Programación Estructurada

Alumnos: Daniel Gutiérrez y Raymundo Mora

Profesor: Emilio Gabriel Rejón Herrera

202

Avance de proyecto



**Antecedentes del sistema**

La Agricultura a sido una actividad fundamental para el ser humano. Este inicio como una actividad necesaria para la supervivencia, después con el paso del tiempo se volvió también una actividad de comercio y hoy es algo indispensable el desarrollo de cualquier país, estado o región, ya que esta además de la alimentación, es parte de la economía de familias, estados y países.

La tecnología avanza muy rápido y se implementan siempre nuevas tecnologías en áreas de salud, de entretenimiento, entre otras y se ha vuelto algo tan común y parte de nosotros que ya es de lo más normal usar tecnología para facilitarnos la vida. Sin embrago en zonas rurales esto es muy difícil que pase, entonces los agricultores no tienen casi avances para saber tomar decisiones que les den más beneficios. Pensando en esto surgió la idea de un programa que ayude a los agricultores a orientar su producción según las condiciones en las que se encuentra.

**Cambios en la Sociedad**

No cabe duda de que la agricultura es algo que cambio el curso de la humanidad en su totalidad, desde el inicio del hombre sedentario, pasando por el trueque de alimentos por otro tipo de cosas, hasta la importación y exportación de alimentos entre países que mueve gigantescas cantidades de dinero en él mercado.

**Descripción del Producto Software**

Con los conocimiento adquiridos en el lenguaje C además de la investigación sobre las características de los diferentes tipos de suelo, se da a luz al un programa que orienta a los agricultores para no tener que recurrir al ya conocido ensayo y error en ese ámbito, este programa por medio de preguntas simples podrá dar un resultado aproximado sobre el tipo de suelo que tiene el agricultor, además este al dar el estado donde se encuentra, el programa dará unas recomendaciones de cultivos con base a las características del suelo y las de la zona, esta no necesitará conexión a internet ya que el programa contiene la información, a su vez el programa tiene dos opciones adicionales, “Los suelos” le da al usuario la opción de saber sobre los tipos de suelos y “Mi Zona” le da al usuario la información productora de cualquiera de los 32 estados de la Republica Mexicana.

**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 especifico 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 optimo 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 él 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.

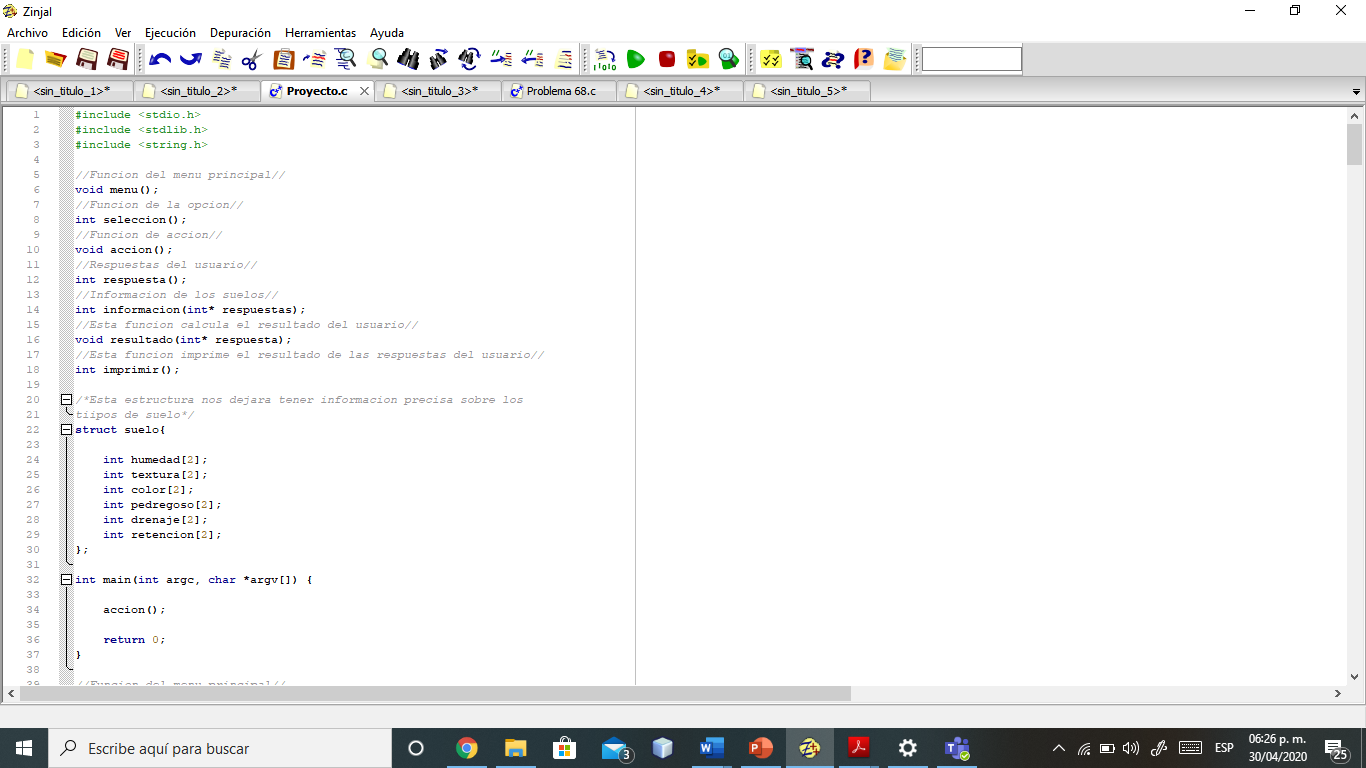
Los requerimientos no funciones cambiaron casi en su totalidad por ser básicamente irrelevantes y en algunos casos erróneos y los actuales son más útiles y se adatan mejor a lo que se necesita a la hora de hacer el programa.

