



# PROPUESTA

Programación Estructurada

## PROFESOR:

Emilio Gabriel Rejón Herrera

## Alumnos:

Daniel Gutiérrez Delfín & Raymundo Mora

## Antecedentes

No existen antecedentes de un programa similar, buscando en internet no encontramos ningún resultado, se habló con productores agrícolas y ganaderos y lo que la mayoría dijo es que para saber toda esa información es hablar con un ingeniero agrónomo o por lo que nosotros conocemos por ensayo y error, siendo así el primer programa en este tipo. De igual forma los productores estuvieron interesados en el programa pues por programas como por ejemplo “sembrando vida” se obtiene un apoyo económico y se les sería más sencillo cumplir con el programa sabiendo que plantas pueden producir según el terreno que tienen.

## Descripción

El Programa tendrá 3 opciones: Quiero aprender, El suelo y Mi zona

**Quiero aprender:** Esta opción le hace al usuario varias preguntas con el fin de recaudar información sobre su terreno, después se le preguntará su región en el país y mediante todo esto se le dará el tipo de suelo que tiene para que se utiliza mayormente y que es lo recomendado sembrar (ya sea productos agrícolas o pastura para animales) en su terreno.

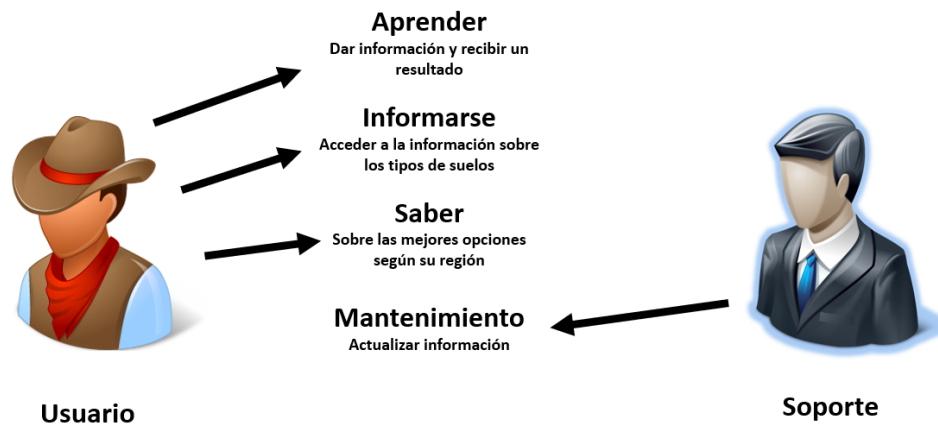
**El suelo:** Esta opción le muestra al usuario los tipos de suelo que existen, según el que seleccione se le brindará la información sobre el mismo.

**Mi zona:** Esta opción le dará al usuario información sobre la zona en la que el indique, no se generalizará todo el espacio en algunas zonas (esto aplica para la primera opción) tomando en cuenta que lugares como Chiapas tiene climas fríos y cálidos.

## Objetivos

- ✚ Orientar a los productores de campo
- ✚ Aumentar el conocimiento agrícola
- ✚ Dar a conocer que producción es la más conveniente según la zona y el tipo de suelo.
- ✚ Reducir en gran medida el problema de ensayo y error.
- ✚ Aumentar las probabilidades de éxito de producción.

