MANUAL TÉCNICO

EMULADOR DE SQL:
PRACTICA 1 DE
ORGANIZACIÓN DE LENGUAJES Y
COMPILADORES 1

BREVE DESCRIPCIÓN

El presente proyecto, es un emulador del lenguaje de SQL que es utilizado para la manipulación de datos dentro de bases de datos relacionales.

HERRAMIENTAS UTILIZADAS

- ## LENGUAJE DE PROGRAMACIÓN JAVA - VERSIÓN 17

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems.

El lenguaje Java proporciona:

- El paradigma de la programación orientada a objetos.
- Ejecución de un mismo programa en múltiples sistemas operativos y plataformas.
- Es posible utilizarlo para múltiples propósitos, desde aplicaciones de escritorio hasta en servidores web.
- Tiene una curva de aprendizaje media pero también toma lo mejor de otros lenguajes orientados a objetos, como C++.

También se utilizaron librerías/bibliotecas externas como:

- **JFlex - Análisis léxico**
- **Java CUP - Análisis sintáctico**

● JAVA SWING - (VISTA)

Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, listas desplegables y tablas.

Es un framework MVC para desarrollar interfaces gráficas para Java con independencia de la plataforma. Sigue un simple modelo de programación por hilos, y posee las siguientes características principales:

Independencia de plataforma.
Extensibilidad: es una arquitectura altamente particionada: los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto. Se puede extender clases existentes previendo alternativas de implementación para elementos esenciales.
Personalizable: dado el modelo de representación programático del framework de Swing, el control permite representar diferentes estilos de apariencia "look and feel" (desde apariencia MacOS hasta apariencia Windows XP, pasando por apariencia GTK+, IBM UNIX o HP UX, entre otros). Además, los usuarios pueden proveer su propia implementación de apariencia, que permitirá cambios uniformes en la apariencia existente en las aplicaciones Swing sin efectuar ningún cambio al código de aplicación.

● IDE: INTELLIJ IDEA ULTIMATE

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) escrito en Java para desarrollar software informático escrito en Java, Kotlin, Groovy y otros lenguajes basados en la JVM. Está desarrollado por JetBrains (anteriormente conocido como IntelliJ) y está disponible como una edición comunitaria con licencia de Apache 2, y en una edición comercial patentada. Ambos se pueden utilizar para el desarrollo comercial.

● GITHUB

Es un sistema de control de versiones de código y gestión de proyectos, a su vez también funciona como una plataforma de estilo red social diseñada para desarrolladores para poder compartir código entre más personas y colaborar en el mismo. Se utilizó GitFlow como metodología

