Tecnológico de Monterrey

PROGRAMACIÓN ORIENTADA A OBJETOS

SITUACIÓN PROBLEMA

IMPARTIDA POR:
Prof. Fabiola Uribe Plata

ELABORADO POR:
Maria Jose Coronado Ramón A00232245

Manuel Alejandro Preciado Morán A01639643

11 de Junio de 2021

SITUACIÓN PROBLEMA: JUEGO SERIO

HISTORIA DE JUEGO:

La historia de nuestro juego se lleva a cabo en la casa de un niño con amnesia. Este niño olvida dónde deja las cosas, es muy desordenado y tiene todo regado por la casa. Para poder salir de su casa a jugar debe de buscar todos los ingredientes para entregarlos a su mamá y pueda preparar una comida. Para ello, debe recorrer las cuatro habitaciones de su casa (Recepción, habitación, comedor, cocina) y buscar todos los ingredientes que necesita.

Básicamente el objetivo del juego es que recorra todas las habitaciones de la casa y se asegure de que tomó todo lo necesario para que su mamá le pueda hacer un sandwich y pueda salir de su casa a jugar.

Lista de personajes

- Personaje principal: niño
- Personaje aliado: mamá

Lista de ingredientes:

- Pan (recepción casa, dentro de bolsa de plástico)
- Queso (rata, busca a cambio una moneda de plata)
- Jamón (Cocina, microondas)
- Lechuga (Cocina, horno)
- Tomate (habitación, buro, cajón)

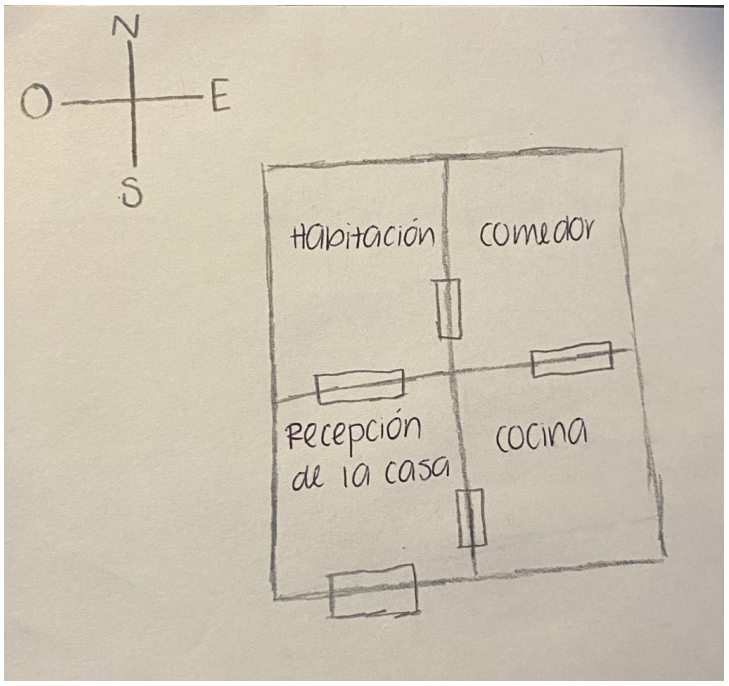
Otros objetos adicionales (no requeridos para ganar):

- Cochinito con monedas(habitación, repisas)

Lista de habitaciones:

- Recepción de la casa: perchero, sillones, tele.
- Habitación: repisa, cama, escritorio, buró (mesa de noche)
- Comedor: mesa, sillas, florero
- Cocina: refrigerador, estufa, horno, microondas

Mapa de habitaciones



IMÁGEN 1.0: Mapa de habitaciones del juego

Clases básicas para el juego:

