

# Universidad ORT Uruguay

## Escuela de Tecnología

RSVPDay

Sistema de Gestión de Eventos

Entregado como Requisito para la Obtención del  
Título de Analista Programador

Lasserre, Ximena - 187746

Maisonneuve, Marianne - 188192

Tutor: Rafael Cohen

2016

# Declaración de Autoría

Nosotras, Lasserre, Ximena y Maisonneuve, Marianne, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto final entregado como requisito para la obtención del título de Analista Programador;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuído por otros, y qué fue contribuído por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Ximena Lasserre



Marianne Maisonneuve

Montevideo, 29 de marzo de 2016

## Agradecimientos

Quisiéramos agradecer a nuestra familia y amigos por habernos apoyado durante la realización del proyecto, así como de la carrera y por entender todas las veces que tuvimos que dejar de lado otras actividades para alcanzar esta etapa.

También quisiéramos agradecer a Diego Febles y Angélica Buffa, así como a todo Modelit, por el apoyo durante el transcurso del proyecto.

Otra persona a agradecer es nuestro tutor, Rafael Cohen, por la ayuda, las dudas y preguntas respondidas y la guía que nos dio para realizar este trabajo.

Gracias.

## Abstract

El presente trabajo consiste en la implementación de un sitio de registro de eventos para la empresa Modelit.

El mismo implica un sitio público al que podrá acceder cualquier usuario. Este se conecta con Salesforce, que es desde donde se crean los eventos de cada compañía, para traer la información pertinente a la misma. Y en el sitio el usuario es capaz de registrarse a ellos.

Como objetivos primordiales se encuentran la implementación del sitio público para que se pueda comenzar a usar al final de este proyecto, la investigación de diferentes frameworks y tecnologías como Meteor, AngularJS y MongoDB e integración del mismo con el sitio privado (donde se crean los eventos) y otras tecnologías mediante APIs REST.

Se utilizan Meteor como framework junto con AngularJS y Javascript como principales lenguajes de programación. También MongoDB para la base de datos. Para el versionado y revisión de código se utiliza GitHub.

## Palabras Clave

RSVPDay, Meteor, eventos, Salesforce, gestión de eventos, APIs, MongoDB, AngularJS, Javascript.

# Índice

Declaración de Autoría .....	2
Agradecimientos .....	3
Abstract .....	4
Palabras Clave .....	5
Índice .....	6
Capítulo 1. Introducción .....	11
1.1. Descripción del Cliente .....	11
1.2. Roles y contacto con Modelit .....	12
Capítulo 2. Anteproyecto.....	13
2.1. Análisis .....	13
2.1.1. Descripción del Problema .....	13
2.1.2. Necesidades que Surgen del Problema .....	16
2.1.3. Objetivos para Satisfacer Dichas Necesidades .....	16
2.2. Proyecto .....	17
2.2.1. Descripción del Entorno .....	17
2.2.1.1. Diagrama de Arquitectura del Sistema .....	17
2.2.1.2. Diagrama Conceptual del Dominio .....	21
2.2.1.3. Modelo Entidad Relación.....	23
2.2.2. Análisis de Riesgos .....	23
2.2.3. Casos de Uso.....	25
2.2.3.1. Diagrama de Casos de Uso .....	25
2.2.3.2. Listado de Casos de Uso .....	26
2.2.4. Backlog del Producto.....	26
2.3. Plan de Proyecto .....	27
2.3.1. Descripción de Herramientas y Tecnologías .....	27
2.3.1.1. Tecnologías a Utilizar .....	27
2.3.1.2 Entorno de desarrollo (IDE).....	29
2.3.2. Marco de Trabajo .....	29
2.3.2.1. Comunicación entre las Partes.....	29
2.3.2.2. Herramientas de Comunicación .....	29
2.3.3. SQA .....	30
2.3.3.1. Estándares Definidos y Convenciones .....	30
2.3.4. SCM .....	38

2.3.4.1. SCM en el Desarrollo .....	38
2.3.4.2. SCM de la Documentación.....	39
2.4. Plan de Testing .....	39
2.5. Planificación .....	40
2.5.1. Metodología .....	40
2.5.2. Anteproyecto: Sprint 1 y 2 .....	40
2.5.3. Cronograma de Desarrollo: Sprint 3 al 8.....	41
2.5.4. Diagrama de Gantt .....	43
2.5.5. Estimación General del Proyecto.....	47
Capítulo 3. Evolución del Anteproyecto.....	48
3.1. Descripción del Entorno .....	48
3.1.1. Diagrama Conceptual del Dominio .....	48
3.1.2. Modelo Entidad Relación.....	51
3.1.3. Estructura de los Documentos JSON .....	51
3.1.3.1. Events.....	51
3.1.3.2. Users .....	56
3.1.3.3. Sponsors.....	58
3.1.3.4. Speakers.....	58
3.1.3.5. Venues.....	59
3.1.3.6. Invoices.....	59
3.1.3.7. Refunds .....	60
3.2. Evolución de Herramientas y Tecnologías.....	61
3.3. Plan de Tesing .....	63
3.4. Planificación .....	63
3.4.1. Metodología .....	63
3.4.2. Cronograma de Desarrollo Final: Sprint 3 al 8 .....	64
Capítulo 4. Desarrollo y Conclusiones .....	70
4.1. Conclusiones del Anteproyecto .....	70
4.2. Seguimiento de los Sprints .....	70
4.2.1. Sprint 3.....	70
4.2.1.1. Detalle.....	71
4.2.1.2. Conclusiones del Sprint.....	71
4.2.2. Sprint 4.....	72
4.2.2.1. Detalle.....	73
4.2.2.2. Casos de Uso.....	73
4.2.2.3. Conclusiones del Sprint.....	74
4.2.3. Sprint 5.....	75
4.2.3.1. Detalle.....	76
4.2.3.2. Casos de Uso.....	77
4.2.3.3. Conclusiones del Sprint.....	77

4.2.4. Sprint 6.....	78
4.2.4.1. Detalle.....	79
4.2.4.2. Casos de Uso.....	80
4.2.4.3. Conclusiones del Sprint.....	80
4.2.5. Sprint 7.....	81
4.2.5.1. Detalle.....	82
4.2.5.2. Casos de Uso.....	84
4.2.5.3. Conclusiones del Sprint.....	84
4.2.6. Sprint 8.....	85
4.2.6.1. Detalle.....	87
4.2.6.2. Casos de Uso.....	89
4.2.6.3. Conclusiones del Sprint.....	89
4.3. Casos de Prueba.....	90
4.3.1. Log in .....	90
4.3.1.1. Log in Exitoso.....	90
4.3.1.2. Log in con Contraseña Inválida .....	91
4.3.1.3. Log in con Email Inválido.....	92
4.3.1.4. Registro Exitoso .....	92
4.3.1.5. Registro con Contraseña Inválida.....	93
4.3.1.6. Log in con Redes Sociales .....	94
4.3.2. Vista en Grilla y Lista.....	94
4.3.2.1. Cambio de Vista de Eventos de Lista a Grilla.....	94
4.3.3. Búsqueda de Eventos .....	95
4.3.3.1. Búsqueda por Nombre de Evento Existente .....	95
4.3.3.2. Búsqueda por Nombre de Evento Inexistente .....	96
4.3.3.3. Búsqueda por Ubicación Existente .....	97
4.3.3.4. Búsqueda por Ubicación Inexistente .....	97
4.3.4. Ordenamiento de Eventos .....	98
4.3.4.1. Ordenamiento por Nombre .....	98
4.3.4.2. Ordenamiento por Precio .....	99
4.3.5. Filtrado de Eventos .....	99
4.3.5.1. Filtrado por Tiempo .....	99
4.3.5.2. Filtrado por Precio .....	100
4.3.5.3. Filtrado por Tipo de Evento .....	101
4.3.6. Barra de Navegación en Detalle de Evento .....	102
4.3.6.1. Correcto Funcionamiento de la Barra de Navegación y las Secciones .....	102
4.3.7. Guardar Evento en Detalle del Evento .....	103
4.3.7.1. Guardar Evento desde Detalle del Evento.....	103

4.3.7.2. Eliminar Evento de Eventos Guardados desde Detalle del Evento .....	103
4.3.8. Datos Personales en My Account.....	105
4.3.8.1. Cambio Exitoso de Contraseña .....	105
4.3.8.2. Cambio de Contraseña Inválida .....	105
4.3.8.3. Cambio de Foto de Perfil.....	106
4.3.8.4. Modificación Exitosa de Datos Personales .....	107
4.3.8.5. Modificación con Errores de Datos Personales.....	108
4.3.9. Eventos Pasados .....	109
4.3.9.1. Ver Eventos Pasados .....	109
4.3.10. Eventos Guardados .....	110
4.3.10.1. Eliminar un Evento Guardado desde la Lista de Eventos Guardados .....	110
4.3.11. Eventos Registrados .....	111
4.3.11.1. Acceder a Eventos Registrados .....	111
4.3.12. Registrarse a un Evento .....	112
4.3.12.1. Seleccionar Tickets .....	112
4.3.12.2. Registro Sin Seleccionar Tickets .....	112
4.3.12.3. Registro a Evento Dentro del Tiempo Establecido ....	113
4.3.12.4. Registro a Evento Fuera del Tiempo Establecido .....	114
4.3.12.5. Registro Exitoso a Evento Gratuito.....	114
4.3.12.6. Registro a Evento con Formulario Inválido .....	115
4.3.12.7. Registro a Evento Pago.....	116
4.3.12.8. Registro a Evento Pago Con Datos de Inválidos .....	117
4.3.13. Eliminar Registro a un Evento .....	119
4.3.13.1. Cancelar Registro a un Evento Gratuito .....	119
4.3.13.2. Cancelar Registro a un Evento Pago.....	119
4.3.14. Consulta de Promo Code .....	120
4.3.14.1. Consultar Promo Code Válido .....	120
4.3.14.2. Consultar Promo Code Inválido .....	121
4.3.15. Obtención de PDF .....	122
4.3.15.1. Imprimir PDF .....	122
4.3.16. Envío de mail.....	123
3.2.16.1. Envío de Email .....	123
4.4. Conclusiones del Proyecto .....	124
4.4.1. Conclusiones en cuanto a la Interfaz de Usuario.....	124
4.4.2. Conclusiones en cuanto a las Funcionalidades .....	124
4.4.2.1. Bugs Conocidos .....	125
4.4.3. Conclusiones en cuanto al Aprendizaje .....	126
4.4.4. Conclusiones en Cuanto a los Riesgos .....	127

4.4.5. Conclusiones en Cuanto a las Estimaciones .....	128
4.4.5. Conclusiones Generales .....	128
Capítulo 5 .....	130
Bibliografía .....	130
Glosario .....	131
Anexo .....	132
A.1. Actas de Reuniones con el Cliente.....	132
A.2. Casos de Uso Detallados.....	143
A.3. Mockups para las Páginas .....	163
A.3.1. Directorio de Eventos (cuadrícula) .....	164
A.3.2. Directorio de Eventos (lista) .....	165
A.3.3. Detalle de Evento.....	166
A.3.4. Registro a un Evento.....	168
A.3.5. Cambio de Contraseña (My Account) .....	169
A.3.6. Datos Personales (My Account) .....	170
A.4. Manual de Usuario .....	171
A.4.1. Introducción .....	171
A.4.2. Login .....	171
A.4.3. Events Directory.....	174
A.4.4. Event Detail .....	177
A.4.5. Event registration .....	181
A.4.6. My Account .....	185
A.4.7. Salesforce .....	189
A.4.7.1. Enviar Speakers .....	189
A.4.7.2. Enviar Sponsors .....	191
A.4.7.3. Enviar Venues .....	194
A.4.7.4. Enviar Events.....	195
A.4.7.5. Promo Codes.....	205
A.5. Manual de Deployment .....	207
A.6. Presentación de la Defensa .....	208
A.7. Especificaciones en cuanto al Código Producido .....	209
A.8. Acceso a Salesforce .....	209
A.9. Carta de Conformidad del Cliente .....	210
A.10. Acerca de la Documentación y Código Presentado en los DVDs .....	210

# Capítulo 1

## Introducción

En el presente documento se establece como objetivo la descripción del desarrollo de RSVPDay, sitio web desarrollado para la compañía Modelit para la inscripción y difusión de eventos organizados por diferentes empresas que utilizan Salesforce.

El proyecto se presenta como una aplicación de uso sencillo para el usuario ya que la experiencia de usuario es esencial para la empresa. En este sentido, gran parte del proyecto está dedicado a la investigación de distintas tecnologías como Bootstrap, AngularJS, la API de Stripe, Facebook connect, Google connect, Google Maps y servicios REST para la integración con nuestra aplicación, desafío importante ya que ninguna de las desarrolladoras tenemos demasiada experiencia en las mismas. Por otro lado, el diseño de la aplicación se vuelve un aspecto a considerar y prestar atención para que refleje los objetivos de la empresa que son la usabilidad y sobre todo que sea una aplicación intuitiva dado la diversidad de edades de los usuarios.

En el anteproyecto se describe al cliente, medios de comunicación con el mismo, procesos actuales que se realizan en la empresa, roles de los actores involucrados en el mismo, el problema junto con debilidades o posibles desafíos, además, se describen algunos de los requerimientos funcionales más importantes, ya que en el desarrollo de las iteraciones del proyecto pueden surgir algunos que no fueron contemplados en un principio.

### 1.1. Descripción del Cliente

Modelit es una empresa uruguaya que hace ya algunos años se encuentra en el mercado ofreciendo servicios de consultoría de software y desarrollo web. Trabajan para empresas de pequeño y mediano tamaño en el exterior. Usan tecnologías basadas en la nube para desarrollar, específicamente Salesforce, un *CRM (Customer Relationship Management* - software para administrar ventas, marketing y la relación con clientes) muy importante en la actualidad. Además de trabajar para clientes, se encuentran en el proceso de desarrollar software propio para comercializar.

La empresa tiene como objetivos la consultoría y desarrollo de productos de software a medida para clientes externos. Asimismo, es una empresa en crecimiento, por lo cual el desarrollo de productos propios está planteado como una meta a corto plazo.

## 1.2. Roles y contacto con Modelit

La aplicación que utilizarán los usuarios externos (pública) será desarrollada por las estudiantes Ximena Lasserre y Marianne Maisonneuve de las carreras de Analista Programador y Analista en Tecnologías de la Información.

El contacto con el cliente se realizará mediante skype, mail y reuniones personales con Diego Febles, Director de Servicios Profesionales de la empresa y Angélica Buffa, Project Manager y Software Architect.

# Capítulo 2

## Anteproyecto

### 2.1. Análisis

#### 2.1.1. Descripción del Problema

El desarrollo de la aplicación web aporta valor al cliente ya que consiste en mejorar los procesos actuales de pagos de asistencia a eventos y la gestión de los eventos a un bajo costo para las empresas que utilizan el sitio web. Por otro lado, también intenta posicionar a la empresa en el mercado de desarrollo de aplicaciones para su comercialización.

Actualmente encuentran el problema de que muchas de las empresas de pequeño y mediano tamaño que utilizan Salesforce no compran aplicaciones de marketing de eventos porque son demasiado caras. Estas aplicaciones, como por ejemplo *Linvio Events* (<http://linvio.com/products/linvio-events>), son adquiridas por las empresas y deben pagar una cuota por usuario, lo cual hace que el costo de las mismas no se justifique. En cambio, utilizan Eventbrite ([www.eventbrite.com](http://www.eventbrite.com)) la cual no se integra del todo bien con Salesforce, ni está pensada para el uso de negocios. De esta forma, resulta difícil para los negociantes realizar informes de impacto de ventas basados en dicha información o mantener una relación personalizada con el cliente en estas situaciones. Al mismo tiempo, al equipo de marketing también se le dificulta alcanzar la eficiencia en los procesos porque los datos de Eventbrite no están disponibles en tiempo real dentro de la cuenta de Salesforce.

Problemas específicos que enfrentan actualmente quienes compran aplicaciones para eventos:

- **Información sobre eventos no disponible en tiempo real en Salesforce.com:** Este problema se da en el caso de los usuarios que utilizan Eventbrite o cualquier otra aplicación que no sea nativa de Salesforce. La información que es modificada en la aplicación no puede ser accedida por los usuarios de Salesforce en tiempo real, por lo tanto implica tardanzas y entorpecimiento del proceso.

- **Costo de las aplicaciones:** Actualmente el costo para utilizar un gestor de eventos nativo de Salesforce es muy elevado, lo cual es un gran obstáculo para las pequeñas y medianas empresas que no tienen el dinero para gastar en algo así.
- **Falta de aplicaciones nativas de Salesforce que sean integrales:** Algunas de las aplicaciones existentes cubren solo una parte de lo que Modelit pretende ofrecer con RSVPDay. Por ejemplo, algunas se centran solo en la compra de tickets para el evento, otras no permiten que un usuario pueda invitar gente, lo cual hace que la falta de difusión e integración con las redes sociales en las aplicaciones actuales es una debilidad.
- **Integración con redes sociales:** Ninguna de las aplicaciones nativas de Salesforce permite integrarlas con redes sociales como Facebook o Twitter y darle promoción al evento por medio de las mismas.
- **Diseño de las aplicaciones:** Con la excepción de Eventbrite (que además tiene aplicación móvil) ninguna es responsive.

## **Principales competidores**

RSVPday tiene dos categorías principales de competidores:

1. Aplicaciones existentes que figuran actualmente en el AppExchange de Salesforce, incluyendo:

Linvio Events - En AppExchange desde diciembre del 2011. Linvio Events es una aplicación de Salesforce para la comercialización del evento y tiene un complemento para el procesamiento de pagos: Linvio PaymentConnect, que debe adquirirse por separado si el cliente quiere albergar eventos pagos.

Soluciones Mylo - Esta aplicación volvió a AppExchange en noviembre 2014 bajo su nuevo nombre, después de su predecesor, una aplicación popular llamada ActevaRSVP. Fue retirada del mercado por mal manejo de los recursos de clientes.

2. Aplicaciones que las SMBs utilizan como alternativas a las aplicaciones mencionadas:

- Eventbrite, que cobra una tarifa que es un porcentaje de cada boleto vendido.

- Aplicaciones gratis sin capacidad de procesamiento de pagos, tales como Meetup.com

- Además, Marketo, un proveedor líder de automatización de marketing, también ofrece la capacidad de marketing de eventos en su producto de entry-level. A un precio de \$895 / mes. No consideramos este producto un competidor directo.

Otros proveedores conocidos como Hubspot, o vendedores de email marketing como MailChimp y Yesware, no ofrecen actualmente la capacidad de marketing de eventos en sus productos. Esto puede cambiar en el futuro, pero hasta la fecha no se ha visto ningún proveedor principal que se centre específicamente en esta parte del mercado.

### Cuadro con análisis de los competidores

	<b>RSVPDay</b>	<b>Linvio Events</b>	<b>Mylo Solutions</b>
<b>Precio</b>	\$ 100/mes/usuario \$ 1000/mes/usuario	Desde \$4175 por organización	\$500 para activarlo + \$100 por evento + \$5 por registro
<b>Edición de Salesforce soportada</b>	Group, Professional, Enterprise, Unlimited, Performance.	Enterprise, Unlimited, Performance.	Professional, Enterprise.
<b>Construida en plataforma nativa de Salesforce</b>	Sí	Sí	No
<b>Procesamiento de pago electrónico</b>	Sí	Tarifa adicional desde \$1425 por año	Tarifa adicional (pagando otra app)
<b>Páginas de eventos configurables</b>	Sí	Sí	Sí
<b>Promoción mediante redes sociales</b>	Sí	Sí	Sí
<b>Marketing por mail</b>	Sí	Sí	Sí
<b>Reportes y dashboards</b>	Sí	Sí	Sí
<b>ROI tracking y</b>	Sí	Sí	No

<b>triggers de ventas</b>			
<b>Optimizado para dispositivos móviles</b>	Sí	No	No

#### 2.1.1.1- Cuadro comparativo competidores

### 2.1.2. Necesidades que Surgen del Problema

A partir de lo anteriormente mencionado, Modelit busca con RSVPDay crear un “Eventrbrite para negocios que utilizan Salesforce.com”. Sería una aplicación nativa de Salesforce que les permita a los comerciantes gestionar y personalizar eventos de negocios y controlar el impacto de los mismos en las ventas directamente desde su cuenta de Salesforce. Una vez que se publiquen los eventos, los usuarios públicos pueden, desde la web o un dispositivo móvil, acceder a la información de los mismos desde un sitio web público, registrarse, y participar de los mismos. Toda la información será almacenada dentro de la cuenta de Salesforce.

El sistema consistirá en dos partes: la privada y la pública. La privada será la administrada por el empleado de la empresa, en donde gestionarán todos los eventos generados para la misma. Esta está siendo desarrollada por Modelit actualmente. La pública es la que pueden ver todos los usuarios y la que forma parte de nuestro proyecto. Los datos que se van a publicar en el sitio público serán obtenidos mediante una API que se comunica con la parte privada.

### 2.1.3. Objetivos para Satisfacer Dichas Necesidades

El objetivo principal que surge en base al análisis de las debilidades de los procesos actuales y de cómo los usuarios utilizarán la aplicación es el de desarrollar un sitio web responsive, que sea nativo de Salesforce, integrado con redes sociales que permita la inscripción y pago de asistencia a eventos organizados por diferentes empresas que utilizan Salesforce en la actualidad.

Los objetivos detallados a continuación son los que atienden las necesidades concernientes a nuestro proyecto, es decir, los que tienen que ver con la parte pública de la aplicación.

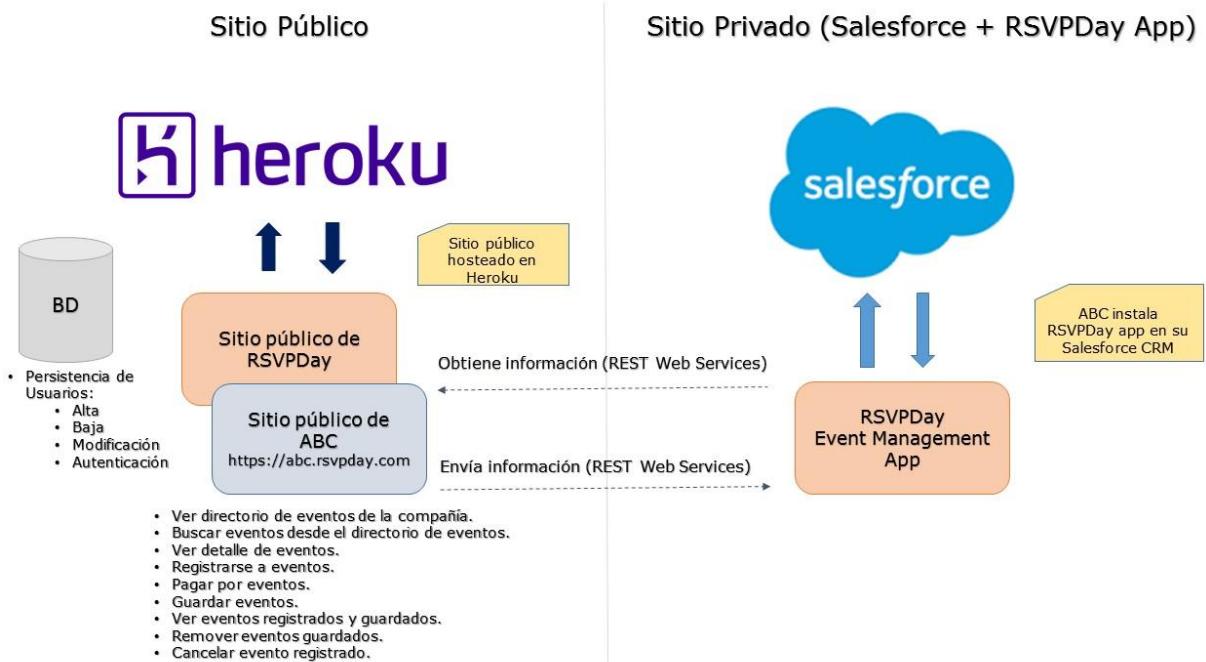
- Gestión de Eventos

- Mostrar los eventos disponibles para la empresa de la que forma parte el usuario.
  - Mostrar los detalles específicos de un evento.
  - Ordenar eventos dentro de la lista de disponibles
  - Buscar eventos dentro de los disponibles
  - Integrar Google Maps para mostrar localización de eventos
  - Ver histórico de eventos en los que se inscribió el usuario
- 
- **Registro Online**
    - Permitir que los participantes puedan registrarse desde la web.
    - Permitir que los participantes puedan pagar por el evento desde la web a través de un servicio externo.
    - Enviar email de confirmación con código QR.
    - Cancelar la inscripción a un evento.
  
  - **Promoción de Eventos**
    - Promover un evento en el directorio de eventos de la empresa, redes sociales y más.
    - Permitir enviar invitaciones a eventos mediante diferentes redes sociales.
  
  - **Estructura del Sitio**
    - El sitio tendrá que ser responsive para poder adaptarse a los dispositivos móviles.
    - El sitio debe ser nativo a Salesforce.
  
  - **Login**
    - Efectuar el login al sitio con Facebook login o Google login (o ambos).

## 2.2. Proyecto

### 2.2.1. Descripción del Entorno

#### 2.2.1.1. Diagrama de Arquitectura del Sistema



## 2.1. Imagen del diagrama de arquitectura

El diagrama anterior explica la arquitectura de las aplicaciones tanto pública como privada. La primera columna muestra el sitio que está desplegado en Heroku y que posee su propia base de datos. La base de datos está conectada mediante Heroku directamente y es una base de datos de Mongo, para gestionarla se utiliza el add-on de Heroku mLab MongoDB. En el diagrama también se especifican las competencias de dicho sitio tales como ver el directorio de eventos, registrarse a un evento, cancelar el registro, etc. La comunicación entre los sitios es mediante servicios REST. Hay dos servicios, uno que recibe los eventos mandados desde Salesforce y otro que se comunica con Salesforce para enviar la información de registro a eventos y la cancelación de los mismos, el primero está implementado con Restivus que es un paquete de Meteor que permite fácilmente hacer CRUD utilizando jsons y el segundo fue desarrollado con Apex REST de Salesforce. En Salesforce, se encuentra el sitio privado que permite el ingreso de nuevos eventos y que, como se mencionó, utiliza una API para enviar los mismos a Heroku. El sitio desarrollado en este proyecto es el que se encuentra en Heroku.

Como parte de la arquitectura de la aplicación fueron utilizados varios paquetes de Meteor y de NPM.

Los paquetes de Meteor básicos que se incluyen por crear una aplicación con dicho framework son:

- meteor-base: paquete básico de toda aplicación Meteor.
- mobile-experience: paquete para desarrollar mobile.
- mongo: paquete para base de datos.
- session: diccionario del lado del cliente.
- jquery
- tracker: librería del lado del cliente para programación reactiva.
- standard-minifiers: minificadores para JS y CSS para producción.
- es5-shim: paquete de compatibilidad para navegadores viejos.

Los siguientes paquetes se agregaron para cumplir diferentes propósitos:

- twbs:bootstrap: paquete para agregar Bootstrap.
- angularui:angular-ui-bootstrap: paquete para utilizar Angular junto con Bootstrap.
- angularui:angular-google-maps: paquete para utilizar Angular y Google Maps.
- Angularui:angular-ui-router: paquete para generar rutas y establecer sus controladores de Angular.
- tmeasday:publish-counts: paquete para contar la cantidad de veces que se comparte un evento en las redes sociales.
- rajit:bootstrap3-datepicker: paquete para agregar la directiva de Angular datepicker.
- accounts-password: paquete para manejar autenticación con contraseñas.
- accounts-google: paquete para autenticación con Google.
- accounts-facebook: paquete para autenticación con Facebook.
- accounts-twitter: paquete para autenticación con Twitter.
- pauli:accounts-linkedin: paquete para autenticación con LinkedIn.
- service-configuration: paquete para manejar la configuración de servicios de terceros.
- nimble:restivus: paquete para crear REST APIs.
- fortawesome:fontawesome: paquete para utilizar iconos y tipografías especiales.
- meteorhacks:npm: paquete para integrar paquetes de NPM como si fueran para Meteor.
- npm-container: paquete que permite utilizar módulos NPM directamente en aplicaciones de Meteor.
- email: paquete para enviar emails.
- accounts-base: paquete para autenticación.
- accounts-ui-modal: paquete modificado especialmente para el proyecto que contiene modales y estilos para el manejo de la cuenta del usuario.
- urigo:angular-blaze-template: paquete que renderiza templates de Blaze como si fueran directivas de angular.
- angular-with-blaze: paquete que renderiza templates de Blaze como si fueran directivas de angular.
- blaze-html-templates: paquete para utilizar templates de Blaze.

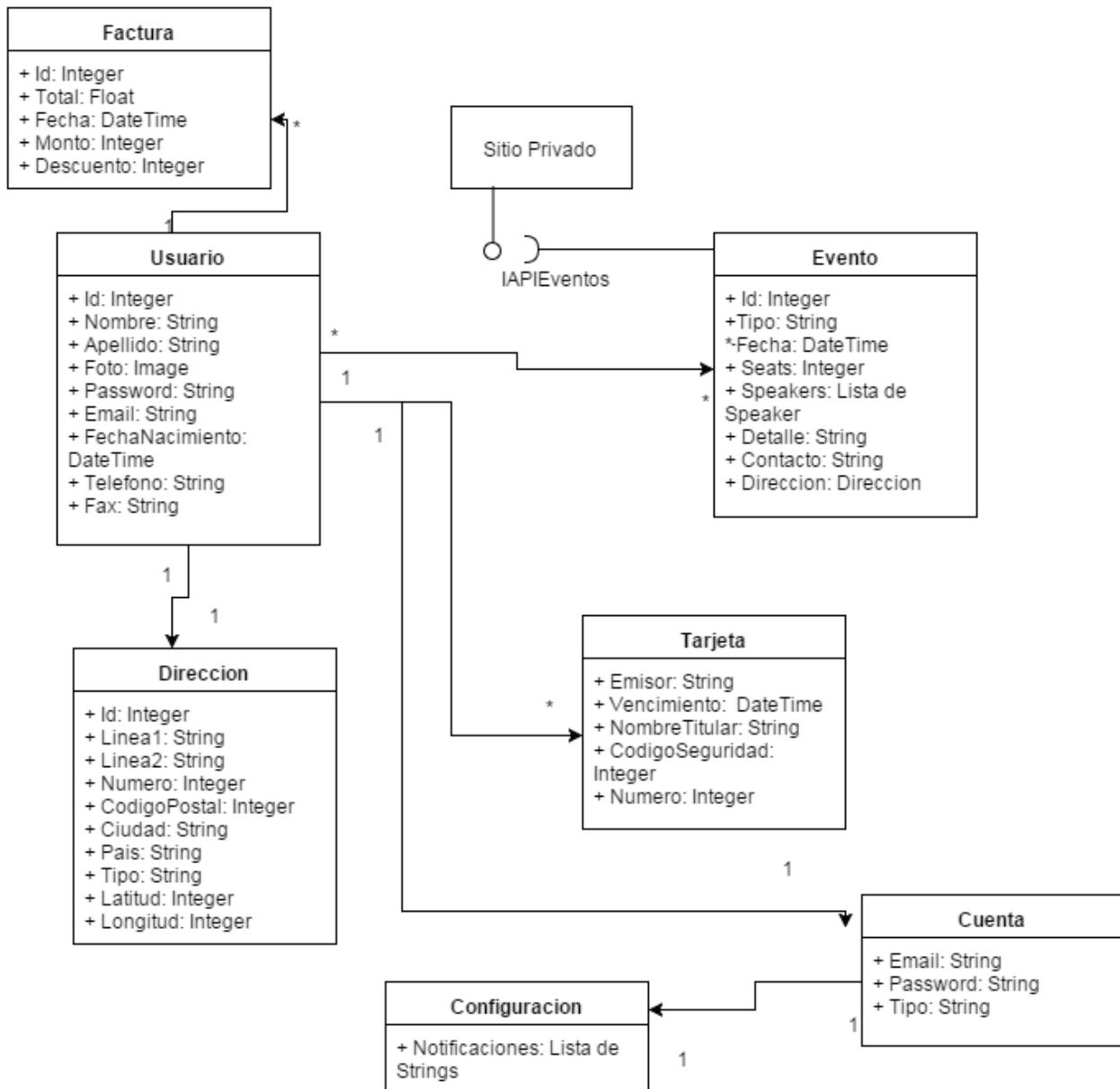
- cfs:standard-packages: paquete que contiene a otros paquetes como son cfs:base-package, cfs:file, etc.
- cfs:gridfs: paquete que permite guardar archivos GridFS para CollectionFS.
- danielfarid:ng-file-upload: paquete para subir archivos con angular.
- jparker:crypto-aes: paquete para encriptar.
- http: paquete que provee una API en el cliente y en el servidor para requests HTTP.
- momentjs:moment: paquete para manejar días y tiempo.
- mrt:moment-timezone: paquete para manejar las zonas horarias.
- mrgalaxy:stripe: paquete para utilizar Stripe.
- angularjs:angular-sanitize: paquete que incluye angular-sanitize como si fuera un paquete de Meteor.
- meteorhacks:ssr: paquete para renderizar templates de Blaze del lado del servidor para enviar mails.
- aaronroberson:ng-mask: paquete para poner masks y formatear el teléfono del usuario.

Los paquetes de NPM utilizados son:

- webshot: paquete para generar un screenshot de la página actual para generar el PDF junto con Phantom.
- jsforce: paquete para manejar las APIs que se comunican con Salesforce.

## 2.2.1.2. Diagrama Conceptual del Dominio

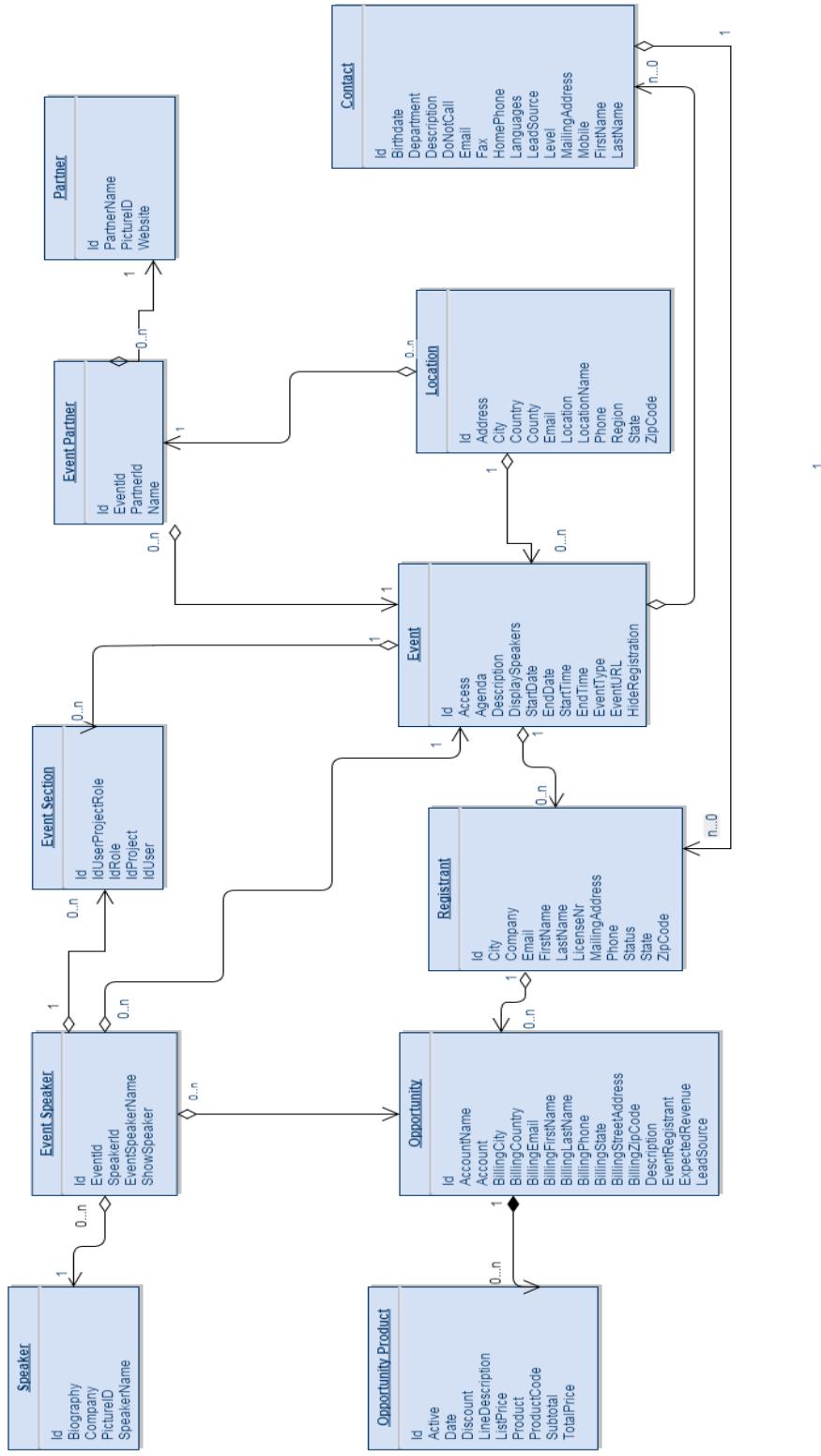
### UML del lado público de la aplicación:



### 2.2. Imagen del UML sitio público

En el diagrama anterior se puede ver el diagrama de las clases de la aplicación del sitio público. Las mismas son: Dirección, Tarjeta, Factura, Usuario, Cuenta, Configuración y Evento.

## **UML del Lado Privado de la Aplicación:**



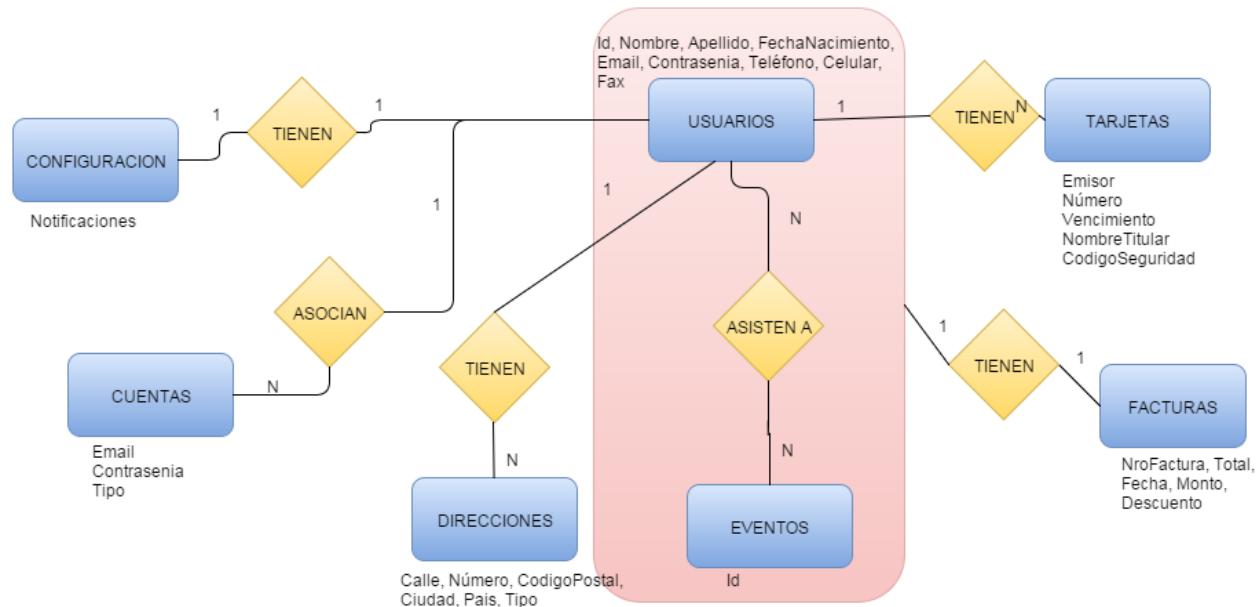
### **2.3. Imagen del UML sitio privado**

En el diagrama se muestran las clases pertenecientes a la parte privada de la aplicación que se encuentra en Salesforce. Las clases utilizadas desde la aplicación pública son: Speaker, EventSpeaker, Registrant, Event, Location y Contact.

### 2.2.1.3. Modelo Entidad Relación

#### **MER del Lado Público de la Aplicación:**

Este MER es el que se va a implementar.



2.4. Imagen del MER sitio público

En el mismo se muestran las relaciones entre las entidades que se encuentran en MongoDB.

### 2.2.2. Análisis de Riesgos

Los riesgos a tener en cuenta para el presente proyecto son:

#### **1) Uso de gran variedad de nuevas tecnologías - Riesgo alto**

Una de las principales razones por las que el proyecto presenta un desafío es por la cantidad y variedad de herramientas y tecnologías a utilizar y con las cuales los estudiantes no están familiarizadas.

En cuanto a la programación, esta se hará principalmente con AngularJS, un framework que ambas conocen y han utilizado, pero no extensivamente. También se utilizará Node.js con el cual ninguna de las dos tiene experiencia, se utilizarán varias APIs como la de Stripe, Facebook y Google, Foundation como front-end framework y MongoDB como base de datos.

Por lo tanto gran parte de la aplicación se llevará a cabo con herramientas inexploradas por cualquiera de las dos estudiantes, lo que significa que habrá una curva de aprendizaje importante para sacar adelante al proyecto.

## **2) Fallas en la estimación de las tareas - Riesgo medio**

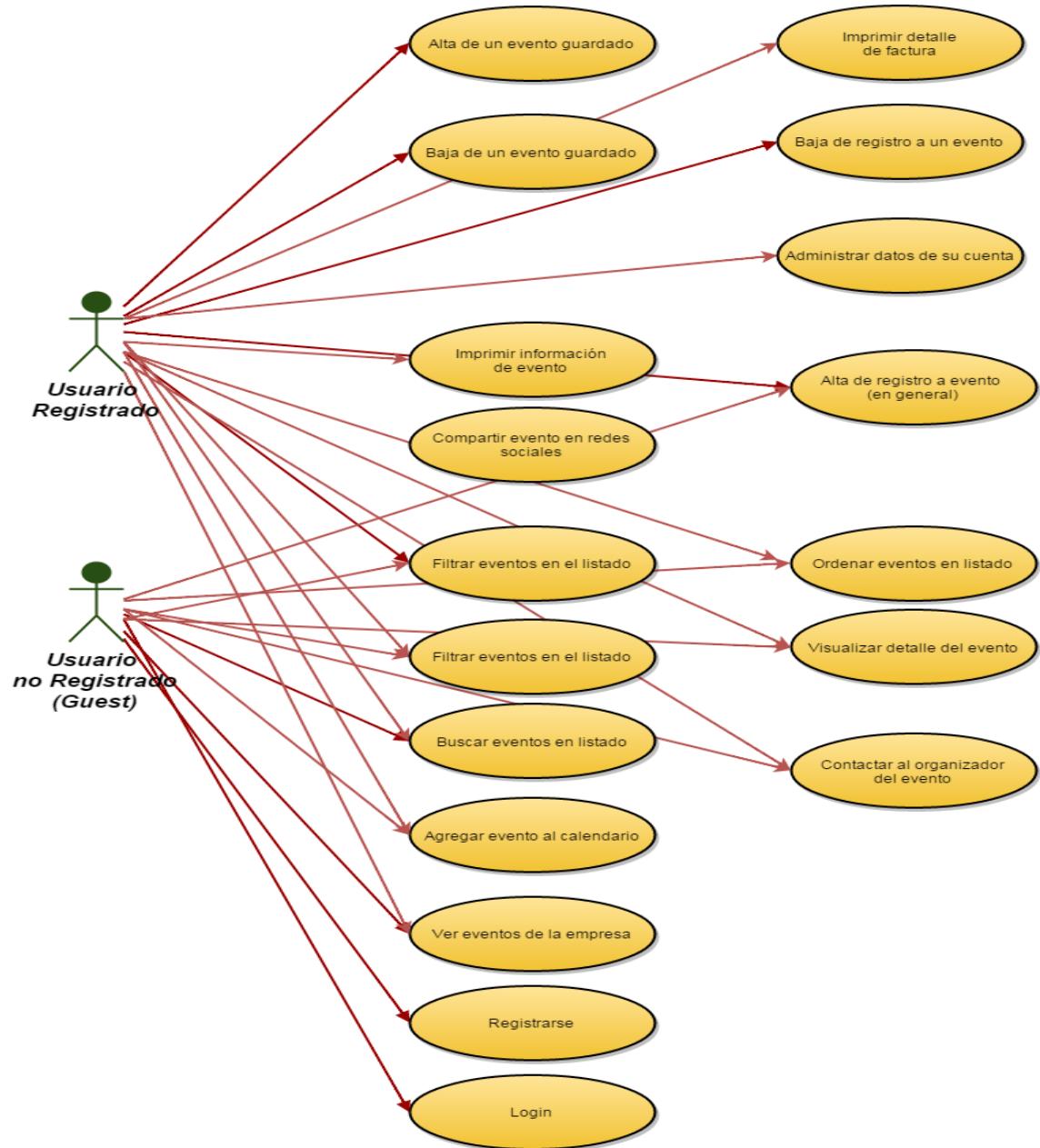
Este riesgo se ve íntimamente relacionado con el número 1, ya que consideramos que la principal causa de fallas en las estimaciones podría ser el aprendizaje de las tecnologías a usar. Sin embargo, creemos que las tareas se pueden ir ajustando de sprint a sprint de acuerdo a cuál fue el resultado del sprint anterior.

No nos parece que este riesgo pueda poner en riesgo todo el proyecto, ya que contamos con la guía y experiencia del tutor que tenemos asignado.

Un plan de contingencia para mitigar este riesgo sería mantener una comunicación permanente con el tutor y el cliente, para ir informándoles de nuestro avance y tratar de ajustar los tiempos de acuerdo a este.

## 2.2.3. Casos de Uso

### 2.2.3.1. Diagrama de Casos de Uso



2.5. Imagen de diagrama de casos de uso

El diagrama de casos de uso muestra qué casos de usos son para cada tipo de usuario (registrado y no registrado). Tanto el usuario registrado como el no registrado tienen acceso al registro de eventos y a eventos con sus búsquedas y filtros. Sin embargo, el

usuario no registrado no puede hacer uso del manejo de la cuenta, ni de los eventos guardados o registrados.

#### 2.2.3.2. Listado de Casos de Uso

**CU-001:** Alta de un registro a un evento sin discount code

**CU-002:** Alta de un registro a un evento con discount code

**CU-003:** Alta de un evento guardado

**CU-004:** Baja de un evento guardado

**CU-005:** Listado de eventos de la empresa

**CU-006:** Filtrar eventos en el listado de eventos de la empresa

**CU-007:** Ordenar eventos en el listado de eventos de la empresa

**CU-008:** Buscar un evento en el sitio

**CU-009:** Login al sistema

**CU-010:** Registro al sistema

**CU-011:** Visualizar detalle de evento

**CU-012:** Agregar evento a calendario

**CU-013:** Imprimir evento

**CU-014:** Imprimir detalle de factura

**CU-015:** Administrar datos de cuenta de usuario

**CU-016:** Administrar listado de eventos guardados por el usuario

**CU-017:** Administrar listado de eventos registrados por el usuario

**CU-018:** Ver listado de eventos pasados a los que se registró el usuario

**CU-019:** Contactar organizadores de un evento

**CU-020:** Compartir evento en redes sociales

**CU-021:** Baja de un registro a un evento

Los detalles de cada caso de uso se encuentran en el anexo de este documento (A.2.).

#### 2.2.4. Backlog del Producto

1. Como usuario quiero poder registrarme a un evento.
2. Como usuario quiero poder registrarme a un evento y utilizar códigos de descuentos.
3. Como usuario quiero poder ver los eventos con una breve descripción de los mismos.
4. Como usuario quiero poder filtrar los eventos por fecha, lugar y tipo.
5. Como usuario quiero buscar eventos en el directorio de eventos.

