

Universidad Nacional Autónoma de México



Facultad de Ingeniería



Nombre del alumno:

Carranza Ochoa José David

Profesora:

M.I. Marco Antonio Martínez Quintana

Materia:

Fundamentos de Programación

Trabajo:

Proyecto final

Número de cuenta:

31804885-7

Generación:

2021-1

Correo electrónico:

deibydcarraza@gmail.com

Fecha: 29/01/21

El presente proyecto expone un software de código abierto desarrollado desde el lenguaje de programación C, en donde recopila los conocimientos adquiridos durante el semestre 2021-1 en la materia de Fundamentos de Programación impartida por la Facultad de Ingeniería, dicha materia representó un método de enseñanza a distancia

La realización de este proyecto fue metódica y sistemática, ya que se asignaron tareas programadas en tiempo y forma con el fin de mantener un control hacía las actividades a desempeñar; generando una concepción profesional como emprendedora encaminada a la conclusión del curso ampliando la visión de los estudiantes más allá de la zona habitual de confort.

Para el planteamiento de los posibles proyectos, al inicio del curso se destinaron los materiales que presentan las bases de alguna problemática actual, empezando con las interrogantes ¿Qué problemática existe?, ¿Quiénes han intentado resolverla?, ¿Cómo puedo ayudar a resolverla?, se pueden generar cientos de problemas los cuales necesitan alguna solución; en mi caso, identifiqué un problema enorme que genera la mayor recaudación económica de manera ilícita y que hasta el momento no está regulado, el robo de dinero electrónico.

Antes de ingresar a temas más avanzados como lo es la delincuencia informática, tuve que comprender el funcionamiento de los sectores actualmente atacados, por ejemplo, los cajeros automáticos exponen una interacción directa entre el individuo y los recursos financieros, presentando puntos de vulnerabilidad que a veces no son detectados; es aquí a donde dirigí mi proyecto, en el cual desarrollé el prototipo de un cajero automático con funciones exclusivas en donde incluí un estado de seguridad en la cuenta ya que es un campo que espero profundizar con mayor detenimiento, así mismo en semestres posteriores desarrollaré de estilo más óptimo el cajero que esta vez presento añadiendo funciones y mejoras en su operación.

El proceso para comprender el funcionamiento de un cajero automático me resultó interesante, desde las opciones que se presentan y como llevar a cabo cada operación, es por ello que a partir de la idealización del concepto se fue convirtiendo cada operación en código accesible, resultado tras evaluar el funcionamiento del algoritmo, la esquematización por diagrama de flujo y finalmente el pseudocódigo. Todos estos están expresados de forma clara a lo largo del documento,

Los avances y habilidades que se desarrollaron al momento de realizar el proyecto fueron complementadas por documentación digital, lo que explicó la comparación entre las habilidades que contaba a un inicio de semestre y las adquiridas gracias a este curso. Presento el siguiente proyecto como probatorio de lo antes mencionado en donde por secciones se detalla cada proceso efectuado, así como tablas de comparaciones y recursos para compartir.

Introducción

Los bancos son organismos públicos los cuales realizan operaciones financieras tales como depósitos o préstamos a personas físicas, así mismo presentan una elevada relevancia en la economía mundial ya que son ellos quienes manejan la moneda tanto nacional como extranjera; su aparición se dice que es tan antigua como la misma aparición del hombre, ya que el humano siempre ha tenido la necesidad de alguna entidad que regula tales transacciones; no es hasta antes del siglo IV a.C. cuando en Grecia surgen de manera pública y reconocida por la sociedad.

Más adelante en la historia, resurgen hasta el siglo XIII en la edad media donde toman impulso por la actividad burguesa presente de la época, consolidando las bases de lo que hoy conocemos como banco; a causa de la circulación de piezas metálicas (monedas) así como la conversión de monedas extranjeras marcaron en el comercio internacional una etapa financiera muy importante que perdurará hasta nuestros tiempos.

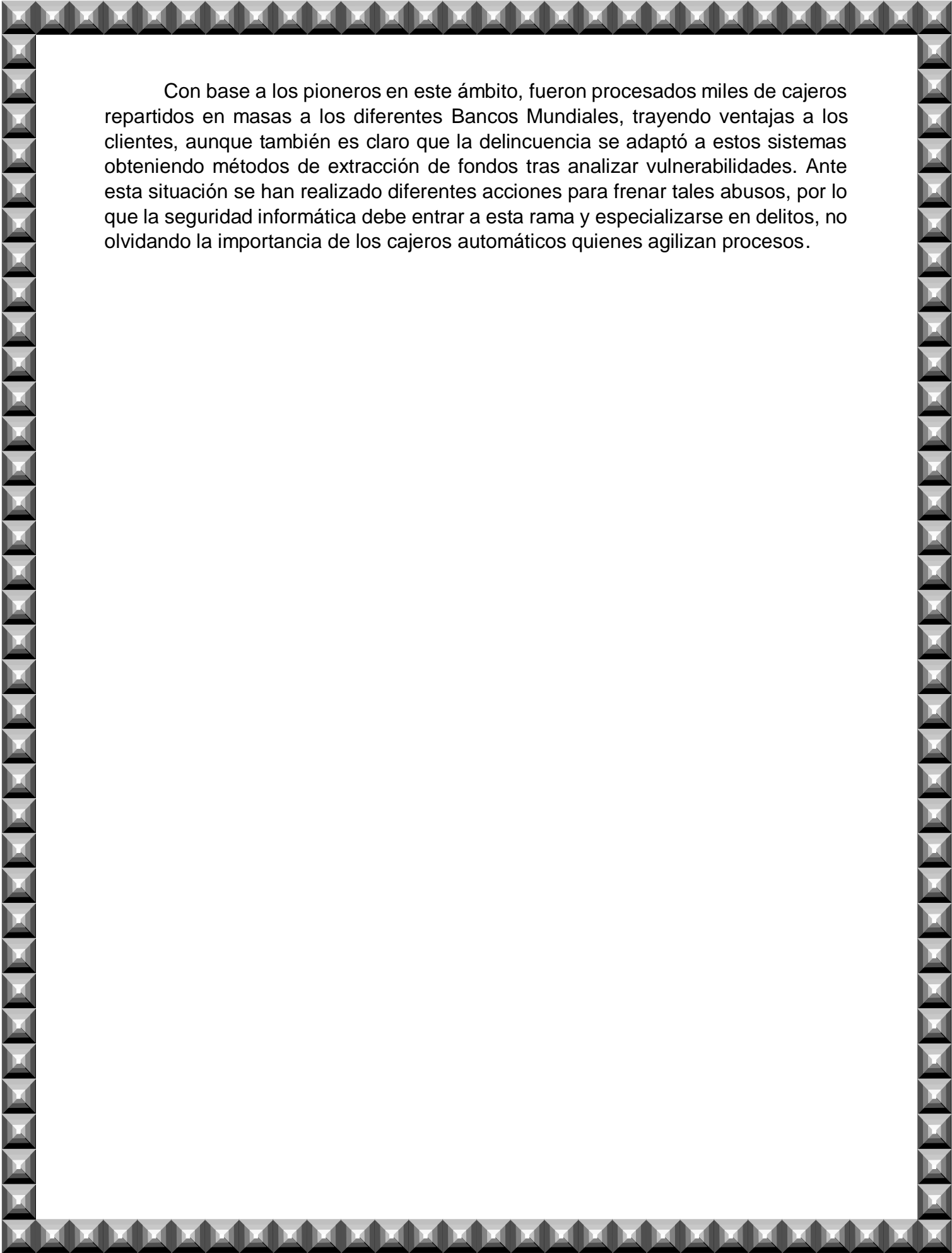
Tras la aparición de las casas bancarias tanto privadas como públicas, se tiene un control más eficiente de la moneda por lo que se le añaden atributos como el financiamiento de negocios, préstamos con garantía, asesoramiento financiero, depósitos, transacciones y también cajas de seguridad.

