



Proyecto de Ingeniería del Software 2013/2014

Where2Night

“La red social para salir de fiesta”

Documentación

Versión 1.4

Componentes del grupo: Javier Alejos Castroviejo
Juan Brugera Monedero
Luis Enrique Cubero
Alicia Daza Castillo
Adrián Díaz Jiménez
Saskya Mosquera Logroño
Beatriz Ortega de Pedro
Álvaro Pascual Crespo
Yuleska Perez Lozada
Sergio Primo Galán
Ismael Requena Andreu

Email de contacto: contacto@where2night.es

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Historial de revisiones

Fecha	Versión	Descripción	Autor
19/01/2014	1.0	Creación Documento	Javier Alejos
21/01/2014	1.1	Revisión y modificación de estilos	Luis Cubero
21/01/2014	1.2	Cambio de tecnología a PHP	Sergio Primo
21/01/2014	1.2.1	Revisión y modificación de estilos	Yuleska Pérez
21/01/2014	1.3	Riesgos y revisión de formato y añadir anexos	Álvaro Pascual

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Tabla de contenidos

1. Introducción	6
1.1. Definición del proyecto	6
1.2. Metodología de trabajo	6
1.3 Tecnologías utilizadas	7
1.3.1 ¿Por qué elegir una App nativa frente a una en HTML5?	7
1.3.2 ¿Por qué desarrollar nuestro proyecto en Android Nativo?	8
1.3.3 Herramientas para el desarrollo Android	8
2. Documentación técnica	17
2.1 Visión	18
2.2 Escenarios	17
2.2.1 Escenario 1	17
2.2.2 Escenario 2	17
2.2.3 Escenario 3	18
2.3 Especificación de requisitos software	22
2.3.1 Especificaciones Local	22
2.3.2 Especificaciones Fiestero	24
2.3.3 Especificaciones DJ	27
2.3.4 Búsquedas	29
2.3.5 Check-In	31
2.3.6 Gramola	31
2.3.6 Gestión de amistades	32
2.3.7 Visión webapp administración y dj	33
2.4 Gestión de riesgos	34
2.5 Casos de uso	46
2.5.1 Registro de local	46
2.3.2 Gramola sin check-in	48

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.3.3 Gramola con check-in	49
2.3.4 Seleccionar canción “Qué está sonando Dj/Local”	51
2.3.5 Seleccionar canción “Qué está sonando (10 +v) Dj/Local”	52
2.3.6 Check-in (manual).....	54
2.6 Diagramas	57
2.6.1 Diagrama de clases	58
2.6.2 Diagrama de la base de datos.....	59
2.7 Mockups.....	59
2.7.1 WEB.....	60
2.7.2 Android	64
3. Planificación.....	66
3.1 EDT y diagrama de Gantt.....	66
3.2 Actividades estructurales (métrica de horas / persona).....	66
3.2.1 Android	67
3.2.2 Web	68
3.2.3 Back-end.....	68
3.3 Burndown chart	70
4. Documentación referencial.....	71
4.1 Guía de estilos.....	71
4.1.2 Introducción	71
4.1.3 Formato	71
4.1.4 Especificaciones generales.....	71
4.1.5 Guía programación	73
5. Documentación de organización	75
5.1 Acta reunión 1	75
5.2 Acta reunión 2	76
5.3 Acta reunión 3	77
5.4 Acta reunión 4	78
5.5 Acta reunión 6	78
5.5 Acta reunión 7	79

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

6. Conversaciones con profesionales del sector	81
6.1 Conversación con DJ.....	81
6.2 Conversación con propietario del local	81
6.3 Conversación con miembro de grupo de música	81
6.4 Conversación sobre listas de discoteca	82
7. Modelo de negocio.....	83
7.1 Listas de locales.....	83
7.2 Promoción de locales	83
8. Anexos.....	84
8.1. ANEXO 1 - EDT.....	84
8.2. ANEXO 2 - Diagrama de Grantt. Planificación de tareas	86
8.3. ANEXO 3 - Diagrama de Grantt.....	90

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

1. Introducción

Este documento contiene una visión del proyecto Where2Night. La visión y la motivación inicial del proyecto, además de documentación software que se ha utilizado.

1.1. Definición del proyecto

Este proyecto se utiliza para planificar y mejorar las salidas en la noche de Madrid. Se divide en cuatro partes principales. Cada una de éstas intenta cubrir las necesidades de los clientes en los distintos ámbitos nocturnos y también facilitar a los locales diversas posibilidades para mejorar la atención a sus clientes.

Cuando una persona sale de fiesta por la noche siempre surge la duda de dónde ir, si el local estará muy lleno, que tipo de gente hay dentro, el precio de las consumiciones o si la música es acorde a sus gustos. Where2Night facilita estas decisiones de tal forma que un usuario no tendrá que preocuparse nunca más por tomar estas decisiones.

Además, Where2Night proporciona a los locales funcionalidades para satisfacer los gustos de los clientes, tales como posibilidad de elegir música, graficas estadísticas para hacer fiestas temáticas o la posibilidad de apuntarse a listas o eventos sin la necesidad de contactar con algún Relaciones Publicas.

El objetivo de este proyecto ha sido crear una aplicación en Android que controlara todas las funciones de un usuario normal, a partir de ahora "Fiestero" y una página Web para las funcionalidades tanto de los Locales como de los Fiesteros.

1.2. Metodología de trabajo

Nuestro proyecto ha seguido la metodología SCRUM, con grupos de 2 a 4 personas que han ido alternando entre los grupos dependiendo de las necesidades del SPRINT. Las reuniones principales han sido mensuales, coincidiendo con el final del Sprint. Además de una reunión semanal vía Skype y contacto prácticamente continuo por otros medios de comunicación.

Hemos seguido un modelo de organización Descentralizado Controlado, donde el Scrum Master ha sido Ismael Requena Andreu.

El repositorio utilizado ha sido **GitHub** (<https://github.com/where2night>). En donde ha habido cuatro ramas: una rama Android, una rama para el BackEnd, una rama para la Web y otra para la Documentación.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Los métodos de comunicación entre los compañeros de proyecto han sido:

- Google Docs para la resolución de preguntas y creación de documentos.
- Reuniones a través de Google Hangouts o Skype
- Comunicación rápida: Whatsapp, Email.

La división de grupos de trabajo final ha sido la siguiente:

- Android: Álvaro Pascual , Ismael requena, Juan Brugera
- BackEnd: Adrián Díaz, Luis Cubero, Sergio Primo
- Web: Javier Alejos, Saskya Mosquera, Alicia Daza, Yuleska Pérez, Beatriz Ortega
- Diseño: Sergio Primo, Yuleska Pérez, Saskya Mosquera

1.3 Tecnologías utilizadas

Para decidir qué tecnología utilizar hemos realizado las siguientes comparativas:

1.3.1 ¿Por qué elegir una App nativa frente a una en HTML5?

1. Mejor experiencia de usuario: Características como NFC, uso de la cámara, notificaciones y vibración no son posibles de usar con html5 (<http://mobilehtml5.org/>) Con HTML5 obtienes una experiencia suficientemente buena pero con una app nativa sacas lo mejor.
2. Rendimiento: Las app nativas son más rápidas y eficientes.
3. Multiplataforma: Aquí HTML5 es mejor, su esencia es “write one, run anywhere” pero la cantidad inmensa de interpretaciones por parte de los navegadores de HTML5 hace que esto cambie y sea “write once, optimize everywhere”.
4. Actualizar la app: HTML5 mejora a Android Nativo, tener la app como una web permite una actualización más rápida.
5. Seguridad: El código de HTML5 es más inseguro, dado que cualquiera puede ver el código fuente. Los datos cacheados sólo pueden ser encriptados con las app nativas. Si se encriptan los datos que intercambia la App con el servidor en HTML5 compromete la velocidad.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

6. Innovación: Cada año salen distintas actualizaciones de los OS mientras que HTML5 ha tardado 5 años en ser ratificado por W3C.
7. Documentación: Google proporciona una documentación de la API totalmente detallada y actualizada a la última versión. Además la comunidad de desarrolladores es más grande en la App nativa y cualquier problema que podamos encontrar puede estar subsanado.

1.3.2 ¿Por qué desarrollar nuestro proyecto en Android Nativo?

Es de esperar que si vamos a realizar Login con Facebook podamos usar las credenciales cacheadas y si el Login es propietario también deberíamos cachear las credenciales para que el usuario no tenga que introducirlas constantemente. (Apartado 5 anterior)

Para las peticiones de amistad podríamos necesitar una notificación para avisar al usuario de que le han agregado (Apartado 1 anterior)

Para hacer CheckIn en un local posiblemente nos veamos obligados a usar tecnologías como Bluetooth o Wifi además de la Geo-Localización y HTML5 no puede hacer uso de la Wifi del dispositivo. (Apartado 1 anterior).

1.3.3 Herramientas para el desarrollo Android

Lo primero que hay que preguntarse es, **¿Qué lenguaje utilizar?**

Existen múltiples plataformas en muchos lenguajes para programar Android, pero ¿Cuál es mejor?

Lo más normal es pensar primero con que lenguaje se siente uno más cómodo, pero lo realmente importante es cuál nos ofrece más ventajas.

En este punto Google decidió que para programar Android de forma nativa se usaría Java, con lo que es la mejor opción y más sencilla, sin embargo ya que la mayoría del SO está escrito en C o C++ existe la opción de usar NDK, aunque los propios desarrolladores en su página indican que no es beneficioso para la mayoría de aplicaciones, únicamente para cosas específicas, con lo cual es ganador es **Java**.

Una vez tenemos definido el lenguaje la cuestión es, **¿Qué herramienta usamos?**

Evidentemente existen infinidad de herramientas para hacer una aplicación Android, no todas con las mismas características y no todas igual de buenas.

Lo más sencillo es contar con un editor gráfico que ahorre mucho código XML necesario para la GUI, con lo que descartamos todas las que no lo tengan.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

A partir de aquí, solo falta ver cual nos da más posibilidades. Lo más conocido es Eclipse (<http://www.eclipse.org>), entorno en el que programan la mayoría de desarrolladores Android, aunque Google ha lanzado un nuevo entorno llamado Android Studio, el cual aún es desaconsejable, puesto que no es un proyecto terminado y genera bugs muy evidentes, además de no tener todas las opciones disponibles.

La otra opción más utilizada es el Plugin para NetBeans. Realmente hay una fuerte discusión sobre cual es mejor, pero al final cada uno acaba usando con la que se siente más cómodo. Parece que el Plugin de Eclipse (<http://developer.android.com/sdk/index.html>) es algo mejor, más sencillo de utilizar, más fiable y más desarrollado puesto que Google le daba su apoyo, aunque el emulador consume más recursos del PC.

Por tanto el ganador es **Eclipse**.

1.3.4 Ruby vs PHP (BackEnd)

Se compararán los lenguajes en varios aspectos, destacando el ganador de cada aspecto. Antes de cada aspecto, se explicará brevemente porque se considera un factor determinante en la comparación.

Nota (07/01/2014): Debido a problemas con el servidor, como la falta de librerías, nos vimos obligados a tomar en cuenta el cambio PHP. Además se sumó esto a los problemas de aprendizaje de Ruby, lenguaje que sólo dominaba un miembro del grupo, el cual no estuvo muy disponible para gestionar todos los problemas relacionados con el lenguaje.

Por tanto para facilitar las cosas al resto de miembros del grupo, que sí conocen en su mayoría PHP, agilizar el desarrollo, y evitar problemas con el servidor, decidimos cambiar a **PHP** a tiempo.

Filosofía.

La filosofía de cualquier lenguaje determina ampliamente el tipo de desarrollo que se puede hacer fácilmente en dicho lenguaje. Además la forma de trabajar también viene determinada por la filosofía del mismo.

Está claro que con cualquier lenguaje podemos hacer de todo, pero elegir el lenguaje que más se adapte a tu desarrollo es vital, por lo que la filosofía no debería de olvidarse.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- **RUBY:** Nace para ser un lenguaje orientado a objetos, divertido y eficiente. Enfatizado sobre todo en las necesidades de los humanos y no de los ordenadores.
- **PHP:** Nace para permitir a los programadores realizar rápidamente páginas web dinámicas.
- **Ganador:** En breves palabras, PHP nace de una necesidad y su crecimiento ha sido descontrolado; mientras que el **Ruby** fue pensado como un lenguaje serio y estructurado.

Facilidad de aprendizaje

Debido al desconocimiento de los lenguajes comparados, se cree importante hacer un listado de los lenguajes a los que se le parecen. Ya que aquel que comparta más lenguajes que conozcamos nos será más fácil de aprender y leer.

- **RUBY:** ADA, C++, PERL, PYTHON (similar a JAVA)
- **PHP:** C, PERL, JAVA, C++, TCL
- **Ganador:** Como en realidad los dos son bastante similares en cuanto a sintaxis, se considerará como un **empate**.

¿Es estricto el lenguaje?

Los lenguajes estrictos pueden verse como una ventaja o desventaja. Es verdad que un lenguaje muy estricto requiere un esfuerzo mayor a la hora de aprender la sintaxis, sin embargo ayuda mucho a la unificación del código. Probablemente se aprendería a programar más rápido en lenguajes poco estrictos, sin embargo se obtendrían malos hábitos que no permitirían un desarrollo sostenible debido a falta del conocimiento profundo de la arquitectura del mismo.

Por lo que consideramos que un lenguaje estricto es una ventaja en un equipo grande debido a la uniformidad y manutención de un gran proyecto. Y, aunque también es perjudicial al crecimiento del proyecto, nosotros nos vamos a enfocar en desarrollar un proyecto sólido.

Restricciones en RUBY:

- No existen parámetros de salida, todas las expresiones devuelven un valor. La última línea en un método se considera como el parámetro de salida.
- No existen corchetes, en vez de eso se define el scope mediante la indentación.
- Es case sensitive.
- No permite mezclarse con HTML

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Restricciones en **PHP**:

- Todas las variables necesitan un “\$” delante.
- Permite mezclarse en el HTML

Ganador: Dado que se busca una uniformidad en el proyecto en general, el ganador es **Ruby** por la indentación y la restricción de no mezclarse en el HTML.

Legibilidad.

La legibilidad de un lenguaje es fundamental a la hora de compartir código y reparar el mismo. Un programador no debería invertir tiempo en descifrar el código de otro sino en colaborar y ayudar.

- Hola mundo en **RUBY**: puts p “hello world” + “ from my part”
- Hola mundo en **PHP**: <p> <?php print(“Hello world”); ?> from my part</p>
- **Ganador:** Evidentemente **RUBY** es más fácil de leer.

Modernidad.

Debido a nuestro dominio, es fundamental que el lenguaje se adapte a metodologías modernas y cuente con herramientas modernas. Se comparará la adaptación con una herramienta de API REST, la cual es fundamental para la comunicación de los clientes con el servidor.

- API REST implementada en **RUBY**:

```
require 'sinatra'

get '/hi' do
  "Hello World!"
end
```

- API REST implementada en **PHP**:

[más de 100 líneas de código]

- **Ganador: Ruby**, ya que al ser un lenguaje moderno la incorporación de herramientas modernas es casi nativa.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Mantenibilidad.

La manutención es la capacidad que los lenguajes tienen para ser reparados. Esta misma depende sobre todo de la legibilidad y de su adaptación a un entorno de pruebas; por lo que se apoyará en aquellos puntos.

- **Ganador:** Como hemos visto antes, **RUBY** es un lenguaje más fácil de leer y de integrar con un sistema orientado a pruebas.

Escalabilidad.

La escalabilidad es la capacidad que nos ofrece el lenguaje de incorporar nuevas funcionalidades al menor coste posible. Idealmente se buscaría que le dijéramos al lenguaje que implemente automáticamente una funcionalidad y que está fuera compatible con toda nuestro proyecto y futuras funcionalidades, pero ya que esto no es real nos enfocaremos en la cantidad de librerías open-source de cada uno.

- Librerías en **RUBY**: RUBY cuenta con un instalador de servicios y módulos llamado GEM. Cada librería o módulo es un GEM, y para utilizarlos en el proyecto basta con declararlos en un fichero de GEM.FILE. Las dependencias de estas se resuelven automáticamente.
- **PHP**: PHP también cuenta con un gestor de paquetes llamado PEAR. Sin embargo la comunidad de PHP no utilizó mucho dicha librería. Se encuentra más opciones sueltas por Internet, por lo que nos vemos obligados a resolver las dependencias manualmente.
- **Ganador:** Consideramos que es una ventaja para **RUBY** el hecho de que su comunidad esté comprometida con su gestor de paquetes.

Orientado a pruebas

Como lo que se desarrollará con el lenguaje elegido será el hueso, corazón y núcleo de nuestra aplicación es fundamental que esté construido a prueba de fuego. La facilidad que ofrecen para ser probados de una forma automática afectará directamente en la calidad de las pruebas y por lo tanto en la calidad final de la aplicación.

- Ejemplo de pruebas en unidad en **RUBY**:

```
test "should not allow access" do

  get :new
  assert_redirected_to new_user_session_path(:format => 'mobile')

end
```

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Ejemplo de pruebas en unidad en **PHP**:

```
--TEST--
foo() function - A basic test to see if it works. :)
--FILE--
<?php
include 'lib.php'; // might need to adjust path if not in the same dir
$bar = 'Hello World';
var_dump(foo($bar));
?>
--EXPECT--
string(11) "Hello World"
```

- **Ganador:** En este caso ambos son muy similares, sin embargo la instalación de las librerías de pruebas en PHP, es mediante PEAR (su instalador de librerías) y unas modificaciones al APACHE (servidor web); mientras que en el caso de **RUBY** su integración es la inclusión de librerías sin necesidad de modificar el servidor web (APACHE/NGIX).

