



Universidad del Norte

Ruko y su Mandado

Alvarado Juan – Cuello Francisco – Castilla Julián

Daladier Jabba Molinares

Ingeniería de Sistemas y Computación

Segundo Semestre

Barranquilla, Mayo 12 de 2025

1. Introducción.....	¡Error! Marcador no definido.
2. Requerimientos mínimos .....	¡Error! Marcador no definido.
3. Como instalar el juego? .....	¡Error! Marcador no definido.
4. Objetivos del Juego.....	¡Error! Marcador no definido.
5. Modo de Juego y Controles .....	¡Error! Marcador no definido.
6. Elementos Gráficos.....	¡Error! Marcador no definido.
7. Descripción de las Pantallas del Juego .....	¡Error! Marcador no definido.
8. Código (Backend/Frontend) .....	¡Error! Marcador no definido.
Backend.....	¡Error! Marcador no definido.
Frontend.....	¡Error! Marcador no definido.
9. Algunos pseudocodigos .....	¡Error! Marcador no definido.
10. Conclusiones.....	¡Error! Marcador no definido.
12. Créditos.....	¡Error! Marcador no definido.

## 1. Introducción

El juego "Ruko y su Mandado" es un juego de aventura en el que el jugador controla a **Ruko**, un niño que debe cumplir con el mandado que su madre le encargó. En el camino, Ruko se enfrenta a varios obstáculos y enemigos que intentan impedir que llegue a su destino. A través de este juego, se busca ofrecer una experiencia entretenida y desafiante, combinando elementos de acción y comedia.

## 2. Requerimientos mínimos

La mayoría de los computadores son capaces de correr el juego sin problemas, pero con nuestras pruebas encontramos que lo mínimo que debe tener tu sistema para correr el juego sin ningún problema debería ser:

**CPU: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.10 GHz**

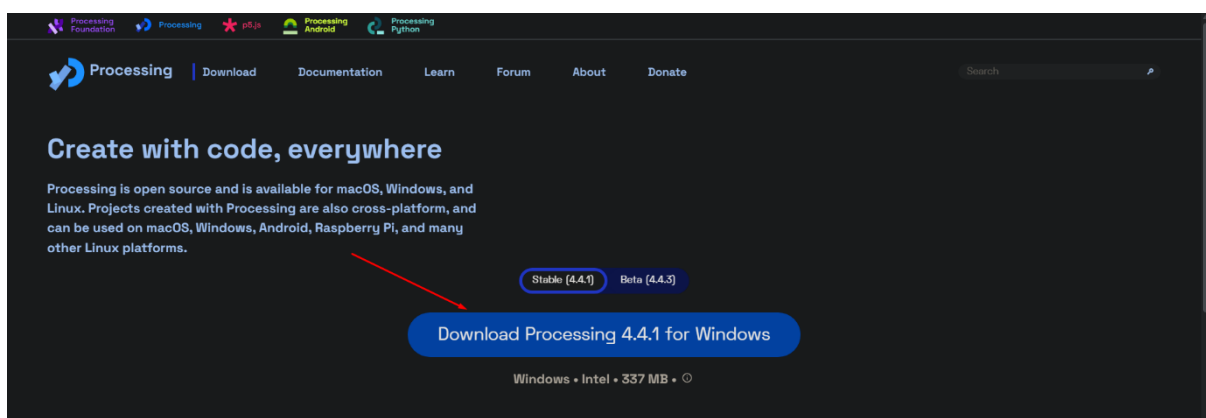
**RAM: 16,0 GB**

**Sistema operativo: Windows 11 (64 Bits)**

## 3. Como instalar el juego?

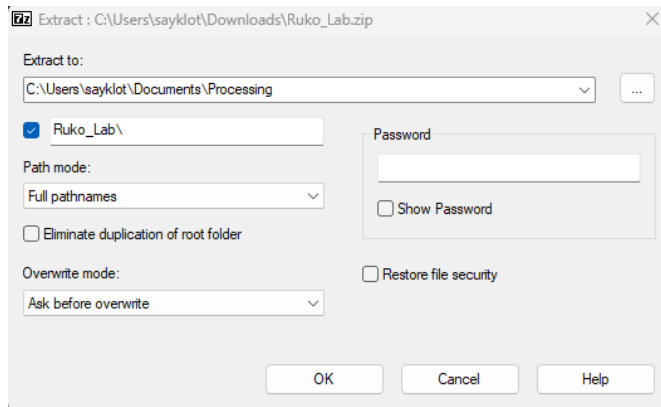
### *Paso 1*

Para poder abrir el juego debes instalar proccesing en tu computador, este es el enlace oficial para su descarga: <https://processing.org/download>



