



# Query Crypter

Manual Técnico



# Versiones de las herramientas utilizadas

NodeJS 20.3.1

NPM 9.6.2

Visual Studio Code 1.83.1

Nodemon 2.0.22

Graphviz 7.1.0

NPX 9.6.2

ReactJS con VITE

# Especificaciones del entorno



The diagram consists of two identical rectangular blocks, one for the client and one for the server. Each block is composed of three overlapping rounded rectangles: a light blue one on the left, a white one in the center, and a light gray one on the right. The word 'CLIENTE' is centered in the white rectangle of the left block, and 'SERVIDOR' is centered in the white rectangle of the right block.

CLIENTE

SERVIDOR

# CLIENTE

- **ReactJS:**

ReactJS es una biblioteca de JavaScript de código abierto desarrollada por Facebook que se utiliza para construir interfaces de usuario modernas y dinámicas. Esta elección se basa en su capacidad para crear componentes reutilizables, su rendimiento excepcional y su comunidad activa de desarrolladores. ReactJS proporciona una base sólida para la construcción de la interfaz de usuario de su aplicación, permitiendo una experiencia de usuario intuitiva y atractiva.

- **Vite:**

Vite es un entorno de desarrollo ultrarrápido para aplicaciones web que está diseñado específicamente para ReactJS y otras bibliotecas modernas. Ofrece una velocidad de desarrollo impresionante al permitir la recarga en caliente, lo que significa que los cambios en su código se reflejan instantáneamente en el navegador sin necesidad de recargar la página. Vite es una elección estratégica para garantizar una eficiencia óptima durante el proceso de desarrollo.

# SERVIDOR

- **Node.js:**

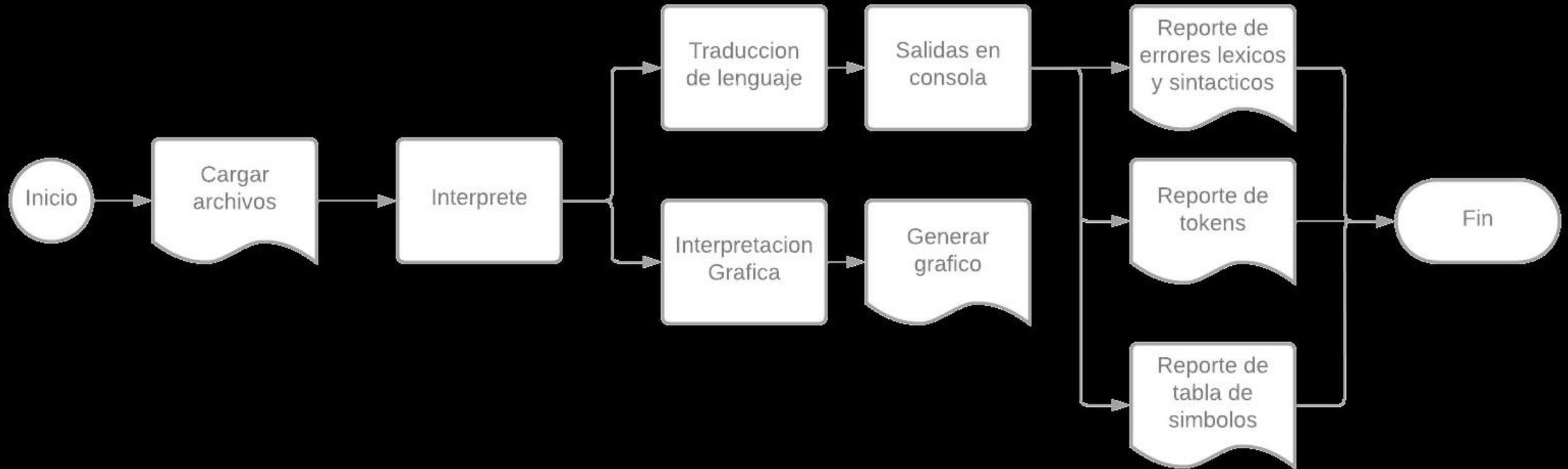
Node.js es un entorno de tiempo de ejecución de JavaScript del lado del servidor que permite la construcción de aplicaciones web escalables y eficientes. Utilizaremos Node.js para crear el servidor de su aplicación, gestionando las solicitudes del cliente y proporcionando datos y lógica empresarial de manera eficiente.