Seguridad

La seguridad en el proyecto no es un factor vital, pero si de alta importancia. Sin embargo el tipo de seguridad requerido es muy básico, dado que la información de nuestro sistema es también información que un usuario comparte en sus redes sociales.

Según algunos comentarios, PHP es menos seguro que Ruby, pero nada completamente demostrado, con lo que el resultado es un empate.

Comunidad

La comunidad de usuario es esencial para resolver dudas y resolver problemas que nos superen. Destacamos principalmente la calidad de la comunidad antes que la cantidad y también miramos mucho en que dicha comunidad sea mayoritariamente de habla hispana.

- **RUBY**, cuenta con una comunidad de usuario de alrededor del 4% sin embargo la calidad de las respuestas suele ser excelente debido a que su comunidad es mayormente de ingenieros.
- **PHP**, cuenta con una comunidad de usuario de alrededor del 7% y con calidades muy variadas, pero en general muy buena.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- **Ganador:** La comunidad de RUBY nos ofrece mejores soluciones, pero quizá no encontremos siempre la solución. Mientras que en PHP siempre encontraremos una respuesta, pero tendremos que ser capaces de juzgar la calidad de dicha. Por lo que se considera a **PHP** como ganador.

Repercusión

Debido a que nuestro futuro aún no está escrito, es importante que el lenguaje tenga una gran demanda de empleo. Sin embargo entre mayor sea la oferta de programadores, menor será el sueldo. Por lo que se valorará la cantidad de la demanda y por otra parte el salario promedio de las ofertas.

- **RUBY:** Las ofertas de trabajo de RUBY son muy pocas pero con un sueldo muy alto; a partir de los 1500€.
- **PHP:** Hay muchas ofertas de trabajo en PHP con sueldos muy variantes, desde sueldo precarios hasta sueldos de 2500€.
- **Ganador:** Consideramos que es una muy buena alternativa destacar, por lo que el ganador es **RUBY**.

Servidor

El acceso al servidor y su precio es algo muy importante a tener en cuenta, así como disponer de una gran variedad de servidores entre los que poder elegir.

- **PHP:** Al ser mucho más utilizado PHP, las librerías están instalados en todos los servidores que se puedan contratar, pudiendo encontrar el servidor que más se adapte a tus necesidades debido a la gran variedad de la que disponemos.
- **RUBY:** Al ser un lenguaje mucho menos utilizado, la oferta de servidores con las librerías adecuadas instaladas es bastante reducida, por tanto los precios de estos suelen ser más altos.
- **Ganador:** Desde luego aquí el ganador es **PHP**, debido a la alta oferta de sus servidores, ofreciendo precios muy competitivos.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Conclusión

Consideramos que los aspectos más importantes en nuestra aplicación son la facilidad de aprendizaje, legibilidad, modernidad, orientados a pruebas y seguridad. Dichos aspectos son los que benefician más al proyecto ya que lo hacen tanto sólido como rápido. Por lo que la tecnología elegida para el desarrollo del servidor de servicio web es **RUBY**.

1.3.5 Comunicación entre Cliente y Servidor

Para la comunicación entre el cliente y el servidor utilizaremos ficheros JSON. Actualmente son los más utilizados, ya que evitan crear código XML, aunque realmente se parecen bastante. Sin embargo existen librerías como Google GSON que convierten automáticamente cualquier objeto a Strings de JSON con lo que facilita mucho el trabajo.

En cuanto a la web también es posible usar JSON, con lo cual también lo utilizaremos.

De esta forma solo será necesario que el backend devuelva un único fichero JSON con cada petición, ya que tanto la web como la aplicación como cualquier plataforma futura serán capaces de convertirlo.

Por lo tanto después de estas comparativas, nuestra selección de tecnologías para el proyecto ha sido:

- Android nativo para app móvil de usuario.
- PHP para el BackEnd
- MySQL para la base de datos.
- HTML5, Javascript, CSS3 para portal Web Usuario y administración del local.

Todas estas tecnologías se iban a usar orientadas a test aunque finalmente por falta de tiempo no ha sido así.

Otras herramientas utilizadas para la gestión del proyecto han sido:

- Balsamiq para diseñar los mockups.
- Trello para gestión del reparto de tareas Scrum.
- GitHub para el control de versiones.
- Office y Google Docs para la creación de documentos

1.3.6 MySQL para la Base de Datos (BackEnd)

MySQL DataBase frente a sus adversarios presenta frecuentes ventajas, que bien pueden ser la razón por la cual es la base de datos de código fuente abierto más usada del mundo.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Decir Open Source (código abierto) representa que el usuario puede usarle libremente y hacer ajustes en el código para maximizar su funcionamiento, anexando mejoras a la base de datos.

De acuerdo a lo que se sabe de los programas de código abierto, se tiene la experiencia de que su éxito depende en mucho del concepto abierto, ya que existen un gran número de desarrolladores que están luchando por mejorar el código y este se mantiene actualizado.

Pero no por decir que es de código abierto, solo es libre, claro que existe la versión comercial que sin duda tendrá complementos a las aplicaciones del programa.

MySQL es un sistema de bases de datos relacional multiusuario que es distribuido bajo licencia GNU GPL.

- Probado con una amplia gama de diversos compiladores.
- Las plataformas que utiliza son muy variadas y entre ellas están LAMP, MAMP, SAMP, BAMP y WAMP (aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Python, sin ser estas las únicas).
- Aplicaciones GNU Automake, AutoConf y Libtool para la portabilidad.
- Multi-roscaado completamente usando los hilos de rosca del núcleo.
- Su conectividad y robustez son invaluables
- Es escalable
- Acceso a la base de datos de forma simultánea.
- Seguridad en cuanto a los permisos y privilegios.
- Permite recurrir a bases de datos multiusuario a través de la web.
- Puede ser ejecutado en una máquina con escasos recursos.
- Facilidad de configuración e instalación.
- Bajo índice ante la probabilidad de corromper datos, inclusive si los errores no se producen en el propio gestor, sino en el sistema en el que está instalada la base de datos.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2. Documentación técnica

Este apartado incluye la documentación técnica de proyecto: Visión, escenarios, requisitos, riesgos, casos de uso, diagramas y mockups.

2.1 Escenarios

2.1.1 Escenario 1

Es un viernes por la tarde. Nuestro protagonista acaba de salir de clase. Tiene ganas de salir de fiesta para calmar el estrés producido durante la semana, pero esa misma noche no tiene ningún plan, así que, coge su Smartphone y se mete en la app Where2Night recién instalada por recomendación de algunos de sus amigos. Se registra introduciendo sus datos (nombre, apellidos, sexo, fecha de nacimiento, email, música y bebida favorita).

El usuario, previamente autenticado, navega por la aplicación. Cuando se encuentra en la pantalla de añadir amigos, ve a todos los amigos que tienen la app ya instalada y los agrega directamente a su lista de amistades.

Después entra en el perfil de los amigos de la facultad y en la opción de posible destino ve que algunos van a ir a “Cats” y otros van a ir a “Moondance”.

Entonces se dirige al buscador y este le indica el número de amigos en cada local. Como van a ir más amigos a “Cats” decide ir hacia allí para apuntarse a la fiesta.

Escribe en la opción posible destino “Cats” y deja un comentario avisando a sus amigos que también va.

2.1.2 Escenario 2

De camino al local, mientras espera al autobús, el usuario entra en el perfil de “Cats” y observa que en el local hay mayor número de chicas que de chicos y que la bebida favorita de la mayoría es Malibu piña.

También se da cuenta de que justo hoy la entrada es más barata, puesto que, hay una oferta especial bastante asequible. En principio parece que el usuario ha elegido bien el destino.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.1.3 Escenario 3

Una vez en el local, el usuario hace Check-In. Al entrar, se encuentra con la mayoría de los amigos que iban a ir (algunos cambiaron de destino gracias a Where2Night porque vieron que la música no les gustaba y finalmente eligieron otro local).

Pasadas unas horas, ponen varias canciones seguidas que no son del gusto del usuario y sus amigos, así que, votan todos por “Rock That Body” de “The Black Eyed Peas” y al poco tiempo la ponen.

2.2 Visión

Para el apartado visión nos hemos basado en preguntas que un fiestero se hace cuando está planeando su noche.

¿Dónde voy a salir?

Esta parte se centra en dar información útil para planificar las salidas nocturnas. Ésta a su vez se dividirá en un perfil de local, un listado de locales favoritos, rutas de locales y un buscador de locales.

Perfil Local

Cada local tendrá una descripción generada por el dueño del local dicha información se hará visible a los usuarios fiesteros para que les facilite la decisión de selección local.

Listado favoritos

Aquí el usuario podrá guardar un listado de sus locales favoritos, que podrá o no compartir con sus amigos.

Ruta de locales

La idea de la ruta de los locales, es para que los usuarios puedan continuar con un mismo ambiente (de personas y/o música) cuando un bar esté cerrando o el ambiente haya cambiado.

También se podrá generar rutas temáticas. Los temas podrían ser: locales “Destroyer” (modo

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

de salir de fiesta que hemos incluido en la aplicación), los más baratos, ruta alternativa, arranca despacio y termina fatal, etc.

Buscador

El buscador ayudará al usuario a encontrar los bares ya sea por precio, ambiente, tipo de personas, cantidad de amigos, frecuencia de visita o trending topic.

¡Quiero escuchar esta canción!

Esta otra parte se centra en el usuario que ya ha elegido el local. Facilitará al usuario la interacción con el bar, desde pedir canciones hasta ligar. Los principales objetivos de esta sección son: solicitar canciones e información sobre la música que se está escuchando.

Solicitar canciones

Las personas podrán votar canciones para escuchar. Estas votaciones pueden ser por canciones específicas, por género o por canciones similares a otra canción. Las canciones específicas deberán estar previamente proporcionadas por el bar en una base de datos, con el fin de evitar que la gente vote por una canción que el bar no tenga.

Las canciones más votadas serán aquellas que probablemente se escuchen. El bar, como siempre se reserva el derecho de hacerles caso a sus clientes.

¿Qué suena? (Inside)

Esta parte de la aplicación intentará imitar el comportamiento de la aplicación “Shazam” (www.shazam.com) donde el usuario puede saber qué música está escuchando actualmente. La diferencia entre la aplicación existente y la nuestra, es que tenemos el contacto directo con la base de datos que está sonando. Y dado que existen varios remixes de canciones, que los mismos DJs crean, Shazam no siempre acierta pero nosotros sí.

Además de esto proveeremos con información adicional de la canción al usuario, como las letras de la canción, lo popular que es la canción dentro del bar (y entre nuestro amigos), el género de la canción, un enlace al perfil del DJ si es un remix propio de él y más.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

¿Dónde están mis amigos? (Connect)

Aquí generaremos conexiones entre los usuarios y los bares. Existirán grupos de personas, grupos de locales y grupos mixtos. La idea es que un usuario pueda tener uno o varios grupos de amigos y a su vez un listado de locales favoritos. La idea de rutas entre locales, también ayudará al usuario a saber dónde ir después de que el local haya cerrado, o bien para planificar toda una noche.

Las rutas de bares podrán ser generadas por los usuarios o automáticamente considerando música similar o ambiente similar.

Además de todo esto, esta parte se centrará también en la de “cotilleo”. Donde un determinado usuario podrá saber dónde están sus amigos, en que plan han salidos y demás.

¿Qué quieren mis clientes? (Help)

Finalmente esta parte proporcionará una serie de herramientas al local para facilitar la comunicaciones de las otras tres con el bar o DJ. La tarea principal de esta aplicación será de notificar al dj las canciones más votadas, avisarle que se le está preguntando acerca de una canción, generar una interfaz cómoda para gestionar las listas y Check-In, realizar ofertas en tiempo real, etc...

En breve, proporcionar al bar de una solución informática de su local orientada al cliente.

Conclusión

Con nuestro proyecto estamos dando un paso hacia el futuro y haciendo simples las decisiones de uno de los aspectos más importantes de nuestras vidas, ¡Pasarlo bien!

- Outside: Elegimos
- Inside : Disfrutamos
- Connect: Planificamos
- Help : Con el apoyo e interés de los locales

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.3 Diferencias entre Where2Night y Facebook

Una de las diferencias más significativas, ya no en cuanto a funcionalidad, sino al tipo de software que estamos creando, es que nuestro proyecto se basa en software libre. Con esto queremos que cualquier persona tenga acceso a él, pueda modificarlo, e incluso realizar aportaciones, al cumplir las 4 libertades del software libre.

Al ser software libre, conseguimos que el producto esté a disposición de cualquiera. Si alguien quisiera utilizar todo o parte de nuestro trabajo, o incluso modificarlo para adaptarlo a sus necesidades, podría hacerlo sin problema.

Es también más visible, en el sentido de que los usuarios pueden ver qué hacemos con sus datos, cómo los gestionamos y hacia dónde van, cosa que no permite Facebook.

2.3.1 Diferencias basadas en usuarios (fiesteros):

- Facebook abarca un público que pertenece a un espacio demasiado extenso, con mucha variedad en cuanto a edad, gusto, trabajo, religión, política, etc. Es una mezcla de todo un poco, que a cierto punto puede resultar un tanto caótico. **Where2night** está enfocado a un público en concreto, acotando principalmente los tópicos de edad y gustos, puesto que **Where2night** sería utilizada en mayor parte por gente joven que le guste salir de fiesta.
- **Where2night** tiene la funcionalidad de seguir a un local. Con ello se permite a los usuarios ver el listado de sus canciones, cuál está sonando en ese momento y al mismo tiempo votar las canciones que deseen escuchar cuando se encuentren dentro de un local, lo que en la especificación se lo conoce como “gramola”.
- **Where2night** ofrece la posibilidad de apuntarte a las listas de los locales, sin la necesidad de conocer a algún relaciones públicas de la discoteca a la que se desea ir.
- **Where2night** también permite al usuario notificar a sus amigos que está ya en el local con el conocido “checkin” de la aplicación.

2.3.2 Diferencias basadas en locales:

Facebook no tiene la mayor parte de la funcionalidad que proporciona **Where2night** a los locales.

- Crear listas para que la gente se apunte directamente. Así, los “fiesteros” pueden recibir descuentos o incluso entrar gratis al local, con el propósito de lograr atraer al mayor número de clientes (lo que viene a ser el trabajo de un relaciones públicas).
- Poder ver las canciones más votadas por las personas que están en el local, y así tener una lista de sugerencias basadas en la preferencias de las canciones que los “fiesteros” desean escuchar.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Ver estadísticas de los “fiesteros” en cuanto a edades, género, estado civil, top 5 bebidas favoritas y top 5 música favorita, basándose en los perfiles de los mismos.

2.4 Especificación de requisitos software

2.4.1 Especificaciones Local

2.4.1.1 Información de la página Inicio

La información que el local verá desde su pantalla de inicio será:

- Editar perfil.
- Subir fotos.
- Crear eventos (foto o comentario).
- Seguidores.

2.4.1.2 Información registro

Datos obligatorios (solicitud)

- Nombre de Empresa
- Nombre del Local
- Dirección exacta (no solo Madrid)
- Teléfono de contacto
- Correo electrónico
- C.I.F

Una vez enviada la solicitud, Where2Night se pondrá en contacto para verificar su información y dar de alta el Local.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Se le enviará:

- Un link para asignar la contraseña y nombre de usuario

Datos de Perfil Interno del Local

- Nombre del Local
- Ubicación
- Estilo música preferida *
- Precio de Entrada *
- Precio bebidas *
- Horarios *
- Foto de perfil *
- Acerca de ti *

* Campos opcionales

2.4.1.3 Criterios de privacidad

Registro Inicial

- Correo de confirmación para la activación de la cuenta y asignación de contraseña.
- El local está de acuerdo con las políticas de protección de datos.

Gestión de contraseñas

- Cambio de contraseña.
- En caso de olvido se enviará al correo una url para restablecerla con una validez de 24 horas.

Visibilidad de los campos del perfil

- Estilo música preferida
- Precio de Entrada
- Precio bebidas
- Horarios

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Foto de perfil
- Acerca de ti

Pueden ser visibles o no. En caso de visibilidad, será a todo el mundo.

Si has rellenado un campo es visible, si no quieres que se vea no lo rellenes (locales solo).

NOTA:

Una vez se concreten los puntos de privacidad a tener en cuenta, será necesario redactar un documento con la política de privacidad de WHERE2NIGHT, la cual será mostrada al Local al principio de su registro de forma obligatoria.

2.4.2 Especificaciones Fiestero

2.4.2.1 Información de Inicio

- Local al que va a ir y botón Check-In
- Editar perfil
- Ánimo en el que vas a salir (Elegir sobre lista o personalizado)
- Novedades
 - Cambio de ánimo para salir de amigos
 - Eventos y promociones de locales favoritos
 - Ubicación de tus amigos. (Voy a <Local>, Estoy en <Local>)
 - Últimas fotos publicadas
 - Eventos de DJs favoritos

2.4.2.2 Información registro

Para usar tanto la app móvil como la página web será obligatorio registrarse y ser mayor de edad. Para ello el usuario dispondrá de dos alternativas: registro mediante su cuenta de Facebook, Google+ o sin ella.

Cuando terminas el registro la siguiente ventana es Editar Perfil.

Tras rellenar el perfil mostrará una un "Tour" de cómo funciona la App.

