

# **Proyecto** **“Truequéalo”**

## **Grupo** **“The Red Chicken”**

### **"DOCUMENTO DE ESPECIFICACIÓN"**

Hito: 0

Fecha entrega: 05-10-16

Versión: 1

#### **Componentes:**

- Raquel Yuste Torregrosa
- Yolanda Torregrosa Hernández
- Juan Fco Bustos Correas
- Pablo Serna Martínez
- Alejandro Torres Mateu

## Contenido

### Contenido

#### 1. Introducción

##### 1.1. Propósito

#### 2. Descripción general

##### 2.1. Ámbito del sistema (Perspectiva del producto).

##### 2.2. Funcionalidades generales.

##### 2.3. Características de los usuarios.

##### 2.4. Interfaces externas.

##### 2.5. Requisitos (suposiciones y dependencias).

##### 2.6. Restricciones.

##### 2.7. Requisitos futuros.

#### 3. Requerimientos específicos

##### 3.1. Requerimientos funcionales.

##### 3.2. Requerimientos no funcionales.

#### 4. Apéndices

##### 4.1. Definiciones, acrónimos y abreviaturas

##### 4.2. Referencias

## 1. Introducción

### 1.1. Propósito

El propósito general del documento es proporcionar una correcta comunicación entre el cliente y el equipo de trabajo que va a desarrollar el proyecto con la finalidad de definir los requisitos y objetivos que debe cumplir. Por esta misma razón, el documento va dirigido no solo al cliente sino también a todos los integrantes del equipo que participan en el proyecto.

## 2. Descripción general

### 2.1. Ámbito del sistema (Perspectiva del producto).

Aplicación móvil y web que permite el intercambio, mediante la economía del trueque, de productos y servicios entre usuarios, mejorando el abanico de aplicaciones actuales. Además, incluirá herramientas para facilitar la comunicación entre los usuarios del sistema como mensajería instantánea o un sistema para conectar usuarios entre los que existe posibilidad de trueque, ya que uno posee algo que el otro necesita, y viceversa.

La aplicación se basará en el diseño *mobile first*, ya que nuestro servicio está orientado a los usuarios procedentes de dispositivos móviles (principalmente desde la app móvil), sin dejar de lado a los usuarios que accedan desde la aplicación web.

El servicio web contará con un panel de administración en el que los administradores del servicio podrán gestionar todo lo relacionado con los contenidos del sistema, como usuarios, anuncios, publicidad, etc. Además, este panel también mostrará métricas que permitan conocer datos sobre la evolución de nuestra aplicación.

### 2.2. Funcionalidades generales

Teniendo en cuenta que *Truequéalo* es un sistema que presenta numerosas funcionalidades que ofrecen la posibilidad al usuario de cumplir su objetivo de manera satisfactoria, en este apartado vamos a enumerar las principales e indispensables.

#### 2.2.1. Desde el punto de vista web y app

##### Usuario No Registrado

- **Crear cuenta nueva:** El cliente que no esté registrado podrá formar parte de nuestro sistema y disfrutar de todas sus ventajas.
- **Filtrar búsqueda:** Adaptar la búsqueda de trueques a unas características específicas.
- **Consultar anuncios:** Un usuario no registrado podrá consultar anuncios publicados en nuestro sistema.
- **Enviar dudas o sugerencias:** Un usuario no registrado podrá contactar con nosotros a través de la web introduciendo un correo electrónico.
- **Compartir anuncios:** Un usuario no registrado podrá compartir anuncios en sus redes sociales.
- **Reportar anuncios:** Denunciar anuncios por contenido inapropiado.

- **Reportar usuarios:** Un usuario podrá reportar a otro si considera que se está comportando inapropiadamente

### Usuario Registrado

Para el usuario registrado, se repiten todas las funcionalidades del usuario no registrado, excepto la de crear una cuenta nueva.

- **Inicio de sesión:** Si el usuario ya está registrado podrá acceder a su cuenta personal autenticándose mediante usuario y contraseña o las principales RRSS.
- **Iniciar un chat:** Un usuario registrado podrá iniciar un chat con otro usuario a través de un anuncio.
- **Enviar mensajes:** Un usuario podrá enviar y recibir mensajes de otro usuario a través del chat. Estos mensajes podrán ser tanto textos, como otros productos o imágenes.
- **Edición de datos de usuario:** Una vez registrado, el usuario podrá modificar su perfil.
- **Publicación de anuncios:** Un usuario registrado tendrá la posibilidad de publicar tantos anuncios como desee.
- **Gestionar favoritos:** Guardar o archivar en una sección de la aplicación todos aquellos anuncios que el usuario haya considerado de interés.
- **Baja del sistema:** El cliente puede eliminar su cuenta y todos sus datos asociados. Por seguridad, los datos eliminados permanecerán en la BD durante los dos años siguientes.
- **Realizar cursos:** Un usuario registrado podrá realizar cursos relacionados con la economía del trueque y los anuncios de productos en internet. Esto le permitirá obtener insignias que mejorarán su perfil de usuario.
- **Ver anuncios sugeridos:** Un usuario registrado podrá visualizar una serie de anuncios sugeridos en base a sus intereses y decidir si le interesa, iniciar un chat con el usuario que publica el anuncio o descartar el producto si no está interesado en él.

### 2.2.2. Desde el punto de vista del BackOffice

Habrà un súper usuario que podrá dar o retirar acceso a los administradores a las diferentes secciones del backoffice.

