

Programación  
Estructurada

# Avances del Proyecto



Alumnos: Daniel Gutiérrez y Raymundo  
Mora

Profesor: Emilio Gabriel Rejón Herrera

202





## Requerimientos Funcionales

1. El programa antes de iniciar cualquier proceso debe preguntarle al usuario que opción desea efectuar, tomar esa decisión y llevarla a cabo.
2. Si el usuario no da un valor valido para efectuar algunas respuestas, elegir una opción o algo parecido el programa debe mantenerse estático hasta que el usuario de un valor valido para el programa.
3. El programa debe limpiar la pantalla al cambiar de acción, esto para un mejor entendimiento del funcionamiento de este ya sea por pruebas o ya en su uso.
4. Las respuestas del usuario serán guardadas en un vector para poder manejarlas juntas de una manera más efectiva, estas deben ser pedidas en un orden específico concorde a las preguntas para no generar errores a la hora de dar un resultado.
5. Las respuestas dadas por el usuario deben ser comparadas con las características de los suelos por medio de un vector de estructuras que contiene la información de estas para poder dar un resultado aproximado más certero.
6. El resultado final debe ser comparado con la zona dada por el usuario, se debe comparar lo que se puede cultivar en el tipo de suelo con lo que se produce mayormente en la zona anteriormente especificada para darle unas mejores sugerencias de cultivo al usuario.
7. Cuando se pida la zona del usuario será de forma escrita y no de forma opcional, se comparará el string con los datos en el programa para identificar la zona esto con <string.h>.
8. Al tener listo el resultado, deberá tener los datos del suelo resultante junto con los cultivos resultantes, así mismo con una descripción/información sobre el mismo en un archivo txt. Cuando el usuario seleccione la segunda opción, se pedirá el tipo de suelo que desea investigar, la opción es múltiple, esta se guardará y se le enviara a una función que imprima la información deseada.
9. Al elegir la tercera opción se pedirá que se escriba la zona (estado) donde el usuario se encuentra, (RF no. 7) está se comparará con los datos del programa para imprimir la información, la cual contiene información sobre la producción de esa área, datos breves sobre el cultivo entre otros datos interesantes.
10. El programa no parará hasta que el usuario seleccione la opción 4 “salir”, hasta que eso no pase le programa seguirá funcionando, tomando las opciones del menú (1-3) las veces que el usuario desee hasta que se seleccione dicha opción.

## Requerimientos No funcionales

1. El uso de estructuras es indispensable para el óptimo funcionamiento del programa, estos se utilizan para almacenar la información de las características de los suelos y hacer las comparaciones con las respuestas del usuario.
2. Para un mejor manejo de información se implementarán apuntadores para un mejor manejo de las respuestas y de la información para acto seguido sacar un resultado.
3. El menú debe tener una pequeña descripción que sirva como guía para que el usuario no se sienta perdido a la hora de usar el programa por primera vez.
4. Las preguntas del programa, así como las respuestas deben tener un lenguaje informal por cuestiones de entendimiento con el usuario.
5. Las imágenes en la muestra de resultados o de información son de vital importancia ya que un ejemplo visual ayuda al usuario a ver en gran medida si el resultado fue correcto o la información buscada es igual a lo que poseen.

## Casos de Uso

-  El usuario accede al programa para seguidamente elegir una de las cuatro opciones disponibles según lo que necesite (Quiero Saber, Tipos de Suelo, Mi Zona y salir).  
Caso contrario: El programa pedirá de nuevo la opción.
-  Si el usuario elige la primera opción deberá contestar una serie de preguntas relacionadas con características del suelo, las respuestas se guardarán en un vector y luego serán comparadas con la información para dar un resultado aproximado.  
Caso contrario: Se pedirá que de una opción válida.
-  Se le preguntará la zona donde se encuentra para hacer una comparación de lo que se puede cultivar en su suelo con lo que mayormente se produce en la zona, el usuario debe escribir la zona(estado) donde se encuentra.  
Caso contrario: Si no se encuentra la zona escrita por el usuario se pedirá que la vuelva a escribir.
-  Si se elige la segunda opción el usuario deberá elegir qué tipo de suelo quiere investigar, una vez seleccionado el tipo de suelo del que se desea saber, el programa buscará la información y luego se le mostrará al usuario, esta información contiene las características del suelo, que es lo recomendable cultivar

en ese suelo y en casos especiales tips para hacer más fácil la germinación y el mantenimiento del cultivo.

Caso contrario: Se pedirá que se vuelva a indicar la opción.

- Al elegir la tercera opción el usuario debe escribir la zona donde se encuentra para poder buscar la información, estas zonas hacen referencia a los estados de la república, esta información contendrá lo que más se produce en ese estado, ya sean frutas vegetales, semillas o pasturas.

