DISEÑO SISTEMA DISTRIBUIDO "PROJECT HUB"

Grupo 2

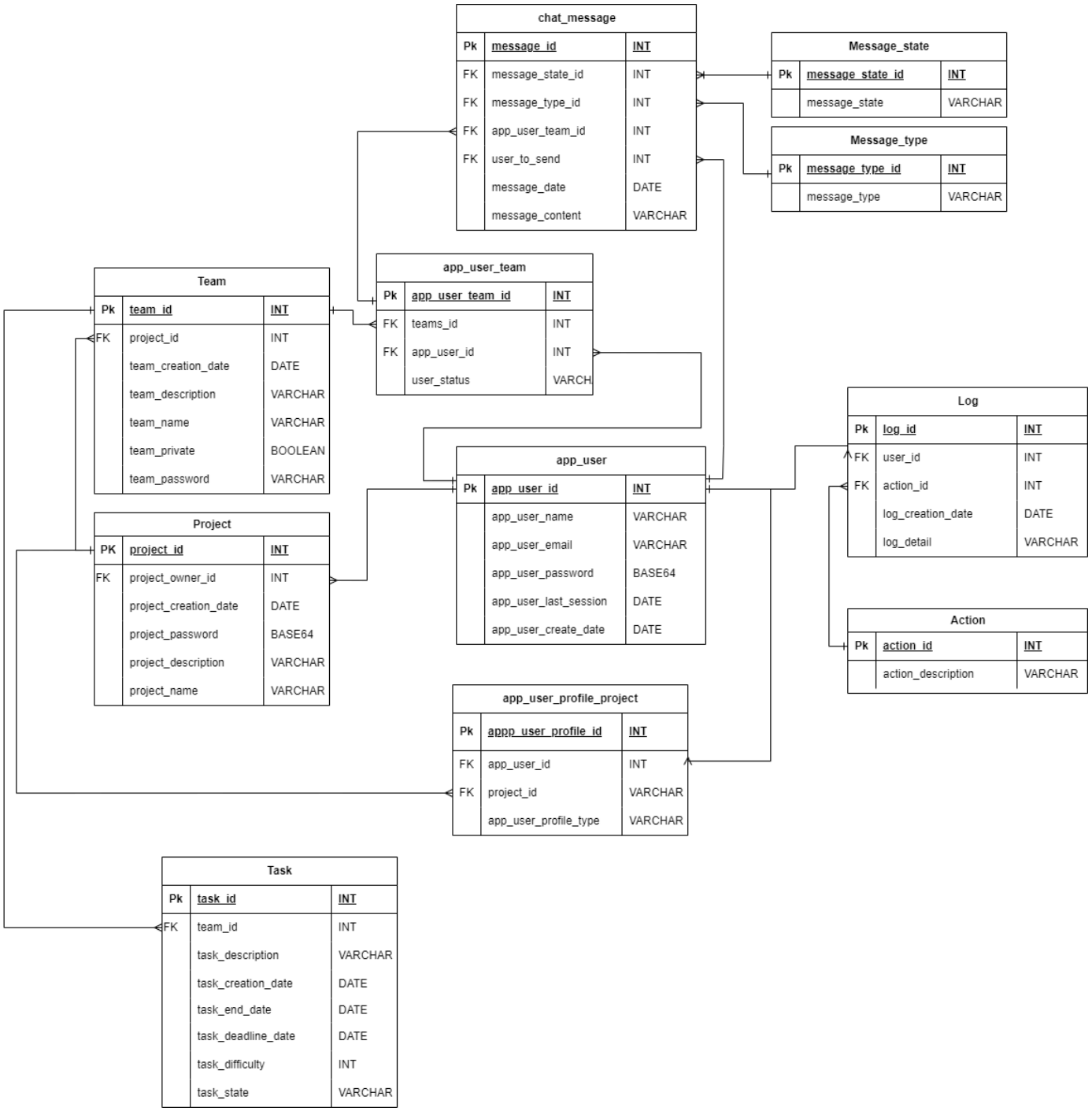
INFO288 - Sistemas distribuidos

Instituto de Informática, Universidad Austral de Chile.

