

# **DOCUMENTACIÓN SÚPER AUTO PETS**

El juego llamado Súper Auto Pets es un juego que funciona en consola y fue desarrollado en el lenguaje java versión 15, se utilizo como herramienta el IDE NetBeans y para control de paquetes maven. Para ejecutar el juego según su sistema operativo puede realizar lo siguiente:

Paquete com.proyecto1.principal

Clase Main

La clase main se encarga de iniciar juego, en esta clase no se cuenta con atributos y únicamente se encarga de mostrar el título del juego y de instanciar el método menuPrincipal que se encuentra en la clase MenuPrincipal.

Inicio Main

Escribir "\*\*\*\*\*"

Escribir "\*SÚPER AUTO PETS\*"

Escribir "\*\*\*\*\*"

Llamar menuPrincipal()

Fin

Clase MenuPrincipal

Esta clase se encarga de mostrar las distintas opciones que tiene el usuario dentro del juego, posee un atributo de tipo entero llamado opcion, dentro de las opciones del menú se encuentran modos de juego, reportes y salir, para capturar la opcion que selecciono el usuario se guarda en el atributo opcion, seguidamente se analiza esta opción en la sentencia de control switch donde en el caso 1 se creara un objeto MenuModoJuego para seguidamente instanciar el método llamado mostrarModosDeJuego(), en el caso 2 se creara el objeto de ReportesDeJuego y se hará uso del método mostrarEstadísticas y en el caso 3 mostrará un mensaje al usuario indicándole que esta saliendo del juego y se finalizará la ejecución, en caso que el usuario ingrese un número que no corresponda a alguna de las opciones indicadas se le mostrará un mensaje de que el valor ingresado es incorrecto y se le mostrará el menú de opciones nuevamente, esto es posible utilizando el ciclo do while.

Inicio Proceso MenuPrincipal

Var opcion

Hacer

Escribir "-MENÚ PRINCIPAL"

Escribir "1. Arena"

Escribir "2. Versus"

Escribir "3. Creativo"

Escribir "4. Reportes"

Escribir "5. Salir"

Leer opcion

caso 1:

    crear objeto de la clase Arena  
    Llamar comenzarBatalla()

caso 2:

    crear objeto de la clase Versus  
    Llamar comenzarBatalla()

caso 3:

    crear objeto de la clase Creativo  
    Llamar comenzarBatalla()

caso 4:

    crear objeto Reportes  
    Llamar estadisticasYReportes()

caso 5:

    Escribir "Saliendo del Juego"

    mientras (opcion !=3)

        Fin

## Paquete modosdejuego

### Clase MenuEntreBatalla

Esta clase se encarga de mostrar todas las opciones que el usuario puede realizar para conformar su equipo que al terminar el turno comenzará la batalla, dentro de esta clase se tienen los métodos aplicarEfectoPermanente(), comprarMascotas(), comprarComida(), ordenarMascotas(), seleccionarCampo(), venderMascotas(), fusionarMascotas().

#### Método aplicarEfectoPermanente()

Al momento de que se compre un alimento y se le de a alguno de los animales dentro del equipo del usuario este será de manera permanente el cual solo cambiara al momento de darle otro alimento que contenga efecto, esto es así debido a que un animal no puede tener más de un efecto en juego.

    Inicio Proceso aplicarEfectoPermanente()

        Fin

#### Método comprarMascotas()

Este método se encarga de mostrar al usuario los animales disponibles por Tier en la tienda, si en caso no aparece un animal de interes para el usuario entonces puede cambiar las opciones que se muestran, si en caso quiere comprar uno se debe validar que el usuario tenga suficiente dinero, luego validar si existe espacio disponible dentro del arreglo de animales y si todo lo anterior se cumple entonces se le mostrará al usuario un mensaje de que la compra ha sido realizada y se le mostrará cuantas monedas le quedan disponibles. En este método al momento de cambiar los animales mostrados también se cambiará los animales mostrados

Inicio Proceso comprarMascotas()

Var final COSTO = 3

Var opcion

Var compra

Hacer

Escribir “1. Comprar Mascotas”

Escribir “2. Cambiar opciones de la tienda”

Leer (opcion)

Usuario usuario = new Usuario()

