

Proyecto 1-Entrega 1
ESTRUCTURAS DE DATOS



Estudiantes:

Laura Valentina Ovalle Benítez

Valentina García Alfonso

Juan Miguel Zuluaga Suárez

Carlos Gabriel López

Nicolás Padilla Medina

PRESENTADO A:

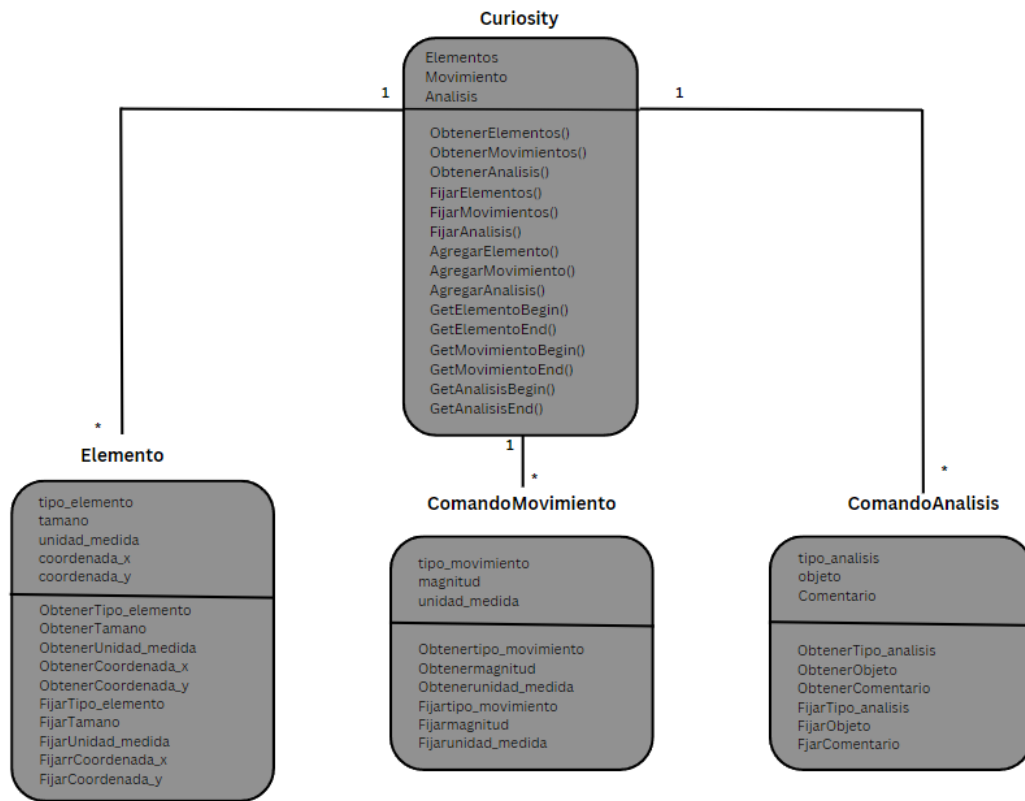
John Jairo Corredor Franco

PONTIFICIA UNIVERSIDAD JAVERIANA

BOGOTÁ D.C

2023

1. Declaración de TADS:



Curiosity

Datos mínimos:

- Elementos, lista de elementos, permite identificar la complejidad del terreno sobre el cual debe moverse el vehículo
- listaMovimientos, lista de comandos de movimiento, permite al robot desplazarse sobre la superficie de Marte
- listaAnalisis, lista de comandos de análisis, permiten al robot investigar la superficie de Marte para analizar los elementos que va encontrando en su desplazamiento

Operaciones:

- ObtenerElementos(), retorna la lista de elementos actual que se encuentran en el suelo que recorre el Curiosity
- ObtenerMovimientos(), retorna la lista de comandos de movimiento actual que debe realizar el Curiosity
- OtenerAnalisis(), retorna la lista de comandos de análisis actual que debe realizar el Curiosity
- FijarElementos(), fija una nueva lista de elementos para el suelo que recorre el Curiosity