## Diagrama de casos de uso

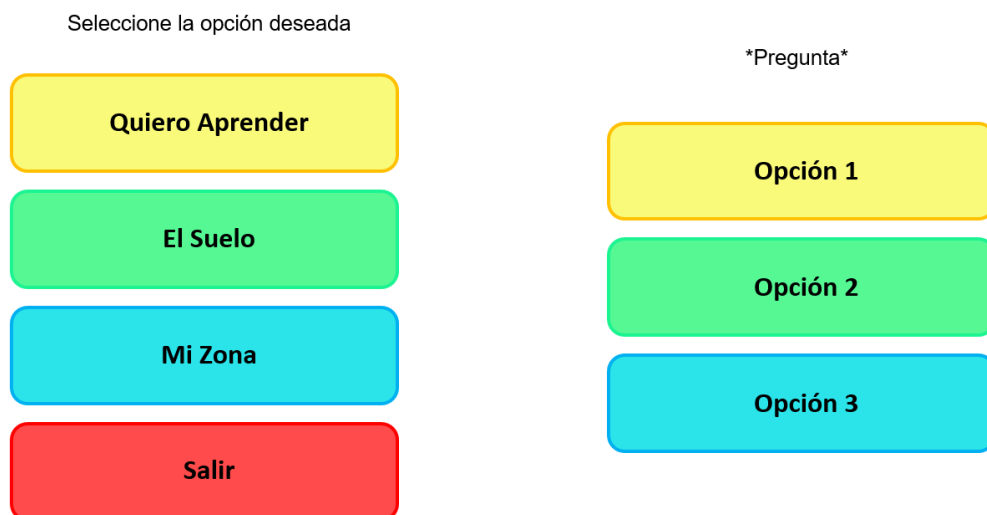


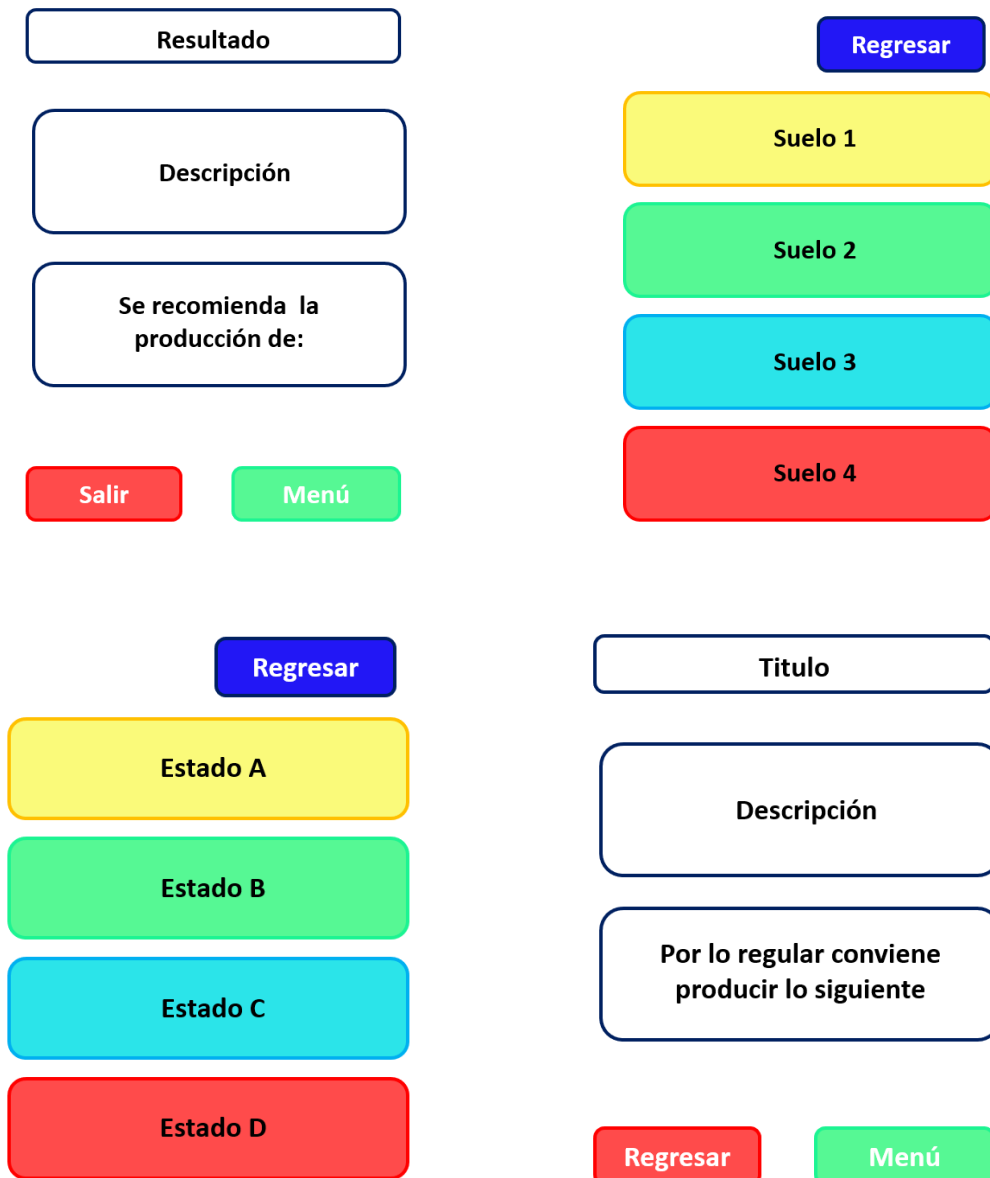
## Tipos de usuario

**Soporte:** Este usuario puede modificar la información y actualizar el programa cuando sea necesario.

**Productor:** Este usuario es capaz de interactuar con el programa en las opciones antes mencionadas ya sea para saber cuales son las mejores opciones para su producción o solo obtener información.

