|  |
| --- |
| FileEXP  Trabajo Final del Curso |
| Team Members  Rodrigo Ticona Esquivel U201819011  Hernán Razo Ballón U2018122706  David Huamani Injante U20181B001 |



**Universidad Peruana de Ciencias Aplicadas**

Ingeniería de Sistemas y Computación | Ingeniería de Ciencias de la Computación

CC52 | Ingeniería de Software

Ciclo 2019-02

CONTENT

[Introducción 3](#_Toc8978784)

[Indice 3](#_Toc8978784)

[Objetivo del Estudiante 3](#_Toc8978784)

[Capitulo 1: Marco conceptual 3](#_Toc8978784)

[Capitulo 2: Diseño del proyecto 3](#_Toc8978784)

[Diseño de un plan de proyecto de desarrollo de soluciones en ingeniería con las restricciones establecidas considerando asignación de recursos, milestones, issues 3](#_Toc8978784)

[Capitulo 3: Diseño del producto 3](#_Toc8978784)

[Definición de las necesidades específicas considerando el impacto en salud pública, seguridad, bienestar y factores globales, culturales, sociales, ambientales y económicos. 3](#_Toc8978784)

[Definición de requisitos funcionales y no funcionales. 3](#_Toc8978784)

[Elaboracion de Diagrama de clases. 3](#_Toc8978784)

[Diseño de interfaz de usuario. 3](#_Toc8978784)

[Diseño de tipos de datos abstractos. 3](#_Toc8978784)

[Seleccion de estructuras de datos. 3](#_Toc8978784)

[Capitulo 4:Diseño del proceso de desarrollo de software 3](#_Toc8978784)

[Diseño de la arquitectura del software. 3](#_Toc8978784)

[Implementacion de las funcionalidades del software. 3](#_Toc8978784)

[Pruebas de Software. 3](#_Toc8978784)

[Conclusiones 3](#_Toc8978784)

[Bibliografia 3](#_Toc8978784)

[Anexos 3](#_Toc8978784)

Ciclo 2018-02 | Sección XXXX

## Introducción

El trabajo final tiene por nombre FILEEXP, consiste en el desarrollo de un software explorador de archivos para el sistema operativo Windows o Linux, el cual debe implementar las características como organizar, indexar, buscar y filtrar por condiciones.

## Objetivo del Estudiante

Nuestro objetivo es aplicar los conocimientos adquiridos en análisis y diseño de tipos de datos abstractos, técnicas sistemáticas con base en matemáticas para la construcción de una aplicación basada en el procesamiento de grandes cantidades de datos considerando el tiempo de respuesta creativamente. 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.

