

Javier Isaías Coyoy Marín  
202030556

### Manual Técnico

La versión de Java utilizada para este proyecto fue Java **15**.

La versión de Netbeans utilizada para este proyecto fue Netbeans **12.6**.

A continuación, presentamos el diagrama de clases correspondiente a este proyecto:

# Konquest

## Menu

- iniciarJuego(): void



## DiseñarMapa

- jugador: boolean
- compu: boolean
- ciego: boolean
- distribuir: boolean
- naves: boolean
- atributos: boolean
- nombre: String

- obtenerAnchura(): int
- obtenerAltura(): int
- obtenerTurnos(): int
- obtenerPlanetasNeutrales(): int
- obtenerPlanetasZombies(): int
- obtenerNombre(): String
- realizarSuma(): int
- control(): void
- jugar(): void



↓ Use

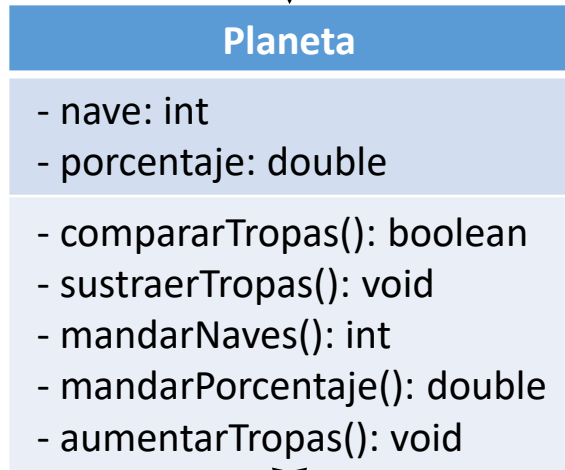
## Batallas

- Player: Planeta
  - Neutrales: Planeta
  - Zombies: Planeta
  - altura: int
  - anchura: int
  - turnos: int
  - planetasDelJU[]: String
  - planetasDelJD[]: String
  - emisor: String
  - coordenada: String
  - jugadorEnTurno: int
- 
- mapa(): void
  - verificarCoordenadas(): void
  - verificarPlaneta(): boolean
  - contarDistancia(): int
  - abrirAccesos(): void
  - verificarTropas(): void
  - restarTropas(): void
  - organizarTurnos(): void
  - confirmarCoordenadas(): void
  - volver Seleccionar(): void
  - terminarTurno(): void
  - mandarAtaque(): void
  - mandarRefuerzos(): void

↓ Use



Use

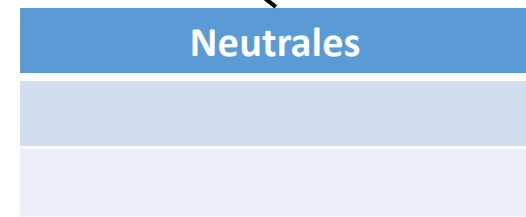
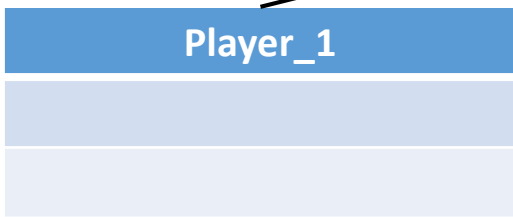


Extends

Extends

Extends

Extends



A continuación, presentamos los métodos escritos en Pseudocódigo que se utilizaron para la realización del proyecto:

Los siguientes métodos sirven para recopilar la información que el usuario ingresa en el panel de diseño de mapa que será utilizados por los métodos siguientes para su evaluación y aceptación.

```
private void jugadorElegido()  
.....  
    compuElegida.setVisible(false)  
Fin  
  
private void compuElegida()  
.....  
    jugadorElegido.setVisible(false)  
Fin  
  
private int obtenerAnchura()  
.....  
    return (int) anchura.getValue()  
Fin  
  
private int obtenerAltura()  
.....  
    return (int) altura.getValue()  
Fin  
  
private int obtenerTurno()  
.....  
    return (int) numeroTurnos.getValue()  
Fin  
  
private int obtenerPlanetasNeutrales()  
.....  
    return (int) numeroPlanetasN.getValue()  
Fin
```

```

private int obtenerPlanetasZombies()
|     return (int) numeroPlanetasZ.getValue()
Fin

private String obtenerNombre()
|     return String.valueOf(nombreMapa.getText())
Fin

private int obtenerTope()
|     int tope = obtenerAnchura() * obtenerAltura()
|     tope = tope - 2
|     return tope
Fin

private int realizarSuma()
|     int suma = obtenerPlanetasNeutrales() + obtenerPlanetasFantasmas() + obtenerPlanetasZombies()
|     return suma
Fin

private void control() {
|     Si (mapaCiego.isSelected()) entonces
|     |     ciego = true
|     |     finSi
|
|     Si (producAcumulada.isSelected()) entonces
|     |     produccion = true
|     |     finSi
|
|     Si (distribuirPlanetas.isSelected()) entonces
|     |     distribuir = true
|     |     finSi

