

jfx-pokemon



Integrantes:

Camilo Campaz Jiménez - A00370258

Camilo González Velasco - A00370263

Daniel Esteban Jaraba Gaviria - A00368822

Johan Stiven Ricardo Sibaja - A0037069

DOCENTE: Juan Manuel Reyes

REQUERIMIENTOS FUNCIONALES	3
REQUERIMIENTOS NO FUNCIONALES	5
DIAGRAMA DE CLASES	6
DISEÑOS DE PRUEBAS UNITARIAS	7

REQUERIMIENTOS FUNCIONALES

El programa debe estar en la capacidad de:

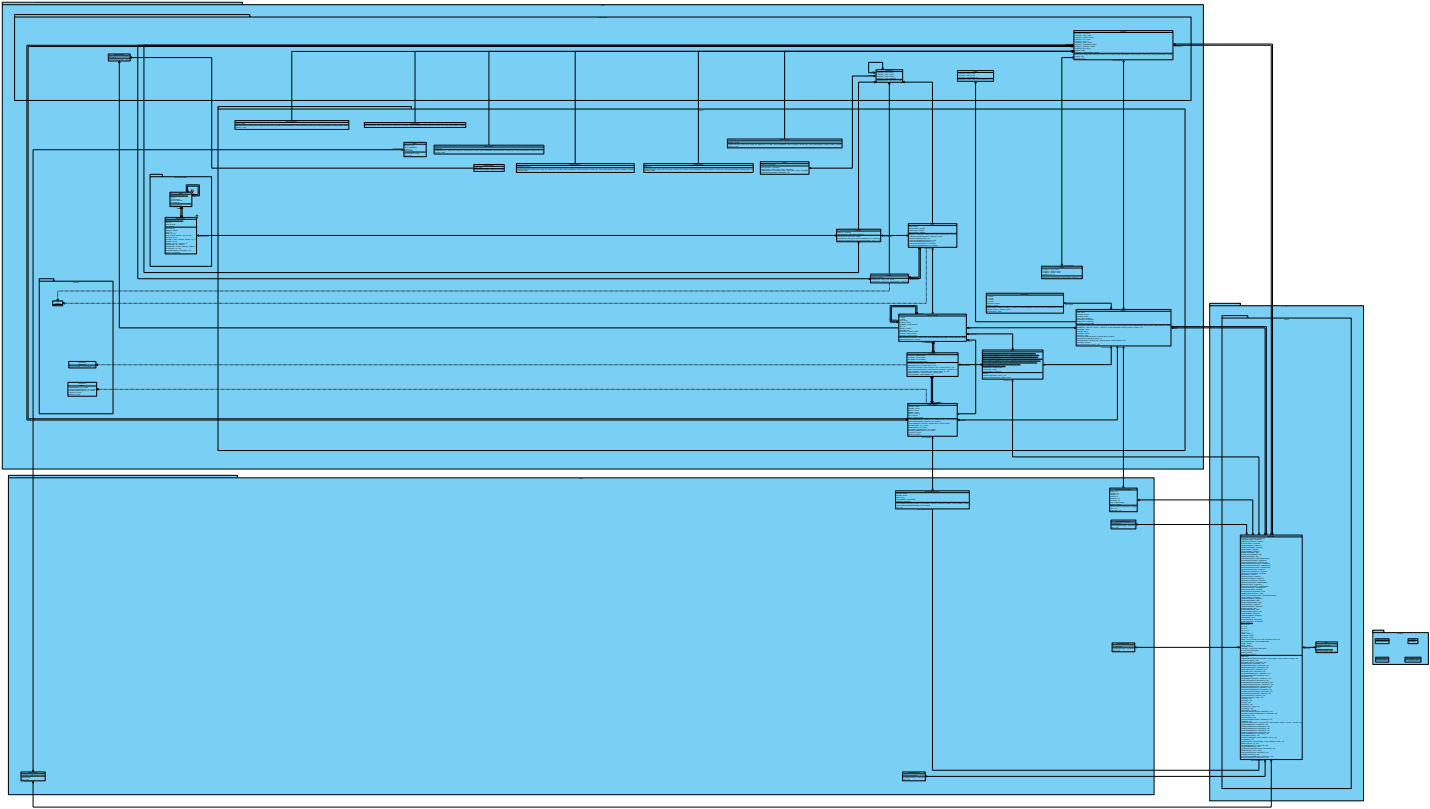
1. Comenzar una nueva aventura con un nuevo entrenador pokémon, con un nombre que será proporcionado por el jugador y un avatar elegido por el mismo, que puede ser masculino o femenino.
2. Elegir el pokémon inicial para el nuevo entrenador pokémon.
3. Cargar los datos de una partida anterior.
4. Acceder a la información de cualquier pokémon por medio de la pokédex. Tan solo es necesario ingresar el nombre del pokemon para visualizar los datos.
5. Guardar manualmente la información de la partida durante la aventura del jugador.
6. Cambiar el volumen de la música en el menú de configuración.
7. Listar en orden los pokemons que el entrenador tiene en sus pokeballs.
8. Cambiar de mapa mientras va avanzando en el mapa principal.
9. Enfrentar de manera automática los pokemons que participan en el torneo frente a el jugador actual.
10. Aumentar el nivel de los pokemons y el nivel de fuerza, energía y defensa.
11. Implementar una mecánica de pelea por turnos por cada enfrentamiento que tenga el entrenador.
12. Mostrar un pequeño tutorial del juego cada vez que se inicie una nueva aventura.
13. Guardar información de un pokémon en la pokédex.
14. Reproducir música de manera continua en cualquier pantalla hasta que el juego se cierre.
15. Capturar algunos pokémon y guardarlos en su colección de pokemons.
16. Reconocer los objetos en los mapas y que el jugador interactúe correctamente con ellos; es decir, si es objeto sólido (como casas , montañas , rocas, etc) el jugador debe chocar con él.

