

```

inicio clase Konquest {

    inicio public static void main(String[] args) {
        MENU M1 = nuevo MENU();
        M1.MENU()

    fin main

fin Konquest

```

```

inicio public class MENU

    var tipo Scanner leer = nuevo Scanner
    var tipo Scanner imprimir = nuevo Scanner

    inicio public void MENU() entonces

        var tipo int respuesta

        mostrar; BIENVENIDO A JUEGOS GYF
        Le presentamos nuestro nuevo juego llamado: Konquest, que consiste en lo siguiente:
        Dos jugadores pueden conquistar planetas, enviando naves con guerreros,
        para poder construir un gran imperio al conquistar todos los planetas disponibles.
        elija un numero de opcion
        1. Iniciar juego
        2. Salir del sistema
        respuesta = leer
        inicio si (respuesta = 1) entonces
            MenuInicio()
        fin si
        inicio pero si (respuesta = 2) entonces
            detener programa
        fin pero si
        inicio pero si no (respuesta != 1 y respuesta !=2) entonces
            mostrar; la opcion no existe
            MENU()
        fin pero si no

    fin MENU ()

    inicio public void MenuInicio() entonces
        var tipo int filas, columnas, planetasNeutrales
        var tipo String Jugador1, Jugador2
        var tipo DisenioMapas M1 = nuevo DisenioMapas()

        mostrar; ingrese el numero de filas para el mapa
        filas = leer

```

```
mosrear; ingrese el numero de columnas para el mapa
columnas = leer
mostrar; ingrese la cantidad de planetas neutrales para el mapa
planetasNeutrales = leer
mostrar; ingrese el nombre del Jugador numero 1
Jugador1 = imprimir
mostrar; ingrese el nombre del Jugador numero 2
Jugador2 = imprimir
mostrar; Diseño del mapa:
    1) Aleatorio
    2) Manual
var tipo int opcionMapa = leer
```

```
inicio switch (opcionMapa) entonces
    caso 1:
        M1.MapaAleatorio(filas, columnas, planetasNeutrales, Jugador1, Jugador2)
        break
    caso 2:
        M1.MapaManual(filas, columnas, planetasNeutrales, Jugador1, Jugador2)
        break
    default:
        mostrar La opcion no existe, elija una de las opciones
fin switch
fin MenuInicio
```

```
fin MENU
```

```
clase DisenioMapas
```

```
var enteras: filas, columnas, planetasNeutrales
var tipo String: Jugador1, Jugador
var tipo planeta[][] matriz
Random random = nuevo Random()
```

```
var tipo string nombre[] = [ABECEDARIO]
```

```
inicio MapaAleatorio(int filas, int columnas, int planetasNeutrales, String Jugador1, String Jugador2)
    constructor
    matriz = nuevo Planeta[filas][columnas]
    var tipo boolean bandera = true
```

```
var tipo entero n = 0
var tipo stringconquistador = "Neutral"
```

```
inicio ciclo while con var boolean
    var tipo decimal rMuerte = random de (0.1 a 0.9999)
    var tipo decimal pMuerte = random con dos decimales

    var tipo entero rDinero = random de (100 a 500)
```

```
var tipo entera rFilas = random de filas
var tipo entera rColumnas = random de columnas
var tipo entera rGeneral = random de (0 a 100)
```

```
inicio si (n = planetasNeutrales) {
    conquistador = Jugador1
fin si
inicio pero si (n = planetasNeutrales + 1) {
    conquistador = Jugador2
fin pero si
```

```
    inicio si (matriz[rFilas][rColumnas] instancia de Tierra O matriz[rFilas][rColumnas] instancia
de Fuego O matriz[rFilas][rColumnas] instancia de Organico O matriz[rFilas][rColumnas] instancia de
Radioactivo O matriz[rFilas][rColumnas] instancia de Agua) entonces
```

```
        fin si
```

```
        inicio peor si (matriz[rFilas][rColumnas] =null) entonces
```

```
            inicio si (rGeneral >= 0 y rGeneral < 45) entonces
```

```
                matriz[rFilas][rColumnas] = nuevo Tierra(nombre[n], pMuerte, rDinero, conquistador)
```

```
                n++
```

```
            fin si
```

```
            inicio pero si (rGeneral >= 45 y rGeneral < 60) entonces
```

```
                matriz[rFilas][rColumnas] = nuevo Fuego(nombre[n], pMuerte, rDinero, conquistador)
```

```
                n++
```

```
            fin pero si
```

```
            inicio pero si(rGeneral >= 60 y rGeneral < 70) entonces
```

```
                matriz[rFilas][rColumnas] = nuevo Organico(nombre[n], pMuerte, rDinero, conquistador)
```

```
                n++
```

```
            fin pero si
```

```
            inicio pero si(rGeneral >= 70 y rGeneral < 75) entonces
```

```
                matriz[rFilas][rColumnas] = nuevo Radioactivo(nombre[n], pMuerte, rDinero,
```

```
conquistador)
```

```
                n++
```

```
            fin pero si
```

```
            inicio pero si(rGeneral >= 75 y rGeneral < 100) entonces
```

```
                matriz[rFilas][rColumnas] = nuevo Agua(nombre[n], pMuerte, rDinero, conquistador)
```

```
                n++
```

```
            fin pero si
```

```
        fin si
```

```
        inicio si (n == planetasNeutrales + 2) entonces
```

```
            bandera = false;
```

```
        fin si
```

```
fin while
```

```
rellenarCampo
```

```
Juego j = uevo Juego(matriz)
```

```
j.menu2()
```