### Usuario administrador

- **Iniciar sesión:** Si el usuario ya está registrado podrá acceder a su cuenta personal autenticándose mediante usuario y contraseña o las principales RRSS.
- **Gestión de reportes:** Una vez recibido el reporte de un usuario, la empresa se encarga de analizar la veracidad de la denuncia y el posible castigo hacia el anuncio que ha sido reportado.
- **Moderación de anuncios:** El administrador del sistema se encargará de revisar y validar los diferentes anuncios creados por los usuarios.

- **Visualización de anuncios:** El administrador del sistema podrá ver una lista de los anuncios publicados, así como buscar anuncios según determinados criterios. La lista podrá ordenarse de forma ascendente o descendente en función de diferentes parámetros.
- **Gestión de categorías:** El administrador del sistema podrá visualizar las categorías de la web, así como editarlas, eliminarlas o crear nuevas.
- **Gestión de newsletter:** El administrador del sistema podrá enviar correos electrónicos con información general a todos los usuarios.
- **Gestión de sugerencias:** El administrador del sistema podrá visualizar las dudas y sugerencias de los usuarios y responderlas desde el panel de administración.
- **Consultar métricas:** El usuario podrá consultar las métricas referidas al número de usuarios registrados, el número de anuncios publicados, etc.

### 2.3. Características de los usuarios.

<b>Tipo de usuario</b>	Registrado
<b>Formación</b>	Básica
<b>Habilidades</b>	Conocimiento básico de uso de interfaces de software
<b>Actividades</b>	Inicio de sesión Filtrar búsqueda Consultar anuncios Iniciar un chat Enviar mensajes Compartir anuncios Edición de datos de usuario Publicación de anuncios Gestionar favoritos Reportar anuncios Bloquear usuarios Baja del sistema Enviar dudas o sugerencias Realizar cursos Ver anuncios sugeridos

<b>Tipo de usuario</b>	No registrado
<b>Formación</b>	Básica
<b>Habilidades</b>	Conocimiento básico de uso de interfaces de software
<b>Actividades</b>	Registrarse Filtrar búsqueda Consultar anuncios Enviar dudas o sugerencias Compartir anuncios

<b>Tipo de usuario</b>	Administrador
<b>Formación</b>	Básica
<b>Habilidades</b>	Manejo básico de herramientas de gestión.
<b>Actividades</b>	Iniciar sesión Gestión de reportes Moderación de anuncios Visualización de anuncios Gestión de categorías Gestión de newsletter Gestión de sugerencias

<b>Tipo de usuario</b>	Máster
<b>Formación</b>	Media
<b>Habilidades</b>	Manejo básico de herramientas de gestión y conocimiento en temas de marketing digital.
<b>Actividades</b>	Iniciar sesión Gestión de reportes Moderación de anuncios Visualización de anuncios Gestión de categorías Gestión de newsletter Gestión de sugerencias Consultar métricas

## 2.4. Interfaces externas.

### 2.4.1. Interfaz de usuario

La interfaz gráfica con la que el usuario final interactúa deberá ser intuitiva de manera que, sin un manual de uso, el usuario identifique rápidamente los componentes y las secciones del sistema. Además, la gama de colores utilizada deberá de contener colores contrastados que faciliten la legibilidad de los textos y la separación entre elementos.

Tanto la interfaz de la web como de la aplicación deberán ser compatibles con los navegadores y sistemas operativos más utilizados.

### 2.4.2. Interfaz con otros sistemas

La aplicación realizará llamadas a las APIs de las principales redes sociales (Facebook, Twitter, Google, etc.) para la autenticación del usuario mediante el protocolo OAUTH.

La plataforma también permitirá a los usuarios compartir cualquier anuncio mediante estas RRSS, para lo que se realizarán llamadas a la API y se integrarán las respectivas interfaces gráficas.

Además, desde la aplicación móvil también se podrá acceder a la cámara del dispositivo (interfaz hardware) para capturar fotografías en las pantallas de creación y edición de anuncios.

## 2.5. Requisitos (suposiciones y dependencias).

Para poder acceder al sistema, es imprescindible tener conexión a internet. Además, para poder descargar y utilizar la aplicación móvil, es necesario que el usuario disponga de un smartphone o tablet cuyo sistema operativo sea iOS o Android.

Se asume que los requisitos descritos en este documento son estables.

## 2.6. Restricciones

- Sólo se podrá acceder a la plataforma web mediante el dominio habilitado.

- Se utilizará una API Rest para realizar consultas a la BD y servir los recursos. La arquitectura utilizada será MVC.
- La BD seguirá un modelo entidad-relación. El sistema de gestión de base de datos utilizado será MySQL.
- La API Rest sólo responderá consultas procedentes de nuestra aplicación.
- El servidor deberá ser capaz de atender peticiones concurrentemente.
- Lenguajes y tecnologías en uso: HTML5, CSS3, AngularJS.
- Se debe hacer uso de los protocolos seguros de intercambio de datos vía internet.
- Se usarán tokens de sesión para limitar el acceso a usuarios no autorizados.
- Se realizarán copias de seguridad externas para garantizar la persistencia de los datos.

## 2.7. Requisitos futuros.

### 2.7.1. Requisito futuro 1

Número de requisito	1
Nombre de requisito	Anuncios en una conversación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

- El usuario podrá enviar automáticamente información sobre sus anuncios publicados desde la pantalla de la conversación (pantalla de detalle de un chat).