- **TypeScript:**

TypeScript es un superconjunto de JavaScript que agrega tipado estático y otras características avanzadas de desarrollo. Al utilizar TypeScript en todo el proyecto, obtendremos ventajas significativas en términos de mantenibilidad, detección temprana de errores y documentación de código. Esto asegura que su aplicación sea confiable y escalable a medida que evoluciona con el tiempo.



# Diagrama de flujo del programa



# Interfaz

Elegir archivos

No se eligió ningún archivo

Crear Nuevo Archivo

Guardar en Archivo Existente

1

1

Ejecutar

Generar Gráfico

Limpiar editor

Limpiar consola

1

# Editor

En este recuadro se ingresarán todas las instrucciones a analizar para posteriormente ver los reportes del compilador.

1

# Terminal

En este recuadro se mostraran los resultados del análisis de cada una de las instrucciones realizadas.



# Ejemplo de marco de trabajo

Elegir archivos pruebas3.bt

Crear Nuevo Archivo

Guardar en Archivo Existente

pruebas3.bt Eliminar

```
1 create table Clientes(  
2     nombre varchar,  
3     edad int,  
4     compra double  
5 );  
6  
7 insert into Clientes (nombre,edad,compra) values("Josue",24,33.5);  
8 insert into Clientes (nombre,edad,compra) values("Ana",21,32.5);  
9  
10 select * from Clientes;  
11  
12 delete from Clientes where edad < 22;  
13  
14 select * from Clientes;  
15
```

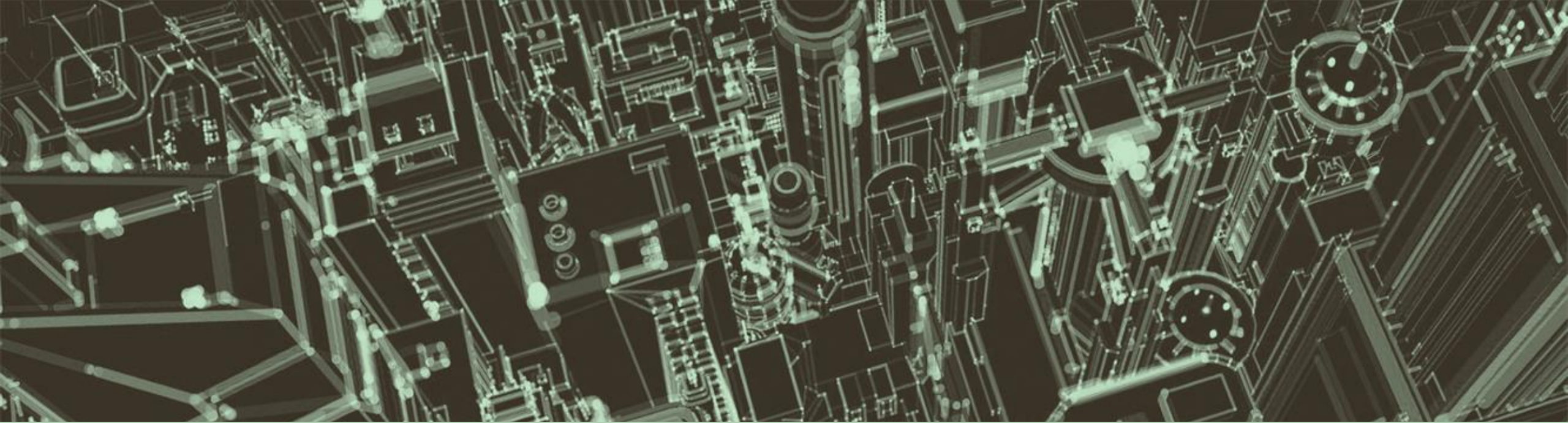
```
1 se creo la tabla Clientes  
2 En la tabla Clientes se inserto nombre con el valor de Josue  
3 En la tabla Clientes se inserto edad con el valor de 24  
4 En la tabla Clientes se inserto compra con el valor de 33.5  
5  
6 En la tabla Clientes se inserto nombre con el valor de Ana  
7 En la tabla Clientes se inserto edad con el valor de 21  
8 En la tabla Clientes se inserto compra con el valor de 32.5  
9  
10 Todos los valores de la tabla Clientes  
11 El valor de nombre es Josue  
12 El valor de edad es 24  
13 El valor de compra es 33.5  
14  
15 El valor de nombre es Ana  
16 El valor de edad es 21  
17 El valor de compra es 32.5  
18  
19  
20 Se eliminaron datos de la tabla Clientes  
21 Todos los valores de la tabla Clientes  
22 El valor de nombre es Josue  
23
```

