los diferentes tipos de roles dentro del prototipo.

3.2.1 Roles

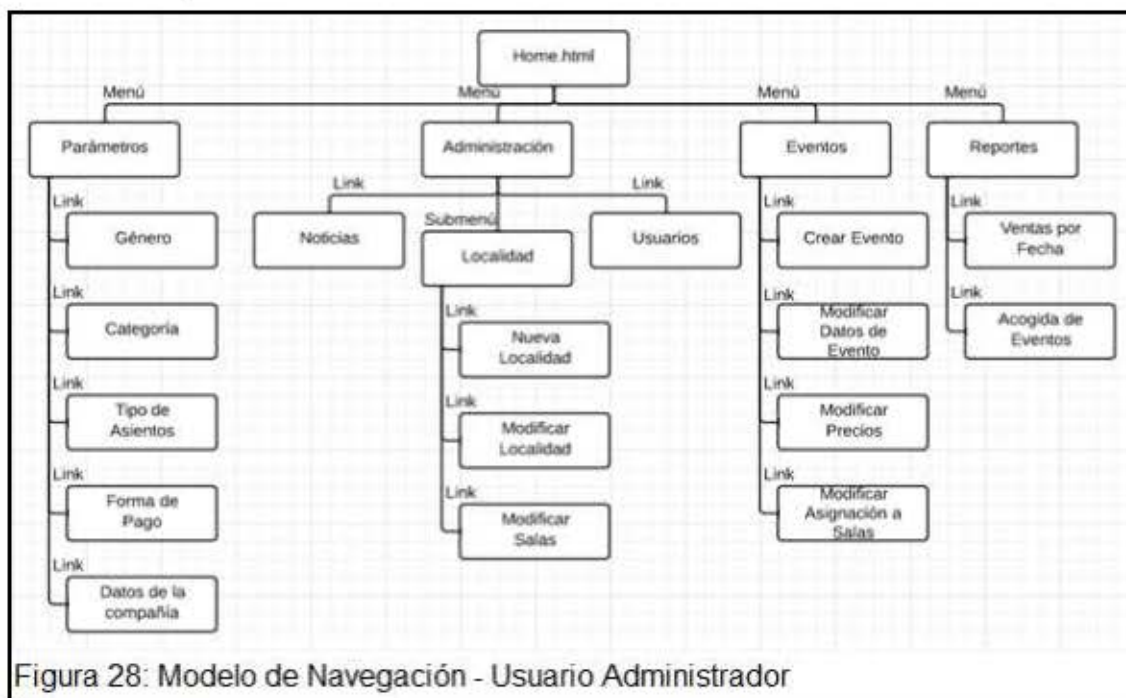
Tabla 8: Roles identificados dentro del prototipo

Roles	Descripción
Usuario sin Registrar	Usuario que podrá tener una navegación limitada de sistema, con la opción a registrarse en el mismo
Cliente	Usuario registrado que aparte de la navegación del usuario sin registro puede también reservar boletos a eventos
Vendedor	Usuario registrado que aparte de la navegación del usuario sin registro puede también vender boletos y pagar reservaciones realizadas
Administrador	Usuario registrado que posee la habilidad de configurar el mismo para su funcionamiento

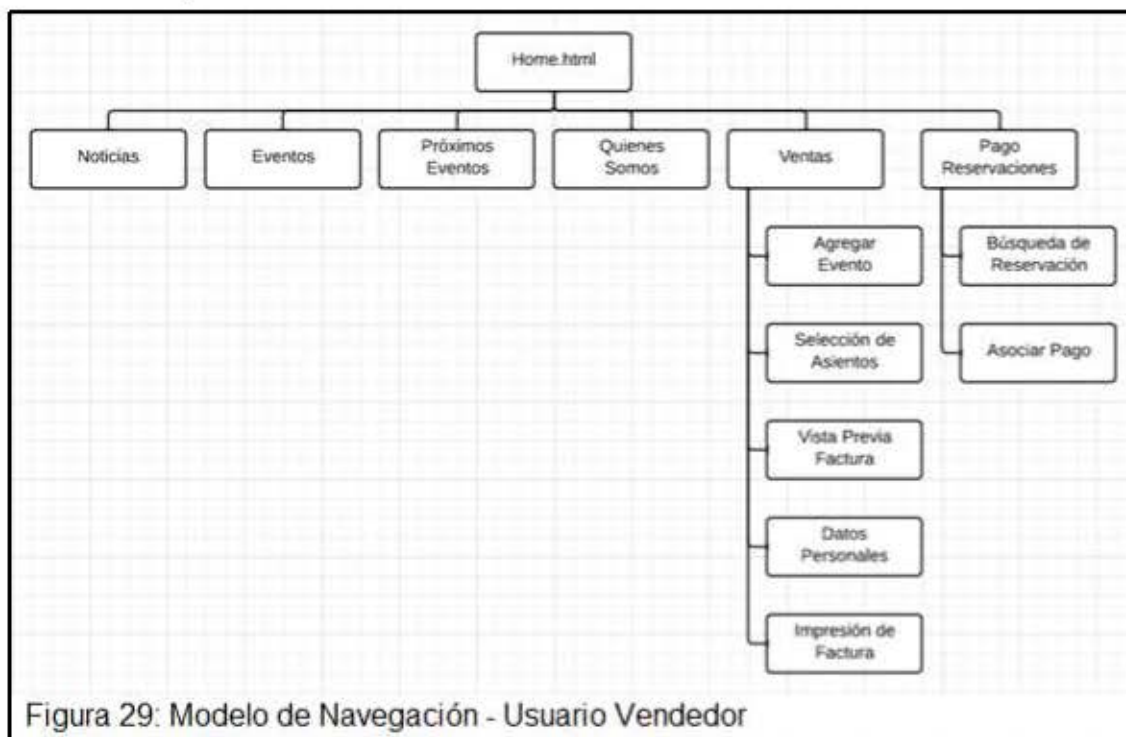
3.2.2 Navegación – Usuario sin cuenta



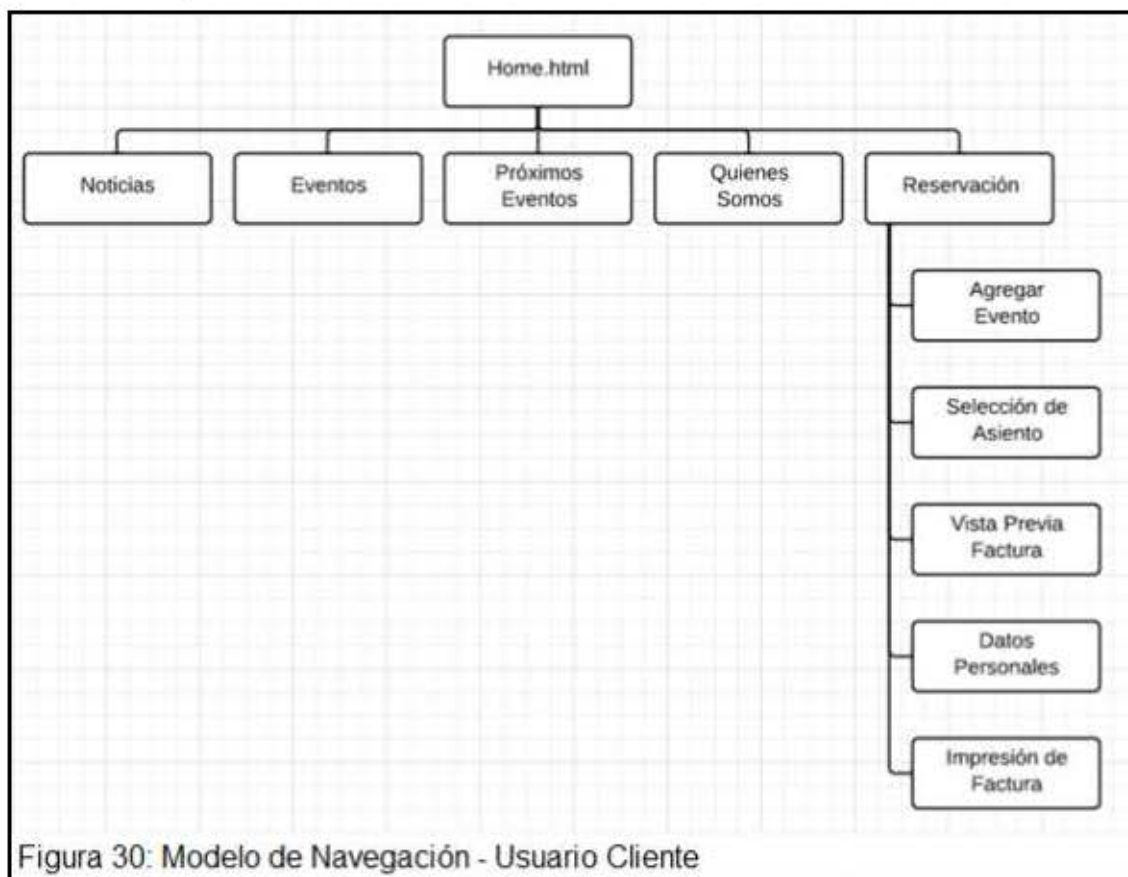
3.2.3 Navegación – Usuario Administrador



3.2.4 Navegación – Usuario Vendedor



3.2.5 Navegación – Usuario Cliente



3.3 WireFrames

El análisis de las interfaces que se encuentran en el prototipo, fue realizado mediante el uso de esta herramienta, que ayudó a crear bocetos que sirvieron de guía para el desarrollo. Para ver todos los wireframes creados dirigirse al Anexo 13.

3.3.1 Rol: Usuario sin Cuenta

Página Principal

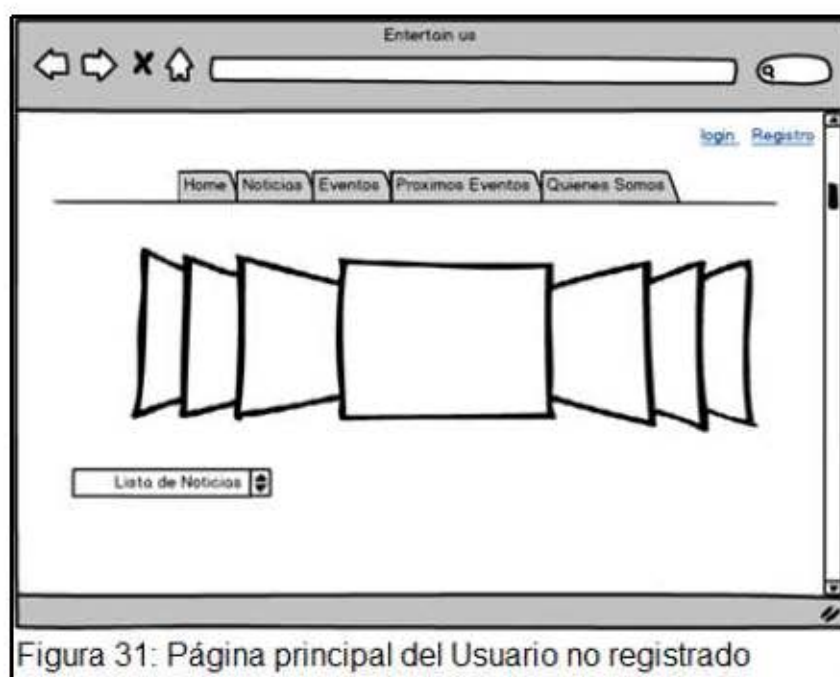


Figura 31: Página principal del Usuario no registrado

Página de Registro

Entertainment us

Inicio Noticias Eventos Proximos Eventos Quiero Sesos

Inicio Registro

Registro

Nombre: Apellidos:

Identificación: Telefono:

Fecha de Nacimiento: Email:

Sexo:

Descripción:

Calle Principal:

Calle secundaria:

Municipio:

Comentario:

Cómo quiere iniciar sesión?

Nombre de la Cuenta:

Contraseña:

Vuelva a escribir la contraseña:

Figura 32: Página de registro de usuarios

3.3.2 Rol: Usuario Administrador

Página de Administración de Localidades

Entertainment us

Inicio Parámetros Administración Eventos Reportes

Administrador

Búsqueda

Búsqueda:

Nombre de la localidad:

localidad	datos de contacto	sala	confirmación
Nombre de la localidad	estado de la localidad	número de sala	

Figura 33: Página de Administración de localidades

Página de Creación de Localidad - Datos

Entertan us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

localidad datos de contacto edits confirmación

Datos de Localidad

Compañía: Fulltheater

Nombre de localidad:

Activa:

Next

Figura 34: Página de Datos de la localidad

Página de Creación de Localidad – Datos de Contacto

Entertan us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

localidad datos de contacto edits confirmación

Direcciones

calle principal: Número:

calle secundaria: Comentario:

teléfonos

Número telefónico:

Guardar teléfono Limpiar

Teléfonos de la localidad

numero de teléfono editor

Back Next

Figura 35: Datos de Contacto de Localidad

Página de Creación de Localidad – Creación de Salas

Administrador

Home Parámetros Administración Eventos Reportes

Búsqueda

localidad datos de contacto salas confirmación

información de sala

nombre de la sala hora de apertura tiempo entre sesiones
 Número de filas en la sala Número de columnas en la sala capacidad

salas

salas de la localidad

nombre de sala	hora apertura	tiempo sesiones	número filas	número columnas	capacidad	acciones
sala 1	09:30	15	8	10	80	editar configuración de sesión eliminar configuración de sesión editar configuración de sesión

Figura 36: Creación de Salas de una Localidad

Página de Creación de Localidad - Confirmación

Administrador

Home Parámetros Administración Eventos Reportes

Búsqueda

Nombre de la localidad

localidad datos de contacto salas confirmación

Confirmación

Compañía Nombre de localidad Activo

Dirección

calle principal Número calle secundaria

Comentarios

Teléfonos

Número de teléfono

Salas

salas de la localidad

nombre de sala	hora apertura	tiempo sesiones	número filas	número columnas	capacidad
sala 1	09:30	15	8	10	80
sala 2	10:30	15	7	10	70
sala 3	09:30	15	8	10	80

Figura 37: Confirmación de Localidad

Página de Modificación de Localidad - Selección

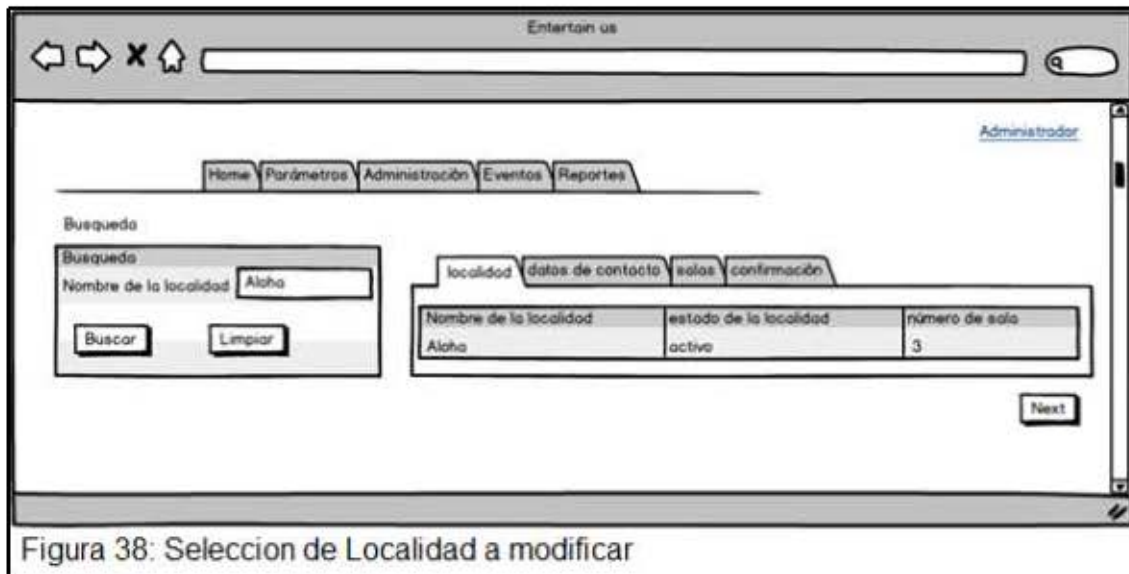


Figura 38: Selección de Localidad a modificar

Página de Modificación de Localidad – Datos de Contacto

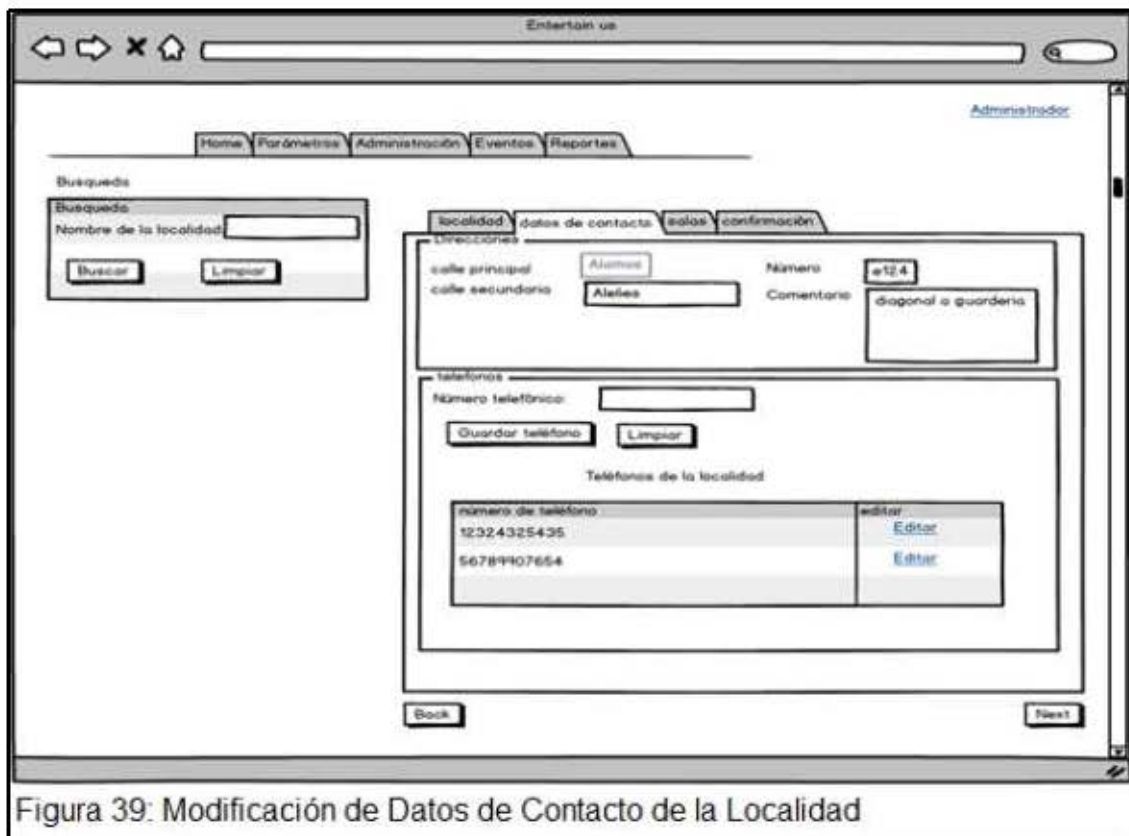


Figura 39: Modificación de Datos de Contacto de la Localidad

Página de Modificación de Localidad – Modificación y Creación de Salas

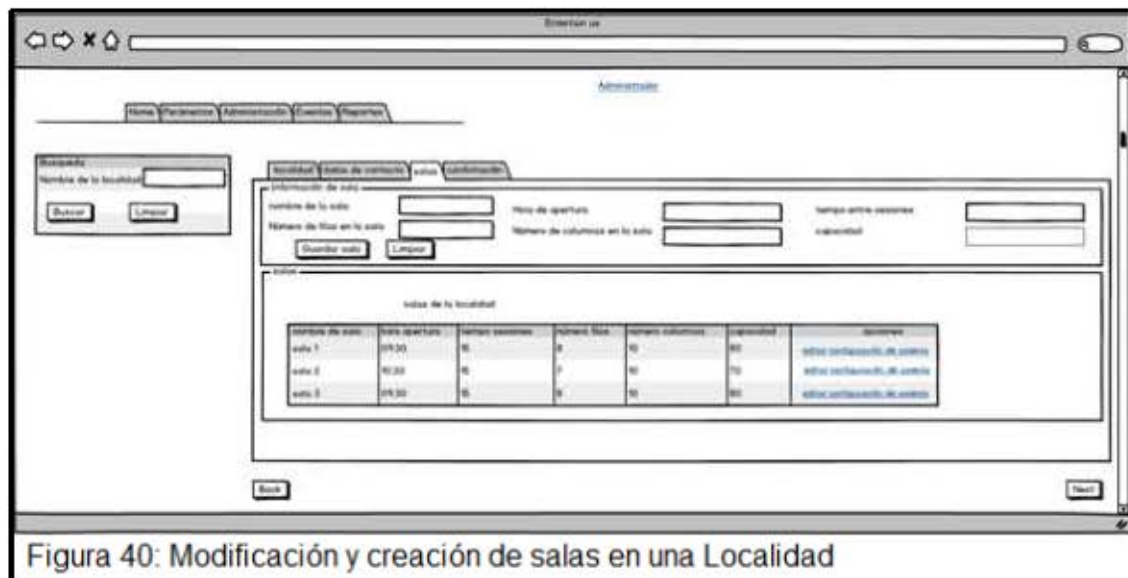


Figura 40: Modificación y creación de salas en una Localidad

Página de Modificación y Creación de Localidad – Configuración de Asientos



Figura 41: Configuración de Asientos de una sala

Página de Modificación de Localidad - Confirmación

Entrenar us

Administrador

Inicio Parámetros Administración Eventos Reportes

Búsqueda

Nombre de la localidad

Buscar Limpiar

Localidad datos de contacto salas confirmación

Confirmación

Completar: Nombre de localidad Año Activo

Dirección:

calle principal Número calle secundaria

Comentario

Teléfonos

número de teléfono

12324325430

5678907654

Salas salas de la localidad

nombre de sala	hora apertura	tiempo sesiones	número filas	número columnas	capacidad
sala 1	09:30	15	8	10	80
sala 2	10:30	15	7	10	70
sala 3	09:30	15	8	10	80

Guardar

Figura 42: Página de Confirmación de modificación de una localidad

3.3.3 Rol: Usuario Cliente

Reservación de Evento - Selección

Entertain us

Cliente

Home Noticias Eventos Proximos Eventos Quiénes Somos Reservación

Evento: Sala: calendario

Hora:

Procesar

Asientos

Siguiente

Figura 43: Reservación de evento - seleccionar datos

Reservación de Evento – Vista Previa de Reservación

Entertain us

Cliente

Home Noticias Eventos Proximos Eventos Quiénes Somos Reservación

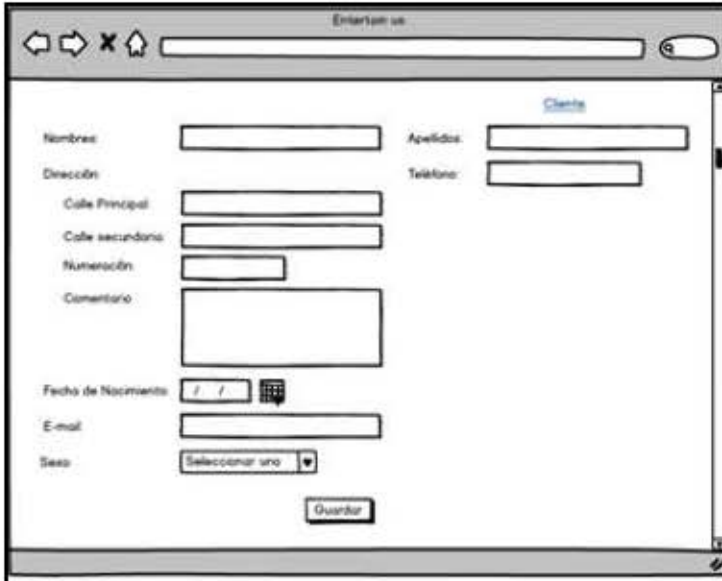
Producto (evento)	Sala/Tipo de Asiento/Asiento	Hora	Fecha	Precio	Eliminar
					Eliminar
					Eliminar
					Eliminar
					Eliminar

Total

Siguiente

Figura 44: Vista Previa de Reservación

Reservación de Evento - Datos de facturación



Entertain us

Cliente

Nombre: Apellido:

Dirección:

Calle Principal:

Calle secundaria:

Número:

Comentario:

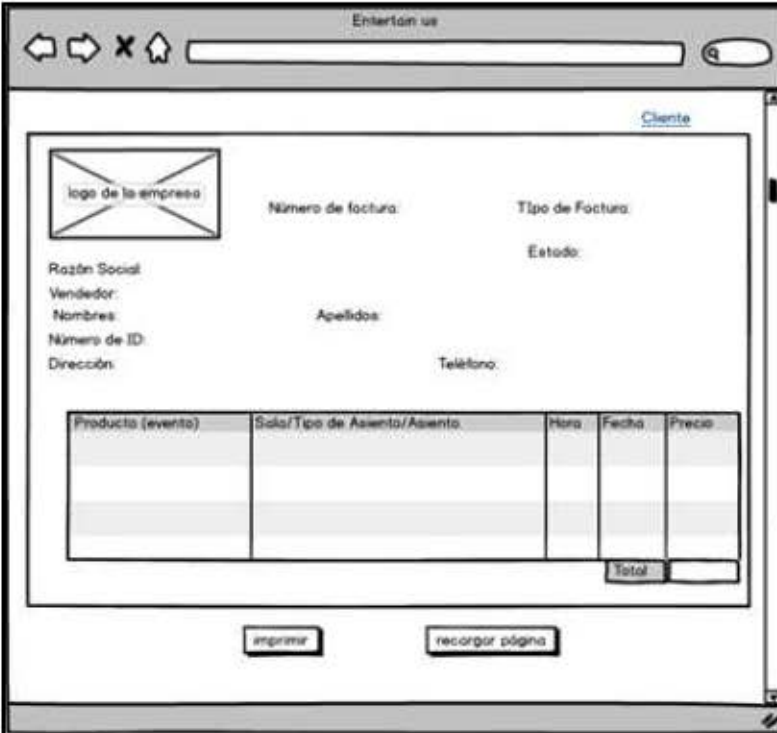
Fecha de Nacimiento: / /

E-mail:

Sexo:

Figura 45: Datos de Facturación

Reservación de Evento – Impresión de Factura



Entertain us

Cliente

Logo de la empresa:

Número de factura: Tipo de Factura:

Razón Social: Estado:

Vendedor:

Nombre: Apellido:

Número de ID:

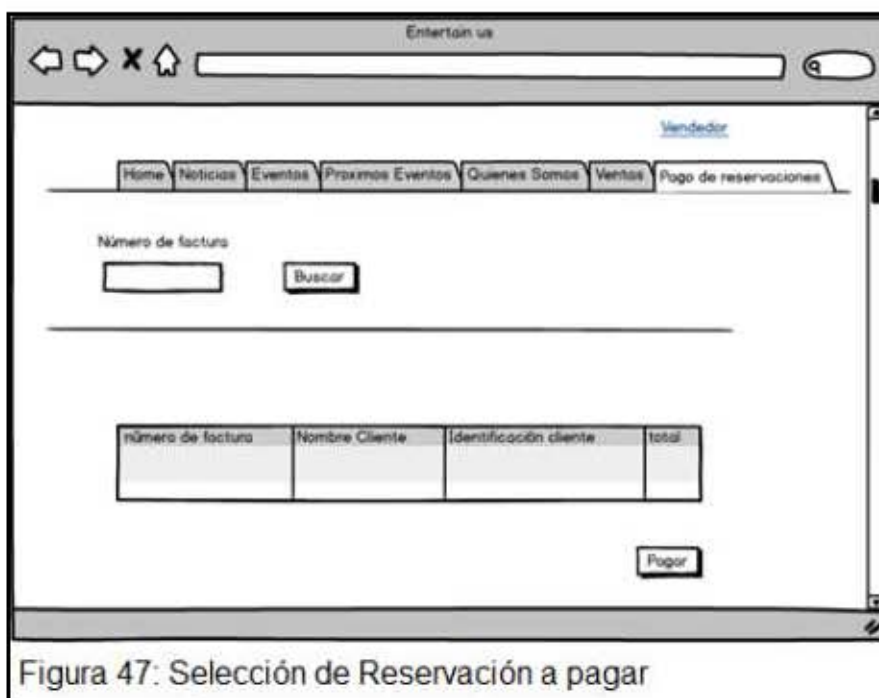
Dirección: Teléfono:

Producto (evento)	Sala/Tipo de Asiento/Asiento	Hora	Fecha	Precio
Total				<input type="text"/>

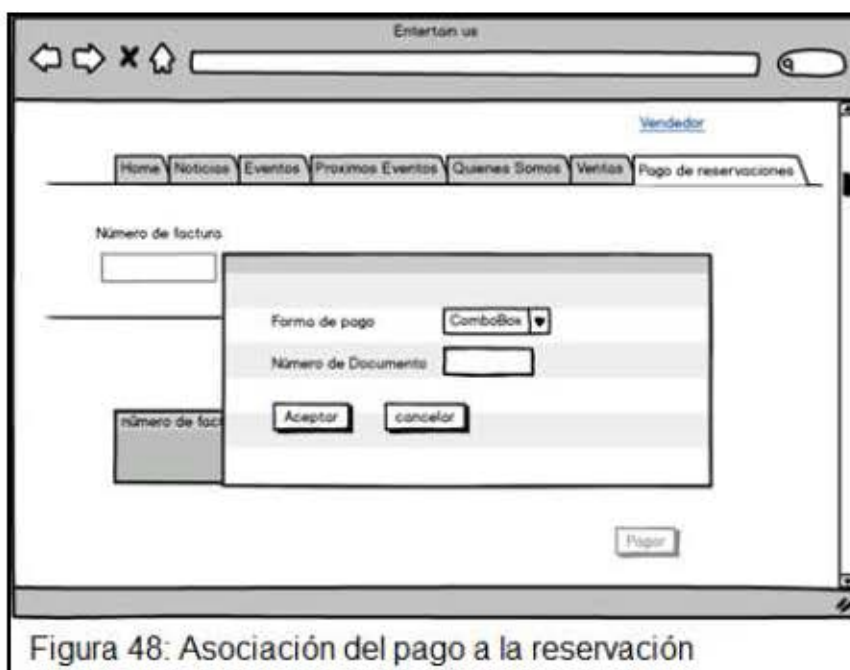
Figura 46: Impresión de Factura de Reservación

3.3.4 Rol: Usuario Vendedor

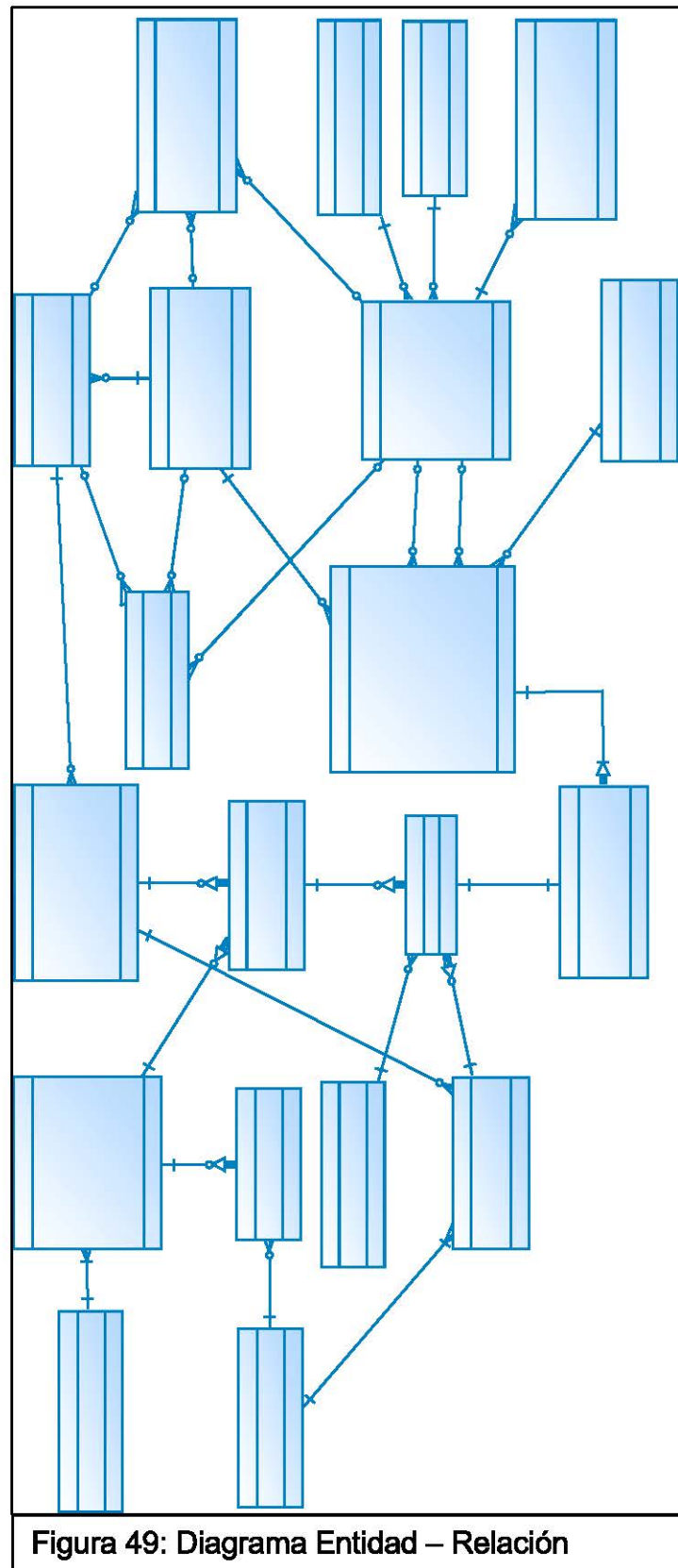
Página de Pago de Reservas - Selección



Página de Pago de Reservas - Asociación de Pago



3.4 Diagrama de Base de datos (Entidad - Relación)



3.5 Diagrama de Clases

La siguiente figura se lo colocó a manera de vista previa. Para visualizar de mejor manera este diagrama por favor ver el archivo adjunto al CD que se encuentra dentro de la carpeta Diagrama de Clases.

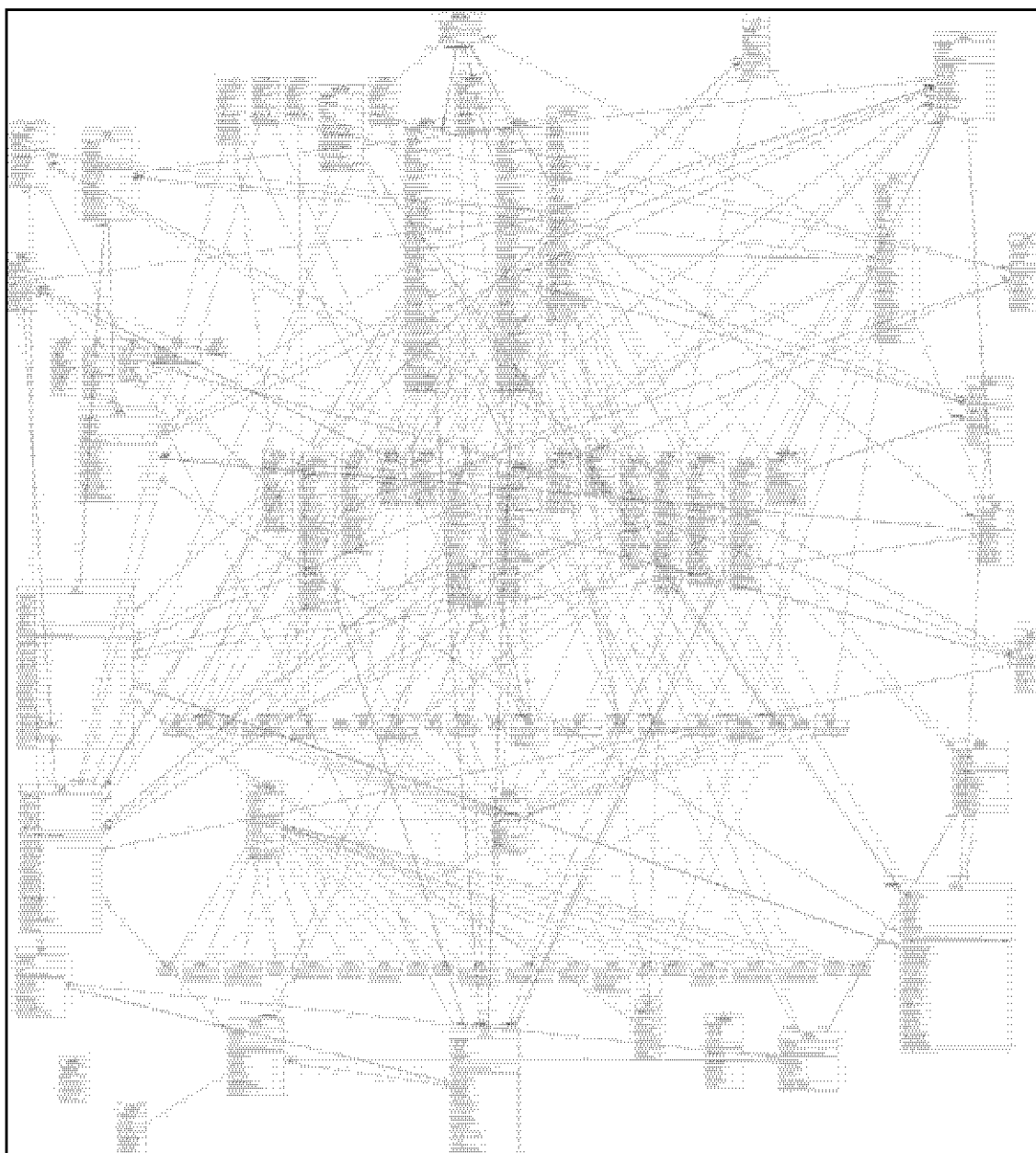


Figura 50: Diagrama de Clases

3.6 Diagrama de Casos de Uso

A continuación se despliegan algunos de los casos de uso desarrollados, para ver la lista completa de ellos por favor dirigirse al Anexo 14

3.6.1 Especificación Caso de Uso: Registro

1. Registro de un usuario que no posee una cuenta

1.1 Descripción

Permite a un usuario no registrado, ingresar su información para la creación de una cuenta de tipo cliente.

