Implementación de juegos de estrategia para ordenador:

L'esprit de Marie Antoinette

Volumen: 1/1

Alumno: Miquel Carbera Vicens

Director: Miquel Barceló Garcia

Departament: ESSI

Data:06/06/2015

DATOS DEL PROYECTO

Titulo del proyecto: **Implementación de juegos de estrategia para ordenador: L'esprit de Marie Antoinette**

Nombre del estudiante: **Miquel Cabrera Vicens**

Titulación: **Ingeniería en informática**

Créditos: **37,5**

Director/Ponente: **Miquel Barceló García**

Departamento: **ESSI**

MIEMBROS DEL TRIBUNAL

Presidente: **Miquel Barceló García**

Vocal: **Jaume Soler Villanueva**

QUALIFICACIÓN

Qualificación numerica:

Qualificación descriptiva:

Fecha:

Agradecimientos

Le agradezco a mi tutor Miquel Barceló, director de este proyecto la oportunidad que me ha dado, para realizar este proyecto, así como el interés mostrado durante toda la realización proyecto. El seguimiento que me ha dedicado y la ayuda que me prestado han sido gran importancia para la realización de este proyecto y para mejorar sus resultado.

También agradezco al estudiante de Estadística de Zaragoza, llamado José Carlos de Diego Guerrero, haber inventado el juego de cartas “L'esprit de Marie Antoinette” sobre el cual trata este proyecto

Finalmente agradecer a mi familia la comprensión mostrada y mis agradecimientos por dejarme hacer este proyecto. Dedicado a mi abuelo, Joaquin y a mi abuela Maria Montserrat.

Índice de contenido

1 INTRODUCCIÓN

1.1 Objetivos del proyecto

El objetivo del proyecto es crear un programa que permita a uno o a varios jugadores jugar al juego “L'esprit de Marie Antoinette” en un ordenador.

Es necesario pensar en cuatro posibles fases principales:

- Implementar una interfaz gráfica del juego agradable e intuitiva para el jugador
- Implementar las reglas del juego (movimientos correctos, movimientos incorrectos)
- Implementar una versión para jugar entre usuarios mediante Internet
- Implementar un programa que pueda jugar contra un adversario artificial con varios niveles de dificultad.

Adicionalmente dicho juego también tendrá las funcionalidades de: registrar un usuario, hacer el login de un usuario, cargar una partida, salvar una partida, ver los récords (clasificación por puntos), ver el ranking (clasificación por numero de partidas ganadas), ver la ayuda con las reglas del juego, deshacer la ultima jugada hecha y pedir que haga una jugada al ordenador hecha por la inteligencia artificial en el turno de un jugador humano a modo de sugerencia o comodín.

1.2 Tecnología

El programa esta siendo implementado con la ultima versión de Java, ya que de este modo usando Java se maximiza la portabilidad con los diferentes sistemas operativos (Windows, Linux, Mac OS X..)

Su desarrollo se realiza mediante la ultima versión de Netbeans y adicionalmente el código del proyecto esta guardado en Google Code (la nube), de este modo cualquier persona desde cualquier ordenador puede obtener el código entero del proyecto tal cual está hasta la fecha, para ser leído y ejecutado, pero sin los permisos de escritura, evitando así que personas que no sean el proyectista puedan borrar parcial o totalmente el código del proyecto. De este modo se facilita el seguimiento del proyecto.

Para acceder al código del juego de cartas de estrategia de mesa "L'esprit de Marie Antoinette", debe instalar Netbeans, abrirlo, hacer clic con el ratón en Team, luego en Checkout y donde pone Repository URL ha de escribir esto:

https://marie-antoinette.googlecode.com/svn/trunk/

Luego se debe hacer clic en Next y en Finish y se importará el proyecto en el Netbeans de su ordenador.

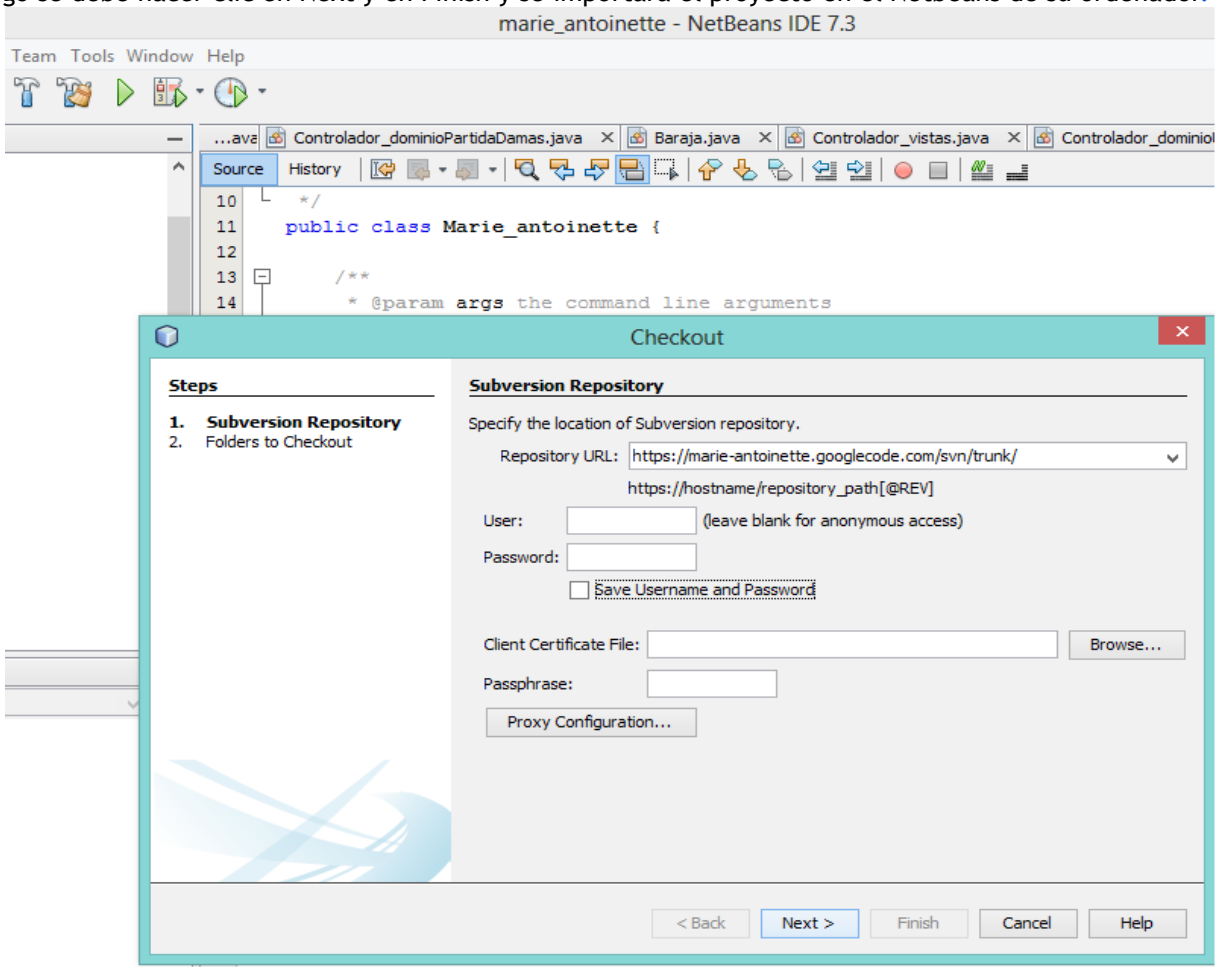


Figura 1. En la Figura 1 se puede ver como sincronizar Eclipse con Google Code para importar el proyecto.

2 EL JUEGO: L'ESPRIT DE MARIE ANTOINETTE.

2.1 Origen del juego

El proyecto consiste en la implementación de un juego de estrategia llamado “L'esprit de Marie Antoinette”. Dicho juego, es un juego de cartas y fue finalista en el concurso de diseño de juegos de mesa Ciudad de Granollers 2009 y segundo en el I Concurso de juegos Print&Play del blog Jugando en pareja.

El juego fue ideado por un estudiante de Estadística de Zaragoza, llamado José Carlos de Diego Guerrero.

En la siguiente pagina web se encuentra mas información sobre el autor y sobre todos sus juegos ideados.

http://www.labsk.net/wkr/autor/

2.2 Descripción del juego: “L'esprit de Marie Antoinette”

El juego de cartas “L'esprit de Marie Antoinette” es para 5 jugadores como máximo.

Dicho juego es bastante difícil de explicar y de entender ya que es un juego que no se parece a ningún juego conocido de cartas. Por ello explicaré (por claridad, y por extensión del informe) una versión simplificada del juego donde se entienda el objetivo de éste así como sus reglas generales, luego se mostrará al final de la página un enlace con las reglas enteras del juego ya que éstas están llenas de detalles que hacen muy difícil la comprensión del juego a la primera.

El juego se basa en hacer perfumes según unas tendencias mostradas al principio de la partida.

Simplificando: Al principio de la partida se muestran las tendencias de los perfumes (tipo de carta) que se tienen que elaborar, dichas tendencias, con sus puntos que se obtienen al conseguir los correspondientes perfumes, son las que marcan el desarrollo de la partida, pues el juego trata de elaborar los perfumes

correspondientes a las tendencias dadas, para así conseguir puntos. El jugador que más puntos consiga, es el que ganará la partida

Ejemplo muy simplificado:

Al principio de la partida se muestran dos tendencias (es decir dos cartas). Una tendencia indica que se debe hacer un perfume con olor a lavanda y lleva una puntuación asociada, que se obtiene si se consigue elaborar el perfume, de 12 puntos. La otra tendencia mostrada indica que debe elaborar un perfume con olor a canela, lleva una puntuación asociada de 9 puntos.

Durante el transcurso de la partida, los jugadores deben conseguir elaborar y presentar un perfume con olor a lavanda y otro con olor a canela para conseguir los 12+9 puntos.

Para elaborar un perfume, se trata de que un jugador consiga hasta 5 cartas con un dibujo de una lavanda y con un dibujo de una canela (cartas llamadas principios olfativos). Los principios olfativos se obtienen robándolos de la baraja en el turno del jugador.

Las reglas del juego están en la siguiente url y tienen una extensión de 10 paginas:

https://www.box.com/s/02h3f9s20p8kzqae9fyw

2.3 Reglas del juego

A continuación se explica como es el juego de que elementos se compone y cómo jugar

2.4 Objetivo

Cada jugador asigna **Principios Olfativos** de su mano entre las diversas **Elaboraciones** de su parfumothèque. Al final de la partida se presentan a **Maria Antonieta**, teniendo en cuenta las **Tendencias** que marcan los **Nobles** parisinos. Aquel que consiga más puntos será el vencedor, una de sus Elaboraciones se convertirá en el "El Espíritu de Maria Antonieta" y el jugador pasará a formar parte de la corte como ¡perfumero real!.

2.5 Componentes

Un mazo de 110 cartas distribuidas de la siguiente manera:

- Principios Olfativos:
    - Frutas (10 cartas)
    - Flores (10 cartas)
    - Hojas (10 cartas)
    - Semillas (8 cartas)
    - Cortezas (8 cartas)
    - Raíces (8 cartas)
    - Almizcles (8 cartas) (del tipo fijador)
    - Resinas (10 cartas) (del tipo fijador)
- Reserva (4 cartas)
- Tendencias (10 cartas)
- Nobles (20 cartas) (10 nobles repetidos dos veces)
- Frascos de Perfume (4 cartas) (del tipo comodín)

2.6 Principios Olfativos

Los Principios Olfativos son los grupos de Esencias con los que se crean las Elaboraciones.

- Hay 72 cartas divididas 8 tipos distintos.
- Los 18 fijadores (10 Resinas y 8 Almizcles) tienen el icono  en dos de sus esquinas.
- El número que aparece en las otras dos esquinas indica el número de cartas de cada tipo en el mazo.

2.7 Frascos de Perfume

Los 4 Frascos de Perfume se pueden añadir al mazo de Encarte opcionalmente.

A efectos de juego cuentan como un Principio Olfativo. Esto significa, por ejemplo, que en una parfumothèque ocupan un espacio en la Elaboración, y que si te pillan alguno al final de la partida en la mano te dan -1 punto.

Se usan como un comodín, así que sirven como cualquiera de los ocho Principios Olfativos (incluidos por tanto, fijadores).

La gracia consiste en que a la hora de bajarlos a tu parfumothèque se pueden bajar de golpe junto con otro Principio Olfativo (por ejemplo, 1 frasco de perfume y 2 semillas), y luego usarlos como otro distinto al realizar las ofertas (por ejemplo, 1 rosa y 2 semillas).

En la Puntuación, a la hora de comparar si tienes cartas de los ocho grupos (para el bono de 10 puntos) no cuentan..

2.8 Elaboraciones

Las Elaboraciones son los perfumes que crea cada perfumero, siguiendo estas reglas:

- Una Elaboración debe tener 1 único fijador.
- Una Elaboración debe tener como máximo 5 Principios Olfativos (incluyendo el fijador).
- No hay límite de Elaboraciones en la parfumothèque.

2.9 Tendencias

Hay 10 distintas. Los iconos son los Principios Olfativos que demanda en **cualquier combinación** (incluso 0, salvo fijadores que por obligación es 1) en una de tus Elaboraciones válidas. Si aparecen dos fijadores significa que puedes elegir uno de los dos en tu Elaboración. La nariz es un grupo cualquiera a tu elección.

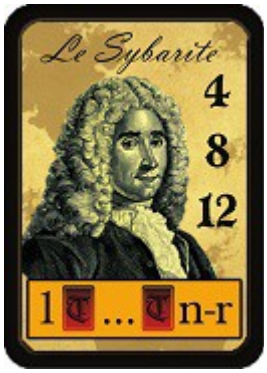
En la carta de ejemplo, podrías elegir uno (1) de los dos fijadores que aparecen (resina o materia animal) y cualquier combinación de cartas de Flores y/o Frutas (0-4, 4-0, 1-3, 3-1 o 2-2). Eso sí, sumando todas incluyendo el fijador, nunca podrás superar 5 cartas en tu Elaboración.

Todas las cartas del mismo Principio Olfativo.

De 2 a 4 cartas de Principios Olfativos diferentes.

2.10 Nobles

Existen 10 Nobles distintos, repetidos 2 veces. Son sucesiones numéricas (x/x/x) que **valoran las Elaboraciones** de 3, 4 y 5 cartas, respectivamente. Los puntos obtenidos al ofertar Elaboraciones lo determina cada Tendencia junto el Noble asignado en ella.



EFECTOS: Cuando se pone un Noble en juego desde la mano dispara un efecto especial, que **siempre** ha de realizarse, salvo que sea imposible, en cuyo caso se ignora:

1. **Le Sybarite (el Sibarita):** Reorganiza todas Tendencias en juego, excepto las reservadas y la suya, a su elección. Los Nobles mantienen su posición.

2. **Le Fou (el Loco):** Añade una nueva Tendencia en juego al final de la línea (a la derecha del todo).

3. **L'Extroverti (el Extrovertido):** Descarta cartas del mazo de Encarte hasta que aparezca un Noble que no este repetido en la línea de Tendencias. Añádelo en la primera posición libre de la línea de Tendencias; si no hay ninguna libre, por ser el último Noble en asignarse, simplemente ignora el efecto.

4. **L'Introverti (el Introvertido):** Elimina un Noble (distinto de L'Introverti) en juego. Si solo queda un único Noble en juego ignora el efecto (dicho de otra forma, siempre tiene que haber al menos un Noble en juego).

5.L'Expérimenté (el Experimentado): Busca 1 Principio Olfativo en el mazo de Encarte, enséñaselo a los demás jugadores y quédatelo en la mano. Después baraja el mazo de Encarte.

6.Le Sophistiqué (el Sofisticado): Este Noble se puede jugar en cualquier Tendencia de la línea, no en la primera libre empezando por la izquierda.

7.Le Diable (el Diablo): Elimina una Tendencia en juego que no tenga un Noble asignado a tu elección. En el caso de que tuviera una reserva la carta de reserva vuelve a la parfumthèque de su propietario.

8. Le Tenace (el Tenaz): Todos los perfumeros adversarios roban 2 cartas del mazo de Robo.

9. Le Romantique (el Romántico): Una vez colocado sobre una Tendencia válida, intercambia Le Romantique por otro Noble en juego.

10. Le Mûr (el Maduro): Sustituye Le Mûr por otro Noble en juego. El otro noble se elimina de juego.

2.11 Setup

Setup inicial para 2 jugadores

1. El perfumero que mejor huela será el repartidor y jugador inicial. Si no os ponéis de acuerdo sugiero que sea entonces el jugador más joven quien reparta.

2. Baraja las 10 Tendencias por separado. Coloca tantas cartas como el **número de jugadores + 2** al azar bocarriba, de izquierda a derecha, formando una fila en el centro de la mesa. Las sobrantes se dejan al final de la línea boca-abajo.

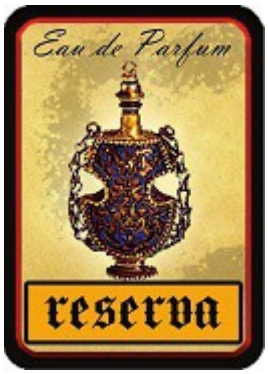
3. Si solo juegan dos jugadores elimina los (6) Nobles de **L'Introverti, L'Extroverti y Le Sophistiqué.**

   Para recordarlo en su carta aparece el icono  .

4. Baraja las cartas de Noble, y coloca una al azar en la **primera Tendencia** libre, empezando por la izquierda. No se aplica el efecto del Noble inicial.

5. Las 19 cartas de Nobles restantes se barajan junto las 80 cartas de Principios Olfativos. Se reparten **9 cartas** en secreto, de 3 en 3, a cada perfumero. Las sobrantes pasan a formar el mazo de Encarte.

6. Da una carta de Reserva a cada **parfumothèque**.



2.12 Mecánica de juego

Se juega en sentido horario. Cada perfumero en su turno debe realizar dos fases en este orden:

1. **Encarte**
2. **Acción** - 2 acciones distintas a elegir de entre estas 6:

    a. Añadir Principios Olfativos a tu parfumothèque, ó

    b. Mover 1 o 2 Principios Olfativos entre Elaboraciones de tu parfumothèque, ó

    c. Poner en juego un Noble, ó

    d. Descartarte una carta de tu mano o de tu parfumothèque, ó

    e. Robar 1 carta adicional del mazo de Encarte.

    f. Reservar una Tendencia.

Una vez finalice, el turno pasará al jugador sentado a su izquierda.

Encarte

- Roba:

- 1 carta del mazo de Encarte; ó
1. 1 o más cartas de Principios Olfativos de la Pila de Descartes, siempre y cuando sean las últimas descartadas, estén consecutivas y pertenezcan al mismo tipo, en cuyo caso, **deben añadirse, junto la posibilidad de agregar cartas del mismo tipo de la mano,** utilizando una de sus dos acciones como una acción a) en alguna Elaboración ya comenzada o deben usarse para crear una nueva.

Acción

- Debe hacer obligatoriamente 2 acciones distintas de estas 6 disponibles:

a. AÑADIR PRINCIPIOS OLFATIVOS

El perfumero puede jugar cualquier número de Principios Olfativos de su mano a su parfumothèque siempre y cuando pertenezcan al mismo grupo.

- Pueden jugarse sobre Elaboraciones ya creadas, o comenzar nuevas, o una combinación de ambas.
- Cada carta puede colocarse en Elaboraciones diferentes.

b. MOVER PRINCIPIOS OLFATIVOS

2. El perfumero puede mover **1 o 2** cartas de Principios Olfativos de su *parfumothèque*:

- Debe moverse a una Elaboración distinta de la origen.
- Si mueves dos cartas no es necesario tengan como objetivo la misma Elaboración.
- Si haces un segundo movimiento no puedes deshacer lo hecho en el primero.
- Puedes crear nuevas Elaboraciones con los Principios Olfativos movidos.

c. JUGAR UN NOBLE

Pones 1 Noble de tu mano en juego. Para ello:

- Ese Noble (con el mismo nombre) no debe estar ya en juego.
- Debe quedar una Tendencia libre donde asignarlo.
3. Si hay varias Tendencias libres, se asigna siempre a la primera de la fila, comenzando por la izquierda, salvo que sea **Le Sophistiqué** (que elige cualquiera).

d. DESCARTAR UNA CARTA

Descarta boca-arriba a la Pila de Descartes:

- 1 carta de la mano (los Nobles se eliminan fuera de juego); ó
4. 1 Principio Olfativo de una de sus Elaboraciones de tu **parfumothèque.**

Nótese que las cartas descartadas se podrán robar en las fases de Encarte de los jugadores.

e. ROBAR UNA CARTA

Roba 1 carta adicional del mazo de Encarte, nunca de la Pila de Descartes.

f. RESERVAR UNA TENDENCIA

Esta acción sirve para reservar una Tendencia. De esta forma un jugador se asegura que será el primer jugador en ofertar Elaboraciones en ella en la fase de Recuento. Puede:

- Poner su carta de Reserva desde su parfumothèque sobre una Tendencia no reservada por otro jugador; ó
- Devolver su carta de Reserva desde una Tendencia reservada por él a su *parfumothèque*.

2.13 Muerte súbita

Cuando todas las Tendencias tengan un Noble asignado se pasa a muerte súbita. A partir de entonces, ya no se pueden jugar Nobles desde la mano (acción c); y además si se roba un Noble del mazo de Encarte se elimina de inmediato y se roba otra carta en su lugar.

2.14 Fin de partida

La partida finaliza de dos maneras:

- En la muerte súbita si uno de los perfumeros se desprende de todas sus cartas de de su mano.
- O cuando un perfumero no pueda robar del mazo de Encarte, ya sea por la fase 1 (encarte) o por la acción e).

Quien finaliza el juego es el jugador activo.

Entonces cada perfumero, salvo el jugador activo, puede realizar una única acción “in extremis” que no sea la c).

2.15 Lo que puntúa

Los perfumeros solo puntúan por las Elaboraciones en su *parfumothèque*. Se calculan, revisando lo siguiente, en este estricto orden:

5. Cada Principio Olfativo en la mano resta 1 punto, y cada Noble 2 puntos. Las cartas se descartan.
6. Si una Elaboración no tiene **entre 3 y 5 Principios Olfativos (inclusive)** resta 3 puntos y todas sus cartas se descartan.
7. Si una Elaboración no tiene un **único fijador** resta **3** puntos y todas sus cartas se descartan.
8. Si entre todas las Elaboraciones válidas (las que quedan) un perfumero ha utilizado los **ocho grupos de Principios Olfativos** obtiene un premio de **+10** puntos; el acorde de su *parfumothèque* ronda la perfección. (Nota: El acorde es la delicada composición de esencias en determinadas proporciones, que el perfumista logra al crear una nueva fragancia).
9. Ofertas: Las modas activas (parejas de Tendencia + Noble) determinan el valor (positivo) que se puede obtener por una Elaboración válida. Mira “Como se puntúa” más abajo.
10. Finalmente, todas las Elaboraciones válidas que no se consigan ofertar dan 0 puntos.

Elaboración no válida

(mínimo 3 cartas)

Elaboración no válida - falta fijador

Elaboración válida.

2.16 Como se puntúa

Solo se puede ofertar una única Elaboración por Tendencia y jugador.

¿Quién oferta? Primero lo hace quien tenga una de Reserva sobre esa Tendencia, después va el jugador activo (en la primera oferta es quien finalizo la partida), y luego se prosigue con los restantes jugadores, siguiendo el sentido de las agujas del reloj, sentados tras el jugador activo. Se continúa de esta forma hasta que todos los perfumeros hayan tenido opción de presentar una (y solo una) de sus Elaboraciones en la Tendencia activa.

¿En qué orden se oferta? En estricto orden de Tendencias, de izquierda a derecha. Se resuelven una a una. En cada Tendencia cada jugador tiene oportunidad de ofertar una única vez.

¿Cómo se oferta? Un jugador:

- Si oferta, recibe los puntos del Noble según la Elaboración que presente sea de 3, 4, o 5 Principios Olfativos; no pudiendo otro jugador posterior presentar una Elaboración con el mismo número de Principios Olfativos a este Noble. Las Elaboraciones ofertadas se quedan en la parfumothèque boca-abajo.
- Puede no presentar ninguna: por no tener o porque prefiere mantener la Elaboración para un Noble posterior que de más puntos; pero ya no podrá presentar ninguna Elaboración a este Noble.

Luego se prosigue con la segunda Tendencia. El jugador activo será el jugador sentado a la izquierda del jugador activo anterior. El primer jugador en tener opción de ofertar será quien tenga una carta de Reserva, después el nuevo jugador activo y luego los restantes jugadores en orden del sentido de las agujas del reloj tras el jugador activo. Se prosigue de esta forma hasta completar la puntuación de todas las Tendencias sobre la mesa.

Ejemplo: *Juegan 2 jugadores. Pedro y María. A continuación se muestran las tendencias activas .*

Parfumthèque de María:

Parfumthèque de Pedro:

María es quien cierra quedándose sin cartas en la mano. Pedro, como acción “in extremis”, decide descartarse de un Principio Olfativo de su mano. Como le quedan otros dos en la mano, pierde 2 puntos.

En ambas parfumthéques todas las Elaboraciones tienen entre 3 y 5 Principios Olfativos; ninguno de los dos se lleva negativos. Igualmente, todas las Elaboraciones tienen 1 único fijador; nadie se lleva negativos. Así mismo, ninguno tiene los 8 Principios Olfativos en sus Elaboraciones válidas, por tanto, nadie obtiene +10 puntos.

Pasemos a la oferta. Se empieza a ofertar siempre por la primera Tendencia empezando por la izquierda. El jugador activo (quien cerro) es María. María decide ofertar su primera Elaboración para así obtener 8 puntos. Pedro que quería ofertar su primera Elaboración no puede porque tiene el mismo número de Principios Olfativos (cartas) que la de María. Sus otras dos Elaboraciones tampoco le sirven, por tanto no puede ofertar en esta primera Elaboración.

Se oferta en la segunda Tendencia. El primero en ofertar en esta ocasión sería Pedro (el jugador sentado a la izquierda del anterior jugador activo: María), pero María tiene una reserva en esa

Tendencia. María oferta su segunda Elaboración para así obtener 3 puntos. Nuevamente impide que Pedro pueda vender su segunda Elaboración.

Se oferta en la tercera Tendencia. Ahora el primero en ofertar sería María (el jugador sentado a la izquierda del anterior jugador activo: Pedro). María oferta su tercera Tendencia y obtiene otros 3 puntos, y además impide de nuevo que Pedro pueda vender su segunda Elaboración.

En la última Tendencia Pedro es el jugador activo. Tras el fiasco de las anteriores ofertas él puede vender su tercera Tendencia y obtiene por ella 8 puntos. María ya no puede ofertar porque no le quedan Elaboraciones en su **parfumthèque.**

2.17 Vencedor

Quien haya conseguido más puntos será el vencedor. Si hay empate se desempata a favor de:

1º) Quien tenga cartas de los 8 Principios Olfativos en su **parfumothèque.**

2º) Quien tenga más Elaboraciones válidas en su **parfumothèque.**

3º) Quien tenga menos cartas en la mano al final de la partida.

4º) Quien haya usado menos Principios Olfativos en sus Elaboraciones.

Ejemplo:

La puntuación final en nuestro ejemplo queda como sigue:

María: 8 + 3 + 3 = 14 puntos Pedro: -2 + 8 = 6 puntos

María es la vencedora.

3 ESTUDIO PREVIO Y ANÁLISIS

3.1 Análisis de requerimientos

3.1.1 Requerimientos funcionales

1. Habrá la posibilidad de que jugadores situados en ordenadores distintos jueguen una partida mediante Internet.
2. El juego tendrá una interfaz gráfica que mostrará al jugador la información necesaria para poder jugar. Como el juego es bastante complicado de jugar el jugador será guiado continuamente para saber que puede hacer.
3. El juego tendrá varias inteligencias de distinto nivel de dificultad contra las cuales el jugador humano podrá jugar. Las inteligencias artificiales podrán jugar contra ellas mismas y contra oponentes humanos.
4. El jugo será multiplataforma y se podrá ejecutar tanto en Windows como en iMac OS como en GNU/Linux
5. Habrá la posibilidad de poder guardar y cargar partidas (siempre claro,que no sean partidas en red)
6. En el juego se podrá deshacer el ultimo movimiento hecho.
7. En el juego se podrá visualizar la lista de movimientos realizados durante la partida.
8. El juego permitirá jugar a L'esprit de Marie Antoinnette aceptando solo aquellas jugadas y movimientos validos.

3.1.2 Requerimientos no funcionales

1. La interfaz gráfica del juego será intuitiva, y ayudará al jugador a entender que es lo que está pasando
2. en todo momento, así como les formas que tiene el jugador de interaccionar con el juego.

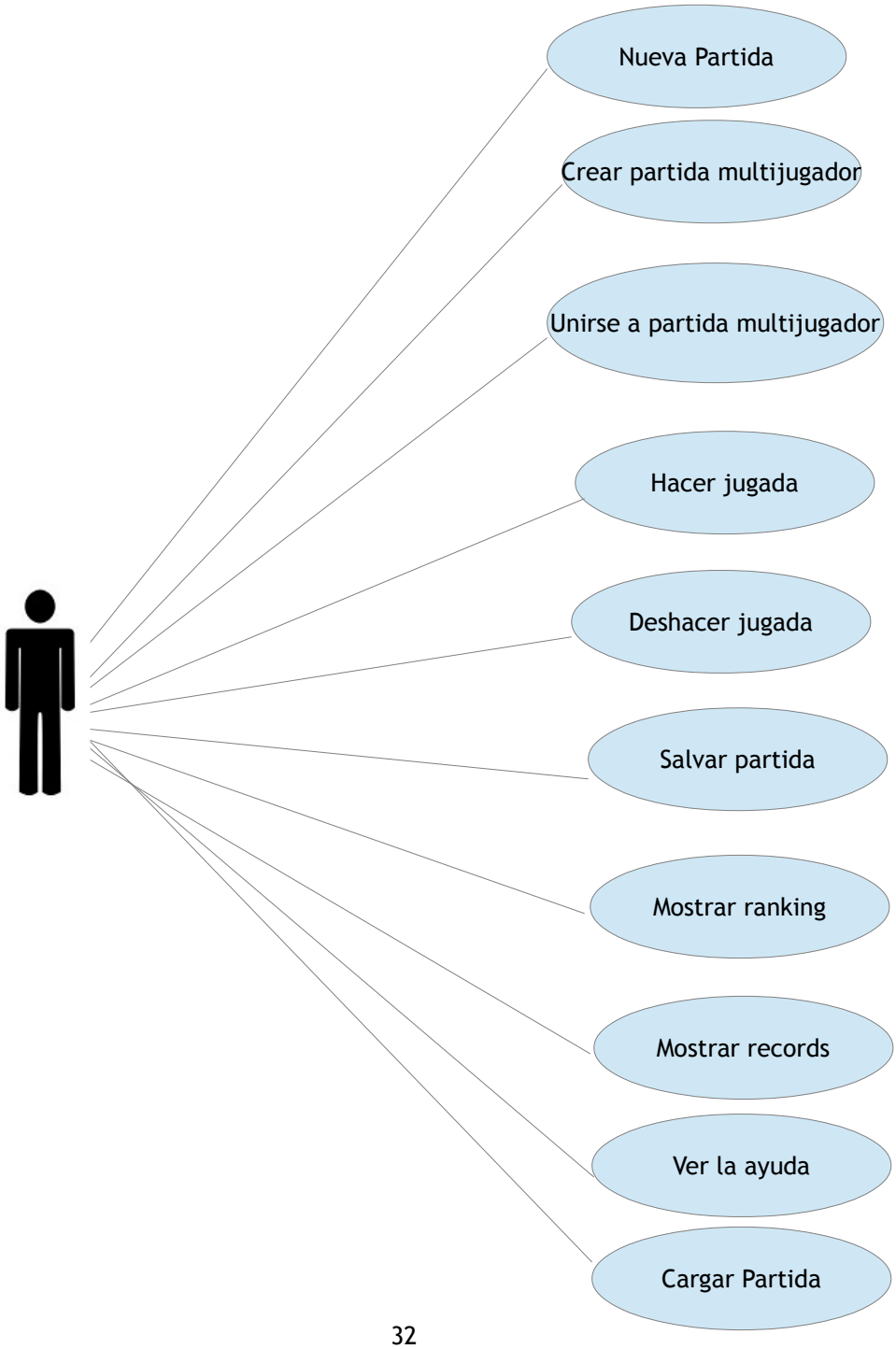
3. Les inteligencias artificiales tendrán un nivel de competencia suficientemente alto como pera presentar un reto al jugador.
4. La inteligencia artificial del juego tardará un tiempo razonable para efectuar y calcular el siguiente movimiento.

4 ESPECIFICACIÓN

4.1 Actores del sistema

El único actor del sistema es el usuario o jugador del juego. Dicho jugador podrá jugar contra rivales humanos (en local o en red) o contra la Inteligencia Artificial.

4.2 Casos de uso



4.2.1 Nueva partida local

Caso de uso: Nueva partida	
Actores	Usuario
Descripción	Inicia una nueva partida con las opciones escogidas por el usuario
Precondiciones	-
Postcondiciones	Se muestra el estado inicial de la nueva partida

Usuario	Sistema
1- Selecciona la opción Nueva Partida.	
	2- Muestra el menú con los tipos de partida disponibles
3- Selecciona la opción Partida local	
	4- Muestra el menú con las opciones para escoger como sera la nueva Partida local
5- Selecciona las opciones que se desee, pulse en Aceptar	
	6- Se inicia la nueva partida con los jugadores que quiere el usuario

Casos alternativos
6-Si se selecciona Hacer login se obliga a todos los jugadores ha hacer el login y sino esta registrado se podrá crear el usuario en el sistema. Luego se inicia la nueva partida. 6-Si se selecciona que un jugador es de tipo en red, la partida pasa automáticamente a ser en red y el juego espera a que dicho jugador se conecte antes de empezar la partida. Luego se inicia la nueva partida.

4.2.2 Crear partida en red

Caso de uso: Crear partida en red	
Actores	Usuario
Descripción	Crea una nueva partida a la cual se puede conectar otra jugador mediante Internet
Precondiciones	-
Postcondiciones	Hay un socket abierto esperando una conexión remota

Usuario	Sistema local	Sistema en red
1- Selecciona la opción Archivo->Nueva Partida.		
	2- Muestra el menú con los tipos de partida disponibles	
3- Selecciona la opción partida en red		
	4- Muestra el menú con los tipos de partida en red.	
5- Selecciona la opción de crear partida en red		
	5-Muestra la pantalla con las opciones de partida en red.	
6- Selecciona el numero de jugadores y si quieres el login o no y pulsa en aceptar.		
	6- El sistema local espera que se conecten el numero de jugadores seleccionados.	
		7- Otro sistema se conecta mediante la red al sistema local host.
	8-Se inicia la nueva partida con los jugadores que quiere el usuario	

Casos alternativos
6-Si se selecciona Hacer login se obliga a todos los jugadores ha hacer el login y sino esta registrado se podrá crear el usuario en el sistema. Luego el sistema espera a que se conecten los jugadores.

