ESTRUCTURAS DE DATOS Y ALGORITMOS I y II

M. en I. Fco. Javier Rodríguez G
Facultad de Ingeniería, UNAM
Version: 1.0, 24/01/21

Pautas para el proyecto

2021-1

Objetivo

El objetivo del proyecto es que apliquen los conocimientos adquiridos durante el curso a una aplicación de la vida real.

Objetivos específicos

1. **Objetivo1:** (pendiente)
2. **Objetivo2:** (pendiente)

ESPECIFICACIONES

El proyecto deberá realizarse en equipo y se entregará la primera semana de finales.

EQUIPOS

EDA I: El equipo único es de 4 integrantes.

EDA II: Cada equipo deberá estar conformado entre 2 y 3 integrantes. Ni menos, ni más. No se permiten equipos de 1 persona.

TEMA

Cada equipo buscará y escogerá un tema que tenga aplicación en la vida real.

Más adelante les enviaré un poco de ayuda para escoger un tema adecuado que puedan realizar en el tiempo establecido.

Si no tienen ideas por favor avísenme para asignarles un proyecto. Ésto último no afectará (no bajará puntos) la evaluación del proyecto.

ELEMENTOS DE EVALUACIÓN

De manera general el proyecto deberá utilizar los siguientes elementos (en la rúbrica que les entregaré más tarde están especificados de manera particular):

EDA I

1. Dos tipos abstractos de nivel superior. Éstos representarán la esencia de su aplicación. Por ejemplo, si su proyecto es sobre inventarios de productos, entonces ustedes deberán tener una API para representar cada **producto**, y una API para representar una **lista** de productos. Un programador que quisiera utilizar el trabajo que ustedes están realizando utilizará este par de APIs, y no se meterá con detalles de la implementación. Por supuesto que podrían tener solamente un tipo abstracto de nivel superior, pero deberán justificarlo.

2. Tipos abstractos para las otras entidades que formen a su aplicación (pilas, colas, listas enlazadas). Los algoritmos NO requieren tipos abstractos.
3. Si usan **listas** éstas deberán estar implementadas como listas doblemente enlazadas, DLLs, para tipos abstractos. Podrán usar DLLs circulares de ser necesario.
    - Si utilizan colas, éstas deberán estar basadas en DLLs.
    - Si utilizan pilas, éstas deberán estar basadas en arreglos.
4. Deberán implementar al menos un algoritmo: voraz, de programación dinámica, o de backtracking. Úsen a este último si están considerando utilizar fuerza bruta.
5. El o los algoritmos que utilicen deberán ser recursivos, a menos que puedan fundamentar la utilización del modelo iterativo (desconocimiento o falta de tiempo no son opciones).
6. Su aplicación deberá estar manejada por menú (*menu driven*).
7. Su programa deberá estar organizado en módulos, con codificación limpia, documentado con Doxygen, y estar alojado en un repositorio Git. Los tipos abstractos de nivel superior DEBEN estar documentados.

EDA II

- Un tipo abstracto de nivel superior. Éste representará la esencia de su aplicación. Por ejemplo, si su proyecto es sobre administración de vuelos de una aerolínea, entonces ustedes deberán tener una API para representar para representar una **grafo** de destinos. Un programador que quisiera utilizar el trabajo que ustedes están realizando utilizará esta API, y no se meterá con detalles de la implementación.
- La API del tipo abstracto de nivel superior deberá incluir operaciones para búsqueda y serialización (Serialize(), Deserialize()), y opcionalmente, para guardar y extraer la información en una base de datos (Save(), Load()).
- Tipos abstractos para las otras entidades que formen a su aplicación (pilas, colas, listas enlazadas). Los algoritmos NO requieren tipos abstractos.
- Si usan **listas** éstas deberán estar implementadas como listas doblemente enlazadas, DLLs, para tipos abstractos.

    - Si utilizan pilas o colas, ambas deberán estar basadas en DLLs.
- Deberán implementar al menos los algoritmos: QuickSort y búsqueda binaria.
- Deberán utilizar al menos una de las siguientes estructuras de datos: tabla Hash, Grafo, o Árbol.
- Deberán utilizar un método de serialización: XML, JSON o YAML. Pueden usar bibliotecas de terceros.
- Deberán paralelizar al menos un algoritmo (búsqueda, ordenamiento, recorridos en árboles y grafos, etc).
- Pueden utilizar la base de datos SQLite.
- Su aplicación deberá estar manejada por menú (*menu driven*).
- Su programa deberá estar organizado en módulos, con codificación limpia, documentado con Doxygen, y estar alojado en un repositorio Git. El tipo abstracto de nivel superior DEBE estar documentado.

INTERFAZ DEL USUARIO

El usuario deberá interactuar con el programa a través de menús en una consola de texto. No están permitidas las interfaces gráficas.

LENGUAJE Y PLATAFORMA

El lenguaje de programación será:

- EDA I: **C**
- EDA II: **C** ó **C++**

El programa podrá ser desarrollado en Windows o Mac, pero deberá compilar y correr desde Linux. Recuerden que yo no uso IDEs y no las voy a instalar.

ORGANIZACIÓN

De preferencia el programa deberá estar organizado en módulos, uno por cada tipo abstracto y estructura de datos utilizado.

Los algoritmos NO pueden considerarse para ser convertidos en módulos, pero deberán estar en su propio archivo (es decir, fuera de la función main()).

Dejar todo el código fuente en un mismo archivo será válido (siempre y cuando funcione) pero les bajará (muchísimos) puntos.

DOCUMENTACIÓN

De preferencia todas sus funciones *públicas* deberán estar documentadas en el formato Doxygen. Utilizar otro formato (uno inventado por Ud) o no documentar le bajará puntos.

REPORTE

Además del código fuente, deberán entregar de manera obligatoria un pequeño reporte. Les mandaré la plantilla más adelante.

EVALUACIÓN

Los proyectos se evaluarán con una rúbrica que les enviaré un poco más adelante.

OTROS

Cualquier asunto no tratado aquí, nos pondremos de acuerdo para solucionarlo.