



TECNOLÓGICO  
NACIONAL DE MÉXICO

TecNM



**TECNOLÓGICO NACIONAL DE MÉXICO**

**Instituto Tecnológico de Mexicali**

**Ing. Sistemas Computacionales**

**Desarrollo de Aplicaciones Web**

**Docente: Bogarin Valenzuela Jose Ramon**

**“Publicación de Frontend con Vercel”**

**Alumna: Rodriguez Herrera Maria Fernanda**

**Mexicali, B.C.**

**20-Nov-2025**

Para comenzar, en el archivo ***docker-compose*** se deben modificar las variables de entorno de keycloak para que el frontend se comuniquen correctamente con keycloak y la API del backend, para lo cual agregamos las siguientes:

**VITE\_KEYCLOAK\_URL:** es la URL de nuestro servidor de keycloak, en este caso en el puerto 8081.

**VITE\_KEYCLOAK\_REALM:** es el nombre de nuestro reino que creamos en keycloak.

**VITE\_KEYCLOAK\_CLIENT\_ID:** es el id de nuestro cliente de keycloak.

**VITE\_API\_BASE\_URL:** es la URL del backend, en este caso tiene el puerto 8080.

```
args:
  VITE_KEYCLOAK_URL: 'https://j6t9s9cj-8081.usw3.devtunnels.ms'
  VITE_KEYCLOAK_REALM: 'Fernys_Realm'
  VITE_KEYCLOAK_CLIENT_ID: 'RoHeFerny'
  VITE_API_BASE_URL: 'https://j6t9s9cj-8080.usw3.devtunnels.ms/api/v1'
```

Seguido de ello, agregamos ***KC\_HOSTNAME***, que es el nombre por el cual nuestro frontend va a ser accesible desde el exterior (que sería el dominio del puerto 8081). Luego con ***KC\_HOSTNAME\_STRICT*** con valor ***false*** permite que keycloak acepte peticiones, aunque el hostname no coincida exactamente. Después con ***KC\_HTTP\_ENABLED*** con ***true*** decimos que se habilite el protocolo http en el servidor. Por último, con ***KC\_PROXY*** con ***edge*** indicamos que estamos usando un proxy (que en este caso tenemos nginx):

```
keycloak:
  image: quay.io/keycloak/keycloak:25.0.4
  container_name: todo-keycloak
  environment:
    KEYCLOAK_ADMIN: admin
    KEYCLOAK_ADMIN_PASSWORD: admin
    KC_LOG_LEVEL: info
    TZ: America/Mexico_City
    # Base de datos Postgres
    KC_DB: postgres
    KC_DB_USERNAME: todo_user
    KC_DB_PASSWORD: todo_password
    KC_DB_URL_HOST: postgres
    KC_DB_URL_PORT: 5432
    KC_DB_URL_DATABASE: todo_db_v2
    # Configuración del túnel público
    KC_HOSTNAME: j6t9s9cj-8081.usw3.devtunnels.ms
    KC_HOSTNAME_STRICT: 'false'
    KC_HTTP_ENABLED: 'true'
    KC_PROXY: edge
```

Luego en el docker del backend indicamos en el **CORS\_ALLOWED\_ORIGINS** la ruta de nuestro proyecto en vercel:

```
todo-back-app:
  image: todo-back-app-2025
  container_name: todo-back-app
  environment:
    # Configuración de base de datos
    SPRING_DATASOURCE_URL: jdbc:postgresql://postgres:5432/todo_db_v2
    SPRING_DATASOURCE_USERNAME: todo_user
    SPRING_DATASOURCE_PASSWORD: todo_password
    SPRING_DATASOURCE_DRIVER_CLASS_NAME: org.postgresql.Driver

    # Configuración de JPA/Hibernate
    SPRING_JPA_HIBERNATE_DDL_AUTO: update
    SPRING_JPA_SHOW_SQL: false
    SPRING_JPA_PROPERTIES_HIBERNATE_FORMAT_SQL: true
    SPRING_JPA_PROPERTIES_HIBERNATE_DIALECT: org.hibernate.dialect.PostgreSQLDialect
    # Configuración de CORS
    CORS_ALLOWED_ORIGINS: https://todo-frontend-six-mu.vercel.app
```

