**"Año de la lucha contra la corrupción y la impunidad"**

**Universidad Peruana De Ciencias Aplicadas (UPC)**

****

**Ingeniería de Software**

**2019-02**

**Trabajo final**

**Curso:**

Algoritmos y estructura de datos

**Sección:**

SV31

**Profesor:**

Carlos Alberto Jara Garcia

**Equipo:**

**Aplicación:**

Desarrollo de un archivo único de identificación (AUI)  basado en un explorador de carpetas.

**Integrantes:**

·         Eduardo Lenin Ortega Esquivel

·         Ramírez Zarate Denilson Nider

·         Huapaya Buitron Ariana

Contenido

[1. Introducción 3](#_Toc24230260)

[2. Objetivo del Estudiante (Student Outcome) 3](#_Toc24230261)

[3. Marco Teórico 4](#_Toc24230262)

[3.1. Explorador de Windows 4](#_Toc24230263)

[3.1.1. Definición 4](#_Toc24230264)

[3.1.2. Historia 4](#_Toc24230265)

[3.1.3. Cambios en el Explorador de Windows 5](#_Toc24230266)

[3.2. Archivo Único de Identificación (AUI) 6](#_Toc24230267)

[3.2.1. Definición del Proyecto 6](#_Toc24230268)

[3.2.2. Uso del Explorador de Windows 6](#_Toc24230269)

[4. Diseño del Proyecto 7](#_Toc24230270)

[5. Diseño del producto 7](#_Toc24230271)

[5.1. Definición 7](#_Toc24230272)

[5.2. Definición de requisitos funcionales y no funcionales 7](#_Toc24230273)

[5.3. Diagrama de clases 8](#_Toc24230274)

[5.4. Interfaz de usuario 9](#_Toc24230275)

[5.5. Tipos de datos abstractos 9](#_Toc24230276)

[5.6. Selección de estructuras de datos 10](#_Toc24230277)

[5.6.1. Archivos 10](#_Toc24230278)

[5.6.2. Lamba 10](#_Toc24230279)

[5.6.3. Listas 10](#_Toc24230280)

[5.6.4. Recursividad 10](#_Toc24230281)

[5.6.5. Pilas y Colas 11](#_Toc24230282)

[5.6.6. Arboles Binarios 11](#_Toc24230283)

[6. Diseño del proceso de desarrollo de la solución 11](#_Toc24230284)

[6.1. Diseño de la arquitectura del software 11](#_Toc24230285)

[6.2. Implementación de las funcionalidades del software. 11](#_Toc24230286)

[6.3. Pruebas del software 11](#_Toc24230287)

[7. Conclusiones 11](#_Toc24230288)

[8. Bibliografía 12](#_Toc24230289)

[9. Anexos (ppt, video, otros) 12](#_Toc24230290)

# **Introducción**

En el presente proyecto desarrollamos  un Archivo Único de Identificación (AUI), donde se simulará el almacenamiento de información, en un explorador de carpetas, que serán denominadas de acuerdo al número de identidad de las personas. Nos basaremos en los principios del Explorador de Windows, ya que nos permitirá guiarnos durante el desarrollo de la aplicación

El proyecto consiste en un archivo único de identificación (AUI), el cual será atribuido a cada una de las personas. Las carpetas  principalmente almacenarán información general como: nombres completos, número de identificación, fecha de nacimiento, lugar de nacimiento, sexo, estado civil, dirección, saldo en tarjetas o monedero virtual. De esta manera todos los datos de una persona serán virtuales facilitando así a las entidades públicas el fácil manejo y uso de esta información.

La ampliación constará de una ventana de registro por el cual se podrá agregar la carpeta de una persona a la base de datos. Asimismo, poseerá una herramienta de búsqueda de personas registradas, y una herramienta que elimina por completo carpetas con toda su información general. Además, se desarrollará una función de clasificación, el cual se encargará de filtrar las carpetas por criterios y una función de ordenamiento de carpetas de manera, ya sea de manera ascendente o descendente.

# **Objetivo del Estudiante (Student Outcome)**

La 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.

# **Marco Teórico**

## **Explorador de Windows**

### **Definición**

De manera general, el explorador de Windows consiste en una aplicación para administrar archivos que forma parte del sistema operativo Microsoft Windows. A través de esta herramienta es posible crear, editar o eliminar carpetas, archivos, etc.

