**MANUAL TÉCNICO**

**JUEGO BLACK JACK**

**ANDRÉS DAVID OCAMPO**

**ELIANA JANNETH PUERTA MORALES**

**UNIVERSIDAD DE ANTIOQUIA**

**INGENIERÍA DE SISTEMAS**

**MEDELLÍN**

**2021**

**CONTENIDO**

[**OBJETIVO** 2](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705837)

[**REQUERIMIENTOS TÉCNICOS** 2](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705838)

[**REQUERIMIENTOS MÍNIMOS DE HARDWARE** 2](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705839)

[**REQUERIMIENTOS MÍNIMOS DE SOFTWARE** 2](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705840)

[**HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO** 2](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705841)

[**JAVA** 2](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705842)

[**CÓDIGO** 3](https://udeaeduco-my.sharepoint.com/personal/eliana_puerta_udea_edu_co/Documents/UDEA/SEMESTRE%20lll/Logica%20y%20Representación%20ll/Practica%201/Manual%20Tecnico%20Picas%20y%20Fijas.docx#_Toc70705843)

# **OBJETIVO**

Informar y especificar al usuario la estructura y conformación del sistema con el fin de que puedan hacer soporte y modificaciones o actualizaciones al sistema en general.

# **REQUERIMIENTOS TÉCNICOS**

### **REQUERIMIENTOS MÍNIMOS DE HARDWARE**

* Procesador: Core
* Memoria RAM: Mínimo: 1 Gigabytes (GB)
* Disco Duro: 500Gb.

### **REQUERIMIENTOS MÍNIMOS DE SOFTWARE**

* Sistema Operativo: Windows 7 en adelante
* Java 15.0
* NetBeans IDE 12.2

# **HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

## **JAVA**

El lenguaje de programación de Java es una herramienta de desarrollo orientada a objetos, fue diseñado para que no dependieran en muchas implementaciones, el cual permite a los desarrolladores ejecutar en cualquier dispositivo sin necesidad de recompilar el código, el cual se considera multiplataforma.

# **CÓDIGO**

Inicialmente se crea la clase Cola con sus respetivos métodos:

Texto

Descripción generada automáticamente

En la clase ListaCartas se crea un ArrayList y se llena con 52 números de 0 al 51 de forma Ramdon:

Texto

Descripción generada automáticamente

Luego en el PrimaryController se crea una cola para encolar los 52 números de forma ramdom y luego ir desencolando de a uno.

Texto

Descripción generada automáticamente con confianza baja

Luego en el botón de jugar se crea el método empezar() que se activan los diferentes controles de la interfaz grafica como los Label, TextFlied, e ImageView y Buttons. Además se generan 52 números del 0 al 52 de forma aleatoria y luego se llena la cola es decir se encolan los 52 números, y finalmente se toma cada imagen y se le asigna un numero que depende del que vaya saliendo al desencolar la cola.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

En el método Suma() se están validando los valores de las cartas de los jugador principal es decir el usuario ya que cuando son Ases se convierten en 11 o 1 dependiendo el caso.

Texto

Descripción generada automáticamente

En el método sumaMesa() se están validando los valores de las cartas de la mesa ya que cuando son Ases se convierten en 11 o 1 dependiendo el caso.

Texto

Descripción generada automáticamente

En el método sumaCp1() se están validando los valores de las cartas del jugador 1 del sistema ya que cuando son Ases se convierten en 11 o 1 dependiendo el caso, y lo mismo pasa con el método sumaCp2() y sumaCp3() y retornan el valor de cada uno respetivamente.

Texto

Descripción generada automáticamente

En el método cartas() recibe un entero que el numero que se va desencolando y luego en un Switch Case se pregunta por el entero y en la variable aux de tipo image se va almacenar la imagen a la cual corresponde el numero por el que se está preguntado.

Texto

Descripción generada automáticamente

En el botón Pedir Carta se crea el método pedirCarta() el cual por medio de un Switch Case dependiendo del valor que tenga la variable contador va desencolando de una carta, sumando el valor que arroje la cola y se manipulan los controles de la interfaz, la variable contador se incrementa cada que el jugador pide una carta. Y en el mismo método lo mismo sucede para cada jugador.

Texto

Descripción generada automáticamente

En el botón Plantarse se crea el método plantarse() se colocan las cartas automáticamente cuando el usuario decide plantarse y esto se realiza por medio de un while y se pregunta a los métodos ya mencionado de las respectivas sumas de la mesa y los jugadores y dependiendo del valor del contador se van desencolando los números para así completar las cartas faltantes hasta llegar a la suma de 21 y dependiendo del valor que arrojen los métodos de las sumas mencionadas por cada jugador se va asignado a cada jugador si perdió, empató o ganó.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

En el botón Doblar se crea el método doblar() si la suma de las cartas es menor a 22 se pueden duplicar los puntos. Inicialmente la variable puntos vale 400 y si pierde se le restan 100 o si se dobla se duplican puntos y se invoca el método de plantarse anteriormente mencionado y allí se suman o restan los puntos según el caso.

Texto

Descripción generada automáticamente