- FijarMovimientos(), fija una nueva lista de comandos de movimiento que debe realizar el Curiosity
- FijarAnalisis(), fija una nueva lista de comando de análisis que debe realizar el Curiosity
- AgregarElemento(), agrega un elemento a la lista de elementos
- AgregarMovimiento(), agrega un movimiento a la lista de movimientos
- AgregarAnalisis(), agrega un análisis a la lista de análisis
- GetElementoBegin(), retorna el inicio de la lista de elementos
- GetElementoEnd(), retorna el fin de la lista de elementos
- GetMovimientoBegin(), retorna el inicio de la lista de movimientos
- GetMovimientoEnd(), retorna el fin de la lista de movimientos
- GetAnalisisBegin(), retorna el inicio de la lista de análisis
- GetAnalisisEnd(), retorna el fin de la lista de análisis

Elemento

Datos mínimos:

- Tipo_elemento, cadena de caracteres, describe el tipo del elemento que fue hallado por el Curiosity en el terreno. Puede ser roca, cráter, montículo o duna.
- Tamaño, entero, es el valor de la dimensión del elemento
- Unidad_medida, cadena de caracteres, define la unidad con la que se realizó la medición del tamaño del elemento.
- Coordenada_x, decimal, indica la posición del elemento sobre el eje x en el plano cartesiano
- Coordenada_y, decimal, indica la posición del elemento sobre el eje y en el plano cartesiano

Operaciones:

- ObtenerTipo_elemento(), retorna el tipo actual que tiene el elemento
- ObtenerTamaño(), retorna el tamaño actual que tiene el elemento
- ObtenerUnidad_medida(), retorna la unidad de medida actual con la que se realizó la medición del elemento
- ObtenerCoordenada_x(), retorna la coordenada actual del elemento en el eje x del plano cartesiano
- ObtenerCoordenada_y(), retorna la coordenada actual del elemento en el eje y del plano cartesiano
- FijarTipo_elemento(), fija un nuevo tipo para el elemento
- FijarTamaño(), fija un nuevo tamaño para el elemento
- FijarUnidad_medida(), fija una nueva unidad de medida para el elemento
- FijarCoordenada_x(), fija una nueva coordenada en el eje x para el elemento
- FijarCoordenada_y(), fija una nueva coordenada en el eje y para el elemento

Movimiento

Datos mínimos:

- Tipo_movimiento, cadena de caracteres, le indica al Curiosity como debe moverse sobre el cielo marciano. Puede ser avanzar o girar.
- Magnitud, de tipo decimal, indica el valor o cantidad del movimiento
- Unidad_medida, entero, es la unidad con la que se hace la medición del movimiento

Operaciones:

- ObtenerTipo_movimiento(), retorna el tipo de movimiento actual que tiene el comando
- ObtenerMagnitud(), retorna el valor o cantidad de movimiento actual que indica el comando
- ObtenerUnidad_medida(), retorna la unidad actual con la que se hizo la medición del movimiento
- FijarTipo_movimiento(), fija un nuevo valor para el tipo de movimiento que indica el comando
- FijarMagnitud(), fija un nuevo valor o cantidad de movimiento para el comando
- FijarUnidad_medida(), fija una nueva unidad con la que se hace la medición del movimiento

Análisis

Datos mínimos:

- Tipo_analisis, cadena de caracteres, le indica al Curiosity como debe analizar un elemento. Puede ser fotografiar, composición o perforar
- Objeto, cadena de caracteres, es el nombre del elemento sobre el cual se hace el análisis
- Comentario, cadena de caracteres, corresponde a información adicional sobre el análisis

Operaciones:

- ObtenerTipo_analisis(), retorna el tipo de análisis actual que debe realizar el Curiosity a un elemento de acuerdo al comando
- ObtenerObjeto(), retorna el nombre actual del elemento sobre el cual se hace el análisis
- ObtenerComentario(), retorna la información adicional actual sobre el análisis
- FijarTipo_analisis(), fija un nuevo tipo de análisis que debe realizar el Curiosity a un elemento
- FijarObjeto(), fija un nuevo tipo de elemento sobre el cual se debe hacer el análisis
- FijarComentario(), fija nueva información adicional para el análisis de un elemento

2. Descripción de entradas, salidas y condiciones:

- Procedimiento principal:

Entradas	Comando ayuda tipo string, lista de todos los comandos de movimiento que se desean realizar tipo list<movimiento>, lista de comandos de análisis que se desean realizar tipo list<analisis>, lista de los elementos de tipo list<elemento>
Salidas	Si se ingresa comando erróneo: Comando no reconocido.