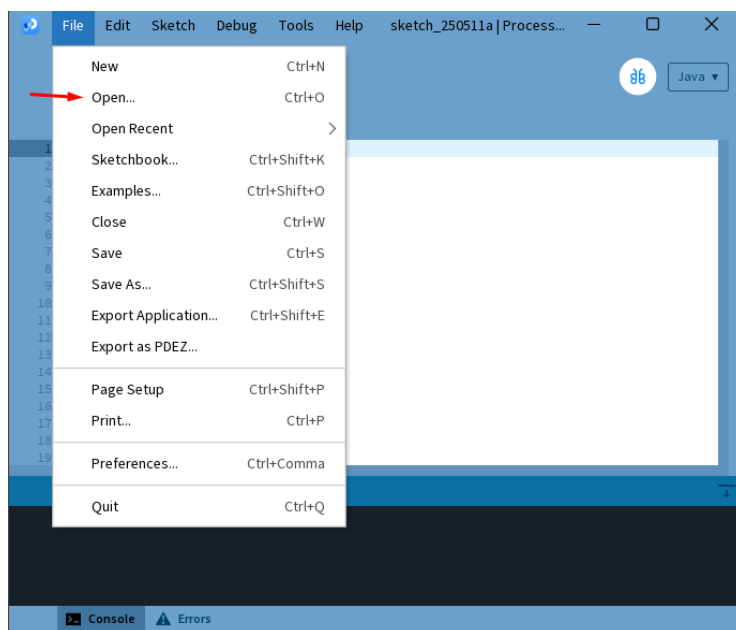
## Paso 2

Después debes descargar el .zip, luego lo extraes preferiblemente en la carpeta de processing que es C:\Users\TuNombreDeUsuario\Documents\Processing.



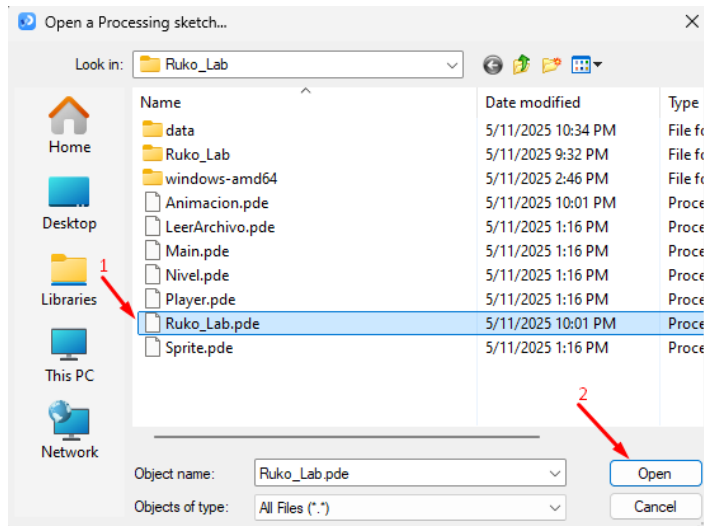
## Paso 3

Luego lo que vas a hacer es abrir processing y darle en donde dice file y luego open como se muestra en la imagen.



#### Paso 4

Después vas a buscar donde extrajiste el .zip y vas a buscar el archivo q dice Ruko\_Lab.pde, y le vas a dar open.



#### Paso 5

Por último, vas a ejecutar el archivo en processing dándole a la flecha tal y como se muestra en la imagen.



## 4. Objetivos del Juego

El objetivo principal de **Ruko** es **recoger el mandado** y llevarlo a la tienda, evitando los obstáculos y enemigos que se presentan en el camino. El juego tiene los siguientes objetivos secundarios:

- **Evitar obstáculos** y enemigos que le restan vidas.
- Ir recolectando ítems para completar su mandado.
- El jugador gana al **completar el mandado** o pierde si se queda sin vidas.

## 5. Modo de Juego y Controles

Controles:

- **Teclas de flechas:** Mueven a Ruko hacia la izquierda, derecha, arriba y abajo.
- **W:** Usado para saltar.
- **Objetivo:** El jugador debe evitar los obstáculos y recoger el mandado para completarlo. El jugador pierde vidas si choca con un enemigo o un obstáculo.
- **Velocidad del Juego:** A medida que el tiempo pasa, la velocidad del juego aumenta, lo que hace que los enemigos y obstáculos sean más difíciles de evitar.

