

# GUÍAS

## para hacer “Delilah Restó”

Para que un proyecto apruebe, tiene que cumplir con todas las condiciones del checklist.

### Guía 1: Diseñado y creando el esqueleto del proyecto \_

#### Objetivo

El objetivo del proyecto Delilah Restó es emular la tarea de un desarrollador backend por lo tanto tus entregables de mínima deben ser:

- Archivos JS
- Archivo SQL
- Archivo de documentación
- README.md con instrucciones de instalación.

El frontend no está incluido en el paquete de entregables de este proyecto.

#### Preparación

- Si todavía no los tenés, instalá Node.js, MySQL y Postman.
- Descargá los recursos del proyecto.

#### Método de evaluación

La tarea de un/a evaluador/a es descargar tu proyecto e instalarlo siguiendo tus instrucciones del README.md.

El objetivo es validar que tu aplicación funcione en cualquier plataforma, por lo tanto todas las pruebas las realizará sobre su entorno local. Ten en cuenta éste método para realizar tu desarrollo.

#### Paso 1: Introducción al proyecto

Si aún no descargaste los materiales hazlo ahora haciendo [click aquí](#). Analiza cada una de las vistas para entender el proyecto en su totalidad. Observa con detenimiento cómo es el comportamiento de la aplicación y comienza a diagramar a grandes rasgos la arquitectura de tu aplicación.

#### Paso 2: Estructura de la información

El segundo paso es armar tu base de datos para que soporte todo el flujo de la aplicación. Revisa nuevamente cada una de las vistas y diagrama

cada tabla en base a la información que extraes de las vistas. Haz [click aquí](#) para ver el ejemplo.

Al finalizar el análisis de todas las vistas tu primera versión del modelo de datos estará lista.

### **Paso 3: Definiendo la especificación**

En base a la información relevada, es momento de crear la especificación OpenAPI de la API que vamos a crear.

Abre el archivo `spec.yml` y completa el listado de endpoints.

### **Paso 4: Creando el entorno**

Es momento de iniciar la carpeta con tu proyecto.

Crea una nueva carpeta en tu computadora, comienza con un repositorio de git, y inicializa tu proyecto de NodeJS creando el archivo `package.json`.

Instala las dependencias necesarias y crea tu servidor web.

### **Paso 5: Conexión a la base de datos**

Extiende el código del servidor con la conexión a la base de datos. Al finalizar este paso deberías tener un servidor escuchando en un puerto local, conectado a una base de datos MySQL.

## **Guía 2: Listado y creación de Productos**

### **Paso 1: Creando la tabla y estructura de productos**

Ahora vas a crear la primer tabla del proyecto!

En base a la especificación creada anteriormente, crea la tabla de productos con las columnas requeridas para cumplir la especificación de la API.

### **Paso 2: CRUD de productos**

Añade Express a tu proyecto, crea tu primer endpoint de `/productos` y genera las operaciones necesarias para poder crear, leer, actualizar y borrar un producto.

Tomate tu tiempo para ir probando y debugueando cada parte por separado. Protip: No dudes en usar la consola integrada en phpMyAdmin para probar tus consultas si es que tenés alguna duda.

## Guía 3: Sistema de usuarios

### Paso 1: Creando la tabla y estructura de usuarios

Confiamos en que ya tenés experiencia creando una nueva tabla. La repetición convierte conceptos en saberes! En este paso debes crear la tabla de usuarios.

### Paso 2: Registro y login de usuarios

Basándose en la especificación, crea el endpoint correspondiente para darle al usuario una forma de crear una nueva cuenta.

Lo siguiente es incorporar tus conocimientos sobre tokens JWT para darle a tus usuarios registrados una forma de iniciar sesión en la plataforma.

### Paso 3: Agregar validación de roles

Por último, generá una estrategia de validación de roles para todos los endpoints existentes. Por ejemplo que solo usuarios administradores puedan crear, editar y eliminar productos, y que los usuarios logueados solo tengan acceso a su información personal.

## Guía 4: Creando pedidos

### Paso 1: Creando la tabla y estructura de pedidos

La tabla de pedidos se relaciona con dos tablas: la de usuarios y la de productos. Al crearla, tené en cuenta las siguientes condiciones: Un pedido puede ser realizado por un único usuario, pero un usuario puede realizar más de un pedido. Por otro lado, un pedido puede contener varios productos y un producto puede formar parte de varios pedidos.

## **Paso 2: Creando y obteniendo pedidos**

Con las tablas ya creadas, generá el primer endpoint para la creación de los productos. Recordá basarte en la especificación para saber qué vas a recibir y qué deberías devolver.

Ahora podés crear un nuevo endpoint para hacer GET de todos los productos. Esta consulta no solo tiene que devolver el detalle de los pedidos sino también el detalle de todos los productos. Es importante que el GET de todos los pedidos solo pueda ser ejecutado por un admin, un usuario logueado solo debe recibir los propios.

## **Paso 3: Edición de pedidos**

El último paso para finalizar la API es brindarle al admin herramientas de edición sobre los pedidos realizados para poder actualizar el estado de los mismos. Solo te falta crear un endpoint para hacer un UPDATE sobre la ruta de pedidos, que debería ser muy parecido al que ya creaste en el CRUD de productos.