**Primer proyecto programado (Matrícula universitaria)**

**Estudiantes: Erick Gamboa**

**Curso: Programación 3 (ISW-411)**

**Profesor: Freddy Rocha Boza**

**III Cuatrimestre del 2020**

**Valor: 15%**

**Año: 2020**

Introducción:

El desarrollo de las aplicaciones programadas ha venido a innovar el mundo. Cuando se habla de aplicaciones se debe mencionar que existen distintos tipos de estas, por ejemplo, las aplicaciones web o las aplicaciones de escritorio, estas segundas han venido a facilitar en gran medida las tareas de las personas, esto debido a que se puede lograr un mejor manejo de los datos o de igual forma es posible desarrollar aplicaciones que optimicen el tiempo de inscripción de un usuario, también se puede restringir algún tipo de información entre usuarios. Las aplicaciones de escritorio permiten desarrollar una gran variedad de ideas, existen distintos lenguajes para el desarrollo como c#, Python, Java, entre otros y de igual manera distintos ambientes de desarrollo (Integrated Development Environment) como NetBeans, pyCharm, Visual Estudio, entre otros.

En este escrito se plasmarán algunos aspectos importantes basados en la elaboración de una aplicación de escritorio cuya función es posibilitar una matricula universitaria. Se debe tomar en cuenta que la aplicación es programada en el lenguaje de programación c# y ambientada en el IDE de Visual Estudios, como se mencionó anteriormente, en este documento se tomarán en cuenta los principales objetivos de la elaboración de este proyecto, también las lecciones que se pudieron aprender durante su desarrollo, de igual manera algunas conclusiones ligadas a los resultados obtenidos y finalmente una bibliografía que mostrará las fuentes utilizadas para la elaboración de dicho trabajo.

Objetivo general:

* Desarrollar una aplicación de escritorio en C# que permita facilitar el proceso de una matrícula universitaria

Objetivos específicos:

* Interpretar el funcionamiento del lenguaje c#
* Identificar distintos problemas u obstáculos que puede presentar una aplicación durante su desarrollo
* Separar las distintas capas de responsabilidades que posee la aplicación

Lecciones aprendidas:

En este apartado se debe mencionar que se pudo comprender el funcionamiento de la división de capas o de responsabilidades. De igual manera se puede decir que se entendió bastante el manejo de la sintaxis del lenguaje c#. Este proyecto también aportó en gran medida en la elaboración de diseño, esto debido a que con el manejo de las interfaces se pudieron intentar elaboraciones que fueran lindas de ver para el usuario. Otra lección aprendida fue la importancia del uso del GIT, este aporta muchísimo en resguardar los avances del proyecto. Finalmente se debe decir que es importante crear un diagrama de flujo o un seudocódigo antes de elaborar el proyecto, esto con la intención de tener mas claro como se debe elaborar el proyecto

Conclusión:

A modo de conclusión se debe mencionar que c# es un lenguaje de programación bastante amigable con el usuario, esto debido a que posee un diseño de interfaz bastante cómodo y la sintaxis no es compleja, de igual manera Visual Estudio es una herramienta sumamente útil que permite una edición del código bastante importante y unas lecturas de los errores muy atinadas. También se puede concluir que las aplicaciones de escritorio son importantes para poder sintetizar trabajos de oficina como una matricula o cualquier otra tarea sin importar el área. Otra conclusión importante es que la utilización de la metodología orientada a objetos fue bastante atinada para la elaboración del proyecto, esto debido a que permitió un orden bastante visible en el código. Finalmente se concluye que independientemente de cual sea el área de trabajo, por medio de una aplicación (web o de escritorio) se pueden obtener más resultados con menos trabajo y menos tiempo, en la actualidad las personas se deben adaptar a la nueva era tecnológica.

Bibliografía:

JSON sample - Code Samples. (2020). Recuperado de: https://docs.microsoft.com/en-us/samples/microsoft/windows-universal-samples/json/

list Class. (2019). Recuperado de: https://docs.microsoft.com/en-us/cpp/standard-library/list-class?view=vs-2019

List<T> Class (System.Collections.Generic). (2020). Recuperado de: https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=netcore-3.1