**Asignatura: Taller de Diseño y desarrollo de Soluciones**

**Sección: D-IEI-N4-P1-C1**

**Nombre del docente:** Armin Vladimir Brun Ruth

**Nombre de los integrantes del grupo:** Nicolas Vicencio

**Fecha de entrega 26/09/2022**

Guía Informe de Proyecto

Unidad 2: Interfaces, codificación, base de datos y configuración del entorno

**Contenido**

[I. Introducción 5](#_Toc1)

[II. Objetivo 5](#_Toc2)

[III. Construir las interfaces gráficas. 6](#_Toc3)

[IV. Codificar la solución. 10](#_Toc4)

[V. Implementar la base de datos. 12](#_Toc5)

[VI. Configurar el entorno que soporta la aplicación. 13](#_Toc6)

[VII. Conclusiones 14](#_Toc7)

[VIII. Referencias bibliográficas 15](#_Toc8)

Indice de Figuras

[Figura 1 - Vista de Login 6](#_Toc1)

[Figura 2 - Vista de Registro 6](#_Toc2)

[Figura 3 - Vista de Tareas 7](#_Toc3)

[Figura 4 - Vista de Eventos 7](#_Toc4)

[Figura 5 - Vista de Calendario 8](#_Toc5)

[Figura 6 - Vista de Timer 8](#_Toc6)

[Figura 7 - Vista de Habitos 9](#_Toc7)

[Figura 8 - Vista de Perfil 9](#_Toc8)

[Figura 9 - Coneccion con supabase 10](#_Toc9)

[Figura 10 - Codigo de gestion de authenticacion de OrganiApp 10](#_Toc10)

[Figura 11 - Variables globales para datos de usuario 11](#_Toc11)

[Figura 12 - Consumiendo datos del contexto useUsers 11](#_Toc12)

[Figura 13 - Instalacion de Dependencias 12](#_Toc13)

[Figura 14 - Variables de Entorno 12](#_Toc14)

[Figura 15 - Conexion Aplicacion/Supabase 12](#_Toc15)

[Figura 16 - Tablas en la nube 12](#_Toc16)

Indice de Tablas

Tabla 1 - Tablas de Dependencias 13

Tabla 2 – Tabla de Navegadores compatibles 13

Tabla 3 - Tabla de Variables de Entorno 13

1. Introducción

Este documento contempla la construccion de las interfaces graficas de la aplicacion OrganiApp para la gestion de tiempo, las interfaces seran construidas en en react con el framework nextjs.

Luego veremos la implementacion de la base de datos en el servicio de supabase y como hacemos la coneccion al servicio a traves del codigo.

Finalmente abordaremos la implementacion y configuracion del entorno de ejecucion, la variables de entorno y paqueterias necesarias para el correcto funcionamiento de nuestra aplicacion.

1. Objetivo

**Objetivo General**

Desarrollar la solucion propuesta anteriormente, utilizando como base, los patrones de diseño Flux y la gestion de base de datos en cloud.

**Objetivos Especificos**

* Crear interfazes sobrias y profesionales.
* Desarrollar la aplicacion con el patron de diseño Flux.
* Utilizar el servicio de Supabase para desplegar nuestra base de datos en la nube.
* Configurar las variables de entorno parar la correcta conexion cliente-servidor.
* Hacer deploy de nuestra aplicacionen Vercel.