4.2.3 Unirse a una partida en red

Caso de uso: Unirse a una partida en red	
Actores	Usuario
Descripción	Se une a la partida creada por otro jugador del ordenador remoto host.
Precondiciones	
Postcondiciones	Conexión establecida. Partida iniciada.

Usuario	Sistema local	Sistema en red
1- Selecciona la opción Nueva Partida.		
	2- Muestra el menú con los tipos de partida disponibles	
3- Selecciona la opción partida en red		
	4- Muestra el menú con los tipos de partida en red.	
5- Selecciona la opción de unirse a partida en red		
	5-Muestra la pantalla de unirse partida en red.	
6- Introduzca la ip del host y si quieres el login o no y pulsa en aceptar.		
	6- El sistema local se conecta con el ordenador en red que hace de host.	
		7- El host o servidor inicia la partida.
	8- Se inicia la partida en el ordenador local cuando todos los jugadores en red se han conectado al servidor.	

Casos alternativos
6-Si se selecciona Hacer login se obliga a todos los jugadores ha hacer el login y sino esta registrado se podrá crear el usuario en el sistema. Luego el sistema se conecta al servidor.

4.2.4 Hacer movimiento

Caso de uso: Hacer jugada	
Actores	Usuario
Descripción	Se efectúa un jugada valida(compuesta por 3 acciones)
Precondiciones	-
Postcondiciones	Se muestra y se guarda la jugada y se pasa el turno al siguiente jugador

Usuario	Sistema
1- El usuario se recomienda que lea la guía, piense la jugada que va a hacer y luego pulse encima la carta a mover.	
	2- El sistema procesa el primer clic, lo guarda para el retroceder jugada y muestra donde puede pulsar el jugador para terminar la jugada en la guía.
3- Se recomienda que el usuario lea la guía y termine el movimiento que empezó seleccionando la posición donde ira parar la carta.	
	4- Se procesa el segundo clic terminando el movimiento y se guarda para el retroceder jugada. Si es el ultimo movimiento de la jugada se pasa a la siguiente sino se continua la jugada. Se muestra el movimiento en el historial, y se actualiza la guía.

Casos alternativos
2- Si el primer clic no tiene sentido se indica en la guía y no se guarda. Se puede volver ha hacer un primer clic para empezar el movimiento valido.
4- Se muestra el movimiento o jugada hecha en el historial y se muestran las opciones que tienes en la nueva jugada. Si se han terminando las 3 jugadas por turno se pasa el turno ha otro jugador.

4.2.5 Deshacer movimiento

Caso de uso: Deshacer jugada	
Actores	Usuario
Descripción	Deshace la ultima acción
Precondiciones	Que exista un movimiento anterior
Postcondiciones	La partida deshace el último movimiento

Usuario	Sistema
1-Una vez empezada una partida, Pulse encima la opción Acción-> Deshacer Jugada	
	2- Retrocede el ultimo movimiento y muestra la mesa de la partida según estaba anteriormente

4.2.6 Salvar partida

Caso de uso: Salvar partida	
Actores	Usuario
Descripción	Guarda el estado completo del juego
Precondiciones	La partida no puede ser en red, debe salvarse al principio del turno de un jugador.
Postcondiciones	Se ha creado un archivo que contiene el estado completo de la partida.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Archivo->Salvar Partida	
	2- Muestra la pantalla pequeña de salvar partida
2- El usuario debe introducir el nombre de la partida a salvar y pulsar en Salvar	
	3- El sistema salva la partida para los jugadores que participan en la partida en curso para

4.2.7 Borrar partida

Caso de uso: Borra partida	
Actores	Usuario
Descripción	Borra una partida previamente salvada
Precondiciones	La partida debe a borrar debe de haber sido guardada previamente y debe de pertenecer a una partida anterior del jugador que la quiere borrar.
Postcondiciones	Se ha borrado la partida salvada.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Archivo->Salvar Partida	
	2- Muestra la pantalla pequeña de salvar partida
2- El usuario debe seleccionar la partida de la lista de partidas salvadas previamente que quiere borrar y pulsar en borrar	
	3- El sistema borra la partida seleccionada por el usuario

4.2.8 Cargar partida

Caso de uso: Cargar partida	
Actores	Usuario
Descripción	Se carga el estado del juego desde un archivo
Precondiciones	La partida ha sido salvada anteriormente.
Postcondiciones	El estado de la partida es el mismo que cuando se guardó la partida.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Archivo->Cargar Partida	
	2- Muestra la pantalla pequeña de cargar partida, con las partidas salvadas del jugador previamente.
2- El usuario debe seleccionar el nombre de la partida a cargar y pulsar en cargar.	
	3- El sistema carga la partida conlos jugadores que participaron y en el mismo estado en que fue guardada.

4.2.9 Abandonar partida

Caso de uso: Borra partida	
Actores	Usuario
Descripción	Abandona una partida previamente empezada
Precondiciones	-
Postcondiciones	Se ha abandonado la partida empezada y se vuelve al menú previo de crear partida.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Archivo->Abandonar partida	
	2- Termina la partida anterior y muestra la pantalla anterior de crear partida.

4.2.10 Reiniciar partida

Caso de uso: Nueva partida	
Actores	Usuario
Descripción	Reinicia una nueva partida con las opciones escogidas por el usuario
Precondiciones	-
Postcondiciones	Se reinicia la partida anterior.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Archivo->Reiniciar partida	
	2- Reinicia la partida anteriormente empezada, con su estado inicial igual.

4.2.11 Mostrar ranking

Caso de uso: Mostrar ranking de ganadores	
Actores	Usuario
Descripción	Muestra el ranking de los jugadores según el numero de partidas ganadas
Precondiciones	-
Postcondiciones	El ranking es visible

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Estadisticas->Ver ranking de ganadadores	
	2- Muestra la pantalla pequeña un listado ordenado de los jugadores que han ganado más partidas.

4.2.12 Mostrar récords

Caso de uso	Mostrar récords
Actores	Usuario
Descripción	Muestra los récords por puntos obtenidos en una partida de todos los usuarios.
Precondiciones	-
Postcondiciones	Los récords son visibles

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Estadisticas->Ver récord de puntos.	
	2- Muestra la pantalla pequeña un listado ordenado de los jugadores con sus récords puntuación de anteriores partidas.

4.2.13 Mostrar ayuda

Caso de uso	Mostrar ayuda
Actores	Usuario
Descripción	Se muestra la ayuda del juego
Precondiciones	-
Postcondiciones	La ayuda del juego es visible

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre Ver ayuda-> Ayuda.	
	2- Muestra la pantalla pequeña con toda la normativa y explicación del juego.

4.2.14 Mostrar guía

Caso de uso	Mostrar ayuda
Actores	Usuario
Descripción	Se muestra la guía del juego
Precondiciones	-
Postcondiciones	La guía del juego se visualiza con consejos actuales.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre <-G	
	2- Aparece una pantalla pequeña dinamicamente con todas los consejos necesarios para continuar jugando según las normas del juego.

4.2.15 Mostrar historial de jugadas

Caso de uso	Mostrar ayuda
Actores	Usuario
Descripción	Se muestra el historial de jugadas del juego
Precondiciones	-
Postcondiciones	La pantalla de historial de jugadas se visualiza.

Usuario	Sistema
1- Una vez se ha empezado una partida, se pulsa sobre <-H	
	2- Aparece una pantalla pequeña dinamicamente con todas los jugadas hechas de todos los jugadores anteriormente.

4.2.16 Mostrar cartas de otros jugadores

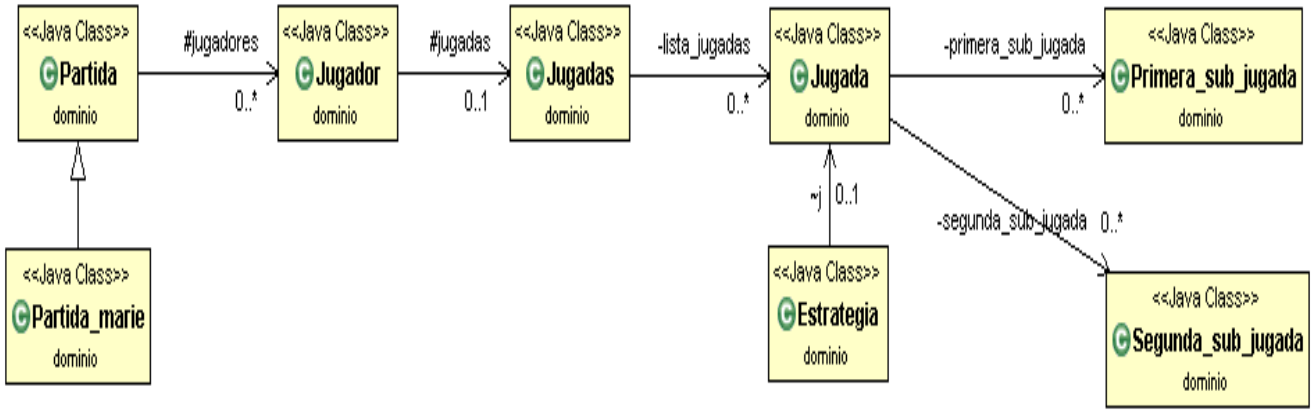
Caso de uso	Mostrar ayuda
Actores	Usuario
Descripción	Sirve para saber cuantas cartas tienen los demás jugadores en tiempo real.
Precondiciones	-
Postcondiciones	Se ven el numero de cartas que tienen el otro jugador.

Usuario	Sistema
1- Una vez ha empezado una partida, se clica en la parte superior derecha de la pantalla donde pone cartas de otros jugadores y encima del jugador que se quiera ver las caras de la mano que tiene.	
	2- Se muestran las cartas de la mano del jugador seleccionado volteadas.

4.3 Modelo conceptual

Dado el tamaño de el modelo conceptual de todo el sistema completo, el modelo conceptual es dividido en diversas paginas. De este modo se facilita el visionado y la compresión de el modelo conceptual. Se empieza pues mostrando el modelo conceptual básico de una partida para ver de que objetos se compone y su relación entre ellos. Las clases se muestran en un principio sin sus variables ni funciones para facilitar la comprensión del diagrama entero en su conjunto. Luego individualmente se muestra en el anexo objecto por objecto tanto sus atributos como sus funciones. Finalmente comentar que no he puesto en el modelo conceptual todas las clases (como por ejemplo clases comparators utilizadas para la ordenación de alguna de las clases) para facilitar la comprensión al lector de como es el juego básicamente.

4.3.1 Diagrama principal del dominio

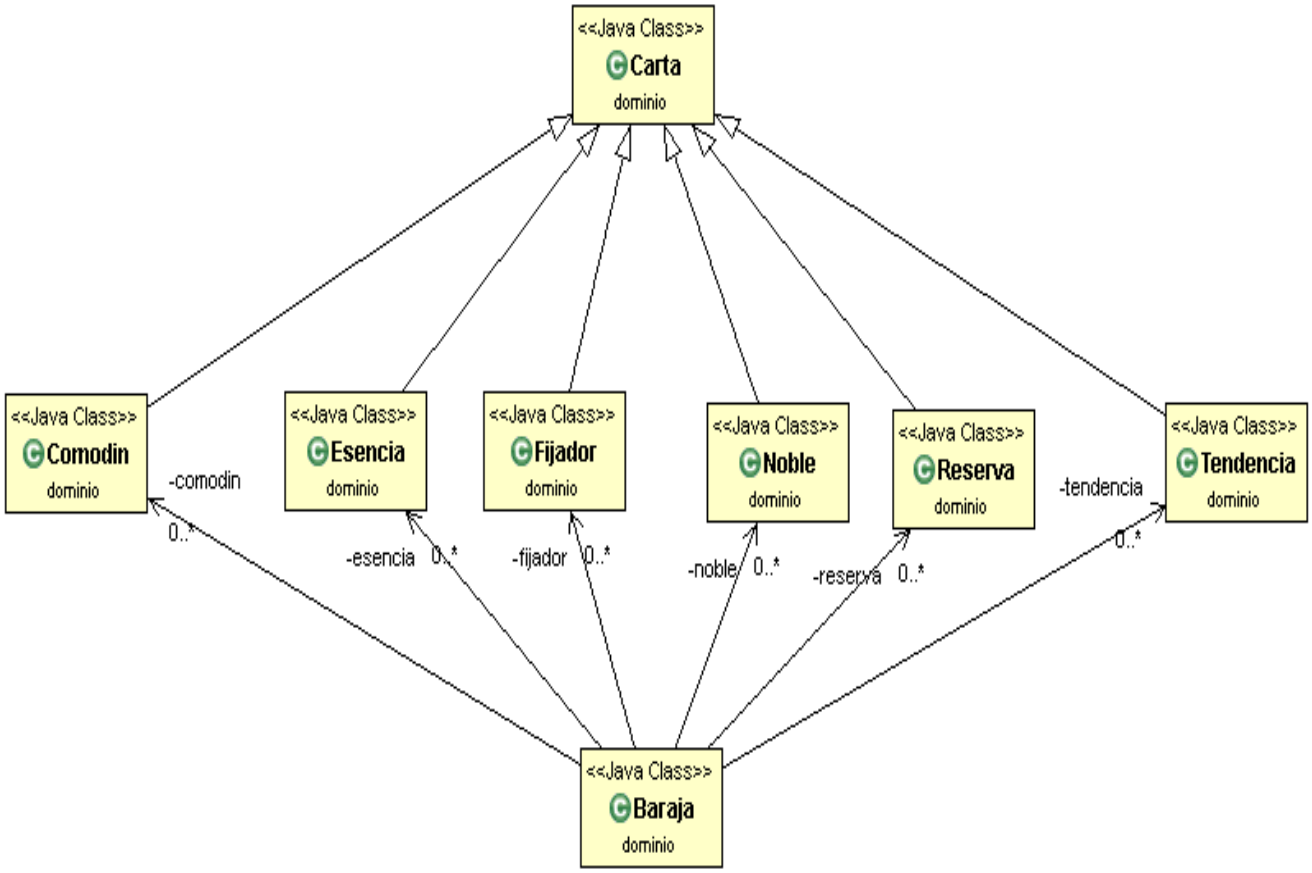


En este diagrama se puede observar como una partida se compone por varios jugadores (la estructura básica). Cada jugador se compone de Jugadas y cada Jugada se compone de dos tipos de tipos distintos de jugadas de acuerdo con las normas. Cada turno se compone de 1 primera jugada (con sus posibles movimientos) y luego se procede a hacer la segunda jugada y luego la tercera. Tanto la segunda jugada como la tercera sus posibles tipos de movimientos son iguales, por eso no hay una tercera subjugada porque es idéntica a la segunda.

Para finalizar la inteligencia artificial devuelve una jugada que contiene la jugada pensada por la inteligencia artificial. En el anterior diagrama se ha obviado las cartas y sus tipos para que resultase más claro. A continuación se explican.

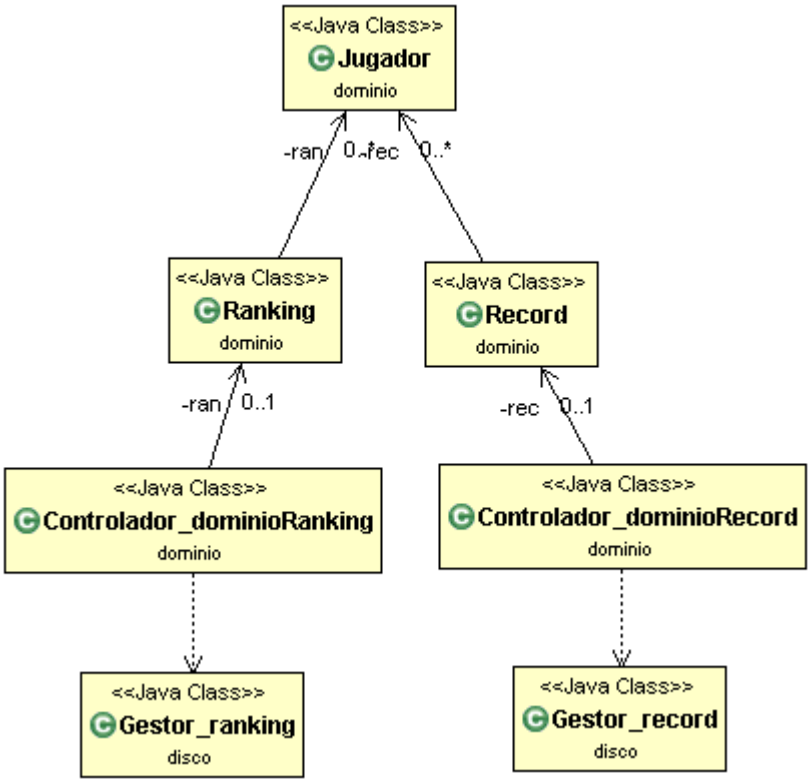
4.3.2 Diagrama cartas

Se muestra a continuación el diagrama de clase de una carta. Vemos que una carta puede ser un comodín una esencia un fijador un noble una reserva o una tendencia. Y que además dichos tipos de cartas pueden formar parte de otros objetos como por ejemplo de la baraja principal del juego donde están todas las cartas.



4.3.3 Diagrama ranking y récord

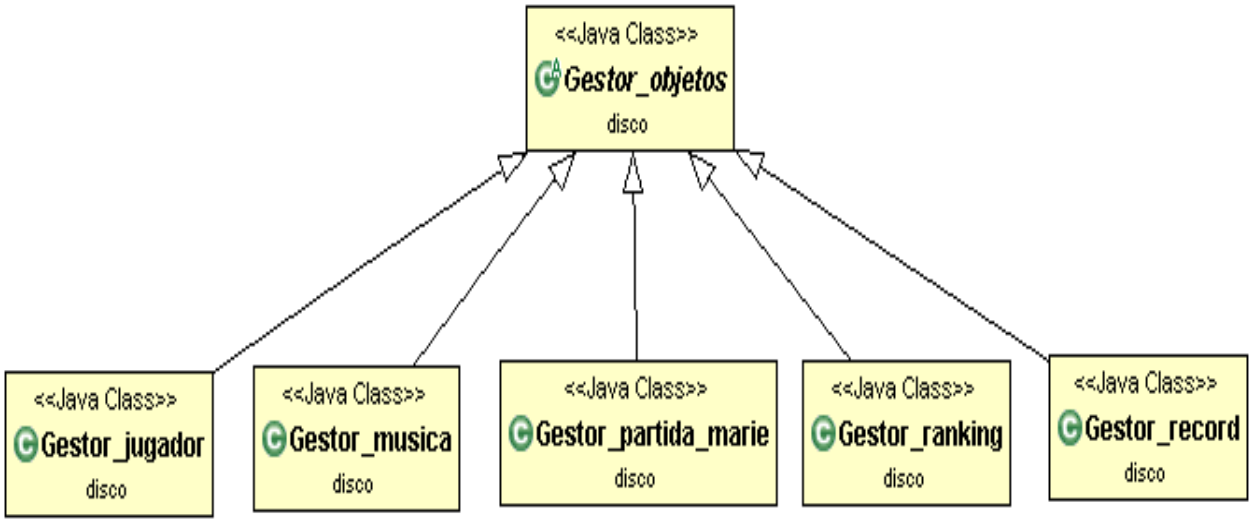
Explicamos que el ranking de partidas y el récord de puntos no son más que un vector ordenados de jugadores como se aprecia en el diagrama siguiente. Aprovecho para introducir los gestores llamados por los controladores siguiendo el patrón MVC que se ha seguido durante todo el proyecto. Es decir que los controladores son los encargados de controlar todo. Los gestores son los que se encargan de grabar el ranking y el récord en el disco duro a memoria permanente. Aúnque aquí no se aprecia ninguna vista se aprecian ya los controladores.



Se aprovecha esta pequeña introducción a los gestores para ver en el siguiente diagrama todos los gestores de disco que son los encargados de grabar todo lo necesario en disco a memoria permanente.

4.3.4 Diagrama de gestores de datos

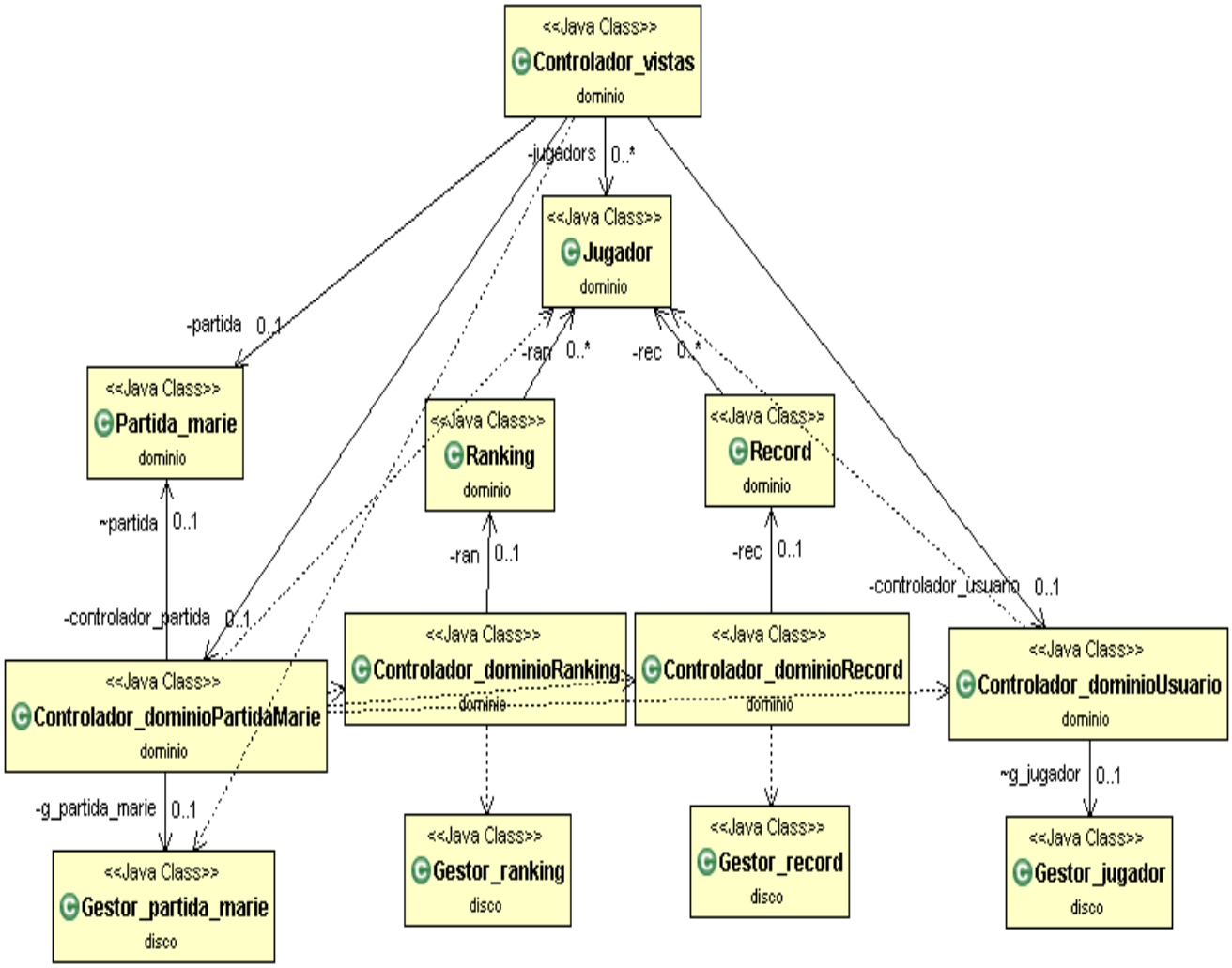
En este diagrama se muestran todas las clases gestoras de datos, es decir las que se encargan de guardar los datos necesarios en el disco de manera permanente. Se aprecia como todos los gestores desciende del gestor objetos.



El gestor jugador sirve para guardar jugadores cuando uno se registra por ejemplo. El gestor Partida marie sirve para guardar partidas. El gestor ranking y récord sirven para guardar como he dicho anteriormente el ranking y el récord y el gestor música fue experimental para guardar música en el disco.

4.3.5 Diagrama de control de datos.

En el siguiente diagrama se muestra el como los controladores, como su nombre indica controlan los datos y se comunican con los gestores encargados de la persistencia a disco (guardar datos). Además se aprecia como las controladores se comunican entre ellos.



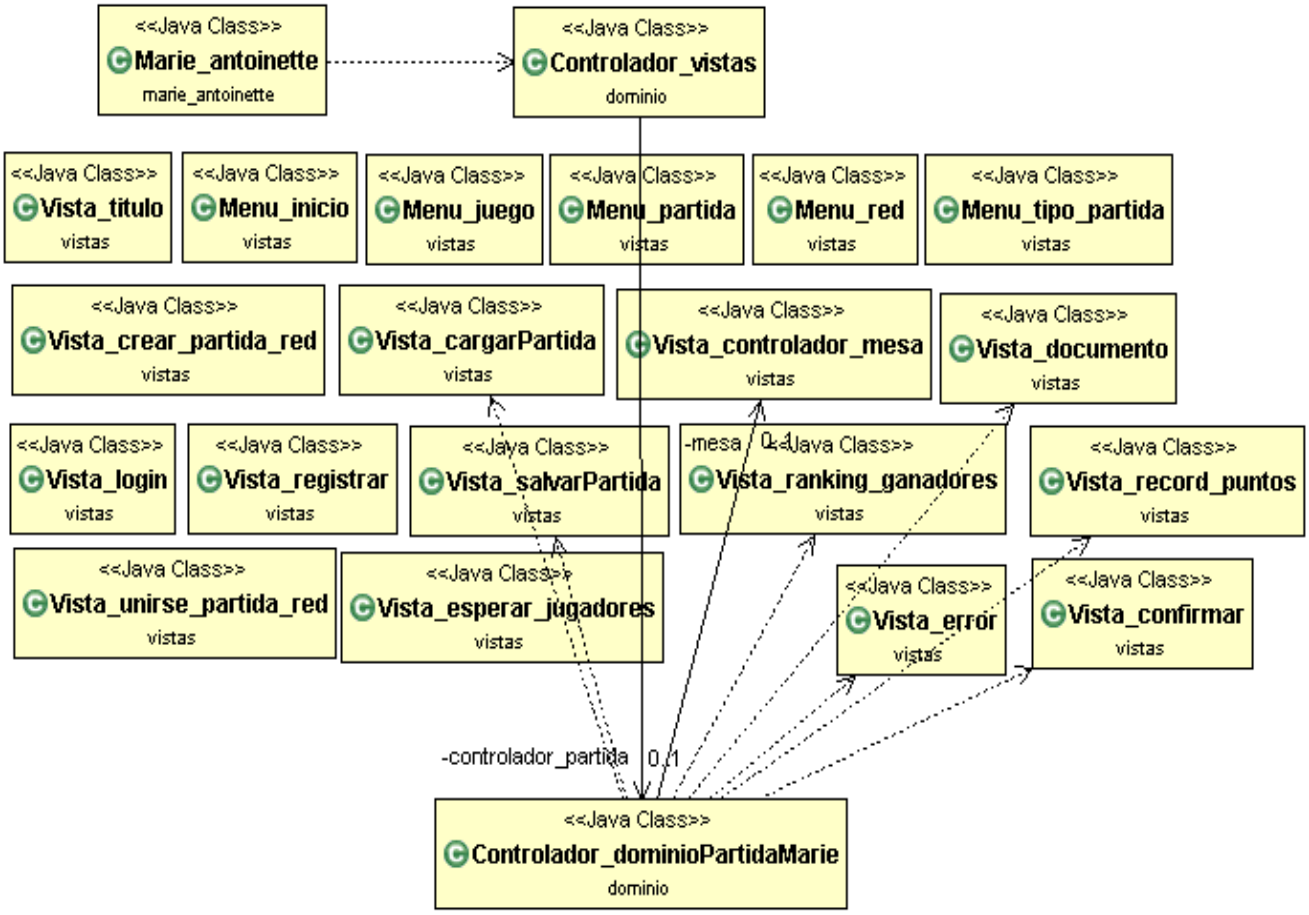
4.3.6 Diagrama de vistas

Ya que en la anterior pagina se han mostrado los gestores de disco ahora se muestran las Vistas y en la siguiente pagina se mostraran los controladores. Se vera así pues la separación entre disco, vistas y controlador y clases de objetos de datos (dominio) que se ha seguido durante todo el proyecto.

En el siguiente diagrama están todas las vistas existentes. Todas estas vistas están controladas por el controlador de vistas (aunque no aparecen las flechas) y por el controlador dominio partida Marie.

El Main del juego esta en la casse Marie antoinette que directamente llama el controlador vistas que es el que se encarga de presentar la pantalla presentación del juego donde pone Marie Antoinette, y luego aparece el menú con las opciones principales del juego, Nueva partida, Cargar Partida... Ver Ranking etc..

El controlador vistas controla todos estos menús y Vistas previos a la pantalla del juego de mesa (donde aparecen las cartas una vez empezado el juego). Cuando se empieza el juego es controlado por el controlador dominio Partida Marie, este controlador controla todo una vez ha empezado la Partida.



5 DISEÑO

5.1 Arquitectura

El proyecto estará estructurado en una arquitectura de 3 capas: presentación, dominio, y datos.

- **Presentación:** Muestra toda la información del juego al usuario como corresponde en cada caso (muestra y quita elementos según corresponda mostrando la vista correcta en cada caso), recibe la interacción del usuario que es consultada y controlada por un controlador de la capa de dominio.
- **Dominio:** Controla la lógica del juego (donde se toman todas las decisiones, el comportamiento del juego), y tiene una representación interna del estado de todos los elementos de la partida. Sabe lo que el usuario ha pulsado consultado la información dada por la capa de presentación y decide como proseguirá el juego.
- **Datos:** Se encarga del almacenamiento persistente de los datos del juego. Es controlada por el dominio y sus controladores.

El objetivo principal de utilizar este modelo es el de minimizar la conexión entre los tres niveles de representación de los elementos del juego: así, la capa de presentación se encargará sólo de la representación visual, la capa de dominio de la representación lógica, y la capa de datos de la representación en el sistema de ficheros. Mantener estas representaciones separadas facilita la sostenibilidad del software.

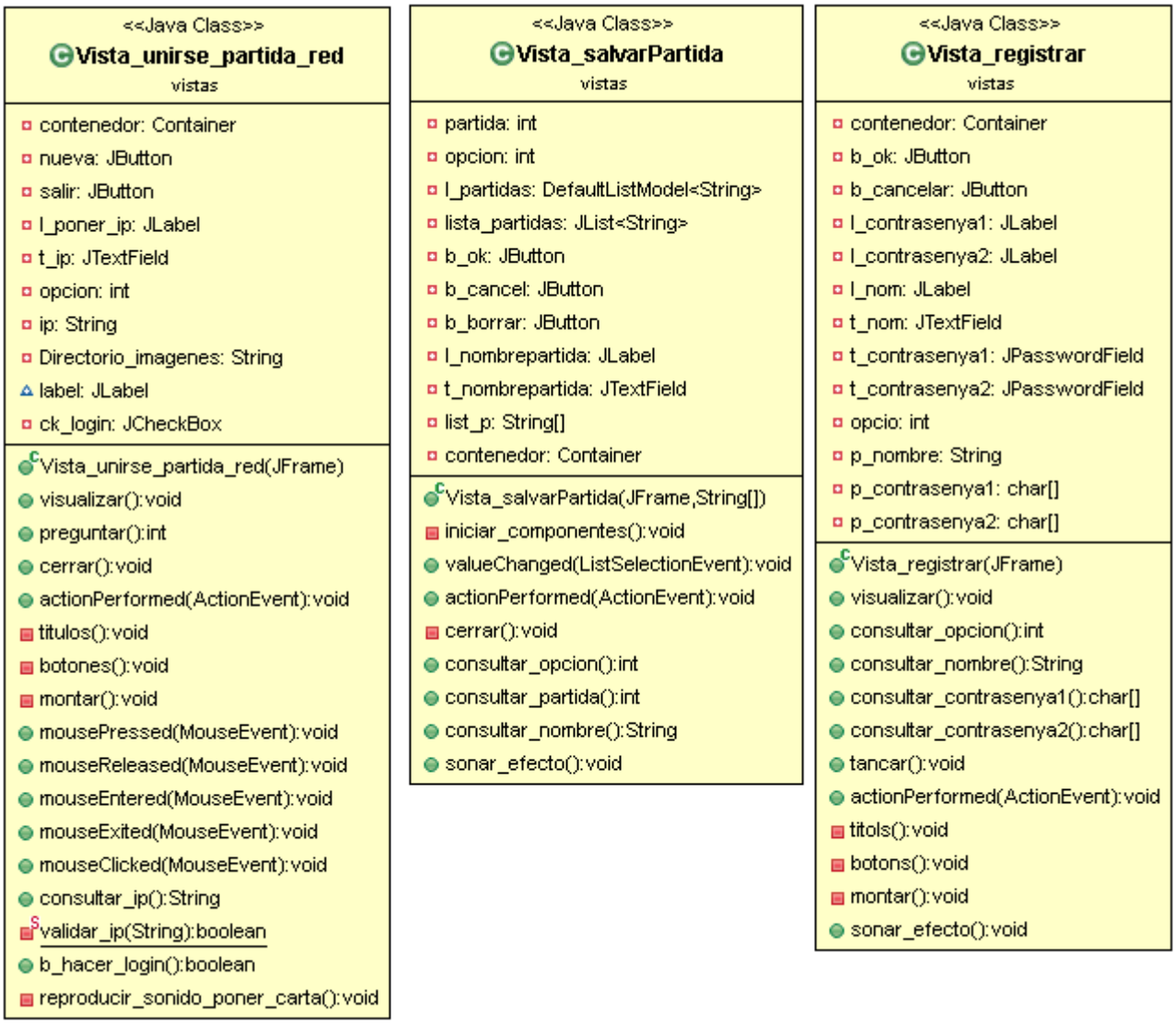
Las conexiones entre las capas serán las mínimas necesarias. Particularmente, se limitarán a las interacciones entre los controladores de cada capa. En la mayoría de casos de uso, el usuario iniciará una interacción con el juego, que será capturado por la capa de presentación. esta, comunicará la información de la interacción del usuario a la capa de dominio (la cual actualizará el estado del juego) con lo que los controladores decidirán sobre la capa de vistas lo que tienen que mostrar y que no, y se comunicará con la capa de datos si es necesario en caso de querer salvar algo a disco.

5.2 Diseño de las capa de presentación

La capa de presentación se compone de una clase o objeto por vista. Cada vista recoge información sobre lo que ha pulsado el usuario, luego en la capa de dominio, los controladores consultan donde ha pulsado el usuario preguntando a la vista que se esta mostrando, y en función de donde se ha pulsado decide que hacer y lo comunica a la capa de presentación. Los objetos que pertenecen a la capa de vista, se podría decir que son tontos, pues no deciden nada, solo comunican la información sobre donde se les ha pulsado a los controladores y muestran lo que los controladores les ordenan.

Las funciones típicas de cada objecto por lo tanto son funciones consultoras (que son consultadas por los controladores), luego también tiene las funciones de mostrar_algo y quitar_algo que son las encargadas de modificar el estado de la vista que visualiza el usuario.

Se muestra ha continuación pues una vista cualquiera donde se aprecian dichas funciones típicas.