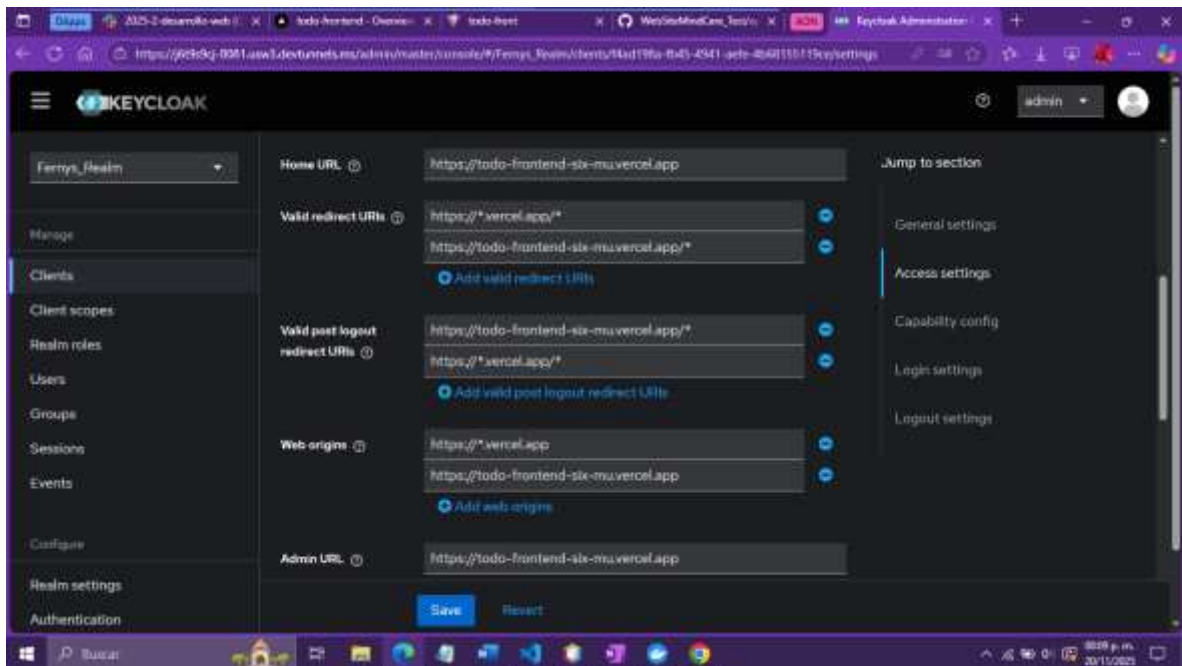
Lo que se hizo también fue un archivo **.env** en el cual indicamos las variables de VITE (que también se encuentran en las variables de entorno de keycloak):

```
🔧 .env.development
1  # Variables de entorno para desarrollo local
2  VITE_KEYCLOAK_URL=https://j6t9s9cj-8081.usw3.devtnnls.ms
3  VITE_KEYCLOAK_REALM=Fernys_Realm
4  VITE_KEYCLOAK_CLIENT_ID=RoHeFerny
5  VITE_API_BASE_URL=https://j6t9s9cj-8080.usw3.devtnnls.ms/api/v1
```