## 6. Elementos Gráficos

El juego cuenta con varios elementos gráficos que incluyen:

Fondo

El fondo del juego muestra un escenario que cambia a medida que Ruko avanza en su misión. La imagen de fondo es clave para crear una atmósfera dinámica. Se dividirá en 3 escenarios para 3 niveles respectivamente.



## Personajes

Los personajes, como **Ruko**, **Sarnoso**, **Chismosa**, y **Carro**, están diseñados en un estilo píxel art, lo que les da un toque único y nostálgico.

- **Imagen de Ruko** : Un niño aficionado al Junior de Barranquilla.



- **Imagen de Sarnoso (Perro)**: El perro enemigo que aparece aleatoriamente en el camino.



-

- **Imagen del Carro:** Un vehículo pixelado que atraviesa la ruta de Ruko.



- **Hombre misterioso:** Este es un hombre misterioso que se aparecerá de la nada y te pondrá a prueba para que demuestres tu habilidad en un juego.



- **Mamá de Ruko:** Está señora fue quien encomendó la misión que inició la aventura

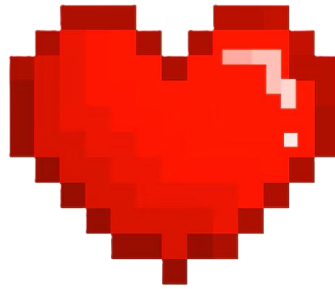




- **Elementos:** Aquí se encuentran los elementos que iras recolectando a medida que avanza el juego, están diseñados en un estilo píxel art, lo que les da un toque único y nostálgico.
  - **Bolsa:** Aquí vas a recolectar todos los ítems que hayas cogido en el transcurso del juego.



- **Corazón:** Tendrás 4 de estos, cada vez que algún enemigo te golpee bajara un corazón, si te quedas sin ninguno pierdes.



## 7. Descripción de las Pantallas del Juego

### 1. Pantalla de inicio

En el menú principal, el jugador verá botones como:

- a. **Iniciar Juego:** Comienza el juego.
  - b. **Opciones:** Info (que te dice la información y la historia del juego), Créditos (donde se le da créditos a los sprites que utilizamos).
  - c. **Salir:** Cierra el juego.
2. **Pantalla de juego:** Una vez que el jugador comienza, verá el camino con los obstáculos (Palo e' sol, Sarnoso, etc.). Deberá avanzar evitando estos obstáculos, recoger el mandado y llegar a la tienda.
  3. **Pantalla de Game Over:** Si el jugador pierde todas sus vidas, aparecerá una pantalla de "Game Over", donde podrá elegir entre reiniciar el juego o salir.

## 8.Código (Backend/Frontend)

**Backend** y **Frontend** son dos componentes fundamentales de cualquier juego o aplicación interactiva. En nuestro caso, el **Backend** se encarga de la lógica del juego, como el movimiento de Ruko, la detección de colisiones y la recolección de objetos. El **Frontend**, por otro lado, se ocupa de la parte visual del juego, como la interfaz de usuario (UI), los botones, y la representación gráfica de los elementos del juego.

## Backend

El **Backend** de nuestro juego maneja la lógica de interacción, como la detección de colisiones entre Ruko y los objetos en el camino, así como el manejo de las vidas y el puntaje. Además, se encarga de crear y mover los objetos en el juego, asegurándose de que todo funcione correctamente a nivel de lógica.

### *Parte del código de Backend:*

```
// CLASES
// =====

class Player {
    boolean moving = false;
    boolean jumping = false;
    float x, y, w, h;
    float vx = 0;
    float vy = 0;
    boolean onGround = false;
    int lives = 4;
    float speed = 8;

    Player() {
        w = 50;
        h = 70;
        x = 50;
        y = groundY - h;
    }

    void update() {
        // — Movimiento horizontal —
        moving = false;
        if (keyPressed) {
            if (key == 'a' || key == 'A') {
                x -= speed;
                if (x < 0) x = 0;
                moving = true;
                facingRight = false;
            }
            if (key == 'd' || key == 'D') {
                x += speed;
                moving = true;
                facingRight = true;
            }
        }
    }
}
```

En este fragmento de código, le damos forma y movimiento al personaje principal del juego. Se define la clase Player, que incluye variables para su posición (x, y), tamaño (w, h), velocidad (vx, vy), y estado (si se está moviendo, saltando o si está en el suelo).

## Frontend

El **Frontend** de nuestro juego se encarga de la creación de la interfaz visual, como el menú y los botones de interacción. Además, gestiona el fondo del juego, la disposición de los botones y la apariencia general del menú.

