

Segunda Entrega.

Utilizar la aplicación REST desarrollada en la primera entrega y realizar las siguientes modificaciones:

- Los endpoints, reglas y entidades deben ser los mismos.
- Agregar soporte para base de datos. Utilizar el ORM del framework elegido.
- Las entidades Profesor y Alumno deben guardarse en una base de datos relacional. La información debe persistir al reiniciar o apagar el servidor.
- Considerar que el script de test ya no va a enviar el Id al momento de crear una entidad. El Id debe ser proporcionado por la base de datos.

Crear un bucket de S3 para subir archivos:

- Crear un bucket público (para mayor facilidad).
- Utilizar el SDK de AWS para subir el archivo a S3.
- Agregar el campo `fotoPerfilUrl` a la entidad Alumno para almacenar la URL.
- Agregar el campo `password` para la entidad Alumno.
- Crear el endpoint:
 - `POST /alumnos/{id}/fotoPerfil`
- El POST debe aceptar una imagen y subirla a S3. Aceptar `multipart/form-data`
- El GET `/alumnos/{id}` debe devolver los datos del alumno y la URL de S3 de la foto de perfil.
- Probar que la imagen subida sea alcanzable por medio del navegador.
- Pueden obtener las credenciales para el SDK en el botón de AWS Details dentro del laboratorio. Utilizar el **key**, **secret** y **sessionToken** para que funcione.
- Subir la imagen con permisos públicos, usando el ACL Public Read. Para Java: <https://stackoverflow.com/a/6524088/8723349>

Crear la base de datos relacional en RDS. Subir el nuevo proyecto al mismo servidor de EC2:

- Crear la base de datos en una subnet pública, con DNS público (para que puedan probar en su local). El tamaño de la instancia puede ser el más pequeño.

Crear un topic de SNS para enviar una notificación al alumno:

- Crear un endpoint:
 - `POST /alumnos/{id}/email`
- El endpoint va a enviar una alerta de SNS la cual enviará un correo a mi correo de la UADY. El contenido serán las calificaciones y la información del alumno (nombre y apellido).
- La suscripción al topic de mi parte será en el momento de la revisión, pueden enviar mensajes de prueba desde la consola.

- SNS no requiere proporcionar un email, solo una suscripción a un topic. Ustedes deben poner el contenido en el topic y a las personas que estén suscritas (yo) les debe llegar el correo.

Crear la tabla sesiones-alumnos en DynamoDB para guardar las sesiones de los alumnos:

- Crear un endpoint de:
 - POST /alumnos/{id}/session/login
 - Este endpoint debe recibir la contraseña del alumno y compararla en la base de datos.
 - Si la comparación es exitosa, debe escribir una entrada en la tabla sesiones-alumnos:
 - id (UUID, string)
 - fecha (Unix timestamp, number)
 - alumnoId (number)
 - active (boolean, true por defecto)
 - sessionString (string)
 - El sessionString debe ser un string aleatorio automáticamente generado de 128 dígitos.
- Crear un endpoint de:
 - POST /alumnos/{id}/session/verify
 - Debe recibir el sessionString y ver si la sesión es válida, debe comprobar el valor de active:
 - 200 si el sessionString es correcto y si active = true
 - 400 en algún otro caso.
- Crear un endpoint de:
 - POST /alumnos/{id}/session/logout
 - Debe recibir un sessionString y poner el valor de active en false.

Consideraciones:

- El proyecto deberá correr 100% sobre la infraestructura de AWS.
- Debe existir una base de datos en RDS conectada al servidor EC2.
- Debe existir un bucket de S3 público para almacenar las fotos.
- Debe existir un topic de SNS y deben llegar correos al correo del alumno.
- Debe existir la tabla de sesiones de DynamoDB
- Debe existir una lambda.
- Pueden utilizar Roles de IAM para conectarse a la base de datos o usuario y contraseña. Cualquier método mientras funcione.

Subir capturas de la creación de los EC2, RDS, SNS topic y la tabla de DynamoDB.

El proyecto se revisará de forma automática. Un script de pruebas realizará una serie de test cases sobre todos los endpoints del proyecto.

Utilizar la rama database del software de testing:

https://gitlab.com/eduardo_rdgz/sicei-autotest/-/tree/database