


| | | |
|---|-------------------------|--|
|  | <h2>Proyecto Final</h2> | |
| Facultad de Ingeniería | Laboratorio de docencia | |

Profesor: MARCO ANTONIO MARTINEZ QUINTANA

Asignatura: ESTRUCTURA DE DATOS Y ALGORITMOS I

Grupo: 17

Proyecto: PLATAFORMA DE APRENDIZAJE

Alumno: José Luis Arroyo Chavarría

No. de Lista : 5

Semestre: 2

Fecha de entrega: 22/05/2020

Observaciones:

CALIFICACIÓN: _____

Objetivo:

Tener una plataforma en donde este los temas que pueden ver tanto niños como adolescentes y/o adultos jóvenes con temarios, definiciones, ejemplos y ejercicios dependiendo del grado o año que este cursando

Alcance de su proyecto:

Este proyecto está pensado para que en situaciones de no poder asistir presencialmente a la escuela como la presencia de un virus, de un paro de actividades educativas o de cualquier otro problema que podría pasar tanto a todo el personal como a un par de alumnos. Uno de estos casos más reciente es el virus COVID-19.

Introducción:

Durante la estancia de la cuarentena he visto como la mayoría de mis conocidos, en noticias y en mi experiencia como esta situación del virus ha afectado la situación académica, tanto así que la mayoría de alumnos no pueden entender todos los temas que las escuelas quieren dar dependiendo de su carrera o de su grado y si esto en un futuro si pasara otra vez y no mejorara podríamos tener el mismo problema o uno peor que el actual.

Una de las soluciones más posibles y actual que tenemos son las páginas web de la universidad como por ejemplo la DCB en parte de la Facultad de ingeniería que tiene temarios, series, bibliografías y entre otros materiales de apoyo, aunque estas páginas nos den una idea no tienen ejemplos a utilizar para el estudio de exámenes o trabajos que el alumno debe presentar y mientras tanto en el caso de primaria y secundaria no tienen esos servicios ya que el gobierno no implementa mucho en este ámbito y que en cierta forma no estaban preparados para una situación a mayor escala.

Pero esto puede solucionarse con un programa que sea igual que actualmente tenemos en nuestra facultad pero mejorado y más agradable para el usuario con temarios, definiciones, ejemplos y ejercicios pero añadiendo referencias a videos, libros y páginas web que el usuario puede reforzar su aprendizaje

Desarrollo:

Una manera de realizar este proyecto podemos utilizar o basarnos en las pestañas que utilizan varias páginas web como una lista simple.



Pero siento que una manera más apropiada para un tipo de prototipo sería mejor utilizar datos abstractos (struct) para así ordenar la información que se va a utilizar dependiendo de las escuelas

- **Definición:**

- **Datos abstractos:**

Un tipo de dato abstracto (TDA) es un conjunto de datos u objetos creado de manera personalizada por un programador para un fin específico. Un TDA es una abstracción que permite modelar las características de un elemento en particular.

Un tipo de dato abstracto se puede manipular de forma similar a los tipos de datos que están predefinidos dentro del lenguaje de programación, encapsulando más información, según se requiera.

La implementación de un tipo de dato abstracto depende directamente del lenguaje de programación que se utilice. En lenguaje C los tipos de dato abstracto se crean mediante las estructuras (struct).

La sintaxis para crear estructuras en lenguaje C está definida por la palabra reservada struct, seguida del nombre de la estructura y, entre llaves, se definen el número y tipo de variables que definan al nodo (abstracción), es decir:

```

struct nodo {

    tipoDato elemento1;

    tipoDato elemento2;

    ...

    tipoDato elementoN;

};

```

Un ejemplo de esto que donde hemos visto a lo largo de la materia es en los ejercicios de la practica 3 del código del nodo de película

```

struct pelicula{
    char *nombre;
    char *genero;
    short año;
    short numDirectores;
    char *directores[10];
};

void imprimirDatosPelicula(struct pelicula);
struct pelicula llenarDatosPelicula(char *, char*, short, short, char *[10]);

int main(){
    char *directores[10];
    directores[0] = "Lana Wachowski";
    directores[1] = "Andy Wachowski";
    struct pelicula matrix = llenarDatosPelicula("The matrix", "Ciencia ficción", 1999, 2, directores);
    imprimirDatosPelicula(matrix);
    return 0;
}

struct pelicula llenarDatosPelicula(char *nombre, char *genero, short año, short numDirectores, char *directores[10]){
    struct pelicula movie;
    movie.nombre = nombre;
    movie.genero = genero;
    movie.año = año;
    movie.numDirectores = numDirectores;
    for (int count = 0 ; count < movie.numDirectores ; count++){
        movie.directores[count] = directores[count];
    }
    return movie;
}

void imprimirDatosPelicula(struct pelicula movie){
    printf("PELICULA: %s\n", movie.nombre);
    printf("GENERO: %s\n", movie.genero);
    printf("AÑO: %d\n", movie.año);
    printf("DIRECTOR(ES):\n");
    for(int count = 0; count < movie.numDirectores; count++){
        printf("%s\n", movie.directores[count]);
    }
}

