

IMPLEMENTACIÓN

MANUAL DEL DESARROLLADOR

PATRONES ARQUITECTÓNICOS APLICADOS

La aplicación sigue una combinación de varias arquitecturas:

ARQUITECTURA DE APLICACIONES DE PÁGINA ÚNICA (SPA)

El frontend de la aplicación se ha desarrollado con React, siguiendo el patrón de Componentes Compuestos. En esta arquitectura, los componentes más grandes se componen de varios componentes más pequeños, creando una estructura de árbol.

ARQUITECTURA ORIENTADA AL DOMINIO (DDD)

Para el backend, hemos seguido la arquitectura DDD. El código se organiza en distintos paquetes para separar la lógica de negocio, los datos y la infraestructura.

INYECCIÓN DE DEPENDENCIAS

Hemos utilizado la Inyección de Dependencias, proporcionada por Micronaut, en todo el backend. Esta técnica permite una mayor flexibilidad, simplifica el código y facilita las pruebas unitarias.

PATRONES DE DISEÑO UTILIZADOS

Hemos utilizado varios patrones de diseño en la aplicación, estos han sido detallados en el capítulo de Diseño, bajo la sección de Patrones de diseño aplicados.

A continuación se listan los patrones de manera resumida:

- **Repository:** Este patrón nos permite encapsular la lógica de acceso a los datos, proporcionando un acceso uniforme a diferentes fuentes de datos.
- **Factory Method:** Este patrón se usa para manejar la creación de objetos, permitiendo que las subclases decidan qué clase instanciar.
- **Singleton:** Usamos este patrón para asegurarnos de que una clase tenga solo una instancia y proporcionar un punto de acceso global a ella.
- **Data Transfer Object (DTO):** Este patrón se usa para transferir datos entre procesos de software, en nuestro caso, entre la interfaz de usuario y los servidores.
- **Builder:** Este patrón nos permite construir objetos complejos paso a paso.
- **Adapter:** Este patrón nos permite convertir la interfaz de una clase en otra interfaz que los clientes esperan.
- **Strategy:** Este patrón se usa para seleccionar un algoritmo en tiempo de ejecución.

LENGUAJES UTILIZADOS

Las tecnologías y lenguajes que se han utilizado para desarrollar esta aplicación son:

BACKEND

- Java 17 LTS
- Micronaut 3.7.0
- PostgreSQL v15
- log4j 2.19
- JUnit 5 con AssertJ y Mockito.
- Liquibase 4

FRONTEND

- JavaScript ES2020
- React 18.2
- Axios 1.3
- dayjs 1.11
- React Oauth Google library 0.7
- Mui/material 5
- Mdb React 6.
- Entorno de ejecución y gestión de paquetes
 - Node 18
 - Npm 9.5

Además, es necesario tener instalado:

- Docker 24.0
- Git 2.39
- Gradle 7.5

PREPARACIÓN DEL ENTORNO DE DESARROLLO

Para preparar tu entorno de desarrollo se necesita:

- **Docker:** Se usa para levantar una instancia de PostgreSQL. Hay un archivo docker-compose en la carpeta resources del backend que facilita este paso.
- **Java 17 LTS:** Para el desarrollo del backend es necesario tener instalado un JDK de Java 17.
- **Node 18 y Npm 9.5:** Para el desarrollo del frontend y la gestión de las dependencias.
- **Gradle:** Para la gestión de las dependencias del backend.
- **IntelliJ IDEA Ultimate:** Como editor de código. Aunque puedes utilizar cualquier editor de tu preferencia, te recomendamos IDEA por su integración con Gradle y su soporte para el desarrollo con React. Como alternativa se puede usar eclipse para el backend y VS Code para el frontend.
- **IntelliJ DataGrip:** Para la gestión de la base de datos.
- **Git 2.39:** Como herramienta de gestión de versiones.

