**Delilah Restó**

Este proyecto plantea la creación de un sistema de pedidos online para un restaurante. Deberás poner en funcionamiento las partes necesarias para montar una REST API que permita realizar altas, bajas, modificaciones y obtención de información sobre una estructura de datos que podría consumir un cliente. Parte del desafío estará enfocado en lograr que el desarrollo del proyecto sea puesto en producción utilizando web services.

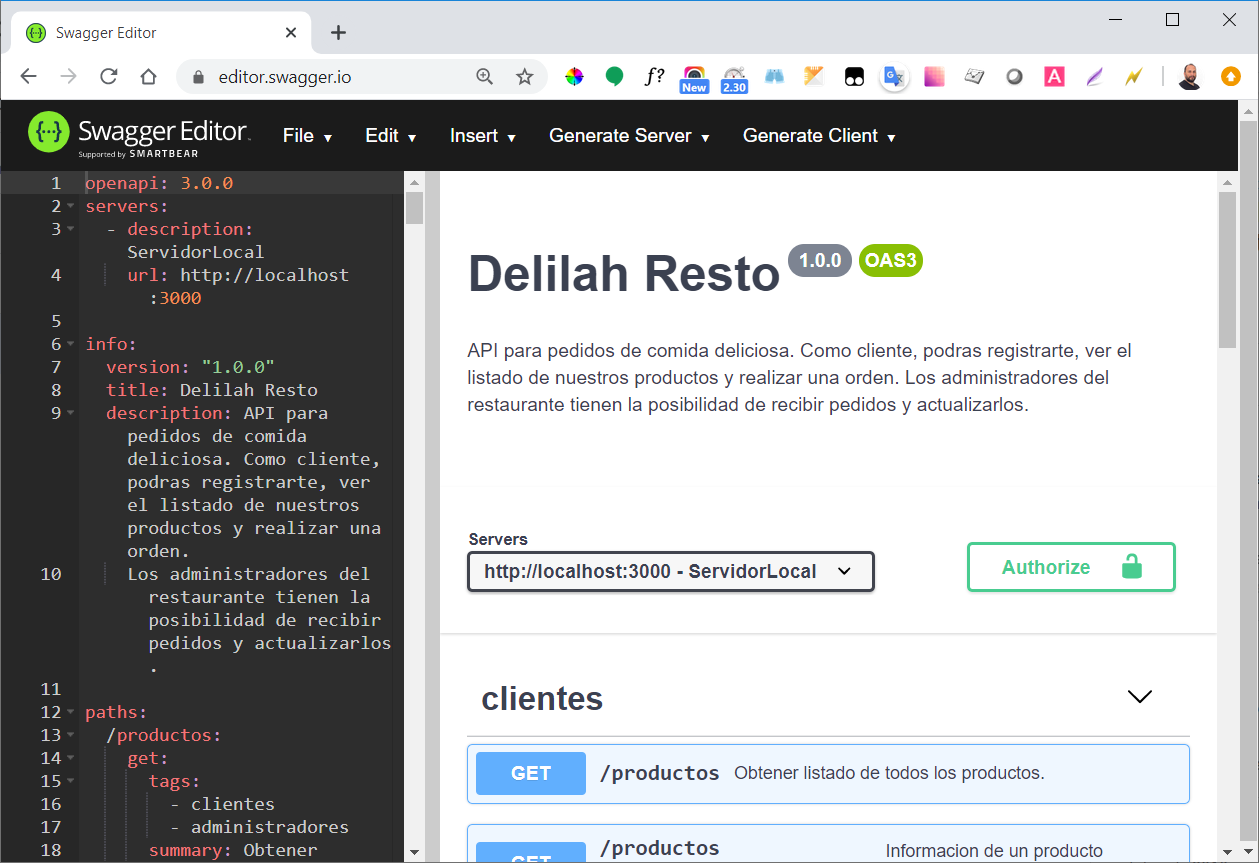
## **Descripción general**

Crearás el backend para un sistema de pedidos online para un restaurante poniendo en funcionamiento las partes necesarias para montar una REST API que permita realizar operaciones CRUD sobre una estructura de datos.

## **Entregable**

El objetivo del proyecto Delilah Restó es emular la tarea de un desarrollador backend. Los entregables de mínima que se esperan son:

* Archivos JS
* Archivo SQL o instalación desde la aplicación
* Archivo de documentación
* README.md con instrucciones de instalación



**Recursos para desarrollar el proyecto**

[Aquí](https://s3.amazonaws.com/resources.acamica.com/contenidos/DWFS-19/recursos_estudiantes/Recursos+proyectos/Proyecto3/delilah_exports.zip) encontrarás los recursos necesarios para realizar el proyecto.

**Para tener en cuenta:** el frontend no está incluido en el paquete de entregables ni recursos de este proyecto. El foco está puesto en el Backend.

## **Uso de librerías**

No está permitido el uso de librerías, plugins o cualquier otro recurso que no esté especificado en esta guía, ya que el objetivo del proyecto es validar los conocimientos de base.

