Título del proyecto a realizar

Primer autor, Segundo autor, Tercer autor

No. de equipo de trabajo: {*Número de Equipo de trabajo*}

I. INTRODUCCIÓN

En esta sección se describe de manera general en qué consiste este documento y su contenido.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

En esta sección se debe describir de manera general el problema que se propone resolver en el proyecto de clase. Se debe especificar el propósito u objetivo general.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

En esta sección se deben mencionar las características y clasificación de los usuarios (perfiles/roles) que utilizarán el producto. La clasificación puede ser en función a la frecuencia de uso, grupo de funcionalidades utilizadas, privilegios de acceso y seguridad, nivel de experiencia y otros aspectos que considere pertinentes.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Los requerimientos funcionales de un sistema son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

Entre los posibles requerimientos funcionales de un sistema, se incluyen:

- Descripciones de los datos a ser ingresados en el sistema.
- Descripciones de las operaciones a ser realizadas en cada pantalla que se presenta.
- Descripción de los flujos de trabajo realizados por el sistema.
- Descripción de los reportes del sistema y otras salidas.
- Definición de quiénes pueden ingresar datos en el sistema.

De esta manera, se deben describir las interacciones que tendrán los usuarios con el software.

Cada funcionalidad se debe especificar así:

- *Nombre de la funcionalidad*

En el título de la funcionalidad, se recomienda utilizar nombres muy descriptivos para cada funcionalidad. No limitarse a nombrarlas “Funcionalidad 1”; en cambio, usar por ejemplo “Autorización de pedido de compra”.

- *Descripción:*
  - Descripción breve de la funcionalidad.
- *Acciones iniciadoras y comportamiento esperado*: Secuencia de acciones del usuario y respuestas esperadas del programa para esta funcionalidad.

Requerimientos funcionales:

Lista detallada de los requerimientos funcionales asociados a esta funcionalidad.

Para cada requerimiento funcional se establece cómo debe mostrarse el software y cuáles comportamientos debe desempeñar para que el usuario pueda realizar la función que necesita.

Es muy importante prever y describir cómo debe responder el software ante condiciones de error y entradas de datos inválidas.

Las funcionalidades mínimas sobre los datos que se manejen deben prever operaciones de:

- Creación
- Actualización
- Eliminación
- Consulta total de los datos
- Búsqueda parcial de datos
- Ordenamiento
- Almacenamiento

Aunque en otros cursos se estudian estrategias de organización y almacenamiento, en este curso, el almacenamiento se requiere principalmente para facilitar las pruebas del prototipo de software. Por lo anterior, para facilitar su implementación, se deja abierta la opción a que se apoyen en el uso de sistemas manejadores de bases de datos, o se haga almacenamiento por archivos, de objetos u otra estrategia que les convenga, siempre que se garantice la implementación y uso de las estructuras de datos vistas en clase en la memoria en el tiempo de ejecución del software.

IMPORTANTE: En cada una de las entregas para reportar el avance en el desarrollo del proyecto, se especificarán las funcionalidades mínimas y las estructuras de datos mínimas requeridas que se deben implementar. También, se debe presentar un análisis (comparativo y asintótico) de la eficiencia de las estructuras de datos usadas.

NOTA: En el siguiente enlace web (URL) puede encontrar una explicación de cómo diferenciar Requisitos Funcionales de los No Funcionales https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Se debe incluir una descripción general de la interfaz de usuario del software. Al proponer el desarrollo de una interfaz de usuario gráfica, deberán usar *mockups* (bosquejos, fotomontajes o figuras) que le permitan al futuro usuario visualizar cómo quedarán los diseños.

NOTA: Existen múltiples herramientas que ayudan en este tipo de tareas. Por ejemplo, podrán encontrar una herramienta útil para esta labor disponible en: https://balsamiq.com/

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

En esta sección se debe describir el entorno en el que se desarrollará el software, así como el entorno en el que funcionará cuando se ponga en operación, para esto último se debe especificar el sistema operativo y el hardware sobre el que operará.

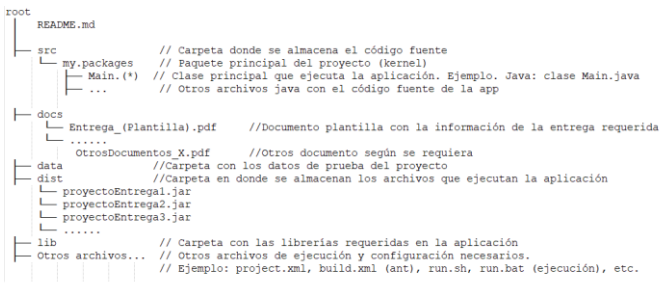
VII. PROTOTIPO DE SOFTWARE INICIAL

Para esta entrega de avance en el desarrollo del proyecto, se debe realizar una primera versión de un prototipo de software funcional de acuerdo con los requisitos que se explican a continuación.

El software desarrollado se debe registrar en un repositorio de software de Github. Para facilitar el uso de esta plataforma, se sugiere estudiar el tutorial disponible en

https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud

Se debe organizar el software en el repositorio de una manera estructurada como se ilustra en la siguiente figura:



Además, para mantener una versión gráfica de desarrollo del repositorio, ustedes se podrán apoyar en el uso de herramientas como Sourcetree, disponible en el siguiente URL: https://www.sourcetreeapp.com/

En este prototipo se deben implementar y aplicar algunas estructuras de datos como se detalla en la siguiente sección.

