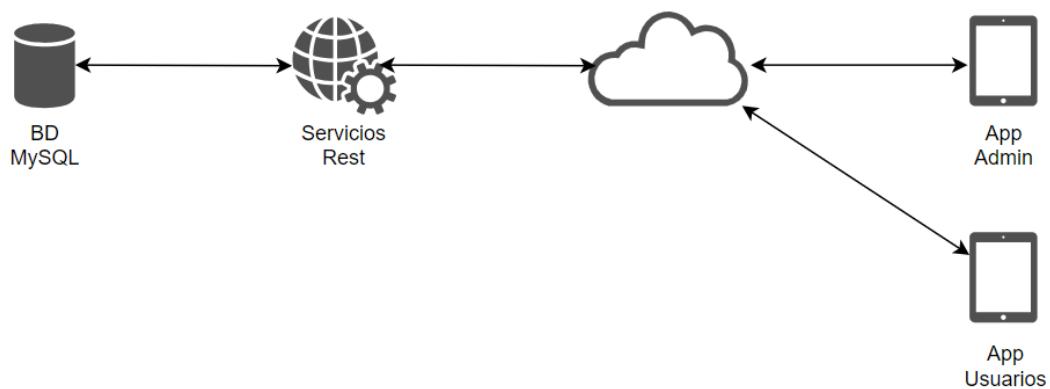


PRÁCTICA 1.

El objetivo de la práctica es crear un sistema de control de presencia, capaz de gestionar los fichajes de los trabajadores de una organización para mantener un control de las horas y trabajos realizados. El funcionamiento del sistema será el siguiente:

1. Desde una **AppAdmin** se crearán los usuarios con sus perfiles.
 - a. Para ello tendremos una página que al igual que los ejercicios guiados nos permite gestionar (Consultar, Insertar, Borrar, Actualizar) los usuarios, así como los trabajos a realizar.
2. Posteriormente habrá una **AppUsuarios** donde los usuarios al acceder podrán:
 - a. Realizar un fichaje y asociarlo a un trabajo (en caso de no tener ningún fichaje previo).
 - b. En caso de disponer de un fichaje previo en un periodo de tiempo inferior a 12 horas, se mostrará dicho fichaje-trabajo permitiendo fichar y cerrando el fichaje.
 - c. Adicionalmente podrán consultar sus fichajes en el día actual (24 horas).
3. Finalmente, desde la **AppAdmin** tendremos dos opciones:
 - a. Mostrar un listado con los diferentes fichajes realizados filtrando por un usuario en un intervalo de fechas.

Para esto se plantea la siguiente arquitectura:



Como se aprecia en la arquitectura propuesta, tendremos dos aplicaciones implementadas ambas mediante **Ionic**, las cuales conectarán con unos **Servicios Rest** implementados mediante **nodejs** (previamente diseñados con **OpenApi** y generados con **swagger**), estos se encargarán de gestionar los datos persistidos en una BD **MySql**.

AppUsuarios

Consistirá en una App con las siguientes páginas:

- **Página Inicial**, con dos opciones:
 - **Registrar fichaje**
 - **Consulta Fichajes**
- **Página Registrar fichaje**, al mostrar la página:
 - En caso de no tener ningún fichaje en 12 horas, se mostrará la opción de iniciar un fichaje nuevo asociándole un trabajo:
 - Seleccionando el trabajo de los creados con la AppAdmin.
 - Generando un nuevo registro que incluya: Fecha-Hora Entrada, Trabajo, Geolocalización.
 - En caso de tener fichaje en el intervalo de 12 horas anteriores, se indicará que se va a finalizar dicho fichaje.
 - Actualizando el registro del fichaje con la fecha-hora de salida.
- **Página Consulta Fichajes**, en esta página, al abrirla se mostrarán los fichajes realizados en ese día en forma de listado incluyendo:
 - Fecha-Hora entrada, Fecha-Hora salida, Trabajo y georreferenciación.

