

## PARCIAL No. 1

### Problema

*Attendance* es una herramienta de software que permite el registro de asistencia de estudiantes a clase. La herramienta tiene tres módulos de usuario: Administración, Personal y Estudiante. El administrador puede crear un curso y agregar detalles del personal, tanto estudiantes como profesores. Los profesores pueden acceder utilizando la identificación y una contraseña. Luego pueden buscar y seleccionar estudiantes para el curso. Los estudiantes acceden y escriben su código y de esa manera marcan su asistencia a clase.

El desarrollo de la aplicación web inicia con el diseño conceptual del modelo de datos. Luego el desarrollo de una micro aplicación encargada de la Creación, lectura, actualización y eliminación de los datos denominada backend y finalmente la construcción de una interfaz gráfica de la aplicación.

### Objetivo y enfoque

El objetivo es diseñar y desarrollar un aplicación web que pueda dar servicio a una interfaz web para la toma de asistencia. Sus capacidades deben ser las siguientes.

1. Diseñar un modelo Entidad-Relación (ER) para la herramienta *Attendance*
2. Convertir el modelo ER a Relacional para la herramienta *Attendance*
3. Crear las instrucciones SQL para llevar el modelo al SGBD PostgreSQL
4. Desarrollar una herramienta prototipo.

### Herramientas y tecnologías

Para este desarrollo, se ha decidido utilizar PostgreSQL, nodeJS + Express y ReactJS.

- **PostgreSQL** es una base de datos avanzada de código abierto, totalmente gratuita, que cumple bien con estándares SQL y su rendimiento es comparable con otros SGBD de uso comercial, este potente sistema de base de datos relacional con más de 30 años de desarrollo activo ha ganado una sólida reputación.
- **Node + Express** es un marco de aplicación web mínimo y flexible que proporciona un conjunto sólido de funciones para aplicaciones web y móviles. Escrito en JavaScript se aloja en el entorno de ejecución de Node.js.
- **ReactJS** es el marco de JavaScript front-end más popular. Los desarrolladores utilizan JSX, una combinación de HTML y JavaScript, para crear vistas de forma natural. Los desarrolladores también pueden crear componentes para bloques que se pueden reutilizar en sus aplicaciones.

Adicionalmente, se puede considerar herramientas como **Git + github**, **Docker + Docker Hub**, para la construcción y despliegue de la herramienta prototipo.

## Capa de datos [60 puntos]

A continuación se definen los lineamientos para la implementación en el SGBD de los modelos conceptuales Entidad-Relación, Relacional que usted desarrolle. Lea a continuación el enunciado.

Un **administrador** tiene un nombre y una contraseña con los que puede acceder a la aplicación. Este puede tener varias **sedes** de las cuales se conoce un nombre, una ubicación y una identificación única. Las sedes tienen un personal asociado que se encarga de lo misional y no misional de la institución educativa. Es por esta razón que se ha considerado, el salario, la dirección, la EPS, el ARL y el nombre completo de cada persona. Los **profesores** hacen parte de los miembros del personal pero no los estudiantes. Cada **estudiante** está matriculado en uno o varios cursos, de ellos se conoce la dirección, el nombre completo y el código de estudiante. Todo el personal y los estudiantes marcan su asistencia a la sede. La asistencia mantiene información de la hora y fecha en que se registra la asistencia.

1. [15] Crear un diseño utilizando el modelo Entidad-Relación (ER) con notación Chen o Chen-Extendida
2. [10] Crear un diseño utilizando el modelo Relacional utilizando como base el diseño ER
3. [10] Crear los archivos con las instrucciones SQL para llevar sus diseños a PostgreSQL
4. [10] Encuentre las dependencias funcionales de cada una de sus tablas.
5. [10] Describa por qué su diseño se encuentra normalizado.
6. [5 ] Despliegue utilizando la versión contenerizada de PostgreSQL

## Capa del servidor [30 puntos]

A continuación se definen los lineamientos para la construcción/desarrollo del backend. Lea a continuación el enunciado.

Utilice el *framework* Express para crear la aplicación web-API. Para ofrecer los servicios de creación, lectura, actualización y eliminación de registros en cada una de las tablas. Cree una serie de endpoints para cada una de las relaciones consideradas en el modelo de datos. Estos deben permitir el CRUD de los registros en la base de datos.

1. [20] Cree los endpoints para administradores, sucursales, personal, asistencia y estudiantes
2. [10] Crear los diagramas de flujo para el acceso del administrador y para el registro de asistencia

## Capa del cliente [10 puntos]

A continuación se definen los lineamientos para la construcción/desarrollo del frontend. Lea a continuación el enunciado.

1. [5] Cree una interfaz de usuario Graphical User Interface (GUI)
2. [5] Crear los diagramas de flujo para el acceso del administrador y para el registro de asistencia

## Entrega

1. Se debe presentar un documento en formato PDF. Presente el análisis de requerimientos, el diseño y el desarrollo. Además, tenga en cuenta diagramas de la arquitectura de la aplicación, un repositorio en github, conclusión y mejoras futuras.
2. Se debe agregar todos los archivos generados, diagramas UML, archivos SQL y código fuente.
3. Cree un video usando meet presentando la herramienta, utilice diapositivas y la aplicación en ejecución.