6. Como usuario quiero poder ordenar los eventos de acuerdo al precio y fecha.
7. Como usuario quiero poder loguearme con redes sociales.
8. Como usuario quiero poder loguearme.
9. Como usuario logueado quiero poder guardar un evento en cualquier momento.
10. Como usuario quiero poder guardar un evento en mi calendario.
11. Como usuario quiero poder guardar un evento en mi cuenta.
12. Como usuario logueado quiero poder borrar un evento guardado en mi cuenta.
13. Como usuario logueado quiero poder administrar mis datos personales, mis datos de pago y mi dirección desde mi cuenta.
14. Como usuario quiero poder visualizar los detalles sobre speakers, fecha, lugar (en un mapa), datos de contacto del responsable del evento, sponsors de un evento.
15. Como usuario quiero que luego de registrarme a un evento pueda imprimir los speakers, fecha y lugar (en un mapa) del evento al que me inscribí.
16. Como usuario logueado quiero administrar mi historial de eventos.
17. Como usuario quiero poder hacer pagos online.
18. Como usuario logueado quiero visualizar los próximos eventos a los que estoy registrado y los que guardé.
19. Como usuario quiero que se me envíe un email con los detalles de un evento luego de registrarme.
20. Como usuario quiero que se me mande un código para poder entrar al evento.
21. Como usuario quiero poder contactar a los organizadores de un evento.
22. Como usuario quiero imprimir el detalle de la factura una vez hecho el pago online.
23. Como usuario, quiero que se me indique el tiempo que tengo para completar un formulario de inscripción a un evento gratuito y a uno pago.

## 2.3. Plan de Proyecto

### 2.3.1. Descripción de Herramientas y Tecnologías

#### 2.3.1.1. Tecnologías a Utilizar

- **Salesforce** - Software para la administración de la relación con los clientes, CRM, por sus siglas en inglés. Se utilizará la Force.com REST API que es una API de servicios web para interactuar con Force.com (en nuestro caso particular, con el sitio privado de la aplicación).

- **AngularJS** - Framework de Javascript con el que se escribirá la mayoría del código. Es de código abierto y mantenido por Google y está basado en el patrón MVC (Modelo-Vista-Controlador: es un patrón de arquitectura de software que consiste en separar los datos y la lógica de la vista y del módulo de eventos y comunicación).
- **Node.js** - Entorno en tiempo de ejecución para el desarrollo de aplicaciones web del lado del servidor. Será usado principalmente para mantener la comunicación con la base de datos (MongoDB).
- **ExpressJS** - Es una framework para aplicaciones web de Node.js que contiene un conjunto de características para aplicaciones web. Permite la creación de APIs con gran facilidad.
- **MongoDB** - Es una base de datos NoSQL orientada a documentos. Las estructuras de datos se guardan en documentos tipo JSON. Esto permite que la integración con la base de datos se haga de una manera más fluida y simple.
- **Heroku** - Es una *Plataforma como Servicio* (*PaaS* por sus siglas en inglés) en la nube que nos va a permitir hacer deploy de la aplicación y correrla desde allí.
- **HTML** - *HyperText Markup Language*. Lenguaje de marcado para elaborar páginas web.
- **CSS** - *Cascading Style Sheets*. Lenguaje para definir la presentación o vista del HTML.
- **Javascript** - Es un lenguaje de programación dinámico, generalmente usado como parte del navegador web para desarrollar mejoras en la interfaz de usuario y páginas web dinámicas.
- **Foundation** - Front-End framework que permite que se desarrolle el diseño de la página más rápidamente. También facilita la creación de páginas responsive.
- **Stripe API** - Stripe es un conjunto de APIs que permite gestionar pagos, por lo tanto facilita la programación en dicho sentido.
- **Facebook Login API** - API de Facebook que permite a los usuarios hacer log in en una aplicación sin tener que ingresar ningún dato, nada más que asociar su cuenta de Facebook a dicho sistema.

- **Google Sign-In API** - API de Google que posibilita a los usuarios hacer log in en una página utilizando su cuenta de Google.
- **Git y Github** - Se van a utilizar para mantener el versionado del código. Nos decidimos por esta herramienta ya que es la de uso del cliente y al mismo tiempo nos sentimos cómodas utilizándola. Además de que es práctica y confiable.

### 2.3.1.2 Entorno de desarrollo (IDE)

Para el desarrollo de la aplicación se utilizará Sublime. Si bien no es un IDE el mismo es un editor de texto libre, lo cual lo hace conveniente.

### 2.3.2. Marco de Trabajo

#### 2.3.2.1. Comunicación entre las Partes

Las dos partes con las que se mantendrá la comunicación son el tutor y el cliente.

La comunicación con el tutor se realizará mediante reuniones semanales o cada dos semanas, dependiendo de la necesidad. Las mismas se coordinarán via email o en la reunión anterior. También el email servirá de medio cuando se quieran hacer preguntas concretas al tutor y si la situación lo ameritara se ha considerado la opción de utilizar Skype para hacer reuniones.

Con el cliente se espera tener una o dos reuniones semanales, dependiendo de la carga de trabajo y si hay algún obstáculo que esté obstruyendo el flujo normal de trabajo. Las mismas también podrán ser presenciales, en las oficinas de Modelit o por Skype o GoToMeeting. También se maneja el email como medio de comunicación para asuntos o preguntas más concretas.

En cuanto a la comunicación entre los estudiantes, esta se hará presencialmente, mediante email o mediante chat de algún tipo.

#### 2.3.2.2. Herramientas de Comunicación

Como fue dicho anteriormente, en cuanto a herramientas de comunicación se utilizarán para videoconferencias tanto Skype como GoToMeeting y como herramienta de chat se utilizará Google Talk.

Asimismo, también se mantendrá la comunicación con el cliente mediante Pivotal Tracker, la herramienta de gestión de proyecto que se utilizará.

### 2.3.3. SQA

#### 2.3.3.1. Estándares Definidos y Convenciones

El trabajo de programación se hará de acuerdo a los estándares y convenciones definidos por Modelit que son con los que trabajan en la empresa.

#### **Convenciones y Estándares Generales:**

Las siguientes convenciones<sup>1</sup> fueron provistas por el cliente:

Principios Generales	
<ul style="list-style-type: none"><li>- <i>DRY - Don't repeat yourself</i> - Evitar repetición</li><li>- Principio de abstracción - Está relacionado a DRY. Dicho principio dice que "Cada pieza funcional significativa en un programa debería implementarse en solo un lugar en el código fuente".</li><li>- <i>KISS (Keep it simple, stupid!)</i> - ¡Hazlo simple, estúpido! La simplicidad (y el evitar complejidades) deberían siempre ser las metas principales. El código simple demora menos en escribirse, tiene menos bugs y es más fácil de modificar.</li><li>- No intentes optimizar el código prematuramente, que sea legible y comprensible.</li><li>- Todo el código en una base de código debería lucir como si una misma persona lo hubiera escrito, incluso cuando hay muchas personas contribuyendo al mismo.</li><li>- Evitar crear un <i>YAGNI (You aren't going to need it)</i>. En español <i>No lo necesitarás</i>) - Deberías tratar de no agregar funcionalidad que no vas a necesitar.</li><li>- Utiliza la manera más fácil con la que podría funcionar - Una buena pregunta para hacerse es "¿Qué es lo más fácil que podría funcionar?"</li><li>- No me hagas pensar - El objetivo es que el código sea fácilmente leíble y comprendido con el menor esfuerzo.</li><li>- Escribe código para el que lo mantendrá - Casi cualquier código que vale la pena escribir vale la pena mantenerlo en el futuro, por ti o por alguien más.</li><li>- Principio de la Responsabilidad Única - Un componente de código (por ejemplo una clase o función) debería realizar una tarea única y bien definida.</li><li>- Minimiza el acoplamiento - Una sección de código (bloque de código, función, clase, etc.) debería minimizar las dependencias de otras áreas del código. Esto se puede lograr usando variables compartidas lo menos posible.</li><li>- Maximiza la cohesión - El código que tiene una funcionalidad similar debería encontrarse dentro del mismo componente.</li><li>- La reutilización de código es buena - Reutilizar código mejora la legibilidad del mismo y disminuye el</li></ul>	

<sup>1</sup> Los ejemplos algunos se tradujeron al español para mostrar también cómo sería en español, pero la mayoría se mantuvo en inglés ya que es cómo se va a escribir el código.

tiempo de desarrollo.
- Separación de Conceptos - Distintas áreas de funcionalidad deberían manejarse por módulos de código distinto y que se superpongan de manera mínima.
- Acepta el Cambio - Este es un concepto de programación extrema y de metodología ágil en general. Deberías esperar y aceptar el cambio.

<b>Convenciones de Nombres</b>	Siempre utiliza las mismas convenciones de nombres para todo el código.	
Variables	<ul style="list-style-type: none"> <li>- Deben empezar con una letra.</li> <li>- Deben ser en <i>camelCase</i>.</li> <li>- No comenzar con signo de pesos (\$)</li> <li>- No usar guiones</li> <li>- Descriptivas. Legibles. Consistentes. Contextuales.</li> </ul>	<i>myVariableName</i> <i>accountStatus</i>
Variables globales y constantes	<ul style="list-style-type: none"> <li>- Deben escribirse en MAYÚSCULA</li> <li>- Deben usar guión bajo</li> </ul>	<i>NUMBER_OF_DAYS</i> <i>NUMBER_PI</i>
Funciones y métodos	<ul style="list-style-type: none"> <li>- Deben empezar con una letra</li> <li>- Deben usar <i>camelCase</i>.</li> <li>- No deben empezar con signo de pesos (\$).</li> <li>- No usar guiones</li> <li>- Descriptivos. Legibles. Consistentes. Contextuales.</li> </ul>	<i>myFunction()</i> <i>myNewFunction()</i> <i>processAccounts()</i> <i>setVar()</i> <i>getVar()</i>
<b>Estándares de Código</b>		
Espacios alrededor de operadores	Siempre dejar espacios alrededor de los operadores ( = + - * / ), y luego de las comas.	<code>var x = y + z;</code> <code>var values = ["Volvo", "Saab", "Fiat"];</code>
Indentación de código	Siempre usar 4 espacios para la indentación de bloques de código.	<code>function toCelsius(fahrenheit) {</code> <code>    return (5 / 9) * (fahrenheit - 32);</code> <code>}</code>
Reglas de Declaraciones	<ul style="list-style-type: none"> <li>- Siempre terminar una declaración simple con punto y coma.</li> <li>- Poner la llave de apertura al final de la primera línea.</li> <li>- Poner un espacio antes de la llave de apertura.</li> <li>- Poner la llave de cierre en una nueva línea, sin espacios antes.</li> <li>- No finalizar una declaración</li> </ul>	<code>var values = ["Volvo", "Saab", "Fiat"];</code>  <code>function toCelsius(fahrenheit) {</code> <code>    return (5 / 9) * (fahrenheit - 32);</code> <code>}</code>  <code>for (i = 0; i &lt; 5; i++) {</code> <code>    x += i;</code> <code>}</code>

	compleja con punto y coma.	<pre>if (time &lt; 20) {     greeting = "Good day"; } else {     greeting = "Good evening"; }</pre>
Largo de línea < 80	<ul style="list-style-type: none"> <li>- Por un tema de legibilidad, no deben haber líneas más largas que 80 caracteres.</li> </ul> <p>Si una sentencia no cabe en una línea, el mejor lugar para pasar a la siguiente es después de un operador o una coma.</p>	<pre>document.getElementById("demo").innerHTML = "Hello Dolly.;"</pre>
Comentarios	<p>Un código bien comentado es extremadamente importante. Tómate tiempo para describir componentes, cómo funcionan, sus limitaciones y la forma en la que están construidos. No dejes que los demás tengan que adivinar el propósito de código raro o poco obvio.</p> <p>- Poner comentarios en una nueva línea sobre su tema.</p> <p>- Mantener largo de línea en un máximo sensato, por ej. 80 columnas.</p> <p>- Escribir comentarios libremente.</p> <p>- Utilizar estilo oracional (Usa mayúscula solo en la primera palabra en una oración y que sean consistentes con la indentación del texto.</p>	<pre>/* =====</pre> <p><i>Section comment block</i></p> <pre>===== */</pre> <p><i>Sub-section comment block</i></p> <pre>===== */</pre> <p>* Descripción corta usando el formato de comentarios del estilo de Doxygen</p> <p>* La primera oración de la descripción larga *comienza aquí y continúa en esta línea *un poco hasta que concluye aquí al final *de este párrafo.</p> <p>* La descripción larga es ideal para explicaciones y documentación más detalladas. Puede incluir ejemplos de HTML, URLs o cualquier otra información que se considere necesario o útil.</p> <p>* @tag Este es un tag llamado 'tag'</p> <p>* TODO: Esta es una declaración TODO que describe una tarea que debe ser completada luego.</p>

		*/  /* Comentario básico */  // Comentario en línea
--	--	---

## 2.1. Tabla de convenciones generales

### Git y Github:

Repository		
Nombre	- Descriptivos. Legibles. Consistentes. Contextuales. - Usar patrón de minúsculas y guiones	<i>an-awesome-repo</i>
Branch		
Nombre	- Descriptivos. Legibles. Consistentes. Contextuales. - Usar patrón de minúsculas y guiones	
Feature/Task/Issue	<i>topic-*</i>	<i>topic-create-user</i>
Bug	<i>bug-*</i>	<i>bug-username-must-be-required</i>
Hotfix	<i>hotfix-*</i>	<i>hotfix-user-cant-login</i>
Pull Request		
Título	- Resumen corto (50 caracteres o menos) de los cambios. - {Nombre de la Tarea, ID de la Tarea} o {Resumen de cambios}	- Agregar formulario de registro, #101 - Add a case support system
Descripción	{Lista of URLs de la historia} {Descripción de items} {Lista de Test Methods relacionados a la historia (si aplica)}	<a href="http://sometrackerurl">http://sometrackerurl</a> <a href="http://anothertrackerurl">http://anothertrackerurl</a>  - Allow people to register to the website - Create a user table - Crear controlador - Crear vista  - <i>TestMethodClass.cls</i> (95%) - <i>AnotherTestMethodClass.cls</i> (95%)
Commit		

Mensaje	<p>Crear mensajes de commit útiles. Esto es una de las mejores cosas que se pueden hacer por los demás que usan el mismo repositorio. Permite que las personas puedan comprender rápidamente los cambios sin tener que leer el código.</p> <p>Descripción de lo que hiciste en infinitivo, en vez de pasado. Ejemplos: “escribir”, “agregar”, en vez de “escribi”, agregué”.</p> <p>Buenos mensajes de commit sirven para al menos tres propósitos importantes:</p> <ul style="list-style-type: none"> <li>- Acelerar el proceso de revisión.</li> <li>- Ayudar a escribir una buena nota de release.</li> <li>- Para ayudar a quienes lo vayan a mantener para que puedan saber por qué un cambio en particular fue hecho o por qué determinada feature se agregó.</li> </ul>	<ul style="list-style-type: none"> <li>- <i>Add a table to the page</i></li> <li>- <i>Change a label</i></li> <li>- <i>Agregar estilos</i></li> <li>- <i>Arreglar bug en formulario de login</i></li> </ul>
---------	--	---

## 2.2. Tabla de convenciones de Git

### Javascript:

Convenciones de Nombres		
Nombre de archivo	<ul style="list-style-type: none"> <li>- Debe empezar con una letra</li> <li>- Debe ser en minúscula</li> <li>- No usar espacios</li> <li>- Usar guiones</li> <li>- Descriptivo. Legible. Consistente. Contextual.</li> <li>- Debe tener .js como extensión.</li> </ul>	<i>my-application-file.js</i> <i>application.js</i>
Estándares de Código		
Reglas de objeto	<ul style="list-style-type: none"> <li>- Poner la llave de apertura en la misma línea que el nombre del objeto.</li> <li>- Usar dos puntos más un espacio entre cada</li> </ul>	<pre>var person = {   firstName: "John",   lastName: "Doe",</pre>

	<p>propiedad y su valor.</p> <ul style="list-style-type: none"> <li>- Usar comillas entre los valores que son strings, no en los valores numéricos.</li> <li>- No agregar una coma luego del último par propiedad-valor.</li> <li>- Poner la llave de cierre en una nueva línea, sin espacios antes.</li> <li>- Siempre terminar una definición de objeto con un punto y coma.</li> </ul>	<pre>age: 50, eyeColor: "blue" };</pre>
--	---	---

### 2.3. Tabla de Convenciones de Javascript

#### CSS:

Convenciones de Nombres		
Nombre del archivo	<ul style="list-style-type: none"> <li>- Debe comenzar con una letra.</li> <li>- Debe ser en minúscula.</li> <li>- No usar espacios.</li> <li>- Usar guiones</li> <li>- Descriptivo. Legible. Consistente. Contextual.</li> <li>- Debe tener una extensión .css.</li> </ul>	<i>my-styles.css</i> <i>styles.css</i>
	<p>Si solo tendrás un archivo CSS para tu aplicación, puedes llamarlo:</p> <i>styles.css</i>	<i>styles.css</i>
	<p>Nombra las hojas de estilo de acuerdo a su función. Como una página web puede tener múltiple hojas de estilo es útil dividir los estilos en distintas hojas dependiendo de la función de esa hoja y los estilos en la misma.</p>	<ul style="list-style-type: none"> <li>- <i>Layout vs Design</i>  <i>layout.css</i>  <i>design.css</i></li> <li>- <i>Secciones de la página</i>  <i>main.css</i>  <i>nav.css</i></li> <li>- <i>Todo el sitio con subsecciones</i>  <i>main-styles.css</i>  <i>sub-page.css</i></li> </ul>
Estándares de Código		
Estructura	<ul style="list-style-type: none"> <li>- Siempre usar 4 espacios para indentar cada propiedad.</li> </ul>	Correcto:

	<ul style="list-style-type: none"> <li>- Añadir dos líneas en blanco entre las secciones y una línea en blanco entre bloques de una sección.</li> <li>- Cada selector debería estar en su propia línea, terminando en una coma o en una llave de apertura.</li> <li>- Los pares de propiedades y valores deben estar en su propia línea, con 4 espacios de indentación y un punto y coma al final. La llave de cierre debe estar a la izquierda, usando el mismo nivel de indentación que el primer selector.</li> </ul>	<pre>#selector-1, #selector-2, #selector-3 {     background: #fff;     color: #000; }  <i>Incorrecto:</i>  <code>#selector-1, #selector-2, #selector-3 { background: #fff; color: #000; }</code>  <code>#selector-1 { background: #fff; color: #000; }</code></pre>
Selectores	<ul style="list-style-type: none"> <li>- Usar minúscula y separar las palabras con guiones.</li> <li>- No usar camelcase y guiones bajos.</li> <li>- Usar selectores legibles que describan a qué elementos le dan estilo.</li> <li>- Los atributos de los selectores deben tener comillas dobles alrededor del valor.</li> <li>- Evitar usar selectores sobrecalificados, div.container puede ser simplemente .container.</li> </ul>	<p>Correcto:</p> <pre>#comment-form {     margin: 1em 0; }  .comment-form {     margin: 1 em 0; }  input[type="text"] {     line-height: 1.1; }  <i>Incorrecto:</i>  <code>#commentForm { /* Evitar camelcase. */ margin: 0; }  #comment_form { /* Evitar guiones bajos. */ margin: 0; }  div#comment_form { /* Evitar sobrecategorización. */ margin: 0; }  #c1-xr { /* ¿Qué es c1-xr? Usar mejor nombre. */</code></pre>

		<pre> margin: 0; }  input[type=text] { /* Debería ser [type="text"] */ line-height: 110% /* Dblemente incorrecto */ } </pre>
Propiedades	<ul style="list-style-type: none"> <li>- Las propiedades deben seguirse por dos puntos y espacio.</li> <li>- Todas las propiedades y valores deberían estar en minúscula, excepto por nombres de fuentes y propiedades específicas del proveedor.</li> <li>- Utilizar código hex para los colores, o rgba() si se precisa opacidad. Evitar el formato RGB y mayúscula, y acortar los valores cuando sea posible: #fff en vez de #FFFFFF.</li> <li>- Tanto como sea posible usar atajos (excepto al hacer override de estilos) para los valores de background, border, font, list-style, margin y padding.</li> </ul>	<pre> #overlay { position: absolute; z-index: 1; padding: 10px; background: #fff; color: #777; } </pre>
Orden de las propiedades:	<p>Sobre todo, elegir algo que tenga sentido semántico de alguna forma:</p> <p>Agrupar por significado: Display Posicionamiento Modelo de cajas Colores and Tipografía Otros</p> <p>o</p> <p>Alfabéticamente</p>	<p>Alfabéticamente:</p> <pre> #overlay { background: #fff; color: #777; padding: 10px; position: absolute; z-index: 1; } </pre>
Valores	<ul style="list-style-type: none"> <li>- Espacio antes del valor, luego del punto y coma.</li> <li>- No usar paréntesis en vez de espacios.</li> <li>- Siempre terminar en punto y coma.</li> <li>- Usar comillas dobles en vez de</li> </ul>	<p>Correcto:</p> <pre> .class { /* Uso correcto de comillas */ background-image: url(images/bg.png); font-family: "Helvetica Neue", sans-serif; } </pre>

	<p>simples, y solo cuando sean necesarias, como cuando el nombre de una fuente tiene espacios.</p> <ul style="list-style-type: none"> <li>- Los valores que son 0 no deberían tener unidades, a menos que sea necesario, como en el caso de <i>transition-duration</i>.</li> <li>- Múltiples valores separados por coma para una propiedad deben estar separados o por un espacio o por una nueva línea, incluso dentro de <i>rgba()</i>.</li> </ul>	<pre>.class /* Uso correcto de valores 0*/ font-family: Georgia, serif; text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.5), 0 1px 0 #fff; }  Incorrecto:</pre> <pre>.class /* Evitar no poner espacio y punto y coma */ background:#fff }  .class /* Evitar poner unidad en valor 0*/ margin: 0px 0px 20px 0px; }</pre>
Comentarios	<ul style="list-style-type: none"> <li>- Comentar, y que sea liberalmente.</li> <li>- Los comentarios largos deberían tener un largo por línea de 80 caracteres.</li> </ul>	<p>Para secciones y subsecciones:</p> <pre>/**   * ## Título de sección  *  * Descripción de sección, ya tenga o no media queries, etc. */</pre> <p>Para comentarios en línea:</p> <pre>/* Este es un comentario sobre este selector*/ .another-selector { position: absolute; top: 0 !important; /* debería explicar por qué es importante*/</pre>

## 2.4. Tabla de Convenciones de CSS

### 2.3.4. SCM

#### 2.3.4.1. SCM en el Desarrollo

La herramienta de SCM a utilizar será Git y Github será la herramienta para alojar el proyecto. La elección se debe a que es el sistema que utilizan en Modelit. Asimismo, ambas estudiantes están familiarizadas con el manejo del mismo y se sienten cómodas utilizándolo. Otra de las razones por la que Git fue seleccionada es porque es considerada una herramienta confiable y eficiente.

El cliente desea que se utilice una cuenta privada para manejar el proyecto. A la misma tendrán acceso el cliente, las estudiantes y el tutor para hacer el seguimiento del desarrollo.

#### 2.3.4.1. SCM de la Documentación

En el caso de la documentación se pensó en Google Drive para compartir y editar archivos, ya sean documentos, planillas, pdfs, etc. Consideramos a la misma como una herramienta de fácil uso y que presenta funcionalidades muy útiles en términos de edición de documentos, como por ejemplo hacer comentarios a los mismos, editarlos de manera colaborativa y en tiempo real, etc. Además, es de gran ventaja que sea gratuita y muy conocida por todo el mundo.

## 2.4. Plan de Testing

El plan de testing va a consistir en distintas partes. Primero se van a realizar tests unitarios para los casos de uso más importantes. La herramienta a usar para crear estos tests aún está en plan de investigación ya que al usar AngularJS y Node.js, lenguajes con los que ninguna está familiarizada, tenemos poco conocimiento de las herramientas para testear. Sin embargo, conocemos de la existencia de algunas tecnologías como Jasmine, que sirve para testear ambos, Karma, para AngularJS, y Mocha, para Node.js. Los test unitarios se van a generar una vez terminada cada historia de usuario, si esta lo requiriera.

En segunda instancia, también se van a hacer tests manuales, basándose en los casos de uso. Estos también se realizará cuando la desarrolladora finalice con la historia de usuario. Se han apartado también algunas horas del último sprint (sprint 8) para repasar todos los casos de uso y manualmente testear la aplicación.

## 2.5. Planificación

### 2.5.1. Metodología

La metodología de planificación a utilizar en el presente proyecto es el scrum. La misma es una metodología de desarrollo ágil para gestionar el desarrollo de un producto. Uno de los principios fundamentales del scrum es que durante el proceso de producción el cliente puede cambiar de parecer en cuanto a lo que desea o necesita del producto y por lo tanto el equipo debe reaccionar de manera rápida al cambio en el proyecto. Según Sommerville "*los métodos ágiles universalmente dependen de un proceso iterativo para la especificación, desarrollo y entrega del software, y principalmente fueron diseñados para apoyar al desarrollo de aplicaciones de negocio donde los requerimientos del sistema normalmente cambiaban rápidamente durante el proceso de desarrollo*" [1:362]. Este es un problema en otras metodologías en las que es más complicado aceptar y cambiar rápidamente si cambian los requerimientos del cliente.

El scrum está dividido en sprints. Estos son iteraciones de una o dos semanas en las que se planifican las tareas en las que trabajará cada uno y se trabaja en las mismas. Al final de cada uno se hace una reunión retrospectiva del sprint para revisar el trabajo completado y lo que no pudo ser completado.

### 2.5.2. Anteproyecto: Sprint 1 y 2

Los sprints 1 y 2 consisten en la planificación del proyecto y creación de la documentación del mismo.

El sprint número 1 consta de las siguientes tareas:

- Descripción del cliente y su visión
- Qué procesos tienen actualmente, por qué y cómo los quieren cambiar
- Descripción general de las tareas a realizar y estimación de esfuerzo

El sprint número 2 consta de las siguientes tareas:

- Herramientas y tecnologías a utilizar
- Arquitectura del sistema
- Gestión de riesgos
- Planificación de las tareas para los siguientes sprints

### 2.5.3. Cronograma de Desarrollo: Sprint 3 al 8

Al considerar el esfuerzo estamos considerando también el testing de la historia.

Sprint/Número Tarea	Historia	Esfuerzo
<b>1. Sprint 3</b>	<b>Investigación de tecnologías y configuración de ambiente</b>	
1.1.	Investigar AngularJS	16
1.2.	Investigar Mean Stack	8
1.3.	Investigar Heroku	8
1.4.	Investigar Node.js	12
1.5.	Investigar Mongo y crear base de datos	8
1.6.	Investigar API Salesforce	8
1.7.	Setear environment	2
1.8.	Interfaz de la API	2
<b>Subtotal</b>		<b>64</b>
<b>2. Sprint 4</b>	<b>Directorio de Eventos, Registro, Login al sitio</b>	
2.1.	Investigar Foundation	4
2.2.	Traer de la API todos los eventos disponibles	4
2.3.	Crear markup de directorio de eventos	6
2.4.	Filtrar eventos por nombre	6
2.5.	Filtrar eventos por fecha	4
2.6.	Filtrar eventos por tipo	4
2.7.	Ordenar eventos por nombre y precio	3
2.8.	Filtrar eventos por location	5
2.9.	Paginar eventos (con lazy loading)	5
2.10.	Investigar API de Google para loguearse	4
2.11.	Investigar API de Facebook para loguearse	4
2.12.	Diseñar login (método regular)	6
2.13.	Añadir Facebook login al login	8
2.14.	Añadir Google SignIn al login	8
2.15.	Diseñar formulario extendido de registro de datos de cuenta del usuario	4
2.16.	Investigar opciones para envío de emails	3
2.17.	Implementar confirmación de email mediante envío de link	6
Bugs		8

<b>Subtotal</b>		<b>92</b>
<b>3. Sprint 5</b>	<b>Detalle de Eventos</b>	
3.1.	Crear markup de detalle de evento (overview)	7
3.2.	Crear markup de detalle de evento (speakers)	7
3.3.	Crear markup de detalle de evento (sponsors)	7
3.4.	Crear formulario (pop up) de contacto con organizador	4
3.5.	Validaciones del formulario de contacto con organizador	4
	Investigar cómo funcionan APIS de calendarios Google Calendar, Yahoo Calendar, Microsoft Outlook, Apple iCal	
3.6.	Investigar cómo funcionan APIS de calendarios Google Calendar, Yahoo Calendar, Microsoft Outlook, Apple iCal	5
3.7.	Implementar adición del evento al calendario de elección	5
3.8.	Investigar cómo generar pdfs	3
	Imprimir datos de eventos a los que no se está registrado en pdf	
3.9.	Imprimir datos de eventos a los que no se está registrado en pdf	5
	Investigar API Facebook para realizar y compartir publicaciones	
3.10.	Investigar API Facebook para realizar y compartir publicaciones	2
3.11.	Investigar API Twitter para realizar y compartir publicaciones	2
	Investigar API LinkedIn para realizar y compartir publicaciones	
3.12.	Investigar API LinkedIn para realizar y compartir publicaciones	2
3.13.	Compartir evento en Facebook	4
3.14.	Compartir evento en Twitter	3
3.15.	Compartir evento en LinkedIn	4
3.16.	Implementar mapa del evento integrado con Google maps	5
3.17.	Compartir evento por email	4
Bugs		6
<b>Subtotal</b>		<b>79</b>
<b>4. Sprint 6</b>	<b>Registro a Eventos y Guardar Eventos</b>	
4.1.	Investigar la documentación de Stripe para adquirir conocimiento sobre la integración	8
4.2.	Crear markup de registro mediante mockups	6
4.3.	Validaciones del formulario de registro	6
4.4.	Implementar pago online	10
4.5.	Implementar pago offline	2
	Implementar conjunto de mensajes a mostrar al usuario dependiendo del resultado con respecto al pago de eventos	
4.6.	Implementar conjunto de mensajes a mostrar al usuario dependiendo del resultado con respecto al pago de eventos	2
4.7.	Investigar cómo se generan códigos QR	3
4.8.	Implementar envío de email con código	4
4.9.	Implementar pago con código de descuento	5

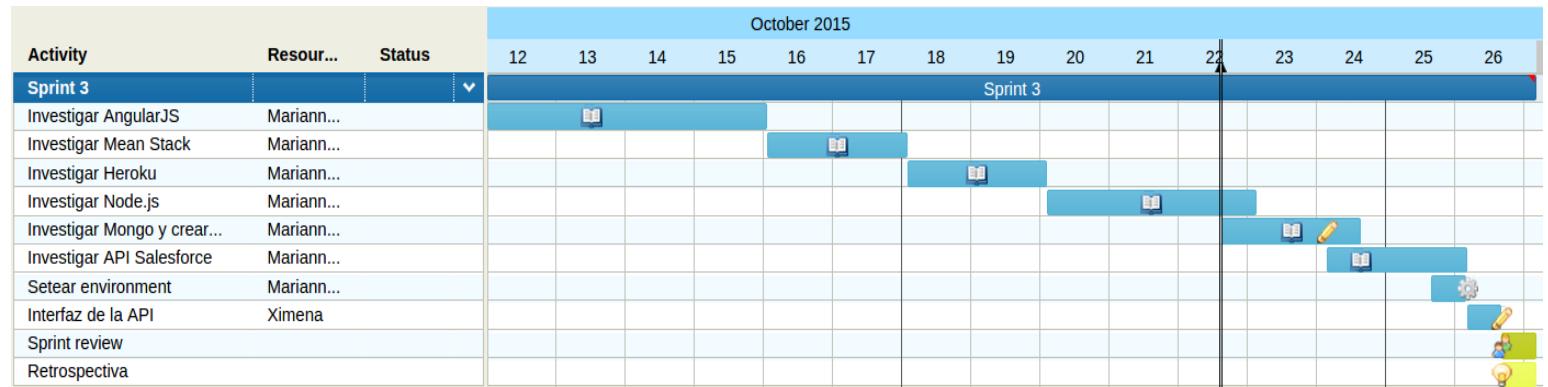
4.10.	Implementar guardado de eventos	4
4.11.	Implementar timeout para completar formulario de inscripción	8
Bugs		6
<b>Subtotal</b>	<b>64</b>	
<b>5. Sprint 7</b>	<b>Continuación Pago y My Account</b>	
5.1.	Investigar API Paypal	3
5.2.	Implementar Pago con Paypal	7
5.3.	Modificar datos personales de mi cuenta	5
5.4.	Modificar contraseña	3
5.5.	Implementar directorio de eventos guardados	5
5.6.	Eliminar evento guardado	4
5.7.	Implementar directorio de eventos a los que estoy registrado	4
5.8.	Cancelar un evento al que estoy registrado	8
5.9.	Imprimir info de pago	5
5.10.	Configuración y asociación de cuentas	10
Bugs		6
<b>Subtotal</b>	<b>60</b>	
<b>6. Sprint 8</b>	<b>Pasos finales</b>	
6.1.	Implementar sección de notificaciones	5
6.2.	Implementar sistema de notificaciones	11
6.3.	Deploy final a Heroku	2
6.4.	Testeo de la aplicación	5
6.5.	Confeccionar manual de usuario	4
6.6.	Finalizar documentación	12
Bugs		12
<b>Subtotal</b>	<b>51</b>	
<b>Total</b>	<b>410</b>	

## 2.5. Tabla del cronograma inicial de desarrollo

### 2.5.4. Diagrama de Gantt

Los siguientes diagramas se encuentran divididos por los sprints que abarca el proyecto.

## Sprint 3



2.6. Imagen de diagrama de Gantt sprint 3

## Sprint 4



2.7. Imagen de diagrama de Gantt sprint 4

## Sprint 5



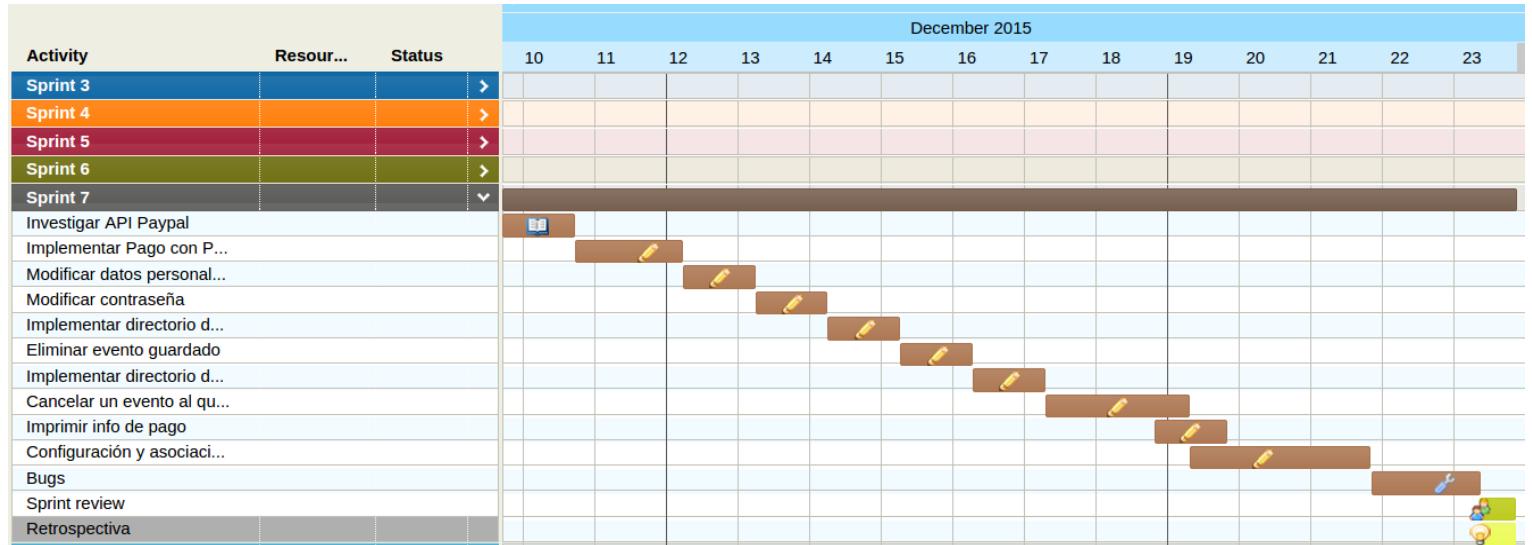
2.8. Imagen de diagrama de Gantt sprint 5

## Sprint 6



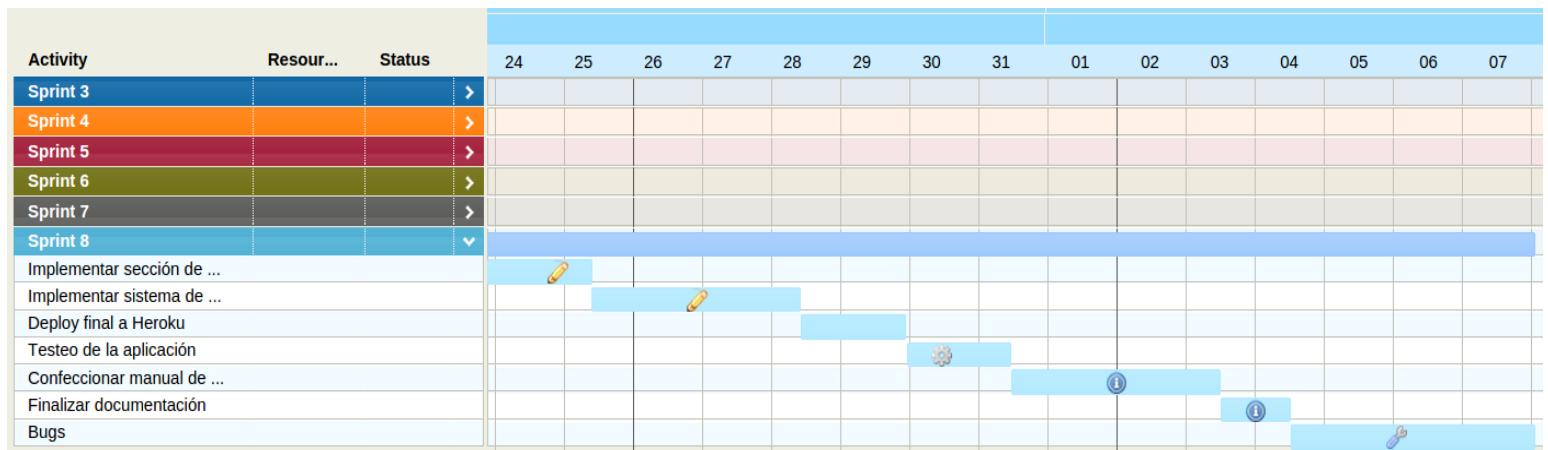
2.9. Imagen de diagrama de Gantt sprint 6

## Sprint 7



2.10. Imagen de diagrama de Gantt sprint 7

## Sprint 8



2.11. Imagen de Diagrama de Gantt sprint 8

Mediante el siguiente link se puede acceder al diagrama:  
<https://www.tomsplanner.com/public/rsvpday>. Contraseña: ProyectoFinal .

## 2.5.5. Estimación General del Proyecto

La estimación general del proyecto es de 410 horas a partir del tercer sprint hasta el octavo, o sea, 4 sprints (8 semanas). La entrega del resultado final del proyecto es para el día 17 de febrero de 2016, para cuando esperamos que todas la historias especificadas en el cronograma de desarrollo estén completadas. Sin embargo, y de la mano de lo que es el scrum, esperamos que existan adelantos y atrasos en el desarrollo del sistema en general. Por lo tanto, las historias de usuario se van a ir ajustando durante cada sprint de acuerdo al progreso del proyecto y al potencial cambio de requerimientos por parte del cliente.

# Capítulo 3

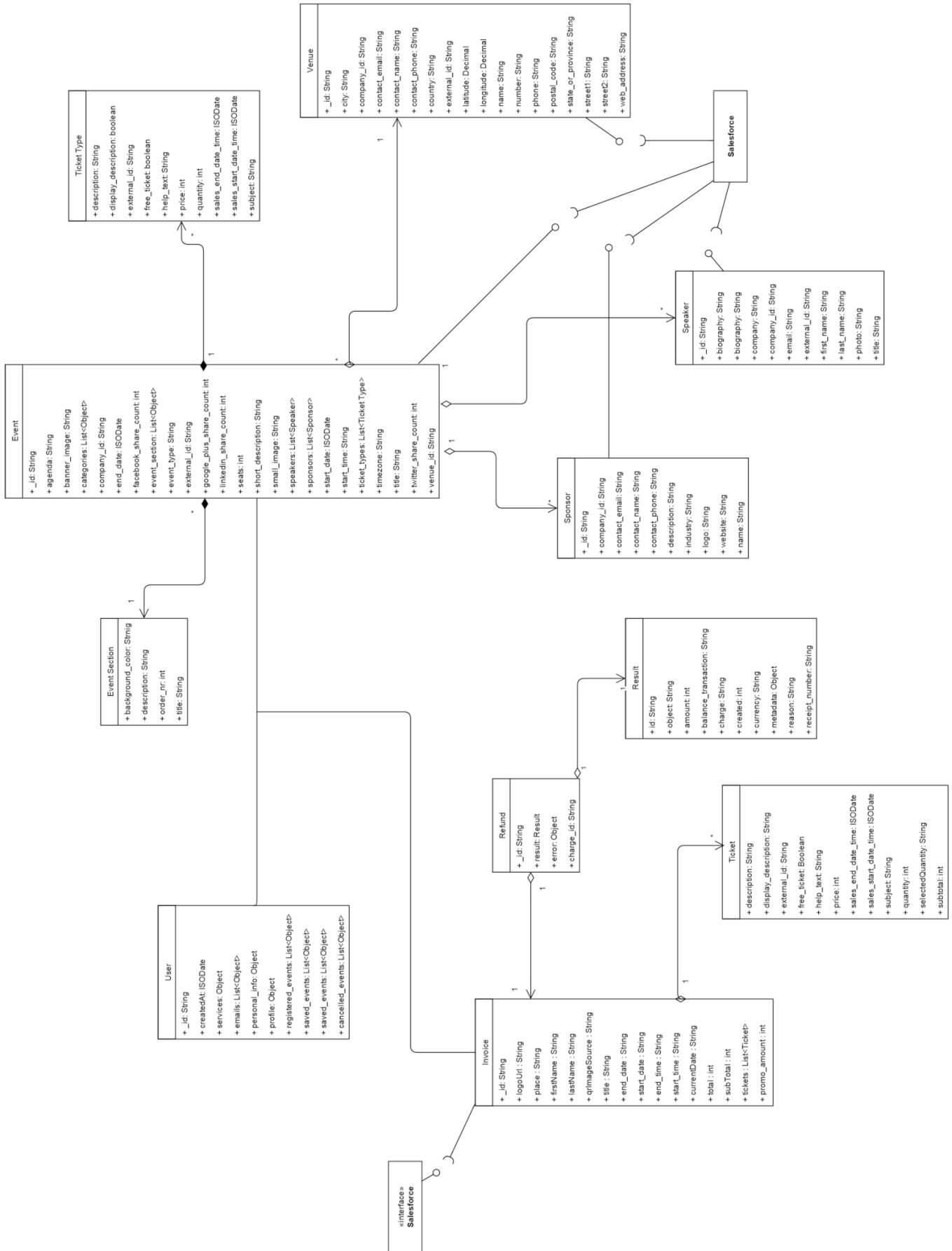
## Evolución del Anteproyecto

En el presente capítulo se muestran los cambios que hubo en el proyecto a medida que fue avanzando en relación a lo planeado en el anteproyecto. No se listan todas las secciones que presenta el anteproyecto ya que muchas de ellas se mantuvieron iguales.

### 3.1. Descripción del Entorno

#### 3.1.1. Diagrama Conceptual del Dominio

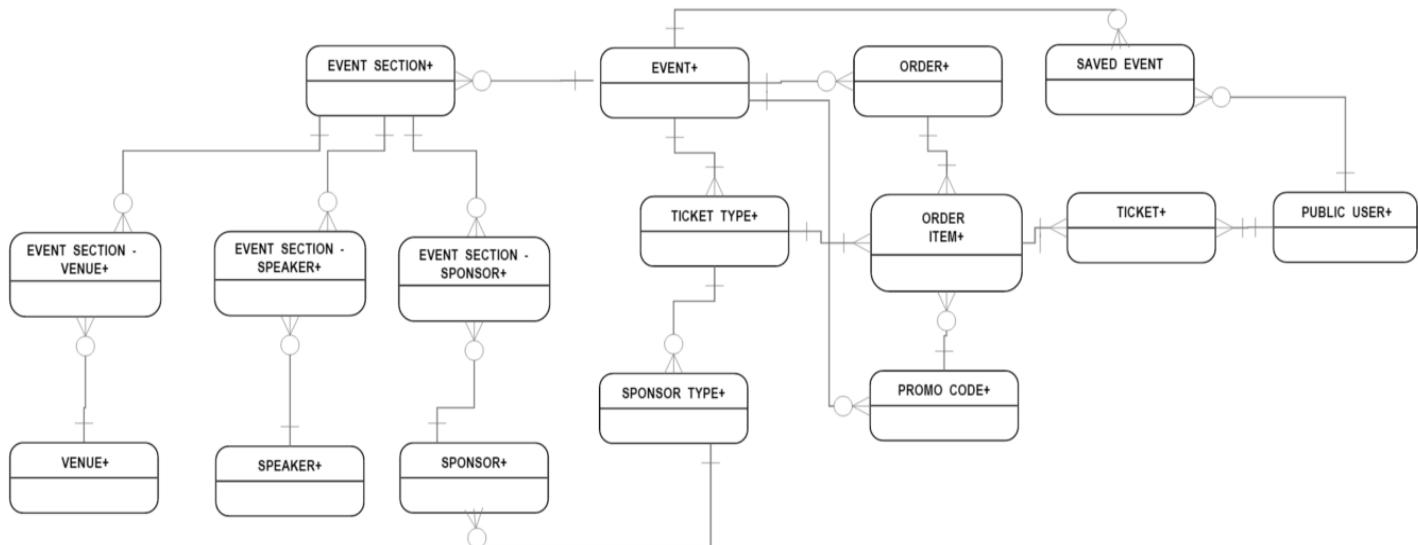
**UML del lado público de la aplicación:**



### 3.1. Imagen de UML sitio público

El diagrama del sitio público cambió con respecto al inicial, ya que en cada iteración lo fuimos ajustando a las necesidades del proyecto. La clase Usuario, se mantuvo con algunos cambios en sus atributos. La clase Factura cambió por Invoice y se incluye en el diagrama nuevo la clase Refund para la devolución de los pagos hechos, que en un principio no fue considerada. Asimismo, se agregó la clase Ticket y Event se desglosó en EventSection, Venue, Sponsor y Speaker que anteriormente formaban parte del mismo. Finalmente, se agregó TicketType que fue una de las últimas modificaciones para reflejar la existencia de distintos tipos de tickets.

### UML del Lado Privado de la Aplicación:



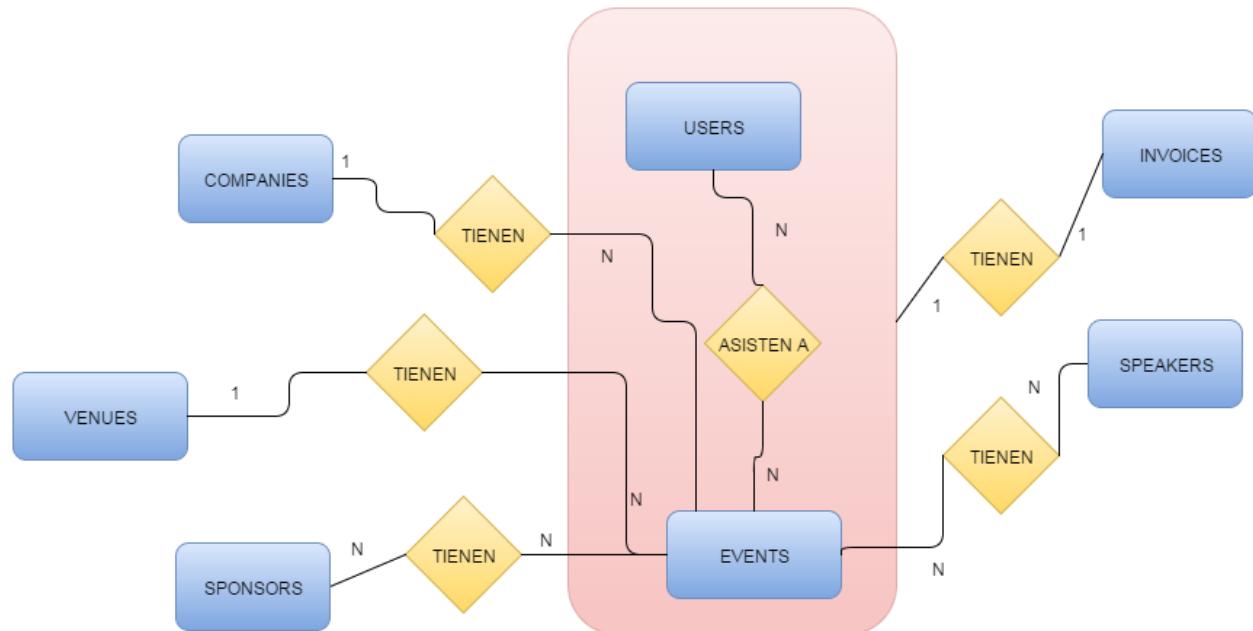
### 3.2. Imagen de UML sitio privado

El UML del sitio privado también sufrió modificaciones que provocaron que se cambiara el sitio público. Como parte de las mismas destacamos la división de Evento en diversas clases explicadas anteriormente y la creación de PromoCode para los descuentos en el registro a un evento.

