



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Sistemas y Tecnologías Web

Anteproyecto – uTopia

Jacobo Labrador González

(alu0101119663@ull.edu.es)

Vlatko Jesús Marchán Sekulic

(alu0101321141@ull.edu.es)

Tanausú Falcón Casanova

(alu0101320878@ull.edu.es)



Índice

Índice	1
Introducción	2
Planificación de las pantallas	2
División de las pantallas en colecciones	7
Arquitectura de la aplicación	8
API	8
Elección de la arquitectura de aplicación	9
Referencias	9



Introducción

A lo largo del presente documento se expondrán las ideas propuestas para el desarrollo del proyecto que se realizará a lo largo de la asignatura Sistemas y Tecnologías Web. La idea principal será implementar una red social llamada uTopia, enfocada principalmente a los usuarios de dispositivos móviles siguiendo un diseño mobile-first. Dicha página tendrá una serie de funcionalidades, como un tablón general donde los usuarios podrán publicar sus opiniones y pensamientos de temas variados, configuración de perfil, etc.

Planificación de las pantallas

Para el diseño de la web uTopia se han propuesto en un principio seis páginas para la implementación de las funcionalidades de la aplicación:

- Una pantalla de inicio de sesión 1 y otra de registro 2, para que los usuarios puedan acceder a la aplicación.

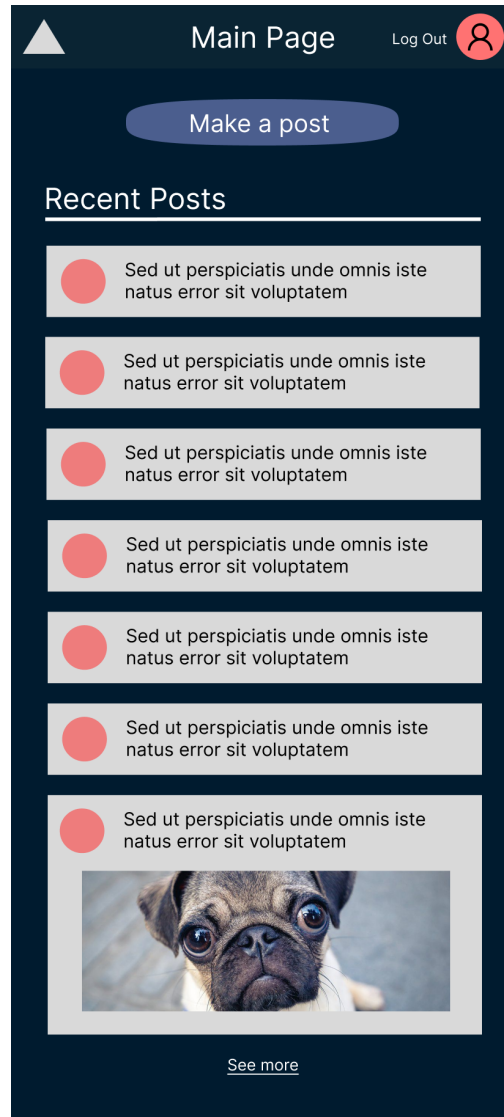
The image displays two side-by-side mobile application screens with a dark blue background. Each screen features a light blue rounded rectangle containing the form elements. At the top of each screen is a grey circular placeholder for a profile picture. The left screen, titled 'Log In', includes input fields for 'Username' and 'Password', a 'Sign in' button, a link 'Don't have an account?' with a 'Register' button below it, and a footer link 'Click here to know about us'. The right screen, titled 'Register', includes input fields for 'Username', 'Account name', 'Email', 'Password', and 'Confirm password', followed by a 'Register' button, and the same footer link.

Pantalla 1 y 2: Inicio de sesión y registro de usuarios

La información de la cuenta que se almacenará de los usuarios será su nombre de usuario, nombre de cuenta, correo electrónico y contraseña.



- Una pantalla principal 3, dónde se muestran los posts realizados por los usuarios.