Como consecuente de la Primera Guerra Mundial en 1914, los bancos toman el impulso necesario para consolidarse completamente, ahora estos estarán regulados y se interconectarán provocando los sistemas bancarios donde se incorporan varias entidades con el mismo propósito.

Con el crecimiento poblacional y la alta demanda del servicio, los bancos se apoyan del desarrollo de nuevas tecnologías, apareciendo nuevos métodos de logística y seguridad para el cliente, creando las tarjetas de crédito (cheques los cuales podían ser recargados) a inicios del siglo XX, generando una revolución económica seguida de mejoras y optimizaciones.

Sin embargo, no fue hasta 1939 cuando Luther George Simjian realizó más de 20 patentes de lo que sería un cajero automático con funciones básicas, no obstante, sus inventos no fueron empelados por el público ya que solo algunas personas selectas tenían acceso a él.

En base al prototipo expuesto, pasaron varios años hasta 1965 cuando John Shepherd-Barron tras llegar tarde por un minuto para retirar dinero planteó la idea de un sistema encargado de las actividades que desempeñaba una persona al distribuir dinero, es por ello que construyó el primer cajero automático adaptándolo a las tarjetas de crédito que ya circulaban en ese entonces; incorporando de la misma forma un código de 6 números como PIN aunque más adelante se cambiaría a 4 números como contraseña.



Con base a los pioneros en este ámbito, fueron procesados miles de cajeros repartidos en masas a los diferentes Bancos Mundiales, trayendo ventajas a los clientes, aunque también es claro que la delincuencia se adaptó a estos sistemas obteniendo métodos de extracción de fondos tras analizar vulnerabilidades. Ante esta situación se han realizado diferentes acciones para frenar tales abusos, por lo que la seguridad informática debe entrar a esta rama y especializarse en delitos, no olvidando la importancia de los cajeros automáticos quienes agilizan procesos.

Desarrollo

El simulador presente de cajero automático presenta características compartidas entre cajeros, como lo son el retiro de efectivo, consulta de saldo, ingreso de dinero y se añadió la opción de estado de seguridad de la cuenta ya que es allí a donde pretendo destinar el proyecto que se expone, describe una interacción real entre el sistema y el cliente que lo solicite, dando respuestas conforme las opciones definidas, el menú principal tiene algunos detalles estéticos con el fin de atraer la atención del consumidor final.

Ya que se recopilaron los conocimientos a lo largo del curso, la inclusión de la mayor cantidad de recursos fue fundamental, retomando aquellos que se encontraban al alcance para evitar caer en un proyecto deficiente y monótono. Cuenta con secuencias de escape al imprimir texto como detalle, la escritura en pantalla, así como su lectura constituyó gran parte del código en el software, el empleo de estructuras de selección fue fundamental en operaciones específicas determinando la ejecución entre un condicional, para las estructuras repetitivas se optó por ejercer el ciclo “do-while” ya que el proyecto reside en un menú principal.

Ahora bien, contando con las “herramientas básicas” al programar, se destinó tiempo para introducir apuntadores cuya finalidad era agilizar parte del código, los arreglos presentes ejecutaron una selección entre dos palabras aleatorias, esto en la parte de seguridad; finalmente, la lectura y escritura de datos en archivos de texto plano determinó la unificación de todas las prácticas ejecutadas durante el semestre

Al tratarse de un prototipo experimental de un cajero automático perfecto, se tiene la idea de un seguimiento prolongado para la mejora en los posibles errores que se presenten durante los próximos semestres; así mismo, la implementación de criptografía avanzada es indispensable al tratarse de recursos económicos que pudieran ser obtenidos de forma ilícita, por consecuente se pretende salvaguardar al cliente de estafas cibernéticas.

Visualizando el proyecto a escalas más grandes, existe esa posibilidad de incluirlo a los bancos actuales mejorando su sistema de seguridad, teniendo esta mentalidad emprendedora sin dificultad podemos adaptar un sistema de cajeros automáticos digitales en los cuales se puedan realizar las transacciones seguras contando con eficiencia como nunca antes, así que al hablar de digitalización hacemos referencia a la adaptación de la plataforma tanto en computadoras personales como a dispositivos móviles.

