

**“UNIVERSIDAD TECMILENIO”  
CERTIFICADO EN FUNDAMENTOS DE LA  
PROGRAMACIÓN**



**FUNDAMENTOS DE LA PROGRAMACIÓN**

**- AVANCE DEL PROYECTO: -  
VEX ROBOTICS: GESTOR DE PROYECTOS**

**Primer Semestre**

**INTEGRANTES:**

**Victor Andres Hernandez Castillo - SENIOR**

**David Durán Rubio - MID**

**Edwin Omar Perez Estrada - MID**

**Blanca Dennis Soto Rubio - JUNIOR**

**Nicolas Flores Verdad Quintana - JUNIOR**

# INTRODUCCIÓN

Lo que buscamos en este proyecto es darle solución a una problemática que nos afecte de manera directa o indirecta, en nuestro caso presentamos una herramienta de apoyo para el taller del TecMilenio “VEX: ROBOTICS”. Dicha herramienta tendrá varias funciones como apoyo a los estudiantes de dicho taller, pues ese es su principal objetivo es ayudar a los alumnos a lograr con mayor facilidad sus objetivos y que cuenten con una mayor organización.

## CONTEXTO Y PROBLEMA

Uno de los integrantes de equipo pertenece a VEX, por lo tanto él considera que las principales flaquezas o problemas que tiene el taller, es la falta de organización, que no se conoce el material que se tiene para el trabajo y que las ideas no están bien aterrizadas, ya que muchas veces no sabemos las actividades pendientes que tenemos que realizar.

Separaremos nuestro programa en cuatro secciones, pues consideramos que contamos con cuatro grandes problemas:

### 1.- Lista de integrantes por sector:

Muchas de las veces, debido a la mala organización, no sabemos a qué sector pertenece cada integrante del equipo. VEX se divide en 3 principales ramas de trabajo que son: *Administración, Mecánica y Programación*; Por ello vamos crear una lista de integrantes, para checar a que paquete pertenece cada integrante y tener de manera más visual los equipos.

MATRICULA	NOMBRE	ROL	SEMESTRE
AL07165261	David Duran Rubio	Mecánica	Primer semestre

*Este es un ejemplo sencillo de lo que buscamos lograr en la lista de integrantes.*

### 2.- Inventario/Materiales:

En VEX el material que se tiene para trabajar es muy limitado o en varias ocasiones no se sabe con cuanto material se cuenta debido a que no hay un registro concreto, por lo tanto en nuestro programa buscamos tener el conteo del material disponible para poder llevar a cabo sus proyectos. Este registro se podría modificar para poder quitar o agregar material según corresponda.

```
INVENTARIO:  
Tornillos: 115/200  
Tuercas: 110/200  
Motores: 2/2
```

*Buscamos un resultado que cumpla el mismo propósito, pero con una mejor calidad al de la imagen.*

### 3.- Lista de pendientes:

Consideramos que una manera más sencilla de lograr los objetivos del taller es especificar y listar las cosas por hacer, la mayoría de las veces cuando los integrantes de VEX se presentan para trabajar, no tienen claro que van a hacer, por eso crear una lista con los objetivos haría que el proceso de trabajo sea más veloz.

```
--- LISTA DE PENDIENTES ---  
1. Revisar el diseño del chasis del robot - Estado: completado  
2. Calibrar los sensores de distancia - Estado: pendiente  
3. Optimizar el código para el movimiento autónomo - Estado: pendiente  
4. Preparar el informe del proyecto - Estado: pendiente
```

*Esta es la idea de la manera en que buscamos mostrar las tareas pendientes.*

### 4.- Barra de progreso:

Consideramos que una manera algo creativa y que motive a nuestros integrantes de VEX es agregando una barra de progreso al código que muestre cuánto falta de los pendientes a realizar.

```
PROGRESO DEL PROYECTO: 25%  
[██████████] ]
```

*Esta barra de progreso estaría enlazada a la lista de actividades pendientes, para saber como va el avance de tareas en el equipo.*

Las soluciones que mostramos, son para mejorar o eliminar las problemáticas que tiene VEX y buscar un mejor desarrollo como equipo, también buscamos que no solo se quede para esta generación de VEX, sino que perdure para las siguientes generaciones y sea una herramienta de ayuda, la cual puede seguir actualizando y mejorando.

