**"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** 4](#_Toc25010908)

[**2.** **Objetivo del Estudiante (Student Outcome)** 5](#_Toc25010909)

[**3.** **Marco Teórico** 5](#_Toc25010911)

[**3.1.** **Explorador de Windows** 5](#_Toc25010912)

[**3.1.1.** **Definición** 5](#_Toc25010913)

[**3.1.2.** **Historia** 5](#_Toc25010914)

[**3.1.3.** **Cambios en el Explorador de Windows** 6](#_Toc25010915)

[**3.2.** **Archivo Único de Identificación (AUI)** 6](#_Toc25010916)

[**3.2.1.** **Definición del Proyecto** 6](#_Toc25010917)

[**3.2.2.** **Uso del Explorador de Windows** 7](#_Toc25010918)

[**3.3.** **Uso de Estructuras de datos en el proyecto** 7](#_Toc25010919)

[**4.** **Diseño del Proyecto** 9](#_Toc25010920)

[**5.** **Diseño del producto** 9](#_Toc25010921)

[**5.1.** **Definición** 9](#_Toc25010922)

[**5.2.** **Definición de requisitos funcionales y no funcionales** 10](#_Toc25010923)

[**5.2.1.** **Visualizar los archivos del programa en un TreeView.** 10](#_Toc25010924)

[**5.2.2.** **Visualizar el contenido de cada archivo en un ListView** 10](#_Toc25010925)

[**5.2.3.** **Buscar archivos de acuerdo a un criterio** 10](#_Toc25010926)

[**5.2.4.** **Filtrar los archivos de acuerdo a un criterio** 11](#_Toc25010927)

[**5.2.5.** **Ordenar los archivos de acuerdo a un criterio** 11](#_Toc25010928)

[**5.3.** **Cuadro de Riesgos** 11](#_Toc25010929)

[**5.4.** **Diagrama de clases** 12](#_Toc25010930)

[**5.5.** **Interfaz de usuario** 13](#_Toc25010931)

[**5.6.** **Tipos de datos abstracos** 13](#_Toc25010932)

[**5.7.** **Selección de estructuras de datos** 14](#_Toc25010933)

[**5.7.1.** **Archivos** 14](#_Toc25010934)

[**5.7.2.** **Lamba** 14](#_Toc25010935)

[**5.7.3.** **Listas** 14](#_Toc25010936)

[**5.7.4.** **Recursividad** 14](#_Toc25010937)

[**5.7.5.** **Arboles Binarios AVL** 14](#_Toc25010938)

[**5.7.6.** **Tabla Hash** 15](#_Toc25010939)

[**6.** **Diseño del proceso de desarrollo de la solución** 15](#_Toc25010941)

[**6.1.** **Diseño de la arquitectura del software** 15](#_Toc25010942)

[**6.2.** **Implementación de las funcionalidades del software.** 15](#_Toc25010944)

[**6.3.** **Pruebas del software** 16](#_Toc25010945)

[**7.** **Conclusiones** 17](#_Toc25010946)

[**8.** **Bibliografía** 18](#_Toc25010947)

[**9.** **Anexos (ppt, video, otros)** 19](#_Toc25010948)

# **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)**

# El desarrollar el proyecto, nos ha ayudado a aplicar la enseñanza que corresponde al nuestra carrera. Así pues, el beneficio de ello es considerar una serie de situaciones considerando la salud pública, seguridad y bienestar para poner el funcionamiento nuestro proyecto e impactar de manera positiva a la sociedad.

# **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.

## **Uso de Estructuras de datos en el proyecto**

* + 1. **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.

* + 1. **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.

* + 1. **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.

* + 1. **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.

* + 1. **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.

* + 1. **Arboles Binarios AVL**

El árbol AVL es un árbol binario de búsqueda, este árbol binario cumple con que cada nodo este dividido por subárboles, separándolos los menores que la raíz a la izquierda mientras que los mayores a la derecha. Así púes, un árbol AVL debe poseer la propiedad de equilibrio para asegurar su profundidad, y no recorrer todo el código hasta hallar el elemento deseado.