AppAdmin

Consistirá en una App con las siguientes páginas:

- **Página inicial**, donde mostraremos un botón para acceder a cada una de las siguientes páginas:
 - Gestión Usuarios.
 - Gestión Trabajos.
 - Gestión Fichajes.
- **Gestión Usuarios**, mostrará un listado de usuarios y permitirá crear nuevos registros, borrar y actualizar.
 - Se mostrarán los campos más relevantes en el listado y en la vista detalle se mostrarán todos los campos.
- **Gestión Trabajos**, mostrará un listado de trabajos y permitirá crear nuevos registros, borrar y actualizar.
 - Se mostrarán los campos más relevantes en el listado y en la vista detalle se mostrarán todos los campos.
- **Gestión Fichajes**, mostrará un listado de fichajes, permitiendo filtrar por fechas y usuarios (searchbar):
 - Al mostrar esta información en una lista, mostraremos usuario, fecha-hora de entrada, fecha-hora de salida, el trabajo realizado y tendremos un botón que al

pulsarlo mostrará un mapa con un punto indicando la dirección donde se ha fichado.

Base de Datos (MySQL)

Ambas Apps, compartirán datos a través de los Servicios Web necesarios, teniendo la siguiente estructura de BD:

- Usuarios
 - IdUsuario: integer, (Clave Primaria, No Nulo)
 - Nombre: string, 100 (No Nulo)
 - Usuario: string, 50
 - Clave: string, 50
- Fichajes
 - IdFichaje: integer (Clave Primario, No Nulo, AutoIncremental)
 - FechaHoraEntrada: datetime
 - FechaHoraSalida: datetime
 - HorasTrabajadas: integer
 - IdTrabajo: integer
 - IdUsuario: integer
 - GeolocalizacionLatitud: float
 - GeolocalizacionLongitud: float
- Trabajos
 - IdTrabajo: integer, (Clave Primaria, No Nulo)
 - Nombre: string, 200 (No Nulo)
- ApiKey
 - idKey:integer, (Clave Primaria, No Nulo, AutoIncremental)
 - Key: string, 50 (No Nulo)

Servicios Web (Nodejs)

Se deberá de diseñar mediante OpenApi los servicios web Rest necesarios con todas las opciones disponibles para un CRUD, así como las consultas necesarias. Posteriormente mediante el codegen de Swagger, generar el stub para añadir en este nuestra lógica de negocio.

Servicios Web necesarios:

- Usuarios
- Fichajes
- Trabajos
- ApiKey

Optativo

Para optar a la máxima nota se debe de implementar un mecanismo de **Login** en la App, el cual al iniciar la aplicación solicite un usuario y contraseña, la aplicación deberá de acceder a un servicio previamente creado en nodejs el cual validará en la tabla de usuarios de la BD.

Se valorará positivamente el empleo de las siguientes características:

- Uso de una ApiKey en los Servicios Web, para validar el uso de las Apis mediante una key almacenada en BD.
- Generación aplicación PWA.
- Generar despliegue Aplicación (Android/iOS).

Entrega y Corrección

- Se deberá entregar por UACloud, mediante una entrega de práctica, los proyectos/scripts involucrados en un ZIP, así como una memoria detallando los pasos necesarios para la puesta en marcha en el control correspondiente a cada grupo de prácticas.
 - Entrega 1:
 - Hasta el 4-11-2024 (Grupos Martes), 6-11-2024 (Grupos Jueves)
 - App Usuarios, así como BD, Servicios Web y aspectos básicos.
 - Entrega 2:
 - Hasta el 25-11-2023 (Grupos Martes), 27-11-2024 (Grupos Jueves)
 - App Admin, los aspectos básicos y optativos de ambas Apps.
- Se realizará una demo con la aplicación desarrollada al profesor en el laboratorio, en los turnos asignados al alumno, contestando correctamente a las preguntas realizadas por el profesor:
 - Corrección entrega 1:
 - 5-11-2024 (Grupos Martes), 7-11-2024 (Grupos Jueves)
 - Corrección entrega 2:
 - 26-11-2024 (Grupos Martes), 28-11-2024 (Grupos Jueves)
- La práctica es de carácter individual.
- Si se constata que el código desarrollado no es original (por tanto, es una copia), la práctica se calificará con un 0.