

---

**Universidad de Valladolid**

**E. T. S. Ingeniería Informática**

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática  
Mención en Tecnologías de la Información

# **Gestor de promociones con código**

Autor:

**D. Alberto Pesquera Senovilla**

Tutores:

**D. Benjamín Sahelices Fernández**

**D. David Gómez González**

## **Resumen**

El trabajo se centrará en la creación de una aplicación que gestione las promociones con código y ofertas vinculadas desde un panel de administración web que proporcione una API y un ejemplo de utilización de dicha API. Se desarrollará con las tecnologías Slim, Twig y Doctrine para la aplicación y Angular2 para el ejemplo de utilización.

Este proyecto se realiza en la empresa Brooktec por lo que se ha aplicado su metodología de trabajo, las tecnologías y los entornos para el desarrollo de esta aplicación.

## Tabla de contenidos

Capítulo 1. Introducción.....	6
Objetivos.....	6
Resumen de contenidos.....	6
Capítulo 2. Contexto.....	8
Metodología.....	8
Scrum.....	8
Kanban.....	12
Scrumban.....	13
Tecnologías.....	13
Vagrant.....	13
Instalación.....	14
Utilización.....	14
Node.js y npm.....	14
Instalación.....	15
Utilización.....	15
Gulp.....	15
Instalación.....	16
Utilización.....	16
Slim.....	17
Instalación.....	17
Utilización.....	17
Twig.....	18
Instalación.....	18
Utilización.....	18
Doctrine.....	19
Instalación.....	19
Utilización.....	19
Angular2.....	20
Instalación y utilización.....	20
Entornos.....	20
Entorno de producción.....	21
Entorno de pruebas.....	21
Entorno de desarrollo local.....	22
Capítulo 3. Análisis.....	24
Glosario de términos.....	24
Roles.....	24
Historias de usuario.....	24
Especificaciones técnicas.....	31
Tecnología a utilizar.....	32
Aplicación web.....	32
API.....	32
Página web del usuario final.....	32
Riesgos.....	33
Tabla de riesgos inicial.....	33
Planificación.....	34
Capítulo 4. Descripción de los sprints.....	35
Sprint 1, del 29/2/2016 al 13/3/2016.....	35
Análisis.....	35

Riesgos.....	37
Historias de usuario.....	38
Criterios de aceptación US-0008.....	38
Sprint 2, del 14/3/2016 al 27/3/2016.....	40
Análisis.....	40
Riesgos.....	40
Historias de usuario.....	41
Criterios de aceptación US-0009.....	41
Sprint 3, del 28/3/2016 al 10/4/2016.....	43
Análisis.....	43
Riesgos.....	43
Historias de usuario.....	44
Criterios de aceptación US-0001.....	44
Criterios de aceptación US-0002.....	46
Sprint 4, del 11/4/2016 al 24/4/2016.....	48
Análisis.....	48
Riesgos.....	48
Historias de usuario.....	50
Criterios de aceptación US-0004.....	50
Criterios de aceptación US-0017.....	50
Sprint 5, del 25/4/2016 al 8/5/2016.....	51
Análisis.....	51
Riesgos.....	53
Historias de usuario.....	55
Criterios de aceptación US-0004 (v2).....	55
Criterios de aceptación US-0013.....	55
Criterios de aceptación US-0014.....	56
Criterios de aceptación US-0015.....	57
Sprint 6, del 9/5/2016 al 22/5/2016.....	58
Análisis.....	58
Riesgos.....	59
Historias de usuario.....	60
Criterios de aceptación US-0012.....	60
Criterios de aceptación US-0016.....	62
Sprint 7, del 23/5/2016 al 5/6/2016.....	62
Análisis.....	62
Riesgos.....	62
Historias de usuario.....	64
Criterios de aceptación US-0003.....	64
Criterios de aceptación US-0005.....	64
Criterios de aceptación US-0006.....	65
Sprint 8, del 6/6/2016 al 19/6/2016.....	65
Análisis.....	65
Riesgos.....	66
Historias de usuario.....	67
Criterios de aceptación US-0018.....	67
Criterios de aceptación US-0019.....	68
Sprint 9, del 20/6/2016 al 26/6/2016.....	68
Análisis.....	68
Riesgos.....	68

Historias de usuario.....	70
Criterios de aceptación US-0010.....	70
Criterios de aceptación US-0011.....	71
Criterios de aceptación US-0007.....	71
Criterios de aceptación US-0020.....	71
Capítulo 5. Diseño de la interfaz.....	72
Acceso al gestor.....	72
Promociones.....	73
Crear promociones.....	73
Editar promociones.....	76
Borrar promociones.....	76
Código.....	76
Ofertas.....	77
Crear, editar y borrar ofertas.....	78
Usuarios del gestor.....	80
Crear, editar y borrar usuarios.....	81
Mi perfil.....	82
Usuarios registrados.....	82
Capítulo 6. Conclusiones y trabajo futuro.....	84
Conclusiones.....	84
Trabajo futuro.....	84
Bibliografía.....	85
Metodología.....	85
Tecnologías.....	85
Documentación y ejemplos prácticos.....	86
Anexo.....	87
Contenido del CD.....	87
Documentación de la API.....	87
Obtener la promoción.....	87
Registrar un usuario.....	88
Identificar un usuario.....	89

# Capítulo 1. Introducción

En este apartado se tratará de una forma breve y concisa el tema elegido para este proyecto. Se hablará sobre los objetivos y se realizará un breve resumen del resto de la memoria.

## Objetivos

Los objetivos que se han fijado para la realización de este proyecto son:

- Se realizará un estudio de la tecnología de servidor Slim, Twig y Doctrine.
- Se realizará un estudio de la tecnología de cliente Angular2.
- Se desarrollará una aplicación web con las tecnologías de servidor.
- La aplicación web será privada por lo que hay que identificarse para utilizarla.
- La aplicación web permitirá la creación, visualización, edición y borrado de promociones y ofertas.
- Las ofertas se agruparán en promociones pudiendo estar en varias de esas promociones al mismo tiempo.
- Se desarrollará una API web para obtener la información de las promociones y sus ofertas asociadas.
- Se desarrollará un ejemplo de página web que utilice la API con las tecnologías de cliente.

## Resumen de contenidos

En primer lugar, en el **capítulo 2** se expondrá el contexto del desarrollo de la aplicación, ya que se ha desarrollado en la empresa Brooktec. Se hablará, sobre todo, de la metodología de trabajo que se utiliza en la empresa y se explicará cada una de las tecnologías utilizadas y los diferentes entornos por los que pasa la aplicación durante el desarrollo.

A continuación, en el **capítulo 3** se hará un análisis del proyecto, en el que se incluirán glosario, roles, historias, especificaciones, tecnología, riesgos y planificación.

En el **capítulo 4** se describirá el desarrollo de cada iteración, detallando las fases de dicha iteración, comenzando por el análisis, los riesgos, las historias de usuario y finalmente los criterios de aceptación. En esta última parte se incluirán imágenes para mostrar que se satisfacen los criterios de aceptación definidos en las historias de usuario.

El **capítulo 5** contendrá la explicación del diseño de la interfaz y su utilización. En este capítulo se podrán observar las funcionalidades implementadas y cómo hacer uso de ellas de forma práctica con ejemplos.

Para terminar con la memoria del proyecto realizado, en el **capítulo 6** se comentarán las

conclusiones y trabajo futuro, valorando posibles áreas de mejora y líneas abiertas con las que se puede continuar trabajando en el futuro. Tras ello se encuentra la bibliografía consultada y el anexo.

# Capítulo 2. Contexto

En este capítulo se va a detallar el contexto del proyecto, ya que este trabajo de fin de grado ha sido propuesto por la empresa Brooktec, una empresa que se dedica al desarrollo de soluciones a medida para agencias de publicidad. Se dividirá en tres apartados, uno para la metodología, otro para las tecnologías y el último para los entornos.

## Metodología

En este apartado se explica la metodología que utiliza Brooktec para el desarrollo de sus proyectos, Scrumban. Esta palabra es el resultado de la combinación de Scrum + kanban. Paso a explicar ambas metodologías por separado para poder entender mejor la combinación de ambas y su potencial de cara a la exitosa finalización de los proyectos.

## Scrum

Es una metodología para la gestión y desarrollo de software. Basada en un proceso iterativo e incremental.

Hay que tener en cuenta una serie de definiciones antes de comenzar a describir el resto de elementos que conforman Scrum:

- **Historias de Usuario:** es un recordatorio de una conversación con cliente donde se explica en un lenguaje coloquial la necesidad que le surge. Las características que debe cumplir una historia de usuarios son:
  - Independientes unas de otras. Se pueden hacer en cualquier orden, no dependen unas de otras.
  - Negociables: no son contratos, sino promesas de comunicación.
  - Valoradas por los clientes o usuarios: siempre debemos dar valor a cliente.
  - Estimables: la conversación alrededor de esa historia desencadena en una estimación, de esta manera nos podremos hacer una idea del tiempo que llevará el proyecto.
  - Pequeñas: abordables en un periodo de tiempo de no más de una semana. Si no fuera así, hay que volver a partir esa historia en dos.
  - Verificables: se debe poder probar la historia para saber que realmente está acabada.
- **Sprint:** espacio de tiempo en el cual el equipo de desarrollo se compromete a entregar una cantidad de historias de usuario finita, que aporte valor a cliente final. Normalmente la duración de un sprint es de entre 2 a 3 semanas. Se fija al principio del proyecto y se mantiene hasta la finalización del mismo.
- **Product Backlog:** Lista no ordenada de historias de usuario.
- **Sprint Backlog:** Lista de historias de usuario ordenadas por prioridad. Es el



documento necesario para comenzar un Sprint.

- **Burndown chart** es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Se puede ver un ejemplo en la figura 1.

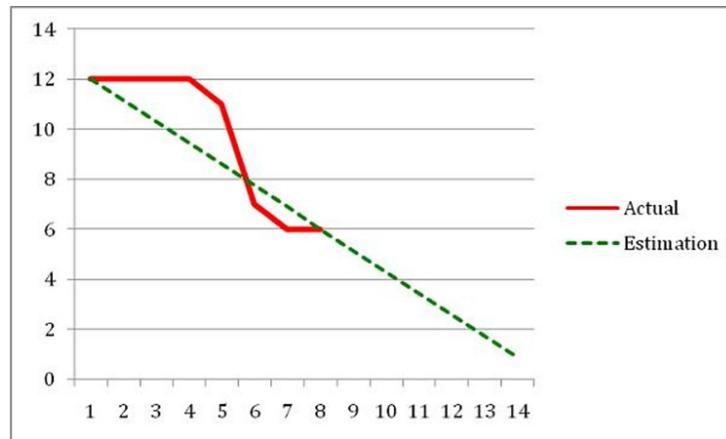


Figura 1: Burndown chart

Dentro del día a día de esta metodología hay que encajar una serie de reuniones y de roles que ayudan a mantener los principios básicos de las metodologías ágiles

Es bastante típico encontrarse la explicación de los roles que se definen dentro del contexto de esta metodología a través del chiste de la figura 2.

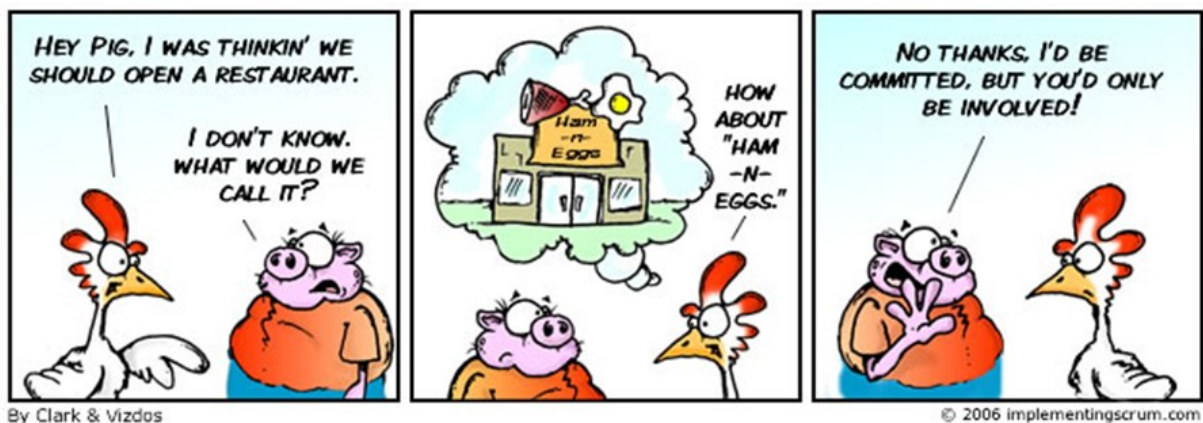
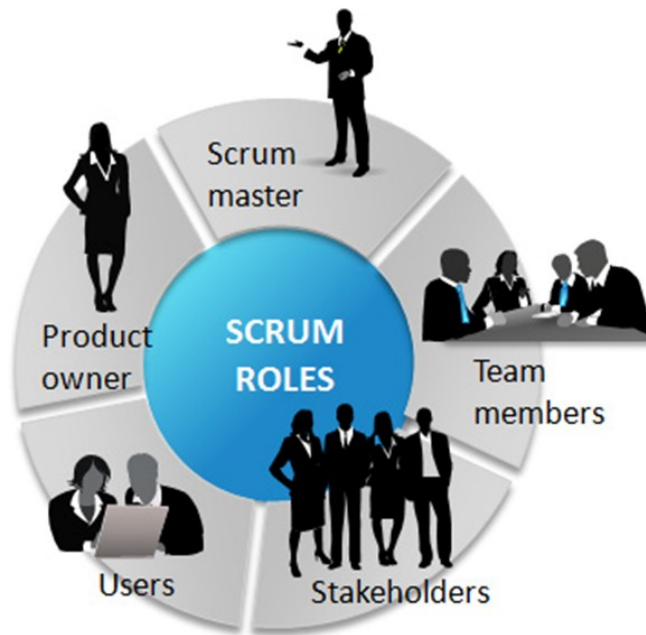


Figura 2: Chiste para explicar Scrum

Explicación del chiste de la figura 2: Un cerdo y una gallina se encuentran en la calle. La gallina mira al cerdo y dice: "Hey, ¿por qué no abrimos un restaurante?" El cerdo mira a la gallina y le dice: "Buena idea, ¿cómo se llamaría el restaurante?" La gallina piensa un poco y contesta: "¿Por qué no lo llamamos "Huevos con jamón?" "Lo siento pero no", dice el cerdo, "Tú sólo estarías involucrada mientras que yo estaría comprometido".

En el gráfico de la figura 3 podemos ver los roles que interactúan entre sí dentro del paraguas de Scrum.



*Figura 3: Roles en Scrum*

#### GALLINAS:

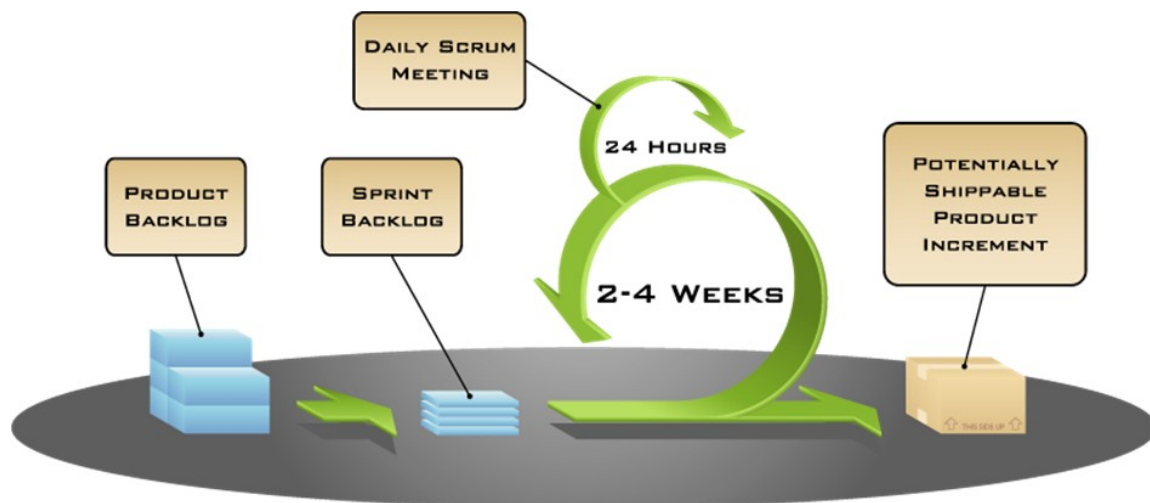
- **Usuarios**, destinatario final del producto.
- **Stakeholders**, la gente que hace posible el proyecto.
- **Managers**, gente que establece el ambiente para el desarrollo del producto.

#### CERDOS:

- **Team/Equipo de desarrollo**, tiene la responsabilidad de entregar el producto. Este equipo, formado principalmente por desarrolladores, puede contener una figura de UX (**User Experience**<sup>1</sup>) o diseñadores. Se comprometen en el Sprint a entregar valor al cliente una vez finalice este.
- **Product Owner**, representa la voz del cliente. Es el encargado de transmitir al equipo las historias de usuario que ha escrito gracias a las conversaciones que ha tenido con cliente y las prioriza en base a las necesidades de cliente.
- **Scrum Master**, eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. Es una figura importante cuando se comienza con Scrum, ya que se encarga de velar porque las normas y buenas prácticas de esta metodología se cumplan, poniendo al servicio del equipo las herramientas necesarias para facilitar el Sprint.

---

<sup>1</sup>La experiencia de usuario o User Experience es el conjunto de factores y elementos relativos a la interacción del usuario, con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Figura 4: Diagrama resumen del flujo Scrum

Durante el flujo de Scrum hay muchas reuniones. El flujo se puede ver en la figura 4. Paso a describir las reuniones que se deben realizar:

- **Daily Scrum Meeting:** reunión diaria en la que participan los miembros del equipo, principalmente de pie, de duración menor a 10 minutos. Se pretende conseguir que se pongan sobre la mesa las tareas que ha realizado cada miembro del equipo y detectar si alguno se ha encontrado con algún problema en su desarrollo.
- **Sprint planning:** reunión al principio de cada Sprint donde el Product Owner lleva el Product Backlog para que el equipo estime las historias de usuario y determine el número de historias que entran a formar parte de ese Sprint planning. Es fundamental que al finalizar el Sprint se entreguen funcionalidades cerradas que el cliente final pueda probar.
- **Sprint Review Meeting:** el equipo se reúne con el Product Owner una vez que ha finalizado el Sprint para hacerle una demo de las historias de usuario que formaban parte del Sprint.
- **Sprint Retrospective:** reunión en la que el equipo evalúa el sprint que ha terminado con la intención de mejorar en los fallos detectados añadiendo técnicas y medidas para que no se vuelvan a repetir.

A mayores, en Brooktec, realizan una reunión inicial de captura de requisitos con el cliente, que ayuda a ambas partes a definir y conceptualizar mejor la idea base del proyecto en curso. Esta reunión recibe el nombre de **Inception**

Desde la empresa mantienen y defienden esta metodología por estos motivos:

- El cliente fija sus expectativas al indicar el valor que le aporta cada requisito del proyecto y cuando espera que esté completado.
- El cliente podría utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo.

- De manera regular el cliente puede redirigir el proyecto según las nuevas prioridades, o según los cambios en el mercado.
- Las personas trabajan más enfocadas y de manera más eficiente cuando hay una fecha límite a corto plazo para entregar un resultado al que se han comprometido.
- La estimación de esfuerzo y la optimización de tareas para completar un requisito es mejor si la realizan las personas que van a desarrollar el requisito, dadas sus diferentes especializaciones, experiencias y puntos de vista.
- Las personas están más motivadas cuando pueden usar su creatividad para resolver problemas y cuando pueden decidir organizar su trabajo.