Tras evaluar el tiempo de realización del mismo considero que su valor monetario actual ronda aproximadamente de \$1000 mxn, dicho valor es asignado por ser un prototipo completo y eficaz, capaz de resolver varios problemas y almacenarlos en su memoria, no obstante, al ser un simulador simplificado actualmente no podría venderse a empresas grandes hasta eliminar posibles fallos

así como la implementación de criptografía en el mismo, por lo tanto se deben optimizar factores que resulten dañinos al consumidor e interconectar plataformas.

Analizando el ciclo de vida de un software el siguiente paso después de la identificación del problema, así como su análisis estructurado, seguimos con el diseño de aquello que puede dar solución al problema planteado; aquí es donde intervienen tres factores muy importantes: el algoritmo, diagrama de flujo (esquemático) y pseudocódigo.

Un algoritmo son los pasos a seguir para resolver cualquier actividad, no necesariamente relacionada a sistemas informáticos; una suma, una receta de cocina, el manual para construir un mueble son ejemplos de algoritmos en la vida cotidiana. Dichos algoritmos son precisos, bien definidos, finitos y eficientes. Cada algoritmo guía el correcto funcionamiento de las actividades, por ello, es indispensable que cumplan con el objetivo propuesto o de lo contrario no serviría.

El desarrollo de un algoritmo que cumpla las necesidades para un cajero automático cuyas funciones son las más comunes es el siguiente. Cabe destacar que debido a la extensión del código se simplificó el algoritmo omitiendo detalles estéticos y de operaciones.

Algoritmo para la creación de un cajero automático

PROBLEMA: Ejecución de un software el cual pueda simular el proceso generado por un cajero automatizado donde plantee diversas características al mostrar opciones, mientras que al finalizar genere un archivo de texto en el cual imprima el saldo actual durante el proceso.

Igualmente tenga la opción de convertir la moneda a la que se le atribuya

RESTRICCIONES: El cajero no puede efectuar retiro de dinero mayor al saldo actual

DATOS DE ENTRADA: Ingreso del tipo de banco al cual se efectuará la operación, banco, cantidad.

DATOS DE SALIDA: Impresión de un archivo con los datos del saldo actual, estado de seguridad, soporte técnico y transferencia.

1. Inicio del algoritmo
2. Mostrar mensaje de bienvenida
3. Solicitar al usuario el tipo de banco al cual pertenece
4. Ingresar al entorno virtual donde se muestran las siguientes opciones: consulta de saldo, retiro de efectivo, transacción de dinero a otra cuenta, estado de seguridad en la cuenta, soporte técnico y cancelar proceso.
5. Si el usuario digita una opción correspondiente a las mencionadas, entonces ejecutar dicha acción
 - 5.1. Consulta de saldo
 - 5.1.1 Mostrar el saldo actual
 - 5.2. Retiro de efectivo
 - 5.2.1 Preguntar la cantidad de dinero a retirar
 - 5.2.1.1 Si el saldo es menor a lo solicitado mostrar error
 - 5.2.1.2 De lo contrario proceder al retiro
 - 5.2.2 Imprimir un ticket como comprobante del saldo actual
 - 5.2.3 Volver al menú principal
 - 5.3. Ingreso de dinero
 - 5.3.1 Ingresar la cantidad de dinero a depositar
 - 5.3.2 Imprimir un ticket como comprobante de lo ingresado
 - 5.3.3 Volver al menú principal
 - 5.4. Estado de seguridad de la cuenta
 - 5.4.1 Informar la seguridad de la cuenta, ya sea segura o insegura
 - 5.4.2 Volver al menú principal
 - 5.5. Soporte técnico
 - 5.5.1 Mostrar los datos de contacto de la empresa o el responsable
 - 5.5.2 Volver al menú principal
 - 5.6. Cancelar proceso (Salir)
6. Preguntar si gusta realizar una donación
 - 6.1 Si quiere entonces restar saldo de la cuenta, si no continuar
7. Si quiere hacer otra operación regresar al punto 5, de lo contrario continuar
8. Imprimir un ticket como comprobante de la cuenta
9. Salir del entorno virtual
10. Fin del algoritmo

Se observa que la escritura de un algoritmo se basa en infinitivos de los verbos los cuales definen las instrucciones a realizar en todo el proceso; igualmente cuenta con un inicio y fin definido lo que lo convierte en un proceso finito.

Retomando el ciclo de vida del software, recordamos que diseñar un problema también existe el desarrollo del diagrama de flujo, cuya finalidad es representar de manera gráfica todos los pasos presentes en el algoritmo, siguiendo una jerarquía al momento de su lectura de arriba hacia abajo y de izquierda a derecha. No obstante, los diagramas de flujo cuentan con características especiales al momento de realizarlos, como por ejemplo el uso correcto de las formas geométricas, ya que su simbología posee atributos especiales estos no pueden ser remplazados ni alterados en las reglas ya establecidas.

La correcta esquematización de los diagramas de flujo son parte indispensable al momento de codificar, ya que desde este punto se observa el cómo se desenvuelve el Software desde el inicio hasta su fin. El uso de conectores (flechas) indica hacia donde se desplazan las instrucciones.

