

## ASPECTOS TEÓRICOS

### CONCEPTOS IMPORTANTES

#### AUTENTICACIÓN Y REGISTRO CON GOOGLE

Se contempla la implementación de un proceso de autenticación y registro basado en el uso de la cuenta de Google del usuario. Este proceso tendrá dos ventajas significativas: por un lado, proporcionará un mecanismo de autenticación seguro y de confianza, y por otro, permitirá el acceso a los datos del calendario de Google del usuario, necesarios para el funcionamiento de la aplicación.

#### OAUTH2

El protocolo que se utilizará para el proceso de autenticación y autorización es OAuth2, un estándar ampliamente aceptado y utilizado para permitir la autorización segura en aplicaciones web, móviles y de escritorio. Este protocolo facilitará un “token de acceso” junto con un “token de refresco” que habrán de ser persistidos y que se utilizarán para autenticar las solicitudes a la API de Google Calendar.

#### ANÁLISIS DE LOS DATOS DEL CALENDARIO

Se llevará a cabo el análisis de los datos extraídos del calendario del usuario. Este análisis implicará el procesamiento de los eventos del calendario para calcular métricas relevantes, como el porcentaje de tiempo total gastado en reuniones, así como una categorización en base al tipo de reuniones, proporcionando así una visión clara de cómo se distribuye el tiempo de trabajo del usuario.

#### API DE GOOGLE CALENDAR

Para acceder a los datos del calendario, se utilizará la API de Google Calendar. Esta interfaz permitirá a la aplicación extraer los datos del calendario del usuario de una manera estructurada y segura.

#### PERSONALIZACIONES DE USUARIO

Se ha de permitir a los usuarios personalizar el análisis de su tiempo. Los usuarios podrán definir aspectos como su horario de trabajo, las horas de comida y el tiempo mínimo entre reuniones que se considerará como tiempo de reunión. Estos ajustes habrán de ser persistidos, de modo que las preferencias del usuario se conserven de cara a todos los análisis futuros que se realicen.

## TECNOLOGÍAS UTILIZADAS

A lo largo del desarrollo del proyecto, se utilizarán diversas tecnologías para cubrir diferentes aspectos del sistema. A continuación, se proporciona una descripción breve de cada una de ellas:

### BACKEND

- **Java 17:** Es el lenguaje de programación que se utilizará para el desarrollo del backend.
- **Micronaut 3.7.0:** Este es el framework que se utilizará para el desarrollo del backend.
- **PostgreSQL 15:** Es el sistema de gestión de bases de datos que se utilizará para el almacenamiento y gestión de datos.
- **Liquibase:** Se utilizará para la gestión de la base de datos, permitiendo un fácil mantenimiento y evolución del esquema de la base de datos.
- **JUnit 5, AssertJ y TestContainers:** Estas son las librerías que se utilizarán para la realización de tests unitarios y de integración, incluyendo aquellos que requieran de una base de datos.

### FRONTEND

- **ECMAScript 2021:** Es la versión de JavaScript utilizada para el desarrollo del frontend.
- **React 18:** Es el framework que se utilizará para el desarrollo del frontend.
- **NPM 9.5:** Es la herramienta utilizada para la gestión de dependencias y la construcción del proyecto.
- **Axios 1.3:** Es la librería que se utilizará para la realización de peticiones HTTP desde el frontend.
- **MDB React 6 y MUI 5.12:** Son las librerías que se utilizarán para la creación de componentes de la interfaz de usuario.
- **Dayjs 1.11:** Es la librería que se utilizará para el manejo de fechas en el frontend.

### HERRAMIENTAS DE DESARROLLO Y DESPLIEGUE

- **Git:** Es el sistema de control de versiones que se utilizará durante el desarrollo del proyecto.
- **GitHub:** Se utilizará como repositorio remoto durante el desarrollo del proyecto. Además, se utilizará GitHub Actions para la integración continua.
- **Azure:** Es el proveedor de servicios cloud que se utilizará para el despliegue de la aplicación. En él se desplegarán tanto el backend como el frontend, y también se alojará la instancia de PostgreSQL que se utilizará en producción.
- **Docker:** Se utilizará para el despliegue de la aplicación, tanto frontend como backend.
- **Qodana:** Es una herramienta de análisis de calidad del código de JetBrains que permite obtener reportes de calidad para detectar problemas, reportar sugerencias y detectar 'code smells'.
- **Google Calendar API:** Se utilizará para la extracción de datos del calendario de Google del usuario.

- **OAuth2:** Se utilizará como protocolo para la autenticación y autorización de los usuarios.
- **Gradle:** Se utilizará para la gestión de dependencias y la construcción del proyecto.