Generalitat de Catalunya
Departament d'Ensenyament
INS Provençana

FINAL PROJECT G1

GRAPHIC ADVENTURE

Final project: A Quarantine Adventure

Institut Provençana Course 2019-2020

Development of multiplatform applications

Students:

María Chacón Alcaide

Alexandra Martínez Pérez

INDEX

1.PROJECT MODULE SHEET (IT DEPARTMENT)

CICLE I GROUP-CLASS

DAM2

TITLE

A Quarantine Adventure

GROUP'S STUDENTS

Alexandra Martínez
María Chacón

DESCRIPTION

The game is about a person who, during the confinement by the Covid-19, tries to skip it and get to see his partner who lives 10km away (the couple and the km can vary, it can be more or less depending on the difficulty, and the couple can be parents or they can be grandparents, whatever). The mission is to first reach the goal, and secondly, gain more points of happiness.

The adventure begins with our protagonist, at home, thinking about that loved one. The happiness score starts at 0, he is very sad and bored, so in an attack of irresponsibility, he decides to put his feet on the street and move towards his destination.

During the map, there are several areas to click on (different streets and squares that stand between the protagonist and the goal). As we go along we can click on these areas.
Each zone leads to an event (random but corresponding to that zone). The user must choose one of the answers provided. Depending on the answer (and mostly on the luck) the protagonist: 1. will be stopped (the game is over), 2. will suffer a penalty in happiness, or 3. an increase in happiness and finally will advance.

If the protagonist is caught by the police in the street, without an excuse or justification, he loses the game and the points counter will be set to 0, also losing any rewards he may have won during the course of this game.
If the protagonist is overcoming the events in a somewhat precarious way, he will be able to reach the end but with a low score.

If, on the other hand, our protagonist is doing great, he will arrive with a high score and set a new record.

There are items and inventory, and a shop to buy them.

REQUIRED MATERIALS AND TECHNICAL SPECIFICATIONS OF THE PROJECT

GlassFish Server or Tomcat (?)
MySQL
Visual Studio 2019 Community
C# Windows .Net Framework

2.FUNCTIONAL REQUERIMENTS WITH THEIR PRIORITY

RF1. Main Window

RF1.1 Register
The user can register and create an account with which he can log in
High

RF1.2 Validate (login)
The user will be able to login with his account previously made
High

RF1.3 Cancellation
The user can unsubscribe at any time, taking into account that he will lose his records.
Medium

RF1.4 Request to modify user data
The user can make a request to change data (password...)
Medium

RF1.5 Configuration
The player will be able to configure options such as sound volume, music...
Medium

RF2- Game start window

RF2.1-Start New Game
The player will be able to start a new game.
High

RF2.1.1 Character selection
The player can select the character he wants to play.
High.

RF2.2 Load Game
The player may resume a saved game.
Medium

RF2.2.1 Delete Game
The player may delete a saved game.
Medium

RF2.3 Record máximo
Show maximum record for this player.
Low

RF2.4 Ranking
The player will be able to consult which position in the ranking he is in.
Low

RF2.5 Achievements
The player will be able to consult his achievements and non achievements.
Low

RF2.6 Configuration
The player will be able to configure options such as sound volume, music...
Medium

RF3-In-game features (character)

RF3.1 Show Game Window
The player will be able to access the shop only at the beginning of the game, and spend their money (collected from other games) on usable items. (optional)
High

RF3.2 Shop (buy)
The player will be able to access the shop only at the beginning of the game, and spend their money (collected from other games) on usable items. (optional)
Low

RF3.3 Movement
The player can click on the area he wants to advance to on the map (within range).
High

RF3.4 Inventory
We will have a drop-down menu to view our inventory, and the player will be able to check their inventory and/or use items.
Medium

RF3.5 Event
The system displays a random event according to the area you are in on the map. It will consist of the statement of a situation and possible answers (actions).
The player will be able to choose an answer.
Depending on what he has chosen, there will also be a consequence before moving on to the next point on the map (positive, negative or neutral).
Among these events there can be things like someone coughing at you, , dodging a patrol...
Game over situations could be that for example you are stopped by the police and you are not allowed to leave (a dog, ...). You would get a fine that would make you lose all the happiness of the game. (game over)

High

RF3.6 Arrival at destination
If the score obtained is higher than the one reflected in your profile, your maximum score will be updated.
Depending on the happiness obtained, the player will have a reward in coins that can be spent in the shop at the beginning of the next game (the first game the user starts will have no coins).
High

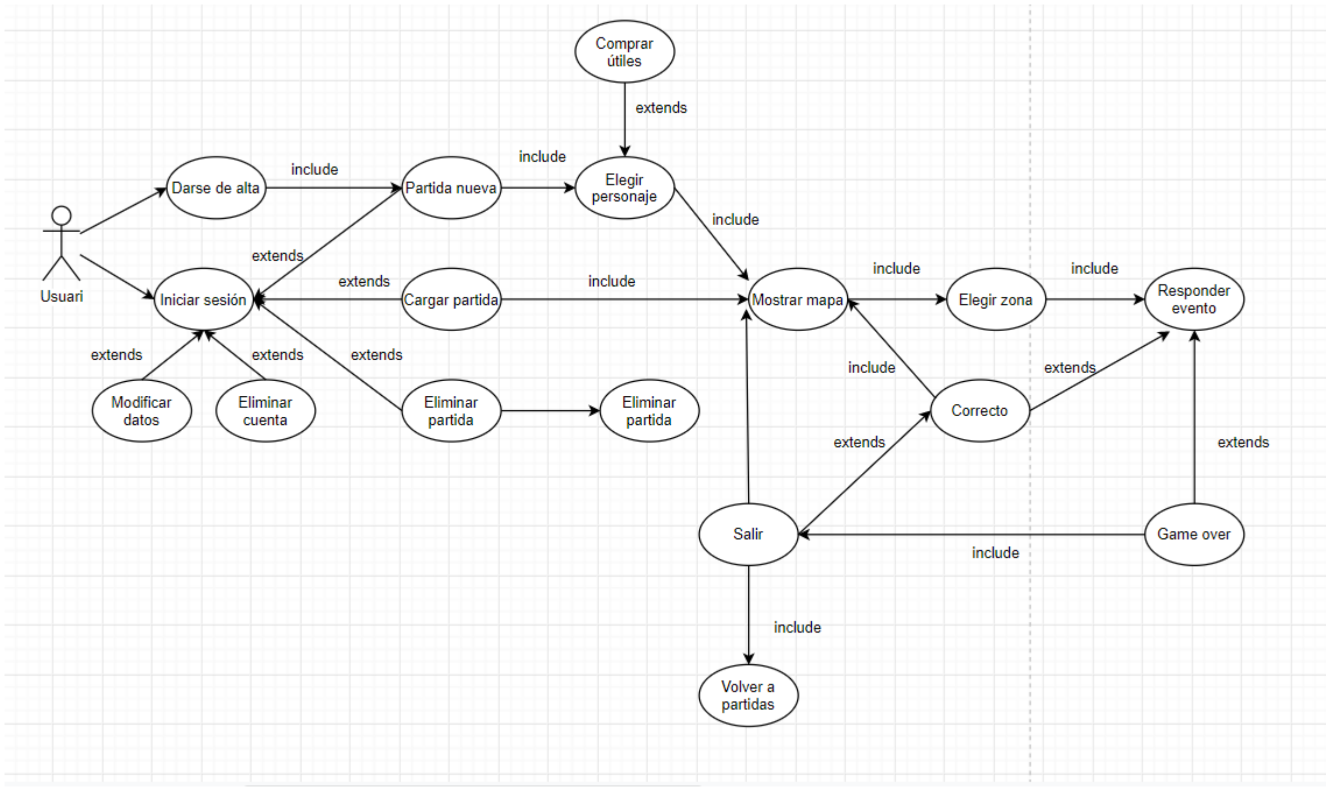
RF3.7 Exit game
The player can leave the game at any time by saving the changes automatically (it would load the map or event that is not completed).
High

RF4. Installation and configuration app

RF4.1 Installation
The user will be able to install the game.
Medium

RF4.2 Uninstall
The user can uninstall the game at any time.
Low

3.USE CASES



3.1.Explanation Cases d'Us (high priority)

Nombre:	**CDU1.1 -Sign up**
Description: The user will be registered and will have his account with his data.	
Actors: User.	
Preconditions: The user selects "Register" in the main window.	
Post-conditions: A message will be sent to the user and the user will be registered.	
Normal flow: 1.- The user is shown a form to fill in data. 2.- The user completes the form with aliases, password, and selects send. 3- The user will be saved in the DB. 4.- The user receives a message as he has been correctly registered.	
Alternative flow: 2.A1.- The alias already exists in the DB, the user is informed to put a different one. A2.- The alias or password is empty. The user can exit the application.	

Name:	CDU1.2-Login
Description: The user will insert his alias and password to log in.	
Actors: User.	
Preconditions: The user must be registered.	
Post-conditions: The user can display the home screen (RF2).	
Normal flow: 1- The user will insert his/her credentials and press the Validate button. 2- The following window will be displayed	
Alternative flow: A- The credentials are incorrect, the user is informed. 1.B- Nickname or password is empty. 1.C. Password or username lenght are not valids.	

Name:	CDU2.1-Start New Game
Description: Starts a new game	
Actors: User	
Preconditions: The user must be logged in.	
Post-conditions: The user will go to character selection.	
Normal flow: 1- The user selects Start new game. 2- A new game is created.	
Alternative flow: 1.A.- The user decides to exit, returns to the main screen.	

Name:	CDU2.1.1 Choose Character
Description: The player will choose the character he wants to play.	
Actors: User	
Preconditions: Having Selected Start New Game	
Post-conditions: The user(player) will be able to access the shop and/or access the map.	
Normal flow: 1- The user must choose one of the characters	
Alternative flow: 1.A.- The user decides to exit, returns to the main screen.	

Name:	CDU3.1 Show Map
Description: The map will be generated with random zones.	
Actors: User.	
Preconditions: The user has selected a character.	
Post-conditions: The user will be able to move his character.	
Normal flow: 1.The system will generate the map of the game, with its zones. 2.The game will start and the player will be able to see the screen with all its elements.	
Alternative flow: 1.A.- The user decides to exit, returns to the main screen.	

Name:	CDU3.3 Movement
Description: The user will choose which area to move to.	
Actors: **U**ser.	
Preconditions: Map loaded.	
Post-conditions: A zone event will be shown.	
Normal flow: 1.User selects which zone he wants to move to. 2. The character moves to that zone.	
Alternative flow: 1.A.- The user decides to exit, returns to the main screen (the game is saved).	

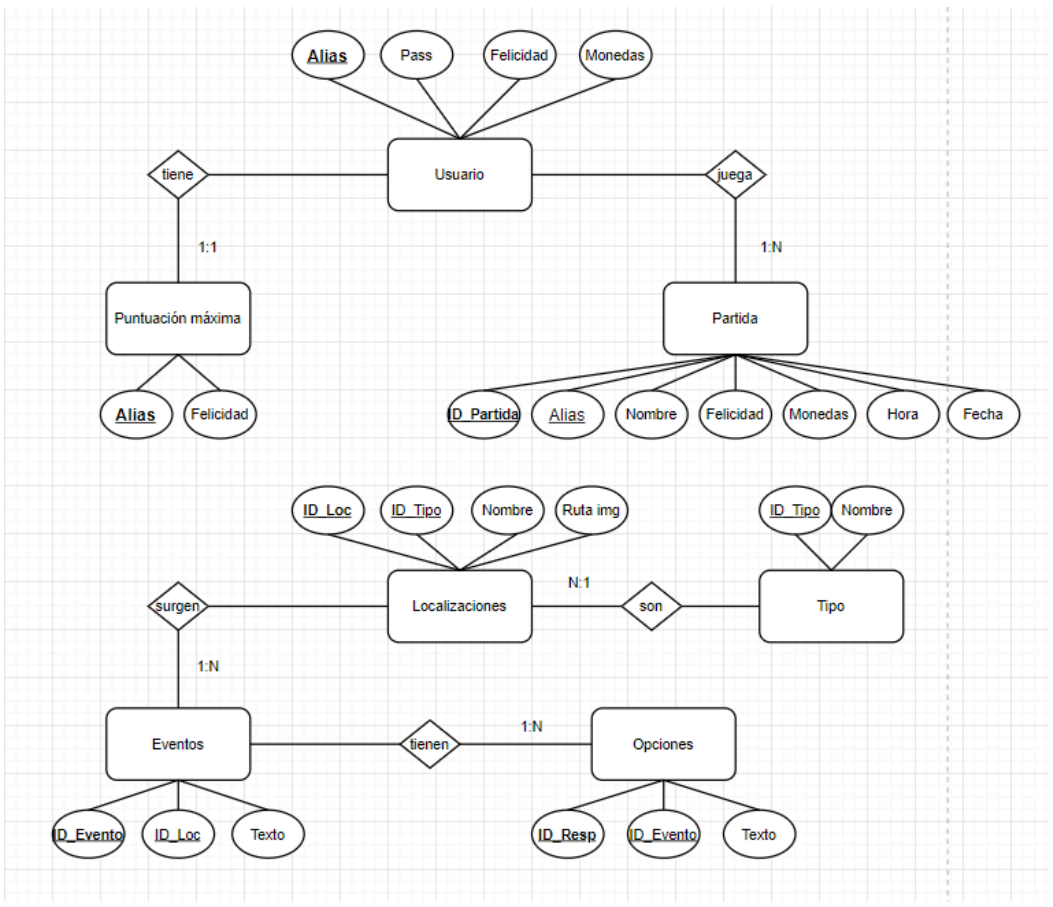
Name:	CDU3.5 Event
Description: The system will launch a random event to the user, depending on the localization.	
Actors: User.	
Preconditions: The character has moved on.	
Post-conditions: The map is displayed again.	
Normal flow: 1.The system launches an event and its possible answer options. 2. The user chooses one of those answer options. 3. The system launches a consequence. 4. The character suffers the negative or positive consequence (it is resolved). 5. The map is displayed again.	
Alternative flow: 4.A1.The character has a consequence that leads him to Game Over (the system notifies him). A2.The user returns to the main screen. A.- The user decides to exit, and returns to the main screen (the game is saved).	

Name:	CDU3.6.End of the game
Description: The character reaches the goal, wins the game and wins happiness (and money).	
Actors: User.	
Preconditions: The user has passed all the zones.	
Post-conditions: The record will be kept in case it is surpassed (or there is no previous record).	
Normal flow: 1. The character reaches the end zone and the final event is found. 2.Win the right way and win the game. 3.Happiness is achieved (final score).	
Alternative flow: 2.A. Follow the wrong path and lose the game.	

