|  |  |
| --- | --- |
|  |  |
| Caso Práctico u6 |  |
|  |  |
|  | 19-02-2025Joaquín García |
|  | Desarrollo Web de Entorno Servidor *DAW, Cesur.* |

Índice

[1. Introducción 3](#_Toc191389412)

[2. Arquitectura y tecnologías. 3](#_Toc191389413)

[3. Desarrollo de proyecto: 3](#_Toc191389414)

[4. Bibliografía. 3](#_Toc191389415)

# 1. Introducción

En este caso práctico de la u6, se va a presentar el desarrollo de una aplicación web para la gestión de empleados que permite:

* Gestionar la información de los empleados de una compañía conectando con una base de datos
* Realizar operaciones CRUD sobre registros de empleados
* Módulo de búsqueda avanzada para filtrar empleados según criterios.
* Recuperar la información de la base de datos dinámicamente en el frontend con estadísticas en tiempo real
* Interfaz moderna e intuitiva.
* Sistema de autenticación mediante log in para proteger los datos.

**Requisitos de Desarrollo**:

* Node.js: Versión 18.17 o superior (preferiblemente Node.js 20.x)
* npm: Versión 9.x o superior
* Git: Versión 2.x o superior
* Editor de código: Visual Studio Code (recomendado) con extensiones:
* ESLint
* Prettier
* TypeScript
* Tailwind CSS IntelliSense

Esta aplicación es compartible para su despliegue en plataforma de hosting compatible con Next.js (Vercel, Netlify..:

# 2. Arquitectura y tecnologías.

1. **Frontend:**

* **Next.js 14**: Es un framework de React (biblioteca de JavaScript para crear interfaces de usuario interactivas) para aplicaciones web que permite renderizado del lado del servidor, quiere decir que el servidor procesa las solicitudes del usuario y genera dinámicamente el HTML necesario con datos actualizados. Tiene soporte de TypeScript nativo.
* **Tailwind CSS:** Tailwind CSS: Es un framework de CSS de utilidad que permite diseñar interfaces de usuario de manera eficiente y coherente mediante clases predefinidas. Facilita la creación de diseños personalizados sin escribir CSS desde cero, clases predefinidas para estilos personalización sencilla.
* **Tremor**: Es una biblioteca de componentes de React diseñada para construir dashboards y visualizaciones de datos. Ofrece más de 35 componentes de código abierto, accesibles y construidos con React, Tailwind CSS y Radix UI. Estos componentes permiten crear interfaces analíticas de manera rápida y con un diseño atractivo

1. **Backend , Base de Datos, Autenticación y Seguridad.**

* **Supabase:** Es una plataforma de Backend-as-a-Service(BaaS) que proporciona una base de datos PostgreSQL con funcionalidades en tiempo real, autenticación y almacenamiento de archivos. Se añaden como dependencie en el paquete package.json y se configuran en el paquete env.local mediante unas credenciales en la cuenta donde esta alojada la base de datos. Supabase Auth cuenta con un sistema de autenticación, con sesiones seguras, gestión de usuario.
* **API Route en next.js,** se ha utilizado para crear endpoints y manejar solicitudes HTTP directamente en el código.

