

# MANUAL TÉCNICO

## **SOLITARIO PRACTICA 1 DE ESTRUCTURAS DE DATOS**

### **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 C++ y GCC

Es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C y añadir mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C + + es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumaron a los paradigmas de programación estructurada y programación orientada a objetos. Por esto se suele decir que el C + + es un lenguaje de programación multiparadigma.

### ● CLION

CLion es un entorno de desarrollo integrado (IDE) para desarrollar software informático escrito en C y C++. Está desarrollado por JetBrains (anteriormente conocido como IntelliJ) y está disponible en una edición comercial patentada.

### ● GITHUB

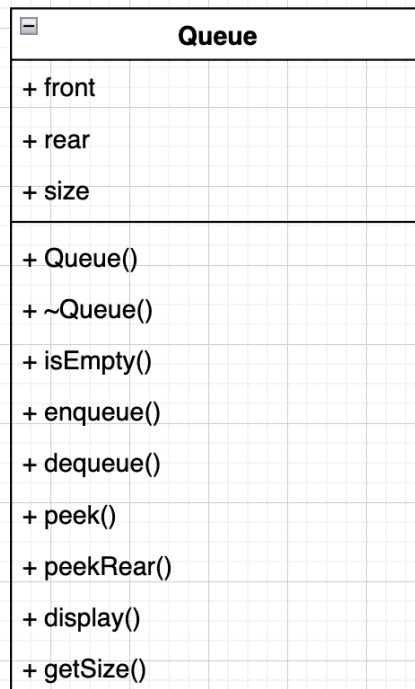
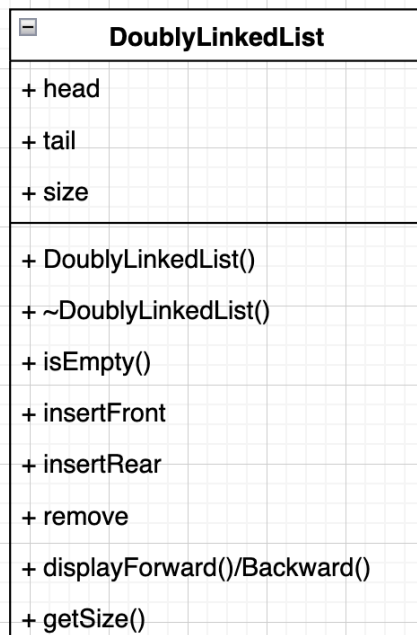
Es un sistema de control de versiones de código y gestión de proyectos, a su vez también funciona como una plataforma de estilo red social diseñada para desarrolladores para poder compartir código entre más personas y colaborar en el mismo.


### ● 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



 <b>CardsHandler</b>
+ cards + randomizedCards + firstQueue + secondQueue + piles + lists - randomizer
CardsHandler() ~CardsHandler() + printCardsArray + prepareGame + prepareStructures() + handleUserChoice() + showMainGame() + seeCard()

 <b>MainGame</b>
+ cardHandler
+ startPlaying()

 <b>Randomizer</b>
+ randomNums
+ generateRandomNumbers + isAlreadyUsed + getRandomNumber + Randomizer

# CLASES Y ALGORITMOS

## **CardsHandler:**

Esta clase se encarga de manejar todo lo respectivo a las cartas generadas dentro del juego, acciones justamente como, generar las cartas, generarlas en forma aleatoria y asignarlas a las colas y listas respectivas.

## **Randomizer:**

Esta clase se encarga como utilidad, de generar números aleatorios y colocarlos dentro de un arreglo fijo de hasta 52 elementos y se puede acceder para obtener un numero aleatorio.

## **Structures:**

Dentro de esta carpeta se encuentran todas las estructuras implementadas para este programa, específicamente, los nodos de cartas, y colecciones de elementos: Pila, Cola y lista doblemente enlazadas, y estas se encargan de ir guardando todos los elementos que el programa vaya requiriendo durante la ejecución.

# INSTRUCCIONES PARA EJECUCIÓN

Para ejecutar este programa solamente es necesario tener un compilador del lenguaje C++ y el programa "make". Luego es necesario estar en el directorio del proyecto desde una terminal y ejecutar el comando "make" y dejará un ejecutable llamado "program" que puede ser ejecutado de forma normal dentro de la misma terminal.