

1. Iteración

a. Roles

- i. Juego<<Controller>>

b. Responsabilidades

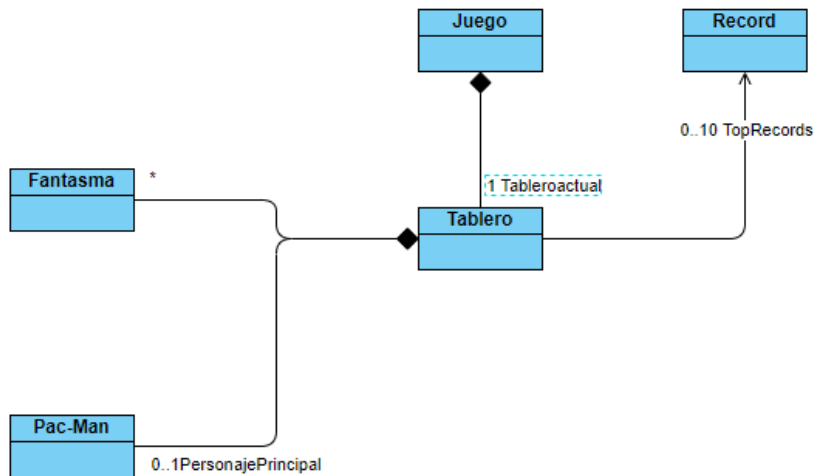
- i. Correr aplicación <<Juego>>
- ii.



2. Iteración

a. Roles:

- i. Pac-Man <<Information Holder>>
- ii. Juego <<Structurer>>
- iii. Fantasma <<Information Holder>>
- iv. Tablero <<Controller>>
- v. Record <<Information Holder>>



b. Responsabilidades:

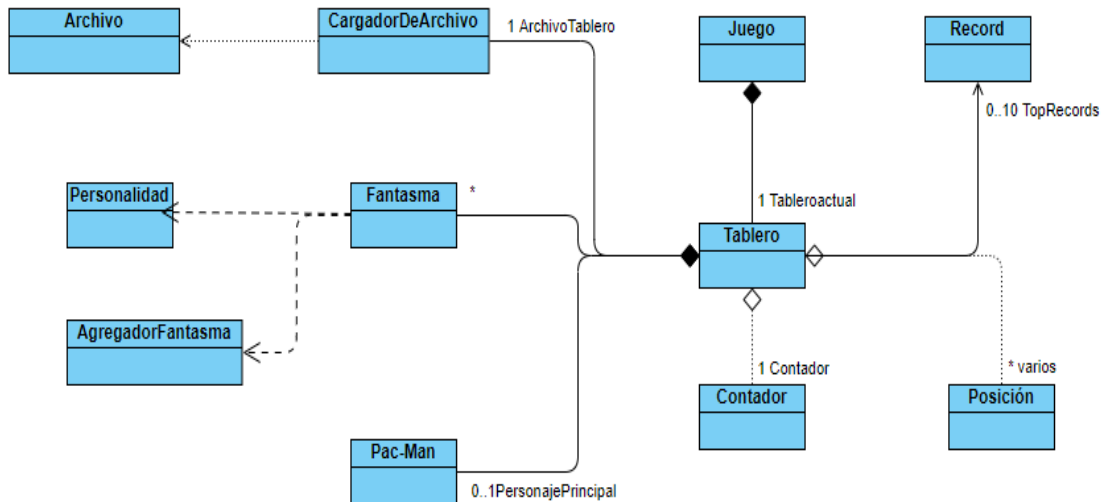
- i.