Cuando el registro no es por Facebook ni por Google+, se envía un link de confirmación y a menos que se active la cuenta ésta será limitada pasada 24 horas.

Datos obligatorios

- Nombre
- Apellidos

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

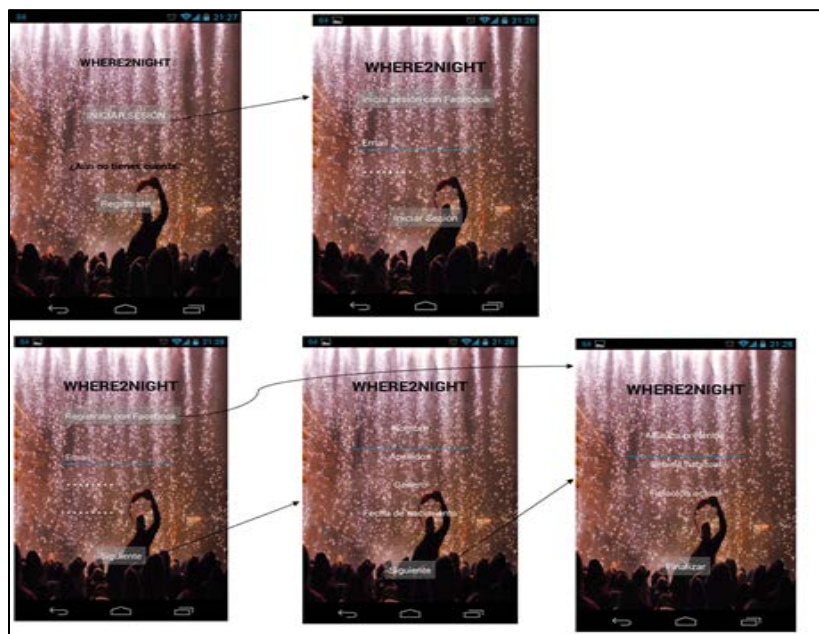
- Fecha de nacimiento
- Sexo
- Correo electrónico
- Contraseña

Datos de Perfil Interno de Fiestero

- Nombre
- Apellidos
- Fecha de nacimiento
- Sexo
- Correo electrónico
- Estilo música preferida *
- Tipo bebida preferida *
- Foto de perfil *
- Estado civil *
- Ciudad donde vives *
- Número de teléfono *
- Acerca de ti *

* Campos opcionales

Procedimiento registro



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.4.2.3 Criterios de privacidad

Registro Inicial

- Correo de confirmación para la activación de la cuenta.
- El usuario está de acuerdo con las políticas de protección de datos.

Gestión de contraseñas

- Cambio de contraseña.
- En caso de olvido se enviará al correo una url para restablecerla con una validez de 24 horas.

¿Quién puede ver las publicaciones que realice (Fotos, comentarios, cambio de estado)?

- Amigos.
- Solo yo.
- Todo el mundo.
- Personalizado (solo ciertos amigos).

¿Quién puede dejar comentarios en las publicaciones?

- Amigos.
- Solo yo.
- Todo el mundo.
- Personalizado (solo ciertos amigos).

¿Quién puede agregarme?

- Amigos de amigos.
- Todo el mundo.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Gestión de bloqueo de usuarios:

- Denuncias de comentarios o fotos.

Visibilidad de los siguientes campos del perfil del usuario

- Música favorita.
- Edad.
- Lugares favoritos.
- Sexo.
- Últimos visitados.
- Estado civil.
- Posible destino.
- Geolocalización.

Pueden ser visibles o no. En caso de visibilidad, podrá ser a todo el mundo o solo a los amigos.

NOTA:

Una vez se concreten los puntos de privacidad a tener en cuenta, será necesario redactar un documento con la política de privacidad de WHERE2NIGHT, la cual será mostrada al usuario al principio de su registro de forma obligatoria tanto en el formato WEB como en la APP.

2.4.3 Especificaciones DJ

2.4.3.1 Información Inicio

- Editar perfil.
- Subir fotos.
- Crear eventos (foto o comentario).
- Seguidores.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.4.3.2 Información registro

Datos obligatorios (Solicitud)

- Nombre
- Apellidos
- Nombre DJ
- Fecha de nacimiento
- Sexo
- Teléfono
- Correo electrónico

Una vez enviada la solicitud, Where2Night se pondrá en contacto para verificar su información y dar de alta el Dj.

Se le enviará:

- Un link para asignar la contraseña y nombre de usuario

Datos de Perfil Interno de Dj

- Nombre
- Apellidos
- Fecha de nacimiento
- Sexo *
- Correo electrónico
- Estilo música que pone*
- Foto de perfil *
- Acerca de ti *

* Campos opcionales

2.4.3.3 Criterios de privacidad

Registro Inicial

- Correo de confirmación para la activación de la cuenta y asignación de contraseña y nombre de usuario.
- El Dj está de acuerdo con las políticas de protección de datos.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Gestión de contraseñas

- Cambio de contraseña.
- En caso de olvido se enviará al correo una url para restablecerla con una validez de 24 horas.

Visibilidad de los campos del perfil

- Estilo música que pone
- Foto de perfil
- Acerca de ti

Pueden ser visibles o no. En caso de visibilidad, será a todo el mundo.

Si has rellenado un campo es visible, si no quieres que se vea no lo rellenes (locales solo).

NOTA:

Una vez se concreten los puntos de privacidad a tener en cuenta, será necesario redactar un documento con la política de privacidad de WHERE2NIGHT, la cual será mostrada al Dj al principio de su registro de forma obligatoria.

2.4.4 Búsquedas

Hay dos tipos de búsquedas: gente o local.

Para buscar gente hay varias opciones:

- Buscar por el nombre o parte de éste.
- Buscar en alrededores.
- Buscar en local específico.

Todas ellas tienen la opción de buscar sólo en amigos, cuando seleccionas a alguien te muestra su perfil (con más o menos cosas dependiendo si es amigo o no) y te da la opción de enviar un mensaje o agregar/quitar en favoritos si sois amigos; o de agregar a amigos si no lo sois.

Para buscar locales hay también varias opciones:

- Buscar por el nombre o parte de éste.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Buscar en alrededores.
- Buscar según donde vayan mis amigos/favoritos (indicar número de amigos/favoritos en cada local que muestre).
- Buscar por características de local (Ej: tipo de música, precio).
- Buscar según estadísticas (Ej: número de gente/chicas/chicos, indicar el número en cada local que muestre).
- Por eventos especiales. (Esto no sé si lo vamos a hacer, si dejamos que los locales añadan eventos se podría buscar así).

Todas ellas tienen la opción de buscar sólo en favoritos, cuando seleccionas un local te muestra los datos del local (número de amigos y estadísticas incluidos) y te da la opción de añadir/quitar en favoritos, indicar si vas a ir ahí o dar tu opinión sobre el local (esto no se si está hablado, pero se me acaba de ocurrir, así que lo pongo por si acaso).

En la aplicación móvil sólo se mostrarían 5 resultados por página, en la versión web se puede ampliar a 20 resultados por página.

Parámetros entrada

Fiestero busca fiestero:

- Nombre.
- Correo electrónico.

Fiestero busca local:

- Nombre del local.
- Locales cercanos a su posición.
- Tipo de música.

Fiestero busca DJ:

- Nombre.

Ordenación

- Popularidad.
- Precio de la entrada.
- Precio de la consumición.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Distancia.
- Aforo total.

2.4.5 Check-In

Formas de realizar check-in

- Check-In manual:
 - Geolocalización.
 - Si el local tiene Wifi, mediante escaneo de la Wifi.

Procedimiento para realizar el Check-In

Cuando el usuario esté en el local y pulse el botón de Check-In la app comprobará en la base de datos si ese local posee Wifi, en caso de que tenga, realizará un escaneo del SSID y si el usuario tiene la Wifi desactivada se realizará por geolocalización de la misma manera que si el local no tuviera Wifi.

2.4.6 Gramola

2.4.6.1 Solicitar canciones

Las personas podrán votar canciones para escuchar. Estas votaciones pueden ser por canciones específicas, por género o por canciones similares a otra canción.

Las canciones específicas deberán estar previamente proporcionadas por el bar en una base de datos, con el fin de evitar que la gente vote por una canción que el bar no tenga. Esta base de datos será visible tanto para el DJ a través de la web como para el usuario a través de la app móvil. Desde esta ventana se podrán filtrar las canciones mediante opciones de búsqueda, para poder realizar la votación.

Se podrá votar por un número limitado de canciones, de géneros o de canciones “similares a”. Por ejemplo, limitar los votos a 5 canciones, 2 géneros y 3 canciones similares a la votada que se quieran escuchar.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Los resultados y las estadísticas de las votaciones se podrán visualizar tanto en el perfil de la web dedicado al DJ como en la app dedicada al usuario.

2.4.6.2 ¿Qué suena?

Se mostrará, en un apartado de la pantalla de la app, la canción que está sonando actualmente. Para ello, el DJ deberá seleccionar la canción que está sonando manualmente, desde el perfil de la web dedicado al DJ.

2.4.6 Gestión de amistades

¿Cómo agregar amigos?

Dependiendo de cómo vayamos a iniciar sesión en nuestra aplicación, vamos a poder buscar nuestros amigos en la red social que escojamos en un principio.

Cuando estemos en la pantalla de añadir amigos, nos mostraría todos nuestros amigos que pertenezcan a dicha red social y tengan la aplicación instalada, dándonos la oportunidad de poder agregarlos a nuestra lista de amigos directamente. En el caso de que el amigo que queremos añadir no tenga la aplicación instalada, que también nos permita la posibilidad de enviar notificaciones a través de la red social, para que empiece a utilizarla si es de su agrado.

Perfil Externo de Amigos

A continuación listamos la información que vamos a almacenar de nuestros amigos siempre y cuando se tenga en cuenta las políticas de privacidad declaradas por el mismo:

- Nombre
- Apellidos
- Fecha de nacimiento
- Sexo
- Correo electrónico
- Estilo música preferida
- Tipo bebida preferida
- Relación actual
- Foto de perfil
- Estado civil

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Ubicación (Voy a <Local>, Estoy en <Local>)
- Ciudad donde vive
- Ánimo para salir
- Número de teléfono
- Acerca de él
- Lugares favoritos
- DJ favoritos
- Últimos visitados
- Amigos

Red social a utilizar

Para una mayor comodidad al momento de iniciar sesión se podrá realizar mediante una red social, en este caso usaríamos “Facebook” mediante el cuál podamos conseguir la mayor cantidad información de los usuarios.

2.4.7 Visión Webapp, Administración y DJ

Lo primero a tener en cuenta es saber si los perfiles de administración y dj son únicamente una Webapp como un software, sin tener en cuenta lo que es la red social, en tema visualización.

Un local y un Dj no deberían tener amigos:

- Los locales sólo clientes, de los que no debería tener información individual ni siquiera una lista que los muestre en plan amigos de la red social.
- El dj sólo debería tener seguidores, de los que sí podrá ver su información y a la vez que pueda dar permisos a estos para poder ver cierta información suya.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.5 Gestión de riesgos

Para la gestión de riesgos utilizaremos las ideas de Boehm, basadas en identificación, análisis, priorización y resolución de los riesgos.

Para ello hemos utilizado un formulario en el cual escribiremos la descripción, indicadores, consecuencias y planes de prevención y mitigación cuando sea posible.

Además usamos la siguiente tabla para medir el impacto en relación a la severidad y la probabilidad del riesgo

Probability Severity	Frequent	Probable	Occasional	Remote	Improbable
Catastrophic	IN	IN	IN	H	M
Critical	IN	IN	H	M	L
Serious	H	H	M	L	T
Minor	M	M	L	T	T
Negligible	M	L	T	T	T
LEGEND	T = Tolerable	L = Low	M = Medium	H = High	IN = Intolerable

Gráfico de Boehm

Control de riesgos

Titulo		Planificación o presupuesto irrealista	
		Descripción	Las estimaciones en tiempo o personal para cada tarea no son realistas.
ID	R1	Indicadores	Tardar más tiempo del previsto en cada tarea.
Severidad	Serio	Consecuencias	Retrasos en las entregas.
Probabilidad	Probable	Prevención	Realizar estimaciones más precisas.
Impacto	Alto	Mitigación	Recalcular las estimaciones y/o dedicar más tiempo o personas a esas tareas si fuera posible.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Titulo		Problemas con miembros del grupo	
		Descripción	Algún miembro del grupo no cumple con su parte.
ID	R2	Indicadores	No entrega su parte a tiempo.
Severidad	Critico	Consecuencias	Retrasos en las entregas.
Probabilidad	Ocasional	Prevención	Presionar más al miembro del grupo.
Impacto	Alto	Mitigación	Recalcular las estimaciones y/o dedicar más tiempo o personas a esas tareas si fuera posible.

Titulo		Problemas con el servidor	
		Descripción	El servidor sufre desconexiones y caídas.
ID	R3	Indicadores	Fallos en las conexiones.
Severidad	Serio	Consecuencias	Imposibilidad de realizar la tarea.
Probabilidad	Improbable	Prevención	
Impacto	Tolerable	Mitigación	Recalcular las estimaciones y/o dedicar más tiempo o personas a esas tareas si fuera posible.

Titulo		Problemas con software de desarrollo	
		Descripción	El software utilizado o el servidor dan problemas.
ID	R4	Indicadores	
Severidad	Critico	Consecuencias	Imposibilidad de realizar la tarea.
Probabilidad	Remoto	Prevención	Subidas continuas al repositorio para evitar la pérdida de datos.
Impacto	Medio	Mitigación	Recalcular las estimaciones y/o dedicar más tiempo o personas a esas tareas si fuera posible.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Título		Miembros del equipo toman decisiones erróneas	
		Descripción	Un miembro del equipo cambia algo o toma una decisión errónea por su cuenta.
ID	R5	Indicadores	Fallo general del sistema
Severidad	Critico	Consecuencias	La aplicación deja de funcionar
Probabilidad	Remoto	Prevención	Consultar ante la menor duda.
Impacto	Medio	Mitigación	Volver al punto anterior.

Título		Pérdida o desaparición de algún miembro del equipo	
		Descripción	Un miembro del equipo deja de hacer sus tareas, o no da señales de lo que está haciendo.
ID	R6	Indicadores	No hay comunicación con algún miembro del equipo.
Severidad	Serio - Catastrófico	Consecuencias	Dependiendo de la función de ese miembro puede ser solucionado fácilmente o producir un retraso muy grande.
Probabilidad	Remoto	Prevención	Comunicación continua.
Impacto	Bajo - Alto	Mitigación	Asignar la tarea a otros miembros.

Título		Elección de lenguaje o tecnología errónea	
		Descripción	Los lenguajes utilizados o las tecnologías no se adecuan a nuestras necesidades reales.
ID	R7	Indicadores	Complicación extrema para continuar el proyecto.
Severidad	Critico - Catastrófico	Consecuencias	Reinicio de esa rama de desarrollo.
Probabilidad	Remoto	Prevención	Investigación exhaustiva previa al inicio del desarrollo.
Impacto	Medio - Alto	Mitigación	Reutilización del código anterior todo lo posible y asignación de recursos extra a esa rama.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Titulo		Planificación incorrecta	
		Descripción	Los tiempos destinados a cada actividad en un sprint no se corresponden con la realidad.
ID	R8	Indicadores	Cada tarea lleva más tiempo del estimado.
Severidad	Serio	Consecuencias	Entregas con retraso.
Probabilidad	Ocasional - Probable	Prevención	Investigación previamente con otros proyectos anteriores tareas parecidas.
Impacto	Medio - Alto	Mitigación	Recalcular las estimaciones, recortar tareas y asignar más recursos si fuera posible.

Otros riesgos

En este apartado presentamos otros riesgos que se podían evitar completamente, de tal forma que los incluimos en este apartado y no en el anterior.

CREACIÓN DE TOKENS

Vamos a explicar cómo proteger nuestra aplicación PHP contra ataques de tipo CSRF. Este tipo de ataques hacen que el usuario realice acciones sobre una web de forma inadvertida. Por ejemplo, en una página vulnerable a este tipo de ataques, podrían construirse páginas de ataque en las que únicamente con la visita del usuario se modificase su información de registro, se publicase información de forma oculta, etc.

Imaginemos que estamos conectados a un web que requiere autenticación y que por ejemplo estamos manteniendo una conversación por chat con otra persona. Esta persona podría enviarnos un enlace a una página que contuviera una imagen oculta que apuntase a una url de la web en la que estamos autenticados. Cuando entrásemos en esa página, la imagen oculta haría que solicitásemos a la web una determinada acción sin ser conscientes de ello, por ejemplo cambiando nuestros datos de registro, publicando información en un foro a nuestro nombre, o cualquier otra cosa que se nos ocurra. El problema por tanto es que la web en la que estamos autenticados, es incapaz de saber si la petición que le llega la hemos realizado nosotros de forma voluntaria o si ha sido trampeada mediante algún ataque CSRF

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Una forma de evitar esta vulnerabilidad es utilizar tokens de autorización en cada acción que se realiza sobre el servidor. Básicamente lo que hace la aplicación es generar un token de autorización y enviarlo al usuario. Cuando se envía información a la aplicación hay que enviar también el token para que el servidor lo compare con el que había generado previamente. Vamos a ver a continuación dos formas diferentes de generar estos tokens.

Token personal por acción en sesión.