1. Construir las interfaces gráficas.

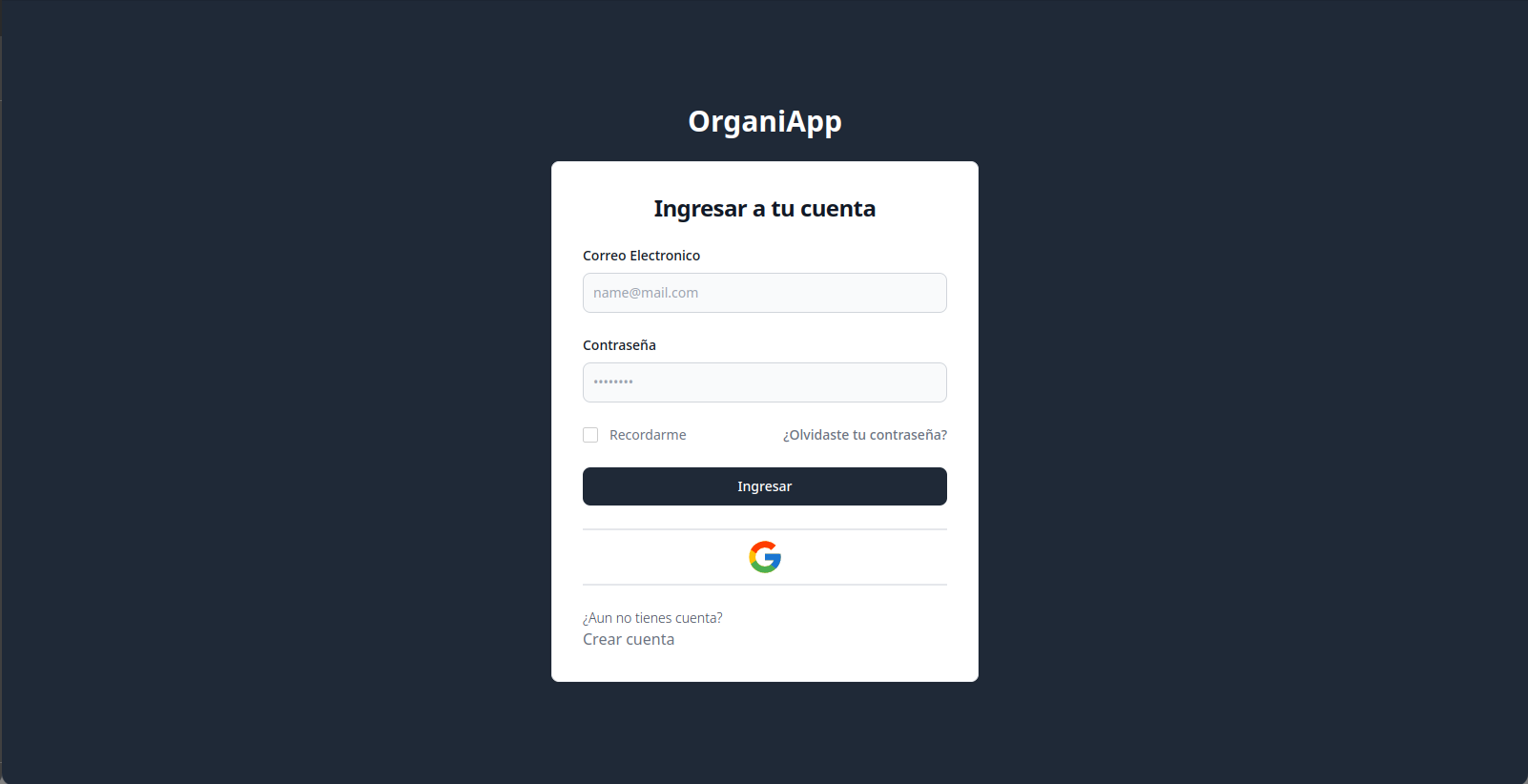


Figura 1 - Vista de Login

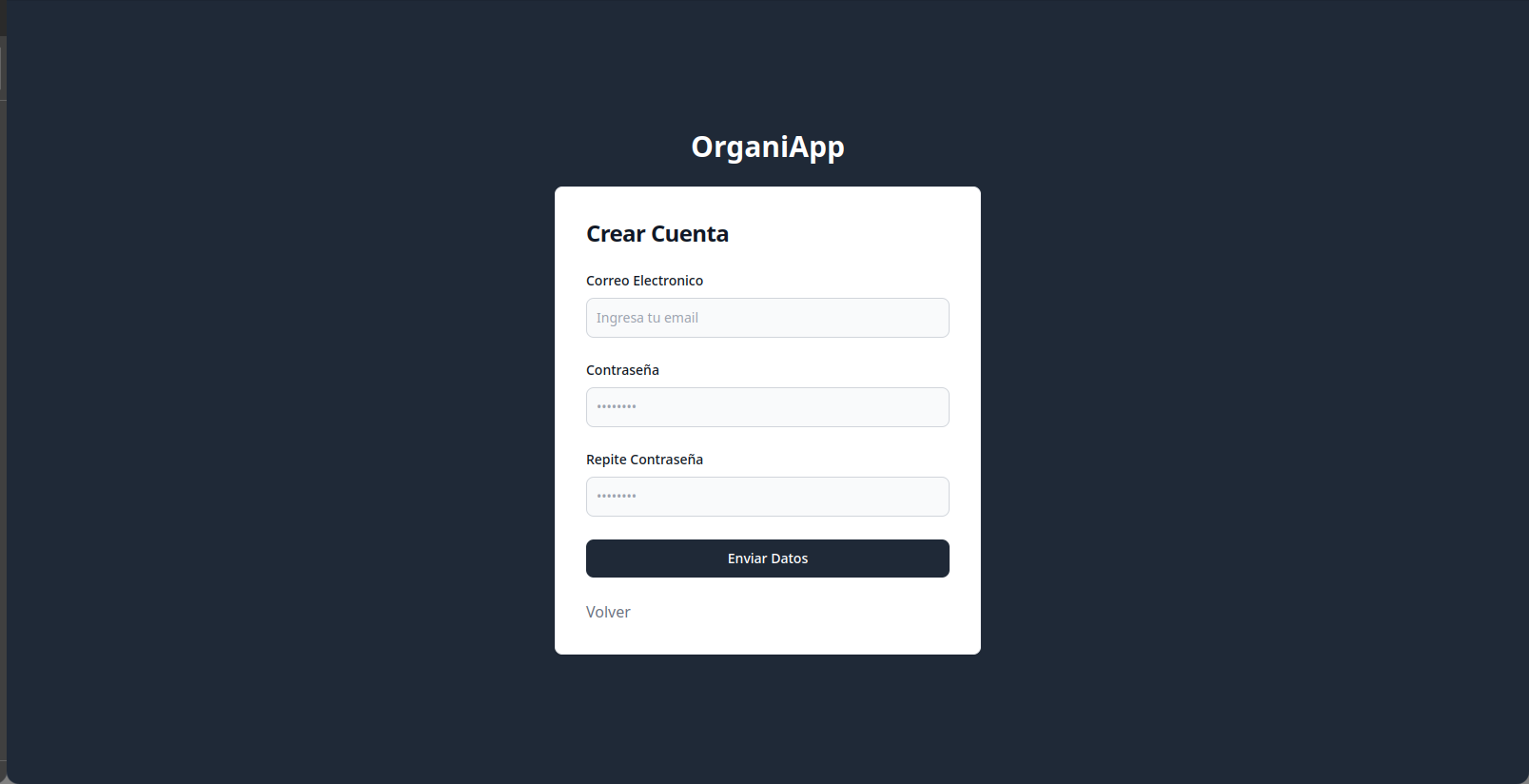


Figura 2 - Vista de Registro

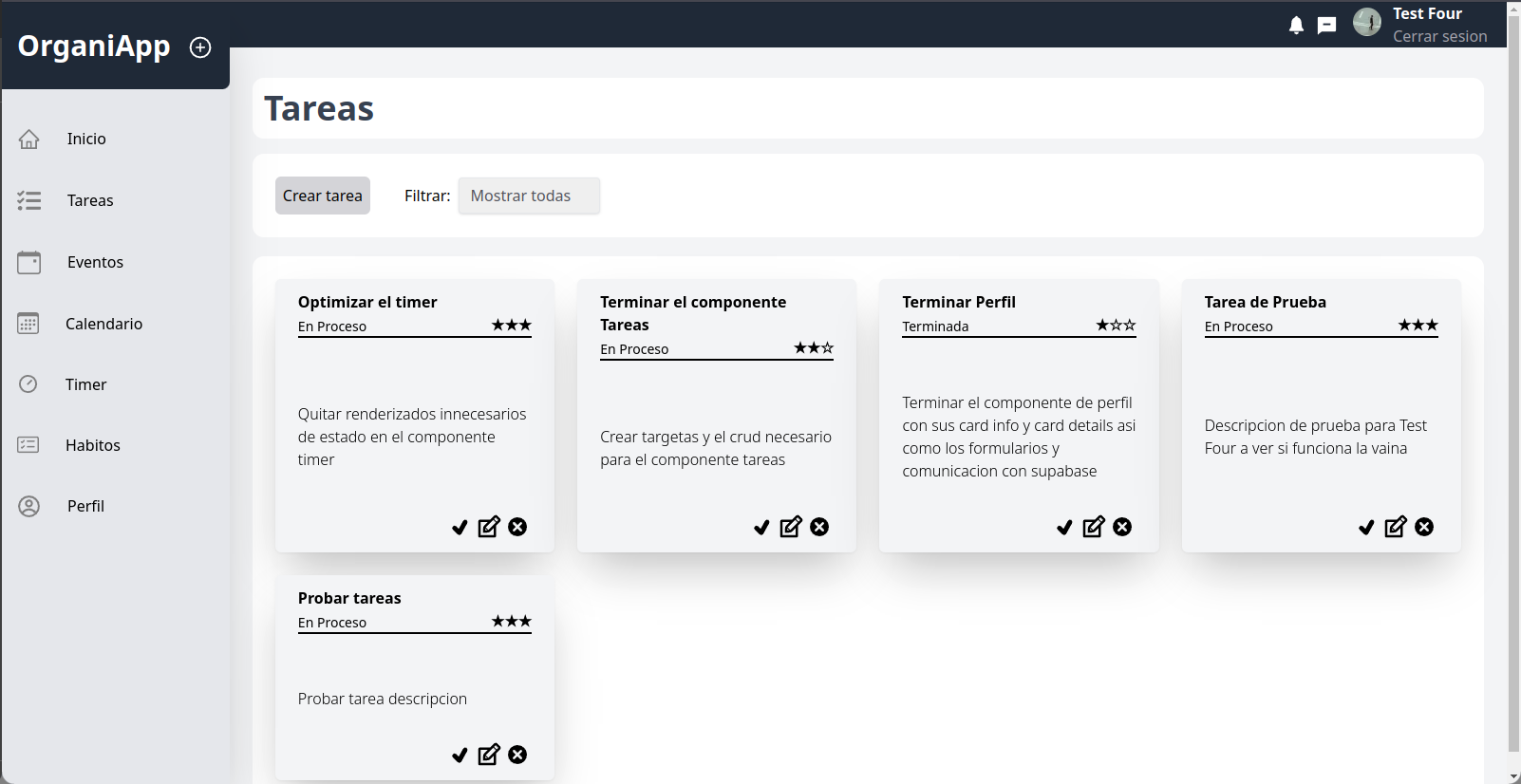