Se puede apreciar también como todas las clases tienen la función sonar efecto, la cual es la encargada de hacer sentir al jugador un efecto sonoro. Luego hay la función visualizar, que muestra la vista entera al usuario y la función “tancar” que se encarga de hacer desaparecer la vista entera (siempre bajo las ordenes del controlador). Obviamente cada vista tiene sus variables propias, según el tipo de elementos que tienen.

Luego finalmente las funciones incializar_algo o títulos, botones y montar son las encargadas de inicializar todas las variables. Estas funciones también son comunes en todas las clases de la capa de presentación.

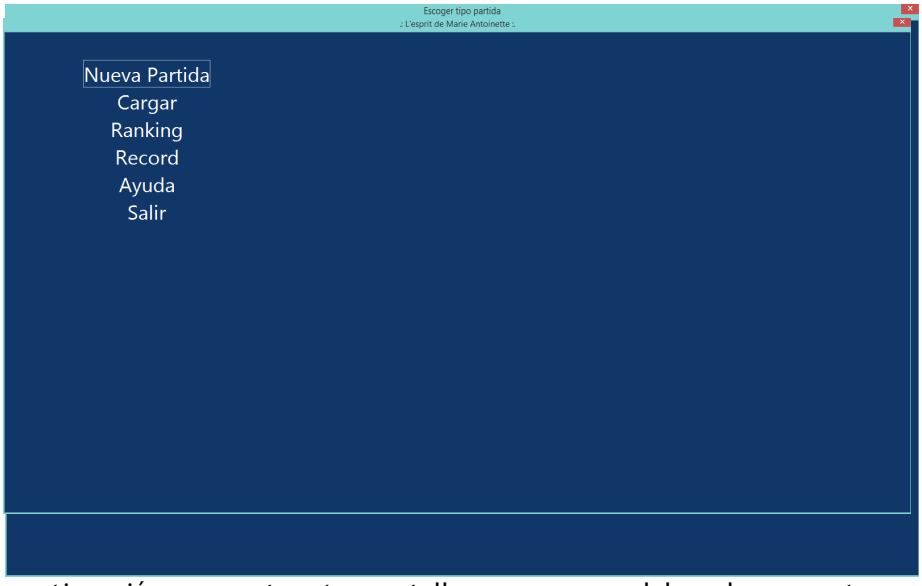
Si el lector quiere puede consultar el anexo donde se puede ver con más detalles todas las clases de la capa de presentación.

Se muestra a continuación unas vistas para que se puedan observar también los elementos en común que tienen todas las clases que pertenecen a la capa de presentación. De esta forma podéis ver visualmente

los elementos comunes dichos anteriormente en las clases mostradas a la pagina anterior.

Primero se muestra la vista del menú principal del juego. Vemos como hay botones incializados por la función botones() y montar() y si se mueve el cursor sobre ellos suenan con la función sonar_efecto ().

La pantalla se muestra con la función mostrar() y se cierra con cerrar().



A continuación se muestra otra pantalla un poco mas elaborada, concretamente la de crear partida local, para mostrar los elementos comunes en el diseño. Vemos como hay botones , estos son incializados por la función botones() y montar() los textos o labels son incializados por la función titols().Esta pantalla también se muestra con la función mostrar() y se cierra con cerrar(). Las opciones pulsadas por el usuario el controlador las sabe gracias a las consultoras. En el apartado del proyecto de la implementación se informa al lector que encontrareis

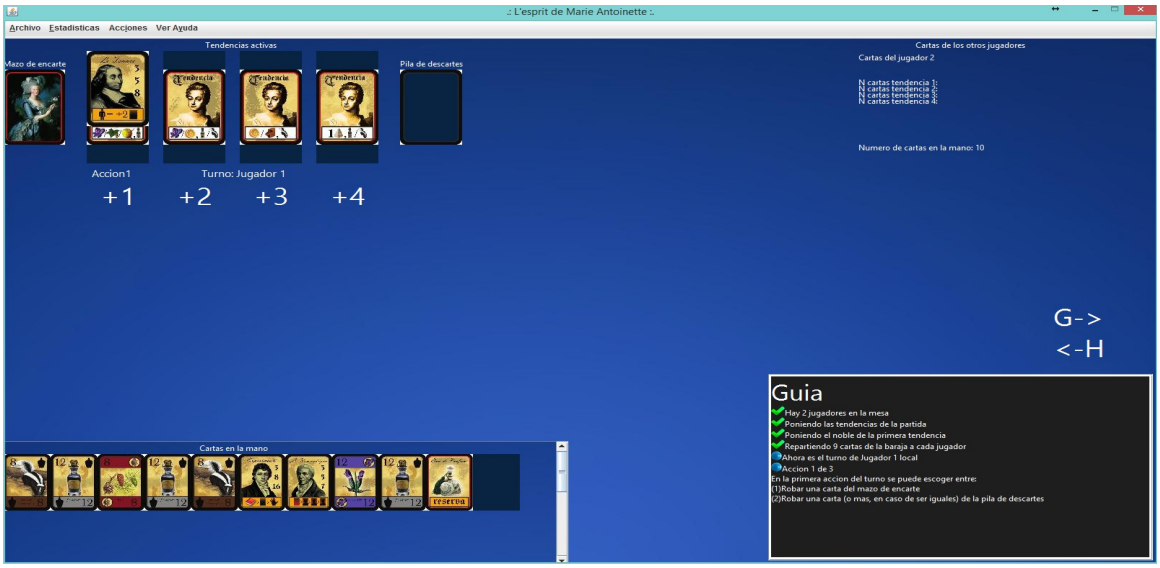
Una vez se le ha explicado al lector como es el diseño de las vistas y sus clases, lo único que resta por explicar del diseño de las vistas, es la vista de la pantalla principal del juego de mientras se juega, de que elementos se compone y

5.2.1 Vista principal

Para componer la vista principal del juego hay primero que entender todo el juego y su funcionamiento.

Ha medida que vas diseñando dicha vista el proyectista se dio cuenta que había en parte un problema de espacio, así que el proyectista tuvo que ir rediseñando dicha vista a medida que iba colocando los distintos elementos. Se nombra ha continuación los distintos elementos.

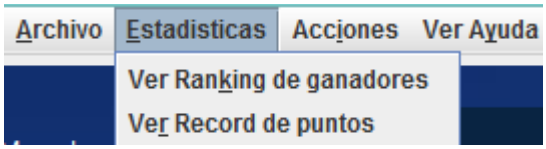
Se muestra a continuación la pantalla del juego o vista principal entera y luego se explica elemento por elemento que es cada cosa y el porqué de su posición



Se empieza primero a explicar uno por uno los elementos comunes para todos los jugadores que son empezando con la barra de funcionalidades de arriba:

1. El menú superior con el Archivo, Acciones,Estadísticas y la Ayuda. El Archivo contiene un submenu desplegable con Reiniciar Partida, Salvar Partida, Cargar Partida , Abandonar Partida y Salir.

2. Las Estadísticas, con Ver Ranking de ganadores y Ver récord de puntos.



3. Acciones, con la funcionalidad de retroceder jugada
4. Ver ayuda funcionalidad la cual muestra la ayuda entera con todas las normas y el juego explicado al usuario.

Una vez visualizado el menú superior o barra de tareas se continua explicando los elementos comunes, esta vez los elementos que forman parte de una partida en curso.

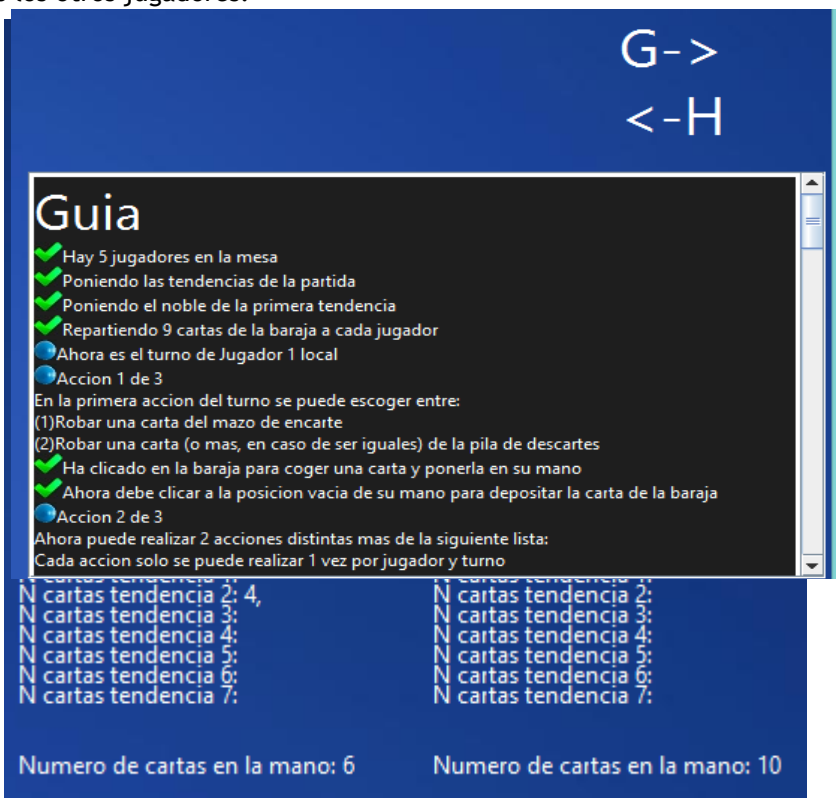
En su conjunto son estos:

1. El Mazo de encarta o la baraja, es de donde se roban todas las cartas todos los jugadores. Como es la jugada más repetida para un jugador y la jugada inicial también, su posición esta por lo tanto en la parte superior izquierda, tal cual se empieza a leer.
2. La pila de descartes es donde se pulsa cuando quieres descartar dicha carta. Como es una jugada que se tiende ha hacer en fases finales de la partida, el proyectista situó su posición más a la derecha.
3. Entremedio hay los Nobles en la parte superior central, los cuales se van colocando a medida que va prosperando la partida.
4. Las Tendencia los cuales están ya todas colocadas desde el principio están colocadas. Es por esto que aparecen entre los nobles y donde irán las Reservas
5. Las Reservas, los cuales no hay ni una colocada desde el principio y están situados debajo las

Tendencias.

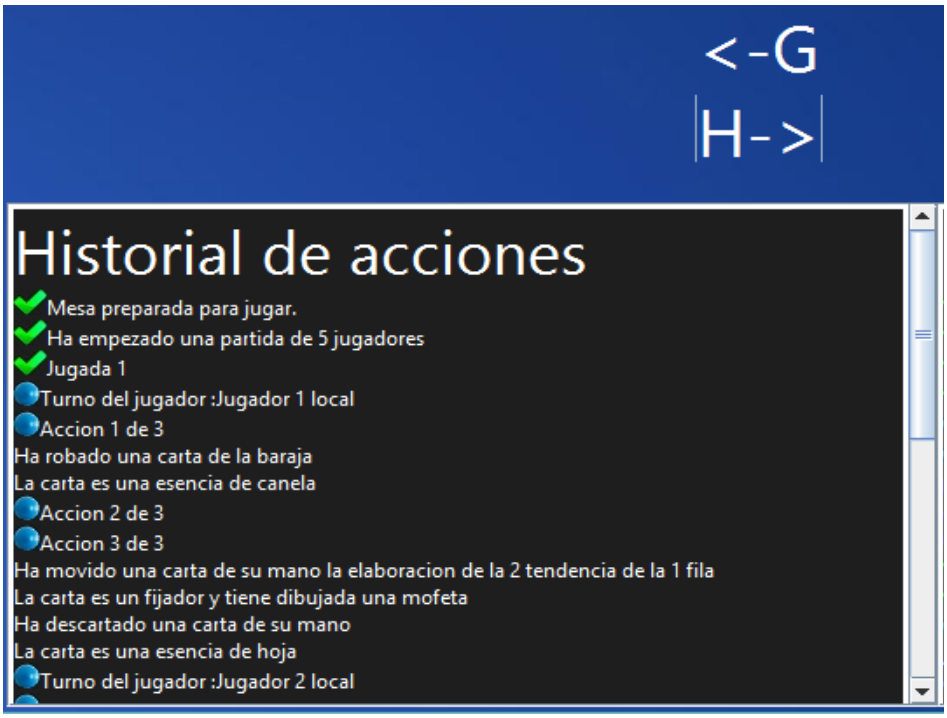
Aunque Nobles,Tendencias y Reservas se solapan en el espacio por motivos de espacio, estas pueden ser mostradas su imagen en su totalidad si se pulsa encima de cualquier de ellas.

6. Información sobre cartas de otros jugadores,seguimos pues con los elementos comunes para todos los jugadores. La información sobre las cartas que tienen otros jugadores esta en la parte superior derecha de la pantalla de juego y muestra toda la información relativa al numero de cartas que tienen todos los otros jugadores.



7. La Guía. Este elemento esta en la parte inferior derecha de la pantalla y muestra información sobre las opciones que puede tener un jugador para jugar en todo momento, es decir le indica donde puede pulsar en el ratón. Este elemento fue hecho para facilitar el juego al usuario y dado la complejidad de las reglas resulta imprescindible. Para visualizar la guía se puede pulsar sobre <-G. Tanto la guía como el historial fueron rediseñados y reimplementados, son 2 pequeñas pantallas que muestran código HTML.

8. El historial esta ubicado en el mismo lugar que la guía (en la parte inferior derecha). Este muestra todas las jugadas realizadas hasta el momento de todos los jugadores. Para ver el historial se puede pulsar sobre <-H. Y para esconderlo ->H o <-G haciendo aparecer la guía.



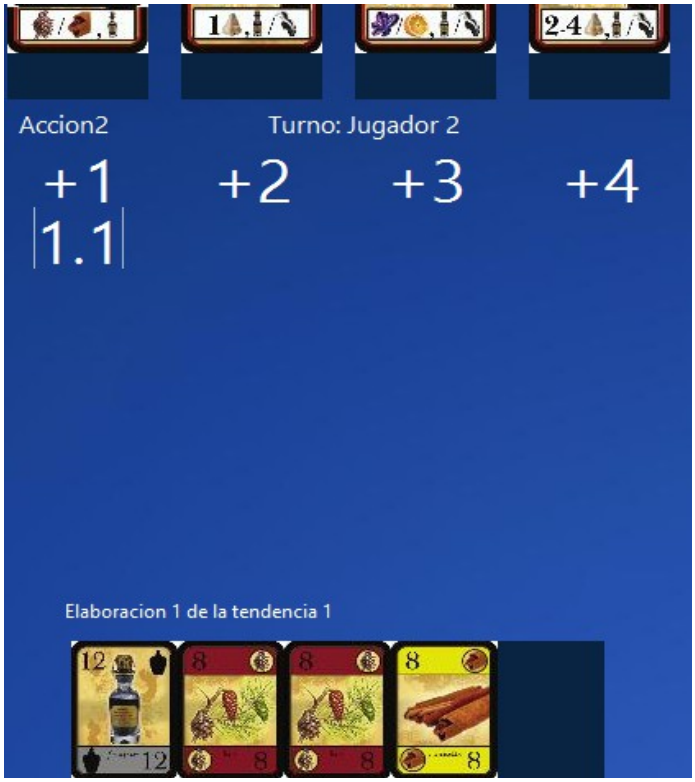
9. El zoom sobre la carta clicada. El zoom esta entre las cartas de la mano y la guía. Esta hecho para ayudar a visualizar una carta en concreto si el jugador no la puede ver, aunque no es necesario de hecho porque su tamaño reducido ya es suficientemente grande para poder ser visualizado. Vemos en el dibujo mostrado a continuación la diferencia entre ambos tamaños y se aprovecha este párrafo para hacer constar que todas las imagines originales son del tamaño grande y posteriormente hay una función que las reduce todas al tamaño pequeño y las guarda a disco, de este modo solo se hace esta operación de reducción una sola vez ganando eficiencia.

Llegados a este punto ahora se explicara los elementos de la partida individuales de cada jugador de la pantalla de juego.

1. Las carta de la mano. Están situadas en la parte inferior izquierda de la pantalla debajo de la baraja de donde roban las cartas. Esto simula la realizad pues la cartas de cada jugador están una partida real están cerca de él y las cartas comunes como la baraja alejadas de él en centro de todos los jugadores.

Si se pulsa en la sección de cartas de otros jugadores y pulsamos sobre un jugador en concreto, entonces se muestran las cartas de la mano de ese jugador giradas al revés pues un jugador no puede ver las cartas de la mano del otro jugador. También aparecen mostradas de este modo cuando se esta jugando en red y es el turno del otro jugador.

2. Las elaboraciones o *parfumothèques* es decir los perfumes que cada jugador crea. Estas están situadas entre la mano y las tendencias. Vemos en la figura de abajo un perfume valido para la primera tendencia. Vemos como los +1 +2 +3.. están justo debajo de cada tendencia activa. Si se pulsa encima de +1, entonces aparece el 1.1, 1,2 y 1,3 y aparece también el espacio para empezar el perfume o elaboración. Se aprecia como encima de cada espacio indica a que elaboración pertenece.



5.3 Diseño de las comunicaciones

El diseño de las comunicaciones está pensado para ser lo más simple posible, y minimizar la cantidad de información que se transmite. Se explica a continuación que el proyectista diseño el típico modelo de partida cliente servidor donde solo hay un jugador en cada ordenador y también creo la posibilidad de que hubiese más de un jugador en cada ordenador.

Las comunicaciones se realizarán mediante sockets utilizando el puerto 80. Para iniciar una partida multijugador, uno de los jugadores (que será el host) debe crear la partida. Cuando lo hace, se inicia un servidor, que escucha un puerto determinado hasta que llega una conexión entrante (del otro jugador, el cliente). Una vez que la conexión se ha establecido, el cliente envía el objeto jugador entero al host o servidor. Luego el servidor crea la partida y la envía al cliente una copia de esta (el objeto entero). Si el lector quiere saber como es la clase partida y jugador se le recomienda ir al anexo y podrá observar todos sus variables que componen ambos objetos. El cliente al empezar la partida no vuelva a inicializar la partida, simplemente utiliza la partida que el servidor le ha pasado.

A partir de este punto, cada jugador enviara al servidor donde ha pulsado y el servidor se encarga de enviar dicha información a los demás jugadores. Los demás jugadores esperan a recibir dicha información del servidor si les toca esperarla o la enviaran al servidor si les toca enviarla (cuando es su turno). Una vez recibida donde ha clicado el jugador en red, el controlador procesa dicha información como si de una jugada local se tratase, a excepción que no muestra la cara de las cartas, es decir muestra el culo o reverso de las cartas de la mano si el jugador no él, es decir si es el turno de un jugador que esta jugando en otro ordenador. Las cartas de la mano de cada jugador solo las ve el propio jugador, los demás ven el culo o reverso de las cartas.

La información que se transmite es la mínima necesaria (es decir donde se pulsa el jugador en red) para que movimiento efectuado de manera unívoca. Concretamente, se envía la información proporcionada la vista de la pantalla de juego que es consultado por el controlador. El único caso donde se envía más información aparte de donde ha clicado, es en una jugada de un noble que baraja de nuevo toda la baraja y la mezcla de nuevo, en este caso único también se envía como información extra toda la baraja, pues ese elemento ha cambiado y debe ser actualizado por los otros jugadores.

5.4 Diseño de dominio

El dominio se caracteriza por ser donde están los objetos o clases que representan cosas reales del juego, por ejemplo un jugador, una carta, una pila de descartes, una elaboración, etc...También es donde están todos los controladores del juego, es decir los que controlan y toman decisiones y las comunican a la capa de presentación o de disco, también modifican si es necesario los objetos que representan cosas reales del juego. Se recomienda leer la sección que corresponde al modelo contextual de este documento y el Anexo para ver con más detalle cómo es la capa de dominio.

5.5 Diseño de la Inteligencia Artificial

En esta sección se explica el diseño de la inteligencia artificial. La inteligencia artificial de juego tiene 4 niveles: fácil, media, difícil y muy difícil. El objetivo a perseguir en la inteligencia artificial es el de crear un máximo numero perfumes para al final poderlos presentar en la fase de la puntuación.

- **Fácil:** la inteligencia artificial fácil esta pensada para ser vencida por el usuario. Presenta una puntuación final muy baja como se apreciar en la comparativa realizada, concretamente la media es de unos -28 puntos. La puntuación es tan baja porque no se realizan jugadas encadenadas, es decir solo se mueven 3 cartas por turno y además no se realiza ninguna estrategia excepto cuando se pone un noble en juego que entonces si que se realiza una buena jugada. La inteligencia artificial fácil puede realizar todos los tipos de jugadas existentes. Esta hace elaboraciones o perfumes validos lo que lo hace con un numero de jugadas elevados ya que solo puede mover una carta en cada turno para hacer una elaboración. Como se ha dicho no presenta ninguna estrategia en las jugadas (excepto nobles) porqué se hacen sin ningún orden o sentido para ganar, simplemente se hacen para que el ordenador que funciona a fácil tenga al final de la partida algún que otro perfume valido a presentar. A continuación se explican algunas de las estrategias de los nobles, por ejemplo en el noble el experimentado se busca un fijador a robar para maximizar el numero de elaboraciones que el jugador podrá hacer y minimiza el numero de cartas que el jugador tendrá en la mano al finalizar la partida. Otro noble mejorado es el diablo que elimina la tendencia en juego que menos daño hace al jugador. Otra mejora realizada entre otras es que el noble de mayor puntuaciones se pone en la tendencia que más conviene al jugador.
- **Media:** la inteligencia artificial media, presenta ya cierta dificultad. La inteligencia artificial media hace ya jugadas encadenadas, es decir que mueve más de tres cartas por turno. Su forma de elaborar perfumes es mucho más optima que en la inteligencia fácil pues puede crear una perfume y empezar otro en solo 2 jugadas, esto es porque busca siempre la carta que más se repite en la mano para realizar el movimiento de crear un perfume, de esta manera se optimiza bastante el movimiento. En la jugada de robar una carta de la pila de descartes, solo se lleva a cabo si la carta que se recoge es un fijador, ya que de esta manera se maximiza el numero de perfumes que se pueden hacer y reduce el numero de cartas no necesarias que se pueden tener en la mano. Otra

mejora realizada es que se descartan las cartas de la mano según su puntuación negativa (pues el tener cartas en la mano resta en la puntuación final), es decir primero se descartan los nobles (-2 puntos) y luego se descartan las esencias(-1). Dicha inteligencia también presenta las mejoras de los nobles de la dificultad fácil.

- **Difícil:** la inteligencia artificial en difícil a diferencia de la inteligencia media en que esta considera el avance del juego. Es decir realiza jugadas distintas en función de si esta al principio del juego o del final. Por ejemplo si se encuentra al principio del juego hay un tendencia clara de robar cartas del mazo de encarte. Esto es así porque de esta manera luego la jugada de crear elaboraciones o perfumes resulta más optima, ya que puedes aprovechar mejor el movimiento moviendo mas cartas iguales. De este modo se pueden crear más elaboraciones a la hora de puntuar, en menos jugadas. Por lo tanto el jugador tiene de este modo más posibilidades de puntuar.

  Otra mejora que resulta haber en la inteligencia difícil es que por ejemplo al final de la partida resulta haber una tendencia en descartar cartas de la mano del jugador. Esto resulta ser así ya que para mejorar la puntuación final y maximizar el numero de perfumes elaborados que se puedan elaborar la estrategia a seguir resultante es que al final de la partida eliminas cartas de la mano. Se eliminan primero las cartas que más penalizan es decir los nobles y luego las esencias.

- **Muy difícil:** La estrategia resultante de la dificultad muy difícil resulta ser la misma que la resultante en la dificultad difícil lo que simplemente esta más refinada para obtener mejores puntuaciones. Es la misma lo que simplemente los movimientos que hace la maquina están mejorados ya que tienen en cuenta más precisamente a que fase esta el juego para realizar la mejor jugada.

Se muestra a continuación las recomendaciones del juego para la inteligencia artificial o para un jugador real dado por el autor del juego copiada directamente:

“Hace algún tiempo ya di algunos **consejos de juego** para L’**Esprit de Marie Antoinette** y basándome en esa entrada creo esta otra.

El motivo es que como están implementándolo para su versión digital y no es un juego de cartas sencillo, su programador (que ha elegido este juego como proyecto de fin de carrera) se ha topado con el “serio

problema” de crear un algoritmo con el que implementar estrategias de juego.

Resulta que en los juegos para más de dos personas no hay un algoritmo más o menos claro como, por ejemplo, el **minimax** para los juegos de dos jugadores. Por eso, para montar inteligencias artificiales (o, más modestamente, programas que jueguen), hay que implementar ciertas estrategias generales de juego (como mínimo unas cuantas para poder tener dos o tres IA’s distintas).

Así que sirva esta entrada como una forma de encontrar algunas posibles estrategias, movimientos y detalles con el fin de maximizar tus posibilidades de ganar en el juego:

1. Fija una Estrategia a medio-largo plazo

Al principio de la partida se hacen públicas algunas Tendencias demandadas en París (el número depende de la cantidad de jugadores), así que lo primero que debes hacer es estudiarlas, y fijarte una estrategia a medio-largo plazo.

Lo primero que debes mirar es que cantidad de fijadores se necesitan de cada tipo, ya que suele ser la mayor restricción u obstáculo con el que te vas a encontrar para progresar en tu *parfumothèque*. Además, recuerdo que tampoco son equiprobables (solo hay 8 fijadores animales y 12 fijadores resina en el mazo) y esto también hay que tenerlo en cuenta.

Lo siguiente es ver que Principios Olfativos son los más demandados; suele ser una buena estrategia quedarse con aquellos que aparecen un mayor número de veces en las Tendencias, sobre todo si en tu mano inicial dispones de algunas de ellos. De esta forma siempre tendrás varias opciones de ofertas al final de la partida.

2. Los primeros turnos

Los primeros turnos suele ser una buena opción robar 1 carta adicional para ir decidiendo en que tipos de Elaboraciones te vas a dedicar. No obstante, si tienes una mano inicial “potente” igual te puede interesar ir a cerrar la partida lo antes posible e irte descartando de cartas, pero ten en cuenta que para ello habrá que jugar todos los Nobles sobre las Tendencias públicas y lo más probable es que, si se percatan de tu estrategia, tus oponentes no lo hagan, entorpeciendo por tanto tu cierre.

3. Nobles

Al principio no hagas demasiado caso a los valores de los Nobles, lo más probable es que cambien a lo largo de la partida.

Lo que si tienes que valorar es aquellos que tienes en la mano inicial sobre que Tendencias jugarlos y que momento es el idóneo, para así irte preparando tu *parfumothèque*, y dar un golpe de efecto. El saber jugar

los Nobles en el momento adecuado es una de las partes más críticas del juego. Normalmente es una buena estrategia guardarte uno o dos para cerrar con ellos y pasar a la muerte súbita.

Si tienes algún Noble ya en juego, como no se puede repetir, una buena opción es descartárselo o antes posible (para que así no te pillen puntos en la mano), salvo que tengas algún Noble que te permita eliminar el que ya hay en juego con el mismo nombre.

4. Elaborar mi parfumothèque

A veces suele funcionar el ir acumulando Principios Olfativos de un mismo tipo en la mano y jugarlas de golpe en una o varias Elaboraciones. Suele ser una buena jugada ya que optimizas acciones (la de poner en juego Principios Olfativos) y sobre todo no desvelas con anterioridad tus intenciones a tus oponentes. Muchas personas esperan el momento para recoger la mejor combinación posible para bajarla de golpe y así ahorrarse acciones. No obstante, debes tener especial cuidado con este tipo de estrategias, ya que si son varios los que deciden acortar la partida pueden pillarte con un buen número de cartas en la mano que te penalizaran.

Algunos de ellos persistentemente quieren esperar tener muchas para asegurarse de que van a formar Elaboraciones de 5 Principios Olfativos con facilidad. No cometas este error. Si tienes algunas cartas que no se ajustan o tienen cabida en tus Elaboraciones, todavía hay una oportunidad de ganarle a tus oponentes. Baja tus cartas tan pronto como sea posible y haz con ellas Elaboraciones aunque luego no puedas venderlas (con que no te lleves negativos en la fase de ofertas es suficiente). Incluso en una situación en la que tienes peores Principios Olfativos que tu oponente, es mejor bajar primero. Es un buen paso por el riesgo y las posibles combinaciones que luego podrás hacer con ellas. Además, hay una acción que te permite mover Principios Olfativos en tu *parfumothèque,* si hiciera falta.

5. Reúne suficientes fijadores válidos

Otro de los grandes retos del juego es reunir suficientes fijadores con los que realizar Elaboraciones válidas que luego poder ofertar. Si en tu mano inicial no tienes suficientes, tu primera estrategia debe ser conseguirlos lo antes posible. Profundiza en el mazo o haz uso de algunos Nobles para tal cometido.

6. Desecha Principios Olfativos

Los Principios Olfativos que aparecen menos en las Tendencias no son óptimos y te juegan en contra en este juego. Si no han sido ya incluidos en algunas de las Tendencias “de nariz”, no hay necesidad de quedárselos. Puedes ayudar al oponente si no te desprendes de ellas. Así que en primer lugar, descarta todos los Principios Olfativos que no quepan en tu juego.

7. No saques las cartas del Descarte a menos que quieras cerrar el juego

El jugador con el Descarte da pistas de tu estrategia a tus oponentes. Lo ideal es que no saques cartas del Descarte a menos que quieras cerrar el juego y pasar a la muerte súbita; o no tengas más remedio (si es un fijador, por ejemplo).

8. Memoriza las cartas descartadas por tus oponentes

L'Esprit de Marie Antoinette es un juego donde no solo juegas en tu turno. Es muy importante que vayas estudiando los movimientos de tus oponentes, sobre todo que Principios Olfativos van colocando en su *parfumothèque* y de que cartas se descartan; y también ir memorizando aquellos Nobles que van saliendo a juego o se van eliminando. Esto te ayudará para saber elegir a que Tendencias dedicarte y cuando utilizar la carta de Reserva (que tiene que servirte para asegurarte la mayor puntuación con ella).

9. Reserva una Tendencia antes de cerrar

Cuando ya tengas claro en que Tendencias vas a ofertar es el momento de reservar una Tendencia, sobre todo si tú eres el que piensa cerrar el juego. Reservar una Tendencia a ciegas no suele ser muy útil, sobre todo si tenemos en cuenta que para cambiar la de sitio tenemos que perder dos acciones.
Calcula en cuál de ellas te conviene reservar la Tendencia para ser el primero en ofertar, si con ello obtienes muchos puntos (normalmente una Preparación de cinco Principios Olfativos) y adicionalmente si con ello haces que alguno de tus rivales pierda esos puntos (de esta manera la ganancia es doble).

10. No te olvides del acorde perfecto

Si entre todas las Elaboraciones válidas (las que quedan tras revisar todas) un perfumero ha utilizado los ocho grupos de Principios Olfativos obtiene un premio de +10 puntos; el acorde de su *parfumothèque* ronda la perfección. (Nota: El acorde es la delicada composición de esencias en determinadas proporciones, que el perfumista logra al crear una nueva fragancia). Estos diez puntos adicionales son vitales.

11. Prepararse para la fase de Ofertas

La fase de Ofertas suele ser lo más complejo del juego con diferencia. En las primeras partidas, cuando uno anda perdido vagando en la partida, no se tiene muy claro como jugar; y llegada la fase de Ofertas apenas puede ofertar sus Elaboraciones porque alguien se le ha adelantado. Lo normal es echarle erróneamente la culpa al azar ya que no se tiene control sobre quien va a terminar la partida y por tanto cuando va a ser tu turno de ofertar. Pues bien, esto es una falacia. Si durante la primera fase has ido posicionándote, elaborando las mejores Elaboraciones posibles, sabiendo usar tu carta de Reserva, y estirar o acortar la duración de la partida con los Nobles según te convenga, cuando llegue la fase de Oferta estarás en una

posición ventajosa. Es más, puede ocurrir incluso que en algunas ocasiones no te interese ser tu quien cierre la partida. Lo que quiero decir con esto, es que es muy importante jugar bien la primera fase del juego porque los fallos o despistes se arrastran a la fase de Ofertas.

A veces puede ser una mala estrategia dedicarse a hacer Elaboraciones del máximo de Principios Olfativos (cinco) cuando puede salir a cuenta hacer mayor número de ellas de 1 Principio Olfativo menos (cuatro), sobre todo si detectas que muchos jugadores van a ofertar en la misma Tendencia ya que solo uno podrá ofertar y los restantes no. En ocasiones, te puede interesar usar la carta de Reserva para asegurarte ser tu quien oferte y fastidiar al resto que verá como una Preparación de 5 Esencias no puede colocarla.

Luego, también es una buena estrategia elegir ofertar en aquellas Tendencias donde la variabilidad de los puntos que dan sus Nobles sea mínima. De esta forma, puedes realizar movimientos de Esencias en tu *parfumothèque* sin miedo a perder demasiados puntos. A veces una retirada a tiempo resulta clave.

See more at: http://labsk.net/wkr/archives/tag/marie-antoinette/#sthash.EZJ1INVj.ng7qtMPN.dpuf "

–

Hasta aquí el texto del autor. El proyectista ha seguido cada uno de los consejos dados por el autor para realizar la mejor inteligencia artificial que el ha sido capaz de crear basándose en los consejos y en su experiencia propia de juego, conociendo todas y cada una de las normas profundamente resultado de haberlas leído entendido y programado. Aún así el proyectista esta convencido en que por ejemplo si este juego se profesionalizase como por ejemplo el ajedrez alguna estrategia surgirá más optima resultado de algunos recovecos en la estrategia que no es consciente y que se han podido pasar por alto.

5.6 Diseño de la persistencia

En este proyecto se guardan los jugadores al registrarlos, las partidas al salvarlas, los rankings y el récord al modificarlos. El proyectista explica también que guardan unos jugadores estándares (solo una vez) por si el usario no le apetece hacer el login como son Jugador en local 1..5, Ordenador, Jugador en red 1...5. Si el usuario no hace el login se usan dichos jugadores estándares según sea el caso.

El proceso de salvar una partida solo puede hacer al principio de cada turno. Esta decisión fue tomada por el proyectista porque en su momento decidió que era mejor salvar las partidas solo al principio de cada turno que salvarlas en medio de un turno con las jugadas a medias. El proyectista quiere hacer notar que esta decisión tomada al inicio del proyecto (que parecía lógica) luego le fue perjudicial a la hora de hacer la funcionalidad de retroceder jugada. Pues no se podían guardar los estados de jugadas entre turnos y hacer dicha funcionalidad se complico bastante.

