

TRABAJO PRÁCTICO - GRUPO 4

Prototipado



Universidad General Sarmiento, Ingeniería de Software, 1S 2025, COM 02

Integrantes:

- Maximiliano Sebastian Barrionuevo
- Ciro Martín López
- Juan Manuel Galante

Profesores:

- Daniel Alvarez.
- Carlos Nelson Jimenez.
-

Fecha de Entrega:

- 21/05/2025.

Introduccion

En este informe, sobre el prototipo construido para demostrar nuestro sistema, explicaremos El alcance del mismo, las tecnologías usadas, y descripción de la visualización por parte de ciertos actores del sistema.

Prototipo

Con el propósito de demostrar una versión inicial del sistema, para detectar posibles errores de entendimiento, construimos un prototipo, el cual cumple requisitos esenciales.

Link a instancia del prototipo:

<https://tp-ing-soft-six.vercel.app/>

Admin:

Usuario: admin

Contraseña: 1234

El prototipo cumple:

- Se pueden visualizar noticias, las cuales se pueden buscar y filtrar por nombre.
- Se puede visualizar un mapa usando una librería, Leaflet. Se pueden visualizar puntos que representan la localidad de las noticias.
- Abstracción de login: No existe un backend o servidor de cuentas, pero se puede loguear con datos estáticos, para visualizar el proceso. Esto también permite la creación de noticias. **Los datos para esto son: usuario “admin”, contraseña “1234”.**
- Normalización de direcciones. Habiendo hecho el login de administrador, en la creación de noticias, al ingresar una dirección de la noticia a crear se muestran direcciones normalizadas en el momento.

El prototipo NO cumple:

- Es solamente una interfaz gráfica local, no posee servidor o base de datos. La permanencia de noticias está “hardcodeada” en un archivo del código fuente.
- No existe sistema de preguntas y respuestas.
- No existe sistema de boletín en ningún aspecto.
- No existe la perspectiva del actor “Vecino”. El sitio solamente representa las visualizaciones de No Registrado, y Administrador.

Como detalle adicional, el prototipo consta de diferentes páginas. Utiliza una sola página, la cual utiliza Typescript (Una alternativa a Javascript) para mostrar “popups” y “acordiones”.

Tecnología

- Construcción de la interfaz de usuario: Utilizamos React con TypeScript para una interfaz robusta, modular y fuertemente tipada.
- Ejecución rápida: Utilizamos Vite para una ejecución rápida y poder agrupar los componentes y dependencias de React.
- Estilos css: Utilizamos Tailwind para tener estilos css rápido y responsivos.
- Normalización: Para poder normalizar direcciones utilizamos la API de USIG (servicios.usig.buenosaires.gob.ar) que nos otorga las coordenadas geográficas necesarias para poder visualizarla en el mapa.
- Mapa: Obtenemos el soporte para el mapa y los marcadores por medio de una librería para React llamada React Leaflet.

Visualización:

El prototipo sirve para mostrar las vistas de No registrado y Administrador.

No registrado:

Se puede visualizar una barra superior, con el potencial logo de la empresa, y un botón de Log In. Al hacer click al botón, un popup aparece, permitiendo el login. Por falta de permisos, esto no es posible.

Sigue más abajo un pantallazo de una noticia guardada. Con botones sutiles a la izquierda y derecha, se puede hacer un *swipe* para ver otras noticias.

Más abajo, en una columna, se visualiza el catálogo de noticias, donde aparecen solapas de noticias, y una barra para buscar y filtrar por nombre. Al hacer click en las solapas, estas se abren como un acordeón, mostrando la información de la noticia.

En otra columna, se visualiza el mapa.

Administrador:

Idéntico a No registrado, con la diferencia de que al tener permisos para hacer login si puede hacerlo. Esto genera un botón nuevo, "Crear Publicación", al costado del botón de Login.

Al hacer click, se abre un popup, que permite el ingreso de datos de una noticia, y la creación de la misma. Al ingresar direcciones, se presentan versiones normalizadas.

Estructura del prototipo:

Raiz:

index.html: HTML base donde se inyecta la app de React.
package.json y package-lock.json: Dependencias y scripts del proyecto.
tsconfig.*.json: Configuración de TypeScript.
vite.config.ts: Configuración de Vite.

SRC:

App.tsx: Componente raíz. Organiza el layout general.
main.tsx: Es el punto de entrada. Renderiza App.tsx en el index.html.
index.css: Estilos base, donde se importa Tailwind.
vite-env.d.ts: Ayuda a Vite con definiciones de tipos personalizados.

SRC/Components:

Contiene los componentes reutilizables y estructurales de la app:

Carousel.tsx: Componente para mostrar imágenes o noticias en un carrusel.
CrearNoticiaForm.tsx: Formulario para crear nuevas noticias.
Header.tsx: Encabezado.
Mapa.tsx: Muestra un mapa interactivo usando React Leaflet y direcciones normalizadas.
NoticiasAccordion.tsx: Muestra las noticias de manera plegable (acordeón).
SesionModal.tsx: Modal para gestionar sesión/login.

SRC/Types:

index.ts: Define Noticias como tipo TypeScript reutilizable.

SRC/Data:

noticiasData.ts: Tenemos las noticias que cargamos.

SRC/Context:

NoticiasContext.tsx: Centraliza el estado global y funciones relacionadas con la gestión de noticias y sesión de administrador.