

Manual TECNICO

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2



29 de septiembre de 2019

|  |  |
| --- | --- |
| Carnet | Nombre |
| 201503712 | Oscar Rene Cuéllar Mancilla |

Contenido

[1. Descripción de la Aplicación y Requerimientos Mínimos 2](#_Toc20484869)

[1.1 Descripción de la aplicación 2](#_Toc20484870)

[1.2 Framework Cliente y Servidor 2](#_Toc20484871)

[1.3 Generador de Analizador Léxico y Sintáctico (Irony) 3](#_Toc20484872)

[1.4 Lenguaje de desarrollo Servidor (C#) 4](#_Toc20484873)

[1.5 Generador de Analizador Léxico y Sintáctico (Jison) 5](#_Toc20484874)

[1.6 Requerimientos Mínimos 5](#_Toc20484875)

# 1. Descripción de la Aplicación y Requerimientos Mínimos

## Descripción de la aplicación

CQL-Teacher es una aplicación cliente–servidor, que permitirá interactuar con los usuarios para que puedan familiarizarse con el uso y beneficios de una base de datos, de una forma bastante fácil e intuitiva, proveerá bastantes modos para que así puedan trabajar de acuerdo con su nivel de conocimiento. Además, el sistema de base de datos también contará con instrucciones y elementos basados de un lenguaje estructurado, tales como variables, ciclos, sentencias condicionales y tipos de datos definidos por el usuario.

La aplicación cliente, CQL-Client, es una aplicación web que permitirá al usuario interactuar con el servidor de bases de datos, esta contará con una serie de componentes que le brindan al usuario una experiencia agradable al momento de utilizarlo. Tiene 3 distintos modos de dificultad para la edición de código de alto nivel, así como un login para garantizar la seguridad de la información.

## Framework Cliente y Servidor

ASP.NET es un entorno para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.



## Generador de Analizador Léxico y Sintáctico (Irony)

Irony es un kit de desarrollo para la aplicación de las lenguas en la plataforma NET. A diferencia de la mayoría de soluciones yacc/lex-style irony no utiliza ningún escáner o el programa de análisis de generación de código a partir de especificaciones de gramática escritos en una meta-lenguaje especializada.

En Irony la gramática idioma de destino se codifica directamente en C # utilizando la sobrecarga de operadores para expresar las construcciones gramaticales. Módulos del escáner y analizador de irony utilizan la gramática codificado como c # clase para controlar el proceso de análisis.

Irony es un analizador que se caracteriza por ser uno de los pocos, o porque no decir el único que funciona en base a la generación del AST (Abstract Syntax Tree ó Árbol de Sintaxis Abstracta); El AST tiene la característica primordial, en que es la base fundamental para la ejecución de código, eso quiere decir que por medio del mismo podemos llevar el control de todo lo que hayamos generado mediante la gramática, pero tomando en cuenta que la generación del árbol implica únicamente la generación, ejecución (esto se debe implementar posteriormente por el programador) pero nunca la modificación del árbol, ya que en base a este árbol, se generará la tabla de símbolos, la cual es la que si debe ser modificable.

El usuario quien implementa una gramática con Irony debe tomar en cuenta que la implementación de la misma puede ser ambigua, re-cursiva por la izquierda y demás, ya que es un analizador ascendente, pero si se aplica ambigüedad en la gramática, existe uno de sus tantos métodos llamado precedence, el cual evalúa la precedencia que le indiquemos para realizar las operaciones necesarias.



## Lenguaje de desarrollo Servidor (C#)

C# (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por el signo #, el cual se lee como sharp en inglés para notación musical. Es un juego de palabras, pues '"C#" significa, musicalmente hablando, "do sostenido", donde el símbolo # indica que una nota (en este caso do, representada por C) debe ser un semitono más alta. Esto es una metáfora de la superioridad de C# sobre su antecesor C++ y a su vez hace alusión a la misma metáfora que se ideó para dar nombre a C++.1​ Además, el símbolo # puede ser imaginado como la unión de cuatro símbolos +, continuando así con el sentido de progresión de los lenguajes C.



## Generador de Analizador Léxico y Sintáctico (Jison)

Jison toma una gramática libre de contexto como entrada y genera un archivo JavaScript capaz de analizar el lenguaje descrito por esa gramática. Luego puede usar el script generado para analizar entradas y aceptar, rechazar o realizar acciones basadas en la entrada. Si estás familiarizado con Bison o Yacc, u otros clones, estás casi listo para rodar.



## Requerimientos Mínimos

Los requerimientos mínimos del proyecto son funcionalidades del sistema que permitirán un ciclo de ejecución básica.

2 Descripción

2.1.5 Login

2.1.6 Panel de información

2.1.7 Salida de datos y Consola

2.1.8 Modos de edición

2.1.8.1 Principiante

2.1.8.2 Avanzado

2.4 Definiciones generales de CQL

2.4.1 Identificadores

2.4.2 Case Insensitive

2.4.3 Comentarios

2.4.4 Valor nulo

2.5 Tipos de dato

2.5.1 Tipos primitivos

2.5.2 Secuencias de escape

2.5.3 User Types

2.5.4 Casteos primitivos en CQL

3 Lenguaje de Definición de Datos (DDL).

3.1 Crear Base de datos

3.2 Sentencia Use 3.4 Crear Tabla

3.4.1 Tipo Dato Counter

3.4.2 Llaves Primarias

3.4.3 Tipos de datos para definición de columnas

3.5 Alter Table

4 Lenguaje de Control de Transacciones (TCL).

4.1 Commit

4.2 Rollback

5 Lenguaje de Control de Datos (DCL).

5.1 Crear Usuario

6 Lenguaje de Manipulación de datos (DML).

6.1 Insertar

6.2 Actualizar

6.4 Seleccionar

6.6 Funciones de Agregación

6.7 Clausula Where

6.7.1 Operaciones Lógicas

6.8 Uso de colecciones y objetos como componentes de una tabla

6.8.3 Listas

6.8.4 User Types

7 Lenguaje de Control de Flujo (FCL)

7.2 Sintaxis

7.2.3 Variables

7.2.4 Declaración de variables

7.2.5 Asignación de variables

7.2.6 Valores predeterminados para las variables

7.2.7 Operadores aritméticos

7.2.8 Operadores relacionales

7.2.9 Operaciones lógicas

7.2.12 Sentencias de Selección

7.2.12.1 Sentencia If

7.2.12.2 Sentencia else

7.2.13 Sentencias cíclicas o bucles

7.2.13.1 Sentencia While

7.2.13.2 Sentencia For

7.2.14 Collections

7.12.14.1 Lists

7.2.15 Funciones

7.2.16 Procedimientos

7.2.19 Sentencias de transferencia

7.2.19.1 Break

7.2.19.2 Return

7.2.20 Cursores

7.2.21 Log

8 Lenguaje Unificado de Paquetes (LUP)

8.1 Solicitudes del cliente al servidor:

8.1.1 Paquete de Inicio de Sesión

8.1.2 Paquete de Fin de Sesión

8.1.3 Paquete de Consulta:

8.2 Respuestas del servidor al cliente:

8.2.1 Paquete de Datos

8.2.2 Paquete de Error

8.2.3 Paquete de Mensaje

8.2.4 Paquete de Estructura

9 El Lenguaje de Almacenamiento de Datos (CHISON)

9.1 Notación

9.1.1 Datos Primitivos

9.2 Definición de los archivos

9.2.1 Archivo Principal

9.3 Importar Archivo

10 Excepciones

11 Apéndice A: Precedencia y asociatividad de operadores

12 Manejo de errores

# 2. Diagrama de Clases