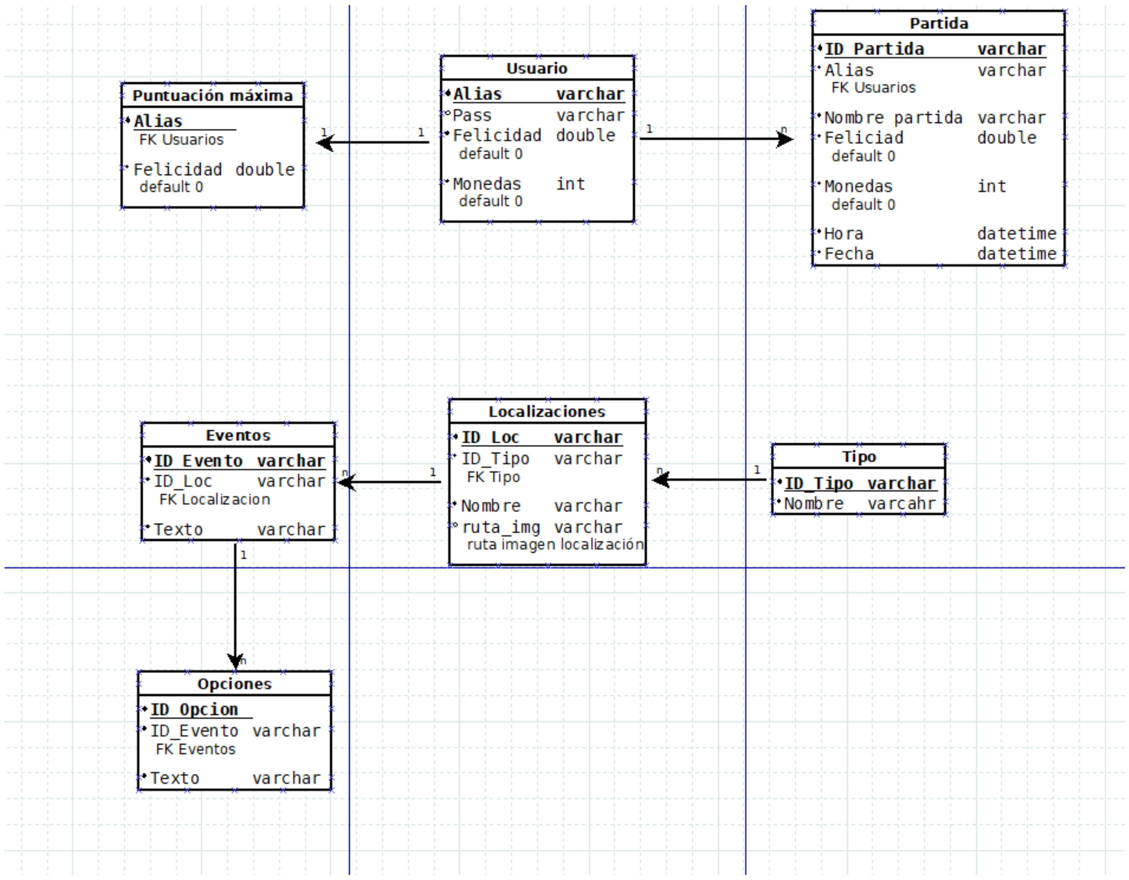
Name:	CDU3.7 Exit application
Description: The user can exit the application at any time.	
Actors: User.	
Preconditions: The application has been initiated.	
Post-conditions: None.	
Normal flow: 1- The user gives Exit. 2- Confirm that you want to exit the application.	
Alternative flow: 2.A. Cancel the action and keep the application open	

4.INITIAL PROTOTYPE

4.1.ER Diagram (initial prototype)



4.2.Class diagram (initial prototype)



5.TECHNOLOGIES USED ON PROJECT'S DEVELOPMENT

We have decided to use the technologies with which we have acquired the most experience during the course.

JAVA: It was the most used language during the course and the one we felt most comfortable with. The main characteristic -and advantage- of this programming language is that it is a platform-independent language, that is, any program created through Java will be able to work correctly on computers of all types and with different operating systems. This is a benefit for programmers, as it makes their work easier since they are no longer forced to create a different program that adapts to Windows, Linux, Mac... As the server part was from Linux and the client from Windows, it was ideal for our project.

Web service API REST: is an interface between systems that uses HTTP requests to obtain data or generate operations on that data in all possible formats, like JSON in our case.

MySQL: Although last year we used only ORACLE, MySQL is the technology used during this year. Among its many advantages is that it is also multiplatform and we have had it more present during this year.

MVC pattern: In our case it has been the most suitable model since we have worked in a project separated by three very well defined layers. This pattern allows us to separate an application into 3 layers, a way of organizing and making a project scalable

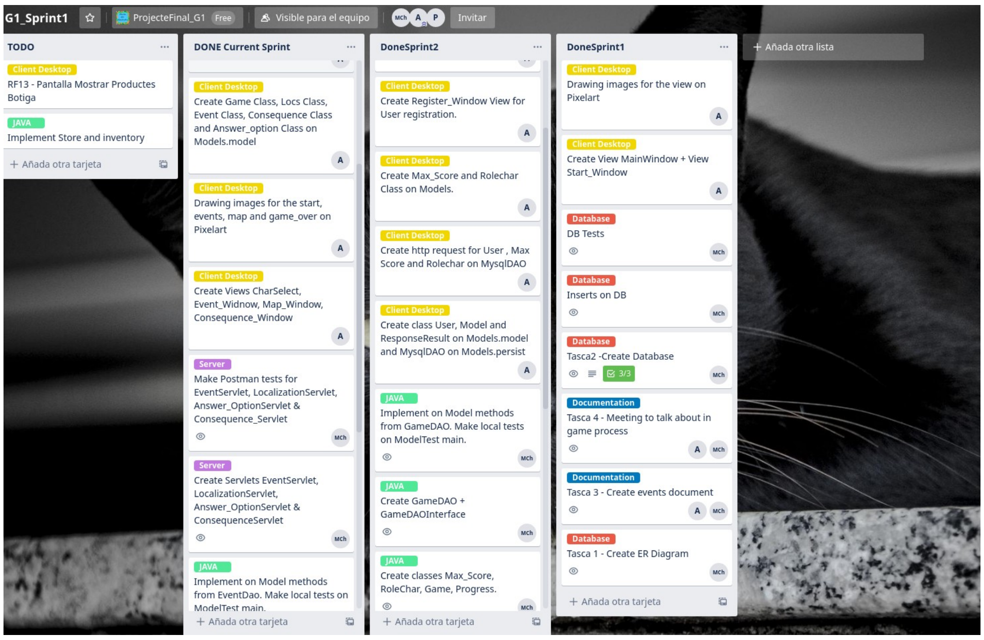
DAO Pattern: As you have an application that is not linked to data access, the DAO pattern is responsible for bringing you the data regardless of where it is stored.

Glassfish server: It is an application server that implements the technologies defined in the JAVAEE platform. We have decided to use it since it is the one we had seen in class.

C# Net Framework: C# is an object-oriented language that allows you to create a wide variety of applications. As it is one of the languages used and learned during the course we decided to use it for the client because of its great usefulness for the design by using WPF , it was ideal for our project.

6.FINAL DOCUMENTATION

We divided the job on frontend and backend. Maria was the responsable of backend (database, java model and web service) and Alexandra of frontend (client desktop and art).
Here is a capture of our trello with the three sprints:



G1_Sprint1 | ProjecteFinal_G1 Free | Visible para el equipo | MCh A P | Invitar

TODO

- Client Desktop — RF13 - Pantalla Mostrar Productes Botiga
- JAVA — Implement Store and inventory

+ Añada otra tarjeta

DONE Current Sprint

- Client Desktop — Create Game Class, Locs Class, Event Class, Consequence Class and Answer_option Class on Models.model — A
- Client Desktop — Drawing images for the start, events, map and game_over on Pixelart — A
- Client Desktop — Create Views CharSelect, Event_Widnow, Map_Window, Consequence_Window — A
- Server — Make Postman tests for EventServlet, LocalizationServlet, Answer_OptionServlet & Consequence_Servlet — MCh
- Server — Create Servlets EventServlet, LocalizationServlet, Answer_OptionServlet & ConsequenceServlet — MCh
- JAVA — Implement on Model methods from EventDao. Make local tests on ModelTest main.

+ Añada otra tarjeta

DoneSprint2

- Client Desktop — Create Register_Window View for User registration. — A
- Client Desktop — Create Max_Score and Rolechar Class on Models. — A
- Client Desktop — Create http request for User , Max Score and Rolechar on MysqlDAO — A
- Client Desktop — Create class User, Model and ResponseResult on Models.model and MysqlDAO on Models.persist — A
- JAVA — Implement on Model methods from GameDAO. Make local tests on ModelTest main. — MCh
- JAVA — Create GameDAO + GameDAOInterface — MCh
- JAVA — Create classes Max_Score, RoleChar, Game, Progress. — MCh

+ Añada otra tarjeta

DoneSprint1

- Client Desktop — Drawing images for the view on Pixelart — A
- Client Desktop — Create View MainWindow + View Start_Window — A
- Database — DB Tests — MCh
- Database — Inserts on DB — MCh
- Database — Tasca2 -Create Database — 3/3 — MCh
- Documentation — Tasca 4 - Meeting to talk about in game process — A MCh
- Documentation — Tasca 3 - Create events document — A MCh
- Database — Tasca 1 - Create ER Diagram — MCh

+ Añada otra tarjeta

+ Añada otra lista

<u>Sprint 1</u>

Documentation: Brainstorming to develop the events on game and functionality of it.

Backend: Create and implement database.

Frontend: Creation of mainwindow and startwindow. Creation of pixelart images for the background of both views.

<u>Sprint 2</u>

Backend: Developed the part of User management (model and servlets), Game management (max_score, game, rolechar and progress model).
Tests on postman for the user servlet.

Frontend: Drawing images for characters, create model for user management, testing of client-server connection.

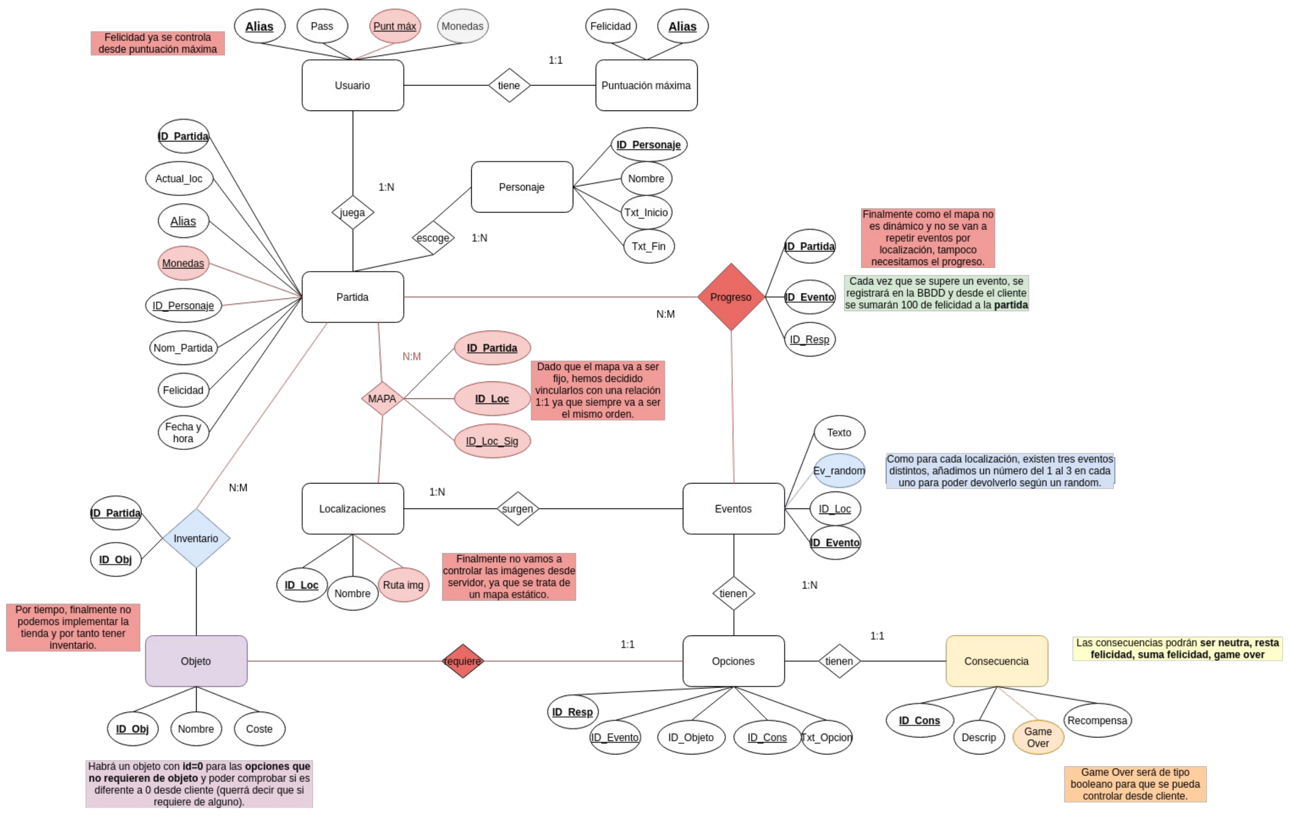
<u>Sprint 3</u>

Backend: Developed the part of Game servlets, and developed Events part (Event, Localization, Answer_option, Consequence) on dao and model. Made testing on local modeltest.
Created part of events servlet.
Tests on postman for the events servlets (EventServlet, LocalizationServlet, Answer_optionServlet, ConsequenceServlet).

Frontend: Drawing images for events and implementation and creation views and testing a full Game experience.

For more detailed information visit our Trello.

6.1.Final ER Diagram



Felicidad ya se controla desde puntuación máxima

Finalmente como el mapa no es dinámico y no se van a repetir eventos por localización, tampoco necesitamos el progreso.

Cada vez que se supere un evento, se registrará en la BBDD y desde el cliente se sumarán 100 de felicidad a la **partida**

Dado que el mapa va a ser fijo, hemos decidido vincularlos con una relación 1:1 ya que siempre va a ser el mismo orden.

Como para cada localización, existen tres eventos distintos, añadimos un número del 1 al 3 en cada uno para poder devolverlo según un random.

