TP Integrador

Objetivo General

Desarrollar un sistema completó que permita gestionar alumnos y cursos, y además un usuario administrador deberá poder inscribir a alumnos a dichos cursos.

Procedimiento

Deberan crear 2 repositorios públicos en github, uno para el front y otro para el back. subirán sus cambios a dicho repo, los cuales podrán ser iterativos, es decir mediante commits/push ir agregando funcionalidad.
El trabajo podrá ser en grupos de hasta 4 integrantes.

Modelo de datos

usuario

- id: INT AUTOINCREMENT PK
- email: varchar(255)
- nickname: varchar(50)
- password: varchar(50)
- rol: varchar(50)

alumno

- id: INT AUTOINCREMENT PK
- nombre: varchar(255)
- apellido: varchar(255)
- dni: varchar(10)
- id_usuario: INT **FK usuario**

curso

- id: INT AUTOINCREMENT PK
- nombre: varchar(255)
- descripcion: varchar(1000)
- imagen: varchar(1000)
- anio: INT
- activo: boolean

alumno_curso

- id_alumno: INT **FK alumno**
- id_curso: INT **FK curso**

BackEnd

1. Crear una base de datos llamada “integrador”
2. Dado el modelo de datos crear las sentencias SQL DDL para crear las tablas, sus PK,PF necesarias para dicho modelo de datos.
3. Crear mediante SQL los siguientes usuarios:
    a. **nickname**: admin, **password**: admin, **email**: admin@admin.com, **rol**: admin
    b. **nickname**: docente, **password**: docente, **email**: docente@docente.com, **rol**: docente
4. Crear un backend en nodejs que permita:
    a. Alumnos
        i. LISTAR: Método GET que recupere la lista de alumnos
        ii. ELIMINAR: Método DELETE que permita dado un ID eliminar dicho alumno de la base de datos.
        iii. CREAR: Método POST que dado un JSON con los datos del alumno permita crearlo
        iv. MODIFICAR: Método PUT que dado un JSON con los datos del alumno y un ID, actualizar para dicho alumno sus datos.
    b. Cursos
        i. LISTAR: Método GET que recupere la lista de cursos
        ii. ELIMINAR: Método DELETE que permita dado un ID eliminar dicho cursos de la base de datos. Este proceso además de eliminar los datos de la tabla CURSOS, debe de eliminar las filas de **alumno_curso** que se correspondan al ID de curso eliminado.
        iii. CREAR: Método POST que dado un JSON con los datos del cursos permita crearlo
        iv. MODIFICAR: Método PUT que dado un JSON con los datos del cursos y un ID, actualizar para dicho curso sus datos.
        v. INSCRIBIR ALUMNO: Dado un ID de curso, y una lista de IDs de ALUMNOS cargar en alumno_curso la relación que corresponde a los alumnos inscriptos a dicho curso.
        vi. LISTAR Alumnos para un ID de curso específico.
    c. Seguridad
        i. LOGIN: Método que dado un nickname y una password retorne un token con los datos del usuario, incorporando al token el ROL.

FrontEnd

Crear un proyecto en react que consuma los endpoints expuestos por el backend.
Se deberá de ser creativos utilizando los conocimientos adquiridos en el curso tratando de aprovechar los componentes que ya disponibiliza el framework bootstrap para facilitar el desarrollo del front.

Páginas públicas

1. HOME:
    a. Menú con opción de login, en la portada
    b. Carrusel con 2 imagenes a eleccion y textos como: “Bienvenido a la plataforma de gestión de cursos de Silicon”, “Creando talento IT en el nordeste Argentino”
2. CURSOS: Lista de cursos pública donde los usuario podrán ver cuales son los cursos que dispone la plataforma.
3. LOGIN: Página que permita ingresar nickname y password. Caso de error mostrar mensaje y caso de login exitoso redirigir.

Página restringidas

1. MENU: Opciones
    a. Nickname usuario, con opción de cerrar sesión.
    b. Alumnos
    c. Cursos
2. ALUMNOS: Pantalla que liste los Alumnos del sistema y tenga opción en cada fila de editar o eliminar dicho alumno.
    a. Botón Editar: debe llevar a pantalla de edición
    b. Botón eliminar: debe eliminar dicho alumno, mostrar mensaje de exito/error, y recargar lista de alumnos.
    c. Botón Crear: en alguna parte de la página ubicar un botón “Crear” que lleve a la pantalla de creación de Alumno
3. CURSOS: Pantalla que liste Cursos, debe tener las opciones:
    a. Botón Crear: en alguna parte de la página ubicar un botón “Crear” que lleve a la pantalla de creación.
    b. Botón Editar: debe llevar a pantalla de edición
    c. Botón eliminar: debe eliminar dicho curso, mostrar mensaje de éxito/error, y recargar lista de cursos.

   d. Botón Gestión Alumnos: Debe llevar a una pantalla que permita gestionar los ALUMNOS que se encuentran inscriptos al curso seleccionado.
4. INSCRIPCIÓN ALUMNOS: pantalla que dado un ID de curso, liste sus alumnos y permita:
   a. Agregar alumnos al curso.
   b. Eliminar alumnos del curso.

Bonus point

1. ALUMNOS
   a. El método debe de retornar un mensaje indicado al usuario en caso de un ERROR por existir una relación con dicho alumno a un curso.
   b. Método CREAR: Verificar que no exista ya un alumno con el mismo DNI.
   c. LISTAR: Retornar la lista Ordenada por apellido
   d. LISTAR: Permitir la paginación de los resultados.
2. Seguridad
   a. Permitir CREAR usuario vinculado a un ALUMNO. Si observan el Modelo de datos el Alumno posee un id de usuario. Dicho endpoint deberá de recibir además de los datos del usuario, el ID de ALUMNO.