

METODOLOGÍA DE LA INVESTIGACIÓN

DESARROLLO DE UN VIDEOJUEGO

MY TA TE TI

Mayra Magiarate Funes

ÍNDICE

1. Introducción
2. Objetivos.
3. La idea.
4. Proceso de Investigación (Planeamiento, ejecución y comunicación).
5. Recursos utilizados.
6. Método Utilizado
7. Resultados
8. Conclusiones

Introducción

La materia Metodología de la investigación, dictada por el profesor Carlos Rodríguez en la Tecnicatura Superior en Programación de la Universidad Tecnológica Nacional, nos introduce en el campo de la investigación, en ella aprendemos las etapas del método científico, las distintas formas de investigar y crear Hipótesis, así como también de una manera más empírica a cómo implementar metodologías para desarrollar nuestra propia investigación la cual aplicaremos luego al desarrollo en la parte de programación de un videojuego.

Nuestro Marco teórico quedará limitado entonces o definido por las técnicas a implementar en la programación del videojuego, la idea de videojuego, el lenguaje elegido para su desarrollo, los procedimientos y la recolección de datos.

Objetivos:

- Desarrollar un videojuego sencillo, que nos permita aplicar lo aprendido en programación como así también ampliar mediante investigación nuestros conocimientos.
- Utilizar software libre para el desarrollo del videojuego.
- Utilizar un lenguaje y una plataforma que permita que el videojuego corra en distintos S.O.
- Desarrollar un tipo de Inteligencia artificial en modo vs con la computadora.
- Implementar GitHub.
- Crear un repositorio Remoto donde se irán guardando los cambios y el avance del proyecto.
- Colaborar dicho repositorio al jefe de la Cátedra para su supervisión.

La Idea

Nos planteamos la posibilidad de desarrollar un juego sencillo como lo es el Tateti, el porqué de ello simplemente es que recientemente nos hemos familiarizado con la biblioteca gráfica de Java, llamada Swing, resultando de fácil implementación y generadora de código automático.

Teniendo en cuenta que:

El tateti es un juego que se juega de a dos (dos personas físicas, o una persona física y una virtual); y que dispone de un tablero que está dividido en tres filas y tres columnas, esto sería un cuadrado que contiene 9 cuadrados. Cada jugador elige entre una "X" o una "O" que lo represente, está será la marca que lo identifique (Para este juego, el primer jugador será siempre el Humano y su marca distintiva será la X). El juego es por turnos, juega uno luego el otro y consiste básicamente (objetivo) en lograr tener una fila, una columna o una diagonal de marcas iguales, o sea lograr tener tres en línea de una misma marca antes que el oponente.

Procedemos entonces al desarrollo del proyecto.

Proceso de investigación

Etapa 1:

1- Planeamiento de acuerdo a la dimensión Epistemológica.

- a) Concebir la idea a investigar: Desarrollar un Tatetí virtual, un tres en línea propio.
- b) Plantear el Problema a investigar: Dibujar un tablero de 3x3 tipo matriz que contenga las marcas elegidas por cada jugador o para cada jugador según su jugada, con contadores, tipos de jugadores (Humano-Máquina), métodos para Jugadas ganadoras, vaciado de tablero, Jugadas de la Máquina, salidas y vueltas al juego.

Elaboración del Marco teórico: Lenguaje de programación Java; entorno de desarrollo integrado: Netbeans. Biblioteca java Swing y standard.

2- Planeamiento de acuerdo a la dimensión Estratégica.

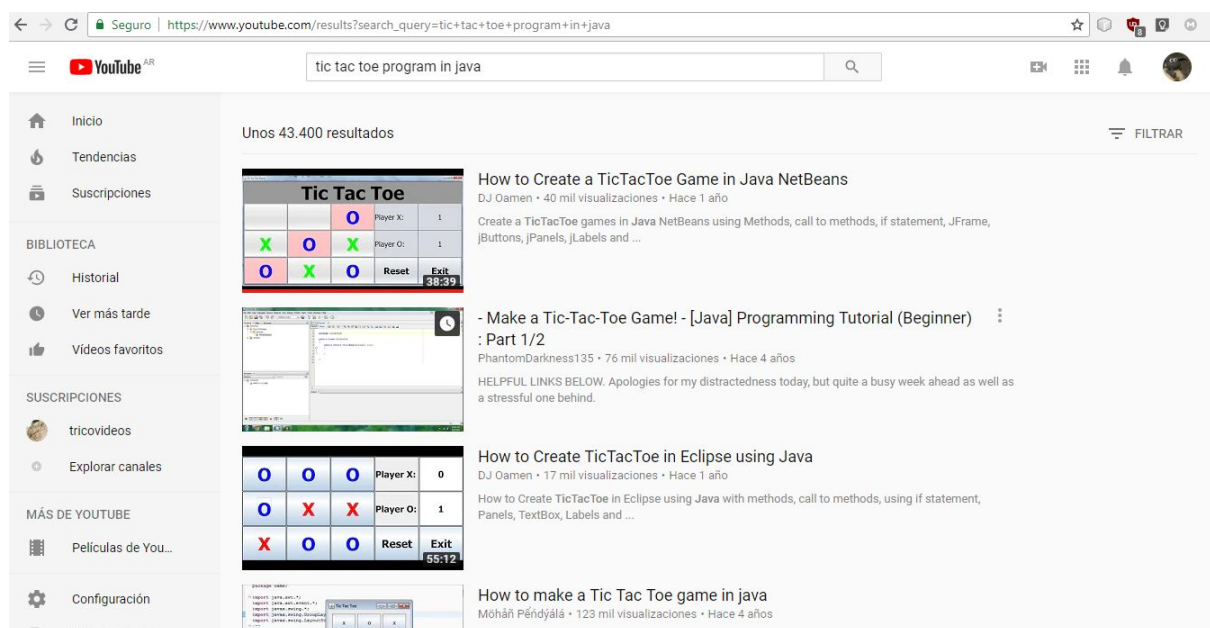
- a) Definir la Investigación: Nuestra investigación será EXPLORATORIA (puesto que al desconocer el cómo del desarrollo del videojuego, nuestra primera etapa consiste netamente en la búsqueda de información, de los cómo y de las existencias de similares y del conocimiento de su funcionamiento).