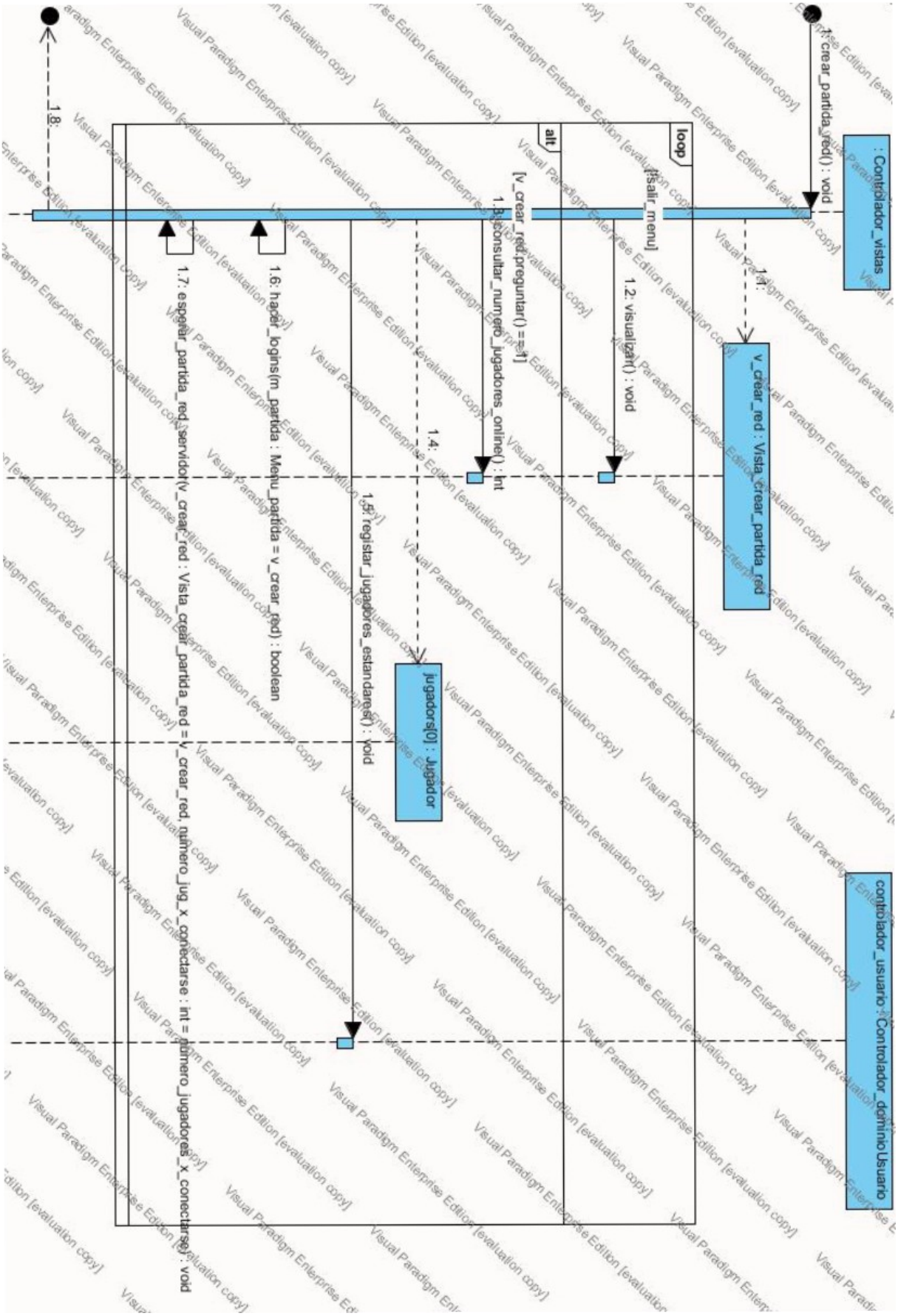
Si el lector desea ver de que se componen dichos objetos en el anexo hay puesto los objetos para que los puedan ver en detalle.

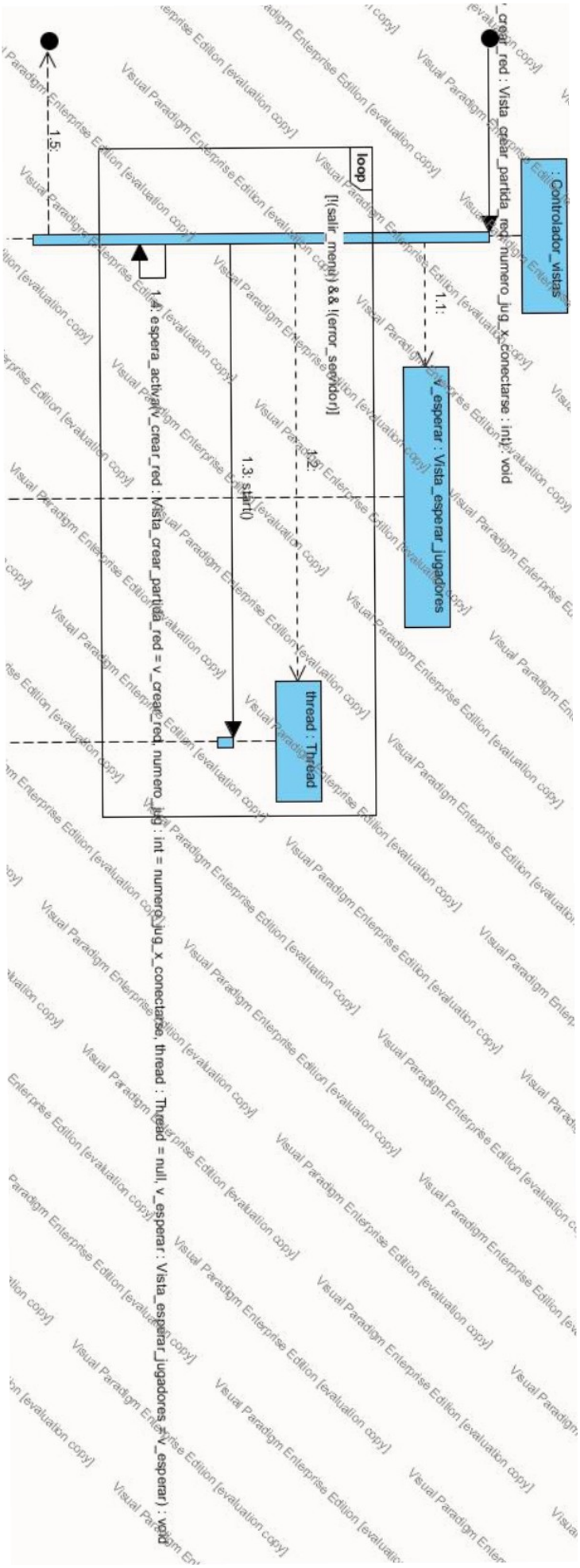
Aparte de los objetos citados anteriormente, también se guardan a disco todas las cartas pero en tamaño reducido. Esto se decidió hacer en su momento para no tener que hacer el cambio de tamaño (resize). Cada vez. De esta manera solo se hace el resize de la imagen una sola vez, que es la primera vez que el usuario ejecuta el juego, en las partidas posteriores de otros días no se vuelven ha hacer el resize puesto que ya están salvadas a disco. De esta manera se optimiza el juego, en ese aspecto.

5.7 Diagramas de secuencia

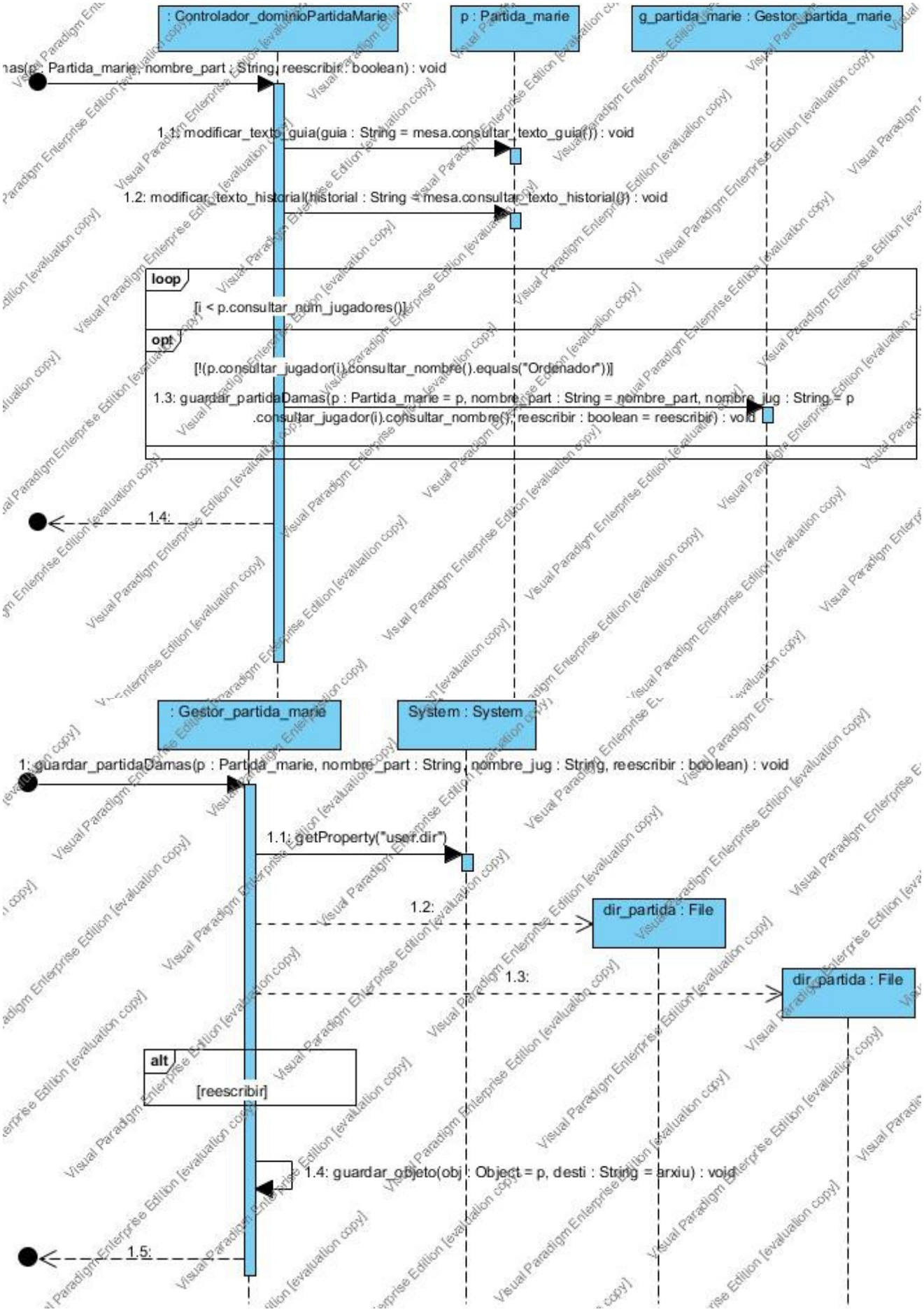
Los diagramas de secuencia fueron creados automáticamente a partir del código usando un programa llamado Visual Paradigm Enterprise Edition. Este programa no es gratuito pero ofrecen un mes de prueba, tiempo más que suficiente para usarlo para generar los diagramas de secuencia a partir del código. Este programa está incluido en el pen drive que se encuentra enganchado a la ultima pagina de la memoria, además hay algunos diagramas de secuencia que resultan demasiado largos para meterlos en una hoja DINA4 que se encuentran también en el pen drive. En el pen drive están todos los que hay aquí y más. Los diagramas de secuencia existentes son algunos de los correspondientes a los casos de uso mostrado en el apartado de la especificación de la memoria. Se recuerda que són: Nueva partida local, Crear partida en red, Unirse a una partida en red, Hacer movimiento, Deshacer movimiento, Salvar partida, Borrar partida, Cargar partida, Abandonar partida, Reiniciar partida, Mostrar ranking, Mostrar récords, Mostrar ayuda, Mostrar guía, Mostrar historial de jugadas y Mostrar cartas de otros jugadores. En este documento están solo algunos de los diagrama de secuencia más simples por cuestión de tamaño. El resto están en el pen drive.

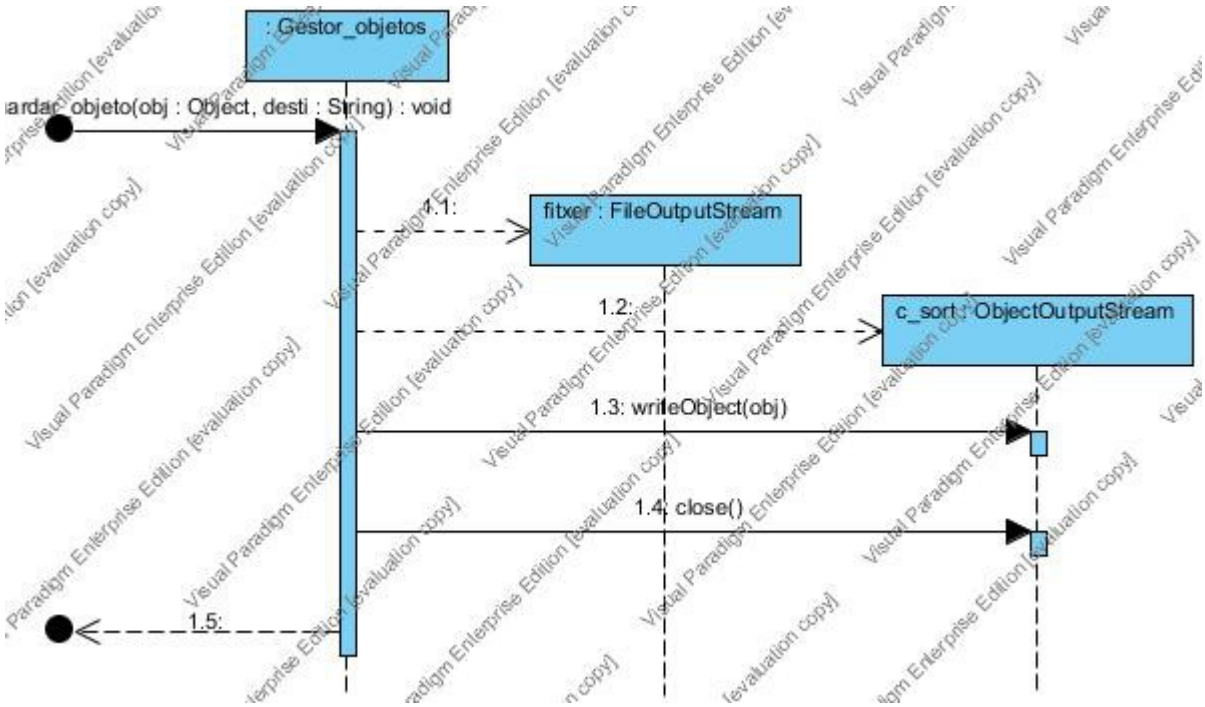
5.7.1 Crear partida en red



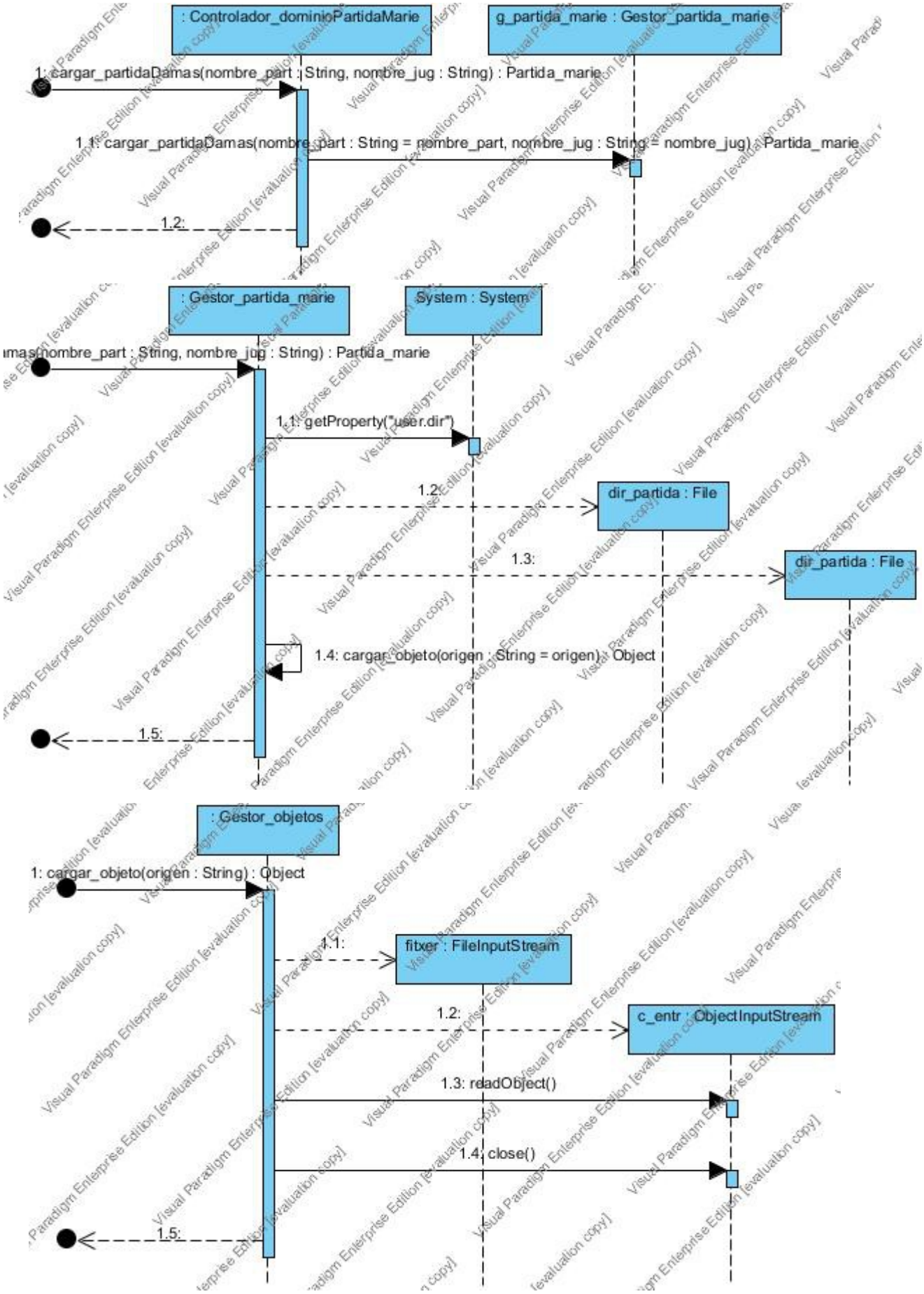

crear_red : Vista_crear_partida_red
: Controlador_vistas
numero_jug_x_conectarse : int) : void
1.1:
v_esperar : Vista_esperar_jugadores
loop
[!(salir_menu) && !(error_servidor)]
1.2:
1.3: start()
thread : Thread
1.4: espera_activa(v_crear_red : Vista_crear_partida_red = v_crear_red, numero_jug : int = numero_jug_x_conectarse, thread : Thread = null, v_esperar : Vista_esperar_jugadores = v_esperar) : void
1.5:

5.7.2 Salvar partida

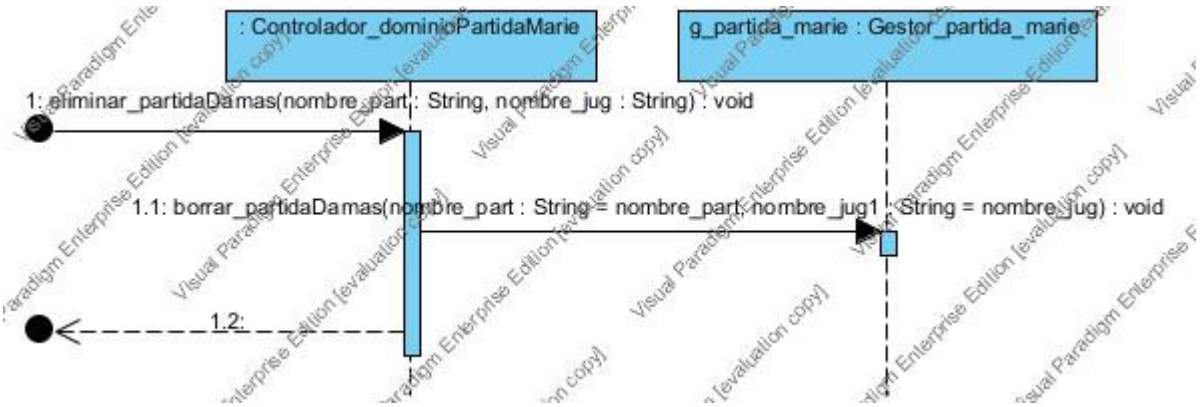


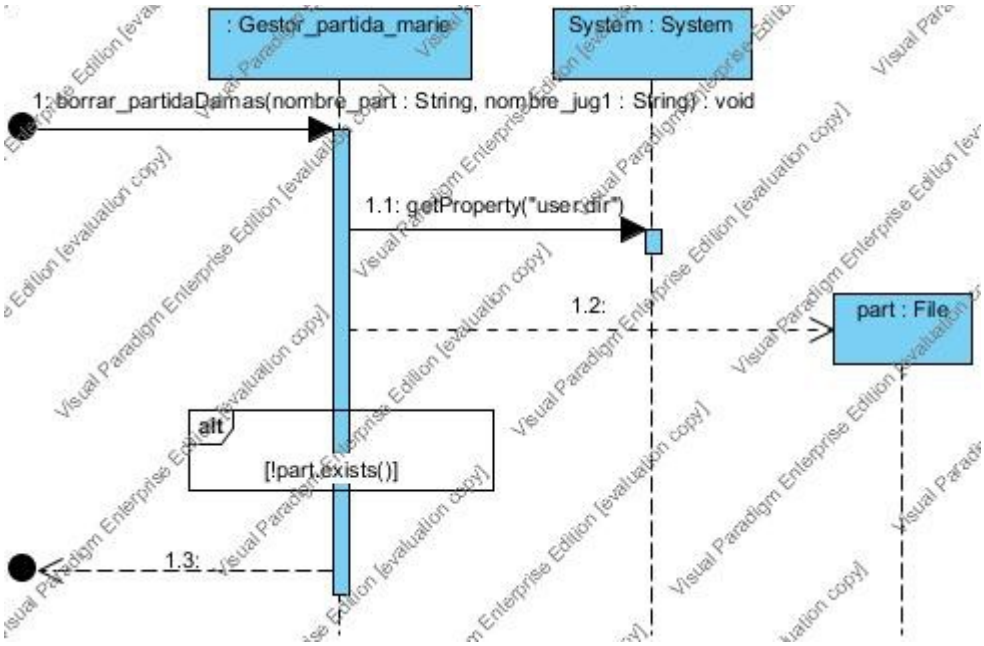

: Gestor_objetos
ardar_objeto(obj : Object, desti : String) : void
1.1:
fitxer : FileOutputStream
1.2:
c_sort : ObjectOutputStream
1.3: writeObject(obj)
1.4: close()
1.5:

5.7.3 Cargar partida

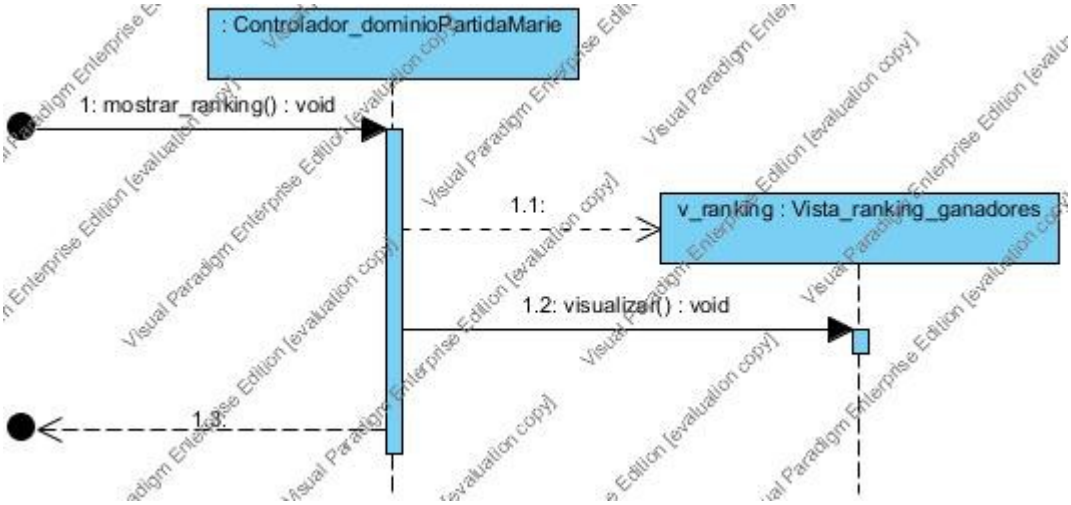


5.7.4 Borrar partida

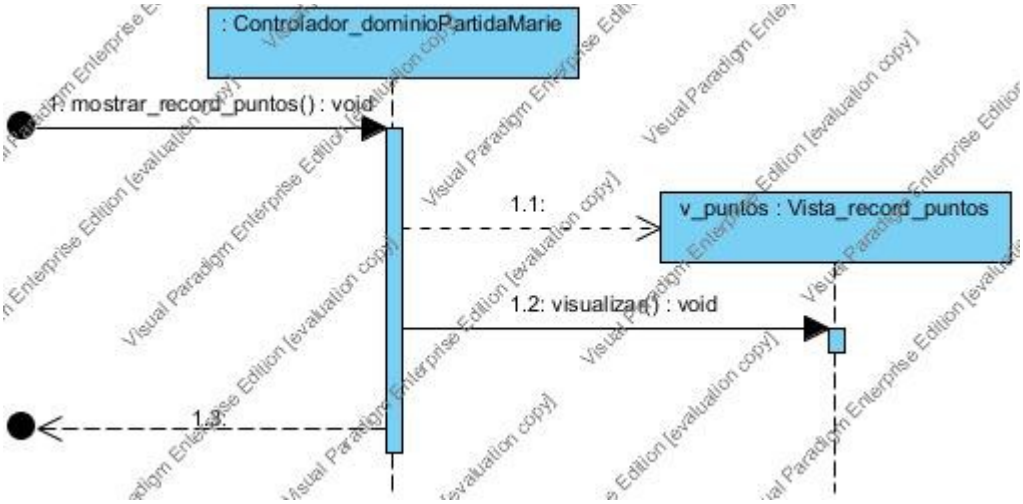




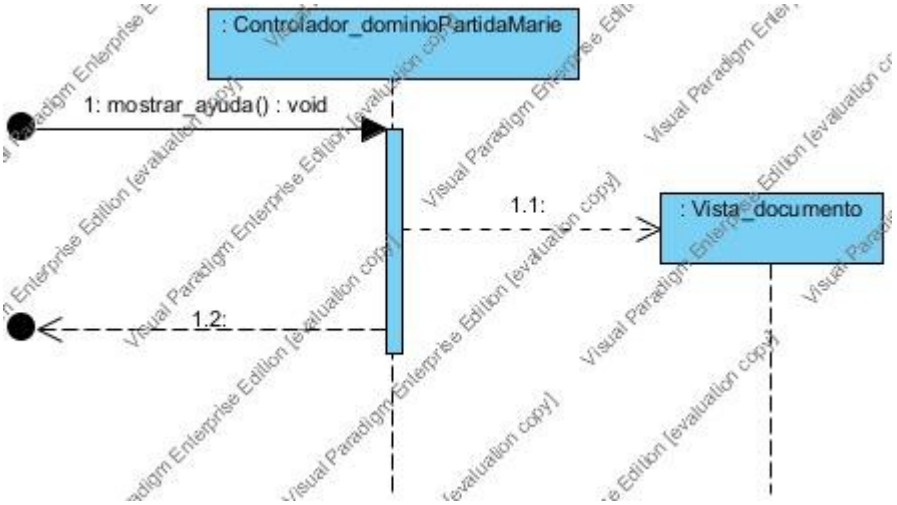
5.7.5 Mostrar ranking



5.7.6 Mostrar récord



5.7.7 Mostrar ayuda



6 IMPLEMENTACIÓN

6.1 Plataforma técnica

Para la implementación del proyecto, hemos decidido utilizar el lenguaje de programación Java, por varias razones.

Principalmente, ya que se pretendía que el programa resultante fuera multiplataforma, una de las primeras opciones a usar es precisamente Java, ya que, debido a que su máquina virtual se puede ejecutar en diversas plataformas, es un lenguaje multiplataforma por excelencia.

Otros factores técnicos importantes a tener en cuenta, aunque que en general son muy frecuentes en los lenguajes más usados hoy en día, es que permite programar dentro del paradigma de la orientación a objetos.

Más allá de eso, es un lenguaje muy extendido y soportado, con muchos años de historia (la primera versión es de 1995), es muy fácil de encontrar apoyo, y existen gran cantidad de librerías disponibles para resolver cualquier necesidad.

Para este proyecto se ha usado el IDE de Netbeans, tanto por familiaridad con él, como por su calidad. Para la interfaz gráfica, se utilizará la librería gráfica Swing y AWT, dado que es la principal herramienta para desarrollar interfaces gráficas para Java, y nos proporciona todos los componentes necesarios para el proyecto. Los diálogos se implementado escribiendo el código a mano como se hizo en su dia en PROP sin la ayuda de WindowBuilder (editor que permite crear interfaces Swing de manera visual fácilmente).

Para multijugador se ha usado sockets, no se ha utilizado UpnP, en su lugar se utiliza el puerto 80 para evitar el bloqueo de puertos de los routers actuales ya que dicho puerto siempre esta abierto.

6.2 Requisitos mínimos del sistema

- Sistemas operativos: Windows, GNU/Linux, o Mac OS X.
- Java: Versión 1.8 o superior.
- Pantalla que soporte una resolución de 1600*900 pixeles.
- RAM: 128 MB
- Espacio en disco: 124 MB para JRE; 2 MB para Java Update+el necesario para el sistema operativo+20MB para el juego.
- Procesador: Mínimo Pentium 2 a 266 MHz

Este proyecto ha sido realizado la mayor parte del tiempo en un portátil Toshiba sobre Windows 8 i5-3230M a 2,6GHz con 6 GB de memoria Ram. Se ha trabajado también en 2 ordenadores más y bajo Ubuntu también.

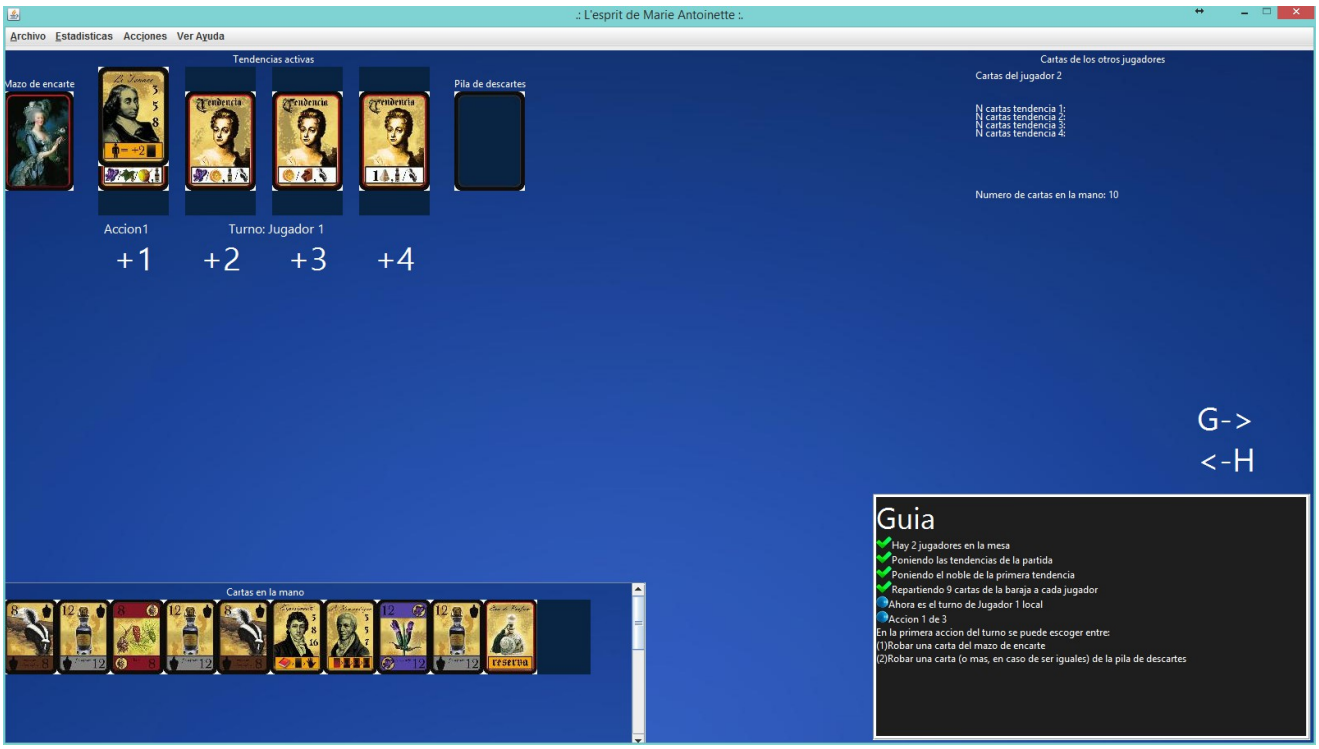
Representa que los requisitos mínimos son los mismos de Java 1,8, aunque hoy en día cualquier sistema operativo que el lector quiera usar supera los requerimientos mínimos del java. És imporante que la resolución de la pantalla este puesta a 1600*900 pixeles para poder visualizar correctamente el juego

6.3 Implementación de la capa de presentación

Se muestran ha continuación las vistas que ofrece el juego una vez ejecutado. Se empieza por la vista principal del juego una vez se ha empezado una partida. Se puede observar en esta sección todas las vistas así como la manera de acceder a ellas, y pasar de una vista a otra. la programación de la interfaz con java usando SWING (librería creada a partir de java AWT). Todas las vistas fue programada a mano sin ningún editor tal y como en su momento se recomendaba en Prop (Proyecto de programación). Es decir que cada uno de los elementos de todas las vistas han sido implementados mediante código escrito, configurando por escrito todos y cada uno de los parámetros necesarios para cada uno de los elementos. Es decir que el proyectista conoce perfectamente funciones como .setBounds entre muchas otras. El proyectista considera que esta manera de programar las vistas es demasiado lenta, ya que escribes todo el código a mano. Una ventaja es que implementas la clases vistas a el gusto del programador, pero en general el proyectista no tiene claro si esta política de programación de vistas es mejor o peor que hacerlas con un editor que te auto-genera el código automáticamente.

Todas las vistas han sido implementadas con elementos clásicos de la librería SWING (botones, labels, etc..), el proyectista quiere hacer notar que el elemento más moderno son la ayuda, la guía y el historial. Tanto la guía como el historial fueron rediseñados y reimplemantos (al principio eran otro elemento mas sencillo, ahora están hechos con HTMLeditKit y HTMLDocument), son 2 pequeñas pantallas que muestran código HTML. Estos elementos hace unos años no existían aún en ninguna librería java y no se podía mostrar código HTML (problema encontrado en una asignatura de la carrera no se si era PXCSO, para solventar el problema en aquel entonces el proyectista uso una librería creada por usuarios que no terminaba de funcionar del todo bien). Al ver que se habitan incorporado dichos elementos el proyectista decidió reimplementar la guía y el historial y usarlos.