	Si se ingresa comando correcto: Ejecuta comando.
Condiciones	El comando debe existir, sus parametros deben ser aquellos aceptados por el mismo. El número de parametros debe coincidir con los requeridos por el comando.

- Comando “cargar_comandos”:

Entradas	Comando “cargar_comandos” tipo string, nombre correspondiente del archivo (nombre_archivo)(string)
Salidas	(Archivo vacio) nombre_archivo no contiene comandos (Archivo erróneo) nombre_archivo no se encuentra o no puede leerse (Resultado exitoso) n comandos cargados correctamente desde nombre_archivo
Condiciones	El archivo debe ser válido, debe poder leerse

- Comando “cargar_elementos”

Entradas	Comando “cargar_elementos” tipo string, nombre correspondiente del archivo (nombre_archivo)(string)
Salidas	(Archivo vacio) nombre_archivo no contiene comandos (Archivo erróneo) nombre_archivo no se encuentra o no puede leerse (Resultado exitoso) n elementos cargados correctamente desde nombre_archivo
Condiciones	El archivo debe ser válido, debe poder leerse

- Comando “agregar_movimiento”:

Entradas	Comando “agregar_movimiento” tipo string, dato “tipo_movimiento” de tipo string, dato “magnitud” tipo float, dato “unidad medida” tipo string
Salidas	(Formato erróneo) La información del movimiento no corresponde a los datos esperados (tipo(string), magnitud(float), unidad(string)). (Resultado exitoso) El comando de movimiento ha sido agregado correctamente
Condiciones	Se debe cumplir con el formato del comando

- Comando “agregar_analisis”:

Entradas	Comando “agregar_analisis” tipo string, dato “tipo_analisis” de tipo string, dato “objeto” tipo string, dato “comentario” tipo string
----------	---

Salidas	(Formato erróneo) La información del movimiento no corresponde a los datos esperados (tipo (string), objeto (string), comentario (string)). (Resultado exitoso) El comando de analisis ha sido agregado correctamente
Condiciones	Se debe cumplir con el formato del comando

- Comando “agregar_elemento”:

Entradas	Comando “agregar_elemento” tipo string, dato “tipo_elemento” de tipo string, dato “tamaño” tipo int, dato “unidad_medida” tipo string, dato “coordenada_x” tipo double, dato “coordenada_y” tipo double
Salidas	(Formato erróneo) La información del movimiento no corresponde a los datos esperados (tipo (string), tamaño (int), unidad (string), x (double), y (double)) (Resultado exitoso) El elemento ha sido agregado correctamente
Condiciones	Se debe cumplir con el formato del comando

- Comando “guardar”:

Entradas	Comando guardar tipo string, dato “tipo_archivo” tipo string, dato “nombre_archivo” tipo string
Salidas	(No hay información) La información requerida no está almacenada en memoria. (Escritura exitosa) La información ha sido guardada en nombre_archivo . (Problemas en archivo) Error guardando en nombre_archivo .
Condiciones	Se debe cumplir con el formato del comando

- Comando “simular_comandos”:

Entradas	Comando simular_comandos” tipo string, dato “coordenada_x” tipo double, dato “coordenada_y” tipo double
Salidas	(No hay información) La información requerida no está almacenada en memoria. (Resultado exitoso) La simulación de los comandos, a partir de la posición (coordX,coordY), deja al robot en la nueva posición (nuevoX ,nuevoY).
Condiciones	Se debe cumplir con el formato del comando

