TRABAJO PRÁCTICO INTEGRADOR

Algoritmos y estructuras de datos I

INTRODUCCIÓN

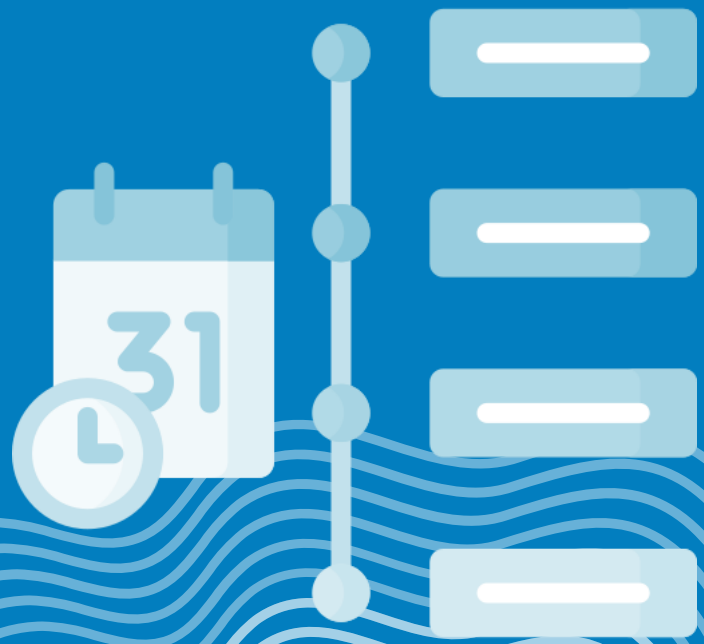
Los estudiantes deberán desarrollar un software completo e incremental como proyecto final para la materia.

Este proyecto tiene como objetivo poner en práctica los conceptos, técnicas y herramientas aprendidas durante el cuatrimestre.

Los alumnos podrán elegir libremente el tipo de software a desarrollar, siempre que cumpla con los requisitos establecidos en cada uno de los hitos.

El proyecto se realizará en tres entregas, cada una con hitos específicos que deben ser alcanzados para considerar la entrega como completa y válida.

Cada instancia de entrega se considerará como una evaluación y la nota será de carácter individual, aunque el desarrollo del proyecto será en equipo.



Lic. Yaps David P.

Temas a evaluar

Matrices. Tuplas. Rebanado y comprensión de listas. Cadena de caracteres. Diccionarios. Conjuntos.

Temas a evaluar

Funciones lambda, map, filter y reduce. Excepciones. Archivos. Expresiones regulares

Temas a evaluar

Repositorio Git. Recursividad. Pruebas unitarias.

Evaluación (Hito 1)

Evaluación (Hito 2)

Final adelantado (Hito 3)

Final Regular (Hito 3)

CRONOGRAMA

El proyecto se realizará en tres entregas incrementales, cada una con hitos específicos que deben ser alcanzados para considerar la entrega como completa y válida.

Lic. Yaps David P.

REQUISITOS DEL PROYECTO

El software deberá ser lo suficientemente amplio y complejo para justificar el uso de todas las temáticas a desarrollar, y debe incluir al menos las siguientes características:

HITO 1

Matrices
Implementar operaciones con matrices como manipulación, búsqueda y modificación de datos.

Tuplas
Emplear tuplas para gestionar conjuntos de datos que no requieren modificación, como configuraciones o constantes.

Rebanado y comprensión de listas
Utilizar técnicas de rebanado para obtener subconjuntos de listas y comprensión de listas para crear listas derivadas.

Cadenas de caracteres
Manipular y procesar cadenas de texto mediante diversas técnicas, funciones y métodos.

Diccionarios y Conjuntos
Utilizar diccionarios para representar asociaciones clave-valor de manera eficiente. Emplear conjuntos para manejar datos sin duplicados y realizar operaciones como intersección o unión.

Lic. Yaps David P.

HITO 2

Funciones lambda, map, filter y reduce
Aplicar funciones lambda y técnicas de programación funcional para transformar y filtrar datos.

Excepciones
Manejar excepciones para controlar errores durante la ejecución del programa.

Archivos
Leer y escribir en archivos para almacenar y recuperar datos.

Expresiones regulares
Validar y buscar patrones en textos utilizando expresiones regulares.

HITO 3

Tests unitarios

Implementar pruebas unitarias para garantizar la funcionalidad del código.

