***Abstract***— ***Una pregunta muy frecuente entre los equipos de programadores es: ¿cuánto se debe cobrar por un sistema de información?, por tal motivo se ha desarrollado una metodología para crear un sistema que estime los costos para la realización de un Sistema de Software .***

***Ana Paulina Salazar B., Jorge Paredes B., Ricardo de la Vega B., José María Leboreiro V., Carlos Jesus Almeraz A., Dante Fernando Bazaldua H., Fernando González,-Valadez B., Leonardo Alberto López R., Jesús Hector Fragoso O., Evelyn Hernández G., Miguel Andrés Izaguirre V.***

S

# Taller de Desarrollo de Aplicaciones

# PROYECTO FINAL

# Descripción del Sistema:

e requiere un sistema, que por medio de los "Function Points Analysis", calcule el costo aproximado de un proyecto de programación.

El sistema que se quiere desarrollar, representa un sistema de captura de información que los usuarios ingresan, de tal manera que estos datos puedan generar un costo final por medio de cálculos. De esta manera, dependerá de la capacidad del sistema para adaptarse a las necesidades requeridas por los usuarios.

Desde la perspectiva del usuario creemos que un diseño atractivo, simple y eficaz es la manera correcta de proponer un programa que permita saber cuánto le cuesta a un programador desarrollar un proyecto de programación en el lenguaje que el requiera.

# Análisis del Sistema:

De acuerdo a la investigación realizada, se pretende diseñar un programa que maneje distintos datos, los cuales se van a utilizar para obtener una fórmula que calcule la cantidad de dinero que le cuesta a un programador hacer un proyecto. Con la investigación antes mencionada, se diseñó en primera instancia un sistema que empíricamente calculaba los costos y cotizaba el precio del sistema de software (denominado como versión 1.0). El programa interactúa con el usuario por medio de preguntas que se le hacen sobre el proyecto que quiere desarrollar. De acuerdo a los datos obtenidos y por medio de fórmulas, se imprimirá el costo total.

El resultado final se obtendrá por medio de distintas fórmulas, las cuales están divididas en distintas funciones. Cada valor que se obtenga con las preguntas realizadas, se va a utilizar para una fórmula en específico. Al final, se va a obtener un resultado, que va a estar dado por los distintos resultados que se generen en cada función.

# Entradas:

* Tipo de Lenguaje de Programación.
* Tipo de Naturaleza del Proyecto.
* Respuesta a catorce preguntas.
* EI: Entradas.
* EO: Salidas.
* EQ: Peticiones.
* ILF: Archivos Internos.
* EIF: Archivos Externos.

# Salidas:

* Mensajes de bienvenida.
* Mensajes de ayuda.
* Puntos de función.
* Puntos de función Ajustados.
* Líneas de código estimadas.
* Lenguaje de programación utilizado.
* Tiempo en meses.
* Esfuerzo.
* Número de personas.
* Costo total.

# Procedimientos/Procesos:

* Realizar catorce preguntas y obtener la suma.
* Calcular el valor del UFP (Unadjusted function points).
* Calcular el valor del CAF (Value adjustment ecuation).
* Con el UFP y el CAF, calcular el FP (Function Points).
* Calcular las líneas de código aproximadas.
* Calcular el esfuerzo aproximado del programador.
* Calcular la cantidad de tiempo (en meses) que tarda en hacer el proyecto.
* Calcular el número de personas que se recomiendan utilizar en el proyecto.
* Calcular el costo total del proyecto.

# Alcances:

* El programa realiza los cálculos por medio de los puntos de función.

# Limitaciones:

* El programa está limitado a cuatro lenguajes de programación (Java, Java script, C, C# o C Sharp )
* Por proyecto está limitado únicamente a un lenguaje.
* Únicamente se consideraron los valores medianos.

# Diseño del Sistema:

# ***Análisis De Requerimientos***

## Descripción de Funcionalidad:

Por medio de un menú, el usuario podrá seleccionar una opción que le permita ingresar datos, los cuales el programa utilizará para calcular el costo. El menú tiene las siguientes opciones:

1. Calcular puntos de función.

2. Calcular complejidad.

3. Seleccionar lenguaje de programación.

4. Seleccionar tipo de proyecto.

5. Calcular el costo total del proyecto.

El programa está diseñado de forma que el usuario tenga la facilidad de utilizarlo cuantas veces quiera. Si es necesario, se puede desplegar un mensaje que explique cómo es que se tiene que utilizar el programa para que funcione.

## Características del Sistema:

Se utilizan algunas herramientas importantes para mejorar el rendimiento y la presentación del sistema de información tales como un mensaje de error al momento de utilizar le programa. De la misma manera el programa fue desarrollado con la intención de evitar cualquier tipo de error al momento de correr el programa o al momento de estarlo utilizando.

