UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1

DOCENTE:
ING. JOSÉ MOISÉS GRANADOS GUEVARA

PROYECTO 1
MANUAL TÉCNICO

ESTUDIANTE:
JORGE ANIBAL BRAVO RODRÍGUEZ

CARNÉ:
202131782

PRIMER SEMESTRE 2022

1. INFORMACIÓN DEL DESARROLLO

El desarrollo de esta práctica fue realizado en el sistema operativo Microsoft Windows en su versión 11, debido a problemas de rendimiento y estabilidad con el sistema operativo Ubuntu Linux. Windows es un sistema operativo de código cerrado desarrollado por la empresa estadounidense Microsoft que no pertenece a la familia de sistemas operativos de tipo UNIX, como Linux, BSD o mac OS. La portabilidad del proyecto a otros sistemas operativos viene dada por la propia plataforma de desarrollo que fue utilizada.

Oracle JDK 15: Esta es una versión de OpenJDK lanzada y mantenida por Oracle. Junto a este también se hizo uso de Maven como gestor-constructor de proyectos.

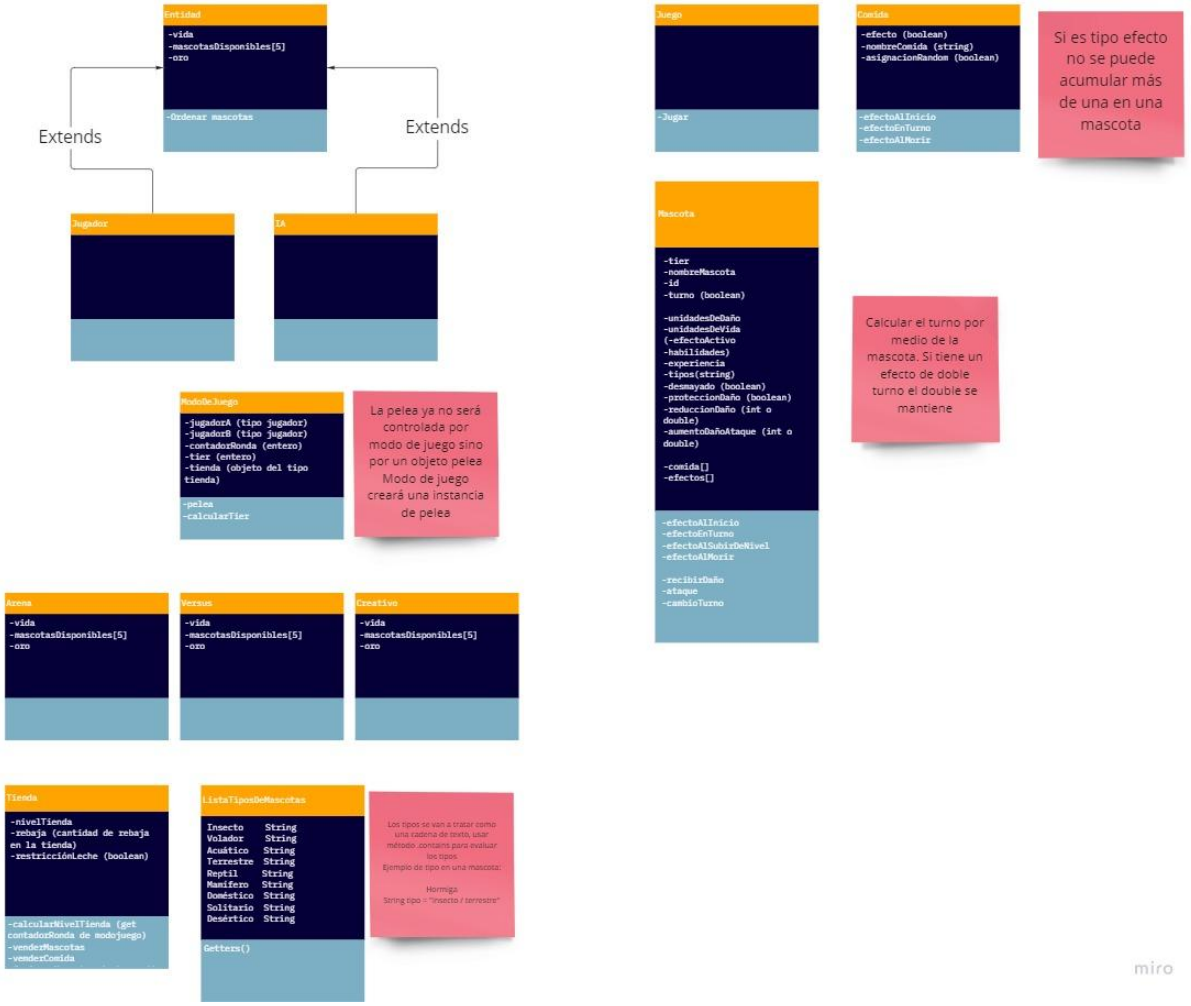
IntelliJ IDEA: Es un IDE desarrollado por la empresa checa Jetbrains, cuyo propósito principal es el desarrollo de software en los lenguajes Java y Kotlin, aunque puede extender su funcionalidad mediante plugins. Este entorno de desarrollo fue escogido por comodidad.