Si (opcion == 1) entonces

Si (usuario.getMonedas > 1) entonces

Escribir tienda tienda = new Tienda()

Escribir “Seleccione la mascota que desee comprar”

Leer (compra)

//Usar metodo de busqueda para seleccionar el animal a comprar

Si (usuario.getMonedas >= 3 ) entonces

Escribir “Compra realizada con éxito”

sino

Escribir “No se compro ningún animal, no cuentas con suficientes monedas”

Finsi

Finsi

sino

Si (usuario.getMonedas > 1 ) entonces

//hacer un random y mostrar mascotas

sino

Escribir “No cuentas con suficientes monedas para realizar esta acción”

Finsi

Finsi

Mientras (opcion > 2 )

Fin

Método comprarComida()

Este método tiene un comportamiento similar al de comprarMascotas(), la diferencia es de que únicamente se mostrarán dos espacios donde se mostraran los alimentos disponibles según el Tier.

Inicio Proceso comprarComida()

Var opcion

Escribir “Elija el alimento que desea comprar”

Leer (opcion)

Si (usuario.getOro() > 3) entonces

Hacer

Si (opcion == 1) entonces

Escribir “Elija el alimento que desee comprar.”

Comida comida = new Comida()

comida.mostrarComidaAleatoria

Leer (Var compra)

Escribir “A que animal le va a aplicar el alimento?”

Leer (Var alimento)

Finsi

Mientras (opcion > 2)

sino

Escribir “No cuenta con suficientes monedas para realizar la compra”

Finsi

Fin

Método ordenarMascotas()

En este método el usuario elije la forma en que colocará los animales de su equipo para que pueda emplear la estrategia que quiera aplicar al momento de entrar en batalla, para el ordenamiento se utiliza el algoritmo llamado algoritmo de ordenamiento

Método venderMascotas()

Método que se encarga de realizar la venta de la mascota o mascotas que el usuario vaya a vender, según el nivel en el que se encuentre el animal será el valor con el cual se define con validaciones.

```
Inicio Proceso venderMascotas()
Var venderAnimal
Escribir "Elija el animal que desea vender"
Leer (venderAnimal)
Usuario usuario = new Usuario()

Si (venderAnimal == 1) entonces
    usuario.setOro(usuario.getOro+1)
    Escribir "Monedas obtenidas por venta: 1"
sino si (venderAnimal = 2) entonces
    usuario.setOro(usuario.getOro + 2)
    Escribir "Monedas obtenidas por venta: 2"
sino
    usuario.setOro(usuario.getOro + 3)
    Escribir "Monedas obtenidas por venta: 3"
Finsi
Fin
```

Método fusionarMascotas()

Por medio de este método se puede fusionar a los animales que tengan el mismo tipo y así poder subirla de nivel, este método realiza una validación que hace que los animales que el usuario quiera fusionar sean del mismo tipo y de lo contrario mostrará un mensaje que indique el error.

```
Inicio Proceso fusionarMascotas()
Var animalSeleccionado

Escribir "Seleccione el animal de la tienda que desea fusionar"
Tienda animales = new Tienda()

Si ()
Fin
```

## Clase Tier

En esta clase se encuentran métodos que se encargan de desbloquear el tier correspondiente a la ronda en la que se encuentra el jugador, el método que se encarga de gestionar la manera de desbloqueo se llama desbloquearTierMascotas().

### Método desbloquearTierMascotas()

Se encarga de desbloquear que animales se estarán disponibles para que en la clase Tienda se pueda mostrar y crear aleatoriamente las mascotas que el usuario pueda comprar.

Inicio Proceso desbloquearTierAnimales(Var ronda)

    Mascotas[] tier

    Si (ronda == 1 && ronda == 2) entonces

        Mascota hormiga = new Mascota("Hormiga", 2, 1, "Da a un aliado al azar (+2/+1)/(+4/+2)/(+6/+3) al morir", "Insecto / Terrestre")

        Mascota pescado = new Mascota("Pescado", 2, 3, "Power-up: Da a todos los aliados (+1/+1)/(+2/+2) al subir de nivel", "Acuatico")

        Mascota mosquito = new Mascota("Mosquito", 2, 2, "Piquete inicial: Al iniciar batalla realiza 1 de daño a (1)/(2)/(3) enemigos", "Volador")

