UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Software y Tecnologías Emergentes

Nombre Alumno:
Eliel Alfonso Ontiveros Ojeda

Grupo:
932

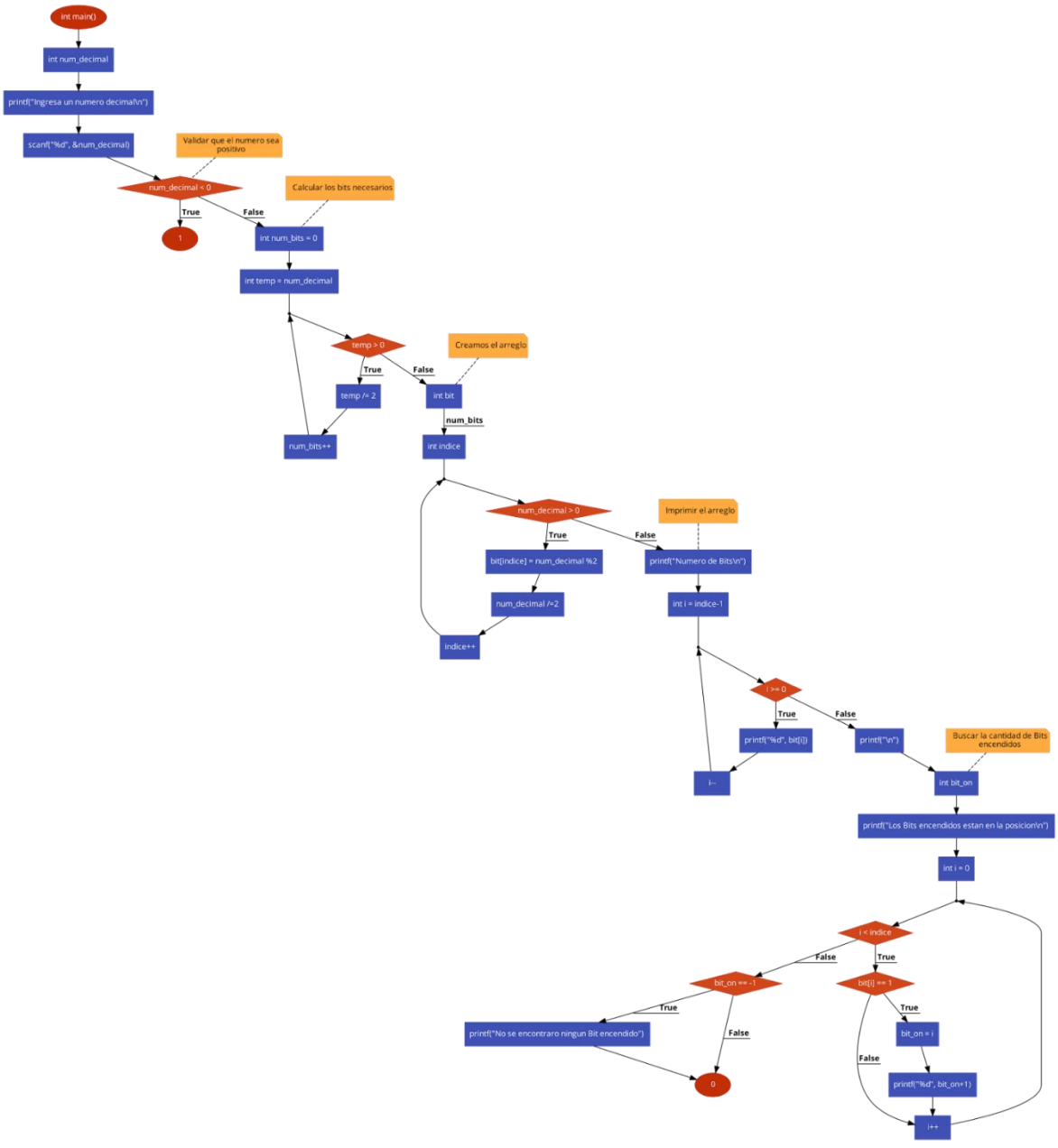
Repositorio:
https://github.com/Eliel-Ontiveros/Practica_2.git

08/09/2023

Actividad 1

Crear un programa que permita a los usuarios ingresar un número entero, especificar el número de bits que se deben considerar y luego analizar el número en términos de bits encendidos, posiciones y representación binaria.