Ejecutar

Generar Gráfico

Limpiar editor

Limpiar consola

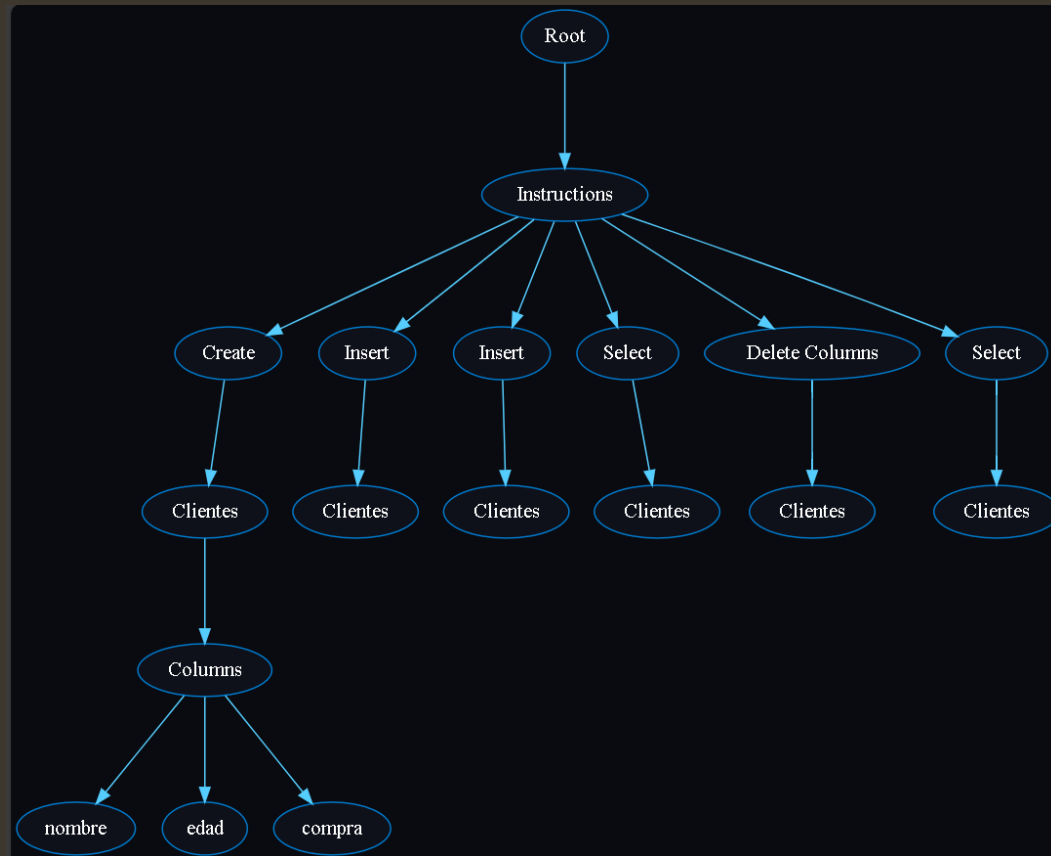


# REPORTES

Herramientas con las que se comprobara el análisis del compilador



# ARBOL AST



(Abstract Syntax Tree)

Es una estructura de datos utilizada en la programación y la compilación de código fuente. Representa la estructura jerárquica y la organización gramatical de un programa o código fuente en un formato más abstracto y simplificado que el código original.

Tipo de error	Descripcion	Linea	Columna
Lexical error	#	1	4
Sintatic error	;	5	1
Lexical error	&	7	5
Sintatic error	;	7	66
Sintatic error	;	10	23

# REPORTE DE ERRORES

En este reporte veras todos los errores encontrados en el analisis lexico con sus respectivas ubicaciones.

Token	Linea	Columna
Cientes	1	13
nombre	2	2
varchar	2	9
edad	3	2
int	3	7
compra	4	2
double	4	9
create	1	0
create	1	0
(	1	0
)	1	0
;	1	0
Cientes	7	12
nombre	7	22

# REPORTE DE TOKENS

En este reporte veras todos los tokens declarados en el analisis lexico con sus respectivas ubicaciones.

Identificador	Tipo	Tipo	Entorno	Linea	Columna
Cientes	identificador	int	Create table	1	0
Cientes	identificador	int	Insert table	7	0
Cientes	identificador	int	Insert table	8	0
Cientes	identificador	int	Select table	10	0
Cientes	identificador	int	Delete table	12	0
Cientes	identificador	int	Select table	14	0

# REPORTE DE SIMBOLOS

En este reporte veras todos los simbolos declarados con su respectiva ubicacion.