* + 1. **Tabla Hash**

La Tabla Hash es una estructura de datos cuyo propósito se centra en llevar acciones básicas como inserción, eliminación u búsqueda de elementos. De esta manera, la función hash de una brinda una acción de buscar un tiempo de ejecución, y de la misma manera convierte un dato para almacenar, en una llave que representa el índice de la estructura.

# **Diseño del Proyecto**

En el presente trabajo se ha divido en tres hitos, cada hito tiene sus respectivos requisitos a implementar en nuestro proyecto. Es por ello, que hemos decidido crear un diagrama de Gantt **[Anexo 1],** 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.

# **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 árboles 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**

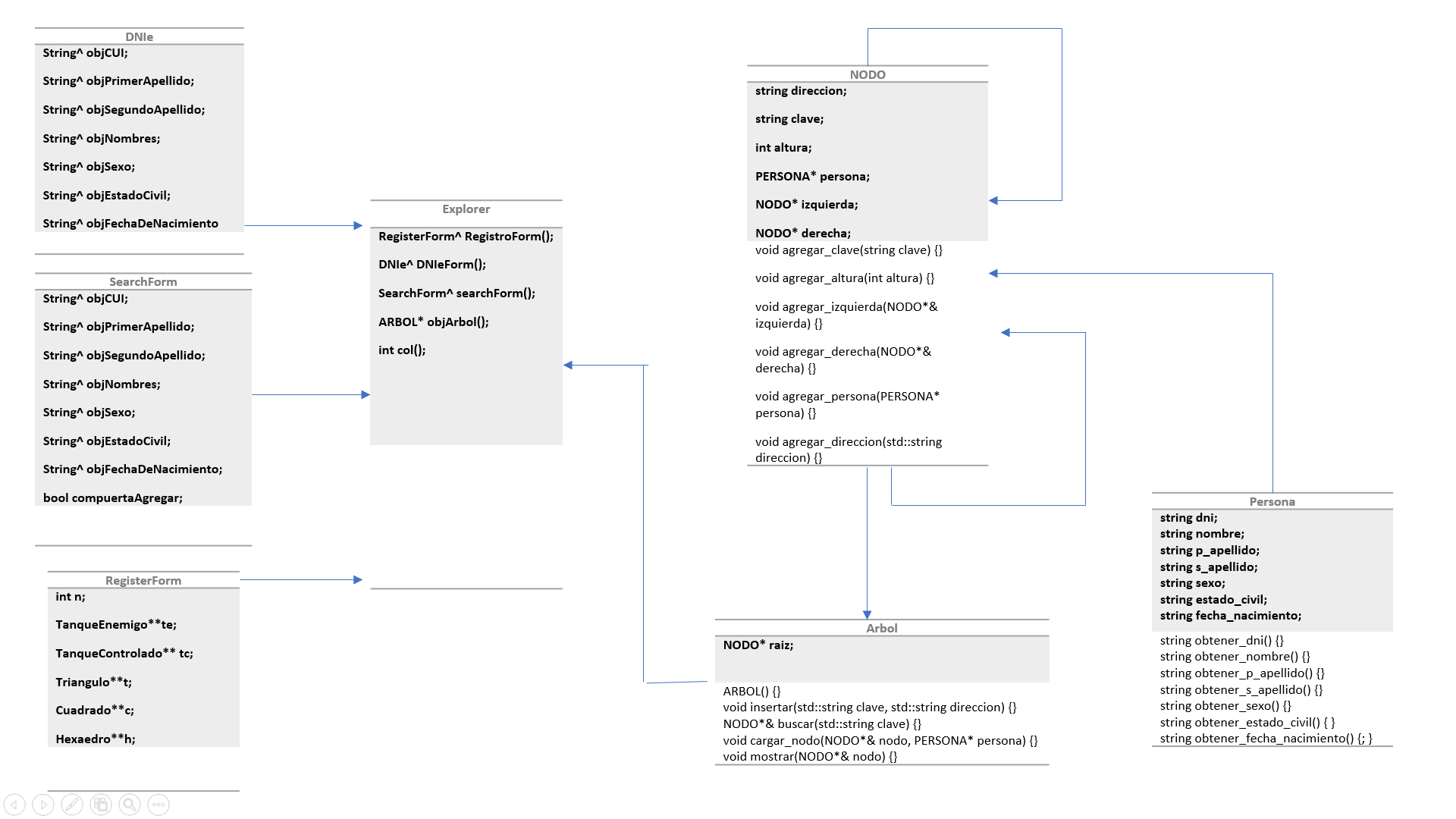
1. **Requisitos Funcionales**

* 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 | Control de contingencias |
| 1 | El software desarrollado puede tener problemas compatibilidad con distintos sistemas operativos. | Alto | La realización de una serie “Prueba y error” con las versiones entre Visual Studio 2015 y 2019 |
| 2 | La computadora donde se está desarrollando el software puede ser afectada por un virus informático. | Alto | El análisis previo de la computadora teniendo en consideración la probabilidad de una nueva instalación anti-virus. |
| 3 | El hardware donde se está almacenando el proceso del programa puede dañarse. | Medio | Realización de una copia de seguridad en un puerto externo. |
| 4 | Un problema de almacenamiento al momento de subir el proyecto al repositorio Github | Medio | Tener conocimientos del uso y manejo del repositorio para poder subir el proyecto. |

## **Diagrama de clases**



## **Interfaz de usuario**

Decidimos dar un uso de interfaz sencillo para que el usuario tenga mayor movilidad y acceso fácil a los dato del propio, es por ello que solo posee funciones básicas. [Anexo 2]

## **Tipos de datos abstracos**

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

* La clase “NODO”
  + Agregar\_Clave(clave)
  + Agregar\_Altura(altura)
  + Agregar\_Izquierda(izquierda)
  + Agregar\_Derecha(derecha)
  + Agregar\_Persona(persona)
  + Agregar\_Direccion(dirección)
* La clase “ARBOL”
  + Insertar (clave, direccion)
  + Buscar(clave)
  + Cargar\_Nodo(nodo, persona)
  + Mostrar(nodo)
* La clase “Persona”
