|  |  |
| --- | --- |
| Colegio Universitario **IES** *Siglo 21* | |
| **INSTANCIA EVALUATIVA FINAL** | |
| **Materia:** Aplicación web 2 | **Docente:** Magaquian Angel |
| **Modalidad:** DISTANCIA | **Fecha: 2024/03/05** |

**Reservado para el alumno**

|  |  |
| --- | --- |
| **Alumno**: Pellizzon Dario | **Carrera:** ANALISTA DE SISTEMAS |
| **DNI**: 32623387 | **Localidad**: Cordoba |

**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 hotel

Se debe desarrollar back-end para la lectura, creación y actualización de datos de datos en base a los siguientes archivos [***JSON***](https://drive.google.com/file/d/1PFIoakPikrXO4Ds61Adn2m2C-gxQEtXw/view?usp=drive_link). Este conjunto de archivos son:

* **Habitaciones**: Guarda la información de las habitaciones existentes del hotel. La estructura que posee es:
  + **id**: una clave compuesta por su piso y la habitación que representa.
  + **piso**: es el piso en el que se encuentra.
  + **habitación**: el número de habitación que representa, es incremental de acuerdo a las demás habitaciones de ese piso.
  + **estado**: estos pueden ser “habilitada”, “ocupada”, “limpieza”, “refacción”. La única habitación que se puede reservar son las que están con el estado de **“habilitada”.**
  + **ocupación:** cantidad de personas que pueden estar en la habitación.
  + **id\_cliente**: id del cliente que la ocupa, solo debe guardar el id en caso de que el estado sea “**ocupada**”, en caso contrario, debe ser null.
* **Cliente**: Guarda la información de clientes. La estructura que posee es:
  + **id**: debe ser autoincremental.
  + **nombre**.
  + **apellido**.

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:

* habitaciones/
* clientes/

**2- (20%)** EndPoint: **nuevoCliente**. Este endPoint debe agregar un nuevo cliente la información requerida en su estructura, todos los campos son obligatorios..

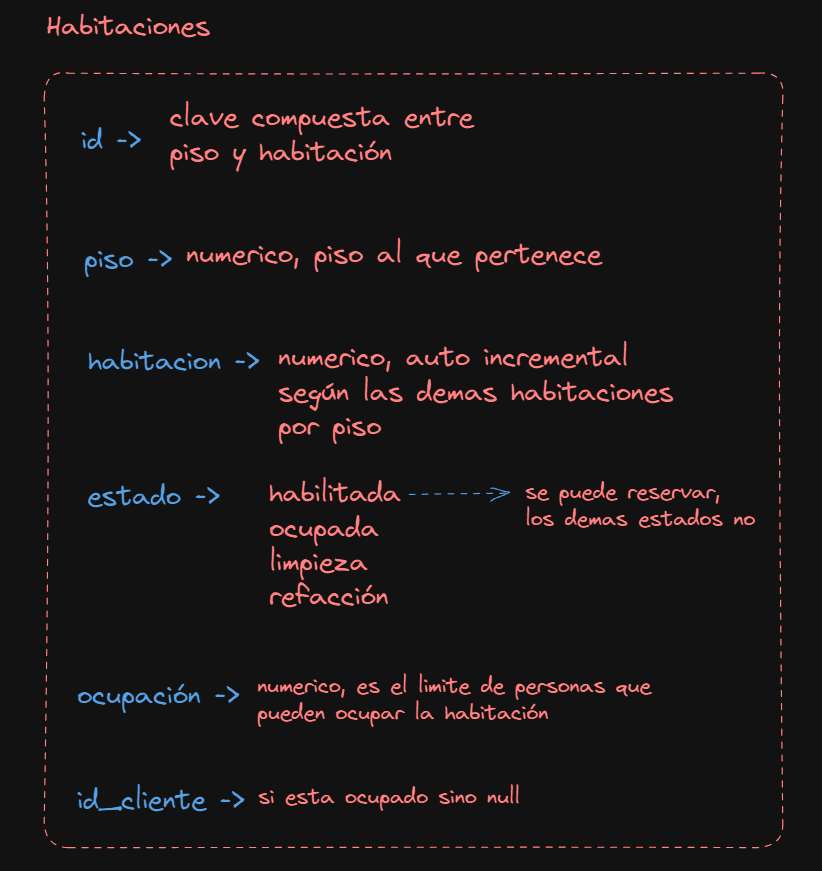
**3-(20%)** EndPoint: **infoHabitaciónPiso**. Este endPoint debe devolver la información completa del JSON de habitaciones en base a un piso que se recibe como parámetro. Si está ocupada, debe mostrar el nombre y apellido del cliente.

**4-(30%)** EndPoint:**filtroEstadoHabitaciones**. Este endPoint debe devolver todas las habitaciones con el estado que se recibe como parámetro.En el caso que ninguna habitación tenga un estado solicitado se debe devolver un array vacio y en el caso de que el estado sea “**ocupado**” debe mostrarse quien es el cliente con su nombre y apellido.

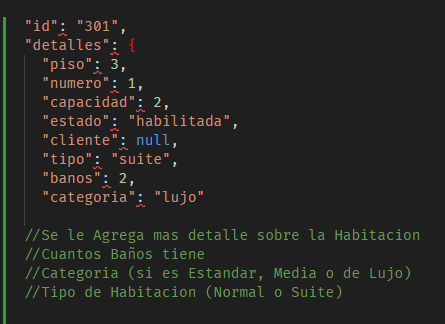
**5-(20%)** La estructura JSON de habitaciones que está planteada no es la mejor. Se solicita que se planteé una nueva estructura para las habitaciones. La presentación de esta estructura puede ser en un archivo JSON con la información o un diagrama. El formato que se elija debe ser adjuntado a este documento o ser subido a github con el resto del proyecto.

No se debe eliminar u obviar las claves actuales.

*Ejemplo de diagrama con la estructura actual (diagrama realizado con* [*excalidraw*](https://excalidraw.com/)*):*

**

Esta estructura es muy básica y sería más conveniente una estructura con más subniveles.



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