## Kanban

Kanban no es una técnica específica del desarrollo software, su objetivo es gestionar de manera general cómo se van completando tareas, pero en los últimos años se ha utilizado en la gestión de proyectos de desarrollo software, a menudo con Scrum (lo que se conoce como Scrumban).

Kanban es una palabra japonesa que significa algo parecido a “tarjetas visuales” (*kan* significa visual, y *ban* tarjeta). Esta técnica se creó en Toyota, y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción. Kanban está dentro de la estrategia *Kaizen*, es decir, la mejora continua y continuada. Se puede ver un ejemplo de la tabla Kanban en la figura 5.

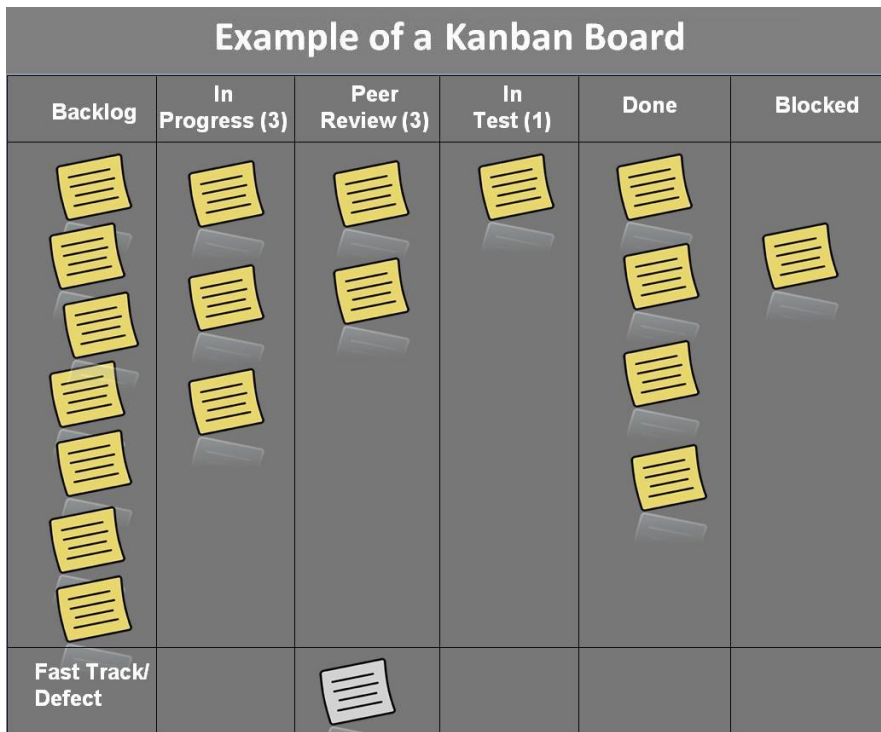


Figura 5: Ejemplo de tablero Kanban

Los principios de Kanban son:

- **Calidad garantizada.** Todo lo que se hace debe salir bien a la primera, no hay margen de error. De aquí a que en Kanban no se premie la rapidez, sino la calidad final de las tareas realizadas. Esto se basa en el hecho que muchas veces cuesta más arreglarlo después que hacerlo bien a la primera.
- **Reducción del desperdicio.** Kanban se basa en hacer solamente lo justo y necesario, pero hacerlo bien. Esto supone la reducción de todo aquello que es superficial o secundario (principio YAGNI).
- **Mejora continua.** Kanban no es simplemente un método de gestión, sino también un sistema de mejora en el desarrollo de proyectos, según los objetivos a alcanzar.
- **Flexibilidad.** Lo siguiente a realizar se decide del backlog (o tareas pendientes acumuladas), pudiéndose priorizar aquellas tareas entrantes según las necesidades del momento (capacidad de dar respuesta a tareas imprevistas).

Las reglas que se deben respetar en kanban son:

- **Visualizar el trabajo** en Kanban y las fases del ciclo de producción, o flujo de trabajo.
- Determinar el límite de “**trabajo en curso**” (Work in progress): el número máximo de tareas que se pueden realizar en cada fase debe ser algo conocido.
- **Medir el tiempo** en completar una tarea: El “lead time” cuenta desde que se hace una petición hasta que se hace la entrega.

## Scrumban

La metodología que utiliza Brooktec está basada en Scrumban. Utilizan el panel kanban para poder visualizar los sprints, tanto los miembros del equipo de desarrollo como el Product Owner y el Scrum Master.

Se reúnen en la daily alrededor de ese panel para comentar las historias de usuario en las que está cada miembro del equipo y se intenta detectar un cuello de botella o problema no contemplado en la primera reunión de Sprint planning.

## Tecnologías

Este apartado describe las tecnologías que se utilizan durante el desarrollo, como Vagrant, así como las que se integran en la aplicación, como Slim. Para cada tecnología se explica qué es, cómo se instala y su utilización.

### Vagrant



*Figura 6: Logotipo de Vagrant*

Vagrant es una herramienta software que crea y configura entornos de desarrollo virtualizados. Puede verse como un envoltorio de alto nivel que embarca herramientas de

virtualización software como VirtualBox, VMWare, KVM, LXC, Docker, etc., y herramientas de software de gestión de configuración como Ansible, Chef, Salt, Puppet, etc.

Originalmente se desarrolló para VirtualBox y sistemas de configuración tales como Chef, Salt y Puppet. Sin embargo desde la versión 1.1 Vagrant es capaz de trabajar con múltiples proveedores, como VMware, Amazon EC2, LXC, DigitalOcean, etc.

Desde la versión 1.6, Vagrant soporta de forma nativa contenedores Docker, que en algunos casos puede servir como un sustituto de un completo sistema operativo virtualizado.

Aunque Vagrant se ha desarrollado en Ruby se puede usar en multitud de proyectos escritos en otros lenguajes, tales como PHP, Python, Java, C# y JavaScript.

## **Instalación**

Para este proyecto se va a utilizar el software de virtualización VirtualBox. Como gestión de configuración un script en bash shell preconfigurado.

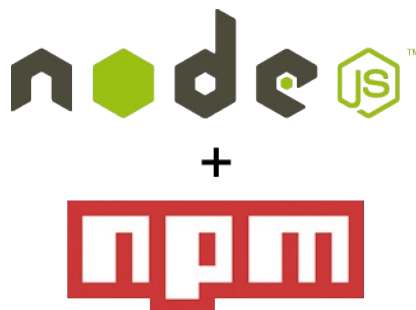
Hay que acceder a la página web de descarga <https://www.vagrantup.com/downloads.html> donde está el instalador para los distintos sistemas operativos: .exe, .dmg, .deb y .rpm. También se puede instalar desde el gestor de paquetes del sistema operativo.

Como se va a utilizar con VirtualBox, hay que tener este programa instalado en la máquina.

## **Utilización**

Ejecutando el comando “vagrant init” nos crea un fichero “Vagrantfile” con la configuración por defecto de Vagrant. Para instalar e iniciar la máquina se ejecuta “vagrant up” y arranca la máquina en segundo plano. Cuando hemos terminado se ejecuta “vagrant halt” para parar la máquina y que no consuma recursos, ya que es una máquina virtual. Y para borrarla por completo se ejecuta “vagrant destroy”.

## **Node.js y npm**



*Figura 7: Logotipo de Node.js y npm*

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado

en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de C, EventMachine de Ruby, vibe.d de D y JEE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS. Node.js incluye un entorno REPL para depuración interactiva.

Tiene un gestor de paquetes llamado npm, Node Package Manager, similar a PEAR para PHP, CPAN para Perl y pip para Python.

### ***Instalación***

Hay que acceder a la página web de descarga <https://nodejs.org/en/download/> donde está el instalador para los distintos sistemas operativos: .exe, .dmg y .deb, entre otros. También se puede instalar desde el gestor de paquetes del sistema operativo.

### ***Utilización***

En este proyecto no se va a utilizar Node.js sino su gestor de paquetes porque tiene muchas utilidades software que facilitan el desarrollo de aplicaciones, como por ejemplo Gulp que se verá a continuación.

Para instalar un paquete hay que ejecutar “npm install <paquete>”. En caso de que se quiera guardar los paquetes para acompañarlo a un proyecto hay que utilizar un fichero llamado “package.json” donde se incluirán los paquetes instalados y su versión. Si en una carpeta que contiene el fichero “package.json” se ejecuta “npm install” se descargarán e instalarán los paquetes listados en la versión que aparece.

## **Gulp**



*Figura 8: Logotipo de gulp*

Gulp se define como “The streaming build system” o sistema de compilado continuo. Sus principales ventajas son:

Automatización: es una herramienta que te ayuda a automatizar tareas dolorosas o que consumen mucho tiempo en tu flujo de trabajo.

Angóstico de plataforma: Se integra con los IDEs más usados y la gente usa gulp con PHP, .NET, Node.js, Java, y otras plataformas.

Ecosistema sólido: Usa los módulos de npm para hacer lo que quieras y tiene más de 2000 plugins comprobados para transformaciones de ficheros.

Simple: Como provee una API pequeña, gulp es fácil de aprender y simple de usar.

## **Instalación**

Como es un paquete creado para Node.js se puede utilizar el gestor de paquetes npm para instalarlo de forma sencilla. En una máquina con Node.js instalado sólo hay que ejecutar “npm install gulp” para que se descargue y se pueda empezar a utilizar. El directorio donde se instala es “node\_modules/gulp” y para ejecutarlo “node\_modules/gulp/bin/gulp.js” pero si se quiere instalar en la máquina para que sea accesible desde cualquier ubicación hay que instalarlo ejecutando “npm install -g gulp” (es posible que requiera permisos de administrador).

## **Utilización**

Gulp se basa en tareas que se ejecutan de forma secuencial o paralela. Por ejemplo se puede tener una lista de ficheros JavaScript que si gulp está en modo desarrollo ejecute unos linters o analizadores de código que muestran los fallos o errores que hay en el código. Por otro lado, si está en modo producción se comprimen, se quitan los comentarios y se concatenan todos los javascript en uno para agilizar la carga de la web.

A continuación se puede ver la tarea “images” en gulp:

```
var gulp = require('gulp'),
    plugins = require('gulp-load-plugins')();

gulp.task('images', () => {
  return gulp.src("img/**/*.*")
    .pipe(plugins.changed("images-min"))
    .pipe(plugins.imagemin())
    .pipe(plugins.size({ title: 'images' }))
    .pipe(gulp.dest("images-min"));
});
```

Esta tarea lee (*gulp.src*) todos los ficheros y subcarpetas de la carpeta “img” que hayan sido modificados (*plugins.changed*). Después se pasa por un programa que hace más pequeñas las imágenes sin perder calidad (*plugins.imagemin*) ya que en internet el tamaño



de las cosas que se descargan es mejor cuanto menos ocupe. Tras haber hecho más pequeñas las imágenes, se muestra por pantalla el tamaño de todas las imágenes (*plugins.size*) y las guarda (*gulp.dest*) en la carpeta “images-min”.

Para lanzar una tarea con gulp basta con ejecutar “gulp <tarea>”, en este caso “gulp images” y tendríamos en la carpeta “images-min” las imágenes con un tamaño menor de forma automática y simple.

## Slim

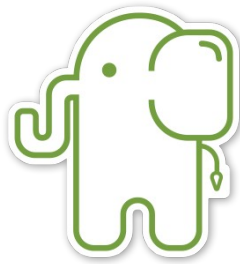


Figura 9: Logotipo de Slim

Slim es un micro framework de PHP que te ayuda a crear rápidamente simple pero robustas aplicaciones web y APIs. En esencia Slim es un despachador que recibe una petición HTTP, invoca la rutina apropiada y devuelve una respuesta HTTP.

En las peticiones y respuestas se trata con objetos. Esos objetos representan la petición HTTP recibida por el servidor web y la respuesta HTTP que se enviará al cliente. Estos objetos implementan la interfaz de PSR 7.

### Instalación

Para descargar Slim sólo hace falta tener instalado PHP y Composer en la máquina. Dado que se va a desarrollar desde un entorno virtualizado y configurado con PHP y Composer instalado, sólo hay que ejecutar “composer require slim/slim”. Este comando descarga la última versión, en este caso la versión 3. Como para este proyecto se necesita la versión 2 de Slim en lugar del comando anterior hay que ejecutar “composer require slim/slim:~2.0”.

### Utilización

La base de Slim es que siempre es el mismo punto de entrada, el fichero que ejecuta “\$app->run()”. El siguiente código es un ejemplo de aplicación muy sencilla de Slim:

```
$app = new \Slim\Slim();
$app->get('/hello/:name', function ($name) {
    echo "Hello, " . $name;
});
$app->run();
```

Suponiendo que ese fichero se llama “index.php” y que el servidor web redirige todo el tráfico a ese fichero, si se accede a la url “/hello/world” aparecerá por pantalla “Hello, world”.

## Twig



Figura 10: Logotipo de Twig

Twig es un moderno motor de vistas para PHP. Es flexible, ya que compila el código PHP al mínimo y lo optimiza permitiendo el uso de vistas en caché; es seguro gracias al modo “caja de arena” o “sandbox” donde se evalúa el código inseguro y permite que los usuarios modifiquen el diseño de plantillas; y es flexible, porque permite al desarrollador definir sus etiquetas y filtros personalizados, además de crear su propio lenguaje de dominio específico.

Mientras que mucha gente puede defender que PHP ya es un motor de vistas en sí, Twig se creó con ese único propósito mientras que el lenguaje PHP ha ido evolucionando y añadiendo características, manteniendo el motor de vistas, pero sin dedicarse exclusivamente en él. Por ello Twig es una solución muy recomendable ya que permite separar el desarrollo a nivel de servidor del nivel de vista.

### Instalación

Como está escrito en PHP se puede utilizar su gestor de paquetes Composer para instalarlo. El comando que hay que ejecutar es “composer require twig/twig:~1.0” y se descarga e instala la última versión. Si se prefiere se puede descargar el código de un fichero comprimido, desde git, PEAR, etc. aunque la forma recomendada es a través de Composer.

### Utilización

Para utilizar Twig hay que importarlo. Después se crean las plantillas que utiliza la aplicación y se agrupan en un cargador. A la hora de querer mostrar cierta plantilla hay que renderizar identificando la plantilla que se quiere utilizar y los datos que se le pasan como argumentos.

```
$loader = new Twig_Loader_Array(array(
    'index' => 'Hello {{ name }}!',
));
$twig = new Twig_Environment($loader);
```

```
echo $twig->render('index', array('name' => 'World!'));
```

En el anterior código se crea una plantilla identificada por “index” que contiene el literal “Hello” y una variable “name”. Tras crear el objeto Twig e imprimir por pantalla el objeto renderizado se muestra “Hello World!” dado que a la hora de renderizar la plantilla se utilizó el literal “World!” como contenido de la variable “name”.

## Doctrine



*Figura 11: Logotipo de Doctrine*

Doctrine es un proyecto con muchas librerías PHP principalmente centradas en el almacenamiento en base de datos y mapeo de objetos. Los proyectos principales son el mapeo objeto-relacional u ORM y la capa de abstracción de bases de datos o DBAL.

En este proyecto se ha utilizado el proyecto de ORM para la gestión de la base de datos. Dado que se ha utilizado el patrón ORM el uso de la base de datos ha sido muy sencillo y fácil de implementar gracias a la API proporcionada por el proyecto de Doctrine ORM.

### **Instalación**

Como Doctrine ORM es una librería para PHP muy extendida se puede instalar desde el gestor de paquetes Composer. Para instalarlo sólo hace falta ejecutar el comando “composer require doctrine/orm”.

### **Utilización**

Esta librería requiere de una base de datos en funcionamiento y un driver de conexión entre PHP y dicha base de datos. La clase principal de Doctrine ORM es el “EntityManager” que gestiona las entidades registradas en la aplicación e interactúa con la base de datos en función de las propiedades de dichas entidades.

Una vez que el EntityManager obtiene de la base de datos la información de una entidad lo transforma a objeto de PHP y permite la manipulación de los datos con código PHP simple así como la creación o actualización de esos datos.

## Angular2



*Figura 12: Logotipo de Angular2*

Angular2 es un framework para el desarrollo de aplicaciones móvil y escritorio sin interacción de servidor. La aplicación se ejecuta en el navegador del cliente con lenguaje JavaScript. Tiene una primera versión que es AngularJS, creada por desarrolladores de Google, y muy extendida por la comunidad de desarrolladores de aplicaciones web.

A diferencia de otras tecnologías, Angular2 es una mezcla de muchas de ellas permitiendo desarrollar en múltiples lenguajes como TypeScript, Dart, JavaScript entre otros. Y no solo es heterogéneo con los lenguajes, lo mismo ocurre con los estándares, siendo los más utilizados SystemJS, CommonJS, ECMAScript5 y ECMAScript6.

### **Instalación y utilización**

La página oficial tiene una guía tutorial para comenzar a desarrollar y aprender esta tecnología desde 0. Este tutorial utiliza TypeScript, CommonJS y ECMAScript5 y se puede acceder desde esta URL: <https://angular.io/docs/ts/latest/quickstart.html>. La instalación que recomiendan es a través del gestor de paquetes de Node.js y descargar el paquete de configuración “package.json” de la guía oficial de Angular2.

Al principio cuesta mucho entender cómo funciona Angular2 si no se ha utilizado ninguna de las tecnologías que lo componen, ya que hay que aprender cada una de esas tecnologías para aprender a utilizar Angular2. El código final HTML, CSS JavaScript que se puede ver en cualquier navegador. El JavaScript modifica la estructura DOM del HTML de forma completamente dinámica y sin intervención de servidor externo. Por ello con servir los ficheros HTML, CSS y JavaScript desde un servidor permite utilizar cualquier aplicación Angular2.

### **Entornos**

En este apartado se explican los entornos que se utilizan en Brooktec durante el desarrollo de cada proyecto. Son tres los distintos entornos y cada uno tiene un propósito diferente. Estos tres entornos son:

- Entorno de desarrollo local
- Entorno de pruebas
- Entorno de producción

¿Qué es lo que diferencia a cada entorno? En el entorno de desarrollo local es necesario que la aplicación guarde en el log todos los detalles de la aplicación para simplificar la depuración, pero la caché tiene que estar deshabilitada para que cualquier cambio realizado se tenga en cuenta de forma instantánea. Por tanto, el entorno de desarrollo se debe optimizar para el programador. El ejemplo más claro es cuando se produce una excepción, el entorno de desarrollo muestra la traza de la excepción, qué la produce, los datos y más información relativa a ésta.

Por otra parte, en el entorno de producción la caché debe estar habilitada y por supuesto se deben mostrar mensajes de error propios en vez de la información relacionada con la excepción producida. Por tanto, el entorno de producción debe estar optimizado para obtener el máximo rendimiento y para conseguir la mejor experiencia de usuario.

El entorno de pruebas debe ser idéntico al entorno de producción, o ser el mismo servidor pero distinta ruta, ya que se probará internamente la aplicación para comprobar que funciona correctamente y puede ser abierta al público. En este entorno el cliente puede ver el resultado del trabajo del equipo y comprobar si se está haciendo lo esperado o por el contrario se tienen que hacer los cambios necesarios.

## **Entorno de producción**

El entorno de producción es un servidor GNU/Linux Ubuntu 14.04.4 Trusty con la arquitectura 64 bits. Tiene como servidor web Apache con la versión 2.4.7 junto a PHP versión 5.5.9 y el sistema gestor de bases de datos MySQL versión 5.5.