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.

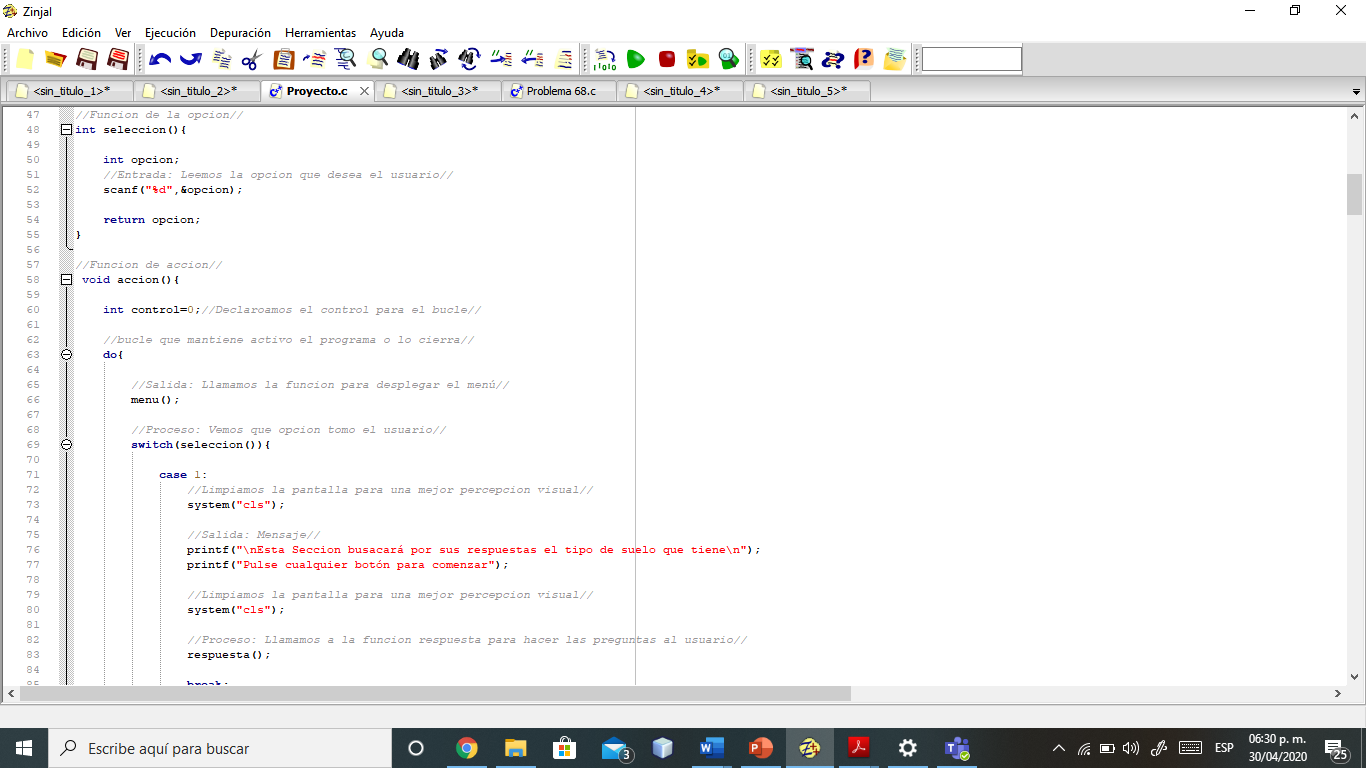
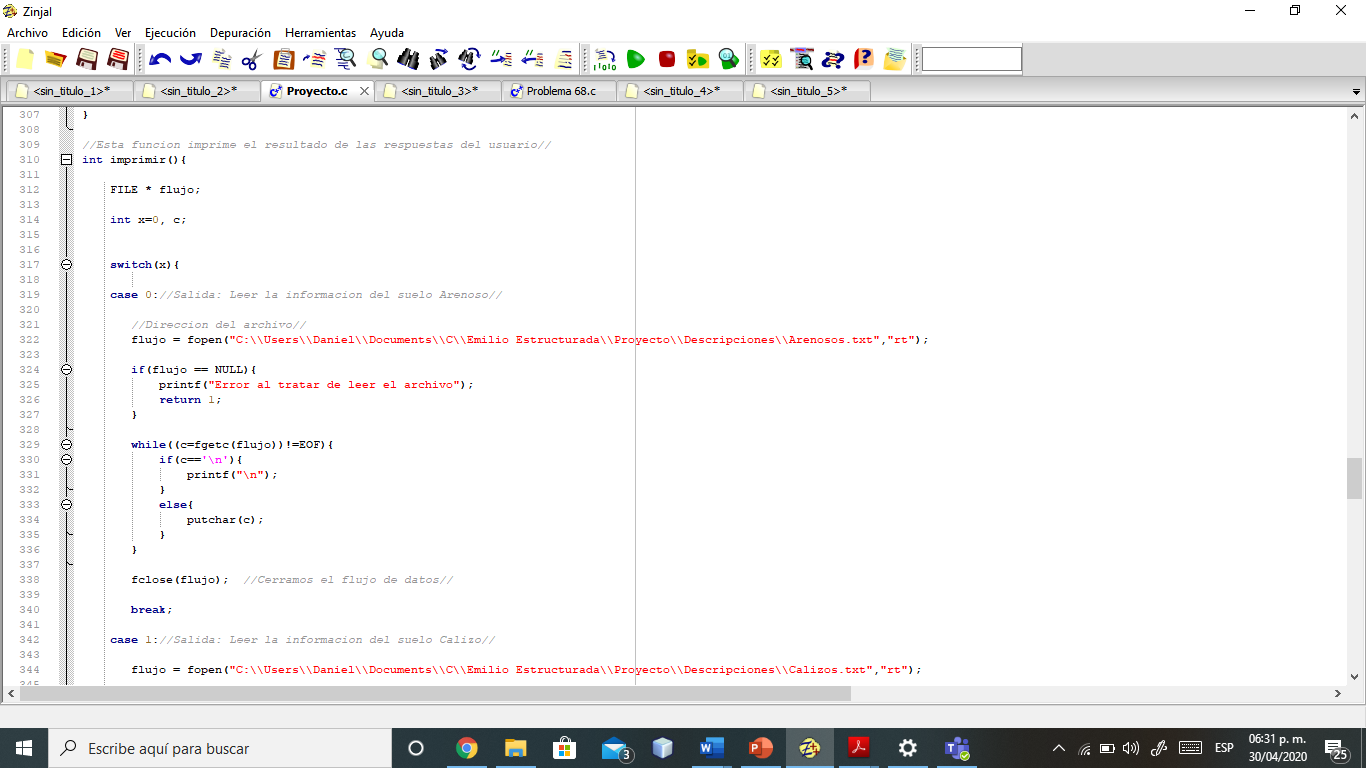
En general el programa va bien, los errores se han podido resolver con eficacia y por el momento el único detalle es el tiempo, ya que la aceptación de la propuesta y las circunstancias pandémicas nos han dejado un poco cortos de tiempo, sin embargo se aumentaran un par de sesiones extra para compensar 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.