### 2.7.2. Requisito futuro 2

Número de requisito	2
Nombre de requisito	Verificación de usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

- El usuario podrá verificar su perfil de distintas formas: conectando su perfil a través de redes sociales, mediante un pago, etc.
- De esta forma el usuario podrá aumentar la confianza con el resto de usuarios.

### 2.7.3. Requisito futuro 3

Número de requisito	3
Nombre de requisito	Nuevos cursos de formación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

- El usuario dispondrá de multitud de cursos específicos sobre diferentes aspectos de la economía de trueque y de cómo publicar un anuncio en internet.
- De esta forma el usuario podrá diferenciarse del resto de usuarios y mejorar su reputación dentro de la aplicación.

### 2.7.4. Requisito futuro 4

Número de requisito	4
Nombre de requisito	Integración de bitcoins
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

- La plataforma integrará una sección de banco de tiempo utilizando bitcoins, donde se podrán intercambiar unos servicios por otros.
- Los usuarios realizarán servicios a otros usuarios y acumularán tiempo en su monedero.
- Este tiempo podrá utilizarse para “comprar” un servicio de otro usuario.

#### 2.7.5. Requisito futuro 5

Número de requisito	5
Nombre de requisito	Sugerencias por packs
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

- El usuario podrá seleccionar varios de sus anuncios para formar packs y encontrar sugerencias de productos que pueda intercambiar por el pack formado.
- Es decir, podrá buscar sugerencias no solo para intercambiar un producto por otro, sino un conjunto de productos por otro con un valor similar al del pack.

#### 2.7.6. Requisito futuro 6

Número de requisito	6
Nombre de requisito	Videos de anuncios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

- Si el anuncio tiene más de una foto, el sistema generará de forma automática un vídeo con las fotos, título y descripción del anuncio.
- Este vídeo se subirá a nuestra cuenta de youtube y se enlazará al anuncio para poder compartirlo.

### 3. Requerimientos específicos

#### 3.1. Requerimientos funcionales.

##### 3.1.1. Requisito funcional 1

Número de requisito	1
Nombre de requisito	Autenticación y acceso
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá loguearse con usuario y contraseña o bien usando redes sociales como Facebook, Google o Twitter.
- Si los datos introducidos por el usuario no son correctos se mostrará un mensaje por pantalla indicándolo.
- Si el usuario olvida su contraseña podrá usar la opción de recuperarla, en la que se le enviará una contraseña nueva al correo registrado.
- También podrá cerrar sesión en cualquier momento, siempre que esté previamente logueado.
- Las contraseñas de los usuarios se almacenan cifradas en la base de datos.



## 3.1.2. Requisito funcional 2

Número de requisito	2
Nombre de requisito	Registro
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá registrarse con usuario, contraseña y otros datos de contacto o bien usando redes sociales como Facebook, Google o Twitter.
- Las contraseñas de los usuarios se almacenan cifradas en la base de datos.

## 3.1.3. Requisito funcional 3

Número de requisito	3
Nombre de requisito	Administración de cuenta de usuario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá cambiar sus datos personales y de contacto, así como su contraseña.
- El usuario podrá eliminar su cuenta, aunque los datos se mantendrán en la base de datos durante un periodo de tiempo por motivos de seguridad.

## 3.1.4. Requisito funcional 4

Número de requisito	4
Nombre de requisito	Navegación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- La interfaz se dispondrá de modo que el usuario pueda acceder de un modo sencillo a todas las funcionalidades de la aplicación. Para ello se dispondrá de un menú de navegación de la aplicación.
- El usuario podrá acceder desde dicho menú hacia: Inicio, Perfil de usuario, Mensajes, Favoritos, Sugeridos, Ajustes, Ayuda.
- Menú de búsqueda en la parte superior de la pantalla principal.
- Habrá una sección en la que se mostrarán los datos referentes a la aplicación, tanto de contacto como legales.

## 3.1.5. Requisito funcional 5

Número de requisito	5
Nombre de requisito	Gestión de anuncios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá publicar anuncios. Estos anuncios contarán con diferentes campos descriptivos: Título, descripción, precio tasado, fotografías, qué se busca a cambio y categoría o subcategoría.
- Cuando el usuario publique un anuncio, quedará pendiente de ser aprobado por parte de un moderador antes de ser publicado.
- El usuario podrá marcar como vendido, o eliminar sus anuncios en cualquier momento.
- El usuario podrá editarlo, destacarlo o contratar el servicio para renovarlo automáticamente.

## 3.1.6. Requisito funcional 6

Número de requisito	6
---------------------	---

Nombre de requisito	Gestión de favoritos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá añadir o eliminar anuncios de sus favoritos.
- El usuario podrá visualizar una lista de los anuncios que ha marcado como favorito.

### 3.1.7. Requisito funcional 7

Número de requisito	7
Nombre de requisito	Visualización de un anuncio
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Cualquier usuario de la aplicación podrá visualizar los anuncios publicados por otros usuarios, independientemente de si está logueado o no.
- El usuario podrá ver todos los datos descriptivos del anuncio.
- Además podrá ver datos adicionales como el número de veces que ha sido marcado como favorito.
- También podrá ver un pequeño resumen de los datos del usuario que ha publicado el anuncio, mostrando por ejemplo valoraciones o si el perfil está verificado.
- El usuario podrá iniciar una conversación con el usuario que ha publicado el anuncio directamente desde el mismo.
- El usuario podrá compartir cualquier anuncio en las principales redes sociales.