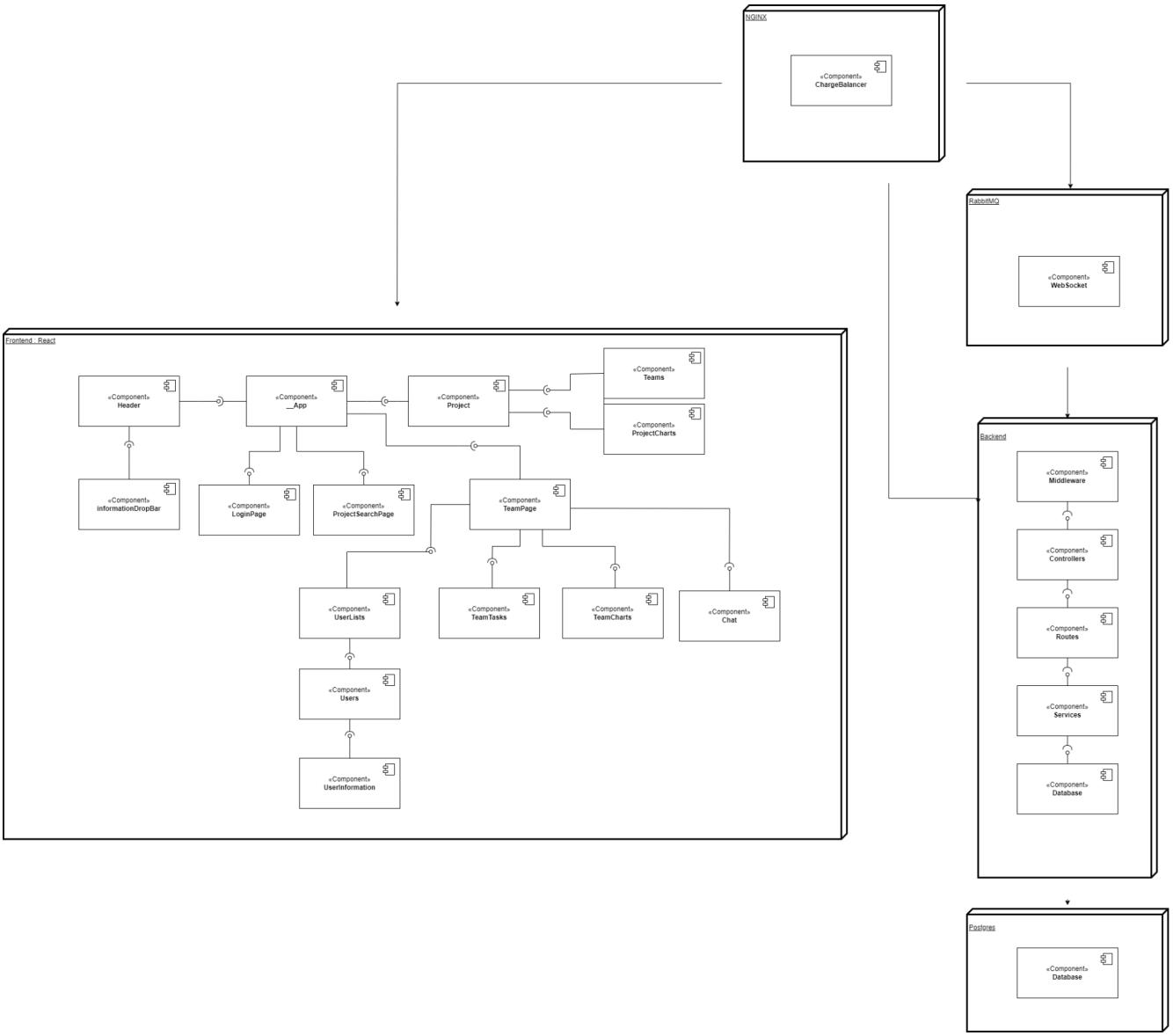
Kevin Medina - Hernán Cornejo - Felipe Prieto