Entre las cuales podemos descartar las siguientes:

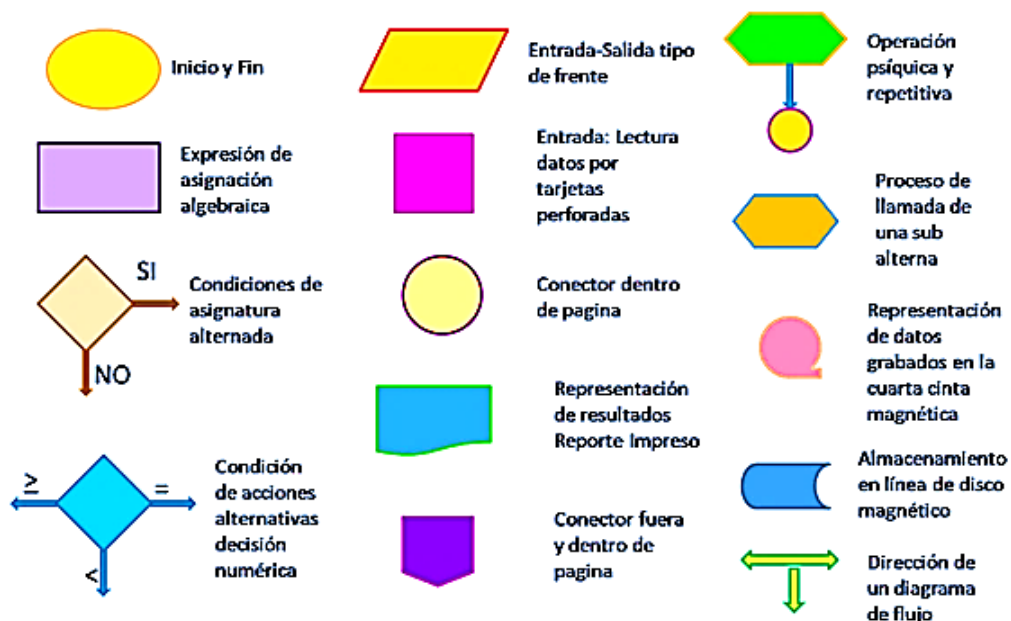
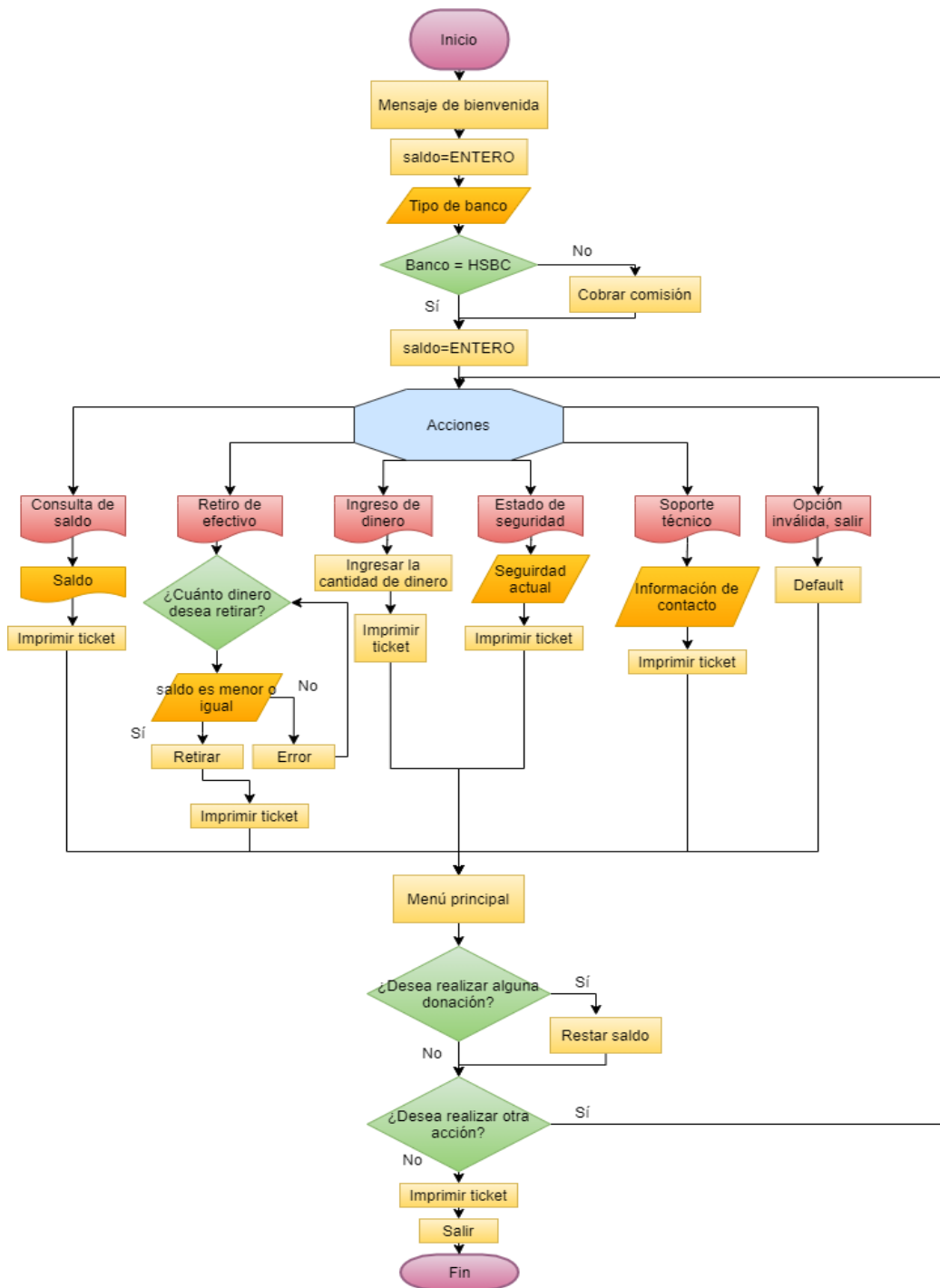


Ilustración 1 Elementos del diagrama de flujo

En relación a los parámetros antes mencionados y retomando el algoritmo establecido, se obtiene el siguiente diagrama de flujo cuya descripción es parcial a causa del tamaño del programa; no obstante, recauda las funciones principales añadiendo nuevas al código a realizar.

Observamos que se incluyen algunas variables a diferencia del algoritmo, igualmente la esquematización es más concisa y ya se interpreta la finalidad.