Etapa 2:

1-Ejecución Según la dimensión Estratégica.

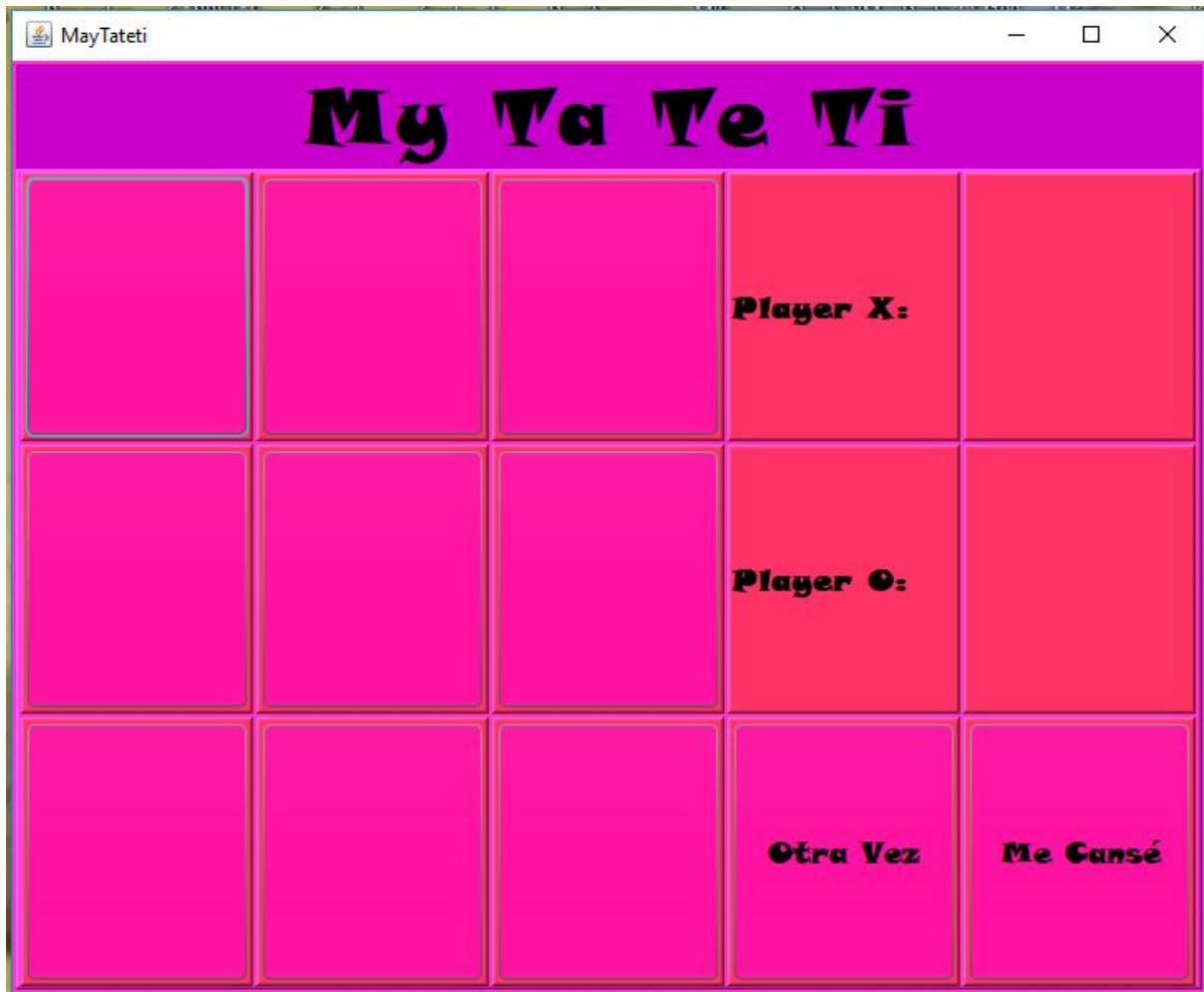
- a) Establecimiento de la Hipótesis: ¿Es posible crear un Tatetí virtual desarrollado netamente en java que nos permita jugar de humano a humano mediante una computadora y también de modo humano-máquina?

Las búsquedas realizadas por nosotros en Google nos dicen que sí, y no solo en java sino también en otros lenguajes de programación y en otros entornos de programación.



Selección del Diseño: Utilizamos JFrame para el diseño de una matriz de 3x5 (3 filas y 5 columnas) 3 filas y 3 columnas para el tablero original donde los jugadores colocarán sus

marcas buscando el objetivo, las 6 casillas restantes es para la visualización del jugador y su puntaje y los botones de salida o vuelta al juego (volver a jugar una partida).