Caso contrario: Se pedirá que se ingrese otra vez la información.

Matriz de Requerimientos			
Requerimiento	Descripción Breve	Prioridad	Opción
RF1	El usuario debe dar primero las instrucciones.	Básico	Menú
RF2	El programa no debe efectuar ninguna acción con datos inválidos	Básico	1 y 2
RF3	El programa debe limpiar la pantalla	Básico	1, 2 y 3
RF4	Las respuestas del usuario deben ser guardadas en un vector	Básico	1
RF5	Las respuestas dadas por el usuario deben ser comparadas	Vital	1
RF6	El resultado concorde a la zona	Vital	1
RF7	Pedir la zona de forma escrita	Básico	1, 3
RF8	Imprimir la información completa del resultado	Vital	1
RF9	Mostrar la información del suelo seleccionado	Vital	2
RF10	Mostrar la información de la zona	Vital	3
RF11	El programa funciona hasta que el usuario salga	Básico	1, 2 y 3
RNF1	El uso de estructuras en el programa	Vital	1
RNF2	El uso de apuntadores para el manejo de información	Vital	1
RNF3	Mensajes Guías para el usuario	Básico	1, 2 y 3
RNF4	Lenguaje informal para un mejor entendimiento	Básico	1, 2 y 3
RNF5	Imágenes como muestra visual del resultado o información	Básico	1, 2, y 3

## Informe del avance general de la implementación de dichos requerimientos

Los requerimientos funciones aumentar y en algunos casos cambiaron por cuestiones de comprensión del programa y porque los requerimientos actuales se adaptan mejor al programa en sí.

Los requerimientos no funciones cambiaron casi en su totalidad por ser básicamente irrelevantes y en algunos casos erróneos y los actuales son más útiles y se adaptan mejor a lo que se necesita a la hora de hacer el programa.

El programa se ha desarrollado muy apegado a los requerimientos actuales y se ha desarrollado de forma organizada, bien documentada y apegada a lo planeado en el plan de desarrollo.

En general el programa va bien, los errores se han podido resolver con eficacia y por el momento el único detalle es el tiempo, ya que la aceptación de la propuesta y las circunstancias pandémicas nos han dejado un poco cortos de tiempo, sin embargo se aumentaran un par de sesiones extra para compensar el tiempo perdido, el programa lleva a la fecha de entrega de este reporte un avance de un 60% aproximadamente y se espera tener terminado el programa para la fecha solicitada.

## Estándar de Codificación

- Los nombres de variables, funciones y apuntadores han sido relativamente trivial y representativo de lo que hacen o representan.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 //Funcion del menu principal//
6 void menu();
7 //Funcion de la opcion//
8 int seleccion();
9 //Funcion de accion//
10 void accion();
11 //Respuestas del usuario//
12 int respuesta();
13 //Informacion de los suelos//
14 int informacion(int* respuestas);
15 //Esta funcion calcula el resultado del usuario//
16 void resultado(int* respuesta);
17 //Esta funcion imprime el resultado de las respuestas del usuario//
18 int imprimir();
19
20 /*Esta estructura nos dejara tener informacion precisa sobre los
21 tipos de suelo*/
22 struct suelo{
23
24     int humedad[3];
25     int textura[3];
26     int color[3];
27     int pedregoso[3];
28     int drenaje[3];
29     int retencion[3];
30 };
31
32 int main(int argc, char *argv[]) {
33
34     accion();
35
36     return 0;
37 }
38
```

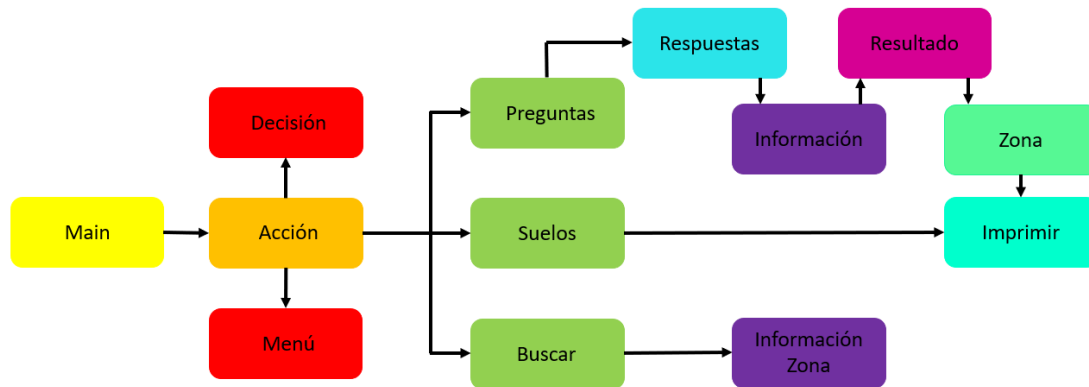
- No se implementaron macros
- Las bibliotecas utilizadas hasta el momento son: <stdio.h>, <stdlib.h>, <string.h>

- La cuestión de entrada, proceso y salida, están repartidas por funciones, en el código hay funciones que sirven como entrada de datos para pasarlos a funciones de proceso y por último pasar a funciones de salida.