## **Entorno de pruebas**

Como se ha dicho anteriormente el entorno de pruebas debe ser idéntico. En este caso al compartir máquina con el entorno de producción las características del servidor son las mismas.

En la figura 13 se puede ver el diagrama del entorno de producción y de pruebas.

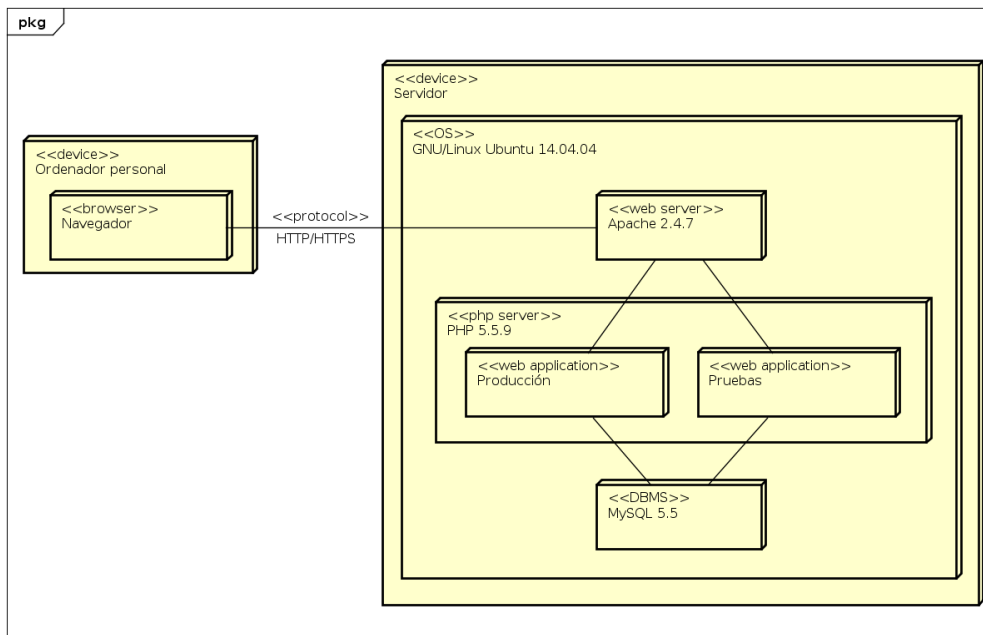


Figura 13: Entornos de producción y pruebas

## Entorno de desarrollo local

En el caso del entorno de desarrollo local habrá distintas máquinas físicas donde se desarrolle. Por esa razón y para tratar de reproducir el entorno de producción de la forma más fiel posible se virtualiza un servidor GNU/Linux Ubuntu 14.04.4 con la arquitectura 32 bits, con servidor web Apache versión 2.4.7 junto a PHP versión 5.5.9 y el sistema gestor de bases de datos MySQL versión 5.5. No se ha elegido la versión para la arquitectura de 64 bits porque no todas las máquinas desde las que se desarrolla son capaces de virtualizar esa arquitectura.

Para crear el entorno de desarrollo local de forma sencilla y automática se utiliza el software "Vagrant" que permite crear una máquina virtual con "VirtualBox" y prepararla con un fichero de instalación previamente configurado para que instale las versiones de Apache, PHP y MySQL necesarias sin intervención del usuario. Con esto se consigue que sea cual sea la máquina desde la que se desarrolle tenga el entorno instalado y configurado en poco tiempo y sin esfuerzo.

En la figura 14 se puede ver el entorno de desarrollo local.

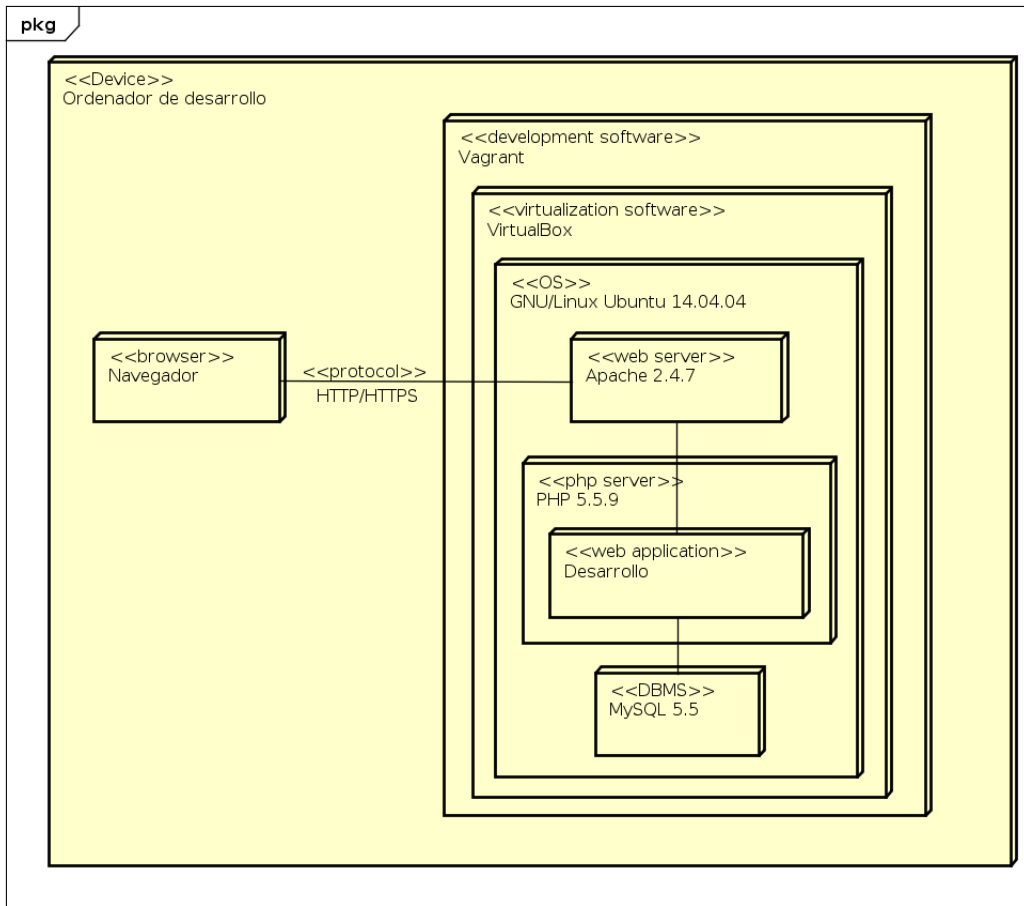


Figura 14: Entorno de desarrollo

## Capítulo 3. Análisis

En el presente capítulo se analiza la información proporcionada por el cliente y se realiza la primera reunión de la planificación de iteraciones. En esta primera reunión se definen los siguientes datos:

- Glosario de términos
- Roles
- Necesidades de cliente (Historias de usuario)
- Especificaciones técnicas
- Tecnología a utilizar para el desarrollo del proyecto
- Riesgos

### Glosario de términos

- **Promoción:** Agrupación de ofertas de una empresa/organización.
- **Oferta:** Promoción de un producto con un iframe de compra asociado.
- **Usuario de la aplicación:** Usuario que puede hacer uso de la aplicación web tras registrarse satisfactoriamente.
- **Usuario registrado:** Usuario que se ha registrado desde una promoción para acceder a las ofertas que tiene dicha promoción. El usuario registrado no puede hacer uso de la aplicación web.

### Roles

- **Usuario final:** persona que hará uso de un descuento a través de una promoción.
- **Gestor:** persona que puede crear, editar y borrar promociones y ofertas y puede ver los usuarios que se han registrado a través de las promociones.
- **Técnico:** persona con un perfil más técnico que es capaz de realizar las mismas funciones que “gestor” pero que además puede descargarse el código de la promoción para replicarla.
- **Superadmin:** persona que puede realizar las mismas funciones que “técnico” pero que además puede crear, editar y borrar usuarios en la aplicación.

### Historias de usuario



### Historia de usuario US-0001

Como superadmin, quiero crear/ver/modificar/eliminar usuarios de cualquier rol que permita el sistema. Cada usuario tendrá los siguientes campos:

- Correo electrónico
- Contraseña
- Rol

#### **Criterios de aceptación:**

- Al crear un nuevo usuario, introducir los datos correctamente y ver que aparece en el listado.
- Al crear un nuevo usuario, introducir un correo electrónico incorrecto y ver que no aparece en el listado.
- Al actualizar un usuario, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar un usuario, introducir un correo electrónico incorrecto y ver que no ha actualizado sus datos.
- Al borrar un usuario, ver que no aparece en el listado.

### Historia de usuario US-0002

Como gestor quiero poder crear/ver/modificar/eliminar promociones. Cada promoción tendrá los siguientes campos:

- Nombre
- Descripción
- Imagen
- Estado: activa u oculta
- Campos del formulario de registro
- Campos del formulario de inicio de sesión

#### **Criterios de aceptación:**

- Al crear una nueva promoción, introducir los datos correctamente y ver que aparece en el listado.
- Al crear una nueva promoción, dejar en blanco cualquier campo y ver que no aparece en el listado.
- Al actualizar una promoción, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar una promoción, dejar en blanco cualquier campo y ver que no ha actualizado sus datos.
- Al borrar una promoción, ver que no aparece en el listado.

### Historia de usuario US-0003

Como gestor quiero poder crear/ver/modificar/eliminar códigos promocionales de las promociones si el campo del formulario de registro está activo. Cada código promocional tendrá los siguientes campos:

- Código

#### **Criterios de aceptación:**

- Al crear un nuevo código promocional, introducir los datos correctamente y ver que aparece en el listado.
- Al crear un nuevo código promocional, dejar en blanco el código y ver que no aparece en el listado.
- Al actualizar un código promocional, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar un código promocional, dejar en blanco el código y ver que no ha actualizado sus datos.
- Al borrar un código promocional, ver que no aparece en el listado.

### Historia de usuario US-0004

Como gestor quiero poder crear/ver/modificar/eliminar ofertas. Cada oferta tendrá los siguientes campos:

- Nombre
- Descripción
- Imagen
- Etiqueta
- Código del video de youtube
- Ciudades
- Testimonios

#### **Criterios de aceptación:**

- Al crear una nueva oferta, introducir los datos correctamente y ver que aparece en el listado.
- Al crear una nueva oferta, dejar en blanco el nombre y ver que no aparece en el listado.
- Al actualizar una oferta, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar una oferta, dejar en blanco el nombre y ver que no ha actualizado sus datos.
- Al borrar una oferta, ver que no aparece en el listado.

### Historia de usuario US-0005

Como gestor quiero poder crear/ver/modificar/eliminar ciudades de las ofertas. Cada ciudad tendrá los siguientes campos:

- Nombre
- Descripción
- URL del iframe
- URL del iframe para dispositivos móviles

#### **Criterios de aceptación:**

- Al crear una nueva ciudad, introducir los datos correctamente y ver que aparece en el listado.
- Al crear una nueva ciudad, dejar en blanco el nombre, la descripción o la URL del iframe y ver que no aparece en el listado.
- Al actualizar una ciudad, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar una ciudad, dejar en blanco el nombre, la descripción o la URL del iframe y ver que no ha actualizado sus datos.
- Al borrar una ciudad, ver que no aparece en el listado.

### Historia de usuario US-0006

Como gestor quiero poder crear/ver/modificar/eliminar testimonios de las ofertas. Cada testimonio tendrá los siguientes campos:

- Testimonio
- Autor

#### **Criterios de aceptación:**

- Al crear un nuevo testimonio, introducir los datos correctamente y ver que aparece en el listado.
- Al crear un nuevo testimonio, dejar en blanco cualquier campo y ver que no aparece en el listado.
- Al actualizar un testimonio, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar un testimonio, dejar en blanco cualquier campo y ver que no ha actualizado sus datos.
- Al borrar un testimonio, ver que no aparece en el listado.

### Historia de usuario US-0007

Como gestor quiero poder ver los usuarios que se han registrado a cada promoción. Cada usuario registrado podrá tener los siguientes campos, siendo todos opcionales:

- Nombre
- Apellidos
- Correo electrónico
- Provincia
- Código promocional
- Promoción

#### **Criterios de aceptación:**

- Desde el panel de administración, en la opción de usuarios registrados, veo el listado de los usuarios que se han registrado y a qué promoción.

### Historia de usuario US-0008

Como usuario que inicia sesión en la parte de administración (gestor, técnico, superadministrador) quiero que para poder acceder a la parte de administración tenga que meter mi correo electrónico y contraseña.

#### **Criterios de aceptación:**

- Introducir el correo electrónico incorrecto y comprobar que indica error.
- Introducir la contraseña incorrecta y comprobar que indica error.
- Introducir el correo electrónico y la contraseña correctos y comprobar que inicia sesión.

### Historia de usuario US-0009

Como usuario que inicia sesión en la parte de administración (gestor, técnico, superadministrador) quiero poder modificar mi email y/o contraseña en cualquier momento.

#### **Criterios de aceptación:**

- Introducir un email inválido y comprobar que indica error.
- Introducir un email correcto y comprobar que actualiza el email.
- Introducir una contraseña inválida y comprobar que indica error.
- Introducir una contraseña correcta y comprobar que actualiza la contraseña.

### Historia de usuario US-0010

Como usuario final, quiero registrarme en una promoción para disfrutar de los descuentos asociados.

#### Criterios de aceptación:

- Rellenar algún campo incorrectamente y comprobar que se muestra error.
- Rellenar todos los campos correctamente y comprobar que se identifica correctamente.

### Historia de usuario US-0011

Como usuario final, quiero identificarme en una promoción para disfrutar de los descuentos asociados.

#### Criterios de aceptación:

- Rellenar algún campo incorrectamente y comprobar que se muestra error.
- Rellenar todos los campos correctamente y comprobar que se identifica correctamente.

### Historia de usuario US-0012

Como gestor quiero que una oferta tenga los campos de otra vinculada si no relleno los datos.

#### Criterios de aceptación:

- Crear una oferta sin todos los datos vinculando otra y comprobar que devuelve toda la información a través de la API.
- Crear una oferta sin todos los datos y sin vincular otra y comprobar que no se crea.

### Historia de usuario US-0013

Como técnico quiero obtener la información de una promoción a través de una API proporcionando el id de la promoción.

#### Criterios de aceptación:

- Realizando la petición a la API, proporciono un id inválido y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y veo que devuelve la información de la promoción.

### Historia de usuario US-0014

Como técnico quiero poder registrar usuarios a una promoción a través de una API proporcionando el id de la promoción y los campos del usuario que requiere dicha promoción para registrarlo.

#### Criterios de aceptación:

- Realizando la petición a la API, proporciono un id inválido y datos incorrectos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id inválido y datos correctos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y datos incorrectos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y veo que devuelve la información de las ofertas asociadas a la promoción.

### Historia de usuario US-0015

Como técnico quiero poder identificar usuarios ya registrados a una promoción a través de una API proporcionando el id de la promoción y los campos del usuario que requiere dicha promoción para identificarlo.

#### Criterios de aceptación:

- Realizando la petición a la API, proporciono un id inválido y datos incorrectos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id inválido y datos correctos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y datos incorrectos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y veo que devuelve la información de las ofertas asociadas a la promoción.

### Historia de usuario US-0016

Como técnico quiero poder tener un código que haga uso de la API.

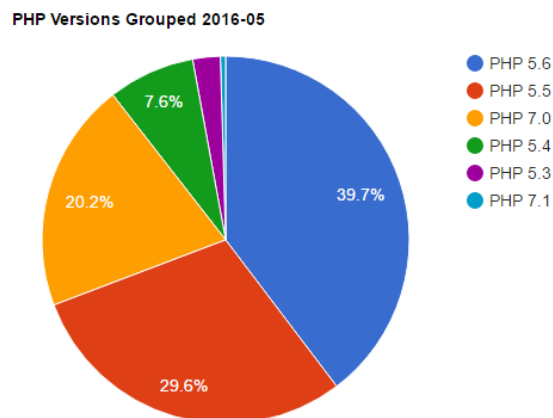
#### Criterios de aceptación:

- Probar el código y comprobar que los datos que aparecen son los mismos que los que hay en la promoción dentro del gestor.

<b>Historia de usuario US-0017</b>
Como gestor quiero poder vincular las ofertas a las promociones.
<p><b>Criterios de aceptación:</b></p> <ul style="list-style-type: none"> <li>Desde el panel de administración, en una promoción, vinculo al menos una oferta y al guardar compruebo que tiene las ofertas vinculadas.</li> <li>Desde el panel de administración, en una promoción, desvinculo todas las ofertas y al guardar compruebo que no tiene ninguna oferta vinculada.</li> </ul>

## Especificaciones técnicas

Para este proyecto se han impuesto las siguientes especificaciones técnicas por parte del cliente. La primera de todas es que debe ser compatible con servidores que tengan una versión de PHP 5.4.0 como mínimo. Esto asegura que la aplicación desarrollada es compatible con aproximadamente el 97,1% de los servidores con PHP instalado. En la figura 15 se puede ver la tasa de utilización de las distintas versiones de PHP a mayo de 2016.



*Figura 15: Fragmentación de PHP en mayo de 2016*

Además se debe visionar correctamente en los navegadores más utilizados en su última versión ya que al ser una aplicación web se espera que su funcionamiento y visualización sean correctos y la misma para todos los navegadores. Los navegadores más utilizados son: Google Chrome, Internet Explorer 11, Microsoft Edge, Mozilla Firefox y Safari. También ha de verse correctamente independientemente del sistema operativo desde el que se acceda, ya sea Windows, Mac OS o GNU/Linux.

Para el desarrollo de la aplicación se tiene que utilizar frameworks para facilitar el desarrollo ya que facilita el desarrollo de ciertas tareas como la creación de la API o la generación de las vistas para el usuario.

## Tecnología a utilizar

En este punto hay 3 partes diferenciadas de la aplicación. Por un lado está la aplicación web que es el gestor de las promociones y que utilizan el superadmin, el técnico y el gestor; por otro está la página web que se muestra al usuario final y desde la que se hace uso de las ofertas y entre medias está la API web que sirve la información del gestor de promociones.

### Aplicación web

Para la aplicación web hay que utilizar un gestor de rutas, un gestor de vistas y un gestor de base de datos. El gestor de rutas que se utilizará es el framework de PHP Slim 2 ya que es muy simple de configurar y utilizar. Aunque existe la versión 3 de este framework, no se puede utilizar porque la aplicación web tiene que ser compatible con la versión 5.4 de PHP y Slim 3 requiere la versión de PHP 5.5 o superior. Entre los beneficios de la versión 3 frente a la 2 se encuentran la implementación nativa del modelo-vista-controlador (MVC), la rapidez, mayor facilidad de desarrollo y nuevas funcionalidades.

Como la aplicación no tiene una gran cantidad de funciones la rapidez no difiere mucho de la versión 3. Ocurre lo mismo con la facilidad de desarrollo. En cambio para la implementación del MVC en Slim 2 sí hay que añadir un poco de complejidad a la aplicación pero es sencillo y se puede utilizar en Slim 2 el MVC.

Para la gestión de vistas se utiliza Twig, un motor de vistas simple, rápido y seguro. La implementación de Twig con Slim es muy sencilla ya que el propio Slim ha incorporado un módulo para la gestión de vistas que permite a Twig incorporarse de forma muy simple. La versión de Twig que se utiliza es la 1.24.

Para la gestión de base de datos se utilizará el patrón mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas ORM) implementado con Doctrine 2 ORM. El uso de este ORM simplifica y delega la gestión de la base de datos tan solo siendo necesaria la definición de las entidades que utiliza la aplicación.