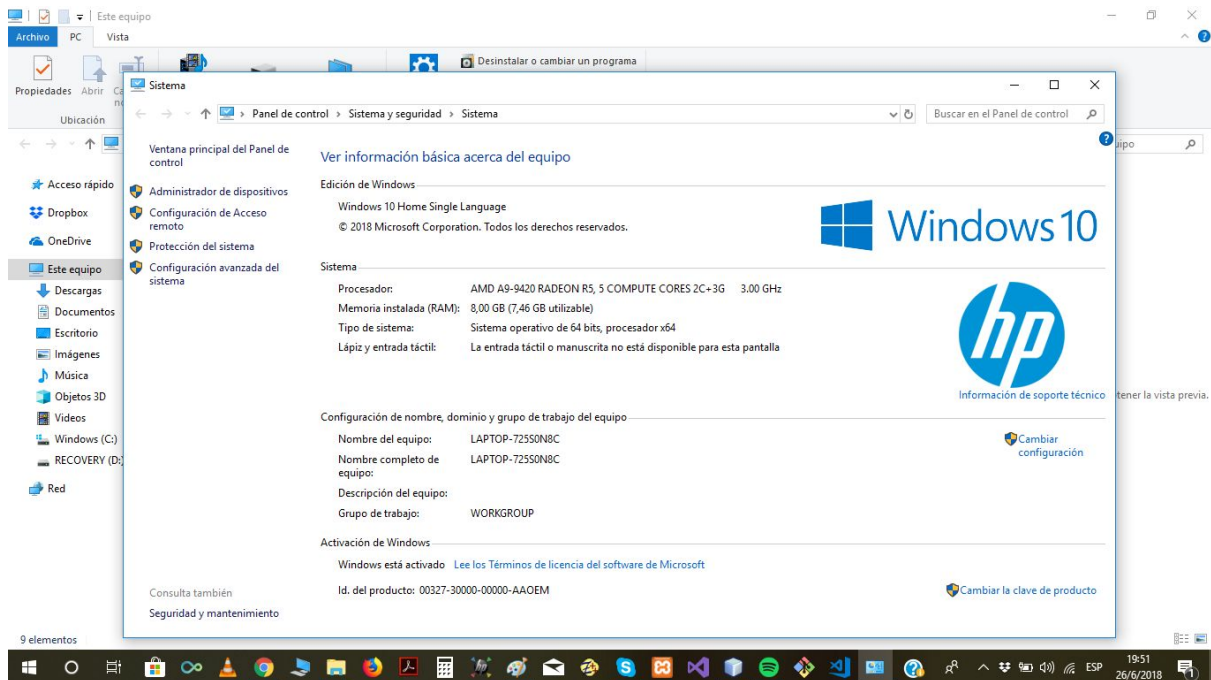
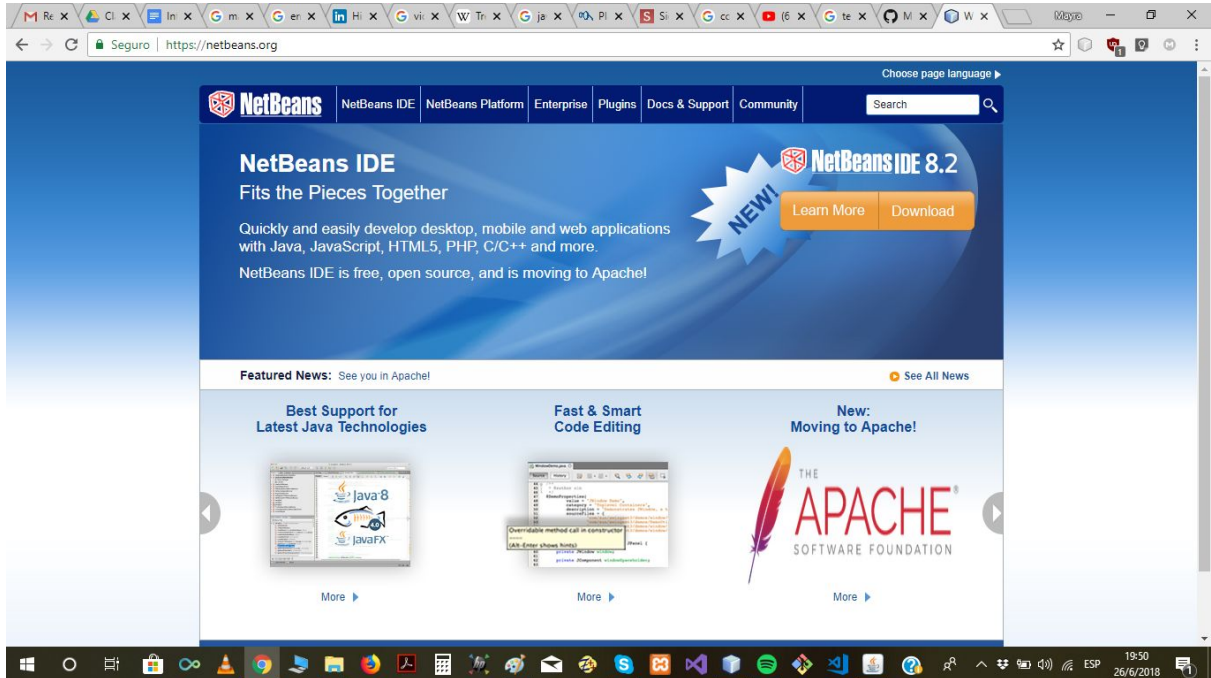