```

```

    Si (mostrarNaves.isSelected()) entonces
        naves = true
    finSi

    Si (mostrarAtributos.isSelected()) entonces
        atributos = true
    finSi
finSi

```

El siguiente método es para la verificación de información en el apartado anterior, lo que hace es verificar que no haya números negativos o ceros que puedan estropear o interrumpir el avance del programa y de haberlos entonces emitirá un mensaje de advertencia para que el usuario sepa donde esta el error y pueda corregirlo.

```

Si (obtenerAnchura() > 1 && obtenerAnchura() < 9 && obtenerAltura() > 1 && obtenerAltura() < 7
    && obtenerTurno() > 0 && obtenerPlanetasNeutrales() >= 0 && obtenerPlanetasFantasmas() >= 0
    && obtenerPlanetasZombies() >= 0 && !"".equals(obtenerNombre())) entonces

    Si (realizarSuma() <= obtenerTope()) entonces

        Si (jugadorElegido.isSelected()) entonces
            jugador = true
            control()
            Batalla pelear = new Batalla(ciego,produccion,distribuir,naves,atributos,
            obtenerAnchura(),obtenerAltura(),obtenerTurno(),obtenerPlanetasNeutrales(),
            obtenerPlanetasFantasmas(),obtenerPlanetasZombies(),obtenerNombre())

            pelear.setVisible(true)

```

```

sino Si (compuElegida.isSelected()) entonces
    compu = true
    control()
sino
    JOptionPane.showMessageDialog(this,"Debes seleccionar a tu Oponente","Jugador",
    JOptionPane.INFORMATION_MESSAGE)
finsi
sino
    JOptionPane.showMessageDialog(this,"El número de Planetas excede el número de Casillas disponibles",
    "Error",JOptionPane.ERROR_MESSAGE)
    JOptionPane.showMessageDialog(this,"La suma de los planetas debe ser igual o menor a " + obtenerTope(),
    "Información",JOptionPane.INFORMATION_MESSAGE)
finsi
sino
Si (obtenerAnchura() == 0) entonces
    JOptionPane.showMessageDialog(this,"Debes seleccionar el Ancho del Mapa","Ancho",JOptionPane.INFORMATION_MESSAGE)
finsi
Si (obtenerAnchura() == 1) entonces
    JOptionPane.showMessageDialog(this,"El Ancho no puede ser 1","Error",JOptionPane.ERROR_MESSAGE)
finsi
Si (obtenerAnchura() >= 9) entonces
    JOptionPane.showMessageDialog(this,"El Ancho tiene que ser Menor de 9","Error",JOptionPane.ERROR_MESSAGE)
finsi
Si (obtenerAltura() == 0) entonces
    JOptionPane.showMessageDialog(this,"Debes seleccionar la Altura del Mapa","Altura",JOptionPane.INFORMATION_MESSAGE)
finsi

```



```

Si (obtenerAltura() == 1) entonces
    JOptionPane.showMessageDialog(this,"La Altura no puede ser 1","Error",JOptionPane.ERROR_MESSAGE)
finsi

Si (obtenerAltura() >= 7) entonces
    JOptionPane.showMessageDialog(this,"La Altura tiene que ser Menor de 7","Error",JOptionPane.ERROR_MESSAGE)
finsi

Si (obtenerTurno() == 0) entonces
    JOptionPane.showMessageDialog(this,"Debes seleccionar los Turnos que tendrá cada Jugador","Turnos",
    JOptionPane.INFORMATION_MESSAGE)
finsi

Si ("".equals(obtenerNombre())) entonces
    JOptionPane.showMessageDialog(this,"Debes escribir el Nombre del Mapa","Nombre",JOptionPane.INFORMATION_MESSAGE)
finsi

Si (obtenerAnchura() < 0 || obtenerAltura() < 0 || obtenerTurno() < 0 || obtenerPlanetasNeutrales() < 0
|| obtenerPlanetasFantasmas() < 0 || obtenerPlanetasZombies() < 0) entonces
    JOptionPane.showMessageDialog(this,"No puedes seleccionar Números Negativos","Error",JOptionPane.ERROR_MESSAGE)
finsi
finsi