Para la realización del juego se decidió crear primeramente una clase “Item” la cual consiste en la descripción de los objetos que se ven involucrados en el juego y que el personaje puede interactuar con ellos. De igual manera, se creó la clase “Room” en la que se describe el lugar donde se encuentra el personaje, las salidas (lugares a los que se puede dirigir) y los items de la habitación. También se creó la clase base “Character” de la que derivan dos subclases: **Niño**, **Mama**. La clase “Character” indica el nombre, la habitación donde se encuentra, el inventario y el número de ítems, y dichas características son las que heredan las subclases mencionadas anteriormente. Por otro lado se creó la clase “Comando” de la que derivan cuatro subclases: **Ayuda, Dar, Desplaza y Toma**. La clase “Comando” indica un comando como lo dice su nombre y una segunda palabra que sigue de este, como puede ser norte, sur, los ingredientes extraviados, etc., y esto va a depender del comando que se utilice. Las cuatro subclases heredan estas características. También está la clase “ListaPalabras” que guarda todas las listas de las palabras y sus comandos relacionados. También hay una clase que es muy importante para el juego que es la clase “Parser” que básicamente funciona como un traductor, es decir, toma lo que escribo, lo procesa y lo ejecuta. Finalmente está la clase “Game” que contiene todos los objetos para que el juego funcione.

Se elaboró el diagrama UML de las clases para que fuera un poco más entendible lo que se quiere realizar y para hacer más fácil la implementación de las clases en C + +.