### 3.1.8. Requisito funcional 8

Número de requisito	8
Nombre de requisito	Visualización del perfil de usuario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Un usuario podrá ver su propio perfil y el de otros usuarios.
- En el perfil de usuario se mostrarán los anuncios publicados por el usuario, las valoraciones, y algunos datos del usuario como su nombre y foto de perfil. En ningún caso se mostrarán datos de contacto.

### 3.1.9. Requisito funcional 9

Número de requisito	9
Nombre de requisito	Listado de anuncios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- En la página principal se podrá ver una lista de los últimos anuncios publicados cerca de él.
- Además, se podrá navegar por las diferentes categorías y subcategorías y visualizar la lista de los anuncios pertenecientes a dicha categoría.
- Al hacer click en un anuncio de la lista, iremos al detalle del anuncio (Requisito funcional 7).

### 3.1.10. Requisito funcional 10

Número de requisito	10
Nombre de requisito	Búsqueda y ordenación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción

Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- La aplicación dispondrá de una barra de búsqueda desde la que el usuario podrá buscar anuncios por palabras clave.
- Además, podrá filtrar dicha búsqueda por categorías, subcategorías, precio mínimo, precio máximo.
- Los campos de búsqueda textuales ofrecerán predicciones de texto, de forma que conforme el usuario va escribiendo, se le sugiera en un desplegable los términos más cercanos existentes.
- La lista de anuncios resultantes se podrá ordenar ascendente o descendientemente según diferentes criterios: fecha, precio, proximidad, etc.

### 3.1.11. Requisito funcional 11

Número de requisito	11
Nombre de requisito	Mensajería instantánea
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá comunicarse mediante mensajería instantánea con otros usuarios.
- La aplicación dispondrá de un apartado de “Mis chats” donde el usuario visualizará una lista de todas sus conversaciones abiertas (iniciadas).
- Para iniciar una conversación, se realizará a través de un anuncio concreto. Es decir, el objeto de la conversación siempre será un producto determinado.
- Cuando un usuario reciba un mensaje mediante el chat, se le notificará a través la interfaz.
- Desde esta pantalla, el usuario podrá eliminar cualquier conversación de la lista.
- Cuando el usuario haga click en una conversación de su lista de chats, se abrirá la pantalla del detalle de la conversación.
- Desde esta pantalla, el usuario podrá visualizar toda los mensajes enviados y recibidos desde que se inició la conversación, así como la fecha y la hora en la que cada mensaje fue enviado.
- Esta página también permitirá al usuario enviar un mensaje mediante un área de texto.

### 3.1.12. Requisito funcional 12

Número de requisito	12
Nombre de requisito	Imágenes en una conversación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá enviar imágenes desde la pantalla de la conversación (pantalla de detalle de un chat).

### 3.1.13. Requisito funcional 13

Número de requisito	13
Nombre de requisito	Valoración de usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Los usuarios podrán valorarse tras realizar un trueque. De esta forma evitamos valoraciones falsas ya que no se podrá valorar sin que el anuncio haya sido marcado como “trueque realizado”.
- Uno de los dos usuarios, una vez marque su anuncio como “trueque realizado” podrá seleccionar al usuario con el que ha hecho el intercambio para valorarlo.

- Si un usuario ha sido valorado, esta valoración no será visible ni para él ni para el resto de usuarios hasta que éste lo valore también.
- El usuario podrá valorar a otro usuario con una puntuación del 1 al 5 y un breve comentario.

#### 3.1.14. Requisito funcional 14

Número de requisito	14
Nombre de requisito	Reporte de anuncios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Un usuario podrá reportar un anuncio si considera que contiene contenido inapropiado.
- Esta funcionalidad estará disponible desde la página de detalle del anuncio.

#### 3.1.15. Requisito funcional 15

Número de requisito	15
Nombre de requisito	Reportar usuario
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Un usuario podrá reportar a otro si considera que está comportándose de forma inadecuada.

#### 3.1.16. Requisito funcional 16

Número de requisito	16
Nombre de requisito	Sugerencias de anuncios de interés (Truequinder)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario tendrá dos formas de obtener sugerencias de anuncios de interés: una accesible desde una sección del menú de navegación y otra accesible desde cada uno de sus anuncios.
- Cuando se acceda desde la propia sección del menú, las sugerencias variarán en función del precio, la proximidad y las categorías de interés presentes en los anuncios que tiene publicados el usuario.
- Cuando se acceda desde un anuncio concreto, las sugerencias variarán en función del precio, la proximidad y la categoría de interés presente en el anuncio desde el cual ha accedido.
- Para cada anuncio que se le sugiera al usuario, éste tendrá tres opciones: marcarlo como favorito, abrir un chat de forma automática con un mensaje predeterminado indicando que está interesado en el producto o descartar el producto si no está interesado en él.
- Si el usuario pulsa en la opción del chat, le aparecerá una ventana emergente indicando que su mensaje ha sido enviado. Al cerrar este mensaje, le seguirán apareciendo nuevas sugerencias.
- Cuando un usuario elija una de esas tres opciones, se mostrará un nuevo anuncio sugerido, y así sucesivamente.

#### 3.1.17. Requisito funcional 17

Número de requisito	17
Nombre de requisito	Buzón de sugerencias/Contacto

Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- La plataforma dispondrá de un formulario desde el cual los usuarios podrán enviar sus dudas o sugerencias a los administradores.