Finalmente no vamos a controlar las imágenes desde servidor, ya que se trata de un mapa estático.

Por tiempo, finalmente no podemos implementar la tienda y por tanto tener inventario.

Las consecuencias podrán **ser neutra, resta felicidad, suma felicidad, game over**

Habrá un objeto con **id=0** para las **opciones que no requieren de objeto** y poder comprobar si es diferente a 0 desde cliente (querrá decir que si requiere de alguno).

Game Over será de tipo booleano para que se pueda controlar desde cliente.

We had to make a few changes during the process. At last we've decided that the "map" on client desktop was static, a sequence of locations that always be on the same order, but the events received from server were random.

Because of this we had to eliminate the map entity and progress (they didn't had any sense on the actual planning).

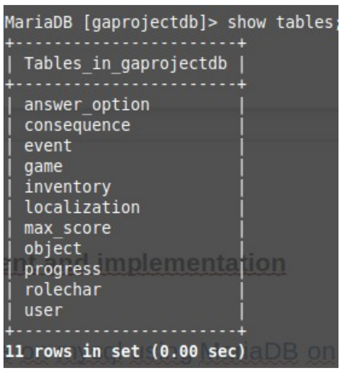
So we decided to make it more interesting incorporating the random events that some of them answer options could result on a "game over situation".

As a matter of resting time to work on the development of our game, we could not incorporate the part of store or inventory to exchange the winned coins.

If we had even one more sprint, we could add more functionalities as save game, load game, inventory and store.

6.2.Database development and implementation

I created our database on mysql using MariaDB on Linux (as seen and used in institute).
You can find the .sql document under resources directory on java project.
The tables created are:

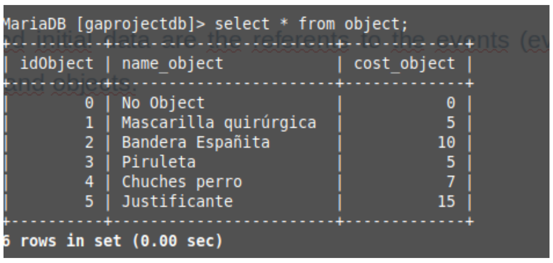


```
MariaDB [gaprojectdb]> show tables;
+-----------------------+
| Tables_in_gaprojectdb |
+-----------------------+
| answer_option         |
| consequence           |
| event                 |
| game                  |
| inventory             |
| localization          |
| max_score             |
| object                |
| progress              |
| rolechar              |
| user                  |
+-----------------------+
11 rows in set (0.00 sec)
```

Data cases

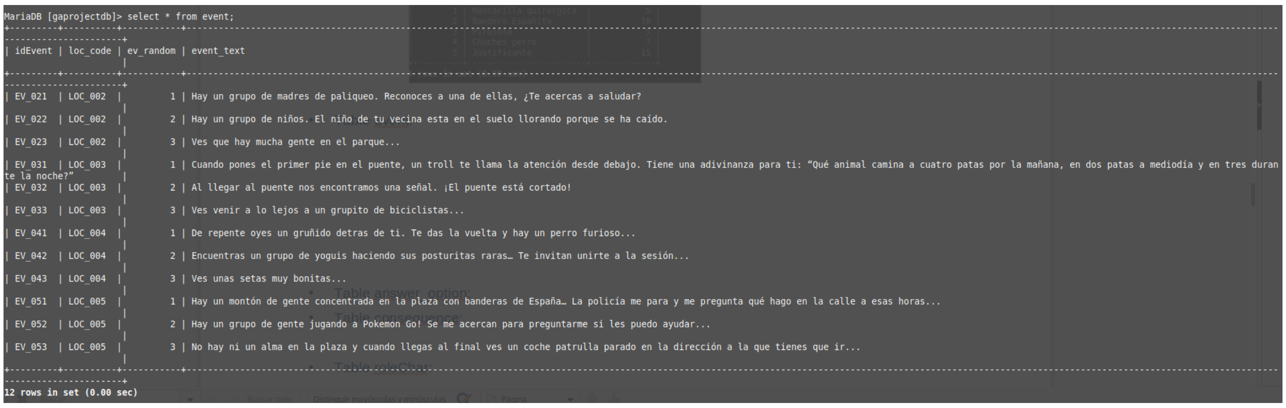
The tables that needed initial data are the referents to the events (event, answer_option, consequence), characters (rolechar), and objects. I attach a capture of the table and some inserts:

- Table objects:
  INSERT INTO `object` (`idObject`,`name_object`,`cost_object`) VALUES
  (0,'No Object',0),
  (1,'Mascarilla quirúrgica',5),
  (2,'Bandera Españita',10),
  (3,'Piruleta',5),
  (4,'Chuches perro',7),
  (5,'Justificante',15);



```
MariaDB [gaprojectdb]> select * from object;
+----------+-----------------------+-------------+
| idObject | name_object           | cost_object |
+----------+-----------------------+-------------+
|        0 | No Object             |           0 |
|        1 | Mascarilla quirúrgica |           5 |
|        2 | Bandera Españita      |          10 |
|        3 | Piruleta              |           5 |
|        4 | Chuches perro         |           7 |
|        5 | Justificante          |          15 |
+----------+-----------------------+-------------+
6 rows in set (0.00 sec)
```

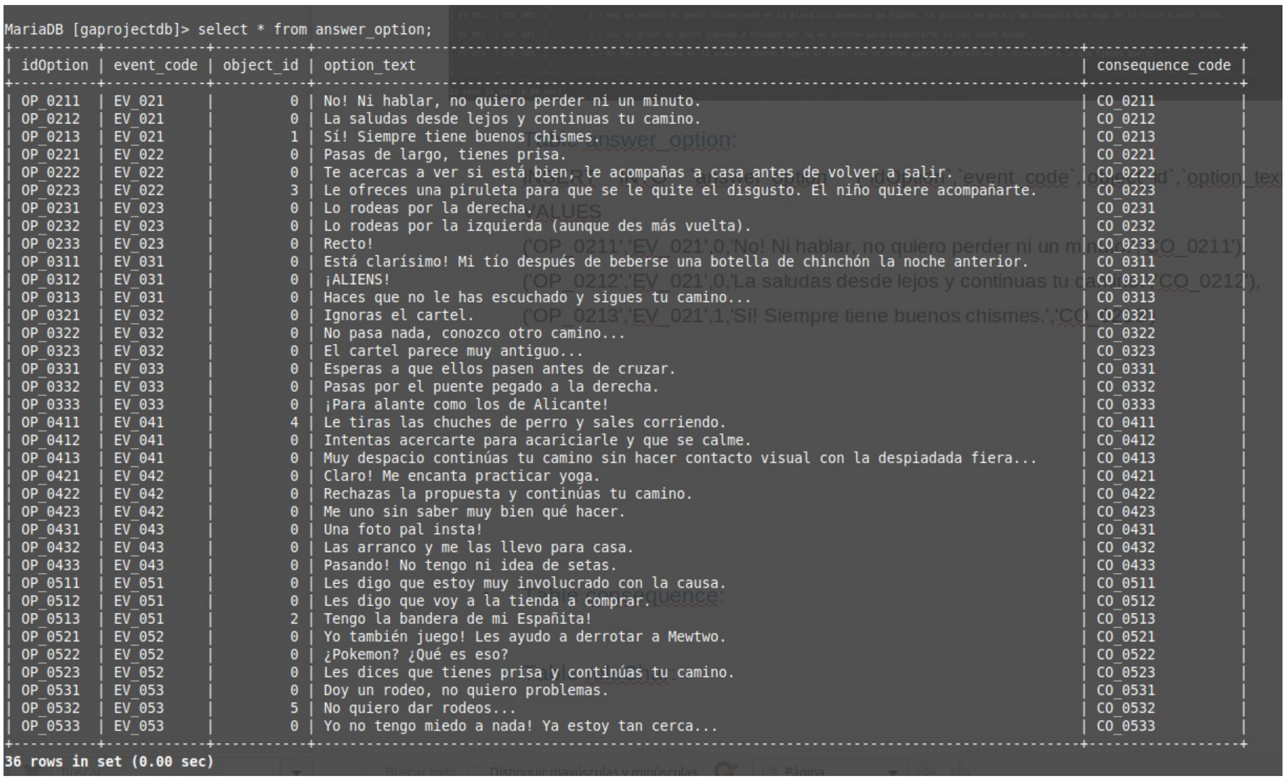
- Table event :
  INSERT INTO `event` (`idEvent`,`loc_code`,`ev_random`,`event_text`) VALUES
  ('EV_021','LOC_002',1,'Hay un grupo de madres de paliqueo. Reconoces a una de ellas, ¿Te acercas a saludar?'),
  ('EV_022','LOC_002',2,'Hay un grupo de niños. El niño de tu vecina esta en el suelo llorando porque se ha caído.'),
  ('EV_023','LOC_002',3,'Ves que hay mucha gente en el parque…')...



```
MariaDB [gaprojectdb]> select * from event;
+---------+----------+-----------+---------------------------------------------------------------------------------------------------------------------------------------------------
---------------------+
| idEvent | loc_code | ev_random | event_text
                     |
+---------+----------+-----------+---------------------------------------------------------------------------------------------------------------------------------------------------
---------------------+
| EV_021  | LOC_002  |         1 | Hay un grupo de madres de paliqueo. Reconoces a una de ellas, ¿Te acercas a saludar?
                     |
| EV_022  | LOC_002  |         2 | Hay un grupo de niños. El niño de tu vecina esta en el suelo llorando porque se ha caído.
                     |
| EV_023  | LOC_002  |         3 | Ves que hay mucha gente en el parque...
                     |
| EV_031  | LOC_003  |         1 | Cuando pones el primer pie en el puente, un troll te llama la atención desde debajo. Tiene una adivinanza para ti: “Qué animal camina a cuatro patas por la mañana, en dos patas a mediodía y en tres duran
te la noche?”        |
| EV_032  | LOC_003  |         2 | Al llegar al puente nos encontramos una señal. ¡El puente está cortado!
                     |
| EV_033  | LOC_003  |         3 | Ves venir a lo lejos a un grupito de biciclistas...
                     |
| EV_041  | LOC_004  |         1 | De repente oyes un gruñido detras de ti. Te das la vuelta y hay un perro furioso...
                     |
| EV_042  | LOC_004  |         2 | Encuentras un grupo de yoguis haciendo sus posturitas raras… Te invitan unirte a la sesión...
                     |
| EV_043  | LOC_004  |         3 | Ves unas setas muy bonitas...
                     |
| EV_051  | LOC_005  |         1 | Hay un montón de gente concentrada en la plaza con banderas de España… La policía me para y me pregunta qué hago en la calle a esas horas...
                     |
| EV_052  | LOC_005  |         2 | Hay un grupo de gente jugando a Pokemon Go! Se me acercan para preguntarme si les puedo ayudar...
                     |
| EV_053  | LOC_005  |         3 | No hay ni un alma en la plaza y cuando llegas al final ves un coche patrulla parado en la dirección a la que tienes que ir...
                     |
+---------+----------+-----------+---------------------------------------------------------------------------------------------------------------------------------------------------
---------------------+
12 rows in set (0.00 sec)
```

