Logotipo

Descripción generada automáticamente con confianza media

**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA**

**CARRERA DE INGENIERÍA DE SOFTWARE**

**CARRERA DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA DE SISTEMAS DE LA INFORMACIÓN**

**CICLO 2022-01**

**ALGORITMOS Y ESTRUCTURAS DE DATOS – CC182**

**Sección SV31**

**INFORME DE PRÁCTICA CALIFICADA**

**“GESTIÓN DE TAREAS”**

**PROFESOR DEL CURSO**

Ricardo Martín Dulanto Ramírez

**INTEGRANTES**

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **CÓDIGO** | **CARRERA** |
| Arroyo Ormeño, André Alonso | U202114714 | Ingeniería de Software |
| Comettant Rubiños, Jessica Elizabeth | U20211C009 | Ciencias de la Computación |
| Huarcaya Chavez, Miguel Alejandro Daniel | U202116207 | Ingeniería de Software |

**Enlace de github:**

<https://github.com/Huarzurita/Algoritmo_Grupo7_PC1>

**FECHA DE ENTREGA MÁXIMA**

17/04/2022

**SEDE MONTERRICO-VILLA, abril del 2022**

**Índice**

1. [Introducción 3](#_Toc99961179)
2. [Objetivos 4](#_Toc99961179)
3. [Diseño de clases 5](#_Toc99961179)
4. [Fundamento teórico del código 6](#_Toc99961179)
5. [Participación del equipo 7](#_Toc99961179)
6. [Anexos 8](#_Toc99961179)
7. **INTRODUCCIÓN**

El presente trabajo busca evidenciar el aprendizaje adquirido por las/los estudiantes del curso de Algoritmos y Estructuras de datos durante las primeras semanas de la unidad N°01. Ello se demostrará a través de la implementación de un gestor de tareas con el lenguaje de programación c++ en el programa Visual Studio 2019. Las técnicas de programación utilizadas enfatizan el uso de los *tipos de datos abstractos, la complejidad asintótica de algoritmos y algoritmo recursivo*. Asimismo, se maneja diferentes tipos de estructuras de datos como *vectores y arreglos*. Todo esto con la finalidad de alcanzar el logro del curso y las competencias postuladas por la universidad (Pensamiento Innovador y Diseño y desarrollo de una solución) con relación a la carrera de Ciencias de la Computación e Ingeniería de Software.

1. **OBJETIVOS**

Para Ciencias de la Computación:

* Diseñar, implementar y evaluar una solución basada en la computación para cumplir con un conjunto de requisitos computacionales en el contexto de la disciplina del programa.

Para Ingeniería de Software:

* Capacidad de aplicar el diseño de ingeniería para producir soluciones que satisfagan necesidades específicas con consideración de salud pública, seguridad y bienestar, así como factores globales, culturales, sociales, ambientales y económicos.

1. **DISEÑO DE CLASES**

Fig.1 Diagrama de clases de la implementación del código

Diagrama

Descripción generada automáticamente

En el diagrama, se visualiza la cantidad de entidades, nombre, métodos y tipo de variables empleadas por el equipo de trabajo.

1. **FUNDAMENTO TEÓRICO DEL CÓDIGO**

* **USO DEL PARADIGMA POO (PROGRAMACIÓN ORIENTADA A OBJETOS)**

El equipo de trabajo empleó los principios fundamentales de Abstracción y Encapsulamiento para la creación de atributos y métodos en cada clase. A través del *encapsulamiento*, las clases *CNota, CTarea, CRecordatorio, CEvento y Controller* fueron capaces de agrupar los métodos (*gets*) y variables requeridas en la implementación (*título, descripción, fecha, hora, urgencia, lugar y completado)* lo que nos permitió incrementar la cohesión del sistema*. Public* nos permitió acceder a los datos de forma abierta, mientrasque *private* otorgaba seguridad a los datos propios de cada clase. Asimismo, la *abstracción* nos facilitó la representación de los atributos y métodos necesarios para lograr concretar la creación de la *app* minimalista de *Microsoft To Do.* Ello convierte a la clase en un nivel superior de abstracción.

* **USO DE TEMPLATES:**

El equipo de trabajo aplicó el uso de *templates* (*titulo, descripción, fecha, hora, urgencia, lugar y completado*) como tipo de dato abstracto fundamental para toda la implementación del código. Gracias a este *TDA,* se evitó la sobrecarga de funciones y variables para diferentes tipos de datos como *int, string,* entre otros lo que facilito el ahorro de líneas de código (programación genérica).

* **USO DE LAMBDAS**

El equipo de trabajo implementó la función anónima *lambda* en la estructura principal del código (*main*) con la finalidad de utilizar las funciones *mostrarTareas(),* *mostrarEventos(), mostrasNotas()* y *mostrarRecordatorios(),* los cuales permiten visibilizar en la consola la información insertada por el usuario en dependencia del tipo de apunte seleccionado.

* **USO DE ESTRUCTURAS DE DATOS**

Las estructuras de datos utilizadas por el equipo de trabajo fueron los *arreglos* y los *vectores*. A través de la librería *vector*, se accedió a métodos (.pushback(), .erase()) que posibilitaron un manejo más dinámico de la memoria.

1. **PARTICIPACIÓN DEL EQUIPO**

|  |  |
| --- | --- |
| **ACTIVIDADES** | **RESPONSABLE** |
| * Realización del informe. * Colaboración con el código del main.cpp y la clase *Controller*. * Creación de las clases *CNotas*, *CEvento, CRecordatorio y CTarea.* | Arroyo Ormeño, André Alonso |
| * Realización del informe * Colaboración con el código del main.cpp y la clase *Controller* * Creación de las clases *CNotas*, *CEvento, CRecordatorio y CTarea* | Comettant Rubiños, Jessica Elizabeth |
| * Realización del informe. * Colaboración con el código del main.cpp y la clase *Controller*. * Creación de las clases *CNotas*, *CEvento, CRecordatorio y CTarea.* | Huarcaya Chavez, Miguel Alejandro Daniel |

1. **ANEXOS**

Diagrama

Descripción generada automáticamente

Fig.2 Diagrama de flujo inicial del código