

DESCRIPCIÓN DEL PROYECTO

Se desea crear una aplicación para que cualquier comercio registrado pueda subir su propia Web con contenido (fotos, texto, etc.), de manera que pueda visualizarse cuando los usuarios busquen por los comercios de una ciudad y de una actividad concreta.

ALCANCE DEL PROYECTO

Se creará un front-end implementando toda la funcionalidad propuesta a continuación.
(El back-end se implementará en el 2Q)

IMPLEMENTACIÓN TÉCNICA

Para el desarrollo del front-end se utilizará NextJS con llamadas al backend con Fetch.

DETALLES

Habrá cuatro tipos de cliente: admins, comercios, usuarios anónimos y usuarios registrados.

Desde una página de back-office, **interfaz para los admins**, se podrá registrar un comercio (por un admin) enviando la siguiente información:

- Nombre del comercio.
- CIF
- Dirección
- E-mail
- Teléfono de contacto

Crear página de login de admins y crear el formulario anterior para dar de alta comercios que, de momento, lo guardaremos en disco local del servidor (véase nota 4 de Requisitos técnicos, al final de este documento). Pero implementar petición POST con fetch para el envío al futuro backend.

Además, desde el interfaz de admins (back-office), podrá consultarse los comercios filtrando por nombre y/o ciudad. También podrá eliminarse un comercio.

Una vez registrado, al comercio se le facilitará acceso con un [id] de página con los cuales podrá subir su contenido (en el 2Q veremos autenticación y JWT) desde su **interfaz de comercio**:

- Ciudad
- Actividad
- Título
- Resumen
- [Textos]
- [Fotos]
- Datos no modificables por el comercio
 - Scoring
 - Número de puntuaciones
 - Reseñas

Crear página de login de comercios y crear formulario que, de momento, lo dejaremos en disco local del servidor. Pero implementar petición POST con fetch para el envío al futuro backend.

Además, desde su interfaz, el comercio podrá modificar su datos, subir texto y fotos (solo el nombre del fichero), ver usuarios de su ciudad interesados en su actividad. Y también podrá borrar su propia página.

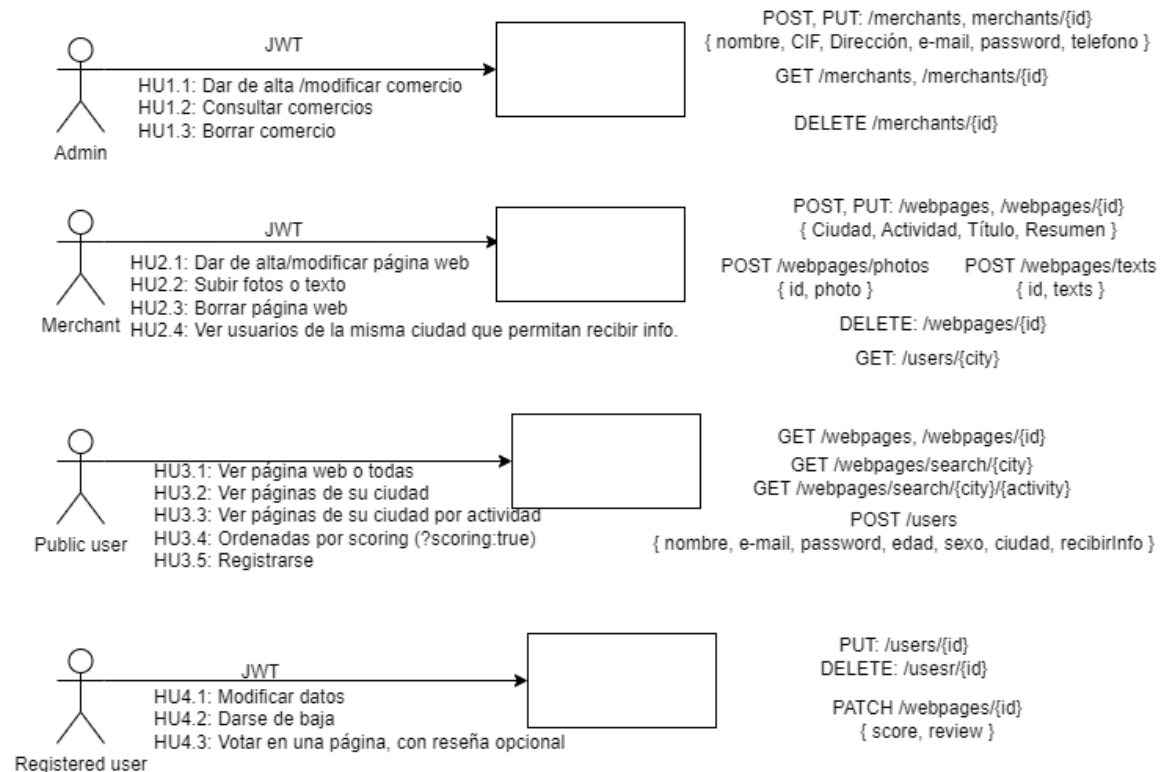
El usuario puede ser anónimo (consultas) o estar registrado para recibir ofertas de su ciudad. Desde su **interfaz de usuario** podrá registrarse tal que:

