| Colegio Universitario **IES** *Siglo 21* | |
| --- | --- |
| **INSTANCIA EVALUATIVA FINAL** | |
| **Materia:** Aplicación web 2 | **Docente:** Magaquian Angel |
| **Modalidad:** DISTANCIA | **Fecha: 2023/02/22** |

**Reservado para el alumno**

| **Alumno**: | **Carrera:** ANALISTA DE SISTEMAS |
| --- | --- |
| **DNI**: | **Localidad**: |

**Reservado para el docente**

| NOTA |
| --- |
|  |

**CONSIDERACIONES GENERALES**

**Objetivos.**Desarrollar un back-end empleando nodejs y el framework de ExpressJs. Lograr estructuras eficientes de enrutamiento utilizando correctamente los métodos de peticiones GET, POST, PUT y DELETE junto con el manejo de código de respuestas en el contexto de consultas a estructuras JSON.

**Criterio de Evaluación.**

Correcta configuración del proyecto npm para la sintaxis de ES6 en adelante, correcto uso de métodos, códigos de respuesta y criterio a la hora de buscar la mayor eficiencia posible y reutilización de funciones. Cada punto del parcial tendrá un valor que podrá variar dependiendo de su funcionamiento y de su calidad.

**Modalidad de Evaluación.**

Escrito e individual.

**Puntaje.**

Para que un programa o ejercicio sea aprobado como mínimo debe cumplir con el criterio de evaluación. Cada ejercicio tiene asignado un porcentaje máximo que disminuirá en función de los métodos empleados y la calidad del código, considerando errores de estructura, errores lógicos y el criterio para resolver el problema. Si el programa cumple con el criterio de evaluación pero tiene una calidad de desarrollo baja, se otorgará al menos un 20%. En caso de que el programa no cumpla con el criterio de evaluación, se asignará un puntaje de 0% en ese punto.

| Puntaje | 1-24 | 25-39 | 40-54 | 55-61 | 62-66 | 67-72 | 73-79 | 80-87 | 88-95 | 96-100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nota | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Valoración | Logro  No satisfactorio | | | Logro básico | Logro satisfactorio | | Logro  Destacado | Logro  Sobresaliente | | Logro  Excelente |

## Proyecto a desarrollar: Back-end de recetas

Se debe desarrollar back-end para la lectura, creación, actualización y eliminación de datos en base a los siguientes archivos [***JSON***](https://drive.google.com/file/d/12WoNeRb8pHxGolaRYzFvXGQOIm0UofOw/view?usp=drive_link). Este conjunto de archivos son:

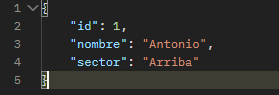
* **Sectores**: guarda la información de sectores de un bar junto al ID de mesas que le pertenece, los mozos encargados de esa zona y un número que representa su ocupación. Su capacidad se mide en base a la cantidad de mesas totales que posee ese sector y la ocupación es por cada mesa, no por la cantidad de personas.
* **Mozos**: solamente guarda la información personal de cada mozo, el dato más relevante es su nombre y su ID.
* **Pedidos**: Se almacena los pedidos realizados por mesa, posee un ID único, el ID de la mesa, ID del mozo y los pedidos realizados con el ID del producto y la cantidad. También posee un **estado** que cuando está en *true significa que la mesa sigue activa* (ocupada con personas) y puede seguir acumulando pedidos pero cuando esta en false (desocupada) *significa que ya esa mesa se liberó* y se cerró por ende no se pueden hacer más pedidos.

Estos archivos deben ser usados para desarrollar el back-end y realizar las siguientes acciones con sus endpoints correspondientes:

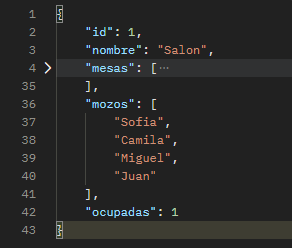
**1-(10%)**Desarrollar la siguiente estructura de rutas para el manejo de los endpoints solicitados según corresponda:

* mozo/
* pedidos/
* sectores/

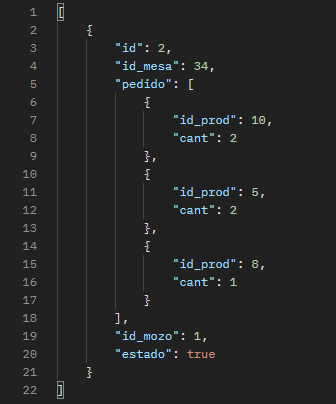
**2- (20%)** EndPoint: **infoMozo**. Este endPoint devuelve la información personal de un mozo en particular y en qué sector se encuentra actualmente.



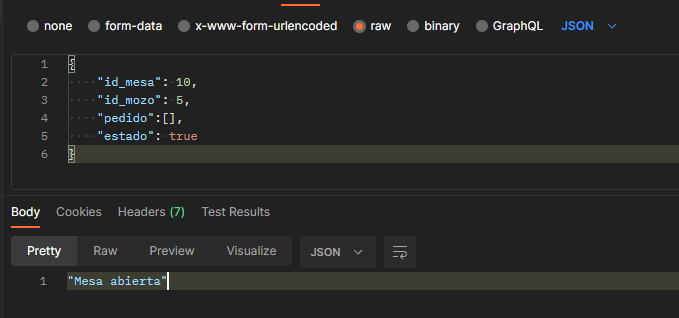
**3-(20%)** EndPoint: **infoSector**. Este endPoint debe devolver la información completa del JSON de sectores en base a un ID de sector que se recibe como parámetro. Se debe reemplazar el ID de los mozos por su nombre.



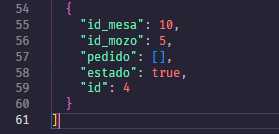
**4-(20%)** EndPoint:**pedidosActivos**. Este endPoint debe devolver todos los pedidos con el estado de true.



**5-(30%)** EndPoint: **abrirNuevaMesa**: Este endPoint debe agregar un nuevo registro a **pedidos** con su estructura básica vacía, solo debe tener el id autoincremental, el ID del mozo, ID de mesa y estructura vacía de pedidos, también debe tener su estado en true. Se debe mandar un mensaje de éxito o error dependiendo del caso.



*Parámetros que se envían al endpoint y respuesta del lado de postman.*



*JSON de pedidos con el nuevo objeto.*

La entrega de este parcial deberá ser un link del repositorio donde esté la solución.