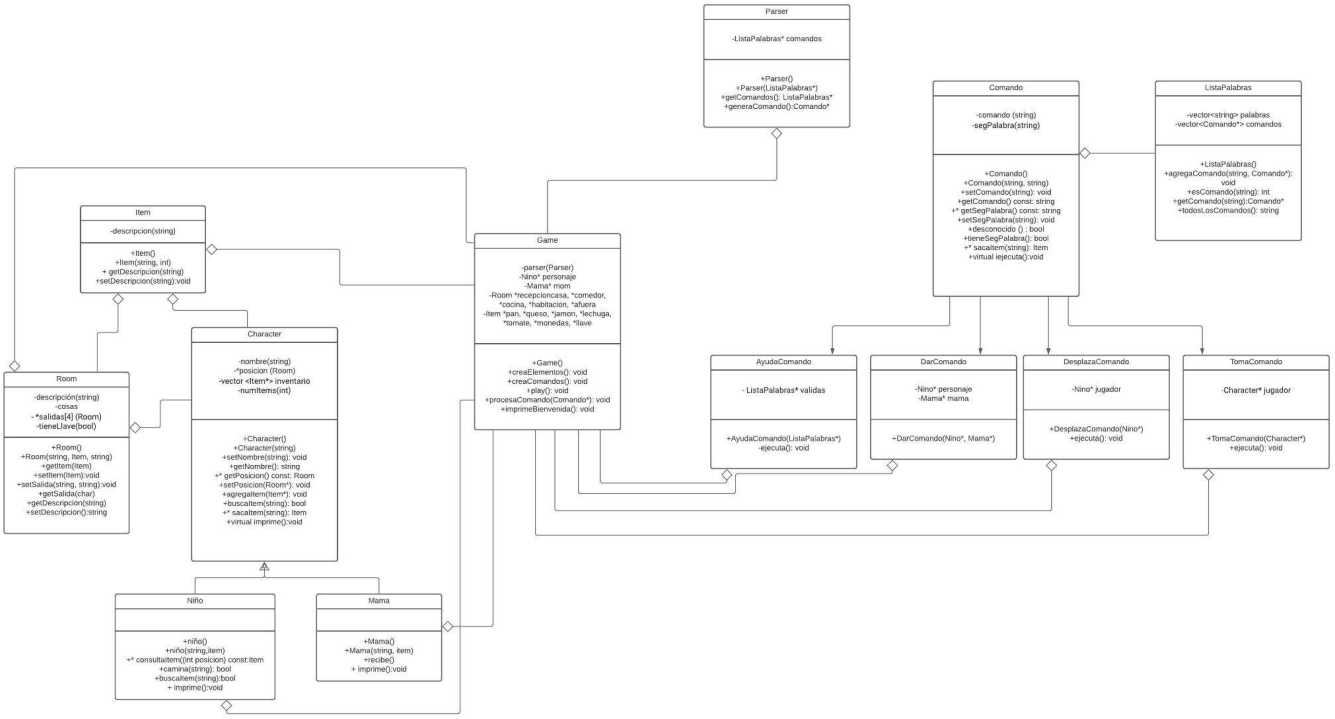


IMAGEN 2.0: Diagrama UML

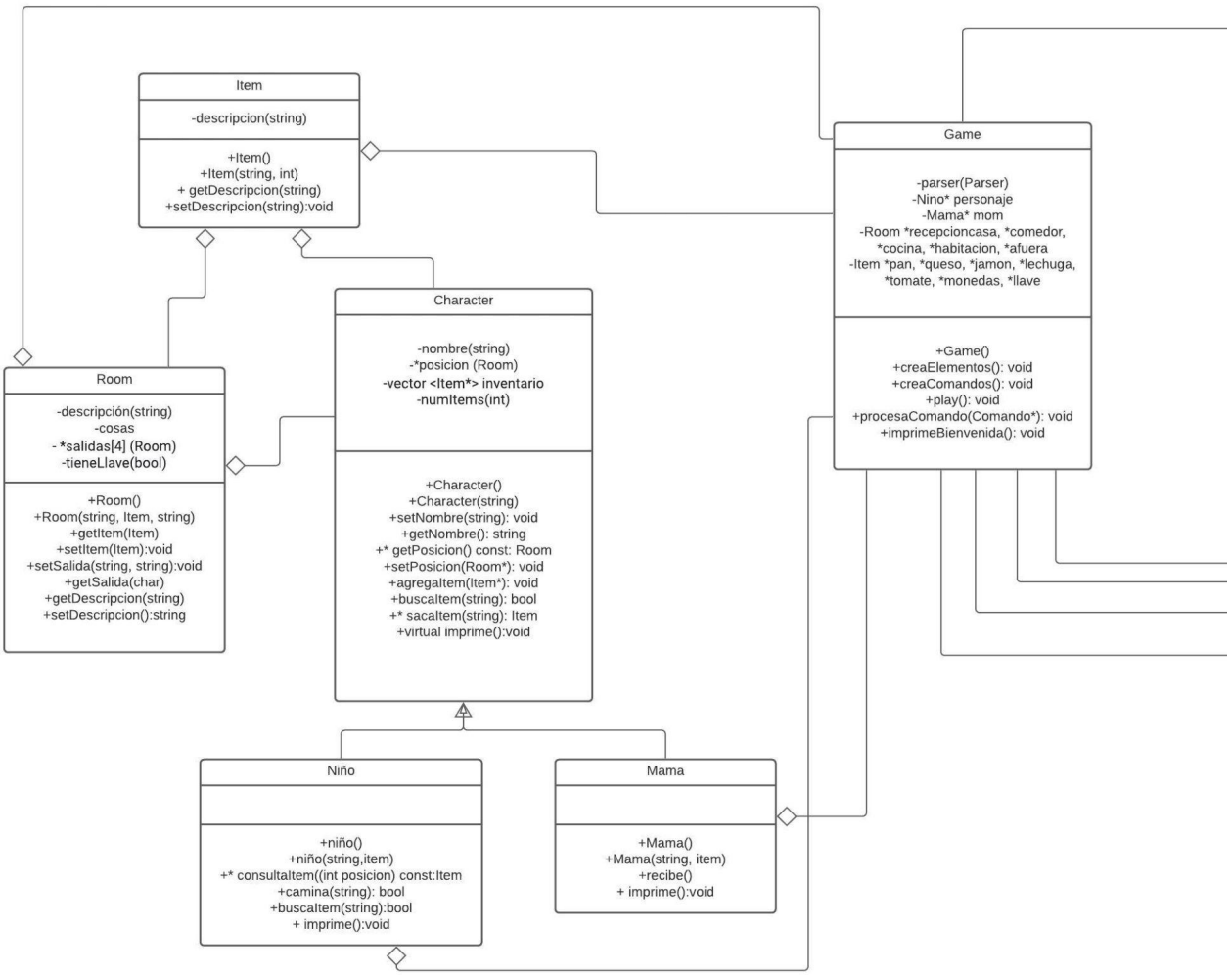


IMAGEN : Diagrama UML (Primera mitad)
2.1

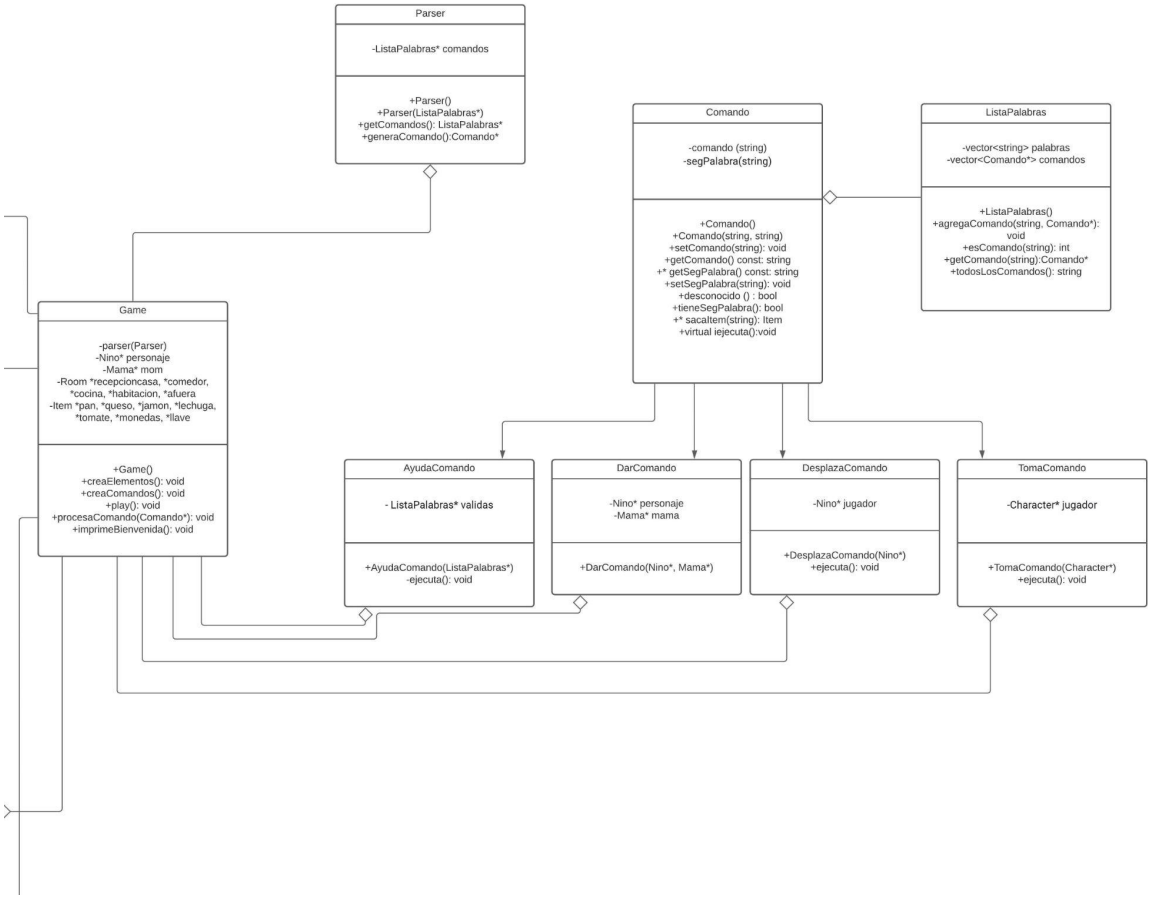


IMAGEN 2.2: Diagrama UML (Segunda mitad)

LINK DE DIAGRAMA UML:
https://lucid.app/lucidchart/0bf0c195-fb93-45da-b305-614a6477927c/edit?shared=true&page=0_0#