También existe la opción de imprimir un instructivo de la forma en la que se tiene que correr el programa. Esto con el fin de evitar cualquier tipo de percance que pueda tener el usuario y evitar que no truene el programa en el momento que lo usa.

## Requerimientos Funcionales:

Se necesitan llenar los campos de información correctamente para enviar los datos correctos a las funciones. La funcionalidad del sistema se encuentra mayoritariamente en manos del usuario, ya que el usuario decide que hacer por medio del menú y por medio de las cosas que necesita. Se debe utilizar la conexión correcta entre funciones, para poder pasar los valores correctos y llegar al resultado esperado. Es por esto, que las funciones están diseñadas para que pasen los parámetros correctos y se puedan utilizar en las fórmulas en las que son requeridas.

## Estructura del Sistema:

La creación de distintas funciones propone una manera correcta de ordenar los datos para mejorar el uso del sistema, por lo que el sistema también cuenta con módulos que se encargan de separar la parte de los cálculos, con la parte del menú. Los usuarios tienen distintas opciones, las cuales se relacionan entre sí para poder obtener todos los datos posibles y hacer lo más preciso el resultado final.

# ***Funciones, Procedimientos y Módulos Individuales***

## 

## **Módulo:** funciones.c

Funciones a utilizar:

1. Calcular UFP (Unadjusted function points):

Esto se hace en una función que recibe como parámetros el número de entradas, salidas, los archivos externos e internos. De esta manera, la función se encarga únicamente de regresar un valor para UFP. Para esto se utiliza la siguiente fórmula:

(EI\*PESO\_EI)+(EO\*PESO\_EO)+(EQ\*PESO\_EQ)+(ILF\*PESO\_ILF)+(EIF\*PESO\_EIF).

Donde todas las demás variables son constantes predefinidas, y se refieren al peso de de los archivos internos y externos, así como de las entradas y salidas.

1. Calcular FP (Function Points):

Antes de obtener el valor total de la variable, se tiene que conocer con anticipación el valor del CAF (Value adjustement equation). Primero, la función recibe como parámetros el valor calculado para UFP y un entero que se le considera el total. Posteriormente, se calcula el CAF aplicando la siguiente fórmula:

65+(.01\*total)

Finalmente, se devuelve el producto del UFP multiplicado por el valor recién calculado del CAF; esto representa entonces a los Function Points.

1. Calcular LDC (líneas de código):

Para calcular las líneas de código, existe una función que recibe como parámetros al FP y un entero llamado ‘lang’, el cual solo puede ser un número entre el 1 y el 4, para representar en qué plataforma se quiere realizar el proyecto. Dependiendo del tipo de lenguaje que sea, se regresará como valor el producto del FP por la constante predefinida para el costo promedio de cada tipo de lenguaje.

1. Calcular Energía (esfuerzo aproximado del programador):

Para calcular la energía gastada aproximadamente por el usuario, se decidió seguir el siguiente procedimiento. Al número de líneas de código se le divide entre 1000. Una vez hecho esto, se toma la decisión de qué tipo de esfuerzo es (orgánico, medio o inorgánico). Luego se aplica la siguiente fórmula:

a\*(pow(ldc,b))

Dependiendo del tipo que sea, las variables ‘a’ y ‘b’ toman valores diferentes. Cabe mencionar que la función recibe el valor de LDC y el entero para el tipo de lenguaje, mientras que los valores de a y b son variables locales dentro de la misma función.

1. Calcular Meses (tiempo en meses para hacer el proyecto):

Para obtener el estimado de tiempo, la función debe recibir como parámetros dos valores; el tipo de lenguaje, y el resultado obtenido previamente para la energía consumida. De igual forma que la función para la energía, dependiendo del tipo de lenguaje en que se vaya a desarrollar el proyecto, se tienen dos variables locales (‘c’ y ‘d’) que con los parámetros anteriores, se obtiene el estimado mediante:

c\*(pow(energia,d))

Así, el resultado de la potencia se regresa a la variable para el estimado del tiempo,

1. Calcular Personas (número de personas en el proyecto):

Para poder obtener el número de personas recomendadas para trabajar en el proyecto, únicamente se divide la energía estimada para trabajar en el proyecto, entre el tiempo estimado en meses para hacer el mismo.

1. Calcular Total (costo total por el proyecto):

Finalmente, nos vamos a la función destinada para cotizar el costo total; es necesario recibir como parámetros a la cantidad de personas para el proyecto, el tiempo estimado en meses (ambas variables calculadas con anterioridad) y el valor número de lang que representa el tipo de lenguaje a utilizar. Una vez seleccionada la plataforma, se sigue la fórmula:

(personas)\*(meses)\*LANG

Donde LANG va a ser la constante predefinida para el promedio de cada lenguaje de programación. Al final se le despliega en pantalla el valor del costo total.