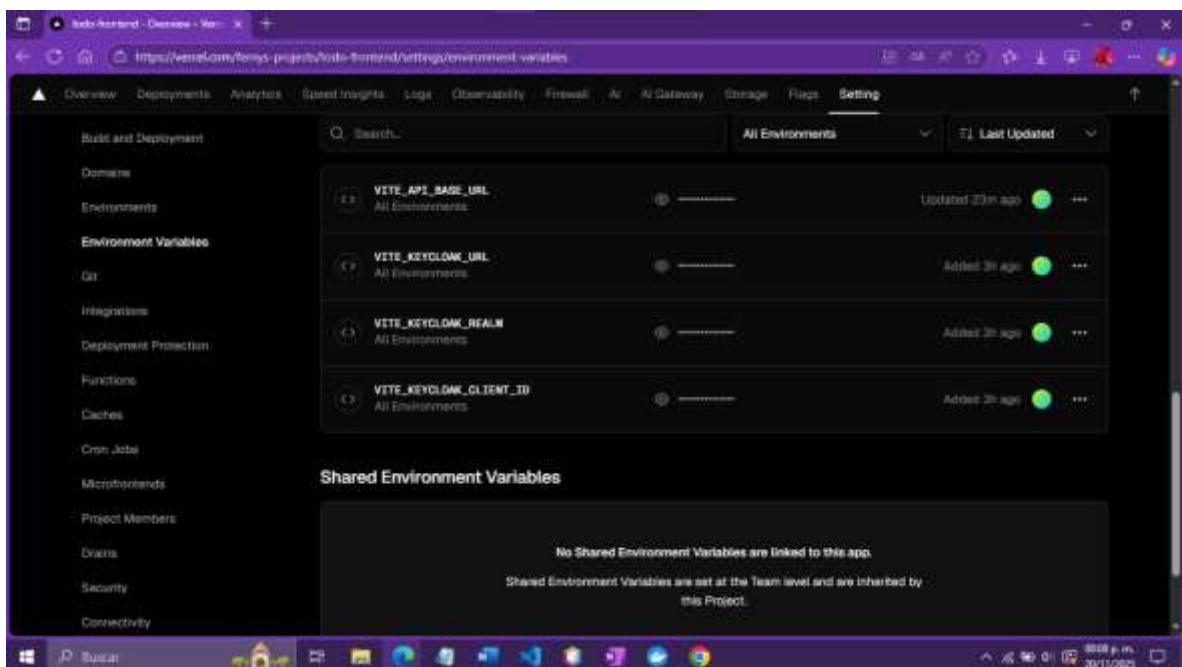
Otra cosa fue cambiar la url en el archivo **todosAPI.ts**, indicándole el puerto del backend:

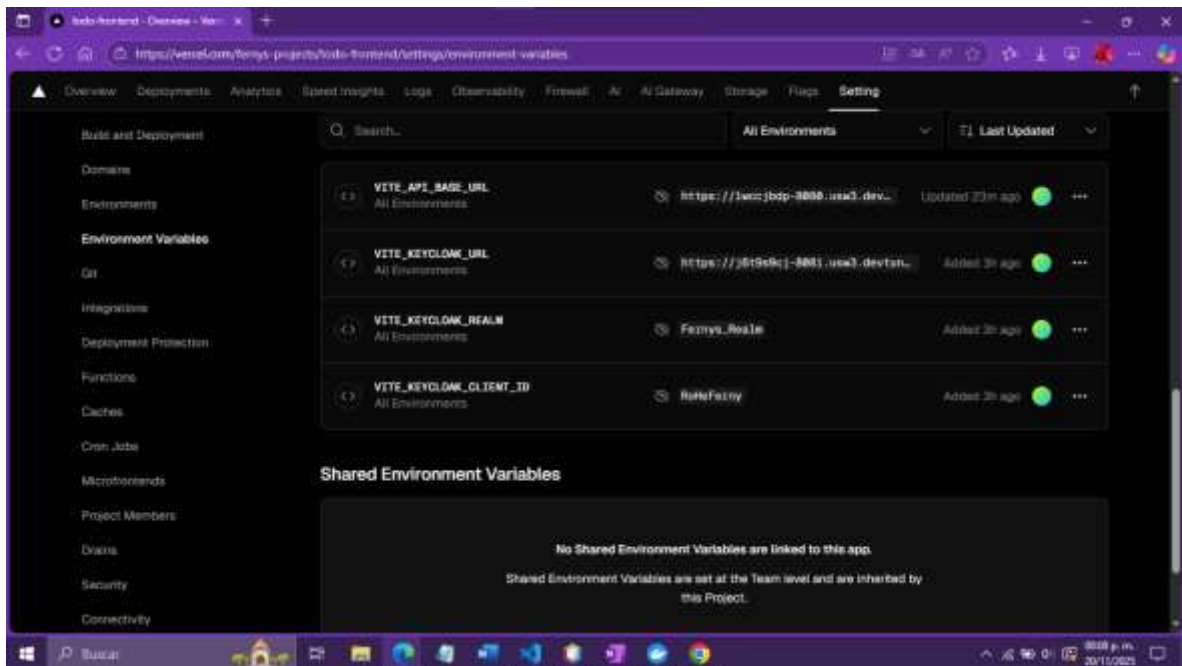
```
// =====
// CONFIGURACIÓN: URL base del backend
// =====
const API_BASE_URL = import.meta.env.VITE_API_BASE_URL || 'https://j6t9s9cj-8080.usw3.devtnnls.ms/api/v1';
```