fin MapaAleatorio

inicio MapaManual(int filas, int columnas, int planetasNeutrales, String Jugador1, String Jugador2)
entonces

```
variable tipo Scanner leerInt = nuevo Scanner
variable tipo Scanner leerString = nuevo Scanner
this.filas = filas
this.columnas = columnas
matriz = nuevo Planeta[filas][columnas]
var tipo entero rGeneral = random (0 a 100)
var tipo decimal rMuerte = random de (0.1 a 0.9999)
var tipo decimal pMuerte = random con dos decimales
var tipo boolean bandera = true
Random random = nuevo Random()
```

```
var tipo entero n = 0;
var tipo String conquistador = "Neutral"
```

inicio ciclo while (bandera) entonces

inicio ciclo for (int i = 0; i < filas; i++) entonces

inicio ciclo for (int j = 0; j < columnas; j++) entonces

var tipo entero TipoPlaneta = 0

inicio si (rGeneral >= 0 && rGeneral < 45) entero

TipoPlaneta = 1

n++

fin si

inicio pero si (rGeneral >= 45 && rGeneral < 60) entonces

TipoPlaneta = 2

n++

fin pero si

inicio pero si(rGeneral >= 60 && rGeneral < 70) entonces

TipoPlaneta = 3

n++

fin pero si

inicio pero si(rGeneral >= 70 && rGeneral < 75) entonces

TipoPlaneta = 4

n++

fin pero si

inicio pero si (rGeneral >= 75 && rGeneral < 100) entonces

TipoPlaneta = 5

n++

fin peor si

mostrar; "Quien será el dueño del planeta " + i + "," + j

mostrar; 1) Neutral

2) Jugador1

3) Jugador2

var tipo entero NumConquistador = scanear con leerInt

```

var tipo String Conquistador = null
inicio switch (NumConquistador) entonces
    caso 1:
        Conquistador = "Neutral"
        break
    caso 2:
        Conquistador = Jugador1
        break
    caso 3:
        Conquistador = Jugador2
        break
    default:
        break
fin switch

```

```

mostrar; Escriba el nombre del planeta
var tipo String nombre = Scanear leerString

```

```

mostrar; Escriba la cantidad dinero que desea entre 100 y 500
var tipo entero Dinero = Scanear leerInt

```

```

    crearPlaneta(TipoPlaneta, nombre, Conquistador, Dinero, i, j)
fin ciclo for
inicio si (n = planetasNeutrales + 2) entonces
    bandera = false
fin si
fin ciclo for

```

```

fin ciclo while

```

```

rellenarCampo()
Juego j = nuevo Juego(matriz)
j.menu2()
fin MapaManual

```

```

inicio rellenarCampo entonces

```

```

    inicio ciclo for (int i = 0; i < filas; i++) entonces
        inicio ciclo for (int j = 0; j < columnas; j++) entonces
            inicio si (matriz[i][j] instanceof Tierra) entonces
                mostrar; "   " + matriz[i][j].getNombre() + matriz[i][j].getConquistador() + "   "
            fin si
            inicio pero si (matriz[i][j] instanceof Fuego) entonces
                mostrar "   " + matriz[i][j].getNombre() + matriz[i][j].getConquistador() + "   "
            fin pero si
            inicio pero si (matriz[i][j] instanceof Organico) entonces
                mostrar "   " + matriz[i][j].getNombre() + matriz[i][j].getConquistador() + "   "
            fin pero si
            inicio pero si (matriz[i][j] instanceof Radioactivo) entonces