## **Consignas**

A continuación, te ofrecemos una guía de pasos sugeridos para construir tu proyecto (¡no es obligatorio que los sigas estrictamente en este orden!).

**a. Método de evaluación de tu proyecto**

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 el/la evaluador/a realizará todas las pruebas sobre su entorno local. Ten en cuenta este método para realizar tu desarrollo.

**b. Preparación**

* Si todavía no los tienes, instala Node.js, MySQL y Postman.
* Descarga los [recursos del proyecto.](https://s3.amazonaws.com/resources.acamica.com/contenidos/DWFS-19/recursos_estudiantes/Recursos+proyectos/Proyecto3/delilah_exports.zip)

**c. Puesta a punto**

* **Paso 1: Introducción al proyecto**
  + Entre los recursos que descargaste, 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 la misma. Haz [click aquí](https://s3.amazonaws.com/resources.acamica.com/contenidos/DWFS-19/imagenes_b3/delilah-resto-screenshot.png" \t "_blank) para ver el ejemplo.
* **Paso 2: Definiendo la especificación**
* En base a la información relevada, es momento de la documentación con Swagger bajo el estándar OpenAPI.
* Crea un archivo spec.yaml en el raíz de tu proyecto y completa el listado de endpoints.
* **Paso 3: 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, e inicializa tu proyecto de NodeJS creando el archivo package.json.
  + Instala las dependencias necesarias y crea tu servidor web.
* **Paso 4: Endpoints**
  + Instala express en tu proyecto y comienza a definir todas las rutas que has especificado en tu documentación, recuerdo utilizar los métodos tipo GET, POST, PUT y DELETE dependiendo la acción que vayas a realizar.
* **Paso 5: Estructura de la información**
  + El siguiente 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.
  + Al finalizar el análisis de todas las vistas tu primera versión del modelo de datos estará lista.
* **Paso 6 : 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.

**d. Listado y creación de productos**

* **Paso 1: Crea la tabla y estructura de productos**
  + Ahora vas a crear la primera 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.
  + Tómate tu tiempo para ir probando y debugueando cada parte por separado.

Pro Tip: No dudes en usar la consola integrada en phpMyAdmin para probar tus consultas si es que tenés alguna duda.

**e. Sistema de usuarios**

* **Paso 1: Crea la tabla y estructura de usuarios**
  + Confiamos en que ya tienes 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, genera 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”.

**f. Creación de 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, ten en cuenta las siguientes condiciones:
  + Un pedido puede ser realizado por un único usuario.
  + Un usuario puede realizar más de un pedido.
  + Un pedido puede contener varios productos.
  + Un producto puede formar parte de varios pedidos.
* **Paso 2: Creando y obteniendo pedidos**
  + Con las tablas ya creadas, genera el primer endpoint para la creación de los productos. Recuerda basarte en la especificación para saber qué vas a recibir y qué deberías devolver.
  + Ahora puedes 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.

## **¡Felicitaciones!**

¡Si llegaste hasta aquí, quiere decir que completaste tu proyecto!

Comparte la URL de tu repositorio con colegas para que den una mirada a tu código fuente.



## **Checklist**

Antes de subir tu proyecto a la plataforma Acámica para ser evaluado, realiza las siguientes verificaciones (son las que los/as evaluadores/as tendrán en consideración al momento de corregir tu trabajo).

### **Condiciones para entregar tu proyecto**

* **Archivos mínimos entregados:** debes incluir como mínimo todos los archivos JS correspondientes al servidor de la API. Debes incluir el package.json con el listado de dependencias.
* **Estructura de base de datos:** debes enviar un archivo con queries de base de datos requeridas para crear la base de la estructura de las tablas requeridas por el proyecto o un instalador que importe todos los datos necesarios en la base de datos.
* **Instrucciones para ejecución:** debes incluir un README.md con pasos requeridos para inicializar el servidor. Incluye la configuración del servidor, instrucciones y archivos para poder tener la estructura base de la base de datos y cómo iniciar el servidor.
* **Documentación:** archivo spec.yml con la documentación de tu API.
* **Enlace al proyecto sobre un repositorio:** envía el enlace a tu proyecto sobre un repositorio. Si aún no tienes repositorio, crea tu usuario gratuito en [GitHub,](https://github.com/) [GitLab,](https://about.gitlab.com/" \t "_blank) [Bitbucket](https://bitbucket.org/" \t "_blank) o cualquier otra plataforma que trabaje con GIT.

### **Condiciones para aprobar**

* Condición 1: Poder registrar un nuevo usuario.
* Condición 2: Un usuario debe poder listar todos los productos disponibles.
* Condición 3: Un usuario debe poder generar un nuevo pedido al restaurante con un listado de platos que desea.
* Condición 4: El usuario con roles de administrador debe poder actualizar el estado del pedido.
* Condición 5: Un usuario con rol de administrador debe poder realizar las acciones de creación, edición y eliminación de recursos de productos (CRUD de productos).
* Condición 6: Un usuario sin roles de administrador no debe poder crear, editar o eliminar un producto, ni editar o eliminar un pedido. Tampoco debe poder acceder a información de otros usuarios.