**EVALUACIÓN FINAL**

**PROGRAMACIÓN WEB II  
2024-20**

**Deberán leer detenidamente cada una de las indicaciones de la evaluación con la finalidad de cumplir con todos los puntos solicitados.**

**CONSIDERACIONES GENERALES PARA EL DESARROLLO DEL TRABAJO / PROYECTO:**

* Actividad Grupal – Los grupos asignados están en la plataforma ISIL+.
* Se debe registrar los apellidos y nombres de los estudiantes en orden alfabético por apellido paterno.
* No se considerará calificación para aquellos miembros del grupo que no estén en la lista de integrantes.
* Todos los integrantes de grupo tienen responsabilidad por la actividad calificada enviada independientemente de si esta fue dividida en partes por el mismo grupo
* Es trabajo de todo el grupo velar por que la actividad calificada sea elaborada bajo los criterios de ética y buenas prácticas académicas de la institución.
* Si el grupo tuviera dudas con respecto a lo solicitado en uno o varios puntos deberá comunicarlo oportunamente al docente para que la duda sea aclarada en un plazo prudente y puedan cumplir con los plazos de entrega de la actividad.
* Es responsabilidad exclusiva del grupo subir adecuadamente el documento solicitado corroborando que sea el correcto y que se haya cargado sin errores a la plataforma ISIL+.
* NO SE REVISARÁN LAS EVALUACIONES ENTREGADAS FUERA DEL PLAZO ESTABLECIDO.

**CONSIDERACIONES DEL ENTREGABLE**

* La presentación de este trabajo se hace a través de un informe escrito.
* El trabajo debe estar ordenado en cuanto a forma y fondo.
* Si se van a incluir imágenes de referencia en la actividad, deberán revisar que estén colocadas de manera ordenada y alineadas al texto. No colocar imágenes de mucho peso o gran tamaño.
* El trabajo debe mostrar los puntos solicitados en el mismo orden en el que se han solicitado.
* Las fuentes de información utilizadas deben ser citadas utilizando las normas APA.
* El trabajo debe contener una carátula que contenga el nombre del curso, el NRC, el título del trabajo, el nombre del docente, los nombres de los integrantes del grupo y el periodo de estudio.

1. **INTEGRANTES**

| **APELLIDOS Y NOMBRES DE LOS ESTUDIANTES:** | **PORCENTAJE DE PARTICIPACIÓN:** | **CORREO ELECTRÓNICO:** |
| --- | --- | --- |
| Mendoza Zapata Renzo Martin | 100% | 77177119@mail.isil.pe |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **FINALIDAD / OBJETIVO DE LA ACTIVIDAD**
2. Utiliza herramientas y estándares para desarrollar y validar soluciones de software, haciendo uso técnicas matemáticas y seleccionando la estructura de datos y los algoritmos que permitan acelerar el tiempo de respuesta y que optimicen el consumo de recursos.
3. Mostrar los conocimientos adquiridos sobre el uso de node.js, express y jwt.
4. Mostrar dominio en el empleo de estas tecnologías de forma combinada.
5. **INSTRUCCIONES DE LA ACTIVIDAD**

El desarrollo de la actividad tendrá dos componentes:

1. Informe de la actividad el que será trabajado en formato de Word con el archivo en el siguiente formato Grupo\_N°. El informe tendrá el siguiente contenido:

|  |
| --- |
| **Contenido del Informe** |
| 1. Carátula. Debe incluir el nombre de la institución, el nombre de su carrera, el título de su trabajo, el nombre del curso, el nombre del profesor, el código y nombre del alumno. |
| 1. Índice de contenidos. |
| 1. Introducción. |
| 1. Resumen. |
| 1. Marco Teórico |
| 1. Descripción del caso de estudio. |
| 1. Objetivos. |
| 1. Desarrollo.    1. Especificaciones del Requerimiento de Software.    2. Análisis de Software.    * Diseño del modelo de la Base de Datos.    * Diseño del API bajo la metodología RUP (Diagrama de casos de uso, Diagrama de clases, Diagrama de objetos, Diagrama de estados, Diagrama de actividades, Diagrama de secuencia, Diagrama de colaboración, Diagrama de componentes y Diagrama de despliegue). |
| 1. Recomendaciones. |
| 1. Conclusiones. |
| 1. Referencias Bibliográficas. |