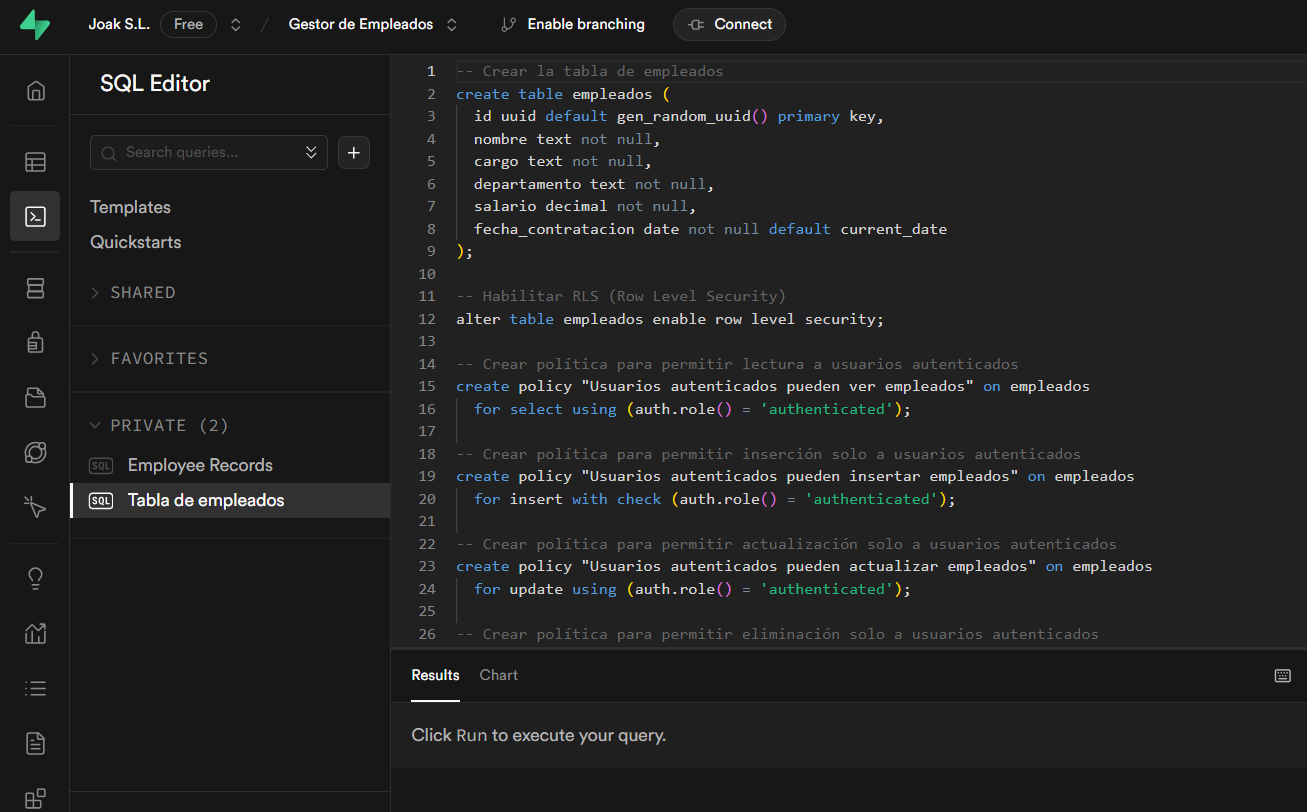
1. **Herramientas de Desarrollo.**

* **Typescript** : Es un superconjunto de JavaScript desarrollado por Microsoft que añade tipado estático y objetos basados en clases, ayuda a mantener un código más seguro y mantenible.
* **JSdocs :** Se han habilitado comentarios en JSdocs para poder desarrollar un documento en el caso de que fuese necesario.

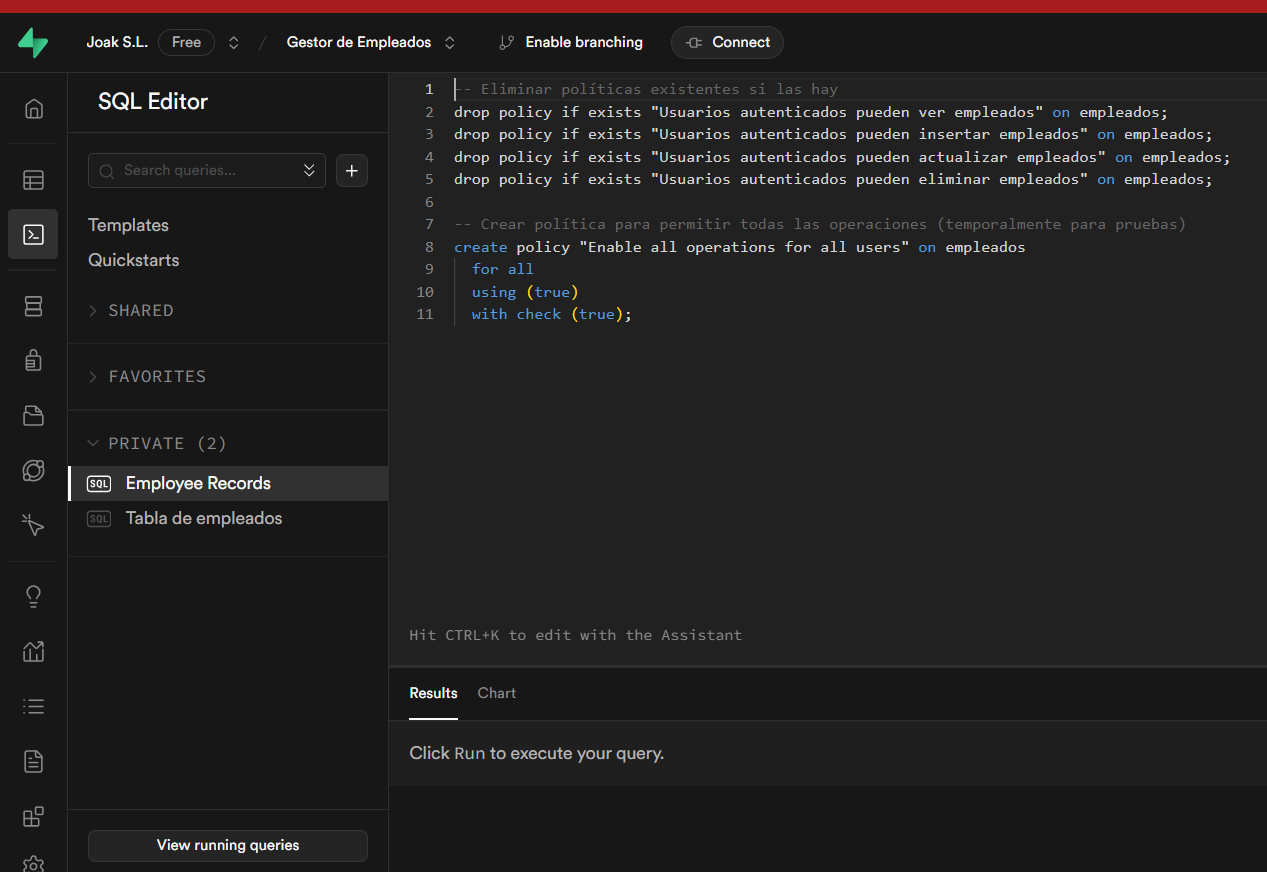
# 3. Desarrollo de proyecto:

**3.1 Creación base de datos.**

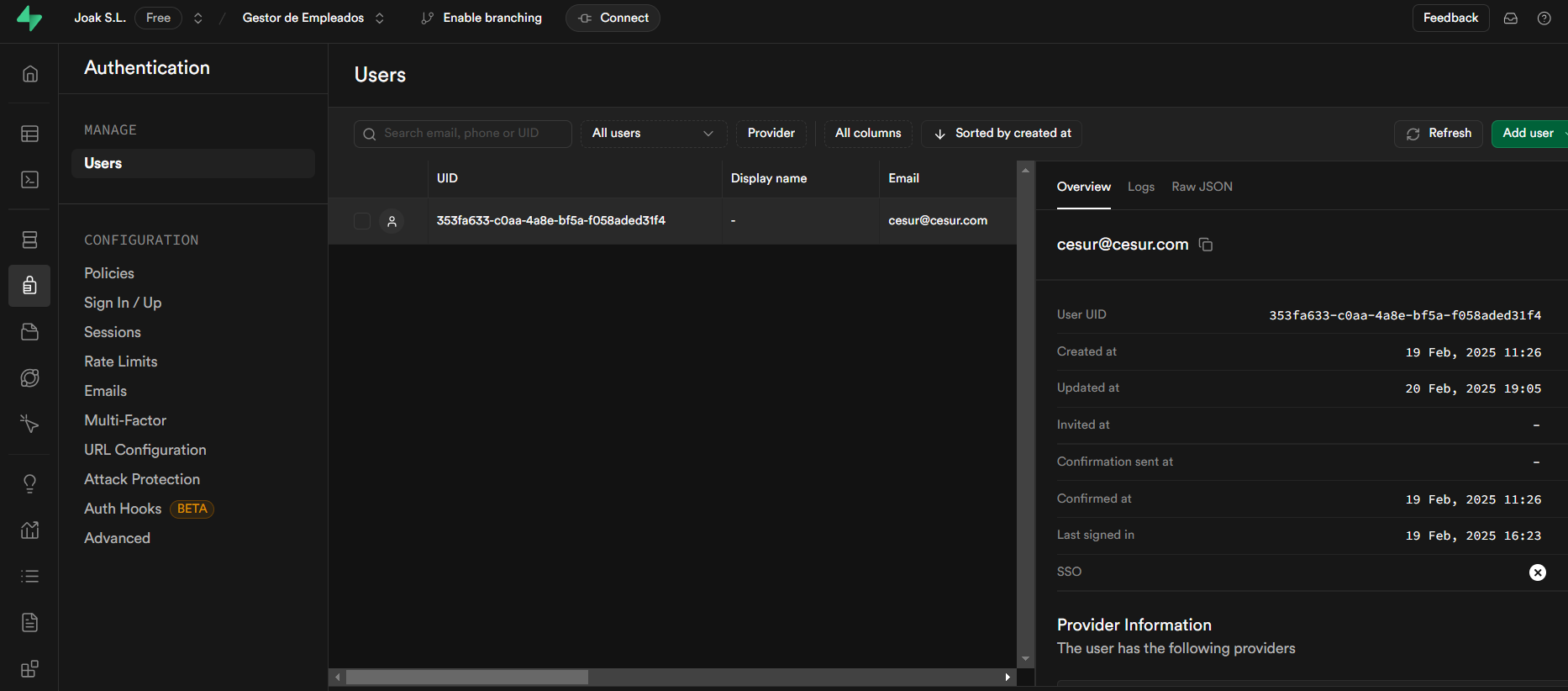
Se ha creado una cuenta en Supabase con las siguientes indicaciones :