## REGLAS DE NEGOCIO

Al iniciar el programa mostrará un menú el cual se divide en las cuatro secciones que mencionamos con anterioridad:

1. Lista de integrantes por sector
2. Inventario/Materiales
3. Lista de pendientes
4. Barra de progreso

Estas estarán listadas por números como se mostró anteriormente, para preguntar a qué sección desea ir, después cada sección tendrá un menú propio. *(Todo este proceso se logrará con una variable “opcion” y un input(), el input() pedirá el número de la sección y la variable guardará para después con un if mandarlo a la sección que corresponde).*

Ahora explicaremos a mayor detalle las secciones, comenzando por la “lista de integrantes por sector”:

Aquí también mostrará un menú pero ahora con dos secciones:

1. Mostrar lista
  - a. General
  - b. Administración
  - c. Mecánica
  - d. Programación
2. Agregar integrante

En la primera sección “Mostrar lista” te preguntará si quieres ver una lista general, o de algún sector en específico. *(Aquí esta parte será muy parecida al menú de inicio, pues cada número te llevará a la sección correspondiente)*

La segunda sección te hará añadir a alguien (Con input() pedirá los datos correspondientes), te solicitará su matrícula, nombre, rol que cumple en el proyecto y el semestre al que pertenece. Por último te preguntará si quieres guardar el registro, si la respuesta es “no” o “n” lo elimina y se repite el ciclo (utilizamos while), si tu respuesta es “si” o “s” se guarda el registro y te regresa a la sección de “lista de integrantes por sector”.

En la segunda sección del menú principal “Inventario/Materiales” se mostrará otro menú el cual contará con las siguientes opciones:

1. Mostrar material: Solo mostrará la lista de materiales con su nombre y cantidad, ejemplo: Tornillos de 6/8 → 26/30.
2. Modificar material: Aquí saldrá una lista de todos los materiales con los que se cuentan, estos están numerados, después un input() te preguntará qué material quieres modificar y lo seleccionamos poniendo el número del material, ya seleccionado podemos modificar la cantidad de material con la que se cuenta.

La parte tres del menú es la “lista de pendientes” esta parte del programa será muy similar a la sección anterior, serán dos opciones:

1. Mostrar pendientes:

Aquí mostrará las actividades que hay pendientes por realizar, el estado en el que se encuentra (La tarea puede estar completa, o pendiente).

2. Modificar lista de pendiente:

Aquí será muy similar a la sección anterior, pues mostrará una lista numerada de las actividades por realizar, un input() preguntará qué actividad desea cambiar su estado (los estados son: completo o pendiente).

Por último contamos con la “barra de progreso” esta al ser seleccionada mostrará cómo va el proceso de tus actividades pendientes, por lo tanto la sección tres y cuatro están conectadas, aquí la manera en la que funciona el código es que el número total de tareas pendientes, se dividirá entre el número de actividades de estado completo, el número que nos de será multiplicado por 100, para poder ir printeando la barra de tareas.

## REQUERIMIENTOS FUNCIONALES

El código tiene que cumplir con una cantidad de requerimientos funcionales, en especial los siguientes:

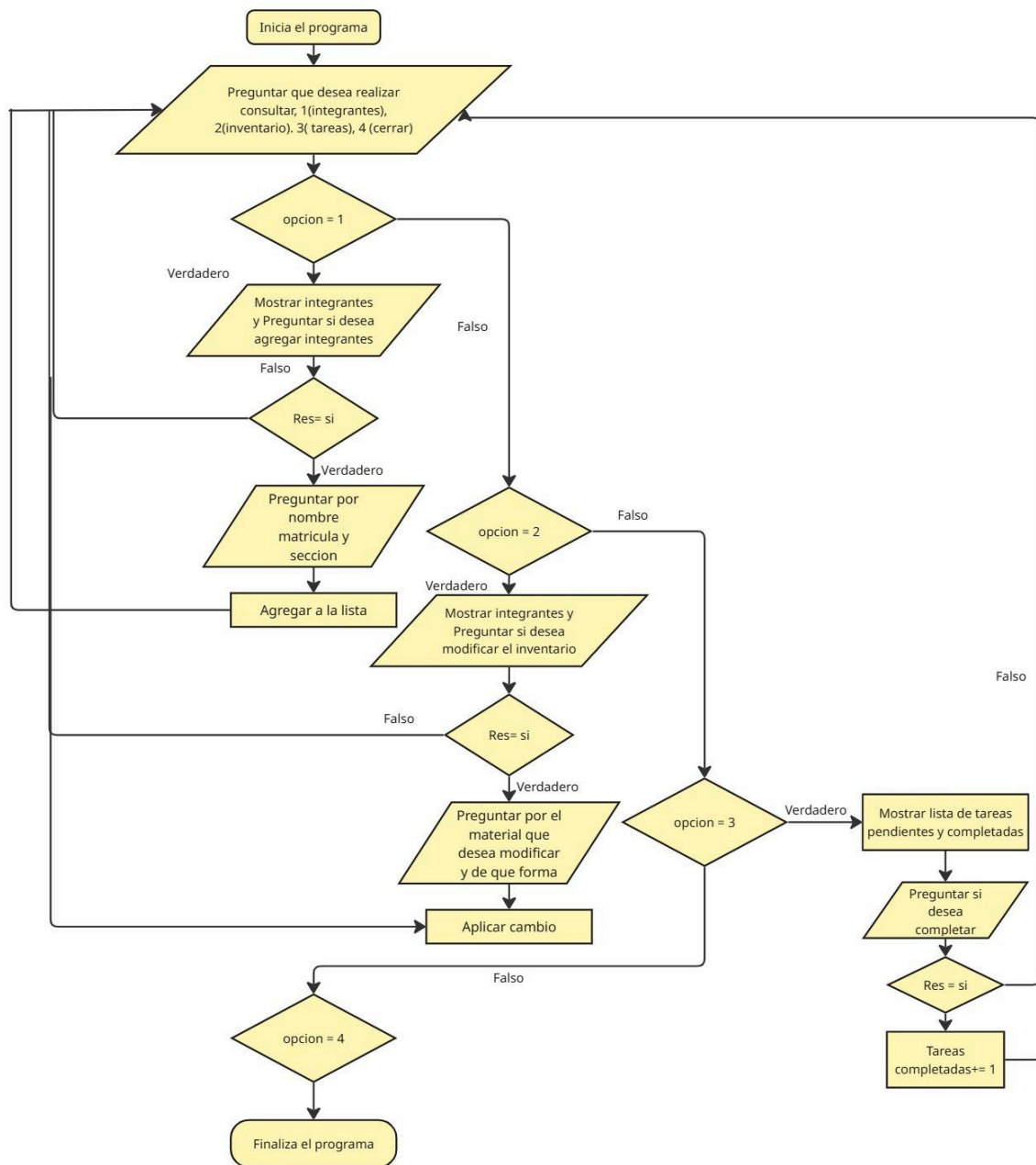
1. Menú: Que tenga un menú ordenado y claro, para que sea entendido por los usuarios que la utilicen, también que dicho menú tenga la facilidad de moverse entre las secciones, osea que no haya dificultad de cambiar de sección en sección.
2. Almacenamiento de integrantes: De las principales funciones, y la más repetida para nuestro código, es el almacenamiento de datos, en este caso el almacenamiento de los integrantes de equipo.
3. Filtro de datos: Así como logra almacenar los datos, consideramos que una manera clara y más sencilla de navegar y poder entenderlos, es con filtros, estos filtros acomodan a los integrantes de equipo según su rol.
4. Almacenamiento de material: Este funciona muy parecido al almacenamiento de integrantes, pues buscamos que nuestro código logre almacenar el tipo de material y la cantidad disponibles para trabajar. También buscamos que dichas cantidades se puedan modificar para saber si se encuentran en uso o no.
5. Lista de pendientes: Aquí buscamos que el programa almacene las actividades que se tienen que realizar y el estado en el que se encuentran, esto del estado es principalmente para poder llevar a cabo la barra de progreso.
6. Barra de progreso: Que muestre en qué porcentaje se encuentran del proyecto, esto es principalmente para agregar una forma más visual de las actividades a realizar.