1.2 Diagramas

1.2.1 Caso de uso



1.3 Actores

- Usuario sin cuenta

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. El caso de uso inicia después de que el usuario selecciona la opción de registro
2. El sistema muestra una pantalla donde se deberá llenar la siguiente información:
 - Nombre de usuario (Requerido)
 - Identificación (Requerido)
 - Contraseña (Requerido)
 - Nombres (Requerido)
 - Apellidos (Requerido)
 - Número de ID (Requerido)
 - Teléfonos (Al menos uno)
 - Dirección: Calle Principal, número, calle secundaria (Requerido)
 - E-mail (Requerido)

- Sexo
 - Fecha de Nacimiento (Requerido)
3. El usuario selecciona la opción guardar
 4. El sistema valida que todos los datos hayan sido ingresados
 5. El sistema valida que el nombre de usuario no se repita con otros ya creados
 6. El sistema valida que la contraseña tenga un máximo de 6 caracteres
 7. El sistema guarda la información
 8. Fin del caso de uso

3.6.2 Especificación Caso de Uso: Administración de localidades

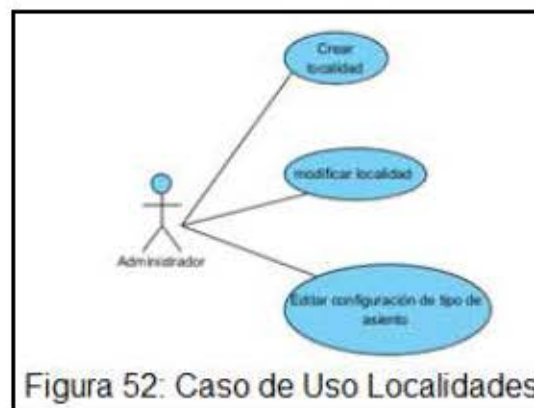
1. Administrar localidades

1.1 Descripción

Permite administrar toda la información de las diferentes localidades existentes.

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. El caso de uso inicia luego de que el Administrador selecciona la opción localidad, el sistema muestra una pantalla donde se visualiza lo siguiente:
 - Localidad
 - Datos de contacto
 - Salas
 - Confirmación
- 1.4.1.1 Crear localidad
1. Este sub flujo inicia después de que el administrador selecciona la opción nueva localidad que está dentro de la opción localidad
 2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Campo inhabilitado Compañía
 - Nombre de localidad
 - Estado
 - Opción para continuar
 3. El administrador ingresa los datos requeridos por el sistema
 4. El administrador selecciona la opción continuar
 5. El sistema muestra la pantalla de Datos de contacto donde se visualiza lo siguiente:
 - Calle principal
 - Calle secundaria
 - Número
 - Comentario
 - Número telefónico
 - Opciones de editar número telefónico
 - Tabla donde se muestran los diferentes números que posee la localidad
 - Opción para guardar número telefónico
 - Opción para limpiar
 - Opción para retroceder
 - Opción para continuar

6. El administrador ingresa el nuevo número telefónico que desea agregar
7. El administrador selecciona la opción guardar
8. El sistema despliega el nuevo número de teléfono
9. El administrador selecciona la opción continuar (caso contrario selecciona la opción retroceder y regresa a la pantalla de localidad)
10. El sistema valida que todos los campos obligatorios se hayan ingresado
11. El sistema muestra la pantalla de salas donde se visualiza lo siguiente:
 - Nombre de la sala
 - Hora de apertura tiempo entre sesiones (minutos)
 - Número de filas en la sala
 - Número de columnas en la sala
 - Capacidad (Deshabilitado)
 - Tabla donde se muestran las salas que tiene la localidad
 - Opciones de editar sala
 - Opción de configuración de asientos, es una ventana emergente donde se visualiza lo siguiente:
 - Gráfico donde se modela el orden en el que están asignados los asientos para los eventos, se genera en base al número de filas y columnas
 - Tipo de asiento
 - Número de filas desde
 - Número de filas hasta
 - opciones de: guardar, limpiar
 - Tabla donde se visualizan los datos de los asientos y sus tipos
 - opción de editar configuración de asientos
 - Opciones: retroceder, avanzar
12. El administrador ingresa los datos requeridos por el sistema

13. El sistema valida que todos los datos requeridos estén ingresados
 14. El usuario selecciona la opción guardar
 15. El sistema muestra la información ingresada en la tabla
 16. El sistema habilita la opción de configuración de asientos
 17. El administrador selecciona la opción configuración de asientos
 18. El sistema muestra un pop up de configuración de asientos
 19. El administrador ingresa todos los datos para los asientos
 20. El sistema muestra la información guardada en la tabla
 21. Se habilita la opción de editar configuración de asientos
 22. El administrador sale del pop up y selecciona la opción continuar
 23. El sistema muestra la pantalla de confirmación donde se visualiza lo siguiente:
 - Datos escogidos anteriormente inhabilitados
 - Opción de grabar
 - Opción retroceder
 24. El administrador selecciona la opción grabar
 25. El sistema guarda todos los datos que se muestran en la pantalla además de crear en la base de datos los registros de los asientos de cada sala que se ha creado.
- 1.4.1.2 Editar configuración de tipo de asiento
1. Este sub flujo inicia cuando el administrador selecciono la opción de configuración de asiento
 2. El sistema habilita todos los campos para editar
 3. El sistema habilita la opción de aceptar y cancelar
 4. El usuario edita la información
 5. El sistema valida que todos los datos requeridos por el sistema se hayan ingresado
 6. El administrador selecciona la opción aceptar
 7. El sistema almacena la información editada

1.4.1.3 Modificar localidad

1. Este sub flujo inicia después de que el administrador selecciona la opción modificar localidad
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Opción de búsqueda
 - Nombre de localidad
 - Opción de buscar y limpiar
 - Tabla con todas las localidades existentes, si el administrador desea ver la información de una localidad específica ingresa el nombre de la localidad en la opción de búsqueda y se despliega solamente la información de dicha localidad
3. Para editar el administrador selecciona la localidad que desea editar haciendo clic en la fila correspondiente a la localidad
4. El administrador selecciona la opción siguiente
5. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Campo inhabilitado de compañía
 - Nombre de localidad
 - Estado
 - Opción siguiente
6. El administrador modifica aquellos datos si lo desea
7. El administrador selecciona la opción siguiente
8. El sistema muestra la pantalla de datos de contacto:
 - Calle principal
 - Calle secundaria
 - Número
 - Comentario
 - Tabla de números telefónicos
 - Opciones de editar número telefónico
 - Opción para guardar número telefónico
 - Opciones para regresar y avanzar
9. El administrador edita los datos deseados

10. El administrador ingresa el nuevo número telefónico que desea agregar
11. El administrador selecciona la opción guardar
12. El sistema despliega el nuevo número de teléfono
13. El sistema valida que todos los campos obligatorios se hayan ingresado
14. El administrador selecciona la opción siguiente (caso contrario selecciona la opción retroceder y regresa a la pantalla de localidad)
15. El sistema muestra la pantalla de salas:
 - Nombre de la sala
 - Hora de apertura tiempo entre sesiones (minutos)
 - Número de filas en la sala
 - Número de columnas en la sala
 - Capacidad (Deshabilitado)
 - Tabla donde se muestran las salas que tiene la localidad
 - Opciones de editar sala
 - Opción de configuración de asientos
 - Opciones: retroceder, avanzar
16. El administrador edita los datos que desee
17. El sistema valida que todos los datos requeridos estén ingresados
18. El administrador selecciona la opción configuración de asientos:
 - Gráfico donde se modela el orden en el que están asignados los asientos para los eventos, se genera en base al número de filas y columnas
 - Tipo de asiento, Número de filas desde, Número de filas hasta
 - Opciones de: guardar, limpiar
 - Tabla donde se visualizan los datos requeridos por el sistema en el modal panel
 - Opción de editar configuración de asientos
19. El administrador edita los campos deseados
20. El administrador selecciona la opción siguiente
21. El sistema muestra la pantalla de confirmación:

- Datos escogidos anteriormente inhabilitados
- Opción de grabar
- Opción retroceder

22. El administrador selecciona la opción grabar

23. El sistema guarda todos los datos que se muestran en la pantalla además de crear en la base de datos los registros de los asientos de cada sala que se ha creado. Si se llegan a reducir el número de asientos de alguna localidad, el sistema además de actualizar los datos también borra los registros de esos asientos en la base de datos.

3.7 Pruebas

Se realizaron pruebas de funcionamiento del sistema en las cuales se produjeron los siguientes errores los cuales se procedió corregir:

- Renderización de cuadros
- Problemas al cargar las fotos del slide de eventos
- Cargar imagen en edición de noticias
- Carga de asientos en gráfico con los datos de búsqueda
- Al momento de renderizar datos desde HTML5 hubieron problemas al pasar información desde java script a java

Además se realizó un control de calidad de funcionamiento del módulo de administración para confirmar que cada componente está realizando la tarea que debe y de la forma que debe.

A continuación se pueden observar algunos de los casos de pruebas realizados

Tabla 9: Prueba de Registro de Cuentas

Caso de Prueba:	Registrar Cliente
Precondiciones:	El usuario no debe poseer una cuenta registrada en el sistema
Resultado:	El usuario podrá entrar en el sistema con un rol de Cliente.
Condiciones:	Los parámetros del sistema deben encontrarse configurados

Caso de prueba: Registrar Cliente
--

- | |
|--|
| <ol style="list-style-type: none"> 1. Llenar la información que solicita el sistema. <ol style="list-style-type: none"> a. Nombres: Carlos b. Apellidos: Gonzales c. Identificación: 1234567890 d. Teléfono: 2755904 e. Fecha de Nacimiento: 15/06/1986 f. E-mail: carlos_gonzales1986@mail.com g. Sexo: Masculino h. Calle Principal: Japón i. Calle Secundaria: Naciones Unidas j. Numeración: oe1-35 k. Comentario: Junto al centro comercial CCNU l. Nombre de cuenta: carlosGon m. Contraseña: 12345 2. Seleccionar la opción de Ok. 3. Si no existe ningún error en los datos ingresados aparece un mensaje indicando que se guardó exitosamente y regresa al usuario a la pantalla de inicio. En caso de que exista algún error aparece un mensaje indicando donde se cometió el error para que sea corregido. |
|--|

Tabla 10: Prueba de modificación de datos de compañía

Caso de Prueba:	Modificar Datos de la Compañía
Precondiciones:	El usuario debe haber ingresado con una cuenta que posea rol de administrador
Resultado:	El usuario podrá modificar la información de la compañía según lo necesite
Condiciones:	N/A

Caso de prueba: Modificar Datos de la Compañía
<ol style="list-style-type: none"> 1. Dar clic en el menú seleccionando la opción parámetros 2. Dentro de parámetros ingresar a Datos de la Compañía 3. Ingresar la información solicitada <ol style="list-style-type: none"> a. No. De Identificación: 174583579001 b. Razón Social: Fun Theater c. Direcciones de la compañía d. Teléfonos de la Compañía e. Quienes Somos: Somos una empresa fundada en Quito con el objetivo de proporcionar entretenimiento a la ciudadanía f. Logo de la empresa: Se selecciona una imagen 4. Seleccionar la opción Guardar. 5. Si no existe ningún error en los datos ingresados aparece un mensaje indicando que se guardó exitosamente. En caso de que exista algún error aparece un mensaje de error donde se cometió el error para que sea corregido.

4. Conclusiones y Recomendaciones

4.1 Conclusiones

- Al usar HTML5, el desarrollador debe estar siempre dispuesto al cambio, ya que esta tecnología todavía se encuentra en fase de desarrollo y pueden cambiar tanto sus especificaciones como deprecarse el uso de algunos tags en el futuro.
- El uso de wireFrames ayuda considerablemente a tener una idea preliminar clara y concisa de la funcionalidad que va a tener un sistema, ya que proporciona una línea base por donde debe seguir el desarrollo, a pesar de que no se detallan exhaustivamente elementos a nivel técnico.
- Para componentes o frameworks de Java se puede encontrar extensa documentación para realizar ejercicios y tutoriales, sin embargo documentación que no tenga un profundo nivel técnico que explique, de manera sencilla, lo que estos componentes son, como están conformados y cómo funcionan es muy escasa, ya que se prefiere realizar documentos completamente técnicos que para gente que recién incursiona en ellos se pueden convertir en algo difícil de interpretar.
- La metodología de desarrollo Scrum ha demostrado ser más que una forma de elaborar un software, esta manera de dirigir proyectos puede llevarse a casi cualquier ámbito, tal se da el ejemplo en este Trabajo de Titulación donde todo fue construido siguiendo estos lineamientos para proyectos.

4.2 Recomendaciones

- El realizar desarrollos utilizando herramientas de software libre es completamente recomendable ya que al tener a la disposición el código fuente de los API's estos pueden incluso llegar a ser modificados buscando un fin establecido.
- Existen personas que consideran mejor el uso de estándares al de frameworks. Sin embargo, dependiendo del tipo de proyecto, su extensión, costo y futuro es recomendable analizar si se lo puede empezar desde cero con el uso de estándares o emplear en el desarrollo herramientas ya existentes que acorten tanto el tiempo de desarrollo como el costo del mismo.
- Al momento de realizar una aplicación utilizando *frameworks* o tecnología que no se conoce o domina, lo mejor es primero realizar tutoriales simples como los famosos "Hola Mundo", para así familiarizarse poco a poco con estos componentes y poderlos emplear de la manera que uno necesite.
- Antes de realizar el diagrama de clases UML, es recomendable tener a la mano el diagrama de base de datos completo, ya que de cierta forma este ayudará a tener una mejor abstracción, facilitando el desarrollo del primero.
- Dado que la aplicación aquí desarrollada llegó a culminarse como un prototipo, en caso de querer colocarla en un ambiente de producción es recomendable primero integrar un componente de seguridad, como Spring Security, ya que en el actual alcance esto no fue contemplado.

Referencias

- Albaladejo, X. (s.f.). *Proyectos ágiles*. Recuperado el 16 de noviembre de 2012 de <http://www.proyectosagiles.org/historia-de-scrum>
- Albaladejo, X. (s.f.). *Proyectos ágiles*. Recuperado el 16 de noviembre de 2012 de <http://www.proyectosagiles.org/que-es-scrum>
- Andrade Castro, J. A. (2012). *Ciencia y Técnica Administrativa C&TA*. Recuperado el 7 de marzo de 2013 de <http://www.cyta.com.ar/ta1101/v11n1a1.htm>
- arcangelsombra. (2009). *SlideShare*. Recuperado el 4 de marzo de 2013 de <http://www.slideshare.net/arcangelsombra/diagramas-de-despligue-uml-1475353>
- Arévalo, M. E. (2010). *Maria Eugenia Arevalo's Blog*. Recuperado el 4 de marzo de 2013 de <http://arevalomaria.wordpress.com/2010/12/02/uml-modelado-dinamico-diagramas-de-distribucion/>
- Avalos, J. L. (2011). *Scribd*. Recuperado el 16 de febrero de 2013 de <http://es.scribd.com/doc/57257203/Metodologia-XP>
- Avsarogullari, E. (2012). *Java Code Geeks*. Recuperado el 4 de marzo de 2013 de <http://www.javacodegeeks.com/2012/04/jsf-2-primefaces-3-spring-3-hibernate-4.html>
- Barrientos, O. (2011). *BlogSpot*. Recuperado el 20 de enero de 2013 de <http://inf162expogruppo7-22011.blogspot.com/>
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Sutherland, J. (2001). *Agile Manifesto*. Recuperado el 16 de noviembre de 2012 de <http://agilemanifesto.org/iso/es/>
- Béjar, E. (2012). El problema es la ausencia de sistemas de pago en internet. (G. Alcívar, Entrevistador) Recuperado el 16 de noviembre de 2012 de <http://www.doctortecno.com/noticia/ecommerceday-ausencia-medios-pago-es-principal-problema-parte-1>

Cáceres Tello, J. (s.f.). *Universidad de Alcalá*. Recuperado el 3 de marzo de 2013 de <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>

Casasola Romero, O. (2010). *Programación en Castellano*. Recuperado el 4 de marzo de 2013 de http://www.programacion.com/articulo/introduccion_a_uml_181/7

Cohn, M. (s.f.). *Mountain Goat Software*. Recuperado el 16 de noviembre de 2012 de <http://www.mountaingoatsoftware.com/topics/scrum>

Doodrich, R. (2013). *TechNewsDaily*. Recuperado el 10 de marzo de 2013 de <http://www.technewsdaily.com/16388-what-is-html5.html>

Dra. Escalona Cuaresma, M. J., & Gutiérrez Rodríguez, J. J. (2007). *Dpto. de Lenguajes y Sistemas Informáticos – Universidad de Sevilla*. Recuperado el 4 de febrero de 2013 de http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/EAActividades.pdf

El Comercio. (2012). Ahora la capacidad del estadio Atahualpa es de 35 742 asientos. *El Comercio*. Recuperado el 16 de noviembre de 2012 de http://www.elcomercio.com/deportes/futbol/Ahora-capacidad-estadio-Atahualpa-asientos_0_758924229.html

Empresa Pública Metropolitana Quito Turismo. (2012). *Quito: Quito Turismo*. Recuperado el 16 de noviembre de 2012 de <http://www.quito.com.ec/index.php/explora-y-disfruta/rutas-y-recorridos-turisticos/quito-tour-bus>

Garat Luque, F. (2012). *SlideShare*. Recuperado el 20 de enero de 2013 de <http://www.slideshare.net/PacoGarat/crystal-clear-11805930>

Gutierrez, D. (2011). *Code Compiling*. Recuperado el 7 de marzo de 2013 de http://www.codecompiling.net/files/slides/UML_clase_04_UML_clases.pdf

- Gutierrez, D. (2011). *Code Compiling*. Recuperado el 7 de marzo de 2013 de http://www.codecompiling.net/files/slides/UML_clase_02_UML_casos_de_uso.pdf
- Herrera, A. (2012). *Postgres*. Recuperado el 7 de marzo de 2013 de <http://alvherre.cl/pgsql/modBasico/node3.html> Descargado 03/07/2013
- HTML5 Rocks. (s.f.). *HTML5 Rocks*. Recuperado el 3 de octubre de 2012 de <http://www.html5rocks.com/en/why>
- Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., . . . Webb, P. (2013). *Spring Framework Reference Documentation*. Recuperado el 14 de marzo de 2013 de <http://static.springsource.org/spring/docs/3.2.x/spring-framework-reference/html/mvc.html>
- Katamreddy, S. (2011). *DZone*. Recuperado el 17 de marzo de 2013 de <http://java.dzone.com/articles/primefaces-quickstart-tutorial>
- Khan, Q. (2010). *Developer Articles*. Recuperado el 17 de marzo de 2013 de <http://developerarticles.com/jsf-2-0-introduction-and-overview/>
- Kieman, J. (s.f.). *About.com*. Recuperado el 4 de marzo de 2013 de <http://webdesign.about.com/od/webdesign/qt/website-wireframes.htm>
- Koch, P. P. (2013). *quirksmode*. Recuperado el 3 de marzo de 2013 de <http://www.quirksmode.org/js/intro.html>
- Letelier, P., & Penadés, M. d. (2006). *Ciencia y Técnica Administrativa*. Recuperado el 16 de noviembre de 2012 de <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Miller, E. (s.f.). *About.com*. Recuperado el 4 de marzo de 2013 de http://graphicdesign.about.com/od/effectivewebsites/ss/wireframes_all.htm

- Osmosis Latina. (2005). *Osmosis Latina*. Recuperado el 4 de abril de 2013 de <http://www.osmosislatina.com/lenguajes/uml/actividad.htm>
- Patel, V. (2010). *viralpatel.net*. Recuperado el 14 de marzo de 2013 de <http://viralpatel.net/blogs/tutorial-spring-3-mvc-introduction-spring-mvc-framework/>
- Pilgrim, M. (2011). *Dive into HTML5*. Recuperado el 3 de marzo de 2013 de <http://diveintohtml5.info/index.html>
- Ploski, R. (2011). *SlideShare*. Recuperado el 17 de marzo de 2013 de <http://www.slideshare.net/rayploski/jboss-application-server-7>
- Rodriguez, Y. (2011). *Scribd*. Recuperado el 16 de febrero de 2013 de <http://es.scribd.com/doc/72420606/METODOLOGIA-XP>
- Rouse, M. (2009). *WhatIs.com*. Recuperado el 17 de marzo de 2013 de <http://whatis.techtarget.com/definition/JBoss>
- Salinas Caro, P. (s.f.). *Universidad de Chile*. Recuperado el 7 de marzo de 2013 de <http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>
- Salinas Caro, P. (s.f.). *Universidad de Chile*. Recuperado el 7 de marzo de 2013 de <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>
- Sookman, J. (2012). *Startup Life*. Recuperado el 7 de marzo de 2013 de <http://www.startuplifeblog.com/tag/product-backlog/>
- Storti, G., Ríos, G., & Campodónico, G. (2007). *Colegio Manuel Belgrano*. Recuperado el 7 de marzo de 2013 de http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf
- Sutherland, J. (s.f.). *scruminc*. Recuperado el 16 de noviembre de 2012 de <http://scrum.jeffsutherland.com/>
- The World Wide Web Consortium (W3C). (2002). *W3C*. Recuperado el 10 de marzo de 2013 de <http://www.w3.org/TR/xhtml1/introduction.html#why>

- The World Wide Web Consortium (W3C). (2008). *W3C*. Recuperado el 10 de marzo de 2013 de <http://www.w3.org/DOM/#what>
- The World Wide Web Consortium (W3C). (2012). *W3C*. Recuperado el 10 de marzo de 2013 de <http://www.w3.org/html/wg/drafts/html/master/>
- UNAM Posgrado. (1999). *UNAM Posgrado*. Recuperado el 7 de marzo de 2013 de <http://www.mcc.unam.mx/~cursos/Objetos/Cap8/cap8.html>
- Universidad Unión Bolivariana. (s.f.). *Universidad Unión Bolivariana*. Recuperado el 20 de enero de 2013 de http://ingenieriadesoftware.mex.tl/59189_Metodologia-Crystal.html
- VersionOne Inc. (s.f.). *versionone.com*. Recuperado el 6 de enero de 2013 de <http://www.versionone.com/Agile101/Agile-Development-Methodologies-Scrum-Kanban-Lean-XP/>
- Way, J. (2011). *net tuts+*. Recuperado el 3 de marzo de 2013 de <http://net.tutsplus.com/tutorials/html-css-techniques/25-html5-features-tips-and-techniques-you-must-know/>
- Wilton Jones, M. (2011). *How to Create*. Recuperado el 17 de marzo de 2013 de <http://www.howtcreate.co.uk/tutorials/javascript/introduction>

Anexos

Anexo 1: Crystal Clear

Crystal es una familia de metodologías de desarrollo ágil que, dependiendo de su enfoque, dan prioridad al tamaño del equipo, a la criticidad del proyecto o a las prioridades de los desarrollos. Se dividen en segmentos de colores, de la siguiente manera:

Familia de metodologías Crystal

		Prioridad por productividad y tolerancia					
Criticidad Los defectos causan perdida de ...	Vidas	L6	L20	L40	L100	L200	L500
	Cantidad Esencial de dinero	E6	E20	E40	E100	E200	E500
	Cantidad Discrecional de dinero	D6	D20	D40	D100	D200	D500
	Conformidad	C6	C20	C40	C100	C200	C500
		1 - 6	7 - 20	21 - 40	41 - 100	101 - 200	201 - 500
		Número de personas involucradas					

Donde las metodologías que se pueden observar son:

- De 1 a 6 personas Crystal Clear
- De 7 a 20 personas Crystal Yellow
- De 21 a 40 personas Crystal Orange
- De 41 a 100 personas Crystal Red
- De 101 a 200 personas Crystal Blue
- De 201 a 500 personas Crystal Violet

En este gráfico es posible evidenciar que ocurre en cada metodología en el caso de que empiecen a aparecer defectos o problemas durante el desarrollo de un proyecto determinado.

Su creador Alistair Cockburn las desarrollo mediante una investigación en los métodos de desarrollo de IBM, donde pudo llegar a la conclusión de que diferentes proyectos requieren de metodologías distintas. En estas metodologías se da un valor de suma importancia a la comunicación, dejando a un lado la documentación y priorizando entrevistas entre las personas involucradas.

Propiedades

Al ser una familia de metodologías todas tienen sus propias características, sin embargo todas comparten las siguientes propiedades:

Entrega Frecuente: Puede referirse tanto a código como a documentación, su importancia radica en conseguir una valoración en cuanto al avance del proyecto, presentar un enfoque claro a pesar de los cambios que puede el usuario final proponer y favorecer estos cambios en una etapa temprana del desarrollo del proyecto.

Comunicación Osmótica: El obligar al equipo de trabajo que desarrolle sus tareas en una sola sala permite que todos los integrantes del equipo estén al tanto de cualquier decisión tomada respecto al mismo y que de esta manera también puedan dar su opinión de sobre cualquier tema que surja en el desarrollo del proyecto.

Mejora Reflexiva: Se trata de mantener una conciencia de mejora continua en todo el equipo, esto se logra proponiéndoles que de vez en cuando es necesario detenerse un momento en las tareas y dedicar unas horas a responderse las siguientes preguntas:

- ¿Qué debemos guardar?
- ¿Dónde estamos teniendo problemas?
- ¿Qué es lo que vamos a hacer en la siguiente iteración?

Seguridad Personal: Propone el generar confianza entre los distintos miembros del equipo, esta confianza radica en la honestidad de cada miembro al indicar la ignorancia o incapacidad sobre algún tema o tarea y la responsabilidad del grupo en ayudar a los integrantes a sobrellevar este cualquier tipo de problema que puedan presentarse.

Enfoque: Todo el equipo debe ser consciente en cuáles son las prioridades del proyecto, así como los tiempos de respuesta esperados para que de esta manera se pueda trabajar con una mayor y sabiendo hacia donde se pretende ir con el proyecto.

Acceso Fácil a los especialistas: El tener a la mano una voz experta puede simbolizar una mejor predisposición al cambio, sin perder el rumbo o la realidad de lo que es posible desarrollar, así como simboliza una gran ayuda para resolver e instruir al equipo de trabajo en áreas concretas que afectan el desarrollo del proyecto.

Técnicas o Actividades

Crystal no ofrece un conjunto de técnicas o actividades, tal como lo hacen las otras metodologías de desarrollo ágil, sino que da uso a las ya existentes, algunos ejemplos son:

Exploración de 360°: Consiste en la verificación o en la toma de los datos del negocio para poder levantar requerimientos, la tecnología necesaria, el modelo del dominio y el proceso que son referentes al proyecto que se va a desarrollar, puede tomar entre 2 a 4 semanas el realizar toda esta actividad.

Victoria Temprana: Es la búsqueda de pequeños triunfos para al final poder aspirar una gran victoria, esta herramienta hace referencia a los pequeños entregables que se extienden al cliente en cortos tiempos para de esta manera ganar la confianza de los inversores y lograr una buena expectativa del proyecto completo.

Esqueleto Ambulante: Al principio del proyecto, es necesario lograr una estructura (esqueleto), que pueda ser de cierto modo independiente, y que simplemente sea añadida toda la funcionalidad necesaria.

Re-Arquitectura Incremental: La arquitectura del proyecto, no debe ser un planteamiento estático sino dinámico, es decir tiene que evolucionar por etapas conforme el proyecto avanza, logrando así mantener el sistema en ejecución mientras la arquitectura cambia.

Roles y los Artefactos que los Definen

Como se ha explicado previamente, dependiendo de cuál integrante de la familia de metodologías Crystal se vaya escoger también de esto depende la cantidad de roles que se plantean. A continuación se describirán los ocho roles correspondientes a Crystal Clear:

Patrocinador: Es el encargado de definir el proyecto completamente, es decir desde conseguir los recursos para realizarlo hasta realizar la declaración de Misión con prioridades de Compromiso

Usuario Experto: Es quien junto con un experto en el negocio se encargara de elabora los archivos de requerimientos y casos de uso, así como producir una lista de actores y objetivos que se desean ver cumplidos en el proyecto

Diseñador Principal: En el entorno de desarrollo ágil se encuentra definido que los profesionales pueden tener tres niveles de experiencia, definidos de la siguiente manera:

- *Nivel 1:* Es capaz de seguir los procedimientos establecidos
- *Nivel 2:* Le es posible el apartarse de los procedimientos existentes para poder encontrar distintos que se puedan acoplar de una buena manera al proyecto.
- *Nivel 3:* Maneja de manera fluida los procedimientos así como los puede rediseñar e inventar para obtener mejores resultados.

El diseñador principal debe corresponder a esta última categoría y ser capaz de ejercer los roles de arquitecto, coordinador, mentor y programador experto, ya que es el encargado de realizar la arquitectura del proyecto a realizarse.

Programador/Diseñador: En Crystal no es posible ser un diseñador si no eres un programador también, ya que esta persona debe realizar (junto con el Diseñador Principal) los borradores de pantallas, modelos de domino, diagramas de diseño, desarrollo del código fuente, pruebas y los sistemas de migración y empaquetado.

Experto en Negocios: Es el encargado de conocer la forma, normas y políticas del negocio donde se va a aplicar el proyecto, ya que junto con el Usuario Experto deben ser los que definan los archivos de donde partirá el proyecto (ver Usuario Experto)

Coordinador: El mapa del proyecto, el plan de entrega, la lista de riesgos, la Agenda de visualización y el estado del proyecto son lo que corresponde a sus responsabilidades por realizar, sin embargo cabe aclarar que para poder realizar esto el Coordinador debe apoyarse en todo el equipo.

Verificador: Corresponde a una o más personas que se dedicaran a encontrar errores y bugs en el proyecto, también es quien levantará un reporte sobre los mismo.

Escritor: Es el que se encargará de realizar todos los manuales que sean necesarios entregar, como el de instalación, manual de usuario, etc.

Anexo 2: Extreme Programming (XP)

Programación Extrema es una metodología de desarrollo de software ágil diseñada por Kent Beck en 1996 con el objetivo de disminuir la burocracia que existe al momento de poder realizar cambios en un proyecto ya iniciado.

XP se basa en potenciar las relaciones entre los miembros de un equipo de trabajo, así como promoviendo el trabajo en equipo, el aprendizaje de los miembros y una retroalimentación continua con el cliente para cumplir con sus expectativas del producto final.



Resumen de XP

Valores

Comunicación: Debe encontrarse presente entre los miembros del equipo así como dentro del sistema en sí. Es uno de los factores de mayor importancia que debe ser cultivado dentro de los integrantes del equipo XP.

Simplicidad: Siempre empezar solo con lo requerido y necesario, para tener un comienzo ordenado y simplificado. El código desarrollado debe mantenerse simplificado y entendible durante todo el proyecto.

Retroalimentación: Este valor va ligado con el de comunicación ya que todo el equipo debe estar enterado del proyecto, incluidos los desarrolladores y el cliente. Además el sistema debe ser capaz de retroalimentar de información a cualquier desarrollador que vaya a tomar un módulo que ya fue trabajado.

Valentía: En XP se dice que “Se debe trabajar para hoy y no para mañana”, el desarrollador debe tener la suficiente valentía para tomar los retos presentados en el día a día y responsabilizarse de todo el proyecto del cual forma parte.

Respeto: Un equipo está conformado por personas individuales pero dentro de la metodología se debe tomar en cuenta que todos conforma un unidad y que por ende deben respetar las opiniones y desarrollos de cada persona. Cada decisión debe ser tomada en conjunto nunca deben ser repentinas autoritarias, a más de que la criticidad del asunto requiera una intervención de esta manera.

Características

Desarrollo Iterativo e Incremental: Todo el desarrollo está orientado a realizar mejoras pequeñas en poco tiempo y una tras otra.

Pruebas Unitarias Continuas: La automatización de pruebas tanto unitarias como de regresión ayuda al desarrollo rápido ya que permite evidenciar que el funcionamiento del sistema no se haya visto alterado por un cambio ni perjudicado, de igual manera se aconseja programar estas pruebas antes de realizar la codificación, para que al momento de empezar el desarrollo, propiamente dicho, se tenga en mente el objetivo funcional de lo que se desea lograr

Programación en Parejas: Este conforma un punto de cuestión entre los administradores de proyectos sin embargo es una de las piedras angulares del desarrollo utilizando la metodología ágil XP, plantea que dos programadores utilicen un solo puesto y una misma computadora al mismo tiempo, aprovechando de esta manera la discusión que se genera al escribir el código, la revisión inmediata que sucede y el aporte de dos personas en cuanto conocimientos para desarrollar un módulo.

Integrar al Cliente al Equipo de Trabajo: Esta metodología plantea que un representante del cliente, que posea conocimientos de la forma en que opera el negocio, se integre como parte del equipo de trabajo para que trabaje tanto en las pruebas como en la aceptación de los módulos programados.

Entrega sin Errores: Al tener programadas pruebas funcionales y de regresión es posible probar el sistema completo en busca de errores y corregirlos antes de lanzar una nueva versión.

Refactorización de Código: Se recomienda que el equipo de desarrollo esté constantemente reescribiendo ciertas partes del código para aumentar la mantenibilidad, rendimiento y legibilidad, sin que esto involucre cambios en el comportamiento del mismo. Las pruebas son las encargadas de verificar que la refactorización no haya causado ningún daño ni alteración a la funcionalidad de los módulos.

Propiedad Colectiva del Código: XP plantea que todo el equipo de trabajo es dueño y responsable del código, dándoles la oportunidad de que cualquier integrante pueda realizar mejoras y cambios a cualquier módulo programado.

Simplicidad: Todo lo programado utilizando esta metodología debe ser simple y capaz de auto-explicarse, es decir se tiene que dar mayor importancia a crear un código simple y entendible, donde se puede ocupar un poco de trabajo extra para modificarlo, que algo complejo que, pueda incluso, no ser utilizado nunca o que necesite de la persona que lo programó para entenderlo. La comunicación en este factor radica en que el sistema tiene que comunicarse por sí solo a los desarrolladores.

Roles y Responsabilidades

Programador: Un personaje de mayor importancia en esta metodología, ya que va a ocuparse de las responsabilidades inherentes al código. Es decir es quién garantiza la programación del código tanto funcional como de pruebas, por ende de la integridad del sistema completo. Siempre debe estar abierto a críticas y tener una buena capacidad de comunicación.

Cliente: Es el encargado de definir las especificaciones funcionales, siempre debe estar influyendo en el desarrollo de los módulos, pero sin controlar a los programadores, es decir debe confiar en ellos. Además es quien debe especificar que deben comprobar las pruebas funcionales.

Encargado de Pruebas: Debe apoyar al cliente en la definición y realización de las pruebas funcionales, así como ejecutarlas y presentar los resultados de las mismas al equipo de desarrollo para una oportuna corrección de errores

Encargado de Seguimiento: Sus tareas involucran la recopilación de información en medio de la marcha del proyecto, pero este trabajo no debe afectar demasiado en el desarrollo del mismo. También debe encargarse de supervisar el cumplimiento de los objetivos en cada iteración y de la corrección de los errores que hayan asomado dentro de cada una.

Entrenador: Su perfil es de un experto en XP, ya que debe dirigir el proyecto en su totalidad identificando posibles desviaciones y reclamando la atención debida para cada caso, sin dañar la seguridad y confianza del equipo de desarrollo ni de ninguno de sus miembros. Igualmente debe intervenir de manera directa si llega a surgir un problema que necesite de su intervención.

Consultor: Puede ser una o varias personas que intervendrán en el proyecto en momentos oportunos y en las áreas que requieran de su apoyo.

Jefe del Proyecto: Tiene plena confianza en su equipo, debe cubrir todas las necesidades del proyecto para así asegurar que se alcancen los objetivos, sin embargo su mayor tarea consisten en favorecer las relaciones entre los usuarios y los desarrolladores.

Funcionamiento

Planeación: Es también llamado juego de planeación, es donde el equipo XP (incluido el Cliente) se junta para crear las llamadas Historias de usuario donde se plantean las necesidades indicadas por el cliente en prioridades de presentación y donde los desarrolladores colocan un tiempo aproximado de elaboración de la tarea (en caso de que una tarea tome más de 2 semanas,

que es el tiempo de cada iteración, se debe pedir al cliente que lo descomponga en subtemas), en cualquier momento del proyecto se pueden incluir nuevas historias que deberán pasar por el mismo proceso.

Una vez se llega a un acuerdo sobre que historias se van a presentar, el equipo de desarrolladores debe clasificarlas siguiendo alguno de los siguientes criterios:

1. Las historias que se implementarán de inmediato
2. Las historias con mayor valor que se integrarán al desarrollo de manera inmediata
3. Las historias de mayor riesgo que deben ser colocadas como primordiales al desarrollo y por ende elaboradas enseguida.

Diseño: Para esta etapa se estimula el uso de tarjetas clase-responsabilidad-colaborador (CRC) donde se organiza según clases orientadas a objetos que abastecerán al sistema ya existente. Toda la etapa se la debe mantener bajo el valor de Simplicidad para así poder seguir con la creación de un prototipo de diseño, llamado solución en punta, para poder minimizar el riesgo cuando comience la verdadera implementación en base a las historias de usuario que conforman el problema.

El diseño es en realidad un artefacto de transición ya que XP no implementa mucha documentación para el proceso del proyecto. Por ende esta es una etapa que se la utiliza tanto antes como después de la codificación y que esta supuesta a gran variedad de cambios.