Una vez teniendo todo lo anterior, lo siguiente es entrar a keycloak y cambiar las urls que se tenían (localhost) por las de vercel (para el dominio):

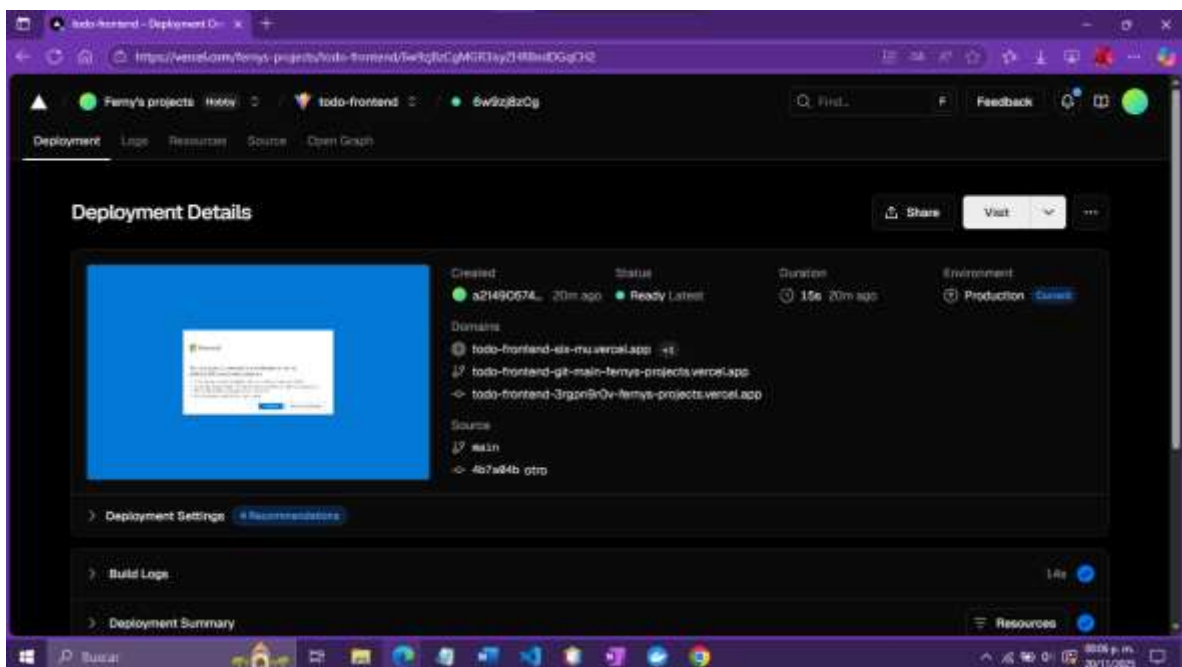


Y de igual manera, las variables de entorno que se especificaron en keycloak también las registraos en vercel (ya que son las principales para que funcione):