Responsabilidad	Rol
Iniciar nuevo juego	Juego
Poner Nuevo tablero	Tablero
Repetir Tablero	Tablero
Mantener juego	Juego
Verificar Record	Record
Recoger galletas	Pac-Man
Perseguir jugador	Fantasma

3. Iteración

a. Roles:

- i. Pac-Man <<Information Holder>>
- ii. Juego <<Structurer>>
- iii. Fantasma <<Information Holder>>
- iv. Tablero <<Controller>>
- v. Record <<Information Holder>>
- vi. CargadorArchivo<<Structurer>>
- vii. Archivo<<Information Holder>>
- viii. Contador<<Information Holder>>
- ix. Posición<<Information Holder>>
- x. Personalidad<<Information Holder>>
- xi. AgregadorDeFantasmas<<Strcuturer>>



b. Responsabilidades:

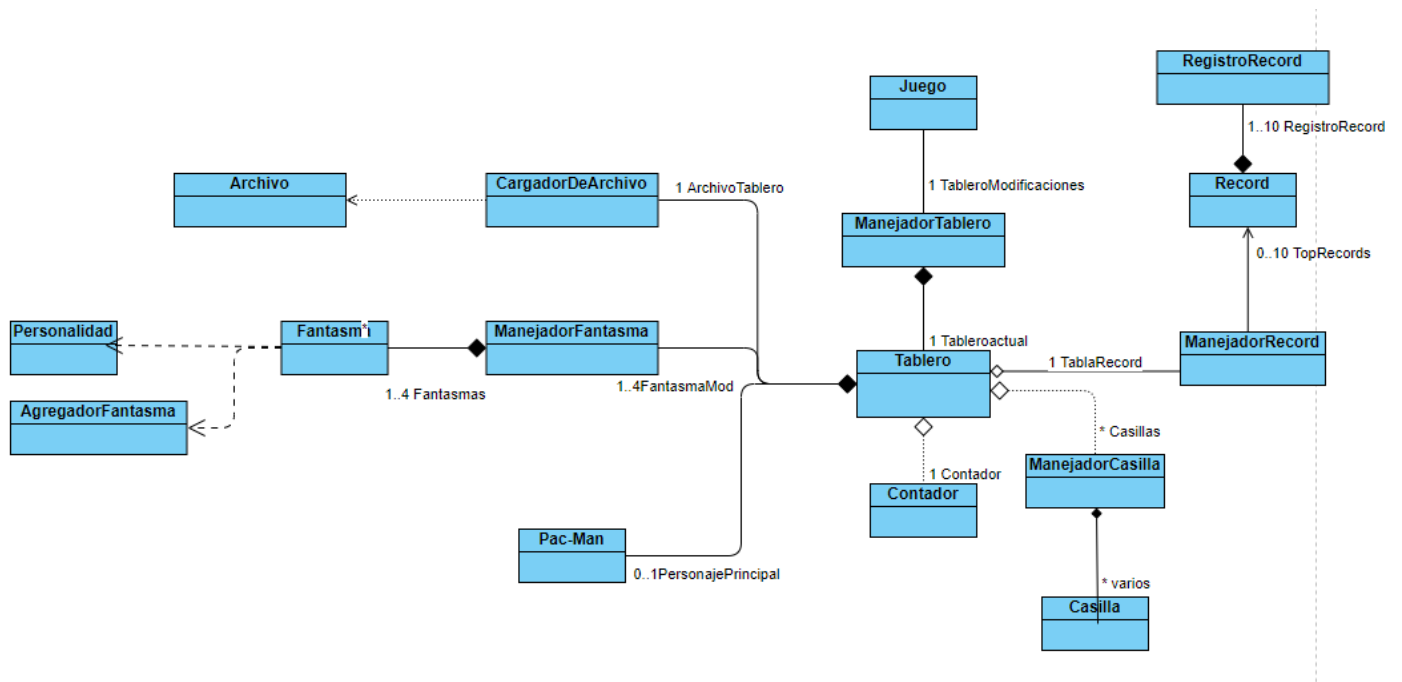
i.