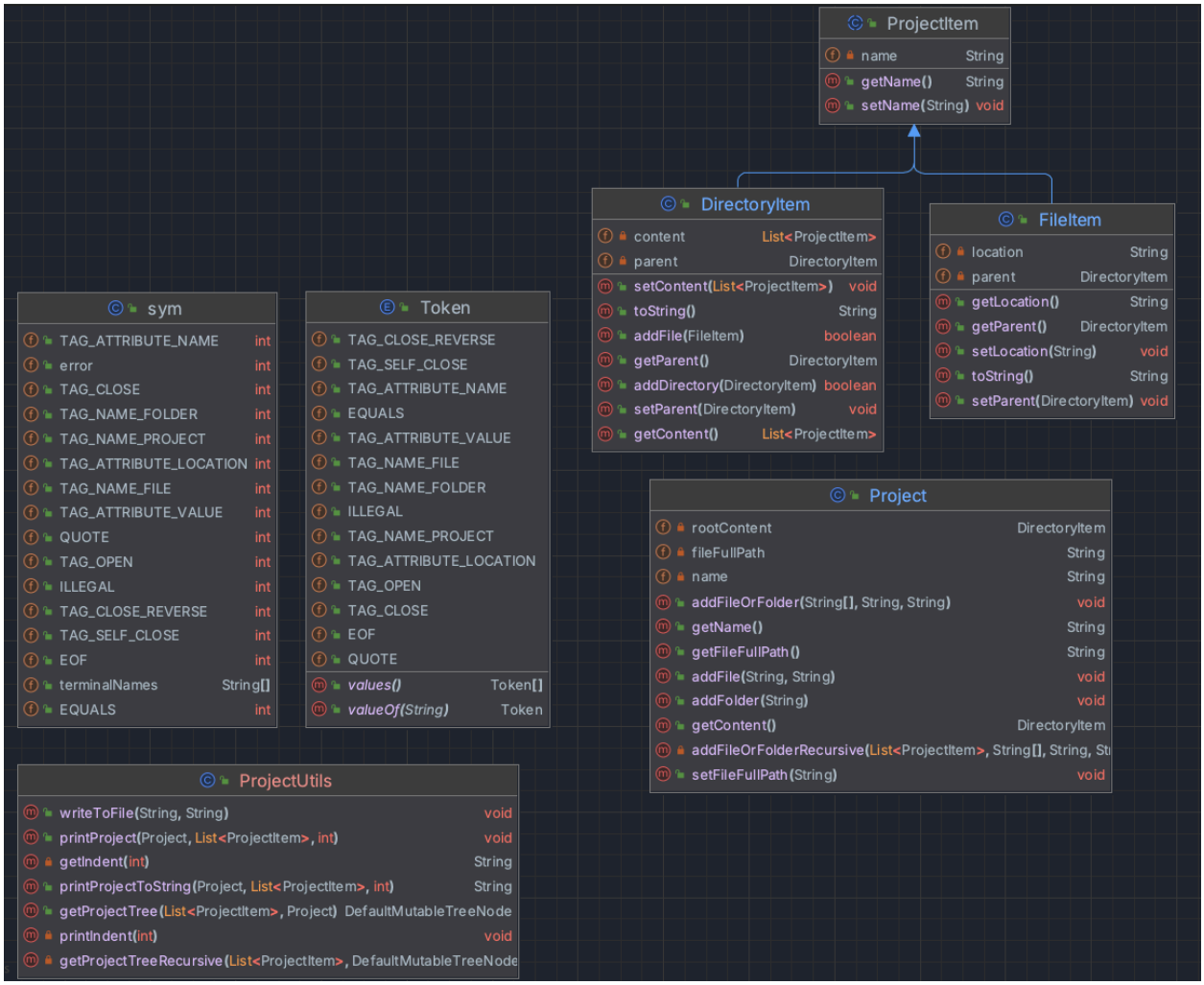
- ## MACOS v13 - SISTEMA OPERATIVO

macOS (previamente Mac OS X, luego OS X) es una serie de sistemas operativos gráficos desarrollados y comercializados por Apple desde 2001. Es el sistema operativo principal para la familia de computadoras Mac de Apple. Dentro del mercado de computadoras de escritorio, portátiles, hogareñas y mediante el uso de la web.