Codificación: Este proceso comienza con la elaboración de las pruebas unitarias que requieren las historias de usuario, y una vez hechas, los desarrolladores se pueden centrar en la implementación de su componente tomando siempre en cuenta que debe pasar las pruebas que realizaron para el mismo.

En esta etapa se usa la herramienta de programación en parejas, una vez que la pareja de programadores termina su componente este debe integrarse al

trabajo de los demás colaboradores. En caso de tener un equipo de integración esta tarea la deben desarrollar ellos y comunicar cualquier inconveniente que haya surgido durante y después del proceso para su corrección.

Pruebas: Todas las pruebas realizadas previamente deben tener la cualidad de poder ser automatizadas y repetidas la cantidad de veces que sea necesario sin causar ningún inconveniente al proyecto, dada la característica de la refactorización las pruebas de regresión deben ser una estrategia tomada en cuenta por los desarrolladores para cualquier aumento o cambio en el sistema.

Siempre que se acabe la integración de un desarrollo las pruebas deben ser ejecutadas por el equipo designado para esta tarea.

Existe un grupo de pruebas adicional que se hace uso en esta etapa, las pruebas de aceptación, donde el cliente se sienta con los testers para calificar las características y funcionalidad del componente agregado.

Cualquier error, bug o corrección que se deba hacer a causa de cualquier prueba debe ser reportado en esta etapa para la corrección, en caso de ser necesaria (esto es si el error impide la liberación de la nueva versión), dentro de la misma iteración o en la siguiente.

Anexo 3: Funcionalidad Adicional de HTML5

Video y Audio

Se los presenta con los nuevos *tags* `<video>` y `<audio>`, ambos aún se encuentran en desarrollo, por lo que no pueden soportar todos los formatos existentes, ni tampoco el mismo formato para los navegadores que lo soportan. Lamentablemente cada navegador, por cuestiones de licencias y derechos, solo soporta formatos específicos para reproducir los videos y pistas de audio.

Navegadores que soportan HTML5 con sus codecs de audio y video

	Formato de Video	Codec de Video	Codec de Audio
Google Chrome	mp4, ogg, webM	H.264, Theora, VP8	mp3, vorbis
Mozilla FireFox	ogg, webM	Theora, VP8	vorbis
Internet Explorer	mp4, webM	H.264, VP8	mp3
Opera	ogg, webM	Theora, VP8	vorbis

Fuera de Línea

A pesar de que una aplicación web, por naturaleza tiene la necesidad de encontrarse en un ambiente que tiene conexión a internet, existe la posibilidad de que esta se pierda y que se pierdan los datos ahí suministrados.

HTML5 da la posibilidad a los programadores web de indicar que acciones realizar al momento que se pierde la conexión, incluso gracias al archivo denominado *Manifest* es posible almacenar, en la computadora cliente, los recursos de la aplicación para poder seguir ejecutándola y guardando información provista por el usuario para luego sincronizarla cuando la conexión vuelva.

Geo Localización

Para poder representar el lugar donde alguien se encuentra en una página web, siempre fue necesario el uso de bibliotecas de terceros, esta versión de

HTML posee un API propio para realizar esta tarea y los navegadores poseen la implementación para obtener la longitud y latitud para poder utilizarlas por medio de JavaScript. Al ser información sensible los navegadores lanzarán una *advertencia* al usuario para que él decida si permite que se envíe su información de ubicación.

Formas de Ingreso de Datos

HTML5 proporciona al desarrollador varios nuevos tipos de ingreso de información, muchos de estos en las versiones anteriores solo eran posibles mediante el uso de librerías de terceros, sin embargo ahora son parte de la especificación. Estos son:

Marcador de Posición

Un marcador de posición es un tipo de texto que se muestra solo si mientras la caja se encuentre vacía, si se empieza a escribir dentro de la caja de texto se pierde el *PlaceHolder*. Para utilizarlo se coloca el atributo *placeholder* dentro de la etiqueta `<input>`



The image shows a user interface element consisting of a text input field and a button. The text input field contains the placeholder text "Ejemplo placeholder". To the right of the input field is a button labeled "Buscar".

Ejemplo placeholder

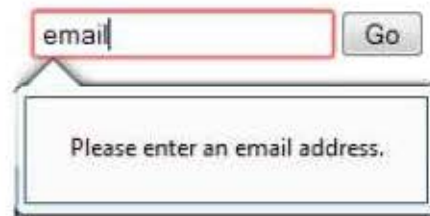
Auto Enfoque

Todas las cajas de texto en esta especificación vienen con el atributo *autofocus* para poder ser utilizado si se desea, este atributo tiene la cualidad de llevar el enfoque del cursor directamente a la caja de texto cuando se carga la página.

Ingreso de Email

Este nuevo tipo de caja de texto posee la ventaja de validar que el texto se encuentre con formato de una dirección de correo electrónico es decir que contenga un símbolo @ y un punto. Además para dispositivos móviles táctiles, cuenta con la facilidad de indicarle al dispositivo que debe hacer aparecer el teclado para e-mails (sin espacios en blanco y con las teclas punto y arroba). Para utilizarla, se coloca la etiqueta `<input type="email">`

En caso de que el navegador no soporte este tipo de caja de texto, se presentará sencillamente como una caja de texto normal.



Ejemplo de Input tipo email

Listas de números

Esta caja de texto viene con la facilidad de un par de botones al lado derecho, que sirven para navegar en la lista de los números, no todos los exploradores lo soportan tal es el caso de *Mozilla FireFox* e *Internet Explorer*. Para utilizarlo se necesita colocar en el atributo *type* el valor "*number*", además de proporcionar valores a los atributos *min*, *max*, *step* y *value*.



Ejemplo de Spinbox numérico

En caso de que no sea soportado se presentará como una caja de texto normal con el valor del atributo *value* como contenido de la caja de texto.

Rangos de números en deslizadores

Se lo utiliza de igual manera que la caja de texto anterior, sin embargo el valor del atributo *type* es "*range*", y este permite desplegar un deslizador donde se puede colocar el cursor en el valor que se desea obtener.



Ejemplo de un Slider numérico

Si el navegador no soporta este tipo de entrada, ocurre exactamente lo mismo que en el caso anterior.

Seleccionador de Fechas

Es un tipo de caja de texto especial que permite ingresar fechas y horas, sin embargo esta puede ser usada de las siguientes maneras:

- **Fecha:** Si en el atributo *type* existe el valor *"date"*, se proporciona la siguiente caja de texto para seleccionar la fecha.

01/12/2013

January, 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Ejemplo de seleccionador de fecha del tipo date

- **Mes:** Si en el atributo *type* existe el valor *"month"*, se proporciona la siguiente caja de texto para seleccionar el mes.

December, 2013

November, 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Ejemplo de seleccionador de fecha del tipo month

- **Semana.** Si en el atributo *type* existe el valor "week", se proporciona la siguiente caja de texto para seleccionar la semana.

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
48	24	25	26	27	28	29	30
49	1	2	3	4	5	6	7
50	8	9	10	11	12	13	14
51	15	16	17	18	19	20	21
52	22	23	24	25	26	27	28
1	29	30	31	1	2	3	4

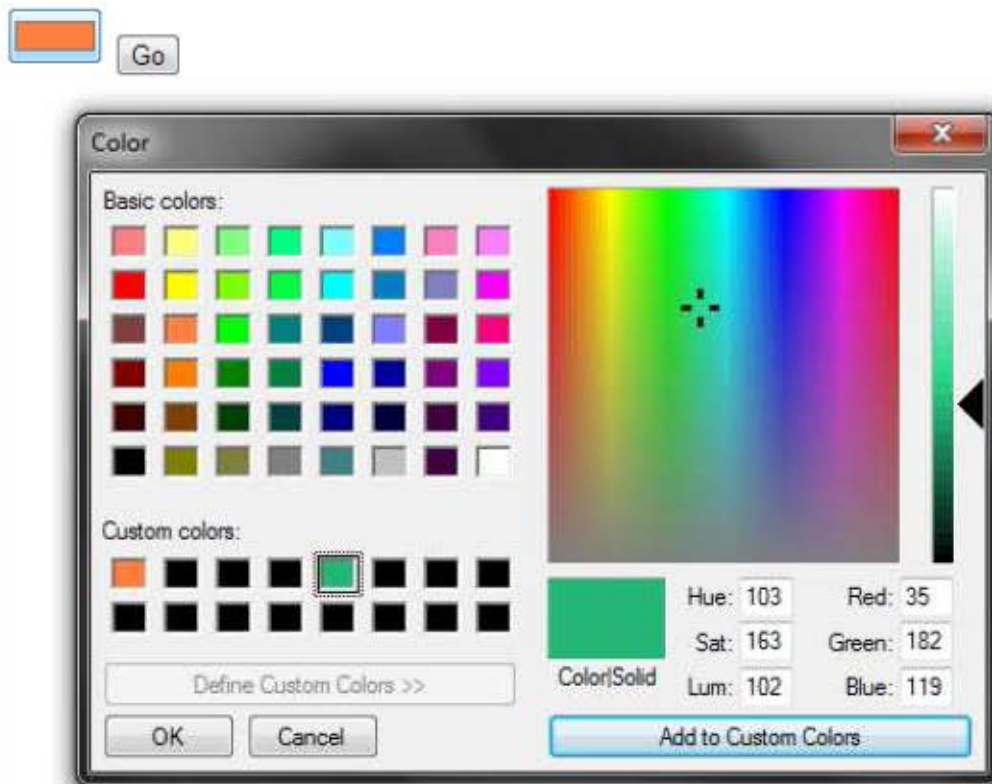
Ejemplo de seleccionador de fecha del tipo week

- **Hora:** Si en el atributo *type* existe el valor "time", se proporciona la siguiente caja de texto para seleccionar la hora.

Ejemplo de seleccionador de fecha del tipo time

Seleccionador de colores

Otro tipo de Ingreso de datos, ahora nativo en HTML5, es el seleccionador de colores, tiene la funcionalidad de abrir la paleta de colores y permitir escoger el color deseado, una vez seleccionado, transfiere al código de la página su valor hexadecimal de la escala RGB. Para utilizarlo se coloca el valor "color" en el atributo *type* de la etiqueta `<input>`. Es posible visualizar este control de la siguiente manera:



Ejemplo de seleccionador de colores

Cajas de búsqueda

La conforma una caja de texto bastante común, sin embargo en algunos exploradores como Chrome y Safari se presenta como una caja donde, cuando se empieza a escribir, aparece una "X" al final, que tiene la función de limpiar el contenido, y dar la apariencia de una caja de búsqueda. A pesar de esto, es necesario implementar la búsqueda con código. Su apariencia es la siguiente:



Ejemplo de una caja de tipo búsqueda

Anexo 4: Ejemplo HTML5

El siguiente ejemplo demuestra las cualidades que pueden presentar las funcionalidades de *canvas* y almacenamiento que presenta HTML5. Fue tomado de la página <http://diveintohtml5.info/index.html>, desarrollado por Mark Pilgrim con algunas modificaciones realizadas por el autor del presente Trabajo de titulación. A continuación una imagen y el código fuente del ejemplo.

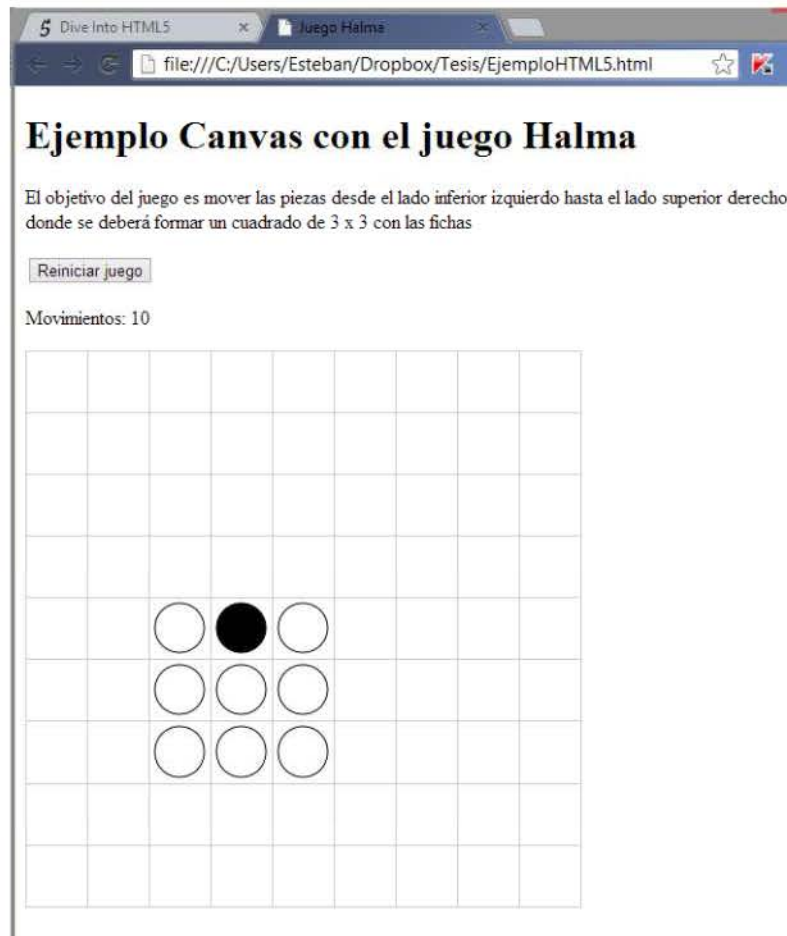


Imagen del ejemplo de HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <title>Juego Halma</title>
    <script>
      var kBoardWidth = 9;
      var kBoardHeight= 9;
      var kNumPieces = 9;
      var kPieceWidth = 50;
      var kPieceHeight= 50;
      var kPixelWidth = 1 + (kBoardWidth * kPieceWidth);
      var kPixelHeight= 1 + (kBoardHeight * kPieceHeight);

      var gCanvasElement;
```

```
var gDrawingContext;
var gPattern;

var gPieces;
var gNumPieces;
var gSelectedPieceIndex;
var gSelectedPieceHasMoved;
var gMoveCount;
var gMoveCountElem;
var gGameInProgress;

function Cell(row, column) {
    this.row = row;
    this.column = column;
}

function getCursorPosition(e) {
    /* returns Cell with .row and .column properties */
    var x;
    var y;
    if (e.pageX != undefined && e.pageY != undefined) {
        x = e.pageX;
        y = e.pageY;
    }
    else {
        x = e.clientX + document.body.scrollLeft +
            document.documentElement.scrollLeft;
        y = e.clientY + document.body.scrollTop +
            document.documentElement.scrollTop;
    }
    x -= gCanvasElement.offsetLeft;
    y -= gCanvasElement.offsetTop;
    x = Math.min(x, kBoardWidth * kPieceWidth);
    y = Math.min(y, kBoardHeight * kPieceHeight);
    var cell = new Cell(Math.floor(y/kPieceHeight),
        Math.floor(x/kPieceWidth));
    return cell;
}
```

```
function halmaOnClick(e) {
    var cell = getCursorPosition(e);
    for (var i = 0; i < gNumPieces; i++) {
        if ((gPieces[i].row == cell.row) &&
            (gPieces[i].column == cell.column)) {
            clickOnPiece(i);
            return;
        }
    }
    clickOnEmptyCell(cell);
}

function clickOnEmptyCell(cell) {
    if (gSelectedPieceIndex == -1) { return; }
    var rowDiff = Math.abs(cell.row -
        gPieces[gSelectedPieceIndex].row);
    var columnDiff = Math.abs(cell.column -
        gPieces[gSelectedPieceIndex].column);
    if ((rowDiff <= 1) &&
        (columnDiff <= 1)) {
        /* we already know that this click was on an empty
        square, so that must mean this was a valid
        single-square move */
        gPieces[gSelectedPieceIndex].row = cell.row;
        gPieces[gSelectedPieceIndex].column = cell.column;
        gMoveCount += 1;
        gSelectedPieceIndex = -1;
        gSelectedPieceHasMoved = false;
        drawBoard();
        return;
    }
    if (((rowDiff == 2) && (columnDiff == 0)) ||
        ((rowDiff == 0) && (columnDiff == 2)) ||
        ((rowDiff == 2) && (columnDiff == 2))) &&
        isThereAPieceBetween(gPieces[gSelectedPieceIndex]
        , cell)) {
        /* this was a valid jump */
        if (!gSelectedPieceHasMoved) {
            gMoveCount += 1;
        }
    }
}
```



```

    }
    gSelectedPieceHasMoved = true;
    gPieces[gSelectedPieceIndex].row = cell.row;
    gPieces[gSelectedPieceIndex].column =
        cell.column;
    drawBoard();
    return;
}
gSelectedPieceIndex = -1;
gSelectedPieceHasMoved = false;
drawBoard();
}

function clickOnPiece(pieceIndex) {
    if (gSelectedPieceIndex == pieceIndex) { return; }
    gSelectedPieceIndex = pieceIndex;
    gSelectedPieceHasMoved = false;
    drawBoard();
}

function isThereAPieceBetween(cell1, cell2) {
    /* note: assumes cell1 and cell2 are 2 squares away
       either vertically, horizontally, or diagonally */
    var rowBetween = (cell1.row + cell2.row) / 2;
    var columnBetween = (cell1.column + cell2.column) / 2;
    for (var i = 0; i < gNumPieces; i++) {
        if ((gPieces[i].row == rowBetween) &&
            (gPieces[i].column == columnBetween)) {
            return true;
        }
    }
    return false;
}

function isTheGameOver() {
    for (var i = 0; i < gNumPieces; i++) {
        if (gPieces[i].row > 2) {
            return false;
        }
    }
}

```

```
    if (gPieces[i].column < (kBoardWidth - 3)) {
        return false;
    }
}
return true;
}

function drawBoard() {
    if (gGameInProgress && isTheGameOver()) {
        endGame();
    }

    gDrawingContext.clearRect(0, 0, kPixelWidth,
                               kPixelHeight);

    gDrawingContext.beginPath();

    /* vertical lines */
    for (var x = 0; x <= kPixelWidth;
         x += kPieceWidth) {
        gDrawingContext.moveTo(0.5 + x, 0);
        gDrawingContext.lineTo(0.5 + x, kPixelHeight);
    }

    /* horizontal lines */
    for (var y = 0; y <= kPixelHeight;
         y += kPieceHeight) {
        gDrawingContext.moveTo(0, 0.5 + y);
        gDrawingContext.lineTo(kPixelWidth, 0.5 + y);
    }

    /* draw it! */
    gDrawingContext.strokeStyle = "#ccc";
    gDrawingContext.stroke();

    for (var i = 0; i < 9; i++) {
        drawPiece(gPieces[i], i == gSelectedPieceIndex);
    }
}
```

```

gMoveCountElem.innerHTML = gMoveCount;

saveGameState();
}

function drawPiece(p, selected) {
    var column = p.column;
    var row = p.row;
    var x = (column * kPieceWidth) + (kPieceWidth/2);
    var y = (row * kPieceHeight) + (kPieceHeight/2);
    var radius = (kPieceWidth/2) - (kPieceWidth/10);
    gDrawingContext.beginPath();
    gDrawingContext.arc(x, y, radius, 0, Math.PI*2,
        false);
    gDrawingContext.closePath();
    gDrawingContext.strokeStyle = "#000";
    gDrawingContext.stroke();
    if (selected) {
        gDrawingContext.fillStyle = "#000";
        gDrawingContext.fill();
    }
}

/*Esta seccion se coloca cuando no se desea usar
el almacenamientode HTML5*/

/*if (typeof resumeGame != "function") {
    saveGameState = function() {
        return false;
    }
    resumeGame = function() {
        return false;
    }
}*/

/*Las 3 siguientes funciones se encargan del
almacenamiento del juego sin embargo en la primera
se verifica si el navegador soporta esta funcionalidad
y en caso de que no lo haga, las siguientes dos

```

```

funciones devuelven el valor de falso para que se
ignore esta capacidad*/

function supportsLocalStorage() {
    return ('localStorage' in window) &&
           window['localStorage'] !== null;
}

function saveGameState() {
    if (!supportsLocalStorage()) { return false; }
    localStorage["halma.game.in.progress"] =
        gGameInProgress;
    for (var i = 0; i < kNumPieces; i++) {
        localStorage["halma.piece." + i + ".row"] =
            gPieces[i].row;
        localStorage["halma.piece." + i + ".column"] =
            gPieces[i].column;
    }
    localStorage["halma.selectedpiece"] =
        gSelectedPieceIndex;
    localStorage["halma.selectedpiecehasmoved"] =
        gSelectedPieceHasMoved;
    localStorage["halma.movecount"] = gMoveCount;
    return true;
}

function resumeGame() {
    if (!supportsLocalStorage()) { return false; }
    gGameInProgress = (localStorage
        ["halma.game.in.progress"] == "true");
    if (!gGameInProgress) { return false; }
    gPieces = new Array(kNumPieces);
    for (var i = 0; i < kNumPieces; i++) {
        var row = parseInt(localStorage
            ["halma.piece." + i + ".row"]);
        var column = parseInt(localStorage
            ["halma.piece." + i + ".column"]);
        gPieces[i] = new Cell(row, column);
    }
}

```

```
gNumPieces = kNumPieces;
gSelectedPieceIndex = parseInt(localStorage
    ["halma.selectedpiece"]);
gSelectedPieceHasMoved = localStorage
    ["halma.selectedpiecehasmoved"] == "true";
gMoveCount = parseInt(localStorage
    ["halma.movecount"]);
drawBoard();
return true;
}

function newGame() {
    gPieces = [new Cell(kBoardHeight - 3, 0),
        new Cell(kBoardHeight - 2, 0),
        new Cell(kBoardHeight - 1, 0),
        new Cell(kBoardHeight - 3, 1),
        new Cell(kBoardHeight - 2, 1),
        new Cell(kBoardHeight - 1, 1),
        new Cell(kBoardHeight - 3, 2),
        new Cell(kBoardHeight - 2, 2),
        new Cell(kBoardHeight - 1, 2)];
    gNumPieces = gPieces.length;
    gSelectedPieceIndex = -1;
    gSelectedPieceHasMoved = false;
    gMoveCount = 0;
    gGameInProgress = true;
    drawBoard();
}

function endGame() {
    gSelectedPieceIndex = -1;
    gGameInProgress = false;
}

function initGame(canvasElement, moveCountElement) {
    if (!canvasElement) {
        canvasElement = document.
            createElement("canvas");
        canvasElement.id = "halma_canvas";
    }
}
```

```

        document.body.appendChild(canvasElement);
    }
    if (!moveCountElement) {
        moveCountElement = document.createElement("p");
        moveCountElement.id = "mCount";
        document.body.appendChild(moveCountElement);
    }
    gCanvasElement = canvasElement;
    gCanvasElement.width = kPixelWidth;
    gCanvasElement.height = kPixelHeight;
    gCanvasElement.addEventListener("click",
                                    halmaOnClick, false);
    gMoveCountElem = moveCountElement;
    gDrawingContext = gCanvasElement.getContext("2d");
    if (!resumeGame()) {
        newGame();
    }
}
</script>
</head>
<body>
    <h1>Ejemplo Canvas con el juego Halma</h1>
    <p>El objetivo del juego es mover las piezas desde el
    lado inferior izquierdo hasta el lado superior derecho
    donde se deberá formar un cuadrado de 3 x 3 con las
    fichas</p>
    <button onclick="newGame()">Reiniciar juego</button>
    <p id = "moves">Movimientos: <span id="mCount">0</span></p>
    <script>
        initGame(null, document.getElementById('mCount'));
    </script>
</body>
</html>

```

Anexo 5: Ejemplo Hola Mundo en Spring 3 MVC

El siguiente ejemplo presenta la forma de acoplar a un proyecto Java el *framework* de *Spring 3 MVC*. Fue tomado del tutorial de la página <http://Viralpatel.net/blogs/spring-3-mvc-create-hello-world-application-spring-3-mvc/>, desarrollado por Viral Patel con algunas modificaciones realizadas por el autor del presente Trabajo de titulación. A continuación una imagen y el código fuente del ejemplo.



Imagen del ejemplo de Spring MVC

HelloWorldController.java

```
package net.example.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class HelloWorldController {

    @RequestMapping("/index")
    public ModelAndView index() {

        return new ModelAndView();
    }
}
```

```

@RequestMapping("/hello")
public ModelAndView helloWorld() {

    String message = "Hello World, Spring 3.2.2";

    return new ModelAndView("hello", "message", message);

}
}

```

Spring-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/
        schema/beans
        http://www.springframework.org/schema/beans/
        spring-beans-3.0.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/
        spring-context-3.0.xsd">
    <context:component-scan base-package="net.example.
        controller" />
    <bean
        id="viewResolver"
        class="org.springframework.web.servlet.view.
        UriBasedViewResolver">
        <property
            name="viewClass"
            value="org.springframework.web.servlet.view.
            JstlView" />
        <property
            name="prefix"
            value="/jsp/" />
        <property
            name="suffix"
            value=".jsp" />
    </bean>
</beans>

```


Web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID"
  version="2.5">
  <display-name>Spring3MVC</display-name>
  <welcome-file-list>
    <welcome-file>/index.html</welcome-file>
  </welcome-file-list>
  <!-- The front controller of this Spring Web application,
  responsible for handling all application requests -->
  <servlet>
    <servlet-name>spring</servlet-name>
    <servlet-class>org.springframework.web.servlet.
    DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <!-- Map all requests to the DispatcherServlet for
  handling -->
  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>*.html</url-pattern>
  </servlet-mapping>
</web-app>

```

Index.xhtml

```

<html>
  <head>
    <title>Spring 3.2.2 MVC Series: Index by
    Viral Parel</title>
  </head>
  <body>
    <script>window.location =
      "http://localhost:8080/Spring3MVC/hello.html"
    </script>
  </body>
</html>

```

Hello.jsp

```
<html>
  <head>
    <title>Spring 3.2.2 MVC Series:
    Hello World by Viral Parel</title>
  </head>
  <body>
    ${message}
  </body>
</html>
```

Index.jsp

```
<html>
  <head>
    <title>Spring 3.2.2 MVC Series: Index
    by Viral Parel</title>
  </head>
  <body>
    <a href="./hello.html">Say Hello</a>
  </body>
</html>
```

Anexo 6: Ejemplo usando PrimeFaces

El siguiente ejemplo presenta la forma de acoplar a un proyecto Java el *framework* de *PrimeFaces*. Fue tomado del tutorial de la página <http://java.dzone.com/articles/primefaces-quickstart-tutorial>, desarrollado por Sivaprasadreddy Katamreddy, con algunas modificaciones realizadas por el autor del presente Trabajo de titulación. A continuación una imagen y el código fuente del ejemplo.

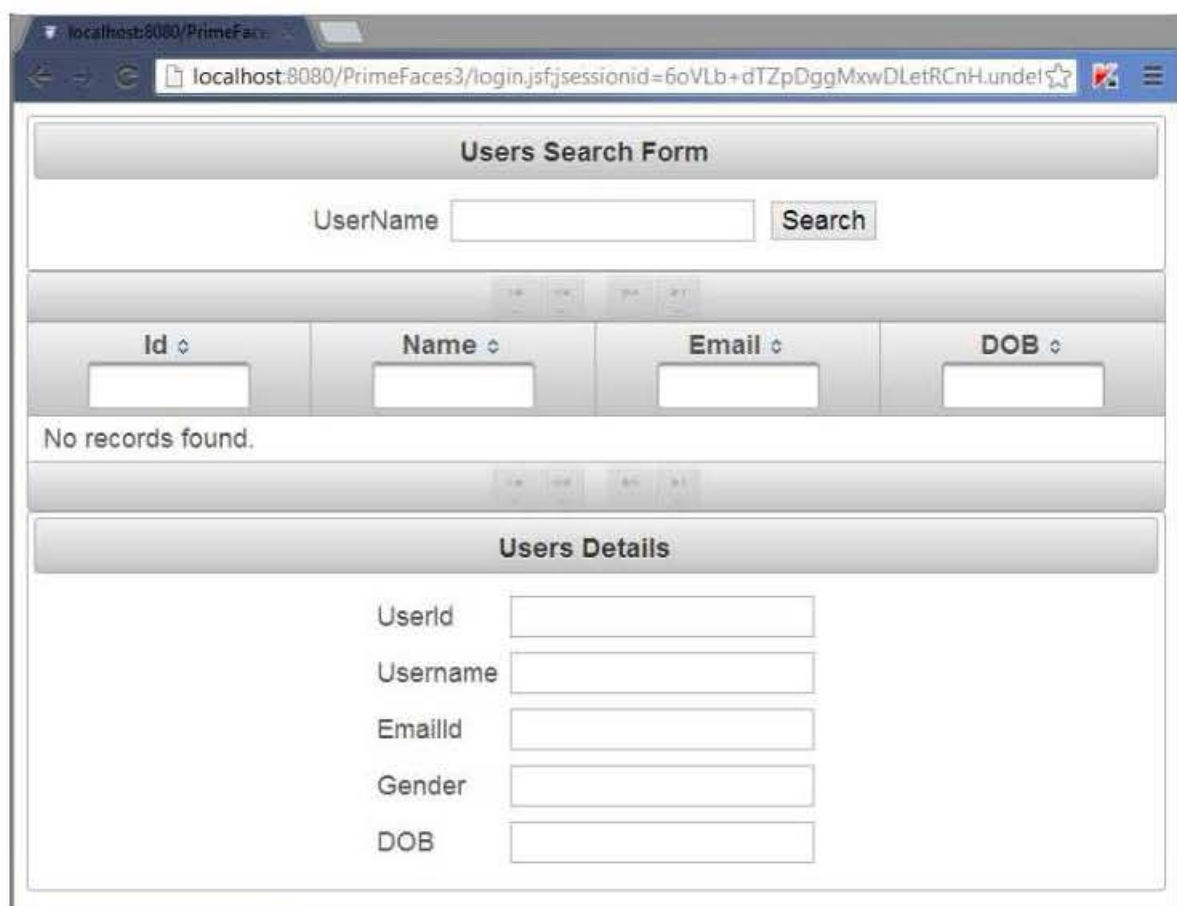


Imagen del ejemplo de PrimeFaces

User.java

```
package net.primefaces.example;

import java.util.Date;

public class User {
    private Integer userId;
    private String username;
    private String emailId;
    private String phone;
    private Date dob;
    private String gender;
    private String address;

    public User() {
    }

    public User(Integer userId, String username,
                String emailId, String phone, Date dob,
                String gender, String address) {
        this.userId = userId;
        this.username = username;
        this.emailId = emailId;
        this.phone = phone;
        this.dob = dob;
        this.gender = gender;
        this.address = address;
    }

    // setter and getters
    public Integer getUserId() {
        return userId;
    }

    public void setUserId(Integer userId) {
        this.userId = userId;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getEmailId() {
        return emailId;
    }
}
```

```
public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}

public Date getDob() {
    return dob;
}

public void setDob(Date dob) {
    this.dob = dob;
}

public String getGender() {
    return gender;
}

public void setGender(String gender) {
    this.gender = gender;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}
}
```

UserManagedBean.java

```
package net.primefaces.example;

import java.util.Collection;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ApplicationScoped;
import javax.faces.bean.ManagedBean;
import javax.faces.context.FacesContext;

import org.primefaces.event.SelectEvent;
import org.primefaces.event.UnselectEvent;

@ManagedBean
@ApplicationScoped
public class UserManagedBean {

    UserService userService = new UserService();

    private String username;
    private String password;
    private String searchUser;
    private Collection<User> searchUsersResults;
    private User selectedUser;

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

```
public User getSelectedUser() {
    if (selectedUser == null) {
        selectedUser = new User();
    }
    return selectedUser;
}

public void setSelectedUser(User selectedUser) {
    this.selectedUser = selectedUser;
}

public Collection<User> getSearchUsersResults() {
    return searchUsersResults;
}

public void setSearchUsersResults(Collection<User>
searchUsersResults) {
    this.searchUsersResults = searchUsersResults;
}

public String getSearchUser() {
    return searchUser;
}

public void setSearchUser(String searchUser) {
    this.searchUser = searchUser;
}

public String login() {
    if ("test".equalsIgnoreCase(getUsername())
        && "test".equals(getPassword())) {
        return "home";
    } else {
        FacesContext context = FacesContext.getCurrentInstance();
        context.addMessage("username", new FacesMessage(
            "Invalid UserName and Password"));
        return "login";
    }
}
```

```

public String searchUser() {
    String username = (this.searchUser == null) ? ""
        : this.searchUser
            .trim();
    this.searchUsersResults = userService.
        searchUsers(username);
    System.out.println(searchUsersResults);
    return "home";
}

public String updateUser() {
    userService.update(this.selectedUser);
    return "home";
}

public void onUserSelect(SelectEvent event) {
}

public void onUserUnselect(UnselectEvent event) {
}
}

```

UserService.java

```

package net.primefaces.example;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class UserService {

    private static final Map<Integer, User> USERS_TABLE =
        new HashMap<Integer, User>();

    static {
        USERS_TABLE.put(1, new User(1, "Administrator",
            "admin@gmail.com", "9000510456", new Date(),
            "M", "Hyderabad"));
        USERS_TABLE.put(2, new User(2, "Guest", "guest@gmail.com",
            "9247469543", new Date(), "M", "Hyderabad"));
    }
}

```



```
USERS_TABLE.put(3, new User(3, "John", "John@gmail.com",
    "9000510456", new Date(), "M", "Hyderabad"));
USERS_TABLE.put(4, new User(4, "Paul", "Paul@gmail.com",
    "9247469543", new Date(), "M", "Hyderabad"));
USERS_TABLE.put(5, new User(5, "raju", "raju@gmail.com",
    "9000510456", new Date(), "M", "Hyderabad"));
USERS_TABLE.put(6, new User(6, "raghav",
    "raghav@gmail.com", "9247469543", new Date(),
    "M", "Hyderabad"));

}

public Integer create(User user) {
    if (user == null) {
        throw new RuntimeException(
            "Unable to create User. User object is null.");
    }
    Integer userId = this.getMaxUserId();
    user.setUserId(userId);
    USERS_TABLE.put(userId, user);
    return userId;
}

public void delete(User user) {
    if (user == null) {
        throw new RuntimeException(
            "Unable to delete User. User object is null.");
    }
    USERS_TABLE.remove(user.getUserId());
}

public Collection<User> getAllUsers() {
    return USERS_TABLE.values();
}

public User getUser(Integer userId) {
    return USERS_TABLE.get(userId);
}
}
```

```
public Collection<User> searchUsers(String username) {
    String searchCriteria = (username == null) ? "" : username
        .toLowerCase().trim();
    Collection<User> users = USERS_TABLE.values();
    Collection<User> searchResults = new ArrayList<User>();
    for (User user : users) {
        if (user.getUsername() != null
            && user.getUsername().toLowerCase().trim()
                .startsWith(searchCriteria)) {
            searchResults.add(user);
        }
    }
    return searchResults;
}

public void update(User user) {
    if (user == null || !USERS_TABLE.containsKey(user.
        getUserId())) {
        throw new RuntimeException(
            "Unable to update User. User object is null " +
            "or User Id ["+ user.getUserId() + "] " +
            "is invalid.");
    }
    USERS_TABLE.put(user.getUserId(), user);
}

protected Integer getMaxUserId() {
    Set<Integer> keys = USERS_TABLE.keySet();
    Integer maxId = 1;
    for (Integer key : keys) {
        if (key > maxId) {
            maxId = key;
        }
    }
    return maxId;
}
}
```

Web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID"
  version="3.0">
  <display-name>PrimeFaces3</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.jsf</url-pattern>
  </servlet-mapping>
</web-app>

```

Faces-config.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<faces-config
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-facesconfig_2_1.xsd"
  version="2.1">
</faces-config>

```

Home.xhtml

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui">
<h:head>
  <link type="text/css" rel="stylesheet"
        href="themes/bluesky/skin.css" />
</h:head>
<h:body>
<center>
  <h:form>
    <p:panel header="Users Search Form" style="width: 700;">
      <h:form>
        <h:panelGrid columns="3" cellpadding="2">
          <h:outputLabel for="#{userManagedBean.
            searchUser}" value="UserName"/>
          <h:inputText value="#{userManagedBean.
            searchUser}" label="UserName"></h:inputText>
          <h:commandButton type="submit"
            value="Search" action="#{userManagedBean.
            searchUser}"></h:commandButton>
        </h:panelGrid>
      </h:form>
    </p:panel>

    <p:dataTable var="user"
      value="#{userManagedBean.searchUsersResults}"
      selection="#{userManagedBean.selectedUser}"
      selectionMode="single"
      rowKey="#{user.userId}"
      dynamic="true"
      onRowSelectUpdate="userUpdateForm"
      onRowUnselectUpdate="userUpdateForm"
      rowSelectListener="#{userManagedBean.onUserSelect}"
      rowUnselectListener="#{userManagedBean.onUserUnselect}"
      paginator="true" rows="5" style="width: 700">
      <p:column sortBy="#{user.userId}"
        filterBy="#{user.userId}">

```

```

        <f:facet name="header">
        <h:outputText value="Id" />
        </f:facet>
        <h:outputText value="#{user.userId}" />
        </p:column>
        <p:column sortBy="#{user.username}"
        filterBy="#{user.username}">
        <f:facet name="header">
        <h:outputText value="Name" />
        </f:facet>
        <h:outputText value="#{user.username}" />
        </p:column>
        <p:column sortBy="#{user.emailId}"
        filterBy="#{user.emailId}">
        <f:facet name="header">
        <h:outputText value="Email" />
        </f:facet>
        <h:outputText value="#{user.emailId}" />
        </p:column>
        <p:column parser="date" sortBy="#{user.dob}"
        filterBy="#{user.dob}">
        <f:facet name="header">
        <h:outputText value="DOB" />
        </f:facet>
        <h:outputText value="#{user.dob}" >
            <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:outputText>
        </p:column>
    </p:dataTable>
    <p:panel id="userDetailsPanelId" header="Users Details"
    style="width: 700;">
    <h:panelGrid columns="2" cellpadding="2"
    id="userUpdateForm" border="0" >
    <h:outputLabel for="#{userManagedBean.
    selectedUser.userId}" value="UserId"/>
    <h:inputText value="#{userManagedBean.
    selectedUser.userId}" style="width: 100;"
    readonly="true"></h:inputText>

```

```

        <h:outputLabel for="#{userManagedBean.
            selectedUser.username}" value="Username"/>
        <h:inputText value="#{userManagedBean.
            selectedUser.username}"
            readonly="true"></h:inputText>

        <h:outputLabel for="#{userManagedBean.
            selectedUser.emailId}" value="EmailId"/>
        <h:inputText value="#{userManagedBean.
            selectedUser.emailId}"
            readonly="true"></h:inputText>

        <h:outputLabel for="#{userManagedBean.
            selectedUser.gender}" value="Gender"/>
        <h:inputText value="#{userManagedBean.
            selectedUser.gender}"
            readonly="true"></h:inputText>

        <h:outputLabel for="#{userManagedBean.
            selectedUser.dob}" value="DOB"/>
        <h:inputText value="#{userManagedBean.
            selectedUser.dob}" readonly="true">
            <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:inputText>

        </h:panelGrid>
    </p:panel>
</h:form>
</center>
</h:body>
</html>

```

Login.xhtml

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:p="http://primefaces.org/ui">
<h:head>
    <link
        type="text/css"
        rel="stylesheet"
        href="themes/bluesky/skin.css" />

```

```

</h:head>
<h:body>
  <center>
    <p:panel
      header="Login Form"
      style="width: 350;">
      <h:form>
        <h:panelGrid
          columns="2"
          cellpadding="2">
          <h:outputLabel
            for="#{userManagedBean.username}"
            value="UserName" />
          <h:inputText
            value="#{userManagedBean.username}"
            label="UserName"></h:inputText>
          <h:outputLabel
            for="#{userManagedBean.password}"
            value="Password" />
          <h:inputSecret value="#{userManagedBean.
password}"></h:inputSecret>
          <h:commandButton
            type="submit"
            value="Login"
            action="#{userManagedBean.login()}">
          </h:commandButton>
        </h:panelGrid>
      </h:form>
    </p:panel>
    <div>
      <h:messages></h:messages>
    </div>
  </center>
</h:body>
</html>

```

Index.jsp

```
<jsp:forward page="login.jsf"></jsp:forward>
```

Anexo 7: Ejemplo usando Spring MVC, PrimeFaces y Base de datos

El siguiente ejemplo presenta la forma de utilizar de manera conjunta los *frameworks* de *PrimeFaces* y *Spring MVC* además de presentar persistencia en una base de datos (En este caso Postgresql). Fue tomado del tutorial de la página <http://www.javacodegeeks.com/2012/04/jsf-2-primefaces-3-spring-3-hibernate-4.html>, desarrollado por Eren Avsarogullari, con algunas modificaciones realizadas por el autor del presente Trabajo de titulación. A continuación una imagen y el código fuente del ejemplo.