Responsabilidad	Rol
Iniciar nuevo juego	Juego
Poner Nuevo tablero	Tablero
Repetir Tablero	Tablero
Mantener juego	Juego
Verificar Record	Record
Recoger galletas	Pac-Man
Perseguir jugador	Fantasma
Agregar Tablero	AgregadorArchivos
Agregar Fantasma	Fantasma
Llevar tiempo en partida	Contador
Dar poder	Posición
Cambiar Personalidad	Personalidad

4. Iteración

a. Roles:

- i. Pac-Man <<Information Holder>>
- ii. Juego <<Structurer>>
- iii. Fantasma <<Information Holder>>
- iv. Tablero <<Strcuturer>>
- v. Record <<Information Holder>>
- vi. CargadorArchivo<<Structurer>>
- vii. Archivo<<Information Holder>>
- viii. Contador<<Information Holder>>
- ix. Casilla<<Information Holder>>
- x. Personalidad<<Information Holder>>
- xi. AgregadorDeFantasmas<<Strcuturer>>
- xii. ManejadorRecord<<Strcuturer>>
- xiii. ManejadorTableros<<Strcuturer>>
- xiv. ManejadorCasilla<<Strcuturer>>
- xv. ManejadorFantasmas<<Strcuturer>>
- xvi. RegistroRecord<<Strcuturer>>



b. Responsabilidades:

i.

Responsabilidad	Rol
Iniciar nuevo juego	Juego
Poner Nuevo tablero	Tablero
Repetir Tablero	Tablero
Mantener juego	Juego
Verificar Record	Record
Recoger galletas	Pac-Man

Perseguir jugador	Fantasma
Agregar Tablero	AgregadorArchivos
Agregar Fantasma	Fantasma
Llevar tiempo en partida	Contador
Dar poder	Posición
Cambiar Personalidad	Personalidad
Registro Usuario	RegistroRecord
Acabar Juego	Juego
Actualizar Tablero	Tablero
Contar Galletas	PacMan
Repetir Tablero	Tablero
Regresar Casilla	Fantasma

5. Iteración

a. Roles

- i. Pac-Man <<Information Holder>>
- ii. Juego <<Structurer>>
- iii. Fantasma <<Information Holder>>
- iv. Tablero <<Strcuturer>>
- v. Record <<Information Holder>>
- vi. CargadorArchivo<<Structurer>>
- vii. Archivo<<Information Holder>>
- viii. Contador<<Information Holder>>
- ix. Casilla<<Information Holder>>
- x. Personalidad<<Information Holder>>
- xi. AgregadorDeFantasma<<Strcuturer>>
- xii. ManejadorRecord<<Strcuturer>>
- xiii. ManejadorTablero<<Strcuturer>>
- xiv. ManejadorCasilla<<Strcuturer>>
- xv. ManejadorFantasma<<Strcuturer>>
- xvi. ConfiguradorCasilla<<Strcuturer>>
- xvii. RegistroRecord<<Strcuturer>>
- xviii. Puntaje<<Information Holder>>
- xix. AlmacenPuntaje<<Structurer>>
- xx. ControladorJuego<<Controller>>
- xxi. MovimientoPacMan<<Structurer>>
- xxii. Vida<<Information Holder>>
- xxiii. ManejadorVida<<Structurer>>
- xxiv. Jugada<< Informaton Holder>>
- xxv. ManejadorJugada<<Structurer>>
- xxvi. AlmacenadorArchivo<<Informaton Holder>>
- xxvii. MovimientoFantasma<<Controlador>>