### 3.1.2. Modelo Entidad Relación

#### MER del Lado Público de la Aplicación:

Este MER es el que se va a implementar.



#### 3.3. Imagen de MER sitio público

El MER sufrió modificaciones para reflejar los cambios en el dominio. Surgieron Speakers, Invoices, Sponsors y Venues manteniéndose Events, Users y Companies.

### 3.1.3. Estructura de los Documentos JSON

#### 3.1.3.1. Events

```
{  
  "_id": "a07G001012jugd1",  
  "agenda": "Morning: snacks. Afternoon: talks",  
  "banner_image": "",  
  "categories": [  
    {  
      "category_id": "013125"  
    }  
  ],
```

```

"company_id":"aaabbbccc",
"end_date":"2016-05-22T00:00:00.000+0000",
"end_time":"18:20",
"event_section": [
    {
        "background_color": "#4A6B8A",
        "description": "<h2>The best of Dreamforce is headed your way.</h2><p>Are you ready for a new kind of customer success? Join us for the all-new Salesforce World Tour in Los Angeles on December 3 and see how to connect with your customers in a whole new way. Access the best speakers, sessions, keynotes, and more - for FREE! Learn firsthand from leading customers how to sell, service, market, and succeed like never before.</p><p><a href='http://www.rsvpd.com'>Register now</a> for a full day of innovation and learning.</p><h2>See new products up close.</h2><p>Learn more about the new products just announced at Dreamforce, including Lightning, SalesforceIQ, and IoT Cloud powered by Thunder.</p>",
        "order_nr": 1,
        "title": "Overview"
    },
    {
        "background_color": "#728CA6",
        "description": "<h2>Agenda.</h2><p>Hear from Salesforce, and other company executives, plus industry-leading customers, partners, and product experts who will show you how to transform your business. We'll also feature customers who use Salesforce products to connect to their customers in a whole new way. <a href='http://www.rsvpd.com'>Register now</a>.</p><p><strong>8:00 a.m. - 6:00 p.m.</strong>Registration & Customer Success Expo*<h4>8:30 a.m. - 9:10 a.m.</h4>Breakout Sessions**<h4>9:45 a.m. - 10:00 a.m..</h4>Pre-show with Peter Coffee, VP of Strategic Research at Salesforce<h4>10:00 a.m. - 11:30 a.m..</h4>Keynote with Patrick Blair, Executive Vice President of Commercial Sales at Salesforce, and special guest speakers<strong>11:30 a.m. - 1:00 p.m..</strong><h4>Lunch in Customer Success Expo<h4>1:00 p.m. - 4:40 p.m..</h4>Breakout Sessions**<h4>4:40 p.m. - 6:00 p.m..</h4>Networking Reception in the Customer Success Expo</p><p>*Customer Success Expo will be closed during the keynote.<br />**All breakout sessions are first come, first served.</p>",
    }
]

```

```
        "order_nr":2,
        "title":"Agenda"
    },
{
    "background_color":"#4A6B8A",
    "description":"<h2>8:45 a.m. - 9:25 a.m.</h2><p> AppExchange  
for Everyone: Apps That Make a Huge Difference With over 2,800 apps on  
the AppExchange you can run your entire business on Salesforce. In  
this session you will hear customers share best the ways to extend  
Sales and Service Cloud as well as how AppExchange can power solutions  
for every department with free and paid apps. Learn why over 79% of  
Salesforce customers use AppExchange apps. Introducing the Salesforce  
IoT Cloud. How do you connect the internet of things with the Customer  
Success Platform? Join Adam Bosworth, EVP of the Salesforce IoT Cloud  
to learn how companies of all sizes are connecting data from their  
devices, sensors, websites and more using our Latest Salesforce  
technology and turning this data into meaningful actions for their  
customers. </p>",
        "order_nr":3,
        "title":"Sessions"
    },
{
    "background_color":"#728CA6",
    "description":"FAQ Got questions? We've got answers. Is there  
a cost to attend? The Salesforce World Tour is a free event! Register  
today. Where will the event take place? Salesforce World Tour Los  
Angeles will take place at Hollywood and Highland Center. What is the  
venue address? 6801 Hollywood Blvd, Los Angeles, CA 90028. What are  
the parking options? Salesforce World Tour will provide full-day  
validation for self-parking in the Hollywood and Highland Center.  
There are entrances on Highland Avenue and Orange Drive. What are the  
public transportation options? The Hollywood and Highland Center can  
be accessed via the Metro Red Line subway. ",
        "order_nr":4,
        "title":"FAQ"
    },
{
    "background_color":"#4A6B8A",
    "description":"This is some information for the speakers  
event section",

```

```

        "order_nr":5,
        "title":"Speakers"
    },
    {
        "background_color":"#728CA6",
        "description":"This is some information for the sponsors
event section",
        "order_nr":6,
        "title":"Sponsors"
    }
],
"event_type":"Symposium",
"external_id":"a00B00000003Rm00",
"facebook_share_count":0,
"google_plus_share_count":0,
"linkedin_share_count":0,
"seats":1400,
"short_description":"The best of Dreamforce is headed your way.",
"small_image":"/images/event1.jpg",
"speakers":[
    {
        "speaker_id":"aaabbdddzzz"
    },
    {
        "speaker_id":"aaabbccczzz"
    }
],
"sponsors":[
    {
        "sponsor_id":"aaabbccddd"
    },
    {
        "sponsor_id":"aaabbccceee"
    }
],
"start_date":"2016-05-21T00:00:00.000+0000",
"start_time":"06:20",
"ticket_types":[
    {
        "description":"This an early bird ticket price",

```

```

        "display_description":true,
        "external_id":"a08B0000002S9yF",
        "free_ticket":false,
        "help_text":"This an early bird ticket price",
        "price":300,
        "quantity":4982,
        "sales_end_date_time":"2016-03-31T03:00:00.000+0000",
        "sales_start_date_time":"2016-02-01T03:00:00.000+0000",
        "subject":"Not Profit Price"
    },
    {
        "description":"For regular for profit company",
        "display_description":true,
        "external_id":"a08B0000002SExJ",
        "free_ticket":false,
        "help_text":"For regular for profit company",
        "price":900,
        "quantity":4992,
        "sales_end_date_time":"2016-01-30T03:00:00.000+0000",
        "sales_start_date_time":"2016-01-08T03:00:00.000+0000",
        "subject":"Full Price Ticket"
    },
    {
        "description":"For regular for profit company",
        "display_description":true,
        "external_id":"a08B0000002SExY",
        "free_ticket":false,
        "help_text":"For regular for profit company",
        "price":600,
        "quantity":5000,
        "sales_end_date_time":"2016-01-08T03:00:00.000+0000",
        "sales_start_date_time":"2016-01-01T03:00:00.000+0000",
        "subject":"Full Price Ticket (Early Bird)"
    }
],
"timezone":"America/Los_Angeles",
"title":"Salesforce World Tour Atlanta",
"twitter_share_count":0,
"venue_id":"a23G000000j2tr3"
}

```

### 3.1.3.2. Users

```
{  
    "_id": "iqBJvRHMcPtXXn4uc",  
    "createdAt": {  
        "$date": "2016-02-09T05:19:26.701Z"  
    },  
    "services": {  
        "google": {  
            "accessToken": "  
ya29.gwJnC4GiMypmjxWMS204hrUJUAADXa8aoZUlex_L9L2Monx8q6R_hxe4c0q9kFk1  
QM4udlU",  
            "idToken": "  
eyJhbGciOiJSUzI1NiIsImtpZCI6IjExOTVkJh1NDlhNTU3MWM4ZTBhYjg3OWU3MGYyY  
2JiYzg0ZDZhYmQifQ.eyJpc3MiOiJhY2NvdW50cy5nb29nbGUuY29tIiwiYXRfaGFzaCI6  
I1lIckpNZThYeldHbjhBU0dpdmp0alEiLCJhdWQiOii5NDExMTk5Mjk2ODAtY3F2ZThhZD  
c1bXNtN2FhdDRtYm9zN2M0Y29oY2s5cXYuYXBwcy5nb29nbGV1c2VyY29udGVudC5jb20i  
LCJzdWIiOiIxMDkyOTc3MTgzMDEzMdg3Nzc0MTQiLCJlbWFpbF92ZXJpZmlIZCI6dHJ1ZS  
wiYXpwIjoiOTQxMTE50TI5",  
            "expiresAt": 1454998766008,  
            "scope": [  
                "https://www.googleapis.com/auth/userinfo.email",  
                "https://www.googleapis.com/auth/userinfo.profile"  
            ],  
            "id": "109297718301308777414",  
            "email": "juanDoe@gmail.com",  
            "verified_email": true,  
            "name": "Juan Doe",  
            "given_name": "Juan",  
            "family_name": "Doe",  
            "picture": "https://lh6.googleusercontent.com/-  
EXSNh7VRVRo/AAAAAAAIAAI/AAAAAAAAh4/HtTgzHlHxoE/photo.jpg",  
            "locale": "es-419",  
            "gender": "male"  
        },  
        "resume": {  
            "loginTokens": [  
                {  
                    "when": {  
                        "at": "2016-02-09T05:19:26.701Z"  
                    }  
                }  
            ]  
        }  
    }  
}
```

```

        "$date": "2016-02-09T05:19:26.709Z"
    },
    "hashedToken":
"Qvh2yim0L7ZzSMf9DU7TbQ60/T7gXXs+3Y1JUX+aAzI="
}
]
}
},
"emails" : [
{
    "address" : "juanDoe@gmail.com",
    "verified" : true
}
],
"personal_info" : {
    "full_name" : "Juan Doe",
    "telephone_number" : "8652334234",
    "mailing_address" : "123 Street",
    "mailing_city" : "San Francisco",
    "mailing_state" : "California",
    "billing_address" : "123 Street",
    "billing_city" : "San Francisco",
    "billing_state" : "California"
},
"profile": {
    "name": "Juan Doe",
    "image" : "zZ3eNWpbXYWhJrHW7"
},
"saved_events": [
{
    "event_id" : "a07G001012jugd1"
}
],
"registered_events": [
{
    "event_id": "a07G050019jEgd1",
    "stripe_id": "",
    "invoice_id": null,
    "salesforce_order_id": "a07G0000fR11122IQA"
}
]
```

```

        ],
      "cancelled_events": [
        {
          "event_id": "a07G050019jEgd1",
          "stripe_cancellation_id": ""
        }
      ]
    }
  }
}

```

### 3.1.3.3. Sponsors

```

{
  "_id" : "aaabbcccccfff",
  "company_id" : "aaabbccccc",
  "contact_email" : "steve@apple.com",
  "contact_name" : "Steve Jobs",
  "contact_phone" : "4578 956 235",
  "description" : "Apple",
  "industry" : "Computers",
  "logo" :
  "https://upload.wikimedia.org/wikipedia/commons/thumb/8/84/Example.svg
  /600px-Example.svg.png",
  "name" : "Apple",
  "website" : "www.apple.com"
}

```

### 3.1.3.4. Speakers

```

{
  "_id" : "aaabbcccccxxx",
  "biography" : "Jane is the daughter of a carpenter and teacher.  
She studied in Harvard. She then became president of the world...",
  "company" : "The World",
  "company_id" : "aaabbccccc",
  "email" : "janetest@gmail.com",
  "external_id" : "a07G0000fR11125",
  "first_name" : "Jane",
  "last_name" : "Test",
  "photo" : "/images/woman2.jpg",
}

```

```
        "title" : "President of the World"
    }
```

### 3.1.3.5. Venues

```
{
    "_id" : "a23G000000j2tr3",
    "city" : "Santa Barbara",
    "company_id" : "aaabbbccc",
    "contact_email" : "meeting@gmail.com",
    "contact_name" : "John Test",
    "contact_phone" : "001-545-589",
    "country" : "United States",
    "external_id" : "a07G0000fR11131",
    "latitude" : 34.417415,
    "longitude" : -119.670726,
    "name" : "Hyatt Santa Barbara",
    "number" : "1111",
    "phone" : "002-225-226",
    "postal_code" : "93103",
    "state_or_province" : "California",
    "street1" : "E Cabrillo Blvd",
    "street2" : "Other Street",
    "web_address" : "www.blabla.com"
}
```

### 3.1.3.6. Invoices

```
{
    "_id" : "XxxPE74SB2sN3eQDJ",
    "logoURL" : "http://localhost:3000/images/RSVPLogoHorizontal.png",
    "firstName" : "Ximena",
    "lastName" : "Lasserre",
    "qrImageSource" :
    "http://localhost:3000/api/eventQr.gif?eventId=a07G001012jugd1",
    "title" : "Salesforce World Tour Atlanta",
    "end_date" : " May 22, 2016",
    "start_date" : " May 21, 2016",
```

```

        "end_time" : "18:20",
        "start_time" : "06:20",
        "place" : "Hyatt Santa Barbara, undefined E Cabrillo Blvd, Santa
Barbara, California",
        "currentDate" : " February 08, 2016",
        "total" : NumberInt(300),
        "subTotal" : NumberInt(300),
        "tickets" : [
            {
                "description" : "This an early bird ticket price",
                "display_description" : true,
                "external_id" : "a08B0000002S9yF",
                "free_ticket" : false,
                "help_text" : "This an early bird ticket price",
                "price" : NumberInt(300),
                "sales_end_date_time" : ISODate("2016-03-
31T03:00:00.000+0000"),
                "sales_start_date_time" : ISODate("2016-02-
01T03:00:00.000+0000"),
                "subject" : "Not Profit Price",
                "quantity" : NumberInt(4976),
                "selectedQuantity" : "1",
                "subtotal" : NumberInt(300)
            }
        ],
        "promo_amount" : NumberInt(0),
        "userEmail" : "juanDoe@gmail.com",
        "eventId" : "a07G001012jugd1"
    }
}

```

### 3.1.3.7. Refunds

```
{
    "_id" : "imiaWsW4Bugkehmx",
    "result" : {
        "id" : "re_17c31fIANrNNTHRsLigh3AEs",
        "object" : "refund",
        "amount" : NumberInt(30000),
        "balance_transaction" : "txn_17c31fIANrNNTHRsTSwcTnsQ",
        "charge" : "ch_17c31NIANrNNTHRs8alXTLb6",
    }
}
```

```

    "created" : NumberInt(1454891831),
    "currency" : "usd",
    "metadata" : {

    },
    "reason" : null,
    "receipt_number" : null
},
"error" : null,
"charge_id" : "ch_17c31NIArNNTHRs8a1XTLb6"
}

```

## 3.2. Evolución de Herramientas y Tecnologías

El siguiente es un listado de las tecnologías que se terminaron usando en el proyecto. Algunas son las mismas que las planeadas en el anteproyecto y otras cambiaron a causa de distintas razones a lo largo del proyecto.

- **Salesforce** - Software para la administración de la relación con los clientes, CRM, por sus siglas en inglés. Se utilizará la Force.com REST API que es una API de servicios web para interactuar con Force.com (en nuestro caso particular, con el sitio privado de la aplicación).
- **Meteor** - Meteor es un framework basado en Javascript que sirve tanto para escribir código frontend como backend, por esta razón decidimos utilizarlo. Además hace bastante fácil el desarrollo de una aplicación web así como la conexión con la base de datos (MongoDB). También produce código multiplataforma, lo cual es muy conveniente para cuando en el futuro el cliente quiera desarrollar esta aplicación para dispositivos móviles. Asimismo, contiene una serie de paquetes que facilitan el codificado de ciertas funcionalidades.
- **AngularJS** - Framework de Javascript con el que se escribió la mayoría del código. Es de código abierto y mantenido por Google y está basado en el patrón MVC (Modelo-Vista-Controlador: es un patrón de arquitectura de software que consiste en separar los datos y la lógica de la vista y del módulo de eventos y comunicación).

- **MongoDB** - Es una base de datos NoSQL orientada a documentos. Las estructuras de datos se guardan en documentos tipo JSON. Esto permite que la integración con la base de datos se haga de una manera más fluida y simple.
- **Heroku** - Es una *Plataforma como Servicio* (*PaaS* por sus siglas en inglés) en la nube que nos permite hacer deploy de la aplicación y correrla desde allí.
- **HTML** - *HyperText Markup Language*. Lenguaje de marcado para elaborar páginas web.
- **CSS** - *Cascading Style Sheets*. Lenguaje para definir la presentación o vista del HTML.
- **Javascript** - Es un lenguaje de programación dinámico, generalmente usado como parte del navegador web para desarrollar mejoras en la interfaz de usuario y páginas web dinámicas.
- **Bootstrap** - En un principio la idea fue diseñar el estilo de la aplicación con Foundation, pero más adelante se decidió hacerlo con Bootstrap ya que ambos frameworks son muy potentes, pero ambas estudiantes tenemos más conocimiento de Bootstrap por ser más popular y usado y permitía llegar al mismo fin que Foundation.
- **Stripe API** - Stripe es un conjunto de APIs que permite gestionar pagos, por lo tanto facilita la programación en dicho sentido.
- **Facebook Login API** - API de Facebook que permite a los usuarios hacer log in en una aplicación sin tener que ingresar ningún dato, nada más que asociar su cuenta de Facebook a dicho sistema.
- **Google Sign-In API** - API de Google que posibilita a los usuarios hacer log in en una página utilizando su cuenta de Google.
- **Google Maps API** - API de Google que posibilita utilizar mapas de Google en la aplicación.
- **Git y Github** - Se utilizaron para mantener el versionado del código. Nos decidimos por esta herramienta ya que es la de uso del cliente y al mismo tiempo nos sentimos cómodas utilizándola. Además de que es práctica y confiable.

- **Apex** - Apex es un lenguaje de programación para programar en Salesforce. Es muy similar a Java y se utilizó para crear la REST API que recibe los datos de las compras.

### 3.3. Plan de Testing

En un principio el plan de testing iba a consistir en utilizar distintos frameworks para poder testear tanto el lado del cliente como el del servidor. Algunos de ellos eran Jasmine, Mocha, Karma. Sin embargo, esto fue pensado al principio del proyecto, cuando la idea era utilizar Node.js y AngularJS, pero al vernos obligadas a cambiar de framework para el desarrollo y utilizar Meteor las herramientas para hacer tests unitarios también cambiaron, así como la forma de usarlas.

La idea principal era usar Velocity, que es el framework oficial de testing para Meteor y hacer los unit tests con Jasmine, que viene integrado a Velocity. Sin embargo, a medida que fueron pasando los sprints y los tiempos quedaban más cortos no nos fue posible implementar los unit tests, ya que implicaba otra curva de aprendizaje para poder utilizar Jasmine junto con Meteor y AngularJS.

A pesar de esto sí fueron escritos casos de prueba para toda la aplicación y se hicieron tests manuales basándose en los mismos. Y también se realizó testing exploratorio para tratar de encontrar bugs que no se puedan encontrar a simple vista o que no sean fáciles de encontrar siguiendo los flujos de los casos de prueba que teníamos.

### 3.4. Planificación

#### 3.4.1. Metodología

En un principio, la metodología de planificación a utilizar en el presente proyecto era scrum. La misma es una metodología de desarrollo ágil para gestionar el desarrollo de un producto. Uno de los principios fundamentales del scrum es que durante el proceso de producción el cliente puede cambiar de parecer en cuanto a lo que desea o necesita del producto y por lo tanto el equipo debe reaccionar de manera rápida al cambio en el proyecto. Este es un problema en otras metodologías en las que es más complicado aceptar y cambiar rápidamente si cambian los requerimientos del cliente.

El scrum está dividido en sprints. Estos son iteraciones de una o dos semanas en las que se planifican las tareas en las que trabajará cada uno y se trabaja en las mismas. Al final de cada uno se hace una reunión retrospectiva del sprint para revisar el trabajo completado y lo que no pudo ser completado.

Al ir avanzando en el proyecto, fuimos encontrando algunos obstáculos que no nos permitieron utilizar scrum propiamente dicho. Sin embargo, pudimos utilizar algunos de sus conceptos como son algunas de sus ceremonias (la planificación y la revisión) y artefactos (sprint, backlog del producto y del sprint). A la hora del seguimiento del estado de las tareas seguimos algunos conceptos de la metodología Kanban, ya que las marcamos como en proceso, finalizadas, aceptadas o declinadas. Para el manejo del backlog utilizamos la herramienta Pivotal Tracker.

### 3.4.2. Cronograma de Desarrollo Final: Sprint 3 al 8

Al considerar el esfuerzo estamos considerando también el testing de la historia.

Sprint/Número Tarea	Historia	Esfuerzo Estimado	Esfuerzo Real	Developer
<b>1. Sprint 3</b>				
1.1.	Investigar AngularJS	16	12	Ambas
1.2.	Investigar Node.js	12	8	Ambas
1.3.	Investigar Mongo	5	5	Ambas
1.4.	Investigar Mean Stack	8	14	Ambas
1.5.	Investigar Heroku	8	6	Ambas
1.6.	Investigar Foundation	4	2	Ambas
1.7.	Investigar Streaming API de Salesforce	2	2	Ambas
1.8.	Investigar API Salesforce (Apex REST)	8	2	Ambas
1.9.	Setear environment	2	6	Ambas
<b>Subtotal</b>		<b>65</b>	<b>57</b>	
<b>2. Sprint 4</b>				
2.1.	Investigar Meteor vs Sails y setear el environment	4	5	Ambas
2.2.	Investigar Foundation vs Bootstrap	2	1	Ambas

2.3.	Crear datos de prueba en Mongo	2	2	Ximena
2.4.	Crear layout del sitio	2	2	Marianne
2.5.	Crear markup de directorio de eventos	6	9	Ximena
2.6.	Filtrar eventos por nombre	6	4	Marianne
2.7.	Filtrar eventos por fecha	4	5	Marianne
2.8.	Filtrar eventos por tipo	4	2	Marianne
2.9.	Ordenar eventos por nombre y precio	3	3	Ximena
2.10.	Filtrar eventos por location	5	5	Marianne
2.11.	Paginar eventos (con load more)	5	7	Ximena
2.12.	Investigar API de Google para loguearse	4	2	Marianne
2.13.	Investigar API de Facebook para loguearse	4	2	Marianne
2.14.	Diseñar login (método regular)	6	4	Marianne
2.15.	Añadir Facebook login al login	8	4	Marianne
2.16.	Añadir Google SignIn al login	8	4	Marianne
Bugs		8	2	
<b>Subtotal</b>		<b>81</b>	<b>63</b>	
<b>3. Sprint 5</b>				
3.1.	Paginar eventos (con load more)	5	6	Ambas
3.2.	Investigar Restivus para permitir el envío de datos desde Salesforce e implementar	5	5	Ximena
3.3.	Crear base de datos	8	5	Ambas
3.4.	Actualizar jsons y código para que se ajuste a las nuevas colecciones separadas	4	4	Ximena
3.5.	Modificar lista de eventos de acuerdo al mockup	5	5	Ximena
3.6.	Crear markup de detalle de evento (overview)	7	9	Ximena
3.7.	Crear markup de detalle de evento (speakers)	7	5	Ximena
3.8.	Crear barra de navegación para el evento	3	3	Ximena
3.9.	Investigar cómo funcionan APIS de calendarios Google Calendar,	5	5	Ximena

	Yahoo Calendar, Microsoft Outlook, Apple iCal			
3.10.	Implementar adición del evento al calendario de elección	5	5	Ximena
3.11.	Filtrar eventos por precio	3	3	Marianne
3.12.	Investigar cómo generar pdfs	3	3	Marianne
3.13.	Agregar burbujas con los filtros aplicados	3	4	Marianne
Bugs		6	2	
<b>Subtotal</b>		<b>69</b>	<b>64</b>	
<b>4. Sprint 6</b>				
4.1.	Crear markup de detalle de evento (sponsors)	7	3	Ximena
4.2.	Investigar API Facebook para realizar y compartir publicaciones	2	1	Marianne
4.3.	Investigar API Twitter para realizar y compartir publicaciones	2	1	Ximena
4.4.	Investigar API LinkedIn para realizar y compartir publicaciones	2	1	Marianne
4.5.	Investigar API Google+ para realizar y compartir publicaciones	2	1	Ximena
4.6.	Investigar opciones para envío de emails	3	4	Marianne
4.7.	Compartir evento en Facebook	4	4	Marianne
4.8.	Compartir evento en Twitter	4	3	Ximena
4.9.	Compartir evento en LinkedIn	4	4	Marianne
4.10.	Compartir evento en Google+	4	1	Ximena
4.11.	Implementar mapa del evento integrado con Google maps	5	7	Ximena
4.12.	Implementar markup login	3	3	Marianne
4.13.	Imprimir detalle de evento en pdf	5	5	Marianne
4.14.	Crear markup para secciones dinámicas	3	4	Ximena
4.15.	Agregar timezone a json	1	1	Ximena
4.16.	Crear markup de registro mediante mockups	6	6	Ximena
4.17.	Compartir evento por email	6	4	Marianne
4.18.	Implementar confirmación de email mediante envío de link	6	6	Marianne

Bugs		6	3
<b>Subtotal</b>		<b>75</b>	<b>62</b>
<b>5. Sprint 7</b>			
5.1.	Actualizar header para que luzca como el mockup.	1	1 Ximena
5.2.	Implementar guardado de eventos	4	4 Ximena
5.3.	Eliminar evento guardado	4	2 Ximena
5.4.	Implementar directorio de eventos guardados	5	4 Ximena
5.5.	Cambiar markup del login para que se parezca a mockup y sea customizable	5	12 Marianne
5.6.	Implementar directorio de eventos a los que estoy registrado	4	3 Ximena
5.7.	Investigar cómo se generan códigos QR	3	5 Marianne
5.8.	Implementar formulario para modificar datos personales de mi cuenta	4	3 Ximena
5.9.	Implementar envío de email con código	4	8 Marianne
5.10.	Implementar directorio de eventos pasados	4	4 Ximena
5.11.	Crear formulario (pop up) de contacto con organizador	4	5 Marianne
5.12.	Validaciones del formulario de contacto con organizador	4	4 Marianne
5.13.	Modificar contraseña	3	2 Ximena
5.14.	Actualizar versión de paquete de angular y arreglar problemas que genera con templates	3	4 Ximena
5.15.	Cambiar botón de página de evento para que diga Get Tickets	1	1 Ximena
5.16.	Cambiar estructura de json para tickets	1	1 Ximena
Bugs		6	11
<b>Subtotal</b>		<b>60</b>	<b>74</b>
<b>6. Sprint 8</b>			
6.1.	Agregar foto de perfil a My Account	4	8 Ximena

6.2.	Hacer que se vea rango de precios de un evento en directorio de eventos y detalle de evento	3	3	Ximena
6.3.	Implementar pago online: Paso 1, modal en detalle de evento. Crear markup.	4	5	Ximena
6.4.	Implementar pago online: Paso 1, modal en detalle de evento. Lógica	3	4	Ximena
6.5.	Implementar pago online: Paso 2, resumen de orden y datos personales	5	5	Marianne
6.6.	Agregar columna derecha en registro para datos del evento	2	3	Ximena
6.7.	Modificar REST API (restivus) para recibir datos para que modifique fechas a date de JS	3	4	Ximena
6.8.	Investigar la documentación de Stripe para adquirir conocimiento sobre la integración	8	8	Marianne
6.9.	Implementar pago online: Paso 3, datos de pago: Stripe	6	6	Marianne
6.10.	Implementar pago online: Paso 3, datos de pago: Comunicación con la API	8	9	Ximena
6.11.	Implementar timeout para completar formulario de inscripción	3	4	Marianne
6.12.	Validaciones del formulario de registro a evento	6	5	Marianne
6.13.	Implementar conjunto de mensajes a mostrar al usuario dependiendo del resultado con respecto al pago de eventos	2	2	Marianne
6.14.	Agregar búsqueda de evento que sea accesible en todo el sitio	2	2	Ximena
6.15.	Implementar pago con código de descuento	5	5	Ximena
6.16.	Implementar API para mandar descuentos a salesforce	2	2	Ximena
6.17.	Imprimir info de pago	5	6	Ximena
6.18.	Implementar mensajes antes y	3	3	Marianne

	después de envío de mail de confirmación de registro al sitio			
6.19.	Modificar llamada a API cuando mandamos la orden y agregar datos de Stripe	1	1	Ximena
6.20.	Agregar evento a eventos registrados en el json luego de compra	2	2	Ximena
6.21.	Agregar página de agradecimiento por la compra luego del registro	3	4	Marianne
6.22.	Mandar mail con info de pago	4	12	Marianne
6.23.	Imprimir pdf en página de agradecimiento	4	5	Marianne
6.24.	Mejorar validaciones de teléfonos	1	1	Ximena
6.25.	Agregar burbujas por filtros y sorts	4	4	Marianne
6.26.	Eliminar el registro de un evento	5	7	Ambas
6.27.	Modificar columna de event detail y agregar banner image	1	1	Ximena
6.28.	Testeo de la aplicación	5	3	Ambas
6.29.	Deploy final a Heroku	1	1	Ambas
6.30.	Confeccionar manual de usuario	4	5	Ambas
6.31.	Crear presentación para defensa	6	6	Ambas
6.32.	Finalizar documentación	15	16	Ambas
Bugs		12	13	
<b>Subtotal</b>		<b>142</b>	<b>165</b>	
<b>Total</b>		<b>492</b>	<b>485</b>	

### 3.1. Tabla de cronograma final de desarrollo

# Capítulo 4

## Desarrollo y Conclusiones

### 4.1. Conclusiones del Anteproyecto

Durante las semanas dedicadas al anteproyecto (sprints 1 y 2) se trabajaron unas 130 horas. Estas se utilizaron en reuniones con el cliente, investigando tecnologías y herramientas para decidir qué solución podía ser la más adecuada, realizando demos, escribiendo documentación, haciendo estimaciones y realizando otras tareas de planificación.

En un principio y teniendo en cuenta lo discutido con el cliente se decidió utilizar *MEAN stack*, ya sea *MEAN IO* o *MEAN JS*. Pero durante el primer sprint (como se detallará más adelante) se terminó decidiendo utilizar Meteor, el cual es un framework que también combina frontend y backend. Las demás herramientas escogidas durante el anteproyecto se mantuvieron hasta la finalización del proyecto.

### 4.2. Seguimiento de los Sprints

#### 4.2.1. Sprint 3

La siguiente tabla muestra las tareas del sprint 3 junto con la estimación de tiempo y el tiempo real que tomó llevarlas a cabo. También se detalla quién la realizó.

1. Sprint 3				
1.1.	Investigar AngularJS	16	12	Ambas
1.2.	Investigar Node.js	12	8	Ambas
1.3.	Investigar Mongo	5	5	Ambas
1.4.	Investigar Mean Stack	8	14	Ambas
1.5.	Investigar Heroku	8	6	Ambas
1.6.	Investigar Foundation	4	2	Ambas
1.7.	Investigar Streaming API de Salesforce	2	2	Ambas
1.8.	Investigar API Salesforce (Apex REST)	8	2	Ambas

1.9.	Setear environment	2	6	Ambas
<b>Subtotal</b>		<b>65</b>	<b>57</b>	

#### 4.1. Tabla de cronograma sprint 3

##### 4.2.1.1. Detalle

El sprint 3 consistía de tareas de investigación más que nada y que por lo tanto debían ser realizadas por ambas desarrolladoras. La idea era familiarizarse con las herramientas que se iban a utilizar y evaluar si eran útiles para este caso o no.

##### 4.2.1.2. Conclusiones del Sprint

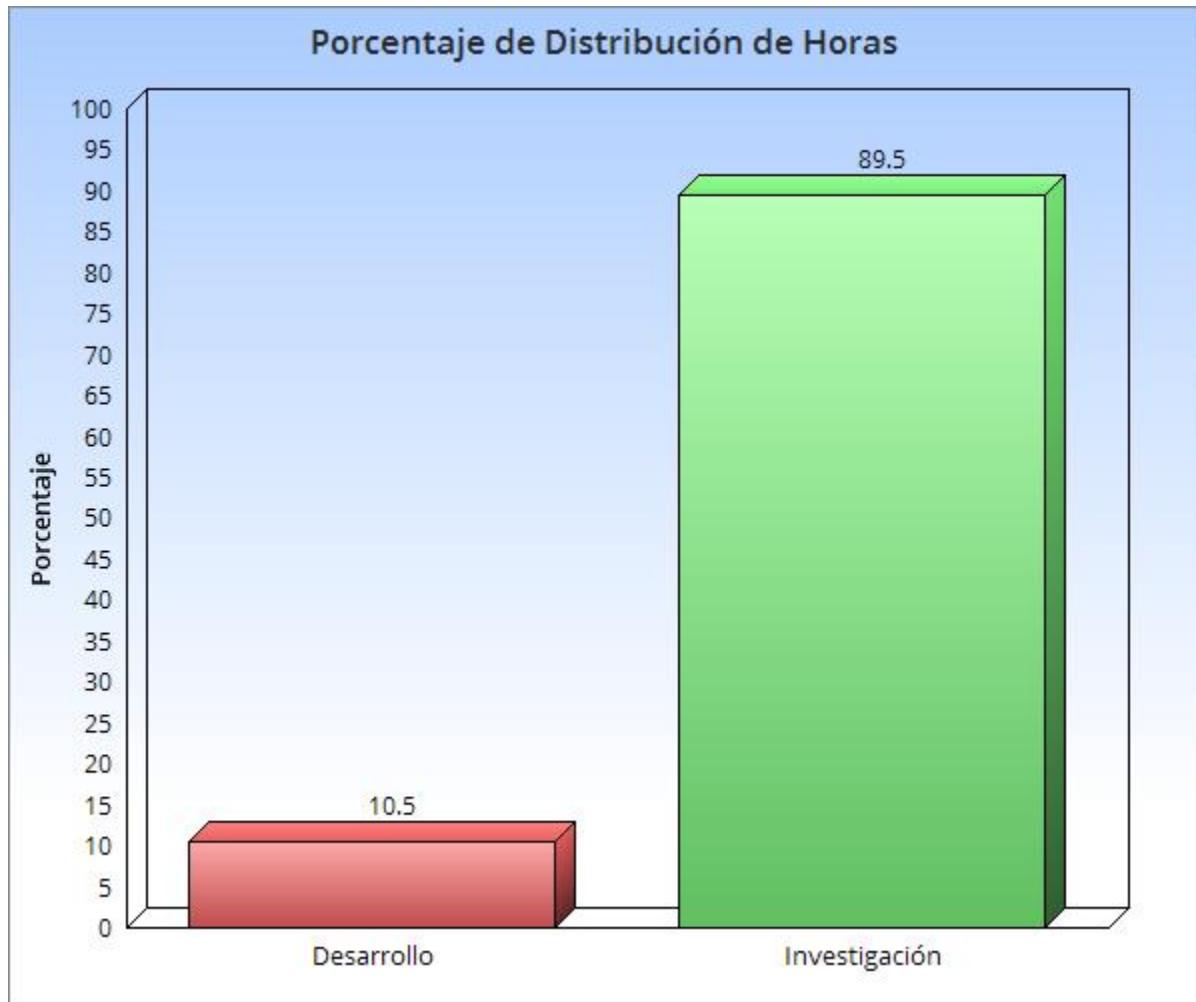
Las tareas en general se llevaron a cabo en orden, para todas se consultaron variadas fuentes en diferente medida, como documentación oficial, tutoriales, foros, cursos online. También se hicieron pequeñas implementaciones (demos) para evaluar la facilidad y compatibilidad de las mismas. Esto último fue lo que nos llevó a desistir de usar Mean Stack (ya sea mean.io o mean.js). Mean.io nos dio bastantes errores al momento de tratar de instalarlo y su integración con redes sociales no era la que esperábamos. Por otro lado, pudimos instalar Mean.js sin problemas, pero encontramos que uno de los problemas actuales del mismo es que hay un bug al momento de crear los módulos entonces no se crean todas las dependencias, rutas, etc. Y dado que era algo que estaban tratando de solucionar y no sabían cuánto tiempo podría pasar para que estuviese arreglado, nos pareció que era una mejor solución buscar otro framework. Esto, sin embargo, nos consumió bastante tiempo ya que no estaba planeado el cambio de framework. Luego de investigar otros frameworks que se adaptaran a las necesidades del proyecto se decidió utilizar Meteor.

Otra situación que se dio en el sprint fue que la investigación de Foundation tampoco pudo hacerse en su totalidad. Y nos pareció más razonable estudiar qué nos aportaba Foundation que no nos aportara Bootstrap que es un framework que ya ambas hemos utilizado.

Por lo tanto tuvimos que pasar algunas historias de usuario al sprint 4, estas son:

- **Investigar Meteor vs Sails y setear el environment** - 4 horas - Estos dos nuevos frameworks fueron los que decidimos investigar como solución al problema de MEAN stack.

- **Investigar Foundation vs Bootstrap** - 2 horas - Ya que teníamos experiencia previa trabajando con Bootstrap y dado que es inmensamente popular, nos pareció que tal vez nos convenía más utilizarlo, así que decidimos investigar las ventajas y desventajas de cada uno.



4.1. Imagen de gráfica de horas del sprint 3

#### 4.2.2. Sprint 4

La siguiente tabla muestra las tareas del sprint 4 junto con la estimación de tiempo y el tiempo real que tomó llevarlas a cabo. También se detalla quién las realizó.

2. Sprint 4				
2.1.	Investigar Meteor vs Sails y setear el environment		4	5 Ambas

2.2.	Investigar Foundation vs Bootstrap	2	1	Ambas
2.3.	Crear datos de prueba en Mongo	2	2	Ximena
2.4.	Crear layout del sitio	2	2	Marianne
2.5.	Crear markup de directorio de eventos	6	9	Ximena
2.6.	Filtrar eventos por nombre	6	4	Marianne
2.7.	Filtrar eventos por fecha	4	5	Marianne
2.8.	Filtrar eventos por tipo	4	2	Marianne
2.9.	Ordenar eventos por nombre y precio	3	3	Ximena
2.10.	Filtrar eventos por location	5	5	Marianne
2.11.	Paginar eventos (con load more)	5	7	Ximena
2.12.	Investigar API de Google para loguearse	4	2	Marianne
2.13.	Investigar API de Facebook para loguearse	4	2	Marianne
2.14.	Diseñar login (método regular)	6	4	Marianne
2.15.	Añadir Facebook login al login	8	4	Marianne
2.16.	Añadir Google SignIn al login	8	4	Marianne
Bugs		8	2	
<b>Subtotal</b>		<b>81</b>	<b>63</b>	

#### 4.2. Tabla de cronograma sprint 4

##### 4.2.2.1. Detalle

El sprint 4 consistía en implementar el registro, login de usuarios y el directorio de eventos. Hay algunas historias que por falta de tiempo en el sprint anterior pasaron a este sprint.

En este sprint se logró implementar el login e integración con redes sociales como son Facebook, Twitter y Google. Por otro lado, se implementó el directorio de eventos con datos de prueba y sus filtros.

Por el momento no se cuenta con casos de prueba definidos formalmente. Se ha ido testeando la aplicación de forma manual a medida que se desarrolla. Sin embargo, en próximos sprints se implementarán casos de prueba.

##### 4.2.2.2. Casos de Uso

Implementa casos de uso redes, CU-005, CU-006, CU-007, CU-008, CU-009, y CU-010. También hay algunas historias que implementaban alguna parte de otros casos de uso pero no la principal, por lo tanto no se mencionan.

Para ver el detalle de cada uno de estos casos de uso dirigirse al anexo A.2.

#### 4.2.2.3. Conclusiones del Sprint

El sprint 4 contiene historias del sprint 3, ya que la investigación de las tecnologías dio como resultado que las mismas no satisfacen las necesidades del proyecto. Por esta razón, en una primera instancia del sprint se realizó la configuración del ambiente para el desarrollo de las historias de usuario sobre el directorio de eventos y sus filtros.

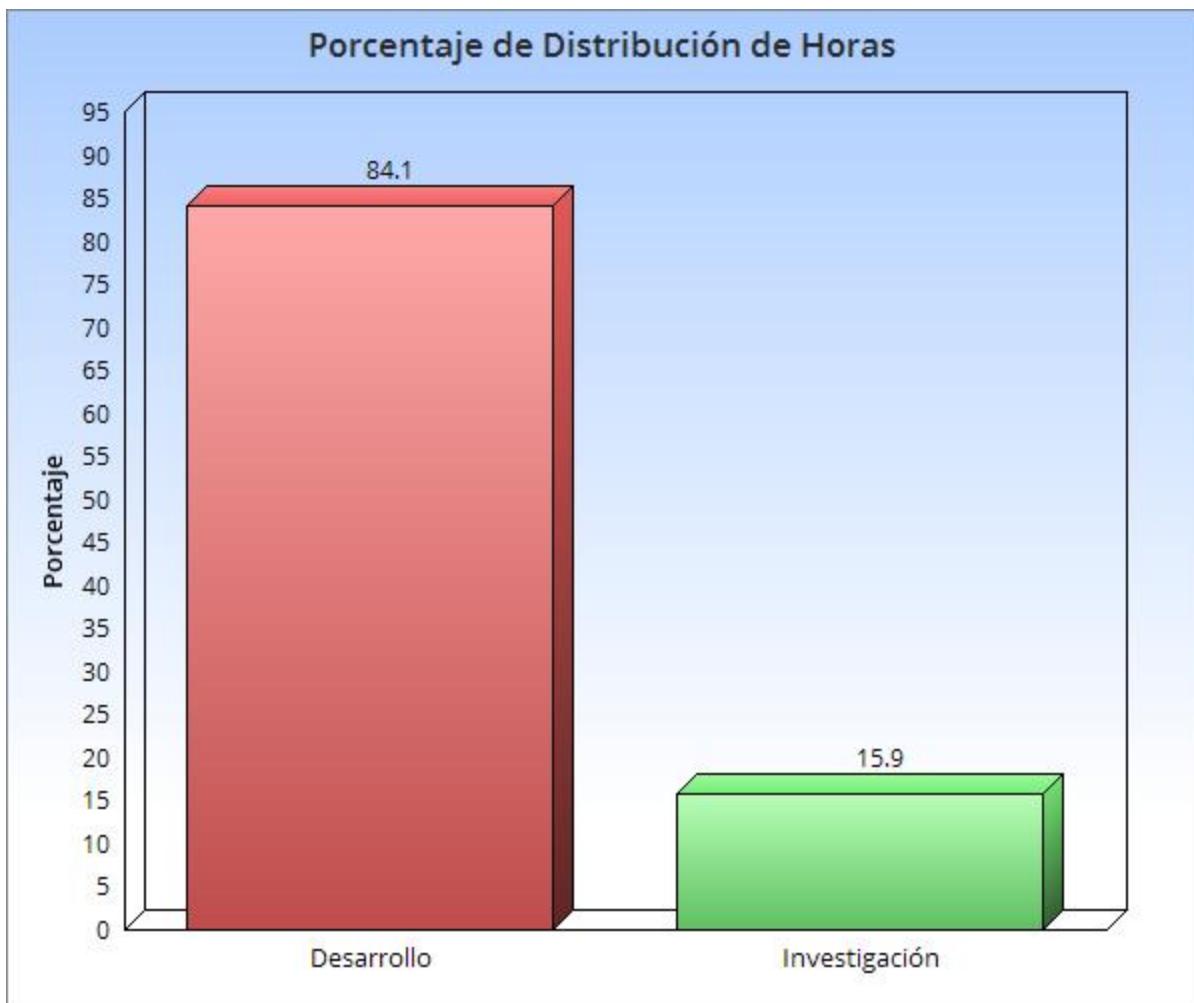
Con respecto a la tareas de investigación sobre Foundation y Bootstrap, decidimos utilizar Bootstrap debido a la familiaridad que contamos con la herramienta y su gran popularidad.

En base a la tabla anterior se puede observar una diferencia en cuanto a las horas estimadas y a las horas reales. Las tareas requirieron menos tiempo, ya que cuando fueron estimadas, la misma se realizó en base a la tecnologías seleccionadas inicialmente. Pudimos notar que la tecnología utilizada (Meteor) es más intuitiva, razón por la que el desarrollo de las tareas de filtrado y ordenamiento se nos facilitaron. Por otro lado, las horas dedicadas a bugs fueron 0 ya que los tests manuales no presentaron errores.

Otra razón para justificar la diferencia de horas, es la base de investigación previa al sprint. Al haber investigado e implementado demos, el trabajo de desarrollo no presentó mayores dificultades.

A pesar de que la cantidad de horas fue menor, hubo una tarea que pasa al siguiente sprint:

**Filtrar eventos por *location*** - Consiste en filtrar los eventos a mostrar en base a las *locations* del evento. Esta tarea no se logró desarrollar, ya que en ese momento el criterio de aceptación no estaba desarrollado.



**4.2. Imagen de gráfica de horas del sprint 4**

#### 4.2.3. Sprint 5

La siguiente tabla muestra las tareas del sprint 5 junto con la estimación de tiempo y el tiempo real que tomó llevarlas a cabo. También se detalla quién realizó cada tarea.

3. Sprint 5				
3.1.	Paginar eventos (con load more)	5	6	Ambas
3.2.	Investigar Restivus para permitir el envío de datos desde Salesforce e implementar	5	5	Ximena
3.3.	Crear base de datos	8	5	Ambas
3.4.	Actualizar jsons y código para que se	4	4	Ximena

	ajuste a las nuevas colecciones separadas			
3.5.	Modificar lista de eventos de acuerdo al mockup	5	5	Ximena
3.6.	Crear markup de detalle de evento (overview)	7	9	Ximena
3.7.	Crear markup de detalle de evento (speakers)	7	5	Ximena
3.8.	Crear barra de navegación para el evento	3	3	Ximena
3.9.	Investigar cómo funcionan APIS de calendarios Google Calendar, Yahoo Calendar, Microsoft Outlook, Apple iCal	5	5	Ximena
3.10.	Implementar adición del evento al calendario de elección	5	5	Ximena
3.11.	Filtrar eventos por precio	3	3	Marianne
3.12.	Investigar cómo generar pdfs	3	3	Marianne
3.13.	Agregar burbujas con los filtros aplicados	3	4	Marianne
Bugs		6	2	
<b>Subtotal</b>		<b>69</b>	<b>64</b>	

#### 4.3. Tabla de cronograma sprint 5

##### 4.2.3.1. Detalle

El Sprint 5 consistía en continuar con el desarrollo del directorio de eventos y comenzar a implementar lo que sería la página en donde se muestra la información de un evento solo, sin aun implementar la compra de tickets para ese evento, sino más bien crear el markup y ver la forma de implementar secciones dinámicas en la página.

En el directorio de eventos hubo que modificar el markup de la vista como lista, ya que al momento del sprint anterior todavía no estaba creado el mockup de esa vista. También hubo que agregar filtros para los eventos, como el de precios.

También se hizo una investigación y se implementó Restivus que es una REST API para permitir llevar a cabo el caso de recibir la información de los eventos, sponsors, speakers, etc. que se envían desde Salesforce.

Además de comenzar a desarrollar el markup de un evento se agregó la funcionalidad de crear un evento en el calendario personal del usuario. Sin embargo cuando esta se

subió al servidor no funcionaba (más allá de que funciona local), por lo tanto va a ser un bug para arreglar en los próximos sprints.