### API

Como Slim 2 es un gestor de rutas de forma muy sencilla y se implementa con Doctrine 2 ORM para acceder a la base de datos estas tecnologías juntas permiten realizar una API de manera sencilla. Por ello se puede simplificar la creación de la API y de la aplicación web al juntar ambas partes.

### Página web del usuario final

La página web tiene que recoger la información de cierta promoción a través de la API de la aplicación. Como la API es web hay que realizar peticiones HTTP/HTTPS para obtener su contenido. Por ello se ha elegido la tecnología en desarrollo Angular2 para la realización de la página web ya que permite realizar páginas web dinámicas con lógica en el navegador del



usuario.

## Riesgos

Scrum no suele tener un control de los riesgos desde el inicio del desarrollo sino cuando se originan, ya que al entregar al final de un sprint parte del producto ayuda a reducir y controlar los riesgos.

En cambio en este proyecto se va a utilizar la gestión de riesgos “**Risk Burndown Chart**” o “Gráfico de quema de riesgos”. Con este sistema se controla iteración a iteración el estado de los riesgos y se puede ver de forma sencilla gracias a los gráficos si está yendo todo correctamente.

Para cada riesgo hay que definir los siguientes campos:

- **Riesgo:** Descripción del riesgo en unas pocas líneas.
- **Probabilidad:** Probabilidad de que se dé el riesgo.
- **Pérdida:** Cantidad de tiempo que se pierde en solucionar el riesgo si éste se produce. Se puede representar con días o puntos de historia.
- **Exposición:** El producto de la probabilidad y la pérdida.

Una vez que se han definido para todos los riesgos se suma toda la exposición y ese es el valor inicial antes de comenzar a desarrollar la aplicación ya que al comenzar cada sprint hay que modificar la tabla porque hay que restar la fecha de caducidad por el tiempo que ha pasado. También se puede modificar la probabilidad del riesgo o añadir nuevos que no se hayan tenido en cuenta previamente o que puedan surgir a partir del sprint realizado.

Como en cada sprint la tabla se modifica, el valor de la exposición total también lo hace y ese valor es el que nos permite visualizar en forma de gráfico la trayectoria de los riesgos del proyecto. Idealmente el gráfico sería una línea descendente desde el valor de exposición inicial hasta el 0 en el último sprint. Si observamos que en sucesivas iteraciones el valor se mantiene por encima de esa línea hay que actuar en el siguiente sprint para solucionar los problemas que causan esa situación.

### Tabla de riesgos inicial

<b>Riesgo</b>	<b>Probabilidad</b>	<b>Pérdida (en días)</b>	<b>Exposición</b>
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	40%	10	4,00
Problemas al integrar en el servidor de pruebas o producción	80%	10	8,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	90%	15	13,50
Añadir nuevas funcionalidades	75%	20	15,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	50%	5	2,50
La tecnología utilizada se discontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
			<b>50,50</b>

Esta tabla se ha creado a partir de la tabla base de Brooktec para los proyectos web a excepción de “La tecnología utilizada se discontinúa o deja de recibir actualizaciones de seguridad”.

Ese riesgo se ha añadido porque se va a utilizar la versión 2 de Slim, ya que está la versión 3 desde el 7 de diciembre de 2015. Se ha puesto una probabilidad del 10% porque es poco probable que esto ocurra, ya que es una versión muy extendida y no es 100% compatible con la versión 3 ya que no soporta PHP 5.3.x y 5.4.x mientras que la versión 2 sí. Si se discontinuara habría que cambiar la tecnología, por eso se estima que 30 días serían necesarios para sustituir Slim 2 por otra.

## **Planificación**

La duración de un sprint en Scrum suele ser de 2 semanas o 1 mes. En este proyecto cada sprint va a ser de 2 semanas debido a que las historias de usuario no tienen mucha complejidad y no deben llevar demasiado tiempo cada una, por lo que un sprint de 1 mes no es necesario en este proyecto.

Por la envergadura del proyecto, las tareas a realizar, el uso de las tecnologías con las que se va a hacer y la similitud con otros proyectos similares la planificación inicial se estima que se puede realizar en 8 iteraciones. Comenzando el 29 de febrero debería estar completado el 19 de junio, permitiendo completar la memoria en el resto del mes de junio y julio.

Al final el resultado ha sido que se han necesitado 9 sprints para terminar el proyecto, terminándolo el día 26 de junio, una semana más tarde de lo planificado inicialmente.

## Capítulo 4. Descripción de los sprints

En este capítulo se va a describir cada uno de los sprints que se han llevado a cabo para realizar este proyecto. En cada sprint se diferencian 4 fases:

- **Análisis:** se realiza un pequeño análisis si fuera necesario para modificar las historias de usuario, cambiar el modelo, etc.
- **Riesgos:** se actualiza la tabla de riesgos para tener unos datos actualizados y comprobar que todo va según lo previsto.
- **Historias de usuario:** se eligen las historias de usuario que se van a realizar en el sprint.
- **Criterios de aceptación:** tras terminar una historia de usuario se comprueba realizando los criterios de aceptación para asegurar que se ha desarrollado correctamente.

### Sprint 1, del 29/2/2016 al 13/3/2016

#### Análisis

Al ser el primer sprint se esboza el modelo de datos con el que se va a trabajar durante el desarrollo. Esto se hace para que todos los miembros del equipo se hagan una idea de cómo va a estar organizado el proyecto. La figura 16 representa el modelo resultante.

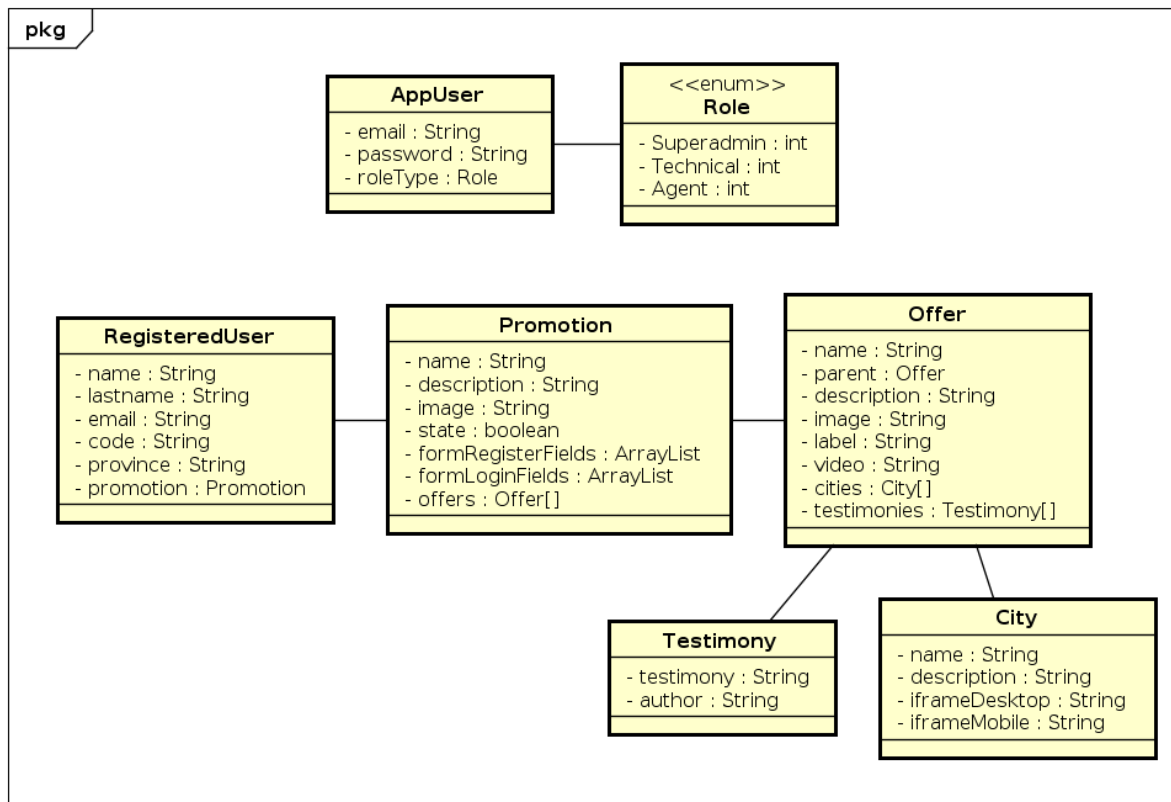


Figura 16: Modelo de datos

Una vez hecho el modelo de datos se eligen las historias de usuario del sprint backlog que el equipo se compromete a realizar. Al ser el primer sprint hay que configurar los entornos de producción, pruebas y desarrollo, comenzar y configurar con el desarrollo de las tecnologías del proyecto, por lo que se tiene que tener en cuenta que, aunque no sean historias de usuario es tiempo consumido por los equipos de desarrollo y de sistemas.

El sprint backlog es el siguiente:

- Historia de usuario US-0008: Como usuario que inicia sesión en la parte de administración (gestor, técnico, superadministrador) quiero que para poder acceder a la parte de administración tenga que meter mi correo electrónico y contraseña.
- Historia de usuario US-0009: Como usuario que inicia sesión en la parte de administración (gestor, técnico, superadministrador) quiero poder modificar mi email y/o contraseña en cualquier momento.
- Historia de usuario US-0001: Como superadmin, quiero crear/ver/modificar/eliminar usuarios de cualquier rol que permita el sistema.
- Historia de usuario US-0002: Como gestor quiero poder crear/ver/modificar/eliminar promociones.
- Historia de usuario US-0004: Como gestor quiero poder crear/ver/modificar/eliminar ofertas.

- Historia de usuario US-0017: Como gestor quiero poder vincular las ofertas a las promociones.
- Historia de usuario US-0013: Como técnico quiero obtener la información de una promoción a través de una API proporcionando el id de la promoción.
- Historia de usuario US-0014: Como técnico quiero poder registrar usuarios a una promoción a través de una API proporcionando el id de la promoción y los campos del usuario que requiere dicha promoción para registrarle.
- Historia de usuario US-0015: Como técnico quiero poder identificar usuarios ya registrados a una promoción a través de una API proporcionando el id de la promoción y los campos del usuario que requiere dicha promoción para identificarle.
- Historia de usuario US-0012: Como gestor quiero que una oferta tenga los campos de otra vinculada si no relleno los datos.
- Historia de usuario US-0016: Como técnico quiero poder tener un código que haga uso de la API.
- Historia de usuario US-0003: Como gestor quiero poder crear/ver/modificar/eliminar códigos promocionales de las promociones si el campo del formulario de registro está activo.
- Historia de usuario US-0005: Como gestor quiero poder crear/ver/modificar/eliminar ciudades de las ofertas.
- Historia de usuario US-0006: Como gestor quiero poder crear/ver/modificar/eliminar testimonios de las ofertas.
- Historia de usuario US-0010: Como usuario final, quiero registrarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0011: Como usuario final, quiero identificarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0007: Como gestor quiero poder ver los usuarios que se han registrado a cada promoción.

## Riesgos

En este sprint no hay que modificar la tabla de riesgos ya que es el primer sprint y la tabla de riesgos queda como descrita anteriormente. El gráfico resultante se puede ver en la figura 17.



*Figura 17: Risk burndown chart del Sprint 1*

## Historias de usuario

Al ser el inicio del proyecto y hay que configurar todo se elige sólomente una historia de usuario:

- Historia de usuario US-0008: Como usuario que inicia sesión en la parte de administración (gestor, técnico, superadministrador) quiero que para poder acceder a la parte de administración tenga que meter mi correo electrónico y contraseña.

Cuando se completa la historia de usuario se sube al entorno de pruebas para la demostración. Al subirlo aparecen problemas con las URLs que no se daban en el entorno de desarrollo. Esto es debido a que la ruta puesta en las redirecciones siempre comienza con '/' en vez de rutas relativas. Por eso, en vez de redirigir desde "www.example.com/tests" a "www.example.com/tests/pagina" que es lo que debería ocurrir, se estaba redirigiendo a www.example.com/pagina. Se ha solucionado en este sprint ya que arreglarlo no lleva mucho tiempo.

## Criterios de aceptación US-0008

- Introducir el correo electrónico incorrecto y comprobar que indica error.

The screenshot shows a login form with the title "Iniciar sesión". At the top, there is a red error message box that says "Usuario o contraseña incorrecto". Below this, there are two input fields: the first contains the email address "admin@example.com" and the second contains five dots representing a password. At the bottom of the form is a blue button labeled "Iniciar sesión".

Figura 18: US-0008: Criterio de aceptación 1

- Introducir la contraseña incorrecta y comprobar que indica error.

This screenshot is identical to Figure 18, showing the login form with the error message "Usuario o contraseña incorrecto" and the "Iniciar sesión" button.

Figura 19: US-0008: Criterio de aceptación 2

- Introducir el correo electrónico y la contraseña correctos y comprobar que inicia sesión.

The screenshot shows a dashboard titled "Gestor de Promociones" with a "Cerrar sesión" link in the top right. On the left is a navigation menu with items: "Resumen" (highlighted), "Promociones", "Ofertas", "Usuarios", and "Usuarios registrados". The main content area is titled "Resumen" and contains a table with the following data:

Promociones	2
Ofertas	2
Usuarios	2
Usuarios registrados	2

Figura 20: US-0008: Criterio de aceptación 3

## Sprint 2, del 14/3/2016 al 27/3/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Como no se ha modificado el sprint backlog ni se plantea modificar el modelo de datos se continúa con el mismo que el del sprint anterior.

### Riesgos

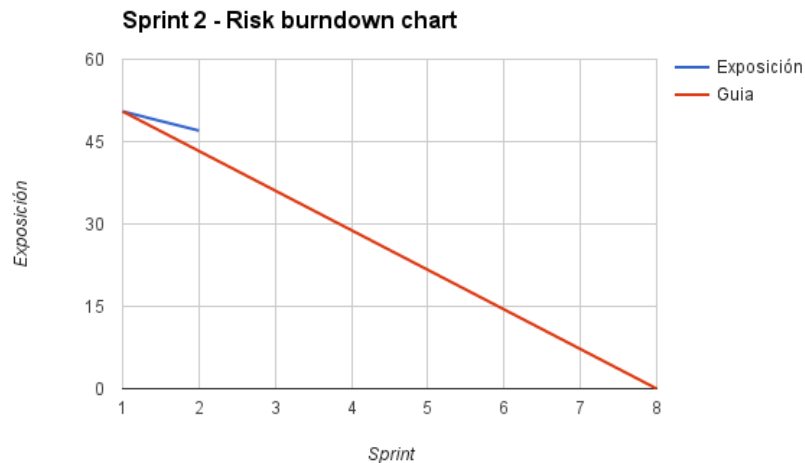
Tras el análisis hay que actualizar la tabla de gestión de riesgos:

Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	30%	10	3,00
Problemas al integrar en el servidor de pruebas o producción	60%	10	6,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	80%	15	12,00
Añadir nuevas funcionalidades	65%	20	13,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	50%	5	2,50
La tecnología utilizada se discontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
			<b>44,00</b>

Al subir la aplicación al servidor de pruebas se han producido algunos problemas por la URL y la forma de manejar las rutas de Slim. Como la URL base es del tipo “www.example.com/tests” no se había tenido en cuenta la ruta relativa y ha dado problemas que ya se han solucionado en el sprint anterior. Ese fue el único problema que dio el integrarlo en el servidor de pruebas. y por ello se reduce la probabilidad de ese riesgo.

Las probabilidades de los riesgos de “Fallo en el servicio de control de versiones”, “Cambios en las historias de usuario” y “Añadir nuevas funcionalidades” se reducen ya que según avanza el tiempo hay menos probabilidad de que ocurra. El gráfico resultante se puede ver en la figura 21.





*Figura 21: Risk burndown chart del Sprint 2*

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las dos siguientes:

- Historia de usuario US-0009: Como usuario que inicia sesión en la parte de administración (gestor, técnico, superadministrador) quiero poder modificar mi email y/o contraseña en cualquier momento.
- Historia de usuario US-0001: Como superadmin, quiero crear/ver/modificar/eliminar usuarios de cualquier rol que permita el sistema.

Al finalizar el sprint aunque se había comprometido a terminar la US-0001, no se ha podido completar la historia de usuario por completo. Queda modificar y eliminar los usuarios. Como no está terminada no se prueban los criterios de aceptación ni se quita del sprint backlog, eso se hará en el siguiente sprint.

## Criterios de aceptación US-0009

- Introducir un email inválido y comprobar que indica error.

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
**Usuarios**  
Todos los usuarios  
Añadir nuevo  
Mi perfil  
Usuarios registrados

Usuario: admin@example.com  
Registrado el 12/05/2016 23:39

Email \*

Introduzca una dirección de correo.

Contraseña

Repetir contraseña

Figura 22: US-0009: Criterio de aceptación 1

- Introducir un email correcto y comprobar que actualiza el email.

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
**Usuarios**  
Todos los usuarios  
Añadir nuevo  
Mi perfil  
Usuarios registrados

Usuario: admin-actualizado@example.com  
Registrado el 12/05/2016 23:39

Email \*

Contraseña \*

Repetir contraseña

Figura 23: US-0009: Criterio de aceptación 2

- Introducir una contraseña inválida y comprobar que indica error.

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
**Usuarios**  
Todos los usuarios  
Añadir nuevo  
Mi perfil  
Usuarios registrados

Usuario: admin@example.com  
Registrado el 12/05/2016 23:39

Email \*

Contraseña \*

Repetir contraseña

Las contraseñas no coinciden

Figura 24: US-0009: Criterio de aceptación 3

- Introducir una contraseña correcta y comprobar que actualiza la contraseña.

Gestor de Promociones
Cerrar sesión

- Resumen
- Promociones
- Ofertas
- Usuarios
- Todos los usuarios
- Añadir nuevo
- Mi perfil
- Usuarios registrados

**Usuario:** admin-actualizado@example.com

Registrado el 12/05/2016 23:39

**Email \***

**Contraseña \***

**Repetir contraseña**

**Actualizar**

Figura 25: US-0009: Criterio de aceptación 4

## Sprint 3, del 28/3/2016 al 10/4/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Como no se ha modificado el sprint backlog ni se plantea modificar el modelo de datos se continúa con el mismo que el del sprint anterior.

### Riesgos

Tras el análisis hay que actualizar la tabla de gestión de riesgos:

Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	20%	10	2,00
Problemas al integrar en el servidor de pruebas o producción	40%	10	4,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	70%	15	10,50
Añadir nuevas funcionalidades	55%	20	11,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	50%	5	2,50
La tecnología utilizada se discontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
			<b>37,50</b>

La probabilidad del riesgo “Problemas al integrar en el servidor de pruebas o producción” se reduce ya que al integrar en el servidor de pruebas y de producción no ha ocurrido ningún problema y se espera que siga así en el futuro. Las probabilidades de los riesgos de “Fallo en el servicio de control de versiones”, “Cambios en las historias de usuario” y “Añadir nuevas funcionalidades” se reducen ya que según avanza el tiempo hay menos probabilidad de que ocurra. El gráfico resultante se puede ver en la figura 26.