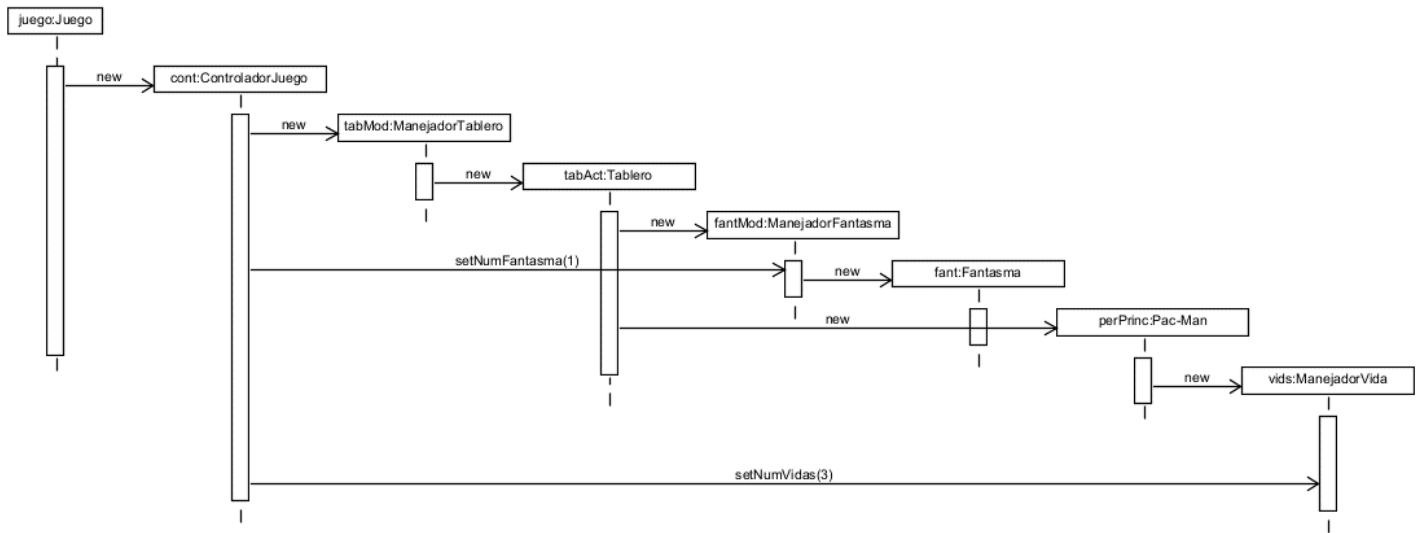
-----El diagrama final se encuentra en el repositorio con mejor calidad-----

Glosario:

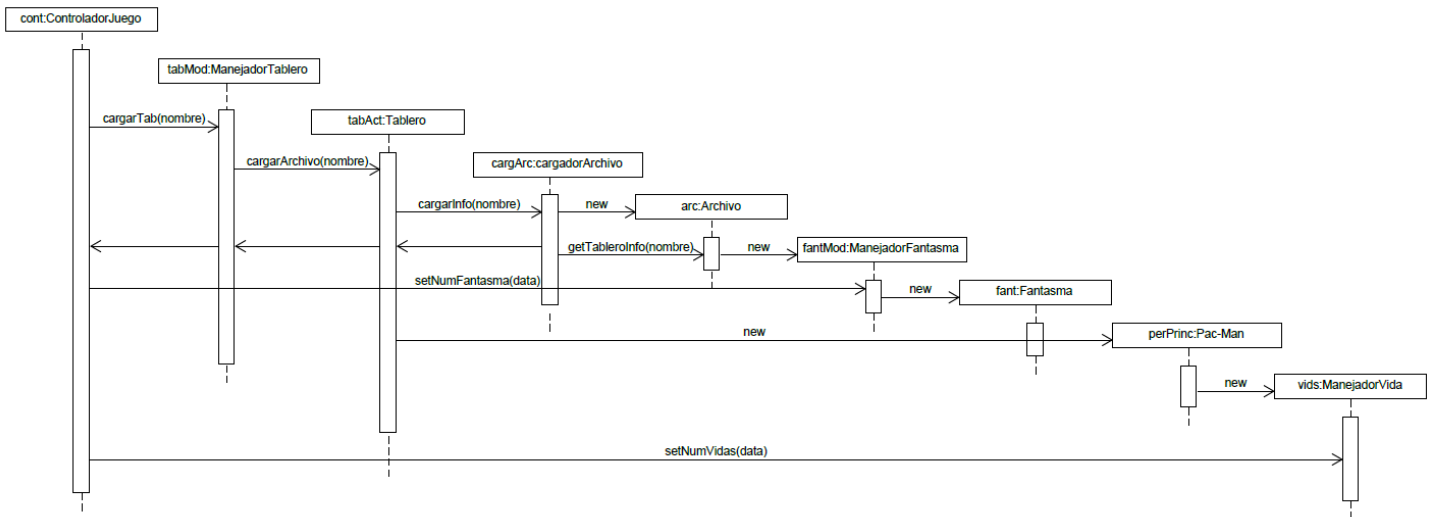
- i. Pac-Man: En este caso, Pac man es el jugador principal de nuestro juego. Es decir, el objeto que el usuario va a poder manipular.
- ii. Juego: En este caso, es el corazón de nuestro juego, es decir, es el componente en el cual recae casi todo el funcionamiento del juego.
- iii. Fantasma: Es el segundo objeto más importante del juego. Puesto que, es el que va a cazar a pac man. La lógica de este objeto es controlada por otra clase para su movimiento.
- iv. Tablero: Se puede considerar como lo que contiene todo sin ser la aplicación. Es decir, todas las casillas y objeto que hay en el juego están ubicados en esta parte.
- v. Record: Es el registro de puntuación de un usuario. Es la marca que se va a guardar para hacer el top de records. Tiene el nombre del usuario que realizo el record y su puntuación.
- vi. Cargador Archivo: Es la estructura encargada de manipular todo lo referente a cargar el archivo de puntuación y del tablero.
- vii. Archivo: Es el archivo de texto del cual se puede sacar la información de un tablero que el usuario quiera cargar.
- viii. Contador: Es un objeto que va a estar en la aplicación el cual le cuenta al usuario el tiempo que lleva en el juego.
- ix. Casilla: Es la estructura base del tablero, es decir, el tablero se forma de casillas. Cada una de estas casillas tiene una función. Por ejemplo, pueden ser o puntos o galletas.
- x. Personalidad: Es una clase que se agrega a fantasma y va a controlar lo que es el patrón con el que el fantasma persigue a pacman, dependiendo de su color.
- xi. AgregadorDeFantasma: Es una clase que se va a encargar de todo lo que conlleva agregar a un fantasma al tablero del usuario. Es decir, definir todos los puntos que lleva el fantasma.
- xii. ManejadorRecord: Es la clase de tipo estructura, que va a permitir armar la tabla de record en el juego. Para que de esta manera se pueda mostrar el top 10.
- xiii. ManejadorTableros: Es la clase que, como su nombre lo indica, va a manejar todo lo relacionado a los tableros: su carga, el tiempo y todo.
- xiv. ManejadorCasilla: Es la clase que se va a encargar de manejar todas las casillas. El total, de que tipo es cada una y todo eso para tener mejor organización en el mapa.
- xv. ManejadorFantasmas: Es la clase que va a manejar todos los fantasmas. Que tienen que hacer cuando desaparecen y ese tipo de cosas.
- xvi. RegistroRecord: Es la clase que se encarga de hacer el registro para cada record. Donde se registra el nombre y la puntuación.

c. Colaboraciones:

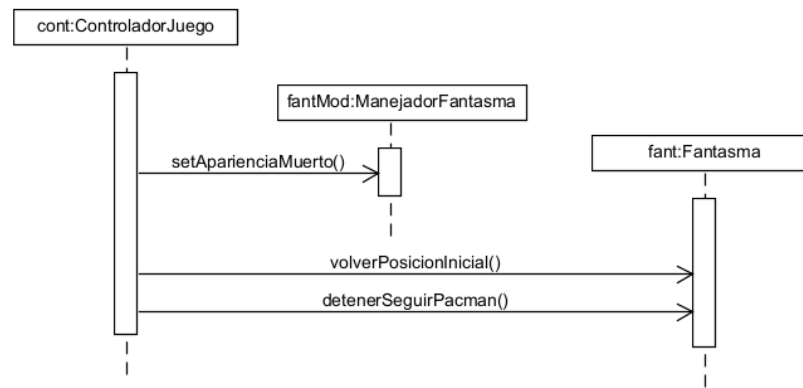
Caso 1: creación de una nueva partida.



Caso 2: cargar partida desde un archivo.

