

**Presentación de Concepto: Juego sobre principios de la informática**

*Integrantes: Juan Diego Mateus, Emilio Aldean y Jairo Perez*

## 

## 

## 

## 

## 

## 

## 

## 

## **Tabla de contenidos:**

[**Tabla de contenidos: 2**](#_26in1rg)

[Introducción 3](#_35nkun2)

[Descripción del Proyecto 3](#_44sinio)

[Objetivos del Proyecto 3](#_z337ya)

[Minijuego: Protocol Patrol 4](#_1y810tw)

[Descripción del Juego 4](#_4i7ojhp)

[Cambios Clave para el Multijugador: 5](#_2xcytpi)

[Tecnologías Utilizadas: 5](#_1ci93xb)

[Minijuego: BashAhead 6](#_2bn6wsx)

[Descripción del Juego 6](#_qsh70q)

[Componentes y Funcionalidades: 7](#_3as4poj)

[Tecnologías Utilizadas: 7](#_1pxezwc)

[Minijuego: The Imitation Race 8](#_2p2csry)

[Descripción del Juego 8](#_147n2zr)

[Características del Juego: 8](#_3o7alnk)

[Tecnologías Utilizadas: 9](#_23ckvvd)

[Conclusión 10](#_32hioqz)

**Especificación de Requerimientos de Software**

### **Introducción**

Este documento detalla la explicación y funcionamiento de un proyecto diseñado para el ámbito educativo, dirigido a estudiantes universitarios. El objetivo principal es brindar una experiencia interactiva y entretenida que facilite la enseñanza de conceptos de programación y el uso de terminales de manera competitiva. El proyecto busca mejorar la enseñanza académica mediante métodos de estudio más dinámicos y prácticos, promoviendo el desarrollo de habilidades necesarias para estudiantes de nivel superior.

### **Descripción del Proyecto**

El proyecto consiste en el desarrollo de múltiples minijuegos y desafíos que abordan conceptos fundamentales de la programación y el uso de terminales como el **Command Prompt** de Windows o terminales en sistemas basados en Unix. Estos minijuegos se distribuyen equitativamente entre dos dispositivos participantes, generando una competencia para ver quién resuelve más rápidamente los problemas planteados. Cada minijuego otorga puntos al usuario que lo resuelva correctamente, y el primero en alcanzar tres puntos será el ganador. El proyecto se desarrollará principalmente en **Python**, utilizando herramientas como **PyQt5** para la interfaz gráfica y **sockets** para la conectividad entre jugadores.

### **Objetivos del Proyecto**

* **Mejorar la enseñanza universitaria** a través de métodos más interactivos y dinámicos.
* **Fomentar el pensamiento lógico y la resolución de problemas** en un entorno competitivo.
* **Introducir a los estudiantes en conceptos avanzados de programación** y el manejo de terminales.
* **Facilitar la familiarización con el uso de terminales** en sistemas operativos.
* **Despertar el interés por la programación** en un ambiente desafiante y estimulante.

### **Minijuego: Protocol Patrol**

### **Descripción del Juego**

**Protocol Patrol** es un juego temático de ciberseguridad donde los jugadores actúan como administradores de red, gestionando paquetes de red entrantes en tiempo real. La versión multijugador permite a dos usuarios competir por ver quién maneja los paquetes más rápida y correctamente. Cada jugador debe identificar y bloquear los paquetes maliciosos mientras intenta superar la puntuación de su oponente. La conectividad es directa, usando una configuración **peer-to-peer (P2P)**, sin necesidad de un servidor central.



### **Cambios Clave para el Multijugador:**

1. **Conexión Cliente-Servidor :** Ambos jugadores se conectarán directamente a un servidor través de sockets en Python. El servidor manejara la logica del juego .Las acciones se sincronizan en tiempo real, y los jugadores intercambian decisiones y puntuaciones.
2. **Actualizaciones en Tiempo Real:** Las acciones de cada jugador, como el manejo correcto o incorrecto de paquetes, se transmiten en tiempo real al otro jugador.
3. **Paquetes Sincronizados:** Ambos jugadores reciben un conjuntos de paquetes similar para actualizar su progreso, asegurando igualdad en los desafíos presentados.
4. **Manejo de Latencia:** La latencia entre las conexiones debe gestionarse para sincronizar correctamente las acciones y evitar desventajas competitivas.
5. **Condiciones de Fin de Juego:** El juego finaliza después de 90 segundos o cuando un jugador alcanza una puntuación predefinida, mostrándose las puntuaciones finales de ambos.

### **Tecnologías Utilizadas:**

1. **Python (Módulo Socket):** Para la conexión cliente servidor de los jugadores al cliente.
2. **PyQt5:** Se usa para desarrollar la interfaz gráfica y la lógica del juego.
3. **Programación Orientada a Objetos (OOP):** Para manejar los eventos del juego y la comunicación entre jugadores.
4. **MongoDB/JSON:** Para almacenar las puntuaciones y gestionar la tabla de clasificación.
5. **Módulo Random:** Para la generación sincronizada de paquetes en ambos dispositivos.