```

El siguiente método permite la selección de los planetas, también es una especie de barra de verificación ya que tiene que pasar por 3 o 4 filtros para que sea procesada la información, y si alguno de los requisitos no se cumple, se lanzará un mensaje.

```

Si (coordenada == 0) entonces

    Si (verificarPlaneta()) entonces
        JOptionPane.showMessageDialog(this,"Coordenada Guardada, seleccione el lugar de destino","Guardado",
        JOptionPane.INFORMATION_MESSAGE)
        coordenada++
        filaInicio = fila
        columnaInicio = columna
    sino
        JOptionPane.showMessageDialog(this,"No puedes realizar ninguna acción desde esta posición","Error",
        JOptionPane.ERROR_MESSAGE)
    finisi

sino (coordenada == 1) entonces

    Si (emisor != planetaEmisor) entonces

        Si (verificarPlaneta2()) entonces
            filaFinal = fila
            columnaFinal = columna

            Si (turno >= contarDistancia()) entonces
                JOptionPane.showMessageDialog(this,"El destino es el Planeta " + emisor,
                " y las tropas llegarán en " + contarDistancia() + " turnos","Verificación",
                JOptionPane.INFORMATION_MESSAGE)
                JOptionPane.showMessageDialog(this,"Si estas deacuerdo, pulsa (Confirmar Coordenadas),
                de lo contrario pulsa (Volver a Seleccionar)","Confirmar Coordenadas",
                JOptionPane.INFORMATION_MESSAGE)
                coordenada++
            sino

```

```
        JOptionPane.showMessageDialog(this, "No hay suficientes turnos para ir a este Planeta,
        selecciona otro destino", "Error", JOptionPane.ERROR_MESSAGE)
    finisi

    sino
        JOptionPane.showMessageDialog(this, "No puedes realizar ninguna acción desde esta posición",
        "Error", JOptionPane.ERROR_MESSAGE)
    finisi

    sino
        JOptionPane.showMessageDialog(this, "No puedes realizar ninguna acción hacia esta posición",
        "Error", JOptionPane.ERROR_MESSAGE)
    finisi

    sino
        JOptionPane.showMessageDialog(this, "Ya no puedes seleccionar otra Coordenada", "Error",
        JOptionPane.ERROR_MESSAGE)
    finisi
```

El siguiente método es para encontrar la distancia o años luz que se necesitarán ya sea para mandar un ataque o para mandar refuerzos, y este método es invocado por el método (verificarCoordenadas), y si la distancia excede los turnos permitidos, entonces lanzará un mensaje y no se le permitirá realizar la acción.

```
private int contarDistancia()  
    Var ancho = 0  
    Var altura = 0  
  
    Si (filaInicio <= filaFinal) entonces  
        altura = filaFinal - filaInicio  
    sino  
        altura = filaInicio - filaFinal  
    finSi  
  
    Si (columnaInicio <= columnaFila) entonces  
        ancho = columnaFinal - columnaInicio  
    sino  
        ancho = columnaInicio - columnaFinal  
    finSi  
  
    return distancia = altura + ancho  
Fin
```

El siguiente método es utilizado para ordenar las confirmaciones de ataque de cada jugador, es decir que cada vez que del jugador de clic en la opción confirmar, entonces los datos se guardarán junto con el turno para así cuando toque llegar a la sección de batalla o refuerzos, si el turno y los ataques de turno conciden entonces se realizara la acción correspondiente. Para este método se hizo uso del ordenamiento burbuja.

```

private void organizarTurnos(int turnosAtaquesJugador[])
    Var tope = 9

    Desde i = 0; hasta 10; 1

        Desde j = 0; hasta tope; 1

            Si (turnosAtaquesJugador[j] > turnosAtaquesJugador[j + 1]) entonces
                int copia = turnosAtaquesJugador[j]
                turnosAtaquesJugador[j] = turnosAtaquesJugador[j + 1]
                turnosAtaquesJugador[j + 1] = copia
            fin_si

        fin_desde

        tope--

    fin_desde
Fin

