

**UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO**

Nombre de la actividad:

# ***Entrega de prácticas individuales***

Nombre de la materia:

**Desarrollo para Dispositivos Inteligentes**

División:

**TECNOLOGÍAS DE LA** **INFORMACIÓN**

**INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE**

Presenta:

[**JESUS GONZALEZ LEAL**](mailto:2021171013@uteq.edu.mx)

Grupo:

**IDGS08**

**Objetivo general de mi proyecto:**

El proyecto consiste en el desarrollo de una aplicación web responsiva que permita a las empresas gestionar sus gerencias, enfocándose en la administración de empleados. Esta aplicación facilitará las siguientes funciones:

1. **Registro de nuevos empleados:**
   * Asignarles un puesto dentro de la empresa.
   * Vincularlos a un departamento correspondiente.
2. **Gestión de empleados existentes:**
   * Editar la información del empleado registrado.
   * Eliminar empleados cuando sea necesario.
   * Cambiar el estatus del empleado (activo/inactivo).
   * Agregar nuevos empleados rápidamente.

La aplicación será accesible desde distintos dispositivos, como teléfonos móviles, tablets y computadoras. Además, se implementará como una **Aplicación Web Progresiva (PWA)** utilizando Firebase para su despliegue, lo que permitirá su instalación en dispositivos móviles y su uso offline en ciertos casos.

### **Problemas a Resolver**

1. **Falta de centralización de datos:**
   * La información de empleados no está organizada en un sistema único, lo que genera confusión y errores.
2. **Procesos manuales y lentos:**
   * Actualizar datos de empleados es ineficiente y propenso a errores.
3. **Acceso limitado a herramientas:**
   * No se cuenta con una plataforma disponible en dispositivos móviles y computadoras.

### **Beneficios Esperados**

1. **Gestión eficiente y centralizada:**
   * Registro y actualización de empleados de manera rápida y organizada en un sistema único.
2. **Accesibilidad multiplataforma:**
   * Compatible con teléfonos, tablets y computadoras, con soporte offline mediante Firebase.
3. **Automatización y modernización:**
   * Reducción de errores, mayor productividad y escalabilidad tecnológica.

### **Funciones Claves**

1. **Autenticación y Usuarios:**
   * **Login y registro de usuarios:** Los usuarios podrán registrarse e iniciar sesión de manera segura.
   * **Validación con la base de datos:** Verificación de credenciales y datos almacenados en la base.
2. **Geolocalización y Clima:**
   * **Información climática:** Provee la estación del clima actual según la ubicación del usuario.
   * **Mapa interactivo:** Muestra las coordenadas (latitud y longitud) utilizando Google Maps.
3. **Integración de APIs Externas:**
   * **API de noticias:** Visualización de noticias relevantes desde un servicio externo.
4. **Estructura Organizacional:**
   * **Listado de estructuras organizacionales:** Presenta de manera jerárquica las divisiones de la empresa.
5. **Gestión de Departamentos:**
   * **CRUD de departamentos:** Crear, leer, actualizar y eliminar departamentos de la empresa.
6. **Gestión de Empleados:**
   * **CRUD de empleados:** Registro, edición, eliminación y cambio de estatus de empleados.
   * **Integración con cámara:** Permite tomar y subir una foto del empleado al momento de registrarlo.

### **Requerimientos Funcionales**

1. **Autenticación de Usuarios:**
   * Permitir el registro y login de usuarios con validación de credenciales frente a la base de datos.
   * Garantizar que solo usuarios autenticados accedan a las funcionalidades del sistema.
2. **Gestión de Empleados:**
   * Registrar nuevos empleados asignándoles un departamento, puesto y foto mediante la cámara integrada.
   * Editar, eliminar o cambiar el estatus de empleados existentes.
3. **Gestión de Departamentos:**
   * CRUD completo para crear, editar, consultar y eliminar departamentos.
4. **Visualización de Estructura Organizacional:**
   * Mostrar la jerarquía de la empresa de forma interactiva y clara.
5. **Geolocalización y Clima:**
   * Proveer información del clima según la ubicación actual del usuario.
   * Mostrar las coordenadas de latitud y longitud en un mapa interactivo con Google Maps.
6. **Noticias:**
   * Integrar un API para mostrar noticias relevantes dentro de la aplicación.

