Evaluación Parcial N° 2 CUY6142

**Docente**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sigla** | **Nombre Asignatura** | **Tiempo Asignado** | **% Ponderación** |
| **CUY6142** | **TELEPRESENCIA Y ENTORNOS INNOVADORES DE COLABORACIÓN HUMANA** | **4** | **33%** |

1. **Situación evaluativa**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | **Ejecución práctica** |  |  | **Entrega**  **de encargo** |  |  | **Presentación** |

1. **Instrucciones generales para el/la estudiante**

Descripción general de la evaluación:

* Esta actividad tiene carácter sumativo y tiene una ponderación del 33 % de la nota de presentación al examen.
* La evaluación deberá realizarse exclusivamente en horario de clases y es individual.
* Tiempo asignado para esta evaluación es de *120 minutos* y se realiza en parejas en laboratorio o taller*.*

Orientaciones para la implementación de la evaluación Sugerencias:

* La evaluación se realiza en el horario de la clase, en 4 bloques, o sea, en 4 horas académicas.
* El laboratorio o taller debe contar audífonos con micrófono.
* Retroalimente a la siguiente clase. Puede usar AVA para esto.
* El estudiante debe levantar la VM Devasc.

Ejecución práctica:

* + Debe generar las evidencias específicas de los requerimientos solicitados en un documento Word o PDF (Puede utilizar este mismo), Sea ordenado, genere explicaciones de los pasos realizados, no es necesario realizar múltiples evidencias de mismo tipo de ítem. De acuerdo con el plazo asignado por el profesor, envíe el documento al profesor a través AVA
  + Debe tomar capturas de pantalla de lo realizado, en este mismo documento, y enviar por AVA a la docente una vez finalizada la clase.

**Instrucciones Específicas**

**Escenario:**

Usted ha sido contratado como desarrollador junior en "API Ltda.", una consultora tecnológica especializada en la integración de sistemas y automatización de procesos. Su primer gran proyecto es con la Biblioteca Nacional, la cual busca modernizar sus sistemas de gestión de inventario y, al mismo tiempo, desarrollar una nueva aplicación para sus visitantes que ofrezca rutas optimizadas para llegar a la biblioteca desde distintos puntos de la ciudad.

El equipo de desarrollo ha decidido utilizar un enfoque basado en microservicios y APIs para asegurar la escalabilidad y mantenibilidad de las soluciones. Su rol es crucial para llevar a cabo dos tareas fundamentales: la primera consiste en poblar y administrar el catálogo de libros digitales a través de la API REST de la biblioteca. La segunda, implica mejorar un prototipo de software de geolocalización que ayudará a los visitantes a planificar su viaje.

Para estas tareas, deberá hacer uso de herramientas estándar en la industria como Postman para la interacción con APIs, Visual Studio Code para el desarrollo en Python, y Git/GitHub para el control de versiones, todo dentro del entorno de desarrollo proporcionado en la máquina virtual DEVASC.

**Requerimientos:**

1. **Gestión de la Biblioteca con Postman y APIs:**

**Dentro de la VM Devasc, utilizando Postman, debe agregar 3 nuevos libros al catálogo de la biblioteca. Los datos para cada libro son los siguientes:**

* + Libro 1: id: 5, título: "Guía de Estudio Para la Certificación Ccna Security", autor: Su-Nombre\_Su-Apellido, isbn: Su RUT (sin puntos y sin guion).
  + Libro 2: id: 6, título: "Redes Cisco. CCNA Routing y Switching.", autor: Su-Nombre\_Su-Apellido, isbn: Su RUT (sin puntos y sin guion).
  + Libro 3: id: 7, título: "Redes Avanzadas 1. DEVNET.", autor: Su-Nombre\_Su-Apellido, isbn: Su RUT (sin puntos y sin guion).
  + Realice una consulta a la API para verificar que uno de los libros (a su elección) fue agregado correctamente.
  + Utilizando la funcionalidad de la API, actualice el título del Libro 1 a: "Guía de Estudio para la Certificación CCNA Security V2.0".

1. **Realice una consulta a la API para ordenar todos los libros del catálogo por autor y asegúrese de que la respuesta incluya el ISBN de cada libro.**
   * Utilizando la funcionalidad de la API, y las librerías de Python, vea la posibilidad de agregar 50 libros adicionales a la biblioteca, estos deben ser generados al azar, tanto en el nombre del autor, como el nombre del libro, como el código isbn
   * Liste la totalidad de los libros creados.
2. **Mejora del Software de Geolocalización con Python:**

Trabajando con el sitio de Graphhopper Directions y el token generado previamente, deberá modificar un script de Python en Visual Studio Code dentro de la VM DEVASC. Las mejoras solicitadas son:

* + Traducir al español todos los textos de interacción con el usuario (solicitudes y respuestas).
  + Asegurar que todos los valores numéricos de salida (distancia, tiempo, etc.) se muestren con un máximo de dos decimales.
  + Implementar una opción para que el usuario pueda salir del programa ingresando "s" o "salir".
  + El programa debe imprimir la narrativa del viaje (las instrucciones paso a paso) en español.