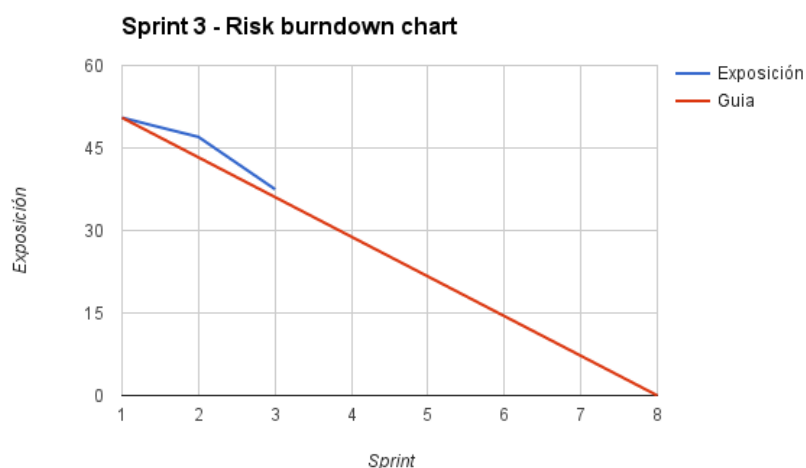


Figura 26: Risk burndown chart del Sprint 3

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las dos siguientes:

- Historia de usuario US-0001: Como superadmin, quiero crear/ver/modificar/eliminar usuarios de cualquier rol que permita el sistema.
- Historia de usuario US-0002: Como gestor quiero poder crear/ver/modificar/eliminar promociones.

## Criterios de aceptación US-0001

- Al crear un nuevo usuario, introducir los datos correctamente y ver que aparece en el listado.

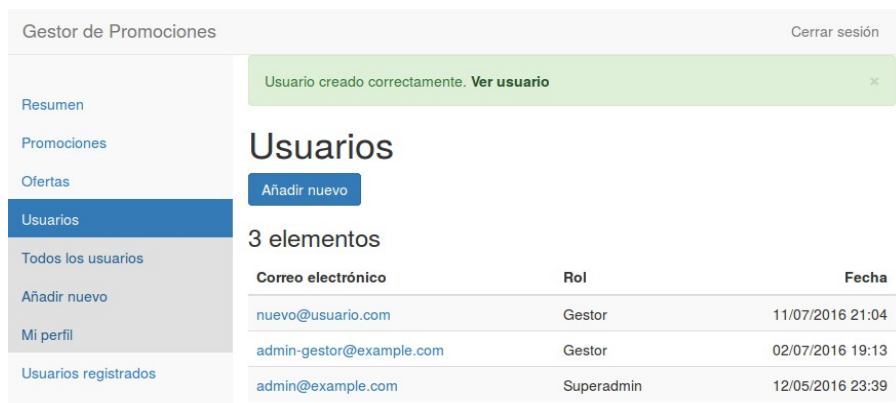


Figura 27: US-0001: Criterio de aceptación 1

- Al crear un nuevo usuario, introducir un correo electrónico incorrecto y ver que no aparece en el listado.

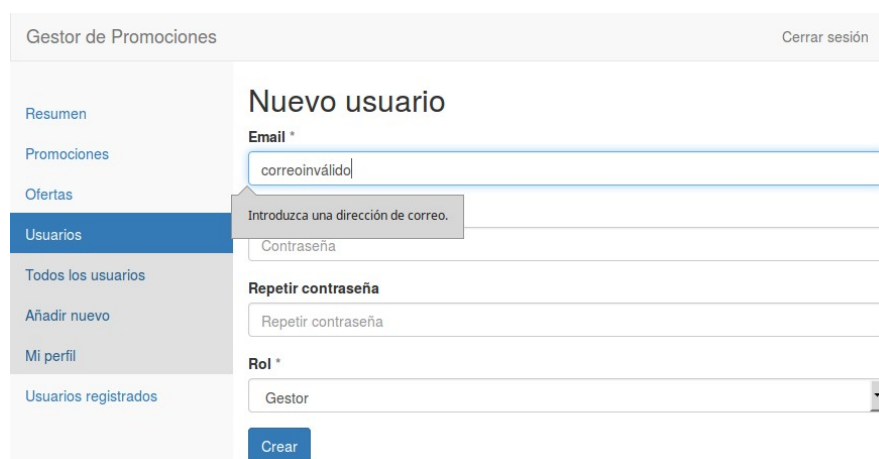


Figura 28: US-0001: Criterio de aceptación 2

- Al actualizar un usuario, introducir los datos correctamente y ver que ha cambiado sus datos.

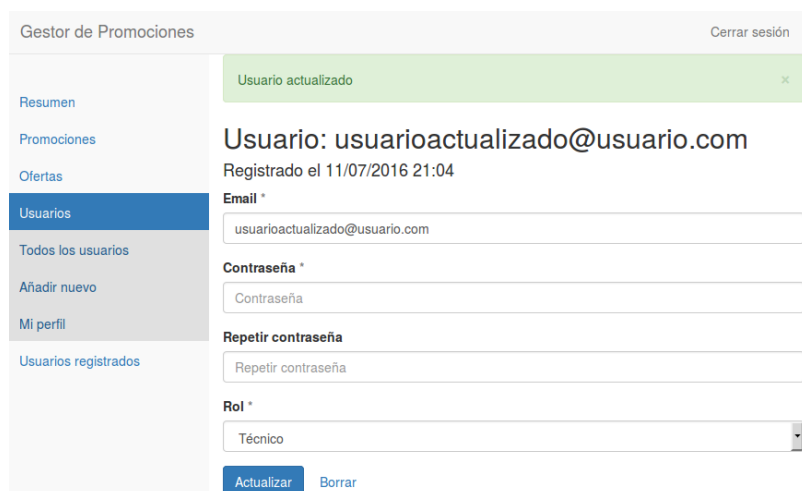


Figura 29: US-0001: Criterio de aceptación 3

- Al actualizar un usuario, introducir un correo electrónico incorrecto y ver que no ha actualizado sus datos.

Figura 30: US-0001: Criterio de aceptación 4

- Al borrar un usuario, ver que no aparece en el listado.

Correo electrónico	Rol	Fecha
admin-gestor@example.com	Gestor	02/07/2016 19:13
admin@example.com	Superadmin	12/05/2016 23:39

Figura 31: US-0001: Criterio de aceptación 5

## Criterios de aceptación US-0002

- Al crear una nueva promoción, introducir los datos correctamente y ver que aparece en el listado.

Gestor de Promociones Cerrar sesión

Promoción creada correctamente x

Resumen

**Promociones**

Todas las promociones

Añadir nueva

Ofertas

Usuarios

Usuarios registrados

### Promoción: Nueva promoción

Creada el 11/07/2016 21:14

**Nombre \***

Nueva promoción

**Descripción \***

Permite etiquetas HTML

Lorem ipsum...

Figura 32: US-0002: Criterio de aceptación 1

- Al crear una nueva promoción, dejar en blanco cualquier campo y ver que no aparece en el listado.

Gestor de Promociones Cerrar sesión

Resumen

**Promociones**

Todas las promociones

Añadir nueva

Ofertas

Usuarios

Usuarios registrados

### Nueva promoción

**Nombre \***

Nombre

Rellene este campo.

Permite etiquetas HTML

Lorem...

Figura 33: US-0002: Criterio de aceptación 2

- Al actualizar una promoción, introducir los datos correctamente y ver que ha cambiado sus datos.

Gestor de Promociones Cerrar sesión

Promoción actualizada x

Resumen

**Promociones**

Todas las promociones

Añadir nueva

Ofertas

Usuarios

Usuarios registrados

### Promoción: Nueva promoción ACTUALIZADA

Creada el 11/07/2016 21:14

**Nombre \***

Nueva promoción ACTUALIZADA

**Descripción \***

Permite etiquetas HTML

NUOVA DESCRIPCIÓN

Figura 34: US-0002: Criterio de aceptación 3

- Al actualizar una promoción, dejar en blanco cualquier campo y ver que no ha actualizado sus datos.

Figura 35: US-0002: Criterio de aceptación 4

- Al borrar una promoción, ver que no aparece en el listado.

Nombre	Imagen	¿Promoción activa?	Nº de ofertas asociadas	Fecha
Promo Renfe		Sí	2	30/05/2016 20:28

Figura 36: US-0002: Criterio de aceptación 5

## Sprint 4, del 11/4/2016 al 24/4/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Como no se ha modificado el sprint backlog ni se plantea modificar el modelo de datos se continúa con el mismo que el del sprint anterior.

### Riesgos

Tras el análisis hay que actualizar la tabla de gestión de riesgos:



Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	10%	10	1,00
Problemas al integrar en el servidor de pruebas o producción	20%	10	2,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	60%	15	9,00
Añadir nuevas funcionalidades	45%	20	9,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	50%	5	2,50
La tecnología utilizada se descontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
			<b>31,00</b>

La probabilidad del riesgo “Problemas al integrar en el servidor de pruebas o producción” se reduce ya que al integrar en el servidor de pruebas y de producción no ha ocurrido ningún problema y se espera que siga así en el futuro. Las probabilidades de los riesgos de “Fallo en el servicio de control de versiones”, “Cambios en las historias de usuario” y “Añadir nuevas funcionalidades” se reducen ya que según avanza el tiempo hay menos probabilidad de que ocurra. El gráfico resultante se puede ver en la figura 37.

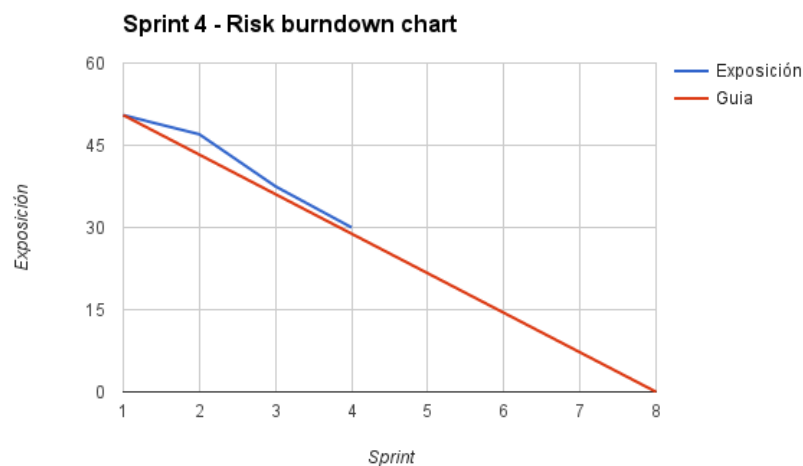


Figura 37: Risk burndown chart del Sprint 4

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las dos siguientes:

- Historia de usuario US-0004: Como gestor quiero poder crear/ver/modificar/eliminar ofertas.
- Historia de usuario US-0017: Como gestor quiero poder vincular las ofertas a las promociones.

### Criterios de aceptación US-0004

- Al crear una nueva oferta, introducir los datos correctamente y ver que aparece en el listado.
- Al crear una nueva oferta, dejar en blanco el nombre y ver que no aparece en el listado.
- Al actualizar una oferta, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar una oferta, dejar en blanco el nombre y ver que no ha actualizado sus datos.
- Al borrar una oferta, ver que no aparece en el listado.

Todos los criterios de aceptación son muy parecidos que los de la historia de usuario US-0002 y el resultado es igual no se han incluido las imágenes ya que se pueden ver más arriba.

### Criterios de aceptación US-0017

- Desde el panel de administración, en una promoción, vinculo al menos una oferta y al guardar compruebo que tiene las ofertas vinculadas.



Figura 38: US-0017: Criterio de aceptación 1

- Desde el panel de administración, en una promoción, desvinculo todas las ofertas y al guardar compruebo que no tiene ninguna oferta vinculada.



Figura 39: US-0017: Criterio de aceptación 2

## Sprint 5, del 25/4/2016 al 8/5/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Tras realizar la historia de usuario US-0017: Como gestor quiero poder vincular las ofertas a las promociones, se observó que es necesario un campo nuevo para las ofertas ya que pueden no diferenciarse correctamente varias ofertas con el mismo nombre. Por ello se modifica la historia de usuario US-0004 para añadir un campo más que permita al gestor identificar correctamente la oferta dentro del gestor de promociones aunque para el usuario final la oferta tenga el mismo nombre:

### Historia de usuario US-0004 (v2)

Como gestor quiero poder crear/ver/modificar/eliminar ofertas. Cada oferta tendrá los siguientes campos:

- Nombre identificativo dentro del gestor
- Nombre público que se muestra al usuario final
- Descripción
- Imagen
- Etiqueta
- Código del video de youtube
- Ciudades
- Testimonios

#### **Criterios de aceptación:**

- Al crear una nueva oferta, introducir los datos correctamente y ver que aparece en el listado.
- Al crear una nueva oferta, dejar en blanco el nombre y ver que no aparece en el listado.
- Al actualizar una oferta, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar una oferta, dejar en blanco el nombre y ver que no ha actualizado sus datos.
- Al borrar una oferta, ver que no aparece en el listado.

Como se ha actualizado una historia de usuario que detalla los campos de una entidad hay que actualizar también el modelo de datos que utiliza la aplicación para reflejar el cambio. La figura 40 representa el nuevo modelo de datos actualizado.

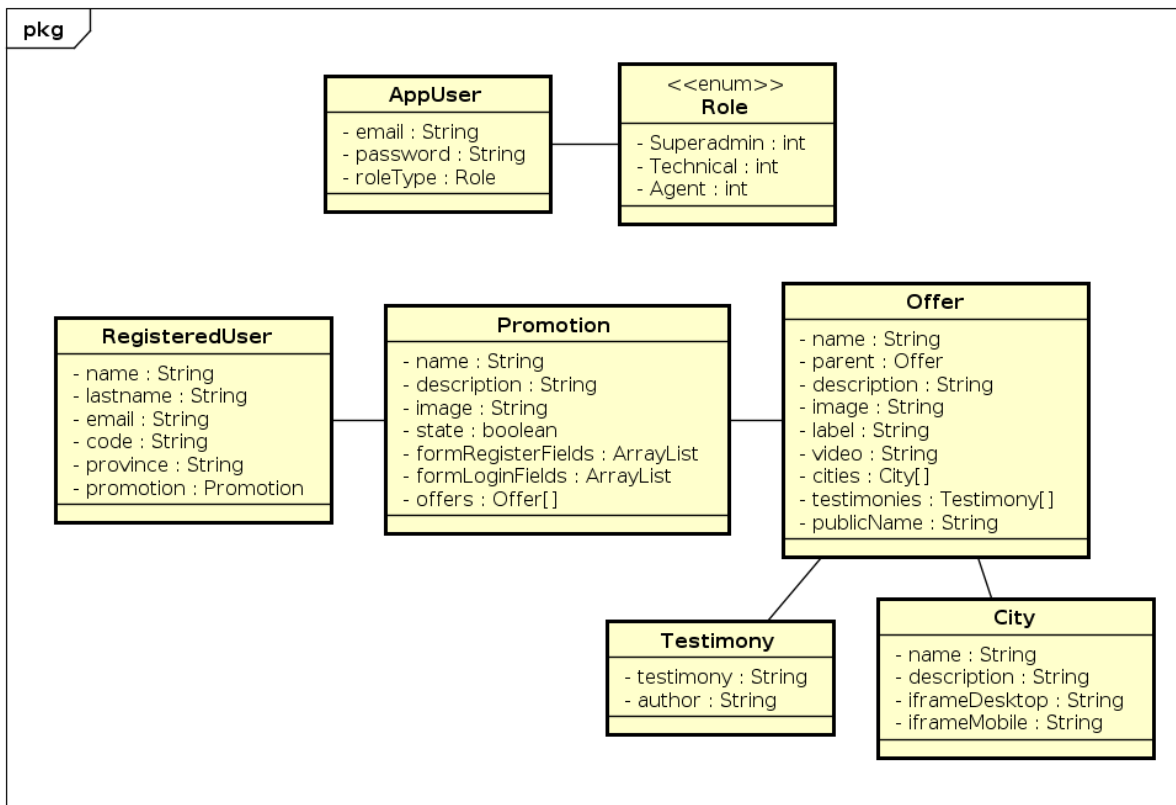


Figura 40: Modelo de datos actualizado en el sprint 5

## Riesgos

Tras el análisis hay que actualizar la tabla de gestión de riesgos:

<b>Riesgo</b>	<b>Probabilidad</b>	<b>Pérdida (en días)</b>	<b>Exposición</b>
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	0%	10	0,00
Problemas al integrar en el servidor de pruebas o producción	0%	10	0,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	35%	15	8,75
Añadir nuevas funcionalidades	35%	20	7,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	50%	5	2,50
La tecnología utilizada se descontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
			<b>25,75</b>

Se ha producido un cambio en una historia de usuario. El cambio es bastante simple y no tiene mucha complejidad. Aún así la probabilidad de que ocurra este riesgo se reduce.

La probabilidad del riesgo “Problemas al integrar en el servidor de pruebas o producción” se reduce ya que al integrar en el servidor de pruebas y de producción no ha ocurrido ningún problema y se espera que siga así en el futuro. Las probabilidades de los riesgos de “Fallo en el servicio de control de versiones” y “Añadir nuevas funcionalidades” se reducen ya que según avanza el tiempo hay menos probabilidad de que ocurra.

En los riesgos “Fallo en el servicio de control de versiones” y “Problemas al integrar en el servidor de pruebas o producción” se ha llegado a la probabilidad de 0%. Esto significa que es muy difícil que ocurra, no que nunca vaya a ocurrir. Por ello se mantiene en la tabla de riesgos aunque se espera que no se de el caso. El gráfico resultante se puede ver en la figura 41.

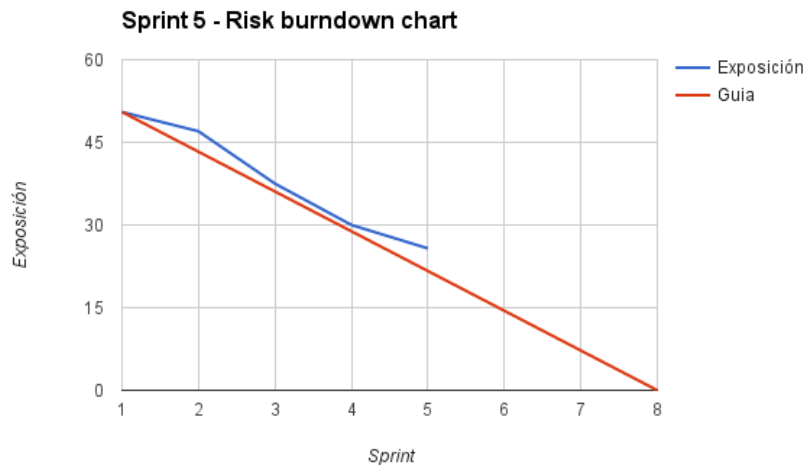


Figura 41: Risk burndown chart del Sprint 5

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las cuatro siguientes:

- Historia de usuario US-0004 (v2): Como gestor quiero poder crear/ver/modificar/eliminar ofertas.
- Historia de usuario US-0013: Como técnico quiero obtener la información de una promoción a través de una API proporcionando el id de la promoción.
- Historia de usuario US-0014: Como técnico quiero poder registrar usuarios a una promoción a través de una API proporcionando el id de la promoción y los campos del usuario que requiere dicha promoción para registrarle.
- Historia de usuario US-0015: Como técnico quiero poder identificar usuarios ya registrados a una promoción a través de una API proporcionando el id de la promoción y los campos del usuario que requiere dicha promoción para identificarle.

### Criterios de aceptación US-0004 (v2)

Como ya se ha probado en el sprint anterior los criterios de aceptación y no se han modificado en la versión 2 de la historia de usuario, se vuelven a probar y el resultado es el mismo que en la ocasión anterior.