Arquitectura

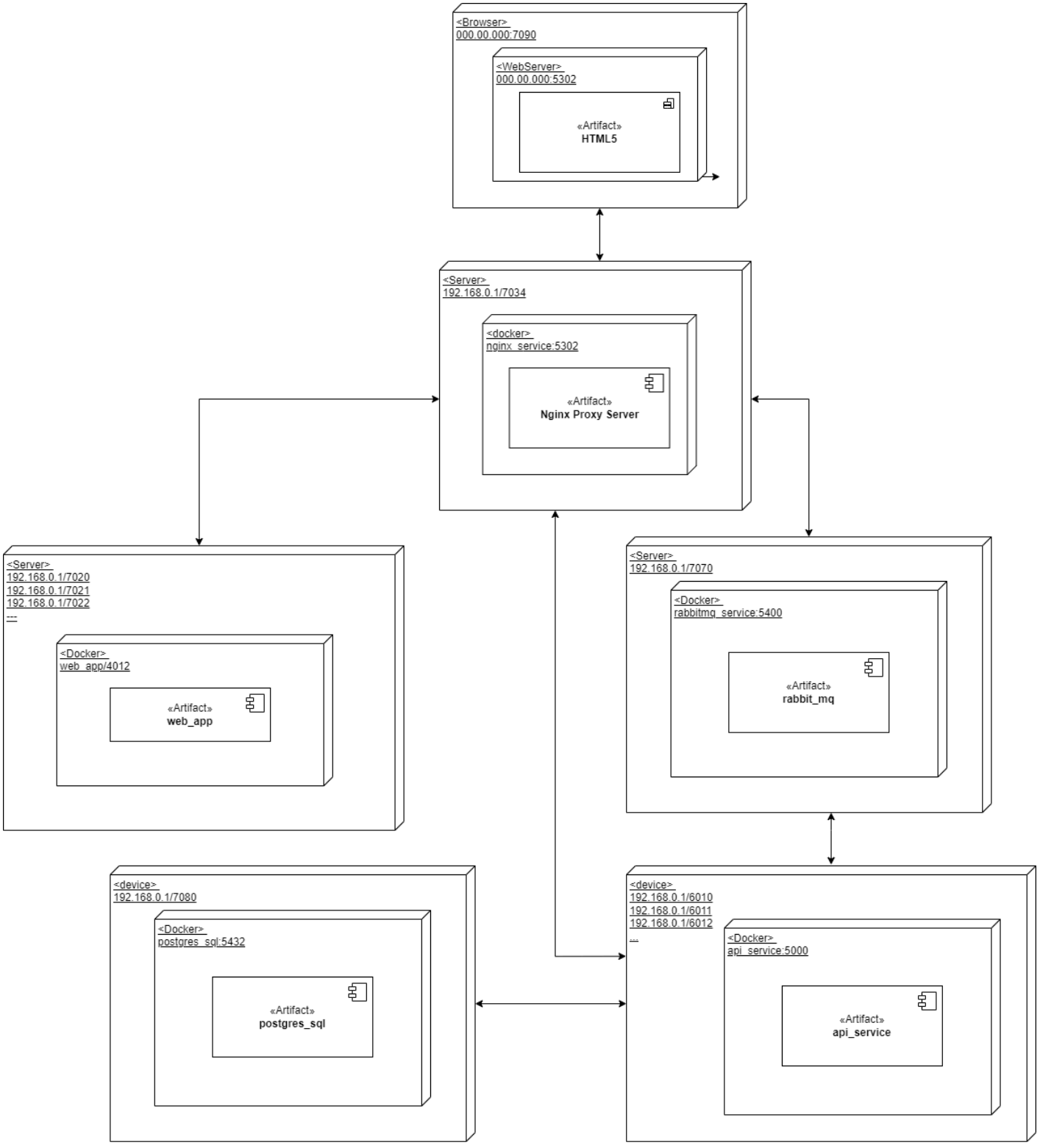
- **Modelo Relacional**



- **Diagrama de componentes**



- **Diagrama de despliegue**



Componentes

Lenguaje de programación

- **TypeScript (5.4.5)**: Extensión de JavaScript la cual añade sistemas de tipos estáticos, permitiendo una detección temprana en errores de código, siendo empleado para la creación del frontend, la capa visible al usuario del sistema.
- **Python (3.10.14)**: Utilizado para la parte lógica del sistema, gestionando la interacción con la base de datos y servicios necesarios. Python tiene una sintaxis la cual facilita la lectura, escritura y comprensión del código, permitiendo desarrollar a mayor velocidad.