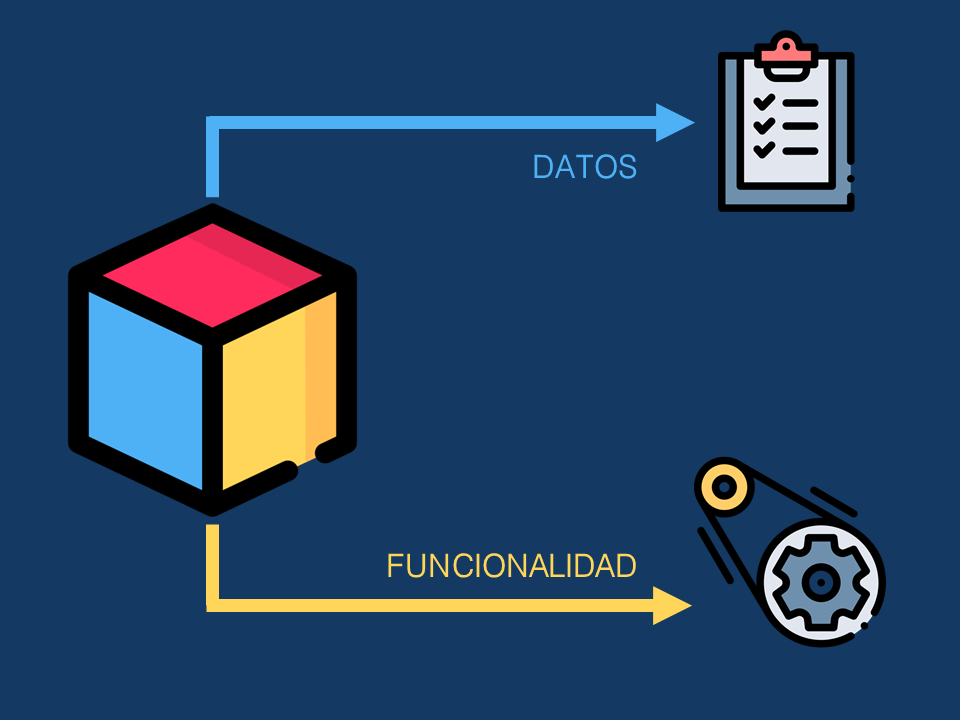
CHAPTER

## Marco Conceptual

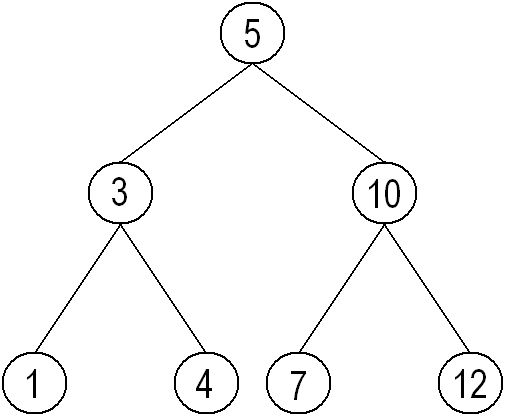
1

### **Programación Orientada a Objetos (POO)**

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación. Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con este tipo de programación. Pensar en términos de objetos es muy parecido a cómo lo haríamos en la vida real. Por ejemplo vamos a pensar en un coche para tratar de modelizarlo en un esquema de POO. Diríamos que el coche es el elemento principal que tiene una serie de características, como podrían ser el color, el modelo o la marca. Además tiene una serie de funcionalidades asociadas, como pueden ser ponerse en marcha, parar o aparcar. Pues en un esquema POO el coche sería el objeto, las propiedades serían las características como el color o el modelo y los métodos serían las funcionalidades asociadas como ponerse en marcha o parar.



### **Arboles binarios de búsqueda balanceada**

La búsqueda más eficiente se efectúa en un árbol binario balanceado. Desafortunadamente, la función Inserta no asegura que el árbol permanezca balanceado, el grado de balance depende del orden en que son insertados los nodos en el árbol. La altura de un árbol binario es el nivel máximo de sus hojas. La altura del árbol nulo se define como -1. Un árbol binario balanceado es un árbol binario en el cual las alturas de los dos subárboles de todo nodo difieren a lo sumo en 1. El balance de un nodo en un árbol binario se define como la altura de su subárbol izquierdo menos la altura de su subárbol derecho. Cada nodo en un árbol binario balanceado tiene balance igual a 1, -1 o 0, dependiendo de si la altura de su subárbol izquierdo es mayor que, menor que o igual a la altura de su subárbol derecho.

### **Archivos**

El archivo de cabecera fstream.h define las clases ifstream, ostream y fstream para operaciones de lectura, escritura y lectura/escritura en archivos respectivamente. Para trabajar con archivos debemos crear objetos de éstas clases de acuerdo a las operaciones que deseamos efectuar. Empezamos con las operaciones de escritura, para lo cual básicamente declaramos un objeto de la clase ofstream, después utilizamos la función miembro open para abrir el archivo, escribimos en el archivo los datos que sean necesarios utilizando el operador de inserción y por último cerramos el archivo por medio de la función miembro close

### 

CHAPTER

## Diseño del proyecto

2

### **Diseño de un plan de proyecto de desarrollo de soluciones en ingeniería con las restricciones establecidas considerando asignación de recursos, miles tones, issues.**

El trabajo fue hecho atreves de GitHub para lograr una mejor comunicación con los integrantes del grupo. El trabajo se dividió en 2 partes la codificación del programa y el informe del trabajo.