Por último, en cuanto a las secciones dinámicas, esto se refiere a que cada usuario puede decidir qué secciones tenga su evento y cada una de ellas se debe mostrar en el orden y de la forma deseada por el mismo.

#### 4.2.3.2. Casos de Uso

Implementa casos de uso redes, CU-005, CU-006, CU-011, CU-012 y CU-013. También hay algunas historias que implementaban alguna parte de otros casos de uso pero no la principal, por lo tanto no se mencionan.

Para ver el detalle de cada uno de estos casos de uso dirigirse al anexo A.2.

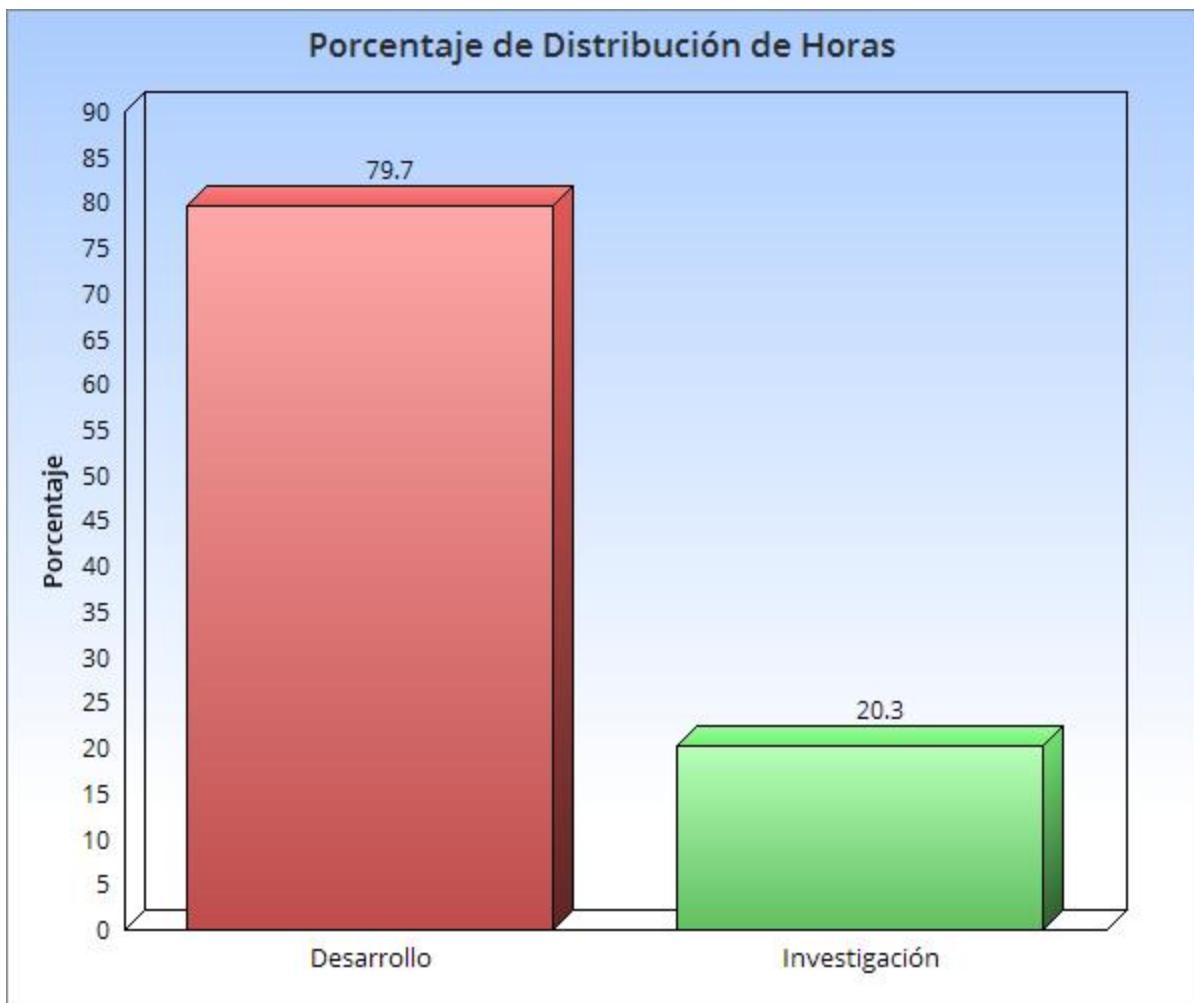
#### 4.2.3.3. Conclusiones del Sprint

En un principio se tenían pensadas más historias de usuario para este sprint, sin embargo por temas de tiempo de las historias y porque hubo que mover algunas del sprint 4 a este, no todas las historias se pudieron completar y hubo que moverlas al sprint 6.

También, como se dijo en el Detalle, surgió un bug para arreglar en el futuro que es el del calendario que no funciona solo en el servidor.

En general las horas utilizadas para cada historia fueron más o menos las horas planeadas para las mismas. Hay algunas, como las relacionadas con el organizador del evento (ver Cronograma Original de Desarrollo, punto 3.3 de la documentación), que todavía no están claras por parte del cliente, por lo tanto fueron pasadas para más adelante.

Y las historias relacionadas con compartir el evento en redes sociales también fueron pasadas al sprint 6 por falta de tiempo.



4.3. Imagen de gráfica de horas del sprint 5

#### 4.2.4. Sprint 6

La siguiente tabla muestra las tareas del sprint 6 junto con la estimación de tiempo y el tiempo real que tomó llevarlas a cabo. También se detalla quién realizó cada tarea.

4. Sprint 6				
4.1.	Crear markup de detalle de evento (sponsors)	7	3	Ximena
4.2.	Investigar API Facebook para realizar y compartir publicaciones	2	1	Marianne
4.3.	Investigar API Twitter para realizar y compartir publicaciones	2	1	Ximena

4.4.	Investigar API LinkedIn para realizar y compartir publicaciones	2	1	Marianne
4.5.	Investigar API Google+ para realizar y compartir publicaciones	2	1	Ximena
4.6.	Investigar opciones para envío de emails	3	4	Marianne
4.7.	Compartir evento en Facebook	4	4	Marianne
4.8.	Compartir evento en Twitter	4	3	Ximena
4.9.	Compartir evento en LinkedIn	4	4	Marianne
4.10.	Compartir evento en Google+	4	1	Ximena
4.11.	Implementar mapa del evento integrado con Google maps	5	7	Ximena
4.12.	Implementar markup login	3	3	Marianne
4.13.	Imprimir detalle de evento en pdf	5	5	Marianne
4.14.	Crear markup para secciones dinámicas	3	4	Ximena
4.15.	Agregar timezone a json	1	1	Ximena
4.16.	Crear markup de registro mediante mockups	6	6	Ximena
4.17.	Compartir evento por email	6	4	Marianne
4.18.	Implementar confirmación de email mediante envío de link	6	6	Marianne
Bugs		6	3	
<b>Subtotal</b>		<b>75</b>	<b>62</b>	

**4.4. Tabla de cronograma sprint 6**

#### 4.2.4.1. Detalle

El Sprint 6 tenía como principal objetivo la integración con redes sociales y el envío de emails tanto de confirmación como para compartir links. También había tareas de maquetación, como terminar una de las secciones del evento, la de sponsors, y comenzar con el markup del formulario de registro a un evento. Asimismo hubo que implementar la creación de un pdf con los datos del evento para que el mismo se pudiera imprimir y la adición de la dirección del evento a un mapa de google maps.

Al no conocer en profundidad las tecnologías fue necesario investigar las APIs para poder compartir con redes sociales, sin embargo la implementación, una vez conocido el procedimiento, fue bastante sencilla lo cual se vio reflejado en la cantidad de horas

que tomó desarrollarlo. Además de compartirlo en redes sociales, también hubo que desarrollar la posibilidad de compartir el evento a través de mail.

En cuanto a mapa que tuvimos que integrar al evento, esta fue una de las tareas que más tiempo tomó del sprint debido a que en un principio teníamos problemas con la scope de angular ya que cuando cargábamos el mapa todavía no teníamos los datos de la latitud y longitud del evento. También nos costó un poco hacer el mapa no dinámico, ya que queríamos que el usuario pudiera manipularlo mismo en la página.

La creación de los markups que había para este sprint se llevó más o menos en la cantidad de tiempo estimado. La única historia que llevó un poco más de tiempo fue la de crear las secciones dinámicas. Ya que en la misma se debían crear directivas adentro de otra directiva principal, y las primeras al principio no cargaban. Sin embargo, se encontró rápidamente una solución para esto. La razón para hacer esto es que el usuario puede crear en salesforce la cantidad y variedad de secciones que quiera. Y estas se deben de mostrar en el evento de acuerdo a cómo este quiera, y en el orden que eligió.

Otra historia relacionada al envío de emails que se llevó a cabo en este sprint, es enviar un mail cuando un usuario se registra al sitio. Al ya tener una idea de cómo implementar esto gracias a la historia de compartir eventos por email, no tomó tanto esfuerzo realizar esta.

#### 4.2.4.2. Casos de Uso

Implementa casos de uso redes, CU-009, CU-010, CU-013 y CU-020. También hay algunas historias que implementaban alguna parte de otros casos de uso pero no la principal, por lo tanto no se mencionan.

Para ver el detalle de cada uno de estos casos de uso dirigirse al anexo A.2.

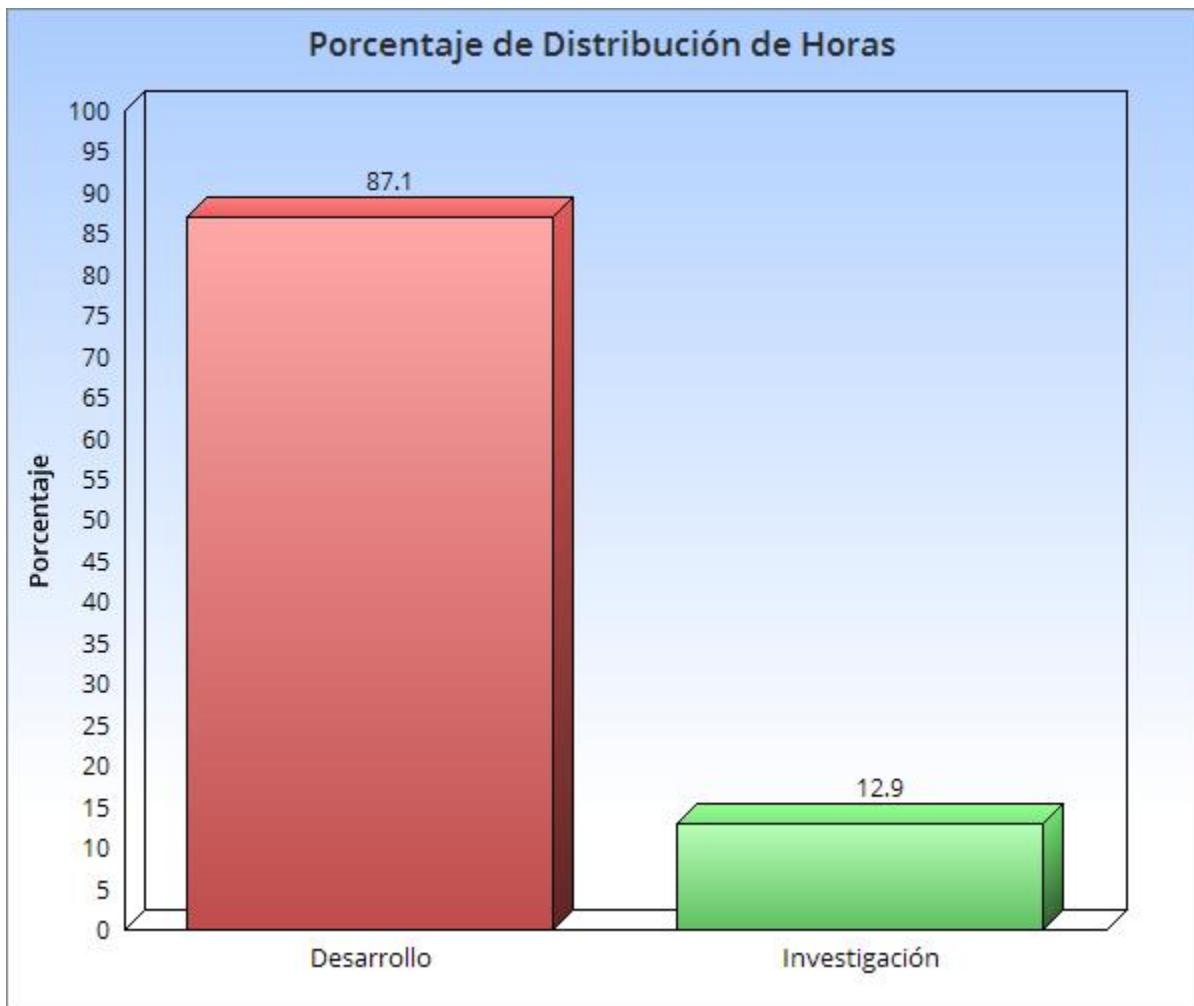
#### 4.2.4.3. Conclusiones del Sprint

Al sprint 6 se incluyeron algunas historias más que no se habían podido realizar al término del sprint 5.

Al analizar la tabla de tareas realizadas, se observa una diferencia entre las horas planificadas y las horas realizadas. Esta diferencia se produce ya que originalmente pensamos que la integración con redes sociales iba a implicar un esfuerzo mayor.

Asimismo, al tener mayor conocimiento de las tecnologías, el esfuerzo necesario para cada tarea es menor ya que no es necesario investigar sobre la sintaxis o el uso de las mismas.

Finalmente, con el sprint 6 se concluye la integración con redes sociales.



4.4. Imagen de gráfica de horas del sprint 6

#### 4.2.5. Sprint 7

La siguiente tabla muestra las tareas del sprint 7 junto con la estimación de tiempo y el tiempo real que tomó llevarlas a cabo. También se detalla quién realizó cada tarea.

5. Sprint 7				
5.1.	Actualizar header para que luzca como el mockup.	1	1	Ximena
5.2.	Implementar guardado de eventos	4	4	Ximena
5.3.	Eliminar evento guardado	4	2	Ximena
5.4.	Implementar directorio de eventos guardados	5	4	Ximena
5.5.	Cambiar markup del login para que se parezca a mockup y sea customizable	5	12	Marianne
5.6.	Implementar directorio de eventos a los que estoy registrado	4	3	Ximena
5.7.	Investigar cómo se generan códigos QR	3	5	Marianne
5.8.	Implementar formulario para modificar datos personales de mi cuenta	4	3	Ximena
5.9.	Implementar envío de email con código	4	8	Marianne
5.10.	Implementar directorio de eventos pasados	4	4	Ximena
5.11.	Crear formulario (pop up) de contacto con organizador	4	5	Marianne
5.12.	Validaciones del formulario de contacto con organizador	4	4	Marianne
5.13.	Modificar contraseña	3	2	Ximena
5.14.	Actualizar versión de paquete de angular y arreglar problemas que genera con templates	3	4	Ximena
5.15.	Cambiar botón de página de evento para que diga Get Tickets	1	1	Ximena
5.16.	Cambiar estructura de json para tickets	1	1	Ximena
Bugs		6	11	
<b>Subtotal</b>		<b>60</b>	<b>74</b>	

#### 4.5. Tabla de cronograma sprint 7

##### 4.2.5.1. Detalle

En el sprint 7 se pensaba implementar la parte de registro y pago a un evento, pero por un tema de que el cliente no tenía aún claros todos los requerimientos involucrados con el registro y pago se decidió mover dicha implementación para el sprint 8.

Es por ello que durante el sprint número 7 se terminaron algunas historias que habían quedado incompletas de los sprints pasados o que no se habían podido comenzar, especialmente las relacionadas con el detalle del evento. Dentro de las mismas tenemos la 5.1. que consistía en actualizar el header porque cuando lo hicimos en un principio el cliente todavía no tenía listos los mockups y nos dijeron más o menos cómo lo querían, entonces en esta historia lo adaptamos mejor a como quedó diseñado. La 5.2. y la 5.3. también están relacionadas al detalle de evento, ya que desde el mismo se puede guardar o eliminar un evento de los guardados (aunque esto último también se puede realizar desde la cuenta del usuario).

Asimismo se comenzó con la sección de My Account, en donde el usuario puede consultar sus datos personales, los eventos que tiene guardados, los que ya pasaron y aquellos a los que está registrado. Las historias dentro de esta sección son las siguientes: 5.3., 5.4., 5.6., 5.8., 5.10., 5.13.

Una de las historias que se agregó y llevó más horas de las estimadas fue la modificación del login (5.5.). Ya que hasta el momento se había utilizado el paquete Accounts de Meteor que crea el markup del login automáticamente, pero el mismo no es customizable. Por lo tanto, hubo que modificar el markup del login para que fuera similar al del mockup y se presentara como una ventana modal y no como un dropdown.

Otras de las historias desarrolladas para este sprint fueron las relacionadas con el envío del código QR luego de que el usuario se registra al evento (5.7. y 5.9.). Se implementó la funcionalidad más allá de que todavía el registro y pago de un evento no estaba listo, por lo tanto luego de que esto estuviera desarrollado, ambas funcionalidades se unirían.

También se trabajó en el formulario de contacto con el organizador (5.11. y 5.12), hubo que actualizar la versión del paquete de angular para meteor que estábamos usando ya que a lo largo del desarrollo del proyecto se publicaron nuevas versiones (5.14.), se cambió el botón para registrarse un evento para que diga “Get Tickets” (5.15.), se cambió la estructura de la colección events específicamente en la parte relacionada a los tickets (5.16.).

Asimismo, se utilizaron varias horas para arreglar bugs. Especialmente uno en el que cuando el usuario hacía logout y estaba en la página My Account, no lo redireccionaba al directorio de eventos o quitaba el link para acceder a My Account. Y otro que sucedía cuando se agregó el botón para contactar al organizador del evento y que hacía que el botón de agregar al calendario que estaba al lado moviera su texto para abajo porque no había espacio para el elemento.

#### 4.2.5.2. Casos de Uso

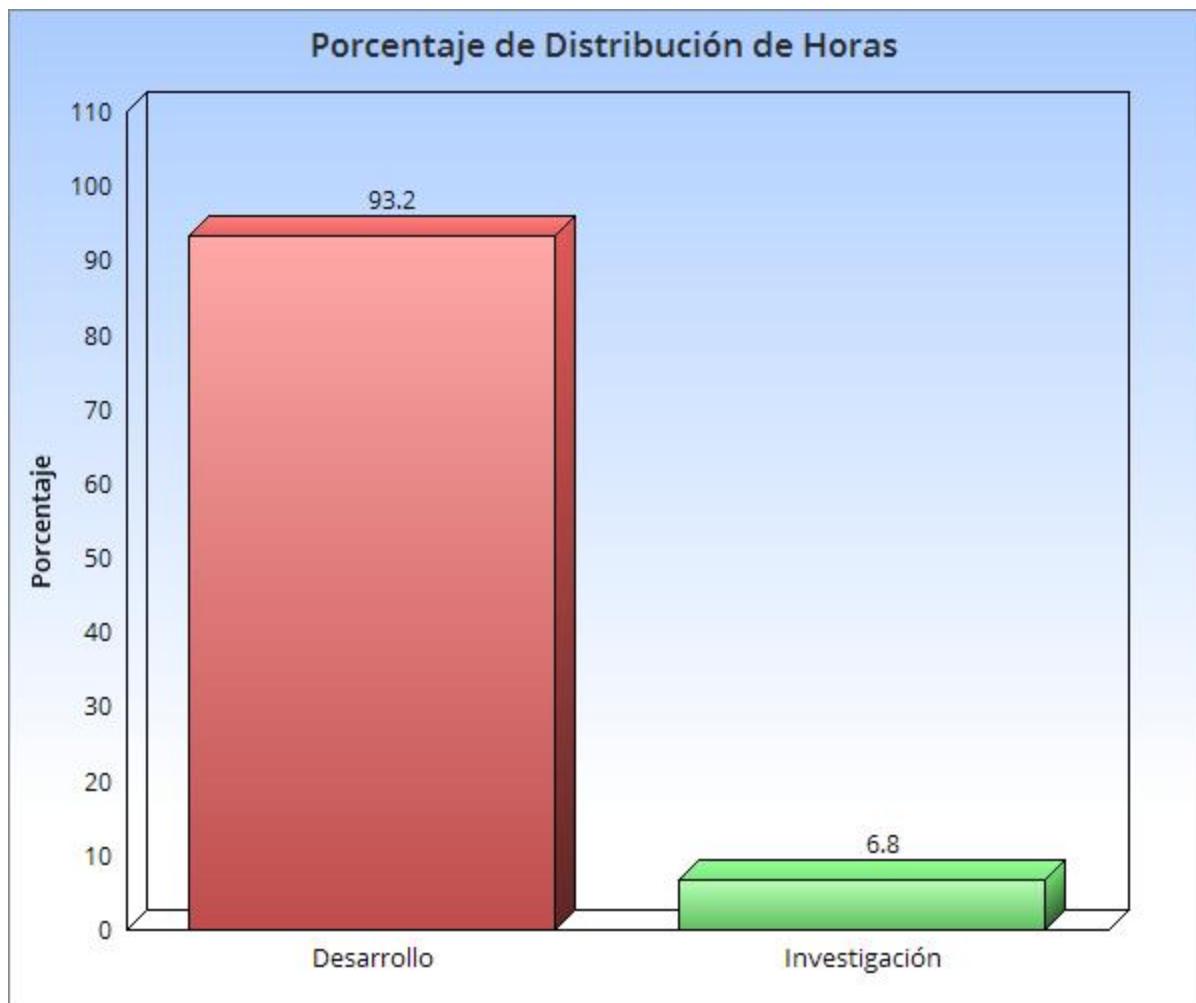
Implementa casos de uso CU-003, CU-004, CU-015, CU-016, CU-017, CU-018 y CU-019. También hay algunas historias que implementaban alguna parte de otros casos de uso pero no la principal, por lo tanto no se mencionan.

Para ver el detalle de cada uno de estos casos de uso dirigirse al anexo A.2.

#### 4.2.5.3. Conclusiones del Sprint

El sprint número 7 fue un sprint bastante productivo en el que se pudieron completar variadas historias. También hubo más bugs para arreglar que en los sprints anteriores, especialmente porque se fueron encontrando algunos de las cosas que ya estaban hechas anteriormente. Quizá, por esta razón fue que las horas reales fueron mayores a las estimadas, además de que hubo algunas historias que tomaron considerablemente más tiempo de lo pensado ya que presentaban algunos obstáculos que tuvimos que sortear.

Asimismo, quedaron la mayoría de las tareas solucionadas como para que en el siguiente sprint se pueda comenzar y enfocarse principalmente en las tareas relacionadas al pago del evento.



4.5. Imagen de gráfica de horas del sprint 7

#### 4.2.6. Sprint 8

La siguiente tabla muestra las tareas del sprint 8 junto con la estimación de tiempo y el tiempo real que tomó llevarlas a cabo. También se detalla quién realizó cada tarea.

6. Sprint 8				
6.1.	Agregar foto de perfil a My Account	4	8	Ximena
6.2.	Hacer que se vea rango de precios de un evento en directorio de eventos y detalle de evento	3	3	Ximena
6.3.	Implementar pago online: Paso 1, modal en detalle de evento. Crear markup.	4	5	Ximena

6.4.	Implementar pago online: Paso 1, modal en detalle de evento. Lógica	3	4	Ximena
6.5.	Implementar pago online: Paso 2, resumen de orden y datos personales	5	5	Marianne
6.6.	Agregar columna derecha en registro para datos del evento	2	3	Ximena
6.7.	Modificar REST API (restivus) para recibir datos para que modifique fechas a date de JS	3	4	Ximena
6.8.	Investigar la documentación de Stripe para adquirir conocimiento sobre la integración	8	8	Marianne
6.9.	Implementar pago online: Paso 3, datos de pago: Stripe	6	6	Marianne
6.10.	Implementar pago online: Paso 3, datos de pago: Comunicación con la API	8	9	Ximena
6.11.	Implementar timeout para completar formulario de inscripción	3	4	Marianne
6.12.	Validaciones del formulario de registro a evento	6	5	Marianne
6.13.	Implementar conjunto de mensajes a mostrar al usuario dependiendo del resultado con respecto al pago de eventos	2	2	Marianne
6.14.	Agregar búsqueda de evento que sea accesible en todo el sitio	2	2	Ximena
6.15.	Implementar pago con código de descuento	5	5	Ximena
6.16.	Implementar API para mandar descuentos a salesforce	2	2	Ximena
6.17.	Imprimir info de pago	5	6	Ximena
6.18.	Implementar mensajes antes y después de envío de mail de confirmación de registro al sitio	3	3	Marianne
6.19.	Modificar llamada a API cuando mandamos la orden y agregar datos de Stripe	1	1	Ximena
6.20.	Agregar evento a eventos registrados en el json luego de compra	2	2	Ximena
6.21.	Agregar página de agradecimiento por la compra luego del registro	3	4	Marianne
6.22.	Mandar mail con info de pago	4	12	Marianne

6.23.	Imprimir pdf en página de agradecimiento	4	5	Marianne
6.24.	Mejorar validaciones de teléfonos	1	1	Ximena
6.25.	Arreglar burbujas por filtros y sorts	4	4	Marianne
6.26.	Eliminar el registro de un evento	5	7	Ambas
6.27.	Modificar columna de event detail y agregar banner image	1	1	Ximena
6.28.	Testeo de la aplicación	5	3	Ambas
6.29.	Deploy final a Heroku	1	1	Ambas
6.30.	Confeccionar manual de usuario	4	5	Ambas
6.31.	Crear presentación para defensa	6	6	Ambas
6.32.	Finalizar documentación	15	16	Ambas
Bugs		12	13	
<b>Subtotal</b>		<b>142</b>	<b>165</b>	

#### 4.6. Tabla de cronograma sprint 8

##### 4.2.6.1. Detalle

Las historias de usuario más relevantes que se llevaron a cabo en el Sprint 8 fueron las relacionadas con el pago de los tickets en un evento y la comunicación mediante la API de Salesforce para mandar los datos de la orden de compra luego de que se compraron los tickets. La razón por la que esta funcionalidad fue dejada para último momento fue porque el cliente todavía estaba desarrollando los requisitos para la misma y la intención era que esta funcionalidad fuera lo más extensible posible a futuras modificaciones o adiciones.

Dentro de las que estaban dedicadas a implementar el pago tenemos las historias 6.3., 6.4., 6.5., 6.6., 6.8., 6.9., 6.11., 6.12., 6.13., 6.15., 6.17., 6.20., 6.21., 6.22. y 6.23. Estas engloban todo desde el markup que se agregó a la página para seleccionar tickets (el modal en el detalle de evento que se despliega al intentar comprar), validaciones, la lógica para manejar los datos durante la compra, el manejo del código de descuento, la comunicación con la API de Stripe para llevar a cabo la compra, el envío de los datos de la compra al email del comprador y la página final de compra en donde se comunica la compra realizada y se muestran botones para ver la factura o ir al directorio.

En cuanto a la comunicación con Salesforce mediante la API destinada a recibir órdenes y consultas de descuentos las historias son: 6.10., 6.16. y 6.19. Estas implicaron implementar una clase con REST Apex en el org de la aplicación en

Salesforce, desarrollar utilizando JSForce (una librería para trabajar con Salesforce utilizando javascript que utilizamos para autenticarnos en Salesforce) y manejar los datos de la orden que se mandan a Salesforce para guardar.

Otra historia relacionada con el registro al evento fue la 6.26., en la que se implementó la eliminación de un registro a un evento. Esta historia no estábamos seguras si íbamos a poder implementarla, ya que no sabíamos si nos iba a dar el tiempo, pero finalmente nos dio. Incluye mandar un *request* a Stripe para hacer void la factura y mandar otro *request* a Salesforce para cancelar la *Order* y devolver los asientos cancelados.

También hubo historias más aisladas, como la 6.1. en la que hubo que implementar el guardado de una foto de perfil en la cuenta del usuario registrado. La 6.2., en la que se decidió mostrar el rango de precios de los distintos tipos de tickets para un evento, en vez de mostrar el precio más bajo entre todos ellos, como se había hecho hasta el momento. La 6.7., en la que hubo que modificar la REST API (Restivus) que se implementó en Meteor y hacerla a mano para que cuando se envía una fecha esta sea guardada de la forma correcta (como fecha de Javascript) como para que después se pueda manipular fácilmente en Meteor. La 6.14., en la que se cambió la caja de texto en la que se buscan los eventos y se ubicó en el header, para que fuera accesible desde cualquier parte del sitio. La 6.18., en la que hubo que implementar mensajes más detallados para cuando un usuario se registraba en el sitio, antes de mandar el mail de verificación y luego de la misma. En la 6.25. hubo que arreglar las burbujas de los sorts y filtros, para que en vez del tipo de filtro mostraran el valor del filtro, o sea, en vez de *Start Date* mostraran 6/2/16 por ejemplo. También se hizo que las burbujas estuvieran más cerca entre sí. En cuanto a la 6.27. se cambió la columna derecha en el detalle del evento, para que estuviera separada por secciones, como la del registro al evento y también se agregó un panel arriba para mostrar la *banner image*.

Las últimas historias de usuario son los últimos pasos hacia la entrega del proyecto. Implican un testeo general de la aplicación utilizando los casos de prueba escritos, resolución de últimos bugs que puedan haber, y último deployment de la aplicación. También contienen tareas de documentación y preparación de la presentación para la defensa así como también se escribió el manual de usuario para que se pueda utilizar la aplicación.

Las horas dedicadas a la resolución de bugs en este sprint fueron bastantes, unas 13 horas. Los bugs que se identificaron fueron uno en el ordenamiento por precio que fue debido al cambio en la colección *events*, más específicamente en los campos que referían a los tickets y sus precios. Otro problema que encontramos era que luego de que cerrábamos el modal del login la barra para hacer scroll de la pantalla desaparecía y no volvía, a pesar de que la ventana seguía teniendo contenido más abajo. Esto fue

un problema que surgió debido a que se agregaba una clase al modal que no se removía luego de que se cerraba este. También sucedía con el modal del login que los botones del acceso mediante redes sociales cambiaban de estilo en distintos momentos cuando se abría el modal. Esto era un tema de un timer que había y se solucionó simplemente con css. Otro bug era que al mostrar la fecha de los eventos, la misma se mostraba un día antes del que tenía en la colección en mongoDB ya que había un problema con las zonas horarias. También faltaban validaciones en la página del registro para cuando determinados valores eran *undefined* y no se creaba un objeto ticket por cada ticket comprado en el objeto que se manda a salesforce para crear la orden. Por último había un bug en el que si el *promo code* era mayor al subtotal de la compra cuando se restaba el total era negativo.

#### 4.2.6.2. Casos de Uso

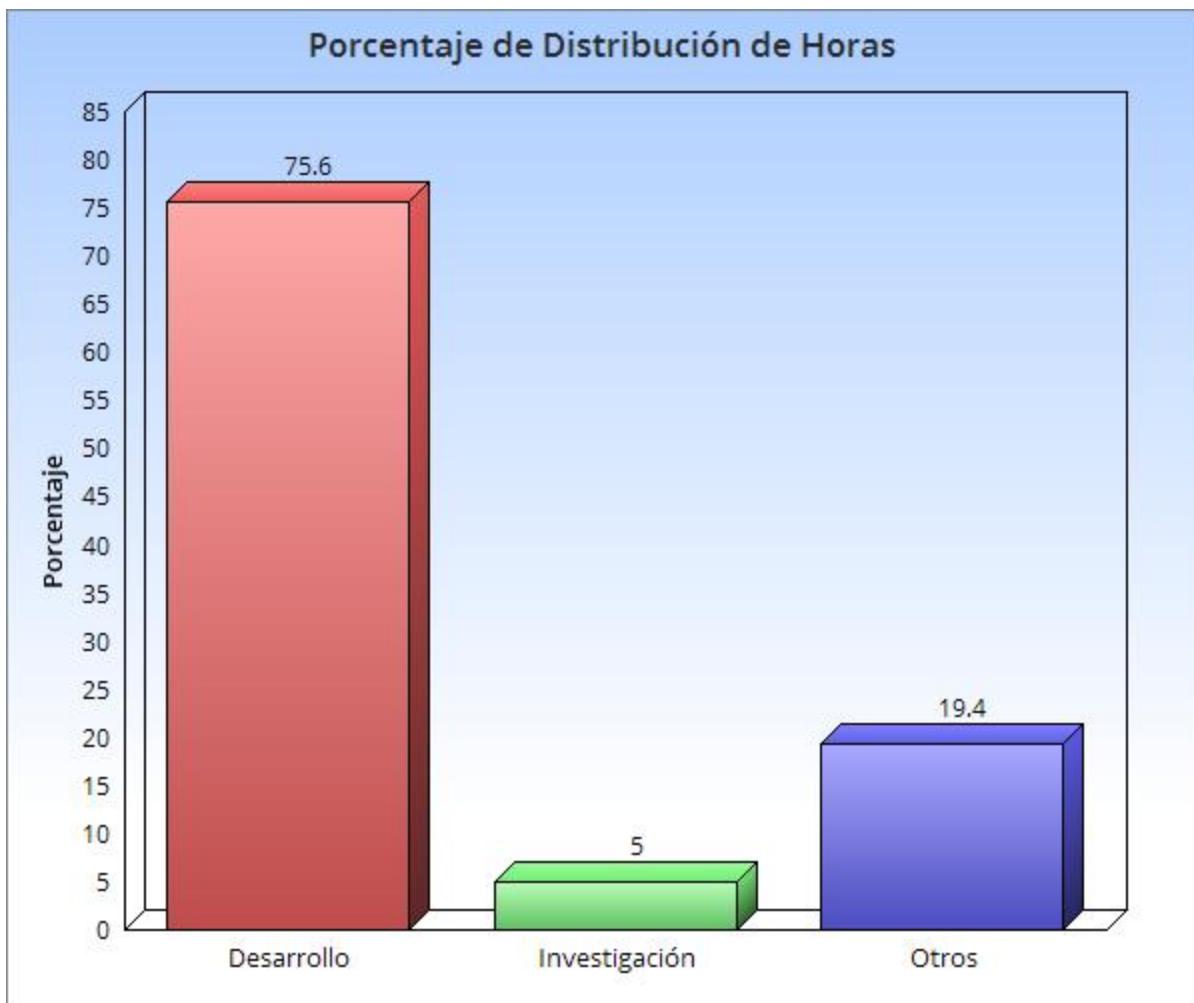
Implementa casos de uso CU-001, CU-002, CU-008, CU-014 y CU-021. También hay algunas historias que implementaban alguna parte de otros casos de uso pero no la principal, por lo tanto no se mencionan.

Para ver el detalle de cada uno de estos casos de uso dirigirse al anexo A.2.

#### 4.2.6.3. Conclusiones del Sprint

Este sprint fue el que nos insumió más horas debido a que las historias de usuario en él eran bastante importantes para la finalización del proyecto. Además también incluía las tareas que había que llevar a cabo para la presentación del mismo, como terminar con la documentación, preparar la defensa, etc.

Sin embargo, el número de horas dedicadas a cada historia del sprint se aproxima bastante a las que estimamos, probablemente porque ya a esta altura teníamos más conocimiento de las herramientas y podíamos tener una idea más clara de los obstáculos que se nos podían presentar y de cómo llevar a cabo determinada tarea.



**4.6. Imagen de gráfica de horas del sprint 8**

### 4.3. Casos de Prueba

#### 4.3.1. Log in

##### 4.3.1.1. Log in Exitoso

<b>ID</b>	1.1
<b>Título</b>	<b>Log in Exitoso</b>
<b>Prioridad</b>	Alta
<b>Propósito</b>	El usuario puede loguearse con su nombre de usuario y contraseña de manera exitosa.

<b>Precondiciones</b>	El usuario está registrado al sistema.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos
<b>Paso 2</b>	El usuario hace clic en el botón de <i>Login or Join</i> .	Se abre un modal para loguearse o registrarse.
<b>Paso 3</b>	El usuario ingresa su email y contraseña existentes.	Se cierra el modal y aparece su email en lugar del botón de login. Se ha logueado exitosamente.

4.7. Tabla de caso de prueba 1.1

#### 4.3.1.2. Log in con Contraseña Inválida

<b>ID</b>	1.2	
<b>Título</b>	<b>Log in con Contraseña Inválida</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que al ingresar una password incorrecta el usuario no pueda ingresar	
<b>Precondiciones</b>	El usuario está registrado al sistema.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos
<b>Paso 2</b>	El usuario hace clic en el botón de <i>Login or Join</i> .	Se abre un modal para loguearse o registrarse.
<b>Paso 3</b>	El usuario ingresa su email (que existe en el sistema), pero ingresa su contraseña con errores.	Aparece un mensaje en el modal que dice <i>Incorrect Password</i> .
<b>Paso 4</b>	El usuario ingresa su email y contraseña existentes.	Se cierra el modal y aparece su email en lugar del botón de login. Se ha logueado exitosamente.

4.8. Tabla de caso de prueba 1.2

#### 4.3.1.3. Log in con Email Inválido

<b>ID</b>	1.3	
<b>Título</b>	<b>Log in con Email Inválido</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que al ingresar un email incorrecto o inexistente el usuario no se pueda loguear.	
<b>Precondiciones</b>	El usuario está registrado en el sistema.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos
<b>Paso 2</b>	El usuario hace clic en el botón de <i>Login or Join</i> .	Se abre un modal para loguearse o registrarse.
<b>Paso 3</b>	El usuario ingresa su email con errores, e ingresa su contraseña correctamente.	Aparece un mensaje en el modal que dice <i>User not found</i> .
<b>Paso 4</b>	El usuario ingresa su email y contraseña existentes.	Se cierra el modal y aparece su email en lugar del botón de login. Se ha logueado exitosamente.

#### 4.9. Tabla de caso de prueba 1.3

#### 4.3.1.4. Registro Exitoso

<b>ID</b>	1.4	
<b>Título</b>	<b>Registro Exitoso</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que un usuario se pueda registrar al sitio.	
<b>Precondiciones</b>		
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos
<b>Paso 2</b>	El usuario hace clic en el botón	Se abre un modal para loguearse o

	de <i>Login or Join</i> .	registrarse.
<b>Paso 3</b>	El usuario hace clic en <i>Create Account</i> .	Desaparece el botón de <i>Create Account</i> , aparece arriba y quedan las cajas de texto para que ingrese email y contraseña.
<b>Paso 4</b>	El usuario ingresa su email y contraseña y hace clic en <i>Create Account</i> .	Se loguea al usuario y se le envía un email para confirmar su registro.
<b>Paso 5</b>	El usuario abre el email y cliquea el link dentro del mismo.	El usuario es redirigido a la aplicación y su registro verificado. Se muestra un modal que dice: <i>Email verified. You are now logged in as [email del usuario]</i> .

4.10. Tabla de caso de uso 1.4

#### 4.3.1.5. Registro con Contraseña Inválida

<b>ID</b>	1.5	
<b>Título</b>	<b>Registro con Contraseña Inválida</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que al ingresar una contraseña con menos de 6 caracteres el usuario no se pueda registrar.	
<b>Precondiciones</b>		
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos
<b>Paso 2</b>	El usuario hace clic en el botón de <i>Login or Join</i> .	Se abre un modal para loguearse o registrarse.
<b>Paso 3</b>	El usuario hace clic en <i>Create Account</i> .	Desaparece el botón de <i>Create Account</i> , aparece arriba y quedan las cajas de texto para que ingrese email y contraseña.
<b>Paso 4</b>	El usuario ingresa su email y	Se mantiene el modal y aparece un

	una contraseña con menos de 6 caracteres.	mensaje que dice: <i>Password must be at least 6 characters long.</i>
<b>Paso 5</b>	El usuario ingresa su email y una contraseña con al menos 6 caracteres.	El flujo continúa de acuerdo al paso 4 del caso de prueba 1.4.

#### 4.11. Tabla de caso de prueba 1.5

##### 4.3.1.6. Log in con Redes Sociales

ID	1.6	
Título	<b>Log In con Redes Sociales</b>	
Prioridad	Alta	
<b>Propósito</b>	Loguearse al sistema mediante alguna de las redes sociales disponibles.	
<b>Precondiciones</b>	El usuario tiene su cuenta de facebook abierta.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos
<b>Paso 2</b>	El usuario hace clic en el botón de <i>Login or Join</i> .	Se abre un modal para loguearse o registrarse.
<b>Paso 3</b>	El usuario hace clic en el botón del logo de facebook.	El sistema le pide autorización para proveer sus datos de facebook.
<b>Paso 4</b>	El usuario acepta.	El modal se cierra y el usuario se loguea. Su email asociado a google aparece en lugar del botón de login.

#### 4.12. Tabla de caso de prueba 1.6

##### 4.3.2. Vista en Grilla y Lista

###### 4.3.2.1. Cambio de Vista de Eventos de Lista a Grilla

<b>ID</b>	2.1
-----------	-----

<b>Título</b>	<b>Cambio de Vista en Eventos de Lista a Grilla</b>	
<b>Prioridad</b>	Media	
<b>Propósito</b>	Verificar que se muestren de manera correcta los eventos en el directorio de eventos	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla. Es como una tabla con celdas en donde está cada evento.
<b>Paso 2</b>	El usuario hace clic en el ícono de la lista que está arriba a la derecha de los eventos.	Este ícono se muestra como seleccionado y los eventos se muestran como una lista, uno abajo del otro.
<b>Paso 3</b>	El usuario hace clic en el ícono de la grilla.	Este ícono vuelve a estar como seleccionado y los eventos se muestran como grilla nuevamente.

4.13. Tabla de caso de prueba 2.1

### 4.3.3. Búsqueda de Eventos

#### 4.3.3.1. Búsqueda por Nombre de Evento Existente

<b>ID</b>	3.1
<b>Título</b>	<b>Búsqueda por Nombre de Evento Existente</b>
<b>Prioridad</b>	Media
<b>Propósito</b>	Verificar el correcto funcionamiento de la búsqueda por nombre de evento.
<b>Precondiciones</b>	Tener eventos guardados en la base

	de datos.	
	Descripción	Resultado Esperado
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario escribe el título de un evento existente.	Se le van sugiriendo títulos.
<b>Paso 3</b>	El usuario hace clic o da enter en uno.	Se lo lleva a los detalles de ese evento.

4.14. Tabla de caso de prueba 3.1

#### 4.3.3.2. Búsqueda por Nombre de Evento Inexistente

<b>ID</b>	3.2	
Título	Búsqueda por Nombre de Evento Inexistente	
Prioridad	Media	
<b>Propósito</b>	Verificar el correcto funcionamiento de la búsqueda por nombre de evento que no existe.	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	Descripción	Resultado Esperado
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario escribe el título de un evento inexistente.	Se le van sugiriendo títulos hasta que ya no queden más sugerencias.
<b>Paso 3</b>	No se cambia de página, se mantiene la página de directorio de eventos.	

4.15.. Tabla de caso de prueba 3.2

#### 4.3.3.3. Búsqueda por Ubicación Existente

<b>ID</b>	3.3	
<b>Título</b>	<b>Búsqueda por Ubicación Existente</b>	
<b>Prioridad</b>	Media	
<b>Propósito</b>	Verificar el correcto funcionamiento de la búsqueda por ubicación existente.	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario escribe una ubicación existente.	Se le van sugiriendo ubicaciones.
<b>Paso 3</b>	El usuario hace clic en la ubicación deseada.	Se muestran los eventos con esa ubicación.

4.16. Tabla de caso de prueba 3.3

#### 4.3.3.4. Búsqueda por Ubicación Inexistente

<b>ID</b>	3.4	
<b>Título</b>	<b>Búsqueda por Ubicación Inexistente</b>	
<b>Prioridad</b>	Media	
<b>Propósito</b>	Verificar el correcto funcionamiento de la búsqueda por ubicación inexistente.	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>

<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario escribe una ubicación inexistente.	Se le van sugiriendo ubicaciones hasta que no haya más sugerencias.
<b>Paso 3</b>	El usuario no ve cambios en los eventos mostrados.	

4.17 Tabla de caso de prueba 3.4

#### 4.3.4. Ordenamiento de Eventos

##### 4.3.4.1. Ordenamiento por Nombre

ID	4.1	
Título	Ordenamiento por Nombre	
Prioridad	Alta	
Propósito	Verificar el correcto funcionamiento del ordenamiento por nombre.	
Precondiciones	Tener eventos guardados en la base de datos.	
Descripción	Resultado Esperado	
Paso 1	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
Paso 2	El usuario hace clic en <i>Sort By</i> .	Se despliega un dropdown mostrando <i>Name (A - Z)</i> , <i>Name (Z - A)</i> , <i>Price (Low to High)</i> , <i>Price (High to Low)</i> .
Paso 3	El usuario hace clic en <i>Name (A - Z)</i> .	Se muestran los eventos ordenados de manera alfabética.
Paso 4	El usuario hace clic en <i>Name (Z - A)</i> .	Se muestran los eventos ordenados desde la Z a la A.

#### 4.18. Tabla de caso de prueba 4.1

##### 4.3.4.2. Ordenamiento por Precio

<b>ID</b>	<b>4.2</b>	
<b>Título</b>	<b>Ordenamiento por Precio</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar el correcto funcionamiento del ordenamiento por precio.	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario hace clic en <i>Sort By</i> .	Se despliega un dropdown mostrando <i>Name (A - Z)</i> , <i>Name (Z - A)</i> , <i>Price (Low to High)</i> , <i>Price (High to Low)</i> .
<b>Paso 3</b>	El usuario hace clic en <i>Price (Low to High)</i> .	Se muestran los eventos ordenados de más baratos a más caros.
<b>Paso 4</b>	El usuario hace clic en <i>Price (High to Low)</i> .	Se muestran los eventos ordenados desde más caros a más baratos.

#### 4.19. Tabla de caso de prueba 4.2

##### 4.3.5. Filtrado de Eventos

###### 4.3.5.1. Filtrado por Tiempo

<b>ID</b>	<b>5.1</b>	
<b>Título</b>	<b>Filtrado por Tiempo</b>	
<b>Prioridad</b>	Media	
<b>Propósito</b>	Verificar que el filtrado por tiempo funciona	

	<b>adecuadamente</b>	
<b>Precondiciones</b>	Tener al menos dos eventos guardados en la base de datos. Uno de ellos debe realizarse esta semana y el otro la semana que viene.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestran todos los eventos en la base de datos.
<b>Paso 2</b>	El usuario hace clic en <i>All Time</i> y luego en <i>This Week</i> .	Se muestran solo el evento que es esta semana.
<b>Paso 3</b>	El usuario hace clic en <i>All Time</i> y luego en <i>This Weekend</i> .	Se muestra solo el evento que es esta semana.
<b>Paso 4</b>	El usuario hace clic en <i>All Time</i> y luego en <i>Next Week</i> .	Se muestra solo el evento que es la semana que viene.
<b>Paso 5</b>	El usuario hace clic en <i>All Time</i> y luego en <i>This Month</i> .	Se muestran solo los eventos que son en este mes.
<b>Paso 6</b>	El usuario hace clic en <i>All Time</i> y luego en <i>Custom Date</i> .	Arriba de los eventos se muestran dos inputs para ingresar la fecha de inicio y la fecha de finalización.
<b>Paso 7</b>	El usuario escoge el día anterior de uno de los eventos como fecha de inicio y el día posterior del mismo evento como fecha de finalización.	Se muestra solo el evento que se tomó como referencia.

4.20. Tabla de caso de prueba 5.1

#### 4.3.5.2. Filtrado por Precio

<b>ID</b>	<b>5.2</b>	
<b>Título</b>	<b>Filtrado por Precio</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el filtrado por precio funciona correctamente.	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>

<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario hace clic en <i>All Prices</i> .	Se despliega un dropdown con las opciones <i>All, Free, Paid</i> .
<b>Paso 3</b>	El usuario selecciona <i>Free</i> .	Se muestran solo los eventos que son gratis y una etiqueta que muestra la palabra <i>Price: Free</i> .
<b>Paso 4</b>	El usuario hace clic en <i>All Prices</i> y luego en <i>Paid</i> .	Se muestran solo los eventos pagos y una etiqueta que muestra la palabra <i>Price: Paid</i> .
<b>Paso 5</b>	El usuario hace clic en <i>All Prices</i> y luego en <i>All</i> .	Se muestran todos los eventos.