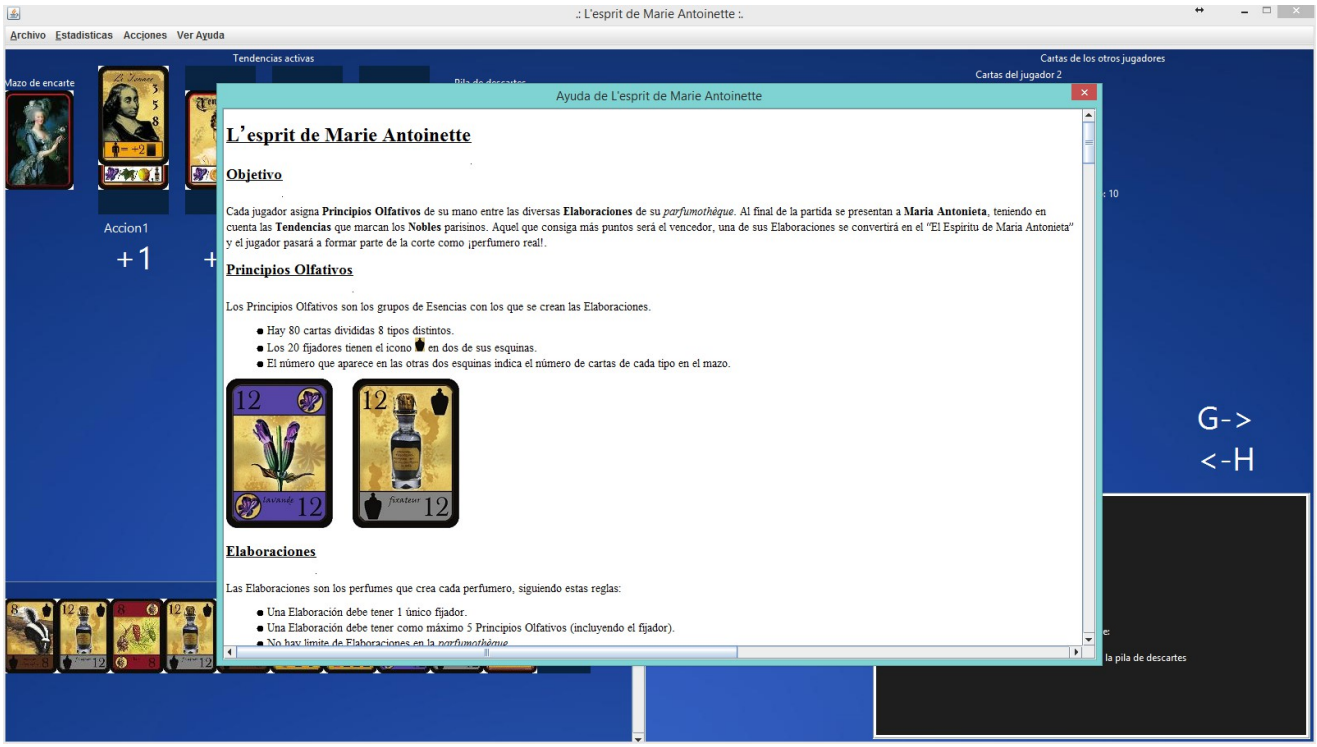
6.3.1 Vista principal



Vemos como el juego en la parte superior tiene las cartas comunes para todos los jugadores y abajo como se muestran las cartas de la mano del jugador que este jugado. En la parte inferior de la su derecha hay la guía del juego, en el mismo espacio se puede mostrar el historial de jugadas. En la parte superior derecha hay información acerca las cartas de todos los jugadores. Se observa como en la parte superior hay una barra con archivo, estadísticas,acciones y ayuda las cuales dan la posibilidad de ejecutar funcionalidades como nueva partida, reiniciar partida, cargar partida, salvar..., retroceder jugada, ver la ayuda, ver el ranking y el récord.

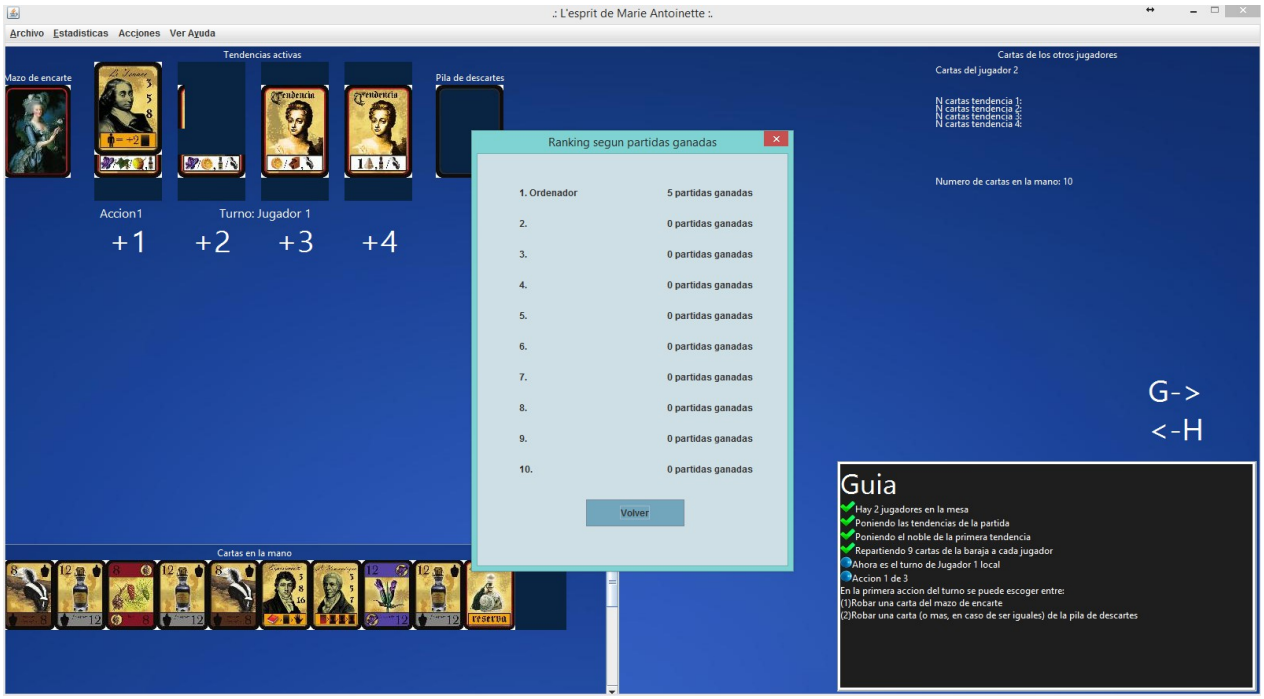
6.3.2 Vista de la ayuda

Se muestra a continuación la ayuda del juego. Es un archivo HTML.



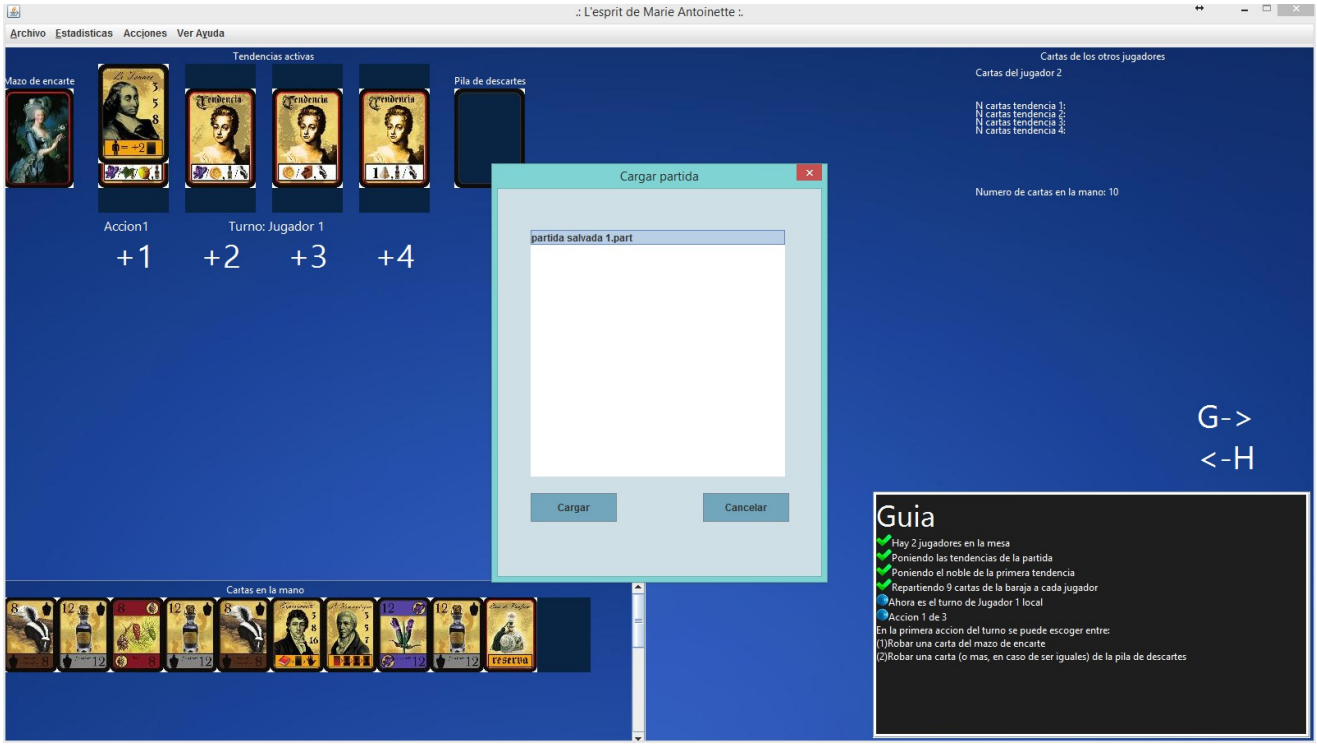
6.3.3 Vista del ranking

Se muestra a continuación una vista del ranking de partidas ganadas. Arriba de todo esta el jugador con más partidas ganadas.



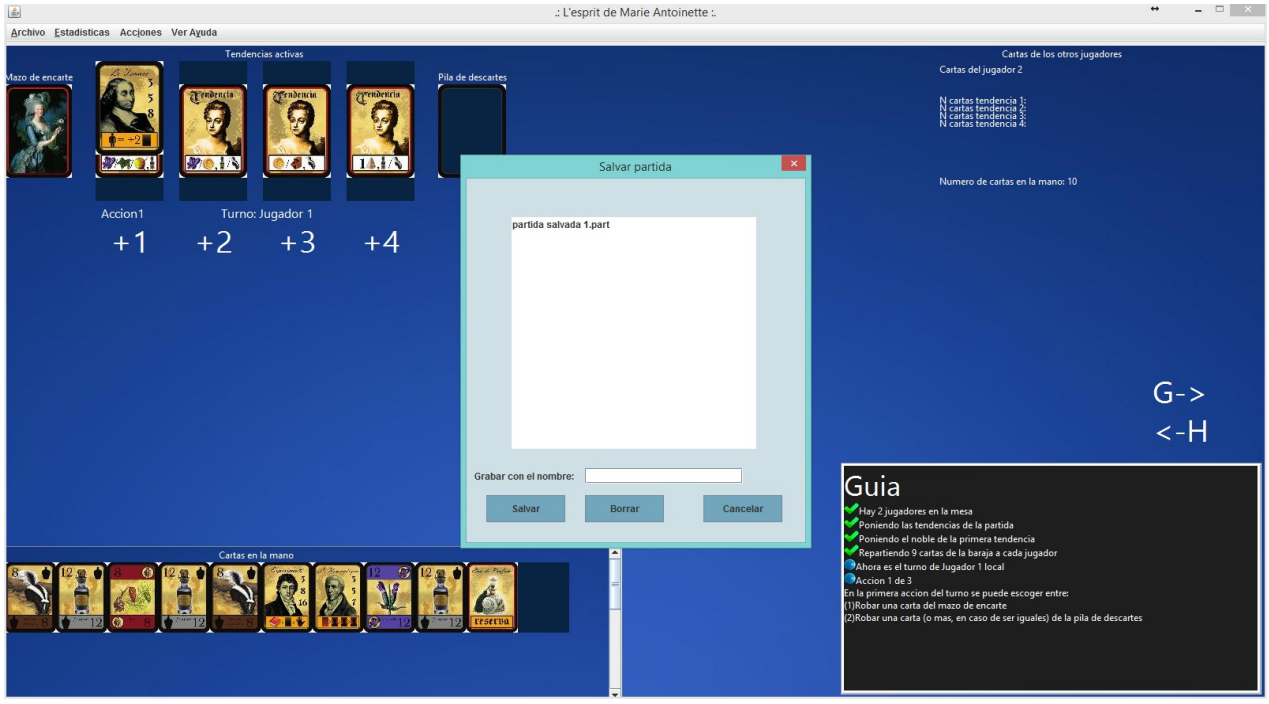
6.3.4 Vista del récord

Se muestra a continuación una captura del récord de puntos del juego. Como más arriba esta el jugador más alta es la puntuación del juego. Se muestra a continuación una vista de la pantalla de cargar partida.

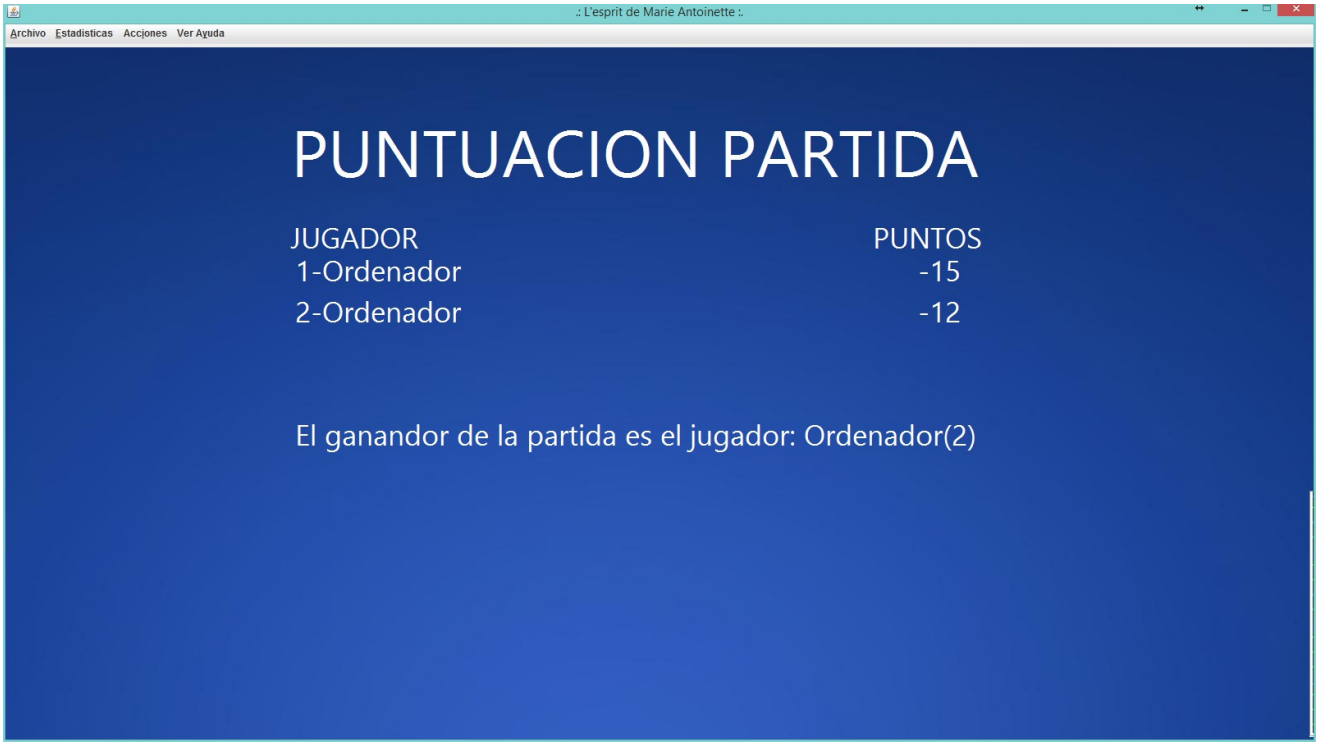


6.3.5 Vista salvar partida

Se muestra a continuación una captura de la vista de salvar partida.



Se muestra a continuación una captura de una vez finalizada la partida como es la pantalla final donde se muestran las puntuaciones

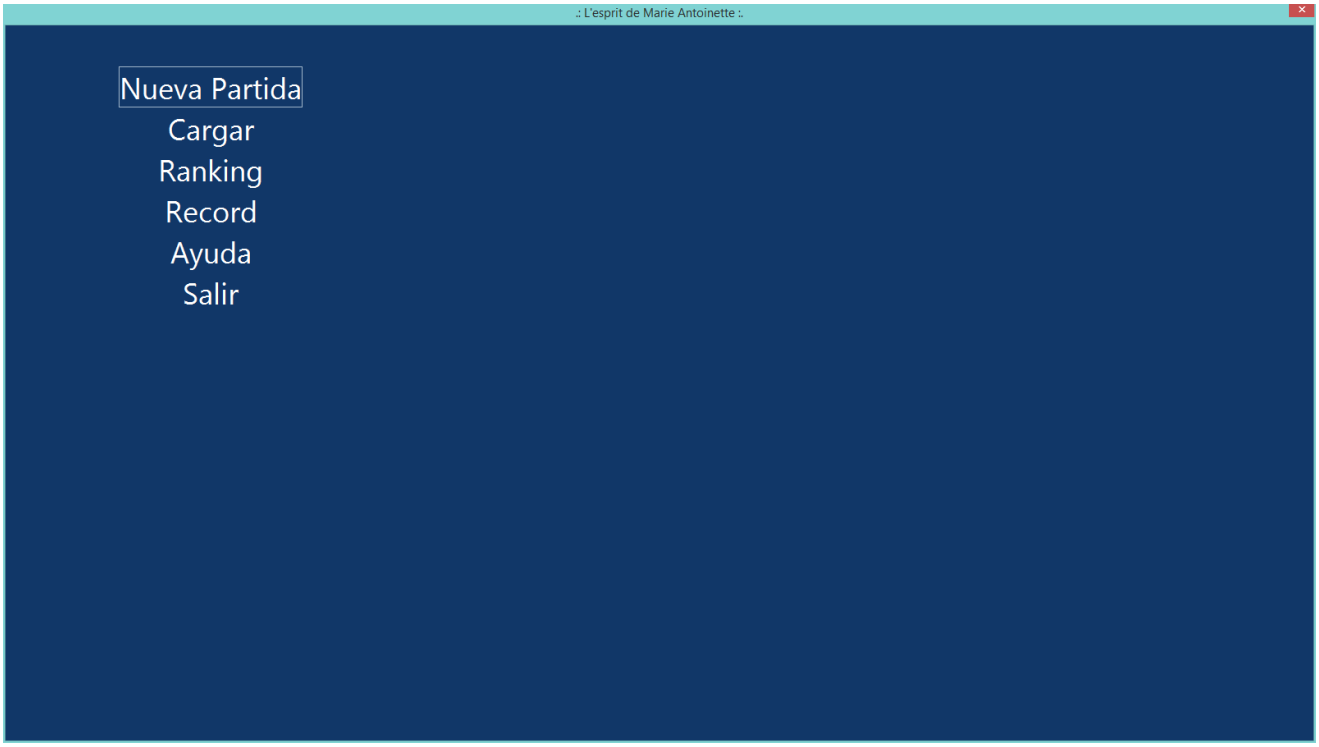


6.3.6 Pantalla de presentación

Se muestra a continuación la primera pantalla del juego que aparece. La pantalla de presentación que solo dura un par de segundos su visionado.

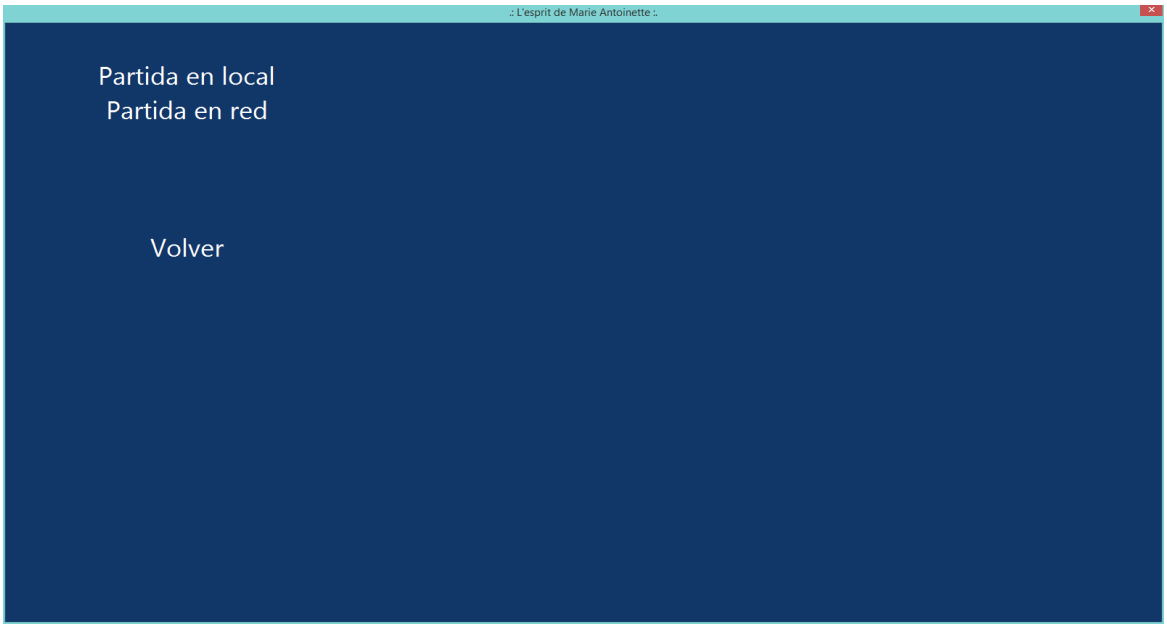
6.3.7 Menú inicial

Se muestra a continuación el menú inicial que aparece justo después de la pantalla anterior.

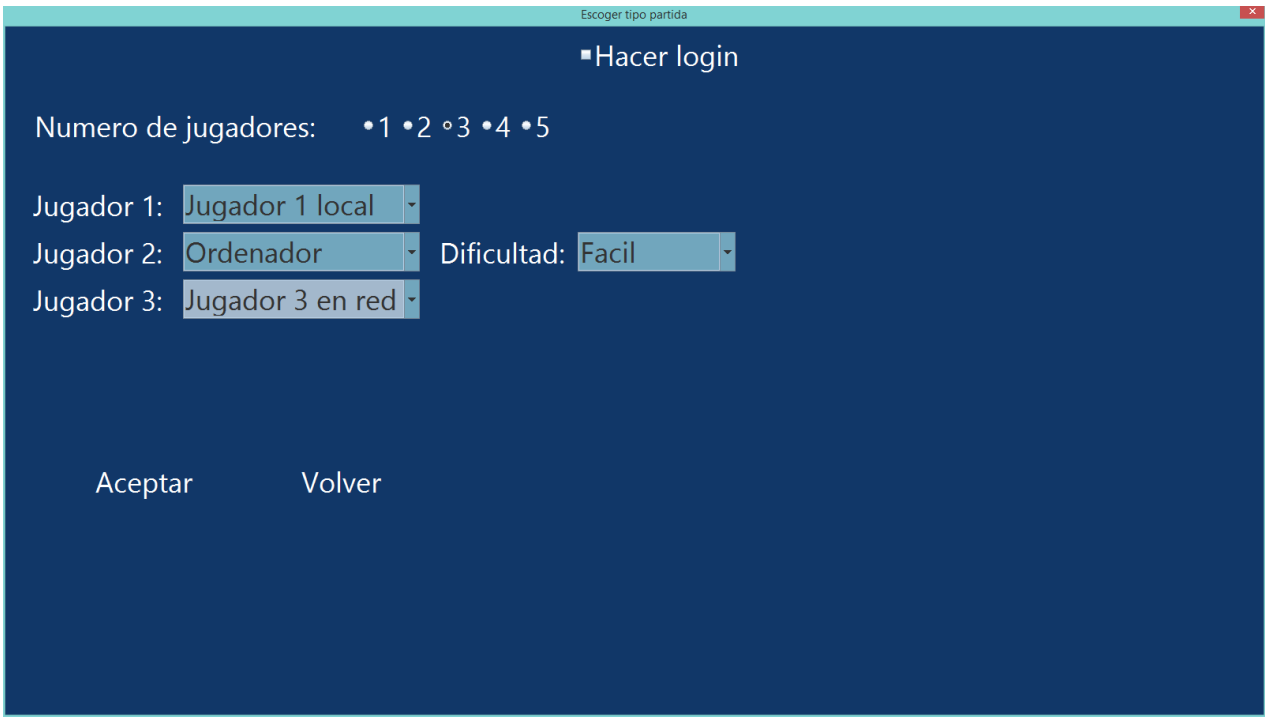


6.3.8 Menú nueva partida

Si en la pantalla anterior se pulsa en nueva partida aparece esta pantalla con las opciones de la nueva partida. Simplemente son 2: la partida local y la partida en red. Si se pulsa encima de volver, se retorna a la pantalla anteriormente mostrada.



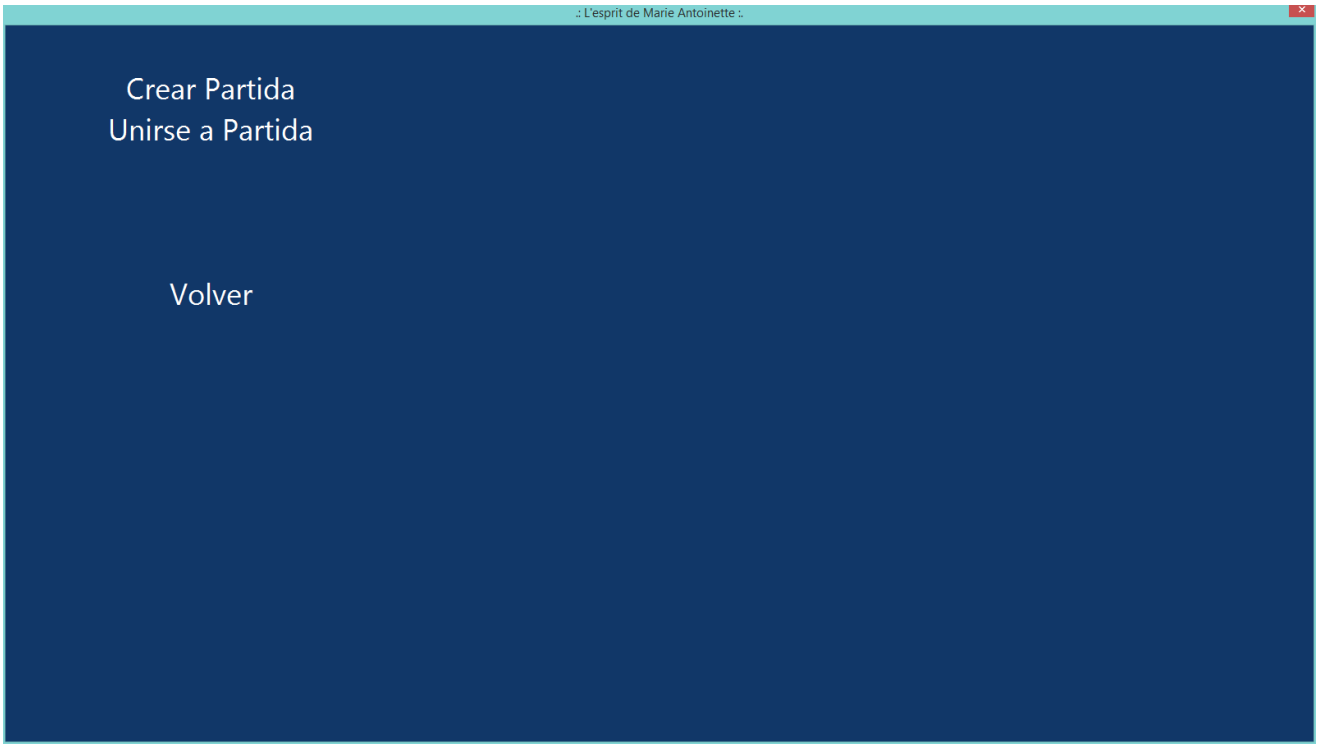
6.3.9 Menú partida local



Si en la pantalla anterior se pulsa sobre partida local aparece esta pantalla con las opciones de la partida local. El proyectista hace notar que un tipo de jugador puede ser en red. En cuyo caso se espera a que se conecte dicho jugador a este ordenador.

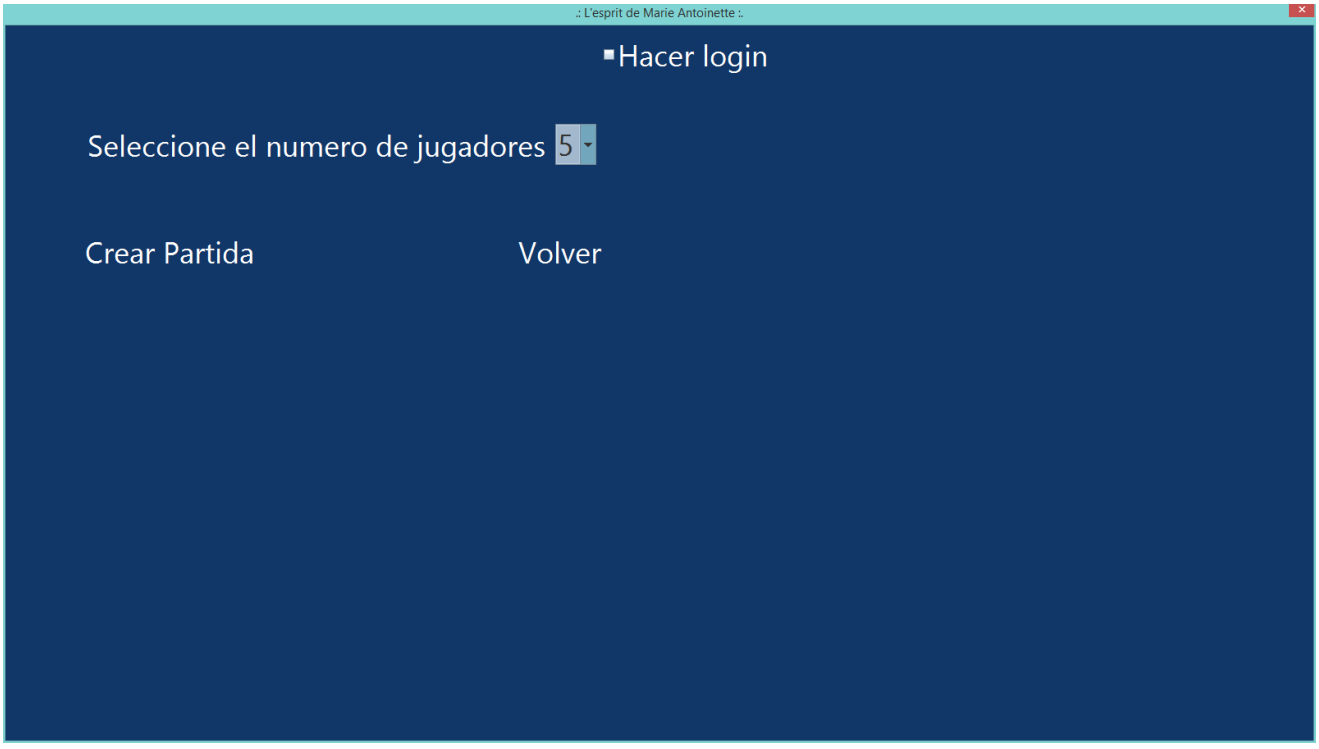
6.3.10 Menú partida en red

Si en el menú nueva partida se pulsa en partida en red aparece este menú. Se observa que se puede hacer de servidor creando la partida o de cliente.



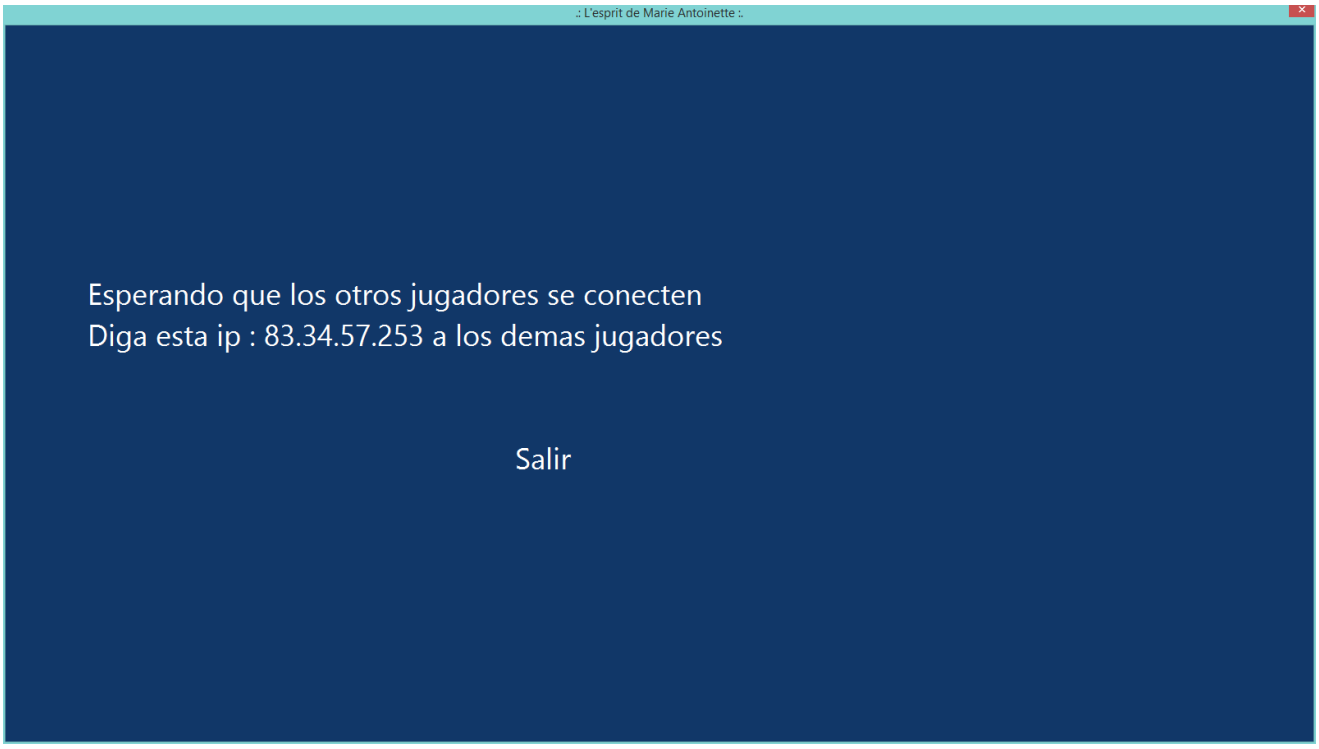
6.3.11 Menú crear partida.

Si en el menú partida en red se pulsa en crear partida aparece esta pantalla con el numero de jugadores ha crear la nueva partida.