- <u>Table answer_option:</u>

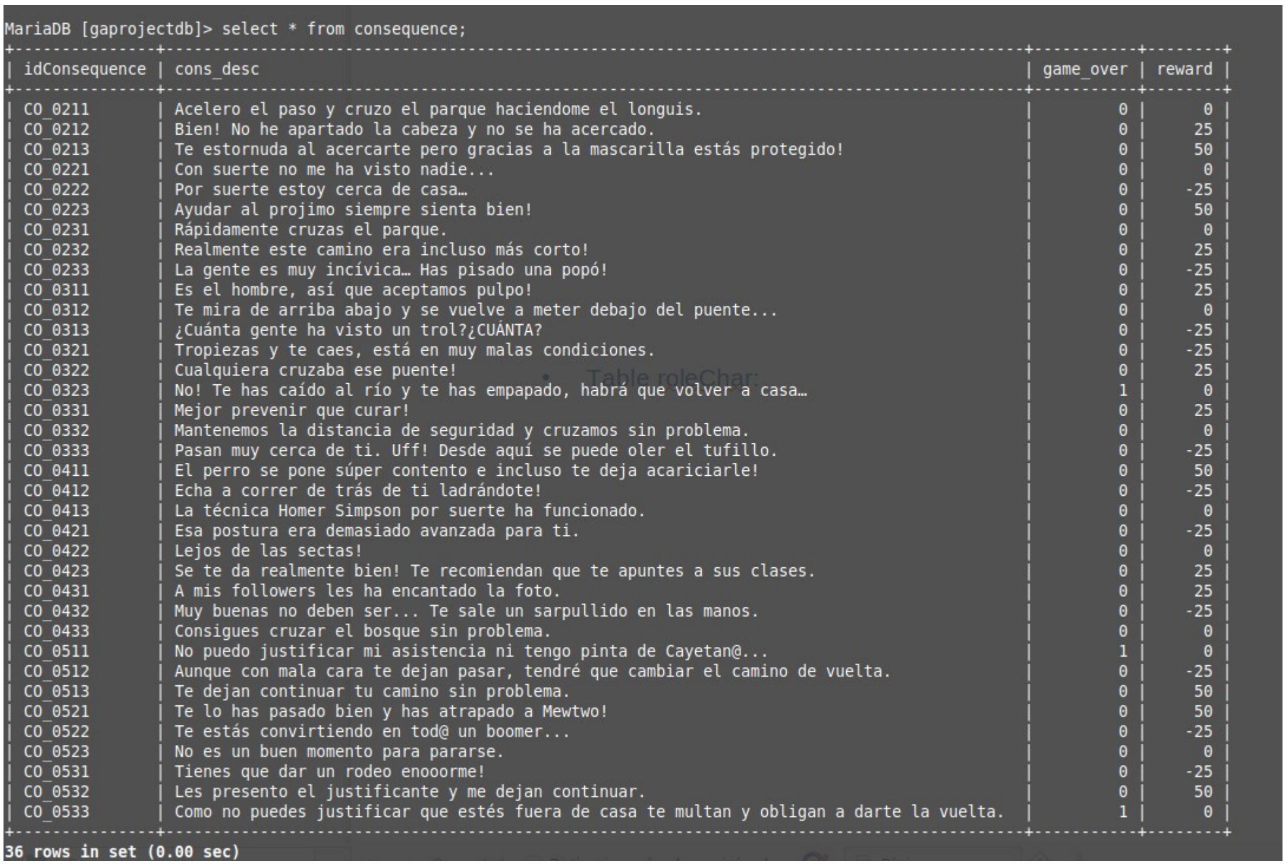
  INSERT INTO `answer_option` (`idOption`,`event_code`,`object_id`,`option_text`,`consequence_code`) VALUES
  ('OP_0211','EV_021',0,'No! Ni hablar, no quiero perder ni un minuto.','CO_0211'),
  ('OP_0212','EV_021',0,'La saludas desde lejos y continuas tu camino.','CO_0212'),
  ('OP_0213','EV_021',1,'Sí! Siempre tiene buenos chismes.','CO_0213')
  …



```
MariaDB [gaprojectdb]> select * from answer_option;
+----------+------------+-----------+-------------------------------------------------------------------------------+------------------+
| idOption | event_code | object_id | option_text                                                                   | consequence_code |
+----------+------------+-----------+-------------------------------------------------------------------------------+------------------+
| OP_0211  | EV_021     |         0 | No! Ni hablar, no quiero perder ni un minuto.                                 | CO_0211          |
| OP_0212  | EV_021     |         0 | La saludas desde lejos y continuas tu camino.                                 | CO_0212          |
| OP_0213  | EV_021     |         1 | Sí! Siempre tiene buenos chismes.                                             | CO_0213          |
| OP_0221  | EV_022     |         0 | Pasas de largo, tienes prisa.                                                 | CO_0221          |
| OP_0222  | EV_022     |         0 | Te acercas a ver si está bien, le acompañas a casa antes de volver a salir.   | CO_0222          |
| OP_0223  | EV_022     |         3 | Le ofreces una piruleta para que se le quite el disgusto. El niño quiere acompañarte. | CO_0223  |
| OP_0231  | EV_023     |         0 | Lo rodeas por la derecha.                                                     | CO_0231          |
| OP_0232  | EV_023     |         0 | Lo rodeas por la izquierda (aunque des más vuelta).                           | CO_0232          |
| OP_0233  | EV_023     |         0 | Recto!                                                                        | CO_0233          |
| OP_0311  | EV_031     |         0 | Está clarísimo! Mi tío después de beberse una botella de chinchón la noche anterior. | CO_0311   |
| OP_0312  | EV_031     |         0 | ¡ALIENS!                                                                      | CO_0312          |
| OP_0313  | EV_031     |         0 | Haces que no le has escuchado y sigues tu camino...                           | CO_0313          |
| OP_0321  | EV_032     |         0 | Ignoras el cartel.                                                            | CO_0321          |
| OP_0322  | EV_032     |         0 | No pasa nada, conozco otro camino...                                          | CO_0322          |
| OP_0323  | EV_032     |         0 | El cartel parece muy antiguo...                                               | CO_0323          |
| OP_0331  | EV_033     |         0 | Esperas a que ellos pasen antes de cruzar.                                    | CO_0331          |
| OP_0332  | EV_033     |         0 | Pasas por el puente pegado a la derecha.                                      | CO_0332          |
| OP_0333  | EV_033     |         0 | ¡Para alante como los de Alicante!                                            | CO_0333          |
| OP_0411  | EV_041     |         4 | Le tiras las chuches de perro y sales corriendo.                              | CO_0411          |
| OP_0412  | EV_041     |         0 | Intentas acercarte para acariciarle y que se calme.                           | CO_0412          |
| OP_0413  | EV_041     |         0 | Muy despacio continúas tu camino sin hacer contacto visual con la despiadada fiera... | CO_0413  |
| OP_0421  | EV_042     |         0 | Claro! Me encanta practicar yoga.                                             | CO_0421          |
| OP_0422  | EV_042     |         0 | Rechazas la propuesta y continúas tu camino.                                  | CO_0422          |
| OP_0423  | EV_042     |         0 | Me uno sin saber muy bien qué hacer.                                          | CO_0423          |
| OP_0431  | EV_043     |         0 | Una foto pal insta!                                                           | CO_0431          |
| OP_0432  | EV_043     |         0 | Las arranco y me las llevo para casa.                                         | CO_0432          |
| OP_0433  | EV_043     |         0 | Pasando! No tengo ni idea de setas.                                           | CO_0433          |
| OP_0511  | EV_051     |         0 | Les digo que estoy muy involucrado con la causa.                              | CO_0511          |
| OP_0512  | EV_051     |         0 | Les digo que voy a la tienda a comprar.                                       | CO_0512          |
| OP_0513  | EV_051     |         2 | Tengo la bandera de mi Españita!                                              | CO_0513          |
| OP_0521  | EV_052     |         0 | Yo también juego! Les ayudo a derrotar a Mewtwo.                              | CO_0521          |
| OP_0522  | EV_052     |         0 | ¿Pokemon? ¿Qué es eso?                                                        | CO_0522          |
| OP_0523  | EV_052     |         0 | Les dices que tienes prisa y continúas tu camino.                             | CO_0523          |
| OP_0531  | EV_053     |         0 | Doy un rodeo, no quiero problemas.                                            | CO_0531          |
| OP_0532  | EV_053     |         5 | No quiero dar rodeos...                                                       | CO_0532          |
| OP_0533  | EV_053     |         0 | Yo no tengo miedo a nada! Ya estoy tan cerca...                               | CO_0533          |
+----------+------------+-----------+-------------------------------------------------------------------------------+------------------+
36 rows in set (0.00 sec)
```

- <u>Table consequence:</u>

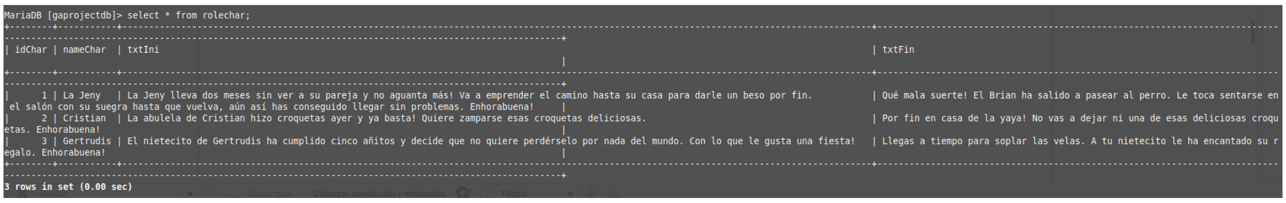
  INSERT INTO `consequence` (`idConsequence`,`cons_desc`,`game_over`,`reward`) VALUES
  ('CO_0211','Acelero el paso y cruzo el parque haciendome el longuis.',false,0),
  ('CO_0212','Bien! No he apartado la cabeza y no se ha acercado.',false,25),
  ...



```
MariaDB [gaprojectdb]> select * from consequence;
+---------------+-----------------------------------------------------------------------------------------+-----------+--------+
| idConsequence | cons_desc                                                                               | game_over | reward |
+---------------+-----------------------------------------------------------------------------------------+-----------+--------+
| CO_0211       | Acelero el paso y cruzo el parque haciendome el longuis.                                |         0 |      0 |
| CO_0212       | Bien! No he apartado la cabeza y no se ha acercado.                                     |         0 |     25 |
| CO_0213       | Te estornuda al acercarte pero gracias a la mascarilla estás protegido!                 |         0 |     50 |
| CO_0221       | Con suerte no me ha visto nadie...                                                      |         0 |      0 |
| CO_0222       | Por suerte estoy cerca de casa…                                                         |         0 |    -25 |
| CO_0223       | Ayudar al projimo siempre sienta bien!                                                  |         0 |     50 |
| CO_0231       | Rápidamente cruzas el parque.                                                           |         0 |      0 |
| CO_0232       | Realmente este camino era incluso más corto!                                            |         0 |     25 |
| CO_0233       | La gente es muy incívica… Has pisado una popó!                                          |         0 |    -25 |
| CO_0311       | Es el hombre, así que aceptamos pulpo!                                                  |         0 |     25 |
| CO_0312       | Te mira de arriba abajo y se vuelve a meter debajo del puente...                        |         0 |      0 |
| CO_0313       | ¿Cuánta gente ha visto un trol?¿CUÁNTA?                                                 |         0 |    -25 |
| CO_0321       | Tropiezas y te caes, está en muy malas condiciones.                                     |         0 |    -25 |
| CO_0322       | Cualquiera cruzaba ese puente!                                                          |         0 |     25 |
| CO_0323       | No! Te has caído al río y te has empapado, habrá que volver a casa…                     |         1 |      0 |
| CO_0331       | Mejor prevenir que curar!                                                               |         0 |     25 |
| CO_0332       | Mantenemos la distancia de seguridad y cruzamos sin problema.                           |         0 |      0 |
| CO_0333       | Pasan muy cerca de ti. Uff! Desde aquí se puede oler el tufillo.                        |         0 |    -25 |
| CO_0411       | El perro se pone súper contento e incluso te deja acariciarle!                          |         0 |     50 |
| CO_0412       | Echa a correr de trás de ti ladrándote!                                                 |         0 |    -25 |
| CO_0413       | La técnica Homer Simpson por suerte ha funcionado.                                      |         0 |      0 |
| CO_0421       | Esa postura era demasiado avanzada para ti.                                             |         0 |    -25 |
| CO_0422       | Lejos de las sectas!                                                                    |         0 |      0 |
| CO_0423       | Se te da realmente bien! Te recomiendan que te apuntes a sus clases.                    |         0 |     25 |
| CO_0431       | A mis followers les ha encantado la foto.                                               |         0 |     25 |
| CO_0432       | Muy buenas no deben ser... Te sale un sarpullido en las manos.                          |         0 |    -25 |
| CO_0433       | Consigues cruzar el bosque sin problema.                                                |         0 |      0 |
| CO_0511       | No puedo justificar mi asistencia ni tengo pinta de Cayetan@...                         |         1 |      0 |
| CO_0512       | Aunque con mala cara te dejan pasar, tendré que cambiar el camino de vuelta.            |         0 |    -25 |
| CO_0513       | Te dejan continuar tu camino sin problema.                                              |         0 |     50 |
| CO_0521       | Te lo has pasado bien y has atrapado a Mewtwo!                                          |         0 |     50 |
| CO_0522       | Te estás convirtiendo en tod@ un boomer...                                              |         0 |    -25 |
| CO_0523       | No es un buen momento para pararse.                                                     |         0 |      0 |
| CO_0531       | Tienes que dar un rodeo enooorme!                                                       |         0 |    -25 |
| CO_0532       | Les presento el justificante y me dejan continuar.                                      |         0 |     50 |
| CO_0533       | Como no puedes justificar que estés fuera de casa te multan y obligan a darte la vuelta. |        1 |      0 |
+---------------+-----------------------------------------------------------------------------------------+-----------+--------+
36 rows in set (0.00 sec)
```

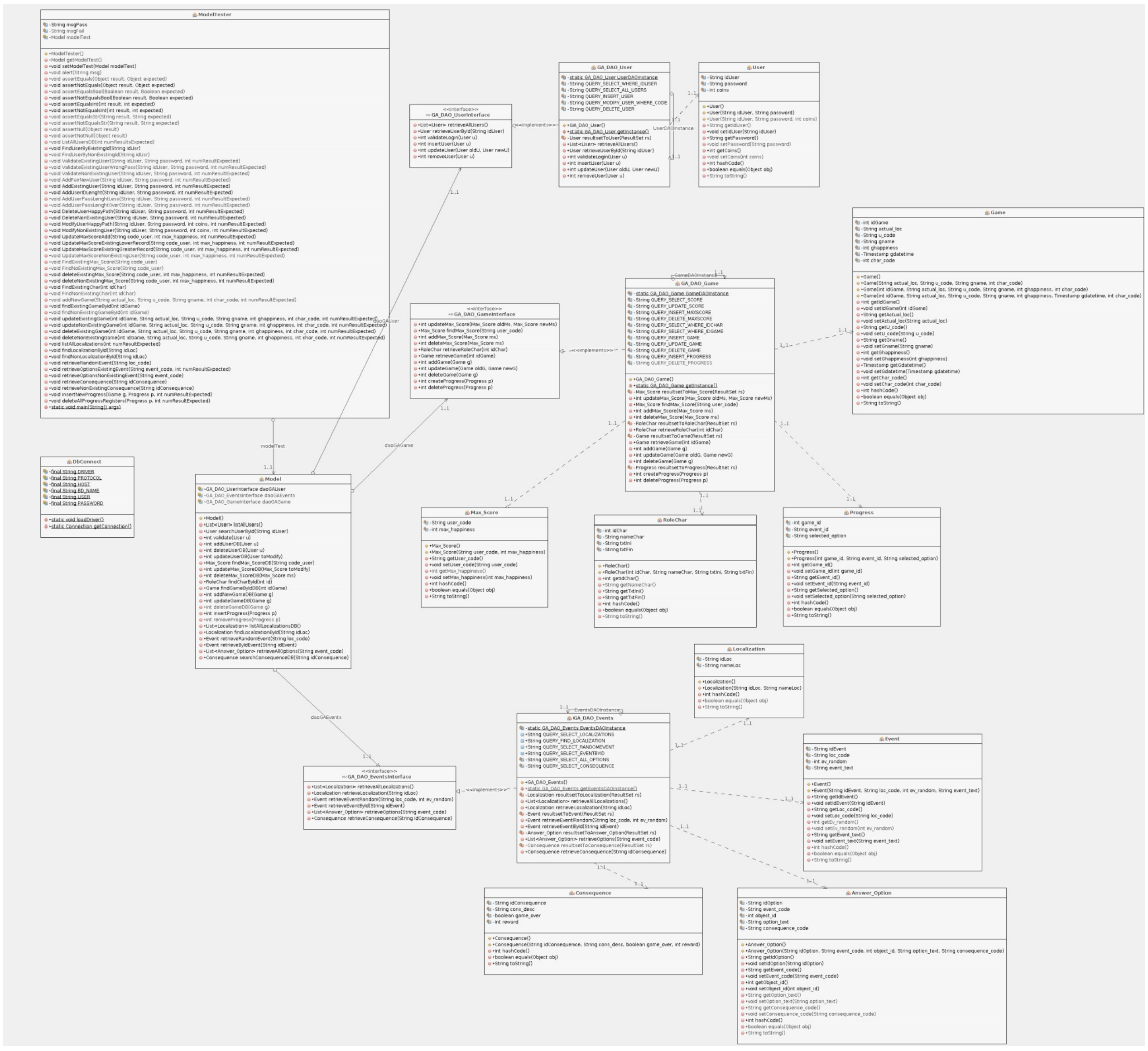
- <u>Table roleChar:</u>

  INSERT INTO `rolechar` (`nameChar`,`txtIni`,`txtFin`) VALUES ('La Jeny','La Jeny lleva dos meses sin ver a su pareja y no aguanta más! Va a emprender el camino hasta su casa para darle un beso por fin.','Qué mala suerte! El Brian ha salido a pasear al perro. Le toca sentarse en el salón con su suegra hasta que vuelva, aún así has conseguido llegar sin problemas. Enhorabuena!'), ('Cristian','La abulela de Cristian hizo croquetas ayer y ya basta! Quiere zamparse esas croquetas deliciosas.','Por fin en casa de la yaya! No vas a dejar ni una de esas deliciosas croquetas. Enhorabuena!'), ('Gertrudis','El nietecito de Gertrudis ha cumplido cinco añitos y decide que no quiere perdérselo por nada del mundo. Con lo que le gusta una fiesta!','Llegas a tiempo para soplar las velas. A tu nietecito le ha encantado su regalo. Enhorabuena!');