Esta forma de generación de tokens se basa en la creación de tokens específicos para cada tipo de acción realizable en la aplicación y el registro de estos tokens en la sesión del usuario. La siguiente función recibe como parámetro el nombre de un tipo de formulario o de acción realizable sobre la web. La función genera un token de forma aleatoria y guarda en sesión, asociado al nombre del formulario recibido, el token generado y la fecha de generación.

```
function generateFormToken($form) {
    // generar token de forma aleatoria
    $token = md5(uniqid(microtime(), true));

    // generar fecha de generación del token
    $token_time = time();

    // escribir la información del token en sesión para poder
    // comprobar su validez cuando se reciba un token desde un formulario
    $_SESSION['csrf'][$form.'_token'] = array('token'=>$token, 'time'=>$token_time);

    return $token;
}
```

Para incluir el token en un formulario habría que añadir un nuevo campo oculto a cada formulario que queramos securizar. En el siguiente ejemplo se ha añadido el campo “auth_token” a un formulario de envío de mensajes. En el campo “value” del campo se incluye el token obtenido indicando que se solicita un token para el formulario “send_message”.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

```
<form action="send_message.php" method="POST">
<input type="hidden" name="auth_token" value="<?=generateFormToken('send_message')? >"/>
<textarea name="message-text"></textarea>
<input type="submit" />
</form>
```

La siguiente función permite comprobar la validez de los tickets generados. La función recibe como parámetros el nombre del formulario, el token recibido y un parámetro opcional con el tiempo de validez del ticket. Se comprueba si hay registrado en sesión un token de autorización para el formulario indicado y, si es así, se compara este token con el recibido. En caso de que sean iguales se tiene en cuenta si se indica un tiempo de validez de ticket. Este tiempo permite establecer un control complementario de forma que damos un tiempo de vida determinado a cada ticket enviado al usuario.

```
function verifyFormToken($form, $token, $delta_time=0) {

    // comprueba si hay un token registrado en sesión para el formulario
    if(!isset($_SESSION['csrf'][$form.'_token'])) {
        return false;
    }

    // compara el token recibido con el registrado en sesión
    if ($_SESSION['csrf'][$form.'_token']['token'] !== $token) {
        return false;
    }

    // si se indica un tiempo máximo de validez del ticket se compara la
    // fecha actual con la de generación del ticket
    if($delta_time > 0){
        $token_age = time() - $_SESSION['csrf'][$form.'_token']['time'];
        if($token_age >= $delta_time){
            return false;
        }
    }

    return true;
}
```

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Para comprobar la validez del ticket cuando se recibe un formulario de envío de mensajes por POST, podemos utilizar el siguiente código, en el que se llama a la función anterior indicando que se quiere comprobar la validez del token recibido para el formulario "send_message" con un tiempo máximo de vida de 300 segundos (5 minutos).

```
$ticket = $_POST['auth_token'];
$valid = verifyFormToken('send_message', $token, 300);
if(!$valid){
    echo "El ticket recibido no es válido";
}
```

Con esta forma de generar los token aumentamos la seguridad de cada formulario de la aplicación por separado. En el caso de que alguno de los tickets generados cayera en poder de un atacante únicamente podría utilizarlo en un formulario y durante un tiempo de vida limitado. En el lado negativo hay que tener en cuenta el espacio requerido en el servidor para almacenar cada token en las sesiones de usuario y algunos problemas de usabilidad. Estos problemas pueden producirse por ejemplo si el usuario utiliza varias pestañas para navegar por la web. En este caso, podría darse el caso de tener dos pestañas abiertas con el formulario de enviar mensaje, pero cada una de ellas con un ticket diferente. Al enviar un mensaje únicamente el último ticket generado sería válido. Una forma de reducir este problema sería comprobando también la fecha de validez del ticket en la generación y no regenerándolo en cada petición de ticket.

Token personal por acción sin sesión

Esta forma de generar los tickets se basan en la utilización de una palabra secreta y el identificador de sesión del usuario. De esta forma se evita la necesidad de tener que almacenar los tokens generados en sesión. A continuación se muestran dos funciones que pueden utilizarse en lugar de las descritas en la sección anterior. Ambas funciones utilizan un valor de configuración prefijado que contiene una palabra secreta a partir de la cual se generan todos los tickets. Esto permite invalidar en cualquier momento todos los tickets modificando simplemente esta palabra.

```
$CONF['csrf_secret'] = 'dfa%d_FA{]2Ñf523scvDAgfasg';

public function generateFormToken($form) {
    $secret = $GLOBALS['CONF']['csrf_secret'];
    $sid = session_id();
    $token = md5($secret.$sid.$form);
    return $token;
}
```


where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

```

public function verifyFormToken($form, $token) {
    $secret      =      $GLOBALS['CONF']['csrf_secret'];
    $sid          =      session_id();
    $correct      =      md5($secret.$sid.$form);
    return        ($token      ==      $correct);
}

```

La función de generación recibe el nombre del formulario a securizar y obtiene el token asociado haciendo un hash md5 sobre la concatenación de la palabra secreta, el identificador de sesión del usuario y el nombre del formulario. De esta forma se obtiene un token de autorización para cada usuario y cada acción. Para comprobar su validez se realiza la misma acción y se compara el token generado con el token recibido.

A diferencia del caso en el que registrábamos los token en sesión utilizamos menos espacio en el servidor y tenemos menos problemas de usabilidad, pero sin embargo no tenemos control sobre el tiempo de validez del ticket, estos caducan en el momento en el que finaliza la sesión de usuario. El siguiente fragmento de código permite comprobar la validez de un token recibido.

```

$ticket = $_POST['auth_token'];
$valid = verifyFormToken('send_message', $token);
if(!$valid){
    echo "El ticket recibido no es válido";
}

?>

```

Utilización de TOKENs en el proyecto

Creemos un Token cuando hacemos un login, ya sea mediante web o mediante móvil.

```

<?php
$sToken = md5(uniqid(mt_rand(), true));
    if ($mobile == 1){
        $query = "SELECT insertTokenMobile('{ $email}', '{ $sToken} )'";
    }
    else{
        $query = "SELECT insertToken('{ $email}', '{ $sToken} )'";
    }

?>

```

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Para cada acción comparamos el token con el de la base de datos. Ya sea peticiones mediante \$_POST, \$_GET, \$_DELETE, el idProfile y el Token siempre serán recibidos de la url. Lo utilizamos de la siguiente forma:

```
<?php
$request = explode("/", substr(@$_SERVER['PATH_INFO'], 1));
    $idProfile = $request[0];
    $token = $request[1];

    $tokenT= _tokenOK($idProfile,$token);

    if (isset($_SERVER['REQUEST_METHOD']) && $tokenT == 1){
        $method = $_SERVER['REQUEST_METHOD'];
        switch ($method) {
            case 'GET':
                break;
            case 'POST':
                break;
            case 'DELETE':
                break;
            default:
                break;
        }
    }
?>
?>
```

ATAQUES DE INYECCIÓN SQL

La mayoría de las aplicaciones web desarrolladas hoy en día hacen uso de una base de datos para ofrecer páginas dinámicas y almacenar información tanto de los usuarios como de la propia herramienta, datos a los que se accede por medio del lenguaje SQL, un lenguaje para interactuar con bases de datos relacionales.

El uso de este tipo de lenguaje ha traído consigo la aparición de una de las vulnerabilidades más peligrosas a la que nos podemos enfrentar. Nos estamos refiriendo al ataque por inyección de código SQL, una vulnerabilidad que no sólo pone en riesgo la integridad de la aplicación, sino de todos los datos almacenados de los usuarios que la utilicen, y que se produce cuando no se filtra de forma correcta la información enviada por los usuarios.

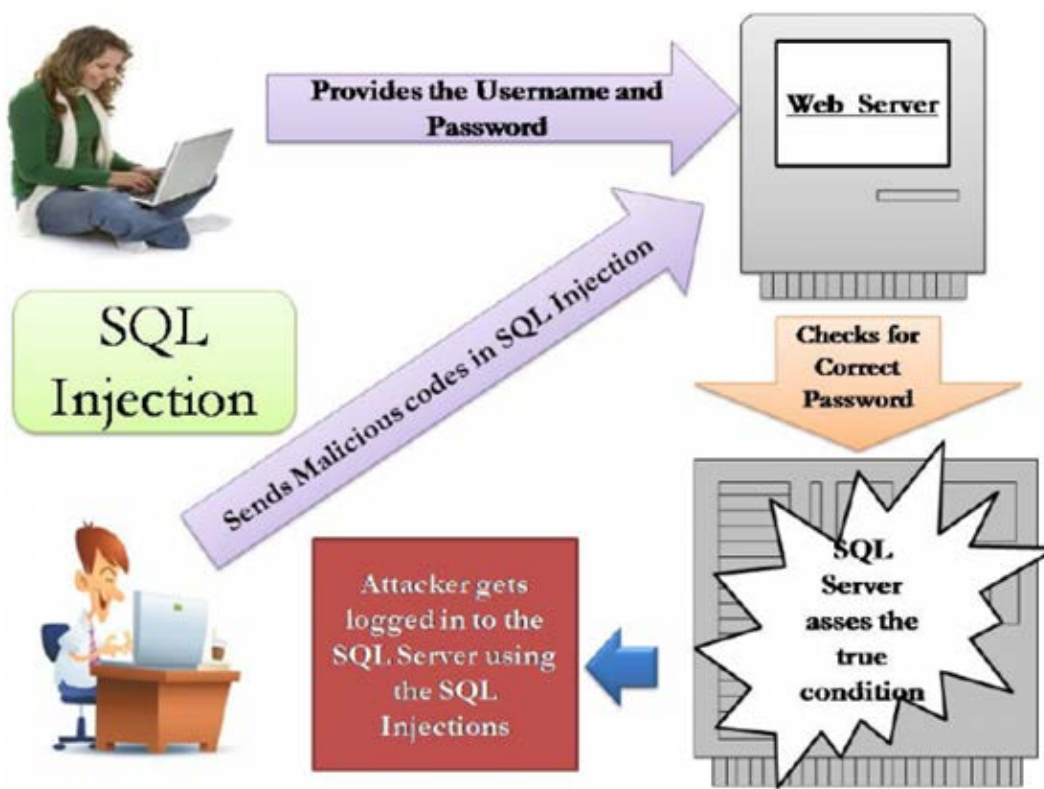
where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

¿Qué es el ataque por inyección SQL?

Consiste en la inserción de código SQL por medio de los datos de entrada desde la parte del cliente hacia la aplicación. Es decir, por medio de la inserción de este código el atacante puede modificar las consultas originales que debe realizar la aplicación y ejecutar otras totalmente distintas con la intención de acceder a la herramienta, obtener información de alguna de las tablas o borrar los datos almacenados, entre otras muchas cosas.

Como consecuencias de estos ataques y dependiendo de los privilegios que tenga el usuario de la base de datos bajo el que se ejecutan las consultas, se podría acceder no sólo a las tablas relacionadas con la aplicación, sino también a otras tablas pertenecientes a otras bases de datos alojadas en ese mismo servidor.

Lo comentado anteriormente es posible gracias a que el uso de ciertos caracteres en los campos de entrada de información por parte del usuario, ya sea mediante el uso de los campos de los formularios que son enviados al servidor mediante POST o bien por medio de los datos enviados mediante GET en las urls de las páginas web, posibilitan coordinar varias consultas SQL o ignorar el resto de la consulta, permitiendo al hacker ejecutar la consulta que elija, de ahí que sea necesario realizar un filtrado de esos datos enviados para evitar problemas.



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Entre las bases de datos susceptibles a este tipo de ataques nos encontramos MySQL, Oracle, Postgres o MS SQL.

Principales problemas que causan los ataques SQL Injection

- **Confidencialidad.** De forma habitual, las bases de datos almacenan información sensible, por lo que la pérdida de confiabilidad es un problema muy frecuente en aquellos sitios que son vulnerables a este tipo de ataques.
- **Autenticación.** Si el sistema de logueo que se utiliza para acceder a una zona restringida de una web es débil, por medio de este tipo de ataques se podría acceder sin la necesidad de conocer ni el usuario ni la contraseña.
- **Integridad.** Al igual que un ataque por inyección SQL permite leer información relevante almacenada en la base de datos, también es posible realizar cambios o incluso borrar toda información mediante este tipo de vulnerabilidad.

¿QUE HEMOS UTILIZADO EN EL PROYECTO PARA EVITAR ESTOS ATAQUES?

```
<?php
// Protege variables tipo string contra posible ataques de inyección SQL
function _protect_data ($string)
{
    if (!get_magic_quotes_gpc())
        $secure = addslashes($string);
    else
        $secure = $string;

    return $secure;
}

function _protect_data_type_int ($object)
{
    if (!isset ($object) || !is_numeric ($object))
        die ("<br/><br/>Value not expected. Possible attempt to hack!");
}

// Protege arrays contra posible ataques de inyección SQL
function _protect_data_array ($array)
{

```

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

```

        if (isset ($array) && is_array ($array))
        {
            if (!get_magic_quotes_gpc())
                foreach ($array as $key => $value)
                    $secure[addslashes ($key)] = addslashes ($value);
            else
                // Revisar la protección de las claves en este caso
                $secure = $array;

            return $secure;
        }
    }

function _protect_data_array_type_int ($array)
{
    if (isset ($array) && is_array ($array))
        foreach ($array as $key => $value)
            if (!is_numeric ($key) || !is_numeric ($value))
                die ("Value not expected. Possible attempt to hack!");
}
?>

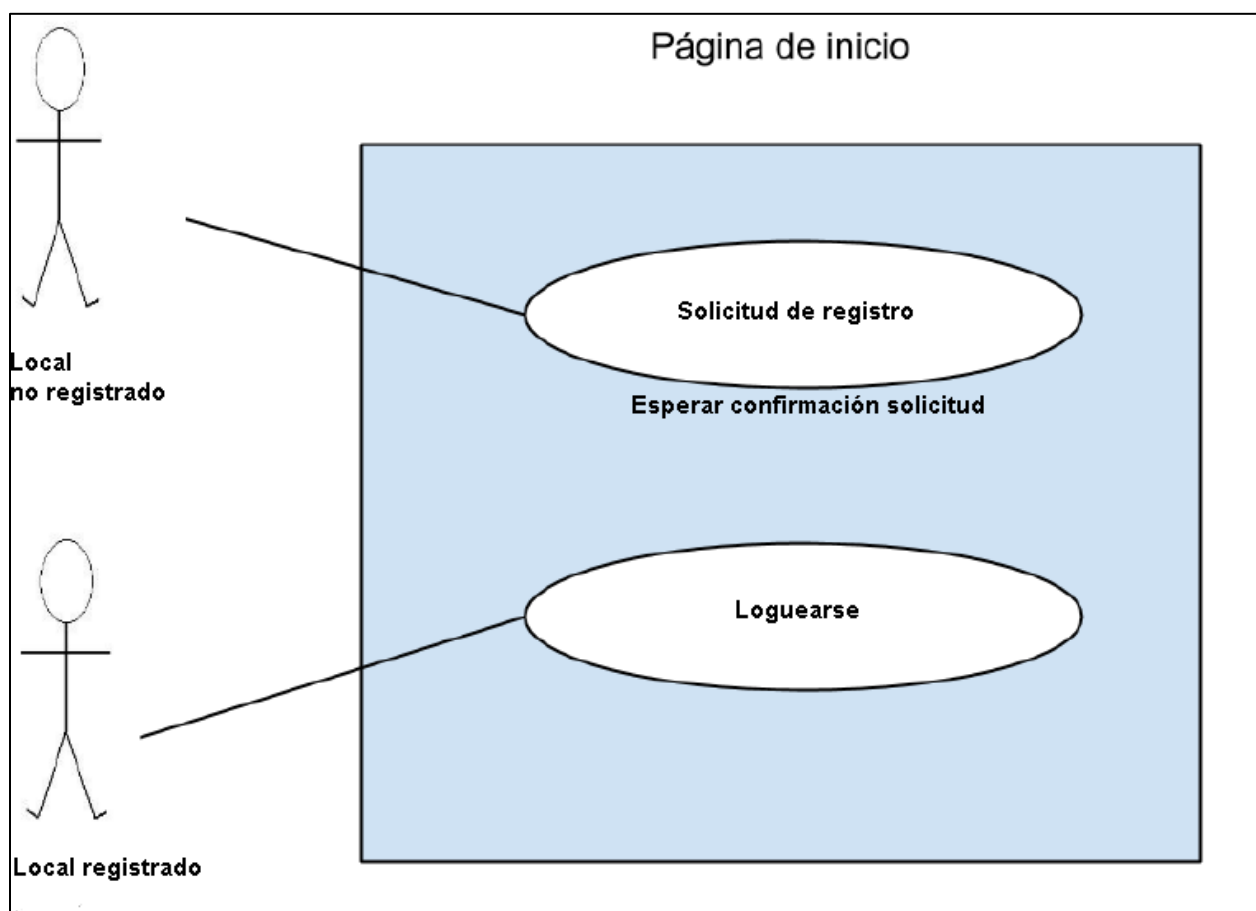
```

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6 Casos de uso

En este punto exponemos los casos de uso que hemos realizado hasta la fecha:

2.6.1 Registro de local



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

CU-01	Registro de local		
Versión	1.0 (26/11/2013)		
Dependencias	Ninguna		
Precondición	El usuario ha pulsado alguno de los enlaces		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el local desee registrarse en nuestra base de datos.		
Secuencia normal	Paso	Acción	
	1	Aparece una pantalla con los campos obligatorios a rellenar por un local: Nombre de la Empresa, Nombre del local, Dirección, Teléfono, Correo electrónico y C.I.F	
	2	El usuario inserta dichos datos	
	3	El sistema enviará, vía e-mail, un link para asignar una contraseña y un nombre de usuario,	
Postcondición	El local queda registrado y los campos quedan		
Excepciones	Paso	Acción	
	2	Si el usuario inserta una dirección de correo que no puede ser válida (teniendo en cuenta cómo está escrita)	
		E.1	El sistema informa de que la dirección no es válida y solicita otra vez los datos.
		E.2	El usuario introduce otra vez los datos.
	E.3	Si el correo es válido, se va al paso 3	
Comentarios	El registro sólo solicita datos obligatorios. Sin embargo, hay más información de local que podrá ser introducida una vez el local haya sido registrado		

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.2 Gramola sin Check-In

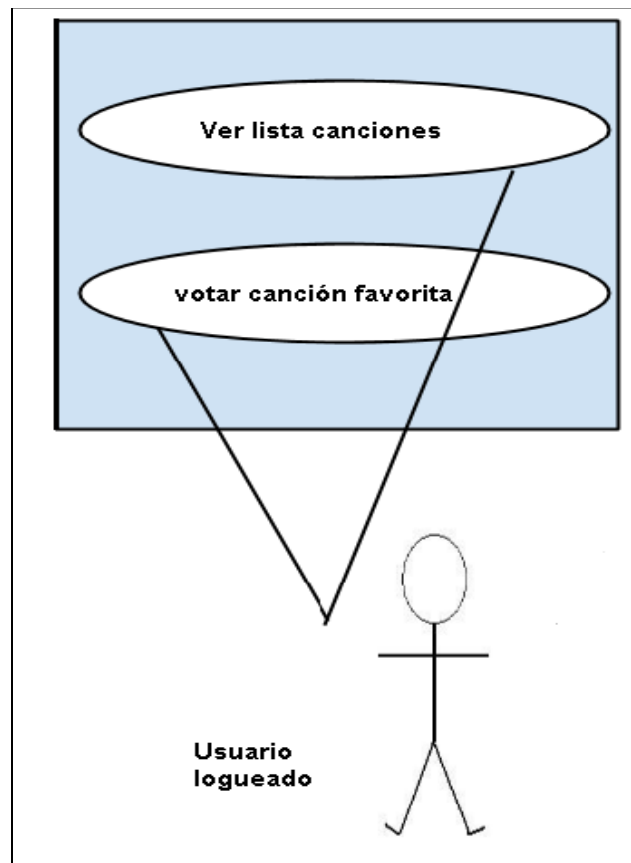
CU-02	Votar Cancion Fiestero (Sin check-in)		
Versión	1.0 (17/12/2013)		
Dependencias	Ninguna		
Precondición	El usuario ha pulsado alguno de los enlaces cuya opción es "Música local"		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee ver la música del local.		
Secuencia normal	Paso	Acción	
	1	Aparece una pantalla con la música que esta sonando en el local y una lista de música de las 10 canciones mas votadas y un boton de votar.	
	2	El usuario sólo puede ver estos datos.	
	3	El usuario pulsa votar.	
Postcondición	El usuario solo puede visualizar los datos.		
Excepciones	Paso	Acción	
	3	Si el usuario pulsa votar sin hacer check-in	
		E.1	El sistema informa de que necesita hacer check-in para poder votar
		E.2	El usuario pulsa aceptar
		E.3	Volvemos al paso 1
Comentario	El usuario sólo puede ver lo que esta sonando y las 10 canciones mas votadas porque no ha hecho check-in		

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.3 Gramola con Check-In

CU-03	Votar Canción Fiestero(Con Check-In)		
Versión	1.0 (17/12/2013)		
Dependencias	Haber hecho Check-in en el local que estamos		
Precondición	El usuario ha hecho check-in en algún local		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee votar canciones.		
Secuencia normal	Paso	Acción	
	1	Aparece una pantalla con la música que está sonando en el local y una lista de música de las 10 canciones más votadas. Y un botón de Votar	
	2	El usuario pulsa votar	
	3	Aparecerá una pantalla con las 10 canciones más votadas y abajo unos botones donde permita buscar canción.	
Postcondición	El usuario vota a las canciones y estas suman un voto.		
Excepciones	Paso	Acción	
	2	Si el usuario vota más de 5 canciones cada día.	
		E.1	El sistema informa de que ha sobrepasado los votos diarios
		E.2	El usuario pulsa aceptar
		E.3	Volvemos al paso 1

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.4 Seleccionar canción “Qué está sonando Dj/Local”

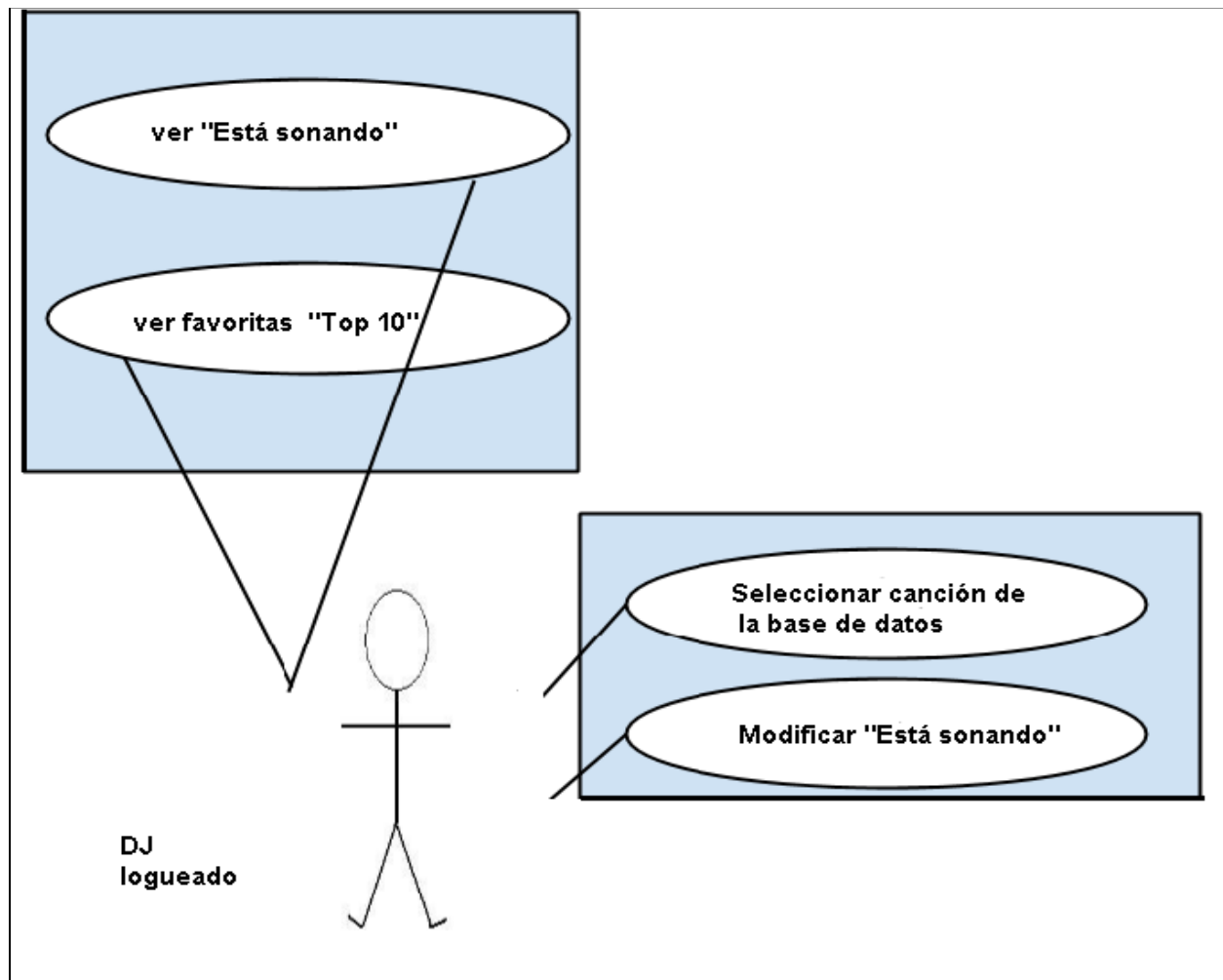
CU-04	seleccionar cancion "Qué está sonando Dj / Local"		
Versión	1.0 (17/12/2013)		
Dependencias	Ninguna		
Precondición	El usuario ha pulsado Gramola.		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el Dj local maneja la Gramola.		
Secuencia normal	Paso	Acción	
	1	Aparece una pantalla dividida en dos. "Está sonando" y "10 canciones más votadas".	
	2	En la parte esta sonando hay un boton "seleccionar cancion de la bbdd"	
	3	el usuario pulsa el boton "seleccionar cancion de la bbdd"	
	4	El sistema abre un pop-up con las canciones de la bbdd generada y un buscador arriba	
	5	El usuario busca la canción, la selecciona y pulsa "ok" y esta se muestra en "Está sonando"	
Postcondición	El usuario selecciona la canción que está sonando.		
Excepciones	Paso	Acción	
	3	Si el usuario no tiene una bbdd generada	
		E.1	El sistema informa de que no existe bbdd.
		E.2	El usuario pulsa aceptar
		E.3	Volvemos al paso 1

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.5 Seleccionar canción “Qué está sonando (10 + v) Dj/Local”

CU-05	seleccionar cancion "Qué está sonando(10 + v) Dj / Local"	
Versión	1.0 (17/12/2013)	
Dependencias	Hay canciones más votadas. Sino orden alfabético	
Precondición	El usuario ha pulsado Gramola.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el Dj local maneja la Gramola.	
Secuencia normal	Paso	Acción
	1	Aparece una pantalla dividida en dos. "Está sonando" y "10 canciones más votadas".
	2	En la parte "10 canciones más votadas" selecciona una cancion.
	3	se habilitan opciones. (borrar de la lista, añadir a está sonando)
	4	El usuario pulsa añadir a "Está sonando"
	5	Esta se muestra en "Está sonando"
Postcondición	El usuario selecciona la canción que está sonando.	

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.6 Check-in (manual)

CU-06	Check-in (manual)		
Versión	1.0 (21/1/2014)		
Dependencias	Encontrarse en el lugar que se quiere hacer el check-in		
Precondición	El usuario hace check-in en local		
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el Usuario hace check-in manualmente.		
Secuencia normal	Paso	Acción	
	1	El usuario busca el local dónde se encuentra. Accede al perfil de este, donde pulsa sobre el boton CHECK-IN	
	2	Mediante geolocalizacion, direccion MAC etc... comprobamos que se encuentra en el local seleccionado	
	3	El sistema enviará, vía e-mail, un link para asignar una contraseña y un nombre de usuario,	
	4	Si se encuentra dicho local aparece un mensaje de conectado con un boton de compartir (facebook, twitter etc...)	
	5	El usuario selecciona o no compartir y finaliza el proceso de check-in	
Postcondición	El usuario hace check-in del local en el que se encuentra.		
Excepciones	Paso	Acción	
	3	Si no se reconoce el local dentro de la lista de locales	
		E.1	El sistema informa de que no existe el local dentro de la bbdd de locales.
		E.2	El usuario pulsa aceptar
		E.3	Volvemos al paso 1

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.7 Añadir canción por un Local

CU-07	Añadir canción Local	
Versión	1.0 (3/4/2014)	
Dependencias	Haber sido logueado como Local	
Precondición	El usuario quiere añadir canción, ha pulsado Canciones.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee añadir canciones.	
Secuencia normal	Paso	Acción
	1	Aparece una pantalla con dos campos Nombre de Cancion y Artista canción.
	2	El usuario rellena estos campos
	3	Pulsa añadir canción y esta se añade.
Postcondición	El usuario ha añadido cancion que puede ver en su lista.	
Excepciones	Paso	Acción
		Si el usuario introduce los nombres repetidos, se detecta que está repetido.
	E.1	El sistema informa de que la canción ya existe.
	E.2	El usuario pulsa aceptar.
	3	E.3 Volvemos al paso 1

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.6.8 Enviar mensajes entre usuarios

CU-08	Enviar Mensaje entre usuarios	
Versión	1.0 (3/4/2014)	
Dependencias	Haber sido logueado como Partier	
Precondición	El usuario quiere enviar un mensaje a un amigo de W2N, para ello pulsa en Mensajes	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee enviar mensajes.	
Secuencia normal	Paso	Acción
	1	Aparece una pantalla con todos los mensajes de la bandeja y arriba a la derecha un boton que pone mensaje nuevo.
	2	El usuario pulsa el botón mensaje nuevo.
	3	Se abre un pop-up con los siguientes campos: Selección de contacto(Para) y el mensaje. y en la parte de abajo un botón "Enviar".
	4	El usuario pulsa enviar.
Postcondición	El usuario ha añadido cancion que puede ver en su lista.	
Excepciones	Paso	Acción
		Si el usuario envía un mensaje a un usuario que tiene la bandeja llena, salta un aviso.
	E.1	El sistema informa de que el usuario X tiene la bandeja de entrada llena.
	E.2	El usuario pulsa aceptar.
	4 E.3	Volvemos al paso 1

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

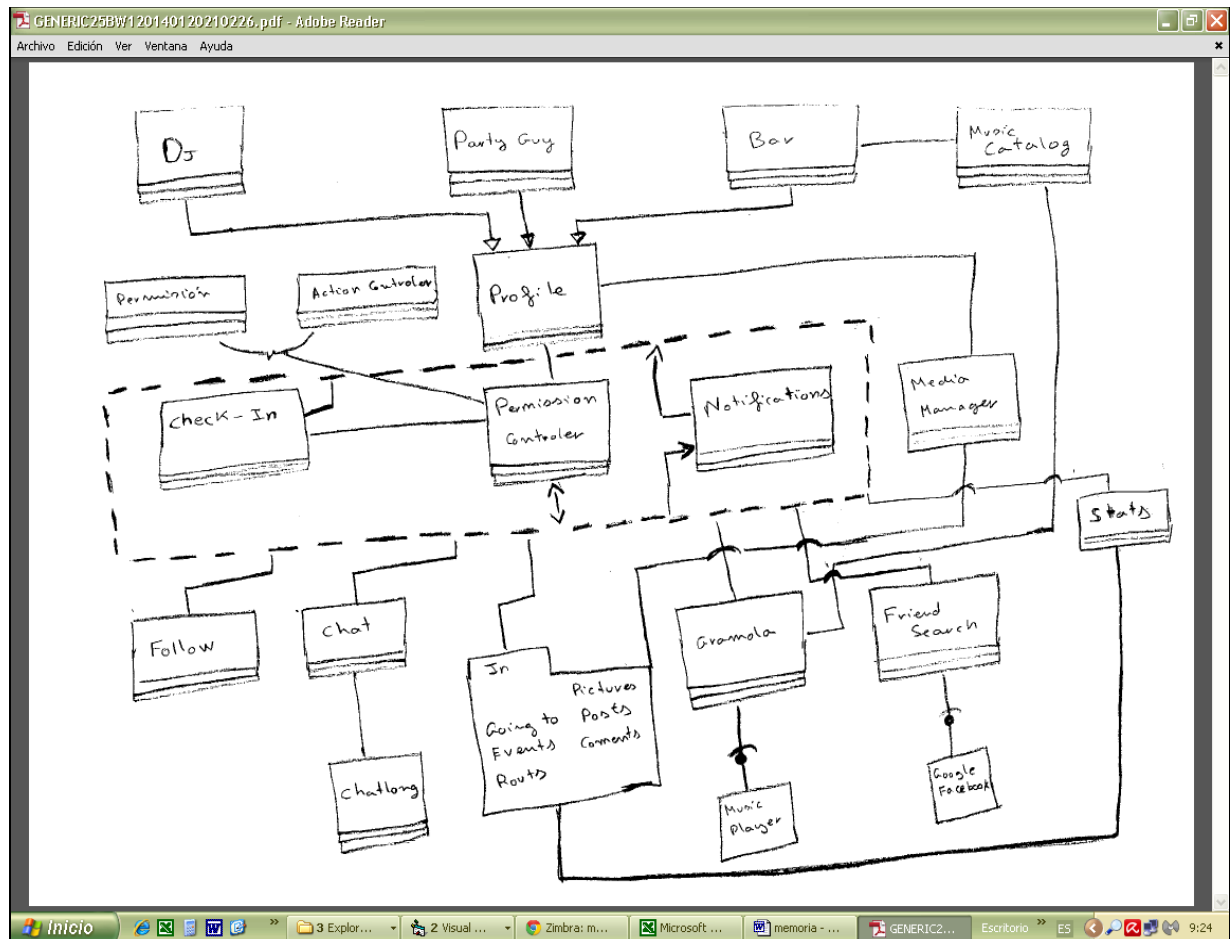
2.6.9 Enviar mensajes entre usuarios

CU-09	Enviar petición de amistad entre usuarios	
Versión	1.0 (3/4/2014)	
Dependencias	Haber sido logueado como Partier	
Precondición	El usuario quiere enviar una petición de amistad a otro usuario de de W2N, para ello pulsa en Amigos	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando el usuario desee enviar mensajes.	
Secuencia normal	Paso	Acción
	1	Aparece una pantalla con todos los amigos del propio partier en la parte superior, y en la parte inferior se muestra una lista de todos los demás usuarios de W2N que no son amigos del Partier en cuestión. El actor pulsa sobre uno de estos usuarios.
	2	Aparece una pantalla con el perfil del usuario que el actor a pulsado. Justo debajo de la foto de perfil aparecerá un botón "añadir fiestero"
	3	El boton pasa a tener el siguiente nombre "solicitud de amistad enviada"
	4	En la pantalla de Amigos, al final del todo podemos ver las solicitudes enviadas que estan pendientes de confirmación.
Postcondición	El usuario ha enviado una peticion de amistad, que puede ver en Amigos.	
Excepciones	Paso	Acción
		Si el usuario cancela esta petición de amistad.
	E.1.1	El sistema borra de la base de datos cualquier fila con los datos de petición y recibimiento.
	4	E.1.2 Volvemos al paso 1
	5	Si el usuario bloquea la petición de amistad.
	E.2.1	El sistema inserta en la base de datos un tipo especial por lo que al ser detectado este usuario no aparecerá nunca más como posible amigo del actor y viceversa.
	E.2.2	Volvemos al paso 1, en este caso sin poder volver a ver a este miembro en la lista.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.7 Diagramas

2.7.1 Diagrama de clases



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.8.1 WEB

2.8.1.1 Editar perfil

Registro Fiestero

← → ↻ ⬆ http://lucidchart.com

Where2Night 🔍 Búsquedas Fiestero Editar Perfil

Privacidad Cerrar Sesión

Mi perfil
Eventos
Fotos
Amigos
Locales
Dj's

Nombre

Apellidos

Fecha de nacimiento Día Mes Año

Sexo ☒ Hombre ☐ Mujer

Música favorita

Bebida favorita

Estado civil

Ciudad actual

Acerca de tí

Guardar

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.8.1.3 Perfil fiestero

←

→

↺

⬆

Registro Fiestero

http://lucidchart.com

Where2Night

🔍

Búsquedas

Fiestero

Mi perfil

Eventos

Fotos

Amigos

Locales

Dj's

Nombre del fiestero

Acerca de mí/cosas perfil

Ánimo

▼

Estoy en/ voy a

Pacha!!!!