VARIABLES DE ENTORNO

Es necesario configurar una serie de variables de entorno tanto en el backend como en el frontend. Para ejecutar el backend en local no será necesario nada más que especificar que se ha de cargar el fichero de configuración application-local.yml, que está preparado para comunicarse con la instancia de PostgreSQL inicializada por el fichero docker-compose.

Para producción, se han de especificar las siguientes variables de entorno en el caso del backend:

- **GOOGLE_CLIENT_ID:** Aquí deberá ir el client ID de Google Cloud que se usará en la comunicación con las APIs de Google.
- **GOOGLE_CLIENT_SECRET:** Secreto de Google Cloud que será usado para comunicarse con las APIs de Google.
- **DB_URL:** URL de conexión de tipo JDBC para conectarse a la base de datos.
- **DB_USERNAME:** Nombre de usuario que se usará en la conexión a base de datos.
- **DB_PASSWORD:** Contraseña del usuario que se usará en la conexión a base de datos.

En el caso del frontend, tanto en local como en producción, hay que especificar las siguientes variables de entorno:

- **REACT_APP_BACKEND_HOST:** URL donde se aloja el backend en el formato {HTTP/HTTPS}://{URL}:{PORT}

MANUAL DE DESPLIEGUE

Este manual proporciona una guía paso a paso para desplegar la aplicación en un entorno de producción.

REQUISITOS

Servidor con Java 17 LTS instalado.

Base de datos PostgreSQL v15.

Docker instalado en el servidor.

Node.js v18 y Npm 9.5 instalados en el servidor.

PASOS PARA EL DESPLIEGUE

1. **Clonar el repositorio del proyecto:** En el servidor de producción, clona el repositorio del proyecto utilizando el comando git clone.
2. **Configurar variables de entorno:** Configura las variables de entorno necesarias para la aplicación, listadas en el manual de desarrollador.
3. **Construir y ejecutar la imagen de Docker para la base de datos PostgreSQL:** Utiliza el archivo docker-compose que se encuentra en la carpeta del backend para levantar una instancia de PostgreSQL. Puedes hacer esto con el comando docker-compose up.
4. **Instalar dependencias del backend:** Navega a la carpeta del backend y ejecuta ./gradlew build para instalar las dependencias necesarias, pasar los tests y construir el proyecto.
5. **Iniciar la aplicación backend:** Una vez construido el proyecto, se puedes iniciar la aplicación backend con el comando:
 - a. `java -jar build/classes/java/main/com/dfernandezaller/Application.class.`
6. **Instalar dependencias del frontend:** Navega a la carpeta del frontend y ejecuta npm install para instalar las dependencias del proyecto.
7. **Construir el proyecto del frontend:** Ejecuta el comando npm run build. Esto creará una versión de producción de la aplicación frontend en la carpeta build.
8. **Desplegar la aplicación frontend:** Según el servidor donde se desplegará la aplicación, los pasos pueden variar. En general, deberás copiar los contenidos de la carpeta build en la carpeta correspondiente del servidor, generalmente '/public'.

VERIFICACIÓN DEL DESPLIEGUE

Para verificar que la aplicación se ha desplegado correctamente, se ha de:

1. Navegar a la URL de tu aplicación frontend y comprobar que la interfaz de usuario se carga correctamente.
2. Realizar pruebas de conexión a la base de datos y comprobar que las tablas se han creado de manera adecuada.
3. Conectarse a los servicios del backend para asegurarte de que todo funciona como se espera. Si se va a la URL del backend debería aparecer un mensaje notificando que el servidor está listo para recibir peticiones.

MANUAL DE USUARIO

INTRODUCCIÓN

Este manual tiene como objetivo guiar a los usuarios a través de las funcionalidades y características de NoMoreMeetings. Aquí encontrará instrucciones paso a paso sobre cómo utilizar cada característica, junto con capturas de pantalla para ilustrar cada paso.