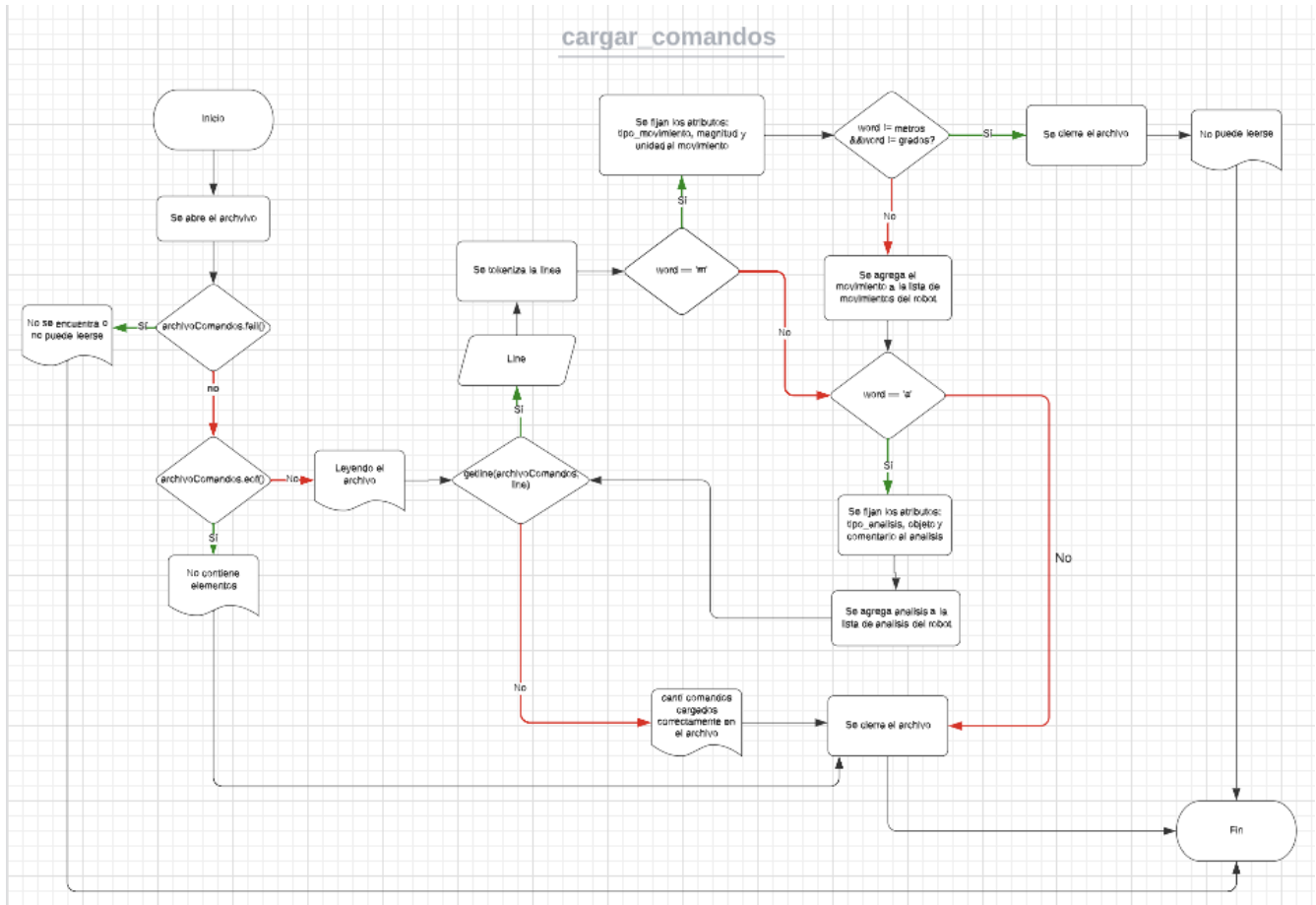
- Comando “salir”:

Entradas	Comando salir tipo string
Salidas	Sin salida por pantalla

Condiciones	No debe recibir ningún otro parámetro más allá del comando base
-------------	---