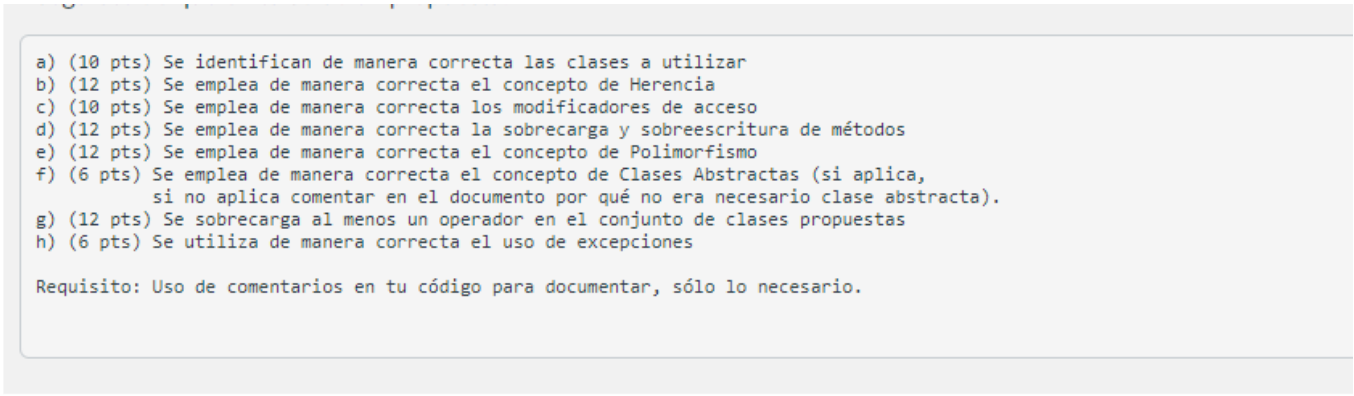
Posteriormente se pasó a la implementación de las clases

Como se mencionó anteriormente, la herencia se verá reflejada en la clase “Character”, de la cual se derivan dos subclases: **Niño** y **Mama,** y también en la clase “Comando” de la que se derivan cuatro subclases: **Ayuda, Dar, Desplaza y Toma**. De igual manera, el polimorfismo estará vinculado con la clase padre “Character” y “Comando”, específicamente en el método “imprime()” de la clase Character y en el método “ejecuta()” de la clase comando que lógicamente se encontrará en sus respectivas subclases.
La razón por la que escogimos que la herencia se diera entre la clase “Character” y las clases “Niño y mamá” es debido a que el niño y la mamá tienen ciertas características en común, y consideramos que sería más fácil hacer una clase base que incluyera dichas características para que simplemente no haya código repetido. Lo mismo ocurre con la clase

“Comando”, se decidió que hubiera una relación de herencia con Ayuda, Dar, Desplaza y Toma porque en la clase Padre ubicamos los métodos y atributos que tienen en común estas cuatro clases.
De igual manera, en la parte del polimorfismo en la clase Character se decidió aplicarlo en el método “imprime()” ya que tanto el niño como la mamá lo necesitan para mostrar sus datos, pero muestran cosas distintas, y en la clase Comando se decidió aplicarlo en el método “Ejecuta()” ya que es algo que necesita desplegarse en las cuatro clases pero de diferente forma.