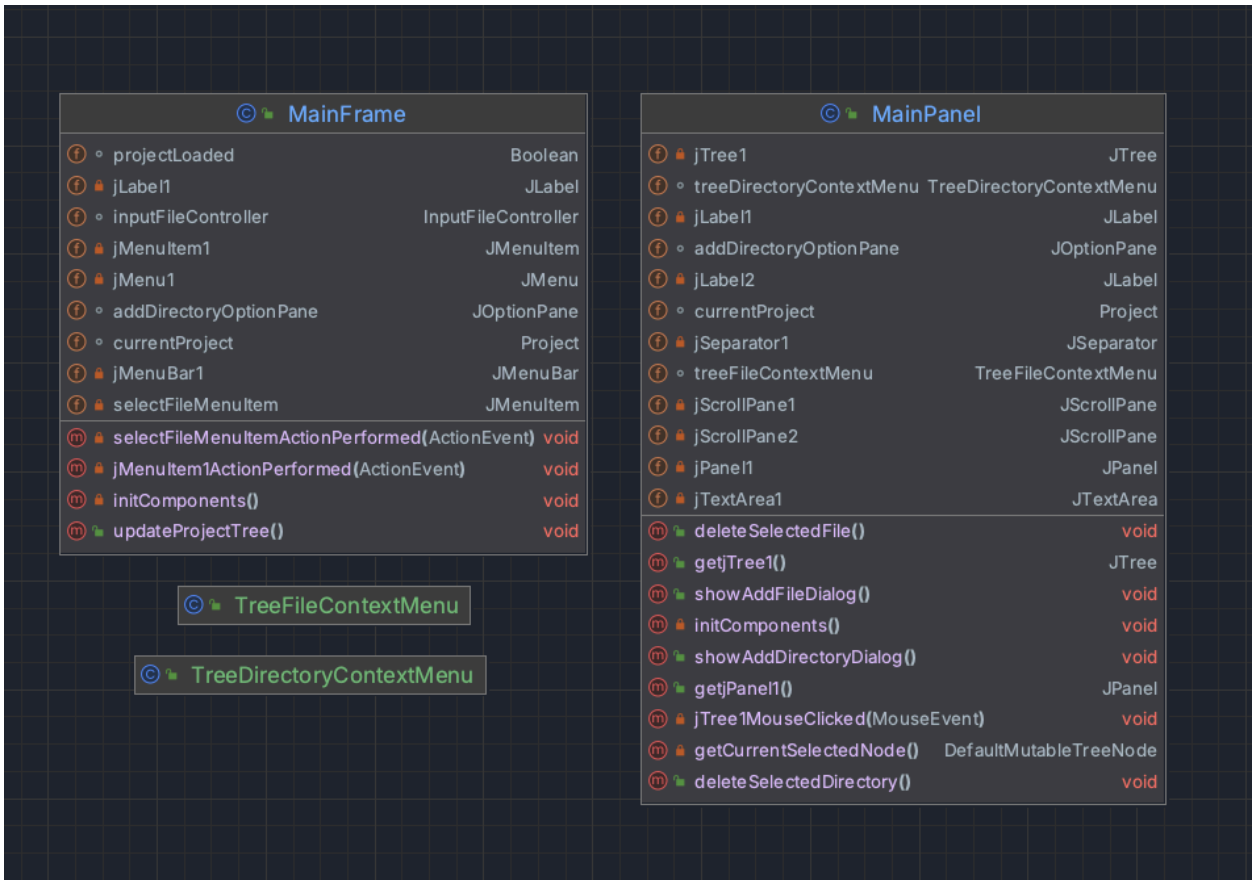
macOS se basa en tecnologías desarrolladas entre 1985 y 1997 en NeXT. Se logró la certificación UNIX 03 para la versión Intel de Mac OS X 10.5 Leopard y todos los lanzamientos de Mac OS X 10.6 Snow Leopard hasta la versión actual también tienen la certificación UNIX 03. macOS comparte su núcleo basado en Unix, llamado Darwin, y muchos de sus frameworks con iOS 16, tvOS y watchOS.