### Criterios de aceptación US-0013

- Realizando la petición a la API, proporciono un id inválido y veo que devuelve error.

```
1 {
2   "code": 1,
3   "message": "Promoción no encontrada",
4   "description": "La promoción no existe"
5 }
```

Figura 42: US-0013: Criterio de aceptación 1

- Realizando la petición a la API, proporciono un id válido y veo que devuelve la información de la promoción.

```
1 {
2   "name": "Promo Renfe",
3   "description": "Solo por ser cliente de Renfe, podrás be
4     Entertainment a un precio tan especial como tú. ¿Qué
5   "image": "logo-576061f988228.png",
6   "formRegister": {
7     "name": true,
8     "lastname": true,
9     "email": true
10  },
11  "formLogin": {
12    "email": true
13  }
14 }
```

Figura 43: US-0013: Criterio de aceptación 2

## Criterios de aceptación US-0014

- Realizando la petición a la API, proporciono un id inválido y datos incorrectos y veo que devuelve error.

```
1 {
2   "code": 34,
3   "message": "Datos inválidos",
4   "description": "Los datos enviados son incorrectos"
5 }
```

Figura 44: US-0014: Criterio de aceptación 1

- Realizando la petición a la API, proporciono un id inválido y datos correctos y veo que devuelve error.

```
1 {
2   "code": 34,
3   "message": "Datos inválidos",
4   "description": "Los datos enviados son incorrectos"
5 }
```

Figura 45: US-0014: Criterio de aceptación 2

- Realizando la petición a la API, proporciono un id válido y datos incorrectos y veo que devuelve error.



```

1  {
2  "code": 34,
3  "message": "Datos inválidos",
4  "description": "Los datos enviados son incorrectos"
5  }

```

Figura 46: US-0014: Criterio de aceptación 3

- Realizando la petición a la API, proporciono un id válido y veo que devuelve la información de las ofertas asociadas a la promoción.

```

1  {
2  "user": {
3  "id": 4
4  },
5  "offers": [
6  {
7  "name": "El Rey León",
8  "description": "Descripción de ERL",
9  "image": "Logo-576061a7f11da.jpg",
10 "label": "PRÓXIMAMENTE",
11 "video": "kPDnw3_1GOI",
12 "cities": [
13 {
14 "name": "Ciudad1",
15 "description": "DescCiudad1",
16 "iframe-desktop": "http://example.com/",
17 "iframe-mobile": "http://example.com/"
18 }
19 ],
20 "testimonies": [
21 {
22 "testimony": "Genial",
23 "author": "Autor"
24 }
25 ]
26 }
27 ]
28 }

```

Figura 47: US-0014: Criterio de aceptación 4

## Criterios de aceptación US-0015

- Realizando la petición a la API, proporciono un id inválido y datos incorrectos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id inválido y datos correctos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y datos incorrectos y veo que devuelve error.
- Realizando la petición a la API, proporciono un id válido y veo que devuelve la información de las ofertas asociadas a la promoción.

Todos los criterios de aceptación son los mismos que los de la historia de usuario US-0015 y el resultado es igual, no se han incluido las imágenes ya que se pueden ver más arriba.

## Sprint 6, del 9/5/2016 al 22/5/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Tras terminar el sprint anterior se vió que es necesario que el usuario gestor vea cómo quedaría la promoción con una vista previa, y que el técnico pueda cargar y descargar el código que haga uso de la API. Por tanto se añaden dos nuevas historias de usuario:

<b>Historia de usuario US-0018</b>
Como gestor quiero poder ver una vista previa de cómo es la promoción para el usuario final.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dentro de la promoción, al pulsar el botón “Vista previa” abre la página web que simula cómo vería un usuario final la promoción.</li></ul>

<b>Historia de usuario US-0019</b>
Como técnico quiero que en la parte de administración haya una opción que permita cargar y descargar el código correspondiente a una promoción en concreto.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• El zip se ha guardado correctamente en mi ordenador.</li><li>• El zip no está corrupto.</li><li>• Están todos los archivos que necesito.</li></ul>

En esta ocasión sí que cambia el sprint backlog ya que se han añadido nuevas historias de usuario y hay que incluirlas de forma priorizada. El sprint backlog es el siguiente:

- Historia de usuario US-0012: Como gestor quiero que una oferta tenga los campos de otra vinculada si no relleno los datos.
- Historia de usuario US-0016: Como técnico quiero poder tener un código que haga uso de la API.
- Historia de usuario US-0003: Como gestor quiero poder crear/ver/modificar/eliminar códigos promocionales de las promociones si el campo del formulario de registro está activo.
- Historia de usuario US-0005: Como gestor quiero poder crear/ver/modificar/eliminar ciudades de las ofertas.
- Historia de usuario US-0006: Como gestor quiero poder crear/ver/modificar/eliminar

testimonios de las ofertas.

- Historia de usuario US-0018: Como gestor quiero poder ver una vista previa de cómo es la promoción para el usuario final.
- Historia de usuario US-0019: Como técnico quiero que en la parte de administración haya una opción que permita cargar y descargar el código correspondiente a una promoción en concreto.
- Historia de usuario US-0010: Como usuario final, quiero registrarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0011: Como usuario final, quiero identificarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0007: Como gestor quiero poder ver los usuarios que se han registrado a cada promoción.

## Riesgos

Tras el análisis hay que actualizar la tabla de gestión de riesgos. Como la tecnología de Angular2 no ha salido todavía la versión final sino que anda con versiones beta y release candidate es posible que cambie su API o modifique funcionalidades. Como se emplea esta tecnología hay que tener en cuenta esta posibilidad por lo que se añade ahora a la tabla de riesgos:

Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	0%	10	0,00
Problemas al integrar en el servidor de pruebas o producción	0%	10	0,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	25%	15	3,75
Añadir nuevas funcionalidades	20%	20	4,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	20%	5	1,00
La tecnología utilizada se discontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
Cambio en la API de Angular2	50%	10	5,00
			<b>21,25</b>

En este sprint se han añadido nuevas funcionalidades. Como estamos llegando al final del

desarrollo de la aplicación y se han añadido estas 2 nuevas historias de usuario se reduce un poco más la probabilidad de que vuelva a ocurrir este riesgo.

Como ya se ha implementado mucha de la parte visual y la parte de la API el probar en todos los navegadores y sistemas operativos y no dar fallo se espera que continúe así y por eso se reduce la probabilidad de no tener tiempo para hacer las pruebas. Como todavía quedan algunas secciones de la parte visual no se puede reducir a 0 su probabilidad.

La probabilidad del riesgo “Cambios en las historias de usuario” se reduce ya que según avanza el tiempo hay menos probabilidad de que ocurra. El gráfico resultante se puede ver en la figura 48.

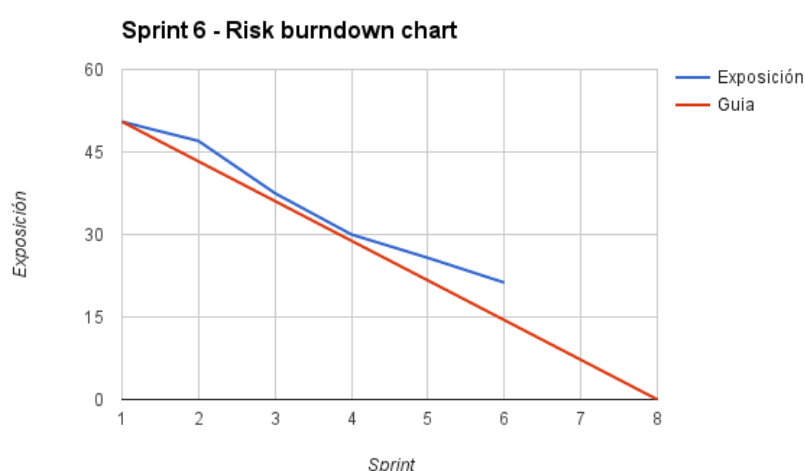


Figura 48: Risk burndown chart del Sprint 6

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las dos siguientes:

- Historia de usuario US-0012: Como gestor quiero que una oferta tenga los campos de otra vinculada si no relleno los datos.
- Historia de usuario US-0016: Como técnico quiero poder tener un código que haga uso de la API.

## Criterios de aceptación US-0012

- Crear una oferta sin todos los datos vinculando otra y comprobar que devuelve toda la información a través de la API.

**Oferta padre**  
Si se selecciona un padre heredará los datos de los campos que se dejen en blanco

Mamma Mía! (Genérica)

**Nombre público**  
Nombre que se mostrará en las promociones

OFERTA SIN TODOS LOS CAMPOS

**Descripción**  
Permite etiquetas HTML

Descripción

**Imagen**  
Tamaño máximo: 1MB. Tipo de imagen aceptada: JPG, PNG y GIF.

Examinar... No se ha seleccionado ningún archivo.

**Label**

Label

Figura 49: US-0012: Criterio de aceptación 1 (1)

```

1  {
2  "user": {
3    "id": 4
4  },
5  "offers": [
6    {
7      "name": "OFERTA SIN TODOS LOS CAMPOS",
8      "description": "Descripción Mamma Mía",
9      "image": "logo-5760619acba92.jpg",
10     "label": "DE GIRA",
11     "video": "kPDnw3_1GOI",
12     "cities": [],
13     "testimonies": [
14       {
15         "testimony": "asdf",
16         "author": "foo"
17       }
18     ]
19   }
20 ]
21 }

```

Figura 50: US-0012: Criterio de aceptación 1 (2)

- Crear una oferta sin todos los datos y sin vincular otra y comprobar que no se crea.

Gestor de Promociones Cerrar sesión

Resumen

Promociones

**Ofertas**

Todas las ofertas

Añadir nueva

Usuarios

• La oferta no tiene todos los campos necesarios

**Oferta: Oferta hija (hereda de Mamma Mía! (Genérica))**

Creada el 12/07/2016 19:25

**Nombre \***

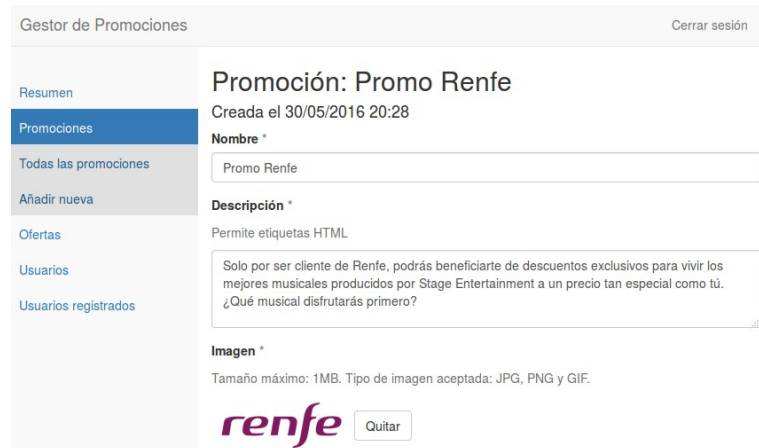
Nombre privado para identificar la oferta en el gestor

Oferta hija

Figura 51: US-0012: Criterio de aceptación 2

## Criterios de aceptación US-0016

- Probar el código y comprobar que los datos que aparecen son los mismos que los que hay en la promoción dentro del gestor.



The screenshot shows a web interface for managing promotions. The main content area displays the following information:

- Gestor de Promociones** (top right: Cerrar sesión)
- Resumen** (left sidebar)
- Promoción: Promo Renfe** (title)
- Creada el 30/05/2016 20:28** (creation date)
- Nombre \*** (input field containing "Promo Renfe")
- Descripción \*** (text area containing: "Permite etiquetas HTML", "Solo por ser cliente de Renfe, podrás beneficiarte de descuentos exclusivos para vivir los mejores musicales producidos por Stage Entertainment a un precio tan especial como tú. ¿Qué musical disfrutarás primero?")
- Imagen \*** (text: "Tamaño máximo: 1MB. Tipo de imagen aceptada: JPG, PNG y GIF.")
- renfe** logo (bottom left) and **Quitar** button (bottom right)

Figura 52: US-0016: Criterio de aceptación 1 (1)



Figura 53: US-0016: Criterio de aceptación 1 (2)

## Sprint 7, del 23/5/2016 al 5/6/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Como no se ha modificado el sprint backlog ni se plantea modificar el modelo de datos se continúa con el mismo que el del sprint anterior.

### Riesgos

Tras el análisis hay que actualizar la tabla de gestión de riesgos:

Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	0%	10	0,00
Problemas al integrar en el servidor de pruebas o producción	0%	10	0,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	15%	15	2,25
Añadir nuevas funcionalidades	10%	20	2,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	20%	5	1,00
La tecnología utilizada se descontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
Cambio en la API de Angular2	50%	10	5,00
			<b>17,75</b>

Las probabilidades de los riesgos de “Cambios en las historias de usuario” y “Añadir nuevas funcionalidades” se reducen ya que según avanza el tiempo hay menos probabilidad de que ocurra. El gráfico resultante se puede ver en la figura 54.

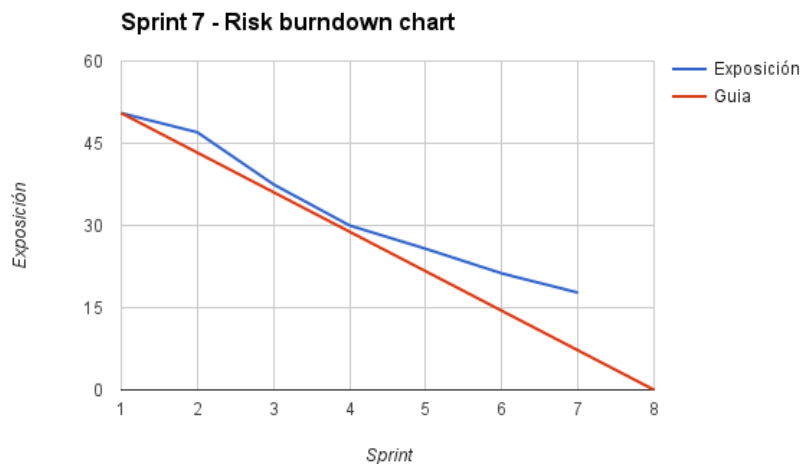


Figura 54: Risk burndown chart del Sprint 7

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las tres siguientes:

- Historia de usuario US-0003: Como gestor quiero poder crear/ver/modificar/eliminar códigos promocionales de las promociones si el campo del formulario de registro está activo.
- Historia de usuario US-0005: Como gestor quiero poder crear/ver/modificar/eliminar ciudades de las ofertas.
- Historia de usuario US-0006: Como gestor quiero poder crear/ver/modificar/eliminar testimonios de las ofertas.

### Crterios de aceptación US-0003

- Al crear un nuevo código promocional, introducir los datos correctamente y ver que aparece en el listado.
- Al crear un nuevo código promocional, dejar en blanco el código y ver que no aparece en el listado.
- Al actualizar un código promocional, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar un código promocional, dejar en blanco el código y ver que no ha actualizado sus datos.
- Al borrar un código promocional, ver que no aparece en el listado.

Todos los criterios de aceptación son muy parecidos que los de la historia de usuario US-0002 y el resultado es igual no se han incluido las imágenes ya que se pueden ver más arriba.

### Crterios de aceptación US-0005

- Al crear una nueva ciudad, introducir los datos correctamente y ver que aparece en el listado.
- Al crear una nueva ciudad, dejar en blanco el nombre, la descripción o la URL del iframe y ver que no aparece en el listado.
- Al actualizar una ciudad, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar una ciudad, dejar en blanco el nombre, la descripción o la URL del iframe y ver que no ha actualizado sus datos.
- Al borrar una ciudad, ver que no aparece en el listado.

Todos los criterios de aceptación son muy parecidos que los de la historia de usuario US-0002 y el resultado es igual no se han incluido las imágenes ya que se pueden ver más arriba.



## Criterios de aceptación US-0006

- Al crear un nuevo testimonio, introducir los datos correctamente y ver que aparece en el listado.
- Al crear un nuevo testimonio, dejar en blanco cualquier campo y ver que no aparece en el listado.
- Al actualizar un testimonio, introducir los datos correctamente y ver que ha cambiado sus datos.
- Al actualizar un testimonio, dejar en blanco cualquier campo y ver que no ha actualizado sus datos.
- Al borrar un testimonio, ver que no aparece en el listado.

Todos los criterios de aceptación son muy parecidos que los de la historia de usuario US-0002 y el resultado es igual no se han incluido las imágenes ya que se pueden ver más arriba.

## Sprint 8, del 6/6/2016 al 19/6/2016

### Análisis

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Se ha añadido una nueva historia de usuario que permite al superadministrador borrar usuarios registrados, no añadir o modificar los datos como al resto de entidades.

Historia de usuario US-0020
Como superadministrador, quiero poder borrar los usuarios que se han registrado a cada promoción.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Eliminar un usuario y comprobar que desaparece de la lista.</li></ul>

En esta ocasión sí que cambia el sprint backlog ya que se han añadido nuevas historias de usuario y hay que incluirlas de forma priorizada. El sprint backlog es el siguiente:

- Historia de usuario US-0018: Como gestor quiero poder ver una vista previa de cómo es la promoción para el usuario final.
- Historia de usuario US-0019: Como técnico quiero que en la parte de administración haya una opción que permita cargar y descargar el código correspondiente a una promoción en concreto.
- Historia de usuario US-0010: Como usuario final, quiero registrarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0011: Como usuario final, quiero identificarme en una

promoción para disfrutar de los descuentos asociados.

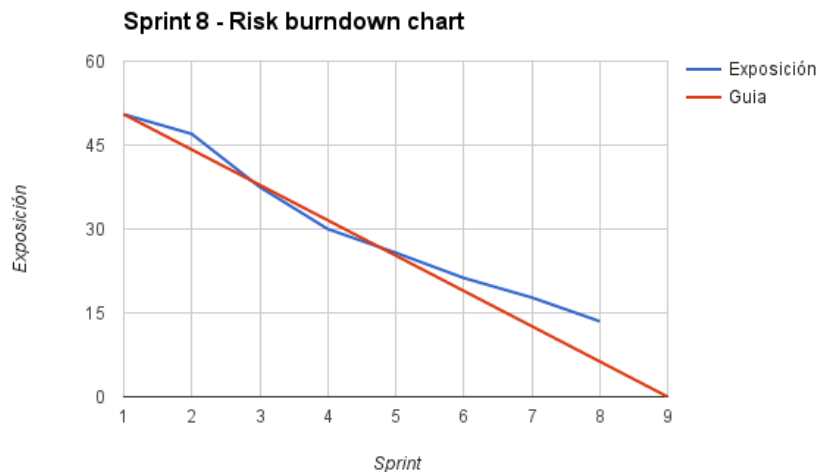
- Historia de usuario US-0007: Como gestor quiero poder ver los usuarios que se han registrado a cada promoción.
- Historia de usuario US-0020: Como superadministrador, quiero poder borrar los usuarios que se han registrado a cada promoción.

## Riesgos

Tras el análisis hay que actualizar la tabla de gestión de riesgos:

Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	15%	10	1,50
Fallo en el servicio de control de versiones	0%	10	0,00
Problemas al integrar en el servidor de pruebas o producción	0%	10	0,00
Fallos de las máquinas de desarrollo	20%	15	3,00
Cambios en las historias de usuario	0%	15	0,00
Añadir nuevas funcionalidades	0%	20	0,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	20%	5	1,00
La tecnología utilizada se discontinúa o deja de recibir actualizaciones de seguridad	10%	30	3,00
Cambio en la API de Angular2	50%	10	5,00
			<b>13,50</b>

En este sprint se han añadido nuevas funcionalidades. Como estamos ya en el final del desarrollo de la aplicación y se han añadido esta nueva historia de usuario se reduce a 0 la probabilidad ya que la aplicación está casi implementada totalmente. Lo mismo ocurre con los cambios en las historias de usuario. El gráfico resultante se puede ver en la figura 55.