Pantalla 3: Página principal



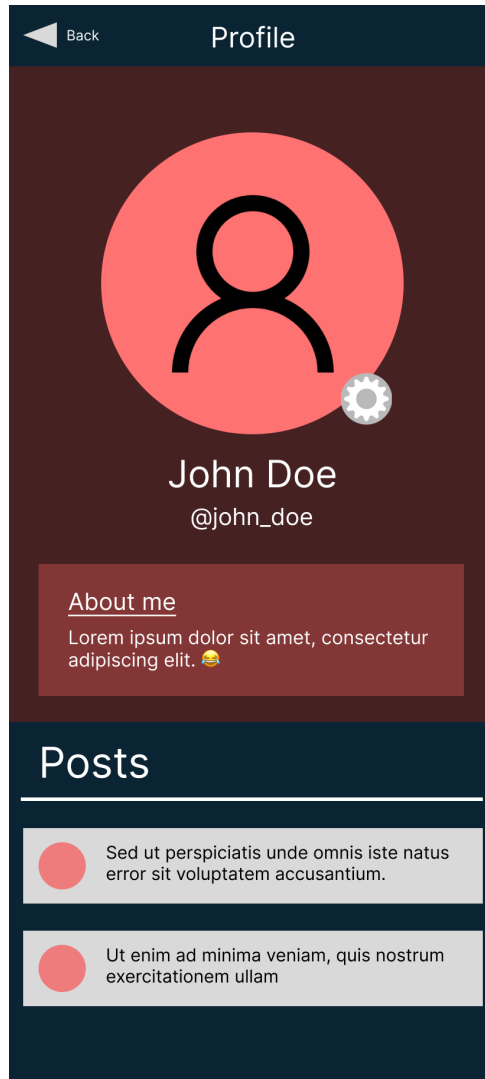
- Una pantalla de post 4, donde el usuario podrá escribir y publicar el contenido que desea.

The screenshot shows a mobile application interface for creating a post. At the top, there is a dark blue header bar with a white back arrow and the text 'Back' on the left, the title 'Posts' in the center, and 'Log Out' with a red user icon on the right. Below the header, the main content area is dark blue. A light gray rounded rectangle represents the post being created. Inside this rectangle, on the left, is a red circular profile picture icon with a white person silhouette, followed by the username '@john_doe'. Below the username is a text input field containing the placeholder text: 'Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.' At the bottom right of the text field, the character count '103/240' is displayed next to a small pencil icon. Below the gray rectangle, there are two buttons: a blue 'Post' button and a gray 'Cancel' button.

Pantalla 4: Publicar un post



- Una pantalla para el perfil del usuario 5, donde se podrá consultar la información pública del mismo, como su nombre de usuario, cuenta, descripción y posts realizados.



Pantalla 5: Perfil de usuario



- Una pantalla de configuración 6, donde el usuario podrá cambiar tanto su información pública de perfil, como su información privada de la cuenta.

Pantalla 6: Configuración de cuenta

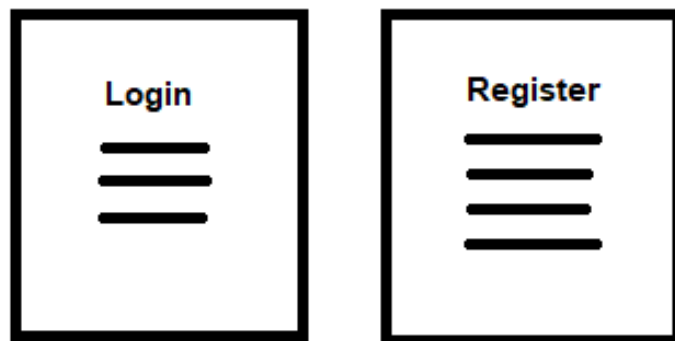


División de las pantallas en colecciones

En este apartado se realizará una clasificación de las pantallas en colecciones lógicas por su funcionalidad, para plasmar posteriormente la arquitectura de la aplicación.

En primer lugar, para la gestión de la autenticación de los usuarios se utilizan las pantallas de inicio y registro de sesión (1 y 2). Una vez que los usuarios estén autenticados, pasarán a la pantalla principal 3.

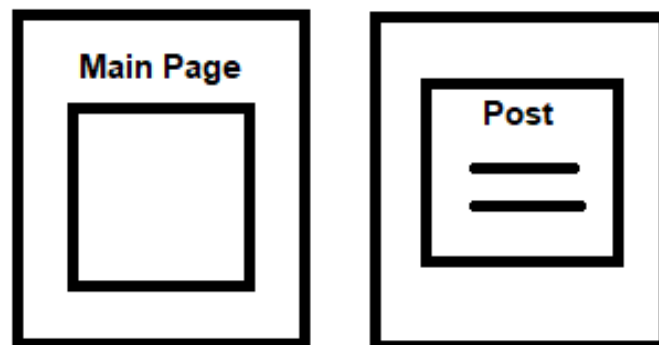
Autenticación:



Colección de pantallas de Autenticación

Desde la pantalla principal, los usuarios podrán acceder a la pantalla de post 4 mediante la opción de "Make a post". Desde aquí, podrán escribir cualquier contenido que deseen, con un máximo de 240 caracteres, y publicarlo con el botón de "Post" o cancelarlo con "Cancel", que devolverá a los usuarios a la pantalla principal.