## Principales Interfaces del Usuario





## Requerimientos Funcionales

1. El sistema desplegara un Menú donde le mostrara las opciones.
2. En caso de que ingrese una opción no valida, el programa se quedara estático hasta recibir una respuesta valida.

3. Podrá regresar al Menú en caso de que desee continuar buscando información.
4. El sistema hilará las respuestas con la siguiente pregunta para poder ser mas certero con el resultado.
5. Una vez iniciadas las preguntas no se podrán corregir las respuestas.
6. Se deberá tomar en cuenta la región para dar una respuesta concreta.
7. Todas las informaciones se darán en un formato predeterminado en el código.

## **Requerimientos No funcionales**

1. El Usuario no podrá modificar el Código Fuente.
2. El código solamente está delimitado a la Región de México.
3. Estará dividido en Estados.
4. Las propuestas de producción serán independientes de la temporada.
5. Las propuestas tendrán opciones tanto para agricultores como ganaderos.
6. Las propuestas serán generalizadas, no se tomará muchas variantes de la misma fruta.
7. Se pondrán principalmente propuestas de mayor mercado para los agricultores y de mayor conveniencia para los ganaderos, esto en cuestión de pastura.

## **Casos de uso para cada funcionalidad**

**Datos no validos:** En caso de dar algún dato no valido no pasará nada.

**Preguntas:** Las preguntas irán tomando un camino distinto según las respuestas del usuario para poder llegar a un resultado

**Respuestas:** Las respuestas serán de opción múltiple y serán con respecto a cada pregunta.

**El Suelo:** Se desplegará una lista de los tipos de suelo, al seleccionar alguno se dará la información de este.

**Mi Zona:** Esta función desplegará una lista de las zonas de México con su respectiva información (según lo seleccionado).

**Regresar:** Esta función retrocede un paso según donde se encuentre el usuario.

**Salir:** Esto da fin al programa

## **Estándar de Codificación**

### **Nombrado de Variables y Funciones**

El nombre de las variables debe ser efectivo, de preferencia que no sea trivial, que lleve un nombre simple y práctico para un mayor entendimiento a la hora de utilizarlas.

De igual forma las funciones deben tener un nombre que sea algo distinguible de los demás para que no se confunda de igual manera que sea simple.

### **Archivos, Bibliotecas, etc.**

En cuanto a los archivos y bibliotecas, se espera que la información de los eventos esté guardada en un archivo y por lo pronto se tiene muy en cuenta las funciones de las bibliotecas <string.h> y <stdlib.h> para el buen manejo de la información

### **Bloque de Comentarios**

**Función:** Cada función en deberá tener un comentario que contenga su objetivo y con qué otras funciones se relacionan (si lo hace)

**Apuntadores:** Cada apuntador deberá tener un comentario que diga a donde va a estar apuntando y cuál va a ser su función en el programa.

**Estructuras:** Las estructuras deben tener en sus comentarios para que sirve y como va a ser implementada, además si está estará anidada dentro de otra(s) estructura(s) debe estar documentado también.

**Operaciones:** Las funciones como strcpy, fflush(stdin), malloc, entre otras deben tener comentado para que se están usando.

**Entrada/Salida/Proceso:** De forma general toda entrada, salida, proceso y similares debe estar comentada para saber con qué se está trabajando.

### **Proceso de Desarrollo**

Se piensa en dividir las tareas en 2 categorías básico y vital, siendo básico cuestiones de menos y cosas no muy complicada y vital cosas que ya requieran

mucha más investigación y tiempo de codificación ya sea por tedioso o por complejo. Estas tareas se piensan poner en un orden para que el programa vaya creciendo de manera adecuada, para eso se piensa usar la herramienta Trello para ponerlas de manera organizada.

De igual forma se dejará el nombre de la persona que hizo la tarea, si ya está listo o en proceso, si tiene algún detalle o falta algo, cuando fue iniciada o terminada entre otras cosas.

Se espera que se hagan 18 horas a la semana de trabajo, este tiempo algo extendido no es nada más para ver conflictos en el código, investigar métodos o funciones y ver que todo esté bien documentado y marche bien.