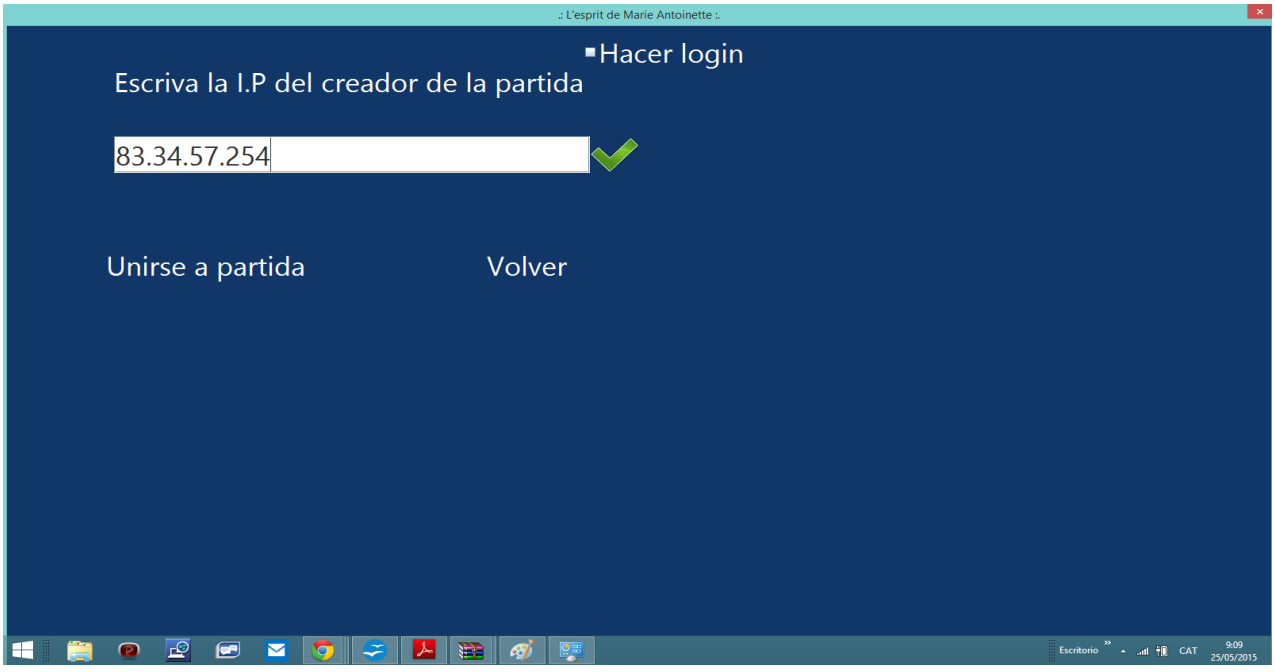
6.3.12 Menú crear partida servidor

Si en el menú crear partida anterior pulsamos sobre crear partida se visualiza esta pantalla. Observamos como se muestra la IP del servidor para que sea comunicada a los demás jugadores.



6.3.13 Menú unirse a partida

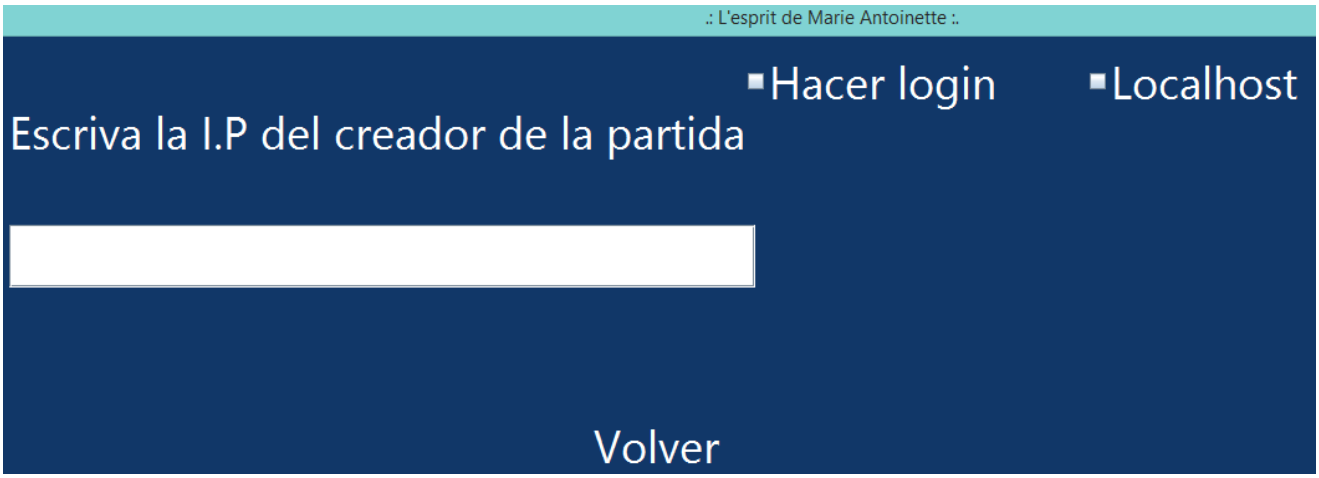
Si en el menú partida en red se pulsa sobre unirse a partida aparece esta pantalla donde debemos introducir la IP. Una vez introducida la IP aparece el botón unirse a partida.



6.4 Implementación de las comunicaciones

Se avisa al lector que es buena idea, antes de leer esta sección haber leído previamente la sección de diseño de las comunicaciones.

La implementación de las comunicaciones fue realizada de dos maneras en distintas fases del proyecto. Fue probada en local (localhost) y en red (una ip). El motivo por el cual fue programada en localhost fue porque al programador le resultaba más practico en la fase de testeo. De esta manera fueron probadas todas las cosas sin que el proyectista tuviera que conectarse a un ordenador remoto situado en otro casa cada vez. Por supuesto que primero fue probada la comunicación entre dos ordenadores remotos comunicados mediante sockets e IP. Una vez se comprobó que la comunicación remota funcionaba fue pasado a una comunicación localhost y finalmente una vez desarrollado todas las comunicaciones y testeado todo, fueron unidas dichas implementaciones (ahora te puedes conectar mediante localhost o te puedes conectar a un ordenador remoto situado en otra casa) sin cambiar código java, solo hace falta pulsar en localhost en la pantalla de unirse partida red para conectarse sobre un mismo ordenador y desclicar dicha casilla si se desea conectarse a un ordenador remoto situada en otra casa. Esta implementación que une dichos modos fue hecha en una fase muy tardía del proyecto, de hecho fue realizada mientras el proyectista escribía estas lineas. Mostramos la pantalla que se usa para unirse a una partida en red para entender lo redactado anteriormente.



Ahora se explicara como se implementa dicha comunicación. Este es el código de un cliente que desea a unirse a una partida creada por un servidor. Como se aprecia se necesita un socket, un puerto , y una ip.

Encaso de conectarse al mismo ordenador se pone localhost al socket en lugar de una ip.

```
public Socket[] socket=new Socket[5];


String s_ip=v_unirse_red.consultar_ip();//aqui se coje la ip escrita por el usuario
boolean localhost=v_unirse_red.b_hacer_localhost();
try {
if (localhost==true)
	{
	socket[0]=new Socket("localhost",80);//aquí se conecta
	}
else
	{
	//socket[0]=new Socket(s_ip,38521);//aquí se conectaría posible puerto utilizado
alternativo
	socket[0]=new Socket(s_ip,80);//aquí se conecta
	}

} catch (Exception e)
{
Vista_error v_error=new Vista_error(new JFrame(), "El jugador que hace servidor no esta a
la espera de clientes");
v_error.visualizar();
break;
}
```

Mostrado el código del cliente ahora se muestra parte del código simplificado del servidor.Se destaca que para leer un objeto se usa un ObjectInputStream obtenido a partir del socket conectado. Con la llamada (Jugador)ois.readObject() por ejemplo se puede leer un jugador enviado.

```
ServerSocket serversocket;
serversocket=new ServerSocket(80);
socket[0]=new Socket();
while ((i_jugadores_x_conectados<=numero_jugadores_x_conectarse))
		{
		socket[i_jugadores_x_conectados]=new Socket();
		System.out.println("Esperant una conexio"+i_jugadores_x_conectados);
		socket[i_jugadores_x_conectados]=serversocket.accept();
		//se tendria de copiar la informacion del jugador conectado
		//lectura objecto
		InputStream in=socket[i_jugadores_x_conectados].getInputStream();
		ObjectInputStream ois =new ObjectInputStream(in);
		jugadors[(i_jugadores_x_conectados)] =(Jugador)ois.readObject();
		jugadors[(i_jugadores_x_conectados)].modificar_es_online(true);
		System.out.println("Jugador leido");
		//enviar el numero jugador que es
		DataOutputStream  salir= new
	DataOutputStream(socket[i_jugadores_x_conectados].getOutputStream());
		salir.writeInt(i_jugadores_x_conectados);
		//guardamos el socket;
		System.out.println("Un jugador se ha conectado");
		i_jugadores_x_conectados++;
		}
```

Para finalizar se muestra como enviar un objecto por ejemplo el objecto partida que envía el servidor a los

clientes. Observamos que para enviar un objecto se usa el `ObjectOutputStream;` Luego con la función

`oos.writeObject((Partida_marie) partida)` se envía el objecto en este caso la partida.

```
// SOLO LO EXCECUTA EL SERVIDOR LA PRIMERA VEZ, CUANDO SE CREA LA PARTIDA
if (partida.consultar_jugador_1().consultar_fa_de_servidor() == true)
{
for (int i_jugadores_conectados = 1; i_jugadores_conectados <
partida.consultar_num_jugadores(); i_jugadores_conectados++)
      {
if (partida.consultar_jugador(i_jugadores_conectados).consultar_online())
            {
            OutputStream os = socket[i_jugadores_conectados].getOutputStream();
            ObjectOutputStream oos = new ObjectOutputStream(os);
            oos.writeObject((Partida_marie) partida);
            System.out.println("Partida enviada");
            }
      }

}
```

Si un jugador se desconecta en algún momento de la partida se comunica al resto y se termina la ejecución del juego avisando antes con un mensaje.

6.5 Implementación del dominio

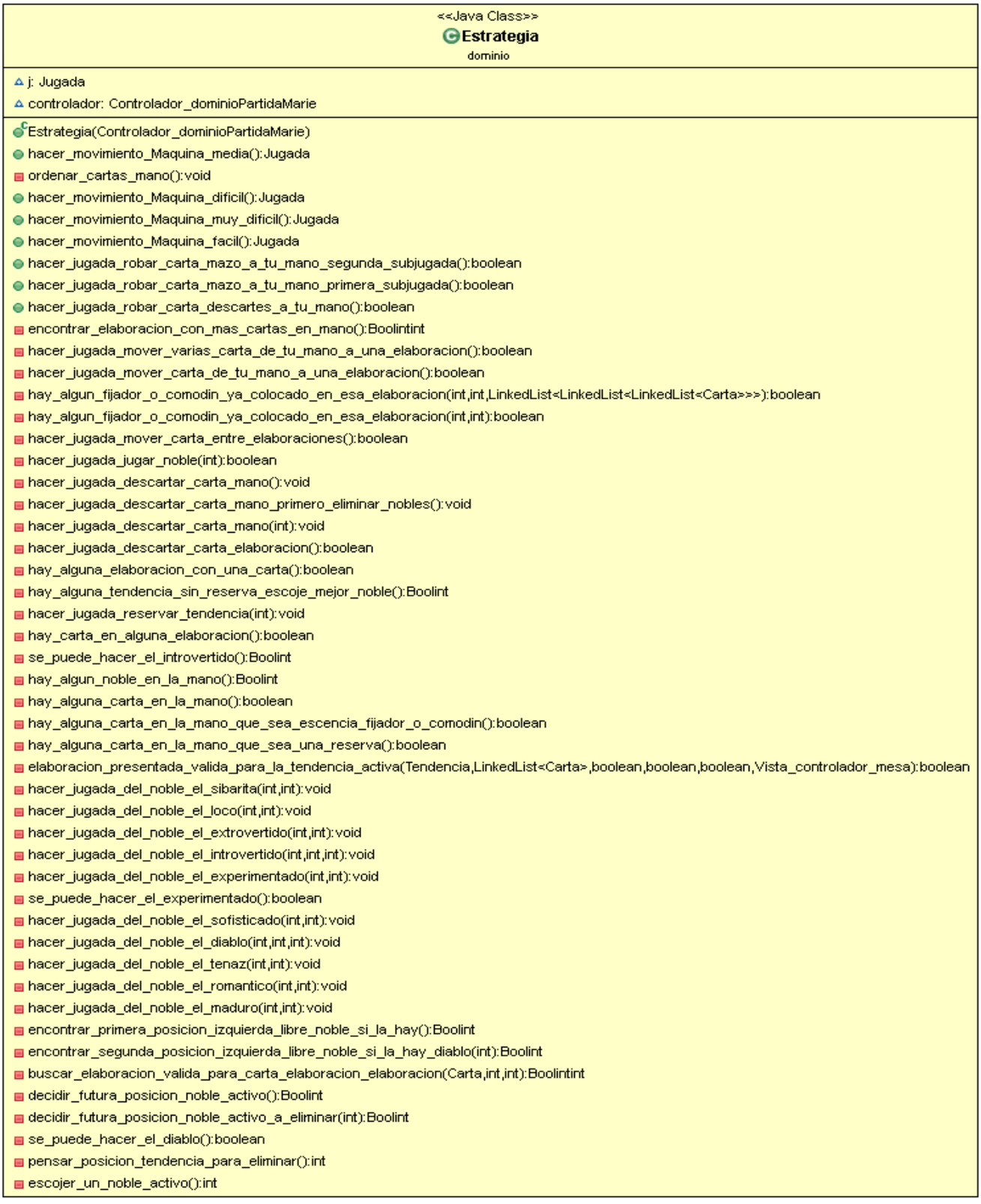
Toda la lógica de una partida en curso está controlada por la clase Controlador Partida Marie Antoinette , se encarga decidir **todo** lo que sucede en una partida, hacer cumplir las normas (toda y cada una de las normas), guiar al usuario, procesar los clicks o las jugadas del usuario (correctas y incorrectas), salvar partidas, cargarlas, mostrar, ranking o récord, retroceder jugadas, mostrar la ayuda, modificar la vista de la partida según lo que pulse el usuario, es decir que controla toda la pantalla que se muestra al jugar una partida, así como de la comunicación con la capa de presentación.

Controlador Partida Marie Antoinette tiene **toda** información del estado de la partida en todo momento ya que se comunica con la vista que muestra la partida llamada Vista controlador Partida (la parte del nombre que pone controlador fue puesto para destacar que esta vista es la que se comunica con el controlador de la partida, es decir es una vista muy importante, no controla nada como se puede ver si consultáis las funciones de dicha vista. Esta vista solo se encarga de mostrar la vista correcta al usuario en cada caso según toque quitando y mostrando los elementos que el controlador le ordena. Dicha vista se comunica con el Controlador Partida Marie para comunicarle que ha pulsado el usuario.

Hay tres tipos de jugadores distintos: jugador local, jugador en red y el ordenador. Si el lector quiere conocer mejor la estructura de los objetos del juego (por ejemplo la clase jugador, o partida) sus variables y funciones, debe ir a la sección de este documento llamado Diseño, y luego se le recomienda ir al Anexo donde puede leer cada una de las variables y funciones que componen cada una de los objectos de este proyecto. Los nombres de las variables y de las funciones son suficientemente claras para saber lo que hacen y se han evitado durante toda la implementación del proyecto nombres ambiguos como por ejemplo i, j en variables y funciones, y en su lugar se han puesto nombres largos o muy largos que explican tanto en variables y funciones que es lo que son y hacen. De esta manera de una manera rápida el lector podrá entender mejor y de forma precisa como han sido realizadas cada una de las clases. De este modo también se puede entender como ha sido implementado todo el proyecto de una manera más fácil. Como el lector entenderá no se explicara 2 veces en este documento algo tan complejo como la implementación entera del proyecto por eso se le insta a ver el apartado de Diseño y el Anexo encarecidamente.

6.6 Implementación de la Inteligencia Artificial

Se recomienda al lector leer antes de esto el apartado de Diseño de la Inteligencia Artificial antes de leer



este. Apreciamos a continuación una repetición de la clase estrategia dado la importancia teórica de la sección.

Se observa como esta clase devuelve una jugada, la jugada que devuelve es la pensada por la maquina como teórica mejor jugada en ese momento para el jugador. Dicha jugada va al Controlador Partida Marie y dicho controlador traduce y procesa toda la jugada. La jugada es visualizada poco a poco por la Vista de la partida de manera que el jugador persona pueda seguir toda la jugada pensada por la maquina como si hubiese sido realizada por un humano. Para que la jugada pensada por la maquina se pudiese ver como si una persona la estuviera haciendo se tubo que modificar la clase correspondiente de la vista de la partida añadiendo algunos paradas de tiempo (delays) para que el jugador persona tuviese tiempo de ver las jugadas encadenadas que realizan mas de un movimiento por jugada, esa sola es una de las modificaciones realizadas.

En este párrafo se explica como se piensa la mejor jugada brevemente. Según sea la dificultad del juego se ejecutara la función hacer jugada fácil, hacer jugada media, difícil o muy difícil. Puede apreciar dichas funciones en la clase mostrada anteriormente.

Cada función hacer_jugada llama a distintas funciones como por ejemplo hay algún noble en la mano luego escoger un noble activo. Luego por ejemplo esta función puede llamar a la función del noble escogido.

Por ejemplo la función hacer jugada descartar carta mano descarta cualquier carta en mano, esta función es para la inteligencia fácil. Pero sin embargo la función hacer jugada descartar carta mano primero nobles, esta función ya establece todas las prioridades de todos los tipos de cartas existentes y descarta de la forma más inteligente posible, es decir descarta las cartas que más penalizan primero por orden.

En esta sección no se explica cada función pues no se trata de eso solo de entender como funciona la inteligencia artificial implementada, además los nombres de las funciones ya son suficientemente claros.

La inteligencia fácil es mucho más simple que la inteligencia difícil. La inteligencia fácil llama a funciones más simples menos complejas, y además no sopesa bien que jugada realizara (la jugada que realiza es bastante aleatoria) es decir tanto puede hacer un tipo movimiento como otro , por ejemplo tanto puede descartar una carta de la mano como de robar una carta de la baraja. En cambio la dificultad muy difícil sopesa siempre que jugada mejor hacer, es decir la encoje no tiene nada de aleatorio. Además la jugada escogida su función esta mucho más elaborada de manera que se escoge mejor la carta que mueve.

Creo que dicho esto se puede entender como funciona la inteligencia artificial diseñada. Y recordamos

como el autor del juego dijo en este juego el minimax de por si solo no resulta lo suficientemente eficiente por lo que escoger resulta importante. En por ejemplo cuando se descarta una carta en la mano se entiende bien, que lo mejor es escoger las cartas que más penalizan descartarlas primero, en lugar de ir descartando una por una todas las cartas de la mano para luego ver cual evaluar cual escenario resulta mejor con un heurístico. En este juego hay muchísimas jugadas donde se puede escoger que carta resulta mejor o que jugada resulta mejor hacer en cada momento. De este modo se optimiza la inteligencia y resulta mejor incluso que ir generando todos los escenarios sin sentido alguno para luego evaluarlo con un heurístico. La mejor estrategia en mi opinión es la de generar todas las elaboraciones posibles porque así la posibilidad de poder presentar más elaboraciones aumenta y por la tanto la puntuación final. El hecho de generar pocas elaboraciones es una mala estrategia porque en este juego nunca se puede estar seguro que elaboraciones podrás presentar pues la ultima ronda, la ultima jugada para cada uno de los jugadores pueden hacer variar totalmente la fase de puntuación y quedarte sin puntuar. Es por eso que escoger jugadas, cartas, es buena estrategia. Este juego presenta muchas jugadas pero en realidad la estrategia general del juego es bastante simple (comparado con la cantidad de jugadas distintas y las reglas tan complejas que tiene) una vez se entienden todas las reglas.

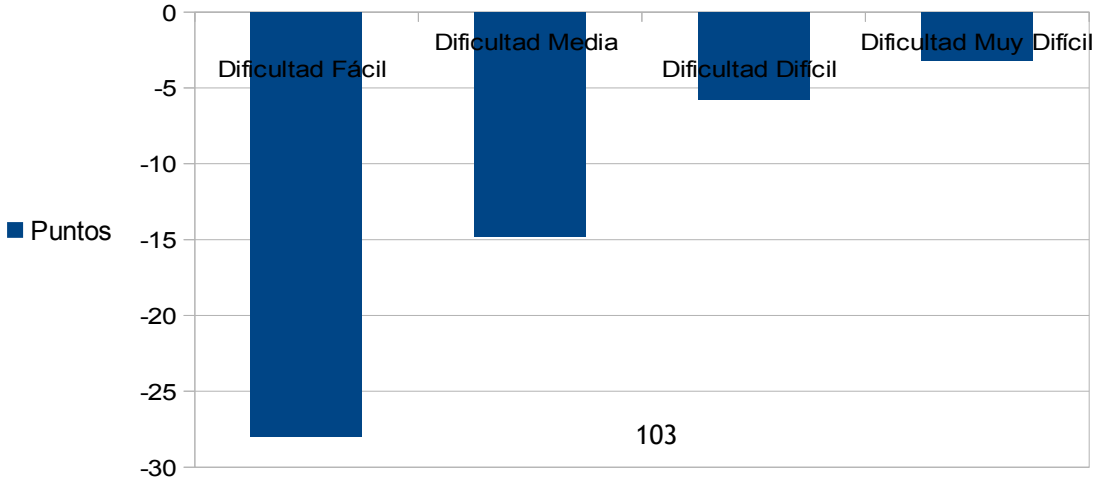
6.7 Comparación entre Inteligencias Artificiales.

A continuación les mostrare unas dadas estadísticas sobre las diferentes Inteligencias artificiales del juego. Vemos pues sin más preámbulos las puntuaciones medias que suelen sacar las diferencias inteligencias artificiales del juego.

Las pruebas han sido realizadas para 2 jugadores, esta tipo de partida hace que haya menos elaboraciones ha presentar y por lo tanto se suelen acumular más cartas en la mano, por lo tanto las puntuaciones tienden a ser mas bajas que con mas jugadores. En una partida con 5 jugadores se tienen a acumular menos cartas en la mano, pues se roban menos cartas del mazo y se pueden presentar más elaboraciones.

Comparativa entre Inteligencias Artificiales		
NIVEL DIFICULTAD	**PUNTUACIONES en puntos**	**PUNTUACION MEDIA (puntos)**
Dificultad Fácil	(-28,-23,-32,-29)	-28
Dificultad Media	(-12, -16,-19,-15,-12)	-14,8
Dificultad Difícil	(-4,4,-4,-19)	-5,75
Dificultad Muy Difícil	(-5,1,-6,-2,-4)	-3,2

Observemos ahora la información de la tabla anterior resumida en un gráfico. Se puede ver como la puntuación más alta es la obtenida por la dificultad muy difícil.



6.8 Implementación de la persistencia

Se recomienda al lector leer antes el diseño de la persistencia. Para salvar los objetos o clases a disco de forma permanente primero se deben de serializar, como se muestra a continuación:

```
public class Partida_marie  extends Partida implements Serializable
```

Una vez serializado el objeto se puede crear la carpeta donde se guardara el objeto si no esta ya creada.

```
String Directorio_partida = System.getProperty("user.dir");
Directorio_partida=Directorio_partida+"/marie_antoinette_files/Partida/";
File dir_partida = new File(Directorio_partida);
if (!dir_partida.exists())
{
dir_partida.mkdir()
}
```

Luego basta con salvar la partida. Para hacer esto todas las clases encargadas de salvar objetos ya sea una partida, un jugador, un ranking o un récord llaman a la funcion guardar objeto.

```
Partida_marie p;
arxiu = Directorio_partida+nombre_part+".part";
guardar_objeto(p, arxiu);
```

Ahora entremos en la función guardar objeto para ver definitivamente como se guardan los objetos.

Finalmente se observa como se guarda con la función writeObject().Observamos que se llama a dicha función usando un ObjectOutputStream.

```
Object obj, String desti
FileOutputStream fitxer = new FileOutputStream(desti);
ObjectOutputStream c_sort = new ObjectOutputStream(fitxer);
c_sort.writeObject(obj);
c_sort.close();
```

Para cargar un objeto ya previamente salvado mostramos directamente el código de cargar objeto simplificado.

```
Object cargar_objeto(String origen) throws Exception {
FileInputStream fitxer = new FileInputStream(origen);
ObjectInputStream c_entr = new ObjectInputStream(fitxer);
Object obj = c_entr.readObject();
c_entr.close();
return obj;
```

Se observa que se carga con la función readObject. Observamos que se llama a dicha función usando un ObjectInputStream.

7 PLANIFICACIÓN

En esta sección llamada planificación se explica tanto la planificación inicial, como la planificación que realmente fue. Luego se hace una estimación económica del coste del proyecto.

7.1 Comparativa entre planificación inicial y final

Para comparar la planificaciones , primero se muestra la planificación inicial y luego se muestra la planificación real que al final tubo lugar. Luego se muestran las conclusiones.

Se muestra pues primero la planificación inicial.

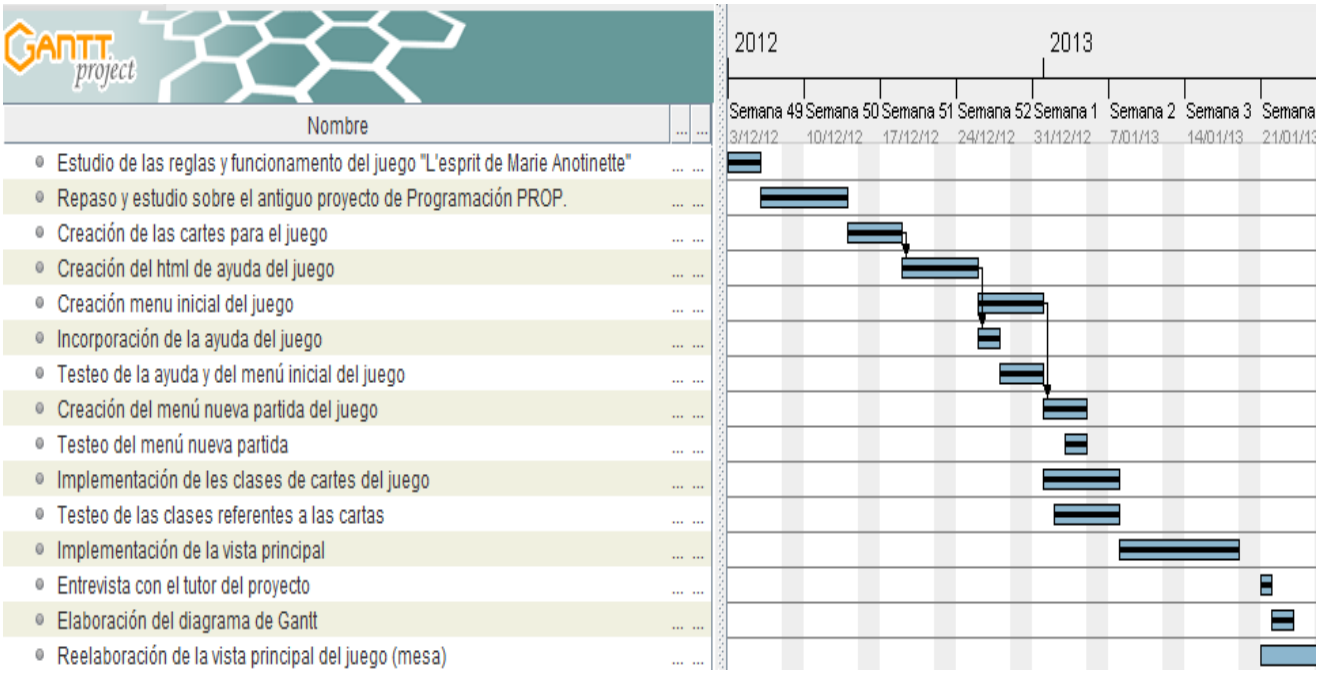
7.1.1 Planificación inicial

En este apartado se muestra la planificación inicial del proyecto tal cual fue hecha a mitad de la realización del proyecto, se muestra primero el diagrama de Gantt de las tareas hasta esa fecha realizadas y se luego el diagrama de Gantt de las tareas que estaban pendientes de realizar con su estimación. La siguiente planificación se hizo en la etapa incial-intermedia del proyecto.

7.1.1.1 Diagrama de Gantt:

Se puede apreciar en el diagrama de Gantt el orden y duración de cada tarea. La duración se expresa en días de trabajo de 8 horas aproximadamente, aunque al principio las horas por día trabajadas eran inferiores a 8. Las flechas indican dependencias entre tareas.

Horas totales trabajadas hasta ahora:20 días laborables del mes de diciembre*4 horas diarias +20 días del mes de enero*5 horas diarias+20 días del mes de febrero*8 horas diarias+13 días del mes de marzo*8 horas diarias=444 horas totales;



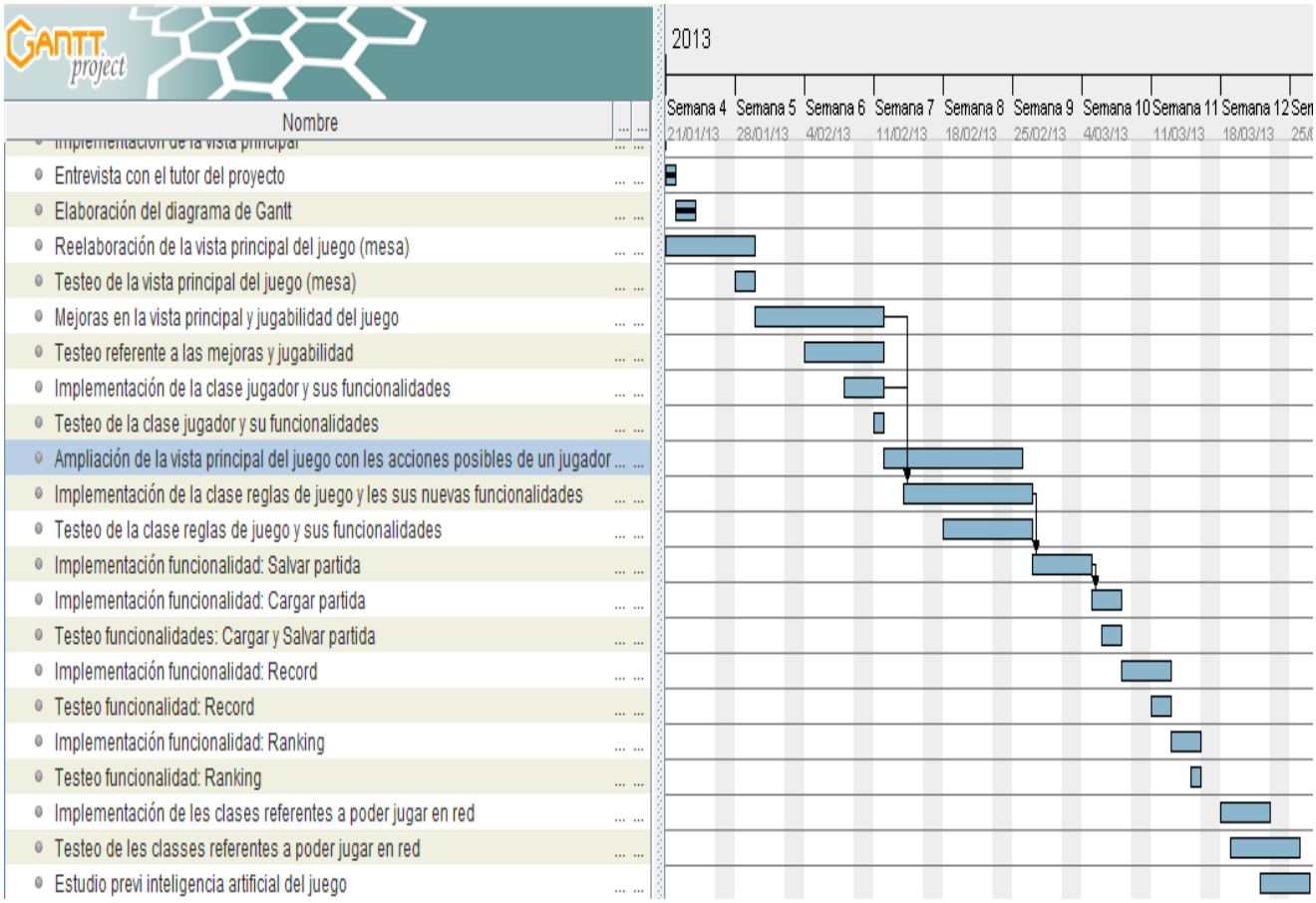
7.1.1.1 Planificación del trabajo a realizar para finalizar el proyecto

El trabajo restante para finalizar el proyecto, eran básicamente tres tareas: hacer posible que puedan jugar dos jugadores mediante Internet, hacer posible que un jugador pueda jugar contra la inteligencia artificial escogiendo diferentes niveles y escribir la memoria. Finalmente se prepararía la presentación del proyecto.