REQUERIMIENTOS NO FUNCIONALES

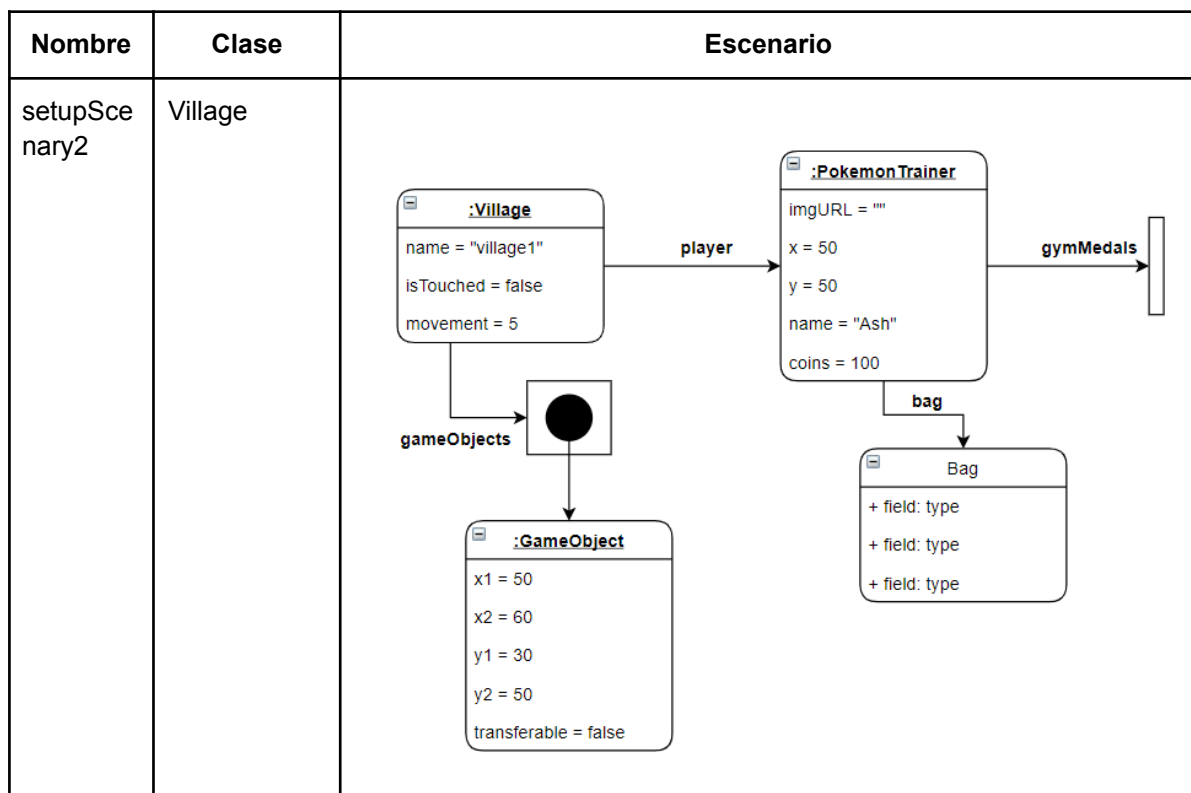
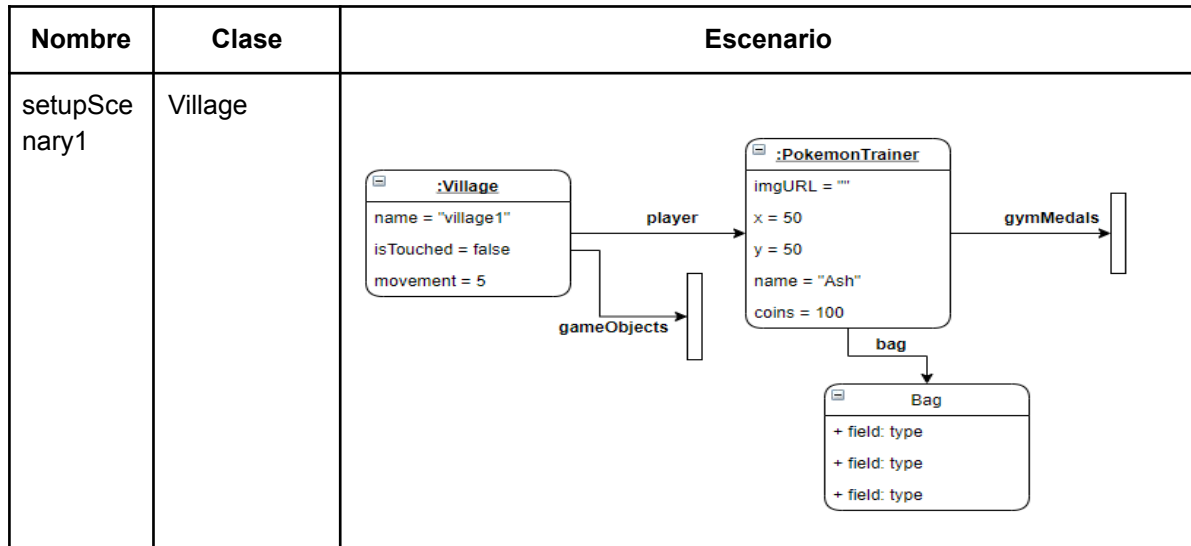
El programa debe cumplir con las siguientes condiciones:

1. Impedir que el usuario salga de los límites del mapa con el movimiento del jugador.
2. Generar un mínimo de 3 objetos de tipo Pokemon por cada mapa del videojuego, a excepción de zonas de intercambio y/o seguras.
3. Cambiar el nombre de cualquier Pokemon que el jugador haya atrapado.
4. Limitar la cantidad de Pokemons que se generan en un mismo mapa.
5. Visualizar NPC en las diferentes aldeas.

DIAGRAMA DE CLASES



DISEÑOS DE PRUEBAS UNITARIAS



Objetivo de la Prueba: Verificar el correcto funcionamiento del sistema de movimiento en dirección ascendente para el personaje modificando de manera correcta las coordenadas del mismo

Clase	Método	Escenario	Valores de Entrada	Resultado
Village	moveUp	setupScenary1	<i>ninguno</i>	se modifica el atributo del objeto PokemonTrainer que posee la clase, restando movement a a su atributo en Y

Objetivo de la Prueba: Verificar el correcto funcionamiento del sistema de movimiento en dirección descendente para el personaje modificando de manera correcta las coordenadas del mismo

Clase	Método	Escenario	Valores de Entrada	Resultado
Village	moveDown	setupScenary1	<i>ninguno</i>	se modifica el atributo del objeto PokemonTrainer que posee la clase, sumando movement a a su atributo en Y

Objetivo de la Prueba: Verificar el correcto funcionamiento del sistema de movimiento en dirección hacia la derecha para el personaje modificando de manera correcta las coordenadas del mismo

Clase	Método	Escenario	Valores de Entrada	Resultado
Village	moveRight	setupScenary1	<i>ninguno</i>	se modifica el atributo del objeto PokemonTrainer que posee la clase, sumando movement a a su atributo en X

Objetivo de la Prueba: Verificar el correcto funcionamiento del sistema de movimiento en dirección hacia la izquierda para el personaje modificando de manera correcta las coordenadas del mismo

Clase	Método	Escenario	Valores de Entrada	Resultado
Village	moveLeft	setupScenary1	<i>ninguno</i>	se modifica el atributo del objeto PokemonTrainer que posee la clase, restando movement a a su atributo en X

Objetivo de la Prueba: Verificar que el jugador no pueda atravesar diferentes objetos que hay por el mapa, por la misma razón no podrá salirse del mismo por zonas no permitidas

previamente,esto se consigue creando objetos los cuales ocupan un espacio en el espacio no permitiendo el paso

Clase	Método	Escenario	Valores de Entrada	Resultado
Game Object	isNear	setupScenary2	x = 50 y = 50	true El jugador no podrá continuar moviéndose en la dirección hacia la derecha ya que está "colisionando" con un objeto el cual no puede atravesar