

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas
Organización de Lenguajes y Compiladores 2
Segundo semestre 2019



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Catedráticos: Ing. Bayron López, Ing Edgar Saban e Ing Erick Navarro

Tutores académicos: Luis Lizama, Julio Arango, Rainman Sian

CQL-Teacher

Hoja de calificación para el primer proyecto de laboratorio

Fecha Calificación: ____/____/____

Sección: _____

Nombre: _____ Registro académico: _____

NOTA: Los requerimientos mínimos del proyecto son funcionalidades del sistema que permitirán un ciclo de ejecución básica. En el caso de no cumplir con **uno o más** de los siguientes requerimientos **no se calificará el proyecto y el estudiante tendrá una nota de cero puntos.**

Requerimientos mínimos	Cumple	No cumple
Login		
Panel de información		
Salida de datos y Consola		
Modos de edición		
Principiante		
Avanzado		
Definiciones generales de CQL		
Identificadores		
Case Insensitive		
Comentarios		
Valor nulo		
Tipos de dato		
Tipos primitivos		
Secuencias de escape		
User Types		
Casteos primitivos en CQL		
Lenguaje de Definición de Datos (DDL).		
Crear Base de datos		
Sentencia Use		
Crear Tabla		
Tipo Dato Counter		
Llaves Primarias		
Tipos de datos para definición de columnas		
Alter Table		

Lenguaje de Control de Transacciones (TCL).		
Commit		
Rollback		
Lenguaje de Control de Datos (DCL).		
Crear Usuario		
Lenguaje de Manipulación de datos (DML).		
Insertar		
Actualizar		
Seleccionar		
Funciones de Agregación		
Clausula Where		
Operaciones Lógicas		
Uso de colecciones y objetos como componentes de una tabla		
Listas		
User Types		
Lenguaje de Control de Flujo (FCL)		
Sintaxis		
Variables		
Declaración de variables		
Asignación de variables		
Valores predeterminados para las variables		
Operadores aritméticos		
Operadores relacionales		
Operaciones lógicas		
Sentencias de Selección		
Sentencia If		
Sentencia else		
Sentencias cíclicas o bucles		
Sentencia While		
Sentencia For		
Collections		
Lists		
Funciones		
Procedimientos		
Sentencias de transferencia		
Break		
Return		
Cursores		
Log		
Lenguaje Unificado de Paquetes (LUP)		
El Lenguaje de Almacenamiento de Datos (CHISON)		
Excepciones		
Apéndice A: Precedencia y asociatividad de operadores		

Ponderación

Los puntos se encuentran repartidos en diez secciones. Cada sección será evaluada sobre cien puntos para facilitar la calificación. El puntaje neto de cada sección se encuentra indicado en la siguiente tabla:

DESCRIPCION	Ponderación	Puntaje Obtenido
1. Archivo requerimientos mínimos	--	
2. Características de CQL-Client	20	
3. Funcionalidades básicas	15	
4. Funcionalidades Intermedias	10	
5. Funcionalidades de nivel avanzado	25	
6. Lenguaje de almacenamiento, CHISON	15	
7. Lenguaje unificado de paquetes, LUP	5	
8. Excepciones	4	
9. Errores léxicos y sintácticos	3	
10. Documentación	3	
TOTAL	100	

1 Archivos de requerimientos mínimos

En esta sección se calificará que cumpla con los requerimientos mínimos.

1.1 Generalidades de CQL

En esta sección se evaluará que se cumpla con los aspectos generales del lenguaje CQL. Se pasará un archivo de entrada con todos los requerimientos mínimos, es de forma obligatoria que el proyecto corra todo el archivo de entrada sin ninguna modificación y debe imprimir la salida exacta, en caso no se pueda ejecutar dicho archivo el estudiante no podrá continuar con la calificación, en caso de que el estudiante logre ejecutar el archivo podrá calificarse todo lo que muestre en el lapso de 2 horas.

1.2 DDL

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten creación de bases de datos y tablas, así como también todos los tipos de datos soportado.

1.3 TCL

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten la persistencia de datos en los archivos CHISON y la lectura de datos en Memoria

1.4 DCL

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten el acceso a la base de datos.

1.5 DML

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten la consulta, inserción y modificación de la información en la base de datos.