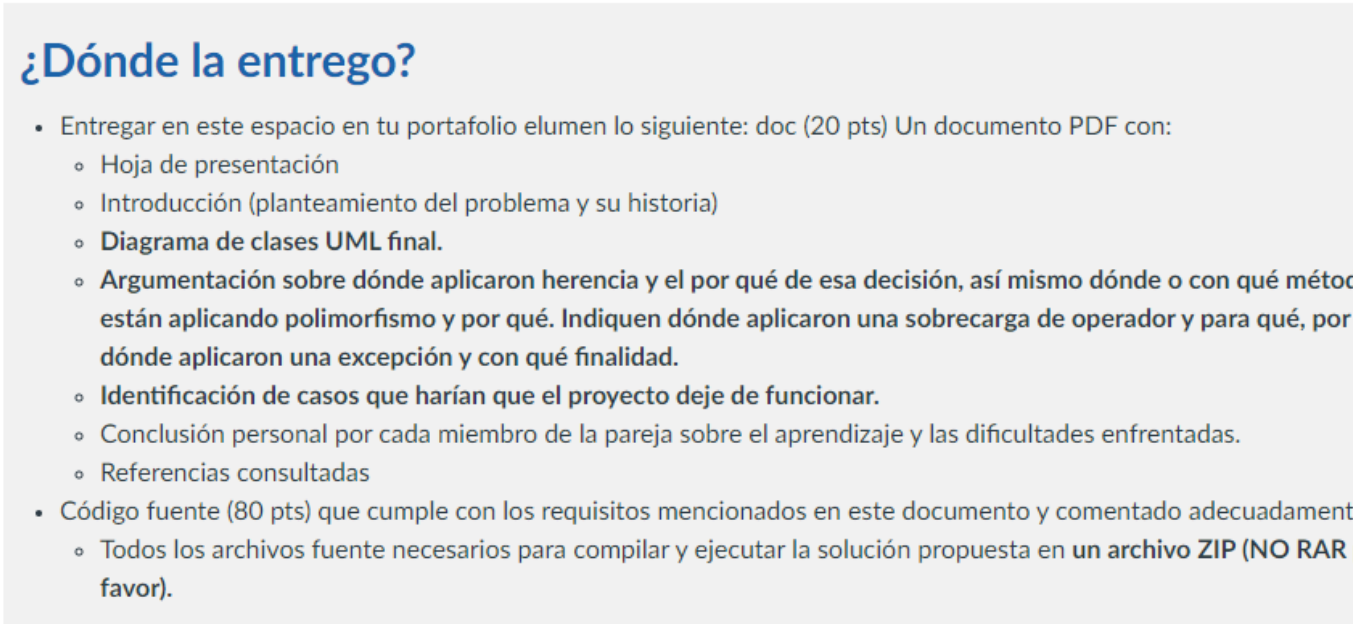
Para el caso de manejo de excepciones se utilizará cuando se te pida introducir un comando. Si el usuario se equivoca al momento de escribirlo, se le dará otra oportunidad para hacerlo en vez de que el juego termine ahí. Consideramos que ahí podría estar bien porque puede ocurrir que el usuario se llegue a equivocar hasta en una letra y usando una excepción lo podríamos ayudar en caso de que si falle.

Por último, utilizamos sobrecarga de operadores en la clase “Character” en los operadores de flujo de salida, con el objetivo de poder imprimir el nombre de la persona y los ingredientes que tiene consigo.



```
a) (10 pts) Se identifican de manera correcta las clases a utilizar
b) (12 pts) Se emplea de manera correcta el concepto de Herencia
c) (10 pts) Se emplea de manera correcta los modificadores de acceso
d) (12 pts) Se emplea de manera correcta la sobrecarga y sobreescritura de métodos
e) (12 pts) Se emplea de manera correcta el concepto de Polimorfismo
f) (6 pts) Se emplea de manera correcta el concepto de Clases Abstractas (si aplica,
          si no aplica comentar en el documento por qué no era necesario clase abstracta).
g) (12 pts) Se sobrecarga al menos un operador en el conjunto de clases propuestas
h) (6 pts) Se utiliza de manera correcta el uso de excepciones

Requisito: Uso de comentarios en tu código para documentar, sólo lo necesario.
```



¿Dónde la entrego?