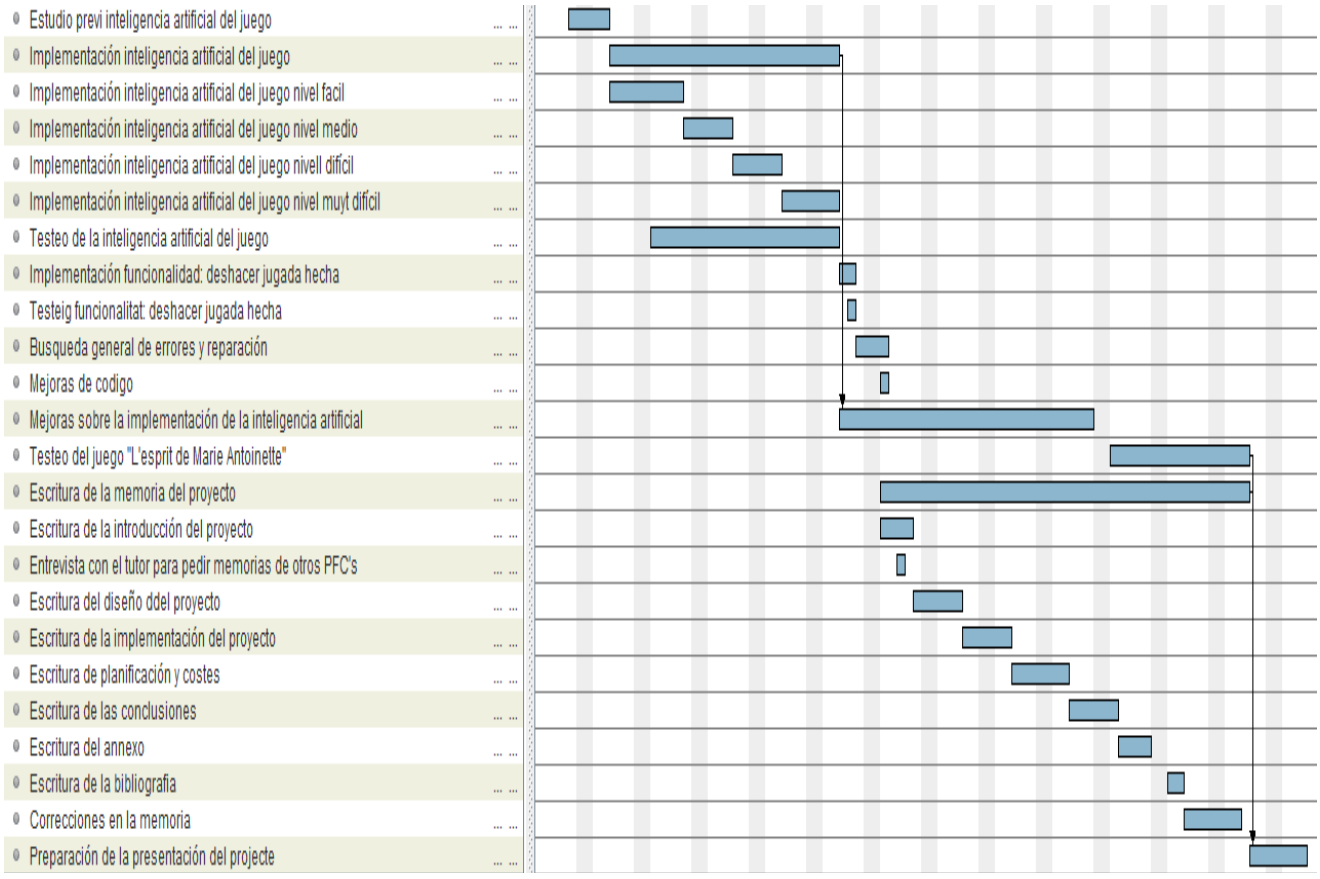
Referente a las tareas de implementar una inteligencia artificial del juego y la tarea de hacer posible poder jugar en red, son para el proyectista, tareas que se prevén que pueden alargarse más de lo calculado debido a la complejidad y a la poca experiencia en el mismo tipo de funcionalidades que ha tendido el proyectista anteriormente

7.1.1.2 Diagrama de Gantt:

Se puede apreciar en el diagrama de Gantt el orden y duración de cada tarea. La duración se expresa en días de trabajo de 8 horas. Las flechas indican dependencias entre tareas.



Se muestra seguidamente la continuación del diagrama de Gantt.



Estimación horas totales restantes para terminar el proyecto: 7 días laborables del mes de marzo*8 horas diarias +20 días del mes de abril*8 horas diarias+20 días del mes del mayo*8 horas diarias+13 días del mes de junio*8 horas diarias=480 horas totales;

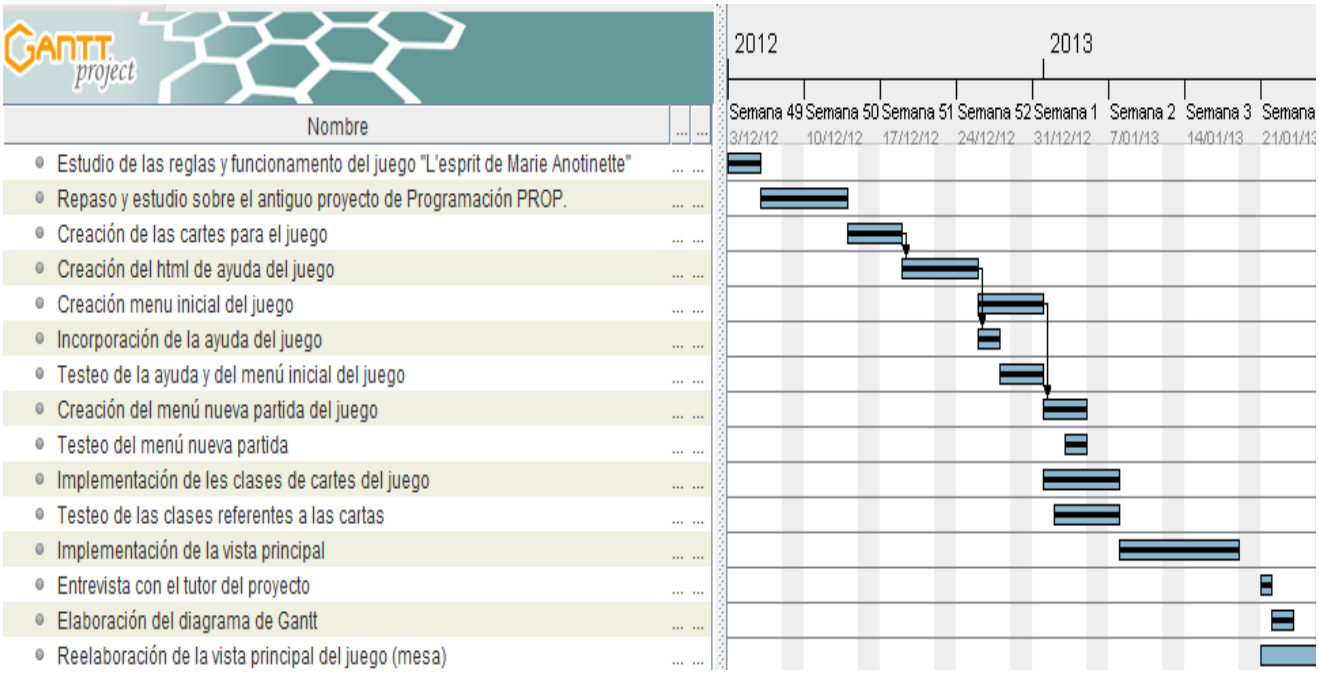
7.1.2 Planificación final

Una vez terminado el proyecto se les muestra a continuación la planificación real del proyecto. Se va a observar como las tareas duran más en general de lo previsto e incluso se retoman tareas ya hechas para mejorarlas. En conjunto el proyecto tarda más de lo planificado en un principio. El primer diagrama de Gantt que se muestra es el de las tareas realizadas y por lo tanto resulta invariable pues ya estaban hechas.

Luego se mostrara la continuación del diagrama de Gantt real completo siguiendo una continuidad y por partes ya que no cabe todo en una sola imagen.

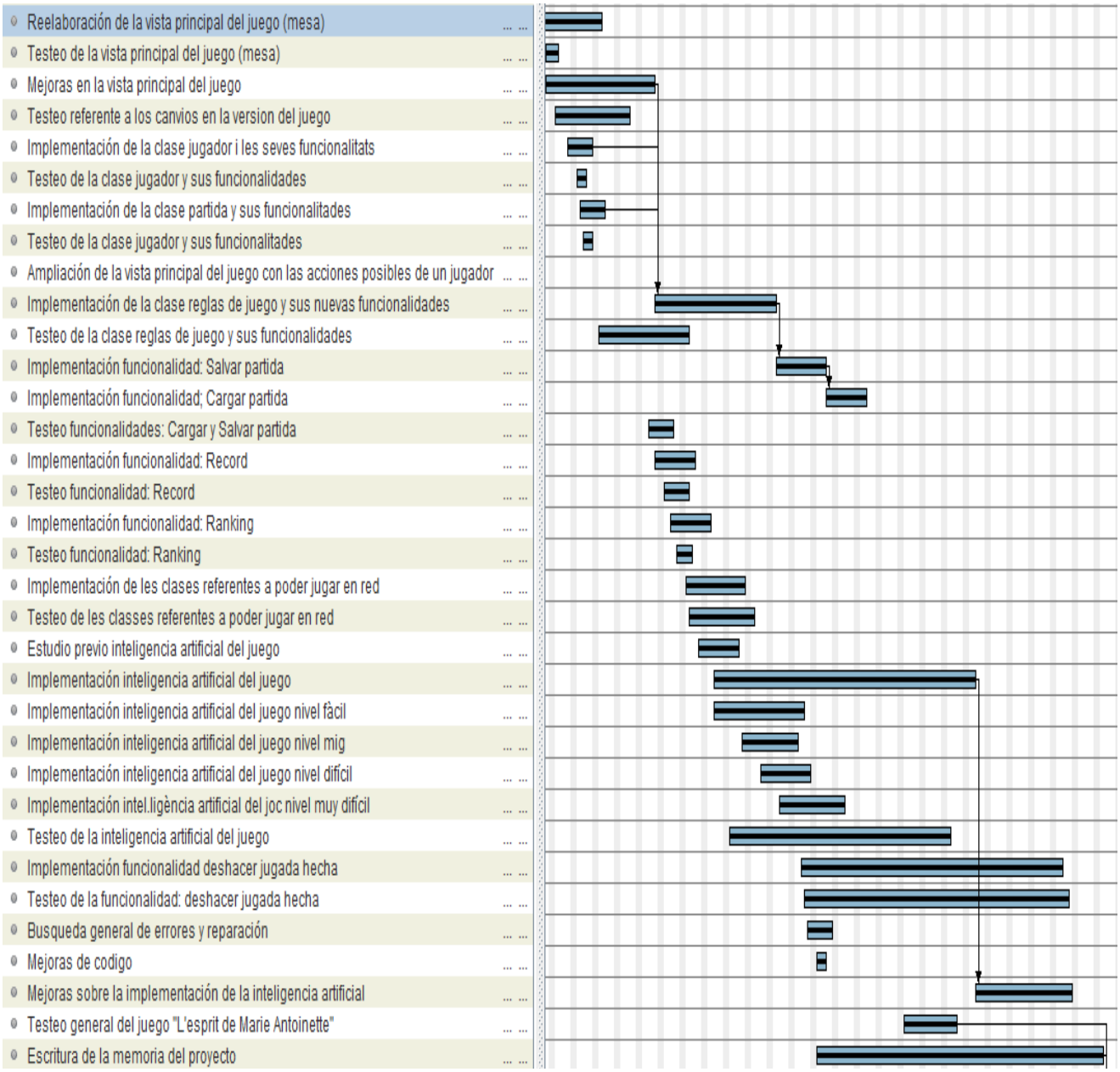
7.1.2.1 Diagrama de Gantt:

El siguiente diagrama de Gantt que se muestra a continuación es exactamente igual que el mostrado en la planificación inicial, el motivo es que se trata de tareas ya hechas en el momento de hacer el diagrama de Gantt de la planificación inicial. El motivo por el cual se repite esta parta de el diagrama de Gantt es el de mostrar el diagrama de Gantt final completo de principio a fin.

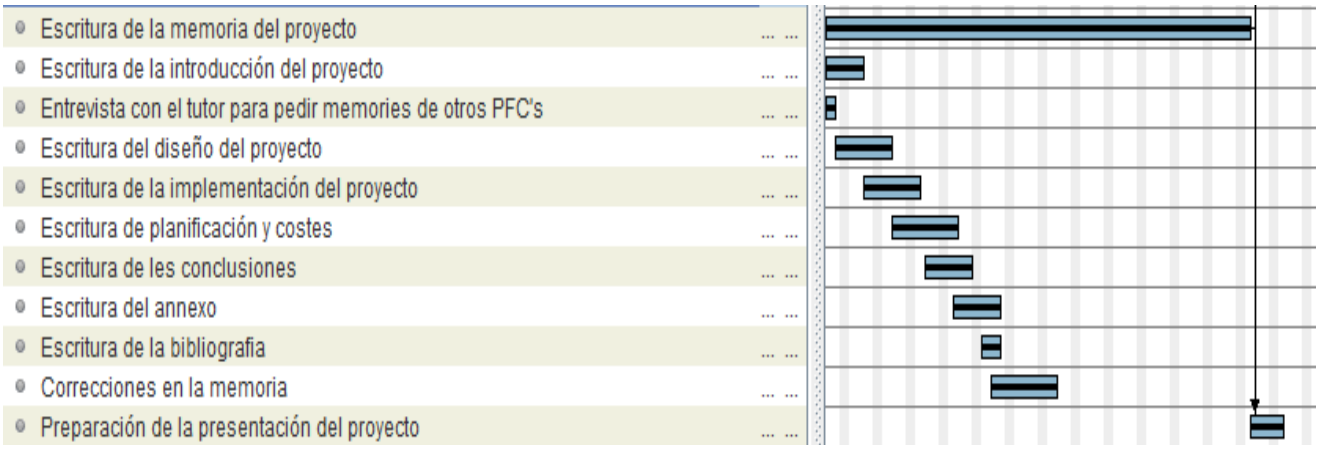


El trozo de diagrama anterior como se ha dicho muestra las tareas ya hechas en el momento de hacer el diagrama de Gantt inicial. A continuación se muestra la continuación del diagrama anteriormente mostrado con la planificación real tal cual ha sido. Esta continuación a diferencia del anterior difiere de la

planificación inicial. Se muestra pues la continuación del diagrama.



Se muestra ahora la continuación del diagrama anterior.



Entre el diagrama inicial de Gantt y el resultante al final de la realización del proyecto la diferencia principal es que las tareas que restaban pendientes se tardó unas tres veces aproximadamente más de lo inicialmente previsto. Aparte algunas no inicialmente previstas surgieron como algunas mejoras y reparación de errores. Estas son pues las diferencias principales.

7.1.3 Conclusiones

Las conclusiones finales principales el proyectista ya era consciente de ellas antes de realizar este proyecto, y básicamente son que lo que se tiene planificado realizar en una unidad de tiempo acabas tardando 3 o 4. Por ejemplo, en este proyecto después de la hacer la planificación inicial cuando solo faltan por hacer la inteligencia, hacer el juego funcional en red, y la funcionalidad de retroceder jugada, se decidió mejorar las algunas características anteriores ya realizadas para hacer el juego más fácil y completo, como por ejemplo en jugadas como mover castas de la mano de un jugador a una elaboración, se decidió mejorar la jugada y detectar automáticamente cuando se puede o no continuar dicha jugada y si no se puede, se procede a terminar la jugada y pasar a la siguiente jugada. De este tipos de mejoras han habido varias. Debido a cumplir los plazos de tiempo algunas tareas, después de no poder presentar el proyecto en su primer semestre matriculado, luego al disponer de más tiempo pude hacer varias mejoras además de terminar el juego.

Otro problema que retraso los tiempos y la planificación del proyecto fue la aparición de un bug de java que ha complicado la vida al proyectista de mala manera.

El proyectista también se ha dado cuenta que independientemente de problemas que puedan surgir en el trabajo, pueden surgir también problemas familiares y personales. Algunos de estos problemas pueden ser resueltos y otros simplemente los tienes que pasar.

Finalmente el proyecto se ha ido alargando en el tiempo y ha habido un factor de desgaste personal, es decir no programas igual ni con las mismas ganas y energía el primer mes que el último, el rendimiento que no las horas de trabajo fue disminuciones a diferencia que las horas de trabajo.

El proyectista también se dio cuenta que tiene que adaptar su lugar de trabajo adecuadamente para que sea usado durante largas temporadas y horas para que este no produjese daño físico. Este factor siempre fue tomado a broma por el proyectista anteriormente y no se si por edad o por el hecho de mantener una postura incorrecta durante mucho tiempo delante el ordenador portátil, el proyectista empezó a sufrir dolores en el cuello realmente fuertes. Debido ha esto el proyectista se compró una mejor silla, uso una mejor mesa de trabajo y actualmente trabaja en el ordenador portátil con un teclado de ordenador de escritorio, el ordenador portátil elevado a la altura de los ojos y con una mesa de escritorio para ordenadores.

Otra cosa que el proyectista ya sabia, es que cuando una persona lleva larguísimos tiempos trabajado en algo sin descanso, el rendimiento va decreciendo a poco a poco hasta que al final aunque no quieras tomarte

un día sin trabajar, cuando llevas ya muchos días trabajando sin tomarte un día de descanso el cuerpo al final te obliga a tomarte un día de descanso aunque no quieras, ya que aunque estés delante el ordenador llega un punto después de muchos días de trabajo seguido encerrado en casa casi, que apenas haces nada. Cuando llegas al limite después de tomarte un par de días de descanso, se vuelve ha empezar a trabajar "con las pilas recargadas". La opinión del proyectista es que por eso existen los días de fiesta, no por las reclamaciones sociales. Todo el punto que ha trabajado realmente duro y sin descanso durante largos periodos de tiempo sabe que existe un limite físico de trabajo.

La conclusión final es que las tareas se alargan, la complejidad real del juego no se ve hasta que se trata de hacer-lo y las complicaciones simplemente surgen. Los diagramas de Gantt están bien pero simplemente hay que trabajar lo máximo físicamente posible cada día y tener en cuenta que muy difícilmente vas tardar a hacer la totalidad de la tarea, en el tiempo inicialmente establecido y pensado pocas veces va a cumplirse en su totalidad.

7.2 Valoración económica del PFC

La valoración económica del software viene dada por la suma de los costes del software con los costes en recursos humanos. A continuación se muestra dichos costes, primero los costes relacionados con el software utilizado para la realización del proyecto y luego los costes resultantes en recursos humanos debido al numero de horas utilizado y al valor económico puesto arbitrariamente a una hora de tiempo humano.

7.2.1 Coste del software

Para calcular el coste del software, se he tendido en cuenta que el software necesario para la realización del proyecto exista en versión libre, y gratuita. De este modo se minimiza el coste del proyecto.

Software	Función	Coste
Ubuntu	Sistema operativo	0 euros
Java JDK	Lenguaje de programación	0 euros
Netbeans	Entorno de desarrollo	0 euros
OpenOffice	Entorno de escritura para la documentación del proyecto	0 euros
Gimp	Programa para el retoque de imágenes	0 euros
Windows 8	Sistema operativo	119 euros
ObjectAid	Programa para la creación de los diagramas	0 euros
Gantt Project	Programa para la creación de los diagramas de Gantt	0 euros

Como se observa en la tabla anterior, no es necesario pagar por ningún programa para la realización del proyecto, ya que existen suficientes programas libres y gratuitos que cubren todas las necesidades. La excepción en costes, es el Windows 8 que se ha usado, que aunque no lo se haya comprado directamente, pues ya venia incluido en el coste del ordenador portátil, se le ha puesto el precio estándar que Microsoft pone a Windows 8.

7.2.2 Coste en recursos humanos

Para calcular el coste en recursos humanos se han dividido las diversas tareas realizadas en diversos tipos según la clase de tarea profesional (cada tipo de tarea podría ser realizada por un perfil distinto de profesional). Luego para cada tipo de tarea hay el numero de horas invertidas en ellas. Una vez fijado el precio hora de cada tipo de tarea se calcula el coste total resultante. Sin mas dilaciones mostramos pues la tabla para calcular el coste total del proyecto.

Tarea	Horas	Coste (Euros/hora)	Coste total (Euros)
Analista/Programador/Tester	1032 horas	8 Euros/hora	8256 Euros
Diseñador gráfico	264 horas	7 Euros/hora	1848 Euros
Documentador	176 horas	6 Euros/hora	1056 Euros
		Coste total	11160 Euros

El calculo final en Euros que se gastan en recursos humanos para la realización del proyecto es de 11160 Euros. Para llegar a este coste se ha utilizado el diagrama de Gantt final mostrado anteriormente y se ha establecido que aproximadamente (para hacerlo más comprensible para el lector se ha cogido el diagrama de Gantt y se han aproximado los tiempos en meses, o en 15 días, aquí se habla de meses, para hacer números redondos) se ha trabajado un mes en hacer la documentación, un mes y medio para hacer la interfaz gráfica funcional y 6 meses aproximadamente para hacer la programación y testeo del juego.

Luego como ya se ha explicado se han fijado unos precio/horas en Euros que aunque seguramente resultan inferiores a los reales, el proyectista ha querido fijar los precios que habitualmente él se ha encontrado que se pagan hoy en día para un recién titulado en según que trabajos. No dejan pues de ser precios hora reales para algunas empresas.

7.2.3 Coste total

El valor estimado del proyecto es el del coste en recursos humanos (11160 Euros) sumado al coste del software (119 Euros) y para hacerlo más real se puede incluso incluir el precio de un par de ordenadores (1000 Euros para hacer números redondos). Por lo tanto se estima que sin tener en cuenta costes de alquiler del local y variables similares (luz..) que tendrían lugar en una empresa coste es de unos **11279 Euros** .

8 Conclusiones

Una vez finalizado el proyecto, en este apartado se valora el resultado final obtenido, así como el trabajo futuro que se puede realizar en este proyecto para mejorarlo y finalmente en el ultimo apartado las sensaciones y conclusiones personales después de la realización de este proyecto.

8.1 Valoración del resultado

En este apartado se valora el resultado final del proyecto y si se han cumplido los objetivos propuestos al inicio.

El objetivo principal del proyecto ha sido completado con éxito: Se ha hecho un juego el cual cumplía con todas las normas del juego, así como todas las funcionalidades propuestas, y adicionalmente algunas de extras como por ejemplo el retroceder jugada o la guía del juego interactiva, la cual te va guiado a lo largo del juego en tiempo real según se va jugado, para que el jugador sepa como continuar la jugada según las normas del juego. La guía en este juego todo y no constar inicialmente en las funcionalidades del proyecto para hacer, la tuve de realizar debido a que el juego tiene demasiadas normas y yo mismo me perdía jugando al testear el juego, es decir no tenia claro como proseguir la jugada. Para no equivocarte debes saberte de memoria todas las reglas del juego (10 paginas aproximadamente) y no descontarse. Descontarse en la jugada en la que estas sucede mucho y saberte todas las jugadas y que no se te olviden lo encuentro inviable. Así que hice la guía interactiva para testear mejor el juego y para que los jugadores pudiesen jugar de una manera fácil, pues la guía les dice las opciones que tiene un jugador y como proseguir la jugada en todo momento. Estoy muy contento del resultado final de la guía interactiva, además la guía tiene algunos detalles escondidos, incluso uno de ellos te puede hacer jugar mejor si lo descubres.

Respecto las funcionalidades básicas inicialmente pactadas con el tutor se han hecho todas: se puede por ejemplo jugar en una interfaz gráfica, crear una partida con los jugadores que se desee, salvar una partida, cargarla, reiniciarla, ver el historial de jugadas hechas, se puede ver una ayuda con todas las normas del juego para aprender a jugar antes de empezar una partida o para consultar toda la normativa cuando un jugador quiera. Se puede jugar en local o en red mediante Internet, e incluso decidí adicionalmente crear un tipo de partida nuevo que es una mezcla de partida en red y partida local, es decir pueden haber varios jugadores jugando en la misma maquina (en local) jugando con varios jugadores en otras maquinas (en red)

en la misma partida. Se puede jugar una partida con inteligencia artificial de varios niveles de dificultad (4 niveles, fácil, media, difícil y muy difícil).También se puede un listado con los jugadores que han ganado mas partidas, así como los jugadores que mejores puntuaciones han tenido.

Respecto los puntos débiles del proyecto son según piensa el proyectista son la interfaz gráfica, y la Inteligencia artificial. La interfaz gráfica el juego inicialmente el proyectista la pensó para que el juego tuviera el mismo aspecto que los juegos de cartas de Windows 8 (el último Windows disponible en la fecha de desarrollo). Así que el proyectista en un principio empezó la programación de la interfaz con java usando SWING (librería creada a partir de java AWT). Lo que sucedió realmente el proyectista no lo esperaba y fue que se encontró con un bug de java, es decir un error interno del lenguaje de programación java (no de la programación del proyectista). El proyectista inicialmente observaba errores aleatorios sin sentido. Inicialmente el proyectista no entendió que era un bug de java pues nunca se había encontrado uno, pensó pues que era un error suyo. El tiempo iba pasando y no encontraba la fuente del error, así que se planteo dedicarle todo su tiempo a ver que ocurría, se dio cuenta que el JDK de java indicaba que la linea del error estaba en una librería de JAVA, es decir en el JAVA propiamente dicho. Así que decidió buscar por Internet que ocurría y entonces se convenció que era un bug. Deshizo código, hasta que le desapareciera el bug, pero no funcionó. Cambio y simplifico un sin fin de código para intentar que el bug no apareciese, quito todos los warnings del proyecto, pero nada parecía dar-le resultado. Uso una complemento llamado FindBugs para intentar solventar la situación y cambio todo aquello que FindBugs le indicaba, y nada parecía funcionar. Finalmente después de consultar semanas enteras por Internet en foros y haciendo todo lo que los usuarios decían en casos similares, encontró una solución que paliaba el problema. Por lo visto los había un problema con los hilos que JAVA no termina de gestionar bien y el InvokeAndWait pareció ser la solución. El bug se resistió tanto que el proyectista pensó de muy seriamente en volver a hacer el proyecto pero esta vez basado en Android. De hecho el tiempo invertido para solventar el bug fue superior al invertido en desarrollar toda la interfaz gráfica en ese momento llegando a ser de casi un par de meses. Debido ha este problema el proyectista decidió parar de mejorar los efectos de la interfaz gráfica, no llegando a ser una interfaz gráfica tan bonita como él deseaba.

Respecto la Inteligencia Artificial (IA) del juego, el proyectista esta bastante satisfecho del resultado, pero cree que una Inteligencia Artificial siempre se puede mejorar.

8.2 Trabajo Futuro

Trabajo futuro ha realizar en este proyecto son tres:

Mejorar la Inteligencia Artificial ya de por si suficientemente eficaz, lo cual conllevaría que el jugador fuera más experto jugando. A más experto fuera el proyectista jugando a este juego mejor seria la Inteligencia Artificial (siempre se puede mejorar). El proyectista para hacer la Inteligencia Artificial se ha tenido en cuenta los consejos del autor del juego así como su experiencia personal jugando adquirida al programar el juego entero. El proyectista aún así esta bastante satisfecho con la IA.

El otro punto a mejorar es la Interfaz gráfica del juego pues también siempre se puede mejorar gráficamente. Aún así el proyectista cree que ha hecho todo lo posible acorde al Swing de java y a sus limitaciones acorde la versión actual de Java existente en la actualidad. El proyectista uso para hacer la guía y el historial de jugadas una ventana basa en código HTML que hace unos años ni tan siquiera estaba disponible ni existía esta posibilidad de incorporar este tipo de ventana.

Adicionalmente se podría añadir varios idiomas en el juego, de manera que el juego fuera comprensible para jugadores de diferentes países.

8.3 Conclusiones personales

Las conclusiones personales de este proyecto son varias. Primero se narran las criticas al proyecto, luego los puntos fuertes.

CRITICAS

Respecto al juego, la opinión del proyectista, es que el estudiante de Estadística de Zaragoza, llamado José Carlos de Diego Guerrero no se hará rico con este juego por lo menos.

El proyectista opina que los juegos que realmente tienen éxito son los simples, (incluso el ajedrez es más simple en normativa). Este juego tiene mucha normativa, demasiada, demasiado tipos de jugadas, distintas, demasiado efectos con los nobles, hasta aprender ha hacer la puntuación final del juego tiene su complicación. Como se dice anteriormente el juego sin ayuda que ofrece la guía interactiva que guía al jugador de que jugada le toca hacer al jugador seria muy muy difícil de jugar. Realmente tanta variabilidad, una vez entiendes todo el juego, como funciona , la estrategia básica a seguir es relativamente simple.

Respecto al Java, el proyectista ya desde pequeño piensa que lo que no haces tu personalmente, y si quieres que algo este realmente bien hecho, lo mejor que puedes hacer es hacerlo tu enteramente. Realmente nunca pensó que un bug de JAVA le daría tantos problemas.

Respecto la planificación del proyecto, el proyectista considera que ha realizado dos errores importantes en este juego. El primero quizás hubiese tenido que realizar el proyecto basado en Android, en lugar de usar la librería Swing de Java, quizás esto hubiera simplificado el proyecto y lo hubiese hecho mas robusto y vistoso. En principio se hizo con Swing debido a que el proyecto no estaba hablado de hacerlo para Android y porque ademas el proyectista no tiene experiencia en Android y empezó este proyecto usando el juego de hecho en la asignatura PROP de la carrera como plantilla.

El otro error que cree que ha cometido el proyectista es quizá la realización de retroceder jugada, pues esta funcionalidad prácticamente pasa desapercibida y conlleva muchísimo trabajo.

VENTAJAS

La ventaja principal de la realización de este proyecto es que el proyectista ha adquirido más practica en la programación de juegos, así que de como realizarlos y diseñarlos.

9 Bibliografía

Se muestra a continuación una pequeña parte de la bibliografía utilizada.

9.1 Api de java

http://docs.oracle.com/javase/7/docs/api/

http://docs.oracle.com/javase/8/

9.2 Foros de java

http://stackoverflow.com/questions/5428196/using-png-in-a-jar-file

http://bugs.java.com/bugdatabase/view_bug.do?bug_id=4671653

http://stackoverflow.com/questions/3433809/java-setvisibletrue-has-no-effect-on-gui

https://community.oracle.com/welcome

http://www.pierotofy.it/pages/extras/forum/14/1036054-jtextpane_or_jtexteditor_to_solve_html_problems/

http://stackoverflow.com/questions/6641457/access-change-jeditorpanes-html-loaded-elements-htmleditorkit-problem-with-un/7032855#7032855

9.3 Tutoriales de java

http://programacion.net/articulo/swing_y_jfc_java_foundation_classes_94/45#classes

http://docs.oracle.com/javase/tutorial/uiswing/components/text.html

https://azerdark.wordpress.com/2008/07/30/java-bubble-sort-in-linked-list/

http://docs.oracle.com/javase/tutorial/uiswing/components/panel.html

http://www.avajava.com/tutorials/lessons/how-do-i-return-an-image-from-a-servlet-using-imageio.html

http://www.java2s.com/Code/Java/2D-Graphics-GUI/WritesanimagetoanoutputstreamasaJPEGfileTheJPEGqualitycanbespecifiedinpercent.htm

http://mindprod.com/jgloss/jeditorpane.html

http://zetcode.com/tutorials/javagamestutorial/basics/

http://ocw.uc3m.es/ingenieria-informatica/programacion/manuales/java2-U-Navarra.pdf

http://codigosimportantes.blogspot.com.es/2012/05/imagen-de-fondo-de-fondo-con-jlabel.html

http://novacreations.net/eclipse-teclas-de-acceso-rapido/

http://www.ctr.unican.es/asignaturas/procodis_3_II/Doc/Procodis_3_01.pdf

https://mail.google.com/mail/u/0/#search/practica+ia/12c0359c57de9165?projector=1

http://mundogeek.net/archivos/2013/04/28/como-mejorar-tu-productivad-con-netbeans-en-un-200/

9.4 Reglas del juego

https://jugandoenpareja.wordpress.com/2011/08/07/juego-de-hoy-l%E2%80%99esprit-de-marie-antoinette/

http://labsk.net/wkr/archives/tag/marie-antoinette/#sthash.EZJ1INVj.dpbs

9.5 Youtube

www.youtube.com

(un ejemplo de los muchos de los videos visionados)

https://www.youtube.com/watch?v=ap3KrB2RlRw

9.6 Google code

https://code.google.com/p/marie-antoinette/

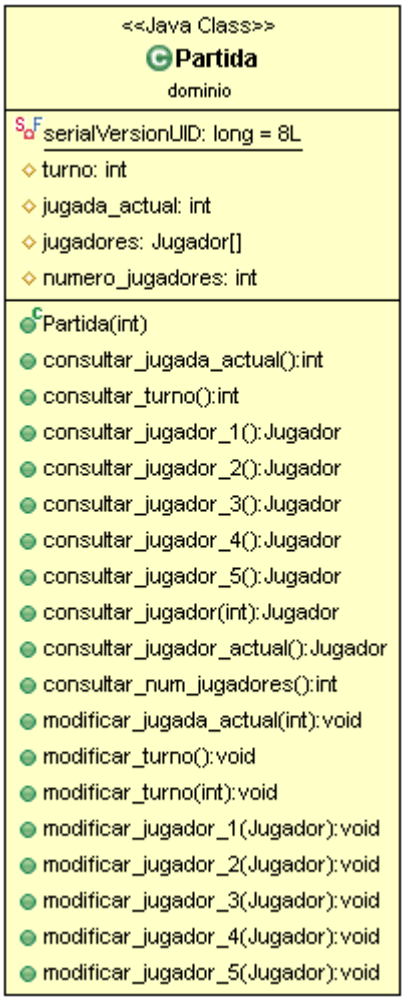
10 Anexo

A continuación se muestran todas las clases del proyecto con sus variables y funciones que componen a cada una de ellas. Primero se muestran las capa de dominio, luego mostrare la capa de disco y finalmente la de vistas. Esta división se mantiene en el proyecto al abrir eclipse o netbeans el proyecto lo podrían observar.

10.1 Capa de Dominio

Como se ha dicho anteriormente hay 3 capas: dominio, vistas, disco. Aqui se muestra primero la capa de dominio. La primera clase a mostrar es partida, una de las más representativas.

10.1.1 Partida

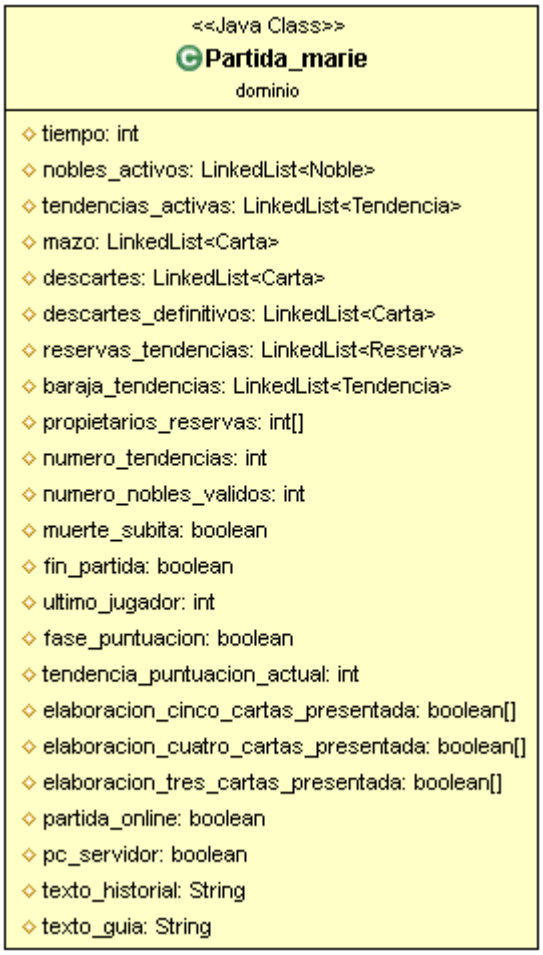


La clase partida es una de las más representativas del juego. Se muestran pues todas sus variables y funciones para que las podáis observar.

