

Realiza una aplicación de backend con nodejs/express que implemente un API para gestión de usuarios tal que se desarrollen los siguientes endpoints:

1) **Registro de usuario (POST):** (1 punto)

[https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/post\\_api\\_user\\_register](https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/post_api_user_register)

Especificaciones técnicas:

- Definir validador para que sea un **email** válido.
- Definir validador para que la **password** contenga al menos 8 caracteres.
- No se podrá registrar con un email ya existente (y validado) en la base de datos (devuelve un error 409 en ese caso).
- La password se guardará cifrada en base de datos.
- Se generará un código aleatorio de seis dígitos y un número máximo de intentos en base de datos para después poder validar el email.
- La respuesta devolverá los datos del usuario (email, status que represente si se ha verificado el email y role por defecto) y el **token JWT**.

2) **Validación del email (PUT o POST):** (1 punto)

[https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/put\\_api\\_user\\_validation](https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/put_api_user_validation)

Especificaciones técnicas:

- Requiere del token JWT recibido en la respuesta del registro
- Definir validador para el código (6 dígitos)
- En el cuerpo se enviará el "code" (en teoría nos llegaría por correo, lo implementaremos más adelante, por ahora, buscadlo en la base de datos para ese usuario).
- Si el código recibido es correcto (es igual al que tenemos en base de datos para ese usuario obtenido del token JWT), modificamos el status a validado y devolvemos un ACK. Si no, devolver un error de cliente 4XX HTTP.

3) **Login (POST):** (1 punto)

[https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/post\\_api\\_user\\_login](https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/post_api_user_login)

Especificaciones técnicas:

- Definir validador para el e-mail y password enviados en el cuerpo de la petición.
- Si las credenciales son válidas devolver datos del usuario y su token JWT
- Si las credenciales son incorrectas devolver un error de cliente 4XX HTTP.

4) **On boarding (PUT y/o PATCH):**

[https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/put\\_api\\_user\\_register](https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/put_api_user_register)

Especificaciones técnicas para los datos personales: (1 punto)

- Requiere Token JWT
- Validar los datos del cuerpo (nombre, apellidos y nif)
- Actualizar el usuario con estos datos

[https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/patch\\_api\\_user\\_company](https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/patch_api_user_company)

Especificaciones técnicas para los datos de la compañía: (1 punto)

- Requiere Token JWT
- Validar los datos (nombre, cif, dirección, etc).
- Si el usuario es autónomo, los datos de la compañía serán sus propios datos personales.

**5) Logo (PATCH):** (1 punto)

[https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/patch\\_api\\_user\\_logo](https://inclined-bonnibelle-bildyapp-1fff10be.koyeb.app/api-docs/#/User/patch_api_user_logo)

Especificaciones técnicas:

- Recibe una imagen como logo (controlad el tamaño)
- Guarda el logo en disco o en la nube (por ejemplo, IPFS) y la url a la imagen en base de datos.

**6) Crea endpoints adicionales siguiendo los requisitos anteriores (validaciones, token, errores, etc.)**

- GET del usuario a partir del token JWT (1 punto)
- DELETE (hard o soft en función de un parámetro query ?soft=false) del usuario a partir del token JWT (1 punto)
- Otras funcionalidades implementadas como consideres, por ejemplo:
  - recuperar contraseña (1 punto),
  - invitar a otros compañeros (para que pertenezcan a tu compañía con el role guest) (1 punto)