4.21. Tabla de caso de prueba 5.2

#### 4.3.5.3. Filtrado por Tipo de Evento

<b>ID</b>	5.3	
<b>Título</b>	<b>Filtrado por Tipo de Evento</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el filtrado por tipo de evento funciona correctamente.	
<b>Precondiciones</b>	Tener eventos guardados en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario hace clic en <i>All Types</i> .	Se despliega un dropdown con los tipos de eventos disponibles entre todos los eventos en la base de datos.
<b>Paso 3</b>	El usuario hace clic en alguna de las opciones disponibles.	Se muestran los eventos que pertenecen a ese tipo y se muestra una etiqueta con el tipo de evento seleccionado.

<b>Paso 4</b>	El usuario hace clic en la cruz dentro de la etiqueta del tipo de evento seleccionado.	Se vuelven a mostrar todos los eventos en la base de datos.
---------------	--	---

4.22. Tabla de caso de prueba 5.3

#### 4.3.6. Barra de Navegación en Detalle de Evento

##### 4.3.6.1. Correcto Funcionamiento de la Barra de Navegación y las Secciones

ID	6.1	
<b>Título</b>	<b>Correcto funcionamiento de la Barra de Navegación y las Secciones</b>	
<b>Prioridad</b>	Media	
<b>Propósito</b>	Verificar el correcto funcionamiento de la barra de navegación en el detalle del evento.	
<b>Precondiciones</b>	Tener al menos un evento guardado en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario selecciona un evento dentro del directorio.	Se muestran los detalles del evento. En la parte superior del evento se muestra una barra con links a todas las secciones del mismo.
<b>Paso 3</b>	El usuario selecciona Overview.	La vista se traslada a la sección Overview.
<b>Paso 4</b>	El usuario sube hacia la parte superior del evento y selecciona otra sección.	La vista se traslada a la sección seleccionada por el usuario.

4.23. Tabla de caso de prueba 6.1

## 4.3.7. Guardar Evento en Detalle del Evento

### 4.3.7.1. Guardar Evento desde Detalle del Evento

<b>ID</b>	<b>7.1</b>	
<b>Título</b>	<b>Guardar Evento desde Detalle del Evento</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el guardado del evento en el detalle del evento se produzca de forma correcta.	
<b>Precondiciones</b>	*Debe al menos haber un evento en la base de datos. *El usuario debe estar registrado en el sistema.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario selecciona un evento.	Se muestra la página del detalle del evento.
<b>Paso 3</b>	El usuario hace clic en <i>Save Event</i> .	Se muestra un modal diciéndole que debe loguearse para guardar un evento. Presenta el siguiente texto: <i>Please log in to save the event.</i>
<b>Paso 4</b>	El usuario hace clic en <i>Accept</i> .	El modal se cierra.
<b>Paso 5</b>	El usuario se loguea y hace clic nuevamente en <i>Save Event</i> .	Aparece un modal que dice que se guardó el evento con el siguiente texto: <i>The event has been saved!</i> En donde estaba el link de <i>Save Event</i> ahora se puede ver <i>Remove Saved Event</i> .
<b>Paso 6</b>	El usuario hace clic en <i>My Account</i> y luego en <i>Saved Events</i> .	Allí se muestra el evento recién guardado.

4.24. Tabla de caso de prueba 7.1

### 4.3.7.2. Eliminar Evento de Eventos Guardados desde Detalle del Evento

<b>ID</b>	7.2	
<b>Título</b>	<b>Eliminar Evento de Eventos Guardados desde Detalle del Evento</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que funcione correctamente la funcionalidad de eliminar un evento guardado desde el detalle del evento.	
<b>Precondiciones</b>	*Tener al menos un evento guardado en la base de datos. *Tener un usuario registrado. *Tener un evento guardado en <i>Saved Events</i> para ese usuario.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea y cliquea sobre <i>My Account</i> y luego sobre <i>Saved Events</i> .	Se muestra el evento que tiene guardado.
<b>Paso 3</b>	El usuario selecciona el evento guardado.	Se muestra el detalle del evento seleccionado.
<b>Paso 4</b>	El usuario hace clic sobre <i>Remove Saved Event</i> .	Se muestra un modal diciéndole que se eliminó el evento de sus eventos guardados. El texto que se muestra es el siguiente: <i>The event has been removed from your saved events!</i>
<b>Paso 5</b>	El usuario hace clic en <i>Accept</i> .	El link que decía <i>Remove Saved Events</i> ahora dice <i>Save Event</i> .
<b>Paso 6</b>	El usuario cliquea sobre <i>My Account</i> y luego sobre <i>Saved Events</i> .	Ya no tiene más eventos guardados. Se muestra un texto que dice <i>You have no saved events!</i>

4.25. Tabla de caso de prueba 7.2

## 4.3.8. Datos Personales en My Account

### 4.3.8.1. Cambio Exitoso de Contraseña

<b>ID</b>		<b>8.1</b>
<b>Título</b>		<b>Cambio Exitoso de Contraseña</b>
<b>Prioridad</b>		Alta
<b>Propósito</b>	Verificar que el sistema permite al usuario cambiar de contraseña.	
<b>Precondiciones</b>	*Debe haber un usuario registrado.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea y se dirige a <i>My Account</i> .	Se muestra la sección de configuración de datos personales ( <i>Settings</i> ).
<b>Paso 3</b>	Dentro de <i>Settings</i> , el usuario cliquea en <i>Password</i> .	Se muestra una pantalla para cambiar la contraseña.
<b>Paso 4</b>	El usuario ingresa su contraseña anterior y una nueva de más de seis caracteres. Hace clic en <i>Save</i> .	Debajo de las cajas de texto le aparece un mensaje diciendo que la contraseña se cambió con éxito. Este dice: <i>Password changed successfully</i> .
<b>Paso 5</b>	El usuario se desloguea y hace clic para loguearse nuevamente.	Se muestra el modal de login.
<b>Paso 6</b>	El usuario ingresa su email y su contraseña nueva.	Se loguea exitosamente al sitio.

4.26. Tabla de caso de prueba 8.1

### 4.3.8.2. Cambio de Contraseña Inválida

<b>ID</b>		<b>8.2</b>
-----------	--	------------

<b>Título</b>	<b>Cambio de Contraseña Inválida</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que cuando el usuario quiere cambiar la contraseña por una inválida el sistema no lo permita.	
<b>Precondiciones</b>	*Debe haber un usuario registrado.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea y se dirige a <i>My Account</i> .	Se muestra la sección de configuración de datos personales ( <i>Settings</i> ).
<b>Paso 3</b>	Dentro de <i>Settings</i> , el usuario cliquea en <i>Password</i> .	Se muestra una pantalla para cambiar la contraseña.
<b>Paso 4</b>	El usuario ingresa su contraseña anterior y una nueva con menos de seis caracteres. Hace clic en <i>Save</i> .	Se muestra un mensaje diciendo que la contraseña debe tener al menos seis caracteres. El texto dice: <i>Password must be at least 6 characters long</i> .
<b>Paso 5</b>	El usuario ingresa nuevamente su contraseña anterior y una nueva con al menos seis caracteres. Hace clic en <i>Save</i> .	Debajo de las cajas de texto le aparece un mensaje diciendo que la contraseña se cambió con éxito. Este dice: <i>Password changed successfully</i> .
<b>Paso 6</b>	Continúa el flujo de la misma forma que el paso 5 en adelante del caso de prueba 8.1.	

4.27. Tabla de caso de prueba 8.2

#### 4.3.8.3. Cambio de Foto de Perfil

<b>ID</b>	<b>8.3</b>
<b>Título</b>	<b>Cambio de Foto de Perfil</b>
<b>Prioridad</b>	Media

<b>Propósito</b>	Verificar que la foto de perfil se cambia correctamente.	
<b>Precondiciones</b>	*Tener un usuario registrado.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea y se dirige a <i>My Account</i> .	Se muestra la sección de configuración de datos personales ( <i>Settings</i> ).
<b>Paso 3</b>	El usuario cliquea en <i>Click here to upload photo</i> y selecciona una foto para foto de perfil.	Se muestra la foto que el usuario seleccionó en el cuadro de foto de perfil.

4.28. Tabla de caso de prueba 8.3

#### 4.3.8.4. Modificación Exitosa de Datos Personales

<b>ID</b>	8.4	
<b>Título</b>	<b>Modificación Exitosa de Datos Personales</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que la modificación de los datos personales se realice correctamente.	
<b>Precondiciones</b>	*Tener un usuario registrado.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea y se dirige a <i>My Account</i> .	Se muestra la sección de configuración de datos personales ( <i>Settings</i> ) en la que el campo del email del usuario logueado está poblado.
<b>Paso 3</b>	El usuario ingresa los siguientes datos: En <i>Personal Info</i> : <i>Full Name</i> : Juan Brum	Aparece un modal comunicando que la información personal fue guardada. El texto dice: <i>Your account information has been</i>

	<p><i>Telephone: 4567778989</i>  <i>Address: 214 2nd Street</i>  <i>City: New York</i>  <i>State: New York</i>  <i>En Billing Address:</i>  <i>Address: 214 2nd Street</i>  <i>City: New York</i>  <i>State: New York</i>  <i>Luego hace clic en Save All.</i></p>	<i>saved!</i>
<b>Paso 4</b>	<p>El usuario se desloguea y se vuelve a loguear. Luego accede a <i>My Account</i>.</p>	Su información personal está pre populada en las cajas de texto correspondientes.

4.29. Tabla de caso de prueba 8.4

#### 4.3.8.5. Modificación con Errores de Datos Personales

<b>ID</b>	<b>8.5</b>	
<b>Título</b>	<b>Modificación con Errores de Datos Personales</b>	
<b>Prioridad</b>	<b>Alta</b>	
<b>Propósito</b>	Verificar que si el usuario ingresa datos incorrectamente o no ingresa datos requeridos no pueda insertar sus datos personales y se le comunique.	
<b>Precondiciones</b>	*Tener un usuario registrado sin datos personales ingresados.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea y se dirige a <i>My Account</i> .	Se muestra la sección de configuración de datos personales ( <i>Settings</i> ) en la que el campo del email del usuario logueado está poblado.
<b>Paso 3</b>	El usuario ingresa los siguientes datos:	La caja de texto del email queda con un borde rojo y se muestra un

	<p>En <i>Personal Info</i>:</p> <p><i>Full Name</i>: Juan Brum</p> <p><i>Telephone</i>: 4567778989</p> <p><i>Address</i>: 214 2nd Street</p> <p><i>City</i>: New York</p> <p><i>State</i>: New York</p> <p>En <i>Billing Address</i>:</p> <p><i>Address</i>: 214 2nd Street</p> <p><i>City</i>: New York</p> <p><i>State</i>: New York</p> <p>Luego borra el email y hace clic en <i>Save All</i>.</p>	<p>mensaje debajo del botón de <i>Save All</i> que dice que hay datos incorrectos o incompletos. El texto dice: <i>Not all required info is correct or complete.</i></p>
<b>Paso 4</b>	El usuario ingresa el email pero sin el arroba. Hace clic en <i>Save All</i> .	El mensaje de error sigue apareciendo y la caja de texto del email sigue marcada con rojo.
<b>Paso 5</b>	El usuario ingresa el email correctamente. Modifica el teléfono a un número con menos de 10 caracteres. Hace clic en <i>Save All</i> .	El mensaje de error debajo del botón sigue apareciendo. También aparece un mensaje debajo de la caja de texto del teléfono que dice que debe ingresar un teléfono válido. El texto dice: <i>Valid phone number is required.</i> Esta caja de texto también se marca en rojo.
<b>Paso 6</b>	El usuario ingresa un número de teléfono con al menos 10 caracteres y hace clic en <i>Save All</i> .	Se guardan los datos correctamente.

4.30. Tabla de caso de prueba 8.5

#### 4.3.9. Eventos Pasados

##### 4.3.9.1. Ver Eventos Pasados

<b>ID</b>	9.1
<b>Título</b>	<b>Ver Eventos Pasados</b>
<b>Prioridad</b>	Media
<b>Propósito</b>	Verificar que los eventos que ya pasaron se muestren en la lista de eventos pasados.
<b>Precondiciones</b>	*Tener eventos en la base de

	<p>datos.</p> <p>*Contar con un usuario registrado.</p> <p>*Contar con al menos un evento que tenga como fecha de finalización una fecha anterior a la actual.</p> <p>*El usuario debe de estar registrado a al menos uno de los eventos mencionados en el segundo punto.</p>	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea, hace clic en <i>My Account</i> y luego en <i>Past Events</i> .	Se muestra una lista de eventos a los que está registrado y ya pasaron.
<b>Paso 3</b>	El usuario hace clic en cualquiera de los eventos de la lista.	Se muestra el detalle del evento. En el cuadro del lado derecho de la pantalla se ve un texto que dice que es un evento pasado. El texto dice: <i>This is a past event, you cannot register.</i>

4.31. Tabla de caso de prueba 9.1

#### 4.3.10. Eventos Guardados

##### 4.3.10.1. Eliminar un Evento Guardado desde la Lista de Eventos Guardados

<b>ID</b>	<b>10.1</b>	
<b>Título</b>	<b>Eliminar un Evento Guardado desde la Lista de Eventos Guardados</b>	
<b>Prioridad</b>	<b>Media</b>	
<b>Propósito</b>	<b>Verificar que se puede eliminar un evento guardado en la lista de eventos guardados.</b>	

<b>Precondiciones</b>	*Tener eventos en la base de datos. *Tener un usuario registrado. *Que el usuario tenga un evento guardado.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario se loguea, hace clic en <i>My Account</i> y luego en <i>Saved Events</i> .	Se muestra el evento que tiene guardado. Del lado derecho del evento se ve una cruz para eliminarlo.
<b>Paso 3</b>	El usuario hace clic en la cruz roja para eliminar el evento.	El evento es quitado y se puede ver un texto diciendo que no tiene eventos guardados. El texto dice: <i>You have no saved events!</i>

4.32. Tabla de caso de prueba 10.1

#### 4.3.11. Eventos Registrados

##### 4.3.11.1. Acceder a Eventos Registrados

<b>ID</b>	11.1	
<b>Título</b>	<b>Acceder a Eventos Registrados</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario pueda acceder a los eventos a los que se registró.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Tener un usuario registrado. *Que el usuario se haya registrado a un evento y este no haya pasado.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.

<b>Paso 2</b>	El usuario se loguea, cliquea en <i>My Account</i> y luego en <i>Registered Events</i> .	Se muestra el evento al que está registrado.
<b>Paso 3</b>	El usuario cliquea en el evento.	Es llevado a la página de detalle del evento donde puede consultar la información para asistir al evento y la información de pago.

4.33. Tabla de caso de prueba 11.1

#### 4.3.12. Registrarse a un Evento

##### 4.3.12.1. Seleccionar Tickets

<b>ID</b>	<b>12.1</b>	
<b>Título</b>	<b>Seleccionar Tickets</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario pueda seleccionar tickets.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle del evento para el que se quiere registrar y hace click en <i>GET TICKETS</i> .	Se muestra un modal en el que se encuentran los tickets disponibles.
<b>Paso 2</b>	El usuario selecciona la cantidad.	Se le permite hacer click en el botón de <i>Checkout</i> .
<b>Paso 3</b>	El usuario hace clic en <i>Checkout</i> .	Es llevado a la página de registro para el pago del evento o para simplemente llenar un formulario con sus datos.

4.34. Tabla de caso de prueba 12.1

##### 4.3.12.2. Registro Sin Seleccionar Tickets

<b>ID</b>	<b>12.2</b>	
<b>Título</b>	<b>Registro Sin Seleccionar Tickets</b>	
<b>Prioridad</b>	Alta	

<b>Propósito</b>	Verificar que el usuario pueda seleccionar tickets y no se le permita registrarse al evento sin haberlo hecho.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle el evento para el que se quiere registrar y hace click en <i>GET TICKETS</i> .	Se muestra un modal en el que se encuentran los tickets disponibles.
<b>Paso 2</b>	El usuario no selecciona la cantidad de las dropdowns que aparecen para cada ticket.	No se le permite hacer clic en el botón de <i>Checkout</i> . El modal no se cierra hasta que el usuario no selecciona la cantidad de tickets o hace clic en cerrar.

4.35. Tabla de caso de prueba 12.2

#### 4.3.12.3. Registro a Evento Dentro del Tiempo Establecido

<b>ID</b>	12.3	
<b>Título</b>	<b>Registro a Evento dentro del Tiempo Establecido</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario complete el formulario dentro del tiempo establecido.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haber seleccionado la cantidad de tickets a comprar.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle el evento para el que se quiere registrar y hace clic en <i>GET TICKETS</i> y selecciona la cantidad de tickets que quiere comprar.	Es dirigido al formulario de registro.

<b>Paso 2</b>	El usuario completa el formulario en el tiempo estipulado por el timer en la página.	Se le permite realizar el registro exitosamente. Se le envían mails con la información y se lo dirige a la página de agradecimiento.
---------------	--	--

4.36. Tabla de caso de prueba 12.3

#### 4.3.12.4. Registro a Evento Fuera del Tiempo Establecido

<b>ID</b>	12.4	
<b>Título</b>	<b>Registro a Evento Fuera del Tiempo Establecido</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario no pueda completar el formulario fuera del tiempo establecido.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haber seleccionado la cantidad de tickets a comprar.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle el evento para el que se quiere registrar y hace clic en <i>GET TICKETS</i> y selecciona la cantidad de tickets que quiere comprar.	Es dirigido al formulario de registro.
<b>Paso 2</b>	El usuario no completa el formulario en el tiempo estipulado por el timer en la página.	Se le muestra un mensaje diciendo que se le ha terminado el tiempo para completar el formulario y que debe volver a realizar el proceso de registro. Se deshabilita el botón de registro.

4.37. Tabla de caso de prueba 12.4

#### 4.3.12.5. Registro Exitoso a Evento Gratuito

<b>ID</b>	12.5	
<b>Título</b>	<b>Registro Exitoso a Evento Gratuito</b>	

<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario se pueda registrar a un evento gratuito.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haber seleccionado la cantidad de tickets a comprar.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle del evento para el que se quiere registrar y hace click en <i>GET TICKETS</i> y selecciona la cantidad de tickets que quiere comprar.	Es dirigido al formulario de registro.
<b>Paso 2</b>	El usuario completa todos los datos del formulario, el mail de confirmación es igual al mail y el teléfono es válido.	No aparecen avisos en el formulario de que falta información o de que algún campo es inválido.
<b>Paso 3</b>	El usuario hace clic en el botón <i>Register</i> .	Se le muestra un modal diciendo que el registro fue exitoso. Se inserta en la base de datos un registro para el invoice y uno para los eventos registrados del usuario. Se envía la Order generada a Salesforce y se quitan la cantidad de ticket que compró del objeto <i>Ticket Type</i> .
<b>Paso 4</b>	El usuario hace click en el botón de cerrar.	Se lo redirige a una página de agradecimiento.

4.38. Tabla de caso de prueba 12.5

#### 4.3.12.6. Registro a Evento con Formulario Inválido

<b>ID</b>	<b>12.6</b>	
<b>Título</b>	<b>Registro a Evento con Formulario Inválido</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario no se pueda registrar si los datos son inválidos o si el formulario está	DATOS INVÁLIDOS: - Teléfono (si tiene menos de 10 números)

	incompleto.	- Email (si no coinciden el mail y el mail de confirmación)  DATOS OBLIGATORIOS: Todos los campos.
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haber seleccionado la cantidad de tickets a comprar.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle del evento para el que se quiere registrar y hace clic en <i>GET TICKETS</i> y selecciona la cantidad de tickets que quiere comprar.	Es dirigido al formulario de registro.
<b>Paso 2</b>	El usuario completa todos los datos del formulario con datos inválidos.	Aparece un mensaje abajo del campo inválido notificando al usuario sobre esto. No se inserta nada en la base de datos.
<b>Paso 3</b>	El usuario cambia los datos.	Se le permite registrar y se muestra modal indicando el registro exitoso. Si los datos continúan siendo inválidos sigue sin poder registrarse.
<b>Paso 4</b>	El usuario hace click en el botón de cerrar.	Se lo redirige a una página de agradecimiento.

4.39. Tabla de caso de prueba 12.6

#### 4.3.12.7. Registro a Evento Pago

<b>ID</b>	12.7
<b>Título</b>	<b>Registro a Evento Pago</b>
<b>Prioridad</b>	Alta
<b>Propósito</b>	Verificar que el usuario se pueda registrar a eventos pagos.
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haber seleccionado la cantidad

	de tickets a comprar.	
	Descripción	Resultado Esperado
Paso 1	El usuario se dirige al detalle el evento para el que se quiere registrar y hace clic en <i>GET TICKETS</i> y selecciona la cantidad de tickets que quiere comprar.	Es dirigido al formulario de registro.
Paso 2	El usuario completa todos los datos del formulario con datos válidos.	Aparece un modal avisando al usuario que se ha registrado al evento. Se le manda un mail con la información de pago y otro con el QR del evento. Se inserta en la base de datos el <i>invoice</i> y se agrega a los eventos registrados del usuario.
Paso 3	El usuario hace clic en el botón de cerrar.	Se lo redirige a una página de agradecimiento.
	DATOS DE PRUEBA - TARJETAS DE CRÉDITO VÁLIDAS:	
	4242424242424242	Visa
	401288888881881	Visa
	4000056655665556	Visa (debit)
	555555555554444	MasterCard
	378282246310005	American Express
	371449635398431	American Express
	6011111111111117	Discover
	6011000990139424	Discover
	30569309025904	Diners Club
	3852000023237	Diners Club

4.40. Tabla de caso de prueba 12.7

#### 4.3.12.8. Registro a Evento Pago Con Datos de Inválidos

ID	12.8
Título	Registro a Evento Pago Con Datos de Pago Inválidos

Prioridad		Alta	
Propósito	Verificar que el usuario no se pueda registrar al evento si los datos de pago son incorrectos.		
Precondiciones	*Contar con eventos en la base de datos. *Haber seleccionado la cantidad de tickets a comprar.		
	<b>Descripción</b>		<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al detalle el evento para el que se quiere registrar y hace clic en <i>GET TICKETS</i> y selecciona la cantidad de tickets que quiere comprar.		Es dirigido al formulario de registro.
<b>Paso 2</b>	El usuario completa todos los datos del formulario con los datos de pago inválidos.		Aparece un mensaje abajo del campo inválido notificando al usuario sobre esto. Si el la tarjeta está vencida se visualizará un mensaje abajo de los campos de fecha. Si la tarjeta es inválida se mostrará un mensaje en un modal indicando el problema. Si el código es inválido, se muestra un cartel avisando al usuario.
<b>Paso 3</b>	El usuario cambia los datos.		Se muestra modal indicando el registro exitoso. Si los datos continúan siendo inválidos sigue sin poder registrarse.
<b>Paso 4</b>	El usuario hace click en el botón de cerrar.		Se lo redirige a una página de agradecimiento.
TARJETAS CON ERRORES ESPECÍFICOS DE STRIPE:			
	4000000000000002	La compra se rechaza con un código de card_declined.	
	41000000000000019	La compra se rechaza con un código de card_declined y una razón fraudulenta.	
	40000000000000069	La compra se rechaza con un código de expired_card.	
	400000000000000119	La compra se rechaza con un código de	

		processing_error.
--	--	-------------------

4.41. Tabla de caso de prueba 12.8

#### 4.3.13. Eliminar Registro a un Evento

##### 4.3.13.1. Cancelar Registro a un Evento Gratuito

ID	13.1	
Título	<b>Cancelar Registro a un Evento Gratuito</b>	
Prioridad	Alta	
Propósito	Verificar que el usuario pueda cancelar un registro a un evento.	
Precondiciones	*Contar con eventos en la base de datos. *Haberse registrado a un evento pago.	
	Descripción	Resultado Esperado
Paso 1	El usuario se dirige a <i>My Account, Registered Events</i> y hace clic en el botón <i>Unregister</i> .	Se abre un modal para confirmar.
Paso 2	El usuario confirma que quiere cancelar el registro.	Si envía la información de cancelación a Salesforce y se devuelven los asientos al <i>Ticket Type</i> , lo inserta como <i>cancelled event</i> en la información del usuario y lo borra de los eventos registrados, se muestra un modal avisando al usuario que se realizó con éxito, de lo contrario, se avisa al usuario del error. Se mantiene al usuario en la misma página y si es el último evento se muestra que no tiene más eventos registrados.

4.42. Tabla de caso de prueba 13.1

##### 4.3.13.2. Cancelar Registro a un Evento Pago

ID	13.2	
----	------	--

<b>Título</b>	<b>Cancelar Registro a un Evento Pago</b>	
<b>Prioridad</b>	Alta	
<b>Propósito</b>	Verificar que el usuario pueda cancelar un registro a un evento.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haberse registrado a un evento pago.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige a <i>My Account, Registered Events</i> y hace clic en el botón <i>Unregister</i> .	Se abre un modal para confirmar.
<b>Paso 2</b>	El usuario confirma que quiere cancelar el registro.	Si envía la información de cancelación a Salesforce y se devuelven los asientos al Ticket Type, lo inserta como <i>cancelled event</i> en la información del usuario, se realiza el <i>void</i> en Stripe, se inserta un <i>refund</i> y lo borra de los eventos registrados, se muestra un modal avisando al usuario que se realizó con éxito, de lo contrario, se avisa al usuario del error. Se mantiene al usuario en la misma página y si es el último evento se muestra que no tiene más eventos registrados.

4.43. Tabla de caso de prueba 13.2

#### 4.3.14. Consulta de Promo Code

##### 4.3.14.1. Consultar *Promo Code* Válido

<b>ID</b>	<b>14.1</b>
<b>Título</b>	<b>Consultar <i>Promo Code</i> Válido</b>
<b>Prioridad</b>	Media

<b>Propósito</b>	Verificar que al consultar por un <i>promo code</i> que sea válido, se obtenga la respuesta correcta.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Contar con <i>promo codes</i> válidos para ese evento en salesforce.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario cliquea en un <i>event</i> .	Se muestra el detalle del evento.
<b>Paso 3</b>	El usuario hace clic en <i>Get Tickets</i> .	Se muestra el modal con los tickets que puede comprar y la caja de texto para ingresar el <i>promo code</i> .
<b>Paso 4</b>	El usuario ingresa el <i>promo code</i> válido en la caja de texto y hace clic en el botón <i>Apply Code</i> .	Se hace una consulta a salesforce y devuelve una respuesta exitosa. Aparece un texto debajo de la caja de texto que dice <i>Valid Code</i> y luego la descripción del <i>promo code</i> . Si el usuario llega a hacer una compra, se le aplicará el descuento correspondiente.

4.44. Tabla de caso de prueba 14.1

#### 4.3.14.2. Consultar *Promo Code* Inválido

<b>ID</b>	14.2
<b>Título</b>	Consultar <i>Promo Code</i> Inválido
<b>Prioridad</b>	Media
<b>Propósito</b>	Verificar que al consultar por un <i>promo code</i> que inválido, se obtenga la respuesta correcta.
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Contar con <i>promo codes</i> para ese evento en salesforce.

	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario se dirige al sitio.	Se muestra la página del directorio de eventos con los eventos en la vista de grilla.
<b>Paso 2</b>	El usuario cliquea en un <i>event</i> .	Se muestra el detalle del evento.
<b>Paso 3</b>	El usuario hace clic en <i>Get Tickets</i> .	Se muestra el modal con los tickets que puede comprar y la caja de texto para ingresar el promo code.
<b>Paso 4</b>	El usuario ingresa el <i>promo code</i> inválido en la caja de texto y hace clic en el botón <i>Apply Code</i> .	Se hace una consulta a salesforce y devuelve una respuesta exitosa, pero de código inválido. Aparece un texto debajo de la caja de texto que dice <i>Invalid Code</i> . El descuento perteneciente a ese <i>promo code</i> no podrá ser usado.
<b>Nota</b>	Esto aplica para los <i>promo codes</i> cuyo código no existe para dicho evento, los que ya no están activos y los que no están dentro de las fechas de vigencia.	

4.45. Tabla de caso de prueba 14.2

### 4.3.15. Obtención de PDF

#### 4.3.15.1. Imprimir PDF

<b>ID</b>	15.1	
<b>Título</b>	Imprimir PDF	
<b>Prioridad</b>	Baja	
<b>Propósito</b>	Verificar que el usuario pueda obtener el PDF correctamente.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Tener un usuario registrado. *Haberse registrado a un evento.	
	<b>Descripción</b>	<b>Resultado Esperado</b>

<b>Paso 1</b>	El usuario se dirige al evento al que está registrado y del que quiere obtener el PDF.	Se muestra la página de detalles del evento.
<b>Paso 2</b>	El usuario cliquea en un el botón para imprimir el PDF ( <i>GET PDF</i> , según corresponda).	Se descarga un archivo PDF que tiene los mismos estilos que la página de la que obtuvo el PDF.
<b>Nota:</b>	Esto también aplica para el botón <i>Print</i> que se encuentra en la página de agradecimiento que aparece luego de realizar un registro.	

4.46. Tabla de caso de prueba 15.1

#### 4.3.16. Envío de mail

##### 4.3.16.1. Envío de Email

<b>ID</b>	16.1	
<b>Título</b>	Envío de Email	
<b>Prioridad</b>	Media	
<b>Propósito</b>	Verificar que el usuario recibe dos emails, uno con el código QR y otro con el invoice.	
<b>Precondiciones</b>	*Contar con eventos en la base de datos. *Haberse registrado a un evento.	
	<b>Descripción</b>	<b>Resultado Esperado</b>
<b>Paso 1</b>	El usuario completa el formulario de registro correctamente y hace clic en <i>Register</i> .	Se le envían dos mails, uno con el QR y otro con el detalle de la compra. Se le muestra el modal de registro exitoso.

4.47. Tabla de caso de prueba 16.1

## 4.4. Conclusiones del Proyecto

### 4.4.1. Conclusiones en cuanto a la Interfaz de Usuario

- Creemos que se hizo un buen trabajo en la interfaz de usuario diseñada para pantallas grandes y medianas. La página está ordenada, no es complicada para comprender por el usuario. Los componentes no tienen comportamientos inesperados.
- En cuanto al objetivo de que la página sea responsive se trató de hacer el mejor trabajo posible, pero tenemos claro que en algunos casos los componentes no se ajustan como quisiéramos a todos los tamaños de pantallas. Sin embargo creemos que se hicieron grandes avances en nuestro conocimiento del tema, ya que ninguna tenía experiencia en diseño responsive. Además, somos conscientes que fue algo que fue quedando para atrás ya que se le dio más relevancia a funcionalidades que eran más importantes para el cliente y que estaban planificadas para seguir adelante con el proyecto en general. Por lo tanto es uno de los objetivos que no está totalmente completado y queda pendiente.
- También creemos que en el detalle de evento hay partes que pueden ser mejoradas, como la presentación de los speakers y sponsors, que como máximo muestra tres por fila, entonces si hay cuatro, por ejemplo, el cuarto se muestra en una segunda fila y a la izquierda.
- El diseño tampoco es demasiado consistente, esto se debió a que hubo algunas páginas que las diseñamos nosotras ya que no tenían mockups para las mismas.

### 4.4.2. Conclusiones en cuanto a las Funcionalidades

- Consideramos fundamental destacar que de manera general se pudieron completar todas las funcionalidades más importantes para el cliente. Se pudieron sobrelevar los obstáculos que surgieron de trabajar con herramientas

nuevas y por esto mismo creemos que fue una buena elección y experiencia haber decidido trabajar de esta forma.

- Más allá del punto anterior sí hubo ciertas demoras y las estimaciones lamentablemente no siempre representaron las horas reales, ya sea a causa de demoras en el desarrollo, bugs, desconocimiento, requisitos que aún no estaban totalmente definidos, etc. Debido a esto, en esta primera versión de la aplicación quedaron algunas historias que hubo que sacar de la planificación ya que no iba a ser posible completarlas dentro del período de preparación del proyecto. Entre estas están:
  - Notificaciones en vivo y configuración de notificaciones
  - Guardado de foto de perfil perteneciente a redes sociales
  - Pago con Paypal
  - Pago offline

Es importante aclarar que estas funcionalidades se dejaron afuera con conocimiento del cliente, quien fue el que decidió qué cosas prefería no tener en esta versión del sistema y en cuáles prefería que nos concentráramos.

- En cuanto a la seguridad de las APIs, somos conscientes que estas no tienen seguridad actualmente. Esto, sin embargo, fue hecho a propósito y discutido con el cliente para facilitar el desarrollo de la aplicación y se planea agregar seguridad a las mismas cuando se lleve la aplicación a producción.
- Consideramos dentro de las fortalezas de la aplicación los siguientes:
  - Todos los datos en MongoDB están sincronizados con Salesforce.
  - Es una aplicación intuitiva.
  - La vista de los eventos es altamente configurable.
  - Se utilizan tecnologías actuales.
  - Las formas de pago son externas a la aplicación y seguras. Esto es importante ya que el cliente no quería manejar dinero.
  - El usuario puede loguearse con redes sociales, más allá de lo discutido más arriba.

#### 4.4.2.1. Bugs Conocidos

Como fue dicho anteriormente, existen bugs que son de nuestro conocimiento y que por un tema de tiempo no han podido ser resueltos. A continuación se detallan los mismos:

- Para que funcione el login con redes sociales se debe utilizar la página con el protocolo http (<http://rsvpday-demo.herokuapp.com>), no https.
- La foto de perfil puede guardarse en una primera instancia, pero a partir de esa vez algunas veces el guardado falla y otras veces no. Todavía no hemos podido identificar cuál es el problema.
- Twitter y Linkedin no devuelven mail cuando el usuario se loguea mediante los mismos. Esto no es un bug nuestro en sí, pero sí el usuario debe ingresar su mail cada vez que lo necesita para algo.
- El botón para mostrar más eventos que se encuentra al final de la página del directorio desplaza al usuario hacia la parte superior de la pantalla cuando se aprieta. Estuvimos investigándolo, pero creemos que es algo que se va a tener que manejar manualmente con código para que se mantenga la vista en el mismo lugar.
- Las fechas de los eventos se cambia para un día antes. Hasta ahora lo arreglamos cambiándole la timezone a todo como si estuviéramos en la zona horaria UTC, pero somos conscientes de que hay que hacer un refactor para manejar mejor las fechas.

#### 4.4.3. Conclusiones en cuanto al Aprendizaje

- El aprendizaje de nuevas herramientas fue una parte fundamental del proyecto. Consideramos que era un muy buen desafío salir de nuestra zona de confort e investigar y utilizar herramientas que no conocíamos o que habíamos usado muy poco en el pasado. Al mismo tiempo nos parece una habilidad muy importante de desarrollar ya que el desarrollador debe someterse a un aprendizaje constante a lo largo de su carrera.
- Sin embargo, somos conscientes de que utilizar tecnologías desconocidas implicaba utilizar gran parte del tiempo del proyecto en leer documentación y aprender por ensayo y error. La curva de aprendizaje que tuvimos podría haberse sustituido por funcionalidades más complejas o por más funcionalidades. Más allá de esto creemos que valió la pena.
- Las herramientas y tecnologías nuevas que se aprendieron fueron:
  - MeteorJS - ningún conocimiento previo

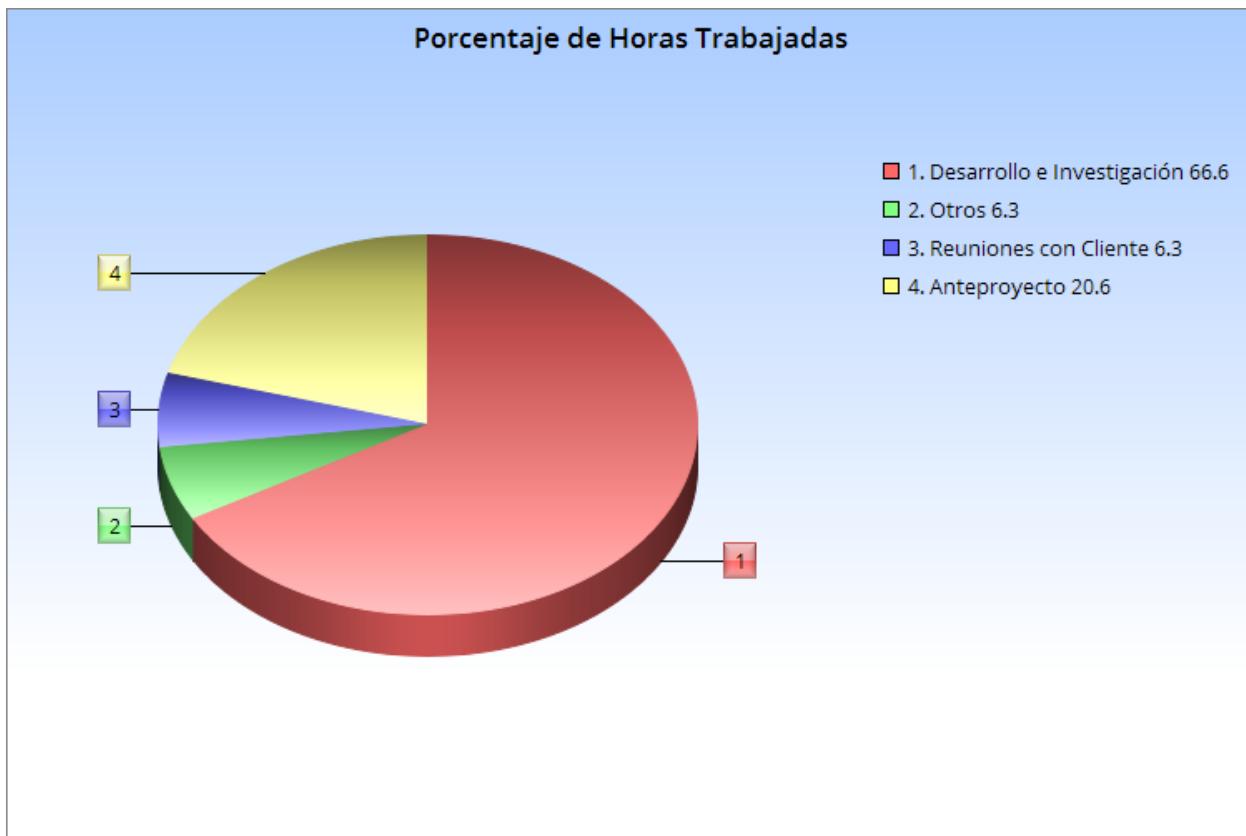
- AngularJS - conocimiento básico previo
  - Bootstrap - conocimiento básico previo
  - MongoDB - ningún conocimiento previo
  - Manejo y creación de APIs - conocimiento anterior pero no específicamente con las APIs utilizadas en el proyecto
  - Heroku - ningún conocimiento previo
- Aprendimos que los tiempos siempre varían en el desarrollo de un proyecto. Más aún cuando hay mucho por investigar. Lo cual hace más difícil hacer estimaciones. Sin embargo, fue una buena práctica para el futuro.
- Aprendimos a mantener una comunicación con un cliente real y a mediar con lo que el cliente quería y los tiempos y por lo tanto a flexibilizar las metodologías de trabajo a las realidades de nuestro proyecto.
- Aprendimos la importancia de estimar y ser capaces de medir nuestro esfuerzo para no comprometernos a cosas que no íbamos a poder cumplir.
- Aprendimos la importancia de tener casos de prueba con resultados esperados bien definidos ya que logramos detectar bugs gracias a ellos.
- Aprendimos la importancia de tener buenas definiciones de tareas con sus respectivos criterios de aceptación ya que varias veces debimos comenzar nuevamente las tareas por falta de claridad en los requerimientos.

#### 4.4.4. Conclusiones en Cuanto a los Riesgos

Los dos riesgos principales que estimábamos que podían poner en riesgo el proyecto eran a causa de desconocimiento de las tecnologías a utilizar y por fallas en las estimaciones. A lo largo del proyecto, ambos representaron obstáculos que tuvimos que solucionar de una manera u otra. Sin embargo, creemos que a largo plazo los mismos fueron vencidos. Esto se debió a que hubo una buena comunicación con el cliente y el tutor y ambos estuvieron al tanto del estado actual del proyecto y nos ayudaron a redirigirlo en los momentos más críticos. Por lo tanto, estos riesgos simplemente terminaron representando cambios en las estimaciones y en las prioridades de los requerimientos con absoluto conocimiento del cliente.

#### 4.4.5. Conclusiones en Cuanto a las Estimaciones

El total de horas trabajadas fue de más de 600 horas. Estas estuvieron repartidas entre 130 horas de trabajo en el anteproyecto, aproximadamente 40 horas de reuniones con el cliente durante el desarrollo de la aplicación, más de 450 horas de desarrollo de la aplicación, y alrededor de 40 horas dedicadas a otras tareas (otras tareas relacionadas con la documentación, reuniones entre nosotras, con el tutor, etc.).



4.7. Imagen de porcentaje de horas trabajadas totales

#### 4.4.6. Conclusiones Generales

- Consideramos que en términos generales los objetivos del proyecto fueron logrados. Existen, sin dudas, cosas que quedaron sin poderse completar, pero estas no representaban un problema para el funcionamiento global de la aplicación y fueron elegidas a conciencia para que queden afuera. Pero como primera versión de una aplicación que implementaba tantas herramientas nuevas, creemos que el resultado final es un éxito.

- Al mismo tiempo, cabe destacar que el cliente quedó satisfecho y a gusto con el resultado obtenido hasta la fecha de la entrega del proyecto.

# Capítulo 5

## Bibliografía

- AngularJS. “AngularJS Developer Guide”. [En línea]. Disponible: <https://docs.angularjs.org/guide>
- IEEE. “IEEE Editorial Style Manual”, octubre, 2014. [En línea]. Disponible: [http://www.ieee.org/documents/style\\_manual.pdf](http://www.ieee.org/documents/style_manual.pdf)
- Meteor. “Meteor. 1.2.1 Docs”. [En línea]. Disponible: <http://docs.meteor.com/#/full/>
- Pressman, R.S. *Ingeniería del Software. Un Enfoque Práctico.* 7ma ed. México D.F., México: McGraw-Hill Educación, 2010.
- Sommerville, Ian. *Ingeniería de Software.* 7ma ed. Madrid, España: Pearson Educación, 2005.

# Glosario

**API:** Siglas en inglés para *Application Program Interface* (interfaz de programación de aplicaciones). Es un conjunto de protocolos, funciones y comandos para desarrollar software y aplicaciones y ser utilizado por otro software.

**CSS:** Siglas en inglés para *Cascading Style Sheets*. Lenguaje para darle estilo a un documento escrito en lenguaje de marcado.

**HTML:** Siglas en inglés para *HyperText Markup Language*. Lenguaje de marcado estándar para la creación de páginas web.

**IDE:** Siglas en inglés para *Integrated Development Environment*. Es una aplicación de software que ofrece determinadas funcionalidades para desarrollar software.

**MEAN stack:** Framework de Javascript que engloba tanto front como backend. Utiliza las herramientas MongoDB, ExpressJS, AngularJS y NodeJS.

**NPM:** *Node Package Manager*. Se utiliza para gestionar paquetes de Node.js.

**REST:** Siglas en inglés para *Representational State Transfer* (transferencia de estado representacional). Estilo de arquitectura de software y enfoque de comunicaciones que a veces se utiliza en el desarrollo de servicios web. Implica una interfaz entre sistemas que utiliza HTTP para obtener datos o ejecutar operaciones.

**ROI:** Siglas en inglés para *Return on Investment*, es una medida de desempeño para evaluar la eficiencia de una inversión.

**RSVP:** Del francés *Repondez, s'il vous plaît*, que significa “Por favor, responda.” Se utiliza actualmente mucho en las redes sociales relacionado a eventos para que los invitados confirmen su asistencia al evento.

**Scrum:** Metodología ágil de desarrollo de software iterativa e incremental para la gestión de un proyecto de desarrollo.

**Servicio web:** En términos generales es un servicio diseñado para soportar la interacción entre un sistema y otro mediante la *World Wide Web*.

**SMBs:** *Small Medium Businesses*. Empresas de tamaño pequeño y medio.

# Anexo

## A.1. Actas de Reuniones con el Cliente

A continuación se detallan las actas de las reuniones llevadas a cabo con el cliente. No todas las ideas presentadas en estas actas terminaron llevándose a cabo, simplemente son una fiel representación de lo que se habló en cada reunión.

El vocabulario y estilo de esta parte difiere un poco del resto del documento porque son notas tomadas al momento de la reunión, por lo tanto hay abreviaciones y palabras en inglés y español mezcladas.