Herramientas

- **NPM (20.9.0):** Herramienta para gestionar paquetes en el ecosistema de Node.js. Siendo una de sus ventajas principales la facilidad de uso, simplificando la gestión de dependencias en un proyecto.
- **Vite (5.2.9):** Bundler que destaca por su velocidad en el desarrollo, permitiendo el precompilar módulos a tiempo real y el esquema de módulos nativos con JavaScript.
- **Nginx (1.25.5):** Servidor web de alto rendimiento y ligero, el cual puede actuar como balanceador de carga y cache del servidor. Permite manejar altos volúmenes de conexiones concurrentes de forma eficiente.
- **Docker(26.0.1):** Plataforma que permite la creación, despliegue y ejecución de aplicaciones. Se utilizará para empaquetar las componentes de software, permitiendo escalar el sistema y hacerlo más eficiente.

Componentes de software

- **_APP:** Componente principal del frontend, lleva toda la parte lógica de la gestión de la web y donde se construyen todos los demás componentes.
- **Header:** Sección superior de la web que contiene el logotipo de la aplicación y un botón relacionado con la cuenta del usuario
- **InformationDropBar:** Listado con las opciones relacionadas con la cuenta de usuario.
- **LoginPage:** Página de inicio de sesión en base a credenciales.
- **ProjectSearchPage**: Página que permite buscar un proyecto existente en base al id del proyecto en cuestión.
- **ProjectPage:** Página principal que se visualiza al entrar a un proyecto específico.
- **Teams**: Listado de equipos relacionados al proyecto.
- **ProjectCharts:** Información respecto a las tareas y progreso general del proyecto
- **TeamPage:** Página principal respecto al equipo seleccionado.
- **UserList:** Listado de usuarios respecto al equipo seleccionado.
- **Users:** Componente perteneciente al listado de usuarios, que permite visualizar al usuario mediante un nombre, foto y estado en el que se encuentra (trabajando, colación, etc).
- **UserInformation:** Información que se ve del usuario cuando se cliquea dentro de la lista.
- **ChargeBalancer:** Componente que se encarga de establecer el cómo se distribuye el acceso a la aplicación del sistema, siendo esta controlada por NGINX

- **WebSocket:** Componente encargado de la comunicación bidireccional entre el usuario y el servidor, permitiendo una comunicación a tiempo real eficiente y escalable en el sistema. Se utilizará RabbitMQ (3.13)
- **Controllers:** Componente que se encarga de recibir la solicitud del cliente y luego estas procesarlas para devolver la respuesta adecuada a su petición.
- **Services**: Encapsulan la lógica del negocio, permitiendo tareas específicas como lo pueden ser el interactuar con la base de datos.
- **Middleware**: Componente de comprobación y validación de datos, permite mantener la integridad y seguridad del sistema.
- **Models**: Componente que define la estructura de los datos que interactúan entre la base de datos y los datos recibidos.
- **Database**: Componente que se encarga de controlar la conexión con la base de datos.

Bibliotecas:

- **FastApi (0.111.0):** Framework centrado en la simplicidad y facilidad, permitiendo crear aplicaciones web de forma rápida, siendo ideal para proyectos que necesitan un desarrollo rápido.
- **React (18.2.0):** Biblioteca que se utiliza para la construcción de interfaces interactivas y dinámicas. Su organización se basa en el concepto de componentes reutilizables, permitiendo desarrollar interfaces completas en componentes pequeños y manejables.

¿ Se utiliza software libre o de pago?