Ahora bien, se nos presenta el pseudocódigo el cual tiene mayor formalidad y de la misma forma, posee una sintaxis definida que estipula el funcionamiento del código, este tipo de diseño representa una idea clara de un lenguaje de programación, puesto que a partir de este se puede desplazar a cualquier lenguaje de programación formando una clave vital como programador.

El siguiente ejemplo es la representación del pseudocódigo del cajero:

INICIO

saldol: = aleatorio

SI banco: =HSBC ENTONCES

saldol: = saldo

FIN SI

DE LO CONTRARIO

saldo: = saldol-2%

FIN DE LO CONTRARIO

HACER

SELECCIONAR (opción) EN

CASO 1 -> Consulta de saldo

saldof: = saldo+0

IMPRIMIR saldof

SI retiroDinero: = no ENTONCES

IMPRIMIR ticket

FIN SI

DE LO CONTRARIO

Salir

FIN DE LO CONTRARIO

CASO 2-> Retiro de efectivo

ESCRIBIR "¿Cuánto dinero desea retirar?"

LEER retiro

saldof: = saldo-retiro

IMPRIMIR ticket

CASO 3-> Ingreso de dinero

ESCRIBIR "¿Cuánto dinero desea ingresar?"

LEER ingreso

saldof: = saldo + retiro

IMPRIMIR ticket

CASO 4 -> Estado de seguridad

ESCRIBIR Seguridad

CASO 5 -> Soporte técnico

ESCRIBIR Datos de contacto

DEFECTO ->

ESCRIBIR "Opción inválida"

FIN SELECCIONAR

MIENTRAS opción !=5

ESCRIBIR "Gracias por usar el programa"

SI donación: = sí ENTONCES

Donación: = saldof-10%

FIN SI

DE LO CONTRARIO

saldof: = saldof+0

FIN DE LO CONTRARIO

IMPRIMIR ticket

FIN

Avanzando en el proyecto nos encontramos con las herramientas suficientes para proceder a la codificación reuniendo los atributos antes descritos, por lo que se tomó la decisión de elegir un ambiente de programación propicio para desenvolver las habilidades adquiridas hasta el momento, al tratarse de un programa universal se optó por el uso del lenguaje C ya que cuenta con una enorme difusión, así como también es recomendado al momento de iniciar en programación.

La sintaxis fue aprendida y desarrollada simultáneamente en todo momento, por lo que sería para este trabajo ineficiente explicar el comportamiento detallado de operadores o estructuras que presenta; ante esta situación, el apoyo de comentarios durante el código facilita su lectura y provoca una visión comprensiva hacia el programador, es en este momento cuando abarcamos la ejecución repetitiva de todo el código compilándolo constantemente analizando los errores comunes.

Los atributos durante todo este proceso hacen hincapié al ciclo de vida del software, respetando cada operación cuyo propósito reafirma su persistencia a lo largo del tiempo, la codificación resultó ser de las mayores al requerir tiempo por lo que se podría tomar como aquella acción compleja a comparación de las anteriores.