### **Historia**

El Explorador de Windows se incluyó por primera vez en Windows 95 como un reemplazo para el administrador de archivos Windows 3.x.​ A lo largo de la historia de Windows, se añadieron o quitaron características; mientras que el "explorador de Windows" es un término usado comúnmente para describir este aspecto del sistema operativo, el proceso del explorador también contiene la funcionalidad y el tipo asociaciones (basadas en extensiones del nombre de fichero), y es responsable de exhibir los iconos correspondientes: el menú inicio, la barra de tareas, y el panel de control. Desde Windows Vista el explorador tiene soporte para mostrar metadatos, hasta 267 columnas iniciales más adicionales especificados por una App cuando se instala (por ejemplo, un documento DOCX sólo mostrará campos Comentarios y Fecha de Impresión después de instalar Microsoft Office). El Explorador de Windows también es conocido como el Administrador de Archivos.

## **Cambios en el Explorador de Windows**

El Explorador de Windows incluye cambios significativos tales como filtración mejorada, clasificar, agrupar y apilar. Combinado con la búsqueda de escritorio integrada, el Explorador de Windows permite que los usuarios encuentren y que organicen sus archivos de nuevas maneras. En la parte inferior, se agregó el "Panel detalles" en el que se permite agregar, cambiar o modificar la información de archivos como imágenes, música, vídeos y documentos. El Explorador de Windows Vista, Windows Server 2008 y Windows 7, introduce un diseño innovador.​Los paneles de tareas de Windows XP y otras versiones anteriores fueron sustituidos por una barra de herramientas en la parte superior de la ventana.

## **Archivo Único de Identificación (AUI)**

### **Definición del Proyecto**

El proyecto consiste en un archivo único de identificación (AUI), el cual será atribuido a cada una de las personas. Las carpetas principalmente almacenarán información general como: nombres completos, número de identificación, fecha de nacimiento, lugar de nacimiento, sexo, estado civil, dirección, saldo en tarjetas o monedero virtual. De esta manera todos los datos de una persona serán virtuales facilitando así a las entidades públicas el fácil manejo y uso de esta información

### **Uso del Explorador de Windows**

La ampliación constará de una ventana de registro por el cual se podrá agregar la carpeta de una persona a la base de datos. Asimismo, poseerá una herramienta de búsqueda de personas registradas, y una herramienta que elimina por completo carpetas con toda su información general. Además, se desarrollará una función de clasificación, el cual se encargará de filtrar las carpetas por criterios y una función de ordenamiento de carpetas de manera, ya sea de manera ascendente o descendente.

# **Diseño del Proyecto**

En el presente trabajo se ha divido en tres hitos, cada hito tiene sus respectivos requisitos ha implementar en nuestro proyecto. Es por ello, que hemos decidido crear un diagrama de Gantt, para poder organizar lo que ha cada integrante le corresponde a realizar. Esto se debe que es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto y es justo que se necesita en nuestro proyecto, porque durante todo el tiempo que corresponda hasta la entrega final y para asegurarse del cumplimiento de los puntos señalados, se seguirán añadiendo o eliminado las actividades y para documentar que cada integrante cumple con su tarea.

* Diagrama de Gantt

# **Diseño del producto**

## **Definición**

Como se ya se mencionado anteriormente, el Archivo Único de Identificación (AUI) es un proyecto que, principalmente almacena información general de una persona, como: nombres completos, número de identificación, fecha de nacimiento, lugar de nacimiento, sexo, estado civil, dirección. Pero de la misma forma, saldo en tarjetas o monedero virtual, por lo que este proyecto es muy útil en varios casos. Entre ellos podemos considerar, con respecto, a la salud pública, en la actualidad

## **Definición de requisitos funcionales y no funcionales**

Para definir los requisitos es importante mencionar los requerimientos(epics) del producto.

### **Visualizar los archivos del programa en un TreeView.**

1. **Requisitos Funcionales**

* Los archivos de del programa se visualizarán de manera vertical.

1. **Requisitos no Funcionales**

* Los archivos se lograrán visualizar debido a la función cargarFile(), ya que este realizara un cargado de archivos constantemente.