DIAGRAMA DE CLASES

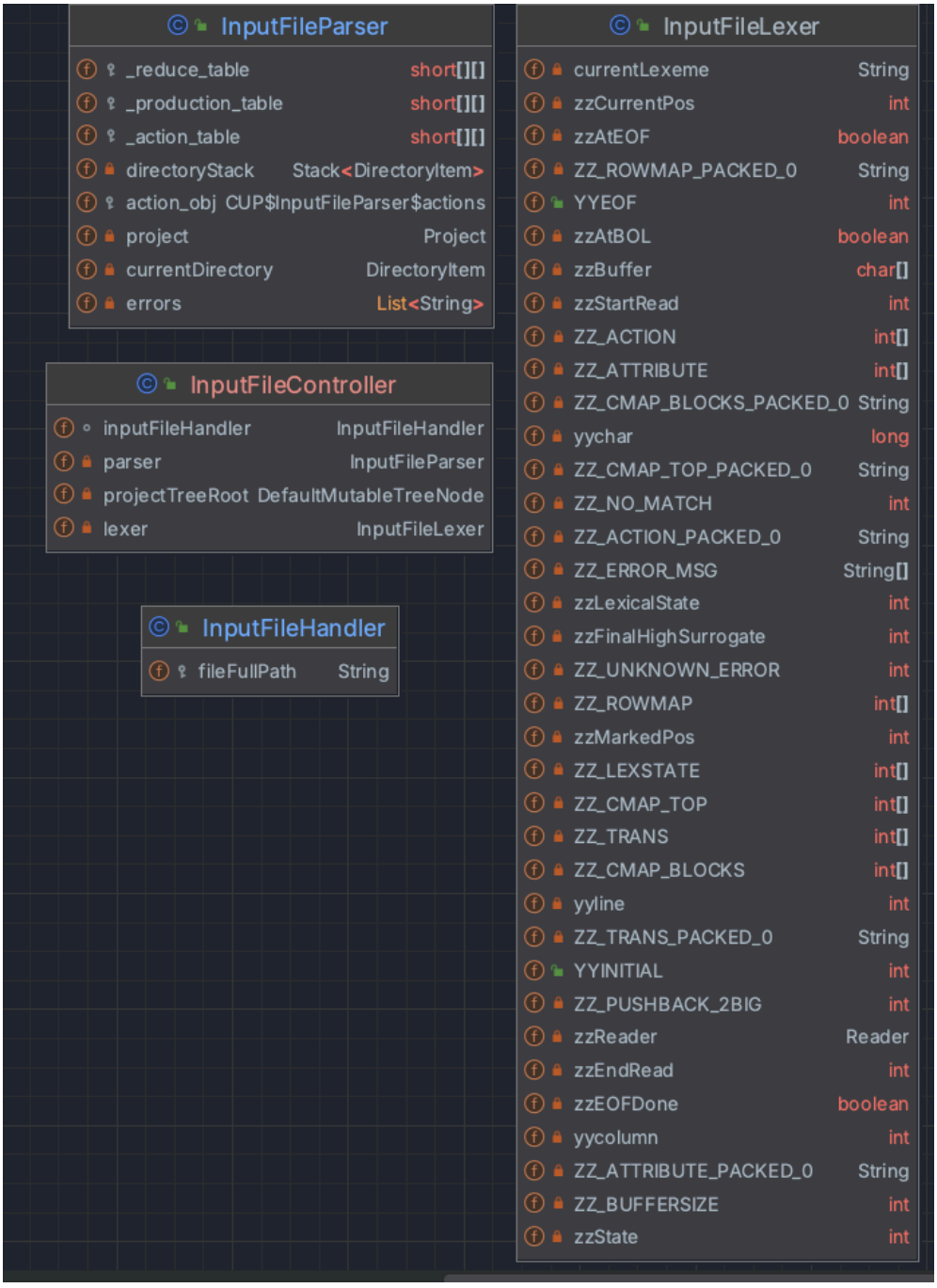
Modelos:



Interfaz:


MainFrame
projectLoaded Boolean
jLabel1 JLabel
inputFileController InputFileController
jMenuItem1 JMenuItem
jMenu1 JMenu
addDirectoryOptionPane JOptionPane
currentProject Project
jMenuBar1 JMenuBar
selectFileMenuItem JMenuItem
selectFileMenuItemActionPerformed(ActionEvent) void
jMenuItem1ActionPerformed(ActionEvent) void
initComponents() void
updateProjectTree() void
TreeFileContextMenu
TreeDirectoryContextMenu
MainPanel
jTree1 JTree
treeDirectoryContextMenu TreeDirectoryContextMenu
jLabel1 JLabel
addDirectoryOptionPane JOptionPane
jLabel2 JLabel
currentProject Project
jSeparator1 JSeparator
treeFileContextMenu TreeFileContextMenu
jScrollPane1 JScrollPane
jScrollPane2 JScrollPane
jPanel1 JPanel
jTextArea1 JTextArea
deleteSelectedFile() void
getjTree1() JTree
showAddFileDialog() void
initComponents() void
showAddDirectoryDialog() void
getjPanel1() JPanel
jTree1MouseClicked(MouseEvent) void
getCurrentSelectedNode() DefaultMutableTreeNode
deleteSelectedDirectory() void

Análisis


InputFileParser
_reduce_table short[][]
_production_table short[][]
_action_table short[][]
directoryStack Stack<DirectoryItem>
action_obj CUP$InputFileParser$actions
project Project
currentDirectory DirectoryItem
errors List<String>
InputFileController
inputFileHandler InputFileHandler
parser InputFileParser
projectTreeRoot DefaultMutableTreeNode
lexer InputFileLexer
InputFileHandler
fileFullPath String
InputFileLexer
currentLexeme String
zzCurrentPos int
zzAtEOF boolean
ZZ_ROWMAP_PACKED_0 String
YYEOF int
zzAtBOL boolean
zzBuffer char[]
zzStartRead int
ZZ_ACTION int[]
ZZ_ATTRIBUTE int[]
ZZ_CMAP_BLOCKS_PACKED_0 String
yychar long
ZZ_CMAP_TOP_PACKED_0 String
ZZ_NO_MATCH int
ZZ_ACTION_PACKED_0 String
ZZ_ERROR_MSG String[]
zzLexicalState int
zzFinalHighSurrogate int
ZZ_UNKNOWN_ERROR int
ZZ_ROWMAP int[]
zzMarkedPos int
ZZ_LEXSTATE int[]
ZZ_CMAP_TOP int[]
ZZ_TRANS int[]
ZZ_CMAP_BLOCKS int[]
yyline int
ZZ_TRANS_PACKED_0 String
YYINITIAL int
ZZ_PUSHBACK_2BIG int
zzReader Reader
zzEndRead int
zzEOFDone boolean
yycolumn int
ZZ_ATTRIBUTE_PACKED_0 String
ZZ_BUFFERSIZE int
zzState int

CLASES Y ALGORITMOS

Modelos:

Las clases dentro del paquete de modelos, representan a las entidades que interactúan en el proceso de la ejecución del programa, y almacena todos los datos necesarios para su funcionamiento y manipulación.
Principalmente el objeto de un proyecto, el cual representa y guarda todos los contenidos necesarios de este para poder interactuar mediante la interfaz de usuario.

Controladores:

En el caso de este proyecto, el único controlador presente se trata sobre el manejo del archivo de entrada. En estas clases están los métodos y propiedades responsables de leer el archivo, crear nuevos y guardarlos dentro del sistema operativo. Y a la vez también las relaciones entre los componentes de análisis léxico y sintáctico.

Utilidades:

En este paquete, están algunas utilidades definidas, que son comúnmente usadas entre varias clases y se han extraído en diferentes clases para evitar la repetición del código.

Frames:

Todas las clases dentro del paquete “FRAMES” son las encargadas de manejar y mostrar todas las interfaces gráficas del programa y conectarse con los controladores para usar los datos guardados por los modelos a través de la ejecución.

INSTRUCCIONES PARA EJECUCIÓN

Para ejecutar este programa solamente es necesario tener instalado en el sistema operativo, el programa Java, al menos en su versión 17

Al descargar el ejecutable .jar, solo es necesario ejecutar desde la terminal el comando:

```
java -jar nombre_del_ejecutable.jar
```