



## **Caso de estudio: UrbanSpot**

**Objetivo:** El proyecto **UrbanSpot** consiste en crear una aplicación web colaborativa que permita a los usuarios descubrir, documentar y valorar **puntos de interés urbano (POIs)** en ciudades.

Los usuarios pueden **añadir nuevos POIs**, subiendo una fotografía y una descripción, mientras que otros usuarios pueden **visualizarlos, añadir nuevas fotos y valorarlos** con las fotos subidas por los usuarios.

El sistema asignará puntos a los usuarios en función de sus aportaciones, de manera que se incentive la participación mediante **mecánicas de gamificación**.

Los usuarios acumularán puntos en dos roles principales:

- **Exploradores (guías):** por descubrir y describir nuevos POIs.
- **Fotógrafos urbanos:** por añadir fotografías de calidad a POIs existentes.

A continuación, se presenta una lista informal de los principales requisitos de la aplicación:

### **A. Almacenamiento.**

El almacenamiento de la información (usuarios, puntos de interés, fotos, valoraciones y puntuaciones acumuladas) se realizará en una **base de datos noSQL** (por ejemplo, MongoDB Atlas, Firestore o Datastore de Google App Engine).

Las fotografías se guardarán en un sistema de almacenamiento de objetos (por ejemplo, Cloud Storage, S3 o Firebase Storage), manteniendo en la base de datos únicamente las URLs y metadatos asociados.

### **B. Identificación de usuarios.**

Se utilizará algún sistema externo de **identificación segura basado en OAuth 2.0** (por ejemplo, Google OAuth o Firebase Auth), de manera que el acceso y las interacciones con la aplicación se realicen de forma autenticada.

Cada usuario dispondrá de un perfil donde se mostrará su nombre, puntuación total (por POIs y por fotos) y contribuciones realizadas.

### **C. Gestión de puntos de interés (POIs).**

Los usuarios autenticados podrán **crear nuevos POIs**, indicando al menos:

- Nombre y descripción.
- Etiquetas (p. ej., movilidad, energía, sostenibilidad, cultura, turismo, etc.) (opcional).
- Localización geográfica (latitud y longitud).
- Imagen inicial.

Cada POI incluirá su **autor, número de valoraciones** y su **valoración media** calculada a partir de las puntuaciones recibidas por otros usuarios.

Los POIs creados serán visibles para todos los usuarios en el mapa de la aplicación.

### **D. Fotografías asociadas a los POIs.**

Los usuarios podrán **subir nuevas fotografías** a cualquier POI existente.

Cada foto almacenará:

- Autor.
- Fecha de subida.



## Desarrollo de Aplicaciones en la Nube

- Descripción breve (opcional).
- Número de valoraciones.
- Valoración media (resultado de las puntuaciones de otros usuarios).

Las fotografías se mostrarán en una galería asociada al POI correspondiente.

### E. Valoraciones.

Cualquier usuario autenticado podrá **valorar** tanto un POI como una fotografía mediante una puntuación numérica entre **0 y 10**.

Cada elemento mantendrá el número de valoraciones y su valoración media, que se recalculará automáticamente cada vez que se registre un nuevo voto.

Un usuario no podrá valorar sus propias contribuciones ni votar más de una vez el mismo elemento.

### F. Sistema de puntos y reputación.

El sistema otorgará **puntos de gamificación** según las acciones realizadas por cada usuario.

Por ejemplo:

- Crear un nuevo POI validado: **+20 puntos**.
- Recibir valoración media superior a 7 en un POI propio: **+10 puntos**.
- Subir una nueva foto a un POI existente: **+5 puntos**.
- Recibir valoración media superior a 7 en una foto propia: **+10 puntos**.
- Valorar POIs o fotos de otros usuarios: **+1 punto**.

La puntuación total determinará la **reputación del usuario**, y se mostrará un **ranking global** de los participantes más activos.

### G. Visualización de mapas e imágenes.

La aplicación mostrará todos los POIs sobre un **mapa urbano interactivo** (por ejemplo, OpenStreetMaps o Google Maps), utilizando marcadores con iconos diferenciados según la etiqueta del POI.

Al seleccionar un marcador, se visualizarán la descripción, las fotos y la valoración media, así como los botones para valorar o añadir nuevas imágenes.

El sistema se encargará además de gestionar el **almacenamiento y la visualización** de todas las imágenes subidas por los usuarios.

### H. Despliegue [opcional].

La aplicación podrá desplegarse en un entorno **PaaS (Platform as a Service)**, como Heroku, Fly.io, Vercel, Firebase o App Engine.

Deberá configurarse la conexión con la base de datos y el almacenamiento de imágenes, así como las variables de entorno necesarias para la autenticación y la seguridad.

---

## Entrega

El caso de estudio se entregará a través del campus virtual mediante un archivo comprimido que contenga:

- a. **El código completo** de la aplicación desarrollada (frontend y backend).
- b. **Una memoria técnica** que incluya:



## Desarrollo de Aplicaciones en la Nube

- Tecnologías empleadas (lenguajes, frameworks, servicios cloud, etc.).
- Descripción técnica de la aplicación web y diseño de la base de datos.
- Funcionalidad implementada.
- Descripción de la API desarrollada y las URL de las APIs externas utilizadas.
- Instrucciones de instalación y despliegue (si son necesarias).
- Limitaciones de la solución y posibles mejoras futuras.

c. **La URL de despliegue** de la aplicación (por ejemplo: <https://idproyecto.vercel.app>) y los nombres y códigos de los participantes del grupo.