Diagrama de Flujo



Documentación

```c
#include <stdio.h>

int main()
{
    int num_decimal;

    printf("Ingresa un numero decimal\n");
    scanf("%d", &num_decimal);

```

Agregamos las librerías que utilizaremos (stdio.h), nombramos la función principal (int main) y declaramos nuestras variables (int num_decimal). Posteriormente le pedimos al usuario que ingrese un número decimal (scanf).

```c
if(num_decimal < 0)
    {
        return 1;
    }
```

Con el uso del condicional if nos encargamos de Validar que el número que ingreso el usuario sea un número positivo.

```c
int num_bits = 0;
    int temp = num_decimal;

    while(temp > 0)
    {
        temp /= 2;
        num_bits++;
    }

```

Calcular la cantidad de bits necesarios para representar el número

```c
int bit[num_bits];
    int indice;

    while (num_decimal > 0)
    {
        bit[indice] = num_decimal %2;
        num_decimal /=2;
        indice++;
    }

```

Crear un arreglo para almacenar los bits.

Convertir el número decimal en su representación binaria y almacenarla en el arreglo

```c
printf("Numero de Bits\n");

    for (int i = indice-1; i >= 0; i--)
    {
        printf("%d", bit[i]);
    }

    printf("\n");
```

Imprimir la representación binaria del número.

```c
int bit_on;

    printf("Los Bits encendidos estan en la posicion\n");

    for(int i = 0; i < indice; i++)
    {
        if (bit[i] == 1)
        {
            bit_on = i;
            printf("%d", bit_on+1);

        }

    }

    if (bit_on == -1)
    {
        printf("No se encontraro ningun Bit encendido");
    }


    return 0;

}
```