```
47 //Funcion de la opcion//
48 int seleccion() {
49
50     int opcion;
51     //Entrada: Leemos la opcion que desea el usuario//
52     scanf("%d", &opcion);
53
54     return opcion;
55 }
56
57 //Funcion de accion//
58 void accion() {
59
60     int control=0; //Declaramos el control para el bucle//
61
62     //bucle que mantiene activo el programa o lo cierra//
63     do{
64
65         //Salida: Llamamos la funcion para desplegar el menú//
66         menu();
67
68         //Proceso: Vemos que opcion tomo el usuario//
69         switch(seleccion()){
70
71             case 1:
72                 //Limpiamos la pantalla para una mejor percepcion visual//
73                 system("cls");
74
75                 //Salida: Mensaje//
76                 printf("\nEsta Seccion buscará por sus respuestas el tipo de suelo que tiene\n");
77                 printf("Pulse cualquier botón para comenzar");
78
79                 //Limpiamos la pantalla para una mejor percepcion visual//
80                 system("cls");
81
82                 //Proceso: Llamamos a la funcion respuesta para hacer las preguntas al usuario//
83                 respuesta();
84
85             ~
86         }
87     }
88
89 //Esta funcion imprime el resultado de las respuestas del usuario//
90 int imprimir() {
91
92     FILE * flujo;
93
94     int x=0, c;
95
96     switch(x){
97
98         case 0: //Salida: Leer la informacion del suelo Arenoso//
99
100             //Direccion del archivo//
101             flujo = fopen("C:\\Users\\Daniel\\Documents\\C\\Emilio Estructurada\\Proyecto\\Descripciones\\Arenosos.txt", "rt");
102
103             if(flujo == NULL){
104                 printf("Error al tratar de leer el archivo");
105                 return 1;
106             }
107
108             while((c=fgetc(flujo))!=EOF){
109                 if(c=='\n'){
110                     printf("\n");
111                 }
112                 else{
113                     putchar(c);
114                 }
115             }
116
117             fclose(flujo); //Cerramos el flujo de datos//
118
119             break;
120
121         case 1: //Salida: Leer la informacion del suelo Calizo//
122
123             flujo = fopen("C:\\Users\\Daniel\\Documents\\C\\Emilio Estructurada\\Proyecto\\Descripciones\\Calizos.txt", "rt");
124
125             ~
126         }
127     }
```






- Por la cohesión de las funciones estas se van llamando de manera secuencial, podríamos decir que es un poco similar a un modelo incremental, pero descartamos llamarlo así porque hay funciones que llaman a mas de una, ya sea porque necesitan obtener datos, por qué se necesita mostrar un menú o mensaje o porque se necesita hacer un proceso diferente.

## Diagrama de Bloques



Actividad	Sesión 1	Sesión 2	Sesión 3	Sesión 4	Sesión 5	Sesión 6	Sesión 7	Sesión 8
Elegir la propuesta del proyecto								
Investigar antecedentes y hacer entrevistas sobre la propuesta								
Planificación del proyecto y de su estructura								
Investigar, redactar y organizar la información								
Codificación								

Investigación sobre tecnologías o metodologías								
Resolución de problemas con la codificación								
Redactar los avances del proyecto								

Nombre	Contribución	Asistencia	Porcentaje
Daniel Eduardo Gutiérrez Delfín	 Idea principal  Proceso de desarrollo  Casos de Uso  Requerimientos Funcionales	6/6	70%
Raymundo Mora Hernández	 Requerimientos No Funcionales	3/6	30%

### Proceso de Desarrollo

Los Requerimientos antes mencionados se fueron implementando conforme se fueron necesitando, aunque algunos tenían una prioridad más alta, estas necesitaban de otras más básicas para poder implementarse.

A nivel individual se investiga como resolver un problema, que funciones se pueden implementar para ello, entre otras, y a nivel grupal se discute la mejor manera de solucionarlo tomando en cuenta el punto de vista de cada quien para llegar a una solución, también se revisa lo que ha hecho cada uno para identificar errores o falta de documentación.