*Figura 55: Risk burndown chart del Sprint 8*

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las dos siguientes:

- Historia de usuario US-0018: Como gestor quiero poder ver una vista previa de cómo es la promoción para el usuario final.
- Historia de usuario US-0019: Como técnico quiero que en la parte de administración haya una opción que permita cargar y descargar el código correspondiente a una promoción en concreto.

Como no se va a poder completar todas las historias de usuario no van a ser 8 sprints como se había estimado, sino que serán 9. Por ello el gráfico de la exposición de los riesgos se ha incrementado a 9 sprints.

## Criterios de aceptación US-0018

- Dentro de la promoción, al pulsar el botón “Vista previa” abre la página web que simula cómo vería un usuario final la promoción.



*Figura 56: US-0018: Criterio de aceptación 1*

## **Criterios de aceptación US-0019**

- El zip se ha guardado correctamente en mi ordenador.
- El zip no está corrupto.
- Están todos los archivos que necesito.

Al descargar el fichero zip, descomprimirlo y abrir el fichero index.html desde un servidor se puede ver la misma página web que en el criterio de aceptación US-0018.

## **Sprint 9, del 20/6/2016 al 26/6/2016**

### **Análisis**

En la primera reunión del sprint se reúne el equipo para analizar los cambios que hay que realizar antes de continuar con el proyecto si hubiera alguno. Como no se ha modificado el sprint backlog ni se plantea modificar el modelo de datos se continúa con el mismo que el del sprint anterior.

### **Riesgos**

Tras el análisis hay que actualizar la tabla de gestión de riesgos:

Riesgo	Probabilidad	Pérdida (en días)	Exposición
Fallo de la conectividad de la red	0%	10	0,00
Fallo en el servicio de control de versiones	0%	10	0,00
Problemas al integrar en el servidor de pruebas o producción	0%	10	0,00
Fallos de las máquinas de desarrollo	0%	15	0,00
Cambios en las historias de usuario	0%	15	0,00
Añadir nuevas funcionalidades	0%	20	0,00
Tiempo insuficiente para probar en todos los navegadores y sistemas operativos	0%	5	0,00
La tecnología utilizada se descontinúa o deja de recibir actualizaciones de seguridad	0%	30	0,00
Cambio en la API de Angular2	0%	10	0,00
			<b>0,00</b>

Este es el último sprint y se espera que no falle la conectividad de la red, ni las máquinas de desarrollo, ni tener tiempo insuficiente para pruebas, ni que deje de recibir actualizaciones de seguridad la tecnología empleada. Por eso se reduce a 0% la probabilidad de cada una de los posibles riesgos y esto genera un valor de 0 a la exposición total. El gráfico resultante se puede ver en la figura 57.

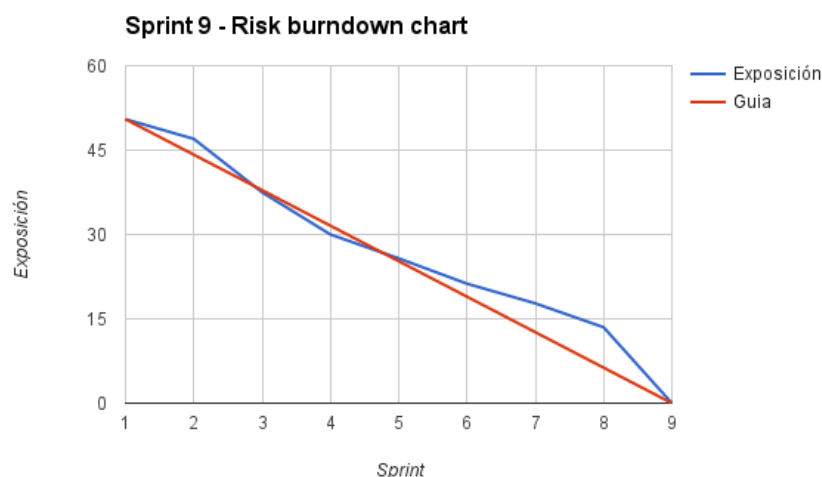


Figura 57: Risk burndown chart del Sprint 9

## Historias de usuario

Una vez terminada la actualización de los riesgos hay que decidir cuántas historias de usuario se realizarán en este sprint a lo que se han elegido las restantes:

- Historia de usuario US-0010: Como usuario final, quiero registrarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0011: Como usuario final, quiero identificarme en una promoción para disfrutar de los descuentos asociados.
- Historia de usuario US-0007: Como gestor quiero poder ver los usuarios que se han registrado a cada promoción.
- Historia de usuario US-0020: Como superadministrador, quiero poder borrar los usuarios que se han registrado a cada promoción.

## Criterios de aceptación US-0010

- Rellenar algún campo incorrectamente y comprobar que se muestra error.

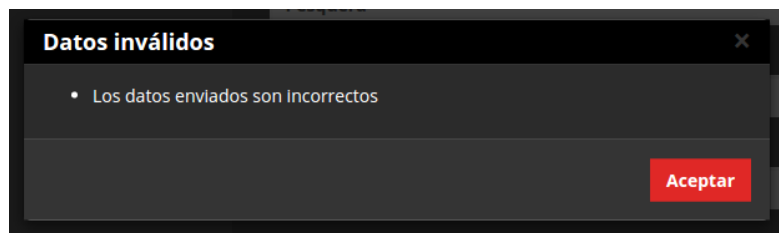


Figura 58: US-0010: Criterio de aceptación 1

- Rellenar todos los campos correctamente y comprobar que se identifica correctamente.



Figura 59: US-0010: Criterio de aceptación 2

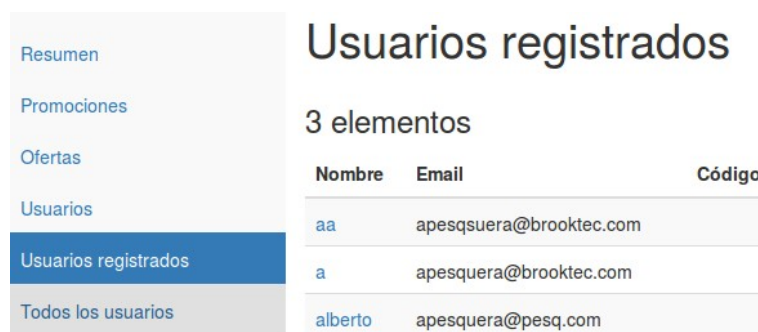
## Criterios de aceptación US-0011

- Rellenar algún campo incorrectamente y comprobar que se muestra error.
- Rellenar todos los campos correctamente y comprobar que se identifica correctamente.

Todos los criterios de aceptación son muy parecidos que los de la historia de usuario US-0011 y el resultado es igual no se han incluido las imágenes ya que se pueden ver más arriba.

## Criterios de aceptación US-0007

- Desde el panel de administración, en la opción de usuarios registrados, veo el listado de los usuarios que se han registrado y a que promoción.

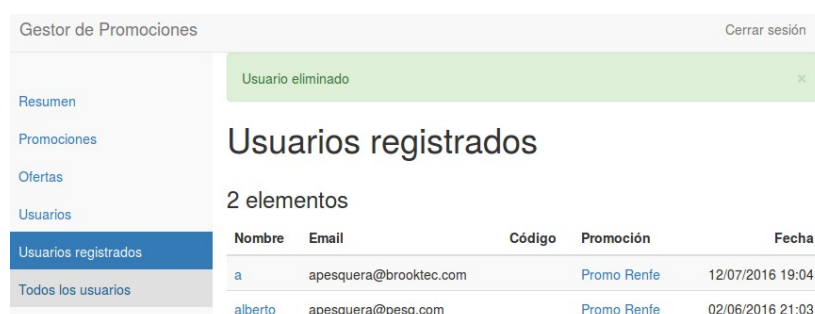


Nombre	Email	Código
aa	apesquera@brooktec.com	
a	apesquera@brooktec.com	
alberto	apesquera@pesq.com	

Figura 60: US-0007: Criterio de aceptación 1

## Criterios de aceptación US-0020

- Eliminar un usuario y comprobar que desaparece de la lista.



Nombre	Email	Código	Promoción	Fecha
a	apesquera@brooktec.com		Promo Rente	12/07/2016 19:04
alberto	apesquera@pesq.com		Promo Rente	02/06/2016 21:03

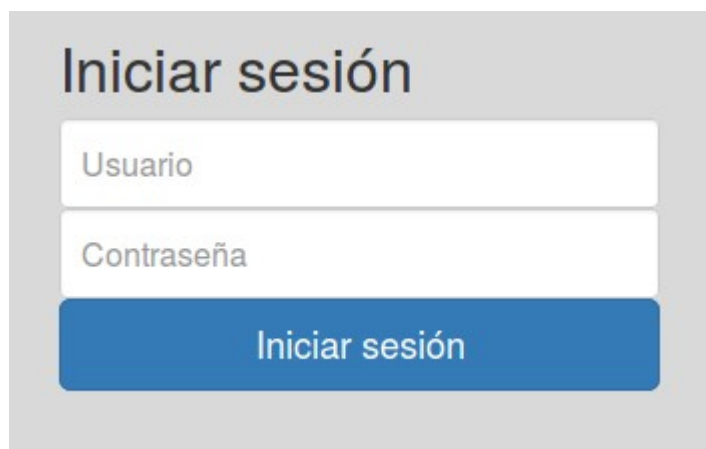
Figura 61: US-0020: Criterio de aceptación 1

## Capítulo 5. Diseño de la interfaz

En este capítulo se muestra la interfaz de la aplicación web así como las distintas funcionalidades que posee.

### Acceso al gestor

Al acceder al gestor aparece un formulario de inicio de sesión con los campos de correo electrónico y contraseña. El formulario es muy simple, se puede ver en la figura 62. Una vez rellenado con datos correctos se pulsa en el botón “Iniciar sesión” para autenticarse en el sistema.



El formulario de inicio de sesión tiene un fondo gris claro. En la parte superior, el título "Iniciar sesión" está en un color gris oscuro. Debajo del título hay dos campos de entrada de texto blancos con bordes grises: el primero está etiquetado "Usuario" y el segundo "Contraseña". Debajo de estos campos hay un botón rectangular azul con el texto "Iniciar sesión" en blanco.

Figura 62: Formulario de inicio de sesión

Una vez que se ha accedido al panel de administración aparece el resumen del gestor. Como se puede ver en la figura 63 el menú se encuentra en la parte izquierda de la pantalla, marcado con el número 1 en la figura 63. Si quisiéramos salir del panel de administración habría que hacerlo con el botón superior derecho, marcado en la figura 63 con el número 2.



El panel de administración tiene un encabezado con el título "Gestor de Promociones" a la izquierda y un botón "Cerrar sesión" a la derecha, ambos en un recuadro gris claro. A la izquierda hay un menú vertical con los siguientes ítems: "Resumen" (destacado con un fondo azul), "Promociones", "Ofertas", "Usuarios" y "Usuarios registrados". A la derecha del menú hay un área de "Resumen" con una lista de ítems:

Promociones	1
Ofertas	4
Usuarios	2
Usuarios registrados	3

El número 1 está debajo del menú y el número 2 está debajo del botón "Cerrar sesión".

Figura 63: Panel de administración



## Promociones

Al seleccionar a la opción del menú “Promociones” se despliega el submenú de esta opción, marcado con un 1 en la figura 64. Las opciones de este submenú son:

- Todas las promociones
- Nueva promoción
- Código

Nombre	Imagen	¿Promoción activa?	Nº de ofertas asociadas	Fecha
Promo Gas Natural		No	0	14/07/2016 18:23
Promo Renfe		Sí	2	30/05/2016 20:28

Figura 64: Todas las promociones

En “Todas las promociones” aparece una lista con ordenada por fecha de más reciente a más antigua con todas las promociones que hay en el sistema, estén activas o no. En la figura 64 se ve que en el recuadro número 2 aparecen dos promociones.

## Crear promociones

Para añadir una promoción se puede hacer pulsando la opción del submenú o el botón superior “Añadir nueva”. Al añadir una nueva promoción aparece un formulario como el de la figura 65 con los campos:

- Nombre
- Descripción
- Imagen
- ¿Activa?
- Campos del formulario de registro
- Campos del formulario de inicio de sesión
- Ofertas

Gestor de Promociones
Cerrar sesión

Resumen

Promociones

Todas las promociones

Añadir nueva

Código

Ofertas

Usuarios

Usuarios registrados

## Nueva promoción

**Nombre \***

**Descripción \***

Permite etiquetas HTML

Descripción

**Imagen \***

Tamaño máximo: 1MB. Tipo de imagen aceptada: JPG, PNG y GIF.

Examinar...

 No se ha seleccionado ningún archivo.
 

¿Activa?  Sí  No

**Campos del formulario de registro \***

Estos campos son los que se tendrá que rellenar el usuario para darse de alta en la promoción.

Nombre

Apellidos

Correo electrónico

Códigos promocionales

**Campos del formulario de inicio de sesión \***

Estos campos son los que se tendrá que rellenar el usuario para iniciar sesión.

Correo electrónico

Código promocional

**Ofertas**

Ofertas disponibles

Buscar...

El Rey León (genérica)
Mamma Mía! (Genérica)
Oferta hija

Ofertas asociadas

Buscar...

Quitar todas

Crear

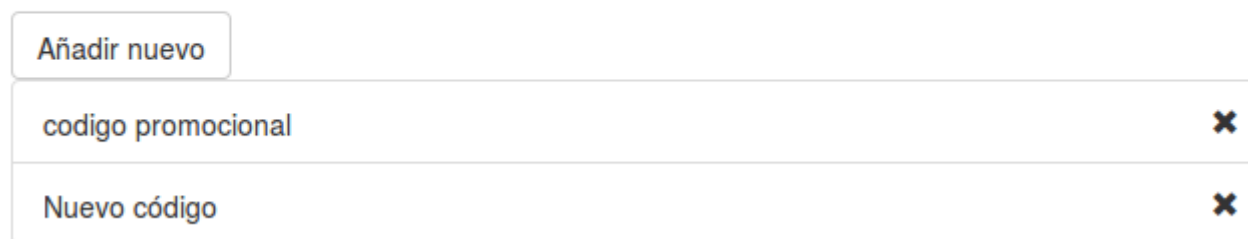
[Vista previa](#)

*Figura 65: Formulario de promoción*

Si en “Campos del formulario de registro” tiene activado “Códigos” aparece un nuevo campo que es la lista de los códigos asociados a la promoción, como aparece en la figura 66. Para crear un nuevo código hay que pulsar el botón “Añadir nuevo”. Tras pulsarlo se despliega una ventana con otro formulario para el código como el de la figura 67. Una vez que se rellenan los campos y se pulsa en “Guardar cambios” se añade el código creado a la lista. Para editar los códigos hay que seleccionarlo y se desplegará el mismo formulario con los datos rellenos.

### Códigos

Los códigos son sensibles a mayúsculas: "Abc", "ABC" y "abc" son 3 códigos distintos.



Añadir nuevo	
codigo promocional	✕
Nuevo código	✕

Figura 66: Códigos promocionales



#### Código

Código \*

Guardar cambios

Figura 67: Formulario para añadir o editar códigos

Si se quiere borrar un código de la lista hay que pulsar en la cruz que tiene ese código en la parte derecha. Esto no pide confirmación y automáticamente borra el código.

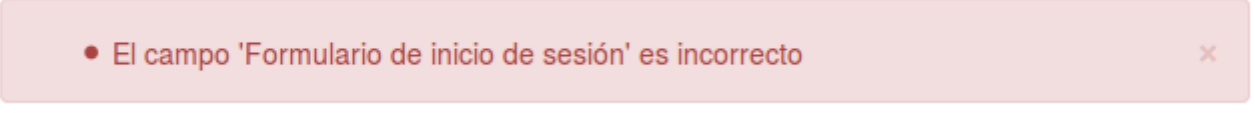
Una vez que se han relleno todos los datos para guardar la promoción hay que seleccionar el botón “Crear” que se encuentra en la parte inferior de la página. Al pulsar el botón comienza el proceso de guardado y mostrará el mensaje de la figura 68.



Promoción creada correctamente

*Figura 68: Mensaje de creación de promociones*

Si se ha producido algún error en la creación porque los datos introducidos no son correctos se muestra un mensaje de error indicando qué ha fallado. Se puede ver un ejemplo en la figura 69.



• El campo 'Formulario de inicio de sesión' es incorrecto

*Figura 69: Mensaje de error en una promoción*


## Editar promociones

Para editar una promoción en la lista de “Todas las promociones” de la figura 64 se selecciona la promoción que se vaya a editar. Esto desplegará la misma página que la figura 65 con los datos de la promoción seleccionada rellenos.

## Borrar promociones

Puede darse el caso de que se quiera borrar una promoción aunque no es lo más recomendable. Si no se quiere tener una promoción se puede marcar el campo “¿Activa?” a “No” y surge el mismo efecto.

Aún así el sistema permite el borrado de promociones. Para ello hay que seleccionar la promoción que se quiere borrar y pulsar el botón borrar que se encuentra en la parte de abajo. Tras pulsarlo aparece un mensaje indicando que se ha borrado, como el de la figura 70.



Promoción eliminada

*Figura 70: Promoción eliminada*

## Código

La opción “Código” permite descargar el código de una promoción. Para ello hay que pulsar el botón “Descargar código” y comienza la descarga de un fichero .zip con el código de la promoción listo para incluir en cualquier página web. Ese botón está marcado con el número 1 en la figura 71.

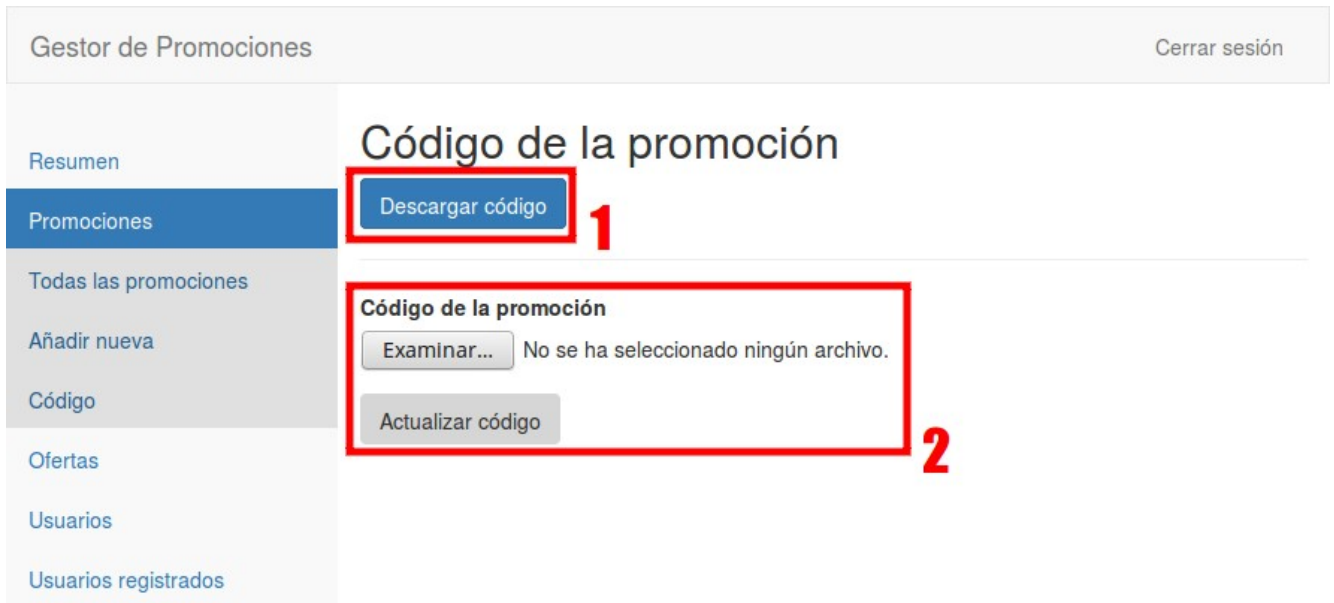


Figura 71: Código de la promoción

Si se quiere actualizar ese código se puede hacer utilizando el formulario representado por el número 2 de la figura 71.