        Mascota grillo = new Mascota("Grillo", 1, 2, "Zombificación: Convoca a un grillo zombie con estadísticas: (1/1)/(2/2)/(3/3) al morir", "Insecto")

        Mascota castor = new Mascota("Castor", 2, 2, "Represa: Da a 2 aliados al azar +1/+2/+3 HP al venderse", "Terrestre / Acuatico")

        Mascota caballo = new Mascota("Caballo", 2, 1, "Rugido aliado: Da +1/+2/+3 ATK a los aliados invocados", "Mamífero/ Doméstico")

        Mascota nutria = new Mascota("Nutria", 1, 2, "Ventaja económica: Da a un aliado aleatorio (+1/+1)/(+2/+2)/(+3/+3) al comprarse", "Mamífero")

        Mascota escarabajo = new Mascota("Escarabajo", 2, 3, "Apetito: Otorga a las mascotas de la tienda +1/+2/+3 HP al comer comida de la tienda", "Insecto")

tier[0] = hormiga.mostrarDatos()

tier[1] = pescado.mostrarDatos()

tier[2] = mosquito.mostrarDatos()

tier[3] = grillo.mostrarDatos()

tier[4] = castor.mostrarDatos()

tier[5] = caballo.mostrarDatos()

tier[6] = nutria.mostrarDatos()

tier[7] = escarabajo.mostrarDatos()

sino si (ronda == 3 && ronda == 4) entonces

Mascota sapo = new Mascota("Sapo", 3, 3, "Metamorfosis: Copia la salud del aliado más saludable", "Terrestre / Acuatico")

Mascota dodo = new Mascota("Dodo", 2, 3, "División de poder: Da al aliado de adelante 50%/100%/150% ATK al iniciar la batalla", "Terrestre / Acuatico")

Mascota elefante= new Mascota("Elefante", 3, 5, "Daño colateral: Inflige 1 daño a 1/2/3 amigos detrás antes de atacar", "Terrestre / Mamífero")

Mascota puercoEspin = new Mascota("Puerco Espin", 3, 2, "Espinazos salvajes: Repartir 2/4/6 dmg a todas las mascotas al morir", "Solitario/ Terrestre")

Mascota pavoreal = new Mascota("Pavoreal", 2, 5, "Espinazos salvajes: Repartir 2/4/6 dmg a todas las mascotas al morir", "Solitario/ Terrestre")

Mascota rata = new Mascota("Rata", 4, 5, "Ayuda hipócrita: invoca 1/2/3 1/1 Dirty Rats al frente para el oponente al morir", "Terrestre/ Solitario")

Mascota zorro = new Mascota("Zorro", 5, 2, "Ataque Rápido: Ataca 2 veces seguidas cada (3 turnos)(2 turnos)(1 turno)", "Solitario/ Terrestre")

Mascota araña = new Mascota("Araña", 3, 2, "Liberación: invoca una mascota de nivel 1/2/3 de nivel 3 como 2/2 al morir", "Insecto")

sino si (ronda == 5 && ronda == 6) entonces

Mascota camello = new Mascota ("Camello", 2, 5, "Joroba: Dar amigo detrás (+1/+2)/ (+2/+4) / (+3/+6)", "Mamífero/Doméstico")

Mascota mapache = new Mapache("Mapache", 5, 4, "Repartir poder: Otorga 1x/2x/3x ATK a las mascotas adyacentes al morir", "Solitario")

Mascota jirafa = new Mascota("Jirafa", 2, 5, "Fortaleza aliada: Da 1/ 2 / 3 amigos por delante +1/+1 al finalizar el turno de compra", "Mamífero/Terrestre")

Mascota tortuga = new Mascota("Tortuga", 1, 2, "Protección aliada: Dar 1/2/3 amigos detrás de Melón Armor al morir", "Reptil")

Mascota caracol = new Mascota("Caracol", 2, 2, "Resurgir: si perdiste la última batalla, dales a todos los amigos (+1/+1)/ (+2/+2) / (+3/+3) al comprar", "Insecto/Solitario")

Mascota oveja = new Mascota("Oveja", 2, 2, "Revolución: invoca dos (2/2)/ (4/4) / (6/6) carneros al morir", "Doméstico/Terrestre")