## DISEÑO DE ENTRADAS Y SALIDAS

ENTRADAS	SALIDAS
Número de la sección que desea visitar.	Te lleva al menú seleccionado y muestra el menú de dicha sección.
En la lista de integrantes, el usuario decide cual es la vista en la que quiere ver la lista (General, Administración, Mecánica, Programación)	Muestra la tabla de integrantes como se la pediste.
Al registrar un nuevo integrante, te pide los datos necesarios (Matrícula, nombre, rol, semestre). Estos datos los almacena para después mostrarlos	Muestra en la tabla de integrantes los datos que fueron anteriormente solicitados.

En el inventario te pide que si quieres ver la lista o modificarla.	Según sea la selección mostrará la lista de material o mostrará una opción para modificar la lista.
Al modificar la lista te pide el tipo de material y la cantidad con la que se cuenta.	Esta información que se solicitó, se mostrará después en la tabla de materiales.
En la lista de pendientes, tu añades la tarea a realizar.	La tarea que vas a realizar se muestra en la tabla de lista de pendientes.
Al modificar la tabla de actividades pendientes, el usuario puede modificar el estado de la tarea.	Ese estado se guarda y se muestra en la lista de actividades pendientes.
En la barra de progreso los valores de entrada son los mismos que los de la lista de pendientes, pues calcula el porcentaje según el número de tareas pendientes y completadas.	Muestra el resultado del cálculo de porcentaje de tareas pendientes, tanto como porcentaje como barra de progreso de manera visual.

# DIAGRAMA DE FLUJO



## PLAN DE TRABAJO

Aquí mostraremos las tareas que iremos realizando durante el periodo de trabajo del proyecto.

*Miércoles 27/08/2025:*

Este día el docente de la clase “Fundamentos de la Programación” nos dio una clase de como hacer uso del GitHub, pues será nuestra principal herramienta para trabajar en este proyecto. También lo que hicimos fue delimitar la idea para llevar a cabo.

*Jueves 28/08/2025:*

Empezamos con el trabajo del proyecto.

Victor (SENIOR) se encargó de la creación del repositorio de GitHub y agregar las carpetas de cada archivo correspondiente.

Nicolas (JUNIOR) se encargó de la creación del esqueleto.

David (MID) se encargó de la redacción del documento.

Dennis (JUNIOR) se encargó de checar y cambiar las correcciones del esqueleto

*Viernes 29/08/2025:*

Este día está enfocado para mandar el primer avance del proyecto, que es un documento que cuente con los siguientes requisitos:

- Portada
- Contexto y problema
- Reglas de negocio
- Requerimientos funcionales
- Diseño de entradas y salidas
- Diagrama de flujo
- Plan de trabajo
- Asignación de roles

De eso se encargó David (MID)

Victor (SENIOR) se encargó del diagrama de flujo.

Dennis (JUNIOR) se encarga de terminar el esqueleto del código.

*Semana del 01/09/2025 al 05/09/2025:*

Veremos los temas del 9 al 12, gracias a estos temas vamos a ver qué información nos puede ayudar a poder mejorar la aplicación e implementarla.

Los integrantes de MID (Edwin y David) se encargaran de seguir con el desarrollo del código, los JUNIOR (Dennis y Nicolas) se encargaran del testeo del código, mientras el SENIOR (Victor) verifica que el trabajo que se está haciendo se esté realizando de manera adecuada y está en los repositorios correspondientes.

*Semana del 08/09/2025 al 12/09/2025:*

En estas semanas veremos los temas del 13 al 16, con la información de estos temas también los integrantes de MID (Edwin y David) mejoraran el código con los temas vistos durante las clases, los JUNIOR (Dennis y Nicolas) se encargaran del testeo del código y recalcar los errores que hay en el código, mientras el SENIOR (Victor) hará las correcciones que sean necesarias y que ya nos han marcado los JUNIOR