```

```

        mostrar"    " + matriz[i][j].getNombre() + matriz[i][j].getConquistador() + "    "
    fin pero si
    inicio pero si(matriz[i][j] instanceof Agua) entonces
        mostrar "    " + matriz[i][j].getNombre() + matriz[i][j].getConquistador() + "    "
    fin pero si
    inicio pero si no entonces
        mostrar "    " + "_____ " + "    " + "    "
    fin pero si no
fin ciclo for

```

```

    fin ciclo for
fin rellenarCampo

```

```

    inicio crearPlaneta(int TipoPlaneta, String nombre, String Conquistador, int Dinero, int Fila, int
    Columna) entonces

```

```

        var tipo decimal rMuerte = random de (0.1 a 0.9999)
        var tipo decimal pMuerte = random con dos decimales

```

```

    inicio switch (TipoPlaneta) entonces

```

```

        caso 1:
            matriz[Fila][Columna] = nuevo Tierra(nombre, pMuerte, Dinero, Conquistador)
            break

```

```

        caso 2:
            matriz[Fila][Columna] = nuevo Fuego(nombre, pMuerte, Dinero, Conquistador)
            break

```

```

        caso 3:
            matriz[Fila][Columna] = nuevo Organico(nombre, pMuerte, Dinero, Conquistador)
            break

```

```

        caso 4:
            matriz[Fila][Columna] = nuevo Radioactivo(nombre, pMuerte, Dinero, Conquistador)
            break

```

```

        caso 5:
            matriz[Fila][Columna] = nuevo Agua(nombre, pMuerte, Dinero, Conquistador)
            break

```

```

        default:
            matriz[Fila][Columna] = null
            break

```

```

    fin switch
fin crearPlaneta

```

```

fin DisenioMapas

```

```

clase Juego entonces

```

```

    var tipo Scanner leer
    var tipo Planeta[][] matriz

```

```

    Juego(Planeta[][] matriz) entonces
        this.matriz = matriz

```

```
inicio menu2() entonces
  leer = nuevo Scanner
  mostrar Los jugadores pueden realizar las siguientes acciones:
    1) Medir una distancias
    2) Ver un planeta
    3) Consultar flota
    4) Enviar flota
    5) Construir una nave
    6) Ir a tienda
    7) Batalla
    Escriba el numero de la opcion que desee
  var tipo entero opcion = scanear leer
```

```
inicio switch (opcion) entonces
  caso 1:
    medirDistancia
    break
  caso 2:
    verPlaneta
    break
  caso 3:
    consultarFlota
    menu2
    break
  caso 4:
    enviarFlota
    menu2
    break
  caso 5:
    construirNave
    menu2
    break
  caso 6:
    irATienda
    menu2
    break
  caso 7:
    Batalla
    menu2
    break
  default:
    mostrar; No es una de las opciones, por favor elija de nuevo
    break
fin switch
fin menu2
```

```
inicio medirDistancia() entonces
    leer = nuevo Scanner
    mostrar La distancia entre planetas se mide en años luz,
        para esto ingrese la ubicacion del planeta origen y la ubicación del planeta de destino,
separadas por coma
    Ejemplo: 1,2. Indicando primero la fila y despues la columna
```

```
var tipo String Ubicacion = leer
var tipo String[] UbSeparada = Ubicacion.split(",")
```

```
menu2()
fin medir distancia
```

```
inicio verPlaneta() entonces
    leer = nuevo Scanner
    mostrar Ud solo puede ver sus planetas y los planetas neutrales,
        para ver uno por favor escriba la ubicacion del planeta separadas por coma
        Ejemplo: 1,2. Indicando primero la fila y despues la columna \nPor favor ingrese la
ubicacion:
```

```
var tipo String Ubicacion = leer
var tipo String[] UbSeparada = Ubicacion.split(",")
```

```
mostrar; El dueño es:
mostrar; matriz[Integer.parseInt(UbSeparada[0])]
[Integer.parseInt(UbSeparada[1])).getConquistador()
mostrar; La cantidad de dinero es:
mostrar; matriz[Integer.parseInt(UbSeparada[0]))[Integer.parseInt(UbSeparada[1])).getcDinero()
mostrar; El tipo de planeta es: =
mostrar; (matriz[Integer.parseInt(UbSeparada[0]))[Integer.parseInt(UbSeparada[1])).getTipo()
menu2()
fin verPlaneta
```

```
fin clase Juego
```

```
inicio public clase Nave
    String nave
    double velocidad
    int espacios, produccion, pVenta
```

```
inicio constructor; public Nave(String nave, double velocidad, int espacios, int produccion, int
pVenta)
    this.nave=nave
    this.velocidad=velocidad
    this.espacios=espacios
    this.produccion=produccion
    this.pVenta=pVenta
```