Mascota conejo = new Mascota("Conejo", 3, 2, "Cariño: Cuando un amigo come comida de la tienda: Le da +1/ +2 / +3 HP", "Mamífero")

Mascota lobo = new Mascota ("Lobo", 3, 4, "Aullido: Si es el último con vida gana (+2/+2)(+3,+3) (+5,+5) permanentemente", "Solitario/Terrestre")

Mascota buey = new Mascota("Buey", 1, 4, "Mejor me protejo: gana armadura de melón y +2/ +4 + 6 ATK cuando la mascota aliada de delante se debilita", "Mamífero")

Mascota canguro = new Mascota("Canguro, 1, 2, "Ya quiero pelear: Ganancia (+2/+2)/ (+4/+4) / (+6/+6) cada que la mascota aliada de delante ataque", "Mamífero/Terrestre")

Mascota buho = new Mascota("Buho", 5, 3, "Suerte amigos: Dar un amigo al azar (+2/+2)/ (+4/+4) / (+6/+6) al vender", "Volador/Solitario")

```
tier[0] = sapo.mostrarDatos()  
tier[1] = dodo.mostrarDatos()  
tier[2] = elefante.mostrarDatos()  
tier[3] = puerco.mostrarDatos()  
tier[4] = pavoreal.mostrarDatos()  
tier[5] = rata.mostrarDatos()  
tier[6] = zorro.mostrarDatos()  
tier[7] = araña.mostrarDatos()
```

sino si (ronda == 7 && ronda == 8) entonces

Mascota venado = new Mascota("Venado", 1, 1, "Venganza: Invoca un (5/5)/(10/10)/(15/15) autobus con Splash attack al morir", "Mamífero")

Mascota loro = new Mascota("Loro", 5, 3, "Copia: al finalizar el turno de compra copia la habilidad del amigo de adelante como lvl 1/2/3 hasta el final de la batalla", "Volador")

Mascota hipopotamo = new Mascota("Hipopótamo", 4, 7, "Robustez: Ganancia (+2/+2)/ (+4/+4) / (+6/+6) al ser él hipopotamo quien debilita a una mascota enemiga", "Acuático/Terrestre")

Mascota delfin = new Mascota("Delfín", 4, 6, "Salpicón: reparte 5/10/15 de daño al enemigo con la salud más baja al comenzar la batalla", "Acuático")

Mascota puma = new Mascota("Puma", 3, 7, "Sigilo: Al cabo de (3)(2)(1) turnos siendo dañado deja una copia de 1/1, mientras está la copia el puede atacar", "Mamífero/Terrestre")

Mascota ballena = new Mascota("Ballena", 3, 8, "Succión: Al iniciar la partida trague al aliado de delante y suéltalo como lvl 1/2/3 después de debilitarse", "Acuático")

Mascota ardilla = new Mascota("Ardilla", 2, 5, "Rebajas: al iniciar el turno de compra realiza un descuento en la compra de alimentos por 1/2/3 de oro", "Doméstico")

Mascota llama = new Mascota("Llama", 3, 6, "Fortaleza individual: al finalizar el turno de compra si tienes 4 mascotas o menos, gana (+2/+2)/ (+4/+4) / (+6/+6)", "Terrestre")

```
tier[0] = venado.mostrarDatos()  
tier[1] = loro.mostrarDatos()  
tier[2] = hipopotamo.mostrarDatos()  
tier[3] = delfin.mostrarDatos()  
tier[4] = puma.mostrarDatos()  
tier[5] = ballena.mostrarDatos()  
tier[6] = ardilla.mostrarDatos()  
tier[7] = llama.mostrarDatos()
```

sino si (ronda == 9 && ronda == 10) entonces

Mascota foca = new Mascota("Foca", 3, 8, "Compartir poder: Al recibir comida de la tienda da a 2 amigos randoms (+1/+1)(+2/+2)(+3/+3)", "Acuático/Mamífero")

Mascota jaguar = new Jaguar("Jaguar", 7, 4, "Venganza felina: Si el que está delante es atacado, daña al que lo atacó por (5)(10)(15) de daño", "Mamífero/Terrestre")

Mascota escorpion = new Mascota("Escorpión", 1, 1, "Aguja; tiene un ataque de veneno innato (el veneno ejecuta a la mascota enemiga sin importar cuánta vida tenga)", "Desértico/Solitario")