Acta de Reunión con el Cliente Nr. 1	
<b>Fecha:</b>	22/09/2015
<b>Personas reunidas:</b>	Diego Febles, Angélica Buffa, Marianne Maisonneuve, Ximena Lasserre
<b>Temas:</b>	<ul style="list-style-type: none"><li>1) Generalidades de RSVPDay</li><li>2) Actuales competidores en el mercado</li><li>3) Descripción de cómo es el mercado actualmente y cómo son las aplicaciones de este tipo</li><li>4) Necesidades del cliente</li><li>5) Estructura y funcionalidades del sistema</li></ul>
<b>Notas:</b>	<ul style="list-style-type: none"><li>1) Usuarios pueden RSVP en el evento</li><li>2) Event management: dos partes privada y pública</li><li>3) Nosotros vamos a hacer la parte pública: cualquier usuario puede ingresar, visualizar eventos, buscarlos, ordenarlos</li><li>4) Mandar invitaciones por mail</li><li>5) Competidores: Linvio Events, Eventbrite</li><li>6) La pública se conecta con la privada por la api, nosotros tendríamos que conectarnos</li><li>7) Event management de Appexchange en salesforce son muy caras y no cubren todos las funciones que se podría, y el costo de instalación también es alto porque manejan el sistema de pagos, ellos quisieran manejar el sistema de pago mediante paypal o authorize.net o stripe.</li></ul>

	8) Quieren que no se maneje dinero, que el usuario que se registra asocie su cuenta de paypal, etc. y se haga mediante eso
	9) Las aplicaciones actuales apuntan o a empresas grandes o a pequeñas, pero no ambas, ellos quieren tanto negocios grandes como chicos puedan usarlo.
	10) Otro problema es el sistema de envío de email, quieren usar un sistema externo (mailchimp, etc), actualmente usan servidores propios, por eso también es más caro.
	11) ¿Por qué Salesforce? SRM líder a nivel mundial, ellos trabajan con Salesforce, dentro del mercado de Salesforce hay un nicho muy grande de mercado para aplicaciones que manejen eventos.
	12) Hosteado heroku, frontend angular, look & feel bootstrap u otro framework (investigar).
	13) Login: lo más probable es que sea facebook y google connect.
	14) Sitio responsive
	15) Parte del registro quieren integrarlo con QR code y que llegue a un mail y la persona pueda ir al evento con ese código, pasar y registrarse
	16) A partir de cada paquete instalado en cada org, se va a generar una url del lado del sitio público para ese org, tipo xyz.rsvpd.com
	17) Posibilidad de integración con redes sociales
<b>Próxima reunión:</b>	29/09/2015

A.1.1. Tabla de acta de reunión con cliente nr. 1

<b>Acta de Reunión con el Cliente Nr. 2</b>	
<b>Fecha:</b>	29/09/2015
<b>Personas reunidas:</b>	Diego Febles, Marianne Maisonneuve, Ximena Lasserre
<b>Temas:</b>	1) Casos de uso
<b>Notas:</b>	1) Directorio de eventos: Los eventos ya van a existir. Filtrar por locations, por regiones (se va a poder buscar por dirección, integrar con google maps, por ejemplo a 10 km de esa dirección), por tipos de eventos (lo más probable que sean tags), busca por palabras, por rango de fechas o por

	<p>período (<i>next week, next month</i>). Los filtros van a ser aditivos, se van a ir agregando fecha y región por ej. La idea sería que a medida que vas agregando se van cargando las búsquedas.</p> <p>También se va a poder hacer un sorting de los eventos, por ej, por evento, por nombre de evento, por región.</p> <p>Paginación, puede ser una paginación de mostrame tantos por pág o un easy loading. [A definir]</p> <p>En cuanto a la vista de los eventos, tienen fecha, nombre, hora, ubicación, descripción, precio, capaz que estaría bueno poner una foto de los speakers chiquita o número de speakers. También podrían aparecer tags de los events.</p>
	<p>2) Detalle de evento: Va a tener overview/descripción del evento que va a venir con imágenes. En este también va a haber sponsors que tendrán logo y url para la página del sponsor. Otra parte que va a poder ver es el detalle, que va a ser texto y otra parte que tenga info de los speakers que va a tener foto, nombre y descripción de los speakers. También la idea es que se vea un mapa con la ubicación y los datos de la location. También se puede ver información del contacto que es un usuario con su info de contacto. El evento arriba va a mostrar de alguna forma fecha, título, botón de registro. También se va a poder imprimir el evento, que va a generar un pdf. También se puede agregar al calendario de google el evento.</p>
	<p>3) Registro a un evento: Va a poder ver descripción del evento, con datos del mismo. Siguen botones de print y add to calendar. Se va a mostrar el order summary, tiene evento, fecha, total y caja de texto para poner discount code. Desde acá hay dos integraciones, una con Stripe que es la que manda pago y un callout a Salesforce para mandar la info del registro. El discount code lo pone ahí y si existe en Salesforce le manda que es válido o de lo contrario error. Abajo el usuario va a poder loguearse o continuar como guest, si sigue como guest se le abre un formulario de pago en donde va a tener que agregar todos los datos de contacto para pagar y procesar tarjeta de crédito. También formulario para tarjeta de crédito. Y checkbox si billing address es la misma que payment address. Si no, pone más cajas de texto para datos de billing address. Si pone pay now entonces le manda mail con la info y QR code al</p>

	usuario. Y algo que le diga que se registró. Ver stripe. El payment se puede hacer online u offline (la persona puede pagar después, mandan cheque, proceso manual). Payment incomplete cuando no se manda el pago.
	4) Login: Facebook connect y google Sign-In si se puede. Va a ver un log in desde la home page. También uno cuando vaya a pagar el evento y en cualquier lado que haga save si no está registrado.
	5) My account del usuario: Se pueden ver los eventos que se aproximan para los que están registrados (upcoming), los eventos que se aproximan y que guardaste, los eventos históricos, puede remover eventos guardados y cancelar eventos a los que está registrado (hay que ver si ya pagó, página para ver si está seguro que quiere cancelar y se haga un void en Stripe). Tienen links para ir al detalle. Dentro de My account puede editar su información de contacto, de pago, también un resend invitation en la parte de upcoming events para que envíe un mail de nuevo con el QR code. Si ya estás registrado a un evento no puede registrarte de nuevo, tiene que avisarte que ya estás registrado. Estaría bueno manejar estados de eventos: registrado, payment offline, payment completed, también poder ver la factura si ya pagaste.
<b>Próxima reunión:</b>	5/10/2015

A.1.2. Tabla de acta de reunión con cliente nr. 2

<b>Acta de Reunión con el Cliente Nr. 3</b>	
<b>Fecha:</b>	5/10/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	1) Casos de uso actualizados 2) Creación de las user stories
<b>Notas:</b>	1) Ver directorio de eventos de la empresa: Tres user stories distintas: event type, location, dates 2) Buscar evento en el directorio de eventos: Una user story distinta para buscar eventos

	3) Ver detalle del evento: ese es el caso de uso general, separado por user stories distintas, que después se puede trabajar por partes, listar speakers, locations, sponsors
	4) Registrarse a un evento: uno puede registrarse al evento, puede elegir si registrarse a su cuenta, trae los datos, luego de registrado se manda email. Si es gratis el evento tiene que completarse el registro en 10 minutos, si es pago tiene 30 minutos por default, pero esa info va a estar dada en el objeto evento. Si el tiempo expiró, le avisa (pop up que le avisa y ahí puede reiniciar el registro o salirse, si reinicia los datos se mantienen) y lo manda al detalle del evento nuevamente.
	5) Pagar evento con tarjeta de crédito: tarjetas visa, master card, amex, luego del pago email de confirmación, son 2 emails que le llegan al usuario si el evento es pago, puede surgir que el usuario tenga que elegir qué asientos pueda tener en el evento. En el formulario de pago se va a generar un resumen, tipo línea de factura, informando cuánto se le va a cargar al usuario. Podríamos agregar caso de uso pagar con paypal.
	6) Guardar eventos: Puede guardar el evento desde el detalle de evento y desde el event directory. Si no estás logueado, pop up para loguear, username, password o facebook login o google login. Si decide registrarse por separado le pide nombre, apellido, email, y password (registro simple), en el mismo pop up, la idea es que no se vaya de la page del evento o en la que estuviera.
	7) Compartir eventos: Usuario puede compartir el evento en las redes sociales. Varias user stories, puede compartirlo via email, si le da share se levanta pop up que le permite poner las direcciones con las que quiere compartir y en el body del mail va a haber los details del evento. La idea es que el evento tenga un share mismo en el detalle. Título del evento, descripción, hora, speakers, location. También compartirlo por facebook, hay que pasarle la info del evento a facebook. Con Twitter también lo mismo. Y con linkedin también.
	8) Loguearse a su cuenta: En la página va a haber un sign up y un log in arriba. El caso normal de

	<p>uso de login. Sin que haya un evento al que está tratando de registrarse o guardar. Si el usuario hace sign up se registra y si es log in, lo mismo de siempre (user y password y facebook y google). Se desprenden varias user stories. En el caso de que se registre enviaría un email de confirmación. Y tendría que confirmar su dirección de email. Que sea un formulario de registro básico (first, last name, email, pwd, confirmación y captcha), después en el profile que pueda agregar la información restante. Cuando esté logueado en vez de sign up y login va a aparecer una foto chica del usuario, su nombre y dropdown en el que pueda acceder a su account.</p>
	9) Crear mi cuenta : Ver anterior.
	10) Ver evento guardado: Se podría agregar un botón al dropdown del nombre y cuenta del usuario en el que diga View Saved Events y ahí lo manda al directorio de eventos guardados. En el directorio de eventos no se pueden ver eventos que no tengan disponibilidad. Pero agreguemos user story para lo que es un evento en tiempo real, si el evento se está quedando sin seats que esto sea visible para el usuario. Pensar caso de uso para manejar eventos que ya no tienen asientos. Eso capaz que se puede poner en una pestaña de eventos upcoming pero no disponibles, que pueden guardarse. Cuando quedan pocos eventos, ahí se puede mandar un mail a todos los que tienen eventos guardados. Y si hay un evento del que se liberan asientos, entonces se puede mandar un email a todos los que tienen eventos guardados. En la parte de profile tendría que haber una parte de notifications, para que el usuario pudiera habilitar y deshabilitar cuando quiere que le lleguen este tipo de notificaciones. También estaría bueno que se le pudieran mandar notificaciones por facebook. Y mostrarlas en la página de rsvpday tmb.
	11) Eliminar eventos registrados: Ya hablamos de esto.
	12) Ver eventos registrados: Desde allí puede imprimir pdf del ticket, cancelar el registro (dos casos: de un evento gratuito y de un evento pago, que ahí habría que cancelar el pago). Hay que investigar qué te piden los sistemas como stripe para cancelar un pago. También desde allí se

	puede contactar al organizador, por email o por otro medio que dé el organizer. Esa info viene de salesforce. Esto también se podría mostrar en el detalle del evento.
	13) Cancelar evento registrado: Si es un guest no hay posibilidad de cancelar desde el site. En los mails va a tener que decir cómo se puede hacer para cancelar un evento. En el caso de los guests tienen que decir que contacte al organizador, etc. En el caso de los usuarios registrados le tiene que decir cómo hacerlo en el sitio. Si un usuario cancela tiene que recibir un email diciéndole que se le hizo el void del pago y que se canceló su registro.
	14) Contactar al organizador : Que sea desde el site. Podría ser un popup con el nombre, email y el mensaje. Y que aparezca un mensaje de confirmación una vez que se haya dado correctamente. Puede ser desde el detalle de eventos como del directorio de eventos registrados.
	15) Agregar al calendario : Si el usuario hace esto y no está registrado le tiene que pedir el login de google. Va a haber un dropdown que muestre los calendarios a los que pueden agregarlo, Google calendar, yahoo calendar.
	16) Imprimir ticket: Es el tema de imprimir el ticket desde el formulario de registro. La idea es que genere un pdf con todo el detalle del evento con el QR code, detalles del pago. Debajo de todo en el medio el QR code. Este es tu ticket de acceso al evento, solo cuando alguien ya se registró.
	17) Modificar cuenta: Actualizar nombre, email, teléfono, dirección de contacto, dirección de pago, foto
	18) Modificar password: En el caso de que no esté usando una red social.
	19) Mapa: Capaz que se podría incluir en eventos registrados un mapa con las ubicaciones de los eventos en ese mapa. Pero esto no es seguro todavía.
<b>Próxima reunión:</b>	6/10/2015

A.1.3. Tabla de acta de reunión con cliente nr. 3

Acta de Reunión con el Cliente Nr. 4	
<b>Fecha:</b>	6/10/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	1) Historias de usuario
<b>Notas:</b>	Investigar herramientas que vamos a usar en primer sprint, setear environment, deploy a heroku en el sprint 3 y dps en el sprint 4 haciendo la interfaz de la api ya podemos empezar con el directorio.
	Como esto van a ser distintas empresas que lo usen capaz que podrían ser distintos colores y diseño para cada una
	Filtrar por location: a partir de la dirección y van a estar las coordenadas para el mapa
	Investigar sendgrid que es para mandar emails.
	Los mockups nos los van a proveer ellos.
	Investigar qué herramientas existen para testear node.js y angular
<b>Próxima reunión:</b>	13/10/2015

A.1.4. Tabla de acta de reunión con cliente nr. 4

Acta de Reunión con el Cliente Nr. 5	
<b>Fecha:</b>	13/10/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	1) Sprint 3
<b>Notas:</b>	Para heroku capaz que hacer un app de prueba y tratar de subirla.
	Reordenar tareas de investigación.
	Ver cómo mean stack ordena la estructura de las carpetas para tratar de tener una buena estructura.
	Pasar credenciales de skype para armar grupo.
	Estaría bueno usar git flow.
	Code review cruzados. [Agregarlo a la documentación]
	Investigar integración con jenkins, gerrit, github para correr los tests. También con heroku. Si no lleva

	mucho trabajo se puede agregar en este sprint como parte de setear el environment.
	Utilizar gulp.
	Al final del sprint tendríamos environment con mean stack que utilice gulp como manejador de tareas y tener la versión en el repo.
<b>Próxima reunión:</b>	20/10/2015

A.1.5. Tabla de acta de reunión con cliente nr. 5

<b>Acta de Reunión con el Cliente Nr. 6</b>	
<b>Fecha:</b>	20/10/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	1) Continuación sprint 3 2) Mean.io vs Mean.js
<b>Notas:</b>	Mean.js viene con utilidades. Vamos a usar mean.js Streaming API de SF investigar para usarla para mandar los eventos del org al sitio público.
<b>Próxima reunión:</b>	27/10/2015

A.1.6. Tabla de acta de reunión con cliente nr. 6

<b>Acta de Reunión con el Cliente Nr. 7</b>	
<b>Fecha:</b>	27/10/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	1) Los eventos van a tener más de un precio? 2) Próximas tareas
<b>Notas:</b>	Agregar event section a db. Es un objeto dentro del event. Tiene content, speaker, image, order, subject. Otro objeto que el event va a tener es un product, este va a tener el unit price, discount. Cómo agregar oauth 2.0 en node Dado que mean.js y meanio presentan problemas para instalar y desarrollar los mismos, investigar sails vs loopback vs meteor.

<b>Próxima reunión:</b>	3/11/2015
-------------------------	-----------

A.1.7. Tabla de acta de reunión con cliente nr. 7

<b>Acta de Reunión con el Cliente Nr. 8</b>	
<b>Fecha:</b>	3/11/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	1) Tareas hechas y a hacer 2) Avances hasta ahora: configuración del ambiente y dummy data
<b>Notas:</b>	Comenzamos con las tareas del directorio de eventos. Marianne hace el html base y Ximena hace el template del directorio. Hay que crear estructura de carpetas del proyecto que incluya server, client, models por lo menos.
<b>Próxima reunión:</b>	10/11/2015

A.1.8. Tabla de acta de reunión con cliente nr. 8

<b>Acta de Reunión con el Cliente Nr. 9</b>	
<b>Fecha:</b>	10/11/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	Tareas del sprint pasado Modelo de los documentos de mongodb. Nuevo mockup de detalle de evento. Rest API (Restivus)
<b>Notas:</b>	Cómo hacemos la funcionalidad de las locations. Traer las locations cuando el usuario va a buscar o cuando se carga la aplicación.
<b>Próxima reunión:</b>	8/12/2015

A.1.9. Tabla de acta de reunión con cliente nr. 9

<b>Acta de Reunión con el Cliente Nr. 10</b>	
<b>Fecha:</b>	8/12/2015

<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	Preguntas acerca de la secciones de un evento: Colores dinámicos. Orden.
	Tests unitarios. La empresa que daba soporte a Velocity dejó de hacerlo.
<b>Notas:</b>	Velocity igual sigue siendo el paquete oficial de testing. Y se supone que meteor development group le va a dar soporte.
	Buscar dos o tres opciones para mandar mails en masa. Sendgrid por ej.
	Investigar rest api de salesforce para mandar info.
<b>Próxima reunión:</b>	15/12/2015

A.1.10. Tabla de acta de reunión con cliente nr. 10

<b>Acta de Reunión con el Cliente Nr. 11</b>	
<b>Fecha:</b>	15/12/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	
<b>Notas:</b>	Menos prioridad offline payment, se van a dejar esas stories para después.
	Se definieron algunos casos de uso nuevos del payment.
	Quieren pasar el payment para después (sprint 8) y antes hacer my account (sprint 7).
<b>Próxima reunión:</b>	12/01/2016

A.1.11. Tabla de acta de reunión con cliente nr. 11

<b>Acta de Reunión con el Cliente Nr. 12</b>	
<b>Fecha:</b>	12/1/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	Pago y registro de un evento
	Bugs actuales

<b>Notas:</b>	Actualmente tenemos dos problemas: el bug de logout que al salir de la sesión siguen viéndose botones de ese usuario o sus datos personales y que tenemos que actualizar la versión de meteor-angular que es el paquete principal que usamos y que cuando actualizamos se cae la aplicación.
	Las historias de usuario del registro a un evento las vamos a agregar a pivotal tracker esta semana.
<b>Próxima reunión:</b>	26/1/2015

A.1.12. Tabla de acta de reunión con cliente nr. 12

Acta de Reunión con el Cliente Nr. 13	
<b>Fecha:</b>	19/1/2015
<b>Personas reunidas:</b>	Marianne Maisonneuve, Ximena Lasserre, Diego Febles
<b>Temas:</b>	Implementación de pago.
	Pasos finales y redondeo para las próximas semanas.
<b>Notas:</b>	
<b>Próxima reunión:</b>	27/01/2016

A.1.13. Tabla de acta de reunión con cliente nr. 13

## A.2. Casos de Uso Detallados

A continuación se especifican los casos de uso:

<b>ID de Caso de Uso:</b>	CU-001
<b>Nombre:</b>	Alta de un registro a un evento sin discount code
Fecha:	28/09/2015
Actores:	Usuario logueado y usuario no logueado
Descripción:	Este caso de uso describe la inscripción de un usuario (que se loguea en el transcurso del caso de uso) a un evento perteneciente a la empresa sin discount code.

Precondiciones:	<ul style="list-style-type: none"> <li>• Ingresar al sitio</li> <li>• Tener eventos disponibles en el sitio que sean en el futuro</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• El usuario se registra exitosamente a un evento</li> <li>• Se ocupa la cantidad de asientos del evento como haya comprado</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona un evento del directorio de eventos.</li> <li>3. El usuario cliquea el botón <i>Get Tickets</i> para registrarse al evento.</li> <li>5. El usuario selecciona los tickets que desea y hace clic en <i>Checkout</i>.</li> <li>7. El usuario se loguea en la página.</li> <li>9. El usuario ingresa sus datos de pago y hace clic en <i>Pay Now</i> para confirmar la compra.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. Se muestran los datos del evento seleccionado.</li> <li>4. Se despliega un modal para que el usuario ingrese un código de descuento y para que seleccione los tipos y cantidad de tickets que quiere.</li> <li>6. El sitio lo envía a la página de registro donde está el formulario de pago y los datos de la compra.</li> <li>8. Es logueado con éxito y se populan sus datos personales si los tenía guardados.</li> <li>10. Se le muestra un mensaje al usuario de que el registro se hizo correctamente y se le envía un mail con los datos del evento y el pago y un código QR para presentar en el evento. Se guardan los datos en Salesforce y se ocupan tantos asientos como haya comprado el usuario.</li> </ol>
Flujos Alternativos:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>11. El usuario ingresa los datos faltantes. Cliquea nuevamente en <i>Pay Now</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>10.1. Faltan datos del usuario o su tarjeta. El sistema notifica los errores al usuario.</li> <li>12. Se sigue con el paso 8 del flujo básico.</li> </ol>
Excepciones:	Falla la comunicación. No se guardan los cambios.
Prioridad:	Alta
Observaciones:	Este flujo también puede desarrollarse sin estar logueado.

A.2.1. Tabla de caso de uso CU-001

<b>ID de Caso de Uso:</b>	<b>CU-002</b>
<b>Nombre:</b>	<b>Alta de un registro a un evento con discount code</b>
Fecha:	28/09/2015
Actores:	Usuario logueado y usuario no logueado
Descripción:	Este caso de uso describe la inscripción de un usuario a un evento perteneciente a la empresa con discount code.
Precondiciones:	<ul style="list-style-type: none"> <li>● Ingresar al sitio</li> <li>● Tener eventos disponibles en el sitio que sean en el futuro</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>● El usuario se registra exitosamente a un evento</li> <li>● Se ocupa la cantidad de asientos del evento como haya comprado</li> <li>● Se desactiva el descuento usado.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona un evento del directorio de eventos.</li> <li>3. El usuario cliquea el botón <i>Get Tickets</i> para registrarse al evento.</li> <li>5. El usuario ingresa un código de descuento y hace clic en <i>Apply Code</i>.</li> <li>7. El usuario selecciona los tickets que quiere y hace clic en <i>Checkout</i>.</li> <li>9. El usuario se loguea.</li> <li>11. El usuario llena los campos del formulario que faltan y hace clic en <i>Pay Now</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. Se muestran los datos del evento seleccionado.</li> <li>4. Se despliega un modal para que el usuario ingrese un código de descuento y para que seleccione los tipos y cantidad de tickets que quiere.</li> <li>6. El sistema le comunica que su código es válido.</li> <li>8. Se muestra el formulario de compra con los datos de la compra que va a efectuar junto con el descuento seleccionado.</li> <li>10. El sistema lo loguea exitosamente y popula los datos del usuario que tenga guardados.</li> <li>12. Se le muestra un mensaje al usuario de que el registro se hizo correctamente y se le envía un mail con los datos del evento y el pago y un código QR para presentar en el evento. Se envían los datos de la compra a Salesforce y el descuento se desactiva. Se restan tantos asientos como haya comprado el usuario.</li> </ol>

Flujos Alternativos:	<p><b>Usuario:</b></p> <p>7. El usuario sigue con la compra sin descuento desde el punto 7 del flujo básico.</p> <p>13. El usuario ingresa los datos que faltan. Cliquea nuevamente en <i>Pay Now</i>.</p> <p><b>Sistema:</b></p> <p>6.1. El usuario es notificado que el código no es válido. Faltan datos del usuario o su tarjeta. El sistema notifica los errores al usuario.</p> <p>12.1. Faltan datos del usuario o su tarjeta. El sistema notifica los errores al usuario.</p> <p>14. Se sigue con el punto 12 del flujo básico.</p>
Excepciones:	Falla la comunicación. No se guardan los cambios.
Prioridad:	Alta
Observaciones:	Este flujo también puede desarrollarse sin estar logueado.

A.2.2. Tabla de caso de uso CU-002

ID de Caso de Uso:	CU-003
Nombre:	<b>Alta de un evento guardado</b>
Fecha:	28/09/2015
Actores:	Usuario registrado
Descripción:	Este caso de uso describe cómo un usuario registrado y logueado guarda un evento en su lista de eventos guardados.
Precondiciones:	<ul style="list-style-type: none"> <li>• El usuario debe haberse logueado</li> <li>• Debe haber al menos un evento en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se guarda exitosamente un evento</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona un evento del directorio de eventos de la empresa.</li> <li>3. El usuario hace clic en Save.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra los detalles del evento seleccionado.</li> <li>4. El sistema le notifica al usuario que el evento ha sido guardado en sus eventos guardados.</li> </ol>

Flujos Alternativos:	
Excepciones:	Falla la comunicación. No se guardan los cambios.
Prioridad:	Alta
Observaciones:	

A.2.3. Tabla de caso de uso CU-003

<b>ID de Caso de Uso:</b>	<b>CU-004</b>
<b>Nombre:</b>	<b>Baja de un evento guardado</b>
Fecha:	28/09/2015
Actores:	Usuario registrado
Descripción:	Este caso de uso describe cómo un usuario que está registrado al sitio y logueado da de baja un evento guardado.
Precondiciones:	<ul style="list-style-type: none"> <li>• Ingresar al sitio</li> <li>• Utilizar un usuario logueado</li> <li>• El usuario debe de tener un evento guardado</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se da de baja exitosamente a un evento guardado.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ingresa a la sección <i>My Account</i> y hace clic en los eventos guardados.</li> <li>3. Selecciona el evento que quiere eliminar de sus eventos guardados.</li> <li>5. Hace clic en el ícono de la cruz en ese evento.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. Se muestran todos los eventos que guardó el usuario.</li> <li>4. Se muestran los detalles del evento seleccionado.</li> <li>6. Se elimina el evento de la lista de eventos guardados.</li> </ol>
Flujos Alternativos:	
Excepciones:	Falla la comunicación. No se guardan los cambios.
Prioridad:	Alta
Observaciones:	

A.2.4. Tabla de caso de uso CU-004

<b>ID de Caso de Uso:</b>	<b>CU-005</b>
<b>Nombre:</b>	<b>Listado de eventos de la empresa</b>
Fecha:	28/09/2015
Actores:	Usuario registrado y usuario no registrado
Descripción:	Este evento describe cómo se accede al listado de eventos que tiene la empresa, tanto para un usuario logueado como para uno no logueado.
Precondiciones:	<ul style="list-style-type: none"> <li>• Existir eventos en la base de datos para esa compañía</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se listan exitosamente todos los eventos que tiene la empresa.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ingresa al sitio web de RSVPDay de la empresa.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra un listado con todos los eventos que posee.</li> </ol>
Flujos Alternativos:	
Excepciones:	
Prioridad:	Alta
Observaciones:	

A.2.5. Tabla de caso de uso CU-005

<b>ID de Caso de Uso:</b>	<b>CU-006</b>
<b>Nombre:</b>	<b>Filtrar eventos en el listado de eventos de la empresa</b>
Fecha:	28/09/2015
Actores:	Usuario registrado y usuario no registrado
Descripción:	

Precondiciones:	<ul style="list-style-type: none"> <li>• Tener eventos en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• El evento filtrado se muestra</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ingresa al sitio web de RSVPDay de la empresa.</li> <li>3. El usuario filtra un evento por ubicación.</li> <li>5. El usuario filtra nuevamente el evento pero esta vez por período de tiempo.</li> <li>7. El usuario filtra nuevamente por categoría.</li> <li>9. El usuario selecciona el evento deseado.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra un listado con todos los eventos que posee.</li> <li>4. El sistema va filtrando el listado de eventos de acuerdo al filtro que agregó el usuario.</li> <li>6. El sistema agrega otro filtro al listado de acuerdo a ese período.</li> <li>8. El sistema agrega otro filtro de acuerdo a esa categoría.</li> <li>10. El sistema le muestra la información sobre el evento que seleccionó.</li> </ol>
Flujos Alternativos:	4.1., 6.1. y 8.1. No existen eventos con tales características, la lista queda vacía.
Excepciones:	
Prioridad:	Media
Observaciones:	

A.2.6. Tabla de caso de uso CU-006

ID de Caso de Uso:	CU-007
Nombre:	<b>Ordenar eventos en el listado de eventos de la empresa</b>
Fecha:	29/09/2015
Actores:	Usuario registrado y usuario no registrado
Descripción:	Este caso de uso describe el proceso para ordenar los eventos en el directorio de eventos de la empresa sin importar si el usuario está logueado o no.

Precondiciones:	<ul style="list-style-type: none"> <li>• Haber eventos en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• El listado de eventos se ordena de la forma esperada para el usuario.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ingresa al sitio web de RSVPDay de la empresa.</li> <li>3. El usuario selecciona ordenar la lista por orden alfabético ascendente según el nombre del evento.</li> <li>5. El usuario selecciona ordenar la lista por orden alfabético descendente según el nombre del evento.</li> <li>7. El usuario selecciona ordenar los eventos por orden ascendente según el precio del evento.</li> <li>9. El usuario selecciona ordenar los eventos por orden descendente según el precio del evento.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra un listado de todos los eventos que posee.</li> <li>4. El sistema ordena los eventos en la lista en orden alfabético ascendente según el nombre.</li> <li>6. El sistema ordena los eventos en la lista en orden alfabético descendente según el nombre.</li> <li>8. El sistema ordena los eventos en la lista en orden ascendente según el precio.</li> <li>10. El sistema ordena los eventos en la lista en orden descendente según el precio.</li> </ol>
Flujos Alternativos:	
Excepciones:	
Prioridad:	Media
Observaciones:	

A.2.7. Tabla de caso de uso CU-007

ID de Caso de Uso:	<b>CU-008</b>
Nombre:	<b>Buscar un evento en el sitio</b>
Fecha:	29/09/2015
Actores:	Usuario registrado y usuario no registrado

Descripción:	Este caso de uso explica el proceso de búsqueda de un evento determinado en el listado de eventos de la empresa.
Precondiciones:	<ul style="list-style-type: none"> <li>• Haber al menos un evento en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se encuentra el evento ingresado por el usuario dentro del directorio de eventos de la empresa</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario ingresa al sitio web de RSVPDay de la empresa.</li> <li>3. El usuario comienza a ingresar el nombre del evento que está buscando en la caja de búsqueda del listado de eventos.</li> <li>5. El usuario selecciona una de las sugerencias que le dio el sistema.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra un listado con todos los eventos que posee.</li> <li>4. El sistema va dándole sugerencias al usuario de acuerdo con el nombre que va ingresando y va filtrando el listado de eventos.</li> <li>6. Se muestra el detalle del evento seleccionado por el usuario.</li> </ol>
Flujos Alternativos:	<p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>4.1. No hay ningún registro entre los eventos que contenga el nombre que está ingresando el usuario.</li> </ol>
Excepciones:	
Prioridad:	Alta
Observaciones:	Este caso de uso se puede realizar desde cualquiera página del sitio.

A.2.8. Tabla de caso de uso CU-008

ID de Caso de Uso:	<b>CU-009</b>
Nombre:	<b>Login al sistema</b>
Fecha:	02/10/2015
Actores:	Usuario registrado

Descripción:	Este caso de uso describe el proceso para los usuarios que quieran loguearse.
Precondiciones:	
Poscondiciones:	<ul style="list-style-type: none"> <li>El usuario accede a su cuenta.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>El usuario accede al sitio y hace clic en <i>Login or Join</i>.</li> <li>El usuario selecciona Facebook.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>Se muestran las opciones para loguearse con usuario y contraseña o mediante redes sociales.</li> <li>El sistema le muestra mensaje de que el login fue exitoso y si no está registrado se lo registra en la base de datos.</li> </ol>
Flujos Alternativos:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>El usuario selecciona Google, Linkedin o Twitter.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>Se continua con flujo del caso base.</li> </ol> <p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>El usuario selecciona ingresar su email y contraseña.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>Se despliega un formulario para que se ingresen la contraseña y el nombre de usuario. Se continua con flujo del caso base.</li> </ol>
Excepciones:	
Prioridad:	Alta
Observaciones:	

A.2.9. Tabla de caso de uso CU-009

ID de Caso de Uso:	CU-010
Nombre:	Registro al sistema
Fecha:	03/02/2016
Actores:	Usuario no registrado
Descripción:	Este caso de uso describe el proceso para los usuarios que quieran registrarse.

Precondiciones:	
Poscondiciones:	<ul style="list-style-type: none"> <li>• El usuario queda registrado y accede a su cuenta.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario accede al sitio y hace clic en <i>Login or Join</i>.</li> <li>3. El usuario selecciona <i>Create Account</i>.</li> <li>5. El usuario ingresa ambos datos y hace clic sobre <i>Create Account</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. Se muestran las opciones para loguearse con usuario y contraseña o mediante redes sociales.</li> <li>4. El sistema le muestra mensaje dos cajas de texto para su email y contraseña.</li> <li>6. El sistema lo registra y lo loguea y le envía un email de confirmación a su casilla.</li> </ol>
Flujos Alternativos:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>7. El usuario ingresa datos válidos y se continúa con el flujo básico.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>6.1. El sistema le comunica que su contraseña o nombre de usuario son inválidos.</li> </ol>
Excepciones:	
Prioridad:	Alta
Observaciones:	

A.2.10. Tabla de caso de uso CU-010

ID de Caso de Uso:	CU-011
Nombre:	<b>Visualizar detalle de evento</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y usuario no registrado
Descripción:	Este caso de uso describe el proceso para visualizar el detalle de un evento seleccionado.
Precondiciones:	<ul style="list-style-type: none"> <li>• Tener al menos un evento en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• El usuario ve el contenido de un evento.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p>

	<p>1. El usuario se dirige al sitio.</p> <p>3. El usuario selecciona el evento a visualizar.</p> <p><b>Sistema:</b></p> <p>2. El sistema muestra en pantalla todos los eventos disponibles.</p> <p>4. El sistema muestra los detalles del evento.</p>
Flujos Alternativos:	
Excepciones:	
Prioridad:	Alta
Observaciones:	

A.2.11. Tabla de caso de uso CU-011

<b>ID de Caso de Uso:</b>	<b>CU-012</b>
<b>Nombre:</b>	<b>Agregar evento a calendario</b>
Fecha:	06/10/2015
Actores:	Usuario registrado
Descripción:	Este caso de uso describe el proceso para agregar evento a calendario del usuario
Precondiciones:	Estar logueado
Poscondiciones:	<ul style="list-style-type: none"> <li>• El evento es agregado al calendario del usuario.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <p>1. El usuario selecciona el evento del cual quiere visualizar los detalles.</p> <p>3. En el detalle del evento el usuario selecciona <i>Add to Calendar</i>.</p> <p><b>Sistema:</b></p> <p>2. Se muestran los datos del evento.</p> <p>4. El sistema agrega al calendario el evento para que el usuario pueda acceder a él luego.</p>
Flujos Alternativos:	No existen flujos alternativos
Excepciones:	
Prioridad:	Media

Observaciones:	
----------------	--

A.2.12. Tabla de caso de uso CU-012

ID de Caso de Uso:	CU-013
Nombre:	<b>Imprimir evento</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y usuario no registrado
Descripción:	Este caso de uso describe el proceso para generar un pdf con la información de un evento.
Precondiciones:	<ul style="list-style-type: none"> <li>• Que haya al menos un evento en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se exporta un pdf con la información.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona el evento del cual quiere ver el detalle.</li> <li>3. Hace clic en el botón de <i>Get PDF</i></li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. Se muestra el evento seleccionado.</li> <li>4. El sistema exporta un pdf con los datos de ese evento.</li> </ol>
Flujos Alternativos:	
Excepciones:	
Prioridad:	Media
Observaciones:	

A.2.13. Tabla de caso de uso CU-013

ID de Caso de Uso:	CU-014
Nombre:	<b>Imprimir detalle de factura</b>
Fecha:	06/10/2015

Actores:	Usuario registrado y usuario no registrado
Descripción:	Este caso de uso describe el proceso para generar un pdf con los detalles de pago de inscripción a evento.
Precondiciones:	<ul style="list-style-type: none"> <li>• Debe haber al menos un evento en la base de datos</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se exporta un pdf con la información de pago.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario hace clic en <i>Get Tickets</i> para registrarse a un evento y completa el modal con los tickets que quiere.</li> <li>3. El usuario completa el formulario de pago y hace clic en Pay Now.</li> <li>5. El usuario hace clic en el botón de imprimir el detalle de factura</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra el formulario de pago.</li> <li>4. El sistema efectúa el pago con stripe.</li> <li>6. El sistema exporta un pdf con los datos de pago.</li> </ol>
Flujos Alternativos:	
Excepciones:	
Prioridad:	Alta
Observaciones:	

A.2.14. Tabla de caso de uso CU-014

ID de Caso de Uso:	CU-015
Nombre:	<b>Administrar datos de cuenta de usuario</b>
Fecha:	06/10/2015
Actores:	Usuario registrado
Descripción:	Este caso de uso describe el proceso para administrar los datos personales de una cuenta.
Precondiciones:	<ul style="list-style-type: none"> <li>• El usuario debe estar registrado y logueado al sistema.</li> </ul>
Poscondiciones:	<ul style="list-style-type: none"> <li>• Se realizan cambios en la información de la cuenta.</li> </ul>
Flujo Básico:	<p><b>Usuario:</b></p>

	<ol style="list-style-type: none"> <li>1. El usuario selecciona el panel de administración de datos <i>My Account</i>.</li> <li>3. Completa el formulario de datos personales</li> <li>5. Hace los cambios deseados y hace clic en guardar.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema le muestra un formulario para llenar con sus datos.</li> <li>4. Se guardan los datos modificados en la base de datos</li> </ol>
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	Problemas de comunicación. No se logran guardar los datos.
Prioridad:	Media
Observaciones:	

A.2.15. Tabla de caso de uso CU-015

ID de Caso de Uso:	CU-016
Nombre:	<b>Administrar listado de eventos guardados del usuario</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y logueado
Descripción:	Este caso de uso describe el proceso para administrar los listados de eventos guardados del usuario.
Precondiciones:	<ul style="list-style-type: none"> <li>● El usuario debe estar registrado y logueado al sistema.</li> <li>● Debe haber eventos en la base de datos.</li> <li>● El usuario debe haber guardado alguno de esos eventos.</li> </ul>
Poscondiciones:	
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario se loguea al sitio y selecciona <i>My Account</i>.</li> <li>3. El usuario hace clic en <i>Saved Events</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra sus datos personales (si los hubiera).</li> <li>4. El sistema le muestra los eventos que tiene guardados.</li> </ol>
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	

Prioridad:	Media
Observaciones:	

A.2.16. Tabla de caso de uso CU-016

ID de Caso de Uso:	CU-017
Nombre:	<b>Administrar listado de eventos a los que se registró el usuario</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y logueado
Descripción:	Este caso de uso describe el proceso para administrar los listados de eventos a los que se registró el usuario.
Precondiciones:	<ul style="list-style-type: none"> <li>● El usuario debe estar registrado y logueado al sistema.</li> <li>● Debe haber eventos en la base de datos.</li> <li>● El usuario debe haberse registrado a alguno de esos eventos.</li> </ul>
Poscondiciones:	
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario se loguea al sitio y selecciona <i>My Account</i>.</li> <li>3. El usuario hace clic en <i>Registered Events</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra sus datos personales (si los hubiera).</li> <li>4. El sistema le muestra los eventos a los que se registró.</li> </ol>
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	
Prioridad:	Media
Observaciones:	

A.2.17. Tabla de caso de uso CU-017

ID de Caso de Uso:	CU-018
--------------------	--------

<b>Nombre:</b>	<b>Ver listado de eventos pasados a los que se registró el usuario</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y logueado
Descripción:	Este caso de uso describe el proceso para acceder al listado de eventos que ya pasaron y a los que se registró el usuario.
Precondiciones:	<ul style="list-style-type: none"> <li>● El usuario debe estar registrado y logueado al sistema.</li> <li>● Debe haber eventos en la base de datos.</li> <li>● El usuario debe haberse registrado a alguno de esos eventos y estos ya deben haber pasado.</li> </ul>
Poscondiciones:	
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>2. El usuario se loguea al sitio y selecciona <i>My Account</i>.</li> <li>3. El usuario hace clic en <i>Past Events</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra sus datos personales (si los hubiera).</li> <li>4. El sistema le muestra los eventos a los que se ha registrado y ya pasaron.</li> </ol>
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	
Prioridad:	Media
Observaciones:	

A.2.18. Tabla de caso de uso CU-018

<b>ID de Caso de Uso:</b>	<b>CU-019</b>
<b>Nombre:</b>	<b>Contactar organizadores de un evento</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y no registrado.

Descripción:	Este caso de uso describe el proceso para contactar a los organizadores de un evento.
Precondiciones:	<ul style="list-style-type: none"> <li>• Debe haber al menos un evento en la base de datos</li> </ul>
Poscondiciones:	Se envía mail a organizadores del evento.
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona el evento desde el directorio de eventos o su lista de eventos.</li> <li>3. El usuario hace clic en <i>Contact Organizer</i>.</li> <li>5. El usuario escribe el email y hace clic en <i>Send</i>.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra el detalle del evento.</li> <li>4. El sistema le muestra un pop up con la información de contacto y un formulario para escribir la consulta.</li> <li>6. El sistema le envía el email al organizador.</li> </ol>
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	
Prioridad:	Media
Observaciones:	

A.2.19. Tabla de caso de uso CU-019

ID de Caso de Uso:	<b>CU-020</b>
Nombre:	<b>Compartir evento en redes sociales</b>
Fecha:	06/10/2015
Actores:	Usuario registrado y no registrado
Descripción:	Este caso de uso describe el proceso para compartir un evento mediante redes sociales.
Precondiciones:	
Poscondiciones:	Se comparte en el muro de la red social elegida.
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario selecciona el evento desde el directorio de eventos o su lista de eventos.</li> <li>3. Hace clic en el botón de la red social de su elección para compartir el evento.</li> </ol>

	<b>Sistema:</b> 2. El sistema muestra el detalle del evento seleccionado. 4. El sistema publica el evento en el muro del usuario de la red social escogida.
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	Falla en la comunicación con los sitios de las redes sociales. No se puede compartir el evento.
Prioridad:	Media
Observaciones:	

A.2.20. Tabla de caso de uso CU-020

<b>ID de Caso de Uso:</b>	CU-021
<b>Nombre:</b>	<b>Baja de un registro a un evento</b>
Fecha:	06/10/2015
Actores:	Usuario registrado
Descripción:	Este caso de uso describe el proceso para dar de baja un evento al que el usuario se registró
Precondiciones:	<ul style="list-style-type: none"> <li>• Tener al menos hay un evento guardado en la base de datos</li> <li>• Tener un usuario registrado</li> <li>• Que el usuario registrado esté registrado al evento</li> </ul>
Poscondiciones:	El usuario elimina de manera exitosa el evento de los eventos a los que está registrado
Flujo Básico:	<p><b>Usuario:</b></p> <ol style="list-style-type: none"> <li>1. El usuario se dirige a su cuenta (<i>My Account</i>), hace clic en <i>Registered Events</i>.</li> <li>3. El usuario hace clic en <i>Unregister</i> sobre el evento.</li> <li>5. El usuario hace clic en Yes.</li> </ol> <p><b>Sistema:</b></p> <ol style="list-style-type: none"> <li>2. El sistema muestra el evento al que está registrado.</li> <li>4. El sistema le muestra un modal pidiéndole la confirmación de si quiere eliminar el evento.</li> <li>6. El sistema elimina el evento de los <i>registered events</i>, anula la factura de stripe si era pago y cancela la orden en</li> </ol>

	Salesforce.
Flujos Alternativos:	No hay flujos alternativos
Excepciones:	Falla en la comunicación con Meteor o Salesforce. No se puede completar la acción.
Prioridad:	Media
Observaciones:	

A.2.21. Tabla de caso de uso CU-021

### A.3. Mockups para las Páginas

Los siguientes son los mockups provistos por el cliente para la creación del markup de cada una de las páginas. Hay que tener en cuenta que los mismos no representan exactamente la visión actual del cliente ya que también tuvimos recomendaciones de obviar y agregar determinadas partes a lo largo del proceso de desarrollo, por lo tanto, son aproximaciones.

### A.3.1. Directorio de Eventos (cuadrícula)

## Roudfusion EVENTS

All Dates ▾ All Types ▾ All Categories ▾ All Prices ▾ North America  Log in or Join

Search event:

10 per page ▾



**GreenPoint Rated New Home**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Dana Point, CA 92629 | \$399\*



**SUCCESS with Existing Homes Training**  
This 1-day course covers the policies, processes and program requirements unique to the New Home...

Oakland, CA | \$150\*



**Big Bash 2015**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Oakland, CA | \$299\*



**GreenPoint Rated New Home**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Dana Point, CA 92629 | \$150\*



**TERRA Alive Post Convention Tour**  
This 1-day course covers the policies, processes and program requirements unique to the New Home...

Oakland, CA | Free



**Accelerate Your Life Toronto**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Toronto, ON | \$20\*



**NewCo Los Angeles**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Los Angeles, CA | \$150\*



**Mustachio Taste**  
Mustachio is an institution. Located at the center of the lower level of the St. Lawrence Market since 1995...

Halifax, NS | Free



**Change the way work gets done**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Toronto, ON | \$20\*



**GreenPoint Rated New Home**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Dana Point, CA 92629 | \$399\*



**SUCCESS with Existing Homes Training**  
This 1-day course covers the policies, processes and program requirements unique to the New Home...

Oakland, CA | \$150\*



**Big Bash 2015**  
This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.

Oakland, CA | \$299\*

1 2 3 4 5 ... 10 →

164

### A.3.1. Imagen de mockup de directorio de eventos (cuadrícula)

### A.3.2. Directorio de Eventos (lista)

**Roudfusion EVENTS**

All Dates ▾ All Types ▾ All Categories ▾ All Prices ▾ North America Log in or Join

Search event: 10 per page ▾

 <p>#gpr #california Nov 2, 2015, 3:00 PM</p> <p><b>GreenPoint Rated New Home</b> This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.</p> <p>Dana Point, CA 92629  \$399*</p>	 <p>#miscellaneous Nov 28, 2015, 1:00 PM</p> <p><b>SUCCESS with Existing Homes Training</b> This 1-day course covers the policies, processes and program requirements unique to the New Home...</p> <p>Oakland, CA  \$150*</p>	 <p>#miscellaneous #party Nov 2, 2015, 3:00 PM</p> <p><b>Big Bash 2015</b> This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.</p> <p>Oakland, CA  \$299*</p>	 <p>#work #california Nov 8, 2015, 1:00 PM</p> <p><b>GreenPoint Rated New Home</b> This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.</p> <p>Oakland, CA  \$150*</p>	 <p>#miscellaneous Nov 28, 2015, 1:00 PM</p> <p><b>SUCCESS with Existing Homes Training</b> This 1-day course covers the policies, processes and program requirements unique to the New Home...</p> <p>Oakland, CA  \$150*</p>	 <p>#enviroment Nov 28, 2015, 1:00 PM</p> <p><b>TERRA Alive Post Convention Tour</b> This 1-day course covers the policies, processes and program requirements unique to the New Home...</p> <p>Oakland, CA  Free</p>	 <p>#food #ontario Nov 28, 2015, 1:00 PM</p> <p><b>Accelerate Your Life Toronto</b> This 1-day course covers the policies, processes and program requirements unique to the New Home rating system.</p> <p>Toronto, ON  \$20*</p>
---	--	--	---	--	---	---

### A.3.2. Imagen de mockup de directorio de eventos (lista)

### A.3.3. Detalle de Evento

The screenshot displays the Roudfusion EVENTS website for the 'Big Bash 2015' event. At the top, the title 'Roudfusion EVENTS' is shown above a search bar and a 'Log in or Join' button. The main banner features a night cityscape with fireworks and the text: 'Celebrate 10 BIG years and experience the technology of the future!' with a '#miscellaneous' tag. A calendar icon indicates the event date as NOV 06. To the right, a sidebar shows the location 'Jack London Aquatic Center, 115 Embarcadero, Oakland, CA' with start time '09:00 AM' and end time '10:00 PM'. Buttons for 'SAVE EVENT' and 'BUY A TICKET' are present. Below the banner, the 'OVERVIEW' section contains a brief description of the event's purpose and highlights. The 'ABOUT THE EVENT' section provides more detailed information about the celebration, including food, speakers, and raffle prizes. It also includes a map of the event venue area and contact information. The 'OFFICIAL THE SPEAKERS' section lists three speakers: Jony Ive, Dieter Rams, and Aaron Draplin, each with their name, title, and a small photo. The timeline at the bottom shows the events for DAY 1, DAY 2, and DAY 3.

**ALL EVENTS / Big Bash 2015**

**Roudfusion EVENTS**

Search event

Log in or Join

**Celebrate 10 BIG years and experience the technology of the future!**

#miscellaneous

NOV 06

We'll celebrate 10 years of helping our community thrive, as well as look ahead to the future of the building industry. There will be something for everyone!

**OVERVIEW**

**ABOUT THE EVENT**

We'll celebrate 10 years of helping our community thrive, as well as look ahead to the future of the building industry. There will be something for everyone!

This year's lively indoor-outdoor celebration will be held at the picturesque Jack London Aquatic Center, where you can amble along the Oakland waterfront, nosh on Off the Grid street food, explore a Tiny Home, be inspired by innovative speakers, and watch our exhibitor, Proto.House, create commemorative 3D printed pins for you to take home! Join us to give kudos to this year's Build It Green Recognition Award honorees.

Join us to touch, see and feel new building technologies, eat, drink and network with other movers and shakers in our industry who are making a difference!

In addition, each guest will receive a complimentary raffle ticket and a chance to win exciting prizes, including:

- A Cupcakes street truck at your home or business for one hour
- 1 night stay at Portola Hotel & Spa in Monterey
- Water saving showerheads from Evolve Technologies
- 4 day passes to the California Academy of Sciences
- Bottles of wine from our Wine Sponsors
- Honeywell WiFi Smart Thermostats
- Grand Lake Ace Hardware Gift Certificate
- 1 year exclusive Wine Pass Membership from Priority Wine Pass
- And much more... check the Details tab regularly to see new raffle prizes

**OFFICIAL THE SPEAKERS**

DAY 1 DAY 2 DAY 3

Jony Ive Apple Senior VP 2:00 pm

Dieter Rams Braun Senior Designer 5:00 pm

Aaron Draplin DDC Founder 7:00 pm

Jack London Aquatic Center  
115 Embarcadero  
Oakland, CA

membership@builditgreen.org

Twitter 12 Google+ 0 Facebook 1258

Via Verde Via Gómez Via Espinosa  
Vista Dr E. Vía Verde Vía Gómez  
Coast Hwy. Rivera Beach and Sea Ranch Camino Capitulano  
Carpintero Cepistreto Surfside Inn ZAK Lantern

Google

166

### A.3.3. Imagen de mockup de detalle de evento (parte 1)

#### ★ PLATINUM SPONSORS



Become a sponsor

#### GREEN LEVEL SPONSORS

Central Coast Energy Compliance | Sherwood Designs | Henri Patey | Central Valley Raters Inc. | Longevity Architectural Products  
Krantz Consultants, LLC | BluePoint Planning | Ashby Lumber | McCutcheon Construction

#### A.3.4. Imagen de mockup de detalle de evento (parte 2)

### A.3.4. Registro a un Evento

The screenshot shows a web-based event registration interface. At the top, the header reads "Roudfusion EVENTS". Below the header, the navigation bar includes links for "ALL EVENTS", "Big Bash 2015", and "Register". A search bar and a user profile "Hello, Robin" are also present.