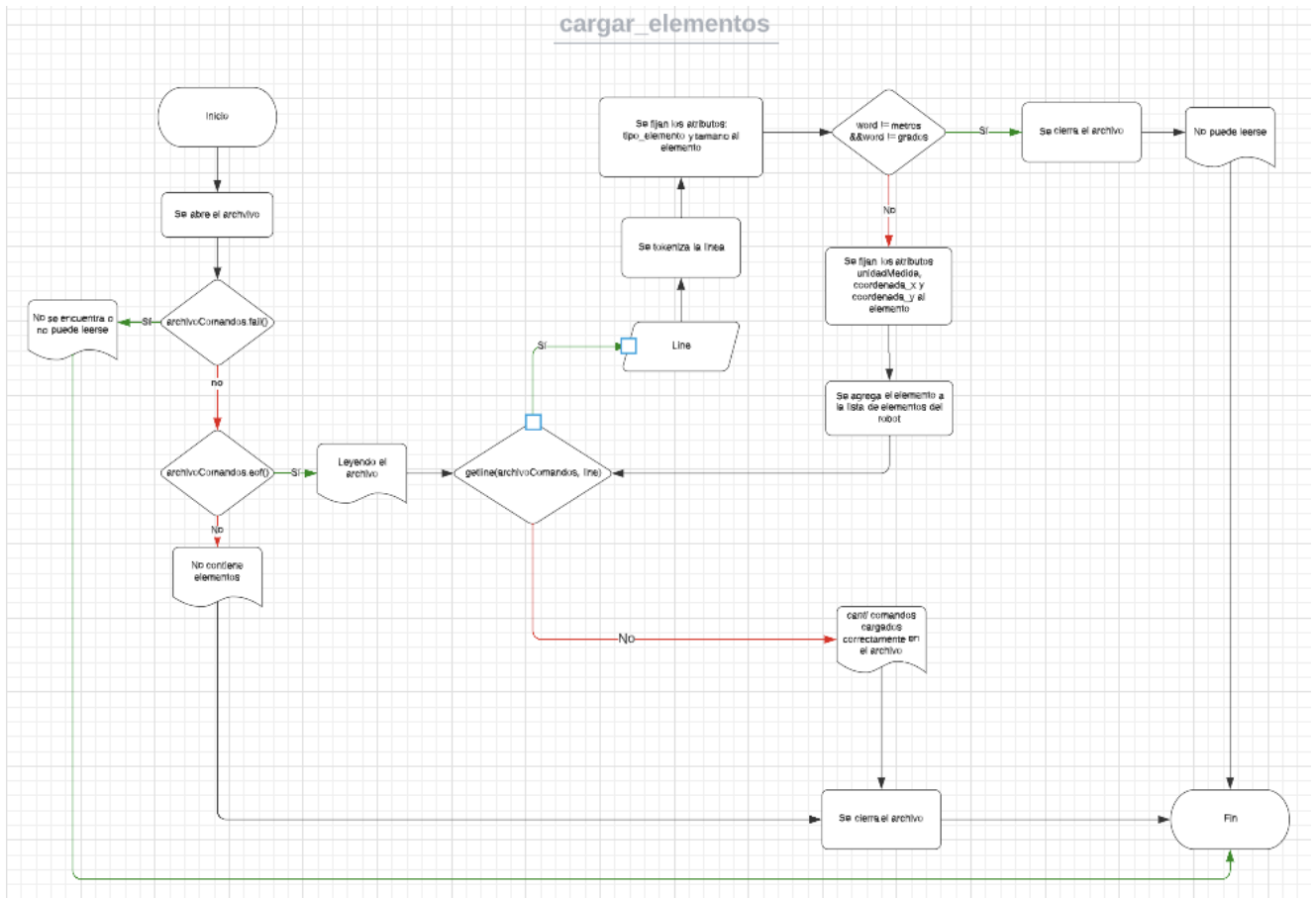
3. Diagramas de Flujo

- Función “cargar_comandos”:



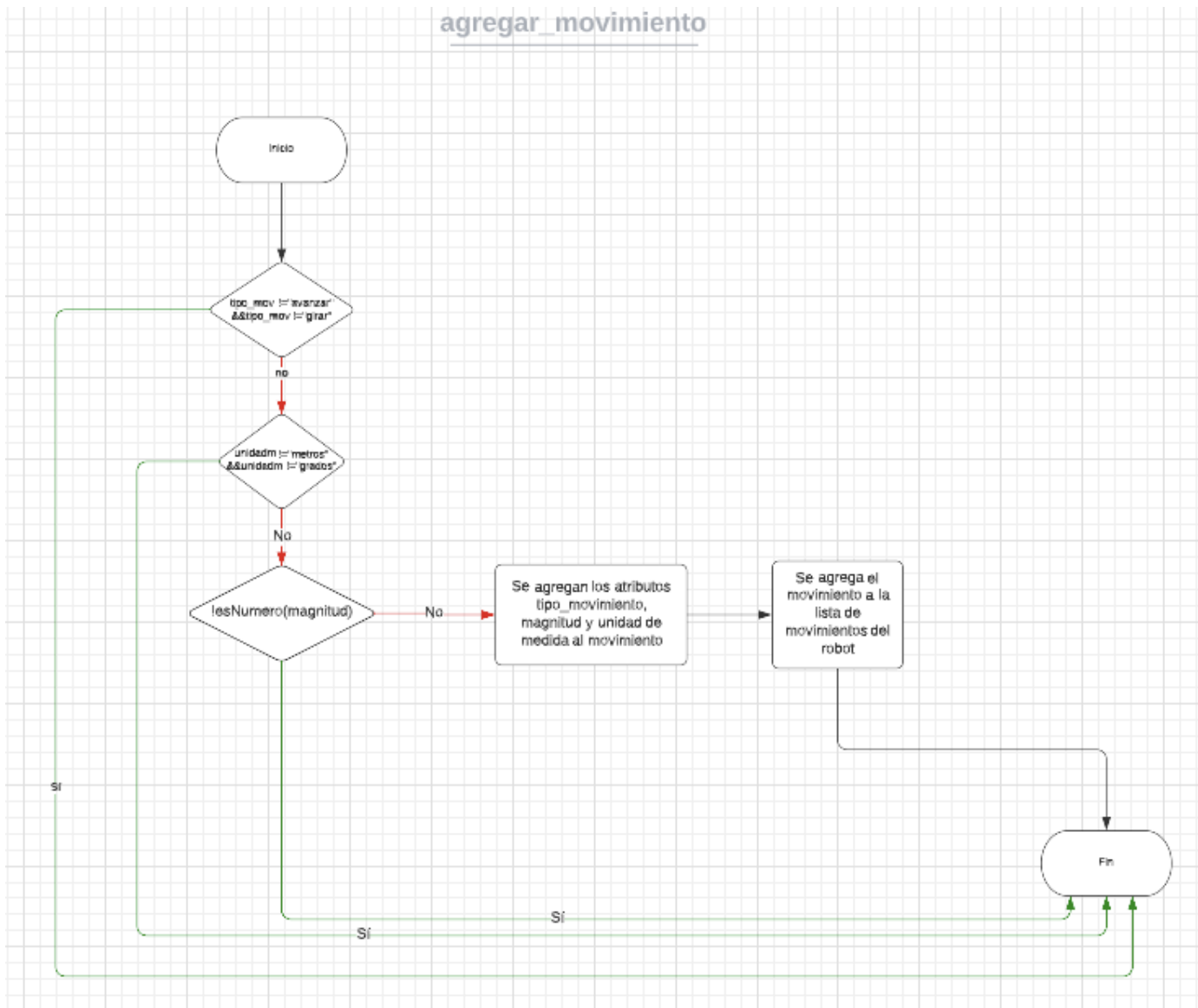
Link del diagrama: https://lucid.app/lucidchart/7e608231-e1fe-4892-a0d0-034e46678766/edit?viewport_loc=-408%2C3%2C2911%2C1383%2C0_0&invitationId=inv_76e2b1b7-a293-4048-8560-cd0e85ec33bf

- Función “cargar_elementos”:



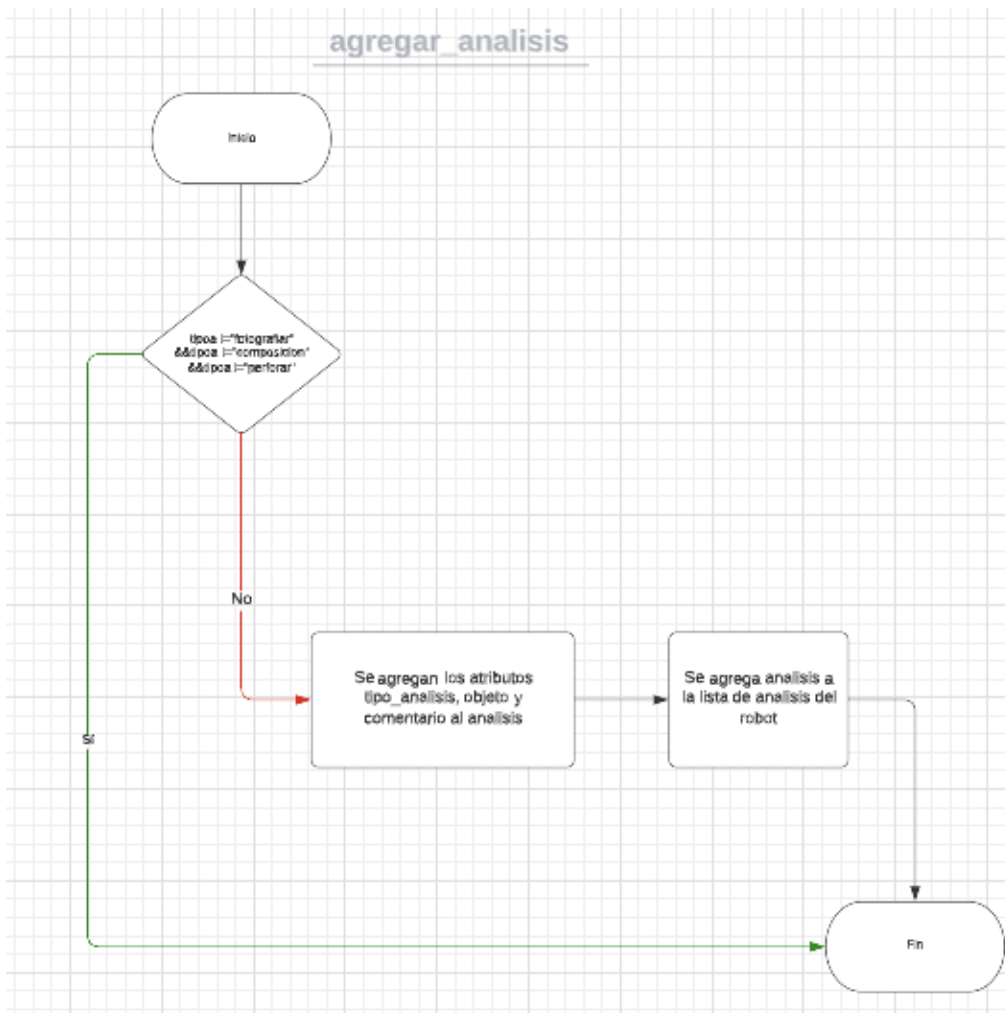
Link del diagrama: https://lucid.app/lucidchart/7e608231-e1fe-4892-a0d0-034e46678766/edit?viewport_loc=-534%2C29%2C2826%2C1342%2CGUZWfcK4cHs&invitationId=inv_76e2b1b7-a293-4048-8560-cd0e85ec33bf

- Función “agregar_movimiento”:



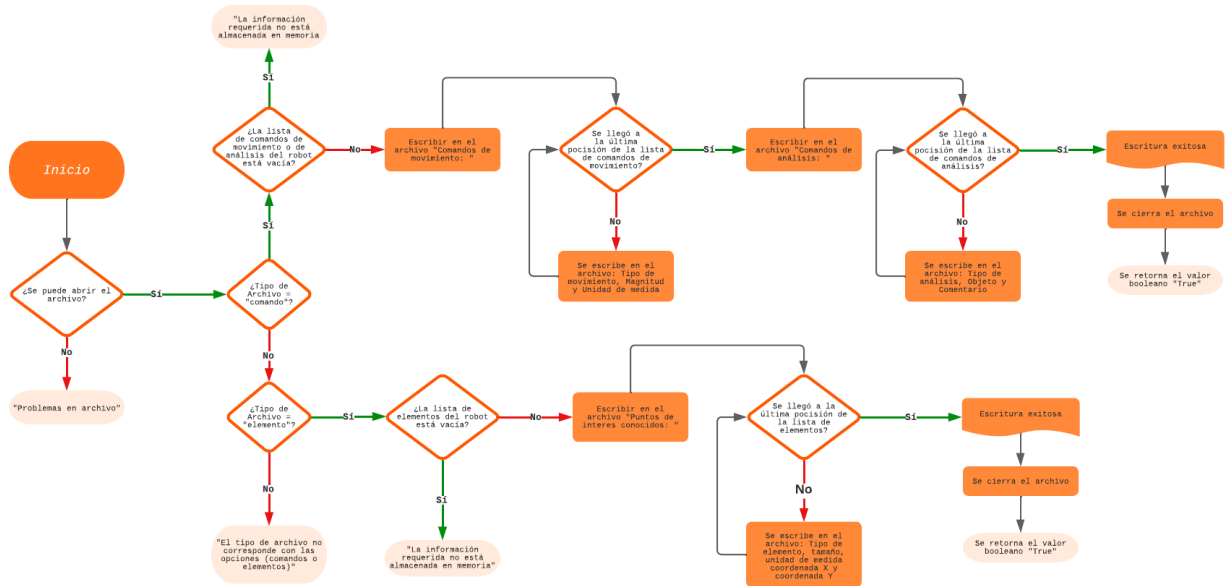
Link del diagrama: https://lucid.app/lucidchart/7e608231-e1fe-4892-a0d0-034e46678766/edit?viewport_loc=-758%2C-97%2C3203%2C1521%2C06ZCH9JY5YVP&invitationId=inv_76e2b1b7-a293-4048-8560-cd0e85ec33bf

