

**ESTRUCTURA DE DATOS**

**GRUPO N°4**

**INTEGRANTES:**

**Keneth Andrade**

**Rafael Buse**

**Henry Tiamba**

**Brayan Patiño**

**Ariel Hidrobo**

**Luis Calvopiña**

**TÍTULO:**

**DISEÑO DE ALGORITMOS UTILIZANDO TÉCNICAS DE REPRESENTACIÓN**

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - ESPE

**Contenido**

[Tema 3](#_Toc75469211)

[Objetivo General 3](#_Toc75469212)

[Objetivos Específicos 3](#_Toc75469213)

[Marco Teórico 3](#_Toc75469214)

[Lenguaje C++ 3](#_Toc75469215)

[Estructura de Datos 3](#_Toc75469216)

[Tabla de amortización 3](#_Toc75469217)

[Desarrollo 4](#_Toc75469218)

[Conclusiones 7](#_Toc75469219)

[Recomendaciones 8](#_Toc75469220)

[Referencias 8](#_Toc75469221)

# 

# Tema

Tabla de amortización.

# Objetivo General

Implementar el Diseño de Algoritmos utilizando técnicas de representación en la resolución de problemas por computador.

# Objetivos Específicos

* Analizar el problema planteado, sobre la tabla de amortización, para poder encontrar una buena solución para implementarlo en un software.
* Demostrar todos los contenidos aprendidos a lo largo del primer parcial mediante la realización de un programa que simule una tabla de amortización.
* Distinguir apropiadamente la aplicación correcta, en la programación, de la diferente teoría aprendida en la materia de Estructura de Datos.

# Marco Teórico

## Lenguaje C++

C++ es un lenguaje de programación que proviene de la extensión de lenguaje C para que pudiese manipular objetos. A pesar de ser un lenguaje con muchos años, su gran potencia lo convierte en uno de los lenguajes de programación más demandados en la actualidad.

Fue diseñado a mediados de los años 80 por el danés Bjarne Stroustrup, con el objetivo de que sirva para manipular objetos, por lo cual se la conoce como un lenguaje de programación multiparadigma. Entre sus ventajas tenemos: alto rendimiento, lenguaje actualizado, multiplataforma y extendido, es decir, que cualquier programa tienen alguna parte escrita en este lenguaje.

## Estructura de Datos

Es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente. También, son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet.

Por lo general, las estructuras de datos son clave para diseñar algoritmos eficientes, convirtiéndose como el factor clave de organización en el diseño. Estos se clasifican en estructura de datos estáticos, dinámicos, lineal y no lineal, dependiendo de la técnica que se use para su almacenamiento o para su recorrido.

## Tabla de amortización

Amortización es el método por el cual se va liquidando una deuda en pagos parciales. El importe de cada pago sirve para solventar los intereses de la deuda, y el sobrante se abona al capital que se debe en ese periodo.

En la amortización se demuestra que:

* El capital va disminuyendo conforme se van dando los pagos, hasta su liquidación total.
* Al ir reduciéndose el capital, los intereses también van descendiendo.
* La amortización del capital va aumentando conforme pasan los periodos, al ir disminuyendo –en la misma proporción– los intereses.
* Si se quieren conocer las amortizaciones de los diferentes periodos, basta multiplicar la primera amortización por la razón

Para poder calcular la el monto a pagar usaremos la siguiente formula donde:

A=monto solicitado

n=años de préstamo

i=interés del préstamo

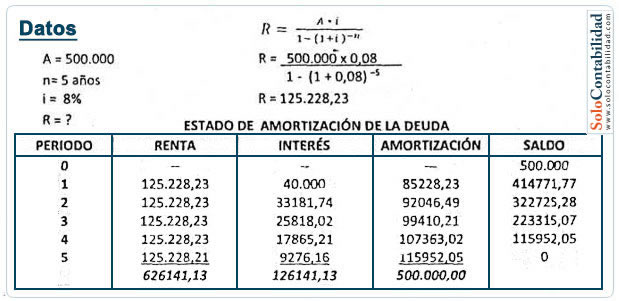


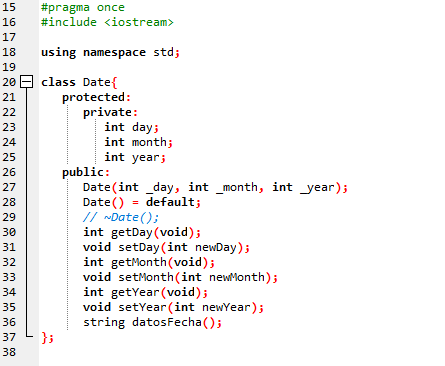
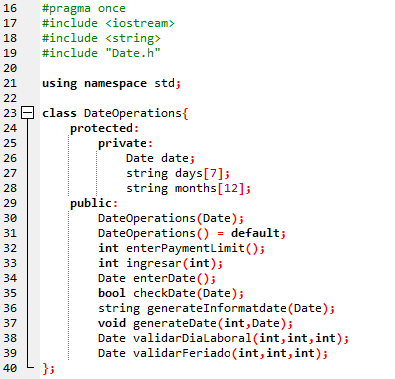
Figura 1. tabla de amortización

# Desarrollo

Para el desarrollo del programa, primer se identificó las clases que se iban a usar, de tal manera que cada una esté ordenada y contenga los atributos y métodos correctamente ubicados, a continuación, se irá describiendo cada clase implementada.

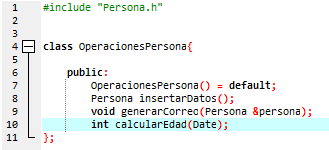
Tenemos la clase “Date”, la cual contiene los datos de las fechas. También una etiquetada como “DateOperations”, la cual está encargada de ingresar, validad, generar, entre otras funciones, las fechas que debe generar el programa a partir de los datos ingresados por el usuario.

A continuación, algunas capturas de las clases implementadas, cabe resaltar que cada método se desarrollo en el .cpp de su propia clase.

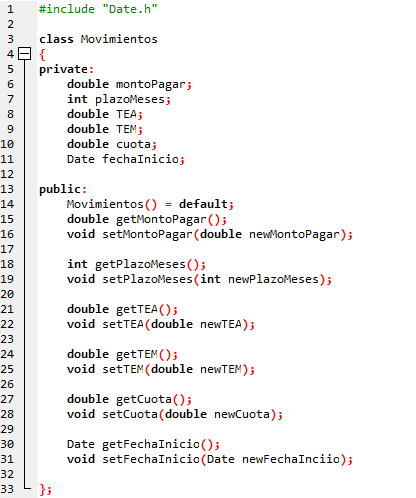
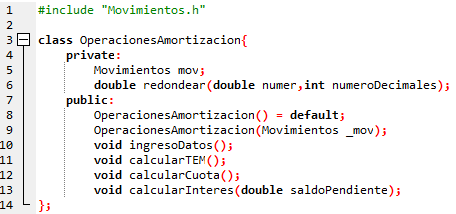
 

De la misma forma, tenemos la clase “persona”, la cual almacena los datos que se requiere del usuario, tales como: nombre, apellido, fecha de nacimiento, cedula, dirección, teléfono, correo electrónico y la edad. También, se ha creado la clase “OperacionesPersona”, la cual contiene solo métodos como el de generar el correo electrónico, calcular la edad e insertar los datos que se piden en la anterior clase nombrada. Ambas quedarían de la siguiente forma:

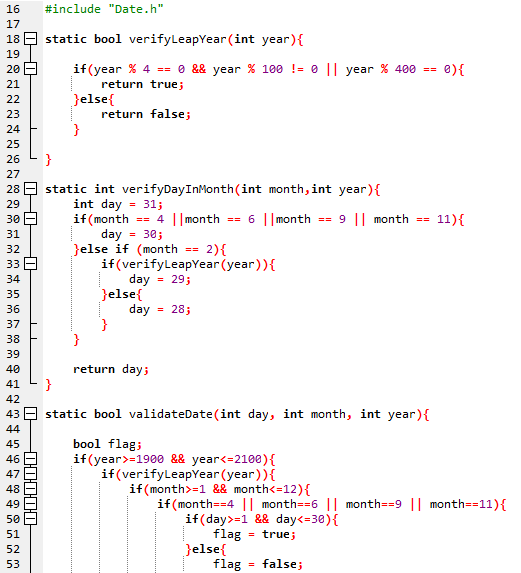
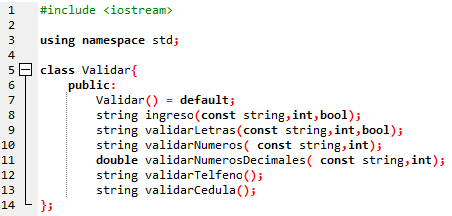
****

****

No obstante, tenemos una clase “Movimientos”, que es la que guarda los valores a pagar, el plazo de meses, la fecha, cuota, entre otras más. Igualmente, se creó una clase “OperacionesAmortizacion” con la función de que contenga los diferentes métodos para calcular los valores de interés, cuota y redondear valores decimales hasta la cifra que se requiera.



Por otro lado, se implementó clases como “Validation” y “Validar”, siendo la primera la encargada de aprobar las fechas ingresadas. La segunda se la usó para la validación de los números, cédula y el teléfono, seguidamente se muestra las clases.

****

Finalmente, también se guardó algunas clases en una carpeta llamada librería, las cuales están clases como “ListaSimple”, “Menu”, “Nodo”, entre otras, y que se usan en los demás archivos .h y .cpp, que sirven como apoyo para poder desarrollar el programa de manera eficiente.

# Conclusiones

* Se implementó satisfactoriamente el Diseño de Algoritmos mediante la utilización de técnicas de representación en la resolución del problema planteado por computador.
* Se logró analizar e identificar correctamente el problema sobre la tabla de amortización, de manera que se planteó un programa que simule su funcionalidad, usando la teoría aprendida en clase.
* Se aplicó cabalmente cada contenido aprendido en Estructura de Datos durante el transcurso del primer parcial, de forma que se pudo dar funcionalidad al problema planteada, por lo cual este refleja los conocimientos adquiridos y comprendidos desde la perspectiva de cada uno de nosotros.
* Se pudo distinguir adecuadamente en qué parte del código fuente aplicar los diferentes conceptos como listas simples, tipo de datos dinámicos o estáticos, entre otros, tratando de optimizar de la mejor manera el tiempo y ejecución del programa.

# Recomendaciones

* Para futuras implementaciones de algoritmo será importante la implementación de más funciones recursivas ya que me ahorran líneas de codificación.
* Es importante realizar esquemas grafico para entender el problema propuesto y así dar solución mediante un algoritmo.
* Es importante el uso de plantillas ya que me permite modificar de una manera muy rápida, simplemente cambiando el tipo de variables con la que desee trabajar las cuales pueden ser: int, char, string, float, double.
* Es importante tener claro los conceptos y criterios de los temas propuestos para la realización de algoritmos claros y eficientes.
* Se podría intentar calendar reuniones de cada grupo con el docente para poder hacer preguntas acerca de dudas acerca del problema que se está resolviendo, de tal forma que el día que se entregue el proyecto, esté bajo todos los criterios establecidos.

# Referencias

Robledano, Á. (2019). *Qué es C++: Características y aplicaciones*. OpenWebinars.net. https://openwebinars.net/blog/que-es-cpp/.

# Velázquez, J. L. (octubre de 2010). *Lenguaje de Programaci´on: C++*. https://www.cimat.mx/~pepe/cursos/lenguaje\_2010/slides/slide\_33.pdf

Martínez, A., Pompa, R., Mata, J., & Viveros, P. (2010). *Matemáticas Financieras* [Ebook] (pp. 195-204). http://fcasua.contad.unam.mx/apuntes/interiores/docs/20181/contaduria/1/LC\_1156\_06047\_A\_Matematicas\_financieras\_Plan2016.pdf.