1. Archivo con el desarrollo de la aplicación el que será entregado en exclusivamente en GitHub.
2. **CRITERIOS DE EVALUACIÓN**

|  |  |
| --- | --- |
| **CRITERIOS** | **PUNTAJE** |
| **Calidad de la redacción:** Respeta el formato APA y mantiene coherencia en su redacción. | **5** |
| **Uso de Estándares en la solución:** Plantea un trabajo que mantiene los estándares para su codificación, incluyendo coherentemente los archivos necesarios y ordenados según la forma de codificación aprendidas en clase. | **5** |
| **Implementación:** Mantiene un orden en el desarrollo de su código, usando las buenas aprendidas en clase. Realiza pruebas de Testing para el código entregado según los casos de prueba planteados. | **5** |
| **Scripts Completos:** Se hace entrega de todos los scripts necesarios para el despliegue de la aplicación. | **5** |
| **TOTAL** | **20** |

1. **CONTENIDO DE LA EVALUACIÓN:**

**5.1 INSTRUCCIONES**

Estimado alumno, esta es tu **Evaluación Final** del curso de **Programación Web II**. Recuerde que esta evaluación es grupal y deberás subirla a la plataforma hasta la fecha programada. Considere entregar todas las evidencias que considere para demostrar la óptima realización de su práctica (Solo con GitHub). Buena suerte…….!!!

**5.2 CASO DE DESARROLLO**

Una cadena de restaurantes de comida italiana, **"Tagliatore"**, se especializa en la gastronomía Toscana y Romana, su carta, incluye una amplia gama de vinos. La empresa desea modernizar su infraestructura tecnológica y ha decidido implementar un API RestFull que permita gestionar sus operaciones diarias de manera más eficiente. El API debe permitir realizar las siguientes operaciones:

* 1. **Gestión de Platillos**
* Crear un nuevo plato: La empresa debe poder añadir nuevos platos a su carta con detalles como nombre, ingredientes, precio e imágenes.
* Obtener detalles de un plato: Permitir la consulta de información detallada de un plato específico por su ID.
* Actualizar un plato existente: Modificar la información de un plato, incluyendo ajustes en el precio e imágenes.
* Eliminar un plato: Permitir la eliminación de platillos que ya no estén disponibles o que hayan sido descontinuados.
  1. **Gestión de Clientes**
* Registrar un cliente nuevo: Añadir nuevos clientes con detalles como nombre, dirección de correo electrónico, número de teléfono y número de DNI.
* Obtener información de un cliente: Consultar los detalles de un cliente específico por su ID.
* Actualizar información del cliente: Permitir la actualización de la información del cliente, como la dirección de correo y el número de DNI.
* Eliminar un cliente: Eliminar los registros de clientes que ya no asisten al restaurante.
  1. **Gestión de Órdenes**
* Crear una nueva orden: Registrar una nueva orden, incluyendo el ID de la mesa, los platillos solicitados y las cantidades.
* Obtener detalles de una orden: Consultar la información de una orden específica por el ID de la mesa.
* Actualizar el estado de una orden: Modificar el estado de la orden (por ejemplo, de "pendiente" a "entregado" o "cancelado").
* Eliminar una orden: Eliminar registros de órdenes canceladas o erróneas.
  1. **Gestión de Categorías**
* Crear una nueva categoría: Añadir nuevas categorías de platillos.
* Obtener todas las categorías: Consultar todas las categorías disponibles.
* Actualizar una categoría: Modificar el nombre o la descripción de una categoría existente.
* Eliminar una categoría: Eliminar categorías que ya no se utilizan.
  1. **Gestión de Meseros**
* Crear un nuevo mesero: Añadir nuevos meseros al sistema.
* Obtener todas los meseros: Consultar una lista de todos los meseros creados.
* Actualizar datos de los meseros: Modificar los datos de los meseros.
* Eliminar meseros: Eliminar usuarios mediante eliminación lógica (No Física).

**5.3 ESPECIFICACIONES TÉCNICAS**

1. El API debe ser desarrollado utilizando Node.js con Express.
2. Utilizar una base de datos como MongoDB.
3. Asegurarse de que el API siga los principios RESTful, incluyendo el uso correcto de los métodos HTTP (GET, POST, PUT, DELETE).
4. Utilizar una JWT para mantenimiento de los meseros y la encriptación de sus contraseñas, siguiendo las buenas prácticas aprendidas en clase.

Enlace al repositorio:

https://github.com/Cavalier556/Grupo10-Final