



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 comunique correctamente con keycloak y la API del backend, para lo cual agregamos las siguientes:

VITE_KEYCLOACK_URL: es la URL de nuestro servidor de keycloak, en este caso en el puerto 8081.

VITE_KEYCLOACK_REALM: es el nombre de nuestro reino que creamos en keycloak.

VITE_KEYCLOACK_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.devtunnels.ms  
3  VITE_KEYCLOAK_REALM=Fernys_Realm  
4  VITE_KEYCLOAK_CLIENT_ID=RoHeFerny  
5  VITE_API_BASE_URL=https://j6t9s9cj-8080.usw3.devtunnels.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.devtunnels.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):

The screenshot shows the Keycloak Administration interface for the 'Ferrys_Realm'. The left sidebar lists various management sections: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, and Authentication. The main configuration area includes fields for Home URL (https://todo-frontend-six-muvercel.app), Valid redirect URLs (https://*vercel.app/*, https://todo-frontend-six-muvercel.app/*), Valid post logout redirect URLs (https://todo-frontend-six-muvercel.app/*, https://*vercel.app/*), Web origins (https://*vercel.app, https://todo-frontend-six-muvercel.app), and Admin URL (https://todo-frontend-six-muvercel.app). A 'Save' button is at the bottom. On the right, a sidebar titled 'Jump to section' lists General settings, Access settings (which is selected), Capability config, Login settings, and Logout settings.

Y de igual manera, las variables de entorno que se especificaron en keycloak tambi  n las registras en vercel (ya que son las principales para que funcione):

The screenshot shows the Vercel project settings for 'todo-frontend'. The left sidebar includes Overview, Deployments, Analytics, Speed insights, Logs, Observability, Firewall, AI, AI Gateway, Storage, Flags, and Settings (selected). Under Environment Variables, four variables are listed: VITE_API_BASE_URL (All Environments, updated 23m ago), VITE_KEYCLOAK_URL (All Environments, added 30 ago), VITE_KEYCLOAK_REALM (All Environments, added 30 ago), and VITE_KEYCLOAK_CLIENT_ID (All Environments, added 30 ago). Below this, a section for Shared Environment Variables states: 'No Shared Environment Variables are linked to this app.' and 'Shared Environment Variables are set at the Team level and are inherited by this Project.'

The screenshot shows the Vercel dashboard for the 'todo-frontend' project. The left sidebar lists various project settings like Overview, Deployments, Analytics, Shared Insights, Logs, Observability, Firewall, AI, AI Gateway, Storage, Flags, and Settings. Under the 'Environment Variables' section, there are four entries:

- VITE_API_BASE_URL: https://3wzjbdp-8888.usr3.dev.. Updated 20m ago.
- VITE_KEYCLOAK_URL: https://3wzjbdp-8888.usr3.dev.. Added 20m ago.
- VITE_KEYCLOAK_REALM: Femyo_Realm. Added 20m ago.
- VITE_KEYCLOAK_CLIENT_ID: RemoteFederate. Added 20m ago.

Below this, under 'Shared Environment Variables', it says: "No Shared Environment Variables are linked to this app. Shared Environment Variables are set at the Team level and are inherited by this Project."