```

Pero no solo utilizaremos struct sino también if y switch para hacer los menús y los ejercicios que reforzara al usuario con los temas a aprender

- **Definición:**

- **IF :**

En esta estructura se evalúa la expresión lógica y, si se cumple (si la condición es verdadera), se ejecutan las instrucciones del bloque que se encuentra entre las

llaves de la estructura. Si no se cumple la condición, se continúa con el flujo normal del programa

La estructura de control de flujo más simple es la estructura condicional if, su sintaxis es la siguiente:

```
if (expresión_lógica) {  
    // bloque de código a ejecutar  
}
```

➤ **Switch :**

La estructura switch-case evalúa la variable que se encuentra entre paréntesis después de la palabra reservada switch y la compara con los valores constantes que posee cada caso (case). Al final de cada caso se ejecuta la instrucción break, si se omite esta palabra reservada se ejecutaría el siguiente caso, es decir, se utiliza para indicar que el bloque de código a ejecutar ya terminó y poder así salir de la estructura.

Si la opción a evaluar no coincide dentro de algún caso, entonces se ejecuta el bloque por defecto (default).

La sintaxis de la estructura switch-case es la siguiente:

```
switch (opcion_a_evaluar){  
    case valor1:  
        /* Código a ejecutar*/  
        break;  
    case valor2:  
        /* Código a ejecutar*/  
        break;  
    ...  
    case valorN:  
        /* Código a ejecutar*/  
        break;
```

default:

/* Código a ejecutar*/

}

Lenguaje a utilizar: C

- **Algoritmo:**

Entradas:

- Seleccionar un tema a aprender

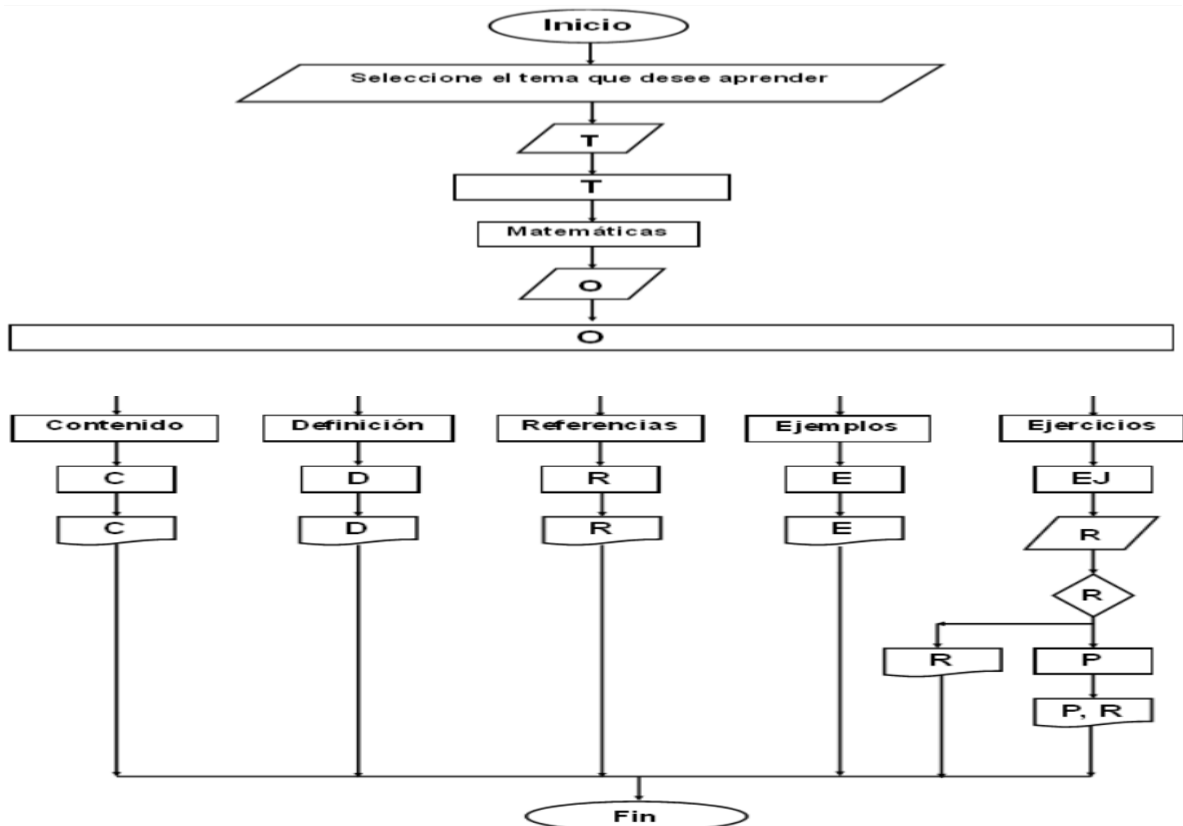
Salida:

- Material educativo sobre el tema a aprender

Proceso:

- Utilizar un menú donde el usuario puede seleccionar lo que desea a saber del tema seleccionado

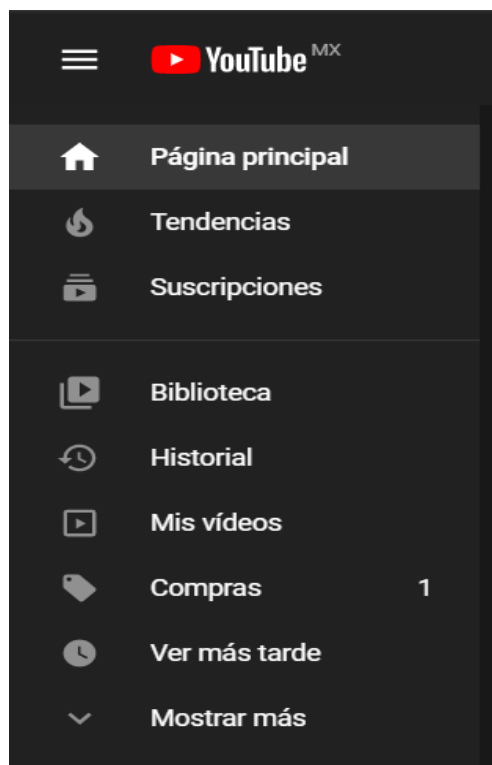
- **Diagrama de flujo (Prototipo):**



| Variable: | Descripción: | Tipo: |
|-----------|--------------|-------------|
| T | Tema | Real |
| O | Opción | Real |
| C | Contenido | Información |
| D | Definición | Información |
| R | Referencia | Información |
| E | Ejemplos | Información |
| EJ | Ejercicios | Información |
| P | Preguntas | Información |
| RES | Respuesta | Real |

Modelo a utilizar:

Un modelo que podríamos utilizar para realizar un menú podría ser uno parecido a uno que usa la página Youtube



De ahí se podría basarse para poner los temas que se van a ver y ahí dividirse en lo que van relacionados como por ejemplo:

Área de matemáticas

- Calculo
 - Temario
 - Definiciones
 - Definiciones
 - Referencias
 - Ejemplos
 - Ejercicios

También podría tomarse un modelo tradicional que se podría uno así:

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                        PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

MENU PRINCIPAL

[1]. Insertar nuevo registro
[2]. Mostrar listado de registros
[3]. Eliminar un registro
[4]. Buscar registro por numero o fecha
[5]. Modificar un registro
[6]. Eliminación física de registros
[7]. Salir

Ingrese su opción: [.]
```

Tendría la misma mecánica pero se le podría añadir una aplicación para que el profesor pueda poner tareas, ejercicios y ejemplos.

Conclusión:

Si este programa puede ser terminado al menos en su parte de prototipo puede hacer grandes cosas y cambios en el sistema educativo tanto presencial como en línea para que los usuarios tengan un mejor aprendizaje y entendimiento en los temas a enseñarse junto con la cercanía de sus profesores. Aunque si es cierto que puede haber unos conflictos en su actualización de los temarios de cada escuela, no se puede dudar que podría ser una gran ayuda en el aprendizaje.

Referencias:

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.