Me apunto

{

Lugares recientes último mes

}

▲

▼

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.8.1.4 Registro fiestero

←

→

↺

⬆

http://lucidchart.com

Registro Fiestero

Where2Night

Iniciar Sesión

Nombre

Apellidos

Fecha de nacimiento

Día

▼

Mes

▼

Año

▼

Sexo

☒ Hombre
☐ Mujer

Correo electrónico

Confirmación correo electrónico

Contraseña

Confirmación contraseña

☐

Acepto los términos y condiciones

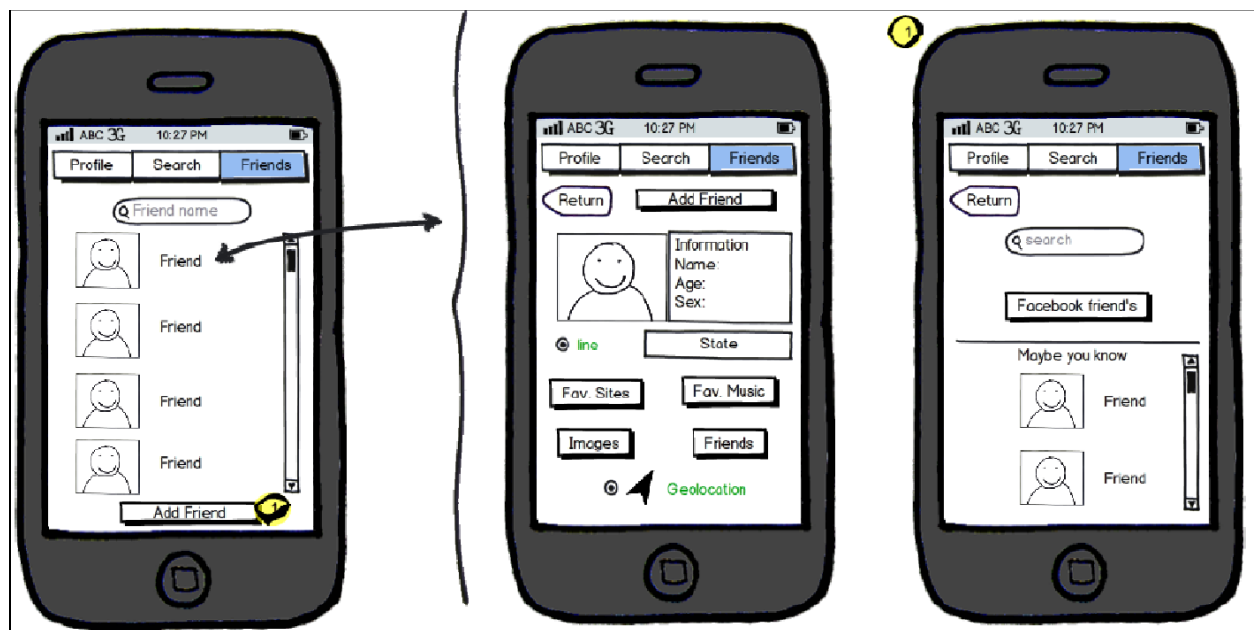
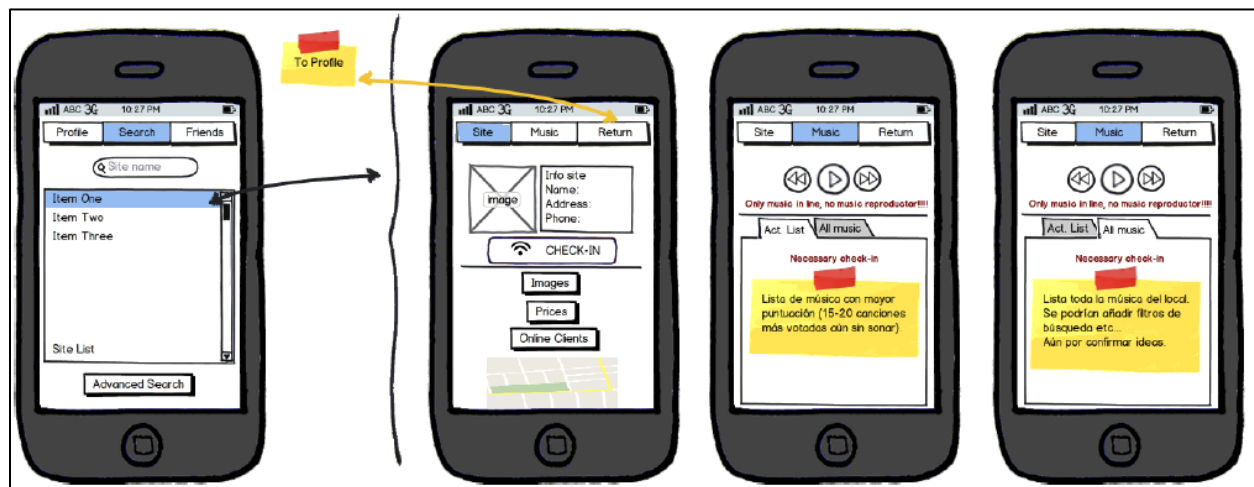
¡Regístrate!

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

2.8.2 Android



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

3. Planificación

3.1 EDT y diagrama de Gantt

La descomposición de tareas, así como los encargados de realizar cada una de ellas y las fechas en las que se realizaron hasta el último sprint aparecen en la siguiente tabla EDT (estructura de descomposición de trabajo). **Ver Anexo 1**

Con la ayuda de la EDT anterior se ha llevado a cabo la planificación del primer cuatrimestre y la del segundo cuatrimestre. Llegando al diagrama de Gantt que se encuentra en los **Anexos 2 y 3**

3.2 Actividades estructurales (Métrica de Persona/Horas)

La división de tareas, así como la estimación de horas que cada una de estas requiere nos ayudan a calcular horas por persona para cada parte del proyecto (Android, Web y Back-End).

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

3.2.1 Android

Android				
Tareas a realizar Desarrollo	Prioridad	Horas/Persona	Tareas a realizar pruebas	Horas/Persona
Login con Facebook	1	10	Login	10
Login con Google Plus	1	20	Registro	2
Login propietario	1	5	Navegación	0
Registro Propietario	1	5	Editar Perfil	2
Navegación por la App	1	5	Inicio	2
Editar Perfil	1	5	Listado Eventos con Búsqueda	1
Inicio	1	10	Listado Amigos con Búsqueda y peticiones de amistad y amigos FB	5
Listado Eventos con Búsqueda	2	10	Listado de Locales con Búsqueda y mapa	5
Listado Amigos con Búsqueda y peticiones de amistad y amigos FB	1	25	Listado de Djs con Búsqueda	1
Listado de Locales con Búsqueda y mapa	1	20	Vista detallada de Local - excepto gramola	10
Listado de Djs con Búsqueda	3	10	Vista detallada del perfil de otro usuario	5
Vista detallada de Local - excepto gramola	1	50	Vista detallada del Dj	1
Vista detallada del perfil de otro usuario	1	35	Galería de imágenes	1
Vista detallada del Dj	3	20	Vista detallada de una imagen	1
Galería de imágenes	5	10	check-in	15
Vista detallada de una imagen	5	5	gramola	15
Implementación del check-in	1	35	Ajustes de privacidad	15
Implementación de la gramola	3	100		
Ajustes de privacidad	1	35		
Total Horas Desarrollo:		415	Total Horas Pruebas:	91

Total Horas: 506

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

3.2.2 Web

Web				
Tareas a realizar Desarrollo	Prioridad	Horas/Persona	Tareas a realizar pruebas	Horas/Persona
Diseño de todas las páginas	2	40		
Iniciar sesión con Facebook (Fiestero, Local y DJ)	1	20	Login	20
DJ)	1	20	Registro	15
Iniciar sesión propio de la app	1	10	Editar Perfil	15
Registro (Fiestero, Local y DJ)	1	20	Gestionar eventos	12
Editar Perfil (Fiestero, Local y DJ)	1	20	Gestionar amigos	12
Home (Fiestero, Local y DJ)	1	10	Gestionar local	10
Gestionar eventos	2	15	Gestionar DJ	10
Búsqueda de amigos	1	10	Gramola DJ	16
Búsqueda Local	1	25	Gestionar listas	18
Agregar amigos	1	10	Gestionar imágenes	15
Seguir Local	3	10	Gestionar favoritos	10
Seguir DJ	3	10		
Gramola DJ	3	50		
Gestionar listas	2	40		
Gestionar imágenes	3	30		
Favorito DJ	4	10		
Favorito Local	4	10	Ajustes de privacidad	20
Ajustes de privacidad	1	40		
Total Horas Desarrollo:		400	Total Horas Pruebas:	173

Total Horas:	573
---------------------	------------

3.2.3 Back-End

Back-end			
Tareas a realizar Desarrollo	Prioridad	Horas/Persona	horas prueba
Creacion adaptacion del html, js y css por las 16 tareas de web (por cada tarea)	1	2	32
instalacion del entorno en el servidor	1	10	
curso intensivo del social stream y rubyOnRails	1	8	
creacion de peticiones JSON y test de unidad (por cada una de las tareas del equipo android)	1	4	60
creacion de distintos perfiles/roles (dj, fiestero, bar)	2	16	
crear base de datos de la musica de cada local	1	8	
interfaz del "escuchando ahora" (de la gramola)	3	16	
conexion con facebook/google+	2	8	
busqueda de amigos en otras redes sociales	3	16	
gestion de permisos segun estado dentro o fuera de checkin	3	16	
test de unidad de "escuchando ahora"	2	4	
gestion de bares en google maps	3	8	
gestion de ramas en github a la hora de hacer deploy (subir la version del sprint)	1	4	
Total Horas Desarrollo:		120	92

Total Horas:	212
---------------------	------------

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

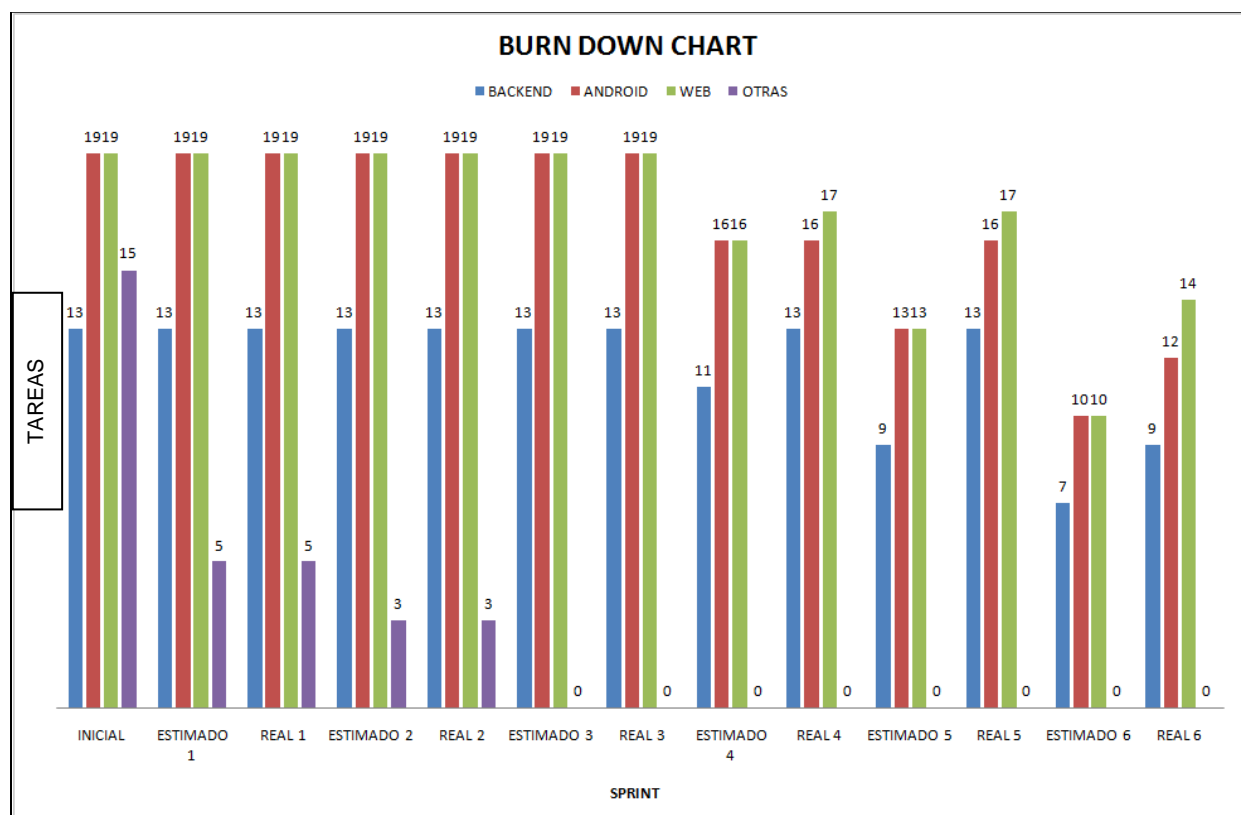
3.3 Burn Down Chart

Los tres primeros sprints los dedicamos a realizar tareas como la visión, guía de estilos... Estimamos que lo ideal es realizarlas en los sprints que los hicimos (estimación perfecta).

En cuanto al resto estimamos que como queríamos acabar en Mayo y los sprints son de aproximadamente un 1 mes, necesitamos un total de siete u ocho sprints.

Total de Tareas / nº total de sprints = Tareas por sprint --> 66/8 aprox. 8

Como hay más tareas en Web y Android hemos considerado que lo ideal sería 8 tareas por cada sprint (3 de Android, 3 de Web y 2 de Back-End).



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

4. Documentación referencial

4.1 Guía de estilos

La guía de estilo que hemos utilizado para realizar los diferentes documentos es la siguiente:

4.1.2 Introducción

El propósito de éste documento es facilitar la labor de dar un formato a nuestros documentos por ese motivo vamos a describir unos estilos que serán implantados entre los miembros del grupo para la creación de documentación y desarrollo del software.

4.1.3 Formato

Todos los documentos de nuestro proyecto (visión, gestión y planificación, requisitos...) estarán estructurados de la siguiente forma:

Portada: incluye nombre de grupo, proyecto, tipo documento y versión.

Historial de revisiones: fecha, versión, breve descripción de las modificaciones y los autores.

Tabla contenidos: índice que será generado automáticamente.

Todas las hojas contendrán una cabecera en la que se indica el nombre del proyecto, fecha y versión de la última modificación junto con el tipo de documento.

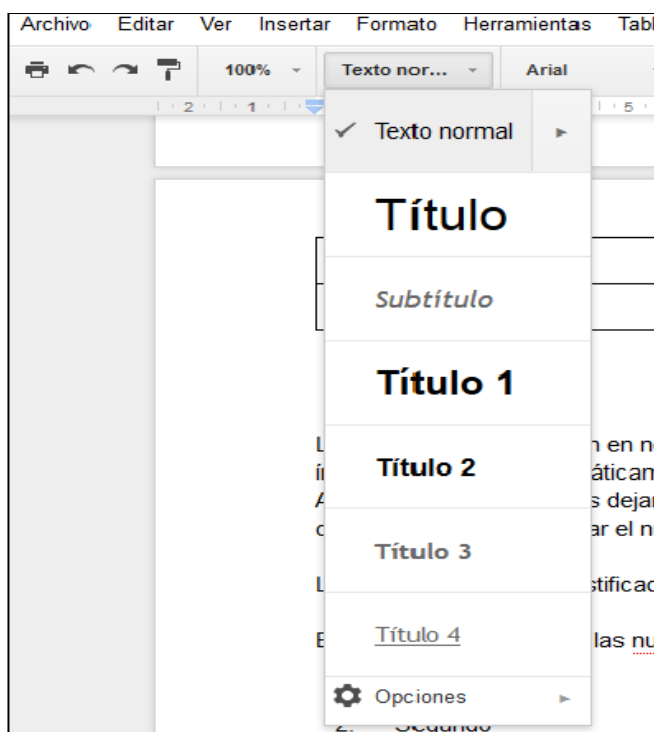
A pie de página se indicará el número de hoja del documento.