Recursividad

Utilizar funciones recursivas cuando la solución sea naturalmente recursiva o para simplificar la lógica de ciertos algoritmos.

Repositorio Git

Deberán utilizar un repositorio Git público para gestionar y entregar el proyecto. Se deberá mantener un historial de *commits* claro y detallado que refleje el progreso del desarrollo.

EJEMPLO DE IMPLEMENTACIÓN

Para ilustrar cómo se puede implementar cada tema en un proyecto, consideremos un ejemplo de sistema de gestión de tareas personales. A continuación, se detallan cómo se aplicaría cada tema solicitado:

HITO 1

Matrices

Implementar una matriz para manejar un calendario mensual, donde cada fila representa una semana y cada columna representa un día, almacenando las tareas planificadas para cada día.

Tuplas

Utilizar tuplas para almacenar configuraciones inmutables, como el formato de fecha o las prioridades de tareas (baja, media, alta).

Rebanado y comprensión de listas

Aplicar rebanado para obtener las primeras 5 tareas próximas a vencer. Utilizar comprensión de listas para filtrar tareas completadas o con prioridad alta.

HITO 1

Cadenas de caracteres

Manipular cadenas de texto para mostrar mensajes al usuario, formatear la salida de las tareas o combinar múltiples atributos de tareas en un solo *string*.

Diccionarios

Utilizar diccionarios para almacenar las tareas, donde cada clave sea un identificador único de la tarea y el valor sea un conjunto de atributos.

Conjuntos

Usar conjuntos para mantener las etiquetas asociadas a las tareas, asegurando que no haya etiquetas duplicadas.

HITO 2

Funciones lambda, map, filter y reduce

Utilizar *filter* para obtener solo las tareas que no han sido completadas. Emplear *map* para aplicar una función que convierta todas las fechas de vencimiento a un formato legible. Usar *reduce* para calcular el total de tiempo estimado para todas las tareas.

Excepciones

Manejar excepciones cuando el usuario ingrese datos inválidos, como una fecha incorrecta o un comando no reconocido.

HITO 2

Archivos

Utilizando el formato JSON, guardar la lista de tareas en un archivo al finalizar el programa y cargarla al iniciar para mantener la persistencia de datos.

Expresiones regulares

Validar el formato de las fechas ingresadas por el usuario con una expresión regular para asegurarse de que estén en el formato correcto (por ejemplo, DD/MM/AAAA).

Lic. Yaps David P.

HITO 3

Tests unitarios

deberían desarrollar pruebas para confirmar que, al agregar una tarea, esta se incluya correctamente en la lista de tareas; al eliminar una tarea, ya no aparezca en la lista; y al listar las tareas, se muestren en el orden y formato correctos.

Recursividad

Implementar una función recursiva para buscar una tarea específica dentro de estructuras de datos anidadas, como subtareas dentro de tareas principales.

Repositorio Git

Deberían crear ramas para desarrollar nuevas características, fusionarlas con la rama principal cuando estén completas. Además, deben asegurarse de que su repositorio sea accesible para el profesor o evaluador, permitiendo así la revisión del código y del historial de desarrollo.

MODALIDAD DE EVALUACIÓN

La evaluación del proyecto será un proceso que combinará el trabajo en equipo con una calificación individual.

Aunque el proyecto se desarrollará en colaboración, cada miembro del equipo será evaluado de manera individual para reflejar su contribución específica.

Para el proyecto en general, se utilizará una rúbrica común que servirá como una guía clara para evaluar cada uno de los puntos expuestos. Esta rúbrica incluirá criterios específicos como el cumplimiento de objetivos, la calidad del contenido, la creatividad, la colaboración, la puntualidad, y la participación activa en el trabajo en equipo.

Durante el proceso, se hará un control detallado de cada criterio para garantizar una evaluación justa y precisa, teniendo en cuenta tanto el desempeño colectivo del equipo como las aportaciones individuales de cada miembro.

Lic. Yaps David P.

MATERIAL DE CLASE

Por medio de la lectura del código QR, se podrá ingresar al material de clase completo.

Lo cual, favorecerá el aprendizaje autónomo y continuo fuera del aula; permitiendo revisar y reforzar los conceptos presentados en clase, profundizar en los temas tratados y ayudar a superar posibles obstáculos para el aprendizaje; y asegurar que podamos alcanzar nuestro máximo potencial académico.