Figura 3 - Vista de Tareas

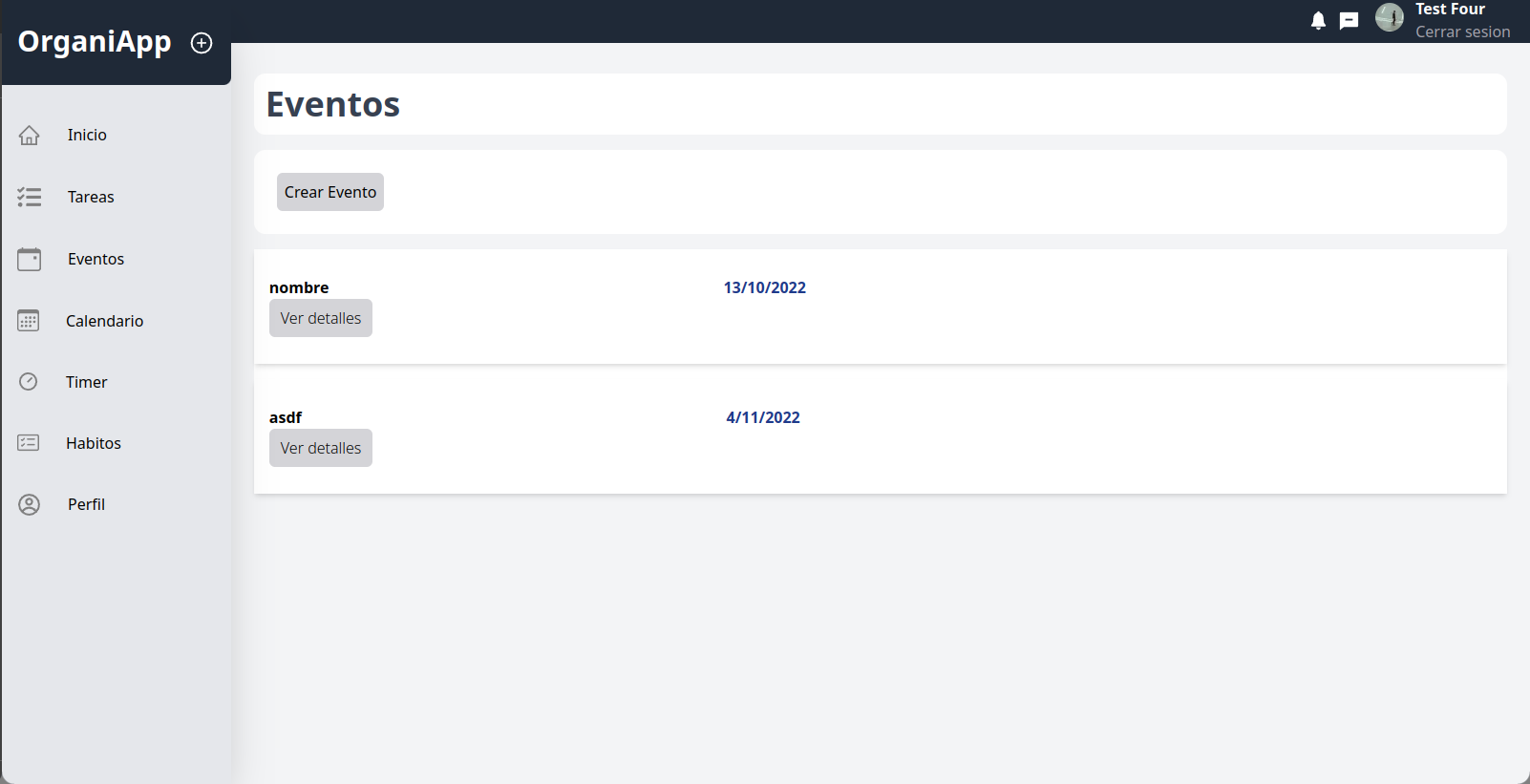


Figura 4 - Vista de Eventos

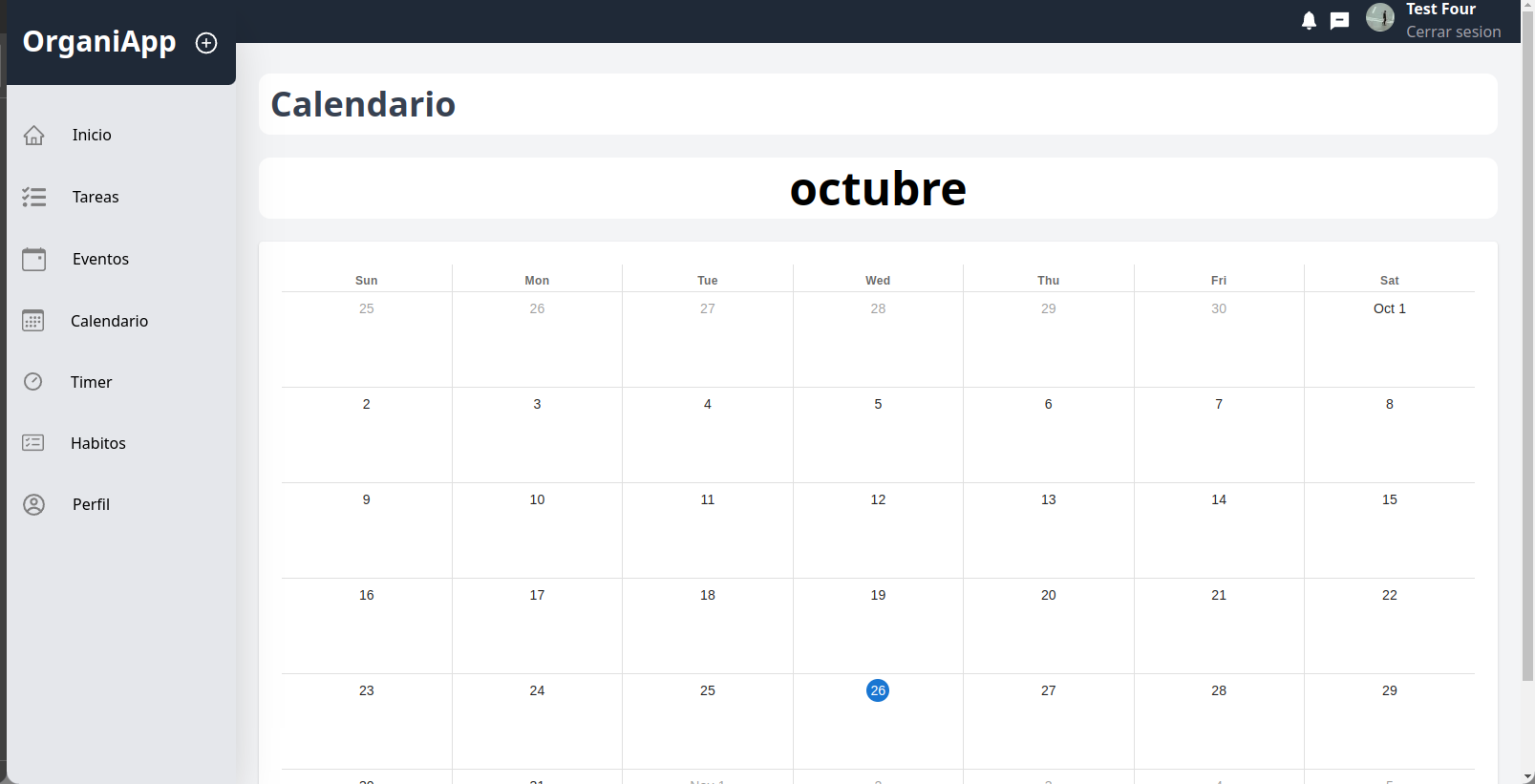


Figura 5 - Vista de Calendario

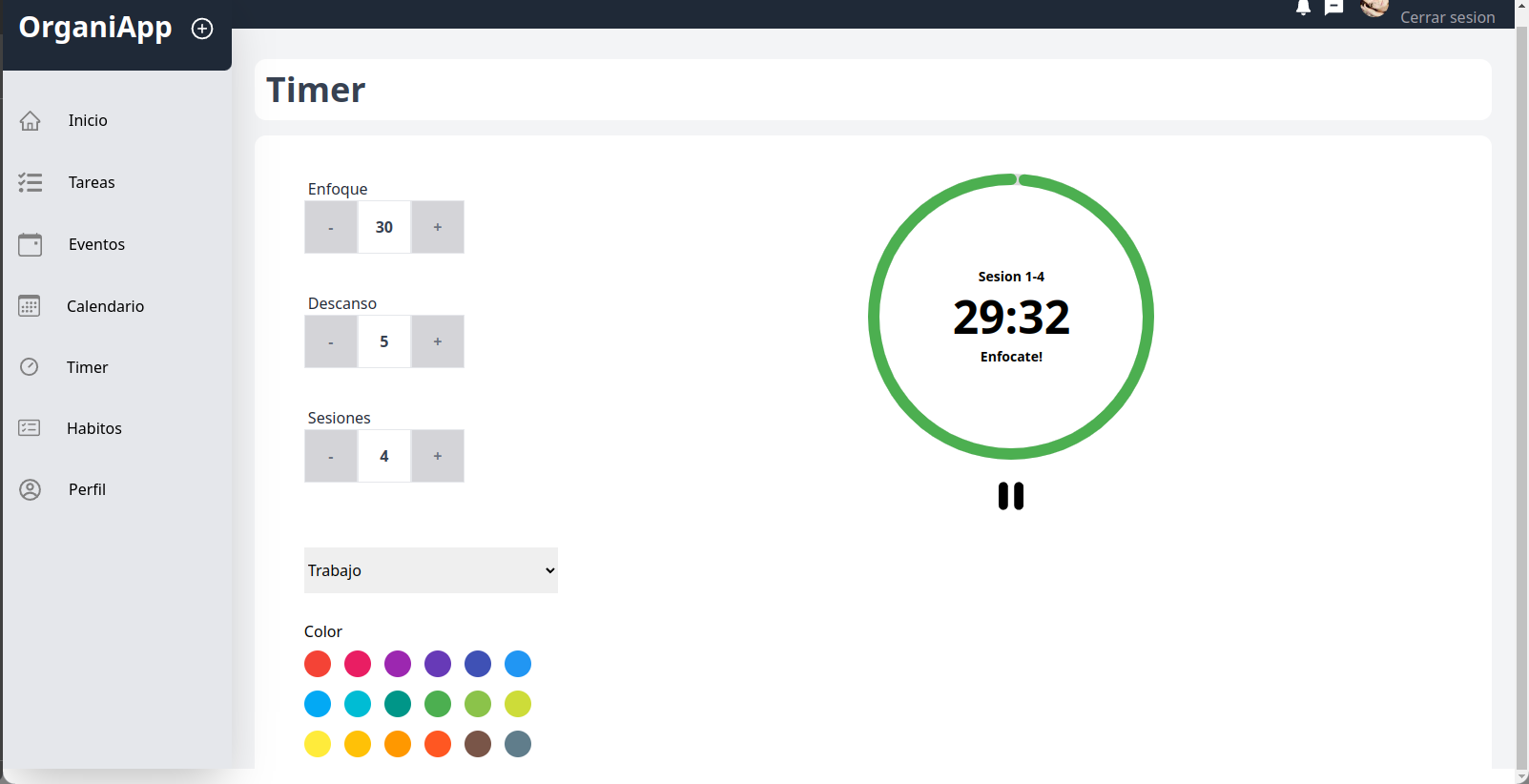


Figura 6 - Vista de Timer