VIII. DISEÑO, IMPLEMENTACIÓN YAPLICACIÓN DE LAS ESTRUCTURAS DE DATOS

En este prototipo se debe implementar por lo menos una instancia de cada una de las estructuras de datos secuenciales (llamadas también lineales o unidimensionales): arreglos, listas encadenadas, pilas y colas. Se reitera que en este tipo de contenedores de datos, los datos se organizan de una manera secuencial o lineal (como en una hilera) de tal manera que cada dato puede tener a lo sumo un antecesor y a lo sumo un sucesor.

Para estas estructuras de datos se deben soportar por lo menos las siguientes operaciones funcionales:

- Creación
- Inserción de un solo dato
- Actualización de un solo dato
- Eliminación de un solo dato
- Búsqueda de un solo dato
- Consulta de todos los datos
- Almacenamiento de los datos

Se debe realizar una descripción del diseño orientado pro objetos, especificando las clases y como estas son usadas para resolver el problema. También, para cada una de las diferentes estructuras de datos implementadas se debe describir brevemente el tipo de implementación realizada y su contribución para llevar a cabo las funcionalidades del prototipo de software.

IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Se deben realizar y documentar las pruebas del prototipo para algunos ejemplos (casos) de prueba para las funcionalidades que tomen más tiempo y realizar un análisis comparativo así:

- Escoger entre tres y cinco funcionalidades que sean las de mayor costo computacional en tiempo;
- Para cada funcionalidad se deben realizar pruebas para varios tamaños de datos de prueba (n), por lo menos para los siguientes valores de n:
    - 10 mil datos,
    - 100 mil datos,
    - 1 millón de datos,
    - 10 millones de datos, y
    - 100 millones de datos.
- Hacer una tabla comparativa de los tiempos que toma realizar cada una de las funcionalidades consideradas para los diferentes tamaños de los datos de prueba.
- Determine y grafique el correspondiente análisis asintótico comparativo entre las estructuras implementadas y su respectiva complejidad, de acuerdo con las pruebas realizadas. Para esto deben usar los datos y tiempos de los ejemplos de prueba; asimismo, se debe realizar el análisis asintótico usando por lo menos la notación O grande (*Big O*).

X. INFORMACIÓN DE ACCESO AL VIDEO DEMOSTRATIVO DEL PROTOTIPO DE SOFTWARE

Aquí se debe incluir el enlace web (URL) para tener acceso al video demostrativo del prototipo de software funcional, en youtube o donde corresponda.

XI. ROLES Y ACTIVIDADES

Se deben describir de manera concisa los roles asignados y las respectivas actividades realizadas por cada uno los integrantes del equipo durante el desarrollo de esta entrega. Para esto, tenga en cuenta los siguientes roles positivos, que puede aplicar durante el desarrollo del proyecto; los mismos deberán ser 'rotados' entre todos los integrantes del equipo en las diferentes entregas.

Rol	Actividades fundamentales
Líder(esa)	Consultar a los otros miembros del equipo, atento que la información sea constante para todos. Aportar con la organización y plan de trabajo.
Coordinador(a)	Mantener el contacto entre todos, Programar y agendar y reuniones; ser facilitador para el acceso a los recursos.
Experto(a)	Líder técnico que propende por coordina funciones y actividades operativas.
Investigador(a)	Consultar otras fuentes. Propender por resolver inquietudes comunes para todo el equipo.
Observador(a)	Siempre está atento en el desarrollo del proyecto y aporta en el momento apropiado cuando se requiera apoyo adicional por parte del equipo.
Animador(a)	Energía positiva, motivador en el grupo.
Secretario(a)	Se convierte en un facilitador de la comunicación en el grupo. Documenta (actas) de los acuerdos/compromisos realizados en las reuniones del equipo.
Técnico(a)	Aporta técnicamente en el desarrollo del proyecto.

Puede utilizar la siguiente tabla para definir los integrantes del grupo, los roles asignados y el listado de actividades durante el desarrollo de la entrega:

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)

XII. DIFICULTADES Y LECCIONES APRENDIDAS

Mencione las dificultades encontradas durante el desarrollo del proyecto. Además, haga alusión a las principales lecciones aprendidas durante el proceso.

XIII. REFERENCIAS BIBLIOGRÁFICAS

[1] Weiss, M.A.: *Data Structures and Algorithm Analysis in C++, 4th Edition*, Pearson/Addison Wesley, 2014.

[2] Hernández, Z.J. y otros: *Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++*, Thomson, 2005.

[3] Shaffer, Clifford A.: *Data Structures and Algorithm Analysis in C++, Third Edition*, Dover Publications, 2013. (En línea.)

[4] Campos Laclaustra, J.: *Apuntes de Estructuras de Datos y Algoritmos*, segunda edición, 2018. (En línea.)

[5] Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos. 2ª Edición, Ed. Garceta, 2013.

[6] Joyanes, L., Zahonero, I., Fernández, M. y Sánchez, L.: Estructura de datos. Libro de problemas, McGraw Hill, 1999.