Mascota rinoceronte = new Mascota("Rinoceronte", 5, 8, "Estampida: Infinge 4/8/12 de daño al primer enemigo", "Desértico/Terrestre")

Turno Final Mascota mono = new Mascota("Mono", 1, 2, "Amistad: Dar al amigo más a la derecha (+2/+3)/(+4/+6) / (+6/+9)", "Mamífero")

Mascota cocodrilo = new Mascota("Cocodrilo", 8, 4, "Mordida: (Comienzo de la batalla) infinge 8/16/24 de daño al último enemigo", "Reptil/Solitario")

Mascota vaca = new Mascota("Vaca", 4, 6, "Leche potenciadora: al comprar reemplaza la tienda de comida con leche gratis que da (+1/+2)/(+2/+4)/(+3/+6)", "Mamífero/Terrestre")

Mascota chompipe = new Mascota("Chompipe", 3, 4, "Solidaridad: Le da (+3/+3)(+6/+6)(+9/+9) a un aliado invocado", "Terrestre/Volador")

tier[0] = foca.mostrarDatos()

tier[1] = jaguar.mostrarDatos()

tier[2] = escorpion.mostrarDatos()

tier[3] = rinoceronte.mostrarDatos()

tier[4] = mono.mostrarDatos()

tier[5] = cocodrilo.mostrarDatos()

tier[6] = vaca.mostrarDatos()

tier[7] = chompipe.mostrarDatos()

sino si (ronda == 11 && ronda == 12) entonces

Mascota panda = new Mascota("Panda", 5, 5, "Fortaleza: Absorbe el (50% / 60% / 80%) del daño recibido", "Mamífero/Solitario")

Mascota gato = new Mascota("Gato", 4, 5, "Maullido: Multiplica el efecto HP y ATK de la comida por 2/3/4", "Mamífero/Doméstico")

Mascota tigre = new Mascota("Tigre", 4, 3, "Repetición: El amigo de adelante repite su habilidad en la batalla como si fuera de nivel 1/2/3", "Terrestre/Mamífero")

Mascota serpiente = new Mascota("Serpiente", 6, 6, "Ataque discreto: infinge 5/10/15 de daño a un enemigo aleatorio cuando una mascota aliada de delante ataca", "Reptil/Terrestre/Desértico")

Mascota mamut = new Mascota("Mamut", 3, 10, "Fuerza compañeros: al desmayarse dará a todos los amigos (+2/+2)(+4/+4)(+6/+6)", "Mamífero/Terrestre/Solitario")

Mascota leopardo = new Mascota("Leopardo", 10, 4, "Zarpazo: al iniciar la batalla infinge 50 % de daño ATQ a 1/2/3 enemigos aleatorios", "Mamífero/Terrestre")

Mascota gorila = new Mascota("Gorila", 6, 9, "Escudo: al recibir daño gana escudo de coco ( 1/2/3 veces por batalla)", "Mamífero/Terrestre")

Mascota pulpo = new Mascota("Pulpo", 8, 8, "Habilidades por nivel (1) Subir de nivel: Gana +8/8. (2) Subir de nivel: gana +8/+8 y una nueva habilidad al azar (3) Antes del ataque: inflige 5 de daño a todos los enemigos", "Acuático/Solitario")

Mascota mosca = new Mascota("Mosca", 5, 5, "Invocación: cuando un aliado se debilita invoca un (5/5)/ (10/10) / (15/15) Zombie Fly en su lugar (tres veces por batalla)", "Volador/Insecto")

tier[0] = foca.mostrarDatos()

tier[1] = jaguar.mostrarDatos()

tier[2] = escorpion.mostrarDatos()

tier[3] = rinoceronte.mostrarDatos()

tier[4] = mono.mostrarDatos()

tier[5] = cocodrilo.mostrarDatos()

tier[6] = vaca.mostrarDatos()

tier[7] = chompipe.mostrarDatos()

sino si (ronda > 12) entonces

Mascota quetzal = new Mascota("Quetzal", 10, 10, "Habilidades por nivel (1) Agrega a su vida la suma de toda la vida de los animales tipos aves (2) Hace lo del nivel 1 y agrega a su daño la suma de todo el daño del daño de todas las aves (3) Hace lo del nivel 2 pero con todos los animales", "Volador/Solitario")