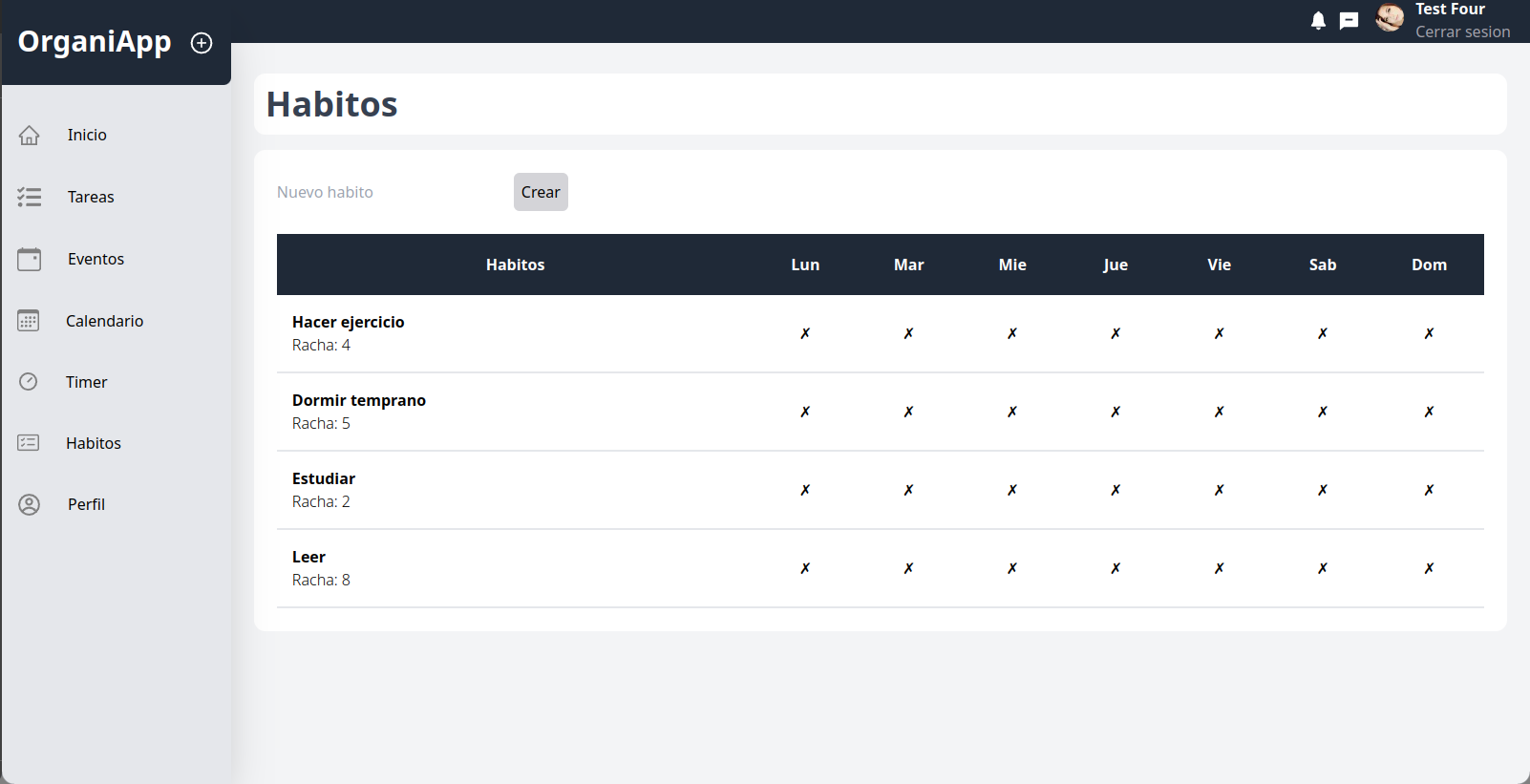


Figura 7 - Vista de Habitos

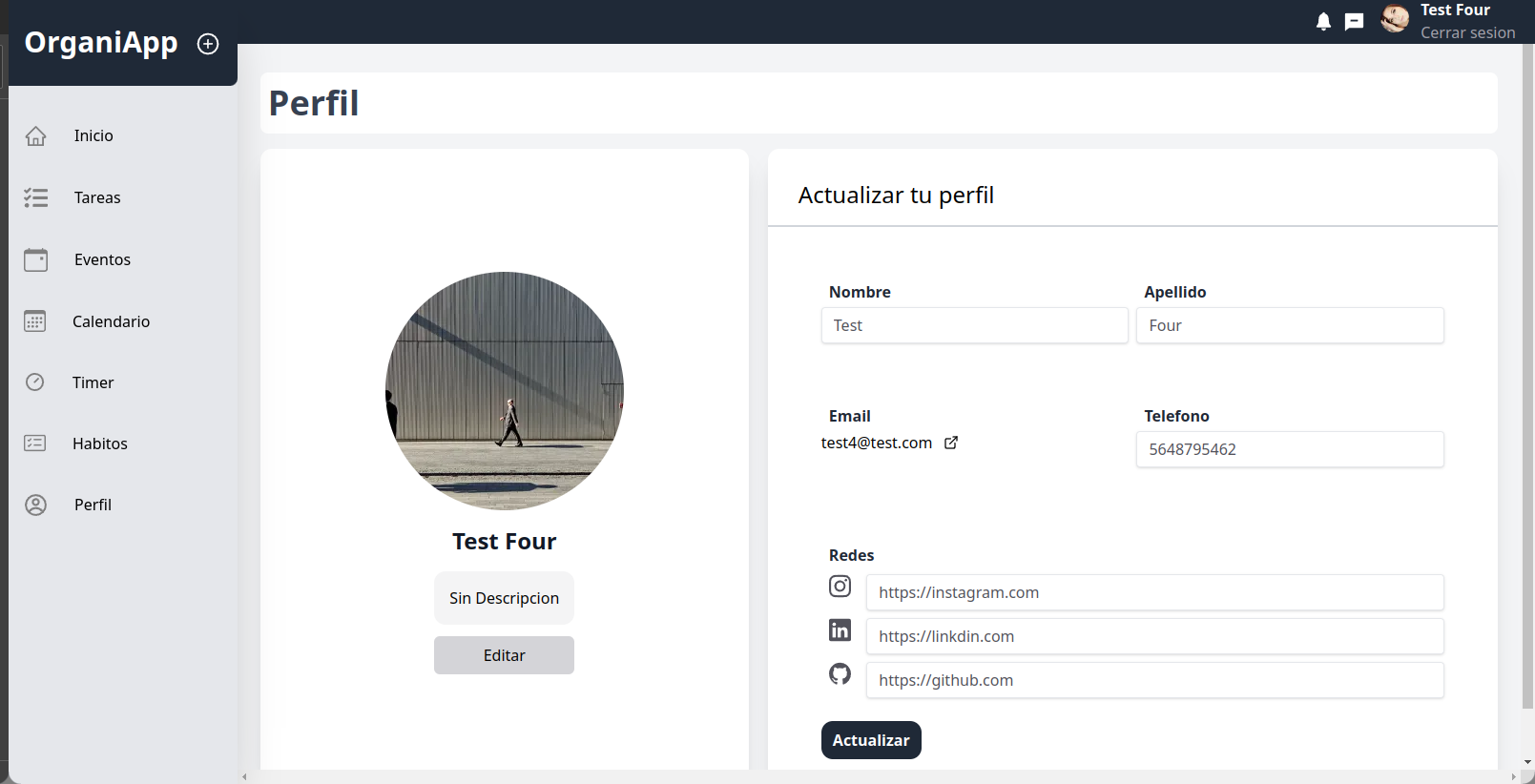


Figura 8 - Vista de Perfil

1. Codificar la solución.

Para conectarnos con el servicio de backend tenemos que codificar el siguiente archivo

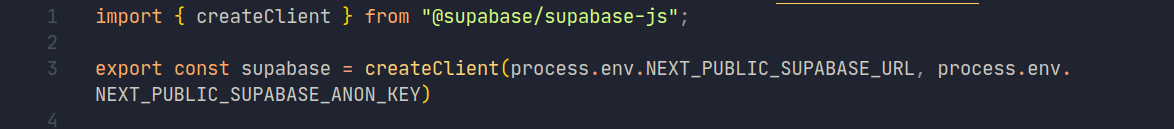


Figura 9 - Conexion con supabase

Creamos una constante llamada supabase y la exportamos, esta constante la utilizaremos en toda nuestra aplicacion para consumir los datos de la nube.

El metodo createClient nos pide dos parametros que son nuestras claves unicas de configuracion de supabase, las manejamos en variables de entorno para asegurar la seguridad de nuestra aplicacion.