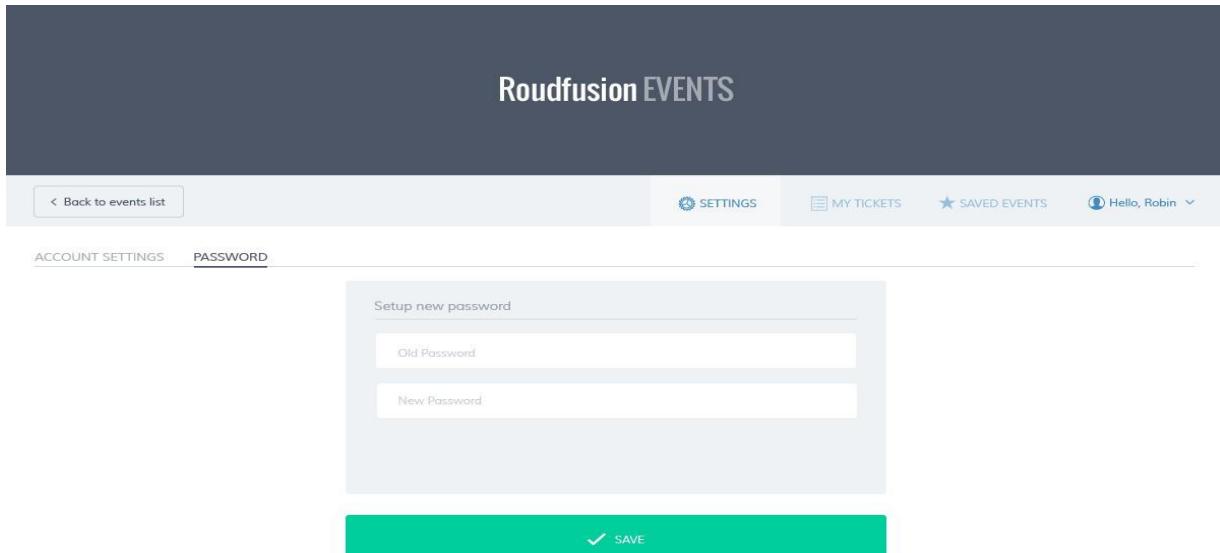
The main content area features a banner with a city skyline at night and the text "Celebrate 10 BIG years and experience the technology of the future!" followed by "#miscellaneous". To the right of the banner, event details are displayed: "Jack London Aquatic Center, 115 Embarcadero, Oakland, CA", "09:00 AM START TIME", "10:00 PM END TIME", and a "SAVE EVENT" button.

The interface is divided into several sections:

- ORDER SUMMARY:** Shows "Celebrate Big Bash 2015", "1 Day Pass", "1 Ticket", and a note that "Sale ends in 30 minutes". The total amount is listed as "USD \$150".
- PAYMENT INFO:** Contains fields for "Payment Info" (Robin Marquez) and "Billing Info" (Card Number, Month, Year, Security Code). The card number field contains "hkjhahkahhkhkjhf" and has a red border, indicating an error or validation failure.
- ATTENDEE INFO:** Fields for "Phone Number", "Email", and "Confirm Email".
- Bottom Buttons:** A large green "PAY NOW" button with a shopping cart icon, and a small note below it stating: "By clicking "Pay Now", I acknowledge that I have read and agree with Build It Green's terms and conditions and privacy policy."

### A.3.5. Imagen de mockup de registro a un evento

### A.3.5. Cambio de Contraseña (My Account)



### A.3.6. Imagen de mockup de cambio de contraseña (My Account)

### A.3.6. Datos Personales (My Account)

The mockup displays the 'Roudfusion EVENTS' interface. At the top, there's a dark header bar with the title 'Roudfusion EVENTS'. Below it is a light-colored navigation bar containing links for 'Back to events list', 'SETTINGS', 'MY TICKETS', 'SAVED EVENTS', and a user profile 'Hello, Robin'. The main content area is divided into two sections: 'ACCOUNT SETTINGS' and 'PASSWORD'. The 'ACCOUNT SETTINGS' section is currently active, showing 'Personal Info' fields for name ('Robin Marquez'), email ('rhnmarquez86@gmail.com'), phone number ('+505 82746485'), and address ('Residencial San Andres 1ra Etapa Casa e16'). It also includes dropdown menus for 'Nicaragua' and 'Managua'. The 'Billing Address' section shows a pre-filled address ('Residencial San Andres 1ra Etapa Casa e16') and city ('Managua'), with dropdown menus for 'Nicaragua' and 'Managua'. A large green 'SAVE' button is located at the bottom right of the form.

### A.3.7. Imagen de mockup de datos personales (My Account)

## A.4. Manual de Usuario

### A.4.1. Introducción

El siguiente manual tiene como objetivo explicar las funcionalidades presentes en la aplicación web RSVPDay y las formas de lograr su mejor uso. La misma consiste en un sitio web (<http://rsvpday-demo.herokuapp.com/>) en el cual se pueden ver eventos y registrarse a los mismos. Es indispensable utilizar http para el login con redes sociales.

La aplicación puede ser utilizada tanto por un usuario registrado como por un usuario no registrado.

Estos eventos, junto con sus datos, pueden mandarse desde Salesforce hacia el sitio de RSVPDay y todos los datos de las compras efectuadas también se envían desde esta aplicación hacia Salesforce.

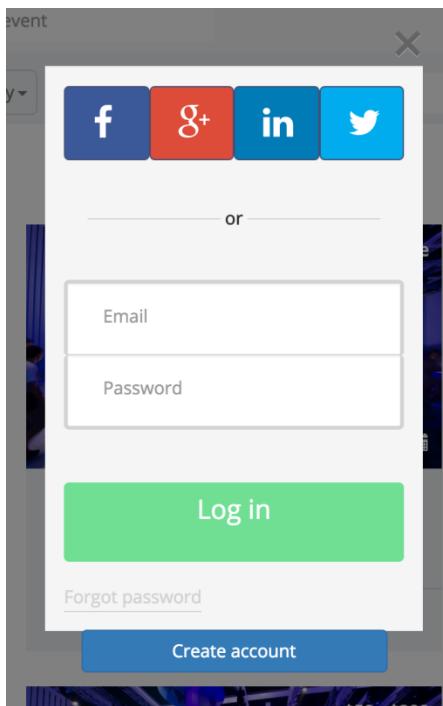
### A.4.2. Login

Para loguearse en el sitio es necesario simplemente ingresar al botón de login presente en el margen superior de la pantalla.



A.4.1. Imagen de botón de login

Al hacer clic se desplegará un formulario en el que se requiere ingresar email y contraseña.



A.4.2. Imagen de modal del login

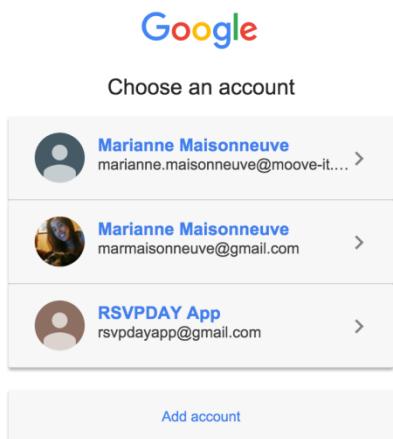
Si la cuenta ya existe con hacer clic en *Log in* se estará logueando al sitio. Si en cambio la cuenta no existe debe hacer clic en *Create Account*. Además está la posibilidad de utilizar las redes sociales para el login.

#### FACEBOOK LOGIN:

A screenshot of a Facebook login interface. At the top, there is a blue bar with the Facebook logo and the text "Facebook". Below the bar, a message says "Inicia sesión para utilizar tu cuenta de Facebook con RSVPday – Public Website Test.". There are two input fields: one for "Correo electrónico o teléfono:" and one for "Contraseña:". Below the password field is a checkbox for "No cerrar sesión". At the bottom right is a link "¿Has olvidado la contraseña?".

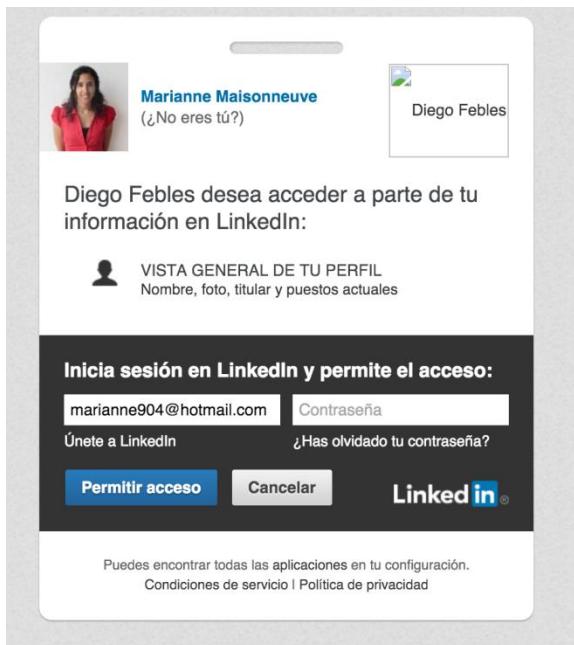
A.4.3. Imagen de login con facebook

## GOOGLE LOGIN:



A.4.4. Imagen de login con google

## LINKEDIN LOGIN:



A.4.5. Imagen de login con linkedin

## TWITTER LOGIN:



[Sign up for Twitter >](#)

**Authorize RSVPday - Public Website to use your account?**

Remember me · [Forgot password?](#)

[Sign In](#) [Cancel](#)

**This application will be able to:**

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

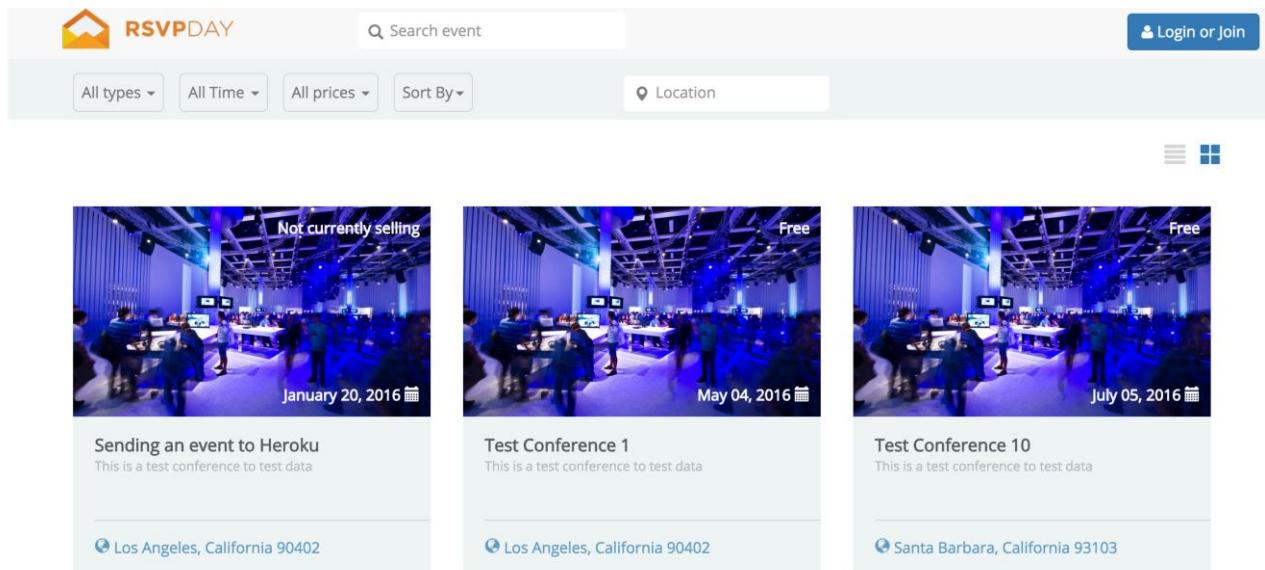
**Will not be able to:**

- Access your direct messages.
- See your Twitter password.

#### A.4.6. Imagen de login con twitter

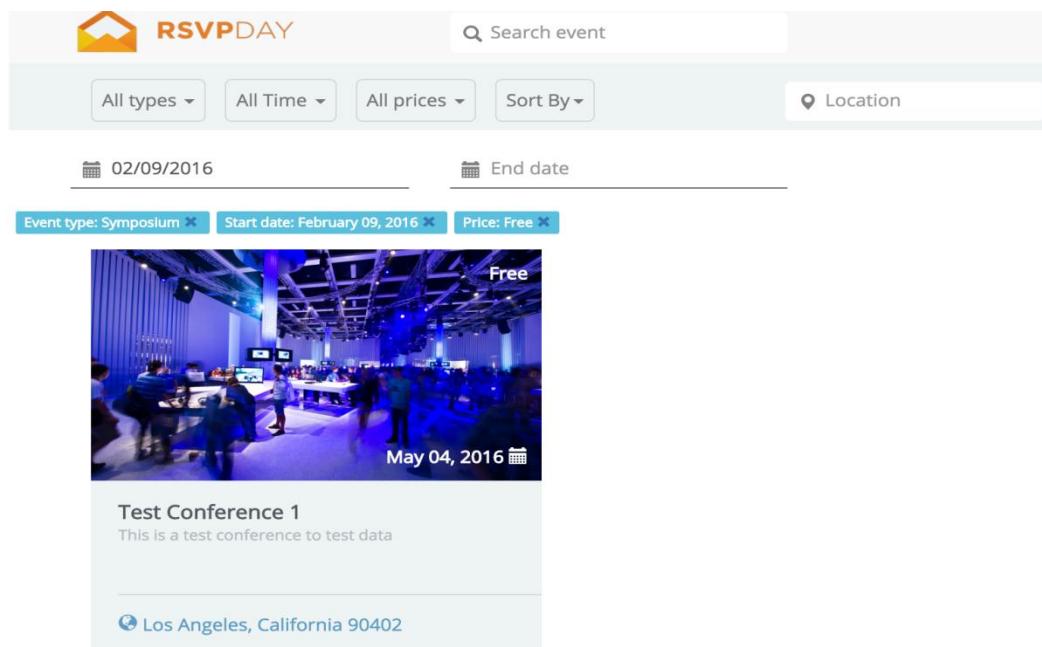
#### A.4.3. Events Directory

Esta página es la que muestra los eventos disponibles. Es la primera página que se ve cuando se ingresa al sitio. Puede ser accedida haciendo clic en el logo de RSVPDay desde cualquier parte del sitio.



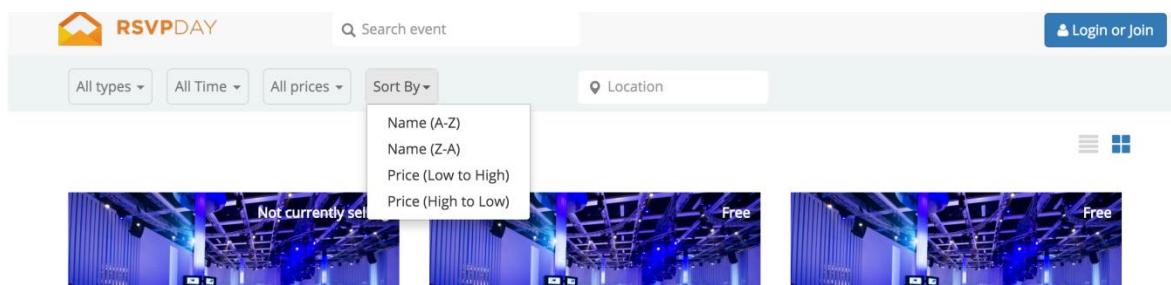
#### A.4.7. Imagen de directorio de eventos

En la misma se muestra la información básica de los eventos como el título, la ubicación, la fecha y una breve descripción. Además, hay una sección de filtrado por tipo, tiempo, precios, ubicación y por evento. La sección de filtrado por evento tiene sugerencias de títulos y al seleccionar uno lleva al usuario a los detalles de ese evento. El resto de los filtros van mostrando aquellos eventos que cumplen con los requisitos y van agregando burbujas con los filtros seleccionados.



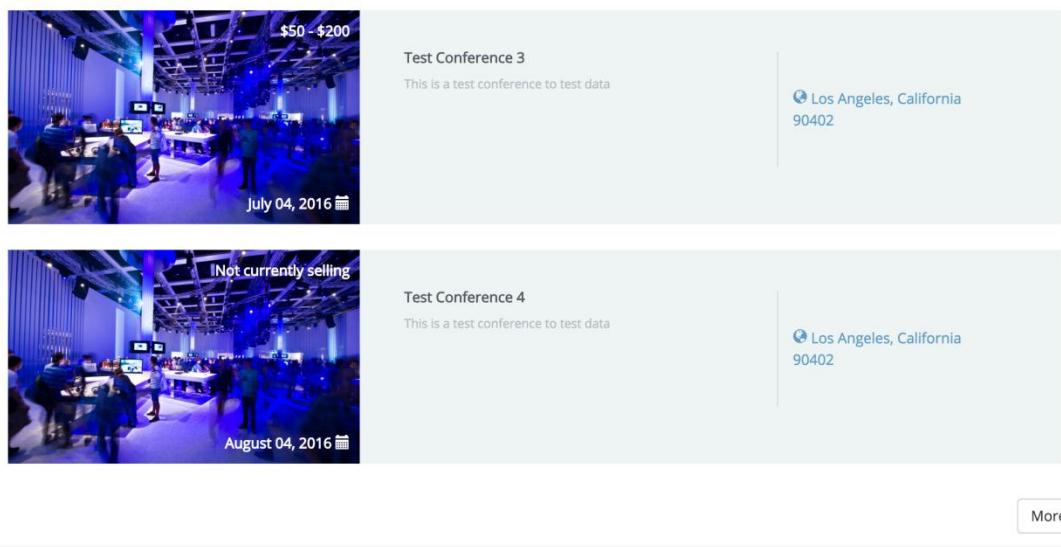
#### A.4.8. Imagen de directorio de eventos filtrado

Al hacer clic en las cruces de las burbujas los filtros se van eliminando. También cuenta con una sección para ordenar por diversos criterios.

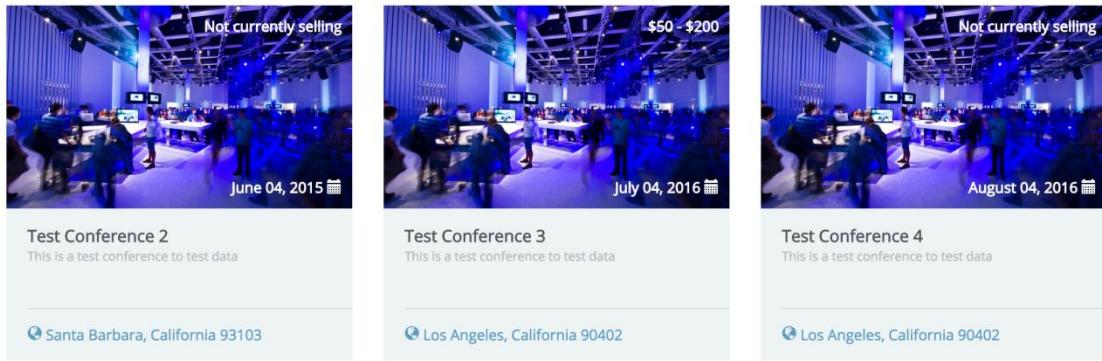


A.4.9. Imagen de directorio de eventos y sort by

Asimismo, tiene dos tipos de vista de los eventos: en lista y en grilla. Para cambiar la forma de visualizar los eventos se debe hacer clic en los correspondientes íconos que se encuentran a continuación de los filtros.



A.4.10. Imagen de directorio de eventos en lista



#### A.4.11. Imagen de directorio de eventos en grilla

Haciendo clic en el botón de *More* se pueden visualizar más eventos.

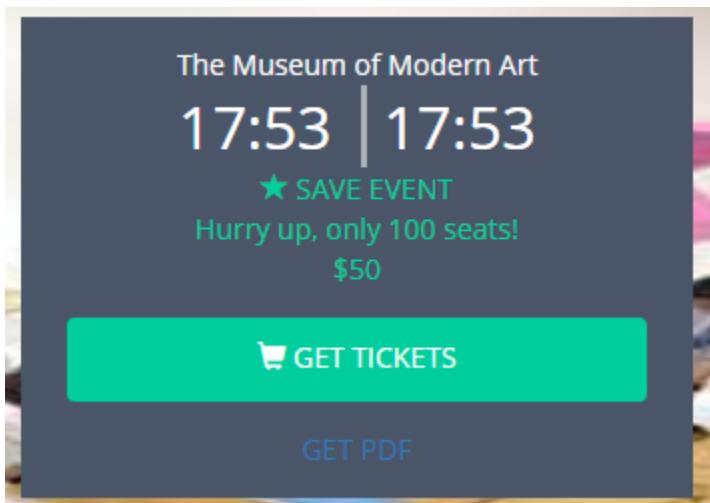
#### A.4.4. Event Detail

El detalle del evento muestra la información del mismo, contiene una sección en la que se muestran las páginas visitadas anteriormente (ej: *ALL EVENTS*) y al hacer clic retorna a la página anterior. Por otro lado, hay un menú para navegar entre las diferentes secciones del evento. Las mismas pueden variar ya que son completamente configurables por los creadores del evento. La presentación de los eventos también puede variar de evento a evento ya que también puede ser configurada cuando se crea el evento en Salesforce.

#### A.4.12. Imagen de detalle de evento

Todos los eventos van a poseer una sección llamada overview que contiene la información esencial del evento como la ubicación (mapa), los horarios, la posibilidad de registrarse al hacer clic en el botón de *GET TICKETS* (si es un evento que ya tuvo lugar no va a dar la posibilidad de registrarse), de guardar un evento que luego se podrá visualizar en los eventos guardados del usuario al hacer clic en *SAVE EVENT* y de obtener un pdf con la información del evento al hacer clic en *GET PDF*.

Si el usuario no está logueado al intentar guardar un evento se le avisará que debe hacerlo.



A.4.13. Imagen de información de evento

También se pueden compartir los eventos mediante los botones de redes sociales a continuación del mapa. Se abrirá una ventana correspondiente a cada red social en la que se podrá agregar información aparte del link al evento.

Where

The Museum of M... [Iniciar sesión](#)

[Ampliar el mapa](#)

Columbus Circle, N.Y.C., N.Y.C.

The Museum of Modern Art

11 W 53rd St  
New York, New York 10019

Organizer

[Contact Organizer](#)

Share

[f 0](#)  [0](#)  [in 0](#)  [g+ 0](#)  [0](#)

A.4.14. Imagen de información de ubicación y organizador de evento y botones para compartir

Sign up >

---

Share a link with your followers

Take a look at this event on RSVP-Day! <https://rsvpday-demo.herokuapp.com/eventDetail/a07G000012jEgd1>

78 [Log in and Tweet](#)

Remember me · [Forgot password?](#)

A.4.15. Imagen de cómo compartir evento por twitter

Al hacer clic en *Contact Organizer* se podrá enviar un mail al organizador del evento.

Contact Organizer

Write your email address here

Write your comments here

Send

**A.4.16. Imagen de formulario para contactar al organizador**

Si el usuario está logueado al guardar un evento se le avisará que el evento fue guardado. Para visualizar el evento debe hacer clic en *MY ACCOUNT*. Asimismo, se visualizará *REMOVE SAVED EVENT* para eliminar ese evento de la lista de eventos guardados.

RSVPDAY

Search event

ALL EVENTS / Test - Salesforce World Tour Atlanta

OVERVIEW AGENDA SESSIONS FAQ SPEAKERS SPONSORS

COBB GALLERIA CENTRE  
06:20 | 18:20  
★ SAVE EVENT  
Hurry up, only 100 seats!  
\$300 - \$900

GET TICKETS GET PDF

OVERVIEW  
ABOUT THE EVENT

The best of Dreamforce is headed your way.

Are you ready for a new kind of customer success? Join us for the all-new Salesforce World Tour in Los Angeles on December 3 and see how to connect with your customers in a whole new way. Access the best speakers, sessions, keynotes, and more - for FREE! Learn firsthand from leading customers how to sell, service, market, and succeed like never before.

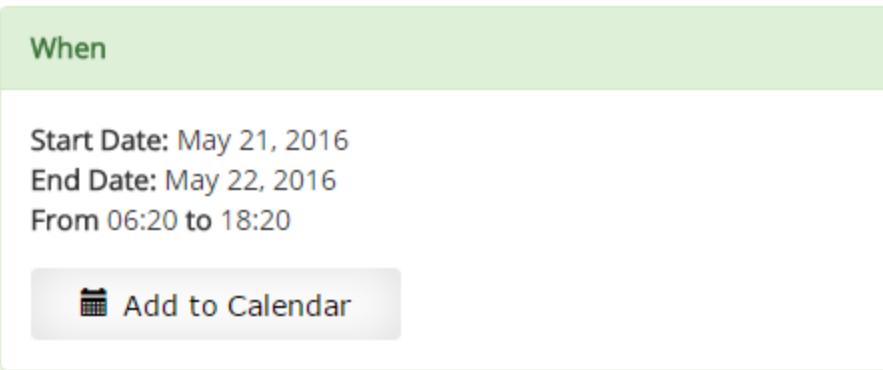
When

Start Date: May 21, 2016  
End Date: May 22, 2016  
From 06:20 to 18:20

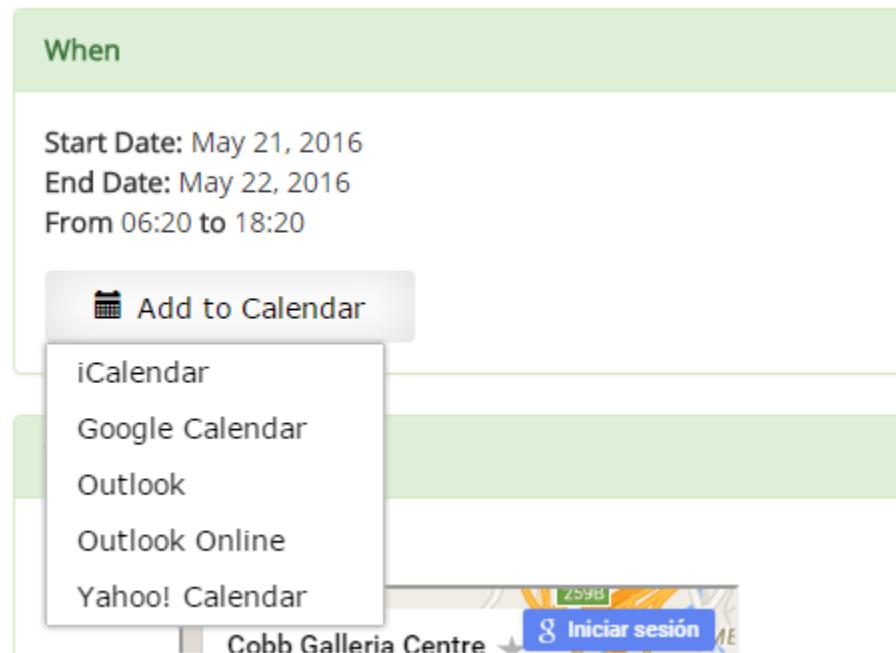
Añadir al Calendario

**A.4.17. Imagen de parte superior de detalle de evento**

Al usuario logueado se le mostrará un botón *Add to Calendar* para agregar ese evento al calendario junto con el tipo de calendario al cual lo quiere agregar.



A.4.18. Imagen de Add to Calendar

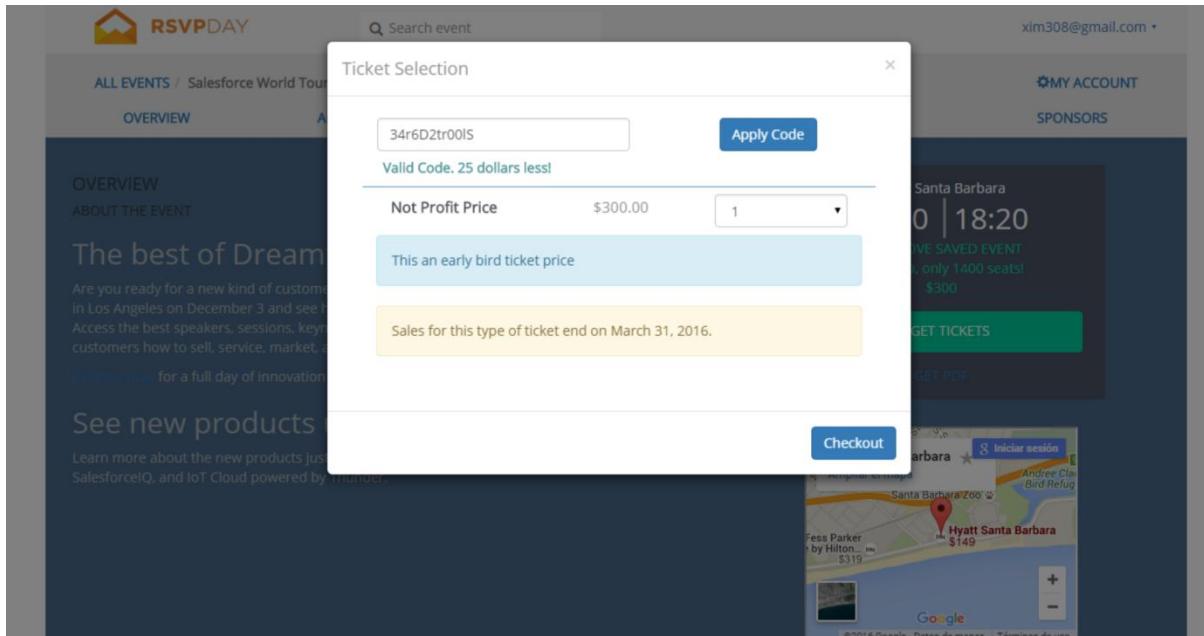


A.4.19. Imagen de Add to Calendar desplegado

#### A.4.5. Event registration

Para registrarse a un evento, se debe hacer clic en el botón de *GET TICKETS*. Al hacerlo se abrirá una sección en la que se debe elegir la cantidad de tickets y aplicar un *promo code*.

A los efectos de corregir la funcionalidad de los *promo codes* y poder usar uno válido se debe ingresar a Salesforce, ir al *event* correspondiente y allí en la sección *Promo Codes* buscar cuál de ellos es válido, si los hay, de lo contrario se pueden crear como explica la sección A.5.7.5. *Promo Codes*.



A.4.20. Imagen de selección de tickets

Luego, se debe completar el formulario de registro con la información de pago (si es un evento pago).

**IMPORTANTE:** Se debe completar el formulario en el tiempo estipulado en el formulario, de lo contrario deberá seleccionar nuevamente la cantidad de tickets.

**Salesforce World Tour Atlanta**

Please complete registration within 27:33 minutes

ORDER SUMMARY			
<b>Salesforce World Tour Atlanta</b>			
Type	Price	Quantity	Subtotal
Not Profit Price	300	1	300
Discount			-25
Total 275 USD			

**ATTENDEE INFO**

Please verify your information below. All fields are required.

Xime
Lasserre
(865) 234-3243
xim308@gmail.com
xim308@gmail.com

**PAYMENT INFO**

1 Payment Info

Visa		
4242424242424242		
February	2017	006

2 Billing Info

123 25th Street	
2nd St	
San Francisco	California
94105	

**When**

Start Date: May 21, 2016  
End Date: May 22, 2016  
From 06:20 to 18:20

**Where**



Hyatt Santa Barbara  
1111 E Cabrillo Blvd  
Santa Barbara, California 93103

**Organizer**

[Contact Organizer](#)

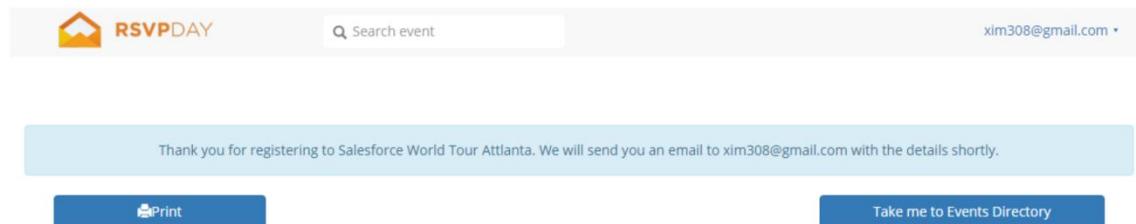
**Share**

[f 0](#) [t 0](#) [in 0](#) [g+ 0](#) [e 0](#)

By clicking "Pay Now" I acknowledge that I have read and agree with the [terms and conditions](#) and [privacy policy](#).

#### A.4.21. Imagen de formulario de registro

Al finalizar el proceso de registro se enviarán dos mails, uno con el código QR al evento y otro con la factura. Además, la aplicación redirige al usuario a una página de agradecimiento en la que se puede obtener un pdf de la factura o dirigirse al directorio de eventos.



#### A.4.22. Imagen de página de agradecimiento

También son enviados los datos de la compra a Salesforce, en donde se crea un objeto *Order* con los datos de la compra. Si hubiera un *promo code*, este se desactiva.

The screenshot shows the Salesforce interface for managing promo codes. The top navigation bar includes links for Home, Events, Speakers, Venues, Sponsors, Ticket Types, Tickets, Orders, Order Items, and Promo Codes. The 'Promo Codes' tab is selected. On the left, a sidebar lists recent items, including a 'Promo' entry. The main content area displays a 'Promo Code Detail' form for a record named 'Promo'. The form fields include:

- Promo Code Name: Active (highlighted in red)
- Amount: \$25.00
- Description: 25 dollars less!
- Expiration Date: 4/15/2016
- Code: 34r6D2tr00IS
- Start Date: 1/27/2016
- Event: Test - Salesforce World Tour Atlanta
- Order: O-25
- Created By: RSVPday Admin
- Last Modified By: RSVPday Admin

A note in red text next to the 'Active' field says: "Este es el campo que se desactiva cuando se usó un promo code para efectuar una compra." Below the form are sections for 'Open Activities' and 'Activity History'.

#### A.4.23. Imagen de un promo code en Salesforce

Y se restan los tickets que se compraron del campo *Quantity* en el objeto *Ticket Type*.

The screenshot shows the RSVPday software interface. In the top navigation bar, there are links for Home, Events, Speakers, Venues, Sponsors, Ticket Types (which is the active tab), Tickets, Orders, Order Items, Promo Codes, and a plus sign. On the right side of the header, there are links for 'RSVPday Admin', 'Setup', 'Help', and 'RSVPday'. Below the header, there is a search bar with placeholder text 'Search...' and a 'Search' button. The main content area has a title 'Ticket Type Not Profit Price'. To the left, there is a sidebar titled 'Recent Items' with a list of ticket types: Not Profit Price, O-43, QI-56, O-58, TKT-000064, TKT-000063, QI-65, O-57, TKT-000062, and TKT-000061. Below this is a 'Recycle Bin' section. The central part of the screen shows a 'Ticket Type Detail' form. The 'Quantity' field is highlighted with a red border. A note in Spanish, 'Este es el campo que se modifica (se le restan tickets) cuando se realiza la compra.', is overlaid on the form next to the quantity field. Other fields in the form include 'Ticket Type Name' (Not Profit Price), 'Description' (This is an early bird ticket price), 'Price' (\$300.00), 'Display Description?' (checked), 'Private Ticket' (unchecked), 'Display Remaining Avail Seat?' (checked), 'Free Ticket?' (unchecked), 'Sales Start Date & Time' (2/1/2016 6:23 AM), 'Sales End Date & Time' (3/31/2016 6:23 AM), 'Event' (Test - Salesforce World Tour Atlanta), and 'Help Text' (This is an early bird ticket price). The 'Created By' field shows 'RSVPday Admin, 1/7/2016 3:41 PM'. The 'Last Modified By' field shows 'RSVPday Admin, 2/13/2016 3:08 PM'. At the bottom of the form are 'Edit', 'Delete', and 'Clone' buttons. Below the form is a section titled 'Open Activities' with buttons for 'New Task' and 'New Event'. A note says 'No records to display'. On the far right, there is a link 'Open Activities Help'.

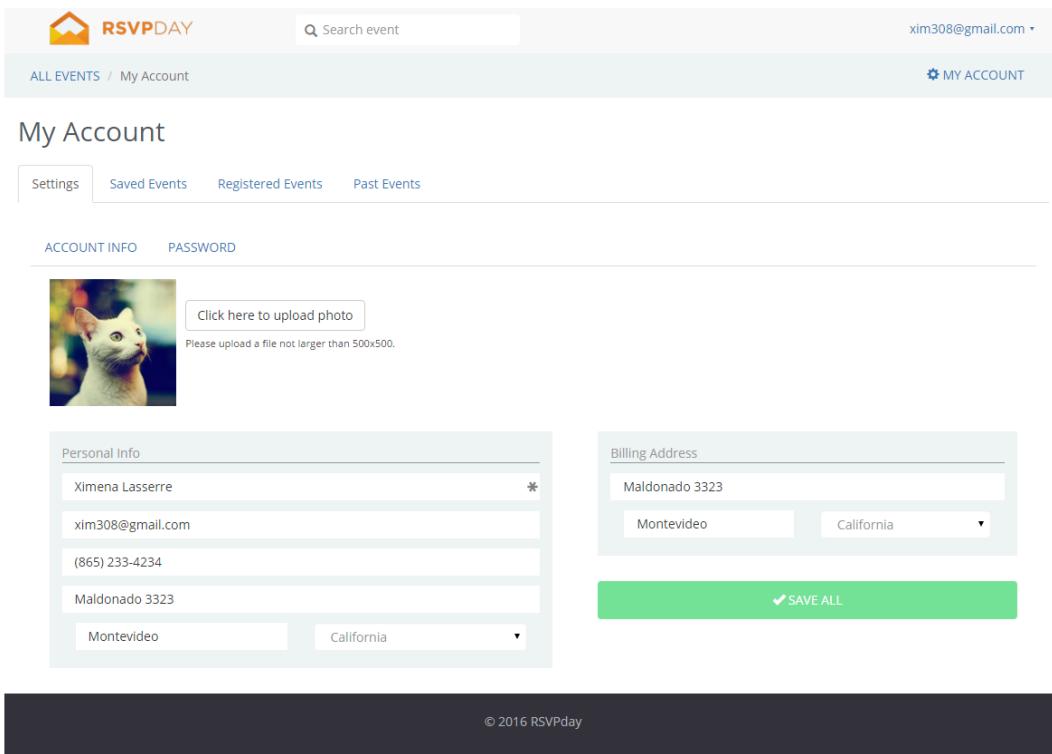
A.4.24. Imagen de un ticket type en Salesforce

#### A.4.6. My Account

Para ingresar a *My Account* primero hay que estar logueado y una vez logueado va a aparecer un link que dice *My Account* en la parte superior derecha de la pantalla.

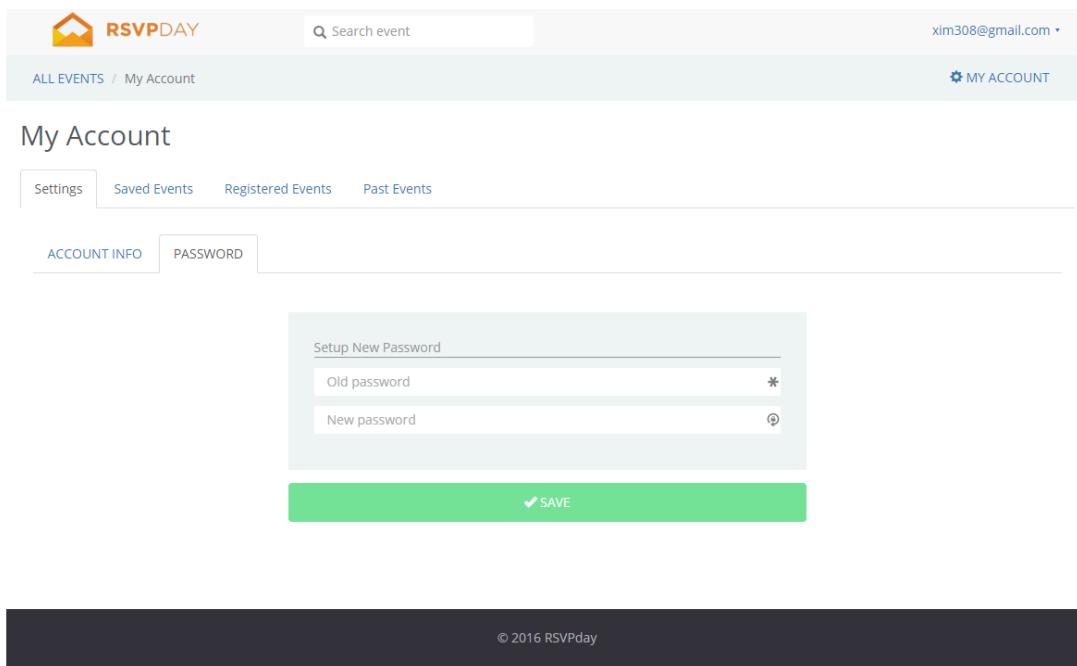
La pantalla principal de *My Account* es la de *Settings*, que está dividida entre *Account Info* y *Password*. Hay otras tres secciones además de *Settings*, estas son *Saved Events* (eventos guardados), *Registered Events* (eventos a los que está registrado el usuario) y *Past Events* (eventos a los que se registró y ya pasaron).

En la sección de *My Account* se van a poder modificar los datos personales del usuario. También se puede subir una foto de perfil del mismo.



#### A.4.25. Imagen de Account Info en My Account

En la sección *Password* se puede cambiar la contraseña del usuario. La misma no puede tener menos de seis caracteres.



#### A.4.26. Imagen de Password en My Account

Para acceder a *Saved Events* hay que hacer clic en la pestaña con ese nombre. Allí se visualizan los eventos que tiene guardados el usuario logueado. Al hacer clic en la cruz roja se puede eliminar como evento guardado, sacándolo de la lista.

The screenshot shows the top navigation bar of the RSVPday website with the logo, a search bar, and a user account dropdown. Below the header, a breadcrumb navigation shows 'ALL EVENTS / My Account'. The main content area is titled 'My Account' and has tabs for 'Settings', 'Saved Events' (which is active), 'Registered Events', and 'Past Events'. Under 'My Saved Events', there is a card for the 'Salesforce World Tour Atlanta' event on May 21, 2016. The card includes a thumbnail image of the event, the event name, a short description, the location ('Santa Barbara, California 93103'), and a red 'X' icon in the bottom right corner, which likely serves as a delete button. At the bottom of the page is a dark footer bar with the text '© 2016 RSVPday'.

A.4.27. Imagen de Saved Events en My Account

Para acceder a *Registered Events* el usuario debe hacer clic en la pestaña del mismo nombre. Allí verá una lista de los eventos a los que está registrado. Al hacer clic en *Unregister*, el usuario podrá eliminar el registro que haya hecho para ese evento y si fuera pago se le hará un reembolso de su dinero.

The screenshot shows the 'My Account' section of the RSVPday website. At the top, there's a navigation bar with the 'RSVPDAY' logo, a search bar, and a user account dropdown. Below the navigation, a breadcrumb trail shows 'ALL EVENTS / My Account'. On the right, there's a 'MY ACCOUNT' button. The main content area is titled 'My Account' and has tabs for 'Settings', 'Saved Events', 'Registered Events', and 'Past Events'. The 'Registered Events' tab is selected. It displays a card for the 'Salesforce World Tour Atlanta' event, which took place on May 21, 2016. The card includes a photo of the event, the name, a short description, the location ('Santa Barbara, California 93103'), and a blue 'Unregister' button. A copyright notice at the bottom of the page reads '© 2016 RSVPday'.

#### A.4.28. Imagen de Registered Events en My Account

Para acceder a *Past Events*, el usuario debe hacer clic en la pestaña del mismo nombre. Allí podrá visualizar todos los eventos a los que se registró y ya pasaron.

The screenshot shows the 'My Account' section of the RSVPday website, similar to the previous one but with the 'Past Events' tab selected. The main content area is titled 'My Account' and has tabs for 'Settings', 'Saved Events', 'Registered Events', and 'Past Events'. The 'Past Events' tab is selected. It displays a card for the 'Test Conference 2' event, which took place on June 04, 2015. The card includes a photo of the event, the name, a short description ('This is a test conference to test data'), the location ('Santa Barbara, California 93103'), and a blue 'Unregister' button. A copyright notice at the bottom of the page reads '© 2016 RSVPday'.

#### A.4.29. Imagen de Past Events en My Account

## A.4.7. Salesforce

Se puede acceder al org de Salesforce (donde se encuentra el lado privado de la aplicación) yendo a <https://login.salesforce.com/>. Allí hay que ingresar nombre de usuario y contraseña.<sup>2</sup> Para enviar datos tales como *Events*, *Speakers*, *Sponsor* o *Venues* al sitio público se deberán seguir los pasos especificados abajo. Para enviar un *Event* con sus *Speakers*, *Sponsors* y *Venue* asociados primero hay que crear estos tres últimos, por eso se empezará por estos y se dejarán los *Events* para lo último.

### A.4.7.1. Enviar Speakers

Para crear un *Speaker* se debe hacer clic en *Speakers* en la barra de navegación. De esta forma se mostrará una lista con los *speakers* que existen en el sistema.

A.4.30. Imagen de lista de Speakers en Salesforce

Luego se debe hacer clic en *New* y se mostrará un formulario para llenar los datos del *speaker*.

<sup>2</sup> Ver punto del Anexo A.8. Acceso a Salesforce para obtener usuario y contraseña.

Speaker Edit  
New Speaker

Last Name: Twain  
First Name: Mark  
Company: Apple  
Job Title: Manager  
Company Name: Apple  
Biography: Mark Twain was born in 1970 in New York.  
Email: marktwain@gmail.com

Save Save & New Cancel

#### A.4.31. Imagen de edición de speaker en Salesforce

Luego de llenar todos los datos con excepción de *Publish* y la *External ID* (este último se populará automáticamente con la id de Meteor cuando se mande el speaker) se debe hacer clic en *Save* y se mostrarán los datos ingresados. En la sección *Notes & Attachments* se debe hacer clic en *Attach File*.

Speaker Detail  
Edit Delete Clone

Last Name: Twain  
First Name: Mark  
Company: Apple  
Job Title: Manager  
Company Name: Apple  
Biography: Mark Twain was born in 1970 in New York.  
Email: marktwain@gmail.com  
Created By: RSVPday Admin, 2/9/2016 5:34 PM  
Last Modified By: RSVPday Admin, 2/9/2016 5:34 PM

Open Activities  
Edit Delete Clone

No records to display

Activity History  
Log a Call Mail Merge Send an Email

No records to display

Event Section - Speaker  
New Event Section - Speaker

No records to display

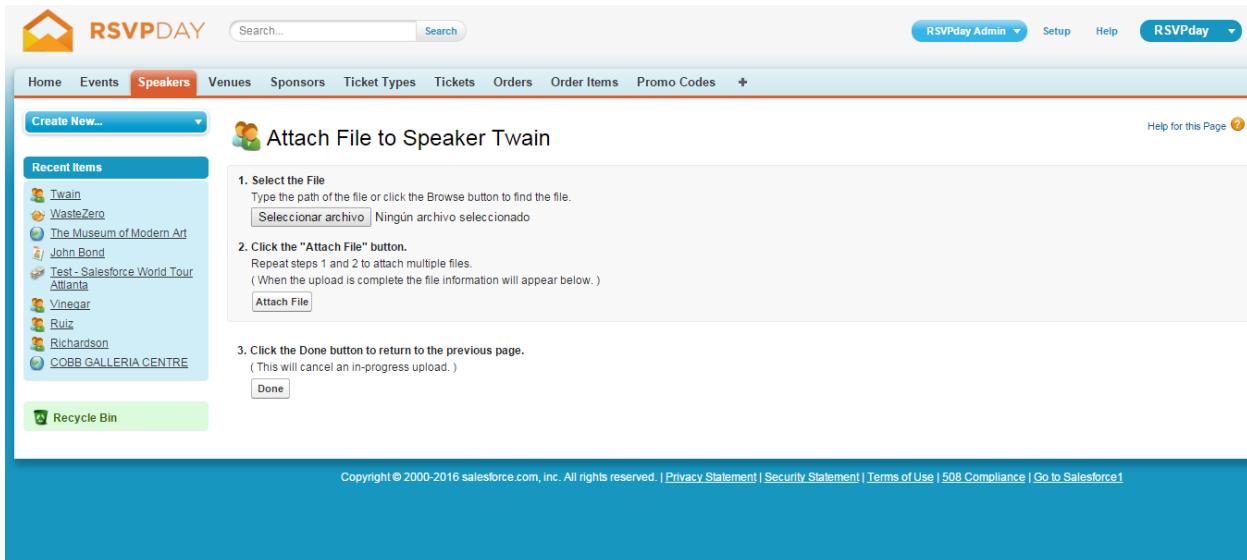
Notes & Attachments  
New Note Attach File

No records to display

Always show me ▾ more records per related list

#### A.4.32. Imagen de layout de speaker en Salesforce

Allí seleccionar la foto del speaker, hacer clic en *Attach File* y luego en *Done*.