  + Obtener\_dni()
  + Obtener\_Nombre()
  + Obtener\_p\_apellido()
  + Obtener\_s\_apellido()
  + Obtener\_sexo()
  + Obtener\_estado\_civil()
  + Obtener\_fecha\_nacimiento()

## **Selección de estructuras de datos**

### **Archivos**

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**

En nuestro proyecto el uso de lambas fue esencial para separar las funciones operativas como el agregar usuario entre otros.

### **Listas**

Las listas nos ayudaron a poder organizar los distintos datos dentro de los archivos.

### **Recursividad**

En nuestro proyecto, la recursividad fue de uso para que las operaciones sean más dinámicas y no presenten problemas al momento de usarlas.

### **Arboles Binarios AVL**

El uso principal en nuestro proyecto fueron los árboles, ya que nos facilitó el buscar las distintos tipos de personas por medio de su nombre, DNI, código.

### **Tabla Hash**

### La tabla de Hash nos ayudó al momento de poder organizar los distintos tipos de archivos que teníamos, junto a la información que poseen cada uno.

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

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

## Según las indicaciones previstas para el desarrollo de nuestro trabajo, y el impacto de nuestro proyecto hacia el usuario, decidimos dar un aspecto simple para que el manejo no presente ningún inconveniente al administrador. Como se ha dicho con anterioridad, nuestra base de diseño ha sido el explorador de Windows, una herramienta básica pero fundamental como organizador y almacenador de archivos. El “Archivo Único de Identidad (AUI)” posee un TreeView que muestra la información en una estructura jerárquica con nodos contraíbles, como el explorador de Windows. De la misma manera, la vista detalla del usuario de divide en el nombre, fecha de creación de usuario, fecha de modificación de usuario, tipo y el tamaño del archivo de usuario. [Anexo 3]

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

Nuestro proyecto “Archivo Único de Identidad (AUI)”, como ya se ha mencionado con anterioridad, simula el almacenamiento de los datos específicos de una persona como: nombres completos, número de identificación, fecha de nacimiento, lugar de nacimiento, sexo, estado civil, dirección, saldo en tarjetas o monedero virtual. Por ello, las funcionalidades son sencillas para el entendimiento y comodidad del usuario.