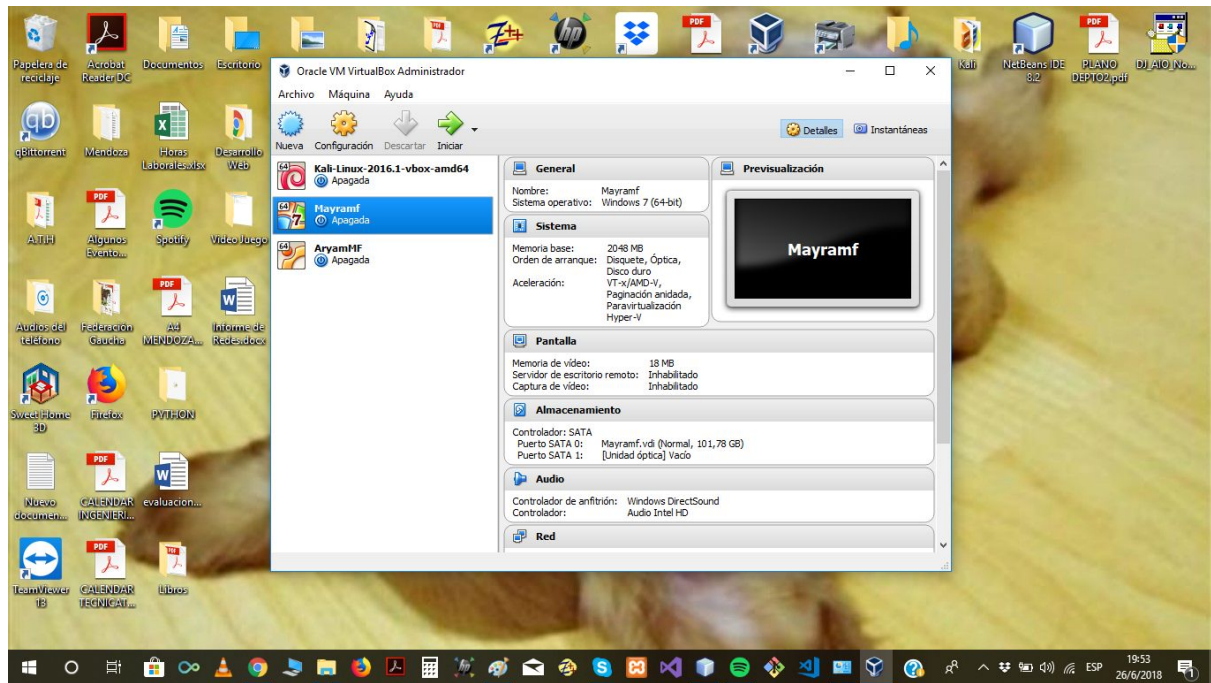
Etapas 3:

1- Comunicación según dimensión Empírica.

- Recolección de datos: Video Tutoriales de desarrollo, los datos se encuentran en las librerías de java.
- Análisis de Datos: Según los videos tutoriales de desarrollo de videojuegos parecidos, estos no contienen una inteligencia artificial y no presentan mallas o matrices estructuradas con métodos que no permitan la superposición de marcas son interfaces para jugar únicamente Humano a Humano.
- Presentación de Resultados: Nuestro juego contiene malla- matriz con métodos de vaciado, análisis de jugada y modo Humano-máquina, implementando condicionales en la programación y bucles básicamente.

Recursos Utilizados





Bibliografía:

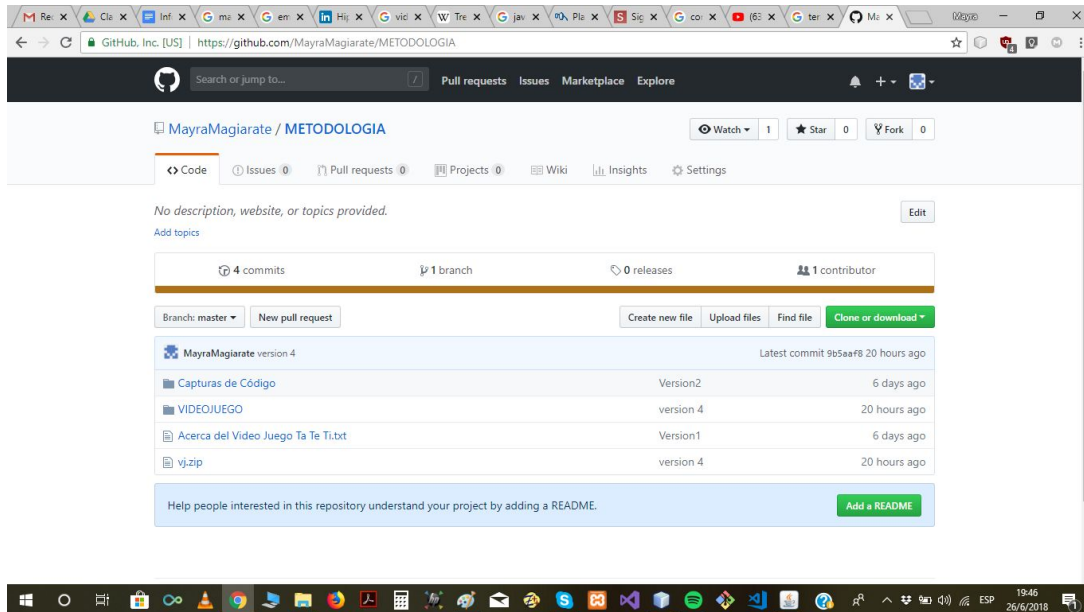
<https://es.wikipedia.org/wiki/Wikipedia:Portada>
<https://www.java.com/es/>
<https://docs.oracle.com/javase/tutorial/>
<https://www.udacity.com/courses/all>
<https://www.youtube.com/>