1.6 FCL

En esta sección se evaluará el correcto funcionamiento de las sentencias que forman parte del lenguaje FCL.

1.7 LUP

En esta sección se evaluará que la aplicación cumpla con los requisitos mínimos durante envío y lectura de paquetes para la comunicación con la base de dato

1.8 CHISON

En esta sección se evaluará que la aplicación cuente con los requisitos mínimos durante el almacenamiento de los archivos chison y su lectura en memoria

1.9 DDL

En esta sección se evaluará que la aplicación cuente con los requisitos mínimos en el procesamiento y manejo de excepciones.

Tiempo estimado para la evaluación de esta sección: **15 minutos.**

DESCRIPCION	Cumple	No Cumple
Generalidades de CQL		
Case insensitive		
Comentarios		
Null		
Tipos de datos primitivos		
User Types		
Casteos implícitos		
Secuencias de escape		
DDL		
Crear base de datos		
Usar base de datos		
Crear Tabla		
TCL		
Commit		

Rollback		
DCL		
Crear usuario		
DML		
Insertar		
Actualizar		
Seleccionar		
FCL		
Variable		
Operaciones		
Sentencias de Selección		
List		
Funciones		
Procedimientos		
Sentencias de transferencia		
Cursorres		
Log		
LUP		
CHISON		
EXCEPCIONES		
CUMPLIMIENTO MINIMOS	SI	NO

2 Características de CQL-Client

En esta sección se calificará que las funcionalidades de la aplicación se hayan implementado correctamente. Para cada punto a calificar se realizarán las acciones indicadas y se deberá comprobar, dentro de la aplicación, que la funcionalidad cumple con lo esperado.

2.1 CQL-Client

En esta sección se evaluará los aspectos de acceso e información del cliente.

2.2 Modo Principiante

En esta sección se evaluará que CQL-Client cumpla con los bloques necesarios para trabajar en modo principiante.

2.3 Modo Intermedio

En esta sección se evaluará que CQL-Client cumpla con los bloques necesarios para trabajar en modo intermedio

2.4 Modo Avanzado

En esta sección se evaluará que el editor cumpla con los requisitos solicitados para poder interactuar con la base de datos.

Tiempo estimado para la evaluación de esta sección: **10 minutos.**

DESCRIPCION	Ponderación		Puntaje Obtenido
CQL-Client	15		
Login		3	
Panel de Información		4	
Salida de consultas		5	
Consola		3	
Modo Principiante	25		
Bloque use		3	
Select		7	
Insert		5	
Update		5	
Delete		5	
Modo Intermedio	45		
Bloque use		3	
Definición de variables		5	
Operaciones con expresiones básicas		8	
Sentencia If		5	
Sentencia Switch		5	
Sentencia While		5	
Sentencia For		5	
Llamada a Función		3	
Llamada a Procedimiento		3	
Llamada a función LOG		3	
Modo Avanzado	15		
Manejo de múltiples pestañas		5	
Guardar		3	
Abrir		3	
Ejecución de archivo básico		4	
TOTAL (100%)	100		

3 Funcionalidades Básicas

En esta sección se calificará que la aplicación sea capaz de ejecutar el flujo completo de ejecución con instrucciones de baja complejidad.

3.1 Generalidades de CQL

En esta sección se evaluará que se cumpla con los aspectos generales del lenguaje CQL

3.2 DDL

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten creación de bases de datos y tablas, así como también todos los tipos de datos soportado.

3.3 TCL

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten la persistencia de datos en los archivos CHISON y la lectura de datos en Memoria

3.4 DCL

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten el acceso a la base de datos.

3.5 DML

En esta sección se evaluará el correcto funcionamiento de las sentencias que permiten la consulta, inserción y modificación de la información en la base de datos.

3.6 FCL

En esta sección se evaluará el correcto funcionamiento de las sentencias que forman parte del lenguaje FCL.

3.7 LUP

En esta sección se evaluará que la aplicación cumpla con los requisitos mínimos durante envío y lectura de paquetes para la comunicación con la base de dato

3.8 CHISON

En esta sección se evaluará que la aplicación cuente con los requisitos mínimos durante el almacenamiento de los archivos chison y su lectura en memoria

3.9 DDL

En esta sección se evaluará que la aplicación cuente con los requisitos mínimos en el procesamiento y manejo de excepciones.