Visual Studio Code: Este es un editor de código basado en tecnologías web que permite un desarrollo de código y un entorno de desarrollo más ligero que el propio IntelliJ. Utilicé esta herramienta cuando solamente necesitaba escribir código con las ayudas del editor de texto.

Requisitos: Al ser un programa desarrollado en Java, una herramienta multiplataforma, se extiende su compatibilidad a la compatibilidad de Java en cuanto a sistema operativo y a requisitos mínimos para ejecutar Java.
El juego no es demandante en cuanto a recursos, por lo que los requisitos mínimos son, en realidad, los requisitos mínimos de un dispositivo que sea capaz de ejecutar Java, sea arquitectura ARM, x86, RISC, etc.
Sucede lo mismo con el sistema operativo, se requiere de un sistema operativo que sea soportado por Java para ejecutar JVM en modo consola. Puede funcionar con Windows, Mac OS, distribuciones GNU/Linux, distribuciones basadas en BSD, Solaris, entre otros.

Se recomienda usar Java 15 o superior, puesto que es en esta versión del Java Development Kit que se realizó el desarrollo.

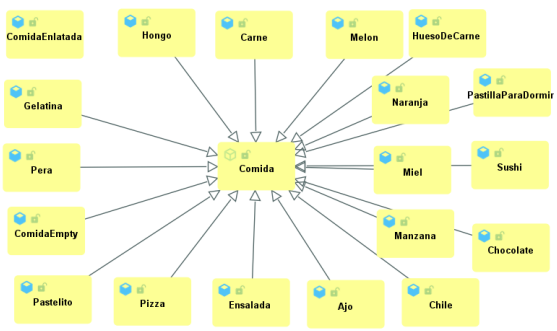
2. DISEÑO DE DIAGRAMAS UML

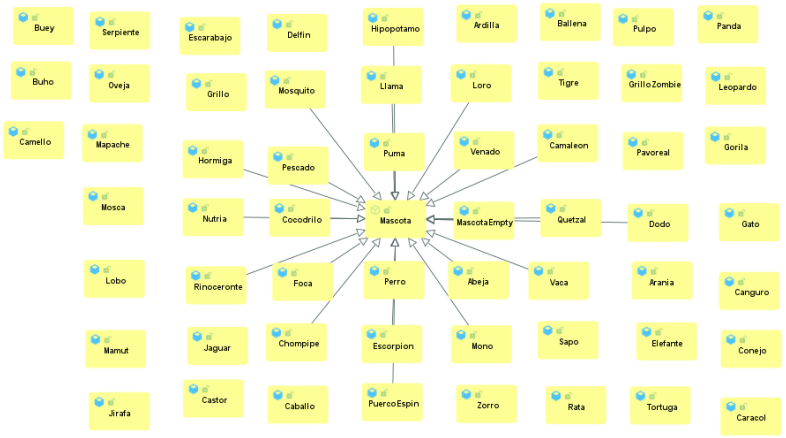


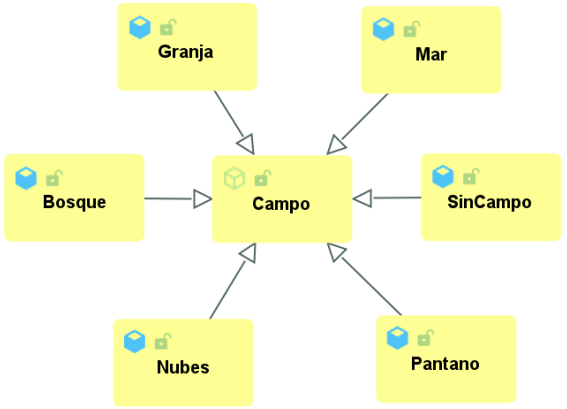
El primer intento para diseñar los diagramas UML que correspondieran con el desarrollo del programa fue hecho en una plataforma online llamada Miro, siendo altamente flexible y ofreciendo un buen rendimiento ejecutándose en navegadores web.