Desarrollo

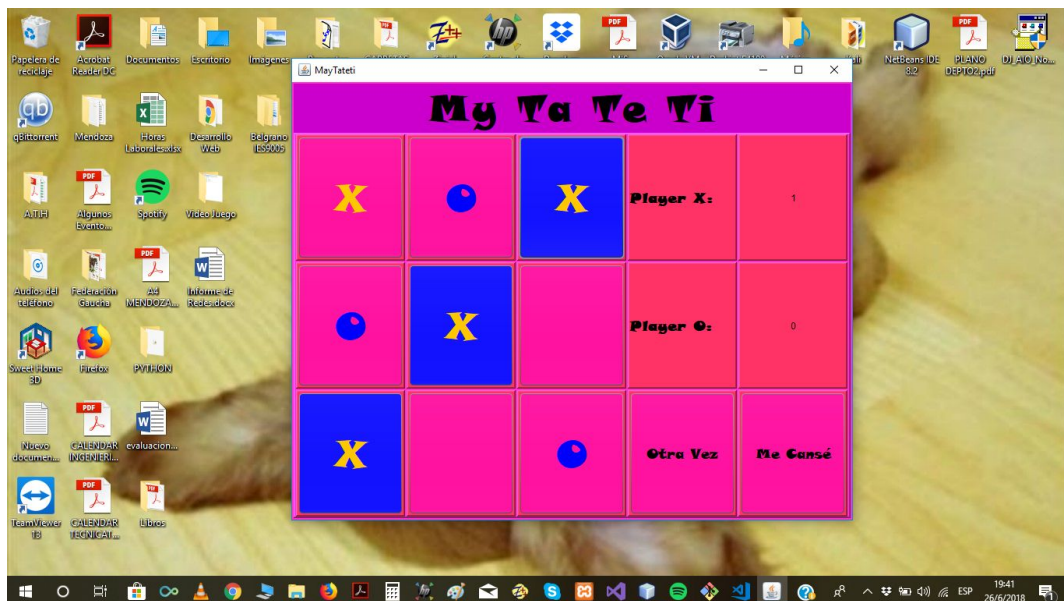
Planificación de Tareas

- a) Actualización de jdk, instalación de Netbeans.
- b) Creación de Proyecto en Netbeans java - java Application llamado VIDEOJUEGO.
- c) Creación de un repositorio remoto.
- d) Primer Commit.
- e) Creación interfaz Gráfica, uso de java Swing JFrame, JPanel, JLabel, JButton.
- f) Creación de Métodos necesarios para el funcionamiento del juego.
- g) Testing del juego en modo Humano-Humano.
- h) Recolección de datos (Errores)
- i) Modificaciones - Guardado
- j) Creación de Métodos de juego para la inteligencia Artificial del jugador Máquina.
- k) Testing del juego en modo Humano-Máquina.
- l) Recolección de datos (Errores)
- m) Modificaciones - Guardado.
- n) Creación de zip que contenga todo el proyecto.

Creación de Repositorio Remoto:



Interfaz Gráfica:



Métodos:

private void puntajesDelJuego() { // Este es el Metodo que creamos para recolectar y calcular los puntajes de cada jugador.

```
    puntosPlayerX.setText(String.valueOf(xConteo));
    puntosPlayerO.setText(String.valueOf(oConteo));
}
```

private void ElegiTuJugador() { // Este Método es para elegir el jugador, por X o por O.

```
    if (ComienzoDelJuego.equalsIgnoreCase("X")) {
        ComienzoDelJuego = "O";
    } // fin del if
    else {
        ComienzoDelJuego = "X";
    }
}
```

} // Fin del Metodo ElegiTuJugador.

// Metodo que determina si es posible hacer una jugada en determinada
// fila y columna del tablero

```
private boolean RealizarJugada(int Fila, int Columna, String Jugador)
{
    boolean SeRealizoLaJugada = false;
    if (EstamosJugando == true){
        if (Tablero[Fila][Columna] == "") {
            Tablero[Fila][Columna] = Jugador;
            SeRealizoLaJugada = true;
        } else {
            SeRealizoLaJugada = false;
        }
    } else {
        SeRealizoLaJugada = false;
    }
    return SeRealizoLaJugada;
}
```

```
private void DibujarBotones() {
    ColorearBoton(jButton1, Tablero[0][0]);
    ColorearBoton(jButton2, Tablero[0][1]);
    ColorearBoton(jButton3, Tablero[0][2]);

    ColorearBoton(jButton6, Tablero[1][0]);
}
```

```
ColorearBoton(jButton7, Tablero[1][1]);
ColorearBoton(jButton8, Tablero[1][2]);
```

```
ColorearBoton(jButton11, Tablero[2][0]);
ColorearBoton(jButton12, Tablero[2][1]);
ColorearBoton(jButton13, Tablero[2][2]);
}
```