- Función “agregar_analisis”:

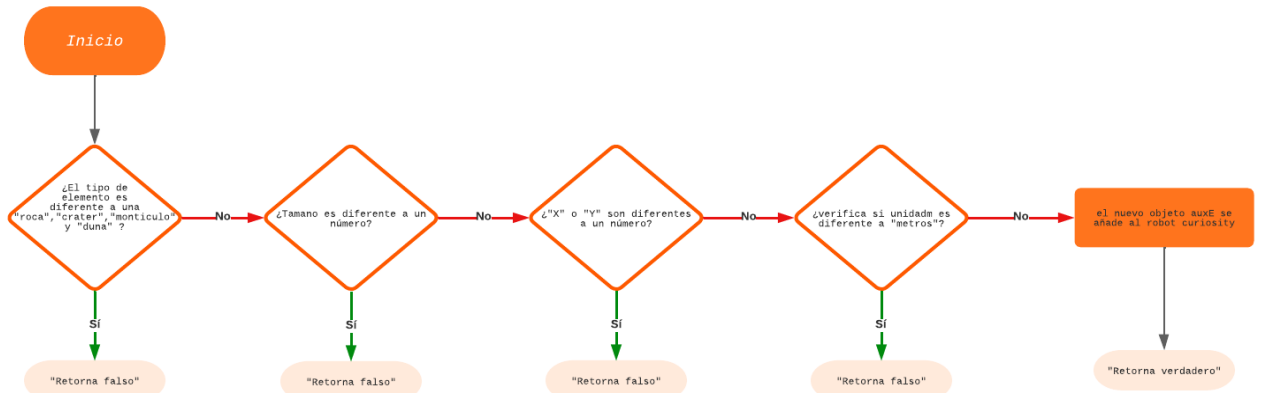


Link del diagrama: https://lucid.app/lucidchart/7e608231-e1fe-4892-a0d0-034e46678766/edit?viewport_loc=-547%2C37%2C3347%2C1590%2Cwk0Cq4n9H5rG&invitationId=inv_76e2b1b7-a293-4048-8560-cd0e85ec33bf

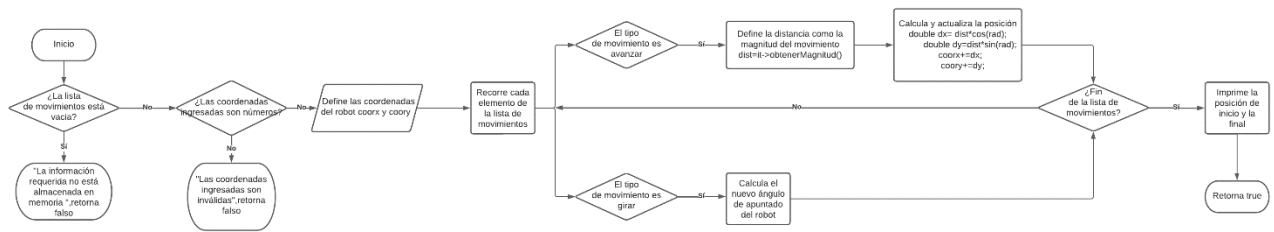
• Función “Guardar”



• Función “Agregar Elemento”



- **Función “Simular Comandos”:**



4. Plan de pruebas comando “simular_comandos”:

-Se proponen los siguientes datos para probar el correcto funcionamiento del comando:

Siguiendo los siguientes comandos de movimientos:

1. avanzar 4 metros
2. girar 360 grados
3. avanzar 3 metros
4. girar 180 grados
5. avanzar 2 metros

Plan de pruebas comando “simular_comandos”			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1.Posición inicial	X=0 Y=0	(3,6)	(3,6)
2.Posición obtenida de la inicial	X=3 Y=6	(6,12)	(6,12)
3.Datos no válidos	X=a Y=C	Datos no válidos	Datos no válidos
4.Un dato válido y uno inválido	X=1 Y=C	Datos no válidos	Datos no válidos