- Por motivo de escasez de recursos, se debe optar por una versión de software libre, permitiendo gestionar los recursos ahorrados a la infraestructura para desplegar el sistema.

Ventajas:

- Permite gestionar recursos en otras áreas.
- Los sistemas open source escogidos son ampliamente utilizados en la industria, generando en consecuencia una gran cantidad de documentación y guías de uso para adaptar fácilmente esta tecnología.

Desventajas:

- El soporte en caso de fallas con el proveedor de la herramienta puede ser acotado en comparación a alguna ofrecida por un proveedor específico. Resultando en tiempos de resolución de sistemas más amplios
- Actualizaciones menos frecuentes, generando problemas de compatibilidad

¿ Se utilizará software desarrollado por nosotros, híbridas?

- Se utilizaran herramientas híbridas, ya que el alcance del proyecto no propone algo trivial, añadido el tiempo de desarrollo que se tendrá. Por lo que el optar por esta opción permitirá centrar las operaciones en otras áreas.

Ventaja:

- Permite avanzar de forma más agilizada el desarrollo.

Desventaja:

- Se debe tener bien estructurado el proyecto para poder adaptarlo a software ya existente

Modelo Físico

- Para la realización de pruebas el sistema será desplegado de forma local dentro de contenedores, teniendo una máquina con las siguientes capacidades.
    - 64GB Ram
    - Windows 11 Pro
    - Procesador: AMD Ryzen 9 5900X 12-Core (3.7GHz)
    - Gráfica: NVIDIA GeForce RTX 3070 (8018 Vram)
    - Disco duro: 1 Tera SSD NVME PCI 4.0
    - Conectividad: 500gb bajada/subida
    - Latencia 1 Ping

Modelo Fundamental

Dado que trabajamos con datos sensibles como sería la elaboración de proyectos, donde la confiabilidad y seguridad es un tema crucial por lo que para garantizar la seguridad y robustez del sistema, implementaremos una serie de prácticas fundamentales que abordan diversos aspectos, desde la protección de datos sensibles hasta la sincronización y la colaboración en tiempo real.

Para el caso de la **seguridad:**

- **Encriptación de Sesiones:** Utilizaremos técnicas de encriptación para proteger las sesiones de proyectos, asegurando que la información transmitida entre el cliente y el servidor esté cifrada y segura.
- **Tokens de Acceso:** Implementaremos tokens de acceso en nuestra API para autorizar el acceso a recursos protegidos, garantizando que solo los usuarios autorizados puedan interactuar con la aplicación y acceder a los datos sensibles.
- **Firewall y Control de Puertos:** Configuraremos un firewall para controlar el tráfico de red y bloquear el acceso no autorizado a los puertos del servidor. Los puertos de la base de datos estarán bloqueados para todas las direcciones IP excepto las autorizadas, lo que aumentará la seguridad del sistema y protegerá la integridad de los datos.

y para el caso de la **sincronización y comunicación:**

- **Middleware para Servicios de Mensajes:** Implementaremos middleware para proporcionar servicios de mensajes entre aplicaciones, permitiendo la comunicación eficiente y segura entre los distintos componentes del sistema. Este middleware también facilitará la comunicación a través de la red, garantizando la sincronización adecuada de los datos y la disponibilidad del sistema.
- **API Restringida por IP:** Configuraremos la API para que solo acepte solicitudes de ciertas direcciones IP autorizadas, lo que restringirá el acceso a la plataforma y protegerá contra posibles ataques maliciosos.
- **Políticas CORS:** Estableceremos políticas CORS para controlar el acceso desde dominios específicos, evitando ataques de tipo cross-site scripting (XSS) y protegiendo la seguridad del navegador del usuario.

Servicios

Encriptación:

- Para asegurar la encriptación de sesiones y datos sensibles, podemos utilizar bibliotecas y herramientas de encriptación como OpenSSL o Libsodium.

Firewall y Seguridad de Red:

- Para configurar y gestionar el firewall y la seguridad de red, podemos utilizar servicios de seguridad en la nube como AWS Security Groups o Azure Network Security Groups.