### **Requerimientos No Funcionales**

1. **Usabilidad:**
   * La aplicación debe ser intuitiva y fácil de usar, con una interfaz responsiva optimizada para dispositivos móviles, tablets y computadoras.
2. **Rendimiento:**
   * Carga rápida de mapas, noticias y otros datos externos.
   * Respuesta eficiente del sistema al realizar operaciones CRUD.
3. **Compatibilidad:**
   * Accesible desde navegadores modernos y adaptable a diferentes tamaños de pantalla.
4. **Disponibilidad:**
   * Implementar como PWA para que sea accesible offline en funcionalidades básicas.
5. **Seguridad:**
   * Cifrado de contraseñas y datos sensibles durante el almacenamiento y transmisión.
   * Validación de entradas para evitar inyecciones SQL o ataques similares.
6. **Escalabilidad:**
   * Posibilidad de integrar nuevos módulos o funcionalidades sin afectar el rendimiento.
7. **Mantenimiento:**
   * Código estructurado y documentado para facilitar la corrección de errores y futuras actualizaciones.
8. **Integración con APIs externas:**
   * Conexión estable y segura con servicios como Google Maps y el API de noticias.

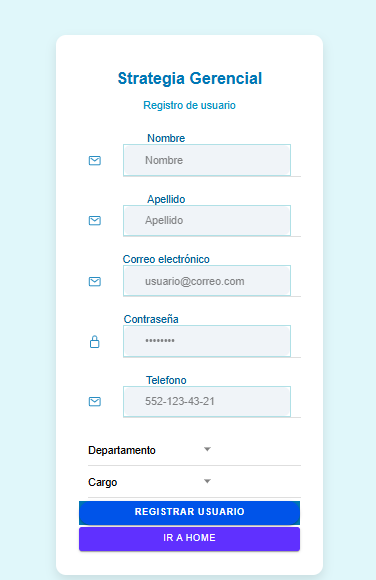
***Manual de usuario:***

***Inicio De Sesión:***

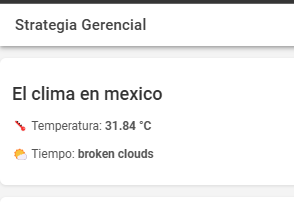
*Mi aplicación cuenta con un módulo de inicio de sesión en el que los usuarios previamente registrados podrán acceder al sistema. Este módulo valida que el usuario que intenta iniciar sesión esté registrado en la base de datos. Para implementar esta funcionalidad, utilicé la librería Axios, ya que la había utilizado previamente en otros proyectos con PHP y JavaScript.*

***Registro de usuarios.***

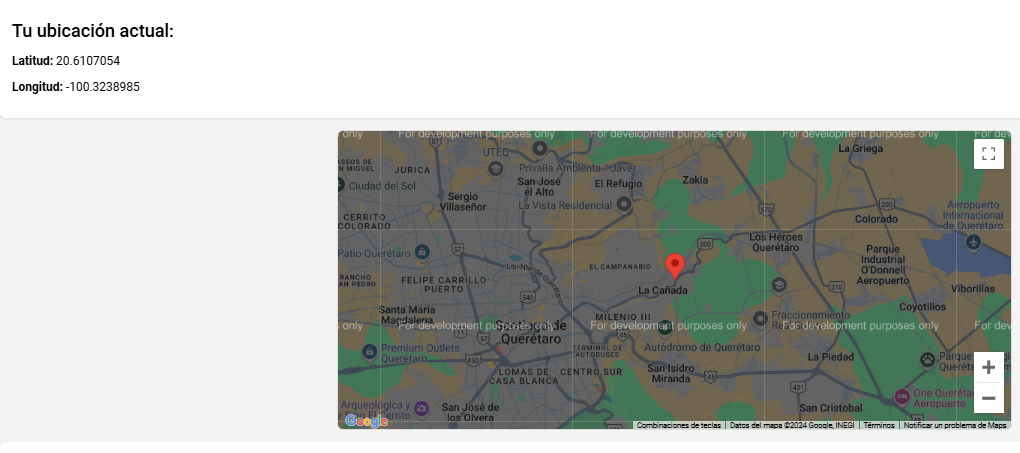
*En el apartado de registro de usuario, el usuario puede registrarse para luego iniciar sesión. Todos los campos están validados. Utilicé un componente select para mostrar los múltiples departamentos y posiciones de la empresa, y la información es consumida mediante un servicio en Axios. El backend está desarrollado en PHP utilizando el framework Codeigniter con el patrón MVC.*