### **Minijuego: BashAhead**

### **Descripción del Juego**

**BashAhead** es un simulador multijugador diseñado para enseñar el uso básico de la terminal **Bash** de Linux. Dos jugadores compiten para navegar un sistema de archivos simulado y eliminar un archivo de "malware" oculto. Este juego no requiere de un servidor central, sino que utiliza una conexión directa entre los jugadores.



### **Componentes y Funcionalidades:**

1. **Interfaz Gráfica (PyQt5):** La pantalla dividida muestra una ventana de terminal para la interacción del jugador y un panel de referencia que rastrea el progreso del oponente.
2. **Conexión Multijugador:** El juego sincroniza las acciones clave entre los jugadores a través de una conexión a un servidor.
3. **Simulación de Terminal:** La ventana de terminal emula comandos básicos de Bash, permitiendo a los jugadores ejecutar operaciones de navegación y manipulación de archivos.
4. **Simulación de Sistema de Archivos:** El sistema de archivos virtual está diseñado para desorientar a los jugadores. Cada uno tiene su propia copia, pero las acciones clave de un jugador afectan al otro.

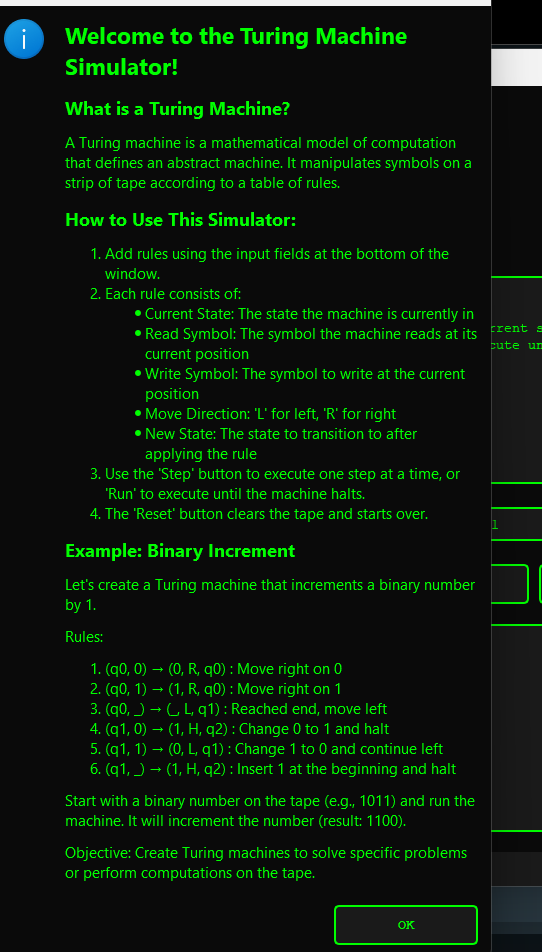
### **Tecnologías Utilizadas:**

1. **Python:** Lenguaje principal del proyecto.
2. **PyQt5:** Utilizado para desarrollar la interfaz gráfica.
3. **Sockets:** Para establecer la conexión cliente servidor entre los jugadores.
4. **OOP:** Para estructurar la lógica del juego.
5. **Simulación de Archivos:** Para manejar los comandos de terminal y simular la estructura de archivos.

### **Minijuego: The Imitation Race**

### **Descripción del Juego**

**The Imitation Race** es una simulación competitiva de una **Máquina de Turing**, transformada en una carrera 1v1. Los jugadores compiten para completar un desafío computacional basado en reglas de una máquina de Turing. El objetivo es diseñar reglas eficientes que manipulen una cinta de símbolos y resolver el problema antes que el oponente.



### **Características del Juego:**

* **Carrera 1v1:** Los jugadores compiten simultáneamente con reglas y cintas propias.
* **Simulación de Máquina de Turing:** Los jugadores definen reglas que controlan el movimiento y la lectura/escritura de símbolos en la cinta.
* **Visualización Interactiva:** Los jugadores observan en tiempo real cómo sus reglas afectan la cinta de símbolos.
* **Ejecución Paso a Paso o Completa:** Los jugadores pueden avanzar manualmente o ejecutar la máquina hasta el final de manera automática.

### **Tecnologías Utilizadas:**

1. **Python:** Para la lógica del juego.
2. **PyQt5:** Para la creación de la interfaz gráfica.
3. **Multijugador en Pantalla Dividida:** Cada jugador controla su máquina de Turing en su propia sección de la pantalla.
4. **Lógica de Máquina de Turing:** Definida por una clase que simula el comportamiento de la máquina en función de las reglas introducidas por los jugadores.
5. **Sincronización y Verificación:** El juego verifica continuamente el progreso de cada jugador para declarar al ganador cuando se cumplan las condiciones.



### **Conclusión**

Este proyecto integra varios elementos fundamentales de la programación y las redes en un entorno educativo competitivo para estudiantes universitarios. Los diferentes minijuegos, como **Protocol Patrol**, **BashAhead** y **The Imitation Race**, no solo sirven como herramientas de aprendizaje, sino que también fomentan el pensamiento crítico y la resolución de problemas en un contexto interactivo. La implementación de tecnologías como **Python**, **PyQt5**, y **sockets** asegura que el proyecto sea tanto funcional como escalable para entornos universitarios, ofreciendo una experiencia educativa integral y estimulante.