### *Parte del código de Frontend:*

```
//==BOTONES==  
float boton1X, boton1Y, botonAncho, botonAlto;  
float boton2X, boton2Y;  
float boton3X, boton3Y;  
float boton4X, boton4Y; // Nuevo botón para el Tres en Raya  
  
// Colores de los botones  
color colorBotonNormal = color(200);  
color colorBotonHover = color(150);  
color colorBoton1, colorBoton2, colorBoton3, colorBoton4, colorBoton5; // Color para el nuevo botón  
  
// Textos de los botones  
String textoBoton1 = "Iniciar";  
String textoBoton2 = "Info";  
String textoBoton3 = "Salir";  
String textoBoton4 = "Creditos"; // Texto del nuevo botón  
boolean estadoBoton1 = false;
```

**En este fragmento de código, se configuran los botones del menú del juego.** Se definen las posiciones y dimensiones de cada botón, así como sus colores en estado normal y cuando el cursor pasa sobre ellos.

También se asignan los textos que aparecerán en cada botón, como “Iniciar”, “Info”, “Salir” y “Créditos”.

### *Menu del juego:*



## Información del juego



## 9. Algoritmos importantes

En esta parte se mostrará algunos pseudocodigos que hicimos al inicio como un borrador para lo que iba a hacer la creación final del algoritmo en proccesing, algunos ejemplos q mostraremos son:

### Movimiento del jugador (Ruko)

*Proceso MovimientoRuko*

*Definir velocidadX, velocidadY, gravedad, enSuelo como Real*

*Definir saltando, moviendose como Logico*

*// Inicialización*

```

velocidadX <- 0
velocidadY <- 0
gravedad <- 0.8
enSuelo <- Verdadero

// Lógica de movimiento
Si TeclaPresionada('A') entonces
    velocidadX <- -8
    moviendose <- Verdadero
    mirandoDerecha <- Falso
FinSi

Si TeclaPresionada('D') entonces
    velocidadX <- 8
    moviendose <- Verdadero
    mirandoDerecha <- Verdadero
FinSi

Si TeclaPresionada('W') Y enSuelo entonces
    velocidadY <- -15
    enSuelo <- Falso
    saltando <- Verdadero
FinSi

// Aplicar gravedad
velocidadY <- velocidadY + gravedad

// Actualizar posición
posicionX <- posicionX + velocidadX
posicionY <- posicionY + velocidadY

// Verificar colisión con el suelo
Si posicionY + altura >= suelo entonces
    posicionY <- suelo - altura
    velocidadY <- 0
    enSuelo <- Verdadero
    saltando <- Falso
FinSi
FinProceso

```

## Sistema de vidas

### *Proceso SistemaVidas*

*Definir vidas, tiempoInmunidad como Entero*

*// Inicialización*

*vidas <- 4*

*tiempoInmunidad <- 0*

*// Cuando ocurre una colisión con enemigo*

*Si ColisionaConEnemigo() Y tiempoInmunidad = 0 entonces*

*vidas <- vidas - 1*

*tiempoInmunidad <- 60 // 1 segundo de inmunidad*

*Mostrar "Vidas restantes: ", vidas*

*FinSi*

*// Reducir tiempo de inmunidad*

*Si tiempoInmunidad > 0 entonces*

*tiempoInmunidad <- tiempoInmunidad - 1*

*FinSi*

*// Verificar game over*

*Si vidas <= 0 entonces*

*CambiarEscena(99) // Escena de Game Over*

*FinSi*

*FinProceso*

## Switch Principal en Draw()

### *Proceso DibujarEscena*

*Segun escena Hacer*

*caso -1: // Menú principal*

*MostrarImagen(menu)*

*DibujarBotones()*

*caso 0: // Escena introductoria*

*MostrarDialogo()*

*caso 1: // Juego principal*

*DibujarNivel()*

```

    MoverEnemigos()
    VerificarColisiones()
    caso 3: // Mini-juego Tres en Raya
        DibujarTablero()
        VerificarGanador()
    caso 4: // Escena final
        MostrarDialogoFinal()
    caso 7: // Escena del Vicioso
        MostrarDialogoVicioso()
    caso 99: // Game Over
        MostrarImagen(gameover)
    De Otro Modo:
        // No hacer nada
    FinSegun
FinProceso