### 3.1.18. Requisito funcional 18

Número de requisito	18
Nombre de requisito	Correos personalizados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Newsletter
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario recibirá de forma semanal un correo electrónico basado en sus preferencias, con anuncios en los que posiblemente esté interesado.

### 3.1.19. Requisito funcional 19

Número de requisito	19
Nombre de requisito	Cursos de formación ( <i>eLearning</i> )
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación / Web
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El usuario podrá realizar un curso formativo sobre la economía del trueque, en el que aprenderá acerca del consumo responsable así como de las mejores técnicas para publicar su anuncio en plataformas de este tipo y cómo interactuar con los demás usuarios para realizar el intercambio.
- Cuando el usuario complete el curso, se le recompensará con una insignia (badge) que será visible en su perfil y que generará valor para los demás usuarios.

### 3.1.20. Requisito funcional 20

Número de requisito	20
Nombre de requisito	Autenticación y acceso al backoffice
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- La plataforma dispondrá de una aplicación web que contendrá las herramientas de gestión de la misma. Se accederá con una URL distinta.
- Los administradores con accesos al backoffice podrán loguearse con usuario y contraseña.
- Si los datos introducidos por el usuario no son correctos se mostrará un mensaje por pantalla indicándolo.
- También podrá desloguearse en cualquier momento siempre que esté previamente logueado.
- Las contraseñas de los usuarios se almacenan cifradas en la base de datos.

### 3.1.21. Requisito funcional 21

Número de requisito	21
Nombre de requisito	Moderación de anuncios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice

Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
-------------------------	--	--	---

- Cuando el usuario publique un anuncio, quedará pendiente de ser aprobado por parte de un moderador antes de ser publicado.
- Desde el panel de administrador, los usuarios autorizados podrán visualizar la lista de anuncios pendientes de moderación y publicarlos o rechazarlos si su contenido es inapropiado.
- Cuando un anuncio sea rechazado, se notificará automáticamente al usuario que lo ha publicado mediante un correo electrónico.
- Los anuncios rechazados no se eliminarán de la BD hasta dos años después de ser moderados.
- Además, el administrador podrá visualizar un listado de los anuncios rechazados que aún permanecen en la BD.

### 3.1.22. Requisito funcional 22

Número de requisito	22
Nombre de requisito	Lista de anuncios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El administrador podrá visualizar un listado de todos los anuncios publicados actualmente en la plataforma.
- Además, esta lista se podrá filtrar en función de la categoría o subcategoría, palabras clave (búsqueda), precio, usuario, id de anuncio, etc.
- La lista de anuncios resultantes se podrá ordenar ascendente o descendientemente según diferentes criterios: fecha, precio, etc.

### 3.1.23. Requisito funcional 23

Número de requisito	23
Nombre de requisito	Gestión de reportes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El administrador podrá visualizar una lista de todos los reportes que han realizado los usuarios.
- Para cada reporte, la plataforma permitirá tratarlo. Si el reporte es veraz, se añadirá al usuario automáticamente a la lista de baneados.
- Un usuario que esté en la lista de baneados no podrá publicar anuncios. Cuando intente publicar un anuncio, se le informará de que no puede hacerlo porque está baneado.

### 3.1.24. Requisito funcional 24

Número de requisito	24
Nombre de requisito	Gestión de categorías
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El administrador podrá visualizar una lista de las categorías de la web y sus subcategorías asociadas.
- Además, podrá crear categorías y subcategorías nuevas, así como editar categorías existentes.
- También podrá eliminar categorías y pasar los anuncios de la categoría eliminada a otra.

## 3.1.25. Requisito funcional 25

Número de requisito	25
Nombre de requisito	Gestión de correos informativos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El administrador podrá enviar correos electrónicos generales a todos los usuarios desde el panel de administrador.
- Para ello, dispondrá de un formulario en el que podrá introducir los diferentes campos del correo como el asunto, el cuerpo del mensaje y una imagen asociada.

## 3.1.26. Requisito funcional 26

Número de requisito	26
Nombre de requisito	Gestión del buzón de sugerencias
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- El administrador dispondrá de una sección en la que podrá visualizar una lista con los mensajes enviados por los usuarios al buzón de sugerencias.
- Además, podrá responder cada uno de estos mensajes directamente desde el backoffice, enviándose un email al usuario.

## 3.1.27. Requisito funcional 27

Número de requisito	27
Nombre de requisito	Mapa de la aplicación (TAG)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Desde el backoffice, el administrador podrá visualizar un gráfico en forma de mapa 3D que ofrecerá información sobre el lugar del que provienen los usuarios registrados o los anuncios publicados en la aplicación.
- Además, este mapa será interactivo y permitirá diferentes funcionalidades como alejarlo, acercarlo, rotarlo, etc.

## 3.1.28. Requisito funcional 28

Número de requisito	28
Nombre de requisito	Métricas de la aplicación (NM)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Desde el backoffice, el usuario máster podrá visualizar las métricas de la aplicación. Esto le permitirá estudiar cómo optimizar la plataforma y ofrecer un mejor servicio a los usuarios.
- Algunas de estas métricas serán: los usuarios registrados, la media de anuncios publicados por usuario, la media de anuncios publicados en un periodo de tiempo, etc.