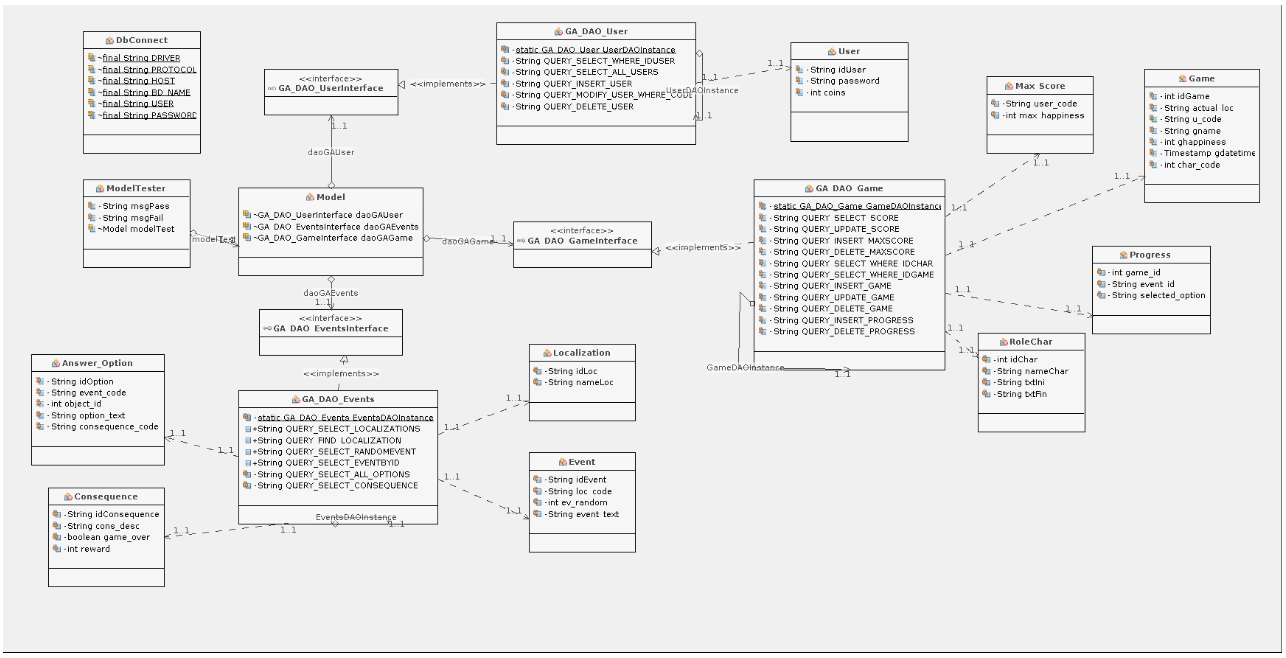


```
MariaDB [gaprojectdb]> select * from rolechar;
+--------+-----------+---------------------------------------------------------------------------------------------------------------------------------------+------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------+
| idChar | nameChar  | txtIni                                                                                                                                | txtFin
                                                                                                        |
+--------+-----------+---------------------------------------------------------------------------------------------------------------------------------------+------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------+
|      1 | La Jeny   | La Jeny lleva dos meses sin ver a su pareja y no aguanta más! Va a emprender el camino hasta su casa para darle un beso por fin.           | Qué mala suerte! El Brian ha salido a pasear al perro. Le toca sentarse en
 el salón con su suegra hasta que vuelva, aún así has conseguido llegar sin problemas. Enhorabuena!     |
|      2 | Cristian  | La abulela de Cristian hizo croquetas ayer y ya basta! Quiere zamparse esas croquetas deliciosas.                                     | Por fin en casa de la yaya! No vas a dejar ni una de esas deliciosas croqu
etas. Enhorabuena!                                                                                      |
|      3 | Gertrudis | El nietecito de Gertrudis ha cumplido cinco añitos y decide que no quiere perdérselo por nada del mundo. Con lo que le gusta una fiesta!   | Llegas a tiempo para soplar las velas. A tu nietecito le ha encantado su r
egalo. Enhorabuena!                                                                                     |
+--------+-----------+---------------------------------------------------------------------------------------------------------------------------------------+------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------+
3 rows in set (0.00 sec)
```

6.3.Final Classes diagram



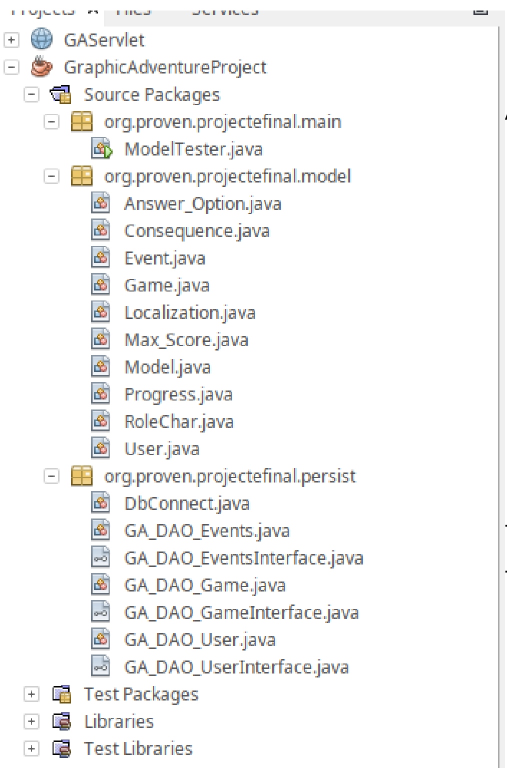
As you can see, we had to simplificate it to make it more legible:



6.4.JAVA Model

This classes diagram corresponds to the JAVA model. Is the one who comunicates servlets with database.

We decided to use a singleton pattern and DAOs to implement our data model.

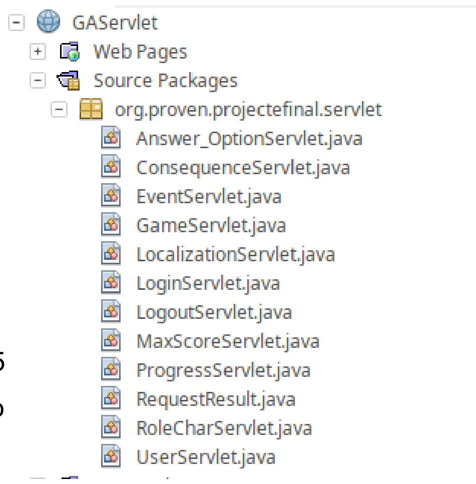


```
+ GAServlet
- GraphicAdventureProject
  - Source Packages
    - org.proven.projectefinal.main
        ModelTester.java
    - org.proven.projectefinal.model
        Answer_Option.java
        Consequence.java
        Event.java
        Game.java
        Localization.java
        Max_Score.java
        Model.java
        Progress.java
        RoleChar.java
        User.java
    - org.proven.projectefinal.persist
        DbConnect.java
        GA_DAO_Events.java
        GA_DAO_EventsInterface.java
        GA_DAO_Game.java
        GA_DAO_GameInterface.java
        GA_DAO_User.java
        GA_DAO_UserInterface.java
  + Test Packages
  + Libraries
  + Test Libraries
```

At last we separated operations on 3 DAOs:

- UserDAO: Manages all operations of User class.
- EventDAO: Manages all operations about events (event, answer_option, consequence and localization classes).
- GameDAO: Manages the operations about game (game, max_score, progress and rolechar classes).

I have to give the name RoleChar to class for characters because there was some type of conflict with that word on mysql, and had to change the name of the class.



```
- GAServlet
  + Web Pages
  - Source Packages
    - org.proven.projectefinal.servlet
        Answer_OptionServlet.java
        ConsequenceServlet.java
        EventServlet.java
        GameServlet.java
        LocalizationServlet.java
        LoginServlet.java
        LogoutServlet.java
        MaxScoreServlet.java
        ProgressServlet.java
        RequestResult.java
        RoleCharServlet.java
        UserServlet.java
```

6.5.Web service

I decided to implement a RestFul API Web Service as seen on M015 assignature.

I developed a single servlet for every class on java model, as we did on our M015 project, to have a more legible and distributed code (a thought it will be easiest to manage from client).

There's a resum of the code developed on the web service:

<u>User Servlet</u>

Operation	Action	Method	Parameters	Error codes
FindAllUsers	findAll	GET	/user?action=findAll	1 OK (return JSON with all users) 0 Empty return
indUserById	findById	GET	/user?action=findById&idUser=nameuser	1 OK (return JSON with User) 0 User doesn't exists
AddUser	add	POST	/user?action=add JSON with User to add	1 OK 0 SQL Exception or error during process -1 User already exists -2 idUser lenght over than 15 -3 password lenght minor than 5 or over 8
DeleteUser	delete	POST	/user?action=delete JSON with User to delete	1 OK 0 Error during process -1 User doesn't exists
ModifyUser	modify	POST	/user?action=modify JSON with User to save on DB	1 OK 0 Error during process -1 User doesn't exists

Login Servlet

Operation	Action	Method	Parameters	Error codes
doPost method on login servlet		POST	/login JSON with User formed with parameters needed to validate (idUser, password)	1 OK 0 User doesn't exists -1 Password is not valid

Logout Servlet

Operation	Action	Method	Parameters	Error codes
processRequest on logout servlet			/logout	1 OK -1 User not authenticated

RoleChar Servlet

Operation	Action	Method	Parameters	Error codes
searchChar	findChar	GET	/rchar?action=findChar&idChar=charid	1 OK (return JSON RoleChar) 0 No char found. -1 Number Format Exception -2 idChar cannot be minor than 1

Max_Score Servlet

Operation	Action	Method	Parameters	Error codes
searchScore	findScore	GET	/maxscore?action=findScore&user_code=nameUser	1 OK (return JSON MAX_SCORE) 0 No record registered for this user yet.
updateScore	setScore	POST	/maxscore?action=setScore JSON with max_score object	2 Record successfully added 1 Record successfully updated 0 Error during process -1 The record registered is greater than the one given -2 No user registered for the code_user given
removeScore	deleteScore	POST	/maxscore?action=deleteScore JSON with max_score object	-1 Max_Score not found 0 Error during process 1 Successfully deleted

Game Servlet

Operation	Action	Method	Parameters	Error codes
searchGame	findGame	GET	/game?action=findScore&u_code=idGame	1 OK (return JSON GAME) 0 No game found by idGame -1 Number Format Exception
insertGame	addGame	POST	/game?action=addGame JSON with Game object	1 OK 0 Error during process -1 gName lenght over 15
updateGame	setGame	POST	/game?action=setGame JSON with Game object	1 OK successfully updated 0 Error during process -1 Game doesn't exists on DB -2 Object stored and recieved are not equals
removeGame	deleteGame	POST	/game?action=deleteGame JSON with Game object	-1 Max_Score not found 0 Error during process 1 Successfully deleted

Localization Servlet

Operation	Action	Method	Parameters	Error codes
searchLocs	findLocs	GET	/loc?action=findLocs	1 OK (return JSON List of Locs) 0 Error during process
findLoc	findLocById	GET	/loc?action=findLocById&idLoc= idLoc	1 OK (return JSON Localization) 0 Localization not found.

Event Servlet

Operation	Action	Method	Parameters	Error codes
retrieveRandomEvent	randEvent	GET	/event?action=randEvent&loc_code=loc_code	1 OK (return JSON with event) 0 Loc code given doesn't have events or doesn't exists
retrieveEventById	findEvent	GET	/event?action=findEvent&idEvent=idevent	1 OK (return JSON with event) 0 Doesn't exists

Answer_Option Servlet

Operation	Action	Method	Parameters	Error codes
findAnswerOptions	findOpt	GET	/answer?action=findOpt&event_code=event_code	1 OK (return JSON with Answer_Option list) 0 Event code given doesn't exists

Consequence Servlet

Operation	Action	Method	Parameters	Error codes
retrieveConsequence	findCons	GET	/cons?action=findCons&idConsequence=idConsequence	1 OK (return JSON with Consequence object) 0 idConsequence doesn't exists

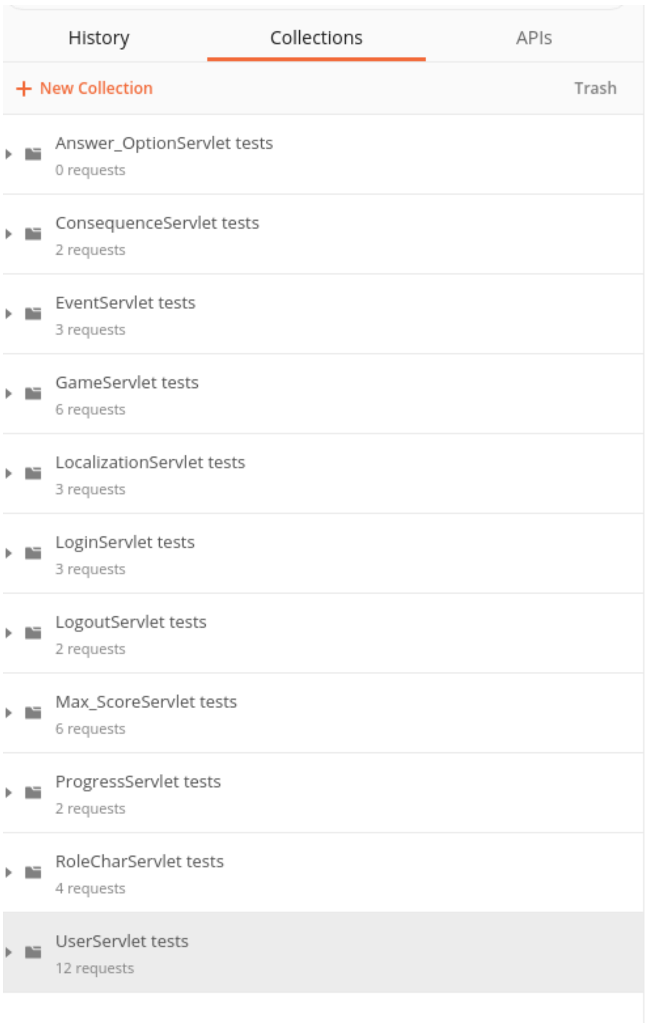
Progress Servlet

Operation	Action	Method	Parameters	Error codes
insertProgress	addPro	POST	/progress?action=addPro JSON with Progress object	1 OK 0 Error during process
removeProgress	deletePro	POST	/progress?action=deletePro JSON with Progress object	1 OK 0 No matches for the game_id of the given Progress object

I did this resum/guide for it to be more easiest to understand from client what Alexandra must recieve and send. Almost the expected code result for every action.
I had to search how to recieve JSON objects and manages them for the POST petitions. I thought it will be difficult but i found it easy to understand and to implement.