### **Visualizar el contenido de cada archivo en un ListView**

* + - 1. **Requisitos Funcionales**
* Se visualizará el contenido textual que contiene cada archivo .txt.
  + - 1. **Requisitos no Funcionales**
* Para lograr visualizar los archivos .txt se usará la función lecturaArchivos():

### **Buscar archivos de acuerdo a un criterio**

* + - 1. **Requisitos Funcionales**
* A través de un textBox se realizará la búsqueda de archivos.
  + - 1. **Requisitos no Funcionales**
* Este requerimiento usara el principio de búsqueda de arboles AVL.

### **Filtrar los archivos de acuerdo a un criterio**

* + - 1. **Requisitos Funcionales**
* Se visualizarán los archivos filtrados se acuerdo a un criterio en el ListView haciendo click en una tira de herramientas(toolStrip).

### **Ordenar los archivos de acuerdo a un criterio**

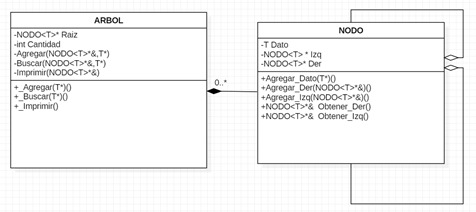
1. **Requisitos Funcionales**

* Se visualizará el ordenamiento de los archivos de acuerdo al criterio del filtrado haciendo click en un botón.

## **Cuadro de Riesgos**

|  |  |  |
| --- | --- | --- |
| Nª | Cuadro de riesgos | Impacto |
| 1 | El software desarrollado puede tener problemas compatibilidad con distintos sistemas operativos. | Alto |
| 2 | La computadora donde se está desarrollando el software puede ser afectada por un virus informático. | Alto |
| 3 | El hardware donde se está almacenando el proceso del programa puede dañarse. | Medio |

## **Diagrama de clases**



## **Interfaz de usuario**

//Formulario

## **Tipos de datos abstractos**

Como se muestra en nuestro diagrama de clase, las principales operaciones que se ha usado en nuestro proyecto son:

* La clase “NODO”
  + Agregar\_Dato(T)
  + Agregar\_Der()() y Agregar\_Izq()()
  + Obtener\_Der() y Obtener\_Izq()
* La clase “ARBOL”
  + \_Agregar()
  + \_Buscar()
  + \_Imprimir()

## **Selección de estructuras de datos**

### **Archivos**

Como sabemos, los archivos o ficheros en el Visual Studio C++, permiten el acceso al disco. Una forma para ingresar o extraer de una carpeta determinada, un archivo en específico. Así pues, en nuestro proyecto lo hemos incluido con esa finalidad, ya que necesitamos guardar todos los datos de la persona y, al mismo tiempo, extraer esa información con la finalidad de facilitar su manejo y uso a las entidades públicas.

### **Lamba**

Una expresión lambda es una función o subrutina sin un nombre que se puede usar siempre que las condiciones que se señalen sean válidas. De esta manera, las expresiones lambda pueden ser funciones o subrutinas y pueden ser de una o varias líneas. En nuestro caso lo hemos usado como———, por eso——-.

### **Listas**

Las listas son un conjunto de elementos llamados nodos conectados mediante un puntero, y una de sus grandes ventajas frente a los arreglos es que son dinámicos por defecto, al mismo tiempo que Insertar o eliminar un elemento a la mitad de la lista no implica mover el resto de elementos.

### **Recursividad**

Este método sirve para definir una función en términos de sí misma, o como se mención en términos de programación, es una función que “para cumplir su objetico se llama así misma”. Es por eso que, es una buena estrategia para resolver problemas al simplificar el problema que progresivamente en cada llamada se va usar a la misma función.

### **Pilas y Colas**

Las pilas y colas son conjuntos dinámicos que cuando se realiza la operación “Eliminar” los elementos eliminados se especifican previamente. Pero en lo que se diferencian es : En la pila, el elemento que se elimina es que se insertó recientemente. Mientras que en la cola, el conjunto que ha permanecido por más tiempo.

### **Arboles Binarios**

En nuestro proyecto hemos hecho un gran uso de los arboles binarios, de la misma forma que la Busqueda Binaria AVL, eso se debe ha que este ultimo es un algoritmo mucho mas eficiente para encontrar un elmentento en una lista ordenada.

# **Diseño del proceso de desarrollo de la solución**

## **Diseño de la arquitectura del software**

//Codigo

## **Implementación de las funcionalidades del software.**

//Codigo

## **Pruebas del software**

//Codigo

# **Conclusiones**

# **Bibliografía**

# **Anexos (ppt, video, otros)**