fin constructor

set's y get's

fin clase Nave

inicio public clase Naboo hereda Nave

```
public Naboo(){  
    super("Naboo N-1", 1.0, 25, 40, 10)
```

set's y get's

fin clase Naboo

inicio public clase StarDestroyer hereda Nave

```
public StarDestroyer()  
    super ("Star Destroyer", 1.75,80,100,25)  
set's y get's
```

fin StarDestroyer

inicio public clase XWing hereda Nave

```
public XWing()  
    super("X-Wing", 1.25, 42, 50, 15)
```

fin Xwing

inicio public clase MillenialFalcon herencia Nave

```
public MillenialFalcon()  
    super("Milenial Falcon", 1.50,58,70,20)
```

set's y get's

fin clase MillenialFalcon

inicio public class Guerrero entonces

```
var tipo String nombre  
var tipo double facMuerte  
var tipo int espacios  
var tipo String ataque
```

inicio constructor public void Guerrero(String nombre, double facMuerte, int espacios, String ataque) entonces

```
    this.nombre = nombre  
    this.facMuerte=facMuerte  
    this.espacios=espacios  
    this.ataque=ataque
```

fin constructor

set's y get's

fin clase Guerrero

inicio public abstract clase Planeta

var tipo String nombre
var tipo double pMuerte
var tipo int cDinero
var tipo String conquistador

constructor public Planeta (String nombre, double pMuerte, int cDinero, String conquistador)
this.nombre = nombre
this.pMuerte = pMuerte
this.cDinero = cDinero
this.conquistador = conquistador

set's y get's

fin clase Planeta

clase Tierra hereda Planeta {

public static final String ANSI_WHITE = "\u001B[37m"
public static final String ANSI_RESET = "\u001B[0m"

constructor Tierra (String nombre, double pMuerte, int cDinero, String conquistador) entonces
super (nombre, pMuerte, cDinero, conquistador)

set's y get's

fin Tierra

inicio clase Radioactivo hereda Planeta

public static final String ANSI_YELLOW = "\u001B[33m"
public static final String ANSI_RESET = "\u001B[0m"

constructor Radioactivo (String nombre, double pMuerte, int cDinero, String conquistador)
super (nombre, pMuerte, cDinero, conquistador);

set's y get's

fin Radioactivo

inicio clase Organico hereda Planeta

public static final String ANSI_GREEN = "\u001B[32m"
public static final String ANSI_RESET = "\u001B[0m"

```
    constructorOrganico ( String nombre, double pMuerte, int cDinero, String conquistador)
        super (nombre, pMuerte, cDinero, conquistador)
    set's y get's
```

```
fin Organico
```

```
inicio clase Fuego hereda Planeta
```

```
public static final String ANSI_RED = "\u001B[31m"
public static final String ANSI_RESET = "\u001B[0m"
```

```
    constructor Fuego( String nombre, double pMuerte, int cDinero, String conquistador )
        super (nombre, pMuerte, cDinero, conquistador);
```

```
    set's y get's
```

```
fin Fuego
```

```
inicio clase Agua hereda Planeta
```

```
    public static final String ANSI_BLUE = "\u001B[34m"
```

```
    public static final String ANSI_RESET = "\u001B[0m"
```

```
    constructor Agua(String nombre, double pMuerte, int cDinero, String conquistador)
        super(nombre, pMuerte, cDinero, conquistador)
```

```
    set's y get's
```

```
fin Agua
```