Tiempo estimado para la evaluación de esta sección: **15 minutos.**

DESCRIPCION	Ponderación		Puntaje Obtenido
Generalidades de CQL	8		
Identificadores		1	
Case insensitive		1	
Comentarios		1	
Null		1	
Tipos de datos primitivos		1	
User Types		1	
Casteos implícitos		1	
Secuencias de escape		1	
DDL	15		
Crear base de datos		4	
Usar base de datos		1	
Crear Tabla	10		
Tipo Dato Counter		2	
Llaves Primarias		4	
Tipos de Datos		4	
TCL	10		
Commit		5	
Rollback		5	
DCL	5		
Crear usuario		5	
DML	15		
Insertar		3	
Actualizar		2	
Seleccionar		3	
Clausula Where	3		
Operaciones lógicas		3	
Uso de Coleccions y User Types	4		
Listas		2	
User Types		2	
FCL	20		
Variable	1		
Tipos		0.25	
Declaración		0.25	
Asignación		0.25	
Valores predeterminados		0.25	
Operaciones Aritméticas	1		
Operaciones relacionales	1		
Operaciones lógicas	1		

Sentencias de Selección	3		
Sentencia If		1	
Sentencia Else		1	
Sentencias cíclicas	3		
Sentencia While		2	
Sentencia For		2	
Collections	2		
Lists		2	
Funciones	2		
Procedimientos	2		
Sentencias de transferencia	1		
Break		0.5	
Return		0.5	
Cursores	2		
Log	1		
LUP	14		
Solicitudes del cliente al servidor	6		
Paquete de Inicio de Sesión		2	
Paquete de Fin de sesión		2	
Paquete de Consulta		2	
Respuestas del servidor al cliente	8		
Paquete de datos		2	
Paquete de error		2	
Paquete de mensaje		2	
Paquete de estructura		2	
CHISON	10		
EXCEPCIONES	3		
TOTAL (100%)	100		

4 Funcionalidades Intermedias

Esta sección permitirá evaluar la implementación correcta de los ámbitos locales utilizando las estructuras en tiempo de ejecución. Una función es recursiva si en su definición existen llamada(s) a sí misma.

4.1 Recursión simple (Utilizando operador ternario)

Una función posee recursividad simple si dentro de su bloque de sentencias solo aparece una llamada a sí misma. Esta parte permitirá la evaluación de la sentencia de selección operador ternario.

4.1.1 Función factorial de un numero n

La función factorial está definida para números enteros y es estrictamente creciente, lo cual facilitará la evaluación del manejo de errores en tiempo de ejecución ya que permite fácilmente superar el rango establecido para el tipo de datos entero. Se espera como salida la factorial de n, es decir, $n!$, permitirá evaluar la recursión simple.

4.1.2 Función potencia

La función potencia calculará la siguiente expresión: x^n , de forma recursiva permitirá evaluar la recursión simple.

4.1.3 Función MCD

La función MCD calculará el máximo común divisor de forma recursiva, permitirá evaluar la recursión simple.

4.2 Recursión múltiple

Una función posee recursividad simple si dentro de su bloque de sentencias existe una o más llamadas a sí misma.

4.2.1 Función Fibonacci

La función Fibonacci como su nombre lo dice calculará el Fibonacci del número que recibe de parámetro, esto permitirá evaluar la recursión múltiple.

4.2.2 Solución al problema Torres de Hanoi

Las Torres de Hanói es un juego matemático cuya solución es recursiva, esto permitirá evaluar la recursión múltiple.

4.2.3 Ordenamiento QuickSort

El ordenamiento QuickSort es un algoritmo de ordenación. Este permitirá evaluar la recursividad múltiple y uso de listas.

4.3 Recursión cruzada o mutua

Dos funciones se llaman mutuamente recursivas si la primera función hace una llamada recursiva a la segunda función y la segunda función, a su vez, llama a la primera.

4.3.1 Función generatriz de sucesiones Hofstadter

Esta función posee recursividad cruzada ya que lo cual representa una dependencia circular entre dos funciones, esta permitirá evaluar la recursión cruzada o mutua.