private void ColorearBoton(javax.swing.JButton Boton, String Jugador) { // Este Método permite cambiar el color de la X o la O.

```
    Boton.setText(Jugador);
    if (Jugador == "X") {
        Boton.setForeground(Color.ORANGE);
    } else {
        Boton.setForeground(Color.BLUE);
    }
}
```

```
private void VaciarTablero() {
    for (int Fila = 0; Fila < 3; Fila++) {
        for (int Columna = 0; Columna < 3; Columna++) {
            Tablero[Fila][Columna] = "";
        }
    }
}
```

private void JugarComputadora() { // Aquí comienza el Método o la inteligencia que le damos a la computadoras.

```
    if (EstamosJugando == true){
        boolean YaJugoLaComputadora = false;
        for (int f = 0; f < 3; f++) {
            int JugadasHumano = ContarJugadasEnFila(f,"X"); // por defecto siempre la X será para el Humano.
```

```
            int JugadasComputadora = ContarJugadasEnFila(f,"O");
            if (JugadasHumano == 2 && JugadasComputadora == 0){
                for (int c = 0; c < 3; c++) {
                    if (Tablero[f][c] == ""){
                        YaJugoLaComputadora = RealizarJugada(f, c, "O");
                    }
                }
                break;
            }
        }
```

} // Este for solo hace la búsqueda por filas. En caso de no encontrar dentro de la fila un espacio vacío habiendo ya dos ocupados lo rellena con una O.

```
    if (YaJugoLaComputadora == false){
        for (int c = 0; c < 3; c++) {
```

```

    int JugadasHumano = ContarJugadasEnColumna(c, "X");
    int JugadasComputadora = ContarJugadasEnColumna(c, "O");
    if (JugadasHumano == 2 && JugadasComputadora == 0){
        for (int f = 0; f < 3; f++) {
            if (Tablero[f][c] == ""){
                YaJugoLaComputadora = RealizarJugada(f, c, "O");
            }
        }
        break;
    }
} // Este for revisa la jugada del Humano por Columnas.
}
if (YaJugoLaComputadora == false){
    int JugadasHumano = ContarJugadasEnDiagonalPrincipal("X");
    int JugadasComputadora = ContarJugadasEnDiagonalPrincipal("O");
    if (JugadasHumano == 2 && JugadasComputadora == 0){
        for (int x = 0; x < 3; x++) {
            int f = x;
            int c = x;
            if (Tablero[f][c] == ""){
                YaJugoLaComputadora = RealizarJugada(f, c, "O");
            }
        }
    }
}
if (YaJugoLaComputadora == false){
    int JugadasHumano = ContarJugadasEnDiagonalContraria("X");
    int JugadasComputadora = ContarJugadasEnDiagonalContraria("O");
    if (JugadasHumano == 2 && JugadasComputadora == 0){
        for (int x = 0; x < 3; x++) {
            int f = 2 - x;
            int c = x;
            if (Tablero[f][c] == ""){
                YaJugoLaComputadora = RealizarJugada(f, c, "O");
            }
        }
    }
}
if (YaJugoLaComputadora == false){ // Aquí la computadora juega al Azar porque no
encontró elementos en las diagonales ni en las columnas y filas basandose en las condiciones
dadas.
    while (YaJugoLaComputadora == false){
        int f = ObtenerAleatorioEntre(0,2);
        int c = ObtenerAleatorioEntre(0, 2);
        YaJugoLaComputadora = RealizarJugada(f, c, "O");
    }
}

```

```
    }  
  }  
}
```

```
private int ObtenerAleatorioEntre(int limiteInferior, int limiteSuperior){  
    Random r = new Random();  
    int Resultado = r.nextInt((limiteSuperior+1) - limiteInferior) + limiteInferior;  
    return Resultado;  
}
```

// Cantidad de jugadas para un determinado jugador en una determinada Fila. Metodo para contar la cantidad de "X o O en una misma fila.

private int ContarJugadasEnFila(int Fila, String Jugador) { // Aqui el metodo pide la fila donde queremos buscar y la letra que queremos buscar, la cual coincide con el jugador.

```
    int Jugadas = 0;  
    for (int i = 0; i < 3; i++) {  
        if (Tablero[Fila][i] == Jugador){  
            Jugadas++;  
        }  
    }  
    return Jugadas;  
}
```

```
private int ContarJugadasEnColumna(int Columna, String Jugador) {  
    int Jugadas = 0;  
    for (int i = 0; i < 3; i++) {  
        if (Tablero[i][Columna] == Jugador){  
            Jugadas++;  
        }  
    }  
    return Jugadas;  
}
```

```
private int ContarJugadasEnDiagonalPrincipal(String Jugador){  
    int Jugadas = 0;  
    for (int i = 0; i < 3; i++) {  
        if (Tablero[i][i] == Jugador){  
            Jugadas++;  
        }  
    }  
    return Jugadas;  
}
```

```
private int ContarJugadasEnDiagonalContraria(String Jugador){  
    int Jugadas = 0;  
    for (int i = 0; i < 3; i++) {  
        if (Tablero[2 - i][i] == Jugador){
```

```

        Jugadas++;
    }
}
return Jugadas;
}

```

private void JugadaGanadora() { // Este Metodo analiza la jugada segun X o O elegido, que resulta ganadora.

```

    String b1 = jButton1.getText();
    String b2 = jButton2.getText(); // Jugada Ganadora de Primera linea o fila
    String b3 = jButton3.getText();

```

```

    String b6 = jButton6.getText();
    String b7 = jButton7.getText(); // Jugada Ganadora de Segunda linea o fila
    String b8 = jButton8.getText();

```

```

    String b11 = jButton11.getText();
    String b12 = jButton12.getText(); // Jugada Ganadora de Tercera linea o fila
    String b13 = jButton13.getText();

```

```

    if (b1 == "X" && b2 == "X" && b3 == "X") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
        "My Tateti", JOptionPane.INFORMATION_MESSAGE);
        xConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
        xConteo.