REGISTRO Y ACCESO

Cuando accedas a la aplicación, te encontrarás con la página de inicio. Aquí podrás iniciar sesión si ya tienes una cuenta, o registrarte si eres un nuevo usuario.

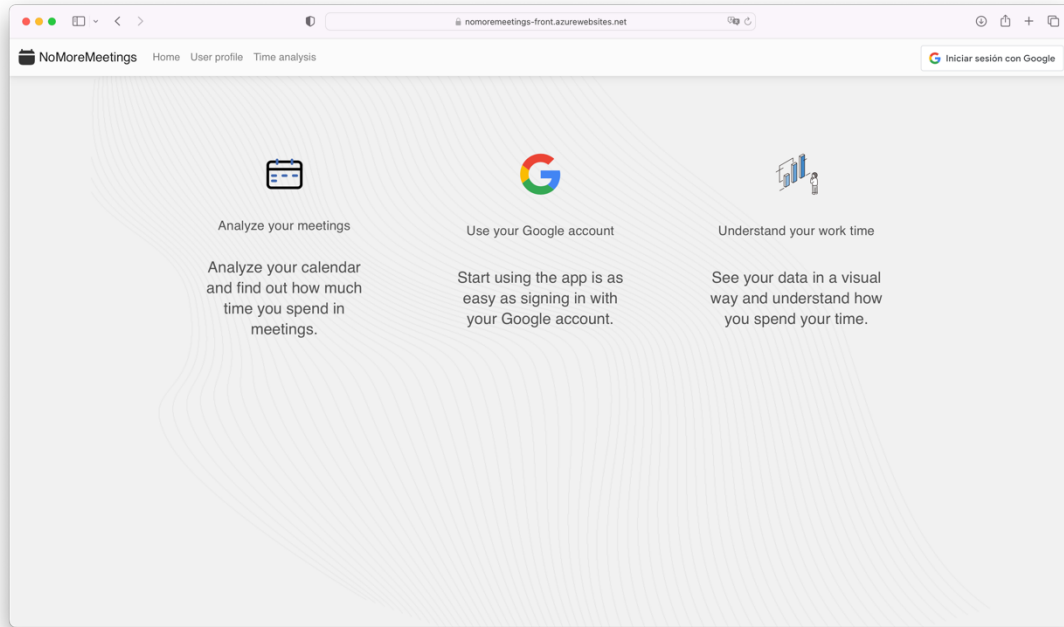


Figura <TBD>: Página de inicio de NoMoreMeetings

INICIO DE SESIÓN

Para proceder al inicio de sesión, se ha de hacer clic en el botón superior izquierdo que dice: Iniciar sesión con Google. Esto te llevara a una página de inicio de sesión de Google donde deberás seleccionar que cuenta quieres usar, y deberás iniciar sesión. Es posible que aparezca un cuadro de selección.

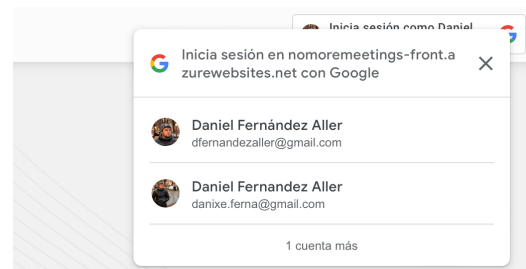
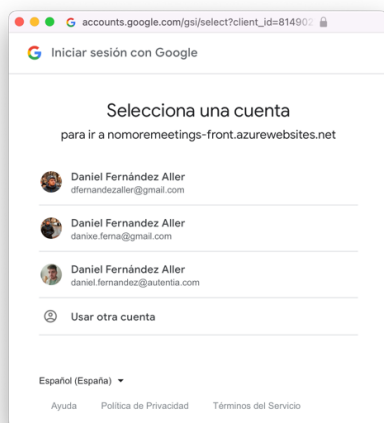


Figura <TBD>: Cuadro de selección de cuenta para iniciar sesión

Figura <TBD>: Página de selección de cuenta

REGISTRO

Para proceder al registro se ha de seguir los mismos pasos que para el inicio de sesión. Una vez que se hayan completado el usuario verá que una alerta informativa al final de la página informa de la necesidad de registro y el botón de inicio de sesión ha cambiado a un botón de registro. Es posible que, en función de la configuración del navegador del usuario, aparezca tras el inicio de sesión directamente una venta de Google para proceder con el registro.

