

# PRÁCTICA: Digitalización de albaranes

---

## Descripción:

Desarrolla un backend web (API) con NodeJS y express que permita la gestión de albaranes (partes de horas o materiales) entre clientes y proveedores. El API debe permitir a través de sus endpoints lo siguiente (puedes basarte en <https://bildy-rpmaya.koyeb.app/api-docs/>):

## OnBoarding (hecho en la práctica intermedia)

- Crear una cuenta de usuario: POST /api/user/register

NOTA: Al crear la cuenta (o al logarte más adelante), recibirás un token JWT desde el backe-end

- Validar el correo a través de un código recibido en el mismo:  
PUT /api/user/validation

NOTA: Como la petición requiere token de sesión, enviaremos la petición con la cabecera Authorization Bearer

- Login en la aplicación (devuelve un token de sesión): POST /api/user/login
- Introducir datos personales del usuario: PATH o PUT /api/user
- Introducir datos de compañía: PATCH /api/user/company

## Endpoint Extras:

- Cambiar contraseña
- Recuperar contraseña
- Invitar a otro usuario como parte de tu compañía
- Eliminar usuario física (hard delete) y lógicamente (soft delete)
- Envía emails con los códigos de verificación

- **Cientes:**

- Crear un cliente: POST /api/client y actualizarlo PUT (o PATCH) /api/client  
(Deberás asociarlo con el usuario que lo creó y asegurarte que no esté creado ya para ese usuario o compañía)
- Si ya hay clientes, mostrarlos todos: GET /api/client  
y uno en concreto: GET /api/client/{id}  
(Mostrarás solo el cliente o los clientes del usuario o los de la company a la que pertenece el usuario)
- Archivar (soft delete) y borrar un cliente (hard delete)
- Lista los clientes archivados y recuperar uno en concreto (pasarlo a no borrado).

- **Proyectos:**

- Crear un proyecto asociado a un usuario y cliente: POST /api/project  
Además, si quiero actualizarlo: PUT /api/project/{projectId}  
(asegúrate que no esté ya creado para ese usuario o compañía)
- Mostrar todos GET /api/project o un proyecto específico: GET /api/project/{id}  
(Mostrarás solo el proyecto o los proyectos del usuario o los de la company a la que pertenece el usuario)
- Archivar (soft delete) y borrar un proyecto (hard delete)
- Lista los proyectos archivados y recuperar uno en concreto (pasarlo a no borrado).

- **Albaranes:**

- Crear albarán (horas o materiales) de un proyecto en concreto: POST /api/deliverynote
  - El albarán creado podrá ser simple (una entrada para las horas realizadas por una persona o un material)
  - El albarán creado podrá contener múltiple personas/horas y materiales
- Listar albaranes: GET /api/deliverynote
- Mostrar un albarán GET /api/deliverynote/{id}
  - Cuando lo hagas trae los datos del usuario, del cliente y del proyecto (usa "ref" en el modelo y "populate" en el controlador)
- Crea y permite descargar albarán en pdf: GET /api/deliverynote/pdf/{id}
  - Para ello, puedes usar la librería pdfKit (lo veremos en clase)
  - Usa cualquier template siempre que contenga los datos del usuario, del client, del proyecto y del albarán (horas o materiales), y la firma si está firmado
  - Solo se pueden descargar albaranes de ese usuario o de un guest de ese usuario.
- Firma un albarán (subiendo la imagen a IPFS u otra nube)
- Una vez firmado, sube también el PDF a la cloud, y para descargar el pdf trata primero de hacerlo de la cloud
- Borra el albarán (solo si no está firmado).

## **Requisitos:**

- Crea todos los modelos y esquemas (de mongodb o mysql) que consideres necesarios.
- Crea todas las rutas solicitadas
- Crea todos los validadores necesarios.
- Crea todos los controladores
- Excepto el registro, login y recuperar contraseña, todos los endpoints requerirán el token JWT (para que solo un usuario pueda cambiar sus datos o los de su compañía si es admin)
- Haz un manejo de errores de cliente 4XX.
- Tu aplicación registrará en un canal de Slack todos los errores 5XX HTTP (aunque no debería haberlos si has hecho una buena gestión de errores).
- Documenta cada endpoint con Swagger
- Realiza testing con JEST para todos tus endpoints.
- Realiza los commits oportunos (no subas todo el código de golpe)

- Realiza el envío de mail para las validaciones
- Sube un .envexample donde se expliquen las variables de entorno utilizadas.
- Sube ejemplos de ficheros.http para los distintos endpoints.

Rúbrica:

Clientes: 1pto

Proyectos: 2pto

Albaranes: 2pto

Testing: 2pto

Swagger: 2pto

Mail: 0.5pto.

Slack: 0.5pto

Además se tendrán en cuenta todos los requisitos anteriores.