```

```

        jButton1.setBackground(Color.YELLOW);
        jButton2.setBackground(Color.YELLOW);
        jButton3.setBackground(Color.YELLOW);
        EstamosJugando = false;
    } // Fin if de la primera jugada Linea 1.

```

```

    if (b6 == "X " && b7 == "X" && b8 == "X") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
        "My Tateti", JOptionPane.INFORMATION_MESSAGE);
        xConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
        xConteo.

```

```

        jButton6.setBackground(Color.MAGENTA);
        jButton7.setBackground(Color.MAGENTA);
        jButton8.setBackground(Color.MAGENTA);
        EstamosJugando = false;
    } // Fin If de la segunda jugada Linea 2.

```

```

    if (b11 == "X" && b12 == "X" && b13 == "X") {

```

```

        JOptionPane.showMessageDialog(this, "El Jugador que elegio la X le ha GANADO!! ",
"My Tateti", JOptionPane.INFORMATION_MESSAGE);
        xConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
xConteo.
        jButton11.setBackground(Color.WHITE);
        jButton12.setBackground(Color.WHITE);
        jButton13.setBackground(Color.WHITE);
        EstamosJugando = false;

    } // Fin If de la tercera jugada Linea 3.
    if (b1 == "X" && b6 == "X" && b11 == "X") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
"My Tateti", JOptionPane.INFORMATION_MESSAGE);
        xConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
xConteo.
        jButton1.setBackground(Color.ORANGE);
        jButton6.setBackground(Color.ORANGE);
        jButton11.setBackground(Color.ORANGE);
        EstamosJugando = false;

    } // Fin if de la 4 jugada Columna1.
    if (b2 == "X" && b7 == "X" && b12 == "X") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
"My Tateti", JOptionPane.INFORMATION_MESSAGE);
        xConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
xConteo.
        jButton2.setBackground(Color.RED);
        jButton7.setBackground(Color.RED);
        jButton12.setBackground(Color.RED);
        EstamosJugando = false;

    } // Fin if 5 jugada Columna 2.
    if (b3 == "X" && b8 == "X" && b13 == "X") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
"My Tateti", JOptionPane.INFORMATION_MESSAGE);
        xConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
xConteo.
        jButton3.setBackground(Color.GREEN);
        jButton8.setBackground(Color.GREEN);
        jButton13.setBackground(Color.GREEN);
        EstamosJugando = false;

```

```

} // Fin if 6 jugada columna 3.
if (b1 == "X" && b7 == "X" && b13 == "X") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
    "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    xConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
    xConteo.
    jButton1.setBackground(Color.CYAN);
    jButton7.setBackground(Color.CYAN);
    jButton13.setBackground(Color.CYAN);
    EstamosJugando = false;

} // fin if Jugada de Diagonal 1.
if (b11 == "X" && b7 == "X" && b3 == "X") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la X la tiene Clara. ",
    "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    xConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
    xConteo.
    jButton11.setBackground(Color.BLUE);
    jButton7.setBackground(Color.BLUE);
    jButton3.setBackground(Color.BLUE);
    EstamosJugando = false;

} // FIN DE LOS IF PARA EL JUGADOR X
if (b1 == "O" && b2 == "O" && b3 == "O") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ",
    "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    oConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
    xConteo.
    jButton1.setBackground(Color.YELLOW);
    jButton2.setBackground(Color.YELLOW);
    jButton3.setBackground(Color.YELLOW);
    EstamosJugando = false;
} // Fin if de la primera jugada Linea 1.

if (b6 == "O " && b7 == "O" && b8 == "O") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ",
    "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    oConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
    xConteo.
    jButton6.setBackground(Color.MAGENTA);
    jButton7.setBackground(Color.MAGENTA);
    jButton8.setBackground(Color.MAGENTA);

```



```

    EstamosJugando = false;

} // Fin If de la segunda jugada Linea 2.

if (b11 == "O" && b12 == "O" && b13 == "O") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la O le ha GANADO!!", "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    oConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a xConteo.
    jButton11.setBackground(Color.WHITE);
    jButton12.setBackground(Color.WHITE);
    jButton13.setBackground(Color.WHITE);
    EstamosJugando = false;

} // Fin If de la tercera jugada Linea 3.
if (b1 == "O" && b6 == "O" && b11 == "O") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ", "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    oConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a xConteo.
    jButton1.setBackground(Color.ORANGE);
    jButton6.setBackground(Color.ORANGE);
    jButton11.setBackground(Color.ORANGE);
    EstamosJugando = false;

} // Fin if de la 4 jugada Columna1.
if (b2 == "O" && b7 == "O" && b12 == "O") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ", "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    oConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a xConteo.
    jButton2.setBackground(Color.RED);
    jButton7.setBackground(Color.RED);
    jButton12.setBackground(Color.RED);
    EstamosJugando = false;

} // Fin if 5 jugada Columna 2.
if (b3 == "O" && b8 == "O" && b13 == "O") {
    JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ", "My Tateti", JOptionPane.INFORMATION_MESSAGE);
    oConteo++;
    puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a xConteo.