6.6.Postman testing

Someone has told me about to make the tests of http petitions with this app, because the addon Rest used on M015 sometimes fails and is not possible to have collections as Postman does.
Actually i have this collections on my Postman app:



Answer_OptionServlet tests
0 requests

ConsequenceServlet tests
2 requests

EventServlet tests
3 requests

GameServlet tests
6 requests

LocalizationServlet tests
3 requests

LoginServlet tests
3 requests

LogoutServlet tests
2 requests

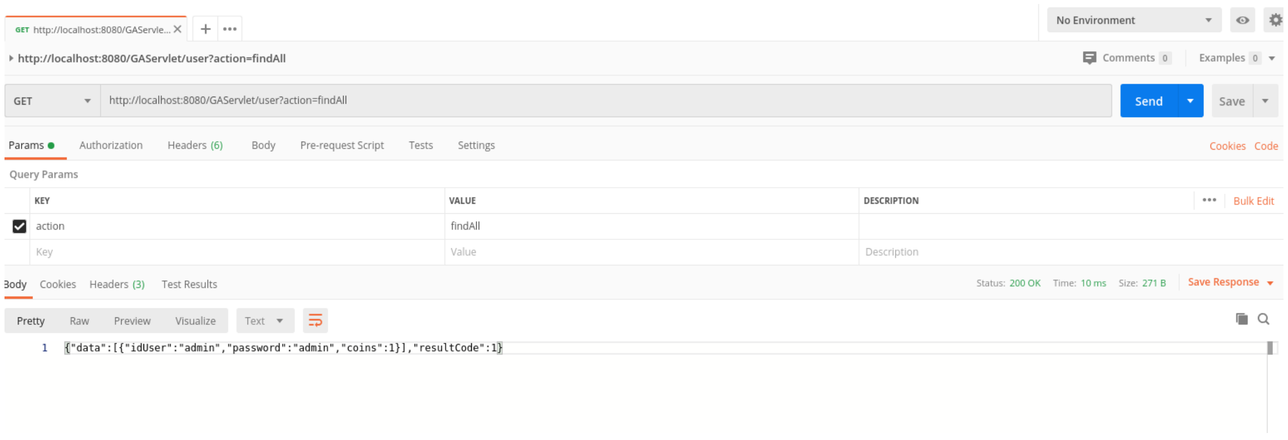
Max_ScoreServlet tests
6 requests

ProgressServlet tests
2 requests

RoleCharServlet tests
4 requests

UserServlet tests
12 requests

I've created one collection to every Servlet to save and recuperate any petition tested before.
Let me show you one test for GET petition and one for POST petition (waiting for a request as a JSON object).
Let's take User's servlet tests as an example. For the GET petition we want to list all the users found on DB:



http://localhost:8080/GAServlet/user?action=findAll

GET http://localhost:8080/GAServlet/user?action=findAll

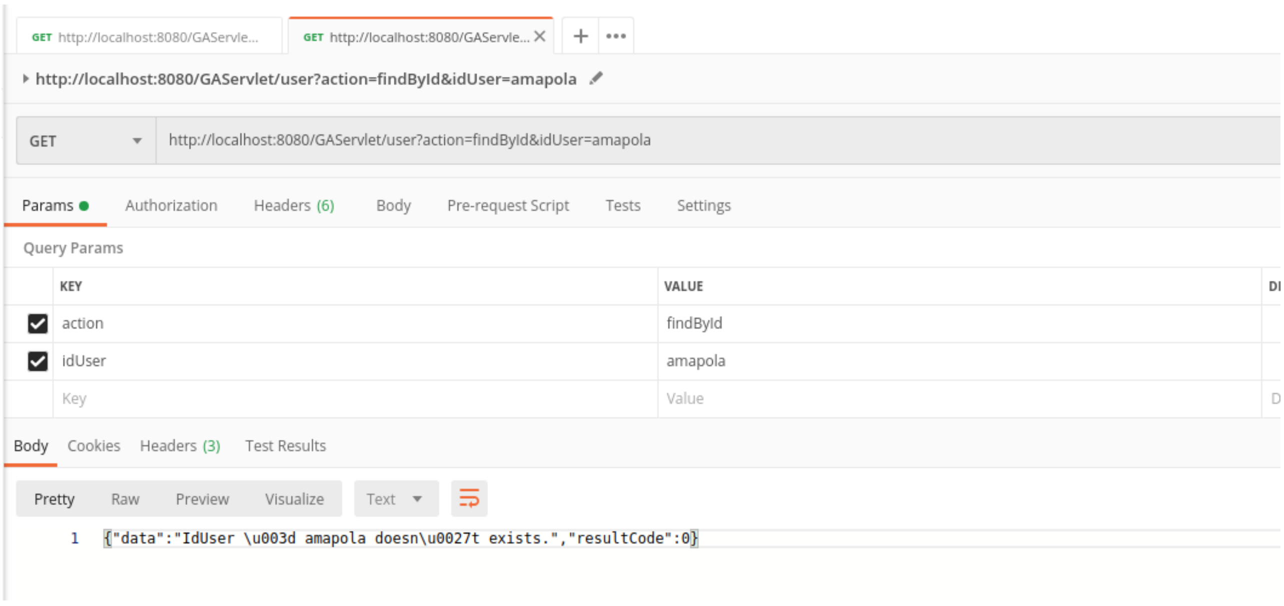
Query Params

KEY	VALUE	DESCRIPTION
✓ action	findAll	
Key	Value	Description

Status: 200 OK Time: 10 ms Size: 271 B

```
{"data":[{"idUser":"admin","password":"admin","coins":1}],"resultCode":1}
```

Let’s try to find a non existing user on DB:


http://localhost:8080/GAServlet/user?action=findById&idUser=amapola

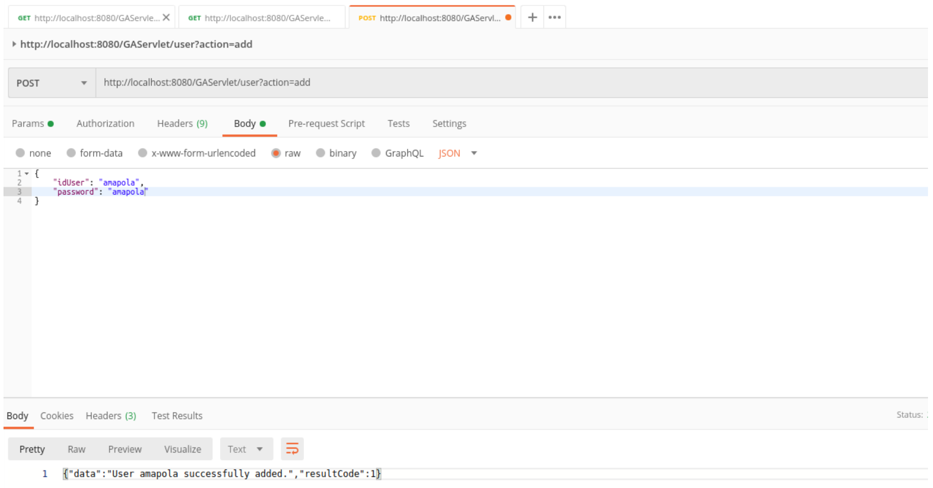
GET http://localhost:8080/GAServlet/user?action=findById&idUser=amapola

Query Params

KEY	VALUE
action	findById
idUser	amapola

```
{"data":"IdUser \u003d amapola doesn\u0027t exists.","resultCode":0}
```

Now let’s see a POST example for adding a new user to DB:


http://localhost:8080/GAServlet/user?action=add

POST http://localhost:8080/GAServlet/user?action=add

```
{
    "idUser": "amapola",
    "password": "amapola"
}
```

```
{"data":"User amapola successfully added.","resultCode":1}
```

As you can see i’m passing as a JSON format the parameter needed for request (an user object).

I will attach all tests made on Postman to the documentation of the project.

6.7.Explanation of client code

The AQuarantineAdventure project is equivalent to the client part of our G1 group.

It contains the windows, classes, images, model and connection to the server through Http requests in JSON format.

It is Model-ViewController, which means that the view controls most of the errors, messages etc. that may arise during the game.

This project is made with Visual Studio 2019 Community
It is a Windows Presentation Foundation desktop application in C# language

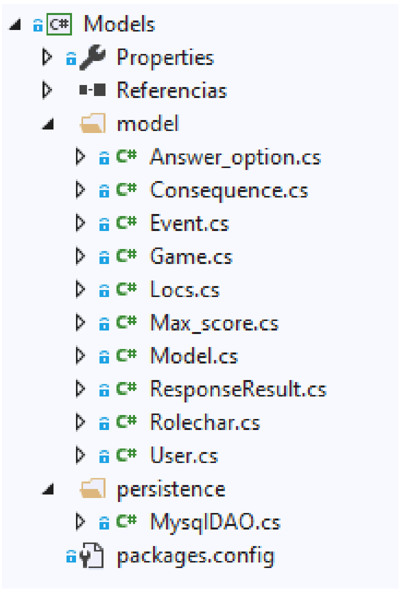
The solution consists of two projects, Models and AQuarantineAdventure which is the view-controller.

Models contains 2 fields:

Models.model: Contains the ADT classes Answer_option, Consequence, Event, Game, Locs, Max_score, ResponseResult, Rolechar and User with their attributes, constructors, setters and getters , and some, tostring, equals and hashcode when needed.

And the Model class, which is the link between the view-controller and the persistence (connection to the server) and contains all the necessary methods to collect the requests that are sent from the view (either when loading the window, clicking a button, etc...) save them, and send them to the persistence to make the required request, receive the response and return it to the view-controller.

It also contains the attributes, the instances of the classes and the getters and setters of the objects that had to be saved when changing views, so that the next screen could retrieve them and use them if needed.



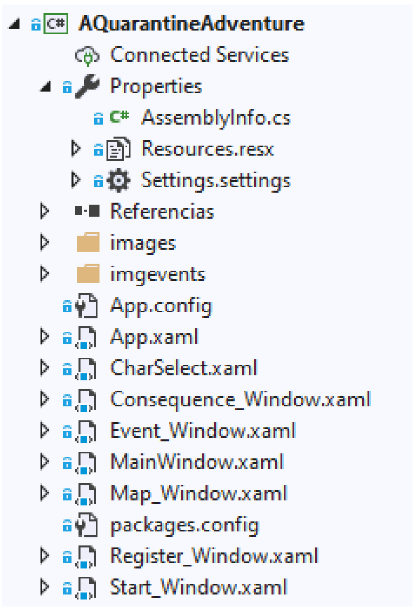
```
Models
  Properties
  Referencias
  model
    Answer_option.cs
    Consequence.cs
    Event.cs
    Game.cs
    Locs.cs
    Max_score.cs
    Model.cs
    ResponseResult.cs
    Rolechar.cs
    User.cs
  persistence
    MysqlDAO.cs
  packages.config
```

Models.persistence: Contains the MysqlDAO class with its attributes, instances and get and post http request methods to send and receive data from the server, which contains the database.

AquarantineAdventure: Contains 2 image folders (images and imgevents) and the CharSelect, Consequence_Window, Event_Window, MainWindow, Map_Window windows. Register_Window and Start_Window.

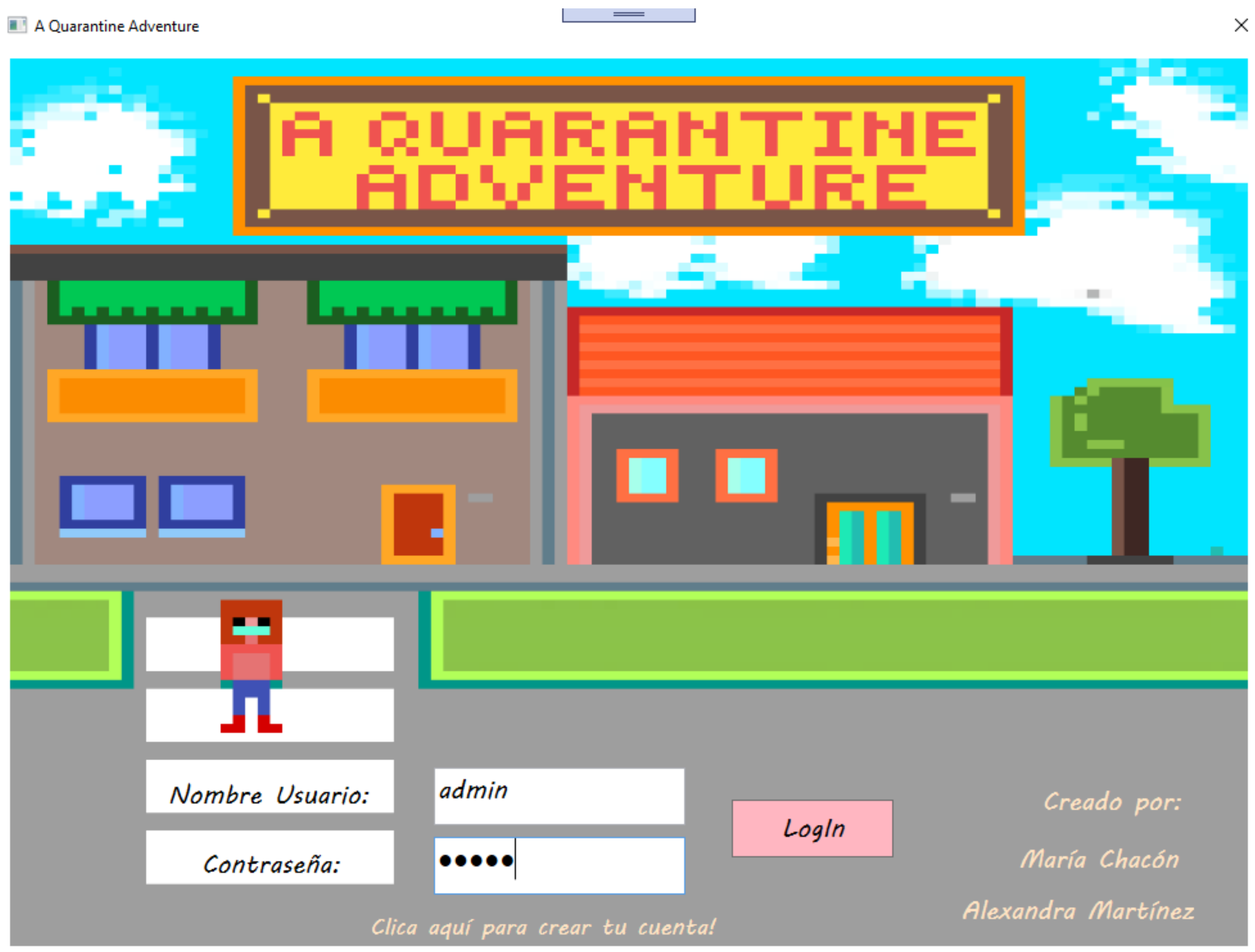
These wpf windows contain the user's interaction with the game and the user's interaction with the code behind it.
These windows, running, allow the user to register, log in, delete his user, start a new game, choose a character and name the game, play (move forward on the map by solving the different events that come up) and reach the end to gain a score, which is then updated in the database.