Una vez aparezca la ventana de registro de Google, bien por presionar el botón de registro o bien por que aparezca de forma automática, se ha de dar los permisos necesarios a NoMoreMeetings para acceder al calendario del usuario y ya se habrá completado el registro.

Es posible que aparezca una alerta informando de que la aplicación no está verificada, en ese caso se ha de presionar en 'Mostrar configuración avanzada' y acto seguido en 'Ir a nomoremeetings...'

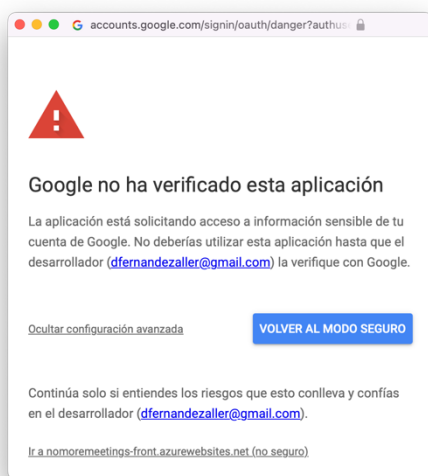


Figura <TBD>: Página de alerta de Google de aplicación no verificada

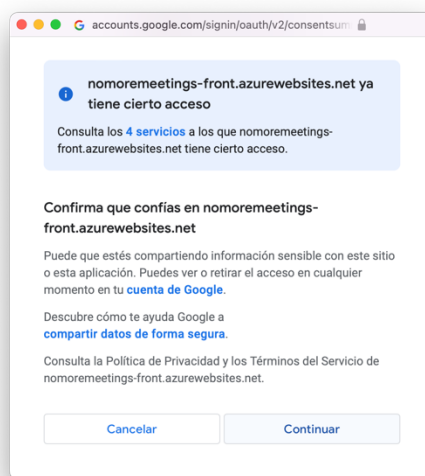


Figura <TBD>: Página de concesión de permisos de Google

PERFIL DEL USUARIO

En esta página el usuario puede ver toda la información relativa a su perfil de usuario en la aplicación así como modificar la configuración del análisis del calendario. Para modificar la configuración relativa al análisis se dispone de tres cuadros donde se puede:

- Seleccionar el tiempo en minutos entre reuniones que contara como tiempo de reunión.
- Seleccionar el número de días previos al actual a analizar.
- Seleccionar el calendario del usuario que se analizará.

Además de estas opciones hay un botón dedicado a poder cambiar el horario laboral del usuario así como su horario de almuerzo. Es durante el periodo de trabajo que se analizara el tiempo dedicado a reuniones del usuario. Al pulsar el botón azul 'Edit working hours' se mostrará una ventana modal que permitirá editar el horario laboral del usuario.

Cualquier cambio en la configuración será guardado cuando se presione 'Save changes'.