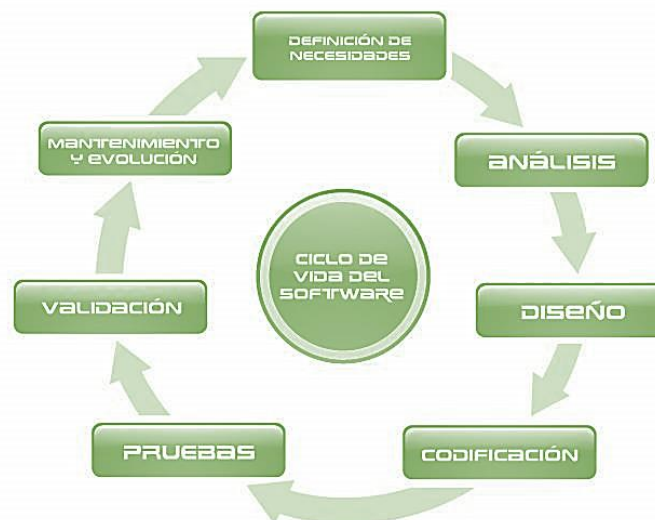


Ilustración 2 Ciclo de vida del Software

Se presenta el código realizado en su formato de texto plano con la finalidad de apreciar lo desarrollado durante el curso, observando los comentarios resaltados en color diferente para su lectura:

```
#include <stdio.h> // printf
```

```
#include <stdlib.h> // rand y RAND_MAX
```

```
#include <unistd.h> // getpid
```

```
#include <time.h> //Time Null
```

```
//Declaración de variables
```

```
int saldo,banco,opcion,final,finalingreso,retiro,ingreso,conver,palabra;
```

```
int dona=100;
```

```
char salir,interr=168,a=219,aa=160,ae=130,ao=162;
```

```
//Case 4 (estado de seguridad de la cuenta por medio de arreglos) firma diseñada
```

```
int numeroAlAzar(int limiteSuperior, int limiteInferior) //Definición de los límites del  
arreglo, esta función conforma el arreglo
```

```
{
```

```
    int valor =rand() % limiteSuperior + limiteInferior;
```

```
    return 0;
```

```
}
```

```
//Función encargada de la dirección de memoria del apuntador obtieneAlAzar
```

```
char *obtieneAlAzar(char *arreglo[], int tam)
```

```
{
```

```
    int indice = rand()%tam+1;//Al igual que en los valores, rand elige un valor a  
partir de la posición 1, no de la 0
```

```
    char *palabra = arreglo[indice];
```

```
    return palabra;
```

```
}
```

```
// Función principal, cuerpo del programa
```

```
int main(){
```

```
    // Hay que alimentar a rand, solamente una vez (seed rand)
```

```
    srand(getpid());
```

```
int saldoInicial = rand()%20000,final; //El saldo es un valor aleatorio, se agrega la
cantidad después de % y toma un valor en ese intervalo
```

```
//Mensaje de bienvenida
```

```
printf("\n\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c",a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a);
```

```
printf("\n\t\t%c Bienvenido a HSBC %c\n",a,a);
```

```
printf("\n\t\t%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c",a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a);
```

```
//Comisión por banco ajeno
```

```
printf("\nIngresa el banco al que perteneces: ");
```

```
printf("\n1) HSBC, 2) SANTANDER, 3) Bancomer, 4) Banamex 5) Otro \n");
```

```
scanf("%i",&banco); //Lectura de la opción seleccionada
```

```
//Condicional de sentencia, se aplica una comisión al tratarse de otro banco
emisor
```

```
if (banco == 1)
```

```
    saldo = saldoInicial+0;
```

```
else
```

```
    saldo = saldoInicial-100; //Cobro de cantidad por banco extranjero
```

```
//Selección de opciones a realizar, aplicación de la sentencia repetitiva do-
while
```

```
do{
```

```
    //Mostrar menú
```

```
    printf("\a\n
```

```
*****");//Sentencia de escape aludiendo al
sonido de una alarma
```

```
printf("\n                * 1) Consulta de saldo                *");
```

```
printf("\n                * 2) Retiro de efectivo                *");
```

```
printf("\n                * 3) Ingreso de dinero                *");
```

```
printf("\n                * 4) Estado de seguridad en la cuenta  *");
```

```
printf("\n                * 5) Soporte técnico                    *,ae);
```

```
printf("\n                * 6) Salir                                *\a\n");
```

```

printf("*****");
//Solicitar la opción que desea
printf("\n\tElige una opci%cn: ",ao);
scanf("%d",&opcion);
switch(opcion)
{
    case 1: //Consulta de saldo
        printf("\n\tSu saldo actual es: %d\n",saldo);

        printf ("\n\t%cDesea realizar otra operaci&cn? 1)Si 2)No
: ",interr,ao);
        scanf ("%d",&salir);
        if(salir== 2) //El uso de == determina que para ese valor
se cumpla la condición
            exit (0); //Detener la ejecución del programa
        else
            printf("\tGracias por consultar su saldo :)");
        break;
    case 2: //Retiro de efectivo
        printf("\tBienvenido, %cCu%cnto dinero deseas retirar?
$,interr,aa);
        scanf("%d",&retiro);
        //sentencia condicional si-de lo contrario
        if(retiro>saldo){//Operadores aritméticos que verifican la
condición
            printf("\t¡No cuentas con fondos suficientes!\n");
        }
        else
        {
            final = saldo - retiro; //Realización de una disminución en
el saldo actual
            printf ("\tEl saldo bancario es: %d\n\n", final);

```

```

    }

    printf ("\n\t%cDesea realizar otra operaci%cn? 1)Si 2)No
: ",interr,ao);

    scanf ("%d",&salir);

    if(salir== 2) //El uso de == determina que para ese valor
se cumpla la condición

        exit (0);

    else

        printf("\tGracias por retirar efectivo :) \n");

    break;

case 3: //Ingreso de dinero

    printf("\tBienvenido. Ingrese la cantidad de dinero a ingresar $");
    scanf("%d",&ingreso); //Lectura de la cantidad solicitada

    if(ingreso>=100000){

        printf("\tCantidad no soportada por el sistema");
//Al ser un valor grande el cajero dicta que la operación no es posible

    }

    else{

        finalingreso=final+ingreso;

        printf ("\tEl saldo bancario es:      %d\n\n",
finalingreso); //Impresión de las operaciones efectuadas o nulas

    }

    printf ("\n\t%cDesea realizar otra operaci%cn? 1)Si 2)No
",interr,ao);

    scanf ("%d",&salir);

    if(salir== 2)

        exit (0);

    else

        printf("\tGracias por ingresar saldo a su cuenta :) \n");

    break;

```



```

        case 4: //Estado de seguridad en la cuenta

            srand(getpid()); //De la misma forma que el saldo inicial, se
            llama de nuevo a la función de valores aleatorios

            srand(time(NULL)); //es un valor aleatorio desde el año 1970 y
            los segundos transcurridos hasta ahora. Nunca se repetirá

            char *palabras[] = {"Segura", "Insegura","Segura",
            "Insegura","Segura", "Insegura"}; //El arreglo de string es llamado por el apuntador
            *palabras

            int tam =sizeof(palabras) / sizeof(char *); // sizeof define
            el tamaño del arreglo, teniendo en primer lugar las columnas y después renglones

            char *palabra = obtieneAlAzar(palabras,tam); //Obtiene
            al azar es una función descrita al inicio del documento, cuyos valores son los
            mostrados

            printf("\tLa cuenta actualmente es: %s\n",palabra);

            break;

        case 5: //Soporte técnico

            printf("\tContacto con el creador del cajero:
            deibydcarranza@outlook.com \n");

            break;

        case 6: //Salir

            break;

        default: //Todo switch cuenta con una función por defecto en caso que
        no este en los intervalos descritos

            printf("\tOpci%cn no v%clida \n",ao,aa);

    }

    }while(opcion!=6); //Hasta no cumplirse este valor el ciclo seguirá su
    funcionamiento

    printf("\n\tGracias por usar el programa");

    printf("\n\t%cDesea realizar alguna donaci%cn de $100 para la Facultad? 1)Si 2)No
    ",interr,ao);

    scanf ("%d",&salir);

    if(salir== 2)

    final=finalingreso+0; //Operación de donación donde se retira una cierta cantidad de
    dinero al saldo

```



```

else
    if(dona>saldo){ //En caso de ser menor a 100 no
se habilita esta opción
        printf("\n¡No cuentas con fondos
suficientes!\n");
    }
    else //De lo contrario se efectúa la donación
    {
        final=finalingreso-dona;
        printf("\n\t\tGracias por su donaci%c\n
:)\n",ao);//ao es el acento ya que se presentan errores al compilar, este es su
número en código ASCII
    }

//Creación de un archivo de texto plano que recopile los datos como saldo
FILE *a; //Se crea un apuntador "a" donde se crea un archivo al llamarse
a=fopen("ticket.txt","w"); //fopen permite abrir un archivo y en caso de no
existir lo crea, los permisos de escritura se expresan con "w"

fprintf(a,"#####
#####");

fprintf(a,"\nCajero autom%ctico realizado por Carranza Ochoa José
David :) \n",aa);

fprintf(a,"-----");

fprintf(a,"\nSu saldo actual es $%d ",final);

fprintf(a,"\nEl tipo de moneda de cambio de su saldo actual es $%d
d%clares\n",conver=final/19,ao);

fprintf(a,"#####
#####"); //Al inicio de las operaciones al trabajar con archivos se debe especificar
el apuntador
printf("\n\tSe ha generado su ticket de comprobante");

fclose(a); //Conclusión de edición del archivo guardando los cambios
return 0;
}