## 3.1.29. Requisito funcional 29

Número de requisito	29
Nombre de requisito	Gestión de SEO
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Backoffice
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

- Desde el backoffice, el administrador podrá gestionar el SEO de la plataforma mediante la introducción de varios parámetros como: *keywords*, *link write* y URLs amigables.
- De esta manera, la plataforma mejorará su indexación en Google.

## 3.2. Requerimientos no funcionales.

## 3.2.1. Requisitos de rendimiento

## 3.2.1.1. Arquitectura

La implementación de la aplicación deberá seguir una arquitectura API Rest MVC.

El servidor deberá ser capaz de atender peticiones por parte de los clientes concurrentemente.

## 3.2.1.2. Número de usuarios

La aplicación deberá garantizar el acceso simultáneo a un número elevado de usuarios. Además, deberá soportar un mínimo de 30 transacciones por segundo.

## 3.2.2. Seguridad

## 3.2.2.1. Protección de los datos/recursos

La API sólo responderá peticiones procedentes de nuestras aplicaciones. Para ello utilizaremos un sistema de tokens.

Además, los recursos privados tales como las imágenes enviadas a través del chat utilizarán un token persistente (que será guardado en la BD). La API rest será la encargada de identificar el token y devolver el recurso asociado a él.

Por último, las contraseñas de los usuarios se almacenarán en la BD encriptadas.

## 3.2.2.2. Protocolo seguro

Todos los recursos de la web se servirán mediante HTTPS, siguiendo el protocolo seguro SSL.

## 3.2.2.3. Persistencia de datos

Cuando un anuncio, un usuario o un chat es eliminado, permanecerán en nuestra base de datos al menos durante dos años. De esta forma, cumpliremos con la normativa legal de persistencia de datos.

Se realizarán copias de seguridad de forma externa para garantizar la persistencia de los datos. Se realizará una copia total el primer fin de semana



del mes y una copia incremental de primer orden el primer lunes de cada mes. Las copias de segundo orden se realizarán el resto de días del mes.

#### 3.2.2.4. Logs de actividad

Todas las actividades de usuarios y administradores serán almacenadas mediante logs de actividad. Esto nos permitirá conocer qué hacen los usuarios en nuestra web, qué les interesa y qué no, además de detectar puntos muertos y comportamientos sospechosos. Para ello, utilizaremos el estándar y la librería Syslog.

#### 3.2.2.5. Autenticación

El sistema validará la identidad del usuario de diferentes formas. Si inicia sesión mediante redes sociales, comprobaremos que el usuario es quien dice ser mediante el protocolo OAUTH. También tendrá la opción de acceder mediante un usuario y contraseña únicos para la aplicación. La contraseña se almacenará cifrada en la BD.

Cuando el usuario se autentique, se le proporcionará un token de sesión de un solo uso que caducará después de determinado tiempo. Se usarán cookies (web) y localStorage (app) para guardar la sesión abierta en una máquina y no tener que identificarse una y otra vez.

#### 3.2.3. Fiabilidad

Utilizaremos las copias de seguridad para garantizar que el sistema funciona correctamente.

#### 3.2.4. Accesibilidad

La plataforma será compatible con los principales sistemas operativos y navegadores.

Además, seguirá el modelo mobile first, sin descuidar el diseño web, ya que los usuarios accederán tanto desde el PC como desde dispositivos móviles.

#### 3.2.5. Disponibilidad

La plataforma estará disponible y funcionando en cualquier momento desde cualquier lugar.

La web será accesible mediante un dominio y la app móvil estará disponible para su descarga tanto en la Play Store como en la App Store.

#### 3.2.6. BDD

La base de datos utilizada será relacional de tipo MySQL. El esquema de la base de datos se encuentra en las referencias del documento.

#### 3.2.7. Analíticas

El sistema utilizará los logs de actividad así como Google Analytics para obtener información de la aplicación en tiempo real, como por ejemplo el número de usuarios activos o el dispositivo desde el cual están accediendo a la plataforma.

Además, como ya hemos comentado antes, generaremos nuestras propias métricas que estarán visibles desde el Backoffice.

### 3.2.8. Requisitos legales

Cumplir con los aspectos legales del país donde se va a prestar servicio, en este caso España. Esto implica que el servicio no es mundial, por lo que habrá que registrar la BD y las normas de seguridad en la agencia española y además aplicar las condiciones que ésta imponga.