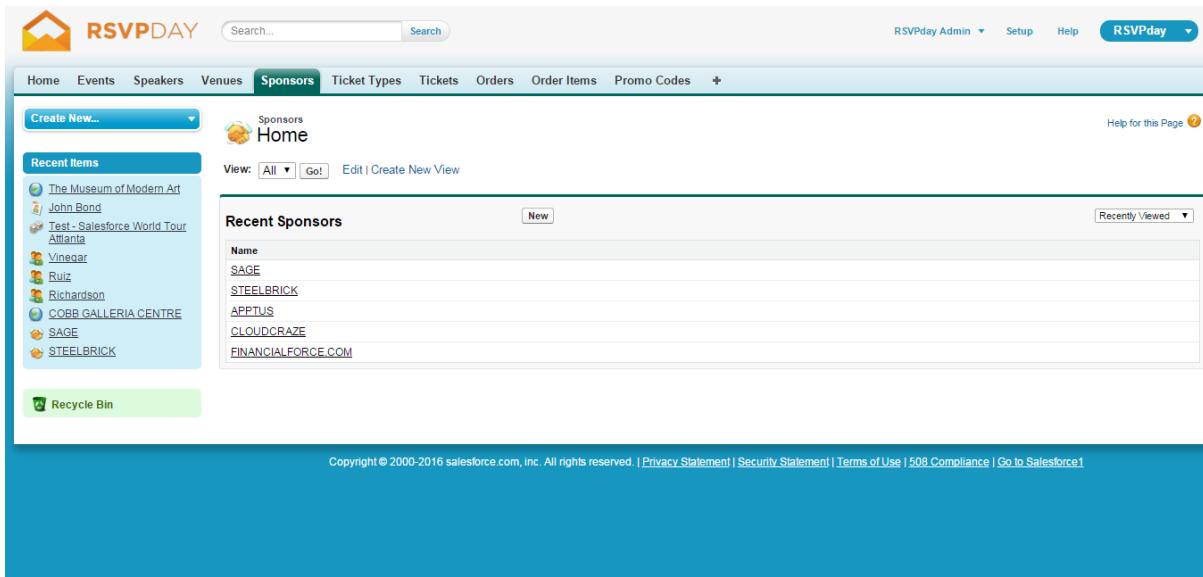


A.4.33. Imagen de pantalla para adjuntar foto de speaker en Salesforce

Al volver al registro editarlo para tildar el campo *Publish* y guardar. Luego de esto se mandarán los datos a Meteor para ser usados y se populará el campo External ID con la id que tiene el registro insertado en Meteor.

#### A.4.7.2. Enviar Sponsors

Para crear un *Sponsor* se debe hacer clic en *Sponsors* en la barra de navegación. De esta forma se mostrará una lista con los *sponsors* que existen en el sistema.



A.4.34. Imagen de lista de Sponsors en Salesforce

Luego se debe hacer clic en *New* y se mostrará un formulario para llenar los datos del sponsor.

The screenshot shows the RSVPday Admin interface. The top navigation bar includes links for Home, Events, Speakers, Venues, Sponsors, Ticket Types, Tickets, Orders, Order Items, Promo Codes, and a search bar. The current page is 'Sponsors'.

The main content area is titled 'Sponsor Edit' for 'WasteZero'. It contains a 'Save' button, a 'Save & New' button, and a 'Cancel' button. A note indicates that the 'Name' field is required information.

The form fields include:

- Name:** WasteZero
- Industry:** Available industries listed include Accident & Health Insurance, Advertising Agencies, Aerospace/Defense - Major Diversified, Aerospace/Defense Products & Services, Agricultural Chemicals, Air Delivery & Freight Services, and Air Services, Other. One industry, 'Waste Management', is selected under 'Chosen'.
- Website:** http://wastezero.com/
- Description:** Waste Management Company
- Contact Email:** wastezero@gmail.com
- Contact Phone:** 220002235
- Publish:** A checked checkbox.
- External ID:** G9LcLrwQSyGkqcWTs

At the bottom of the form are three buttons: Save, Save & New, and Cancel.

At the very bottom of the page, there is a copyright notice: Copyright © 2000-2016 salesforce.com, inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | 508 Compliance | Go to Salesforce!

#### A.4.35. Imagen de edición de sponsor en Salesforce

Luego de completar todos los campos (con excepción de *Publish* y *External ID*) y guardarlos, se debe ir a la sección de *Notes & Attachments* y hacer clic en *Attach File* para subir la foto correspondiente al sponsor.

The screenshot shows the Salesforce interface for the RSVPday application. The top navigation bar includes links for Home, Events, Speakers, Venues, Sponsors (which is the active tab), Ticket Types, Tickets, Orders, Order Items, Promo Codes, and a plus sign. A search bar is located at the top right. The main content area is titled "Sponsor WasteZero". It displays "Recent Items" on the left, including links to WasteZero, The Museum of Modern Art, John Bond, Test - Salesforce World Tour Atlanta, Vinegar, Ruiz, Richardson, COBB GALLERIA CENTRE, and SAGE. Below this is a "Recycle Bin" section. The "Sponsor Detail" section contains fields for Name (WasteZero), Industry (Waste Management), Website (<http://wastezero.com/>), Description (Waste Management Company), Contact Email (wastezero@gmail.com), Contact Phone (22002235), Publish (checkbox), and External ID. The "Created By" field shows "RSVPday Admin, 2/9/2016 4:57 PM" and the "Last Modified By" field shows "RSVPday Admin, 2/9/2016 5:18 PM". Buttons for Edit, Delete, and Clone are available. Below the detail section are sections for "Open Activities", "Activity History", "Event Section - Sponsor", and "Notes & Attachments". The "Notes & Attachments" section is highlighted with a red border. At the bottom, there are links for "Back To Top", "Always show me", and "more records per related list". The footer contains copyright information and links to Privacy Statement, Security Statement, Terms of Use, and Go to Salesforce.

#### A.4.36. Imagen de layout de sponsor en Salesforce

Allí seleccionar el archivo y hacer clic *Attach File* y luego en *Done*.

The screenshot shows the "Attach File to Sponsor" dialog for WasteZero. The top navigation bar and recent items list are identical to the previous screenshot. The dialog box is titled "Attach File to Sponsor WasteZero". It contains three numbered steps: 1. Select the File (with a note about selecting a file or using the Browse button), 2. Click the "Attach File" button (with a note about repeating steps 1 and 2 for multiple files), and 3. Click the Done button to return to the previous page (with a note about canceling an in-progress upload). A "Done" button is at the bottom. The footer contains copyright information and links to Privacy Statement, Security Statement, Terms of Use, and Go to Salesforce.

#### A.4.37. Imagen de pantalla para adjuntar foto de sponsor en Salesforce

Editar nuevamente el registro y tildar el campo *Publish* y guardar. De esta forma se enviará la información a Meteor para ser usada. Se debe tener en cuenta que para que se puedan enviar los datos deben estar completados todos los campos.

#### A.4.7.3. Enviar Venues

Para crear una *Venue* hacer clic en *Venues* en la barra de navegación. Allí se mostrará una lista con las *venues* actuales del sistema.

The screenshot shows the RSVPday application interface. At the top, there is a navigation bar with links for Home, Events, Speakers, Venues (which is highlighted in red), Sponsors, Ticket Types, Tickets, Orders, Order Items, Promo Codes, and a search bar. On the right side of the header, there are links for 'RSVPday Admin', 'Setup', 'Help', and 'RSVPday'. Below the header, there is a toolbar with buttons for 'Create New...', 'New Venue', 'Change Owner', and a filter dropdown set to 'All'. A 'Recent Items' sidebar on the left lists various venue names and IDs, such as 'Vinegar', 'Ruiz', 'Richardson', 'QI-48', 'TKT-000044', 'TKT-000043', 'QI-49', 'Q-42', 'Test - Salesforce World Tour Atlanta', and 'Full Price Ticket (Early Bird)'. The main content area displays a table with columns for 'Action' and 'Name'. One row is selected, showing 'Edit | Del' next to 'COBB GALLERIA CENTRE'. Above the table, there is a grid of letters from A to Z. At the bottom of the page, there are pagination controls ('1-1 of 1', '0 Selected'), navigation links ('Previous', 'Next'), and a copyright notice: 'Copyright © 2000-2016 salesforce.com, inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | 508 Compliance | Go to Salesforce1'.

A.4.38. Imagen de lista de Venues en Salesforce

Al hacer clic en *New Venue* se mostrará un formulario para llenar los datos de la *Venue*. Asegurarse de insertar un lugar conocido para Google Maps de lo contrario no lo reconocerá al crear el mapa del *event detail*.

The screenshot shows the RSVPday application's 'Venues' section. A modal window titled 'Venue Edit' is open for 'The Museum of Modern Art'. The form contains the following information:

Field	Value
Name	The Museum of Modern Art
Street 1	11 W 53rd St
Street 2	
City	New York
State/Province	New York
Postal Code	10019
Country	United States
Website	moma.org
Contact	John Bond
Contact Phone	(223) 344-4551
Geolocation (Latitude)	40.76
Geolocation (Longitude)	-73.98
Other Information	(empty)
Archive	<input type="checkbox"/>
Contact Email	johndoe@gmail.com
Publish	<input checked="" type="checkbox"/>
External ID	a04B0000002estN

Buttons at the bottom of the modal include 'Save', 'Save & New', and 'Cancel'.

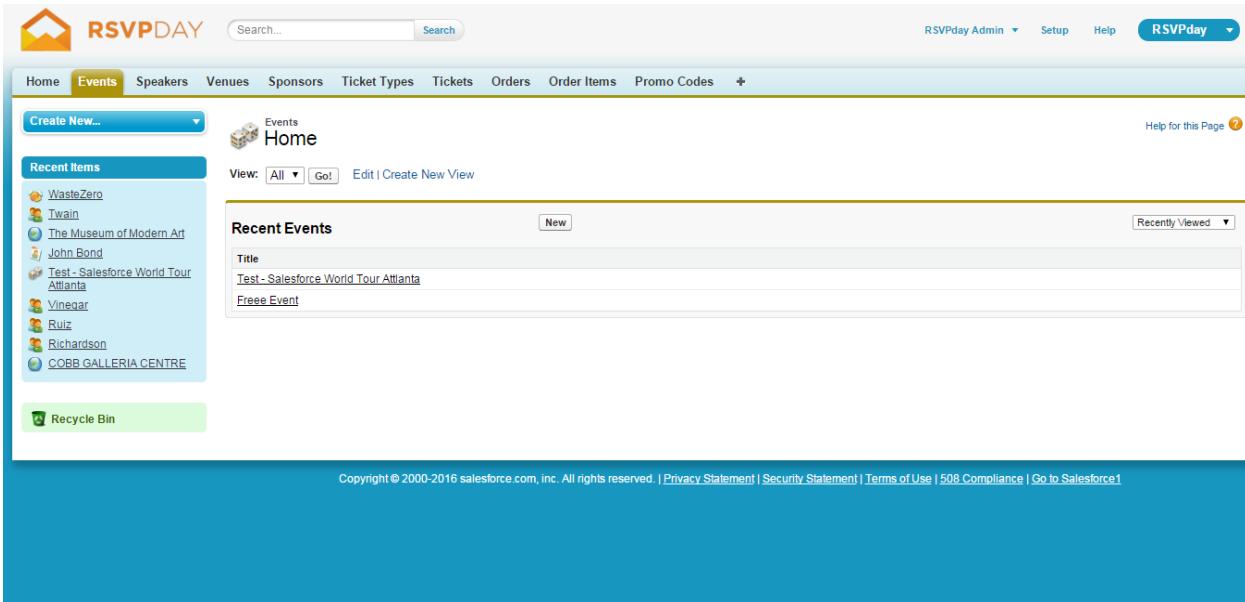
#### A.4.39. Imagen de edición de venue en Salesforce

Es importante que el campo *Publish* no esté tildado al guardar por primera vez y que *External ID* no sea completado, ya que se populará una vez que se mande el *venue* a Meteor. Luego de guardar, editar nuevamente y cliquear en *Publish* y luego en *Save*.

Así el registro será mandado a Meteor y podrá ser utilizado.

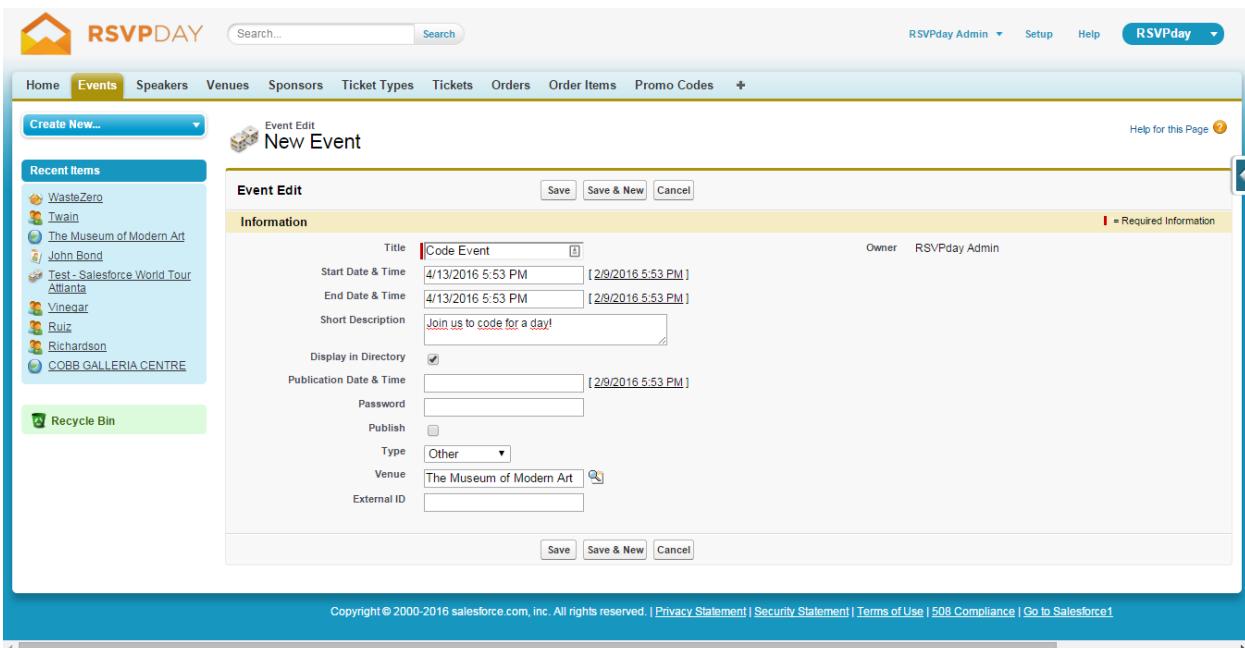
#### A.4.7.4. Enviar Events

Para crear un *Event* se debe hacer clic en *Events* en la barra de navegación. De esta forma se mostrará una lista con los *events* que existen en el sistema.



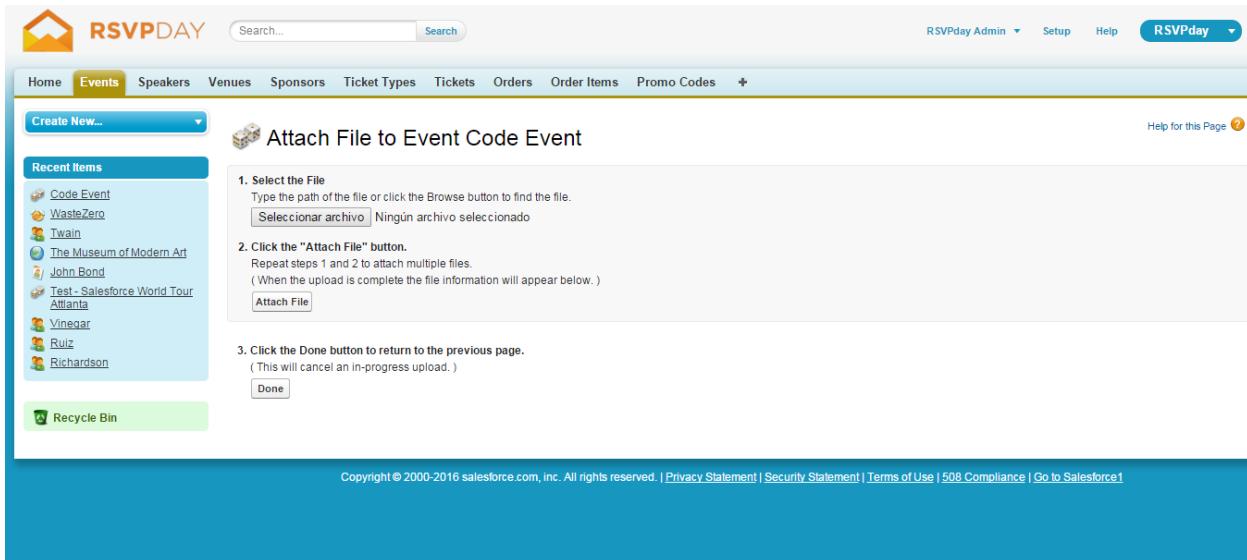
#### A.4.40. Imagen de lista de Events en Salesforce

Hacer click en *New*, allí aparecerá el formulario para crear el registro del *event*. Se deben completar todos los datos del formulario con excepción de *Publish* y *External ID* y hacer clic en *Save*.



#### A.4.41. Imagen de edición de event en Salesforce

Luego de cliquear en *Save* se mostrarán los datos ingresados y las secciones del evento. En la sección de *Notes & Attachments* hacer clic en *Attach File*.



A.4.42. Imagen de pantalla para adjuntar foto de evento en Salesforce

Seleccionar la foto del evento, hacer clic en *Attach File* y luego en *Done*.

The screenshot shows the RSVPday application interface. At the top, there is a navigation bar with links for Home, Events, Speakers, Venues, Sponsors, Ticket Types, Tickets, Orders, Order Items, Promo Codes, and a 'Create New...' button. The 'Events' tab is selected. On the left, there is a sidebar titled 'Recent Items' containing links to various event, speaker, and venue records. Below the sidebar is a 'Recycle Bin' section.

The main content area is titled 'Event Code Event'. It contains a 'Event Detail' section with fields for Title (Code Event), Start Date & Time (4/13/2016 5:53 PM), End Date & Time (4/13/2016 5:53 PM), Short Description (Join us to code for a day!), Display in Directory (checked), Publication Date & Time, Password, Publish (unchecked), Type (Other), Venue (The Museum of Modern Art), External ID, Created By (RSVPday Admin), and Last Modified By (RSVPday Admin). There are also 'Edit', 'Delete', and 'Clone' buttons.

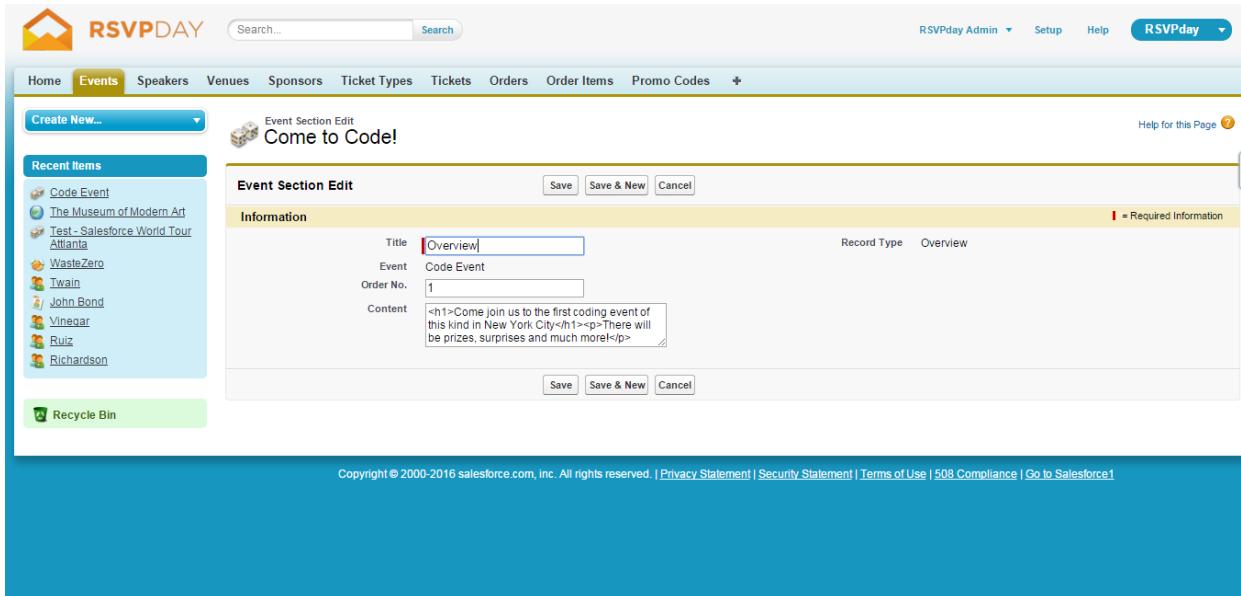
Below the event detail are several sections for managing related data:

- Event Sections**: A section for creating new event sections, with a 'New Event Section' button and a note that no records are displayed.
- Event Section - Speakers**: A section for creating new event section speakers, with a 'New Event Section - Speaker' button and a note that no records are displayed.
- Event Section - Venue**: A section for creating new event section venues, with a 'New Event Section - Venue' button and a note that no records are displayed.
- Sponsor Types**: A section for creating new sponsor types, with a 'New Sponsor Type' button and a note that no records are displayed.
- Event Section - Sponsors**: A section for creating new event section sponsors, with a 'New Event Section - Sponsor' button and a note that no records are displayed.
- Ticket Types**: A section for creating new ticket types, with a 'New Ticket Type' button and a note that no records are displayed.
- Orders**: A section for creating new orders, with a 'New Order' button and a note that no records are displayed.
- Promo Codes**: A section for creating new promo codes, with a 'New Promo Code' button and a note that no records are displayed.
- Notes & Attachments**: A section for managing notes and attachments, with a 'New Note' button, an 'Attach File' button, and a 'View All' link. It lists one attachment: 'codeevent.jpg' (Last Modified: 2/9/2016 6:09 PM, Created By: RSVPday Admin).

At the bottom of the page, there is a footer with copyright information: 'Copyright © 2000-2016 salesforce.com, inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | 508 Compliance | Go to Salesforce'.

#### A.4.43. Imagen de layout de event en Salesforce

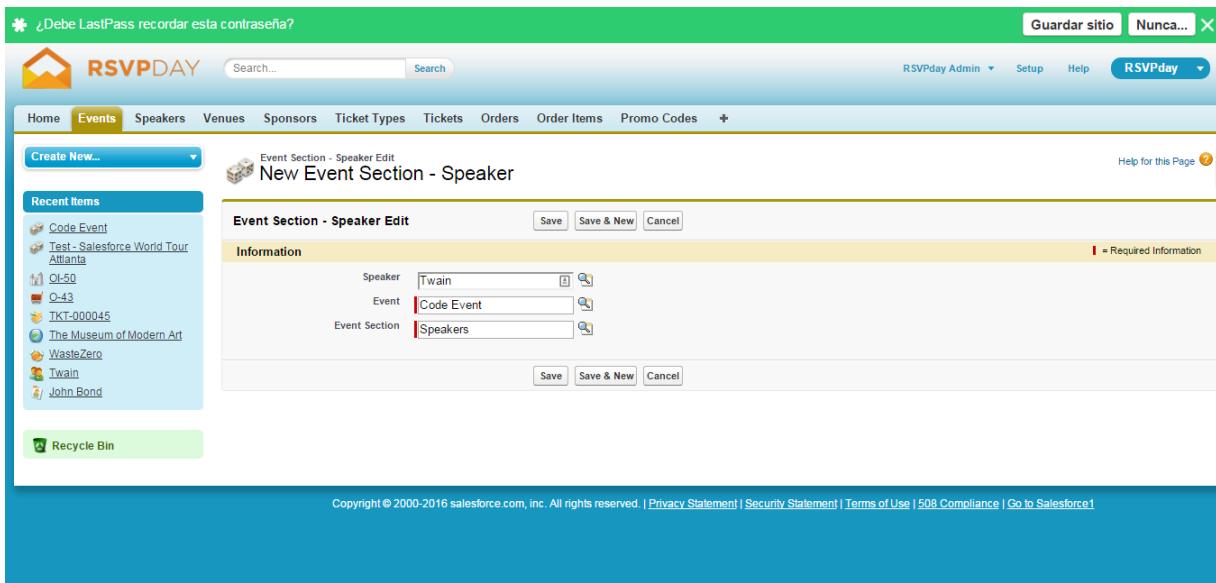
Dentro de *Event Sections* es importante tener en cuenta que se deben agregar obligatoriamente tres secciones: *Overview*, *Speakers* y *Sponsors*. Para hacer esto se debe hacer clic en *New Event Section* y allí se abrirá una página con un *dropdown* para elegir el tipo de sección y luego de cliquear en *Continue* se mostrará el formulario para ingresar la sección.



A.4.44. Imagen de edición de event section en Salesforce

Cabe recordar que la sección *Overview* siempre debe tener el *Order Nr.* 1, *Sponsors* siempre debe tener el último número en el orden (si son 10 *event sections* debe ser el número 10) y *Speakers* el penúltimo (si son 10 *event sections* debe ser el número 9). Si se quiere agregar una sección personalizada se debe elegir la opción *Custom*. También es importante destacar que para que se vea de la mejor forma en el detalle del evento en Meteor, el campo *Content* debe tener formato html como en la imagen.

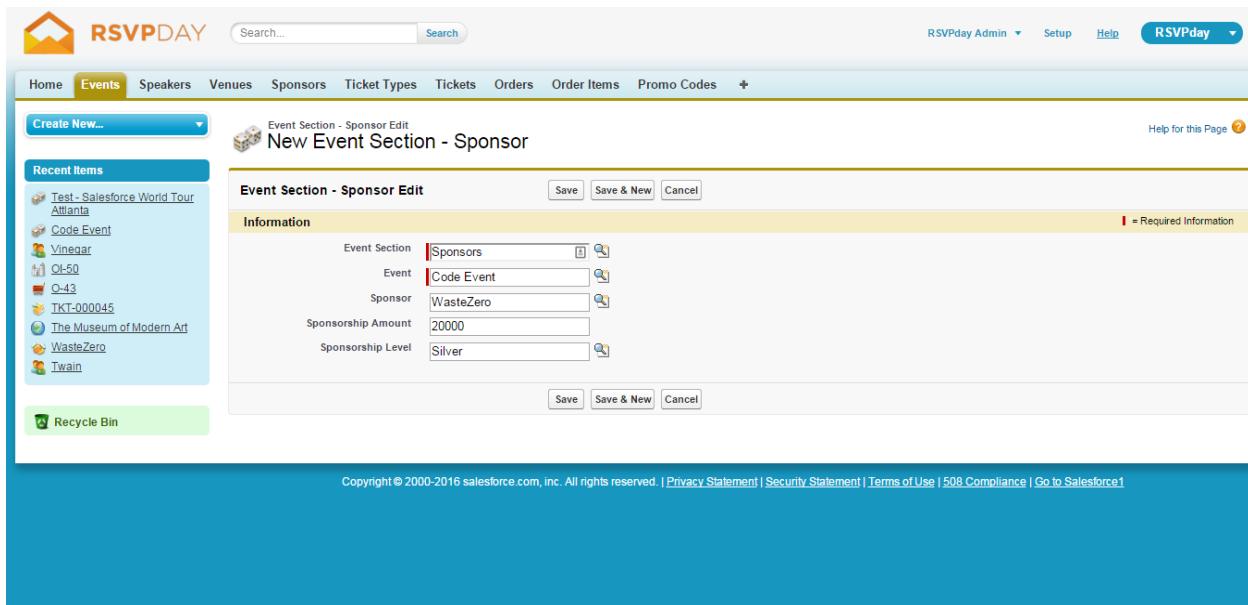
En la *Event Section - Speakers* se asocian los speakers que van a ser parte del evento.



A.4.45. Imagen de edición de event section - speaker en Salesforce

El nombre que se ingrese en el campo *Event Section* debe ser *Speaker*.

Luego de guardar pasamos a *Event Section - Sponsors*, allí hacemos clic en *New Event Section - Sponsors*. En el campo *Event Section* ponemos *Sponsors*, asociamos el sponsor que queremos e ingresamos los demás datos.



A.4.46. Imagen de edición de event section - sponsor en Salesforce

Luego de guardar volvemos al *event* y tenemos que ingresar los *Ticket Types*, que son los tipos de tickets que se van a vender. Hacemos clic en *New Ticket Type* y nos mostrará el formulario para ingresar el *ticket type*.

Ingresamos los datos del *ticket type*, entre ellos cabe destacar que para que se vea la *description* tenemos que tildar *Display Description*, que si el *Price* es 0 entonces debemos tildar *Free Ticket?*, que las fechas de venta del *ticket type* deben ser anteriores a la fecha de realización del evento y el *ticket type* aparecerá a la venta en Meteor solo si la fecha actual se encuentra entre estas dos fechas.

También es importante recordar que para que se muestren correctamente en el sitio, los ticket types deben ingresarse con sus precios de menor a mayor, o sea, primero el de menor precio y así hasta llegar al mayor.

**Ticket Type Edit**

**Information** | = Required Information

**Ticket Type Name:** Cheap Ticket

**Description:** This ticket is cheaper than the rest

**Quantity:** 200

**Price:** 50

**Display Description?**

**Private Ticket:**

**Display Remaining Avail Seat?**

**Free Ticket?**

**Sales Start Date & Time:** 2/10/2016 6:05 AM [2/10/2016 6:05 AM]

**Sales End Date & Time:** 3/29/2016 6:05 AM [2/10/2016 6:05 AM]

**Event:** Code Event

**Help Text:**

**Buttons:** Save, Save & New, Cancel

A.4.47. Imagen de edición de ticket type en Salesforce

Al guardar se mostrarán nuevamente todos los datos del evento.

The screenshot displays the RSVPday application interface. At the top, there is a navigation bar with links for Home, Events, Speakers, Venues, Sponsors, Ticket Types, Tickets, Orders, Order Items, Promo Codes, and a 'Create New...' button. The 'Events' tab is currently selected.

**Event Detail:**

- Title: Code Event
- Start Date & Time: 4/13/2016 5:53 PM
- End Date & Time: 4/13/2016 5:53 PM
- Short Description: Join us to code for a day!
- Display in Directory: ✓
- Publication Date & Time: (empty)
- Password: (empty)
- Publish: (checkbox)
- Type: Other
- Venue: The Museum of Modern Art
- External ID: TKT-000045
- Created By: RSVPday Admin, 2/9/2016 6:03 PM
- Last Modified By: RSVPday Admin, 2/9/2016 6:03 PM

**Event Sections:**

Action	Title	Order No.	Record Type	Content
Edit   Del	Overview	1	Overview	<h1>Come join us to the first coding event of this kind in New York City</h1><p>There will be prizes, surprises and much more!</p>
Edit   Del	Speakers	2	Speaker	(empty)
Edit   Del	Sponsors	3	Sponsor	(empty)

**Event Section - Speakers:**

Action	Speaker
Edit   Del	Twain

**Event Section - Venue:**

Action	Venue
No records to display	

**Sponsor Types:**

Action	New Sponsor Type
No records to display	

**Event Section - Sponsors:**

Action	Event Section - Sponsor: No.	Sponsor	Sponsorship Level	Sponsorship Amount
Edit   Del	ESS-0000000005	WasteZero	Silver	\$20,000.00

**Ticket Types:**

Action	Ticket Type Name
Edit   Del	Cheap Ticket
Edit   Del	More Expensive Ticket

**Orders:**

Action	New Order
No records to display	

**Promo Codes:**

Action	New Promo Code
No records to display	

**Notes & Attachments:**

Action	Type	Title	Last Modified	Created By
Edit   View   Del	Attachment	codeevent.jpg	2/9/2016 6:09 PM	RSVPday Admin

**Footer:**

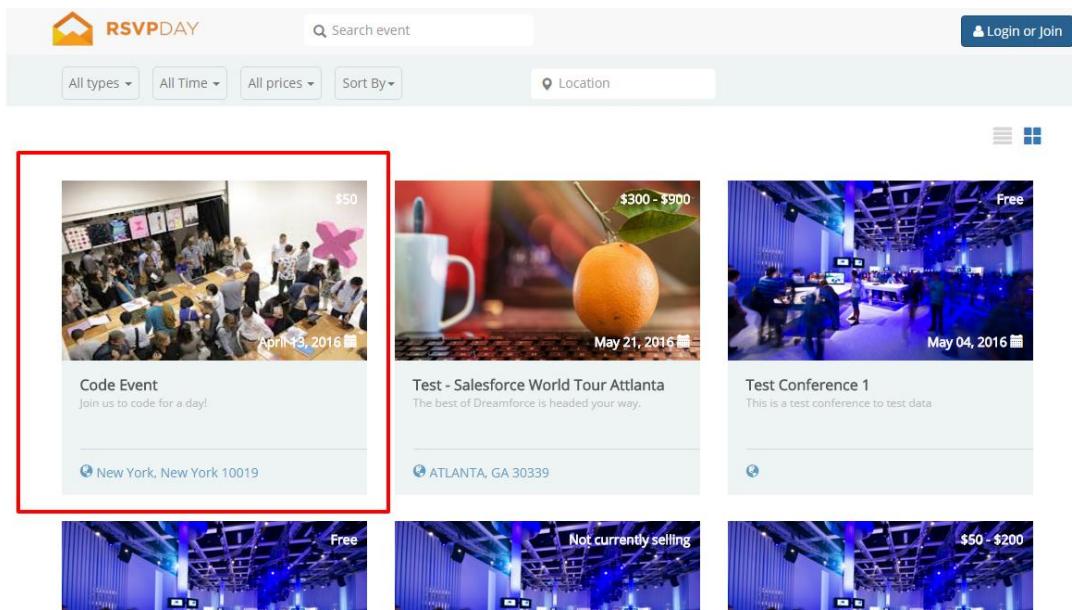
Always show me ▾ more records per related list  
Copyright © 2000-2016 salesforce.com, inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | 508 Compliance | Go to Salesforce!

#### A.4.48. Imagen de layout de event completo en Salesforce

Cabe recordar que al campo *External ID* no se le debe ingresar ningún dato, ya que este se popula luego de que se envía el registro a Salesforce con la id del documento en MongoDB.

Si ya no hay más datos que se quieran ingresar entonces se debe hacer clic en *Edit* y tildar el campo *Publish* y guardar. Esto mandará los datos del evento a Meteor para ser usado allí.

En las imágenes siguientes se muestran cómo quedó el evento en Meteor, primero en el directorio y luego su detalle.



A.4.49. Imagen de event enviado desde Salesforce en directorio de eventos en Meteor

**RSVPDAY**

Search event

Login or Join

ALL EVENTS / Code Event

OVERVIEW SPEAKERS SPONSORS

The Museum of Modern Art  
17:53 | 17:53  
★ SAVE EVENT  
Hurry up, only 100 seats!  
\$50  
GET TICKETS GET PDF

OVERVIEW ABOUT THE EVENT

Come join us to the first coding event of this kind in New York City

There will be prizes, surprises and much more!

When

Start Date: April 13, 2016  
End Date: April 13, 2016  
From 17:53 to 17:53  
Añadir al Calendario

Where

The Museum of Modern Art  
11 W 53rd St  
New York, New York 10019

Organizer

Contact Organizer

Share

f 0    t 0    in 0    g+ 0    e 0

#### THE SPEAKERS



Mark Twain  
Manager  
Apple

#### THE SPONSORS

**WasteZero®**

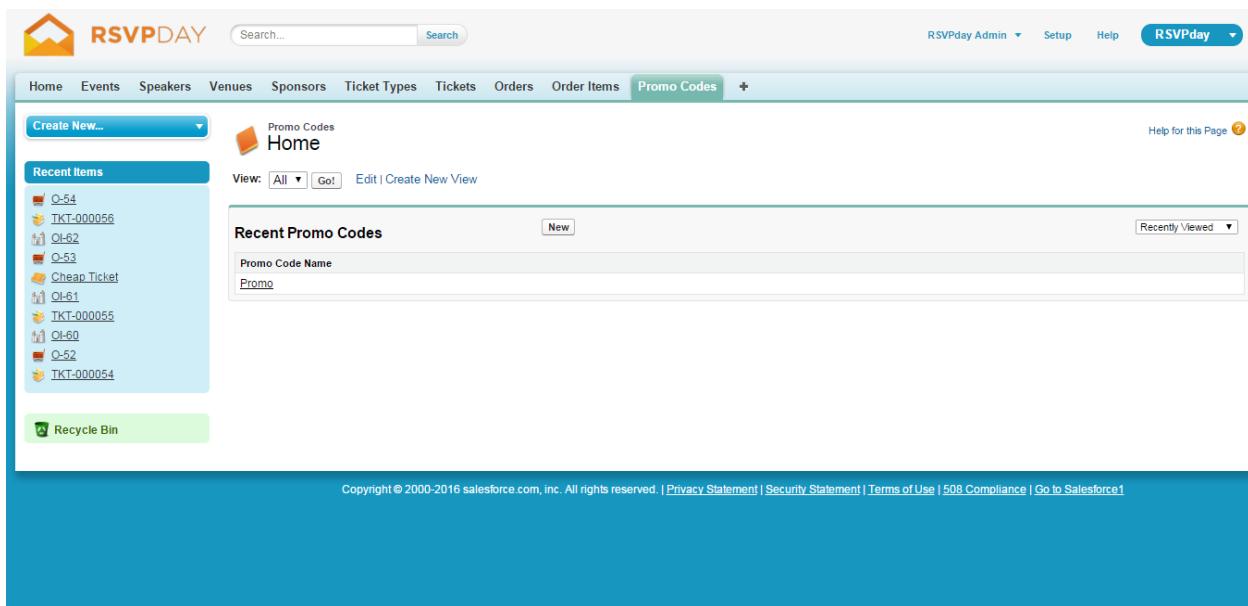
© 2016 RSVPday

A.4.50. Imagen de evento enviado desde Salesforce en Meteor

#### A.4.7.5. Promo Codes

Los *Promo Codes* no se envían a Meteor, pero sí se realiza una consulta a Salesforce desde Meteor para saber si determinado código (que está guardado en Salesforce) es válido.

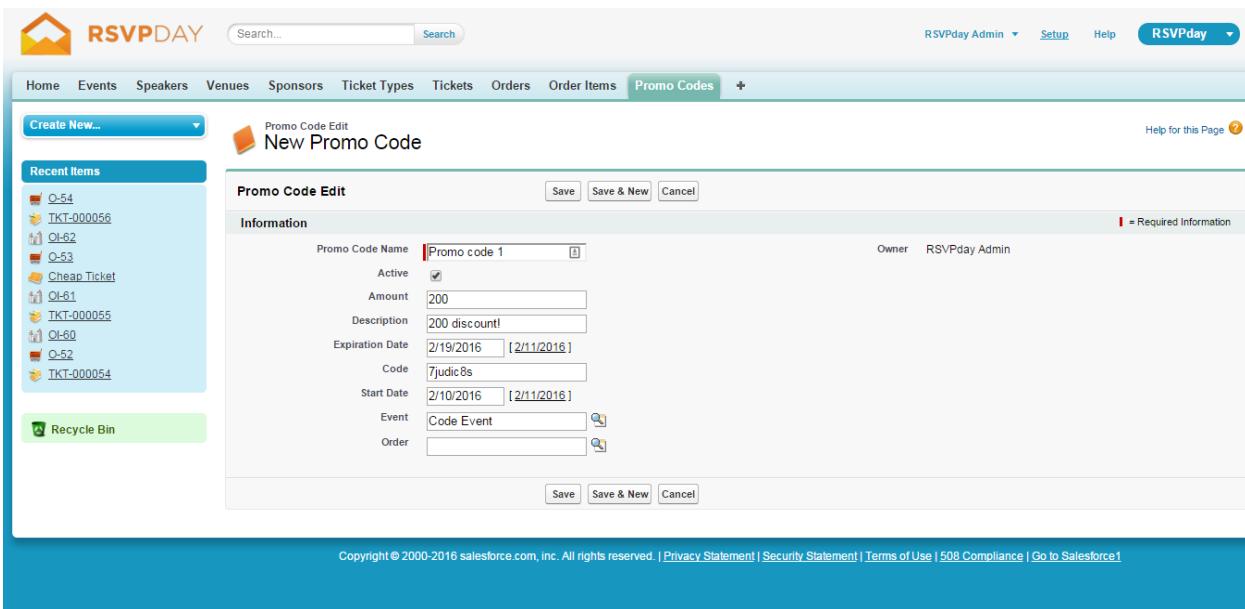
Para crear un *promo code* en Salesforce hay que hacer clic en *Promo Codes* en la barra de navegación. Allí se mostrarán todos los promo codes recientes. Hacer clic en *New*.



A.4.51. Imagen de lista de promo codes en Salesforce

Se mostrará un formulario para completar la información del *promo code*. Allí, para que el *promo code* funcione, debe estar activo, o sea el campo *Active* debe estar tildado. También la fecha en la que se manda la consulta para ser usado debe ser una fecha entre el *Start Date* y el *Expiration Date*. Y también debe estar asociado a determinado evento.

El campo *Order* debe dejarse en blanco. Allí se asocia el *promo code* a la orden luego de que se hizo la compra usando este *promo code*.



A.4.52. Imagen de edición de promo code en Salesforce

La creación de un *promo code* también se puede hacer desde el registro del *event*, en la sección *Promo Codes*. Allí hacer clic en *New Promo Code* para asignarle uno al evento.

## A.5. Manual de Deployment

El despliegue del proyecto es sumamente sencillo, se debe tener una copia local del repositorio remoto y tener configurado en la carpeta .git el archivo config con la siguiente información:

```
[remote "heroku"]
  url = https://git.heroku.com/rsvpday-demo.git
  fetch = +refs/heads/*:refs/remotes/heroku/*
```

Previamente, se debe de haber instalado el toolbelt de Heroku y haberse logueado en Heroku con:

```
heroku login
```

Luego hay que agregar el *buildpack* de Meteor para que Heroku permita hacer el *deployment* de una aplicación de Meteor a Heroku.

```
heroku buildpacks:set https://github.com/AdmitHub/meteor-buildpack-horse.git
```

El siguiente paso es agregar a git el *remote* de la aplicación de la siguiente manera:

```
heroku git:remote -a rsvpday-demo
```

Una vez hecho eso, estando en la rama master (*git checkout master*) se debe ejecutar en la consola el comando:

```
git push heroku
```

Luego, Heroku hace los builds y genera las dependencias.

Heroku posee variables de entorno que son aquellas configuraciones necesarias para la aplicación. En RSVPDAY se necesitan configurar las siguientes:

```

==== rsvpday-demo Config Vars ====
FACEBOOK_ID: 1111111111111111
FACEBOOK_SECRET: 1a1a1a1aa1a1a1a1a1aa1a1a
GOOGLE_ID: 1111111111111111
GOOGLE_SECRET: 1a1a1a1aa1a1a1a1a1aa1a1a
LINKEDIN_ID: 1111111111111111
LINKEDIN_SECRET: 1a1a1a1aa1a1a1a1a1aa1a1a
MAIL_URL:
smtp://mailejemplo%40gmail.com:contraseña@smtp.gmail.com:465/
MONGOLAB_URI:
mongodb://heroku_63djtm6j:llv62qv14m8sf3faoph4mn54cb@ds051334.mongolab.co
m:51334/heroku_63djtm6j
MONGO_URL: mongodb://heroku@x.mongolab.com:51334/heroku_x
ROOT_URL: http://rsvpday-demo.herokuapp.com
TWITTER_ID: 1111111111111111
TWITTER_SECRET: 1a1a1a1aa1a1a1a1a1aa1a1a
process.env.MAIL_URL: smtp://rsvpdayapp@RsvpDay.App:25

```

Las variables de entorno relacionadas a redes sociales se obtienen por tener una app registrada en el centro de desarrolladores de cada red social. MAIL\_URL es la variable usada para el envío de emails al igual que process.env.MAIL\_URL. Las variables de entorno relacionadas a MongoDB son extraídas de las configuraciones de la base de datos creada en Heroku.

Para configurar los valores de dichas variables sólo se debe ejecutar el comando:

```
heroku config:set LINKEDIN_ID=1111111111111111
```

Finalizado este proceso la aplicación queda lista para ser usada.

## A.6. Presentación de la Defensa

Para acceder a la presentación que mostramos durante la defensa del proyecto se puede ingresar al siguiente link:  
[http://prezi.com/y08hqx\\_\\_r4bh/?utm\\_campaign=share&utm\\_medium=copy&rc=ex0share](http://prezi.com/y08hqx__r4bh/?utm_campaign=share&utm_medium=copy&rc=ex0share).

## A.7. Especificaciones en cuanto al Código Producido

Además de todo el código escrito para la aplicación del lado público, también se escribieron algunas clases y métodos del lado privado. El propósito de esto fue crear la REST API que recibe los datos de las compras de tickets y proveer información sobre la validez de los *promo codes*. Además de en los DVDs presentados también se puede acceder a estas clases yendo a <https://login.salesforce.com/> ingresando el usuario y contraseña provisto por nosotras<sup>3</sup>, luego yendo a *Setup* y allí a *Develop > Apex Classes*. Allí se muestran todas las clases en el org. Las que fueron escritas por nosotras fueron:

- OrderItemUtility
- OrderRestResource
- OrderUtility
- PromoCodeResponse
- PromoCodeRestResource
- PromoCodeUtility
- TicketTypeUtility
- TicketUtility
- OrderItemTrigger

## A.8. Acceso a Salesforce

A los efectos de la corrección las credenciales del usuario proporcionado a los correctores para ingresar a Salesforce serán `admin@rsvpday.dev` y la contraseña `rsvpd4y!`

Luego de ingresar usuario y contraseña el sistema pedirá un código que va a llegarle al mail que esté asociado al usuario de salesforce. En este caso la persona que quiera ingresar deberá mandarle un mail a Ximena Lasserre a su casilla [xlasserre@gmail.com](mailto:xlasserre@gmail.com) para pedirle el código de verificación. Este código se pide cada vez que alguien desea entrar desde un nuevo equipo.

La url de acceso es <https://login.salesforce.com/>.

---

<sup>3</sup> Ver punto del Anexo A.8. Acceso a Salesforce.

## A.9. Carta de Conformidad del Cliente



Montevideo, Lunes 15 de Febrero de 2016

A quien corresponda,

A través de la presente queremos hacer de su conocimiento que estamos plenamente satisfechos con el resultado del proyecto RSVPday y con el trabajo realizado por Ximena Lasserre y Marianne Maisonneuve.

La buena comunicación y el compromiso que ellas demostraron durante el desarrollo del proyecto permitieron cumplir con los objetivos propuestos. Cumpliendo con los plazos de entrega y la totalidad de las tareas asignadas según las especificaciones dadas.

Saluda atentamente,

*Diego Febles*

Diego Febles  
Director de Servicios Profesionales

## A.10. Acerca de la Documentación y Código Presentado en los DVDs

En los DVDs presentados para la entrega además de la documentación en formato pdf que se encuentra en la carpeta *Documentación*, hay otras dos carpetas. Una de ellas, llamada *rsvpday-site*, contiene todo el proyecto de la aplicación pública, el cual fue desarrollado en su totalidad por nosotras. La otra, llamada *rsvpday-private*, contiene las clases de Salesforce que creamos o modificamos nosotras para crear la REST API en Salesforce y así comunicarnos con Salesforce desde la aplicación pública y enviar los

datos de la compra, cancelar *Orders* y consultar *Promo Codes*. En dicho org de Salesforce existen otras clases que no modificamos o creamos nosotras y no están agregadas a esta carpeta. *Solo agregamos las clases a las que contribuimos nosotras*. Este código no podrá correrse localmente, porque es código de Salesforce, desarrollado en Apex, pero sí se verá reflejado en el org de Salesforce al que se puede acceder mediante las instrucciones dadas en el Anexo A.8.