The image shows a web browser window with the address bar displaying 'localhost:8080/FullExample/'. The main content area features a form titled 'Add User'. The form contains three text input fields: 'Id' with the value '1715920490', 'Name' with the value 'Esteban', and 'Last Name' with the value 'Capello'. Below these fields are two buttons: a blue 'Add user' button and a grey 'Reset' button.

Imagen del ejemplo de Spring MVC, PrimeFaces y Base de datos

IUserDAO.java

```
package net.example.dao;

import java.util.List;

import net.example.model.User;

public interface IUserDAO {

    public void addUser(User user);

    public void updateUser(User user);

    public void deleteUser(User user);

    public User getUserById(int id);

    public List<User> getUsers();

}
```

UserDAO.java

```
package net.example.dao;

import java.util.List;
import net.example.model.User;
import org.hibernate.SessionFactory;

public class UserDAO implements IUserDAO {

    private SessionFactory sessionFactory;

    @Override
    public void addUser(User user) {
        sessionFactory.getCurrentSession().save(user);
    }

    @Override
    public void updateUser(User user) {
        sessionFactory.getCurrentSession().update(user);
    }

}
```

```

@Override
public void deleteUser(User user) {
    getSessionFactory().getCurrentSession().delete(user);
}

@SuppressWarnings("rawtypes")
@Override
public User getUserById(int id) {
    List list = getSessionFactory().getCurrentSession()
        .createQuery("from User where id = ?")
        .setParameter(0, id).list();
    return (User) list.get(0);
}

@SuppressWarnings({ "unchecked", "rawtypes" })
@Override
public List<User> getUsers() {
    List list = getSessionFactory().getCurrentSession()
        .createQuery("from User").list();
    return list;
}

public SessionFactory getSessionFactory() {
    return sessionFactory;
}

public void setSessionFactory(SessionFactory sessionFactory) {
    this.sessionFactory = sessionFactory;
}
}

```

UserManagedBean.java

```

package net.example.managed.bean;

import java.io.Serializable;
import java.util.List;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.ManagedProperty;
import javax.faces.bean.RequestScoped;

import net.example.model.User;
import net.example.services.IUserService;

```

```
import org.springframework.dao.DataAccessException;

@ManagedBean(name = "userMB")
@RequestScoped
public class UserManagedBean implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID =
        -6071667423110114801L;
    private static final String SUCCESS = "success";
    private static final String ERROR = "error";

    @ManagedProperty(value = "#{UserService}")
    IUserService userService;

    List<User> userList;

    private int id;
    private String name;
    private String lastname;

    public String addUser() {
        try {
            User user = new User();
            user.setId(getId());
            user.setName(getName());
            user.setLastname(getLastname());
            getUserService().addUser(user);
            setUserList(getUserService().getUsers());
            return SUCCESS;
        } catch (DataAccessException e) {
            e.printStackTrace();
        }
        return ERROR;
    }

    public void reset() {
        this.setId(0);
        this.setName("");
        this.setLastname("");
    }
}
```

```
public IUserService getUserService() {
    return userService;
}

public void setUserService(IUserService userService) {
    this.userService = userService;
}

public List<User> getUserList() {
    return userList;
}

public void setUserList(List<User> userList) {
    this.userList = userList;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getLastname() {
    return lastname;
}

public void setLastname(String lastname) {
    this.lastname = lastname;
}
}
```

User.java

```
package net.example.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

import org.hibernate.annotations.GenericGenerator;

@Entity
@Table(name = "users")
public class User {

    private int id;
    private String name;
    private String lastname;

    @Id
    @GenericGenerator(name = "generator", strategy = "increment")
    @GeneratedValue(generator = "generator")
    @Column(name = "id", unique = true, nullable = false)
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Column(name = "name", nullable = false)
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Column(name = "lastname", nullable = false)
    public String getLastName() {
        return lastname;
    }
}
```

```
public void setLastname(String lastname) {
    this.lastname = lastname;
}

@Override
public String toString() {
    return "User [id=" + id + ", name=" +
name + ", lastname=" + lastname
        + "]";
}
}
```

IUserService.java

```
package net.example.services;

import java.util.List;
import net.example.model.User;

public interface IUserService {

    public void addUser(User user);

    public void updateUser(User user);

    public void deleteUser(User user);

    public User getUserId(int id);

    public List<User> getUsers();
}
```

UserService.java

```
package net.example.services;

import java.util.List;
import net.example.dao.IUserDAO;
import net.example.model.User;
import org.springframework.transaction.annotation.Transactional;

@Transactional(readonly = true)
public class UserService implements IUserService {

    IUserDAO userDAO;

    @Transactional(readonly = false)
    @Override
    public void addUser(User user) {
        getUserDAO().addUser(user);
    }

    @Transactional(readonly = false)
    @Override
    public void updateUser(User user) {
        getUserDAO().updateUser(user);
    }

    @Transactional(readonly = false)
    @Override
    public void deleteUser(User user) {
        getUserDAO().deleteUser(user);
    }

    @Override
    public User getUserId(int id) {
        return getUserDAO().getUserById(id);
    }

    @Override
    public List<User> getUsers() {
        return getUserDAO().getUsers();
    }

    public IUserDAO getUserDAO() {
        return userDAO;
    }

    public void setUserDAO(IUserDAO userDAO) {
        this.userDAO = userDAO;
    }
}
```

Faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-facesconfig_2_1.xsd"
  version="2.1">
  <application>
    <el-resolver>org.springframework.web.jsf.
      el.SpringBeanFacesELResolver</el-resolver>
  </application>
  <navigation-rule>
    <from-view-id>/index.xhtml</from-view-id>
    <navigation-case>
      <from-outcome>success</from-outcome>
      <to-view-id>/pages/success.xhtml</to-view-id>
    </navigation-case>
    <navigation-case>
      <from-outcome>error</from-outcome>
      <to-view-id>/pages/error.xhtml</to-view-id>
    </navigation-case>
  </navigation-rule>
</faces-config>
```

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID"
  version="3.0">
```



```

<display-name>FullExample</display-name>
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/applicationContext.xml
  </param-value>
</context-param>
<listener>
  <listener-class>org.springframework.web.
    context.ContextLoaderListener</listener-class>
</listener>
<listener>
  <listener-class>org.springframework.web.
    context.request.RequestContextListener</listener-class>
</listener>
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Development</param-value>
</context-param>
<welcome-file-list>
  <welcome-file>index.xhtml</welcome-file>
</welcome-file-list>
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.
    FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
</web-app>

```

applicationContext.xml

```

<beans
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/
    spring-context-3.0.xsd">

```

```
<bean
  id="User"
  class="net.example.model.User" />
<bean
  id="UserService"
  class="net.example.services.UserService">
  <property
    name="userDAO"
    ref="UserDAO" />
</bean>
<bean
  id="UserDAO"
  class="net.example.dao.UserDAO">
  <property
    name="sessionFactory"
    ref="SessionFactory" />
</bean>
<bean
  id="DataSource"
  class="com.mchange.v2.c3p0.ComboPooledDataSource"
  destroy-method="close">
  <property
    name="driverClass"
    value="org.postgresql.Driver" />
  <property
    name="jdbcUrl"
    value="jdbc:postgresql://localhost:5432/example" />
  <property
    name="user"
    value="postgres" />
  <property
    name="password"
    value="postgres" />
  <property
    name="maxPoolSize"
    value="10" />
  <property
    name="maxStatements"
    value="0" />
  <property
    name="minPoolSize"
    value="5" />
</bean>
```

```

<bean
  id="SessionFactory"
  class="org.springframework.orm.hibernate4.
  LocalSessionFactoryBean">
  <property
    name="dataSource"
    ref="DataSource" />
  <property name="annotatedClasses">
    <list>
      <value>net.example.model.User</value>
    </list>
  </property>
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">org.hibernate.
      dialect.PostgreSQLDialect</prop>
      <prop key="hibernate.show_sql">true</prop>
    </props>
  </property>
</bean>
<tx:annotation-driven transaction-manager="txManager" />
<bean
  id="txManager"
  class="org.springframework.orm.hibernate4.
  HibernateTransactionManager">
  <property
    name="sessionFactory"
    ref="SessionFactory" />
</bean>
</beans>

```

Index.xhtml

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui">
<h:head>
  <title>Welcome to a example of Spring hibernate Primefaces
  Project</title>
</h:head>
<h:body>
  <h:form>
    <center>

```

```
<p:panel
  header="Add User"
  style="width:350;">
  <h:panelGrid columns="3">
    <h:outputLabel
      for="id"
      value="Id: " />
    <p:inputText
      id="id"
      value="#{userMB.id}"
      <f:converter converterId="javax.
        faces.Integer" />
      <p:ajax
        event="blur"
        update="idMsg" />
    </p:inputText>
    <p:message
      id="idMsg"
      for="id"
      display="icon" />
    <h:outputLabel
      for="name"
      value="Name: " />
    <p:inputText
      id="name"
      value="#{userMB.name}"
      <f:validateLength minimum="5" />
      <p:ajax
        event="blur"
        update="nameMsg" />
    </p:inputText>
    <p:message
      id="nameMsg"
      for="name"
      display="icon" />
    <h:outputLabel
      for="lastname"
      value="Last Name: " />
    <p:inputText
      id="lastname"
      value="#{userMB.lastname}"
      <f:validateLength minimum="5" />
      <p:ajax
        event="blur"
        update="lastnameMsg" />
    </p:inputText>
```

```

        <p:message
            id="lastnameMsg"
            for="lastname"
            display="icon" />
        <p:commandButton
            value="Add user"
            action="#{userMB.addUser()}"
            ajax="false"/>
        <p:commandButton
            value="Reset"
            action="#{userMB.reset()}"
            ajax="false"/>
    </h:panelGrid>
</p:panel>
</center>
</h:form>
</h:body>
</html>

```

Success.xhtml

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui">
<h:head>
    <title>Added Successfully</title>
</h:head>
<h:body>
    <h:form>
        <center>
            <p:panel header="Users List">
                <p:dataTable
                    id="users"
                    var="user"
                    value="#{userMB.userList}"
                    style="width: 100%">
                    <p:column>
                        <f:facet name="header">
                            <h:outputText value="ID" />
                        </f:facet>
                        <h:outputText value="#{user.id}" />
                    </p:column>

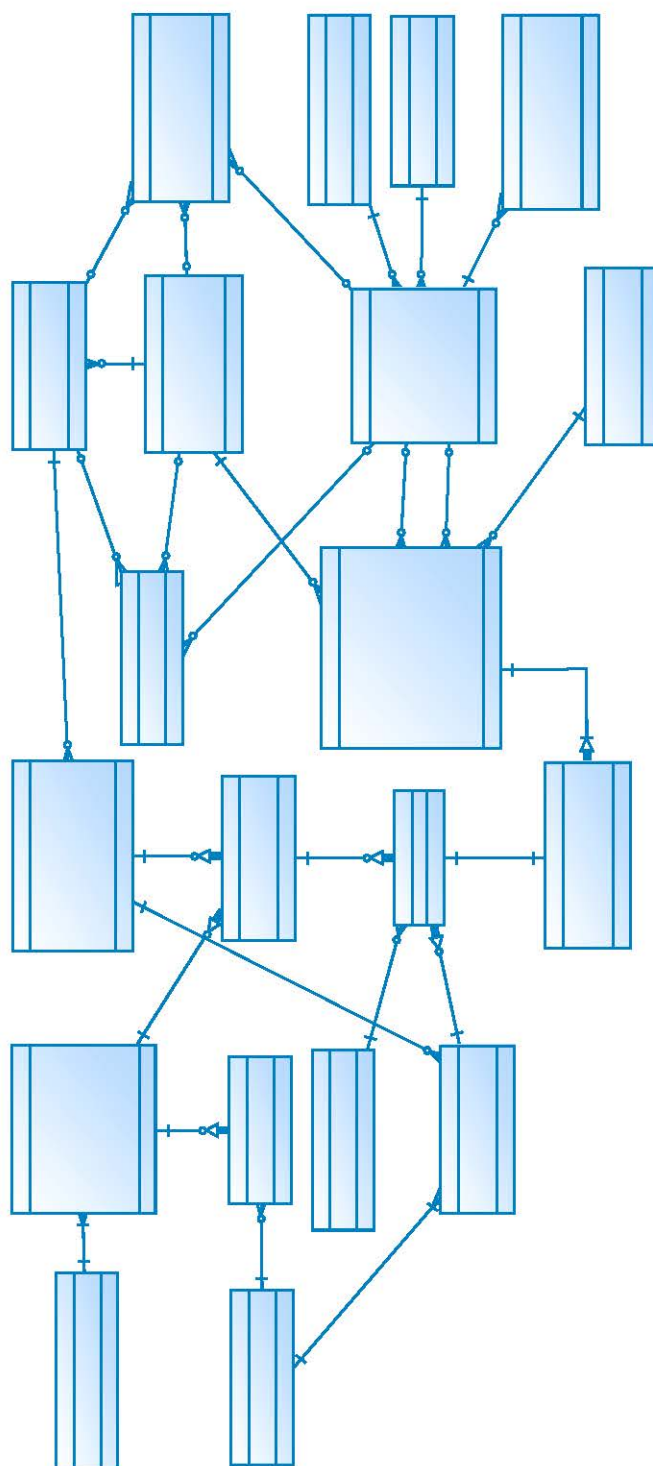
```

```
<p:column>
  <f:facet name="header">
    <h:outputText value="Name" />
  </f:facet>
  <h:outputText value="#{user.name}" />
</p:column>
<p:column>
  <f:facet name="header">
    <h:outputText value="Last Name" />
  </f:facet>
  <h:outputText value="#{user.lastname}" />
</p:column>
</p:dataTable>
<p:spacer height="10" />
<p:commandButton
  value="Return"
  action="../index.xhtml"
  ajax="false" />
</p:panel>
</center>
</h:form>
</h:body>
</html>
```

Error.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui">
<h:head>
  <title>Error!!!</title>
</h:head>
<h:body>
  <h:form>
    <f:view>
      <h:outputText value="Transaction error has occurred!" />
      <p:commandButton
        value="Return"
        action="../../index.xhtml"
        ajax="false" />
    </f:view>
  </h:form>
</h:body>
</html>
```

Anexo 8: Modelo Conceptual de la base de datos



Anexo 9: Manual de instalación

Introducción

El presente documento se presenta en forma de un manual con los pasos necesarios para poder instalar y comprobar el funcionamiento del prototipo de sistema de reservación y venta de entradas al cine. Explicando sus prerequisites de funcionamiento así como los lugares donde se deben colocar los archivos.

Prerrequisitos

De Software

Para la ejecución del sistema es necesario tener instalados las siguientes aplicaciones:

- Java JDK versión 1.6 release 24 como mínimo (Advertencia: no se ha probado el funcionamiento de la aplicación con la versión 1.7 del JDK de Java)
- Motor de base de datos PostgreSQL versión 9.2
- Servidor de aplicaciones JBoss 7.1

De Hardware

Los siguientes componentes se los plantea como los mínimos para una correcta ejecución del sistema:

- Procesador: Intel Core Duo
- Memoria RAM: 2GB
- Espacio en Disco duro de 200 MB

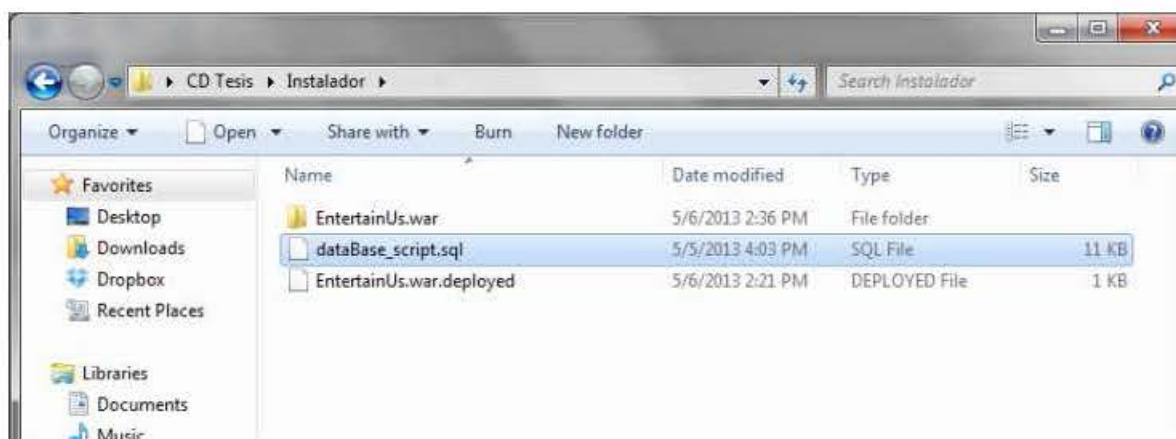
Para cualquier condicionante de las aplicaciones previamente colocadas en esta sección a nivel de Hardware, por favor remitirse a los manuales de instalación de las mismas

Situación

Para comenzar con la instalación se toma en cuenta que los prerequisites tanto de software como de hardware se han cumplido.

Proceso de Instalación

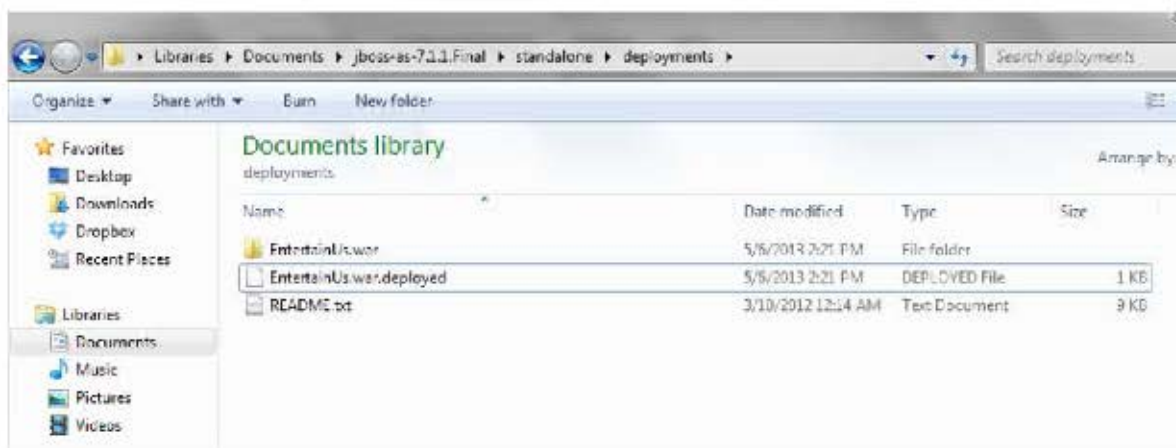
1. Abrir el CD provisto junto con este manual y dirigirse a la carpeta: "UnidadCD:\CD Tesis\Instalador" y ejecutar el script SQL de nombre "dataBase_script.sql" en el motor de base de datos PostgreSQL.



Ruta del instalador en el CD

2. Abrir el directorio donde se encuentra instalado el servidor de aplicaciones JBoss y dirigirse a la ruta: "{JBoss_Dir}\jboss-as-7.1.1.Final\standalone\deployments" y copiar los archivos:
 - "EntertainUs.war"
 - "EntertainUs.war.deployed"

Que fueron provisto en el CD, en la misma ruta del script de la base datos.



Ruta de instalación de la aplicación en Jboss 7.1

3. Abrir una consola de comandos dirigirse al directorio de instalación de Jboss, una vez ahí digitar el siguiente comando para que el servidor Inicie junto con la aplicación: `"$JBASS_HOME/bin/standalone.bat -b=0.0.0.0"`

```

C:\Windows\system32\cmd.exe - standalone.bat -b=0.0.0.0

C:\Users\Esteban\Documents>cd jboss-as-7.1.1.Final
C:\Users\Esteban\Documents\jboss-as-7.1.1.Final>cd bin
C:\Users\Esteban\Documents\jboss-as-7.1.1.Final\bin>standalone.bat -b=0.0.0.0
Calling "C:\Users\Esteban\Documents\jboss-as-7.1.1.Final\bin\standalone.conf.bat"
=====
JBoss Bootstrap Environment

JBASS_HOME: C:\Users\Esteban\Documents\jboss-as-7.1.1.Final
JAVA: C:\Program Files (x86)\Java\jdk1.6.0_41\bin\java