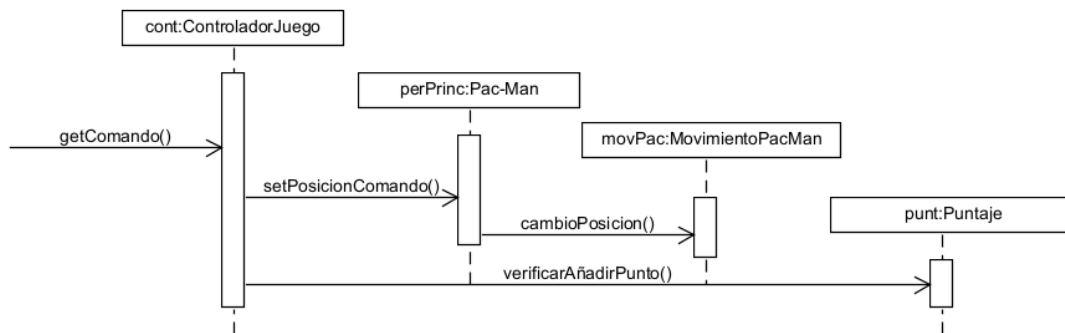


Caso 3: muerte de un fantasma durante partida.



Nota: la muerte del personaje principal presenta la misma lógica, bajo las clases relacionadas a Pac-Man.

Caso 4: movimiento de Pac-Man por el tablero conforme consume puntos.



4. Reflexiones

Teniendo en cuenta el diseño mencionado anteriormente pudimos identificar diversas ventajas y desventajas del mismo, principalmente una de las principales ventajas que encontramos fue la división de roles y del mismo mapa uml, al hacer esto ponemos en práctica el concepto de dividir y conquistar el cual básicamente permite tener todos los requerimientos definidos y que estos no tengan que depender entre ellos todo el tiempo para poder llevar a cabo alguna función como podría ser mirar el récord más alto. Una desventaja sobre el diseño es la dependencia tan alta del tablero para el funcionamiento de gran parte de las clases, a pesar de que sea claro que el tablero es muy importante para el programa este podría tener una menor relevancia en cuanto a ciertas clases por ejemplo con almacenamiento de algunos datos desde nuestra perspectiva no es necesario que tenga una relación directa con el tablero.

Un trade-off que nosotros llevamos a cabo fue compartir la función de movimiento para pac-man y los fantasmas, este elemento se decidió de esta forma debido a que son clases bastante similares ya que al final ambas dependen del tablero para existir y llevan a cabo acciones en el mismo. Para este caso se pudo haber cambiado y agregar en el diseño funciones distintas ya que al ser dos clases distintas sería lo mejor que no compartieran la función del movimiento, ya que uno de los requisitos más importantes es la misma movilidad del pacman y los fantasmas, pero al optar por la compartir funciones se ahorran recurso y posibles errores.

El diseño de este proyecto empezó por el uml desde el cual a partir del enunciado empezamos a crear, la primera iteración fue la más simple ya que este mapa solo se dividía en 5 partes, a partir de ese punto las siguientes 3 iteraciones fueron más complicadas ya que se tenía que tener en cuenta el anterior mapa para lograr ampliarla sin cometer algún fallo, aparte de esto al expandir los mapas uml se iban asignando los roles lo cual fue de gran ayuda ya que teniendo definidos los roles se volvía más claro en que podíamos estar fallando y en que estábamos bien. Ya cuando se finalizaron los uml y la definición de roles se hizo la creación de la secuencia la cual intentamos hacerla de la forma más congruente posible a lo que se hizo

en los pasos anteriores, también se intentó seguir una lógica eficaz y óptima debido a que las colaboraciones determinarían que partes se necesitan entre si.

Lo más complicado de diseñar del proyecto fueron las relaciones entre objetos esto se debe a que los objetos cumplen una gran cantidad de funciones por lo que para crear una relación entre ellos se tiene que tener en cuenta diversos factores, también es complicado definir la secuencia que va a seguir cada una de las acciones que el programa va a llevar a cabo ya que la idea es hacer el programa lo más eficiente posible por lo que si se define mal una secuencia se puede perder cierta eficiencia.