```

Resultados

El proyecto se realizó mediante la herramienta impartida por GNU siendo este el compilador GCC de software libre, de la misma forma, el depurador para la corrección de errores y detección de cambios en las variables; sin embargo, el empleo de Dev-C++ como IDE facilitó a gran medida la depuración a diferencia de un editor de textos como NotePad++.

El proceso de codificación requirió gran tiempo y paciencia al presentarse errores, igualmente la investigación de librerías como de funciones determinó el correcto funcionamiento y control del cajero automático, obteniendo los siguientes resultados en la terminal de Windows.

Mensaje de bienvenida

```
Bienvenido a HSBC
Ingresa el banco al que perteneces:
1) HSBC, 2) SANTANDER, 3) Bancomer, 4) Banamex 5) Otro
```

Selección del banco emisor, en caso que sea derechohabiente no se cobra comisión

Menú principal con las acciones disponibles para realizar

```
Bienvenido a HSBC
Ingresa el banco al que perteneces:
1) HSBC, 2) SANTANDER, 3) Bancomer, 4) Banamex 5) Otro
1

*****
* 1) Consulta de saldo *
* 2) Retiro de efectivo *
* 3) Ingreso de dinero *
* 4) Estado de seguridad en la cuenta *
* 5) Soporte técnico *
* 6) Salir *
*****

Elige una opción:
```

Consulta de saldo (saldo inicial aleatorio):

```
*****
* 1) Consulta de saldo                      *
* 2) Retiro de efectivo                     *
* 3) Ingreso de dinero                     *
* 4) Estado de seguridad en la cuenta      *
* 5) Soporte técnico                       *
* 6) Salir                                 *
*****

Elige una opción: 1

Su saldo actual es: 2264

¿Desea realizar otra operación? 1)Si 2)No : 1
Gracias por consultar su saldo :)
```

Retiro de efectivo

```
*****
* 1) Consulta de saldo                      *
* 2) Retiro de efectivo                     *
* 3) Ingreso de dinero                     *
* 4) Estado de seguridad en la cuenta      *
* 5) Soporte técnico                       *
* 6) Salir                                 *
*****

Elige una opción: 2
Bienvenido, ¿Cuánto dinero deseas retirar? $64
El saldo bancario es: 2200

¿Desea realizar otra operación? 1)Si 2)No : 1
Gracias por retirar efectivo :)
```

Ingreso de dinero a la cuenta

```
*****
* 1) Consulta de saldo                      *
* 2) Retiro de efectivo                     *
* 3) Ingreso de dinero                     *
* 4) Estado de seguridad en la cuenta      *
* 5) Soporte técnico                       *
* 6) Salir                                 *
*****

Elige una opción: 3
Bienvenido. Ingrese la cantidad de dinero a ingresar $300
El saldo bancario es: 2500

¿Desea realizar otra operación? 1)Si 2)No 1
Gracias por ingresar saldo a su cuenta :)
```

Estado de seguridad de la cuenta (simulación aleatoria entre segura e insegura)

```
*****
* 1) Consulta de saldo *
* 2) Retiro de efectivo *
* 3) Ingreso de dinero *
* 4) Estado de seguridad en la cuenta *
* 5) Soporte técnico *
* 6) Salir *
*****

Elige una opción: 4
La cuenta actualmente es: Segura
```

Soporte técnico

```
*****
* 1) Consulta de saldo *
* 2) Retiro de efectivo *
* 3) Ingreso de dinero *
* 4) Estado de seguridad en la cuenta *
* 5) Soporte técnico *
* 6) Salir *
*****

Elige una opción: 5
Contacto con el creador del cajero: deibydcarraza@outlook.com
```

Salida del menú principal e interrogante de donación

```
*****
* 1) Consulta de saldo *
* 2) Retiro de efectivo *
* 3) Ingreso de dinero *
* 4) Estado de seguridad en la cuenta *
* 5) Soporte técnico *
* 6) Salir *
*****

Elige una opción: 6

Gracias por usar el programa
¿Desea realizar alguna donación de $100 para la Facultad? 1)Si 2)No
```

Impresión del ticket como comprobante del saldo

```
Gracias por usar el programa
¿Desea realizar alguna donación de $100 para la Facultad? 1)Si 2)No 1

Gracias por su donación :)

Se ha generado su ticket de comprobante
```

*ticket: Bloc de notas
Archivo Edición Formato Ver Ayuda

```
#####
Cajero automático realizado por Carranza Ochoa José David :)
-----
Su saldo actual es $2400
El tipo de moneda de cambio de su saldo actual es $126 dólares
#####
```

La presente tabla compara las posturas tanto antiguas como actuales de los recursos empelados al finalizar y aquellos que se esperaba a un inicio del proyecto.

	Octubre 2020	Enero 2021
Software y Hardware	Software tales como entornos de desarrollo integrado, cursos de programación y estructura de datos, también una computadora para realizar el proyecto.	Materiales informáticos, se consultaron cursos de programación y de lógica, manejo de la información e investigaciones sobre funciones específicas, empleo de computadoras y dispositivos móviles al momento de codificar.