The URLs needed to make http requests are all located in Properties.Settings.settings. So if you need to change the URL for one reason or another, just change the URL of that file.

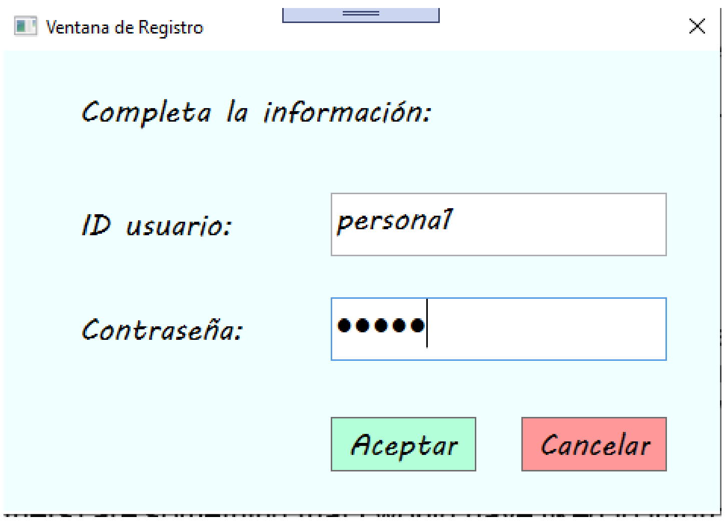


```
AQuarantineAdventure
  Connected Services
  Properties
    AssemblyInfo.cs
    Resources.resx
    Settings.settings
  Referencias
  images
  imgevents
  App.config
  App.xaml
  CharSelect.xaml
  Consequence_Window.xaml
  Event_Window.xaml
  MainWindow.xaml
  Map_Window.xaml
  packages.config
  Register_Window.xaml
  Start_Window.xaml
```