4.1.4 Especificaciones generales

En primer lugar vamos a definir el texto normal que será del tipo de letra Arial y tamaño 11, es el tipo que viene predefinido por google docs. El tipo de letra se aplicará a todo el documento incluyendo los títulos y subtítulos con la salvedad de que estos modificarán su tamaño.

Los títulos y subtítulos irán en negrita y su tamaño será respectivamente 18 y 13. Para que el índice se actualice automáticamente hay que usar Título 1 y 2 que deben estar configurados con el tipo de letra.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014



Antes de cada título debes dejar dos líneas en blanco para que el texto se vea más ordenado y claro, así el podrá destacar el nuevo apartado.

La alineación debe ser justificada a la izquierda y el interlineado de 1.15.

El uso de las listas serán las enumeradas, ejemplo:

Primero
Segundo
Tercero
 Tres punto 1

Se aconseja que cuando se vaya a empezar un nuevo documento coja la base de éste o de cualquier otro ya terminado y lo modifique encima para que la configuración sea la correcta

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

4.1.5 Guía programación

La guía está enfocada para el lenguaje java pero las ideas se pueden extraer para cualquier otro, es importante reseñar que todo el código y comentarios deberán realizarse en inglés.

4.1.5.1 Formato de líneas

No usar más de 80 caracteres por línea, de esta forma se puede visualizar el código correctamente desde un editor de texto o una hoja impresa de tamaño DIN4

Cada vez que se comience un nuevo bloque de instrucciones debe estar sangrado adecuadamente, a la hora de sangrar es recomendable utilizar entre 2-4 espacios en vez de una tabulación para evitar incompatibilidades entre editores de texto.

4.1.5.2 Ficheros

Incluir una sola clase o interfaz por fichero, si es necesario realizar algún comentario global deberá de realizarse al principio.

Si la clase forma parte de algún paquete deberá ser mencionado al principio y a continuación los paquetes importados si fuera necesario.

4.1.5.3 Clases

En primer lugar irán los comentarios de la clase, las variables irán ordenados de la siguiente forma:

- Estáticas
- Públicas
- Protegidas
- Privadas

A continuación se declararán las constructoras, y lo seguirán las funciones ordenadas según su interrelación y no por visibilidad. Por ejemplo si un método público A utiliza dos privados B y C , A irá en primer lugar; esto se hace de esta forma para leer el código y sea más fácil realizar un seguimiento de su funcionalidad.

Es importante no dejar un espacio entre el nombre de la función y el paréntesis de los parámetros.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

4.1.5.4 Identificadores

Escoger identificadores significativos y lo más breves posible

Las clases comenzarán con las iniciales de cada palabra en mayúscula: `class UserParty;`

Las variables irán en minúscula en caso de necesitar dos palabra se usará “_” : `int id_local;`

Las constantes se representan igual que las variables pero en mayúsculas.

Los métodos deben ser nombrados a ser posibles con un verbo si se precisan más palabras irán con la inicial en mayúsculas: `public void getUser(int valuer){`

4.1.5.5 Sentencias de control

Colocar la apertura de bloque (`{`) al final de la línea inicial de la sentencia; el fin de bloque (`}`) colocarlo en una línea a parte, si la sentencia es un `if- else`, el `else` debe situarse en la siguiente línea de fin de bloque:

```
if (a==0){
    date=6;
}
else {
    date=2;
}
```

Aunque solo haya una sentencia en el bloque colocar los paréntesis.

4.1.5.6 Documentación

Documentar el código nos puede facilitar a guiarnos y a aclararnos de la finalidad de la clase o función por ese motivo se aconseja:

En la cabecera añadir una pequeña descripción que incluya información acerca de su objetivo.

Indicar la fecha de modificación y sus autores, por si surgiera alguna duda para tenerlos identificados y facilitar la comunicación con ellos.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

5. Documentación de organización

En las primeras semanas los sprint fueron más cortos y se hicieron reuniones largas los miércoles en clase de laboratorio de ingeniería del software (salvo alguna reunión online) con el objetivo de definir bien el proyecto, elegir lo mejor posible las tecnologías y escoger una metodología de trabajo que se seguirá (salvo alguna excepción) durante todo el curso. Estas son las actas:

5.1 Acta reunión 1

Fecha: 6 de noviembre de 2013

Datos generales

- Asistencia: presentes todos los miembros del grupo.
- Duración aproximada: 2 horas.

Propósito de la reunión

- Toma de contacto de los diez miembros del grupo.
- Primera definición del proyecto.

Decisiones

- Aceptación de la visión:
 - Todos de acuerdo en la visión conjunta del proyecto. Faltan concretar más cosas de la misma.
- Requisitos:
 - Desestimación de la idea de las bebidas en el local
 - Posible incorporación a listas de discoteca en app.
 - Estados distintos para saber si se está dentro del local y así activar la gramola (va a ir, está, se fue), que se registrara por MAC
 - Bebida preferida en el perfil para mejorar la experiencia del local.
- Tecnologías:
 - Uso de Balsamiq para el diseño de los mockups de web y app.
 - Uso de una herramienta aún por decidir, integrable con GitHub, para seguir bien

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- la metodología SCRUM.
 - Andrés realizara un documento para decidir que tecnología usamos en el back-end y decidirlo en la siguiente reunión
 - Uso de MySQL (según Ruby o PHP) y SQLite para la base de datos.
- Organización
 - Pretendemos cambiar de tecnología únicamente entre la gente que quiera, pero habrá gente especializada en una tecnología, los cambios serán voluntarios.
 - Reparto inicial:
 - Android: Álvaro, Ismael, Beatriz, Sergio, Juan, Luis
 - Web: Javier, Saskya, Alicia, Yuleska
 - BackEnd: Andrés, Adrián
 - Diseño inicial de la aplicación: Sergio, Yuleska, Saskya

5.2 Acta reunión 2

Fecha: 7 de noviembre de 2013

Hora: 17:20 después de clase de ingeniería del software.

Datos generales

- Asistencia: presentes todos los miembros del grupo menos Luis.
- Duración aproximada: media hora.

Propósito de la reunión

- Reparto inicial de tareas.

Decisiones

- Requisitos:
 - Tres perfiles: DJ, Local, Usuario normal (fiestero).
- Tareas a repartir
 - Concretar la visión quitando etc y decidir sobre eso la tecnología de la BD:
 - Perfil del usuario – Ismael
 - Perfil del local y dj – Adrian
 - Gestión amistades -- Saskya
 - Gestión de locales -- Sergio

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Checkin -- Juan
- Buscadores -- Beatriz
- Información sobre el local -- Alvaro
- Privacidad -- Yuleska
- Gramola -- Alicia
- Escenarios posibles de uso -- Javier

5.3 Acta reunión 3

Fecha: 13 de noviembre de 2013

Hora: laboratorio de ingeniería del software.

Datos generales

- Asistencia: presentes todos los miembros del grupo.
- Duración aproximada: 2 horas aproximadamente.

Propósito de la reunión

- Proponer más requisitos. Revisar las tareas realizadas por cada miembro del grupo y juntarlas en un documento. Concretamos cada punto de la visión para poder especificar en un futuro el diseño

Decisiones

- **Requisitos**
 - El registro se realizará mediante Facebook y Google+ o manualmente. Twitter solo se utilizará para compartir.
 - Se hará un tour virtual por la aplicación para conocer su uso.
 - Se bloquearán las cuentas pasadas 24 horas si no se confirma la cuenta si se usa el registro de la aplicación
- **Tareas a repartir**
 - Especificación - Yuleska, Saskya, Ismael, Juan, Sergio, Beatriz.
 - Selección de herramientas y tecnologías - Andrés.
 - Hablar con terceros – Adrian
 - Definir los estilos y cambiar los documentos - Luis.
 - Definir los recursos necesarios por campos y personas - Alicia.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Hacer tabla EDT - Javier.
- Investigar tecnologías Android - Álvaro.

5.4 Acta reunión 4

Fecha: 19 de noviembre de 2013

Hora: 22:15.

Datos generales

- Asistencia: Beatriz, Ismael, Juan, Saskya, Sergio y Yuleska.

Propósito de la reunión

- Recopilación de requisitos para la especificación.

Decisiones

● Requisitos

- Búsquedas: fiestero busca fiestero, fiestero busca local, fiestero busca DJ
- Idea posible de implementar en un futuro :
 - Sergio: Añadir a las búsquedas los locales más frecuentados con la posibilidad de ser actualizados cada semana.
 - Ismael y Sergio: Ganar puntos por ir a un local. El que más pague dará más puntos (foursquare).
 - Sergio: Ganar estrellas si se realiza check in.
- Check in : Se realizará manualmente, wifi o gps . Si el local cambia el router nos tendrá que comunicar.
- Privacidad: El nombre siempre es visible.

5.5 Acta reunión 6

Fecha: 20 de noviembre de 2013

Hora: laboratorio de ingeniería del software.

Datos generales

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Asistencia: Todos los miembros del grupo menos Andrés.

Propósito de la reunión

- Revisar las tareas realizadas. Repartir más tareas.

Decisiones

- Tareas a repartir
 - Casos de uso checkin - Andrés
 - Casos de uso registros locales - Alicia
 - Unificación de la visión detallada -- Luis (hecho).
 - Continuación contacto con locales – Adrian
 - Entrega documento herramienta backend - Andrés (Anterior Sprint)
 - EDT - Javier
 - Comunicación del servidor con la app - Álvaro
 - Mockups - grupo de diseño

5.5 Acta reunión 7

Fecha: 27 de noviembre de 2013

Hora: laboratorio de ingeniería del software.

Datos generales

- Asistencia: Todos los miembros del grupo menos Andrés.

Propósito de la reunión

- Revisar las tareas realizadas. Repartir más tareas.

Decisiones

- Tecnologías
 - Usaremos Trello:
 - Tarjetas Verde → Android
 - Tarjetas Roja → BackEnd

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

- Tarjetas Morada → WebApp
- Tarjetas Morado Azul → Diseño Web
- Tarjetas Verde Azul→ Diseño Android

- Tareas a repartir:
 - Reunir los diagramas de Gannt. - Javier
 - Definir en trello todas las AEs Android - Álvaro, Ismael
 - Definir en trello todas las AEs BackEnd - Adrian, Andrés
 - Definir en trello todas las AEs WebApp - Alicia, Luis
 - Definir en trello todas las AEs Diseño Web - Yuleska, Saskya, Beatriz
 - Definir en trello todas las AEs Diseño Android - Sergio, Juan

A partir de esta fecha el resto de reuniones serán reuniones rápidas para ponerse al día y reuniones entre los miembros del grupo en cada final de sprint (back-end, android o web) para reparto de tareas.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

6. Sugerencias de profesionales del sector

Con el objetivo de obtener más requisitos y para saber qué es lo que pediría un cliente para una mejora futura del proyecto y una posibilidad de darle salida en el mercado se iniciaron conversaciones con DJs profesionales y locales para conocer su opinión.

6.1 Conversación con DJ

Al no poder mostrar la música que está sonando automáticamente, sino que debemos marcarla manualmente, poder grabar cada x tiempo unos 10s de música y que el usuario pueda escucharlo desde donde esté.

6.2 Conversación con propietario del local

Generar un apartado donde poder gestionar la administración del local, gastos, ingresos etc. A nivel mensual y a nivel anual, todo ello que se pueda imprimir o poder crear Excel para poder retocarlo.

Dentro del perfil de local generar el apartado de fotos, pero únicamente del local vacío, para poder conocer instalaciones. (Incluido ya)

Apartado para posible contacto con local, para informar de privados, reservas etc...

Reproductor para gestionar las listas de música que haga la gente mediante votaciones. (Esta idea ya la teníamos para la gramola, pero para hacer esto deberíamos incluir un reproductor propio).

Gestión de almacén de locales, cantidades en barra, bebidas que entran y que salen.

6.3 Conversación con miembro de grupo de música

Generarlo también para grupos, donde puedan poner música (ya sea por soundcloud o incluida en su perfil...), donde la cuenta del grupo pueda admitir a varios miembros y pueda mostrar su información.

Para poder hacer esto directamente sería generar un perfil único para este tipo de gente.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

6.4 Conversación sobre listas de discoteca

Cada local tiene su propio convenio.

Para esta idea entonces debemos pensar en que el administrador (local) pueda crear, borrar y modificar listas.

Deberíamos añadir un apartado en la webapp para este tema.

Creación de listas (administración)

Crear nueva lista

- Fecha (día para la lista)
- Fecha finalización
- Hora de finalización
 - o Cuando finalice deberíamos mandar un mail al correo del administrador en el que pueda ver todas las personas apuntadas en la lista mediante la app.
- Título de la lista
- Información (Entrada gratuita, copas gratis...)

Usuario (como apuntarse a las listas)

Pestaña Listas

- Mostramos todas las listas creadas por el local (sólo serán visibles las que no hayan caducado)
 - o Selecciona la que quiera y ya queda automáticamente apuntado, mostrando un aviso de que se ha apuntado correctamente.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

7. Modelo de negocio

7.1 Listas de locales

Hemos considerado que las listas de locales es una de las posibilidades a explotar para conseguir dinero. Los locales gastan una suma importante en promocionarse, relaciones públicas, consumiciones de invitación... todo esto simplemente para que posibles clientes entren en su local. Nosotros pretendemos ahorrar una parte de ese dinero con nuestra funcionalidad de listas.

Un local que use nuestra aplicación podrá gestionar sus listas sin necesidad de un relaciones públicas, además de que estará todo informatizado y no en papeles como ocurre ahora.

Por este servicio nuestra idea es cobrar una pequeña cantidad por cada persona que entre al local apuntándose en una de nuestras listas.

7.2 Promoción de locales

En cualquier aplicación en la que hay búsquedas de resultados, alguien puede tener preferencia por salir más arriba o más destacado. Nuestra idea es ofrecer esa posibilidad a los locales, de tal forma que puedan aparecer en cualquier búsqueda, o simplemente que su local aparezca el primero en búsquedas filtradas.

Por este servicio cobraremos una cantidad que produciría beneficios en nuestro posible negocio.

7.3 Publicidad

Es posible añadir banners o pantallas extra con anunciantes. Estos anuncios podrán ser de cualquier tipo, sin restricción por temas. Habitualmente esta práctica es una de las que más beneficio supone a una aplicación.

7.4 Conclusiones

No es fácil hacer dinero por una aplicación gratuita, lo más probable es que la mayoría de ingresos puedan llegar por publicidad si el número de usuario es grande, ya que tendrán una visibilidad enorme.