```

El siguiente método se utilizara cuando el jugador ya no desee enviar un ataque a un determinado planeta y en ese caso dandole al botón Volver a seleccionar le permitira volver a escoger las coordenadas.

```
coordenada = 0
filaInicio = 0
filaFinal = 0
columnaInicio = 0
columnaFinal = 0
planetaEmisor = null
emisor = null
abrir = 0
numeroAtaque.setEnabled(false)
enviarA.setEnabled(false)
numeroRefuerzo.setEnabled(false)
enviarR.setEnabled(false)
JOptionPane.showMessageDialog(this,"Puedes volver a Seleccionar","Empezar",
JOptionPane.INFORMATION_MESSAGE)
```

El siguiente método es utilizado para enviar las tropas que nosotros hemos seleccionado previamente y lo que hace es que almacena información esencial que la utilizara cuando el turno de ataque sea igual al turno de la ronda.

```
Si (jugadorEnTurno == 1) entonces
    Si (verificarTropas()) entonces
        restarTropas()
        Desde i = 0; hasta 10; 1
            Si (turnosAtaquesJugadorUno[i] == 100) entonces
                turnosAtaquesJugadorUno[i] = contarDistancia() + turno
                planetasQueAtacanJU[i] = planetaEmisor
                planetasAAtacarJU[i] = emisor
                numeroDeTropasJUA[i] = obtenerNumeroAtaque()
                break
            fin_si
        fin_desde

        organizarTurnos()
        enviarA.setEnabled(false)
        numeroAtaque.setEnabled(false)
        coordenada = 0
    sino
        JOptionPane.showMessageDialog(this,
            "No tienes ese número de Tropas disponibles en este Planeta",
            "Estado de las Tropas", JOptionPane.ERROR_MESSAGE)
    fin_si
sino
```

```
Si (verificarTropas()) entonces
    restarTropas()
    Desde i = 0; hasta 10; 1

        Si (turnosAtaquesJugadorDos[i] == 100) entonces
            turnosAtaquesJugadorDos[i] = contarDistancia() + turno
            planetasQueAtacanJD[i] = planetaEmisor
            planetasAAatacarJD[i] = emisor
            numeroDeTropasJDA[i] = obtenerNumeroAtaque()
            break;
        fin_si

    fin_desde

    organizarTurnos()
    enviarA.setEnabled(false)
    numeroAtaque.setEnabled(false)
    coordenada = 0
sino
    JOptionPane.showMessageDialog(this,
    "No tienes ese número de Tropas disponibles en este Planeta",
    "Estado de las Tropas",JOptionPane.ERROR_MESSAGE)
fin_si

fin_si
```

```
confirmarCoordenadas.setEnabled(true)
volverASeleccionar.setEnabled(true)
```



El siguiente método es utilizado para mandar las tropas de apoyo a otro planeta que sean del mismo jugador y de esa manera poder mandar refuerzos cuando sea necesario.

```
Si (obtenerNumeroRefuerzo() != 0) entonces

    Si (jugadorEnTurno == 1) entonces

        if (verificarTropas()) entonces
            restarTropas()
            Desde i = 0; hasta 10; 1
                Si (turnosRefuerzosJugadorUno[i] == 0) entonces
                    turnosRefuerzosJugadorUno[i] = contarDistancia() + turno
                    planetasAReforzarJU[i] = emisor
                    numeroDeTropasJUR[i] = obtenerNumeroRefuerzo()
                    break
                fin_si
            fin_desde

            organizarTurnos1()
            enviarR.setEnabled(false)
            numeroRefuerzo.setEnabled(false)
            coordenada = 0
        sino
            JOptionPane.showMessageDialog(this,
            "No tienes ese número de Tropas disponibles en este Planeta",
            "Estado de las Tropas",JOptionPane.ERROR_MESSAGE);
        fin_si
    sino
```

```

Si (verificarTropas()) entonces
    restarTropas()
    Desde i = 0; hasta 10; 1
        if (turnosRefuerzosJugadorDos[i] == 0) entonces
            turnosRefuerzosJugadorDos[i] = contarDistancia() + turno
            planetasAReforzarJD[i] = emisor
            numeroDeTropasJDR[i] = obtenerNumeroRefuerzo()
            break
        fin si
    fin desde

    organizarTurnos1()
    enviarA.setEnabled(false)
    numeroRefuerzo.setEnabled(false)
    coordenada = 0
sino
    JOptionPane.showMessageDialog(this,
        "No tienes ese número de Tropas disponibles en este Planeta",
        "Estado de las Tropas",JOptionPane.ERROR_MESSAGE)
    fin si
fin si

confirmarCoordenadas.setEnabled(true)
volverASeleccionar.setEnabled(true)
sino
    JOptionPane.showMessageDialog(this,"Las tropas no pueden ser 0",
        "Tropas",JOptionPane.ERROR_MESSAGE)
fin si