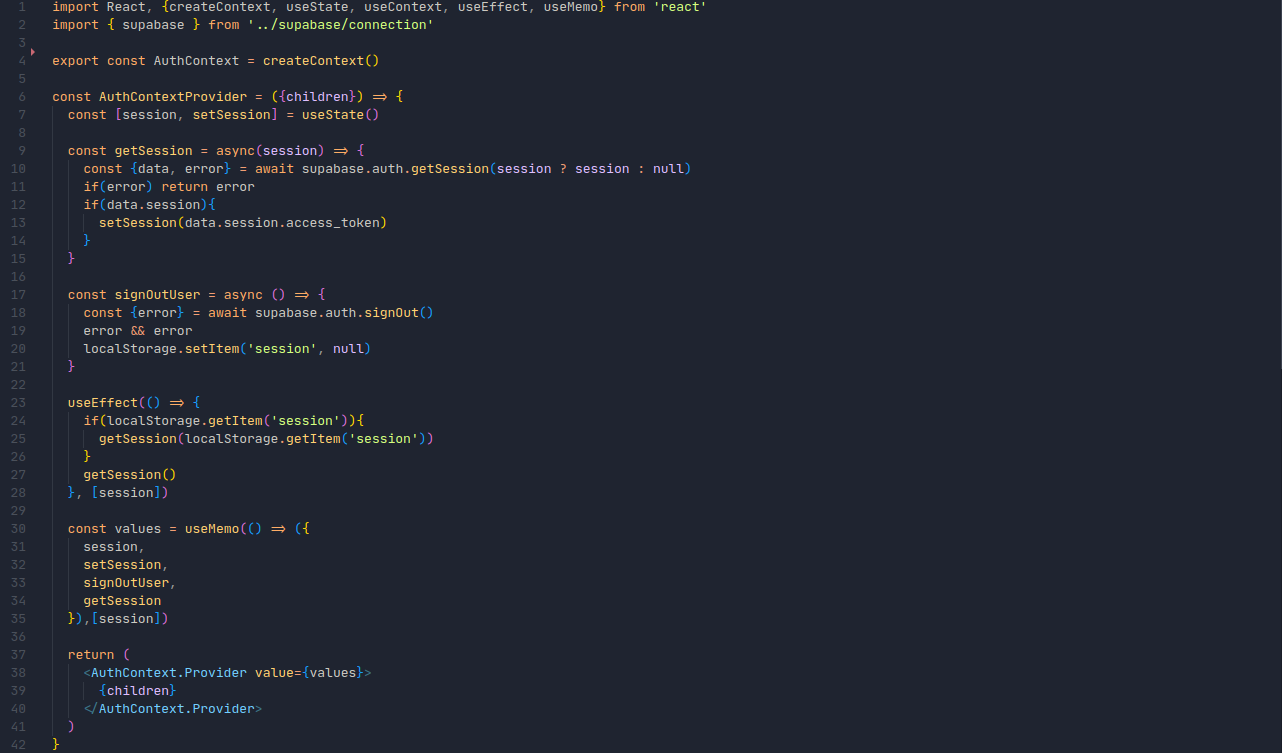


Figura 10 - Codigo de gestion de authenticacion de OrganiApp

Para seguir con el patron de diseno Flux creamos un contex que seria nuestro store de variables globales, estas variables estaran disponibles para toda nuestra aplicacion.

Creamos un metodo para iniciar sesion y que sera llamado por el componente Login, en el momento en el que el usuario ingresa credenciales validas, supabase nos entregara un json web token que le pasaremos como parametro al metodo de getUser para obtener los datos de la sesion activa.

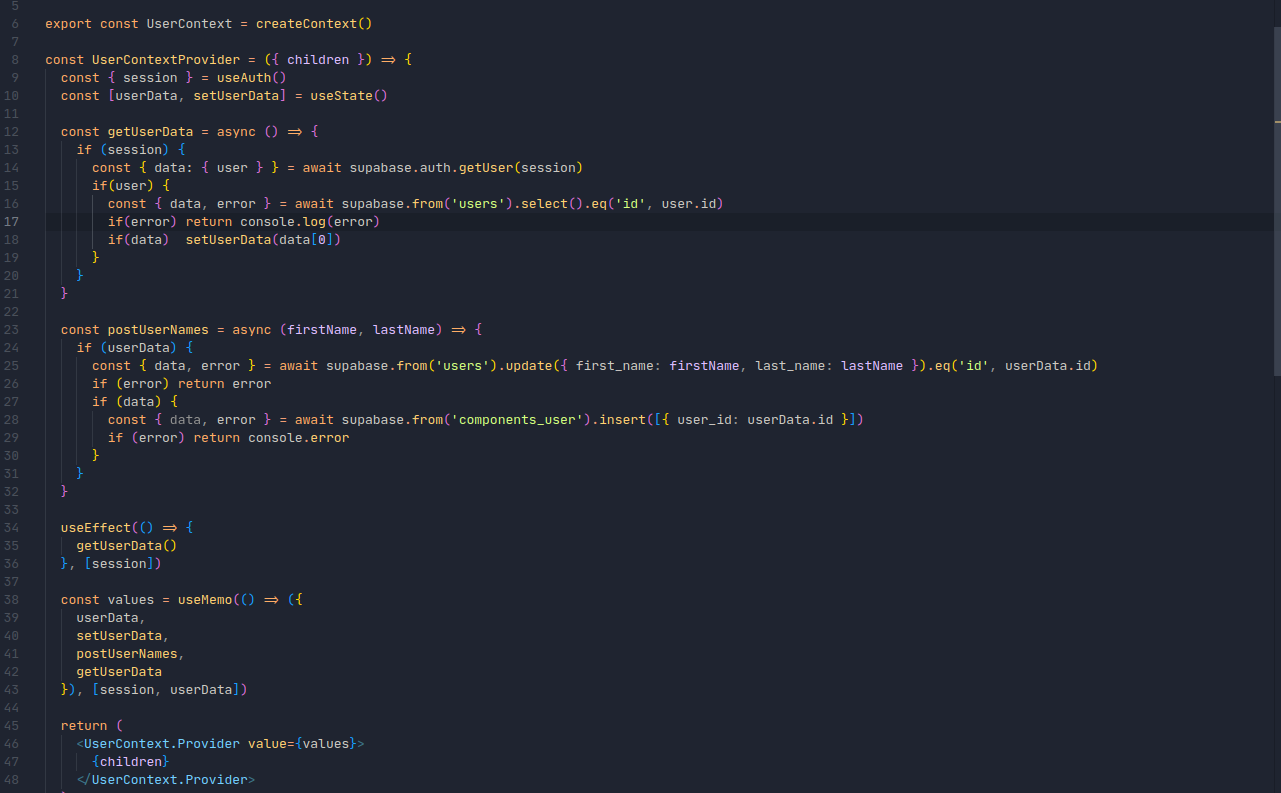


Figura 11 - Variables globales para datos de usuario

Creamos otro contexto para almacenar los datos de usuario de la sesion activa y compatirlos con el resto de componentes de la aplicacion.