***Home***

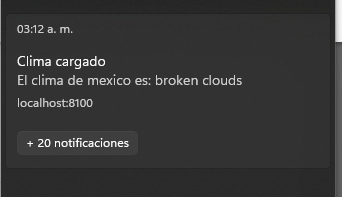
*Una vez que el usuario entra al sistema, accede a la pestaña de 'Home', que contiene múltiples secciones. Una de estas secciones muestra el clima actual en México y el pronóstico del clima. Para esto, consumí una API de WeatherService y la consulté mediante Axios.*

*Otra sección del sistema muestra la ubicación actual del dispositivo. Para esto, utilicé la librería de geolocalización, que obtiene en tiempo real la latitud y longitud del dispositivo que accede a la aplicación. Una vez obtenida la latitud y longitud, usé la librería de Google Maps para ubicar la posición en el mapa con un marcador.*

*Otra sección del sistema muestra las noticias más populares de México. Utilicé una API de newsService y la consumí mediante Axios, mostrando las noticias en un scroll view que se actualiza en tiempo real con las nuevas noticias.*

******

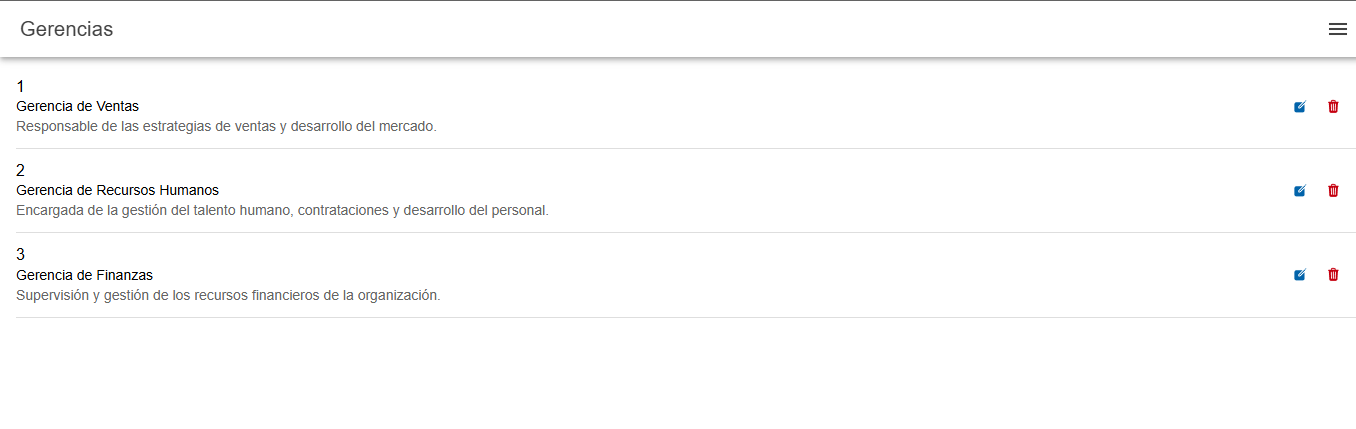
*La última sección de la pestaña 'Home' es un apartado para el consumo de vídeo en streaming. Utilicé la API de YouTube y la librería YouTubePlayerModule para consumir la API de YouTube. Primero obtuve la API key desde Google Cloud Console, luego agregué una variable para guardar el ID del video y ajusté las medidas del video para que se muestre correctamente.*

*También en la página de 'Home' agregué la funcionalidad de notificaciones. Las notificaciones indican tu ubicación actual enviando la latitud y longitud, y también te informan sobre la cantidad de noticias nuevas que se están cargando. Para esto, utilicé la librería LocalNotifications, enviando notificaciones cada vez que se cargan nuevas noticias o se obtiene la ubicación actual.*