```

El siguiente método es para terminar el turno de un jugador y que el mando pase al otro jugador, además de que verifica para ver si los turnos de espera coinciden con el turno que se está jugando, y si este coincide entonces se realice ya sea un ataque o un refuerzo.

```
turno++
```

```
Si (turno <= turnos) entonces
```

```
    Desde i = 0; hasta 10; 1
```

```
        Si (turno == turnosAtaquesJugadorUno[0]) entonces
```

```
            naves1 = numeroDeTropasJUA[0]
```

```
            porcentaje1 = mandarPorcentaje(planetasQueAtacanJU[0])
```

```
            naves2 = mandarNaves(planetasAAatacarJU[0])
```

```
            porcentaje2 = mandarPorcentaje(planetasAAatacarJU[0])
```

```
            mandarAtaque(planetasQueAtacanJU[0], planetasAAatacarJU[0])
```

```
        finSi
```

```
        Si (turno == turnosRefuerzosJugadorUno[0]) entonces
```

```
            naves1 = numeroDeTropasJUR[0]
```

```
            mandarRefuerzos(planetasAREforzarJU[0])
```

```
        finSi
```

```
        Si (turno == turnosAtaquesJugadorDos[0]) entonces
```

```
            naves1 = numeroDeTropasJDA[0]
```

```
            porcentaje1 = mandarPorcentaje(planetasQueAtacanJD[0])
```

```
            naves2 = mandarNaves(planetasAAatacarJD[0])
```

```
            porcentaje2 = mandarPorcentaje(planetasAAatacarJD[0])
```

```
            mandarAtaque(planetasQueAtacanJD[0], planetasAAatacarJD[0])
```

```
        finSi
```

```

        Si (turno == turnosRefuerzosJugadorDos[0]) entonces
            naves1 = numeroDeTropasJDR[0]
            mandarRefuerzos(planetasAReforzarJD[0])
        fin_si
    fin_desde

Si (jugadorEnTurno == 1) entonces
    jugadorEnTurno = 2
    letreroDelJugEnTurno.setText("Turno Player_2")
sino
    jugadorEnTurno = 1
    letreroDelJugEnTurno.setText("Turno Player_1")
fin_si

numeroDeTurno.setText("turno: " + turno)

int contador = 0
int contador1 = 0

Desde i = 0; hasta 10; 1
    Si (planetasDelJU[i] != null) entonces
        contador++
    fin_si
fin_desde

Desde i = 0; hasta 10; 1
    Si (planetasDelJD[i] != null) entonces
        contador1++
    fin_si
fin_si

```

```

Si (contador == 0) entonces
    JOptionPane.showMessageDialog(this,
        "El Player_1 se quedo sin Planetas, Gana el Player_2",
        "Ganador",JOptionPane.INFORMATION_MESSAGE)
    this.dispose()
finisi

```

```

Si (contador1 == 0) entonces
    JOptionPane.showMessageDialog(this,
        "El Player_2 se quedo sin Planetas, Gana el Player_1",
        "Ganador",JOptionPane.INFORMATION_MESSAGE)
    this.dispose()
finisi

```

```

sino
    JOptionPane.showMessageDialog(this,
        "Ya no hay más turnos, termino el juego","Fin del Juego",
        JOptionPane.INFORMATION_MESSAGE)
    int contador = 0
    int contador1 = 0

```

```

Desde i = 0; hasta 10; 1
    Si (planetasDelJU[i] != null) entonces
        contador++
    finisi
fin_desde

```

```

Desde i = 0; hasta 10; 1
    if (planetasDelJD[i] != null) entonces
        contador1++
    finisi
fin_desde

```

```
if (contador > contador1) entonces
    contador--
    JOptionPane.showMessageDialog(this,
        "El Player_1 ganó ya que conquisto " + contador + " planetas",
        "Ganador",JOptionPane.INFORMATION_MESSAGE)
finsi

Si (contador1 > contador) entonces
    contador1--
    JOptionPane.showMessageDialog(this,
        "El Player_2 ganó ya que conquisto " + contador1 + " planetas",
        "Ganador",JOptionPane.INFORMATION_MESSAGE)
finsi

Si (contador1 == contador) entonces
    contador--
    JOptionPane.showMessageDialog(this,
        "Empate, ambos Jugadores conquistaron " + contador + " planetas",
        "Empate",JOptionPane.INFORMATION_MESSAGE)
finsi

this.dispose()
```

```
finsi
```