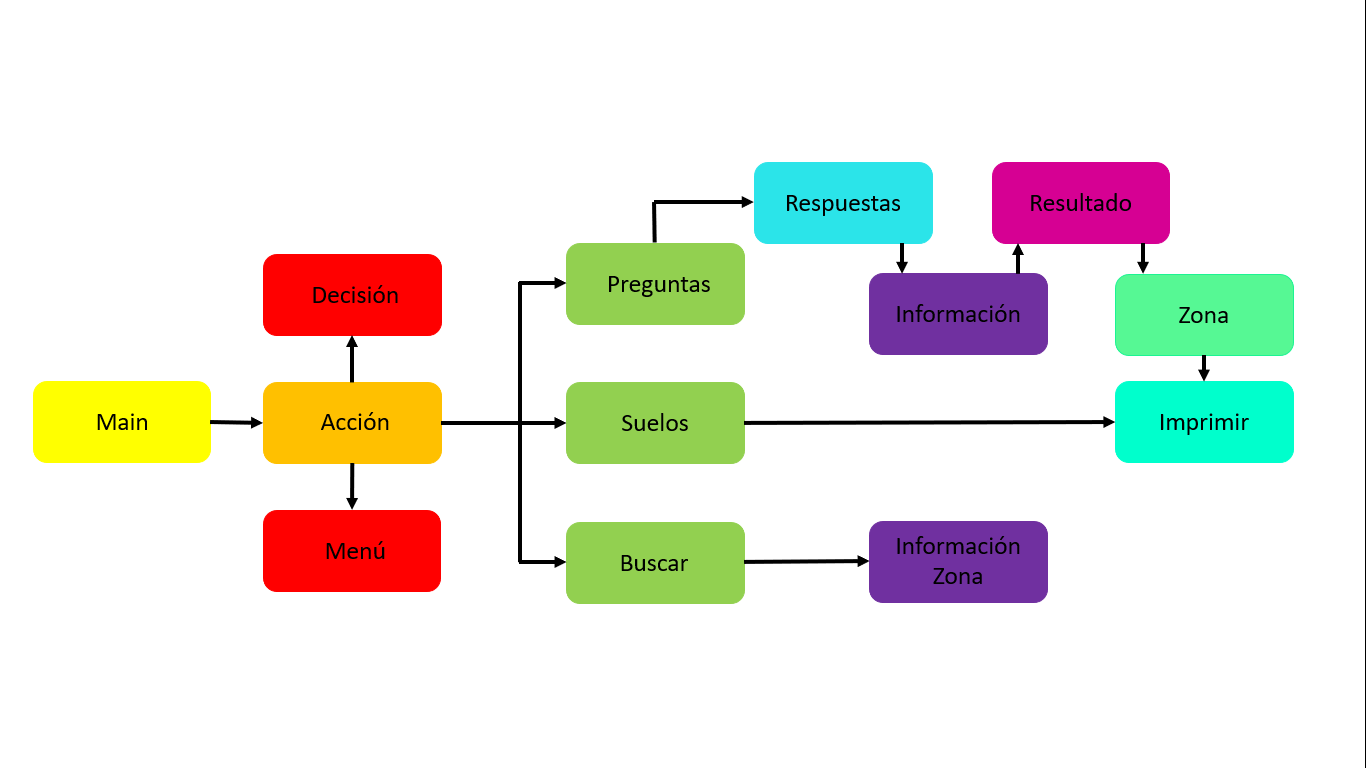


* No se implementaron macros
* Las bibliotecas utilizadas hasta el momento son: <stdio.h>, <stdlib.h>, <string.h>
* La cuestión de entrada, proceso y salida, están repartidas por funciones, en el código hay funciones que sirven como entrada de datos para pasarlos a funciones de proceso y por último pasar a funciones de salida.

* 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** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **S11** |
| 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 |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Contribución** | **Asistencia** | **Porcentaje** |
| Daniel Eduardo Gutiérrez Delfín | * Idea principal * Proceso de desarrollo * Casos de Uso * Requerimientos Funcionales | 11/11 | 60% |
| Raymundo Mora Hernández | * Requerimientos No Funcionales | 6/11 | 40% |

**Reporte Individual (Delfín)**

* Sesión 1: Propuso la idea del proyecto
* Sesión 1: Investigo sobre programas similares en distintas plataformas.
* Sesión 2: Propuso las bases de la estructura del código
* Sesión 2: Estableció la información que se iba a requerir
* Sesión 3: Inicio la codificación del programa
* Sesión 4: Comenzó a investigar métodos para el pase de datos de estructuras
* Sesión 5: Se empezó a implementar la validación de valores dadas por el usuario
* Sesión 6: Dio valores a la información de los suelos
* Sesión 7: Inicio pruebas para el uso de getch junto con el código ascii y el pase de valores de estructuras por referencias
* Sesión 8: Redacto el avance del proyecto
* Sesión 9: Terminó el análisis de respuestas para obtener un resultado y comenzó el proceso de sugerencias.
* Sesión 10: Termino la función “Quiero Aprender” y valido que la lectura de cadenas de la función “Mi Zona” fueran nombres de los estados de la república.
* Sesión 11: Implementó la función getch y que el programa siga funcionando las veces que el usuario quiera hasta que le de salir
* Sesión 11: Busco y resolvió errores en el código, en datos de suelos, rutas alternativas y en la conversión de valores char a ascii y viceversa.

**Reporte Individual (Mora)**

* Sesión 1: Complementó la idea del proyecto
* Sesión 1: Investigo sobre programas similares en distintas plataformas.
* Sesión 2: Propuso mejoras en estructura del codigo
* Sesión 2: Investigó los principales suelos que se iba a usar
* Sesión 5: Inicio la codificación de la función “Los Suelos”
* Sesión 7: Termino la función “Los Suelos” y comenzó a codificación de la función “Mi Zona”
* Sesión 9: Investigo la información de los 32 estados de la republica y cada uno en su respectivo archivo .txt
* Sesión 10: Arregló problemas en orden de datos lo que causaba que se dieran resultados incorrectos.

**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.