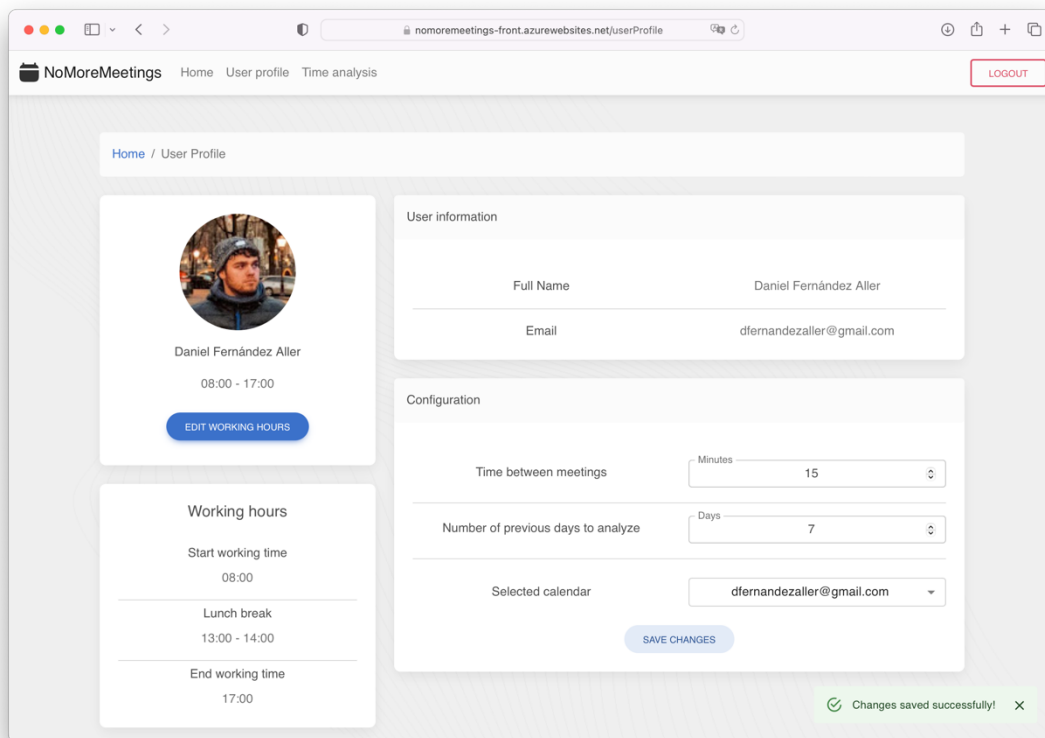


Figura <TBD>: Página de perfil del usuario y configuración

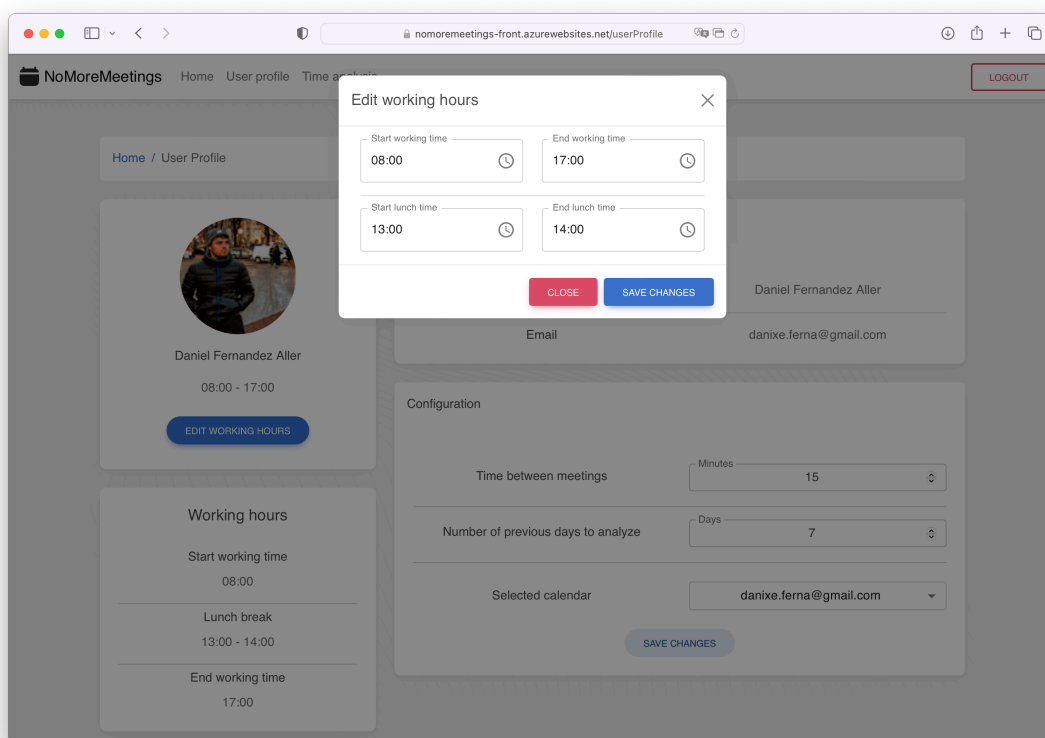


Figura <TBD>: Cuadro modal de modificación del horario laboral

ANÁLISIS DEL TIEMPO

En esta página se pueden visualizar, en un formato de tabla y en una gráfica semi-circular, los datos relativos al tiempo dedicado a las reuniones por el usuario de acuerdo con la configuración establecida en la vista anterior.