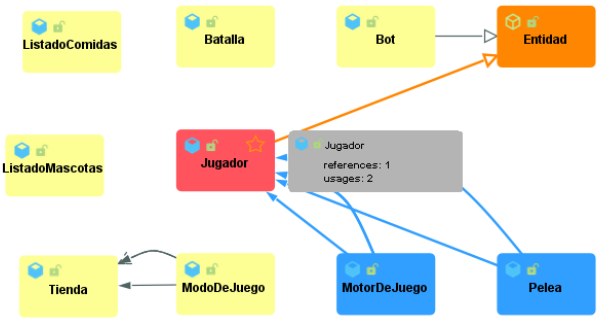
Sin embargo, estos diagramas sufrieron bastantes modificaciones a lo largo del desarrollo, esto es debido a la familiarización que iba obteniendo con el lenguaje a medida que iba implementando nuevas funcionalidades.
Esto trajo como consecuencia que necesitara reescribir muchas partes del código en más de una ocasión para poder ajustar los elementos para que se comunicaran entre ellos de manera correcta.

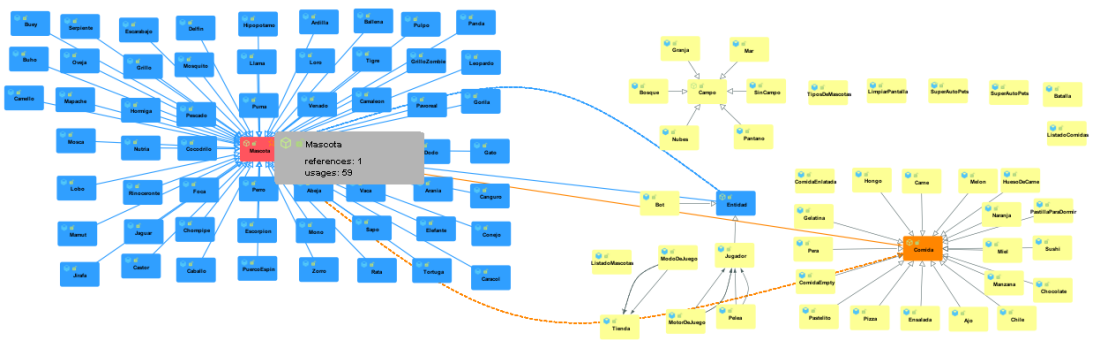
A continuación se muestran los diagramas donde se representan las relaciones entre las clases generados por IntelliJ:




Buey
Serpiente
Escarabajo
Delfin
Hipopotamo
Ardilla
Ballena
Pulpo
Panda
Buho
Oveja
Grillo
Mosquito
Llama
Loro
Tigre
GrilloZombie
Leopardo
Camello
Mapache
Hormiga
Pescado
Puma
Venado
Camaleon
Pavoreal
Gorila
Mosca
Nutria
Cocodrilo
Mascota
MascotaEmpty
Quetzal
Dodo
Gato
Lobo
Rinoceronte
Foca
Perro
Abeja
Vaca
Arania
Canguro
Mamut
Jaguar
Chompipe
Escorpion
Mono
Sapo
Elefante
Conejo
Jirafa
Castor
Caballo
PuercoEspin
Zorro
Rata
Tortuga
Caracol


Granja
Mar
Bosque
Campo
SinCampo
Nubes
Pantano


ListadoComidas
Batalla
Bot
Entidad
ListadoMascotas
Jugador
Jugador
references: 1
usages: 2
Tienda
ModoDeJuego
MotorDeJuego
Pelea


Mascota
references: 1
usages: 59

3.FUNCIONAMIENTO

El juego consiste en lo siguiente:

Existen dos entidades que serán los jugadores, cada uno con un equipo de animales (mascotas) que poseen características propias de cada tipo, con una cantidad de vida, de daño, nivel, habilidades propias y habilidades que pueden adquirir mediante powerups que cada jugador puede adquirir mediante una tienda.