Mascota camaleon = new Mascota("Camaleón", 8, 8, "Habilidades por nivel (1) Copia la vida del enemigo más fuerte (2) Copia la vida y el daño del enemigo más fuerte (3) Copia la vida, el daño y la habilidad del enemigo más fuerte", "Reptil/Solitario")

tier[0] = quetzal.mostrarDatos()

tier[1] = camaleon.mostrarDatos()

Finsi

Fin

Metodo desbloquearTierComida(Var ronda)

El comportamiento es similar al que se encarga de desbloquear el tier de animales, la diferencia es que el arreglo que se va a mostrar es de comida las cuales se podrán dar a las mascotas en el menú entre batallas.

Inicio Proceso desbloquearTierComida()

comida[0] = new Manzana("Manzana", "Da 1 de vida y 1 de daño a un animal seleccionado", 1, 1);

comida[1] = new Naranja("Naranja", "Efecto", "Hace que regrese un 10% de daño");

comida[2] = new Miel("Miel", "Efecto", "Cuando la mascota a la que se le dio la miel muere, es remplazada con una abeja de 1 de vida y 1 de daño");  
 comida[3] = new Pastel("Pastelito", "Da 3 de vida y 3 de daño por la siguiente ronda de pelea", 3, 3);  
 comida[4] = new HuesoDeCarne("Hueso de Carne", "Efecto", "Da al animal 5 de daño extra", 0, 5);  
 comida[5] = new PastillaParaDormir("Pastilla para dormir", "Hace que nuestra mascota se desmaye en la fase de menú entre batalla", 0, 0);  
 comida[6] = new Ajo("Ajo", "Efecto", "Da armadura que hace que reciba 2 de daño menos", 2, 0);  
 comida[7] = new Ensalada("Ensalada", "Da 1 de vida y 1 de daño a 2 animales aleatorios del equipo", 1, 1);  
 comida[8] = new ComidaEnlatada("Comida enlatada", "Da a los animales que estan en la tienda y proximos que aparezcan 1 de vida y 2 de daño", 1, 2);  
 comida[9] = new Pera("Pera", "Da al animal escogido 2 de vida y 2 de daño", 2, 2);  
 comida[10] = new Chile("Chile", "Efecto", "Hace que el animal portador haga 5 de daño al animal que se encuentra detrás del que atacó");  
 comida[11] = new Chocolate("Chocolate", "Da 1 de experiencia a un animal de tu equipo", 0, 0);  
 comida[12] = new Sushi("Sushi", "Da 1 de vida y 1 de daño a 3 animales aleatorios del equipo", 1, 1);  
 comida[13] = new Melon("Melón", "Efecto", "El animal que lo porte no recibira daño la primera vez que sea atacado");  
 comida[14] = new Hongo("Hongo", "Efecto", "Cuando el animal muere vuelve a aparecer con 1 de vida y 1 de daño");  
 comida[15] = new Pizza("Pizza", "Da 2 de vida y 2 de daño a 2 animales aleatorios del equipo", 2, 2);  
 comida[16] = new Carne("Carne", "Efecto", "Hace que el animal que lo porte haga 20 de daño adicional 1 vez por ronda", 0, 20);  
 comida[17] = new Gelatina("Gelatina", "Efecto", "El animal que lo porte puede tener un tipo extra");  
  
 Si (ronda >= 1 && ronda <= 3) entonces  
 Si (ronda == 1 || ronda == 2) entonces  
 comidaMostradaPorRonda = 3  
 Desde 0; hasta i < CANTIDAD\_COMIDA\_MOSTRADA: 1  
 Var indice = Llamar funcion aleatorio() \* comidaMostradaPorRonda  
 Escribir "comida[indice]"  
 findesde  
 sino si (ronda == 3) entonces  
 comidaMostradaPorRonda = 6;  
 Desde i = 0; hasta i < CANTIDAD\_COMIDA\_MOSTRADA: 1  
 Var indice = Llamar funcion aleatorio() \* comidaMostradaPorRonda);  
 Escribir "comida[indice]"  
 findesde

finsi

sino si (ronda >= 4 && ronda <= 6) entonces

Si (ronda == 4) entonces

comidaMostradaPorRonda = 6

Desde 0; hasta i < CANTIDAD\_COMIDA\_MOSTRADA; 1

Var indice = Llamar funcion aleatorio() \* comidaMostradaPorRonda

“Escribir “comida[indice]”;

findesde

sino si (ronda == 5 || ronda == 6) entonces

comidaMostradaPorRonda = 10;