NOTA: esta opción sólo está disponible para los usuarios técnicos y superadministradores.

## Ofertas

El funcionamiento de las ofertas es igual que el de las promociones. El submenú tiene las siguientes opciones:

- Todas las ofertas
- Añadir nueva

Como se puede ver en la figura 72 el panel para las ofertas es idéntico al de las promociones.

Gestor de Promociones Cerrar sesión

Resumen

Promociones

**Ofertas**

Todas las ofertas **1**

Añadir nueva **2**

Usuarios

Usuarios registrados

## Ofertas

3 elementos

Nombre	Nombre público	Oferta padre	Imagen	Fecha
Oferta hija	OFERTA SIN TODOS LOS CAMPOS	Mamma Mía! (Genérica)		12/07/2016 19:25
Mamma Mía! (Genérica)	Mamma Mía!			30/05/2016 20:17
El Rey León (genérica)	El Rey León			30/05/2016 20:16

Figura 72: Todas las ofertas

## Crear, editar y borrar ofertas

La forma de crear, editar y borrar ofertas es idéntica a lo visto en el apartado de las promociones. En lo que se diferencia es en los campos. Como se puede ver en la figura 73 los campos de las ofertas son:

- Nombre
- Oferta padre
- Nombre público
- Descripción
- Imagen
- Label
- Video
- Ciudades
- Testimonios

Las ciudades y los testimonios tienen el mismo comportamiento para la creación, edición y borrado de los códigos de las promociones. El formulario de las ciudades es el de la figura 74 y tiene los campos:

- Nombre
- Descripción
- URL del iframe
- URL del iframe para móvil

El formulario de los testimonios es el de la figura 75 y tiene los campos:

- Testimonio

- Autor

Gestor de Promociones Cerrar sesión

Resumen

Promociones

**Ofertas**

Todas las ofertas

Añadir nueva

Usuarios

Usuarios registrados

## Nueva oferta

**Nombre \***  
Nombre privado para identificar la oferta en el gestor

**Oferta padre**  
Si se selecciona un padre heredará los datos de los campos que se dejen en blanco

**Nombre público**  
Nombre que se mostrará en las promociones

**Descripción**  
Permite etiquetas HTML

**Imagen**  
Tamaño máximo: 1MB. Tipo de imagen aceptada: JPG, PNG y GIF.

 No se ha seleccionado ningún archivo.

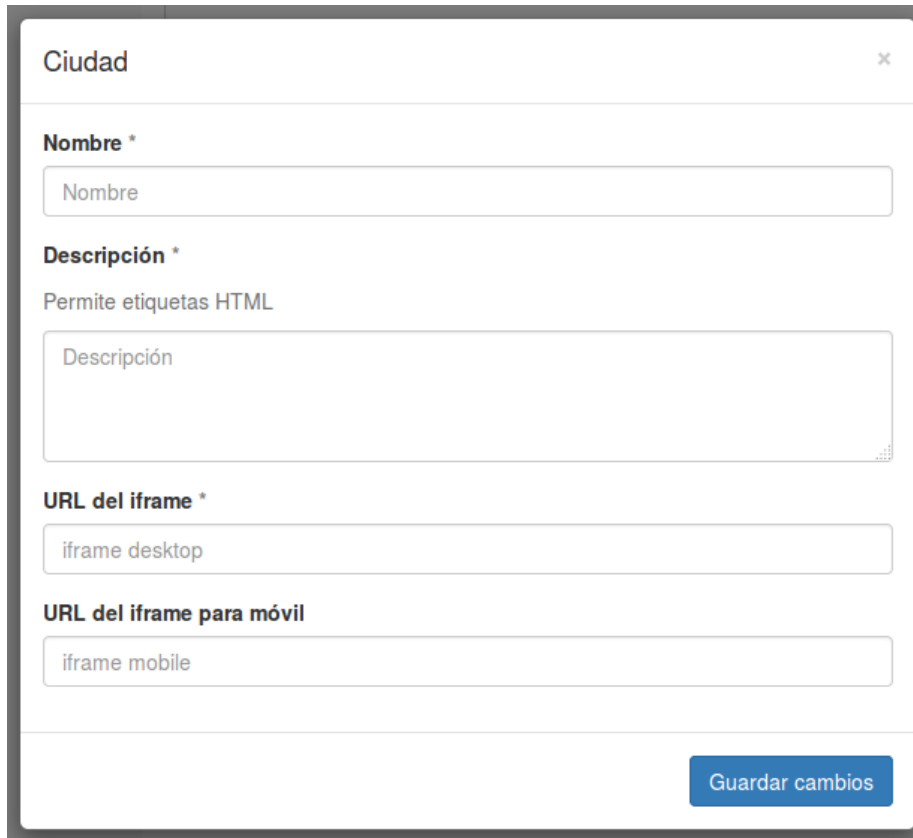
**Label**

**Video**

**Ciudades**

**Testimonios**

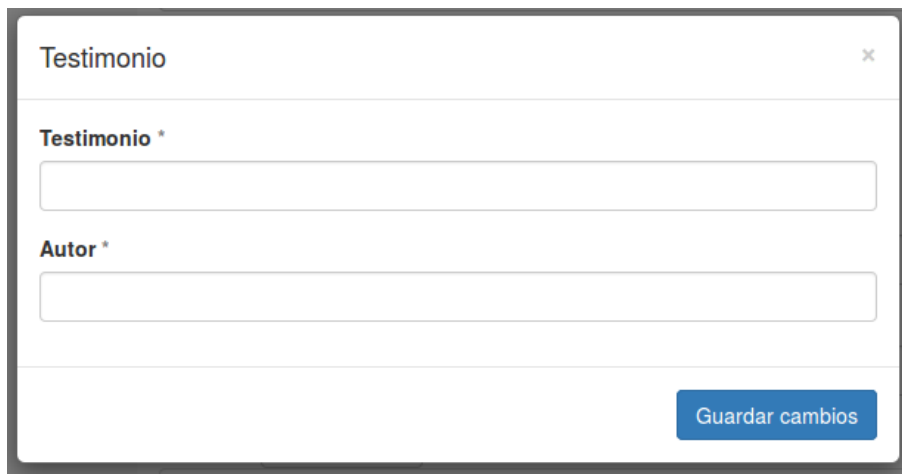
Figura 73: Formulario de oferta



Formulario de ciudad con los siguientes campos:

- Ciudad** (título de la ventana)
- Nombre \*** (campo de texto con el valor "Nombre")
- Descripción \*** (campo de texto con el valor "Descripción", incluye la opción "Permite etiquetas HTML")
- URL del iframe \*** (campo de texto con el valor "iframe desktop")
- URL del iframe para móvil** (campo de texto con el valor "iframe mobile")
- Botón **Guardar cambios**

Figura 74: Formulario de ciudad



Formulario de testimonio con los siguientes campos:

- Testimonio \*** (campo de texto)
- Autor \*** (campo de texto)
- Botón **Guardar cambios**

Figura 75: Formulario de testimonio

## Usuarios del gestor

El funcionamiento de los usuarios es igual que el de las promociones. El submenú tiene las siguientes opciones:

- Todos los usuarios
- Añadir nuevo
- Mi perfil



Como se puede ver en la figura 76 el panel para los usuarios es idéntico al de las promociones.

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
**Usuarios**  
Todos los usuarios  
Añadir nuevo  
Mi perfil  
Usuarios registrados

## Usuarios

Añadir nuevo

2 elementos

Correo electrónico	Rol	Fecha
admin-gestor@example.com	Gestor	02/07/2016 19:13
admin@example.com	Superadmin	12/05/2016 23:39

Figura 76: Todos los usuarios

NOTA: excepto “Mi perfil”, el resto de opciones son únicas de los superadministradores.

## Crear, editar y borrar usuarios

La forma de crear, editar y borrar usuarios es idéntica a lo visto en el apartado de las promociones. En lo que se diferencia es en los campos. Como se puede ver en la figura 77 los campos de los usuarios son:

- Email
- Contraseña
- Rol

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
**Usuarios**  
Todos los usuarios  
Añadir nuevo  
Mi perfil  
Usuarios registrados

## Nuevo usuario

Email \*

Contraseña \*

Repetir contraseña

Rol \*

Crear

Figura 77: Formulario de usuario

## Mi perfil

Acceder a la opción de “Mi perfil” muestra el formulario de edición pero para el usuario con el que estamos identificados. No permite borrar o editar el rol del propio usuario.

## Usuarios registrados

Para los usuarios registrados la única opción que aparece es la de “Todos los usuarios” como se puede ver en la figura 78. Esto es porque sólo se permite acceder a la información de los usuarios que se han registrado.

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
Usuarios  
Usuarios registrados  
**Todos los usuarios** 1

### Usuarios registrados 2

3 elementos

Nombre	Email	Código	Promoción	Fecha
a	apesquera123@brooktec.com	codigo promocional	Promo Renfe	13/07/2016 00:00
a	apesquera@brooktec.com		Promo Renfe	12/07/2016 19:04
alberto	apesquera@pesq.com		Promo Renfe	02/06/2016 21:03

Figura 78: Todos los usuarios

Al seleccionar un usuario de la lista se muestra un formulario con los datos del usuario elegido aunque no se pueden actualizar. Los datos que se muestran son:

- Nombre
- Apellidos
- Código
- Email
- Provincia
- Promoción

Si se accede con un usuario superadministrador se permite borrar el usuario como se ve en la figura 79 en la parte inferior. Para el resto de usuarios esta opción no está disponible.

Gestor de Promociones Cerrar sesión

Resumen  
Promociones  
Ofertas  
Usuarios  
**Usuarios registrados**  
Todos los usuarios

**Usuario: alberto**  
Registrado el 02/06/2016 21:03

**Nombre**

**Apellidos**

**Código**

**Email**

**Provincia**

**Promoción**  
[Promo Renfe](#)  
[Borrar usuario](#)

*Figura 79: Formulario de usuario registrado*

## Capítulo 6. Conclusiones y trabajo futuro

En este capítulo se expondrán las conclusiones extraídas del desarrollo de este proyecto. También se definirán las líneas futuras que pueden seguirse para la continuación del trabajo realizado.

### Conclusiones

El resultado de este proyecto ha sido satisfactorio ya que se ha conseguido el objetivo de crear una aplicación web que gestione las promociones y sus ofertas. Además ha servido para aprender a desarrollar en el framework PHP Slim, el gestor de vistas Twig, el gestor de base de datos Doctrine y la tecnología Angular2. Otras de las tecnologías con las que se ha tenido la oportunidad de trabajar son el software de configuración de entornos virtualizados Vagrant, el automatizador de tareas de desarrollo Gulp, las tecnologías web HTML5, JavaScript y CSS, entre otras.

Además adquirir los conocimientos informáticos, realizar el proyecto con la metodología que se utiliza en la empresa Brooktec ha ayudado a aprender el funcionamiento de metodologías ágiles, en especial Scrum, tanto de forma teórica como práctica.

### Trabajo futuro

Una posible mejora de este proyecto es mejorar la parte visual del panel de administración del gestor, ya que se ha priorizado que sea funcional frente a que sea bonito estéticamente.

Otra línea de trabajo futuro puede ser el desacoplamiento de la funcionalidad del gestor y de la parte visual con la creación de una API, de tipo Rest por ejemplo, que permita la creación, lectura, actualización y eliminación de las promociones y las ofertas a través de peticiones HTTP/HTTPS con esa API. Esto permite a cualquier desarrollador crear distintas formas visuales de acceder a los datos manteniendo siempre la misma funcionalidad en lugar de obligar a utilizar el gestor creado.

También se puede portar la aplicación utilizando la potencia de Slim3 para los servidores que soporten la versión de PHP 5.5 o superior.

Si no se prefiere PHP, se puede hacer una aplicación web similar a ésta en otras tecnologías como Java y Spring, Node.js y Express, Python y Flask, etc. Si se diera este caso, se podría hacer un estudio sobre las prestaciones de las distintas tecnologías para la misma tarea, o un análisis de complejidad de desarrollo, curva de aprendizaje y tamaño de la aplicación por cada tecnología.

# Bibliografía

## Metodología

- [1] *Manifiesto por el Desarrollo Ágil de Software.*  
<http://www.agilemanifesto.org/iso/es> [Consulta: 08 de julio de 2016]
- [2] Wikipedia, *Historias de usuario*  
[https://es.wikipedia.org/wiki/Historias\\_de\\_usuario](https://es.wikipedia.org/wiki/Historias_de_usuario) [Consulta: 08 de julio de 2016]
- [3] Javier Garzás, *Inception en contextos ágiles: dejemos las ideas claras desde el primer momento*  
<http://www.javiergarzas.com/2015/09/inception-en-contextos-agiles.html> [Consulta: 09 de julio de 2016]
- [4] Satheesh Thekku Veethil, *Risk Management in Agile*  
<https://www.scrumalliance.org/community/articles/2013/2013-may/risk-management-in-agile> [Consulta: 09 de julio de 2016]
- [5] Mike Cohn, *Managing Risk on Agile Projects with the Risk Burndown Chart*  
<http://www.mountangoatsoftware.com/blog/managing-risk-on-agile-projects-with-the-risk-burndown-chart> [Consulta: 09 de julio de 2016]

## Tecnologías

- [6] Página oficial de Vagrant  
<https://www.vagrantup.com/> [Consulta: 06 de julio de 2016]
- [7] Página oficial de Node.js  
<https://nodejs.org/> [Consulta: 08 de julio de 2016]
- [8] Página oficial de Gulp  
<http://gulpjs.com/> [Consulta: 08 de julio de 2016]
- [9] Página oficial de Slim  
<http://www.slimframework.com/> [Consulta: 08 de julio de 2016]
- [10] Eric L. Barnes, *PHP Version Stats: May 2016*  
<https://laravel-news.com/2016/06/php-version-stats/> [Consulta: 06 de julio de 2016]
- [11] Página oficial de Twig  
<http://twig.sensiolabs.org/> [Consulta: 09 de julio de 2016]
- [12] Página oficial de Doctrine

<http://www.doctrine-project.org/> [Consulta: 25 de junio de 2016]

[13] Página oficial de Angular2

<https://angular.io/> [Consulta: 25 de junio de 2016]

## Documentación y ejemplos prácticos

[14] Documentación oficial de Slim

<http://docs.slimframework.com/> [Consulta: 11 de julio de 2016]

[15] Ravi Tamada, *How to create REST API for Android app using PHP, Slim and MySQL*

<http://www.androidhive.info/2014/01/how-to-create-rest-api-for-android-app-using-php-slim-and-mysql-day-12-2/> [Consulta: 14 de junio de 2016]

[16] Documentación oficial de Doctrine ORM

<http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/index.html> [Consulta: 11 de julio de 2016]

[17] Stack Overflow, *Angular2 validator which relies on multiple form fields*

<http://stackoverflow.com/questions/31788681/angular2-validator-which-relies-on-multiple-form-fields> [Consulta: 22 de junio de 2016]

[18] ng-book, *The Ultimate Guide to Forms in Angular 2*

<http://blog.ng-book.com/the-ultimate-guide-to-forms-in-angular-2/> [Consulta: 22 de junio de 2016]

[19] Radzishvskyi Sergey, *Angular2 – REST sample app, part 2*

<http://radzserg.com/2016/04/11/angular2-rest-sample-app-part-2-create-update/>  
[Consulta: 28 de junio de 2016]

# Anexo

## Contenido del CD

El contenido del soporte digital que acompaña esta memoria es:

- Archivo **memoria.pdf**: este fichero.
- Archivo **readme.txt**: fichero de texto plano cuya lectura es recomendada dado que contiene información relevante sobre el contenido del CD.
- Carpeta **servidor-webapp**: el directorio con código del proyecto del gestor de promociones.
- Carpeta **cliente-promocion**: el directorio con el código del ejemplo de uso de la API en Angular2.

## Documentación de la API

A continuación se definen las rutas relativas al servidor donde está alojada la aplicación. Para mostrar ejemplos de una URL completa se supondrá que la aplicación está en el servidor [www.example.com](http://www.example.com).

### Obtener la promoción

- Ruta de la llamada: `"/api/v0/promotion/<id>"`
  - Parámetros:
    - `id`: integer. El ID de la promoción.
  - Ejemplo: `www.example.com/api/v0/promotion/42`
- Tipo: GET
- Respuesta:
  - Headers:

<i>Content-Type: application/json</i>
---------------------------------------

- Body:

```
{
  "name": "Promo 42",
  "description": "Solo por ser cliente de Promoción, podrás beneficiarte...",
  "image": "logo-abcdefghij012345679.jpg",
  "formLogin": {
    "code": true,
    "email": true
  },
}
```

```
"formRegister": {
  "code": true,
  "name": true,
  "lastname": true,
  "email": true,
  "province": true
}
```

## Registrar un usuario

- Ruta de la llamada: “/api/v0/promotion/<id>/register”
  - Parámetros:
    - id: integer. El ID de la promoción.
  - Ejemplo: [www.example.com/api/v0/promotion/42/register](http://www.example.com/api/v0/promotion/42/register)
- Tipo: POST
- Petición:
  - Headers:

```
Content-Type: application/json
```

- Body:

```
{
  "name": "Alberto",
  "lastname": "Pesquera",
  "email": "alberto@example.com",
  "code": "codigopromocional"
}
```

- Respuesta:
  - Headers:

```
Content-Type: application/json
```

- Body:

```
{
  "user": {
    "id": 42
  },
  "offers": [
```



```

    {
      "name": "Nombre de la oferta",
      "description": "Descripcion de la oferta",
      "image": "logo-4lf4num3r1c0.jpg",
      "label": "Etiqueta",
      "video": "codigo-youtube",
      "cities": [
        {
          "name": "Ciudad",
          "description": "Descripcion",
          "iframe-desktop": "codigo-iframe",
          "iframe-mobile": "codigo-iframe"
        }
      ],
      "testimonies": [
        {
          "testimony": "Testimonio real",
          "author": "Autor del testimonio"
        }
      ]
    }
  ]
}

```

## Identificar un usuario

- Ruta de la llamada: “/api/v0/promotion/<id>/login”
  - Parámetros:
    - id: integer. El ID de la promoción.
  - Ejemplo: [www.example.com/api/v0/promotion/42/login](http://www.example.com/api/v0/promotion/42/login)
- Tipo: POST
- Petición:
  - Headers:

*Content-Type: application/json*

- Body:

```
{  
  "email": "alberto@example.com"  
}
```

- Respuesta:
  - Headers:

```
Content-Type: application/json
```

- Body:

```
{  
  "user": {  
    "id": 42  
  },  
  "offers": [  
    {  
      "name": "Nombre de la oferta",  
      "description": "Descripcion de la oferta",  
      "image": "logo-4lf4num3r1c0.jpg",  
      "label": "Etiqueta",  
      "video": "codigo-youtube",  
      "cities": [  
        {  
          "name": "Ciudad",  
          "description": "Descripcion",  
          "iframe-desktop": "codigo-iframe",  
          "iframe-mobile": "codigo-iframe"  
        }  
      ],  
      "testimonies": [  
        {  
          "testimony": "Testimonio real",  
          "author": "Autor del testimonio"  
        }  
      ]  
    }  
  ]  
}
```

*I*  
}