1. **Control de Versiones y Documentación con Git/GitHub:**

Mediante el uso de Git y GitHub, suba el archivo de Python modificado (del software de geolocalización) a un repositorio.

* + El repositorio local y el remoto en GitHub deben llamarse "Evaluacion2" y ser de acceso público.
  + Cree un archivo README.md en su repositorio que incluya una breve descripción del proyecto y las instrucciones de cómo ejecutar el script de Python.
  + Debe agregar el enlace a su repositorio de GitHub en el documento de entrega de la evaluación.

**Debe generar un informe detallado con las evidencias de todos los requerimientos solicitados, estructurado con una introducción, el desarrollo de cada punto y una conclusión.**

**Las capturas de pantalla deben ser claras y representar lo solicitado en cada paso. Es fundamental incluir imágenes que muestren los códigos de estado HTTP 200 (o los correspondientes a "OK") entregados por las APIs en Postman para validar las operaciones de creación, actualización y consulta**

1. **Pauta de Evaluación**

|  |  |  |
| --- | --- | --- |
| **Categoría** | **% logro** | **Descripción niveles de logro** |
| **Muy buen desempeño** | **100%** | Demuestra un desempeño destacado, evidenciando el logro de todos los aspectos evaluados en el indicador. |
| **Buen desempeño** | **80%** | Demuestra un alto desempeño del indicador, presentando pequeñas omisiones, dificultades y/o errores. |
| **Desempeño aceptable** | **60%** | Demuestra un desempeño competente, evidenciando el logro de los elementos básicos del indicador, pero con omisiones, dificultades o errores. |
| **Desempeño incipiente** | **30%** | Presenta importantes omisiones, dificultades o errores en el desempeño, que no permiten evidenciar los elementos básicos del logro del indicador, por lo que no puede ser considerado competente. |
| **Desempeño no logrado** | **0%** | Presenta ausencia o incorrecto desempeño. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Indicador de Evaluación** | **Categorías de Respuesta** | | | | | **Ponderación Indicador de Evaluación** |
| **Muy buen desempeño**  **100%** | **Buen desempeño 80%** | **Desempeño aceptable**  **60%** | **Desempeño incipiente 30%** | **Desempeño no logrado**  **0%** |
| **IL 2.1 - Emplea la sintaxis de un lenguaje de**  **programación para desarrollar scripts simples.** | Utilizando Postman, crea los 3 libros solicitados con  todas las  características. | Utilizando Postman, crea los 2 libros solicitados con  todas las  características. | Utilizando Postman, crea los 1 libros solicitados con todas las características. | Crea los libros solicitados con todas las características, pero no utiliza Postman. | No realiza nada de lo solicitado | **25%** |
| **IL 2.2 - Aplica estructuras de datos y control de flujo para resolver problemas mediante programación**  **estructurada.** | Utilizando Postman con APIs, ordena los libros por autor y que incluya el ISBN  de cada libro | Utilizando Postman con APIs, Pero solo ordena los libros por autor o solo incluye  el ISBN de cada libro | N/A | No utiliza Postman con APIs, para ordenar los libros por autor y que incluya el ISBN de cada  libro | No logra realizar lo solicitado | **25%** |
| **IL 2.3 - Implementa procesos de interacción con servicios** | Utilizando el sitio de Graphhopper  Directions, realiza | Utilizando el sitio de Graphhopper  Directions, solo | Utilizando el sitio de Graphhopper  Directions, solo realiza | Utilizando el sitio de Graphhopper  Directions, solo realiza | No logra realizar lo  solicitado | **25%** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **externos mediante API REST.** | las 4 actividades solicitadas. | realiza 3 actividades solicitadas | 2 actividades solicitadas | 1 actividades solicitadas |  |  |
| **IL 2.4 Desarrolla soluciones que integran el consumo de APIs junto con el uso de Git y GitHub para el control de versiones y la documentación pertinente.** | Se crea el repositorio local y el repositorio en GitHub con el nombre Evaluación 2, se sube correctamente el archivo de Python de Graphhopper  Directions. | Se crea el repositorio local y el repositorio en GitHub con otro nombre, pero se sube correctamente el archivo de Python de Graphhopper Directions. | Se crea el repositorio local y el repositorio en GitHub con el nombre Evaluación 2, pero no sube el archivo de Python de Graphhopper Directions. | Solo crea el repositorio local ó el repositorio en GitHub con el nombre Evaluación 2, pero no sube el archivo de Python de  Graphhopper Directions. | No logra realizar lo solicitado | **25%** |
|  |  |  |  |  |  |  |