Desde 0; hasta i < CANTIDAD\_COMIDA\_MOSTRADA; 1

Var indice = Llamar funcion aleatorio()\* comidaMostradaPorRonda

Escribir “comida[indice]”

findesde

finsi

sino si (ronda >= 7) {

Si (ronda == 7 || ronda == 8) entonces

comidaMostradaPorRonda = 13

Desde i = 0; hasta i < 5; 1 entonces

Var indice = (int) (Math.random() \* comidaMostradaPorRonda

Escribir “comida[indice]”

sino si (ronda == 9 || ronda == 10) entonces

comidaMostradaPorRonda = 15

Desde 0; hasta i < 5; 1

Var indice = Llamar funcion aleatorio()\* comidaMostradaPorRonda

Escribir “comida[indice]”

sino si (ronda == 11 || ronda == 12) entonces

comidaMostradaPorRonda = 17

Desde (int i = 0; i < 5; 1

Var indice = Llamar funcion aleatorio()\* comidaMostradaPorRonda

Escribir “(comida[indice]”

sino si (ronda > 12) entonces

comidaMostradaPorRonda = 18

```

    Desde 0; hasta i < 5; 1
    Var indice = Llamar funcion aleatorio() * comidaMostradaPorRonda
    Escribir "(comida[indice])"
finsi
Finsi
Fin

```

## Clase Tienda

En esta clase se crean los animales los cuales se mostrarán en el menú entre batallas, gestiona aleatoriamente que animales se mostrarán por tier y cantidad de animales que se mostrarán al usuario. La clase cuenta con los métodos mostrarMascotasPorRonda(), mostrarMascotasAleatorias(), mostrarAlimentosRandom(), mostrarMascotas().

### Método mostrarMascotasAleatorias(Var ronda)

Se encarga únicamente de llamar al método tier y va a mostrar por ronda los animales y cantidad de animales requeridos por las reglas de juego

```

Inicio Proceso mostrarMascotasAleatorias(Var ronda)
    crear objeto Tier mostrarTier
    mostrarTier.desbloquearTierAnimales()
Fin

```

### Método mostrarComidaAleatoria()

Se encarga de llamar al método desbloquearTierComida() que se encarga de mostrar por tier la comida que se mostrará en tienda para que el usuario y darselo a los animales.

```

Inicio Proceso mostrarComidaAleatoria()
    crear objeto Tier mostrarTier
    mostrarTier.desbloquearTierComida()
Fin

```

## Paquete com.proyecto1.jugadores

### Clase Avatar

Esta clase es la clase padre de las clases Computadora, Usuario y JugadorImportado, cada clase define el comportamiento que tendrán estos al momento de comenzar comenzar una batalla dentro de las opciones disponibles y que estos mismo métodos y atributos también poseen las demás clases. Dada las reglas del juego el usuario como la IA tendrán inicialmente una vida de 10 puntos, victorias acumuladas que será iniciada en 0 y un

arreglo donde se almacenarán los animales que conformarán el equipo. Los método que posee esta clase y que va a heredar a las demás clases son perderVida() y ganarTrofeo y construirEquipo().

#### Método construirEquipo()

A través de este método el usuario podrá almacenar la cantidad de 5 animales los cuales ya habrá comprado en el menú entre batallas

Inicio Proceso construirEquipo()

Var posicion

Hacer

Escribir "En qué posición desea guardar el animal comprado: "

Leer posicion

caso 1:

Mientras (posicion > 5)

super.equipo[]

Fin

Paquete com.proyecto1.jugadores

#### Clase Animal

Esta clase es la clase padre que va a heredar métodos y atributos que serán utilizados y sobreescribirlos por las clases hijas, dentro de los atributos se tiene nombre, unidadDeDaño, unidadDeVida, un arreglo de tipo, experiencia.

Inicio Proceso Animal()

Var nombre

Var unidadDeDaño

Var unidadDeVida

Var tipo[]

metodo constructor()

Fin