

Taller Nº 3

Introducción al Desarrollo Web Móvil - 17 Noviembre 2025

Valentina Henríquez - Cristhian Rabi

Historia del Proyecto

La startup **DataMobile** busca desarrollar una aplicación web móvil que permita a sus usuarios visualizar datos dinámicos provenientes de una base de datos en tiempo real, de forma clara, moderna e interactiva. El objetivo principal es crear un **panel (dashboard)** que muestre información relevante mediante gráficos, filtros y métricas clave.

El equipo desarrollador deberá implementar la aplicación utilizando **Next.js** como framework full stack, aprovechando sus capacidades tanto del lado del cliente como del servidor para construir la API y el frontend desde un mismo entorno.

La aplicación debe permitir a los usuarios:

- Visualizar un conjunto de **registros almacenados en una base de datos** (por ejemplo: productos, usuarios, mediciones, o KPIs).
- Acceder a **una vista general (dashboard)** que presente los datos en formato tabular y gráfico.
- Navegar a **una vista detallada** por cada registro, donde se muestre información individual. Aplicar **filtros y ordenamientos dinámicos**.
- Al aplicar filtros o configuraciones (por ejemplo: rango de fechas, categorías, tipo de gráfico, etc.), estos deben **mantenerse activos incluso si el usuario cambia de página y luego vuelve**.
- Observar los datos representados mediante **gráficos interactivos** (barras, líneas, tortas, etc).

La aplicación debe ser **responsiva, usable en dispositivos móviles** y ofrecer una experiencia fluida y atractiva.

Criterios Técnicos y Requerimientos

Arquitectura y Framework

- Utilizar **Next.js** (versión 14 o superior) como framework full stack.
- Implementar la **API** dentro del propio proyecto Next.js (rutas bajo **/api**).
- Conectarse a una **base de datos relacional o no relacional** (por ejemplo: PostgreSQL, MySQL o MongoDB).
- Usar un ORM o cliente de base de datos (por ejemplo: Prisma o Mongoose).

Gestión de Estado

- Integrar **Redux Toolkit** para manejar el estado global de la aplicación.
- Utilizar Redux para controlar el flujo de datos entre componentes, filtros y resultados visualizados.

Visualización de Datos

- Integrar una **librería de gráficos** (por ejemplo: Chart.js, Recharts o Nivo).
- Incluir al menos **5 tipos diferentes de gráficos** en el dashboard.
- Los gráficos deben actualizarse dinámicamente al aplicar filtros o actualizar los datos.

Diseño y Responsividad

- Implementar un enfoque **Mobile First**.
- Adaptar el diseño a móviles, tablets y escritorio.
- Se recomienda el uso de **alguna librería de componentes**.
- Mantener coherencia visual, legibilidad y buena experiencia de usuario.

Datos y API

- Implementar al menos **una entidad principal** (por ejemplo: "usuarios", "productos", "sensores", "transacciones" o "cultivos").
- La API debe permitir las operaciones **CRUD** (Create, Read, Update, Delete).
- Validar y manejar errores en las respuestas de la API.
- La información debe provenir de la base de datos, no de archivos locales o JSON estáticos.

Organización del Proyecto

- El trabajo se realizará en **equipos de 3 a 4 integrantes**.
- Se debe subir al Campus Virtual:
 - El **link al repositorio de GitHub** con todo el código fuente.
 - Un **archivo README.md** con
 - Número de grupo.
 - Nombre, apellido y RUT de cada integrante.
 - Descripción breve del proyecto.
 - Instrucciones para instalar y ejecutar la aplicación localmente.

Fecha de Entrega



Entrega: 12/12/2025



Formato: Link al repositorio (GitHub) + README.md completo



Consultas: cristhian.rabi@ce.ucn.cl - valentina.henriquez@ce.ucn.cl