## 4. Apéndices

### 4.1. Definiciones, acrónimos y abreviaturas

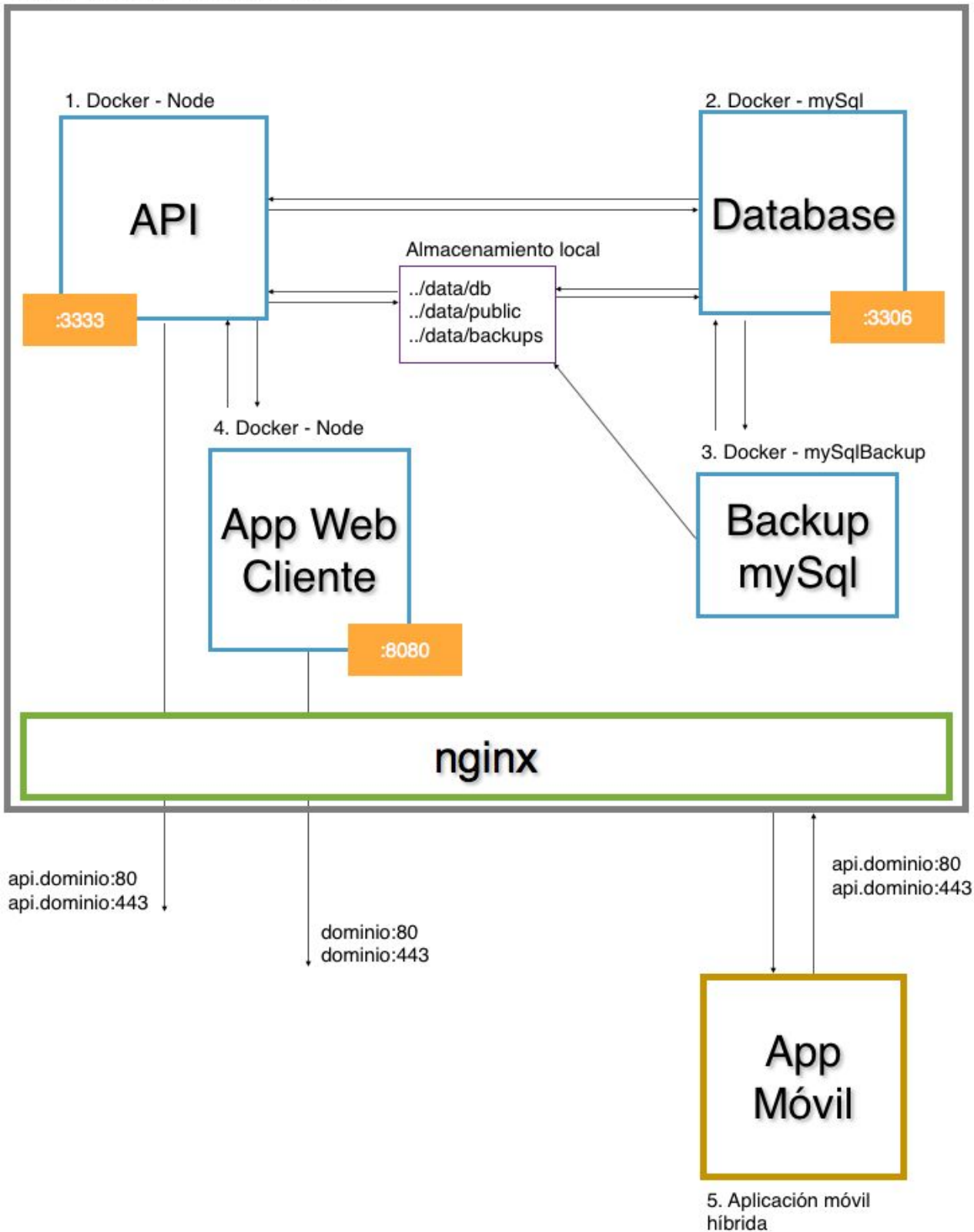
Nombre	Definición
BD	Base de datos. Bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.
SGBD	Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.
MySQL	MySQL es un sistema de gestión de base de datos relacional de código abierto, basado en lenguaje de consulta estructurado (SQL).
HTML	HiperText Markup Language (Lenguaje de Marcación de Hipertexto) es un lenguaje es se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes.
CSS	Cascading Style Sheets es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML.
Mobile first	Concepto que nace de el actual cambio en la manera de navegar de las personas, y consiste en diseñar para los terminales móviles, y luego adaptarlo a las pantallas de ordenador.
API	Una API (siglas de 'Application Programming Interface') es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.
REST	La Transferencia de Estado Representacional (Representational State Transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.
OAUTH	OAuth (Open Authorization) es un protocolo que permite flujos simples de autorización para sitios web o aplicaciones informáticas. Se trata de un protocolo propuesto por Blaine Cook y Chris Messina, que permite autorización segura de una API de modo estándar y simple para aplicaciones de escritorio, móviles y web.
RRSS	Siglas de Redes Sociales. Página web en la que los internautas intercambian información personal y contenidos multimedia de modo que crean una comunidad de amigos virtual e interactiva.

MVC	El modelo–vista–controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
HTTPS	Hypertext Transfer Protocol Secure (ó HTTPS) es una combinación del protocolo HTTP y protocolos criptográficos. Se emplea para lograr conexiones más seguras en la World Wide Web (WWW).
SSL	Secure Sockets Layer es un protocolo diseñado para permitir que las aplicaciones para transmitir información de ida y de manera segura hacia atrás.

4.2. Referencias

4.2.1. Arquitectura del servidor

0. OVH VPS SSD - 4GB RAM - 20GB



## 0. Servidor

El servidor será una instancia VPS con disco SSD de 20GB en OVH (proveedor). En el servidor se compondrá de los siguientes servicios:

- Sistema Operativo: Ubuntu 14.04.
- Docker.
- Jira para la gestión del proyecto.
- Servidor SSH.
- nginx

Mediante nginx usando un *reverse-proxy* enrutaremos las aplicaciones al exterior. Se tratará de implementar un certificado digital para usar *HTTPS*.

## 1. API

En el API se dispondrá de una instancia de Docker con una imagen de Node. El puerto interno será **3333**, el puerto externo expuesto será el **80** apuntando a un dominio del estilo **api.dominio:80/443**.

Los ficheros de la aplicación estáticos (imágenes de usuario, de los anuncios, etc) se guardarán en una carpeta en la máquina Host, es decir fuera del contenedor.