Los siguientes métodos son utilizados para verificar los planetas y ver que los correspondan al jugador correspondiente.

```
private boolean verificarPlaneta()
    boolean permitir = false;

    Si (jugadorEnTurno == 1) entonces

        Desde i = 0; hasta 10; 1
            if (emisor.equals(planetasDelJU[i])) entonces
                permitir = true
                planetaEmisor = emisor
                break
            fin_si
        fin_desde

    sino
        Desde = 0; i < 10; i++) entonces
            Si (emisor.equals(planetasDelJD[i])) entonces
                permitir = true
                planetaEmisor = emisor
                break
            fin_si
        fin_desde

    fin_si

    return permitir
Fin
```

```
private boolean verificarPlaneta2()  
    boolean permitir = false  
    int entrar = 0  
  
    if (jugadorEnTurno == 1) {  
        Desde i = 0; hasta 10; 1  
            Si (emisor.equals(planetasDelJU[i])) entonces  
                permitir = true  
                entrar = 1  
                abrir = 1  
                break  
            fin_si  
        fin_desde  
  
        Si (entrar == 0) entonces  
            Si (!"Espacio".equals(emisor)) entonces  
                permitir = true  
                abrir = 0  
            fin_si  
        fin_si  
  
    sino
```



```
Desde i = 0; hasta 10; 1
```

```
    Si (emisor.equals(planetasDelJD[i])) entonces  
        permitir = true  
        entrar = 1  
        abrir = 1  
        break  
    finsi
```

```
fin_desde
```

```
Si (entrar == 0) entonces
```

```
    Si (!"Espacio".equals(emisor)) entonces  
        permitir = true  
        abrir = 0  
    finsi
```

```
finsi
```

```
finsi
```

```
return permitir
```

```
Fin
```

El siguiente método es utilizado para organizar los turnos y así hacer que siempre en la posición cero esten la información que se necesita para lograr atacar o reforzar una planeta.

```
Var tope = 9

Si (jugadorEnTurno == 1) entonces
    Desde i = 0; hasta 10; 1
        Desde j = 0; hasta tope; 1
            Si (turnosAtaquesJugadorUno[j] > turnosAtaquesJugadorUno[j + 1]) entonces
                int copia = turnosAtaquesJugadorUno[j]
                turnosAtaquesJugadorUno[j] = turnosAtaquesJugadorUno[j + 1]
                turnosAtaquesJugadorUno[j + 1] = copia
                int copial = numeroDeTropasJUA[j]
                numeroDeTropasJUA[j] = numeroDeTropasJUA[j + 1]
                numeroDeTropasJUA[j + 1] = copial
                String copia2 = planetasQueAtacanJU[j]
                planetasQueAtacanJU[j] = planetasQueAtacanJU[j + 1]
                planetasQueAtacanJU[j + 1] = copia2
                String copia3 = planetasAAatacarJU[j]
                planetasAAatacarJU[j] = planetasAAatacarJU[j + 1]
                planetasAAatacarJU[j + 1] = copia3
            finSi
        fin_desde
    tope--
```

```

    fin_desde
sino

    Desde i = 0; hasta 10; 1 entonces
        for (int j = 0; j < tope; j++) {

            Si (turnosAtaquesJugadorDos[j] > turnosAtaquesJugadorDos[j + 1]) entonces
                int copia = turnosAtaquesJugadorDos[j]
                turnosAtaquesJugadorDos[j] = turnosAtaquesJugadorDos[j + 1]
                turnosAtaquesJugadorDos[j + 1] = copia
                int copial = numeroDeTropasJDA[j]
                numeroDeTropasJDA[j] = numeroDeTropasJDA[j + 1]
                numeroDeTropasJDA[j + 1] = copial
                String copia2 = planetasQueAtacanJD[j]
                planetasQueAtacanJD[j] = planetasQueAtacanJD[j + 1]
                planetasQueAtacanJD[j + 1] = copia2
                String copia3 = planetasAAtacarJD[j]
                planetasAAtacarJD[j] = planetasAAtacarJD[j + 1]
                planetasAAtacarJD[j + 1] = copia3
            finSi

        fin_desde

        tope--

    fin_desde
finSi