4.3.2 Par o impar

Esta parte será un algoritmo recursivo que determine la paridad de un numero n permitirá la evaluación de la recursividad cruzada.

4.4 Tabla Hash con estructuras del lenguaje

Se hará uso de las estructuras propias del lenguaje para crear una tabla hash de objetos, un map será encargado de realizar el mapeo y se tendrá una lista para resolver las colisiones. Se podrán realizar operaciones de inserción, eliminación y modificación sobre dicha tabla.

4.5 Tabla Hash con estructuras del lenguaje

Se creará una tabla hash por medio de los collections nativos del lenguaje, usará de base un Map de clave valor como la tabla hash y en su interior tendrá el uso de collections para la resolución de las colisiones. Se hará uso de funciones recursivas para el recorrido de todas las estructuras al momento de necesitar un reajuste en la tabla hash

Tiempo estimado para la evaluación de esta sección: **10 minutos.**

DESCRIPCION	Ponderación		Puntaje Obtenido
Recursividad simple	12		
Función Factorial de un número n		4	
Función Potencia		4	
Función MCD		4	
Recursividad múltiple	14		
Función Fibonacci		4	
Torres de Hanoi		5	
QuickSort		5	
Recursión cruzada	14		
Función de Hofstadter		7	
Par o Impar		7	
Recursividad Anidada (Ackerman)	30		
m=3, n=4 (sin aumento de memoria)		10	
m=3, n=6 (sin aumento de memoria)		20	
Tabla Hash con estructuras del lenguaje	30		
Salida Inserción		10	
Salida Modificación		10	
Salida Eliminación		10	
TOTAL (100%)			

5 Funcionalidades de nivel avanzado

En esta sección se evaluarán especialmente el uso correcto de estructuras haciendo uso de los user types y con recorridos recursivos combinando funcionalidades avanzadas de objetos, cadenas y collections.

También se evaluará todo el uso correcto de cursores, se contará con una serie de procedimientos que retornaran distintos cursores, verificando así el correcto uso de procedimientos y retornos múltiples.

Dentro de los procedimientos se encontrarán sentencias batch y se hará uso de la recursividad con funciones para una inserción masiva de datos, datos los cuales serán los que analice los cursores anteriormente mencionados.

5.1 Árbol Binario de Búsqueda

Este archivo de entrada tendrá el user type de nodo y todas las funciones necesarias para la implementación de un árbol Binario de Búsqueda, el cual permitirá evaluar todos los componentes del lenguaje FCL, incluyendo los collections y recorridos recursivos.

5.2 Árbol AVL

Este archivo de entrada tendrá el user type de nodo y todas las funciones necesarias para la implementación de un árbol AVL, el cual permitirá evaluar todos los componentes del lenguaje FCL, incluyendo los collections y recorridos recursivos.

5.3 Matriz dispersa

Este archivo de entrada tendrá los user types y todas las funciones necesarias para la implementación de una matriz dispersa, el cual permitirá evaluar todos los componentes del lenguaje FCL, incluyendo los collections y recorridos recursivos.

Tiempo estimado para la evaluación de esta sección: **20 minutos.**

DESCRIPCION	Ponderación		Puntaje obtenido
Árbol ABB	21		
Salida inserción		7	
Salida búsqueda		7	
Salida eliminación		7	
Árbol AVL	21		
Salida inserción		7	
Salida búsqueda		7	
Salida eliminación		7	
Matriz dispersa	23		
Salida inserción		7	
Salida búsqueda		7	
Salida eliminación		9	
Procedimientos	9		
Múltiple retorno de cursores		9	
Batch	13		
Ejecución de todas las instrucciones conjuntas		6	
Descartar cualquier instrucción ejecutada al detectar error		7	
Cursores	13		
Almacenamiento de la consulta realizada		6	
Recorrido de todos los datos almacenados		7	
TOTAL (100%)		100	

6 Lenguaje de almacenamiento, CHISON

Esta sección permitirá evaluar la correcta implementación del lenguaje de almacenamiento de la base de datos. Se entregará un archivo chison y se realizarán 8 consultas para evaluar la correcta recuperación de los datos. Las consultas se realizarán tanto en el modo principiante como en el modo avanzado para validar si se obtiene el mismo resultado.

6.1 Consulta 1

Permite eliminar un conjunto de alumnos que cumplan con condiciones planteadas.