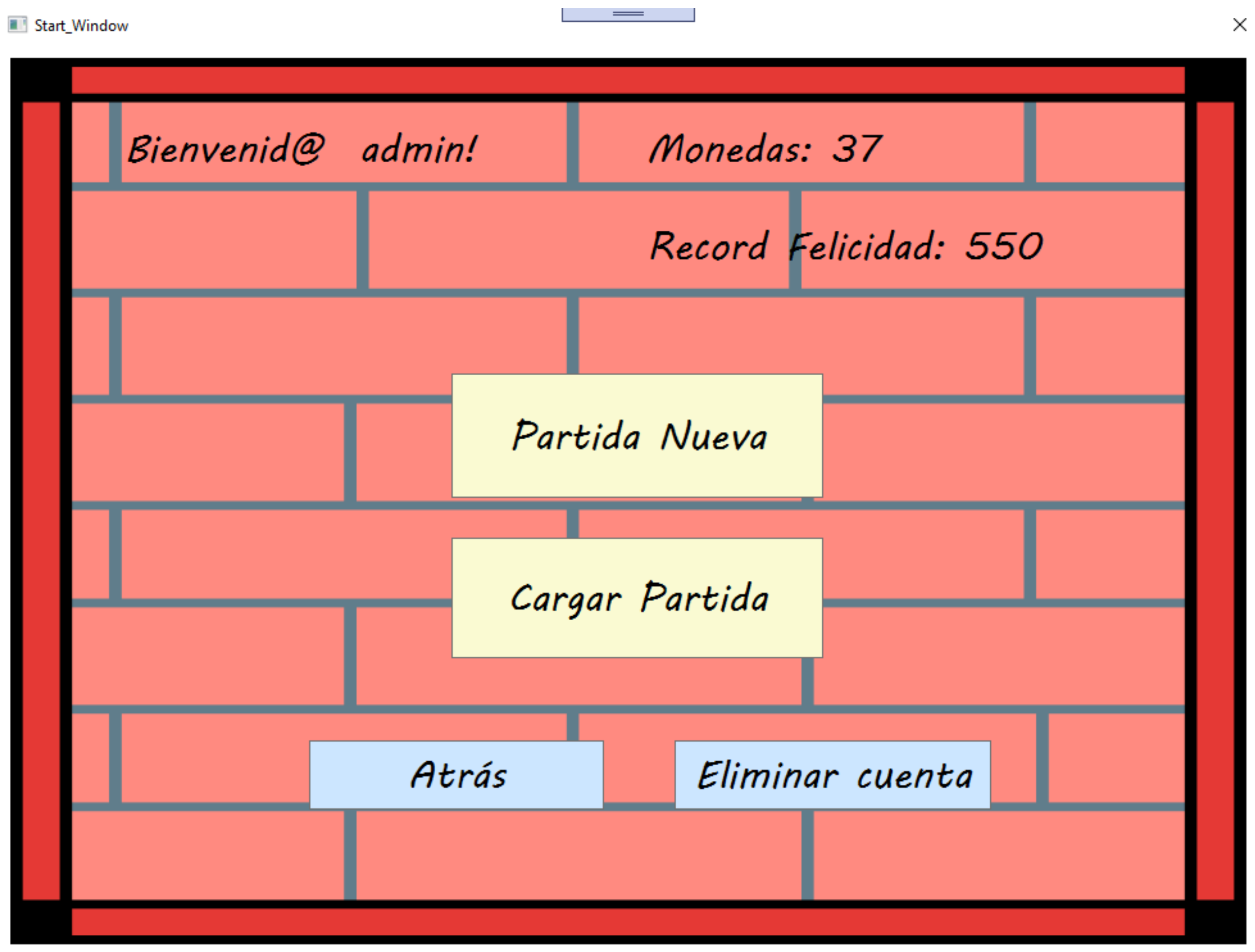
6.8.Prototypes



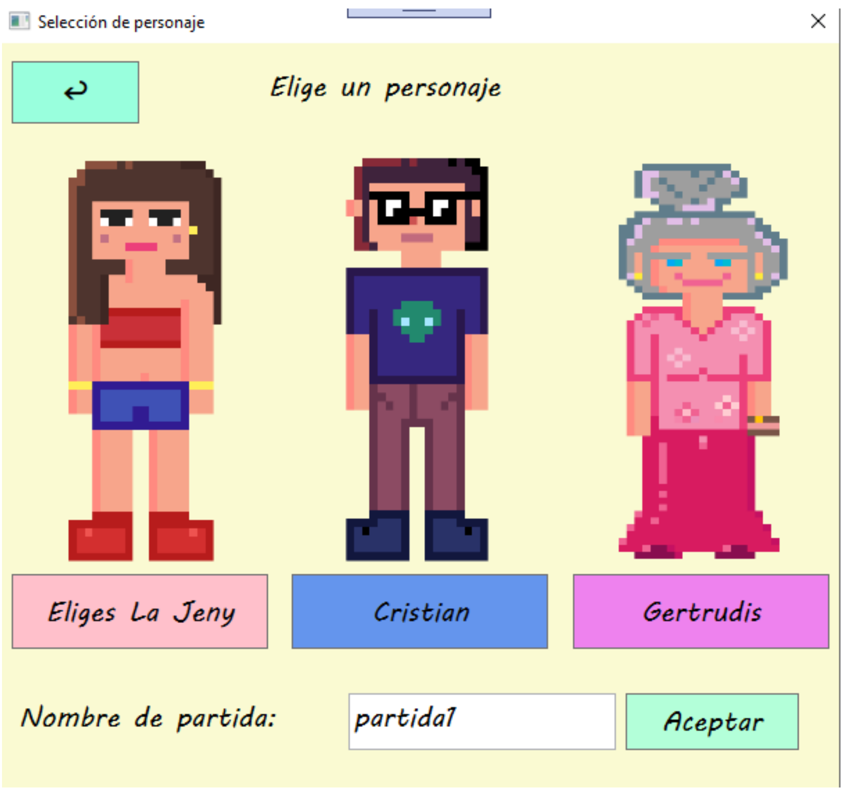
RF1 Main Window



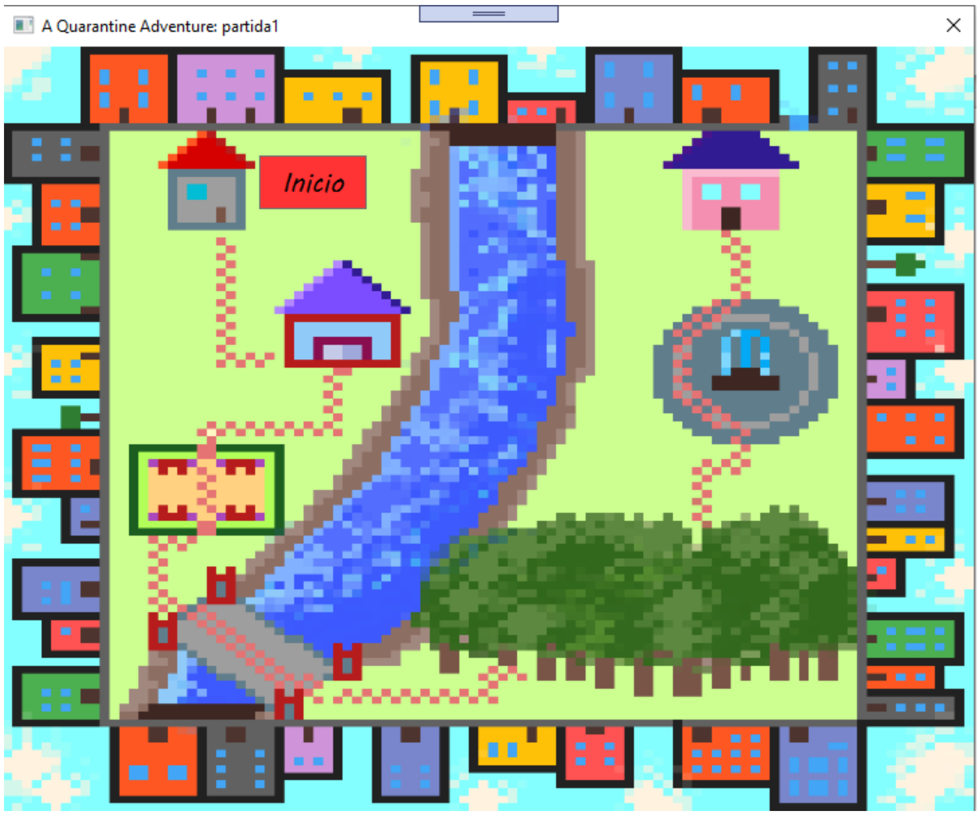
RF1.1 Register



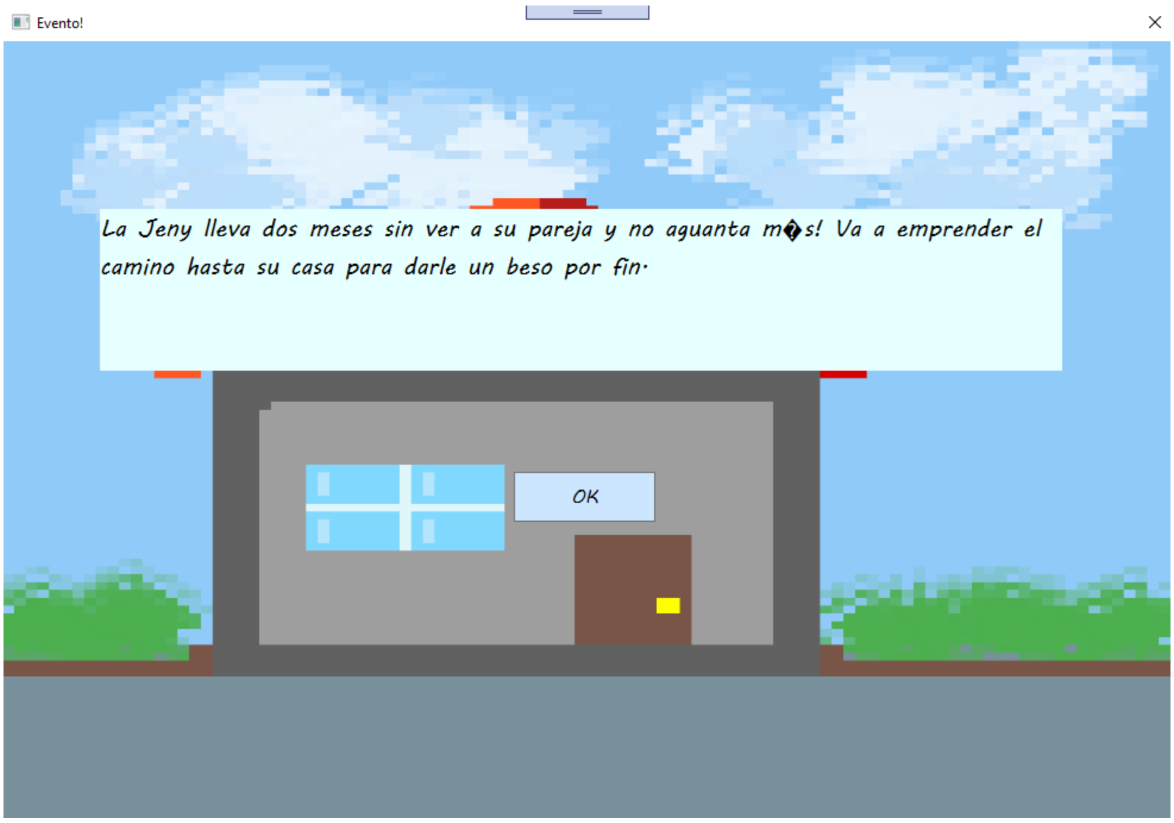
RF2- Game start window



RF2.1-Start New Game RF2.1.1 Character Selection

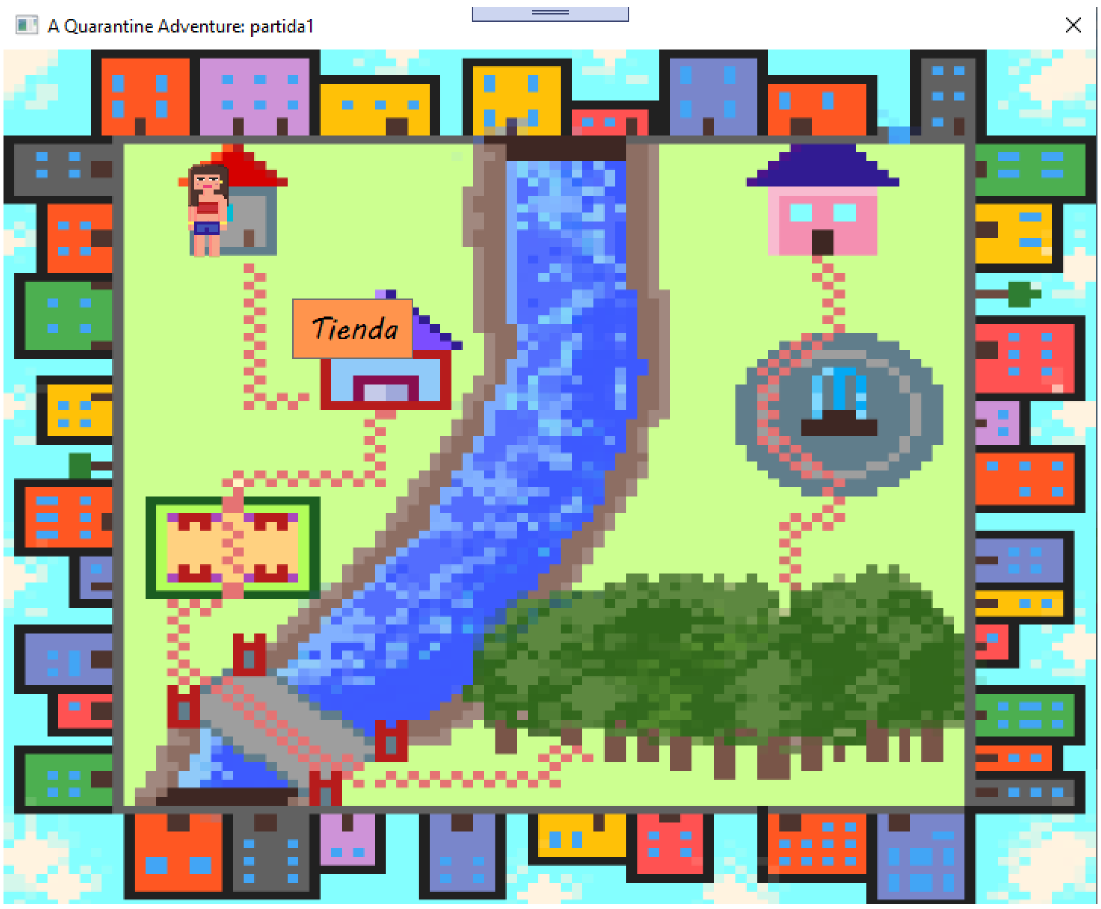


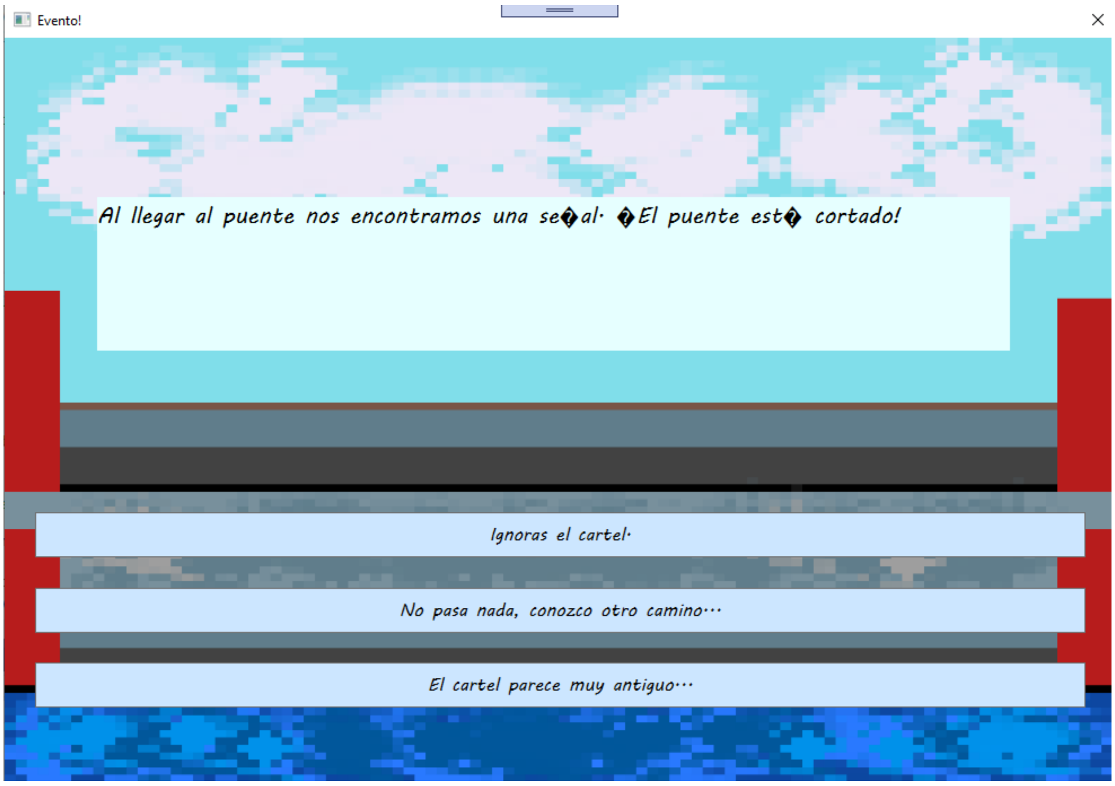
RF3.1 Show Game Window



La Jeny lleva dos meses sin ver a su pareja y no aguanta m�s! Va a emprender el camino hasta su casa para darle un beso por fin.

RF3.5 Event





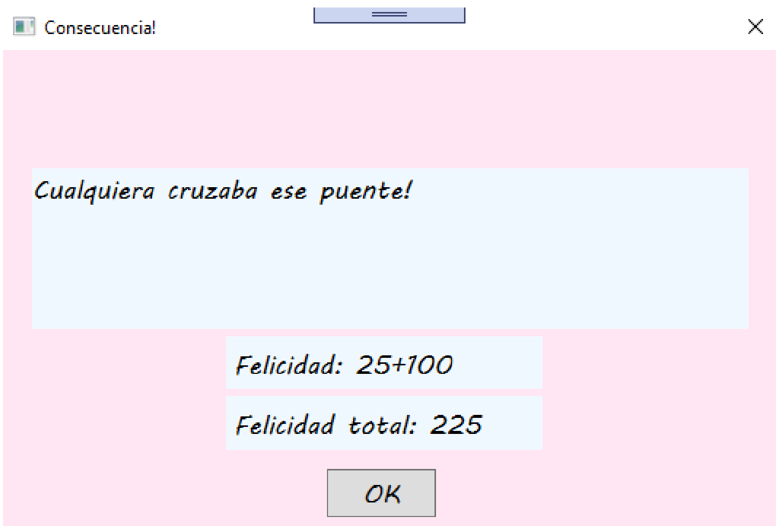
Al llegar al puente nos encontramos una se�al· �El puente est� cortado!

Ignoras el cartel·

No pasa nada, conozco otro camino···

El cartel parece muy antiguo···

RF3.5 Event - answers

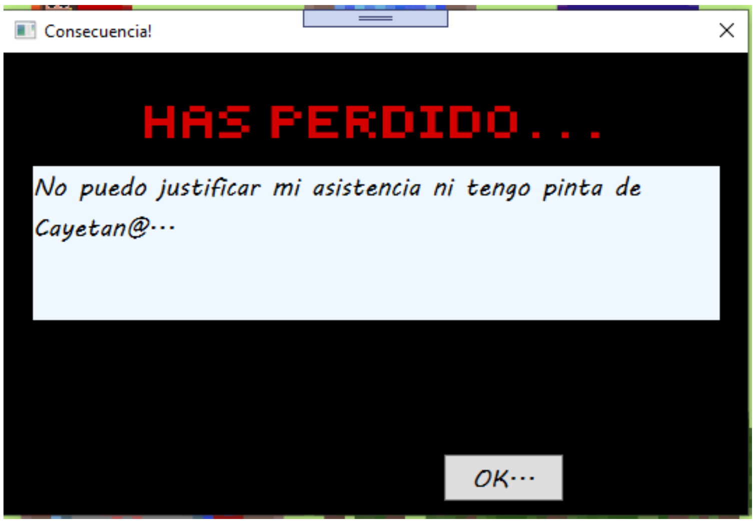

Consecuencia!

Cualquiera cruzaba ese puente!

Felicidad: 25+100

Felicidad total: 225

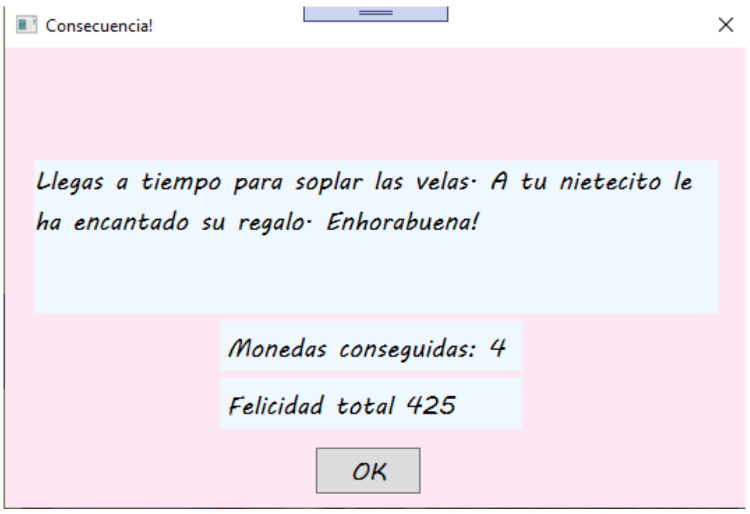
RF3.5 Event- consequence


Consecuencia!

HAS PERDIDO...

No puedo justificar mi asistencia ni tengo pinta de Cayetan@...

RF3.5 Event - game over


Consecuencia!

Llegas a tiempo para soplar las velas. A tu nietecito le ha encantado su regalo. Enhorabuena!

Monedas conseguidas: 4

Felicidad total 425

RF3.6 Arrival at destination

<u>Problems encountered</u>

At first we had a lot of problems with the client-server connection. This connection error was due to a compatibility bug related to Netbeans on Windows, which was solved by running the server on Linux (netbeans java) on a laptop while the client (visual studio c#) was running on another pc on windows.

<u>Comments</u>

Screens
I think the screens are not optimized well. For example, the map window is permanently open and displayed in the background during events, because if I hide or close it, the progress of the buttons (locations) display during the game is lost. I understand that there will be much more efficient ways to handle window switching.
Also, not all "close" button events are implemented.
These two things (and others) are something that I would have liked to improve with more time.

Symbols
When loading the text of characters, events and consequences with http requests using JSON format and receiving the responses from the database, there is a problem with decoding htmlEncode characters which is causing wrong characters to appear in symbols outside the English alphabet.

Learning
During the development of this project I have improved the creation of methods, calls to methods from other classes, pass through parameters and instances. I have learned to connect to a server through http requests the client in visual studio, something that we had not seen in class. .

7.INSTALLATION GUIDE

AquarantineAdventure

Welcome to our application!

As our application is not common, it cannot be installed with an exe file. This is because it requires a server to provide you with data. Here are the steps you have to follow to enjoy it.

1.Download the GraphicAdventureServerCode project (GAServlet server):
https://github.com/mchaconalcaide/GraphicAdventureServerCode.git

Download the zip and unzip the file.

2.Download the GraphicAdventureJAVACode (GraphicAdventureProject data model):
https://github.com/mchaconalcaide/GraphicAdventureJAVACode.git

Download the zip and unzip the file.

Once you have the two projects, open them in your IDE. You will have to add the following libraries to each one:
To the GAServlet:

- JAVA EE Web 7 API Library - javaee-web-api-7.0.jar
- gson-2.8.6.jar
- mysql-connector-java-8.0.20.jar
- GraphicAdventureProject
- JDK 1.8
- Glassfish Server 5.1

To the GraphicAdventureProject:

- mysql-connector-java-8.0.20.jar
- JDK 1.8

For the application to work you need to load the database. You have the MySQL code inside the resources directory in GraphicAdventureProject.

3.Download the AQuarantineAdventure client project:
https://github.com/amartinezperez92/AQuarantineAdventure.git

Open it in Visual Studio and in your nugget package manager install the following packages:

- Newtonsonft.Json

In the solution browser: in Properties - Settings.settings you have to change the ip of the URLs to the ip of the server (localhost if it is the same machine as the client). Also you can change the ip from App.config archive.
Once all the previous steps have been carried out, to make the application work you have to execute the server, selecting the GAServer project and clicking on run. Once an explorer window has been opened, it will be ready to work. Now let's go to the client and we can run our application.

We hope you enjoy it!

A Quarantine Adventure Project Team.