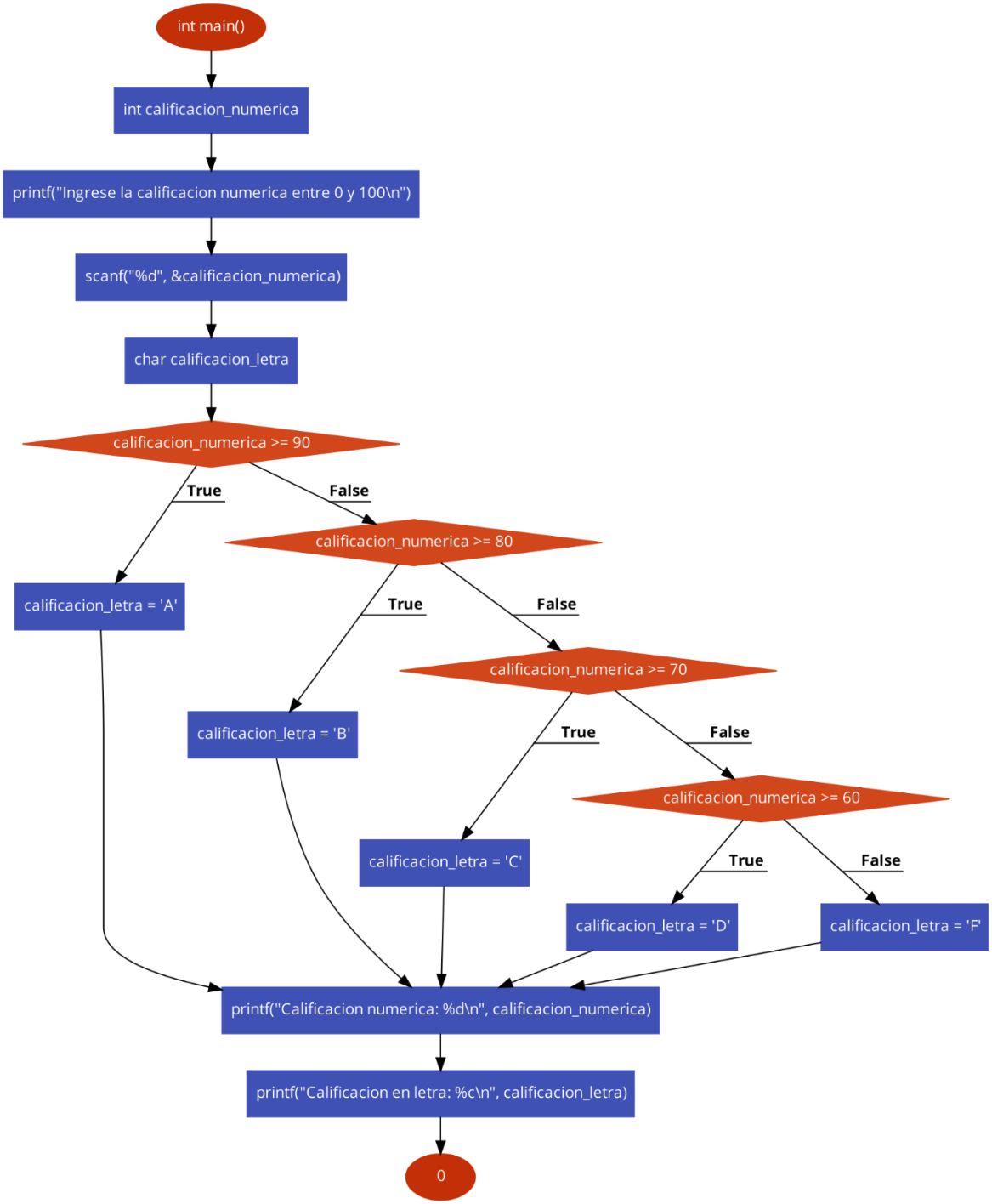
Declaramos nuestra variable en la cual guardaremos la cantidad de Bits encendidos.

Con la utilización del ciclo For buscamos la cantidad de Bits encendidos

Actividad 2

Crear un programa que tome una calificación numérica y la convierta en una calificación en letra utilizando operadores ternarios.

Diagrama de Flujo



Documentación

```c
#include <stdio.h>

int main()
{
    int calificacion_numerica;
    printf("Ingrese la calificacion numerica entre 0 y 100\n");
    scanf("%d", &calificacion_numerica);
```

Agregamos las librerías que utilizaremos (stdio.h), nombramos la función principal (int main) y declaramos nuestras variables (calificación_numerica)

```c
char calificacion_letra = (calificacion_numerica >= 90) ? 'A' :
                          (calificacion_numerica >= 80) ? 'B' :
                          (calificacion_numerica >= 70) ? 'C' :
                          (calificacion_numerica >= 60) ? 'D' : 'f';
```

Declaramos la variable en la cual guardaremos la equivalencia en letra de la calificación en decimal.

Procedemos a la utilización de las funciones ternarias para desarrollar la lógica de nuestro problema.