JAVA_OPTS: -XX:+TieredCompilation -Dprogram.name=standalone.bat -Xms64M -Xmx512M -XX:MaxPermSize=256M -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djboss.server.default.config=standalone.xml
=====

```

Ejecución de Jboss 7.1

4. Finalmente abrir un Explorar de internet y digitar la siguiente dirección para abrir la aplicación: <http://localhost:8080/EntertainUs/>



Anexo 10: Product Backlog

ID Historia	Nombre de la Historia	Fecha de Culminación	Tamaño (días)	Sprint	Comentarios
1	Planteamiento de ideas para el capítulo 1	6/8/2012	1	1	
2	Escritura de antecedentes, justificación y objetivos	7/8/2012	0,5	1	
3	Planeación de alcance y áreas de aplicación	7/8/2012	0,5	1	
4	Escritura de alcance y áreas de aplicación	8/8/2012	1	1	
5	Escritura de la introducción	9/8/2012	1	1	
6	Revisión del capítulo 1	10/8/2012	1	1	
7	Correcciones al capítulo 1	12/8/2012	1,5	1	
8	Planteamiento de ideas para el capítulo 2	13/8/2012	1	2	
9	Investigación de metodologías de desarrollo de software tradicionales	13/8/2012	0,5	2	
10	Escritura de lo investigado sobre metodologías de desarrollo tradicionales	14/8/2012	0,5	2	
11	Investigación de metodologías de desarrollo de software ágiles	15/8/2012	1	2	
12	Escritura de lo investigado sobre metodologías de desarrollo ágiles	15/8/2012	0,5	2	
13	Desarrollo de un cuadro comparativo sobre los tipos de metodologías	16/8/2012	1	2	
14	Investigación sobre Scrum	18/8/2012	1,5	2	
15	Escritura de metodología Scrum (Partes y análisis de ventajas y desventajas)	20/8/2012	2	3	
16	Investigación de metodologías XP y Cristal Clear	23/8/2012	3	3	
17	Desarrollo de un cuadro comparativo sobre metodologías ágiles	24/8/2012	1	3	

18	Planteamiento de ideas de herramientas de apoyo para la elaboración del prototipo	26/8/2012	2	4	
19	Investigación de Wireframes	27/8/2012	1	4	
20	Escritura de WireFrames	27/8/2012	0,5	4	
21	Investigación herramientas UML	29/8/2012	2	4	
22	Escritura Diagrama de Base de datos	30/8/2012	0,5	4	
23	Escritura Diagrama de Actividades	N/A	0	4	Se remueven por falta de tiempo para elaboración e implementación de los diagramas
24	Escritura Diagrama de Secuencia	N/A	0	4	Se remueven por falta de tiempo para elaboración e implementación de los diagramas
25	Escritura Diagrama de Clases	30/8/2012	0,5	4	
26	Escritura Diagrama de Casos de Uso	31/8/2012	0,5	4	
27	Revisión del capítulo 2	1/9/2012	1	5	
28	Correcciones capítulo 2	3/9/2012	2	5	
29	Investigación normas APA	4/9/2012	1	5	
30	Cambio de capítulos 1 y 2 para que cumplan con normas APA	6/9/2012	2,5	5	
31	Investigación de HTML5	8/9/2012	2	6	
32	Desarrollo de Ejemplo de HTML5	10/9/2012	1,5	6	
33	Escritura de HTML5	11/9/2012	1	6	Se deben escoger las características de HTML5 que se van a utilizar en el prototipo
34	Investigación JavaScript	11/9/2012	0,5	6	
35	Desarrollo de Ejemplo de JavaScript	N/A	0	6	Se lo remueve debido a que en el ejemplo de HTML5 ya se encuentra el uso de JavaScript
36	Escritura de JavaScript	12/9/2012	0,5	6	
37	Investigación de Jboss AS	13/9/2012	1	6	

38	Escritura de sección para Jboss AS	13/9/2012	0,5	7	No se realizará ningún ejemplo de utilización ni instalación de Jboss
39	Investigación de Spring 3 MVC	15/9/2012	2	7	Se decide cambiar el uso de solo el MVC de spring por varios componentes de Spring Framework
40	Investigación Spring 3 Framework	17/9/2012	2	7	
41	Desarrollo de Ejemplo de Spring 3 MVC	19/9/2012	1,5	7	
42	Escritura sección Spring 3 Framework	20/9/2012	1,5	8	
43	Investigación de PrimeFaces	21/9/2012	1	8	
44	Desarrollo de Ejemplo de PrimeFaces	23/9/2012	1,5	8	
45	Escritura de sección PrimeFaces	23/9/2012	0,5	8	
46	Investigación de Hibernate	24/9/2012	1	8	
47	Escritura de sección Hibernate	25/9/2012	0,5	8	
48	Revisión capítulo 3	26/9/2012	1,5	9	
49	Correcciones capítulo 3	28/9/2012	2	9	
50	Planificación de roles para el prototipo	30/9/2012	1,5	9	
51	Planificación de WireFrames	3/10/2012	3	9	
52	Desarrollo de WireFrames 1	8/10/2012	5	10	
53	Desarrollo de WireFrames 2	13/10/2012	5	11	
54	Desarrollo de WireFrames 3	18/10/2012	5	12	
55	Revisión de WireFrames	19/10/2012	1	13	
56	Corrección de WireFrames	20/10/2012	1	13	
57	Planificación para la Base de Datos	23/10/2012	3	13	
58	Desarrollo del modelo entidad-relación de la BDD	26/10/2012	3	13	Una vez terminado se lo va a ir modificando según el desarrollo del prototipo lo exija
59	Escritura de del script para la base de datos	26/10/2012	0,5	14	
60	Planificación del Diagrama de Actividades	N/A	0	14	Se lo remueve para dar mayor prioridad al desarrollo del prototipo

61	Desarrollo del diagrama de actividades	N/A	0	14	Se lo remueve para dar mayor prioridad al desarrollo del prototipo
62	Planificación del diagrama de clases 1	31/10/2012	5	14	
63	Planificación del diagrama de clases 2	2/11/2012	2	14	
64	Desarrollo del diagrama de clases	5/11/2012	3	15	Este diagrama va a ser modificado conforme el desarrollo del prototipo lo exija, de manera de poder ir optimizando lo que ya fue planificado
65	Planificación diagramas de casos de uso 1	10/11/2012	5	15	
66	Planificación diagramas de casos de uso 2	15/11/2012	5	16	
67	Planificación diagramas de casos de uso 3	20/11/2012	5	17	
68	Planificación diagramas de casos de uso 4	25/11/2012	5	18	Tarea agregada
69	Planificación diagramas de casos de uso 5	28/11/2012	3	18	Tarea agregada
70	Planificación diagramas de casos de uso 6	30/11/2012	2	19	Tarea agregada
71	Desarrollo de diagramas de casos de uso 1	5/12/2012	5	19	
72	Desarrollo de diagramas de casos de uso 2	10/12/2012	5	20	
73	Desarrollo de diagramas de casos de uso 3	15/12/2012	5	21	
74	Desarrollo de diagramas de casos de uso 4	20/12/2012	5	22	
75	Desarrollo de diagramas de casos de uso 5	23/12/2012	3	23	
76	Desarrollo de diagramas de casos de uso 6	28/12/2012	5	23	
77	Desarrollo de diagramas de casos de uso 7	2/1/2013	5	24	Tarea agregada
78	Desarrollo de diagramas de casos de uso 8	6/1/2013	4	25	Tarea agregada
79	Desarrollo de diagramas de casos de uso 9	8/1/2013	2	25	Tarea agregada

80	Realización del proyecto en JAVA	9/1/2013	1	25	
81	Configuración del proyecto JAVA	10/1/2013	1	26	
82	Desarrollo clases de entidades para prototipo	11/1/2013	1	26	
83	Desarrollo de DAO	12/1/2013	1	26	
84	Desarrollo de Vistas con Primefaces 1	17/1/2013	5	26	
85	Desarrollo de Vistas con Primefaces 2	22/1/2013	5	27	
86	Desarrollo de Vistas con Primefaces 3	27/1/2013	5	27	
87	Desarrollo de Vistas con Primefaces 4	30/1/2013	3	27	Tarea agregada
88	Desarrollo de controladores y servicios 1	2/2/2013	3	28	
89	Desarrollo de controladores y servicios 2	5/2/2013	3	28	
90	Desarrollo de controladores y servicios 3	8/2/2013	3	29	
91	Pruebas de controladores y servicios 1,2 y 3	10/2/2013	2	30	
92	Corrección de errores	13/2/2013	3	30	
93	Corrección de errores	15/2/2013	2	31	Tarea agregada
94	Desarrollo de controladores y servicios 4	18/2/2013	3	31	
95	Desarrollo de controladores y servicios 5	21/2/2013	3	31	
96	Desarrollo de controladores y servicios 6	23/2/2013	2	32	
97	Pruebas de controladores y servicios 4,5 y6	24/2/2013	1	32	
98	Corrección de errores	27/2/2013	3	32	
99	Corrección de errores	1/3/2013	2	33	Tarea agregada
100	Desarrollo de controladores y servicios 7	4/3/2013	3	33	
101	Desarrollo de controladores y servicios 8	7/3/2013	3	33	
102	Desarrollo de controladores y servicios 9	9/3/2013	2	34	

103	Pruebas de controladores y servicios 7,8 y 9	11/3/2013	2	34	
104	Corrección de errores	14/3/2013	3	34	
105	Corrección de errores	16/3/2013	2	35	Tarea agregada
106	Corrección de errores	18/3/2013	2	35	Tarea agregada
107	Desarrollo de controladores y servicios 10	21/3/2013	3	35	
108	Desarrollo de controladores y servicios 11	24/3/2013	3	36	
109	Pruebas de controladores y servicios 10 y 11	26/3/2013	2	36	
110	Corrección de errores	29/3/2013	3	36	
111	Corrección de errores	31/3/2013	2	37	Tarea agregada
112	Corrección de errores	2/4/2013	2	37	Tarea agregada
113	Agregar secciones de HTML5 en el prototipo	4/4/2013	2	37	
114	Corrección de errores	5/4/2013	1	37	Tarea agregada
115	Pruebas generales del sistema	8/4/2013	3	38	
116	Corrección de errores	10/4/2013	2	38	
117	Estabilización del sistema	13/4/2013	3	38	
118	Estabilización del sistema	16/4/2013	3	39	Tarea agregada
119	Agregar CSS al prototipo	N/A	0	39	Se remueve por falta de tiempo, se tiene planeado agregarlo después de entregada la tesis
120	Desarrollo de análisis económico	17/4/2013	1	39	
121	Desarrollo de Conclusiones y Recomendaciones	20/4/2013	3	39	
122	Revisión general de la tesis	23/4/2013	3	40	
123	Correcciones de la Tesis	26/4/2013	3	40	

Anexo 11: Sprints

Sprint 1

Fecha Inicio	6/ago/2012
Fecha Fin	12/ago/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planteamiento de ideas para el capítulo 1	Esteban Capello	Hecho	5	3	6/8/2012	6/8/2012
Escritura de antecedentes, justificación y objetivos	Esteban Capello	Hecho	2.5	2	7/8/2012	6/8/2012
Planeación de alcance y áreas de aplicación	Esteban Capello	Hecho	2.5	2	7/8/2012	7/8/2012
Escritura de alcance y áreas de aplicación	Esteban Capello	Hecho	5	2	8/8/2012	7/8/2012
Escritura de la introducción	Esteban Capello	Hecho	5	3	9/8/2012	8/8/2012
Revisión del capítulo 1	Esteban Capello	Hecho	5	3	10/8/2012	9/8/2012
Correcciones al capítulo 1	Esteban Capello	Hecho	7.5	3	12/8/2012	9/8/2012
Planteamiento de ideas para el capítulo 2	Esteban Capello	Hecho	5	3	13/8/2012	10/8/2012
Investigación de metodologías de desarrollo de software tradicionales	Esteban Capello	Hecho	2.5	1.5	13/8/2012	11/8/2012
Escritura de lo investigado sobre metodologías de desarrollo tradicionales	Esteban Capello	Hecho	2.5	1.5	14/8/2012	11/8/2012
Investigación de metodologías de desarrollo de software ágiles	Esteban Capello	Hecho	5	2	15/8/2012	12/8/2012

Escritura de lo investigado sobre metodologías de desarrollo ágiles	Esteban Capello	Hecho	5	2	15/8/2012	12/8/2012
---	-----------------	-------	---	---	-----------	-----------

Sprint 2

Fecha Inicio	12/ago/2012
Fecha Fin	18/ago/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de un cuadro comparativo sobre los tipos de metodologías	Esteban Capello	Hecho	5	2	16/8/2012	13/8/2012
Investigación sobre Scrum	Esteban Capello	Hecho	7.5	5	18/8/2012	14/8/2012
Escritura de metodología Scrum (Partes y análisis de ventajas y desventajas)	Esteban Capello	Hecho	10	5	20/8/2012	15/8/2012
Investigación de metodologías XP y Cristal Clear	Esteban Capello	Hecho	15	12	23/8/2012	18/8/2012
Desarrollo de un cuadro comparativo sobre metodologías ágiles	Esteban Capello	Hecho	5	3	24/8/2012	18/8/2012

Sprint 3

Fecha Inicio	18/ago/2012
Fecha Fin	24/ago/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planteamiento de ideas de herramientas de apoyo para la elaboración del prototipo	Esteban Capello	Hecho	10	7	26/8/2012	20/8/2012
Investigación de Wireframes	Esteban Capello	Hecho	5	3	27/8/2012	21/8/2012
Escritura de WireFrames	Esteban Capello	Hecho	2.5	2	27/8/2012	21/8/2012
Investigación herramientas UML	Esteban Capello	Hecho	10	6	29/8/2012	22/8/2012
Escritura Diagrama de Base de datos	Esteban Capello	Hecho	2.5	2	30/8/2012	23/8/2012
Escritura Diagrama de Clases	Esteban Capello	Hecho	2.5	3	30/8/2012	24/8/2012
Escritura Diagrama de Casos de Uso	Esteban Capello	Hecho	2.5	3	31/8/2012	24/8/2012

Sprint 4

Fecha Inicio	24/ago/2012
Fecha Fin	31/ago/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Revisión del capítulo 2	Esteban Capello	Hecho	5	5	1/9/2012	29/8/2012
Correcciones capítulo 2	Esteban Capello	Hecho	10	7	3/9/2012	31/8/2012

Sprint 5

Fecha Inicio	31/ago/2012
Fecha Fin	6/sep/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Investigación normas APA	Esteban Capello	Hecho	5	10	4/9/2012	1/9/2012
Cambio de capítulos 1 y 2 para que cumplan con normas APA	Esteban Capello	Hecho	12.5	15	6/9/2012	4/9/2012
Investigación de HTML5	Esteban Capello	Hecho	10	6	8/9/2012	5/9/2012

Sprint 6

Fecha Inicio	6/sep/2012
Fecha Fin	13/sep/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de Ejemplo de HTML5	Esteban Capello	Hecho	7.5	8	10/9/2012	8/9/2012
Escritura de HTML5	Esteban Capello	Hecho	5	2	11/9/2012	9/9/2012
Investigación JavaScript	Esteban Capello	Hecho	2.5	4	11/9/2012	11/9/2012
Escritura de JavaScript	Esteban Capello	Hecho	2.5	2	12/9/2012	12/9/2012
Investigación de Jboss AS	Esteban Capello	Hecho	5	2	13/9/2012	12/9/2012
Escritura de sección para Jboss AS	Esteban Capello	Hecho	2.5	2	13/9/2012	13/9/2012

Sprint 7

Fecha Inicio	13/sep/2012
Fecha Fin	19/sep/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Investigación de Spring 3 MVC	Esteban Capello	Hecho	10	8	15/9/2012	16/9/2012
Investigación Spring 3 Framework	Esteban Capello	Hecho	10	2	17/9/2012	17/9/2012
Desarrollo de Ejemplo de Spring 3 MVC	Esteban Capello	Hecho	7.5	4	19/9/2012	19/9/2012

Sprint 8

Fecha Inicio	19/sep/2012
Fecha Fin	25/sep/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Escritura sección Spring 3 Framework	Esteban Capello	Hecho	7.5	10	20/9/2012	21/9/2012
Investigación de PrimeFaces	Esteban Capello	Hecho	5	2	21/9/2012	22/9/2012
Desarrollo de Ejemplo de PrimeFaces	Esteban Capello	Hecho	7.5	3	23/9/2012	22/9/2012
Escritura de sección PrimeFaces	Esteban Capello	Hecho	2.5	3	23/9/2012	23/9/2012
Investigación de Hibernate	Esteban Capello	Hecho	5	3	24/9/2012	23/9/2012
Escritura de sección Hibernate	Esteban Capello	Hecho	2.5	5	25/9/2012	25/9/2012

Sprint 9

Fecha Inicio	25/sep/2012
Fecha Fin	3/oct/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Revisión capítulo 3	Esteban Capello	Hecho	7.5	3	26/9/2012	26/9/2012
Correcciones capítulo 3	Esteban Capello	Hecho	10	4	28/9/2012	27/9/2012
Planificación de roles para el prototipo	Esteban Capello	Hecho	7.5	8	30/9/2012	29/9/2012
Planificación de WireFrames	Esteban Capello	Hecho	15	18	3/10/2012	3/10/2012

Sprint 10

Fecha Inicio	3/oct/2012
Fecha Fin	8/oct/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de WireFrames 1	Esteban Capello	Hecho	25	24	8/10/2012	8/10/2012

Sprint 11

Fecha Inicio	8/oct/2012
Fecha Fin	13/oct/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de WireFrames 2	Esteban Capello	Hecho	25	25	13/10/2012	13/10/2012

Sprint 12

Fecha Inicio	13/oct/2012
Fecha Fin	18/oct/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de WireFrames 3	Esteban Capello	Hecho	25	28	18/10/2012	18/10/2012

Sprint 13

Fecha Inicio	18/oct/2012
Fecha Fin	26/oct/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Revisión de WireFrames	Esteban Capello	Hecho	5	3	19/10/2012	19/10/2012
Corrección de WireFrames	Esteban Capello	Hecho	5	4	20/10/2012	20/10/2012
Planificación para la Base de Datos	Esteban Capello	Hecho	15	22	23/10/2012	25/10/2012
Desarrollo del modelo entidad-relación de la BDD	Esteban Capello	Incompleto	15	2	26/10/2012	26/10/2012

Sprint 14

Fecha Inicio	26/oct/2012
Fecha Fin	2/nov/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo del modelo entidad-relación de la BDD	Esteban Capello	Hecho	13	8	26/10/2012	28/10/2012
Escritura de del script para la base de datos	Esteban Capello	Hecho	2.5	3	26/10/2012	29/10/2012
Planificación del diagrama de clases 1	Esteban Capello	Incompleto	25	20	31/10/2012	2/11/2012
Planificación del diagrama de clases 2	Esteban Capello	Falta	10	0	2/11/2012	

Sprint 15

Fecha Inicio	2/nov/2012
Fecha Fin	10/nov/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planificación del diagrama de clases 1	Esteban Capello	Hecho	5	8	31/10/2012	4/11/2012
Planificación del diagrama de clases 2	Esteban Capello	Hecho	10	6	2/11/2012	5/11/2012
Desarrollo del diagrama de clases	Esteban Capello	Hecho	15	8	5/11/2012	7/11/2012
Planificación diagramas de casos de uso 1	Esteban Capello	Incompleto	25	18	10/11/2012	10/11/2012

Sprint 16

Fecha Inicio	10/nov/2012
Fecha Fin	15/nov/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planificación diagramas de casos de uso 1	Esteban Capello	Hecho	7	9	10/11/2012	11/11/2012
Planificación diagramas de casos de uso 2	Esteban Capello	Hecho	25	22	15/11/2012	15/11/2012

Sprint 17

Fecha Inicio	15/nov/2012
Fecha Fin	20/nov/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planificación diagramas de casos de uso 3	Esteban Capello	Hecho	25	22	20/11/2012	20/11/2012

Sprint 18

Fecha Inicio	20/nov/2012
Fecha Fin	28/nov/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planificación diagramas de casos de uso 4	Esteban Capello	Falta	25	0	25/11/2012	
Planificación diagramas de casos de uso 5	Esteban Capello	Falta	15	0	28/11/2012	

Sprint 19

Fecha Inicio	28/nov/2012
Fecha Fin	5/dic/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planificación diagramas de casos de uso 4	Esteban Capello	Hecho	25	20	25/11/2012	5/12/2012
Planificación diagramas de casos de uso 5	Esteban Capello	Hecho	15	8	28/11/2012	5/12/2012

Sprint 20

Fecha Inicio	5/dic/2012
Fecha Fin	10/dic/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Planificación diagramas de casos de uso 6	Esteban Capello	Hecho	10	8	30/11/2012	7/12/2012
Desarrollo de diagramas de casos de uso 1	Esteban Capello	Hecho	25	28	5/12/2012	10/12/2012

Sprint 21

Fecha Inicio	10/dic/2012
Fecha Fin	15/dic/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de diagramas de casos de uso 2	Esteban Capello	Hecho	25	12	10/12/2012	11/12/2012
Desarrollo de diagramas de casos de uso 3	Esteban Capello	Incompleto	25	18	15/12/2012	15/12/2012

Sprint 22

Fecha Inicio	15/dic/2012
Fecha Fin	20/dic/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de diagramas de casos de uso 3	Esteban Capello	Hecho	7	5	15/12/2012	16/12/2012
Desarrollo de diagramas de casos de uso 4	Esteban Capello	Hecho	25	20	20/12/2012	20/12/2012

Sprint 23

Fecha Inicio	20/dic/2012
Fecha Fin	28/dic/2012

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de diagramas de casos de uso 5	Esteban Capello	Hecho	15	10	23/12/2012	22/12/2012
Desarrollo de diagramas de casos de uso 6	Esteban Capello	Hecho	25	15	28/12/2012	28/12/2012

Sprint 24

Fecha Inicio	28/dic/2012
Fecha Fin	2/ene/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de diagramas de casos de uso 7	Esteban Capello	Hecho	25	10	2/1/2013	30/12/2012

Sprint 25

Fecha Inicio	2/ene/2013
Fecha Fin	9/ene/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de diagramas de casos de uso 8	Esteban Capello	Hecho	20	27	6/1/2013	6/1/2013
Desarrollo de diagramas de casos de uso 9	Esteban Capello	Hecho	10	8	8/1/2013	7/1/2013
Realización del proyecto en JAVA	Esteban Capello	Hecho	5	4	9/1/2013	8/1/2013
Configuración del proyecto JAVA	Esteban Capello	Hecho	5	8	10/1/2013	9/1/2013

Sprint 26

Fecha Inicio	9/ene/2013
Fecha Fin	17/ene/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo clases de entidades para prototipo	Esteban Capello	Hecho	5	3	11/1/2013	10/1/2013
Desarrollo de DAO	Esteban Capello	Hecho	5	5	12/1/2013	11/1/2013
Desarrollo de Vistas con Primefaces 1	Esteban Capello	Hecho	25	20	17/1/2013	16/1/13

Sprint 27

Fecha Inicio	15/feb/2013
Fecha Fin	22/feb/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de Vistas con Primefaces 2	Esteban Capello	Hecho	25	18	22/1/2013	16/2/2013
Desarrollo de Vistas con Primefaces 3	Esteban Capello	Hecho	25	18	27/1/2013	17/2/2013
Desarrollo de Vistas con Primefaces 4	Esteban Capello	Hecho	15	9	30/1/2013	18/2/2013
Desarrollo de controladores y servicios 1	Esteban Capello	Hecho	15	10	2/2/2013	18/2/2013
Desarrollo de controladores y servicios 2	Esteban Capello	Hecho	15	10	5/2/2013	19/2/2013
Desarrollo de controladores y servicios 3	Esteban Capello	Hecho	15	9	8/2/2013	19/2/2013
Pruebas de controladores y servicios 1,2 y 3	Esteban Capello	Hecho	10	12	10/2/2013	20/2/2013

Corrección de errores	Esteban Capello	Hecho	15	15	13/2/2013	21/2/2013
Corrección de errores	Esteban Capello	Hecho	10	5	15/2/2013	22/2/2013
Desarrollo de controladores y servicios 4	Esteban Capello	Hecho	15	12	18/2/2013	22/2/2013
Desarrollo de controladores y servicios 5	Esteban Capello	Falta	15		21/2/2013	

Sprint 28

Fecha Inicio	22/feb/2013
Fecha Fin	27/feb/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de controladores y servicios 5	Esteban Capello	Hecho	15	12	21/2/2013	23/2/2013
Desarrollo de controladores y servicios 6	Esteban Capello	Hecho	10	12	23/2/2013	24/2/2013
Pruebas de controladores y servicios 4,5 y6	Esteban Capello	Hecho	5	5	24/2/2013	24/2/2013
Corrección de errores	Esteban Capello	Hecho	15	15	27/2/2013	27/2/2013

Sprint 29

Fecha Inicio	27/feb/2013
Fecha Fin	7/mar/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Corrección de errores	Esteban Capello	Hecho	10	5	1/3/2013	28/2/2013
Desarrollo de controladores y servicios 7	Esteban Capello	Hecho	15	18	4/3/2013	4/3/2013
Desarrollo de controladores y servicios 8	Esteban Capello	Hecho	15	16	7/3/2013	7/3/2013

Sprint 30

Fecha Inicio	7/mar/2013
Fecha Fin	14/mar/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de controladores y servicios 9	Esteban Capello	Hecho	10	9	9/3/2013	9/3/2013
Pruebas de controladores y servicios 7,8 y 9	Esteban Capello	Hecho	10	10	11/3/2013	11/3/2013
Corrección de errores	Esteban Capello	Hecho	15	7	14/3/2013	13/3/2013

Sprint 31

Fecha Inicio	14/mar/2013
Fecha Fin	21/mar/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Corrección de errores	Esteban Capello	Hecho	10	10	16/3/2013	15/3/2013
Corrección de errores	Esteban Capello	Hecho	10	7	18/3/2013	16/3/2013
Desarrollo de controladores y servicios 10	Esteban Capello	Hecho	15	15	21/3/2013	19/3/2013
Desarrollo de controladores y servicios 11	Esteban Capello	Incompleto	15	10	24/3/2013	21/3/2013

Sprint 32

Fecha Inicio	21/mar/2013
Fecha Fin	29/mar/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de controladores y servicios 11	Esteban Capello	Hecho	5	4	24/3/2013	22/3/2013
Pruebas de controladores y servicios 10 y 11	Esteban Capello	Hecho	10	7	26/3/2013	23/3/2013
Corrección de errores	Esteban Capello	Hecho	15	15	29/3/2013	26/3/2013
Corrección de errores	Esteban Capello	Hecho	10	12	31/3/2013	29/3/2013

Sprint 33

Fecha Inicio	29/mar/2013
Fecha Fin	5/abr/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Corrección de errores	Esteban Capello	Hecho	10	8	2/4/2013	31/3/13
Agregar secciones de HTML5 en el prototipo	Esteban Capello	Hecho	10	16	4/4/2013	4/4/2013
Corrección de errores	Esteban Capello	Incompleto	5	8	5/4/2013	5/4/2013

Sprint 34

Fecha Inicio	5/abr/2013
Fecha Fin	13/abr/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Corrección de errores	Esteban Capello	Hecho	0	3	5/4/2013	6/4/2013
Pruebas generales del sistema	Esteban Capello	Hecho	15	17	8/4/2013	10/4/2013
Corrección de errores	Esteban Capello	Hecho	10	14	10/4/2013	13/4/2013
Estabilización del sistema	Esteban Capello	Falta	15		13/4/2013	

Sprint 35

Fecha Inicio	13/abr/2013
Fecha Fin	20/abr/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Estabilización del sistema	Esteban Capello	Hecho	15	15	13/4/2013	16/4/2013
Estabilización del sistema	Esteban Capello	Hecho	15	17	16/4/2013	20/4/2013

Sprint 36

Fecha Inicio	20/abr/2013
Fecha Fin	29/abr/2013

Tarea Tomada	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha Planeada fin	Fecha de Culminación
Desarrollo de análisis económico	Esteban Capello	Hecho	5	4	17/4/2013	21/4/2013
Desarrollo de Conclusiones y Recomendaciones	Esteban Capello	Hecho	15	10	20/4/2013	23/4/2013
Revisión general de la tesis	Esteban Capello	Hecho	15	12	23/4/2013	26/4/2014
Correcciones de la Tesis	Esteban Capello	Hecho	15	14	26/4/2013	29/4/2016

Anexo 12: Daily Meetings

Sprint 1

Fecha Inicio	6/8/2012
Fecha Fin	12/8/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
1 - 2	6/8/2012	Nada	Escribir ideas para el capítulo 1, antecedentes , justificación y objetivos	No	Listo	OK!
3 - 4	7/8/2012	Escribir ideas para el capítulo 1, antecedentes , justificación y objetivos	Escribir el alcance y áreas de aplicación	No	Adelantado en tareas	Perfecto
5	8/8/2012	Escribir el alcance y áreas de aplicación	Escribir la introducción	No	Adelantado en tareas	Perfecto
6 - 7	9/8/2012	Escritura de la introducción	Revisar y corregir el capítulo 1	No	Adelantado en tareas	OK!
8	10/8/2012	Se corrige el capítulo 1	Escribir ideas para el capítulo 2	No	Adelantado en tareas	OK!
9 - 10	11/8/2012	Se tiene el planteamiento para el capítulo 2	Investigar y escribir sobre metodologías tradicionales	No	Se adelantan tareas del 2do sprint	Chévere
11 - 12	12/8/2012	Se escribió la sección de metodologías tradicionales	Investigar y escribir sobre metodologías ágiles	No	Se adelantan tareas del 2do sprint	OK!

Sprint 2

Fecha Inicio	12/8/2012
Fecha Fin	18/8/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
13	13/8/2012	Investigar y escribir sobre metodologías ágiles	Un cuadro comparativo de los tipos de metodologías	No	Listo	OK!
14	14/8/2012	Un cuadro comparativo de los tipos de metodologías	Investigar Scrum	No	Listo	Chévere
15	15/8/2012	Investigar Scrum	Escribir la sección de scrum	No	Listo	OK!
16	16/8/2012	Escribir la sección de scrum	Investigar XP y Cristal Clear	No	Listo	OK!
16	17/8/2012	Investigar XP y Cristal Clear	Escribir los anexos de XP y Cristal Clear	No	Listo	OK!
16 - 17	18/8/2012	Escribir los anexos de XP y Cristal Clear	Hacer un cuadro comparativo de la metodologías ágiles	No	Listo	Perfecto

Sprint 3

Fecha Inicio	18/8/2012
Fecha Fin	24/8/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
18	19/8/2012	Hacer un cuadro comparativo de la metodologías ágiles	Investigar sobre las herramientas de apoyo que utilizaré en esta tesis	No	Listo	OK
18	20/8/2012	Investigar sobre las herramientas de apoyo que utilizaré en esta tesis	Continuar con la investigación	No	Listo	OK
19 - 20	21/8/2012	Investigación de herramientas	Investigar un poco más de wireframes y escribir dicha sección	No	Listo	OK
21	22/8/2012	Escribir la sección de wireframes	Investigar sobre algunas herramientas de UML	No	Listo	OK
22	23/8/2012	Investigar UML	Escribir la sección del diagrama de Base de datos	No	Listo	OK
25 - 26	24/8/2012	Escribir la sección del diagrama de Base de datos	Escribir las secciones de los diagramas de clases y de casos de uso	No	Listo	OK

Sprint 4

Fecha Inicio	24/8/2012
Fecha Fin	31/8/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
27	25/8/2012	Escribir las secciones de los diagramas de clases y de casos de uso	Revisar el capítulo 2, para buscar correcciones que deba hacer	No	Listo	OK, sigue así
27	26/8/2012	Revisar el capítulo 2	Seguir con la revisión	No	Listo	OK
27	27/8/2012	Revisar el capítulo 2	Seguir con la revisión	No	Listo	OK
27	28/8/2012	Revisar el capítulo 2	Seguir con la revisión	No	Listo	OK
27	29/8/2012	Revisar el capítulo 2	Seguir con la revisión	No	Listo	Chévere
28	30/8/2012	Revisar el capítulo 2	Corregir el capítulo 2	No	Listo	OK
28	31/8/2012	Corregir el capítulo 2	Terminar las correcciones del capítulo 2	No	Listo	OK

Sprint 5

Fecha Inicio	31/8/2012
Fecha Fin	6/9/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
29	1/9/2012	Corregir el capítulo 2	Investigar las normas APA	No	Listo	OK
30	2/9/2012	Investigar las normas APA	Cambiar los capítulos 1 y 2 en base a las normas APA	No	Listo	OK
30	3/9/2012	Cambiar los capítulos 1 y 2 en base a las normas APA	Cambiar los capítulos 1 y 2 en base a las normas APA	No	Listo	OK
30	4/9/2012	Cambiar los capítulos 1 y 2 en base a las normas APA	Cambiar los capítulos 1 y 2 en base a las normas APA	No	Listo	OK
31	5/9/2012	Cambiar los capítulos 1 y 2 en base a las normas APA	Investigar sobre HTML5	No	Listo	Dado a que estas adelantado en tareas no hay problema en que descanses mañana
N/A	6/9/2012	Investigar sobre HTML5	Este día voy a descansar	N/A	N/A	N/A

Sprint 6

Fecha Inicio	6/9/2012
Fecha Fin	13/9/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
32	7/9/2012	Descanso	Hacer un ejemplo de HTML5	No	Listo	OK
32	8/9/2012	Hacer un ejemplo de HTML5	Terminar dicho ejemplo	No	Listo	OK
33	9/9/2012	Terminar ejemplo de HTML5	Escribir la sección de HTML5	No	Listo	OK
34	10/9/2012	Escribir la sección de HTML5	Investigar sobre Javascript	No	Listo	OK
34	11/9/2012	Investigar javascript	Continuar con la investigación	No	Listo	OK
36 - 37	12/9/2012	Investigar javascript	Escribir sobre javascript e investigar sobre JBOSS	No	Listo	OK
38	13/9/2012	Investigar JBOSS	Escribir la sección de JBOSS	No	Listo	OK

Sprint 7

Fecha Inicio	13/9/2012
Fecha Fin	19/9/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
39	14/9/2012	Escribir la sección de JBOSS	Investigar Spring MVC	No	Listo	OK
39	15/9/2012	Investigar Spring MVC	Investigar Spring MVC	No	Listo	OK
39	16/9/2012	Investigar Spring MVC	Investigar Spring MVC	No	Listo	OK
40	17/9/2012	Investigar Spring MVC	Investigar Spring Framework	No	Listo	OK, sin embargo ya no tenemos un adelanto, estamos igualados al cronograma
41	18/9/2012	Investigar Spring Framework	Hacer un ejemplo de spring mvc	No	Listo	OK
41	19/9/2012	Hacer un ejemplo de spring mvc	Hacer un ejemplo de spring mvc	No	Listo	OK

Sprint 8

Fecha Inicio	19/9/2012
Fecha Fin	25/9/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
42	20/9/2012	Hacer un ejemplo de spring mvc	Escribir sobre spring framework 3	No	Listo	OK