- Entregar en este espacio en tu portafolio elumen lo siguiente: doc (20 pts) Un documento PDF con:
  - Hoja de presentación
  - Introducción (planteamiento del problema y su historia)
  - **Diagrama de clases UML final.**
  - **Argumentación sobre dónde aplicaron herencia y el por qué de esa decisión, así mismo dónde o con qué método están aplicando polimorfismo y por qué. Indiquen dónde aplicaron una sobrecarga de operador y para qué, por dónde aplicaron una excepción y con qué finalidad.**
  - **Identificación de casos que harían que el proyecto deje de funcionar.**
  - Conclusión personal por cada miembro de la pareja sobre el aprendizaje y las dificultades enfrentadas.
  - Referencias consultadas
- Código fuente (80 pts) que cumple con los requisitos mencionados en este documento y comentado adecuadament
  - Todos los archivos fuente necesarios para compilar y ejecutar la solución propuesta en **un archivo ZIP (NO RAR favor).**

Casos que harían que el proyecto deje de funcionar:

Un caso en el que el programa truene y deje de funcionar sería cuando estás en la cocina con la mamá e intentas dar un ingrediente que no tienes en el inventario ya que no tienes nada que entregar y por eso se corta. En unos casos en donde el primer ingreso incorrecto te da una segunda oportunidad y ya si fallas la segunda chanza, se corta.

Conclusiones individuales:

MARIA JOSÉ

Tras haber realizado este proyecto puedo decir que reforcé los conocimientos adquiridos en clase, e incluso lo que no aprendí bien (manejo de excepciones). En mi caso, al principio estaba nerviosa y tenía algo de miedo porque nunca había hecho algo así, pero a su vez me gustó

la idea y me retó a esforzarme un poquito más para poder sacarlo adelante y entregar un buen trabajo junto a mi compañero.
Si se nos presentaron dificultades obviamente, como el manejo de las excepciones (fue más dificultad porque no tenía un dominio del tema y no sabía cómo implementarlo), también se presentó otra dificultad a la hora de que la mamá debía de entregar la llave al niño, no podíamos hacer eso y fue un poco frustrante. Por último, otro ejemplo de una de nuestras dificultades fue todos los errores que nos salían las primeras veces que lo corríamos porque muchos no sabíamos de dónde venían hasta que leíamos bien el código.
En general le doy un 7-8 de nivel de dificultad, no fue algo facilísimo pero tampoco algo imposible y estoy contenta con lo que se logró.
Por último me gustaría mencionar que me gustó haber participado en el proyecto, y aunque no soy la mejor programadora ni es mi área se me hizo un proyecto padre.

MANUEL
Considero que este proyecto nos permitió poder aplicar todo el conocimiento adquirido durante las pasadas 5 semanas de una manera completamente interesante. Fue una experiencia divertida por un lado, ya que se nos permitió crear toda una historia acerca de un niño desordenado y su mamá la cual le permite salir a jugar siempre y cuando junte todos los ingredientes para que el niño coma antes de jugar. De hecho, el buscar ingredientes dentro de la casa fue inspirado por el programa de televisión nacional “Master Chef”. También, el ir acomodando como iban a pasar las cosas dentro del juego y acomodar los ingredientes dentro de la casa fue algo que se disfruto.

Por otro lado, lo desafiante del proyecto estuvo cuando debíamos aplicar el manejo de excepciones, ya que este fue el último visto durante clase, sin embargo, no hubieron actividades extras como para terminar de comprenderlo e identificar todas nuestras dudas o dificultades con el tema.

En conclusión, este proyecto fue divertido, interesante y nos permitió demostrar el conocimiento adquirido durante las pasadas semanas. También las últimas semanas me permitieron mejorar mis habilidades de programación con el lenguaje de C + +, y también entender mejor toda esta parte de la programación orientada a objetos.

Referencias

Observatorio de Innovación educativa, ITESM (2018) ¿Qué son los serious games? Consultado en: https://observatorio.tec.mx/edu-news/que-son-los-serious-games

Gamelearn (2017) Todo lo que necesitas saber sobre los serious games y el game-based learning, explicado con ejemplos. Consultado en: https://www.game-learn.com/lo-que-necesitas-saber-serious-games-game-based-learning-ejemplos/

Educación 3.0 (2018) Serious games: los videojuegos también educan. Consultado en: https://www.educaciontrespuntocero.com/noticias/serious-games-frenar-acoso-escolar-fomentar-la -creatividad/