Cada jugador puede tener un máximo de 5 mascotas en su equipo, y cada mascota puede portar una habilidad extra en un espacio denominado comida (siempre y cuando esta contenga un efecto, de lo contrario se le puede dar a la mascota comida cuantas veces se quiera).

La tienda funciona como menú entre partidas, donde podemos adquirir mascotas, vender mascotas al precio del nivel en el que se encuentren, adquirir comida, ordenar las mascotas como queramos y seleccionar un campo para que nuestras mascotas peleen.

Los campos otorgan ciertos beneficios a algunos tipos de mascotas pero también pueden perjudicar a otras, según sea el caso.

El sistema en el que se organiza el juego es sencillo, diferenciando las diferentes partes en las que se desarrolla el combate entre ambos equipos, siendo el siguiente la abstracción más general:

```
hacer{
        sumar +1 al contador de ronda
        calcular el Tier actual
        mostrar el menú intermedio (tienda)
        iniciar batalla
}mientras( jugador1 y jugador2 tengan vida > 0 )
```

Cuando uno de los jugadores (o ambos) se quede sin vida entonces se procede a calcular quién es el jugador que ha perdido y quién es el jugador que ha ganado.

El Tier es un concepto de desbloqueo, en el que a medida que avanzamos en las rondas del juego podremos desbloquear en la tienda nuevas mascotas y nuevas comidas para estas. El Tier depende directamente del contador de ronda, siendo de la siguiente manera:

- Tier 2 se desbloquea en la 2° ronda
- Tier 3 se desbloquea en la 4° ronda
- Tier 4 se desbloquea en la 6° ronda
- Tier 5 se desbloquea en la 8° ronda
- Tier 6 se desbloquea en la 10° ronda
- Tier 7 se desbloquea en la 12° ronda

Asimismo cada jugador iniciara la partida con 10 de vida, restándole vida por cada derrota que tenga en base a lo siguiente:

- Ronda 1, 2 y 3 el jugador recibe 1 de daño por derrota
- Ronda 4, 5 y 6 el jugador recibe 2 de daño por derrota
- Ronda 7+ el jugador recibe 3 de daño por derrota

La tienda también tendrá una evolución por rondas, de la siguiente manera:

- Ronda 1, 2, y 3 habrán únicamente 3 animales en tienda
- Ronda 4, 5 y 6 habrán únicamente 4 animales en tienda
- Ronda 7+ habrán 5 animales en tienda

Siempre habrán dos alimentos en la tienda generados de manera aleatoria.

Mecánica de la batalla:
La batalla entre los equipos de ambos jugadores se realiza de la siguiente manera:

- Muestra los equipos de ambos jugadores
- Se lleva a cabo las bonificaciones de los campos de juego
- Se activan todos los efectos al inicio de las mascotas (incluidos los efectos al inicio de la comida que lleven en inventario)
- Se desplazan ambos equipos buscando la posición 0 del arreglo
- Las mascotas en la posición 0 del array atacan y reciben daño. Al recibir daño existe una comprobación para verificar si hay algo que los proteja de ese daño o no.
- Si una mascota en la posición 0 de su equipo llega a 0 unidades de vida se procede a activar el efecto al morir. Este efecto al morir puede invocar a una nueva mascota en su lugar.
- Si en el paso anterior no se invocó una nueva mascota entonces se procede a eliminar esta mascota del equipo y vuelve a empezar hasta que no uno de los jugadores se quede sin mascotas vivas en el campo.

Las mascotas de cada jugador se almacenan en arrays del tipo Mascota. Cada jugador posee dos arrays de este tipo, uno siendo el equipo destructivo (el que pelea) y el otro el equipo de respaldo desde el cual se cargan las mascotas a pelear.
En caso de que algún efecto adquirido en la tienda sea temporal este se carga en el array destructivo (el que pelea), de esa manera me aseguro que no siga activo en las rondas siguientes. Para las copias de los arrays se utiliza el método .clone(), evitando de esta manera que ambos arrays apunten al mismo objeto y ambos se conviertan en destructivos.