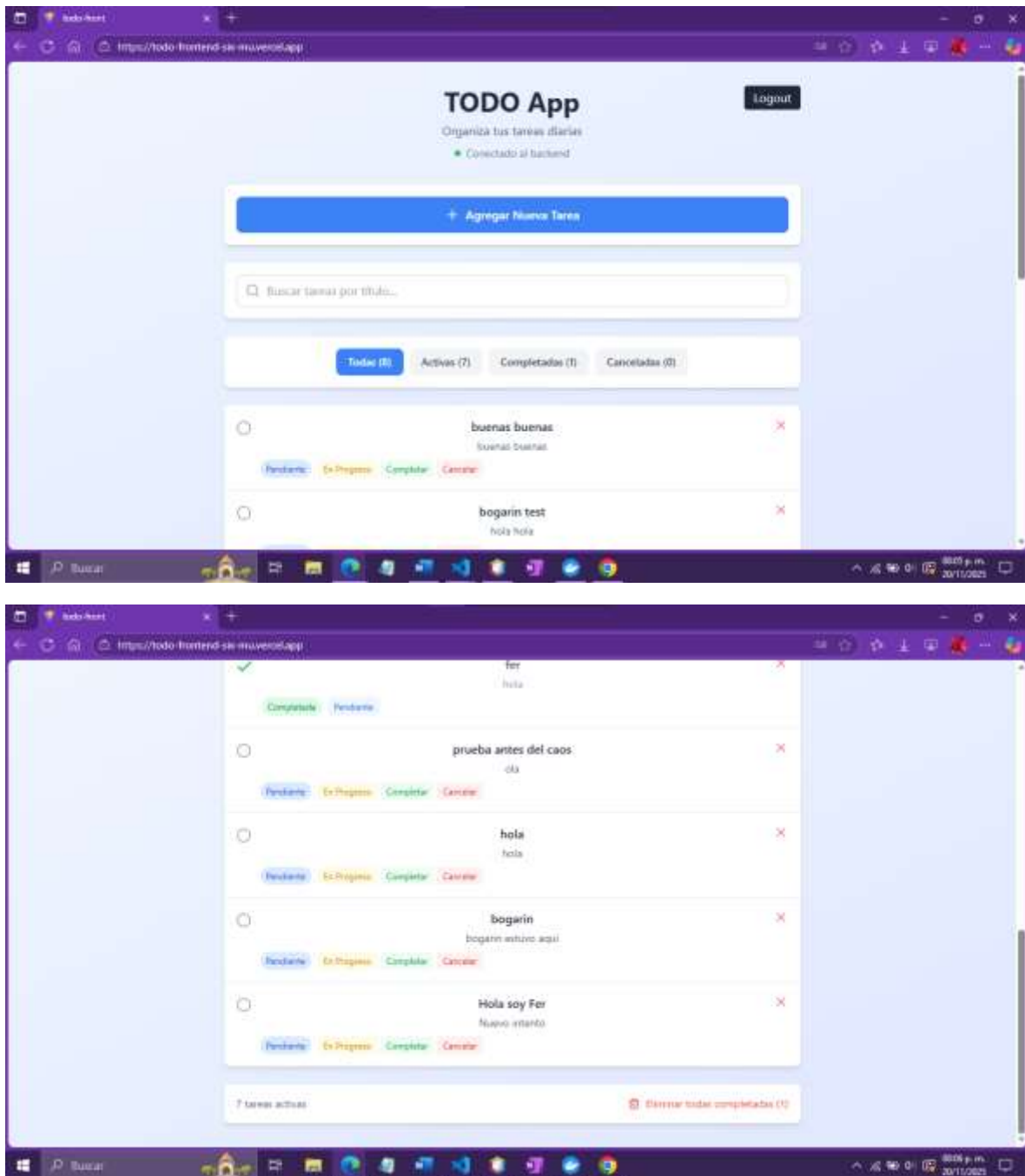




Y teniendo listo todo lo anterior, realizamos un deploy para aplicar los cambios e ingresamos a la url del dominio que se nos generó:



Y con eso se nos pide entrar con nuestro correo y contraseña de keycloak (previamente registrados) y ya podremos acceder a lo que es nuestra aplicación y registrar nuestras tareas:



**Nota importante:** la configuración de mi backend y frontend estaba bien, pero por cuestiones de mi máquina no se sincronizaban los cambios (por lo que no estaba haciendo lo que debería de manera correcta). Por lo cual se pidió ayuda a un compañero y lo que se hizo fue pasarle mis urls que me proporcionó vercel (para mi dominio) y las sustituyó, y de igual manera yo tuve que sustituir la url de mi puerto de backend (8080) por la de mi compañero (en donde se indicó anteriormente). Y ya con eso funcionó la aplicación.