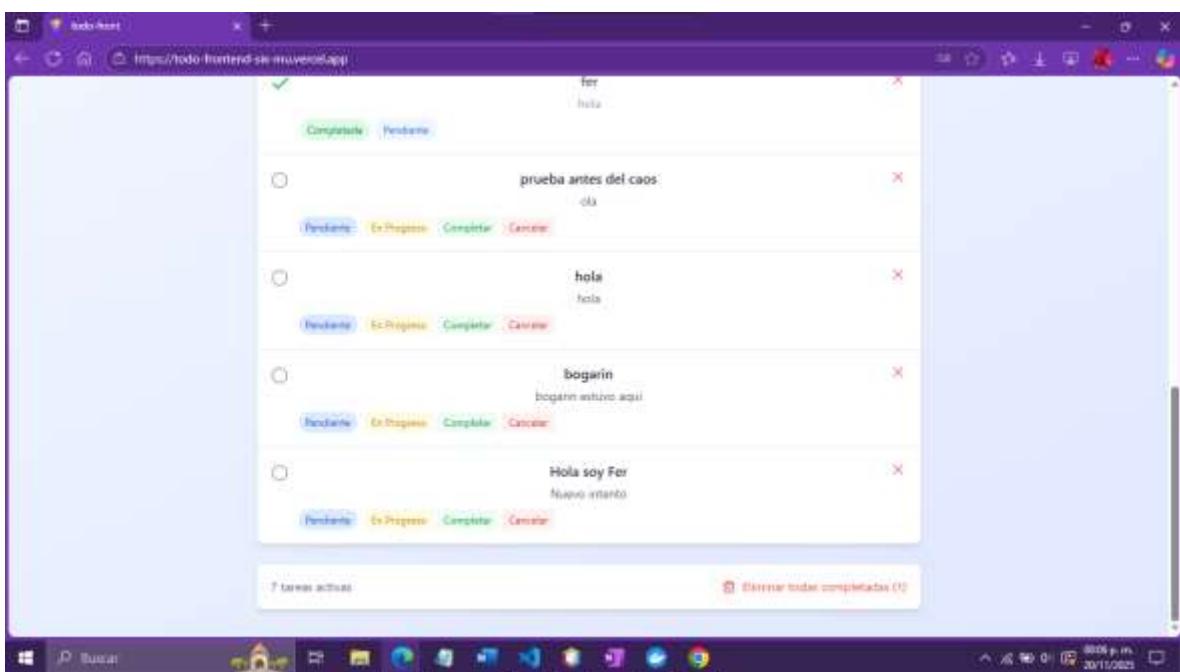
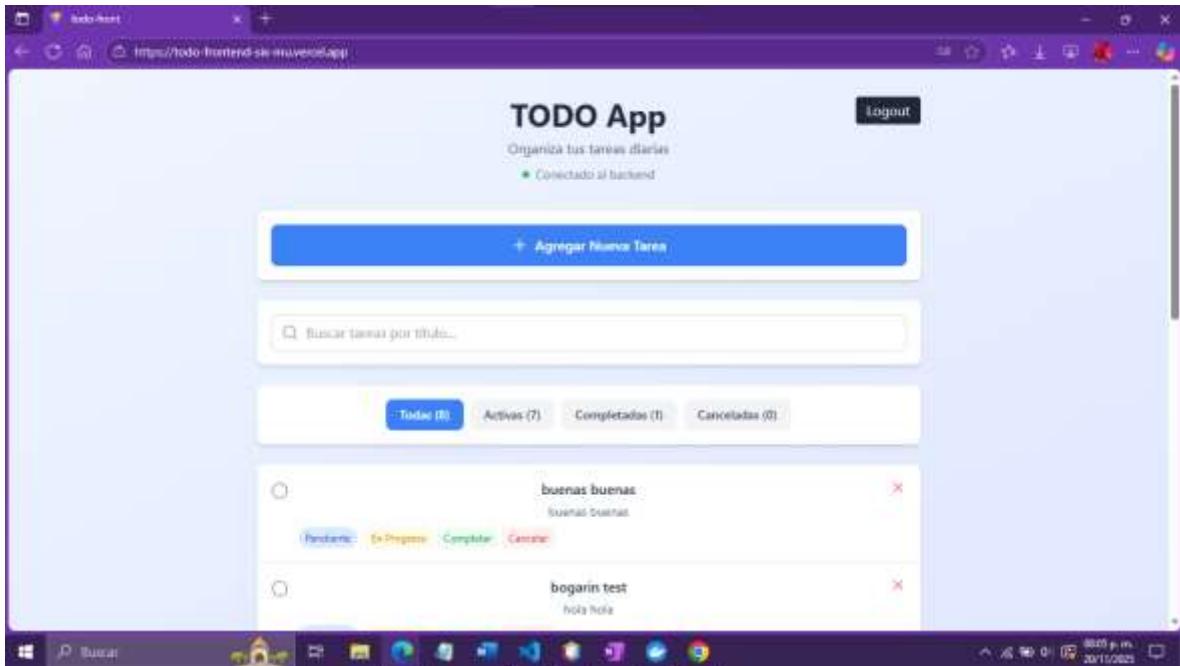
Y teniendo todo lo anterior, realizamos un deploy para aplicar los cambios e ingresamos a la url del dominio que se nos generó:

The screenshot shows the 'Deployment Details' page for the 'todo-frontend' deployment. It includes sections for Deployment, Logo, Resources, Source, and Open Graph. The deployment details are as follows:

- Deployment:** a21490674... (20m ago)
- Status:** Ready Latest
- Duration:** 15s 20m ago
- Environment:** Production
- Domains:** todo-frontend-81e-mu.vercel.app, todo-frontend-gt-main-femyo-projects.vercel.app, todo-frontend-3rgpn@Ov-femyo-projects.vercel.app
- Source:** main, -467a64b (trunk)

Below this, there are sections for Deployment Settings, Build Logs, and Deployment Summary. A 'Resources' button is also present.

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 pasarme 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.