42	21/9/2012	Escribir sobre spring framework 3	Escribir sobre spring framework 3	No	Atrasado	OK
43 - 44	22/9/2012	Escribir sobre spring framework 3. Esta tarea ocupó más tiempo de lo planeado	Investigar sobre primefaces y realizar un ejemplo	No	Listo	Debes realizar alguna tarea con mayor velocidad para poder igualarnos al cronogram a otra vez
45 - 46	23/9/2012	Investigar sobre primefaces y realizar un ejemplo	Escribir la sección de primefaces e investigar sobre hibernate	No	Listo	OK
47	24/9/2012	Escribir la sección de primefaces e investigar sobre hibernate	Escribir la sección de hibernate	No	Adelantado	OK
47	25/9/2012	Escribir la sección de hibernate	Escribir la sección de hibernate	No	Listo	Perfecto, igualados otra vez

Sprint 9

Fecha Inicio	25/9/2012
Fecha Fin	3/10/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
48	26/9/2012	Escribir la sección de hibernate	Revisar el capítulo 3	No	Listo	OK
49	27/9/2012	Revisar el capítulo 3	Corregir el capítulo 3	No	Adelantado	Sigue así
50	28/9/2012	Corregir el capítulo 3	Pensar ideas para los roles que poseerá el prototipo	No	Adelantado	Perfecto
50	29/9/2012	Roles para el prototipo	Roles para el prototipo	No	Listo	OK
51	30/9/2012	Roles para el prototipo	Ideas para los wireframes	No	Listo	OK
51	1/10/2012	Ideas para los wireframes	Seguir con las formas que van a tener las pantallas	No	Listo	OK
51	2/10/2012	Ideas para los wireframes	Ideas para los wireframes	No	Listo	OK
51	3/10/2012	Ideas para los wireframes	Ideas para los wireframes	No	Igualado	OK

Sprint 10

Fecha Inicio	3/10/2012
Fecha Fin	8/10/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
52	4/10/2012	Ideas para los wireframes	Hacer los wireframes	No	Listo	Este sprint solo posee esta tarea, intenta adelantar la tarea en lo que puedas
52	5/10/2012	Wireframes	Wireframes	No	Listo	OK
52	6/10/2012	Wireframes	Wireframes	No	Listo	OK
52	7/10/2012	Wireframes	Wireframes	No	Listo	OK
52	8/10/2012	Wireframes	Wireframes	No	Listo	OK

Sprint 11

Fecha Inicio	8/10/2012
Fecha Fin	13/10/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
53	9/10/2012	Wireframes	Seguir haciendo los Wireframes	No	Listo	OK
53	10/10/2012	Wireframes	Wireframes	No	Listo	OK
53	11/10/2012	Wireframes	Wireframes	No	Listo	OK
53	12/10/2012	Wireframes	Wireframes	No	Listo	OK
53	13/10/2012	Wireframes	Wireframes	No	Listo	OK

Sprint 12

Fecha Inicio	13/10/2012
Fecha Fin	18/10/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
54	14/10/2012	Wireframes	Seguir haciendo los Wireframes	No	Listo	OK
54	15/10/2012	Wireframes	Wireframes	No	Listo	OK
54	16/10/2012	Wireframes	Wireframes	No	Listo	OK
54	17/10/2012	Wireframes	Wireframes	No	Listo	OK
54	18/10/2012	Wireframes	Wireframes	No	Listo	OK

Sprint 13

Fecha Inicio	18/10/2012
Fecha Fin	26/10/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
55	19/10/2012	Wireframes	Revisar todos los wireframes hechos	No	Listo	OK
56	20/10/2012	Revisar wireframes	Corregir algunos wireframes	No	Listo	OK
57	21/10/2012	Corregir wireframes	Diseñar la base de datos para el prototipo	No	Listo	OK
57	22/10/2012	Diseñar la base de datos para el prototipo	Diseñar la base de datos para el prototipo	No	Listo	OK
57	23/10/2012	Diseñar la base de	Diseñar la base de datos para	No	Listo	OK

		datos para el prototipo	el prototipo			
57	24/10/2012	Diseñar la base de datos para el prototipo	Diseñar la base de datos para el prototipo	No	Listo	OK
57	25/10/2012	Diseñar la base de datos para el prototipo	Diseñar la base de datos para el prototipo	No	Tomó más tiempo de lo planeado	OK
58	26/10/2012	Diseñar la base de datos para el prototipo	Hacer el diagrama entidad relación	No	No se alcanzó a terminar esta tarea	Debes esforzarte por volverte a igualar en cuanto a los tiempos

Sprint 14

Fecha Inicio	26/10/2012
Fecha Fin	2/11/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
58	27/10/2012	El diagrama Entidad - Relación	Continuar con el diagrama E - R	No	Atrasado	Tenemos que igualarnos
58	28/10/2012	El diagrama Entidad - Relación	Finalizar el diagrama E - R	No	Atrasado	Dale con fuerza estamos atrasados por dos días
59	29/10/2012	El diagrama Entidad - Relación	El script para la base de datos	No	Atrasado	Ahora ya son 3 días, si puedes volver a alcanzar el cronograma

62	30/10/2012	El script para la base de datos	Se empieza la planificación del diagrama de clases	No	Atrasado	Atento el atraso continua
62	31/10/2012	Diagrama de clases	Diagrama de clases	No	Atrasado	Seguimos atrasados
62	1/11/2012	Diagrama de clases	Diagrama de clases	No	Atrasado	Intenta dedicarle un poco más de tiempo a estas tareas para igualarnos al cronograma
62	2/11/2012	Diagrama de clases	Diagrama de clases	No	No se alcanzó a realizar una tarea de este sprint	La tarea de planificación que no se alcanzó se la deberá continuar en el siguiente sprint

Sprint 15

Fecha Inicio	2/11/2012
Fecha Fin	10/11/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
62	3/11/2012	Diagrama de clases	Diagrama de clases	No	Atrasado	Estamos por atras del cronograma por 4 días
62	4/11/2012	Diagrama de clases	Diagrama de clases	No	Atrasado	Dale
63	5/11/2012	Diagrama de clases	Diagrama de clases	No	Atrasado	El atraso aminoró por un día, sigue así

64	6/11/2012	Diagrama de clases	Hacer el diagrama de clases	No	Atrasado	Dale
64	7/11/2012	Hacer el diagrama de clases	Hacer el diagrama de clases	No	Atrasado	Muy bien ya son 2 días de atraso
65	8/11/2012	Hacer el diagrama de clases	Planificar Casos de Uso	No	Atrasado	Si podemos igualarnos
65	9/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	Dale
65	10/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	No se alcanzó a terminar esta tarea se la continuará en el siguiente sprint	Ok, pero recuerda que ya tenemos un atraso

Sprint 16

Fecha Inicio	10/11/2012
Fecha Fin	15/11/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
65	11/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	EL atraso es de 1 día
66	12/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	Dale
66	13/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	Dale
66	14/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	Dale

66	15/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Listo	Felicitaciones lograste igualarte
----	------------	-------------------------	-------------------------	----	-------	-----------------------------------

Sprint 17

Fecha Inicio	15/11/2012
Fecha Fin	20/11/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
67	16/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Listo	OK
67	17/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Listo	OK
67	18/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Listo	OK
67	19/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Listo	OK
67	20/11/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Listo	OK

Sprint 18

Fecha Inicio	20/11/2012
Fecha Fin	28/11/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
N/A	21/11/2012	Planificar Casos de Uso	La computadora no responde, parece que se me dañó	Sí, computadora dañada	Atrasado y sin equipo para trabajar	Revísale, voy a estar atento para cuando podamos volver a trabajar

N/A	22/11/2012	N/A	La computadora sigue dañada, parece que es el disco duro	Si, computadora dañada	Atrasado y sin equipo para trabajar	Necesitas reemplazar el disco
N/A	23/11/2012		Si es el disco duro de mi laptop	Si, computadora dañada	Atrasado y sin equipo para trabajar	¿Cuándo vas a comprar el nuevo disco duro?
N/A	24/11/2012	N/A	El 30 me pagan el sueldo, ahí voy a tener dinero para comprar un nuevo disco duro	Si, computadora dañada	Atrasado y sin equipo para trabajar	OK, estoy atento cualquier novedad
N/A	25/11/2012	N/A	Esperando para comprar el Disco	Si, computadora dañada	Atrasado y sin equipo para trabajar	
N/A	26/11/2012	N/A	N/A	Si, computadora dañada	Atrasado y sin equipo para trabajar	
N/A	27/11/2012	N/A	He solicitado vacaciones para poder igualarme en el trabajo atrasado	Si, computadora dañada	Atrasado y sin equipo para trabajar	Esa fue una buena idea solo necesitas el disco nuevo ahora. ¿Hasta cuándo son?
N/A	28/11/2012	N/A	Son hasta el 11 de diciembre	Si, computadora dañada	Atrasado y sin equipo para trabajar	OK

Sprint 19

Fecha Inicio	28/11/2012
Fecha Fin	5/12/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
N/A	29/11/2012	N/A	Comprar el disco nuevo	No	Atrasado y sin equipo para trabajar	Me avisas cuando la termines de arreglar
N/A	30/11/2012	N/A	Instalando el sistema operativo	No	Atrasado	OK
N/A	1/12/2012	N/A	Hoy termino de instalar todo lo que necesitamos	No	Atrasado	OK
68 - 69	2/12/2012	Arreglar la laptop	Seguir con la planificación de casos de uso	No	Atrasado	De acuerdo, tenemos un retraso de más o menos 11 días, ahora que estas de vacaciones debes poder igualarte
68 - 69	3/12/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	OK
68 - 69	4/12/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	OK
68 - 69	5/12/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	Ahora son 8 días de retraso

Sprint 20

Fecha Inicio	5/12/2012
Fecha Fin	10/12/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
70	6/12/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	OK
70	7/12/2012	Planificar Casos de Uso	Planificar Casos de Uso	No	Atrasado	El atraso sigue en 8 días
71	8/12/2012	Planificar Casos de Uso	Hoy empiezo a hacer los casos de uso	No	Atrasado	OK
71	9/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	OK
71	10/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	Muy bien!, el atraso ya son 5 días

Sprint 21

Fecha Inicio	10/12/2012
Fecha Fin	15/12/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
72	11/12/2012	Casos de Uso	Casos de Uso	No	Hoy se terminan mis vacaciones	OK, ya solo tenemos un atraso de 1 día
73	12/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	OK
73	13/12/2012	Casos de	Casos de	No	Atrasado	OK

		Uso	Uso			
73	14/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	OK
73	15/12/2012	Casos de Uso	Casos de Uso	No	Atrasado, y no se pudo terminar la tarea	OK, dale con fuerza que ya solo tenemos un día de retraso

Sprint 22

Fecha Inicio	15/12/2012
Fecha Fin	20/12/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
73	16/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	Atraso de 1 día
74	17/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	OK
74	18/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	OK
74	19/12/2012	Casos de Uso	Casos de Uso	No	Atrasado	OK
74	20/12/2012	Casos de Uso	Casos de Uso	No	Listo	Perfecto, ya lograste igualarte

Sprint 23

Fecha Inicio	20/12/2012
Fecha Fin	28/12/2012

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
75	21/12/2012	Casos de Uso	Casos de Uso	No	Listo	OK
75	22/12/2012	Casos de Uso	Casos de Uso	No	Hoy termino esta tarea	OK
N/A	23/12/2012	N/A	N/A	No	Como no hay atraso me voy a tomar estos 3 días	
N/A	24/12/2012	N/A	N/A	No	Feliz Navidad	Feliz Navidad para ti también
N/A	25/12/2012	N/A	N/A	No		
76	26/12/2012	Casos de Uso	Casos de Uso	No	Listo	OK
76	27/12/2012	Casos de Uso	Casos de Uso	No	Listo	OK
76	28/12/2012	Casos de Uso	Casos de Uso	No	Listo	OK

Sprint 24

Fecha Inicio	28/12/2012
Fecha Fin	2/1/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
77	29/12/2012	Casos de Uso	Casos de Uso	No	Listo	OK
77	30/12/2012	Casos de Uso	Casos de Uso	No	Adelantado para celebrar el año nuevo	Perfecto
N/A	31/12/2012	N/A	N/A	No		
N/A	1/1/2013	N/A	N/A	No	Feliz Año	Feliz Año
N/A	2/1/2013	N/A	N/A	No		

Sprint 25

Fecha Inicio	2/1/2013
Fecha Fin	9/1/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
78	3/1/2013	Casos de Uso	Casos de Uso	No	Listo	OK
78	4/1/2013	Casos de Uso	Casos de Uso	No	Listo	OK
78	5/1/2013	Casos de Uso	Casos de Uso	No	Listo	OK
78	6/1/2013	Casos de Uso	Casos de Uso	No	Listo	OK
79	7/1/2013	Casos de Uso	Casos de Uso	No	Adelantado	Perfecto

80	8/1/2013	Casos de Uso	Crear el proyecto en Java	No	Adelantado, hoy me avisaron que el 17 viajo a Panamá por trabajo	Eso nos va afectar en el cronograma, ¿cuánto tiempo te quedas allá?
81	9/1/2013	Proyecto Java	Configurar el proyecto	No	Adelantado, me quedo por cuatro semanas	OK

Sprint 26

Fecha Inicio	9/1/2013
Fecha Fin	17/1/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
82	10/1/2013	Configurar el proyecto	Crear las clases de las entidades	No	Listo	OK
83	11/1/2013	Crear las clases de las entidades	Crear los DAOs	No	Listo	OK
84	12/1/2013	Crear los DAOs	Se empiezan a hacer las pantallas del prototipo	No	Listo	OK
84	13/1/2013	Pantallas	Pantallas	No	Listo	Ya mismo viajas, lo mejor va a ser posponer los sprints durante ese tiempo para que regreses a

						igualarte
84	14/1/2013	Pantallas	Pantallas	No	Listo	OK
84					Listo, pedí permiso en el trabajo para que cuando regrese me tomo una semana (me toca recuperar ese tiempo después) para igualarme con la tesis	Perfecto, en ese tiempo debes igualarte
	15/1/2013	Pantallas	Pantallas	No		
84					Mañana me envían de viaje a Panamá por trabajo	
	16/1/2013	Pantallas	Pantallas	No		OK
N/A	17/1/2013			Si, viaje	De viaje	

Sprint 27

Fecha Inicio	15/2/2013
Fecha Fin	22/2/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
85	15/2/2013	Ya regrese del viaje y tengo hasta el 22 para igualarme	Continuar haciendo las pantallas en el prototipo	No	Atrasado	OK, haz lo posible por igualarte que tenemos un atraso grande
85	16/2/2013	Pantallas	Pantallas	No	Atrasado	OK
86	17/2/2013	Pantallas	Pantallas	No	Atrasado	OK
87 - 88	18/2/2013	Pantallas	Terminar las pantallas y empezar con los controladores y servicios	No	Atrasado	OK
89 - 90	19/2/2013	Controladores y Servicios	Controladores y Servicios	No	Atrasado	OK
91	20/2/2013	Controladores y Servicios	Pruebas de Controladores y servicios hechos	No	Atrasado	OK
92	21/2/2013	Pruebas de Controladores y servicios hechos	Corregir errores	No	Atrasado	OK
93 - 94	22/2/2013	Corregir errores	Corregir errores y continuar con los Controladores y Servicios	No	Atrasado, falta una tarea para cumplir con los sprints	OK

					atrasados	
--	--	--	--	--	-----------	--

Sprint 28

Fecha Inicio	22/2/2013
Fecha Fin	27/2/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
95	23/2/2013	Controladores y Servicios	Controladores y Servicios	No	Atrasado	OK
96 - 97	24/2/2013	Controladores y Servicios	Controladores y servicios, junto con pruebas sobre estos nuevos	No	Listo	Felicidades lograste igualarte de nuevo
98	25/2/2013	Pruebas de controladores y servicios	Corregir errores	No	Listo	OK
98	26/2/2013	Corregir errores	Corregir errores	No	Listo	OK
98	27/2/2013	Corregir errores	Corregir errores	No	Listo	OK

Sprint 29

Fecha Inicio	27/2/2013
Fecha Fin	7/3/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
99	28/2/2013	Corregir errores	Corregir errores	No	Listo	OK
100	1/3/2013	Corregir errores	Controladores y servicios	No	Listo	OK
100	2/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
100	3/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
100	4/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
101	5/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
101	6/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
101	7/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK

Sprint 30

Fecha Inicio	7/3/2013
Fecha Fin	14/3/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
102	8/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
102	9/3/2013	Controladores y servicios	Controladores y servicios	No	Listo	OK
103	10/3/2013	Controladores y servicios	Pruebas de controladores y servicios	No	Listo	OK
103	11/3/2013	Pruebas de controladores y servicios	Pruebas de controladores y servicios	No	Listo	OK
104	12/3/2013	Pruebas de controladores y servicios	Corregir errores	No	Listo	OK
104	13/3/2013	Corregir errores	Corregir errores	No	Listo, si me adelanto quisiera descansar un poco mañana	OK, no hay problema
N/A	14/3/2013			No	Descanso	

Sprint 31

Fecha Inicio	14/3/2013
Fecha Fin	21/3/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
105	15/3/2013	Descanso	Corregir errores	No	Listo	OK
106	16/3/2013	Corregir errores	Corregir errores	No	Listo	OK
107	17/3/2013	Corregir errores	Controladores y Servicios	No	Listo	OK
107	18/3/2013	Controladores y Servicios	Controladores y Servicios	No	Listo	OK
107	19/3/2013	Controladores y Servicios	Controladores y Servicios	No	Listo	Muy bien vamos con un buen adelanto
108	20/3/2013	Controladores y Servicios	Controladores y Servicios	No	Listo	OK
108	21/3/2013	Controladores y Servicios	Controladores y Servicios	No	Listo	Perfecto

Sprint 32

Fecha Inicio	21/3/2013
Fecha Fin	29/3/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
108	22/3/2013	Controladores y Servicios	Controladores y Servicios	No	Listo	OK, tenemos un adelanto de 2 días

109	23/3/2013	Controladores y Servicios	Pruebas de los controladores y servicios	No	Listo	OK
110	24/3/2013	Pruebas de los controladores y servicios	Corregir errores	No	Listo	OK
110	25/3/2013	Corregir errores	Corregir errores	No	Listo	OK
110	26/3/2013	Corregir errores	Corregir errores	No	Listo	OK
111	27/3/2013	Corregir errores	Corregir errores	No	Listo	OK
111	28/3/2013	Corregir errores	Corregir errores	No	Listo	OK
111	29/3/2013	Corregir errores	Corregir errores	No	Listo	Perfecto

Sprint 33

Fecha Inicio	29/3/2013
Fecha Fin	5/4/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
112	30/3/2013	Corregir errores	Corregir errores	No	Listo	OK
112	31/3/2013	Corregir errores	Corregir errores	No	Listo	OK
113	1/4/2013	Corregir errores	Realizar las partes que utilizan HTML5 en el prototipo	No	Listo	OK
113	2/4/2013	Partes HTML5	Partes HTML5	No	Listo	OK
113	3/4/2013	Partes HTML5	Partes HTML5	No	Listo	OK
113	4/4/2013	Partes HTML5	Partes HTML5	No	Listo	OK

114	5/4/2013	Partes HTML5	Corregir errores, esta tarea no se logró terminar en este sprint ya que las partes de HTML5 se alargaron más de lo debido	No	Atrasado	OK
-----	----------	--------------	---	----	----------	----

Sprint 34

Fecha Inicio	5/4/2013
Fecha Fin	13/4/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
114	6/4/2013	Corregir errores	Corregir errores	No	Atrasado	OK
115	7/4/2013	Corregir errores	Pruebas de todo el prototipo	No	Atrasado	Intenta recuperar el tiempo que lleva atrasado
115	8/4/2013	Pruebas de todo el prototipo	Pruebas de todo el prototipo	No	Atrasado	OK
115	9/4/2013	Pruebas de todo el prototipo	Pruebas de todo el prototipo	No	Atrasado	OK
115	10/4/2013	Pruebas de todo el prototipo	Pruebas de todo el prototipo	No	Atrasado	OK
116	11/4/2013	Pruebas de todo el prototipo	Corregir errores	No	Atrasado	OK
116	12/4/2013	Corregir errores	Corregir errores	No	Atrasado	OK
116	13/4/2013	Corregir errores	Corregir errores	No	Atrasado	OK

Sprint 35

Fecha Inicio	13/3/2013
Fecha Fin	20/3/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
117	14/4/2013	Corregir errores	Se continua con pruebas y correcciones para estabilizar el sistema	No	Listo	OK
117	15/4/2013	Pruebas y correcciones	Pruebas y correcciones	No	Listo	OK
117	16/4/2013	Pruebas y correcciones	Pruebas y correcciones	No	Listo	OK
118	17/4/2013	Pruebas y correcciones	Pruebas y correcciones	No	Listo	OK
118	18/4/2013	Pruebas y correcciones	Pruebas y correcciones	No	Listo	OK
118	19/4/2013	Pruebas y correcciones	Pruebas y correcciones	No	Listo	OK
118	20/4/2013	Pruebas y correcciones	Pruebas y correcciones	No	Listo	OK

Sprint 36

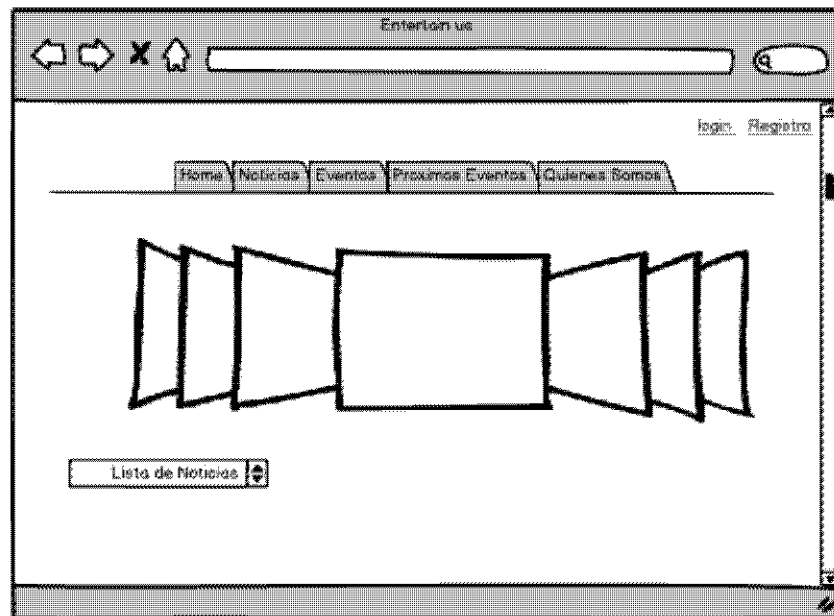
Fecha Inicio	20/3/2013
Fecha Fin	29/3/2013

# de Historia	Fecha	¿Qué has hecho desde ayer?	¿Qué vas a hacer hoy?	¿Existe algún impedimento?	Estado del Equipo	Estado Scrum Master
120	21/4/2013	Pruebas y correcciones	El análisis económico	No	Listo, que tal si cambiamos este sprint para ya terminar todo	OK, entonces hagámosle hasta el 29
121	22/4/2013	El análisis económico	Escribir las conclusiones y recomendaciones	No	Listo	OK
121	23/4/2013	Conclusiones y Recomendaciones	Conclusiones y Recomendaciones	No	Listo	OK
122	24/4/2013	Conclusiones y Recomendaciones	Revisión de toda la tesis	No	Listo	OK
122	25/4/2013	Revisión de toda la tesis	Revisión de toda la tesis	No	Listo	OK
122	26/4/2013	Revisión de toda la tesis	Revisión de toda la tesis	No	Listo	OK
123	27/4/2013	Revisión de toda la tesis	Corrección de últimos errores	No	Listo	Dale ya falta poco
123	28/4/2013	Corrección de últimos errores	Corrección de últimos errores	No	Listo	OK
123	29/4/2013	Corrección de últimos errores	Corrección de últimos errores	No	Listo	Perfecto hemos terminado

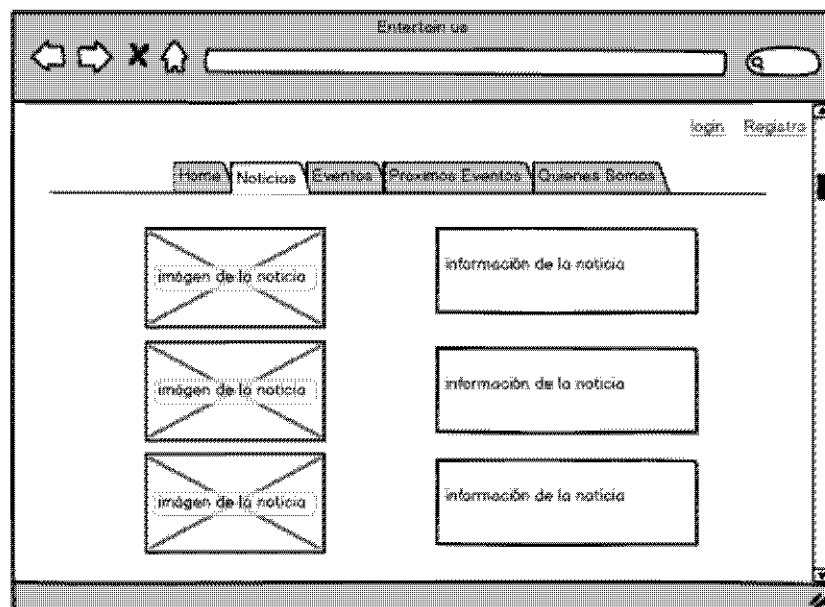
Anexo 13: WireFrames

Rol: Usuario sin Cuenta

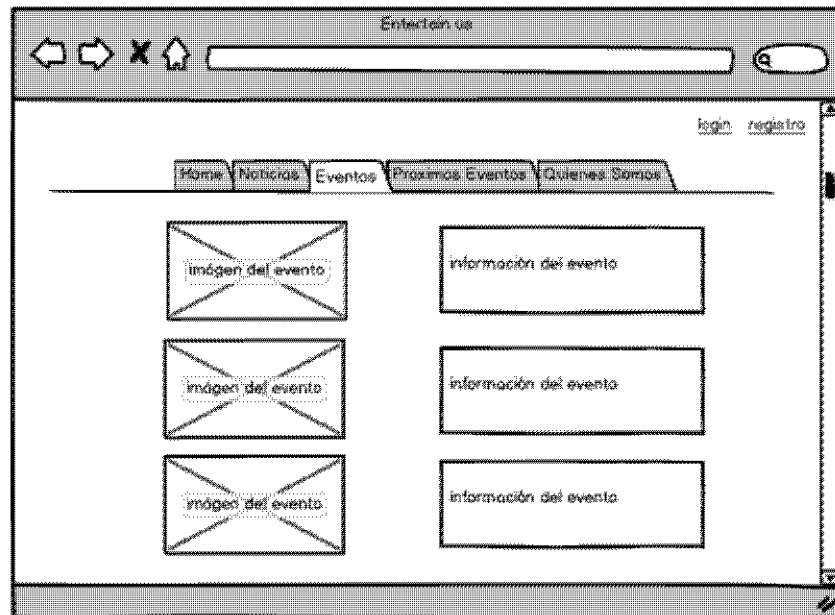
Página Principal



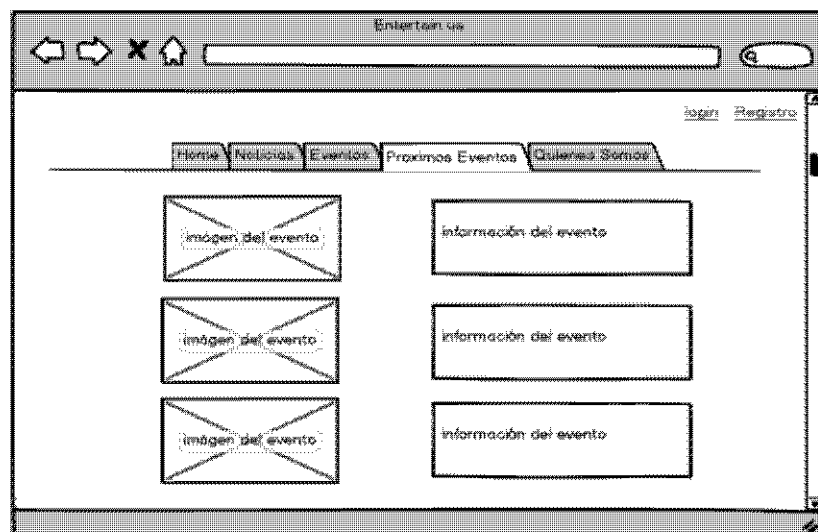
Página de Noticias



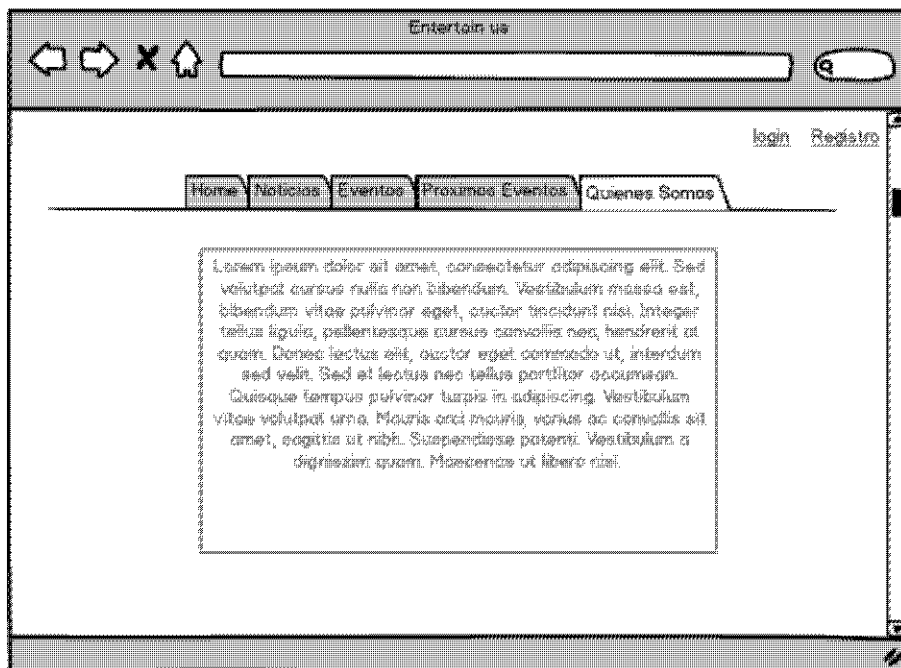
Página de Eventos



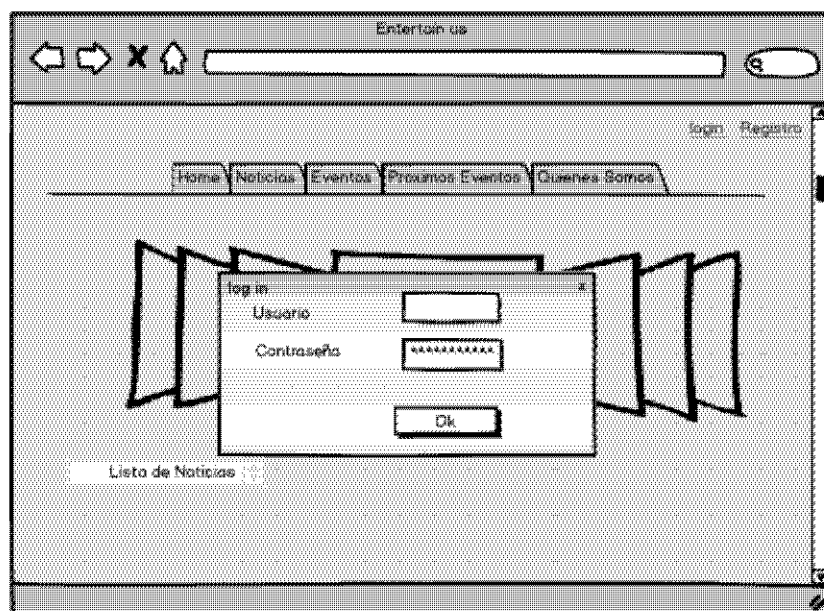
Página de Próximos Eventos



Página de Quienes Somos



Pop-up de Log in



Página de Registro

Entertain us

login Registro

Home Noticias Eventos Proximos Eventos Quienes Somos

Registro

Nombre: Apellidos:

Identificación: Teléfono:

Fecha de Nacimiento: / / E-mail:

Sexo:

Dirección:

Calle Principal:

Calle secundaria:

Numeración:

Comentario

Cómo quiere iniciar sesión?

Nombre de la Cuenta

Contraseña

Vuelva a escribir la contraseña

Rol: Usuario Administrador

Página de Género

Entertain us

Administrador

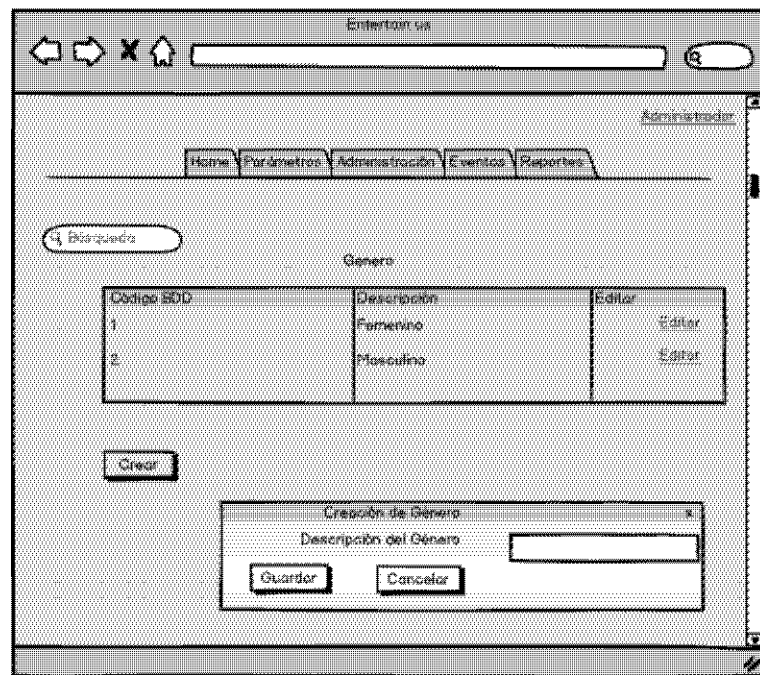
Home Parámetros Administración Eventos Reportes

Búsqueda

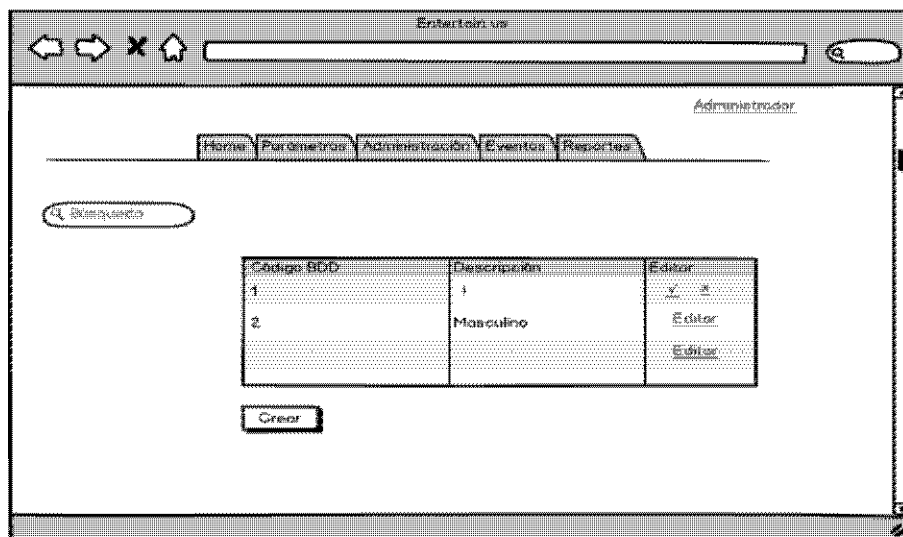
Genero

Código BOD	Descripción	Editar
1	Tarjeta de Crédito	Editar
2	Tarjeta de Débito	Editar
3	Efectivo	Editar

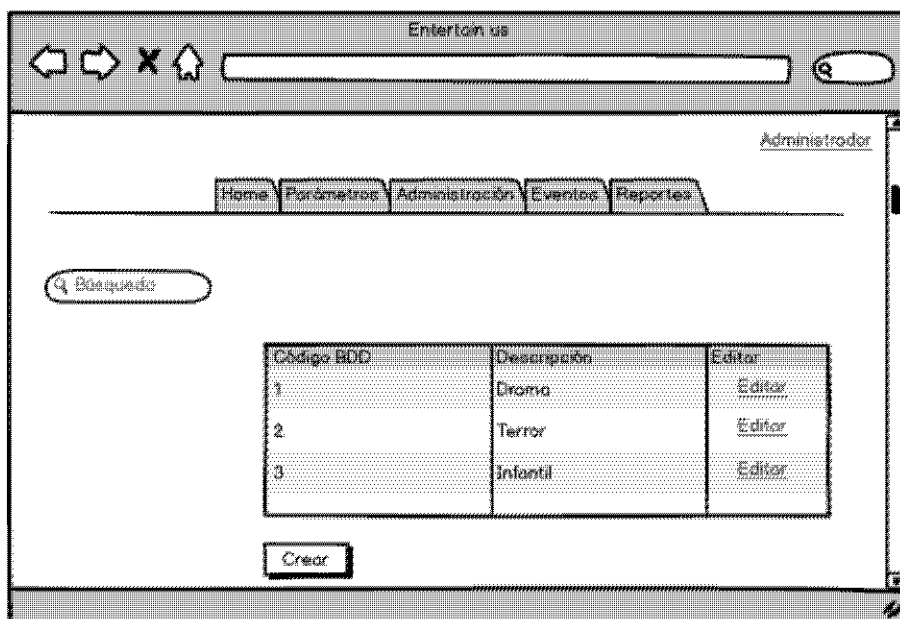
Página de Creación de Género



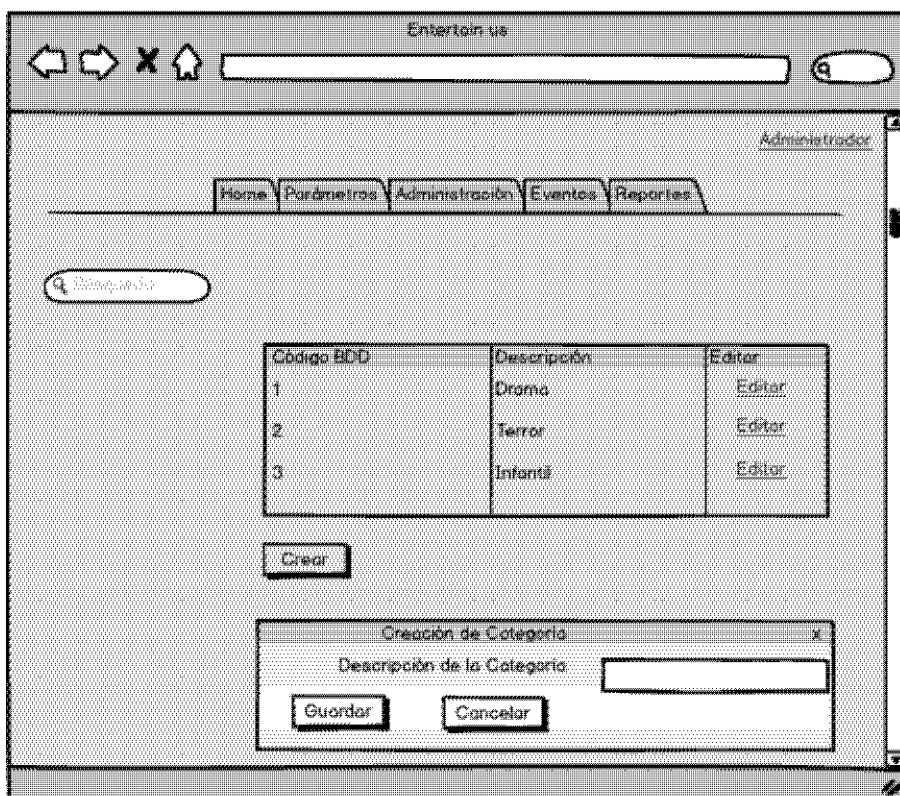
Página de Modificación de Género



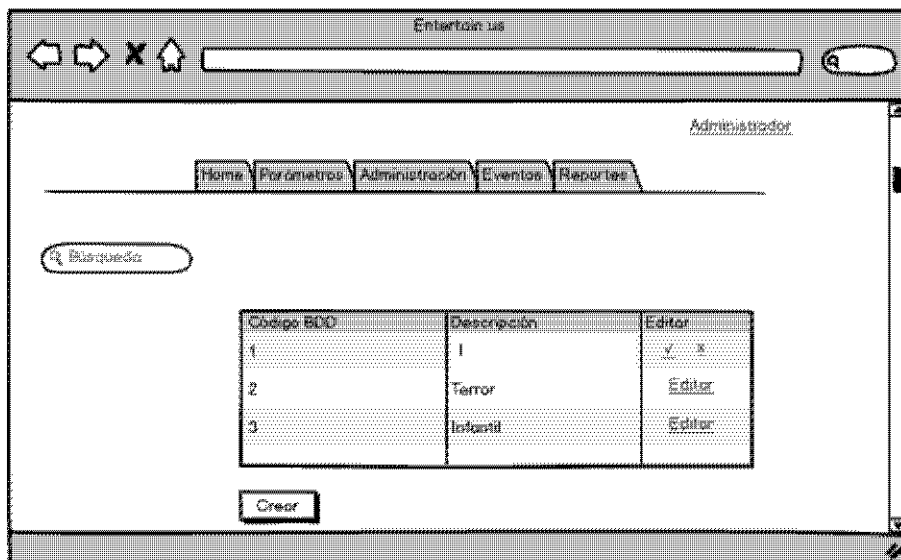
Página de Categoría



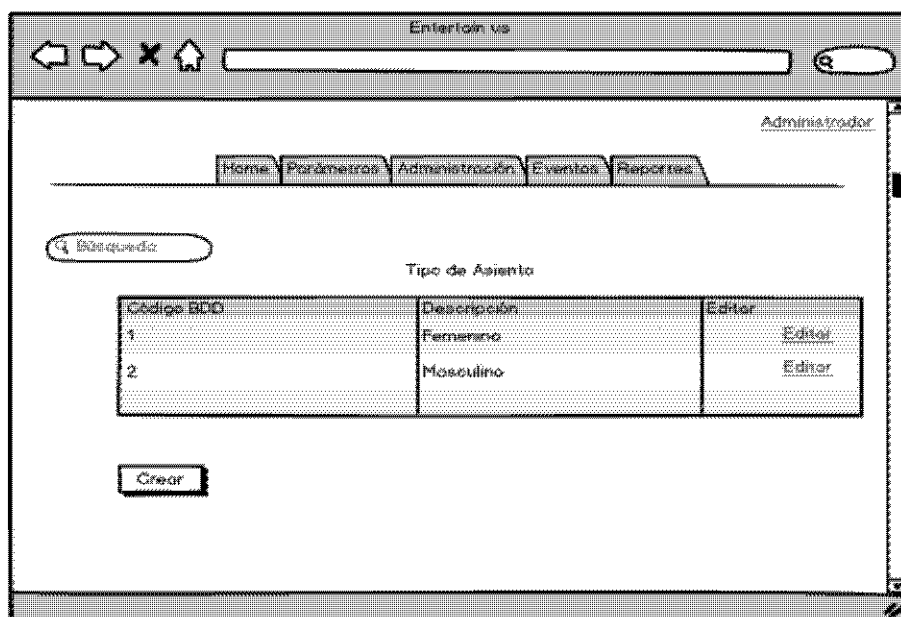
Página de Creación de Categoría



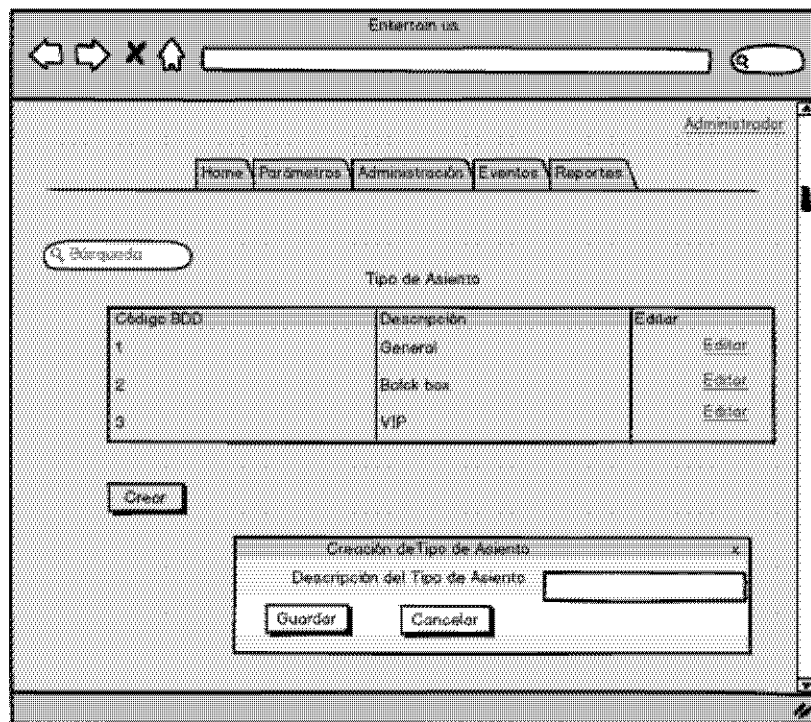
Página de Modificación de Categoría



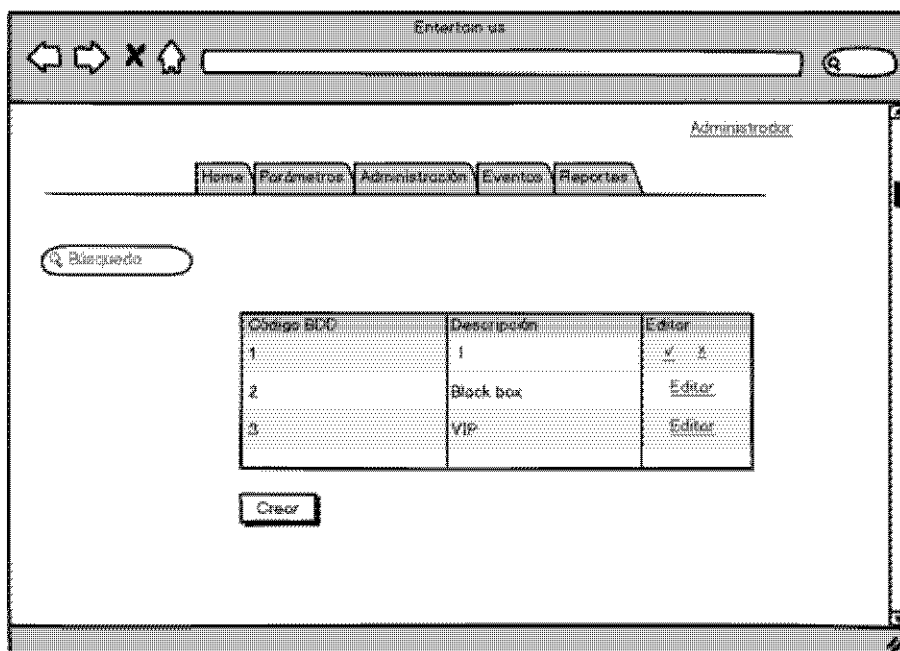
Página de Tipo de Asiento



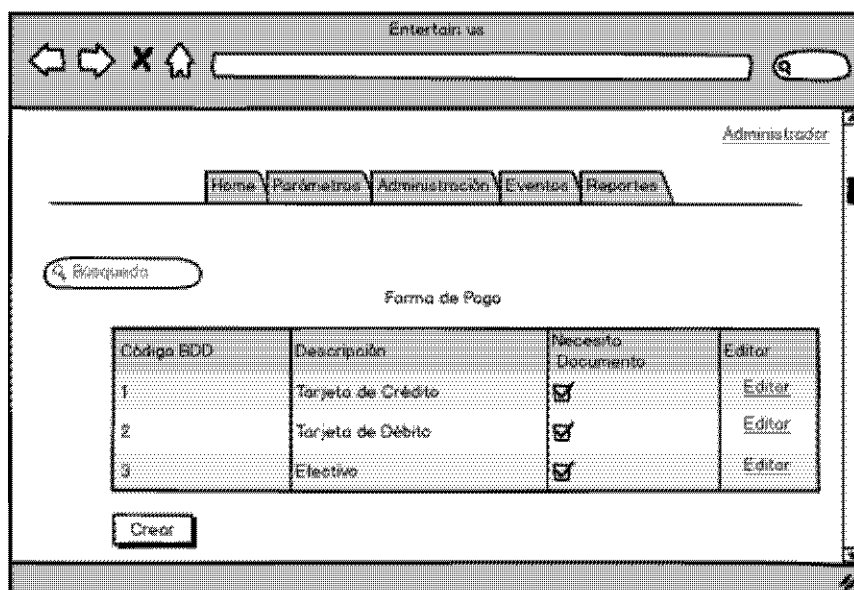
Página de Creación de Tipo de Asiento



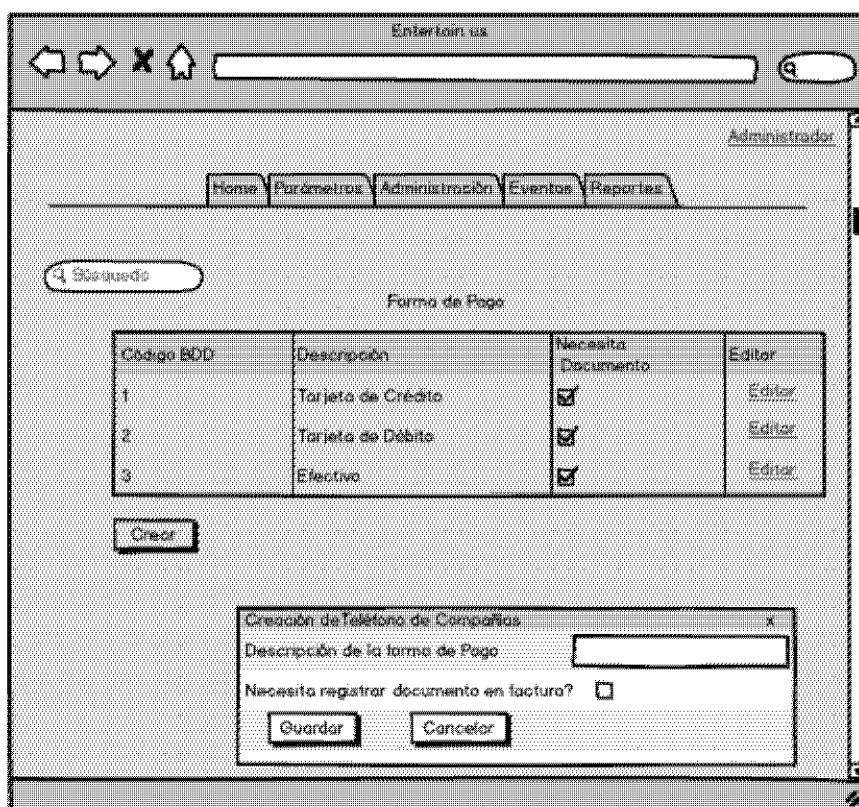
Página de Modificación de Tipo de Asiento



Página de Tipo de Pago



Página de Creación de Tipo de Pago



Página de Modificación de Tipo de Pago

Entibertoan.us

Administrador

Home Parámetros Administración Eventos Reportes

Suscrito

Forma de Pago

Código BDD	Descripción	Necesita Documento	Editar
1	Tarjeta de Crédito	<input checked="" type="checkbox"/>	Editar
2	?	<input type="checkbox"/>	✓ ✕
3	Electivo	<input checked="" type="checkbox"/>	Editar

Crear

Página de Modificación de Datos de la Compañía

Administrador

Inicio | Administración | Administración de Usuarios | Administración de Compañías

Inicio > Administración de Compañías > Modificar Datos de la Compañía

Datos Generales

No. de Identificación: Razón Social:

Direcciones de la Compañía				
Código principal	Número	Calle	Localización	Editar
001	1234	Avenida	Caracas, Venezuela	Editar
				Editar
				Editar

Telefonos de la Compañía

Número de Teléfono	Editar
0212 12345678	Editar
0212 98765432	Editar
	Editar

Datos Especiales

Guionera Gráfica:

Logo de la Empresa:

Actualizar Logo

Descripción de Firma Electrónica	Fecha	Estado	Comprobador
	12/12/2012	Activo	

Logo:

Datos Locales

Locales de la Compañía		
Nombre Localidad	Código de la Localidad	Mensajes de Datos

Página de Creación de direcciones de la Compañía

Administrador

Inicio Parámetros Administración Clientes Reportes

Buscar

Datos de la Compañía

Datos Generales

No. de identificación: 123456789 Razón Social: Fujá Theater

Direcciones de la Compañía

Calle principal	Número	Calle Secundaria	Comentarios	Acción
				Editar
				Editar
				Editar

Creación de Direcciones de Compañías

Calle Principal:

Número:

Calle Secundaria:

Comentarios:

Acción:

Datos Específicos:

Datos Localidades:

Página de Modificación de Direcciones de Compañía

Administrador

Inicio Parámetros Administración Clientes Reportes

Buscar

Datos de la Compañía

Datos Generales

No. de identificación: 123456789 Razón Social: Fujá Theater

Direcciones de la Compañía

Calle principal	Número	Calle Secundaria	Comentarios	Acción
				Editar
				Editar
				Editar

Creación de Direcciones de Compañías

Calle Principal:

Número:

Calle Secundaria:

Comentarios:

Acción:

Datos Específicos:

Datos Localidades:

Página de Creación de Teléfonos de Compañía

Administrador

Inicio | Parámetros | Administración | Usuarios | Reportes

Inicio

Datos de la Compañía

Datos Generales

No. de Identificación: 1234567890 Registro Social: FULCRUM

Direcciones de la Compañía

Calle principal	Número	Calle Secundaria	Comentarios	Editar
				Editar

Crear Dirección

Teléfonos de la Compañía

Número de Teléfono	Editar
1234567890	Editar
	✓ ✗
	Editar

Crear Teléfono

Datos Especiales

Datos Localidades

Guardar Datos de Compañía

Página de Modificación de Teléfonos de Compañía

Administrador

Inicio | Parámetros | Administración | Usuarios | Reportes

Inicio

Datos de la Compañía

Datos Generales

No. de Identificación: 1234567890 Registro Social: FULCRUM

Direcciones de la Compañía

Calle principal	Número	Calle Secundaria	Comentarios	Editar
				Editar

Crear Dirección

Teléfonos de la Compañía

Número de Teléfono	Editar
1234567890	Editar
	✓ ✗
	Editar

Crear Teléfono

Datos Especiales

Datos Localidades

Guardar Datos Compañía

Página de Administración de Noticias

Administración

Home Parámetros Administración Eventos Reportes

Búsqueda

Fecha de Publicación

Desde:

Hasta:

Fecha de Caducidad

Desde:

Hasta:

Buscar Limpiar

Genero

Código BCID	Descripción	Fecha de Publicación	Fecha de Caducidad	Imagen	Editor
1	Prueba1	26/04/2013	29/04/2013		Editor
2	Prueba2	30/04/2013	02/05/2013		Editor

Crear

Página de Creación y Edición de Noticias

Administración

Home Parámetros Administración Eventos Reportes

Búsqueda

Datos de la Noticia

Fecha de Creación: 24/04/2013

Fecha de Caducidad:

Descripción:

Subir Imagen

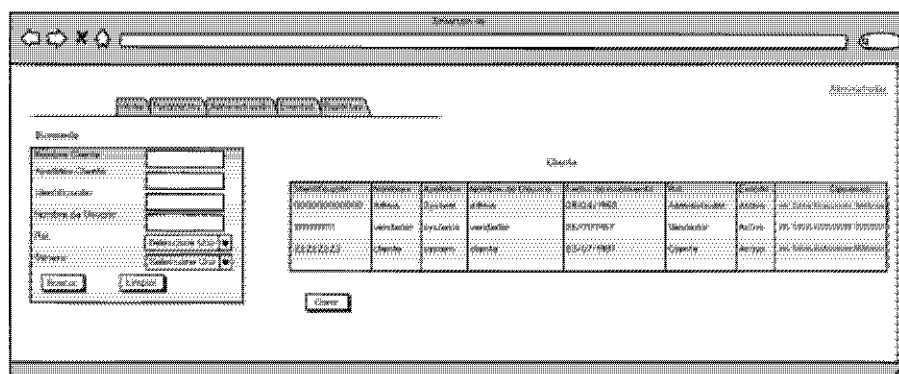
+ Choose

Promocion de Cine Urbano 1.62KB Upload Cancelar

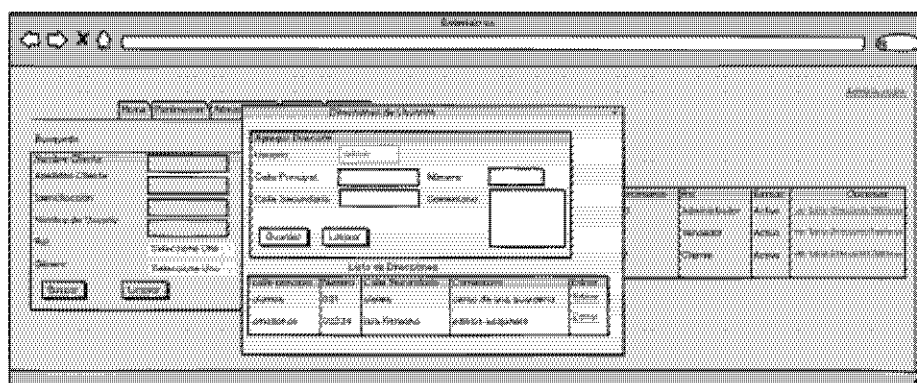
Imagen de la Noticia

Guardar Cancelar

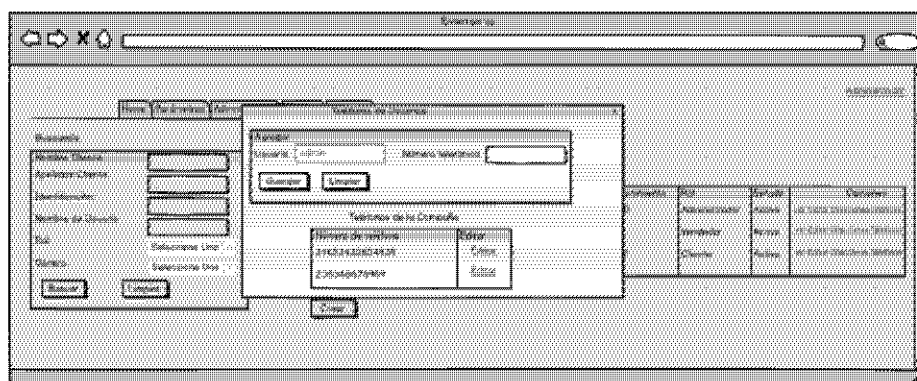
Página de Administración de Usuarios



Página de Creación y Edición de Direcciones de Usuario



Página de Creación y Edición de Teléfonos de Usuario



Página de Modificación de Usuarios

Entrenar us

Administrador

Home Parámetros Administración Eventos Reportes

Búsqueda

Nombre Cliente:

Apellido Cliente:

Identificación:

Nombre de Usuario:

Rol:

Género:

Selección Libre

Selección Uno

Buscar Limpiar

Creación de Usuario

Identificación: Email:

Nombre: Apellido:

Fecha de Nacimiento: Género:

rol: Nombre de Usuario:

Activo:

Guardar Limpiar

Identificación	Rol	Estado	Opciones
	Administrador	Activo	ver Editar Desactivar Eliminar
	Vendedor	Activo	ver Editar Desactivar Eliminar
	Cliente	Activo	ver Editar Desactivar Eliminar

Página de Visualización de Usuario

Entrenar us

Administrador

Home Parámetros Administración Eventos Reportes

Búsqueda

Nombre Cliente:

Apellido Cliente:

Identificación:

Nombre de Usuario:

Rol:

Género:

Selección Libre

Selección Uno

Buscar Limpiar

Creación de Usuario

Identificación: Email:

Nombre: Apellido:

Fecha de Nacimiento: Género:

rol: Nombre de Usuario:

Activo:

Guardar Limpiar

Identificación	Rol	Estado	Opciones
	Administrador	Activo	ver Editar Desactivar Eliminar
	Vendedor	Activo	ver Editar Desactivar Eliminar
	Cliente	Activo	ver Editar Desactivar Eliminar

Página de Administración de Localidades

Entrenar us

Administrador

Home Parámetros Administración Eventos Reportes

Búsqueda

Búsqueda

Nombre de la localidad:

Buscar Limpiar

Localidad: Fecha de la localidad: Número de esta:

Next

Página de Creación de Localidad - Datos

The image shows a web browser window with the address bar containing 'Entertain us'. The browser's navigation buttons (back, forward, home, refresh) are visible on the left. The page content includes a top navigation menu with 'Home', 'Parámetros', 'Administración', 'Eventos', and 'Reportes'. A 'Busqueda' section is present, with sub-tabs for 'localidad', 'datos de contacto', 'salas', and 'confirmación'. The 'localidad' tab is active, displaying a form titled 'Datos de Localidad'. The form contains the following fields:

- Compañía:** A text input field containing the value 'Fulltheater'.
- Nombre de localidad:** An empty text input field.
- Activa:** A checkbox that is currently unchecked.

A 'Next' button is located at the bottom right of the form area. The browser's status bar at the bottom right shows a double-slash icon.

Página de Creación de Localidad – Datos de Contacto

Entertain us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

localidad datos de contacto salas confirmación

Direcciones

calle principal Número

calle secundaria Comentario

telefonos

Número telefónico:

Guardar teléfono Limpiar

Teléfonos de la localidad

número de teléfono	editar
<input type="text"/>	<input type="text"/>

Back Next

Página de Creación de Localidad – Creación de Salas

Administrador

Inicio | Parámetros | Administración | Eventos | Reportes

Búsqueda

Información de sala

nombre de la sala: Hora de apertura: tiempo entre sesiones:
 Número de filas en la sala: Número de columnas en la sala: capacidad:

Salas

salas de la localidad

número de sala	hora apertura	tiempo sesiones	número filas	número columnas	capacidad	acciones
sala 1	09:30	15	4	10	60	<input type="button" value="editar sala"/> <input type="button" value="eliminar sala"/> <input type="button" value="confirmar sala"/>

Página de Creación de Localidad - Confirmación

Administrador

Inicio | Parámetros | Administración | Eventos | Reportes

Búsqueda

Nombre de la localidad:

Confirmación

Compañía: Nombre de la localidad: Activo:

Dirección

calle principal: Número: calle secundaria:

Comentarios:

Teléfonos

Número de teléfono:

Salas

salas de la localidad

número de sala	hora apertura	tiempo sesiones	número filas	número columnas	capacidad	acciones
sala 1	09:30	15	4	10	60	<input type="button" value="editar sala"/> <input type="button" value="eliminar sala"/> <input type="button" value="confirmar sala"/>
sala 2	10:30	15	7	10	70	<input type="button" value="editar sala"/> <input type="button" value="eliminar sala"/> <input type="button" value="confirmar sala"/>
sala 3	11:30	15	4	10	60	<input type="button" value="editar sala"/> <input type="button" value="eliminar sala"/> <input type="button" value="confirmar sala"/>

Página de Modificación de Localidad - Selección

Entertan.us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

Busqueda
Nombre de la localidad: Aloha

Buscar Limpiar

localidad datos de contacto seias confirmación

Nombre de la localidad	estado de la localidad	número de esta
Aloha	activo	3

Next

Página de Modificación de Localidad – Datos de Contacto

Entertan.us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

Busqueda
Nombre de la localidad:

Buscar Limpiar

localidad datos de contacto seias confirmación

Direcciones

calle principal: Phoenix Número: a129

calle secundaria: Alejes Comentario: diagonal a guardería

teléfonos

Número telefónico:

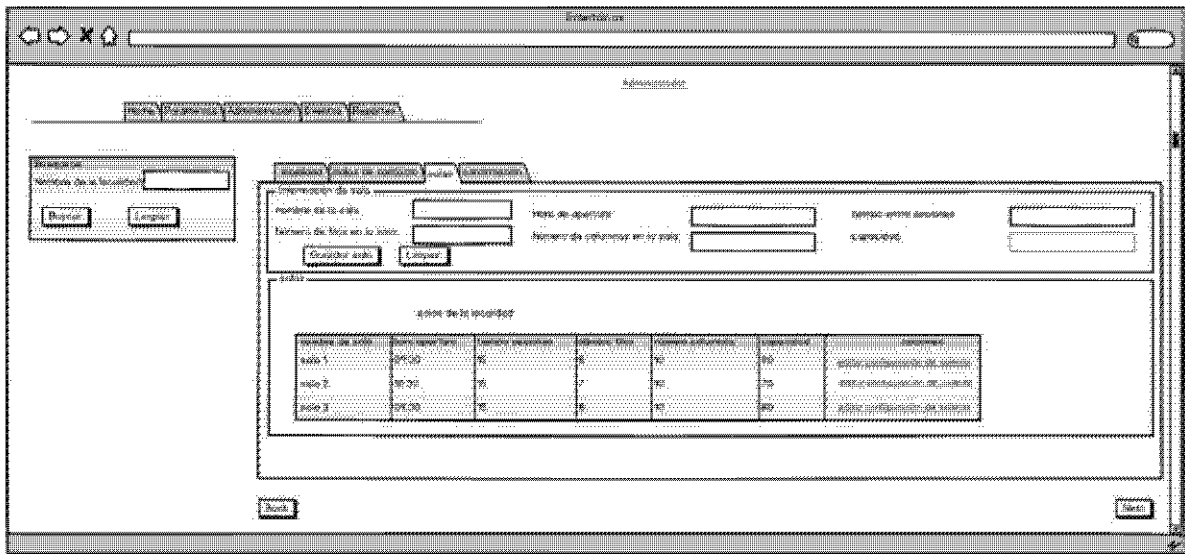
Guardar teléfono Limpiar

Teléfonos de la localidad

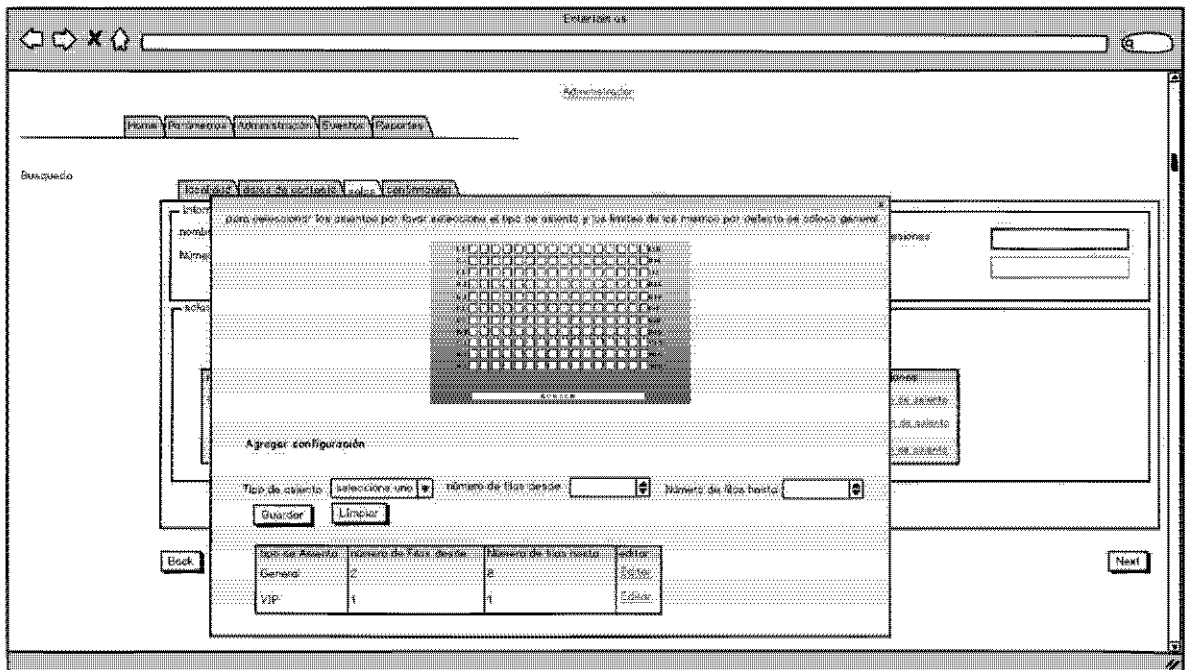
número de teléfono	editar
12324525435	Editar
56789907654	Editar

Back Next

Página de Modificación de Localidad – Modificación y Creación de Salas



Página de Modificación y Creación de Localidad – Configuración de Asientos



Página de Modificación de Localidad - Confirmación

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

Busqueda
Nombre de la localidad:
Buscar Limpiar

Localidad datos de contacto salas confirmación

Confirmación

Compañía: Nombre de localidad: Activa

Dirección

calle principal: Número: calle secundaria:

Comentario:

Teléfonos

número de teléfono:
12324325435
55789407654

Salas salas de la localidad

nombre de sala	hora apertura	tiempo sesiones	número días	número sesiones	capacidad
sala 1	09:30	15	8	10	20
sala 2	10:30	15	7	10	30
sala 3	09:30	15	8	10	50

Grabar

Página de Administración de Sala

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

Busqueda
Nombre de sala:
capacidad sala:
localidad:
Buscar Limpiar

salas editar salas administración

nombre de sala	hora de	tiempo	número	número	capacida	localida
sala 1	09:30	15	8	10	80	alfoha
sala 2	09:30	15	8	10	80	alfoha
sala 3	09:30	15	7	10	70	alfoha

Next

Página de Modificación de Sala

Comercio.es

Inicio Parámetros Administración Eventos Reservas

Búsqueda

Nombre de sala:

Identificador sala:

Localidad:

Tipo sala:

Búsqueda

Información de sala

Nombre de la sala:

Número de sillas en la sala:

Número de asientos:

Número de columnas en la sala:

Espacio entre sillas:

Capacidad:

Configuración de asientos

Borrar

Página de Modificación para la configuración de asientos

Comercio.es

Inicio Parámetros Administración Eventos Reservas

Configuración de asientos

Nombre de la sala:

Número de sillas en la sala:

Número de asientos:

Número de columnas en la sala:

Espacio entre sillas:

Capacidad:

Configuración de asientos

Borrar

Tipo de asiento	Número de Asientos	Número de filas Asiento
Block seat	1	1
Vip	2	2
Normal	3	3

Página de Confirmación de Sala

Confirmación

Administrador

Inicio Parámetros Administración Eventos Reportes

Administración

Inicio Administración Confirmación Configuración

Localidad: Código de la sala: Piso de referencia:

Número de sillas en la sala: Número de categorías en la sala:

Configuración de sillas

Tipo de sillas	Número de sillas	Número de filas
Sillas	15	3
Sillas	15	3

Guardar Cancelar

Volver

Página de Administración de eventos - Creación

Administración de eventos

Administrador

Inicio Parámetros Administración Eventos Reportes

Administración de eventos

Inicio Administración de eventos Creación Actualización de eventos Configuración

Título: Categoría: Dirección:

Límite de edad: Página web: Página web del mapa:

Presentar en el sitio:

Descripción:

Guardar Cancelar

Volver

Página de Creación de eventos - Precios

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

eventos precios colocar en salas confirmación

precio del evento

Tipo de asiento: General valor: 4

Guardar precio Limpiar

precios del evento

tipo de asiento	precio	editar
VIP	10.0	editar
general	4.0	editar

Back Next

Página de Creación de eventos – Colocar en Sala

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

eventos precios colocar en salas confirmación

horarios en salas

localidad: aloha sala: sala 3 Fecha de estreno: 21/05/2013

fecha de término: 25/05/2013 hora de inicio: 11:00

Guardar Horario Limpiar

Horarios del Evento

localidad	sala	fecha de estreno	fecha de término	hora de inicio	editar
aloha	sala 3	21/05/2013	25/05/2013	11:00	editar

Back Next

Página de Creación de eventos - Confirmación

Entertain us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

eventos precios colocar en salas confirmación

datos del evento

Título: House of Mickey Mouse categoría: Infantil duración: 1:30
 Limite de edad: 0 página web: http://www.disney.com/ página web del trailer: http://www.disney.com/
 es un evento próximo: presentar en el inicio:

descripción:
 película infantil

poster:

precios

tipo de asiento	precio
VIP	10.0
general	4.0

horarias

localidad	sala	fecha de estreno	fecha de termino	hora de inicio
aloha	sala 3	21/05/2013	26/06/2013	11:00

Guardar

Back

Página de Modificación de Evento - Selección

Entertain us

Administrador

Home Parámetros Administración Eventos Reportes

Busqueda

título del evento:
 limite de edad:
 categoría: terror

buscar limpiar

título del evento	duración	descripción	categoría	limite de edad	proximamente	editar
Silent Hill Revelaton 3D	02:05	no apta para cardiacos	terror	15	<input type="checkbox"/>	editar

Página de Modificación de Evento – Datos de Evento

Editar evento

Administrador

Home Parámetros Administración Eventos Reportes

editar evento

Título: categoría: Subir poster del evento

Duración: Límite de edad:

Página web: Página Web del trailer:

es un próximo evento: presentar en el inicio:

descripcion:

poster

Página de Modificación de Horario de Evento - Selección

Editar evento

Administrador

Home Parámetros Administración Eventos Reportes

Administración

Busqueda

título del evento:

límite de edad:

categoría:

título del evento	duración	descripcion	categoría	límite de edad	proximamente	editar
Silent Hill Revelation 3D	02:05	no apto para niños	terror	15	<input type="checkbox"/>	editar

Página de Modificación de Horario de Evento - Datos

editar evento

Localidad: sala: fecha de estreno:

fecha de termino: hora de inicio:

dia	sala	fecha de estreno	fecha de termino	hora de inicio	editar
dia	sala 1	25/04/2013	08/05/2013	17:30	Editar
dia	sala 2	25/04/2013	08/05/2013	20:30	Editar
dia	sala 1	25/04/2013	08/05/2013	21:15	Editar
dia	sala 2	25/04/2013	08/05/2013	22:50	Editar

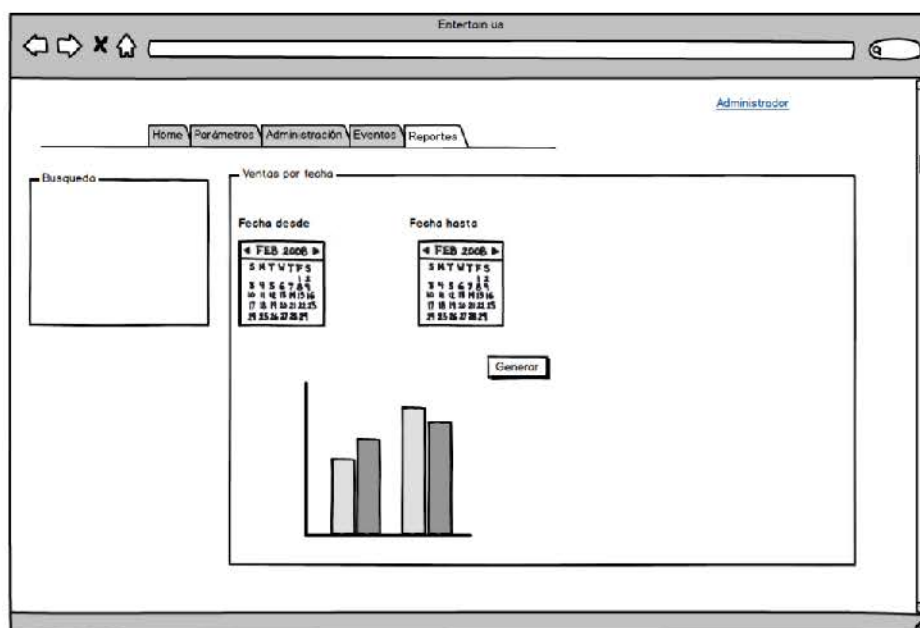
Página de Modificación de Precios de Evento - Datos

editar tipo de evento

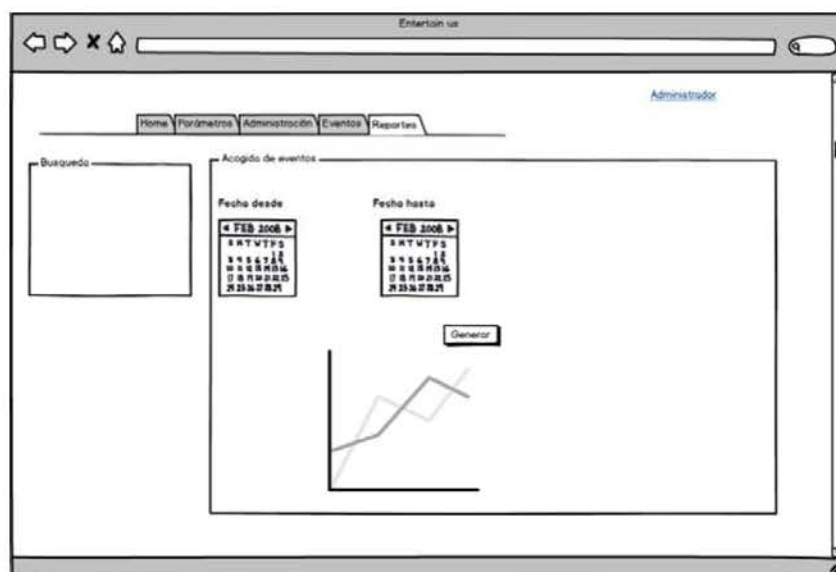
tipo de evento: valor:

tipo de evento	precio	editar
vip	10.0	Editar
General	4.0	Editar

Página de Reportes - Venta por fechas



Página de Reportes - Acogida de eventos



Rol: Usuario Cliente

Reservación de Evento - Selección

Entertain us

Cliente

Home Noticias Eventos Proximos Eventos Quienes Somos reservación

Evento Sala calendario

Hora

Procesar

Asientos

Siguiete

Reservación de Evento – Vista Previa de Reservación

Entertain us

Cliente

Home Noticias Eventos Proximos Eventos Quienes Somos reservación

Producto (evento)	Sala/Tipo de Asiento/Asiento	Hora	Fecha	Precio	Eliminar
					Eliminar
					Eliminar
					Eliminar
					Eliminar

Tota

atras Siguiete

Reservación de Evento - Datos de facturación

Entertan us

Clientes

Nombre: Apellido:

Dirección:

Calle Principal:

Calle secundaria:

Numeración:

Comentarios:

Fecha de Nacimiento: / /

E-mail:

Sexo:

Reservación de Evento – Impresión de Factura

Entertan us

Clientes

Logo de la empresa:

Número de factura: Tipo de Factura:

Razón Social: Estado:

Vendedor:

Nombre: Apellido:

Número de ID:

Dirección: Teléfono:

Producto (evento)	Sala/ Tipo de Asiento/ Asiento	Hora	Fecha	Precio
Total:				<input type="text"/>

Rol: Usuario Vendedor

Página de Venta de Boletos – Selección

Entertain us

Vendedor

Home Noticias Eventos **Proximos Eventos** Quienes Somos Ventas Pago de reservaciones

Evento: Sala: calendario: FEB 2008

Hora:

Asientos

Página de Venta de Boletos – Vista Previa de Compra

Entertain us

Cliente

Home Noticias Eventos Proximos Eventos Quienes Somos Ventas **Pago de reservaciones**

Producto (evento)	Sala/Tipo de Asiento/Asiento	Hora	Fecha	Precio	Eliminar
					Eliminar
					Eliminar
					Eliminar
					Eliminar
Total					

Página de Venta de Boletos – Impresión de Factura

Entertain us

Vendedor

logo de la empresa

Número de factura: Tipo de Factura:

Estado:

Razón Social:

Vendedor:

Nombres: Apellidos:

Número de ID: Teléfono:

Dirección:

Producto (evento)	Sala/Tipo de Asiento/Asiento	Hora	Fecha	Precio
Total				

imprimir recargar página

Página de Pago de Reservaciones - Selección

Entertain us

Vendedor

Home Noticias Eventos Proximos Eventos Quienes Somos Ventas Pago de reservaciones

Número de factura

Buscar

número de factura	Nombre Cliente	Identificación cliente	total

Pagar

Página de Pago de Reservas – Asociación de Pago

Entertain us

Vendedor

Home Noticias Eventos Proximos Eventos Quienes Somos Ventas Pago de reservas

Número de factura

Forma de pago ComboBox

Número de Documento

número de fac

Aceptar cancelar

Pagar

Anexo 14: Casos de Uso

Especificación Caso de Uso: Administración de Género

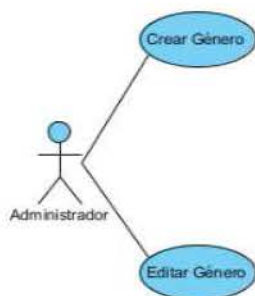
1. Administrar género

1.1 Descripción

Permite administrar los géneros para el registro de información que lo requiera dentro del sistema. Este caso de uso posee dos sub flujos la creación y la edición de este parámetro.

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo básico

1. El caso de uso inicia luego de que un usuario de tipo Administrador accede a la sección de Género
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Una tabla con los siguientes datos:
 - Código BDD
 - Descripción
 - Opción de Edición
 - Opción de Creación

1.4.1.1 Creación de Género

1. Este sub flujo comienza cuando el Administrador selecciona la opción Crear.

2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Descripción del Género
 - Opción para Guardar
 - Opción para Cancelar
 3. El Administrador coloca la información de la descripción.
 4. Selecciona la opción de Guardar.
 5. El sistema guarda la información.
 6. La información se muestra en la tabla.
- 1.4.1.2 Modificación de Género
1. Este sub flujo comienza cuando el Administrador selecciona la opción editar en cualquier registro de la tabla.
 2. El sistema habilita los campos y opciones dentro de la tabla, donde se visualiza lo siguiente:
 - Descripción del Género
 - Opción para Guardar
 - Opción para Cancelar
 3. El Administrador modifica la información de la descripción.
 4. Selecciona la opción de Guardar.
 5. El sistema guarda la información.
 6. La información se muestra en la tabla.

Especificación Caso de Uso: Administración de categorías

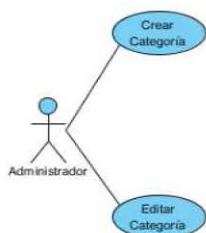
1. Administrar categorías

1.1 Descripción

Permite administrar los diferentes tipos de categorías de los eventos que se presentan

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo básico

1. El caso de uso inicia luego de que un usuario de tipo Administrador accede a la sección de Categorías
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Una tabla con los siguientes datos:
 - Código BDD
 - Descripción
 - Opción de Edición
 - Opción de Creación

1.4.1.1 Crear Categoría

1. Este sub flujo comienza cuando el Administrador selecciona la opción Crear.
2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Descripción de la Categoría
 - Opción para Guardar
 - Opción para Cancelar
3. El Administrador coloca la información de la descripción.
4. Selecciona la opción de Guardar.
5. El sistema guarda la información.
6. La información se muestra en la tabla.

1.4.1.2 Modificar Categoría

1. Este sub flujo comienza cuando el Administrador selecciona la opción editar en cualquier registro de la tabla.
2. El sistema habilita lo campos y opciones dentro de la tabla, donde se visualiza lo siguiente:
 - Descripción de la Categoría
 - Opción para Guardar
 - Opción para Cancelar
3. El Administrador modifica la información de la descripción.
4. Selecciona la opción de Guardar.
5. El sistema guarda la información.
6. La información se muestra en la tabla.

Especificación Caso de Uso: Administración de tipos de asiento

1. Administrar tipos de asiento

1.1 Descripción

Permite administrar los diferentes tipos de asientos de las diferentes salas

1.2 Diagramas

1.2.1 Casos de Uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo básico

1. El caso de uso inicia luego de que un usuario de tipo Administrador accede a la sección de Tipo de Asiento
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Una tabla con los siguientes datos:
 - Código BDD
 - Descripción
 - Opción de Edición
 - Opción de Creación

1.4.1.1 Crear Tipo de Asiento

1. Este sub flujo comienza cuando el Administrador selecciona la opción Crear.
2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Descripción del Tipo de Asiento
 - Opción para Guardar
 - Opción para Cancelar
3. El Administrador coloca la información de la descripción.
4. Selecciona la opción de Guardar.
5. El sistema guarda la información.
6. La información se muestra en la tabla.

1.4.1.2 Modificar Tipo de Asiento

1. Este sub flujo comienza cuando el Administrador selecciona la opción editar en cualquier registro de la tabla.
2. El sistema habilita los campos y opciones dentro de la tabla, donde se visualiza lo siguiente:
 - Descripción del Tipo de Asiento
 - Opción para Guardar
 - Opción para Cancelar
3. El Administrador modifica la información de la descripción.
4. Selecciona la opción de Guardar.
5. El sistema guarda la información.

6. La información se muestra en la tabla.

Especificación casos de Uso: Administración de Forma de Pago

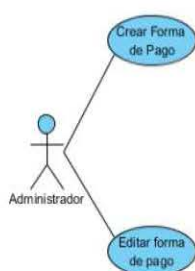
1. Administrar Forma de Pago

1.1 Descripción

Permite administrar las diferentes formas de pago del cliente.

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo básico

1. El caso de uso inicia luego de que un usuario de tipo Administrador accede a la sección de Forma de Pago
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Una tabla con los siguientes datos:
 - Código BDD
 - Descripción
 - Indica si necesita registrar un número de documento mediante una opción de si o no
 - Opción de Edición
 - Opción de Creación

1.4.1.1 Crear Forma de pago

1. Este sub flujo comienza cuando el Administrador selecciona la opción Crear.

2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Descripción de la forma de pago
 - Opción de indicar si necesita asociar un documento
 - Opción para Guardar
 - Opción para Cancelar
3. El Administrador coloca la información de la descripción e indica si es necesario un documento.
4. Selecciona la opción de Guardar.
5. El sistema guarda la información.
6. La información se muestra en la tabla.

1.4.1.2 Modificar Forma de pago

1. Este sub flujo comienza cuando el Administrador selecciona la opción editar en cualquier registro de la tabla.
2. El sistema habilita los campos y opciones dentro de la tabla, donde se visualiza lo siguiente:
 - Descripción de la forma de pago
 - Opción de indicar si necesita asociar un documento
 - Opción para Guardar
 - Opción para Cancelar
3. El Administrador modifica la información de la descripción e/o indica si es necesario un documento.
4. Selecciona la opción de Guardar.
5. El sistema guarda la información.
6. La información se muestra en la tabla.

Especificación Caso de Uso: Acciones de Administración de Compañía

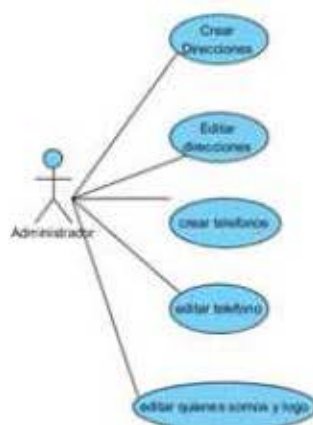
1. Administrar datos Compañía

1.1 Descripción

Permite administrar todos los datos de las compañías organizadoras de eventos

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- **Administrador**

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. El caso de uso inicia luego de que el **Administrador** selecciona la opción **Datos de la compañía**
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - **La primera sección o Datos Generales**
 - **Número de Identificación**
 - **Razón Social**
 - **Direcciones**
 - **Calle Principal**
 - **Número**
 - **Calle Secundaria**
 - **Comentario**
 - **Opción para editar Dirección**
 - **Opción para crear Dirección**
 - **Teléfonos**
 - **Número de Teléfono**
 - **Opción para editar Teléfono**
 - **Opción para crear Teléfono**
 - **Segunda sección o Datos Especificos**

- Información de Quienes Somos
- Logo de la Compañía
- Tercera sección o Localidades
 - Tabla de localidades
 - Nombre de Localidad
 - Estado Localidad
 - Número de Salas
- Guardar Datos de la Compañía

1.4.1.1 Crear Dirección de la Compañía

1. Este sub flujo inicia cuando el Administrador selecciona la opción Crear direcciones
2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Calle Principal
 - Número
 - Calle Secundaria
 - Comentarios
 - Opción para guardar
 - Opción para cancelar
3. El Administrador ingresa todos los datos requeridos por el sistema
4. El Administrador selecciona la opción Guardar
5. El sistema valida que todos los datos obligatorios hayan sido ingresados.
6. El sistema almacena la información.

1.4.1.2 Editar Dirección de la Compañía

1. Este sub flujo inicia cuando el Administrador selecciona la opción Editar en la tabla de direcciones
2. El sistema habilita los campos de la dirección seleccionada
3. El sistema presenta las siguientes opciones:
 - Aceptar
 - Cancelar
4. El Administrador modifica los datos que desee

5. El Administrador selecciona la opción Aceptar
 6. El sistema valida que todos los datos obligatorios hayan sido ingresados.
 7. El sistema almacena la información.
- 1.4.1.3 Crear Números telefónicos de la Compañía
1. Este sub flujo inicia cuando el Administrador selecciona la opción Crear en la tabla de teléfonos
 2. El sistema muestra un modal panel donde se visualiza lo siguiente:
 - Número telefónico
 - Opción para Guardar
 - Opción para Cancelar
 3. El Administrador ingresa todos los datos requeridos por el sistema
 4. El Administrador selecciona la opción Guardar
 5. El sistema valida que todos los datos obligatorios hayan sido ingresados.
 6. El sistema almacena la información.
- 1.4.1.4 Editar Número telefónico de la Compañía
1. Este sub flujo inicia cuando el Administrador selecciona la opción Editar en la tabla de Números telefónicos
 2. El sistema habilita el campo del teléfono seleccionado
 3. El sistema presenta las siguientes opciones:
 - Aceptar
 - Cancelar
 4. El Administrador modifica el teléfono deseado
 5. El Administrador selecciona la opción Aceptar
 6. El sistema valida que todos los datos obligatorios hayan sido ingresados.
 7. El sistema almacena la información.
- 1.4.1.5 Editar sección Quienes Somos y logo de la compañía
1. Este sub flujo inicia cuando el Administrador despliega la pestaña de Datos Específicos

2. El sistema muestra un cuadro de texto en el que el Administrador ingresa información general de la Compañía
3. El sistema muestra una herramienta para subir imágenes donde se presentan las siguientes opciones:
 - Opción escoger imagen
4. El sistema redirecciona a la carpeta de Imágenes del computador
5. El Administrador selecciona el logo que desea publicar
6. El sistema muestra el logo seleccionado en pantalla con las siguientes opciones:
 - Subir
 - Cancelar
7. El Administrador selecciona la opción subir
8. El sistema carga el logo seleccionado por el Administrador
9. Fin del caso de Uso.

Especificación Caso de Uso: Administración de Noticias

1. Administrar Noticias

1.1 Descripción

Permite administrar las diferentes noticias que se publican en el sitio web

1.2 Diagramas

1.2.1 Casos de Uso



1.3 Actores

- **Administrador**

1.4 Flujo de Eventos

1.4.1 Flujo básico

1. El caso de uso inicia luego de que el Administrador selecciona la opción Noticias
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Buscador donde se observan la siguiente información
 - Fecha de publicación: Desde, Hasta.
 - Fecha de caducidad: Desde, Hasta.
 - Opciones:
 - Buscar
 - Limpiar
 - Tabla donde se visualiza los siguiente
 - Código de BDD
 - Descripción
 - Fecha de Publicación
 - Fecha de Caducidad
 - Imagen
 - Opción Editar
 - Opción Crear

1.4.1.1 Crear Noticia

1. Este sub flujo comienza cuando el Administrador selecciona la opción Crear
2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Fecha de creación: se coloca el día en el que el usuario quiere publicar la noticia.
 - Fecha de caducidad
 - Descripción
 - Subir imagen de la Noticia
 - Opción guardar
 - Opción Cancelar

3. El Administrador llena los campos en blanco requeridos por el sistema
4. El Administrador selecciona la opción choose
5. El sistema se direcciona a la carpeta de imágenes del computador
6. El Administrador selecciona la foto que desea subir
7. El sistema carga la foto en pantalla en una herramienta para subir imágenes
8. Se habilitan las opciones de subir y cancelar
9. El Administrador selecciona la opción Subir
10. El Administrador selecciona la opción Guardar
11. El sistema valida que se hayan llenado todos los datos requeridos por el sistema
12. El sistema guarda la información
13. El sistema muestra la información en la tabla.

1.4.1.1 Editar Noticias

1. Este sub flujo comienza cuando el Administrador selecciona la opción Editar
2. El sistema muestra una ventana emergente donde se visualiza lo siguiente:
 - Fecha de creación: campo bloqueado.
 - Fecha de caducidad
 - Descripción
 - Imagen de la noticia
 - Opción guardar
 - Opción cancelar
3. Todos estos campos vienen precargados con los datos de la noticia que se quiere editar
4. El Administrador edita los campos que desee
5. El Administrador selecciona la opción Guardar
6. El sistema valida que se hayan llenado todos los datos requeridos por el sistema
7. El sistema guarda la información

8. El sistema muestra la información en la tabla.

9. Fin del caso de uso.

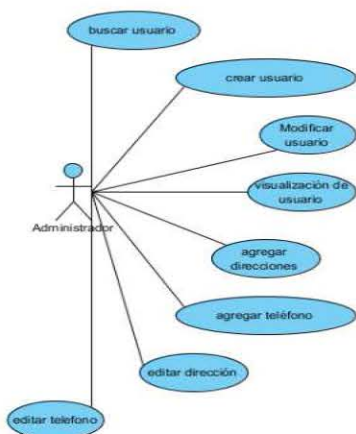
Especificación Caso de Uso: Administrar Usuario

1.1 Descripción

Permite administrar todos los perfiles de cualquier tipo de usuarios del sistema

1.2 Diagramas.

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1.1 Flujo Básico

1. El caso de uso inicia luego de que el Administrador selecciona la opción Usuario
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Buscador donde se observan la siguiente información
 - Nombres del cliente
 - Apellidos del cliente
 - Identificación
 - Nombre de usuario
 - Rol
 - Género

- Opción para buscar
- Opción para limpiar
- Tabla donde se visualiza los siguiente
 - Identificación
 - Nombres
 - Apellidos
 - Nombre de Usuario
 - Fecha de Nacimiento
 - Rol
 - Estado
 - Opción Crear
 - Opción Ver
 - Opción Editar
 - Opción Direcciones
 - Opción Teléfonos

1.4.1.2 Buscar Usuario

1. Este sub flujo inicia después de que el Administrador quiere hacer una búsqueda
2. El Administrador ingresa los datos requeridos en el buscador
3. El Administrador selecciona la opción Buscar
4. El sistema valida que se hayan llenado todos los datos requeridos
5. Para realizar una nueva búsqueda el Administrador selecciona la opción limpiar

1.4.1.3 Crear Usuario

1. Este sub flujo inicia después de que el Administrador selecciona la opción crear
2. El sistema muestra una ventana emergente donde se visualiza los siguiente
 - Identificación
 - Nombres
 - Fecha de Nacimiento

- Rol
 - Estado
 - Email
 - Apellidos
 - Género
 - Nombre de Usuario
 - Opción de Restablecer contraseña
 - Opción Guardar y cancelar
3. El Administrador ingresa todos los datos requeridos por el sistema
 4. El Administrador selecciona la opción Guardar
 5. El sistema valida que se hayan llenado todos los datos requeridos
 6. El sistema guarda la información
 7. El sistema presenta la información creada en pantalla
- 1.4.1.4 Modificar Usuario
1. Este sub flujo inicia después de que el Administrador selecciona la opción editar
 2. El sistema muestra una ventana emergente donde se visualiza los siguiente
 - Identificación
 - Nombres
 - Fecha de Nacimiento
 - Rol
 - Estado
 - Email
 - Apellidos
 - Género
 - Nombre de Usuario
 - Opción de Restablecer contraseña
 - Opción Guardar y cancelar
 3. Todos estos datos vienen cargados con la información que se desea editar
 4. El Administrador modifica los datos que desee

5. El Administrador selecciona la opción Guardar
6. El sistema valida que se hayan llenado todos los datos requeridos
7. El sistema guarda la información
8. El sistema presenta la información creada en pantalla

1.4.1.5 visualización de usuario

1. Este sub flujo inicia cuando el usuario selecciono la opción ver
2. El sistema presenta una ventana emergente con los siguientes datos (estos datos se presentan inhabilitados):
 - Identificación
 - Nombres
 - Fecha de nacimiento
 - Rol
 - Estado
 - E-mail
 - Apellidos
 - Género
 - Nombre de usuario
 - Opciones de: Guardar y cancelar

1.4.1.6 Agregar dirección

1. Este sub flujo inicia después de que el usuario selecciona la opción direcciones
2. El sistema presenta una ventana emergente donde se visualiza lo siguiente:
 - Calle principal
 - Calle secundaria
 - Número
 - Comentario
 - Opción para guardar
 - Opción para Limpiar
3. Una tabla donde se visualizan las distintas direcciones que tiene el usuario con los siguientes datos:
 - Calle principal

- Número
- Calle secundaria
- Comentario
- Opción de Editar

4. El Administrador ingresa los datos que requiere el sistema
5. El Administrador selecciona la opción guardar
6. El sistema valida que todos los campos requeridos estén llenos
7. El sistema guarda la información

1.4.1.7 Agregar teléfono

1. Este sub flujo inicia después de que el usuario selecciona la opción teléfono
2. El sistema presenta una ventana emergente donde se visualiza lo siguiente:
 - Usuario (este campo está deshabilitado)
 - Número telefónico
 - Opción para guardar
 - Opción para limpiar
3. Una tabla donde se visualizan los distintos teléfonos que tiene el usuario, cada uno con opción de editar
4. El Administrador ingresa los datos que requiere el sistema
5. El Administrador selecciona la opción guardar
6. El sistema guarda la información

1.4.1.8 Editar dirección

1. Este sub flujo inicia después de que el usuario selecciona la opción direcciones
2. El sistema presenta una ventana emergente donde se visualiza lo siguiente:
 - Calle principal
 - Calle secundaria
 - Número
 - Comentario
 - Opción para guardar

- Opción para limpiar
3. Una tabla donde se visualizan las distintas direcciones que tiene el usuario con los siguientes datos:
 - Calle principal
 - Número
 - Calle secundaria
 - Comentario
 - Opción de Editar
 4. El Administrador selecciona la opción editar de cualquiera de las direcciones ingresadas
 5. El sistema habilita los campos de la dirección seleccionada para poder ser editados
 6. El Administrador modifica los datos que necesite
 7. El Administrador selecciona la opción guardar
 8. El sistema valida que todos los campos requeridos estén llenos
 9. El sistema guarda la información
- 1.4.1.9 Editar teléfono