```

## Manejo del mouse

```

Proceso ManejarMouse
    // Para menú principal
    Si escena = -1 entonces
        Para cada boton en botones Hacer
            Si MouseClicEn(boton) entonces
                Segun boton.id Hacer
                    caso 1: // Iniciar
                        escena <- 0
                    caso 2: // Info
                        escena <- 2
                    caso 3: // Salir
                        escena <- 99
                    caso 4: // Tres en Raya
                        escena <- 3
                        InicializarJuego()
                    caso 5: // Créditos
                        escena <- 6
                FinSegun
            FinSi
        FinPara
    FinSi

```



```

// Para Tres en Raya
Si escena = 3 entonces
    Si MouseClicEn(celda) Y celdaVacía(celda) entonces
        MarcarCelda(celda, jugadorActual)
        VerificarGanador()
        CambiarJugador()
    FinSi
FinSi
FinProceso

```

## Causas de Muerte/Game Over

### Proceso VerificarGameOver

```

// 1. Perder todas las vidas
Si vidas <= 0 entonces
    GameOver()
FinSi

// 2. No tener suficientes encomiendas al terminar el Tres en Raya
Si juegosTTT >= 3 Y encomiendas < 4 entonces
    GameOver()
FinSi

// 3. Tiempo límite excedido (100 segundos)
Si tiempoTranscurrido > (60 * 100) entonces
    GameOver()
FinSi
FinProceso

```

### Proceso GameOver

```

escena <- 99
MostrarImagen(gameover)
ReproducirSonido(derrota)
FinProceso

```

## Sistema de tres en raya

*Proceso TresEnRaya*

*Definir tablero[3,3] como Entero*

*Definir jugadorActual, juegosJugados como Entero*

*Definir juegoTerminado como Logico*

*// Inicialización*

*jugadorActual <- 1*

*juegosJugados <- 0*

*juegoTerminado <- Falso*

*// Lógica del juego*

*Mientras juegosJugados < 3 Y NO juegoTerminado Hacer*

*Si MouseClicEn(celda) Y celdaVacía(celda) entonces*  
*tablero[celda.fila, celda.col] <- jugadorActual*

*Si VerificarGanador(jugadorActual) entonces*

*juegoTerminado <- Verdadero*

*Si jugadorActual = 1 entonces*

*encomiendas <- encomiendas - 1*

*FinSi*

*Sino Si TableroLleno() entonces*

*juegoTerminado <- Verdadero*

*Sino*

*jugadorActual <- 3 - jugadorActual // Alternar entre 1 y 2*

*FinSi*

*FinSi*

*FinMientras*

*// Después de 3 juegos*

*Si encomiendas >= 4 entonces*

*CambiarEscena(4) // Escena final*

*Sino*

*GameOver()*

*FinSi*

*FinProceso*

## 10. Conclusiones

El desarrollo de *"Ruko y su Mandado"* ha sido finalizado con éxito. El juego cuenta con un menú principal totalmente funcional que permite al jugador iniciar la partida, acceder a opciones y salir del juego, todo dentro de una interfaz clara, fluida y fácil de navegar.

Durante el desarrollo, se implementaron todos los elementos esenciales de la jugabilidad: el movimiento completo de Ruko, incluyendo saltos y desplazamientos, la interacción con enemigos y obstáculos, el sistema de vidas representado por corazones, la recolección de encomiendas, y un límite de tiempo que añade presión al jugador. También se integró un minijuego de tres en raya y condiciones específicas que pueden llevar al jugador a perder la partida.

Visualmente, el juego presenta escenarios animados, un fondo desplazable, personajes con animaciones por frames y pantallas para cada etapa del juego, como la introducción, la pantalla de juego, el final narrativo, el game over y los créditos. Cada uno de estos elementos fue diseñado para brindar una experiencia atractiva y coherente.

*"Ruko y su Mandado"* combina narrativa, jugabilidad y diseño visual para ofrecer una experiencia divertida y completa. Este proyecto representa un gran logro en cuanto a planificación, integración de mecánicas y desarrollo de videojuegos, cumpliendo con todos los objetivos establecidos desde su concepción hasta su versión final.

## 11. Créditos

- Ruko inspiración gif: <https://www.artstation.com/artwork/aYe1ZJ> creadora Rada Ursu
- Carro gif: <https://www.artstation.com/artwork/PeoANo> creador Pixinho
- Perro gif: <https://co.pinterest.com/pin/330029478964214305/> creador SHOPJEEN
- Fondo png: <https://www.dreamstime.com/pixel-bit-game-level-design-art-modern-city-buildings-panorama-high-quality-illustration-image324606490> creadora Anna Khomchenko