Cuando necesitos consumir los datos en otro componente necesitamos exporar nuesto context y destructurar de el objeto que retorna el dato que necesitemos.



Figura 12 - Consumiendo datos del contexto useUsers

1. Implementar la base de datos.

Para conectar nuestra aplicacion con supabase necesitamos instalar las dependencias de supabase con el siguiente comando.



Figura 13 - Instalacion de Dependencias

Luego creamos un archivo .env.local en el cual ingresamos nuestras credenciales personales



Figura 14 - Variables de Entorno

El siguiente codigo configura la conexion de supabase con el resto de la aplicacion, al llamar a la constante supabase accedemos a los metodos necesarios para realizar peticiones.

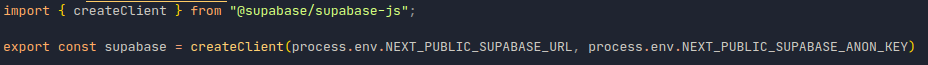


Figura 15 - Conexion Aplicacion/Supabase

En Supabase creamos nuestras tablas, a la derecha vemos las tablas y a la izquierda las filas.

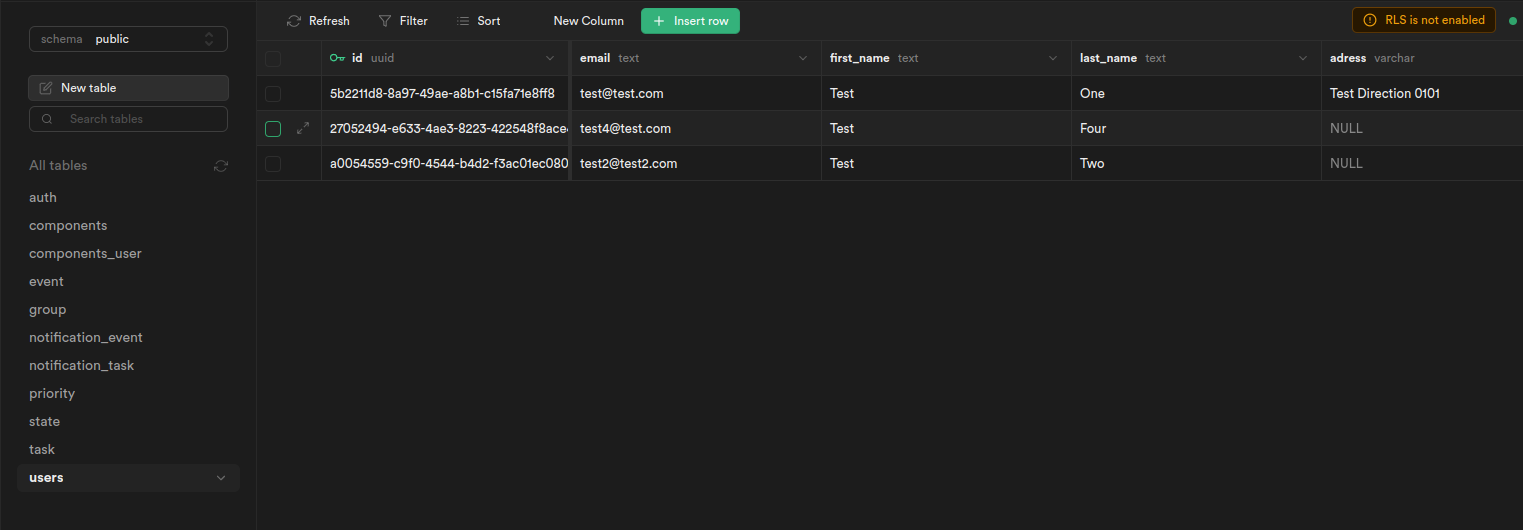


Figura 16 - Tablas en la nube

1. Configurar el entorno que soporta la aplicación.

Para el correcto funcionamiento y despliegue de nuestra aplicacion necesitamos que el servidor cuente con las siguientes dependencias:

|  |  |
| --- | --- |
| **Dependencia** | **Version** |
| NodeJS | 16.16.o |
| Npm | 8.19.0 |
| NextJS | 12.3.1 |
| ReactJS | 18.2.0 |
| PostgreeSQL | 14.0.1 |

Tabla 1 - Tablas de Dependencias

Para el correcto uso de la aplicacion como cliente es necesario uno de los siguientes navegadores

|  |
| --- |
| **Navegador Compatibles** |
| Google Chrome |
| Mozilla Firefox |
| Vivaldi |
| Yandex Browser |
| Opera |
| Cualquier navegador basado en chromiun |

**Tabla 2 – Tabla de Navegadores compatibles**

Para realizar correctamente el deploy en Vercel.com necesitamos otorgale las variables de entorno de nuestro proveedor de Supabase, para que este se conecte con los datos alojados en supabase.

|  |  |
| --- | --- |
| **Nombre Variable de Entorno** | **Valor Variable de Entorno** |
| NEXT\_PUBLIC\_SUPABASE\_URL | Project URL (app.supabase.com) |
| NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY | Project API anon public (app.supabse.com) |

Tabla 3 - Tabla de Variables de Entorno

Si se quiere ejecutar la aplicacion de forma local, en la raiz del projecto, cree un archivo con nombre “.env.local” y escriba las variables de entorno.

1. Conclusiones

En este documento vimos las interfaces graficas de la aplicacion , ademas de la implementacion de la base de datos en supabase y los requerimientos parar el correcto funcionamiento de la aplicacion en el servidor, los navegadores disponibles y la ejecucion para desarrollo en local.

1. Referencias bibliográficas

Open Source Supabase (2022). Documentación oficial de Supabase - <https://supabase.com/docs>

Vercel (2022). Documentacion oficial de NextJs - <https://nextjs.org/learn/foundations/about-nextjs?utm_source=next-site&utm_medium=nav-cta&utm_campaign=next-website>

Hook API Reference (2022). Documentacion de hook reactJS - <https://reactjs.org/docs/hooks-reference.html>