Recursos económicos (Inversión y ganancias)

Tabla comparativa con las referencias tanto antiguas como vigentes, los datos fueron extraídos de investigaciones anteriores

	Octubre 2020	Enero 2021
Inversión y ganancias	Por el mismo proyecto que estoy desarrollando cobraría un aproximado de \$1,000 hasta terminarlo por completo, e invertiría solo la electricidad consumida	Actualmente cobraría \$2,000 por el mismo proyecto debido a su extensión y complejidad, la información asociada y la investigación. Invertí \$50 por un breve curso

Diagramas de Gantt

Las siguientes tablas exponen los diagramas presentes en la logística y planeación de datos correspondientes a las fechas para cada tarea, se compara entre diferentes épocas

Octubre 2020

[illegible]

Enero 2021

Tareas	Semana		Octubre				Noviembre				Diciembre				Enero			
	Inicio	Final	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Objetivo	1	1																
2. Diseño del algoritmo	2	3																
3. Diseño del diagrama de flujo	4	5																
4. Desarrollo del pseudocódigo	6	7																
5. Desarrollo del código	8	12																
6. Prueba del código	13	15																
7. Implementación del proyecto	15	16																
8. Exposición de resultados	16	16																

Finalmente se presentan los enlaces de referencia tanto del canal de YouTube como del repositorio creado a inicios del semestre 2021-1, dicho canal de YouTube se encuentra en formato público y tiene es el siguiente:

- YouTube: <https://www.youtube.com/channel/UC2NcCgADqoZrtZg6jTWvf3w>

Mientras que el repositorio de GitHub adscrito en la documentación secuencial de las habilidades tomadas fue:

- GitHub: https://github.com/DeibydCarranza/practica1_fdp

Conclusiones

El desarrollo de este proyecto fue fundamental para comprender principios básicos de programación, los conocimientos y destrezas adquiridas representaron un cambio en la visión de lo que es la Ingeniería en Computación. Con anterioridad visualizaba el programar como sentarse frente a una computadora a escribir líneas de texto corrido, sin embargo, ahora puedo comprender el error al cual me enfrentaba y es distinguir los diferentes procesos que se llevan a cabo antes de codificar un programa, comprendí la importancia de cada etapa en el Ciclo de vida del Software hasta el momento de diseñar mi proyecto.

El software libre determinó la compilación y edición del código actual, gracias a este entorno descubrí la importancia del Software libre como medida en apoyo a los avances tecnológicos, ya que no representa un espacio cerrado para los sectores privados, si no que siempre está en constante mejora debido a la comunidad que aporta conocimiento mientras adquiere el mismo, dicha retroalimentación ocasiona que el software libre se constituya como el perfecto candidato en la creación de Software de manera abierta.

Richard Stallman asentó las bases la FSF (Free Software Fundation) para el crecimiento tecnológico creando así una nueva revolución digital orientada a la mejora en los procesos al trabajar en conjunto como comunidad. Por su parte, la IEEE siendo uno de los organismos reguladores en tecnología, se apoya de los conocimientos generales para convertirlos en estándares hacia la población; es aquí donde surge la denominada Revolución Industrial 4.0 por la cantidad de avances (principalmente digitales) que se han disparado en las últimas décadas.

Retomando el proyecto citado, fue vital para su óptimo desarrollo la conjunción de todas estas áreas ya mencionadas, trayendo nuevas ideas en cada momento que realizaba alguna modificación a mi proyecto; esto lo ejemplifico tras haber cambiado de proyecto a mitad de su desarrollo por otro, en el cual trabajé de forma cómoda y segura, no esperaba adquirir tanto conocimiento en la materia y hasta este punto me retracto de mi pensamiento inicial.

Con las investigaciones realizadas sobre crímenes cibernéticos me encuentro más entusiasmado para combatir a aquellos delincuentes; por ello, este proyecto define el inicio de una trayectoria de mejoras y optimizaciones hacia el mismo, esperando ejercer las habilidades demostradas en el proyecto.

Finalmente, reconozco la importancia de la impartición de esta materia en los sectores estudiantiles, ya que fomenta un pensamiento más crítico y lógico para el procesamiento de información, resolución de problemas y actividades.

Referencias

Historia de los bancos - CMF Educa. Comisión para el Mercado Financiero. (s. f.). EDUCA. Recuperado 29 de enero de 2021, de <https://www.cmfeduca.cl/educa/600/w3-printer-26922.html>

¿Qué servicios ofrecen los Bancos? - CMF Educa. Comisión para el Mercado Financiero. (s. f.). EDUCA. Recuperado 29 de enero de 2021, de <https://www.cmfeduca.cl/educa/600/w3-article-27428.html>

Utilizar, N. (2020, 14 julio). *Historia de los cajeros automáticos.* BBVA NOTICIAS. <https://www.bbva.com/es/historia-de-los-cajeros-automaticos/>

Lucas, J. (2020, 10 septiembre). *Qué es C: Características y sintaxis.* OpenWebinars.net. <https://openwebinars.net/blog/que-es-c/>

P. (2019, 26 febrero). *Obtener números aleatorios en C.* Parzibyte's blog. <https://parzibyte.me/blog/2019/03/21/obtener-numeros-aleatorios-c/>

Facultad de Ingeniería. (2018, 6 abril). *Manual de prácticas del laboratorio de Fundamentos de programación.* UNAM. http://odin.fi-b.unam.mx/salac/practicasp/MADO-17_FP.pdf

DevDocs. (2018). DevDocs. <https://devdocs.io/c/chrono/time>

ISO/IEC 9899:TC3. (2007). Open STD. <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1256.pdf>

Vídeo: Tutorial de cómo Seleccionar al Azar en Lenguaje C, DuarteCorporation Tutoriales (2016). Enero 2021 recuperado de <https://www.youtube.com/watch?v=J47WHf8lQI8>

Vídeo: 24. Programación en C - Condicionales - La sentencia switch, Programación ATS (2016). Enero 2021 recuperado de <https://www.youtube.com/watch?v=UbK RifnA0FU>

Vídeo: 20.- Curso Básico de C - Apuntadores Llamadas por Referencia, CódigoFacilito (2017). Enero 2021 recuperado de <https://www.youtube.com/watch?v=1S9cv6LNF9M>