Es una parte fundamental permitir las políticas para que los usuarios puedan realizar operaciones sobre la base de datos.



Se creará un usuario como [cesur@cesur.com](mailto:cesur@cesur.com) con contraseña cesur1234, para que el usuario final pueda autenticarse en la aplicación. Mediante un log in de credenciales.



Por úlitmo Supabase nos dará unas credenciales en NEXT\_PUBLIC\_SUPABASE\_URL

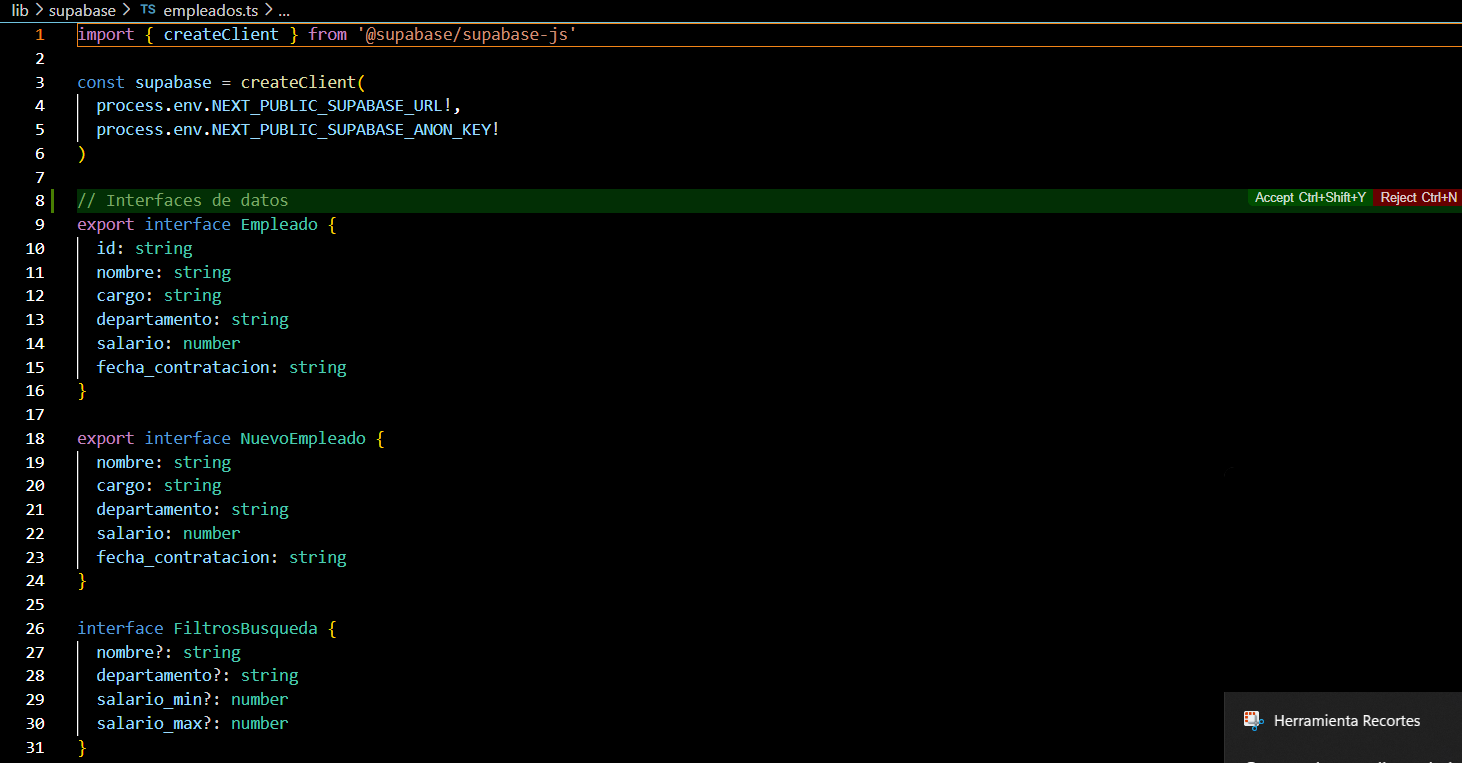
NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY, que le añadiremos en el paquete env.local a nuestro proyecto para enlazar la base de datos.