Backup y Recuperación de Datos:

- Para asegurar la disponibilidad y la recuperación de datos en caso de desastres, podemos utilizar servicios de backup y recuperación como AWS Backup o Google Cloud Storage. Estos servicios nos permiten realizar copias de seguridad regulares de los datos y establecer planes de recuperación para restaurar la información en caso de fallos o pérdidas.

Preguntas al cliente

1. **¿Cuál es el sistema operativo con el que contarán los servidores?**
   R: Windows Server 2019 Standard Edition
2. **¿Cuáles son las características con las que cuentan los servidores?**
   R: Intel Xeon E-2224G 4 núcleos y 8 hilos.
   16GB de RAM DDR4 ECC
   (SSD) de 1TB
3. **¿Existe algún requisito con los lenguajes de programación a utilizar en el sistema?**
   R: Sí, para el frontend se deberá utilizar Typescript + React, y para el backend python.

4. **¿Existen integraciones planificadas con sistemas externos que debamos tener en cuenta durante el desarrollo?**
   R: Por el momento, no se consideran integraciones con sistemas externos.

5. **Respecto al balanceador de carga, ¿Considera necesario crear uno personalizado?**
   R: Por las necesidades del sistema, y el poco tiempo para desarrollar la solución, no será necesario invertir tiempo extra en uno personalizado. Se permitirá utilizar balanceadores de carga ya existentes, como Nginx.

6. **¿Hay algún requisito particular en términos de seguridad y privacidad de la información?**
   R: Sí, los datos de nuestros usuarios deberán ser encriptados y el sistema deberá contar con estándares de seguridad básicos, como el uso de variables de entorno y protección de la base de datos.

7. **¿Qué sistemas o herramientas externas utilizan ACTUALMENTE para la gestión de proyectos y equipo? ¿Hay alguna característica de estos que se deba considerar para nuestra aplicación?**
   R: Actualmente, utilizamos Docs y Sheets de google para gestionar proyectos, y Microsoft Teams para el equipo. El problema es que necesitamos algo más personalizado, que permita monitorear todo desde una sola aplicación y además permita crear artefactos de metodologías ágiles, como mostrar un backlock de actividades, realizar seguimiento de tareas, roles asignados, etc.

8. **¿Cuáles son los requisitos de disponibilidad y tiempo de respuesta para el sistema? ¿Hay algún horario de mantenimiento o ventana de tiempo en la que el sistema no debería estar disponible?**
   R: El sistema deberá estar disponible mientras sea posible, durante todo el día. En la madrugada, existirá un horario en el cual el sistema se podrá actualizar. De momento, no contamos con una hora exacta, pero lo más posible es que deba ser entre las 4 y 5 de la madrugada. Con respecto a los tiempos de respuesta, no deberían ser mayores a 3 segundos al mostrar información o realizar una acción.

9. **¿Cómo se realiza actualmente el seguimiento del progreso de los proyectos y el rendimiento de los equipos en su organización? ¿Hay métricas o indicadores clave que debamos incluir en el sistema?**
   R: Actualmente, utilizamos KANBAN para seguir el progreso de los proyectos y una tabla DO-DOING-DONE con los equipos. No contamos con métricas ni indicadores, pero esperamos que con el sistema podamos incluirlas. Algunas de las más importantes son el porcentaje completado de las tareas, la velocidad con que avanza el equipo, el tiempo promedio por tarea, la cantidad de tareas nuevas que se crean que no estaban contempladas en un inicio, etc.

10. **¿Cuál es el volumen de usuarios que proyecta utilizará el sistema en el corto y largo plazo? ¿Espera que el sistema pueda hacerse público en un futuro?**
    R: El sistema no tiene intenciones de abrirse públicamente para el resto de usuarios, en el corto y mediano plazo será una solución personalizada para nuestra empresa. Con respecto al volumen de usuarios, a pesar de que en el futuro esperamos contar con mayor cantidad de trabajadores, no esperamos superar los 300 usuarios en el largo plazo.