Posts

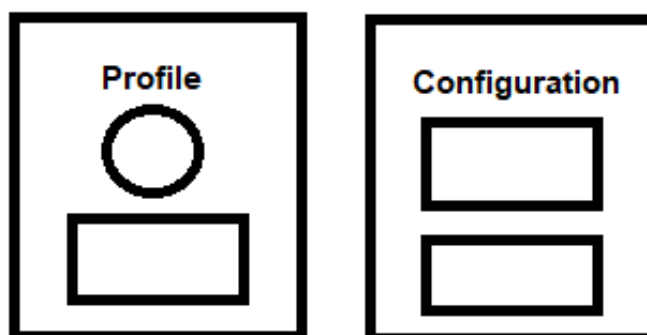


Colección de pantallas de Posts



Por último, desde la pantalla principal el usuario podrá seleccionar su icono para acceder a su perfil (pantalla 5), y desde aquí, acceder a la configuración de la cuenta (pantalla 6) con el icono del engranaje.

Configuración



Colección de pantallas de Configuración

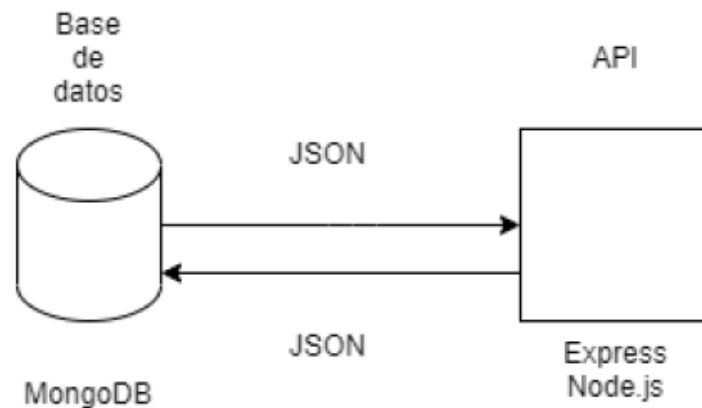
Arquitectura de la aplicación

La aplicación seguirá una arquitectura de páginas simples y funcionales para los usuarios, siguiendo un diseño correcto para realizar intercambios de datos e información, además de transacciones entre la base de datos y las diferentes páginas, todo esto mientras que los usuarios interactúan con la misma, consiguiendo de esta manera la autenticación y la persistencia de la información necesaria para cada consulta.

API

La interacción del usuario con la página se llevará a cabo usando una base de datos para almacenar los datos pertinentes y aportar información al mismo. Esto será esencial para empezar a diseñar la arquitectura con la que trabajaremos. Se dispondrá de una API REST construida con Node.js y Express que nos permitirá interactuar con la base de datos de MongoDB.

La arquitectura que se va a desarrollar se construirá una API REST con Express y Node.js para la comunicación con la base de datos de MongoDB.



Esquema estándar MERN API REST, usando MongoDB, Express, React y NodeJS.

Elección de la arquitectura de aplicación

Dentro de la arquitectura de la aplicación se dispondrá de un database con MongoDB caracterizada por ser no relacional y permitiendo unas consultas más rápidas que sus contrapartes relacionales. Esta base de datos se comunicará con la API basada en la tecnología de Express, la cual se encuentra ampliamente extendida por el mercado, y Node.js como rol de Proxy. Con lo que respecta al frontend de la aplicación, se realizará con Node.js y React debido a que ofrecen módulos que ayudan a la separación en componentes, y permiten la reutilización de los mismos. Además, se utilizará Typescript como lenguaje de desarrollo para utilizar su tipado de variables fuerte.

Referencias

[1]: Planificación de una aplicación web:

<https://selftaughtcoders.com/plan-web-application/>

[2]: Planificación preliminar del stack software:

<https://www.sitepoint.com/planning-mean-stack-application/>