El usuario no ha de hacer nada para poder visualizar los datos, es posible que aparezca una pantalla de carga si los datos tomaran mucho tiempo para ser calculados, pero en cualquier caso no se requiere de interacción del usuario.

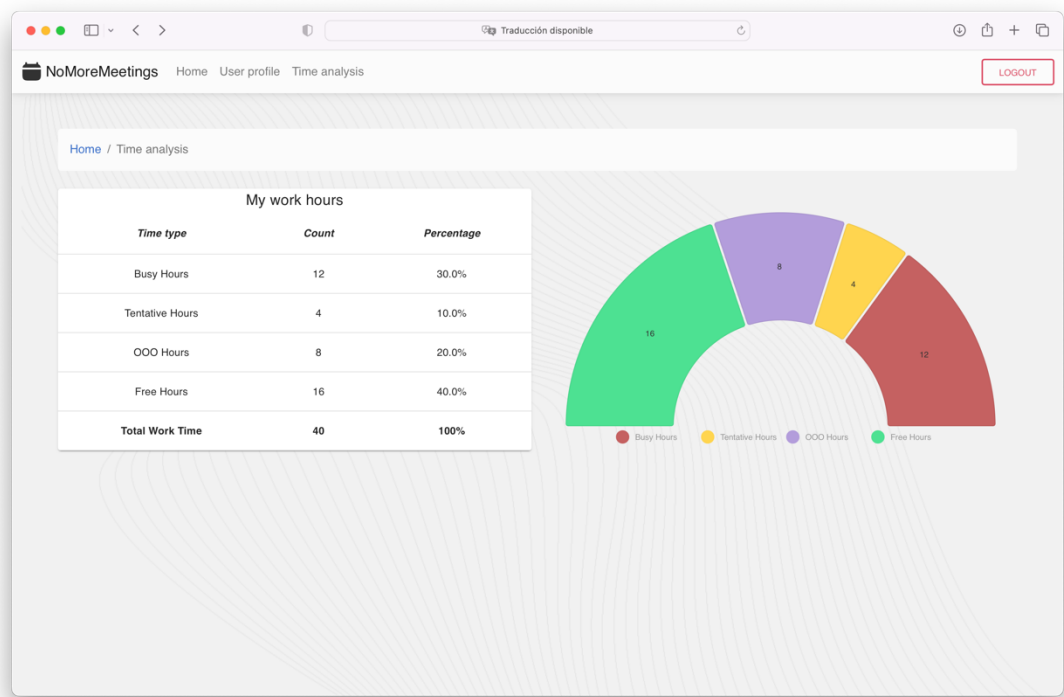


Figura <TBD>: Página de resultados del análisis del calendario del usuario