6.2 Consulta 2

Modificará los nombres y edades de los alumnos que cumplan con las condiciones planteadas.

6.3 Consulta 3

Ingresara un nuevo estudiante únicamente utilizando su nombre y carné.

6.4 Consulta 4

Obtendrá el nombre, carnet y cursos aprobados de los 10 estudiantes que cumplan con los requerimientos planteados.

6.5 Consulta 5

Obtendrá las ultimas goleadas en partidos amistosos.

6.6 Consulta 6

Obtendrá las marcas de los pantalones que son vendidos en talla 32 pero no en 31 y tienen color verde disponible.

6.7 Consulta 7

Llamada al procedimiento OrdenarLista que permitirá ordenar una lista e imprimir los valores en la consola.

6.8 Consulta 8

Llamada al procedimiento eliminarPrimo que elimina una serie de números primos y retorna los resultados

Tiempo estimado para la evaluación de esta sección: **10 minutos.**

DESCRIPCION	Ponderación		Puntaje obtenido
Implementación con funciones TCL	20		
Almacenamiento con sentencia commit		9	
Recuperación con sentencia Rollback		11	
Archivo 1	80		

Consulta 1		5	
Ejecución en Blockly		3	
Ejecución en Editor		2	
Consulta 2		5	
Ejecución en Blockly		3	
Ejecución en Editor		2	
Consulta 3		8	
Ejecución en Blockly		4	
Ejecución en Editor		4	
Consulta 4		8	
Ejecución en Blockly		4	
Ejecución en Editor		4	
Consulta 5		10	
Ejecución en Blockly		5	
Ejecución en Editor		5	
Consulta 6		12	
Ejecución en Blockly		5	
Ejecución en Editor		7	
Consulta 7		15	
Ejecución en Blockly		5	
Ejecución en Editor		10	
Consulta 8		17	
Ejecución en Blockly		5	
Ejecución en Editor		12	
TOTAL (100%)		100	

7 Lenguaje Unificado de Paquetes (LUP)

En esta sección se evaluarán el correcto uso y estructura de cada paquete enviado y recibo desde el servidor.

Para la calificación de esta sección será necesario que en la interfaz gráfica del cliente exista 2 secciones de visualización, para poder revisar los paquetes enviados y recibidos, en caso no se tenga estas secciones de visualización **NO SE CALIFICARÁ**, debido a que no habrá forma de verificar que se esté usando de forma correcta este lenguaje.

Tiempo estimado para la evaluación de esta sección: **10 minutos**.

DESCRIPCION	Ponderación	Puntaje obtenido
-------------	-------------	------------------

Sentencias LUP	90		
Paquete inicio de sesión		15	
Envío del cliente al servidor		7.5	
Respuesta del servidor		7.5	
Paquete de fin de sesión		15	
Envío del cliente al servidor		7.5	
Respuesta del servidor		7.5	
Paquete de consulta		15	
Envío del cliente al servidor		7.5	
Respuesta del servidor		7.5	
Paquete de datos		10	
Paquete de error		10	
Paquete de mensaje		10	
Paquete de estructura		15	
Envío del cliente al servidor		7.5	
Respuesta del servidor		7.5	
Área de visualización de paquete enviado	5		
Área de visualización de paquete recibido	5		
TOTAL (100%)		100	

8 Excepciones

En esta sección se evaluarán las validaciones de semántica definidas en el enunciado, en general serán excepciones.

Se contará con un archivo que contendrá las instrucciones necesarias para generar cada una de las excepciones mencionadas a continuación:

Tiempo estimado para la evaluación de esta sección: **10 minutos**.