## 2. Base de datos

Para la base de datos usaremos una instancia de Docker corriendo una imagen de mySql. Al tratarse Docker de un sistema de imágenes stateless, los datos no persisten al reiniciar los contenedores, es por esto que los datos de la base de datos se guardarán también fuera del contenedor.

## 3. Copia de la base de datos

Usaremos un contenedor específico con una pequeña aplicación para hacer las copias de seguridad de la base de datos que se guardarán también en la máquina Host.

## 4. Aplicación cliente web

Dentro de otro container meteremos la aplicación web cliente. Está aplicación expondrá el puerto **8080** internamente y externamente el puerto **80** usando la raíz del dominio (**dominio:80**).

## 5. Aplicación móvil híbrida

Por último, y fuera del servidor, tendremos una aplicación móvil híbrida que se conectará al servidor mediante la **API**.

### 4.2.2. Definición de la API Rest

## Anuncios

- GET **/advert** - Obtener todos los anuncios ordenados por actualización.
  - **Parámetros**
    - lat=...&long=...: Anuncios ordenados por cercanía.
    - cat: Filtrar por categoría.
    - search: Búsqueda en título y descripción.
    - price\_min & price\_max: Rango de precios.
    - order: Ordenación.
    - user: Productos del usuario.
  - POST **/advert** - Crear un anuncio.
  - GET **/advert/{id-advert}** - Obtener un anuncio con id.
  - PUT **/advert/{id-advert}** - Actualizar un anuncio.
  - DELETE **/advert/{id-advert}** - Eliminar un anuncio.
  - GET **/advert/{id-advert}/fav** - Obtener todos los usuarios que han marcado como favorito el anuncio.
  - GET **/advert/{id-advert}/report** - Obtener todos los reportes del anuncio.
  - POST **/advert/{id-advert}/report** - Crear un nuevo reporte.
  - GET **/advert/{id-advert}/report/{id-report}** - Obtiene un reporte.
  - PUT **/advert/{id-advert}/report/{id-reprot}** - Actualizar el estado del reporte.
  - DELETE **/advert/{id-advert}/report/{id-reprot}** - Eliminar reporte.
  - DELETE **/advert/{id-advert}/suggestion** - Obtiene todas las sugerencias para un determinado anuncio.
- 

## Usuarios

- GET **/user** - Obtiene todos los usuarios.
- POST **/user** - Crea un nuevo usuario.
- GET **/user/{id-user}** - Obtener datos de un usuario.
- PUT **/user/{id-user}** - Actualizar un usuario.
- DELETE **/user/{id-user}** - Eliminar un usuario.
- GET **/user/{id-user}/advert** - Obtener los anuncios del usuario.
- DELETE **/user/{id-user}/suggestion** - Obtiene todas las sugerencias para un determinado usuario.
- GET **/user/{id-from}/relationship** - Obtiene la relación con otros usuarios.
- POST **/user/{id-from}/relationship** - Crea una relación con otro usuario.
- GET **/user/{id-from}/relationship/{id-to}** - Obtiene la relación con otro usuario.

- PUT **/user/{id-from}/relationship/{id-to}** - Actualiza una relación con otro usuario.
  - DELETE **/user/{id-from}/relationship/{id-to}** - Elimina la relación entre dos usuarios.
  - GET **/user/{id-user}/review** - Obtiene las valoraciones de un usuario.
  - POST **/user/{id-user}/review** - Crea una nueva valoración al usuario.
  - GET **/user/{id-user}/review/{id-review}** - Obtiene una valoración de usuario.
  - PUT **/user/{id-user}/review/{id-review}** - Actualiza una valoración de usuario.
  - DELETE **/user/{id-user}/review/{id-review}** - Elimina una valoración de usuario.
  - GET **/user/{id-user}/fav** - Obtener todos los favoritos del usuario.
  - POST **/user/{id-user}/fav** - Añadir un nuevo favorito.
  - GET **/user/{id-user}/fav/{id-fav}** - Obtener el favorito.
  - DELETE **/user/{id-user}/fav/{id-fav}** - Eliminar el favorito.
- 

## Conversaciones

- GET **/user/{id-user}/chat** - Obtener todas las conversaciones del usuario.
  - POST **/user/{id-user}/chat** - Crear una nueva conversación.
  - GET **/user/{id-user}/chat/{id-chat}** - Obtener el chat.
  - PUT **/user/{id-user}/chat/{id-chat}** - Actualiza un chat.
  - DELETE **/user/{id-user}/chat/{id-chat}** - Borrar chat.
  - GET **/chat/{id-chat}/message** - Obtener todos los mensajes del chat.
  - POST **/chat/{id-chat}/message** - Crear un nuevo mensaje.
- 

## Categorías

- GET **/category** - Obtener todas las categorías.
- POST **/category** - Crear una nueva categoría.
- GET **/category/{id-category}** - Obtener una categoría.
- PUT **/category/{id-category}** - Actualizar una categoría.
- DELETE **/category/{id-category}** - Eliminar una categoría.
- GET **/category/{id-category}/subcategory** - Obtener todas las subcategorías de una categoría.
- POST **/category/{id-category}/subcategory** - Crear una nueva subcategoría.
- GET **/category/{id-category}/subcategory/{id-subcat}** - Obtener una subcategoría.
- PUT **/category/{id-category}/subcategory/{id-subcat}** - Actualizar una subcategoría.

- DELETE /category/{id-category}/subcategory/{id-subcat} - Eliminar una subcategoría.

#### 4.2.3. Esquema de la BD