- Nombre
- E-mail
- Password
- Edad
- Ciudad
- Intereses
- PermiteRecibirOfertas

Crear formulario de registro de usuario que, de momento, lo dejaremos en el disco local del servidor. Pero implementar petición POST con fetch para el envío al futuro backend.

Desde el interfaz de usuario podrá realizar consultas de comercios filtrados por ciudad, actividad, etc. Y si el usuario está registrado, permitirá puntuar al comercio y dejar una reseña.

HISTORIAS DE USUARIO



REQUISITOS FUNCIONALES

- Admins:
 - Acceso a la página de administración mediante login y password
 - Es el único perfil que puede registrar un comercio. Una vez registrado le deberá facilitar credenciales.
 - Pueden consultar y filtrar comercios por nombre y ciudad
 - Pueden eliminar comercios
- Comercios
 - Únicamente pueden subir y actualizar contenido en su propia página. Para ello deberán hacer uso de credenciales.
 - No pueden modificar el Scoring, la cantidad de votaciones ni las reseñas.
 - Pueden darse de baja y borrar su propia página.
 - Pueden consultar los intereses de los usuarios de su ciudad para enviarles un correo. Para esto, dada una consulta, recibirá los mails de todos los usuarios de su ciudad interesados en un tema concreto y que tengan el campo "permitirRecibirOfertas" a true.

- Usuario anónimo
 - Puede consultar:
 - los comercios de una ciudad,
 - los comercios de una ciudad y de una actividad.
 - un comercio concreto a partir de su identificador
 - Se puede registrar (deja de ser anónimo)
- Usuario registrado
 - Puede actualizar sus datos (ciudad, intereses)
 - Puede activar y desactivar el flag `recibirOfertas`
 - Puede escribir reseñas en un comercio
 - Puede darse de baja

INTERFACES (páginas)

- Administrador:
 - página de login
 - alta de comercios
 - búsqueda de comercios
- Comercios
 - página de login
 - consulta de datos de usuarios (que han dado su aprobación)
 - modificación de datos de la página de su comercio
- Usuarios no registrados
 - búsqueda de comercios
 - información de un comercio
- usuarios registrados
 - página de login
 - página de preferencias
 - búsqueda de comercios
 - página de comercio con edición de reseñas y votos

REQUISITOS TÉCNICOS

- 1.- Se requiere que la Web sea responsive, y el diseño es abierto (custom css, bootstrap, tailwind, ...).
- 2.- Puedes crear las páginas y funcionalidades con Next que requieras.
- 3.- Se valorarán muy positivamente mejoras funcionales y/o técnicas a las propuestas.
- 4.- Hasta que programemos un backend en el 2Q, almacenaremos los datos en el disco local del servidor. Ejemplo:

```
import {NextResponse} from 'next/server'
import { readFileSync, writeFileSync} from 'fs';

export async function POST(request) {
  const data = await request.json()
  try{ // Read user.txt from disk and concatenate with data from request
    const users = JSON.parse(readFileSync("users.txt"))
    writeFileSync("users.txt", JSON.stringify([...users, data]))
  } catch(e){ // If user.txt file does not exist, create it with data from request
    writeFileSync("users.txt", JSON.stringify([data]))
  }
  return NextResponse.json({
    message: "Guardando datos..."
  })
}
```