1. Este sub flujo inicia después de que el usuario selecciona la opción teléfono
 2. El sistema presenta una ventana emergente donde se visualiza lo siguiente:
 - Usuario (Deshabilitado)
 - Número telefónico
 - Opción para guardar
 - Opción para limpiar
 3. Una tabla donde se visualizan los distintos números telefónicos que la compañía y la opción para editar
 4. El Administrador selecciona la opción editar de cualquier número telefónico ingresado
 5. El sistema habilita los campos de la dirección seleccionada para poder ser editados
 6. El Administrador modifica los datos que requiera

7. El Administrador selecciona la opción guardar
8. El sistema valida que todos los campos requeridos estén llenos
9. El sistema guarda la información

Especificación Caso de Uso: Administración de sala

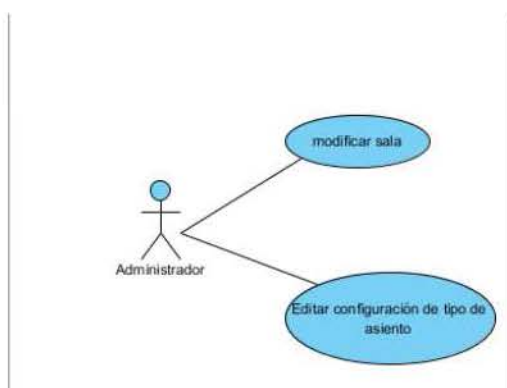
1. Administrar sala

1.1 Descripción

Permite administrar toda la información de las diferentes salas existentes.

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. El caso de uso inicia luego de que el Administrador selecciona la opción de editar sala

1.4.1.1 Modificar sala

1. Este sub flujo inicia después de que el administrador selecciona la opción modificar sala que está dentro de la opción localidad, la que está dentro de la opción Administración
2. el sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. opción de búsqueda
 - i. nombre de la sala

- ii. capacidad de la sala
 - iii. localidad (combobox)
 - b. opciones de:
 - i. salas
 - ii. editar sala
 - iii. confirmación
 - c. lista de las diferentes salas que existen siempre y cuando se haya realizado una búsqueda, caso contrario no se despliega ninguna información
 - d. opción next
- 3. el administrador realiza una búsqueda
- 4. el sistema despliega la tabla con las diferentes salas que tiene esa localidad
- 5. el administrador selecciona la localidad que desea modificar
- 6. el administrador selecciona la opción next
- 7. el sistema muestra la pantalla de Editar sala donde se visualiza lo siguiente
 - a. campos de llenado cargados con la información de la localidad:
 - i. nombre de la sala
 - ii. hora de apertura
 - iii. tiempo entre sesiones (minutos)
 - iv. numero de filas en la sala
 - v. numero de columnas en la sala
 - vi. capacidad: campo deshabilitado se despliega dependiendo del número de filas y columnas que se asigne a la sala
 - b. opción de:
 - i. configuración de asientos
 - ii. next
 - iii. back
- 8. el usuario selecciona la opción configuración de asientos
- 9. el sistema despliega un pop up donde se visualiza lo siguiente:

- a. grafico donde se visualiza la organización de los asientos en la localidad
 - b. combo box tipo de Asiento
 - c. scrollbar:
 - i. selección de filas desde
 - ii. selección de filas hasta
 - d. tabla donde se muestran los tipos de asientos creados hasta el momento
 - i. opción editar tipo de asiento
10. el administrador edita la información requerida por el sistema
11. el sistema valida que todos los campos de llenado estén en el formato adecuado
12. el sistema valida que todos los datos requeridos se hayan llenado
13. el administrador selecciona la opción next
14. el sistema muestra la pantalla de confirmación donde se visualiza lo siguiente:
- a. campos deshabilitados:
 - i. localidad
 - ii. nombre de la sala
 - iii. hora de apertura
 - iv. tiempo entre sesiones (minutos)
 - v. número de filas en la sala
 - vi. número de columnas en la sala
 - vii. capacidad
 - b. tabla donde se muestra la configuración de los asientos
 - c. opciones de:
 - i. grabar
 - ii. back
15. el administrador selecciona la opción grabar
16. El sistema guarda todos los datos que se muestran en la pantalla además de crear en la base de datos los registros de los asientos de cada sala que se ha creado. Si se llegan a reducir el número de asientos

de alguna localidad, el sistema además de actualizar los datos también borra los registros de esos asientos en la base de datos.

1.4.1.2 Editar configuración de tipo de asiento

1. Este sub flujo inicia cuando el administrador selecciono la opción de configuración de asiento
2. El sistema habilita todos los campos para editar
3. El sistema habilita la opción de aceptar y cancelar
4. El usuario edita la información
5. El sistema valida que todos los datos estén en el formato adecuado
6. El sistema valida que todos los datos requeridos por el sistema se hayan ingresado
7. El administrador selecciona la opción aceptar
8. El sistema guarda la información editada

Especificación Caso de Uso: Administración de eventos

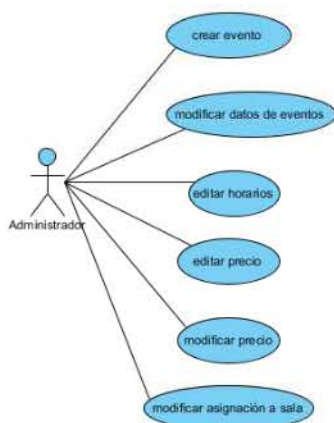
1. Administrar eventos

1.1 Descripción

Permite administrar toda la información de los eventos disponibles en las diferentes localidades existentes

1.2 Diagramas.

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. El caso de uso inicia luego de que el Administrador selecciona la opción eventos
2. El sistema muestra el menú de eventos:
 - a. Crear evento
 - b. Modificar datos de Evento
 - c. Modificar precios
 - d. Modificar Asignación a salas

1.4.1.2 Crear evento

- Este sub flujo inicia cuando el administrador selecciona la opción crear evento
- el sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. tabs:
 - i. Evento
 - ii. Precios
 - iii. Colocar en sala
 - iv. Confirmación
 - b. Campos de llenado
 - i. Título
 - ii. Categoría
 - iii. Duración
 - iv. Límite de edad
 - v. Página web
 - vi. Página web del tráiler
 - vii. Descripción
 - c. Check box:
 - i. Es un próximo evento
 - ii. Presentar en el inicio
 - d. Opción choose para seleccionar el poster del evento
- El Administrador ingresa todos los datos requeridos por el sistema
- El Administrador selecciona la opción choose

- El sistema se direcciona a la carpeta de imágenes del computador
 - El Administrador selecciona la foto que desea subir
 - El sistema carga la foto en pantalla en una herramienta para subir imágenes
 - Se habilitan las opciones de subir y cancelar
 - El Administrador selecciona la opción Subir
 - El sistema carga la foto seleccionada
 - El sistema valida que todos los datos requeridos se hayan ingresado
 - El sistema valida que todos los datos estén en el formato requerido
 - El administrador selecciona la opción next
 - El sistema presenta la pantalla de precios donde se visualiza lo siguiente:
 - a. Precios del evento
 - i. Tipo de asiento
 - ii. Valor
 - iii. Opciones de:
 - iv. Guardar precio
 - v. Limpiar
 - b. Tabla donde se visualizan los precios asignados a los diferentes asientos
 - c. Opción de editar precios
 - d. Opción back y next
15. El administrador ingresa los datos requeridos por el sistema
16. El sistema valida que todos los campos requeridos estén llenos
17. El administrador selecciona la opción guardar precio
18. El sistema presenta en pantalla el precio ingresado
19. El administrador selecciona la opción next
20. El sistema muestra la pantalla de colocar en salas donde se visualiza lo siguiente:
 - a. Combos:
 - i. Localidad
 - ii. Sala

- b. Campos de llenado:
 - i. Fecha de estreno
 - ii. Fecha de termino
 - iii. Hora de inicio
- c. Opciones:
 - i. Guardar horario
 - ii. Limpiar
 - iii. Next
 - iv. Back
- d. Tabla donde se visualizan los horarios de los eventos en la respectivas salas
- e. Opción:
 - i. Editar horario

21. El usuario ingresa los datos obligatorios que requiere el sistema
22. El sistema valida que todos los datos obligatorios se hayan ingresado
23. El administrador selecciona la opción guardar horario
24. El sistema presenta en pantalla la información guardado por el administrador
25. El administrador selecciona la opción next
26. El sistema presenta la pantalla de confirmación donde se visualiza lo siguiente:
 - a. Todos los datos guardados anteriormente, todos estos campos están deshabilitados solo se puede visualizar
 - b. Opciones:
 - i. Grabar
 - ii. Back
27. El administrador selecciona la opción grabar
28. El sistema guarda los datos que se visualizan en pantalla

1.4.1.3 Editar precios

1. Este sub flujo inicia cuando el administrador selecciona la opción de editar precio

2. El sistema habilita todos los campos para editar
3. El sistema habilita la opción de aceptar y cancelar
4. El usuario edita la información
5. El sistema valida que todos los datos estén en el formato adecuado
6. El sistema valida que todos los datos requeridos por el sistema se hayan ingresado
7. El administrador selecciona la opción aceptar
8. El sistema guarda la información editada

1.4.1.4 Editar horarios

1. Este sub flujo inicia cuando el administrador selecciona la opción de editar horarios
2. El sistema habilita todos los campos para editar
3. El sistema habilita la opción de aceptar y cancelar
4. El usuario edita la información
5. El sistema valida que todos los datos estén en el formato adecuado
6. El sistema valida que todos los datos requeridos por el sistema se hayan ingresado
7. El administrador selecciona la opción aceptar
8. El sistema guarda la información editada

1.4.1.5 Modificar Datos de evento

1. Este sub flujo inicia cuando el administrador selecciona la opción Modificar datos de evento
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. opción de búsqueda donde se visualizan los siguientes datos:
 - i. Título del evento
 - ii. Límite de edad
 - iii. Categoría
 - b. Opciones:
 - i. Buscar
 - ii. Limpiar

3. El Administrador ingresa los datos requeridos por el sistema
4. El Administrador selecciona la opción buscar
5. El sistema presenta en pantalla una tabla donde se visualizan los datos del evento que ingreso en el campo de búsqueda los datos son los siguientes:
 - a. Título del evento
 - b. Duración
 - c. Descripción
 - d. Categoría
 - e. Límite de edad
 - f. Próximamente
 - g. Opción de editar
6. El administrador selecciona la opción de editar
7. El sistema presenta un pop up donde se visualiza lo siguiente:
 - a. Campos de llenado cargados con los datos del evento seleccionado
 - i. Titulo
 - ii. Categoría (combo box)
 - iii. Duración
 - iv. Límite de edad
 - v. Página web
 - vi. Página web del tráiler
 - vii. Es un próximo evento (check box)
 - viii. Presentar en el inicio (check box)
 - ix. Descripción
 - x. Herramienta para cargar imágenes
 - Opción choose
 - xi. Campo de visualización de la imagen cargada
 - xii. Opciones:
 - Guardar
 - Cancelar
8. El administrador modifica los datos que desea
9. El sistema valida que todos los datos obligatorios estén llenos

10. El sistema valida que todos los datos estén en el formato adecuado
11. El administrador selecciona la opción guardar
12. El sistema muestra la información guardada en la tabla

1.4.1.6 Modificar precio

1. Este sub flujo inicia cuando el administrador selecciona la opción Modificar precio
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. opción de búsqueda donde se visualizan los siguientes datos:
 - i. Título del evento
 - ii. Límite de edad
 - iii. Categoría
 - b. Opciones:
 - i. Buscar
 - ii. Limpiar
3. El Administrador ingresa los datos requeridos por el sistema
4. El Administrador selecciona la opción buscar
5. El sistema presenta en pantalla una tabla donde se visualizan los datos del evento que ingreso en el campo de búsqueda los datos son los siguientes:
 - a. Título del evento
 - b. Duración
 - c. Descripción
 - d. Categoría
 - e. Límite de edad
 - f. Próximamente
 - g. Opción de editar
6. El administrador selecciona la opción de editar
7. El sistema presenta un pop up donde se visualiza lo siguiente:
 - a. Campos de llenado cargados con los datos del evento seleccionado

- i. Tipo de asiento (combo box)
 - ii. Valor
 - iii. Opción de Guardar precio
 - iv. Opción de limpiar
 - v. Tabla donde se visualiza el tipo de asiento y el precio
 - vi. Opción de editar precio del evento
 - vii. Opción de Cancelar
8. El administrador ingresa los valores requeridos en la sección de precios
9. El sistema valida que todos los datos requeridos estén llenos y en el formato adecuado
10. El administrador selecciona la opción guardar precio
11. El sistema muestra en la tabla el precio guardado
12. El administrador selecciona la opción de editar precio
13. El sistema habilita todos los campos de la tabla
14. El sistema habilita las opciones de aceptar y cancelar
15. El Administrador selecciona la opción aceptar
16. El usuario selecciona la opción guardar cambios
17. El sistema guarda los cambios hechos por el administrador y cierra el pop up

1.4.1.7 Modificar Asignación a salas

1. Este sub flujo inicia cuando el administrador selecciona la opción Modificar asignación a sala
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - b. opción de búsqueda donde se visualizan los siguientes datos:
 - i. Título del evento
 - ii. Límite de edad
 - iii. Categoría
 - c. Opciones:
 - i. Buscar
 - ii. Limpiar

3. El Administrador ingresa los datos requeridos por el sistema
4. El Administrador selecciona la opción buscar
5. El sistema presenta en pantalla una tabla donde se visualizan los datos del evento que ingreso en el campo de búsqueda los datos son los siguientes:
 - a. Título del evento
 - b. Duración
 - c. Descripción
 - d. Categoría
 - e. Límite de edad
 - f. Próximamente
 - g. Opción de editar
6. El administrador selecciona la opción de editar
7. El sistema presenta un pop up donde se visualiza lo siguiente:
 - a. Asignación de horarios
 - b. Campos de llenado cargados con los datos del evento seleccionado
 - i. localidad(combo box)
 - ii. sala (combo box)
 - iii. fecha de estreno
 - iv. fecha de termino
 - v. hora de inicio
 - vi. opciones:
 - guardar horario
 - limpiar
 - c. tabla donde se visualizan todos los horarios del evento seleccionado
 - d. opción de editar horario
 - e. opción guardar
 - f. opción cancelar
8. El administrador ingresa los valores requeridos en la sección asignación de horarios

9. El sistema valida que todos los datos requeridos estén llenos y en el formato adecuado
10. El administrador selecciona la opción guardar horario
11. El sistema muestra en la tabla el precio guardado
12. El administrador selecciona la opción de editar horario en la tabla
13. El sistema habilita todos los campos de la tabla
14. El sistema habilita las opciones de aceptar y cancelar
15. El Administrador selecciona la opción aceptar
16. El usuario selecciona la opción guardar cambios
17. El sistema guarda los cambios hechos por el administrador y cierra el pop up

Especificación Caso de Uso: Administración de reportes

1. Reportes

1.1 Descripción

Permite al administrador controlar los reportes de ventas y acogida de los distintos eventos que se presentan en las localidades

1.2 Diagramas

1.2.1 Casos de uso



1.3 Actores

- Administrador

1.4 Flujo de eventos

1.4.1 Flujo básico

1. El caso de uso inicia después de que el administrador selecciona la opción reportes
2. El sistema muestra un menú donde se visualizan las siguientes opciones

- a. Venta por fecha
- b. Acogida de eventos

1.4.1.1 Venta por fechas

1. Este sub flujo inicia después de que el administrador selecciona la opción ventas por fecha
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. calendarios:
 - i. fecha desde
 - ii. fecha hasta
 - iii. opción: generar
3. El administrador selecciona las fechas que desea visualizar
4. El administrador selecciona la opción generar
5. El sistema despliega un gráfico de barras donde indica cuanto se ha vendido dependiendo de la fecha que el administrador haya seleccionado en los calendarios.

1.4.1.2 Acogida de eventos

1. Este sub flujo inicia después de que el administrador selecciona la opción acogida de eventos
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. Calendarios:
 - i. Fecha desde
 - ii. Fecha hasta
 - b. Opción: generar
3. El administrador selecciona las fechas que desea visualizar
4. El administrador selecciona la opción generar
5. El sistema despliega un cuadro de líneas donde indica cuanta acogida han tenido los eventos en las localidades, dependiendo de las fechas que haya seleccionado el administrador en los calendarios.

Especificación Caso de Uso: Reservación de eventos

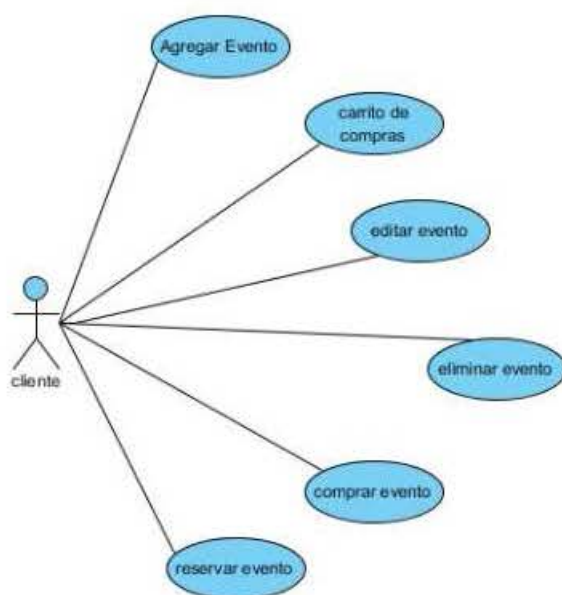
1. Compra de boletos

1.1 Descripción

Permite al usuario reservar los boletos de los eventos en el sistema cuando el usuario se ha ingresado como cliente, se genera una factura en la cual se puede acercar a los locales a cancelar su deuda e ingresar al evento

1.2 Diagramas.

1.2.1 Casos de uso



1.3 Actores

- Cliente

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. el caso de uso inicia luego de logearse como cliente
2. el sistema habilita la pestaña de reservación

1.4.1.1 Agregar reservación de evento

Este sub flujo inicia cuando el cliente ha seleccionado la pestaña reservación.

1. El cliente selecciona la opción Agregar
2. El sistema muestra un pop up donde se visualiza lo siguiente
 - a. Evento: se despliega un combo box
 - b. Sala: se despliega un combo box
 - c. Hora: se despliega en un combo box
 - d. Calendario: un calendario para que el cliente escoja el día en el que quiere asistir al evento.
 - e. Gráfico en el que el cliente selecciona el asiento que desea adquirir para el evento.
 - f. Opción Siguiente.
3. El cliente selecciona el asiento deseado siempre y cuando esté habilitado para su compra.
4. El cliente selecciona la opción siguiente.
5. El sistema guarda la información

1.4.1.2 Visualización de factura previa

1. Este sub flujo inicia cuando el vendedor selecciona la opción siguiente
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. Tabla con los datos de la reservación realizada
 - b. Opción:
 - i. Siguiente
 - ii. Atrás
 - iii. Eliminar reservación
1. Deshacer compra de evento
2. Este sub flujo inicia cuando el cliente ha seleccionado la opción Eliminar
3. El cliente selecciona la opción Eliminar
4. El sistema borra todos los datos del evento seleccionado.

1.4.1.4 Factura

1. Este sub flujo inicia cuando el vendedor selecciona la opción siguiente en la pantalla de visualización de factura previa
2. El sistema muestra toda la información requerida por la factura, estos campos ya no se pueden editar esto solamente es para visualizar la factura definitiva de la reservación.
3. El sistema muestra las opciones de:
 - a. Imprimir: imprime la factura definitiva
 - b. Recargar página: (para poder realizar una nueva reserva)

1.4.1.5 Flujo excepción

Este sub flujo inicia cuando el usuario se logea como cliente

1. el cliente hace click en el nombre de usuario que se muestra en la esquina superior derecha puede editar sus datos personales.
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. Nombre de usuario (Requerido)
 - b. Contraseña (Requerido)
 - c. Nombres (Requerido)
 - d. Apellidos (Requerido)
 - e. Número de ID (Requerido)
 - f. Teléfonos (Al menos uno)
 - g. Dirección: Calle Principal, número, calle secundaria (Requerido)
 - h. E-mail (Requerido)
 - i. Sexo
 - j. Fecha de Nacimiento (Requerido)
3. El cliente selecciona la opción guardar
4. El sistema valida que todos los datos obligatorios hayan sido ingresados
5. El sistema valida que no existan espacios en blanco en ningún campo de llenado
6. El sistema verifica que la información ingresada esté en el formato adecuado
7. Fin del caso de uso

Especificación Caso de Uso: Venta de tickets de eventos

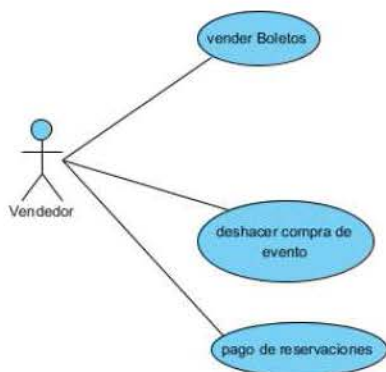
2. Venta de boletos

2.1 Descripción

Permite al vendedor generar la factura de la venta realizada al cliente, solamente de modo presencial.

2.2 Diagramas.

2.2.1 Casos de Uso



1.3 Actores

- Vendedor

1.4 Flujo de Eventos

1.4.1 Flujo Básico

1. El caso de uso inicia luego de logearse como vendedor.
2. en la página principal se habilita la siguiente opción:

- ventas
- pago de reservaciones

1.4.1.1 Vender boletos.

Este sub flujo inicia cuando el vendedor selecciona la opción ventas.

1. El vendedor selecciona la opción vender
2. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - Combobox:
 - a. eventos: el vendedor selecciona el evento al que el cliente desea ir
 - b. Sala: el vendedor selecciona la sala que el cliente desea

c. Hora: el vendedor selecciona la hora del evento en la que el evento esté disponible y el cliente desea ir

- Calendario: fechas en las que está disponible el evento
- Opción de procesar: el vendedor selecciona esta opción para que se despliegue un gráfico en el cual el sistema muestra los asientos disponibles en la sala que el cliente selecciona.

3. El vendedor valida si hay o no asientos disponibles en la sala.
4. El vendedor selecciona otra sala en caso de no existir asientos disponibles
5. El vendedor valida si hay boletos disponibles para la hora y el día seleccionado por el cliente.
6. El vendedor selecciona otra hora y otro día en caso de no existir boletos para la información requerida.
7. El sistema valida que todos los datos obligatorios hayan sido ingresados.
8. Opción siguiente: el vendedor selecciona esta opción después de validar que todos los datos estén correctos.
9. El sistema muestra un modal panel de vista de factura.

1.4.1.2 Vista previa de factura

1. El sistema muestra una pantalla en donde se visualiza lo siguiente:
 - a. Una factura en la cual aparecen los siguientes campos esta factura se presenta en un modal panel:
 - i. Producto (Evento).
 - ii. Sala / tipo de Asiento/Asiento.
 - iii. Hora.
 - iv. Fecha.
 - v. Precio.
 - vi. Total
 - b. Listado de eventos seleccionados.
2. En cada uno de los elementos del listado de eventos seleccionados hasta el momento se muestra las siguientes opciones:

- a. Opción para eliminar un evento seleccionado
3. En la pantalla se visualiza las siguientes opciones
 - a. Atrás
 - b. Siguiente
 - c. Cancelar

1.4.1.3 Deshacer compra de Evento

Este sub flujo inicia cuando el vendedor selecciona la opción Eliminar

1. El vendedor selecciona la opción Eliminar
2. El sistema borra todos los datos del evento seleccionado.

1.4.1.4 Vista de Factura definitiva

Este sub flujo inicia cuando el vendedor selecciona la opción siguiente.

1. El sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. Una factura en la cual aparecen los siguientes campos:
 - i. Producto (Evento).
 - ii. Sala / tipo de Asiento/Asiento.
 - iii. Hora.
 - iv. Fecha.
 - v. Precio.
 - vi. Total
2. En la pantalla se visualizan las opciones de:
 - a. Imprimir:
 - i. El sistema imprime la factura que se visualiza en la pantalla
 - b. Recargar página:
 - i. El sistema recarga la página para hacer una nueva venta
3. Fin de caso de uso

1.4.1.5 Pago de reservaciones

1. Este sub flujo inicia cuando el vendedor selecciona la opción pago de reservaciones

2. el sistema muestra una pantalla donde se visualiza lo siguiente:
 - a. campo de llenado: número de factura
 - b. opción : buscar
3. el vendedor llena los datos requeridos para realizar la búsqueda
4. el sistema despliega la información del número de factura que el vendedor ingreso en el campo de llenado, en la tabla se visualiza lo siguiente:
 - a. Número de factura
 - b. Nombre del cliente
 - c. Identificación del cliente
 - d. Total a pagar
 - e. Opción: pagar
5. El sistema muestra un pop up donde se visualiza lo siguiente:
 - a. Forma de pago (combo box)
 - b. Número de documento
 - c. Opción: aceptar y cancelar
6. El sistema valida que los datos ingresados estén en el formato adecuado
7. El vendedor selecciona la opción guardar
8. El sistema guarda la información.
9. Fin del caso de uso.

Anexo 15: Análisis Económico

La estimación de costos que se va a tener con la implementación, manejo y elaboración del software de manejo de eventos y cine, esta estimación involucra la inversión inicial, el costo de mano de obra directa, la tasa mínima aceptable de retorno, el valor actual neto, tasa interna de retorno, razón beneficio /costo.

Inversión inicial: en esta tabla se puede visualizar cuales son los valores a invertir dependiendo de los activos que se vayan a utilizar en este caso se puede observar que se va a utilizar computadores, muebles y enseres y el software para obtener después una ganancia, al final de todo se observa que la inversión total que se debe hacer para poder implementar el software es de 29.800 usd.

INVERSIÓN INICIAL				
ACTIVOS FIJOS	unidades	valor unitario en USD	total	depreciación
Computadores	2,00	\$1.500,00	\$3.000,00	\$1.000,00
Muebles y Enseres	1,00	\$2.800,00	\$2.800,00	\$580,00
Software	1,00	\$1.500,00	\$1.500,00	\$500,00
SUBTOTAL ACTIVOS FIJOS			\$7.300,00	\$2.080,00

ACTIVOS DIFERIDOS	valor totalizado en USD
Capacitación	\$ 800,00
Gastos Legales de Constitución	\$ 700,00
Viajes (Contactos con Proveedores/clientes)	\$ 1.000,00
SUBTOTAL ACTIVOS DIFERIDOS	\$ 2.500,00
Capital de Trabajo Inicial	\$ 20.000,00

INVERSIÓN TOTAL	\$ 29.800,00
-----------------	--------------

Presupuesto de ventas: en esta tabla se observa la estimación de producción por periodo y asumiendo que se vende el 100% de lo que se elaboró se puede ver que en un periodo de 6 meses los ingresos son mayores a la inversión que se hizo.

PRESUPUESTO DE VENTAS	SE VENDE 100%	SE VENDE 100%	SE VENDE 100%	SE VENDE 100%	SE VENDE 100%
VOLUMEN ESTIMADO DE VENTAS	periodo 1	periodo 2	periodo 3	periodo 4	periodo 5
EN UNIDADES DE PRODUCTO					
SOFTWARE ENTERTAIN US	5,00	6,00	7,00	8,00	9,00
TOTAL PRODUCCIÓN EN UNIDADES	5,00	6,00	7,00	8,00	9,00
PRECIO ESTIMADO EN DÓLARES					
SOFTWARE ENTERTAIN US	\$9.500,00	\$9.500,00	\$10.000,00	\$10.000,00	\$12.000,00
INGRESOS TOTALES	\$47.500,00	\$57.000,00	\$70.000,00	\$88.000,00	\$108.000,00

Costo de mano de obra directa: en esta tabla se puede observar la remuneración que va a tener nuestra mano de obra en este caso serán los programadores, aquí se calcula la remuneración mensual tanto individual como total y la remuneración anual.

MANO DE OBRA DIRECTA				
OBREROS	CANTIDAD	REMUNERACIÓN MENSUAL INDIV.	REMUNERACIÓN MENSUAL TOTAL	REMUNERACIÓN ANUAL
PROGRAMADORES	2	\$ 800,00	\$ 1.200,00	\$ 14.400,00
	0	\$ 0,00	\$ 0,00	\$ 0,00
TOTAL	2,00	\$ 800,00	\$ 1.200,00	\$ 14.400,00

Tasa mínima aceptable de retorno: Se está trabajando con una TMAR Global de 15,00%.

EMPRESA XYZ

TASA MÍNIMA ACEPTABLE DE RETORNO (TASA DE CORTE "K")

período 1

	% de Aporte	TM AR	Ponderación
Accionistas	100,00%	15%	15,00%
Crédito	0,00%	10%	0,00%
TMAR Global			15,00%

Valor actual neto

EMPRESA XYZ

VALOR ACTUAL NETO EN BASE A LA TMAR
--

período 1		
Años	Flujo Operacional	VAN
	TMAR	15,00%
Inversión Inicial	-\$ 29.800,00	-\$ 29.800,00
período 1	\$ 9.154,21	\$ 7.960,18
período 2	\$ 14.776,33	\$ 11.173,03
período 3	\$ 21.906,06	\$ 14.403,59
período 4	\$ 27.799,59	\$ 15.894,51
período 5	\$ 44.492,58	\$ 22.120,68
TOTAL	\$ 88.328,77	\$ 41.751,98
	VAN=	\$ 53.806,08

Tasa interna de retorno: esta tabla nos indica el valor de TIR utilizado que es del 51.21% este valor se calcula con la suma de todos los flujos operativos.

TASA INTERNA DE RETORNO (TIR)	
período 1	
Años	Flujos Operativos
Inversión inicial	-\$ 29.800,00
período 1	\$ 9.154,21
período 2	\$ 14.778,33
período 3	\$ 21.906,06
período 4	\$ 27.799,59
período 5	\$ 44.492,58
TIR	51,21%

Razón beneficio costo en esta tabla se puede observar todos los flujos operativos descontados por periodo, estos datos sumados nos dan como resultado el V.A que es de 71.551.98

RAZÓN BENEFICIO / COSTO	
período 1	
Años	Flujos Operativos descontados
Inversión inicial	-\$ 29.800,00
período 1	\$ 7.960,18
período 2	\$ 11.173,03
período 3	\$ 14.403,59
período 4	\$ 15.894,51
período 5	\$ 22.120,68
Sumatoria de flujos descontados (V.A.)	\$ 71.551,98

R =

$$\frac{\text{V.A.}}{\text{Inversión}}$$

R =

\$ 2,40

Periodo real de recuperación de la inversión: en esta tabla se puede identificar cuanto es lo que se va a recuperar viendo cual es la inversión inicial y la de cada periodo, también los flujos operativos que se tienen por año, para obtener la sumatoria del flujo neto se suma el valor del periodo que se quiere calcular + flujo operativo descontado del periodo anterior, con estos datos se calcula la diferencia entre flujos e inversiones. Dependiendo de estos datos nos sale como resultado el parcial del año sobrante, en este caso se observa que de

0,259 en el periodo 3, y así observamos que el periodo real de recuperación de la inversión es de 3,259 años

PERÍODO REAL DE RECUPERACIÓN DE LA INVERSIÓN					
período 1					
Años	Flujos Operativos descontados	sumatoria flujo neto	Diferencia entre flujos e inversión	parcial del año sobrante	PRI en años
0	Inversión Inicial	-\$ 29.800,00			
1	período 1	\$ 7.960,18	\$ 7.960,18	-\$ 21.839,82	
2	período 2	\$ 11.173,08	\$ 19.133,21	-\$ 10.666,79	
3	período 3	\$ 14.403,59	\$ 33.536,80	\$ 3.736,80	0,259
4	período 4	\$ 15.894,51	\$ 49.431,30	\$ 19.631,30	
5	período 5	\$ 22.120,68	\$ 71.551,98	\$ 41.751,98	
6					
7					
	Total	\$ 41.751,98			

PRI