DESCRIPCION	Ponderación		Puntaje obtenido
Excepciones	100		
TypeAlreadyExists		4	
Dentro de try		2	
Fuera de try		2	
TypeDontExists		4	
Dentro de try		2	
Fuera de try		2	
BDAAlreadyExists		4	
Dentro de try		2	
Fuera de try		2	

BDAAlreadyExists		4	
Dentro de try		2	
Fuera de try		2	
BDDontExists		4	
Dentro de try		2	
Fuera de try		2	
UseBDEXception		4	
Dentro de try		2	
Fuera de try		2	
TableAlreadyExists		4	
Dentro de try		2	
Fuera de try		2	
TableDontExists		4	
Dentro de try		2	
Fuera de try		2	
CounterTypeException		4	
Dentro de try		2	
Fuera de try		2	
UserAlreadyExists		4	
Dentro de try		2	
Fuera de try		2	
UserDontExists		4	
Dentro de try		2	
Fuera de try		2	
ValuesException		4	
Dentro de try		2	
Fuera de try		2	
ColumnException		4	
Dentro de try		2	
Fuera de try		2	
BatchException		4	
Dentro de try		2	
Fuera de try		2	
IndexOutOfRangeException		6	
Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
ArithmeticException		6	
Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
NullPointerException		6	

Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
NumberReturnsException		6	
Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
FunctionAlreadyExists		6	
Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
ProcedureAlreadyExists		6	
Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
ObjectAlreadyExists		6	
Dentro de try		2	
Fuera de try		2	
Lanzado por sentencia throw		2	
Log de exception.message		2	
TOTAL (100%)		100	

9 Errores léxicos y sintácticos

Se **evaluará la detección de un error inducido**, se debe detener la ejecución y mostrar el reporte de error.

9.1 Errores léxicos

Se **evaluará la detección de un error inducido** es importante tener en cuenta que se elegirá aleatoriamente un símbolo que no pertenezca al lenguaje y se debe reportar el mismo. Para la sección de recuperación se introducirán dos o más errores léxicos separados por el Token centinela para verificar que el estudiante haya implementado la recuperación de errores en modo pánico.

9.2 Errores sintácticos

Se **evaluará la detección de un error inducido** es importante tener en cuenta que se elegirá aleatoriamente un error de estructura que represente un error sintáctico y se debe reportar el mismo. Para la sección de recuperación se introducirán dos o más errores sintácticos separados por el Token centinela para verificar que el estudiante haya implementado la recuperación de errores en modo pánico.

Tiempo estimado para la evaluación de esta sección: **10 minutos**.

DESCRIPCION	Ponderación		Puntaje obtenido
Errores léxicos	50		
Prueba 1 (a discreción del tutor)		25	
Detección		10	
Recuperación		15	
Prueba 2 (a discreción del tutor)		25	
Detección		10	
Recuperación		15	
Errores sintácticos	50		
Prueba 1 (a discreción del tutor)		25	
Detección		10	
Recuperación		15	
Prueba 2 (a discreción del tutor)		25	
Detección		10	
Recuperación		15	
TOTAL (100%)		100	

10 Documentación

En esta sección se calificará que las funcionalidades de la aplicación se hayan implementado correctamente. Para cada punto a calificar se realizarán las acciones indicadas y se deberá comprobar, dentro de la aplicación, que la funcionalidad cumple correctamente con lo esperado.

10.1 Manual de usuario

En un manual de usuario se debe definir todo lo necesario para que el usuario comprenda y pueda ejecutar la aplicación de forma correcta, lo mínimo que debe contener este manual de usuario, y lo que se calificará, es:

10.1.1 Pasos para las modalidades de CQL-Teacher

Como parte de la aplicación, esta tiene distintas modalidades y cada uno con diferente manejo de funcionalidades, se debe describir como maneja la aplicación estas modalidades, como comenzar cada una y las funcionalidades aplicadas a tal modalidad.

10.1.2 Pasos para armar una entrada a partir de bloques

Una de las funcionalidades más innovadoras de la aplicación son los bloques en la interfaz para armar un archivo de entrada, se debe describir como armar una entrada únicamente con bloques, cual es alcance de los bloques en la modalidad aplicada y descripción de cada uno de todos los bloques existentes.

10.1.3 Instrucción de cada modalidad

Se debe definir instrucciones específicas para que el usuario pueda manejar cada modalidad en su totalidad, esto debe de incluir todas las instrucciones CQL que puede utilizarse seguido de un ejemplo de un archivo de entrada aplicado en cada modalidad.

10.1.4 Flujo de la aplicación

Se debe definir el flujo de la aplicación de forma general, como es que obtiene la información, que información puede obtener de la base de datos y como es que maneja los entornos de ejecución la aplicación.

