**PROTOCOLO INSTRUMENTO DE EVALUACIÓN**

|  |  |
| --- | --- |
| **SEDE:** | **Posadas** |
| **FACULTAD:** | **Ingeniería, Tecnología y Arquitectura** |
| **CARRERA: Ingeniería en Sistemas de Información** | **MODALIDAD (tildar la opción que corresponda)**  X Presencial  ☐Digital |
| **NOMBRE DE LA ASIGNATURA: Programación Estructurada** | **MODALIDAD (tildar la opción que corresponda)**  X Presencial  ☐Presencial Digital  ☐Presencial Mixta  ☐Digital |
| **NOMBRE Y APELLIDO DEL DOCENTE** | **Lic. Vega José María** |
| **NOMBRE Y APELLIDO DEL/LOS ESTUDIANTE/S** | **Polcownuk, Matías Nicolas** |
| **DNI DEL/LOS ESTUDIANTE/S** | **47108706** |
| **FECHA Y HORARIO DE REALIZACIÓN Y/O DE ENTREGA: 01/10/2024** | |

|  |  |  |
| --- | --- | --- |
| **TIPO DE EVALUACIÓN:**  **(tildar la opción que corresponda)** | ☐ Trabajo Práctico I / 1er evaluación (Régimen especial)  ☐Trabajo Práctico II / 2da evaluación (Régimen especial)  X Evaluación Parcial / 3ra evaluación (Régimen especial)  ☐Evaluación Integradora / 4ta evaluación (Régimen especial) | |
| **MODALIDAD DE INSTANCIA EVALUATIVA**  **(tildar la opción que corresponda)** | X Individual  ☐ Parejas  ☐Grupal | X Sincrónica  ☐Asincrónica |
| **CONTENIDOS A EVALUAR:**  **Eje Temático 2: Estructuras de datos Estáticas**  **Eje Temático 3 Estructuras de Datos Dinámicas** | | |
| **CRITERIOS DE EVALUACIÓN**   * Resolución del algoritmo * Capacidad del programa para manejar errores y situaciones excepcionales * Se valorará el uso correcto de buenas prácticas de programación, como la claridad de las variables y la modularización. * Se evaluará que el sistema esté correctamente implementado en ambos lenguajes: Python y C++. La lógica del programa debe ser equivalente en ambas versiones, garantizando que los alumnos comprendan ambos lenguajes * Se valorará la puntualidad * Se valorará la capacidad del alumno para exponer su trabajo con una buena oratoria. * Se premiará la creatividad en la solución, tanto en la eficiencia de los algoritmos como en la experiencia del usuario. * Solo serán evaluados los Códigos ejecutables y pseudocódigos levantados al Aula virtual. | | |
| **CONSIGNAS DE EVALUACIÓN / ASIGNACIÓN DE PUNTAJE**  **Descripción General:**  El objetivo de este parcial es desarrollar un sistema de gestión de biblioteca digital utilizando estructuras dinámicas de datos, métodos de búsqueda y ordenamiento, y manejo de archivos. El sistema debe permitir gestionar libros y lectores, realizando acciones como agregar, buscar, ordenar, solicitar y devolver libros. El desarrollo deberá implementarse en dos lenguajes: **Pytho**n **y C++.** Este proyecto se debe entregar en un plazo de 1 semana, y la evaluación final se realizará mediante una defensa oral presencial.  **Requisitos Técnicos:**   1. **Lenguajes de programación permitidos:**    * El sistema deberá ser implementado en **Python** y **C++**. El alumno podrá elegir en qué lenguaje desarrollar primero, pero ambos lenguajes deben ser utilizados para el desarrollo final. 2. **Estructura de los libros:** Los libros deben estar organizados en una **lista enlazada**, donde cada nodo contiene los siguientes campos:    * Título    * Autor    * Año de edición    * Editorial    * ISBN    * Número de páginas 3. **Estructura de los lectores:** Los lectores deberán gestionarse mediante una **cola de solicitudes de préstamo**. Cada lector puede solicitar un libro a la vez, y se almacenarán los siguientes datos:    * Nombre del lector    * Número de identificación (DNI)    * Libro solicitado (referencia al título del libro en la lista de libros) 4. **Funciones del sistema:** a) **Agregar libros a la lista enlazada:**    * Los libros se cargarán inicialmente desde un archivo llamado biblioteca.txt, que contendrá la información de los libros. El sistema debe permitir también la **adición manual de nuevos libros**. 5. b) **Ordenar libros:**    * Implementar un algoritmo de **ordenamiento de los libros por título**, usando el método de ordenamiento que el alumno elija (burbuja, quicksort, mergesort, etc.), con uso de **recursividad**. 6. c) **Buscar libros:**    * El sistema debe permitir **buscar libros** por título, autor o ISBN, utilizando un método de búsqueda a elección (búsqueda binaria, búsqueda lineal, etc.), y mostrando los detalles del libro encontrado. 7. d) **Solicitar libros:**    * Los lectores pueden solicitar libros, y sus solicitudes deben encolarse en una **cola de préstamos**. Si el libro ya está solicitado por otro lector, el sistema debe informar que el libro está reservado y colocar la solicitud en la cola de espera. 8. e) **Devolver libros:**    * Los lectores pueden devolver libros. Una vez que un libro es devuelto, se debe atender la primera solicitud pendiente en la cola, si la hubiera. 9. f) **Guardar y cargar datos:**    * Al cerrar el sistema, los datos de los libros y las solicitudes de los lectores deben guardarse en dos archivos (biblioteca.txt y solicitudes.txt), y deben cargarse al iniciar el programa. 10. **Manejo de excepciones:**     * Implementar manejo de errores y excepciones para casos como:       + Intentar solicitar un libro que no existe.       + Ingresar datos inválidos (por ejemplo, un ISBN con formato incorrecto).       + Acceder a archivos que no existen. 11. **Historial (Pila de operaciones):**   Implementar un sistema de **pila** que almacene las acciones realizadas (agregar, eliminar, solicitar, devolver), de modo que se puedan **deshacer** las últimas operaciones en orden inverso  **Fechas Importantes:**   * **Entrega del código:** Jueves 31/10/2024. * **Defensa oral y presentación:** Jueves 31/10/2024 y Viernes 1/11/2024.   El día de la defensa, el alumno deberá presentarse con su código listo y preparado para realizar los ajustes que el profesor pueda solicitar. Las defensas se realizarán en el horario de clases asignado. Se sugiere practicar la presentación y tener claro el funcionamiento del sistema.  La entrega será vía aula virtual, enviando el link del repositorio de GitHub, donde se incluirán ambos códigos (Python y C++), debidamente separados en carpetas.  El código presentado deberá ser original, y el uso de herramientas de IA será detectado en caso de dependencia excesiva.  **Recomendaciones:**   * **Comenzar por la estructura:** Es recomendable que los alumnos desarrollen primero la estructura de datos principal (listas enlazadas, colas, etc.), para luego avanzar con las funciones específicas. * **Validación de datos:** Se sugiere implementar las validaciones de entrada de datos desde el inicio, para evitar problemas durante la ejecución.   **Pruebas continuas:** Se aconseja realizar pruebas continuas de las funcionalidades desarrolladas, para asegurar que cada componente funcione correctamente antes de avanzar | | |

Link al GitHub:

https://github.com/ZeroKava/parcial\_programacion\_estructurada