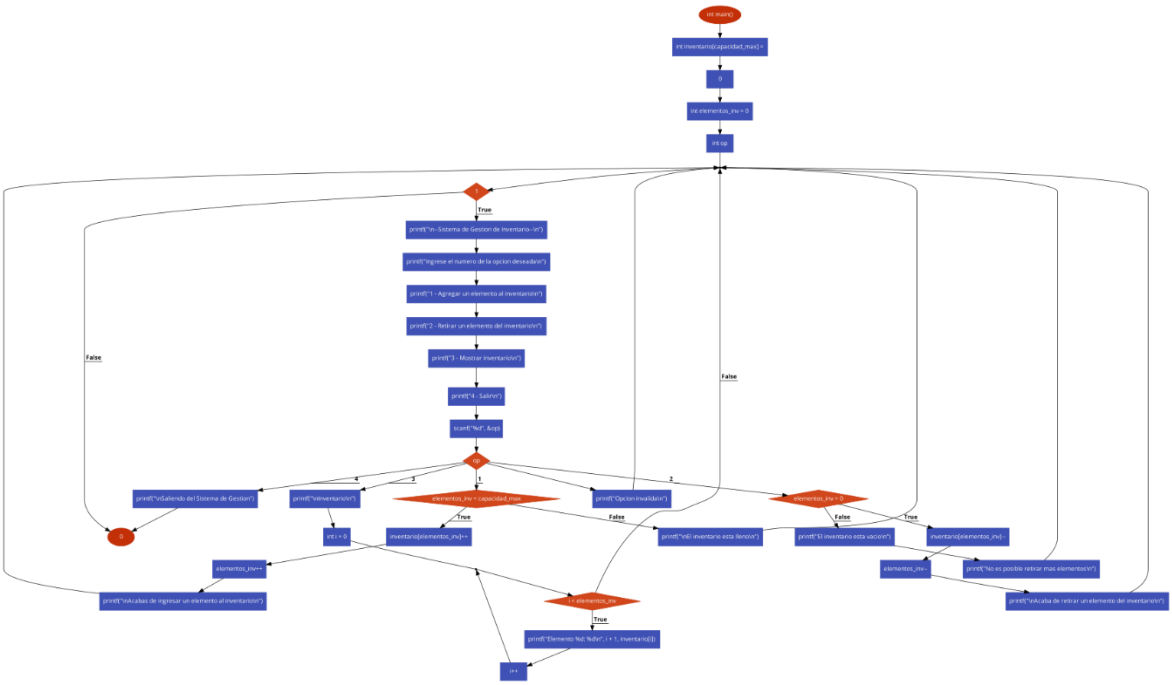
```c
    printf("Calificacion numerica: %d\n", calificacion_numerica);
     printf("Calificacion en letra: %c\n", calificacion_letra);

     return 0;
}
```

Imprimimos las equivalencias de las calificaciones de numérica a letra y finalizamos nuestra función principal.

Actividad 3

El objetivo de este ejercicio es simular un sistema básico de gestión de inventario en C, donde los usuarios pueden agregar y retirar elementos del inventario.

Diagrama de Flujo



Documentación

El inventario se representa como un arreglo de enteros inventario con una capacidad máxima definida por la constante capacidad_max. Aquí tienes una explicación detallada de cada parte del código:

- Se define la capacidad máxima del inventario con la constante capacidad_max.
- Se inicializa un arreglo inventario de tamaño capacidad_max con todos los elementos inicializados a 0.
- Se declaran las variables elementos_inv para llevar un registro de la cantidad de elementos en el inventario y op para almacenar la opción ingresada por el usuario.
- Se inicia un bucle while (1) que permite que el programa se ejecute continuamente hasta que el usuario elija la opción de salir.
- Dentro del bucle, se muestra un menú al usuario y se lee la opción ingresada con scanf.
- El programa utiliza una estructura de control switch para ejecutar la funcionalidad seleccionada por el usuario.
- Para agregar un elemento al inventario (case 1), se verifica si hay espacio disponible y se incrementa el contador de elementos y el valor en el arreglo.
- Para retirar un elemento del inventario (case 2), se verifica si hay elementos en el inventario y se decrementa el contador de elementos y el valor en el arreglo.
- Para mostrar el inventario (case 3), se recorre el arreglo y se muestra el contenido.
- Para salir del programa (case 4), se imprime un mensaje y se retorna 0 para finalizar el programa.
- Se maneja una opción por defecto para manejar entradas inválidas del usuario.