* Botón “Agregar Usuario”: Como su nombre indica, es una herramienta para añadir un nuevo usuario, junto a su archivo. Debajo de este botón se encuentra un textbox que sirve para añadir el DNI y verificar si la persona se encuentra en el sistema. De no ser así, al momento de pulsarlo se mostrara una nueva ventana pidiendo los datos de la persona [Anexo 2].
* Botón “Buscar”: Como su nombre indica, sirve para buscar a una persona por medio del nombre, DNI, edad. Así pues, al momento de pulsarlo, aparecerá de manera automática una nueva ventana mostrando lo que anteriormente se ha mencionado. El sistema al encontrarlo mostrara los datos de dicha persona. [Anexo 2].
* Botón “Ver DNI”: Como su nombre indica, muestra el DNI de la(s) persona(s) que se ha buscado o se encuentra registrado en el sistema. Automáticamente aparece los datos de dicha persona, junto a su foto-perfil correspondiente.
* Filtrar: De la misma manera, tenemos la opción de filtrar por medio del Nombre, tamaño, edad, DNI a las personas que ya se encuentran en el sistema.
* Ordenar: El usuario o el operador, puede ordenar de manera ascendente o descendente a las personas que se encuentran en el sistema.

## **Pruebas del software**

Para poder desarrollar el código con precisión, se tuvo que realizar varias pruebas, con el fin de no tener ninguna falla, bug o incompatibilidad de versión.

* + - Versiones del Código: Durante la realización del proyecto, se realizaron una serie de cambios. Por esta razón, se llegaron a realizar tres versiones hasta poder alcanzar el resultado esperado.
    - Versiones del Visual Studio: El desarrollo del proyecto se llevó por medio del Visual Studio 2019. Con el fin de no tener ningún problema de compatibilidad, se probó el código en Visual Studio 2015. Teniendo resultados positivos.

# **Conclusiones**

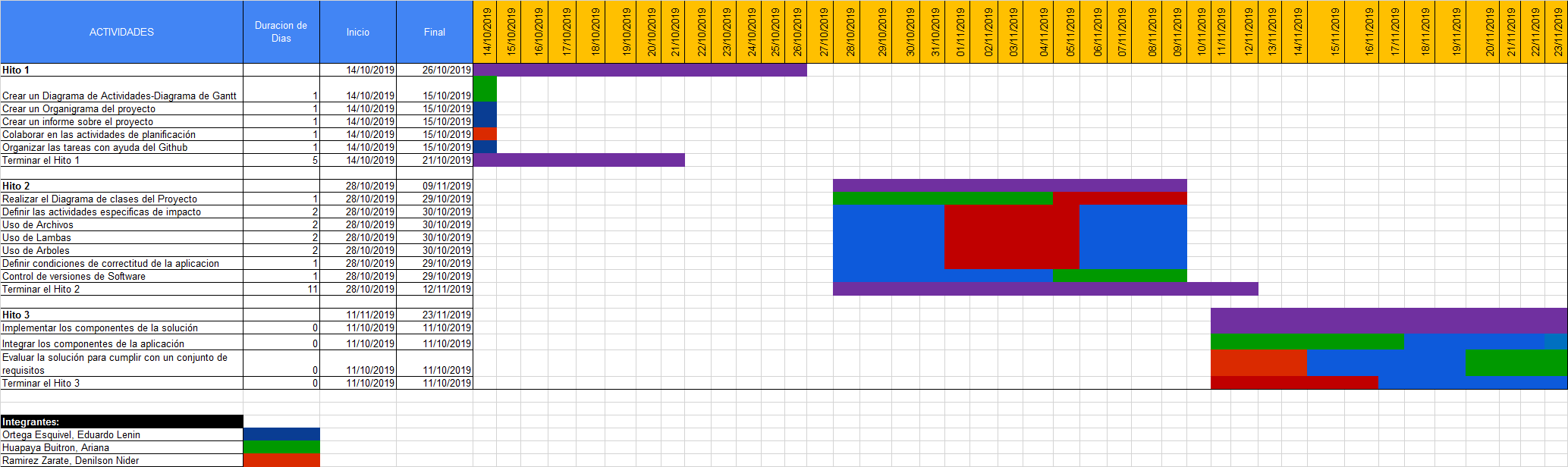
El archivo único de Identidad almacena la información general como: nombres completos, número de identificación, fecha de nacimiento, lugar de nacimiento, sexo, estado civil, dirección, saldo en tarjetas o monedero virtual por medio de carpetas, ya que nos basamos en los principios del Explorador de Windows.