**3.2 Recuperación y gestión de la información.**

Controlador Principal de Empleados se encuentra en lib/supabase/empleados.ts este controlador centraliza todas las operaciones CRUD:

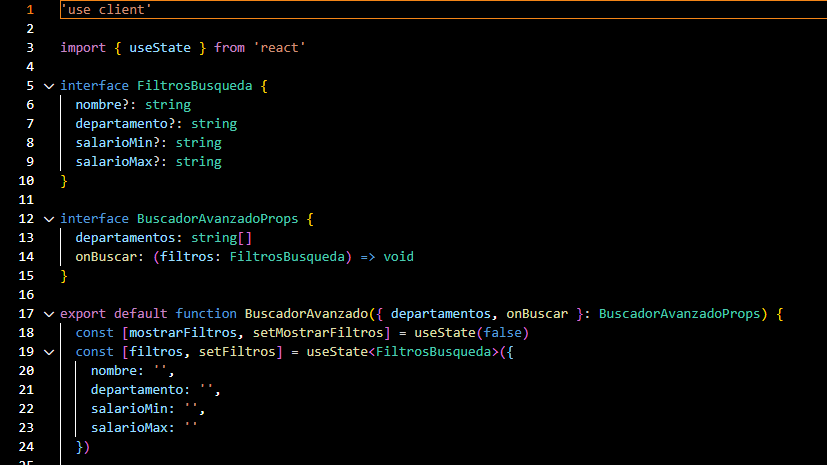
* obtenerEmpleados(): Recupera lista completa
* obtenerEmpleado(id): Recupera empleado específico
* agregarEmpleado(datos): Crea nuevo registro
* actualizarEmpleado(id, datos): Modifica registro existente
* eliminarEmpleado(id): Elimina registro

Será necesario crear las interfaces de Empleado, FiltrosBusqueda, Nuevo empleado para definir la estructura y el tipo de los datos que manejamos en la aplicación.

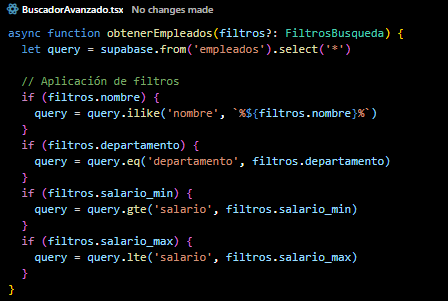


**3.3 Búsqueda Avanzada y Filtrado de empleados**

Mediante la interface BuscadorAvanzadoProps en components/BuscadorAvanzado.tsx se han implementado unos filtros de búsqueda tales como: Nombre, Departamento, Salario. Es fundamental la gestion del estado, importando useState de React para manejar datos dinámicos y en tiempo real.



Se integran la función de búsqueda con el backend mediante:



El flujo de los datos para su recuperación quedaría de la siguiente manera:

* Cliente hace petición → page.tsx
* Middleware verifica autenticación → middleware.ts
* Controlador procesa petición → lib/supabase/empleados.ts
* Supabase ejecuta consulta → Base de datos PostgreSQL
* Datos retornan al cliente → Componentes React

**3.4 Publicación dinámica de la información.**

En el archivo components/Dashboard.tsx es un componente crucial que proporciona una visión general del estado de los empleados, combinando datos estadísticos con visualizaciones interactivas. Aquí importamos tremor que nos proporciona una biblioteca para gráficos dinámicos.

Mediante ‘use client’, indica que el componente se renderiza en el cliente.



**3.5 Seguridad y Auntenticación**

En el archivo middleware.ts actúa como una capa de seguridad, control de acceso y gestión de sesiones. Mediante Supabase Auth se implementa el sistema de autenticación.



# 

* <https://nextjs.org/learn/react-foundations/what-is-react-and-nextjs>
* <https://tailwindcss.com/docs/installation/using-vite>
* <https://github.com/app-generator/sample-tremor-github-charts>
* <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
* https://supabase.com/docs