```

Este método es para realizar ataques a un planeta que fue seleccionado previamente y a través de los cálculos determinar cuál es el planeta ganador y así darle como premio el planeta que conquistó.

```
naves1 = (int) (naves1 * porcentaje1)
naves2 = (int) (naves2 * porcentaje2)
int navesRestantes1 = naves1 - naves2
int navesRestantes2 = naves2 - naves1
```

```
Si (navesRestantes1 <= 0 && navesRestantes2 > 0) entonces
    JOptionPane.showMessageDialog(this,
        "El planeta " + atacante + " ataco al" + defensor + " pero resistio al Ataque",
        "Informe del Ataque",JOptionPane.INFORMATION_MESSAGE)
    reordenarTurnos()
finsi
```

```
Si (navesRestantes1 <= 0 && navesRestantes2 <= 0) entonces
    JOptionPane.showMessageDialog(this,
        "El planeta " + atacante + " ataco al" + defensor + " pero resistio al Ataque",
        "Informe del Ataque",JOptionPane.INFORMATION_MESSAGE)
    reordenarTurnos()
finsi
```

```
Si (navesRestantes1 > 0 && navesRestantes2 <= 0) entonces
    JOptionPane.showMessageDialog(this,
        "El planeta " + atacante + " conquisto el planeta " + defensor,
        "Informe del Ataque",JOptionPane.INFORMATION_MESSAGE)
```

```
if (jugadorEnTurno == 1) {  
    Desde i = 0; hasta 10; 1  
        Si (planetasDelJU[i] == null) entonces  
            planetasDelJU[i] = defensor  
            break  
        fin_si  
    fin_si  
  
    Desde j = 0; hasta 10; 1  
        if (planetasDelJD[j].equals(defensor)) entonces  
            planetasDelJD[j] = null  
            break  
        fin_si  
    fin_desde  
  
    reordenarTurnos()  
sino  
  
    Desde i = 0; hasta 10; 1  
        Si (planetasDelJD[i] == null) entonces  
            planetasDelJD[i] = defensor  
            break  
        fin_si  
    fin_desde
```

```

    Desde i = 0; hasta 10; 1
        Si (planetasDelJD[i] == null) entonces
            planetasDelJD[i] = defensor
            break
        fin_si
    fin_desde

    Desde j = 0; hasta 10; 1
        Si (planetasDelJU[j].equals(defensor)) entonces
            planetasDelJU[j] = null
            break
        fin_si
    fin_desde

    reordenarTurnos()
fin_si

```

Es siguiente método nos permite la asignación, y Evita que un mismo planeta sea renombrado.

```

private void organizarPlanetas()
    Var nombre = null
    Var indice = 0

    Desde i = 0; hasta altura; 1
        Desde j = 0; hasta anchura; 1
            mapa[i][j] = "Espacio"

            Si (i == 0 && j == 0) entonces
                mapa0.setText("Espacio")
            finSi

            Si (i == 0 && j == 1) entonces
                mapa1.setText("Espacio")
            finSi

            Si (i == 0 && j == 2) entonces
                mapa2.setText("Espacio")
            finSi

            Si (i == 0 && j == 3) entonces
                mapa3.setText("Espacio")
            finSi

            Si (i == 0 && j == 4) entonces
                mapa4.setText("Espacio")
            finSi

            Si (i == 0 && j == 5) entonces
                mapa5.setText("Espacio")
            finSi

```

```

Si (i == 0 && j == 6) entonces
    mapa6.setText("Espacio")
finSi

Si (i == 0 && j == 7) entonces
    mapa7.setText("Espacio")
finSi

Si (i == 1 && j == 0) entonces
    mapa8.setText("Espacio")
finSi

Si (i == 1 && j == 1) entonces
    mapa9.setText("Espacio")
finSi

Si (i == 1 && j == 2) entonces
    mapa10.setText("Espacio")
finSi

Si (i == 1 && j == 3) entonces
    mapa11.setText("Espacio")
finSi

Si (i == 1 && j == 4) entonces
    mapa12.setText("Espacio")
finSi

Si (i == 1 && j == 5) entonces
    mapa13.setText("Espacio")
finSi