```

```

        jButton3.setBackground(Color.GREEN);
        jButton8.setBackground(Color.GREEN);
        jButton13.setBackground(Color.GREEN);
        EstamosJugando = false;

    }// Fin if 6 jugada columna 3.
    if (b1 == "O" && b7 == "O" && b13 == "O") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ",
        "My Tateti", JOptionPane.INFORMATION_MESSAGE);
        oConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
xConteo.
        jButton1.setBackground(Color.CYAN);
        jButton7.setBackground(Color.CYAN);
        jButton13.setBackground(Color.CYAN);
        EstamosJugando = false;

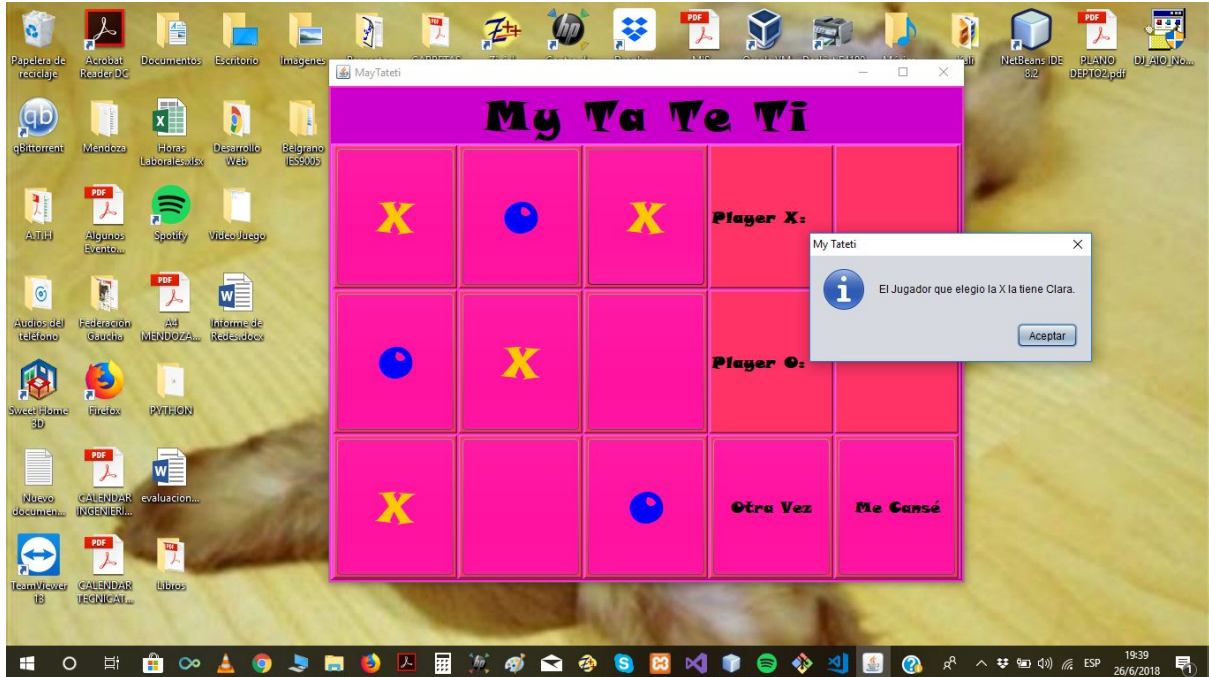
    }// fin if Jugada de Diagonal 1.
    if (b11 == "O" && b7 == "O" && b3 == "O") {
        JOptionPane.showMessageDialog(this, "El Jugador que elegio la O la tiene Clara. ",
        "My Tateti", JOptionPane.INFORMATION_MESSAGE);
        oConteo++;
        puntajesDelJuego(); // LLamo al metodo puntajesDelJuego para darle valor a
xConteo.
        jButton11.setBackground(Color.BLUE);
        jButton7.setBackground(Color.BLUE);
        jButton3.setBackground(Color.BLUE);
        EstamosJugando = false;
    }

    // ¿Que pasa si se completan todos los casilleros pero no hay ninguna jugada
ganadora?
    int CasillerosOcupados = 0;
    for (int f = 0; f < 3; f++) {
        for (int c = 0; c < 3; c++) {
            if (Tablero[f][c] != ""){
                CasillerosOcupados++;
            }
        }
    }
    if (CasillerosOcupados == 9){
        EstamosJugando = false;
    }

} // Fin del Metodo Jugadas Ganadoras

```

Testing:



Conclusiones y Resultados

Nuestra investigación ha sido Exploratoria. Los métodos y recursos utilizados, y su implementación en el desarrollo de videojuego Tatetí nos llevan a afirmar que es posible desarrollar un videojuego sencillo con interfaz gráfica utilizando sólo las librerías básicas del paquete standar de java.

Los conocimientos básicos necesarios para el desarrollo del videojuego son los conceptos y la implementación mínima de los condicionales y bucles en java, así como el uso de métodos y creación de clases, funciones y adquirir un manejo mínimo del uso de la librería java Swing (conocimiento adquirido por videotutoriales y manuales online).