La parte de codificación fue dividida en 3 hitos:

* El primer Hito consta del indexado y serialización para la buscar cierta ruta elegida.
* El segundo Hito consta de la búsqueda de archivos junto al ordenamiento final.
* El tercer Hito consta del Filtrado de datos por columnas el cual fue avanzado por todos los integrantes.

La parte del informe fue dividida en 2 hitos:

* El primer Hito consta del diseño de la aplicación, el diagrama de clases, la arquitectura del software y las pruebas del software.
* El segundo Hito consta de terminar de completar todas las partes faltantes del documento

CHAPTER

## diseño del producto

CHAPTER

3

### **Definición de las necesidades específicas considerando el impacto en salud pública, seguridad, bienestar y factores globales, culturales, sociales, ambientales y económicos**

**Definición de requisitos funcionales y no funcionales**

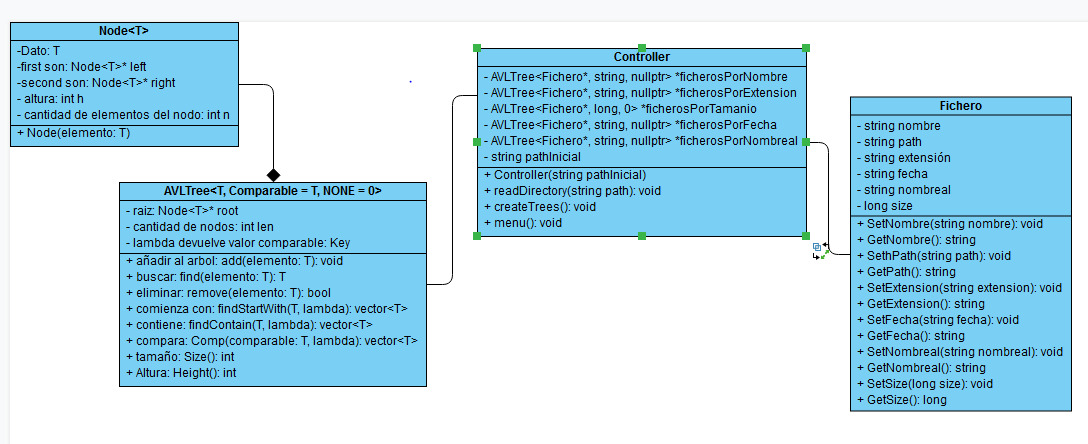
#### Requerimientos funcionales

### Los requerimientos funcionales hacen referencia a la descripción de las actividades y servicios que un sistema debe proveer. Normalmente este tipo de requerimientos están vinculados con las entradas, las salidas de los procesos y los datos a almacenar en el sistema.

#### Requerimientos no funcionales

### Por otra parte los requerimientos no funcionales describen otras prestaciones, características y limitaciones que debe tener el sistema para alcanzar el éxito. Los requerimientos no funcionales engloban características como rendimiento, facilidad de uso, presupuestos, tiempo de entrega, documentación, seguridad y auditorías internas.

**Elaboración de diagrama de clases**



**Diseño de interfaz de usuario**

**Diseño de tipos de datos abstractos**

.

**Selección de estructuras de datos**

CHAPTER

## diseño del proceso del desarrollo de

4

## Software

**Diseño de la arquitectura del Software**

**Implementación de las funcionalidades del software**

**Prueba del Software**

## Conclusiones

## 

## Bibliografía

<http://www.utm.mx/~jahdezp/archivos%20estructuras/balanceo%20de%20arboles.pdf>

<https://desarrolloweb.com/articulos/499.php>

<http://www.programacionenc.net/index.php?option=com_content&view=article&id=69:manejo-de-archivos-en-c&catid=37:programacion-cc&Itemid=55>

<https://ingenieriadesoftwareutmachala.wordpress.com/2017/01/20/requerimientos-funcionales-y-no-funcionales/>

## anexos