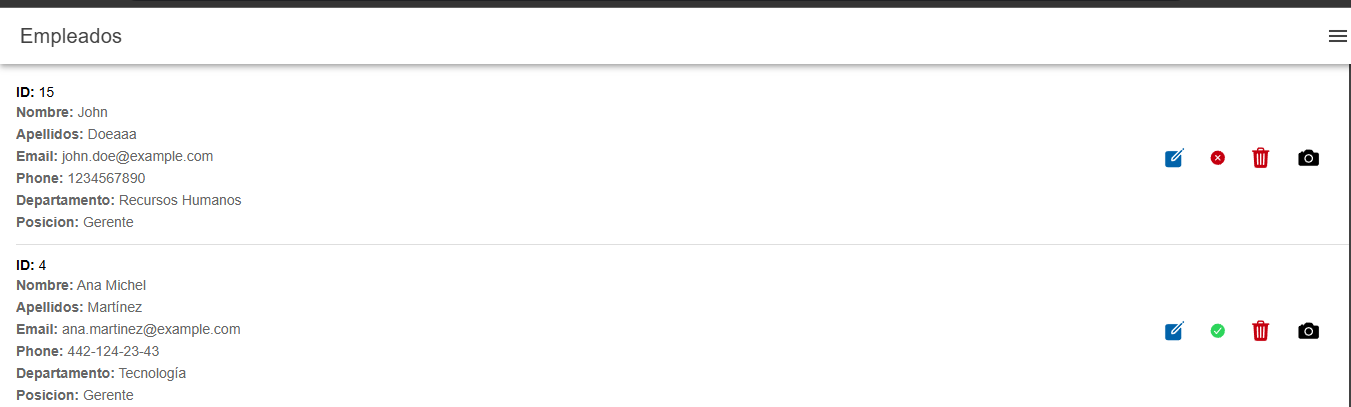
***En el apartado de estructura organizacional.***

*Es un módulo donde se muestra la estructura organizacional de la empresa. Para esto, consumí un servicio con Axios y lo mostré en una lista.*

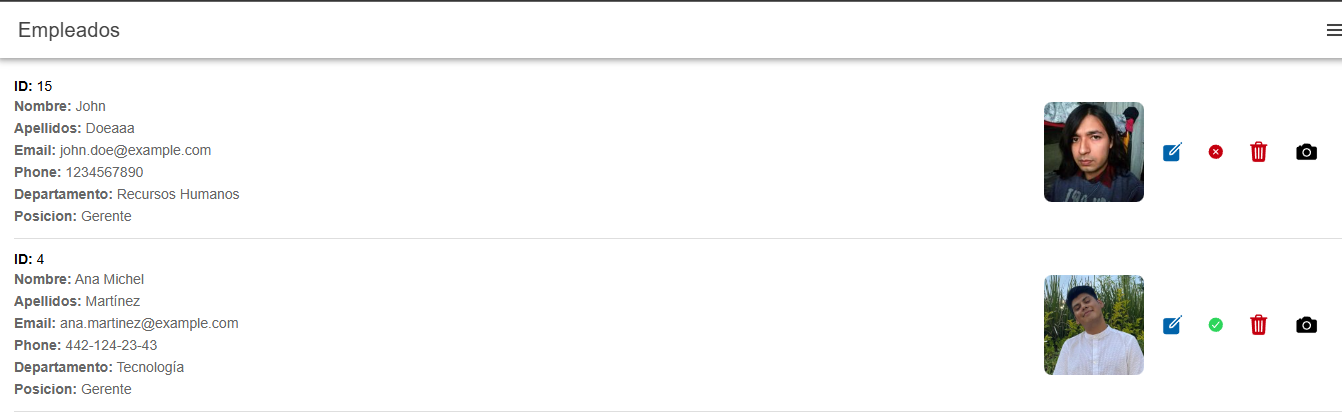
**En el apartado de gerencias**

En este apartado, implementé un CRUD sencillo que muestra las gerencias dadas de alta en la base de datos. Se pueden eliminar con borrado lógico y editar nuevas gerencias. Todo esto lo hice utilizando la librería Axios.

**En el apartado de empleados.**

*En este apartado, implementé un CRUD más complejo que permite visualizar todos los empleados registrados en la base de datos. Se puede cambiar el estatus de los empleados, mostrando en rojo a los inactivos y en verde a los activos. También se pueden eliminar los empleados con borrado lógico y editar los nuevos empleados. Toda esta funcionalidad la implementé utilizando Axios y modales de Bootstrap.*

*También agregué la funcionalidad para tomar una foto y guardarla en el perfil del empleado, activando la cámara en dispositivos Android y la galería en computadoras. Para activar la cámara, utilicé la librería capacitor/camera, y para guardar la imagen, utilicé la librería Axios.*



*También subí la aplicación a Firebase utilizando @angular/service-worker y firebase-tools para posicionarla en un proyecto de Firebase y permitir que cualquier persona pueda testear la aplicación.*

*Link:*[*https://control-gerencias-app.web.app/inicio*](https://control-gerencias-app.web.app/inicio)