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

8. Anexos

8.1. ANEXO 1 - EDT

Fase	DEFINICIÓN		DESARROLLO			MANTENIMIENTO	
Actividades estructurales	comunicación con el cliente	Planificación y gestión de riesgos	Análisis y diseño software	Generación de código	Pruebas	Corregir defectos	Ampliar funcionalidad
Acción	Ingeniería de Requisitos						
Proyecto	Tarea: Visión - Perfil Usuario						
	Realizada por: Isma						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Perfil local y Dj						
	Realizada por: Adrián						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Gestión amistades						
	Realizada por: Saskya						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Gestión locales						
	Realizada por: Sergio						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Checkin						
	Realizada por: Juan						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Buscadores						
	Realizada por: Bea						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Información local						
	Realizada por: Alvaro						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						
Proyecto	Tarea: Visión - Privacidad						
	Realizada por: Yuli						
	Inicio: 07/11/2013						
	Fin: 13/11/2013						

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Fase	DEFINICIÓN	DESARROLLO	MANTENIMIENTO
Proyecto	Tarea: Visión - Gramola Realizada por: Alicia Inicio: 07/11/2013 Fin: 13/11/2013		
Proyecto	Tarea: Visión - Escenarios Realizada por: Javi Inicio: 07/11/2013 Fin: 13/11/2013		
Proyecto	Tarea: ruby vs php Realizada por: Andrés Inicio: 07/11/2013 Fin: 14/11/2013		
Proyecto	Tarea: Hablar con terceros Realizada por: Adrián Inicio: 13/11/2013 Fin: 21/11/2013		
Proyecto	Tarea: Especificación Realizada por: Isma, Juan, Bea, Yuli, Saskya, Sergio Inicio: 13/11/2013 Fin: 21/11/2013		
Proyecto	Tarea: Selección herramientas y tecnología Realizada por: Andrés Inicio: 13/11/2013 Fin: 21/11/2013		
Proyecto	Tarea: Definir estilos y cambiar documentos Realizada por: Luis Inicio: 13/11/2013 Fin: 21/11/2013		
Proyecto	Tarea: Definir recursos necesarios (campos y personas) Realizada por: Alicia Inicio: 13/11/2013 Fin: 21/11/2013		
Proyecto	Tarea: Investigar tecnologías android Realizada por: Alvaro Inicio: 13/11/2013 Fin: 21/11/2013		
Proyecto	Tarea: Caso de uso check-in Realizada por: Andrés Inicio: 21/11/2013 Fin: 27/11/2013		
Proyecto	Tarea: Caso de uso registro de locales Realizada por: Alicia Inicio: 21/11/2013 Fin: 27/11/2013		
Proyecto	Tarea: Unificación de la visión detallada Realizada por: Luis Inicio: 21/11/2013 Fin: 27/11/2013		
Proyecto	Tarea: Contacto con locales Realizada por: Adrián Inicio: 21/11/2013 Fin: 27/11/2013		
Proyecto	Tarea: Comunicación del servidor con la app Realizada por: Alvaro Inicio: 21/11/2013 Fin: 27/11/2013		
Web, Android	Tarea: Mockups Realizada por: Yuli, Bea, Saskya, Sergio, Isma, Juan Inicio: 21/11/2013 Fin: 27/11/2013		

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Fase	DEFINICIÓN	DESARROLLO	MANTENIMIENTO
Proyecto	Tarea: Casos de uso, Diagramas UML		
	Realizada por: Alicia, Andres Fin: 18/12/2013		
Web		Tarea: Prototipo final Realizada por: Alicia, Javier, Saskya, Y uleska Fin: 18/12/2013	
		Tarea: Prototipo final Realizada por: Alvaro, Ismael Fin: 18/12/2013	
Backend		Tarea: Prototipo final Realizada por: Adrian, Andres, Luis Fin: 18/12/2013	
		Tarea: Funcionalidad Fiestero Realizada por: Alicia, Javier, Saskya, Y uleska Fin: 22/01/2014	
Android		Tarea: Aplicación Android estática Realizada por: Juan, Beatriz, Sergio, Alvaro, Ismael Fin: 22/01/2014	
		Tarea: Base de datos y PHP logins registros y editar Realizada por: Adrian, Andres, Luis, Alvaro Fin: 22/01/2014	

8.2. ANEXO 2 - Diagrama de Gantt. Planificación de tareas

Nombre	Duración	Comienzo	Fin
Visión - Perfil Usuario	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Perfil local y DJ	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Gestión de amistades	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Gestión locales	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Check - in	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Buscadores	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Información local	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Privacidad	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Visión - Gramola	5 días	07 noviembre 2013 8:00	13 noviembre 2013 17:00
Escenarios	5 días	15 noviembre 2013 8:00	21 noviembre 2013 17:00
Ruby vs PHP	6 días	07 noviembre 2013 8:00	14 noviembre 2013 17:00
Especificación de la visión	5 días	13 noviembre 2013 8:00	19 noviembre 2013 17:00
Selección de herramientas y tecnología	5 días	20 noviembre 2013 8:00	26 noviembre 2013 17:00
Hablar con terceros	5 días	13 noviembre 2013 8:00	19 noviembre 2013 17:00
Definir los estilos y cambiar los documentos	5 días	13 noviembre 2013 8:00	19 noviembre 2013 17:00
Definir los recursos necesarios (campos y personas)	5 días	13 noviembre 2013 8:00	19 noviembre 2013 17:00
Investigar tecnologías Android	5 días	13 noviembre 2013 8:00	19 noviembre 2013 17:00

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Nombre	Duración	Comienzo	Fin
Caso de uso registro locales	5 días	21 noviembre 2013 8:00	27 noviembre 2013 17:00
Contacto con locales	5 días	20 noviembre 2013 8:00	26 noviembre 2013 17:00
Unificación de la visión detallada	5 días	20 noviembre 2013 8:00	26 noviembre 2013 17:00
Comunicación del servidor con la app	6 días	20 noviembre 2013 8:00	27 noviembre 2013 17:00
Mockups	11 días	20 noviembre 2013 8:00	04 diciembre 2013 17:00
Prototipo Final	11 días	04 diciembre 2013 8:00	18 diciembre 2013 17:00
BackEnd - Manejo del login	16 días	18 diciembre 2013 8:00	08 enero 2014 17:00
Android - Login estatico	16 días	18 diciembre 2013 8:00	08 enero 2014 17:00
WebApp - Login estatico	16 días	18 diciembre 2013 8:00	08 enero 2014 17:00
BackEnd - pruebas de login	11 días	09 enero 2014 8:00	23 enero 2014 17:00
Android - Conexion del login	10 días	09 enero 2014 8:00	22 enero 2014 17:00
BackEnd - Conexion del login	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Prueba de login	10 días	09 enero 2014 8:00	22 enero 2014 17:00
WebApp - Prueba de login	10 días	09 enero 2014 8:00	22 enero 2014 17:00
BackEnd - Manejo del registro	16 días	18 diciembre 2013 8:00	08 enero 2014 17:00
Android - Registro estatico	16 días	18 diciembre 2013 8:00	08 enero 2014 17:00
WebApp - Registro estatico	16 días	18 diciembre 2013 8:00	08 enero 2014 17:00
BackEnd - Pruebas Registro	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Conexion Registro	10 días	09 enero 2014 8:00	22 enero 2014 17:00
BackEnd - Adaptación Registro	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Prueba Registro	10 días	09 enero 2014 8:00	22 enero 2014 17:00
WebApp - Prueba Registro	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Final de Sprint	0 días	22 enero 2014 8:00	22 enero 2014 8:00
Android - Navegación por la App	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Editar Perfil Estático	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Inicio Estático	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Conexión Editar Perfil	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Conexión Inicio	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Prueba Navegación por la App	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Prueba Inicio	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Prueba Editar Perfil	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Listado de amigos, peticiones de amistad y FB	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Listado Locales con Búsqueda y mapa	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Conexión Listado amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Conexión listado locales	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Prueba listado Amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Prueba listado locales	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Vista detallada del local (sin gramola)	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Final de Sprint	0 días	19 febrero 2014 17:00	19 febrero 2014 17:00
Android - Vista detallada de un usuario	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - implementación del check-in	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Conexión vista detallada de un local	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Conexión vista detallada de un usuario	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Conexión del check-in	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Conexión ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Prueba vista detallada del local	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Android - Prueba vista detallada de un usuario	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
Final Sprint	0 días	19 marzo 2014 17:00	19 marzo 2014 17:00

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Nombre	Duración	Comienzo	Fin
Android - Prueba check-in	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Prueba ajustes de privacidad	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Listado Eventos	10 días	09 enero 2014 8:00	22 enero 2014 17:00
Android - Conexión Listado Eventos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Prueba Listado Eventos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
Android - Listado de Djs	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Conexión Listado de Djs	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Prueba Listado de Djs	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Vista detallada Dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Conexión Vista detallada Dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Prueba vista detallada Dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Galeria de imágenes	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Conexión Galeria de imágenes	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Pruebas Galeria de imágenes	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Vista detallada de una imagen	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Conexión vista detallada imagen	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Android - Prueba Vista detallada imagen	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Final Sprint	0 días	16 abril 2014 17:00	16 abril 2014 17:00
Android - Implementación de la gramola	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
Android - Conexión gramola	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
Android - Pruebas Gramola	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
WebApp - Editar Perfil (Fiestero, Local, DJ)	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Adaptación Editar Perfil (Fiestero, Local, DJ)	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Prueba Editar Perfiles	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Home (Fiestero, Local, DJ)	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Adaptación Home (Fiestero, Local, DJ)	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Prueba Home (Fiestero, Local, DJ)	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Búsqueda de amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Adaptación Búsqueda amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Prueba Búsqueda amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Búsqueda Local	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Adaptación Búsqueda local	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Agregar amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Adaptación agregar amigos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
WebApp - Ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Adaptación ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Prueba Ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00

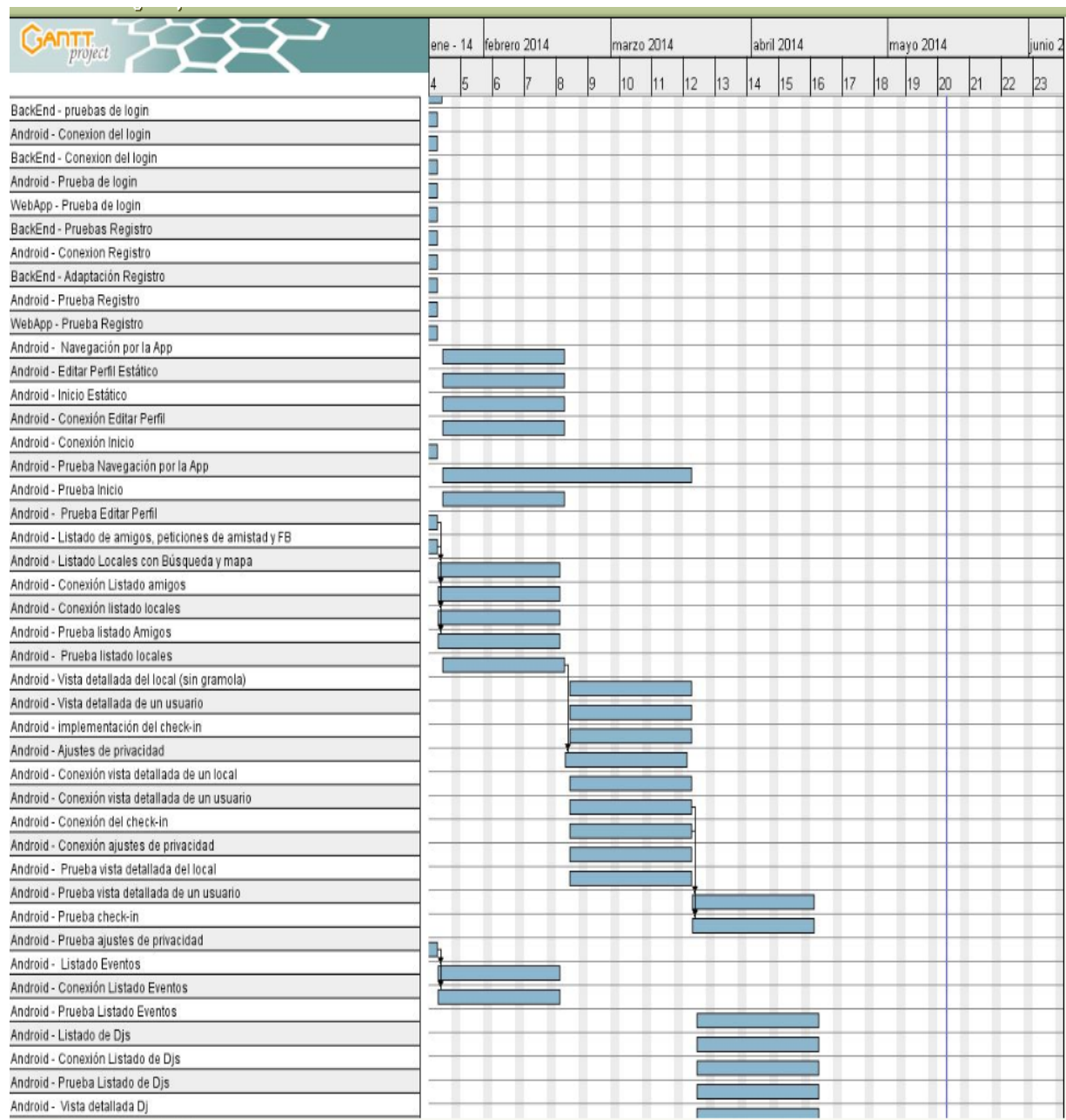
where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

Nombre	Duración	Comienzo	Fin
BackEnd - Adaptación ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Prueba Ajustes de privacidad	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Gestionar eventos	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Adaptación gestionar eventos	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Prueba gestionar eventos	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Gestionar Listas	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Adaptación gestionar listas	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Prueba gestionar listas	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Seguir local	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Adaptación seguir local	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Seguir Dj	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Gramola Dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
BackEnd - Adaptación gramola dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
WebApp - Prueba gramola dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
WebApp - Gestionar imágenes	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Adaptación gestionar imágenes	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Prueba gestionar imágenes	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
WebApp - Favorito Dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
BackEnd - Adaptación Favorito Dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
WebApp - Favorito local	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
BackEnd - Adaptación Favorito local	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
WebApp - Prueba gestionar amigos	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
WebApp - Prueba gestionar local	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
WebApp - Prueba gestionar Dj	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
BackEnd - Petición JSON y test unidad Login Fb, g+ y propietario	10 días	09 enero 2014 8:00	22 enero 2014 17:00
BackEnd - Petición JSON y test unidad registro	10 días	09 enero 2014 8:00	22 enero 2014 17:00
BackEnd - Petición JSON y test unidad listado eventos	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Petición JSON y test unidad listado dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
BackEnd - Petición JSON y test unidad listado locales	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Petición JSON y test unidad listado amigos y peticiones amis	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Búsqueda amigos otras redes sociales	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Petición JSON y test unidad vista detallada local	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
BackEnd - Petición JSON y test unidad vista detallada usuario	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00

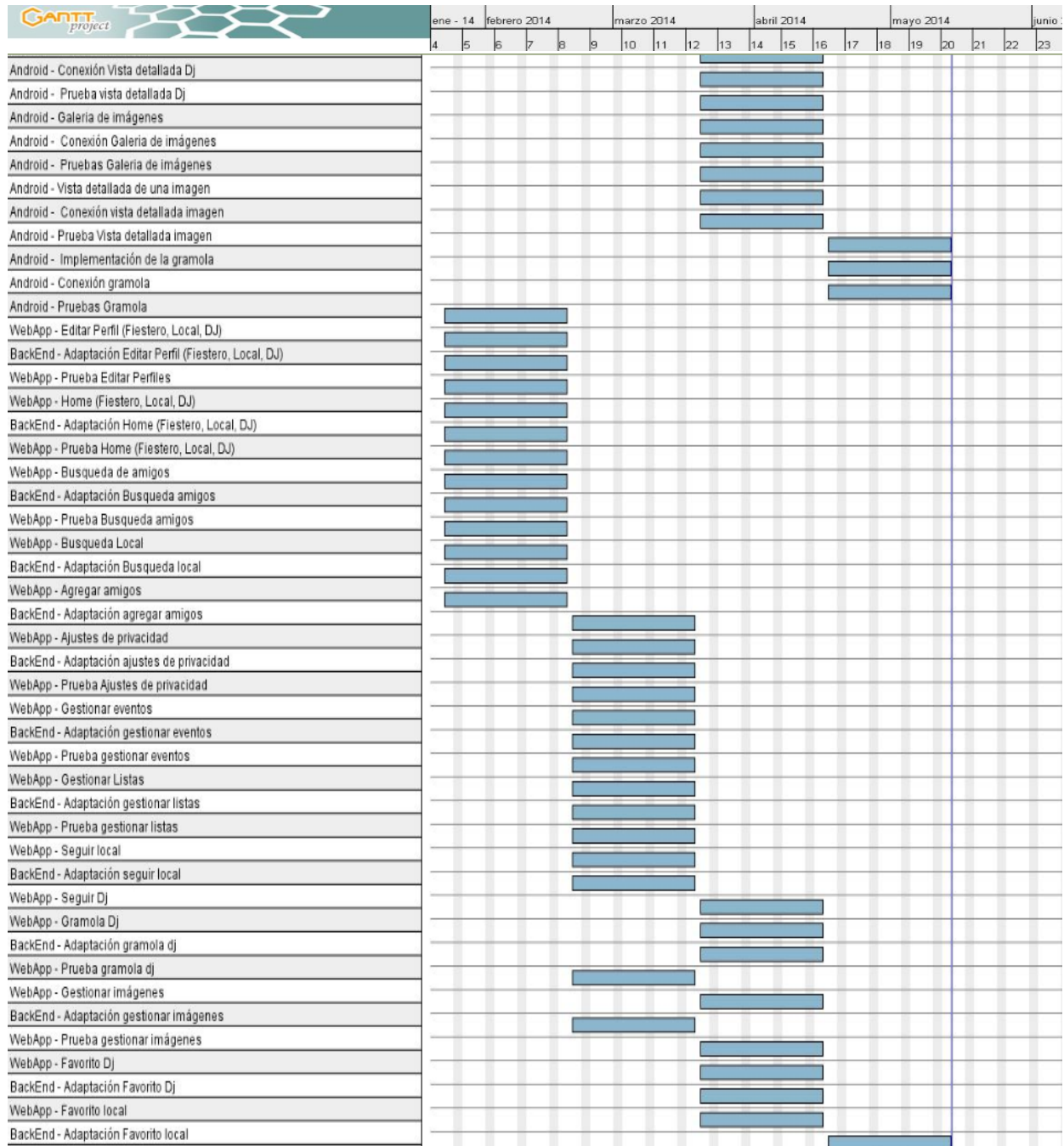
Nombre	Duración	Comienzo	Fin
BackEnd - Petición JSON y test unidad vista detallada usuario	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Petición JSON y test unidad vista detallada dj	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
BackEnd - Petición JSON y test unidad galeria de imágenes	20 días	20 marzo 2014 8:00	16 abril 2014 17:00
Final Sprint	0 días	16 abril 2014 8:00	16 abril 2014 8:00
BackEnd - Petición JSON y test unidad gramola	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
BackEnd - Petición JSON y test unidad checkin	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - gestion permisos checkin	20 días	20 febrero 2014 8:00	19 marzo 2014 17:00
BackEnd - Gestion locales google maps	20 días	23 enero 2014 8:00	19 febrero 2014 17:00
BackEnd - Interfaz escuchando ahora gramola	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
BackEnd - Crear base de datos de música de cada local	20 días	17 abril 2014 8:00	14 mayo 2014 17:00
BackEnd - Crear distintos perfiles	20 días	23 enero 2014 8:00	19 febrero 2014 17:00

where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

8.3. ANEXO 3 - Diagrama de Gantt.



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014



where2night	Versión: 1.4
Documentación Final	Fecha: 05/06/2014