```

```
Si (i == 1 && j == 6) entonces
    mapa14.setText("Espacio")
finisi
```

```
Si (i == 1 && j == 7) entonces
    mapa15.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 0) entonces
    mapa16.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 1) entonces
    mapa17.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 2) entonces
    mapa18.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 3) entonces
    mapa19.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 4) entonces
    mapa20.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 5) entonces
    mapa21.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 6) entonces
    mapa22.setText("Espacio")
finisi
```

```
Si (i == 2 && j == 7) entonces
    mapa23.setText("Espacio")
finisi
```

```
Si (i == 3 && j == 0) entonces
    mapa24.setText("Espacio")
finisi
```

```
Si (i == 3 && j == 1) entonces
    mapa25.setText("Espacio")
finisi
```

```
Si (i == 3 && j == 2) entonces
    mapa26.setText("Espacio")
finisi
```

```
Si (i == 3 && j == 3) entonces
    mapa27.setText("Espacio")
finisi
```

```
Si (i == 3 && j == 4) entonces
    mapa28.setText("Espacio")
finisi
```

```
Si (i == 3 && j == 5) entonces
    mapa29.setText("Espacio")
finisi
```



```
Si (i == 3 && j == 6) entonces
    mapa30.setText("Espacio")
finsi

Si (i == 3 && j == 7) entonces
    mapa31.setText("Espacio")
finsi

Si (i == 4 && j == 0) entonces
    mapa32.setText("Espacio")
finsi

Si (i == 4 && j == 1) entonces
    mapa33.setText("Espacio")
finsi

Si (i == 4 && j == 2) entonces
    mapa34.setText("Espacio")
finsi

Si (i == 4 && j == 3) entonces
    mapa35.setText("Espacio")
finsi

Si (i == 4 && j == 4) entonces
    mapa36.setText("Espacio")
finsi

Si (i == 4 && j == 5) entonces
    mapa37.setText("Espacio")
finsi

Si (i == 4 && j == 6) entonces
    mapa38.setText("Espacio")
finsi
```

```
Si (i == 5 && j == 0) entonces
    mapa40.setText("Espacio")
finsi

Si (i == 5 && j == 1) entonces
    mapa41.setText("Espacio")
finsi

Si (i == 5 && j == 2) entonces
    mapa42.setText("Espacio")
finsi

Si (i == 5 && j == 3) entonces
    mapa43.setText("Espacio")
finsi

Si (i == 5 && j == 4) entonces
    mapa44.setText("Espacio")
finsi

Si (i == 5 && j == 5) entonces
    mapa45.setText("Espacio")
finsi

Si (i == 5 && j == 6) entonces
    mapa46.setText("Espacio")
finsi

Si (i == 5 && j == 7) entonces
    mapa47.setText("Espacio")
finsi
```

fin\_desde

fin\_desde

```

mientras (indice < 2) hacer
    int y = (int) (Math.random()*altura)
    int x = (int) (Math.random()*anchura)

    Si (indice == 0) entonces
        nombre = "Player_1"
    sino
        nombre = "Player_2"
    finSi

    Si ("Espacio".equals(mapa[y][x])) entonces
        mapa[y][x] = nombre
        asignarPlanetas(y,x,nombre)
        indice++
    finSi

fin_mientras

indice = 0

mientras(indice < neutrales) hacer
    int y = (int) (Math.random()*altura)
    int x = (int) (Math.random()*anchura)

    Si (indice == 0) entonces
        nombre = "Neutral_A"
    finSi

    Si (indice == 1) entonces
        nombre = "Neutral_B"
    finSi

```

```

Si (indice == 2) entonces
    nombre = "Neutral_C"
finSi

Si (indice == 3) entonces
    nombre = "Neutral_D"
finSi

Si (indice == 4) entonces
    nombre = "Neutral_E"
finSi

Si (indice == 5) entonces
    nombre = "Neutral_F"
finSi

Si ("Espacio".equals(mapa[y][x])) entonces
    mapa[y][x] = nombre
    asignarPlanetas(y,x,nombre)
    indice++
finSi

fin_mientras

indice = 0;

```

```

mientras (indice < zombies) hacer
    int y = (int) (Math.random()*altura)
    int x = (int) (Math.random()*anchura)

    Si (indice == 0) entonces
        nombre = "Zombie_A"
    finSi

    Si (indice == 1) entonces
        nombre = "Zombie_B"
    finSi

    Si (indice == 2) entonces
        nombre = "Zombie_C"
    finSi

    Si (indice == 3) entonces
        nombre = "Zombie_D"
    finSi

    Si ("Espacio".equals(mapa[y][x])) entonces
        mapa[y][x] = nombre
        asignarPlanetas(y,x,nombre)
        indice++
    finSi
fin_mientras

```

Fin