Así pues, el diseño de nuestro proyecto es simple para el manejo cómodo del usuario o el administrador siendo una manera más rápida para modificar o acceder. 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.

# **Bibliografía**

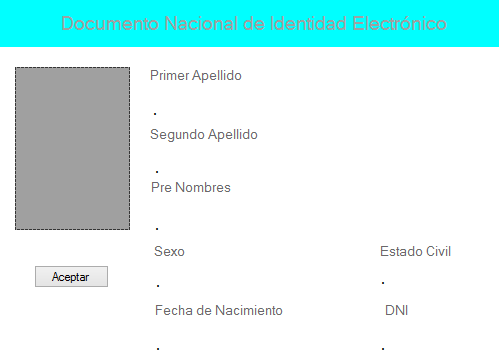
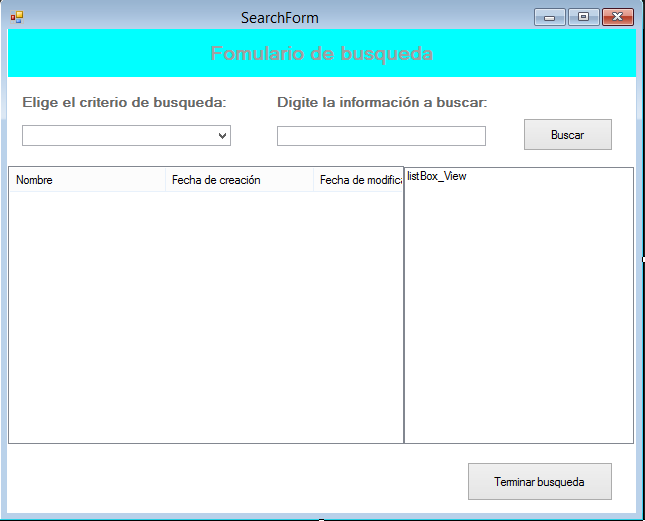
* Barrionuevo Salazar, O., Loechle Chale, M., Palomino Yarasca, I., & Zavaleta Mejía, W. (2019). Recuperado de: <https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/618260/PEREZ_RJ%20%20WORD.pdf?sequence=1&isAllowed=y> [Consultado: 17 de Noviembre de 2019]
* Explorador de archivos. Recuperado de: <https://es.wikipedia.org/wiki/Explorador_de_archivos> [Consultado: 17 de Noviembre de 2019]
* TABLAS HASH. Recuperado de: <http://decsai.ugr.es/~jfv/ed1/tedi/cdrom/docs/tablash.html> [Consultado: 17 de Noviembre de 2019]
* González, A. (2001). *Clase Lista C++ Estándar.* Recuperado de: <http://profesores.elo.utfsm.cl/~agv/elo326/list.pdf> [Consultado: 17 de Noviembre de 2019]

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

**Anexo 1**

**Anexo 2**

****



Anexo 3