10.2 Manual técnico

En un manual técnico se debe definir toda la arquitectura de la aplicación y debe de incluir múltiples diagramas para una mejor comprensión de la distribución del código, lo mínimo que debe contener este manual técnico, y lo que se calificará, es:

10.2.1 Descripción de aplicación requerimientos mínimos

Esta sección describe el flujo de la aplicación y sus principales componentes, además también debe describir todos los requerimientos mínimos para que la aplicación y el servidor puedan ser levantados y modificados.

10.2.2 Diagrama de clases

Debe ser un diagrama de clases UML que contenga las clases principales de la aplicación, cada clase con sus atributos y funciones.

10.2.3 Diagrama de paquetes y herencia

Debe ser un diagrama de paquetes que represente la organización de paquetes y cada uno de sus elementos, además también, debe incluir las relaciones conforma a la herencia aplicada.

10.2.4 Descripción de clases y métodos principales

Se debe definir cada clase y método principal del proyecto, esto con finalidad de entender las principales características y como se van relacionando.

10.2.5 Explicación de acciones semánticas

Debe ser una explicación general de cómo se organizó y se ejecutó la semántica en los archivos de entrada.

10.3 Gramáticas

Una de las características principales en este proyecto es la definición de gramáticas, en esta sección se verificará la correcta implementación de estas y se realizará preguntas estratégicas para verificar que el estudiante maneja el tema.

Tiempo estimado para la evaluación de esta sección: **10 minutos.**

DESCRIPCION	Ponderación	Puntaje Obtenido
Manual de usuario	24	
Pasos para entrar a cualquier modo de CQL-Teacher	4	
Pasos para armar una entrada a partir de bloques	4	
Instrucciones para modo principiante	4	
Instrucciones para modo intermedio	4	

Instrucciones para modo avanzado	4	
Flujo de aplicación	4	
Manual técnico	30	
Descripción de aplicación y requerimientos mínimos	5	
Diagrama de clases	5	
Diagrama de paquetes o de herencia	5	
Descripción de herramientas utilizadas	5	
Descripción de clases y métodos principales	5	
Explicación de acciones semánticas	5	
Gramáticas	46	
Gramática de CQL (todos los lenguajes)	16	
Pregunta sobre la TDS utilizada	8	
Pregunta sobre la gramática	8	
Gramática de Chison	10	
Gramática LUP (servidor)	10	
Gramática LUP (cliente)	10	
TOTAL (100%)	100	

Restricciones:

- El proyecto es individual.
- El proyecto deberá funcionar con los archivos que sean proveídos por lo auxiliares para la calificación, sin modificación alguna, si se modifica algún archivo, no se tendrá acceso a ningún porcentaje de la nota correspondiente a la sección que represente.
- No será permitido compartir los archivos de entrada durante ni después de la calificación.
- La calificación del proyecto será personal y durará como máximo 120 minutos.
- No pueden estar personas ajenas a la calificación presentes en la misma, de lo contrario no se calificará el proyecto.
- Si el incumplimiento de alguna de las anteriores restricciones o bien alguna anomalía detectada durante la calificación los tutores académicos finalizaran la calificación y anularan la nota obtenida hasta el momento, seguido se notificará a los catedráticos y si fuese necesario a la Escuela de Ingeniería en Ciencias y Sistemas.

Consideraciones:

- La calificación se realizará presencialmente y sobre los archivos ejecutables enviados, si la ejecución de estos produce alguna excepción no controlada por el estudiante, no se calificará ya que es su responsabilidad entregar ejecutables probados.
- Únicamente se tomará como correcta la salida, si es exactamente la esperada, de existir discrepancia, por mínima que sea se tendrá una nota de cero puntos.
- Anomalías o copias detectadas de proyectos tendrán de manera automática una nota de cero puntos en el laboratorio y los involucrados serán reportados a la Escuela de

Ingeniería en Ciencias y Sistemas, para que se apliquen las sanciones correspondientes.

- Anomalías detectadas en los archivos entregables tendrá de manera automática una nota de cero puntos, por ejemplo: no se envió código el código correcto, se envió parte del código y no el código completo, archivos ajenos a los entregables del proyecto, no se hizo uso de las herramientas descritas en el enunciado de cada proyecto, entre otras.
- Los archivos de entrada contendrán errores semánticos, sintácticos y léxicos para la verificación de recuperación de errores de la aplicación.