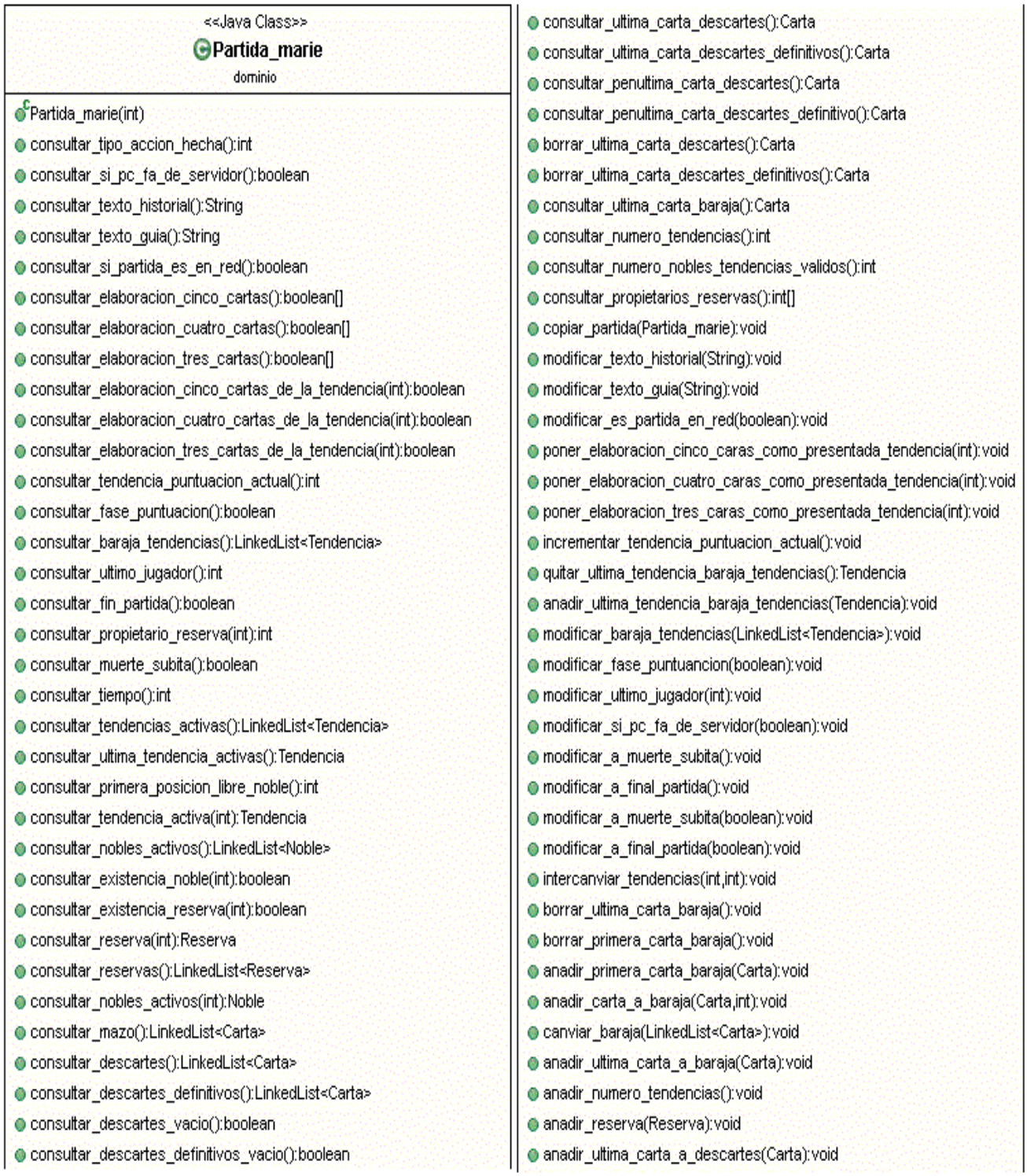
10.1.2 Partida Marie

(Variables)

Esta clase es bastante compleja así que se muestra a dos paginas, primero las variables y luego las funciones.



Partida Marie (funciones)



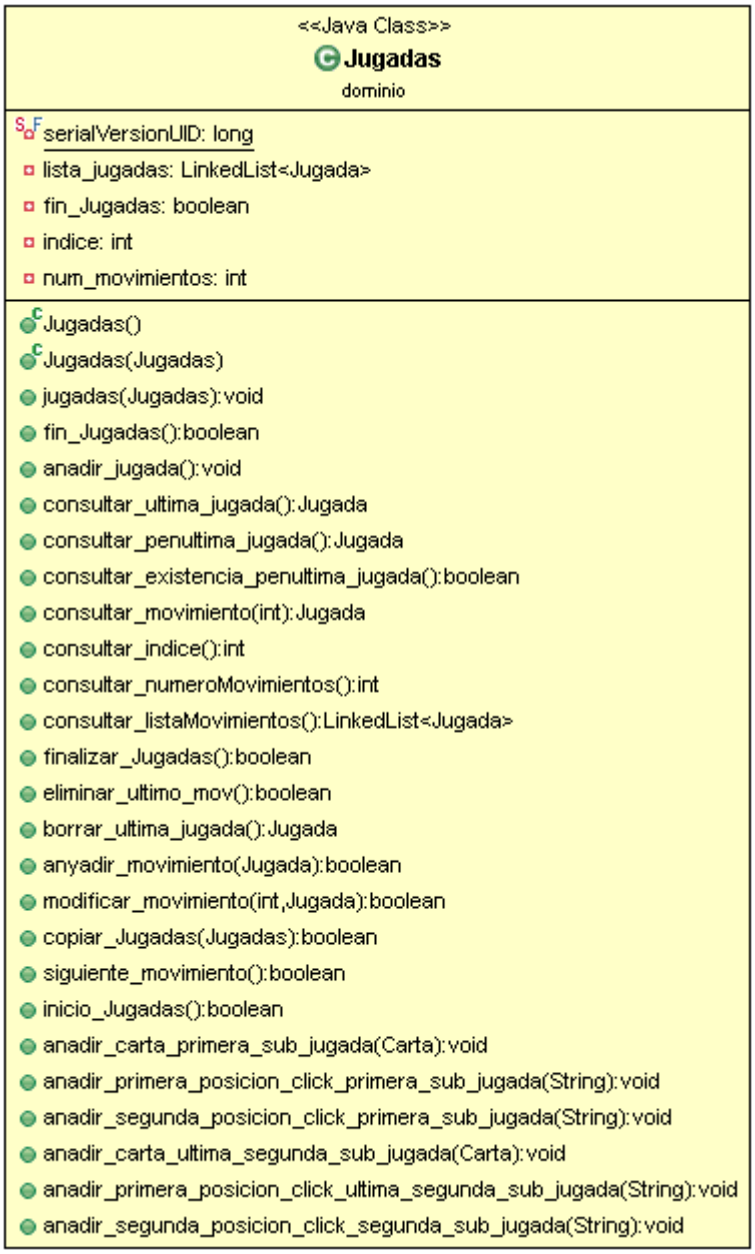

anadir_ultima_carta_a_descartes_definitivos(Carta):void
anadir_reserva(Reserva,int):void
quitar_reserva(int):void
modificar_reservas(LinkedList<Reserva>):void
modificar_tiempo(int):boolean
modificar_descartes(LinkedList<Carta>):void
modificar_descartes_definitivos(LinkedList<Carta>):void
modificar_descartes_anadir_carta(Carta):void
modificar_descartes_definitivos_anadir_carta(Carta):void
modificar_nobles_activos(LinkedList<Noble>,int):void
borrar_noble_activos(int):void
borrar_reserva_activa(int):void
anadir_noble_activos(int,Noble):void
substituir_noble_activos(int,Noble):void
anadir_noble(Noble):void
modificar_tendencias_activas(LinkedList<Tendencia>):void
anadir_tendencias_activa(Tendencia):void
borrar_tendencias_activas(int):void
anadir_tendencias_activas(int,Tendencia):void
quitar_ultima_tendencia_activas():void
modificar_mazo(LinkedList<Carta>):void
modificar_propietario_reserva(int,int):void
quitar_nobles_restantes_mazo():void
modificar_tipo_accion_hecha(int):void
eliminar_ultimo_tipo_accion_hecha():void
reinicar_jugadas():void

10.1.3 Jugador

Se muestra a continuación la clase jugador con sus funciones y variables. La imagen esta partida en 2 para aprovechar el espacio.

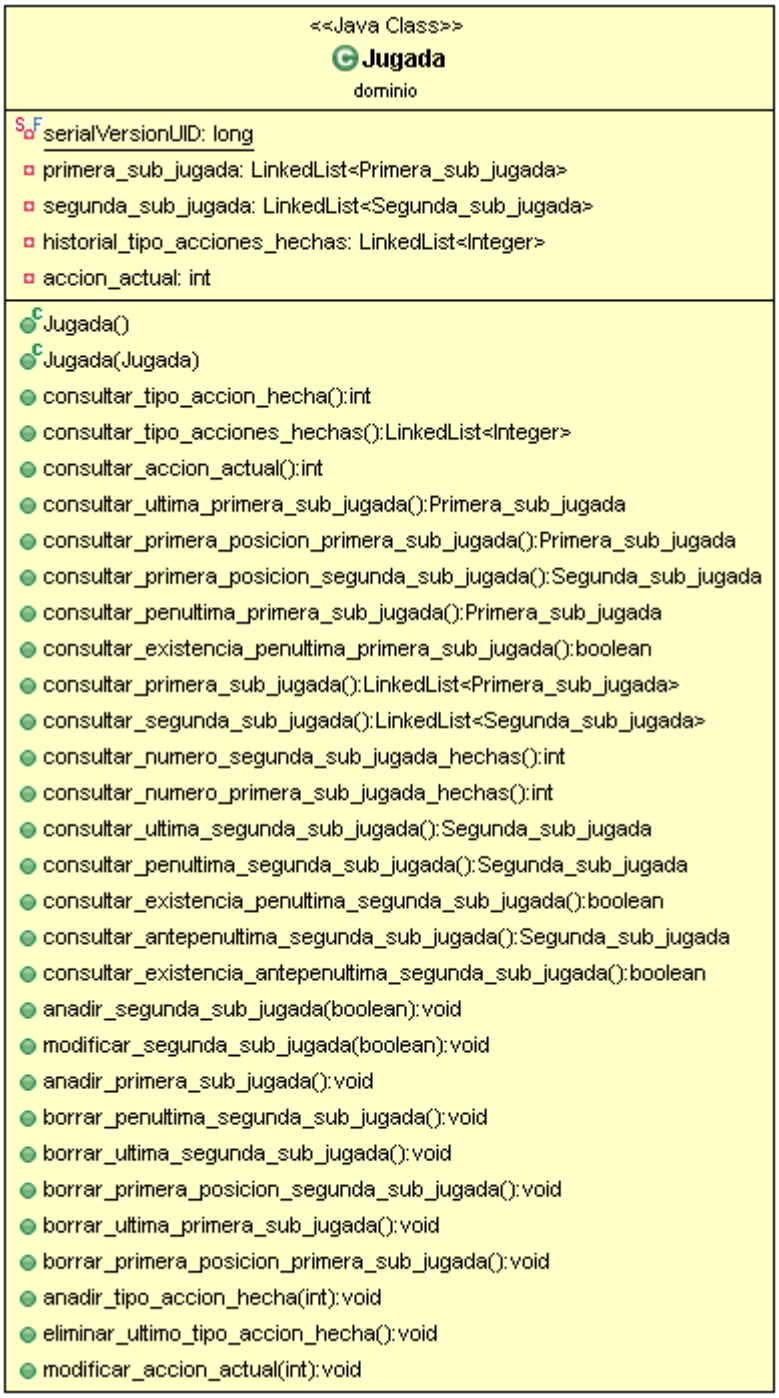
10.1.4 Jugadas

Se muestra a continuación la clase jugadas con sus funciones y variables.



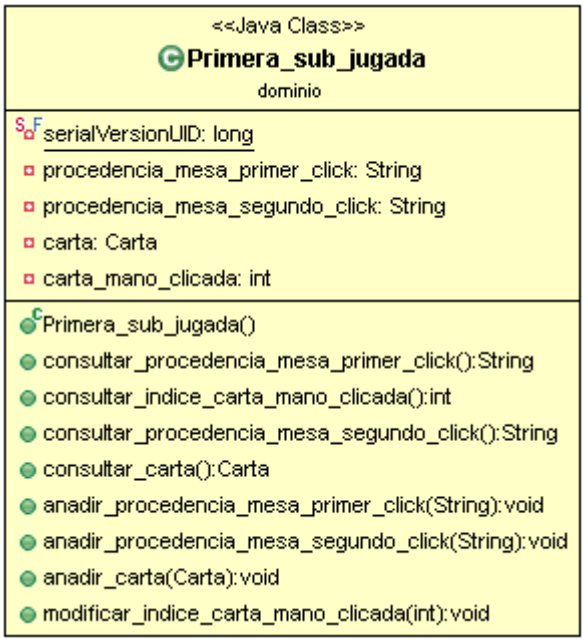
10.1.5 Jugada

Se muestra a continuación la clase jugada con sus funciones y variables.

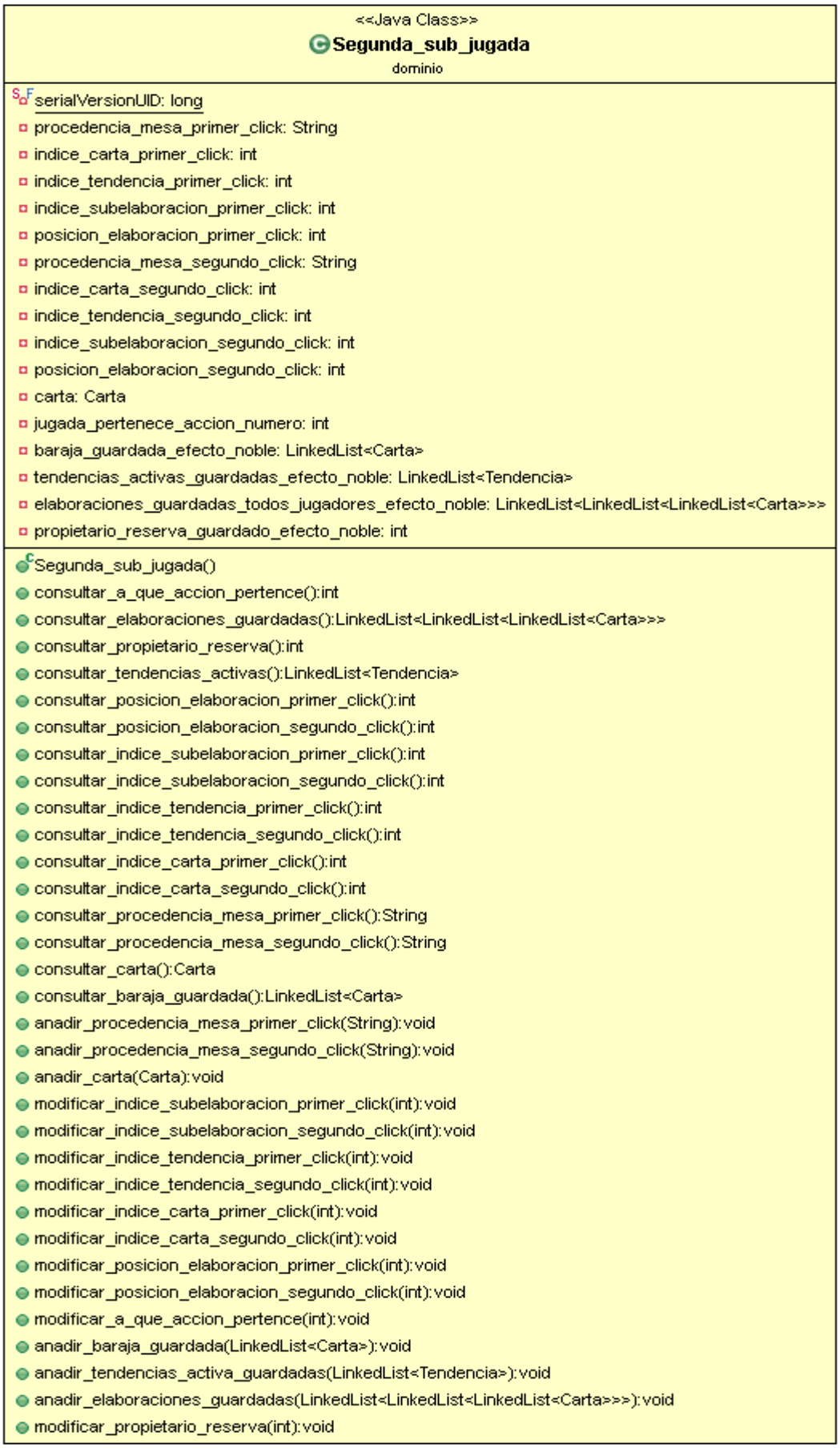


10.1.6 Primera subjugada

Se muestra a continuación la clase primera subjugada con sus funciones y variables. Dicha clase forma parte de la clase jugada.

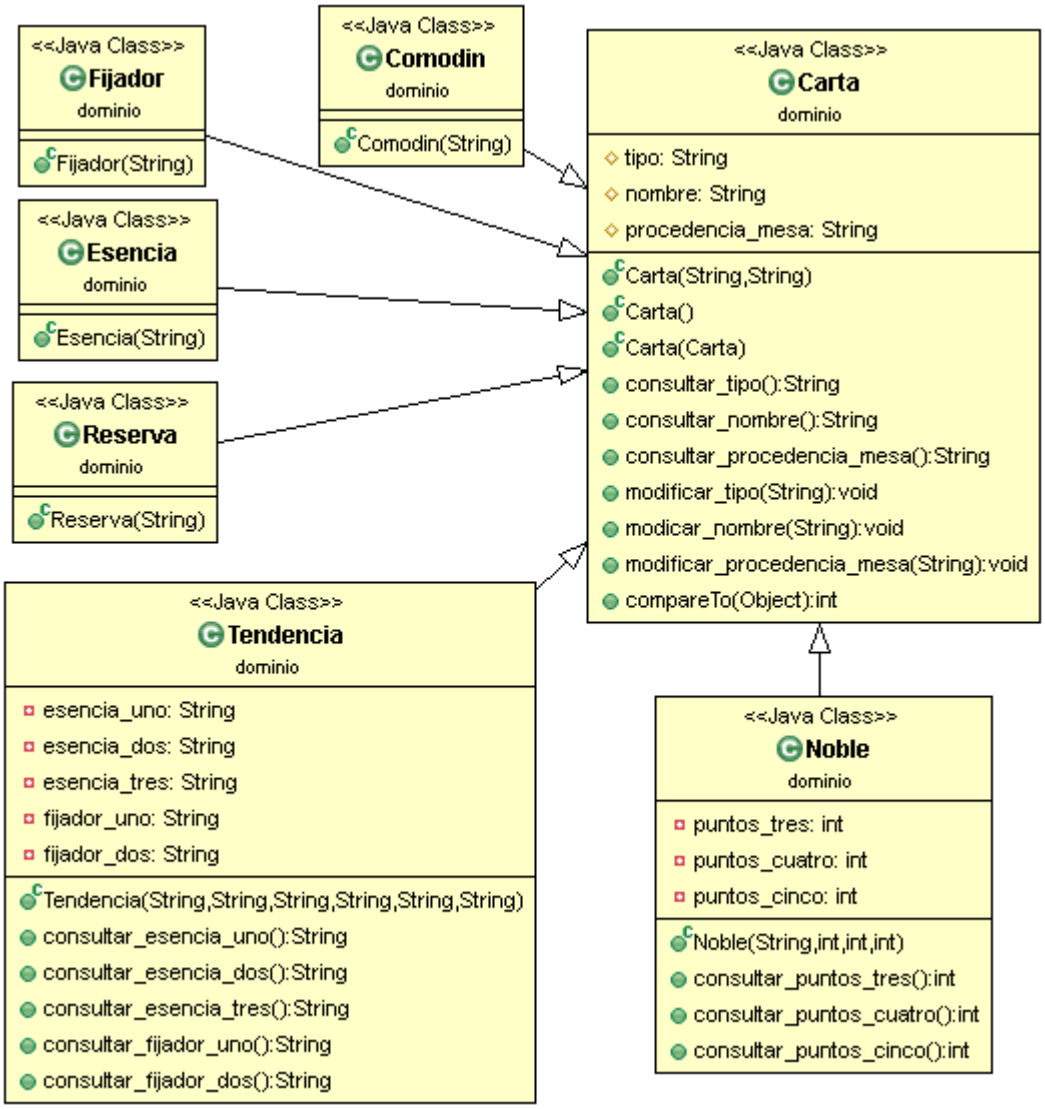


10.1.7 Segunda subjugada

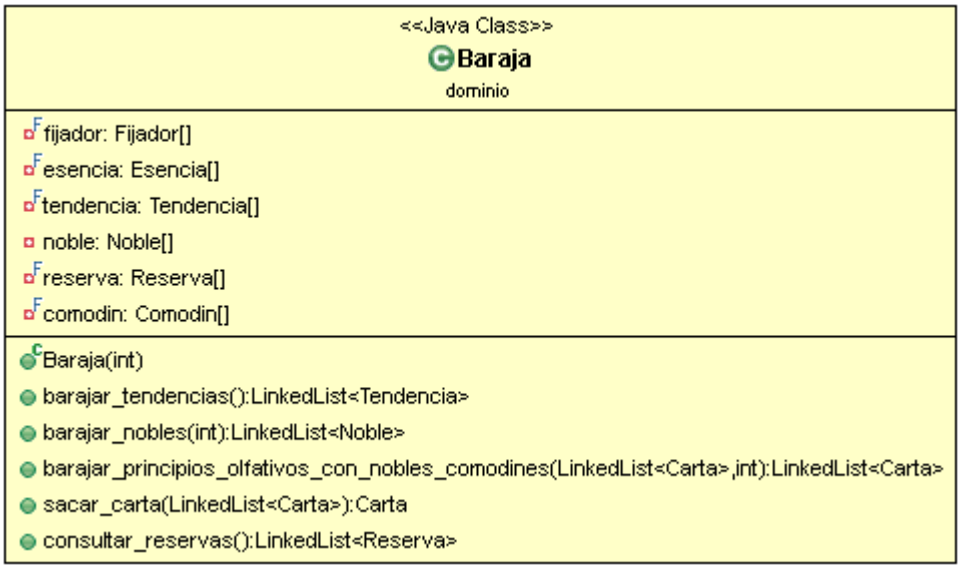


10.1.8 Cartas y tipos de cartas

Se muestra a continuación la clase carta así como los tipos de cartas existentes con sus funciones y variables.

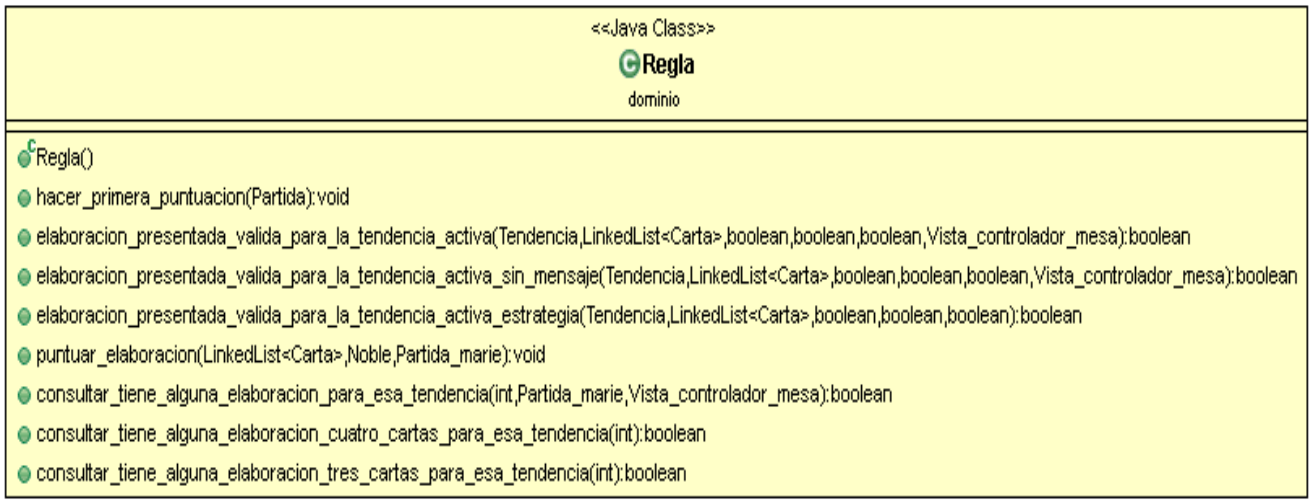


10.1.9 Baraja



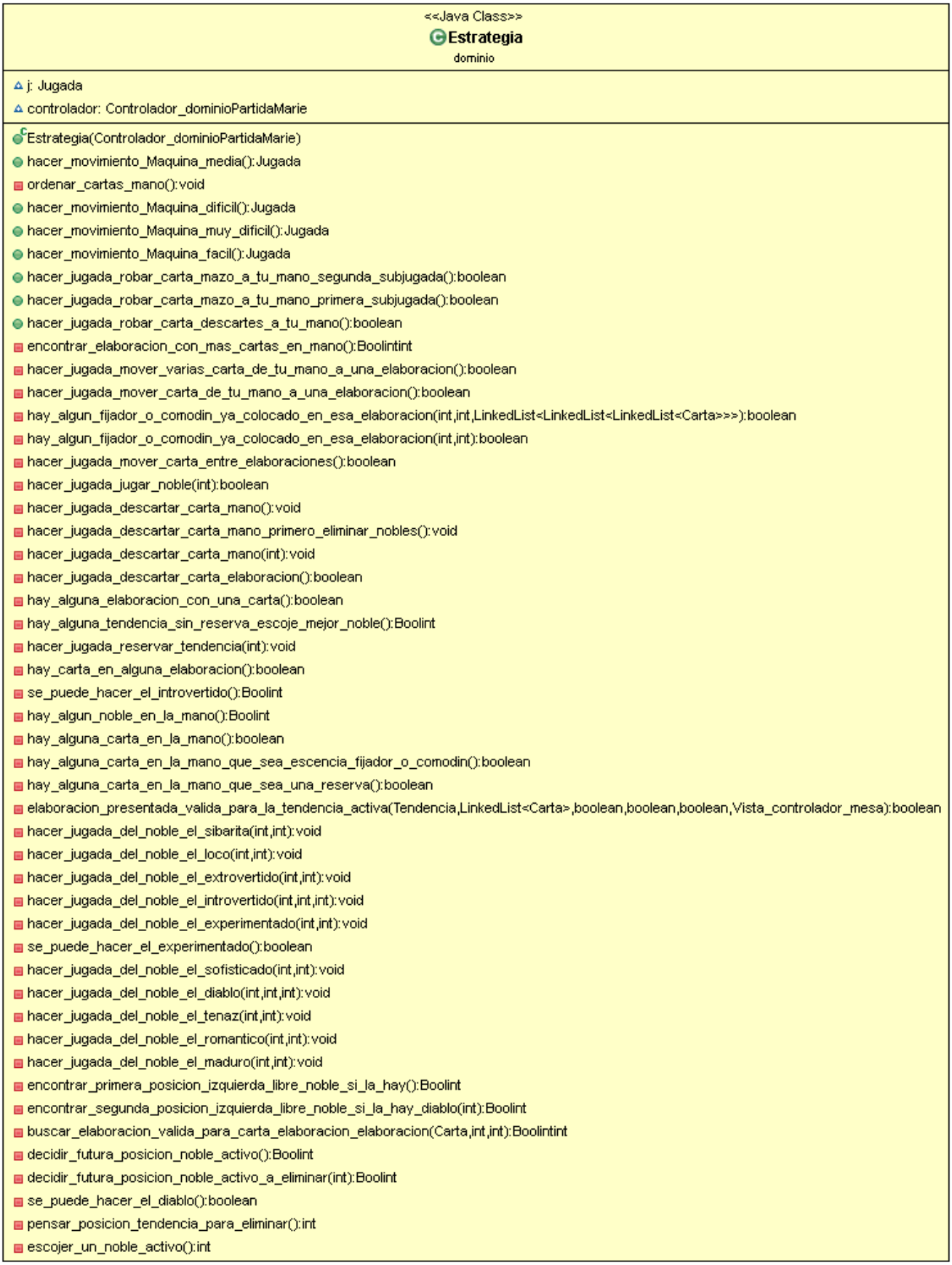
10.1.10 Regla

La clase regla es la encargada de validar las puntuación final, así como validar las elaboraciones presentadas en la fase final de la puntuación.



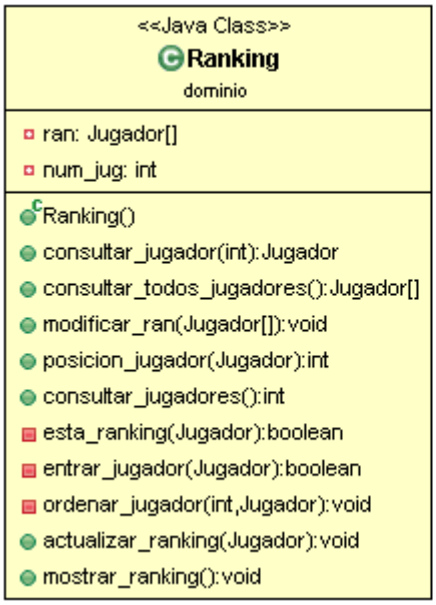
10.1.11 Estrategia

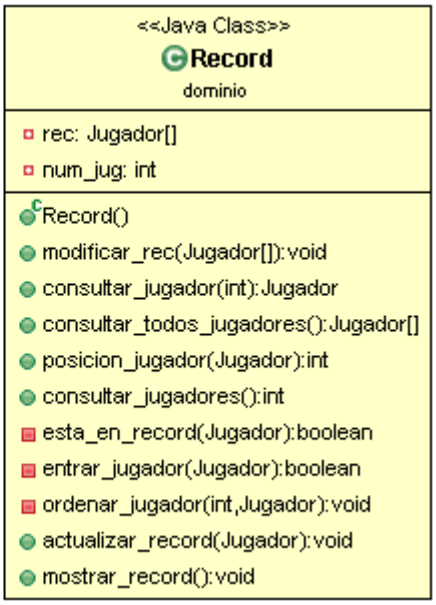
Esta clase es la encargada de realizar la inteligencia artificial del juego.



10.1.12 Ranking y récord

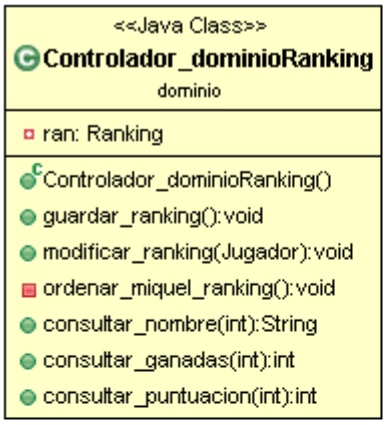
Se muestra a continuación las clases ranking y récord con sus funciones y variables.

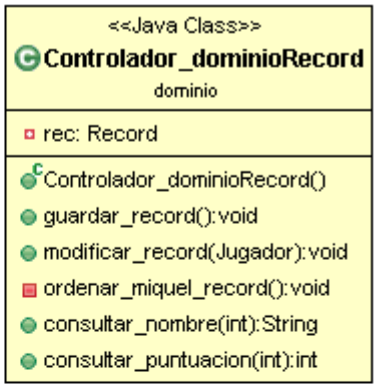


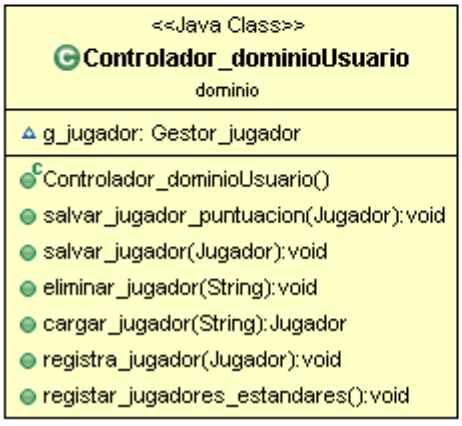


10.1.13 Controlador ranking, récord y usuario

Se muestra a continuación las clases controlador ranking y récord y usuario con sus funciones y variables.

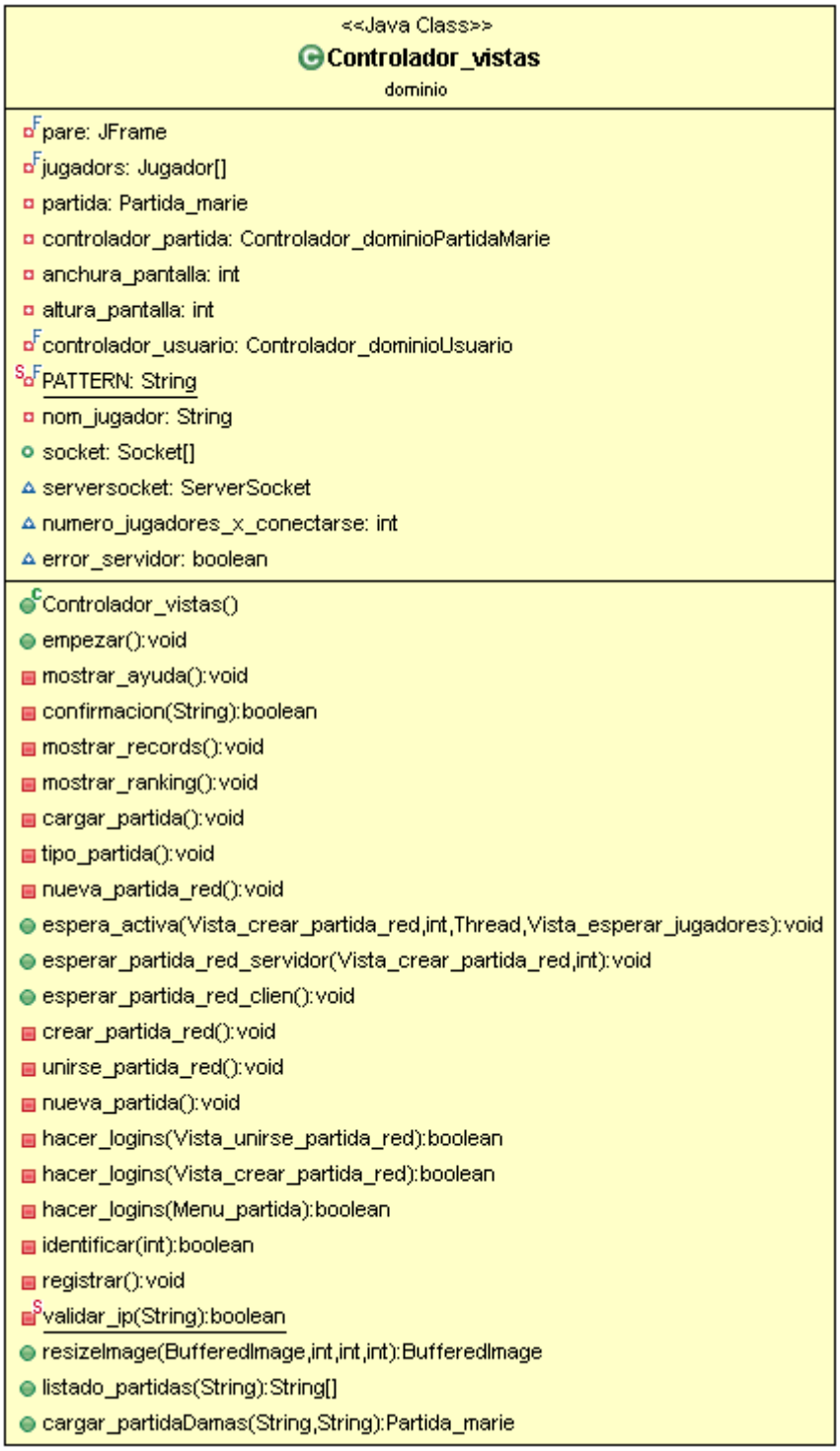




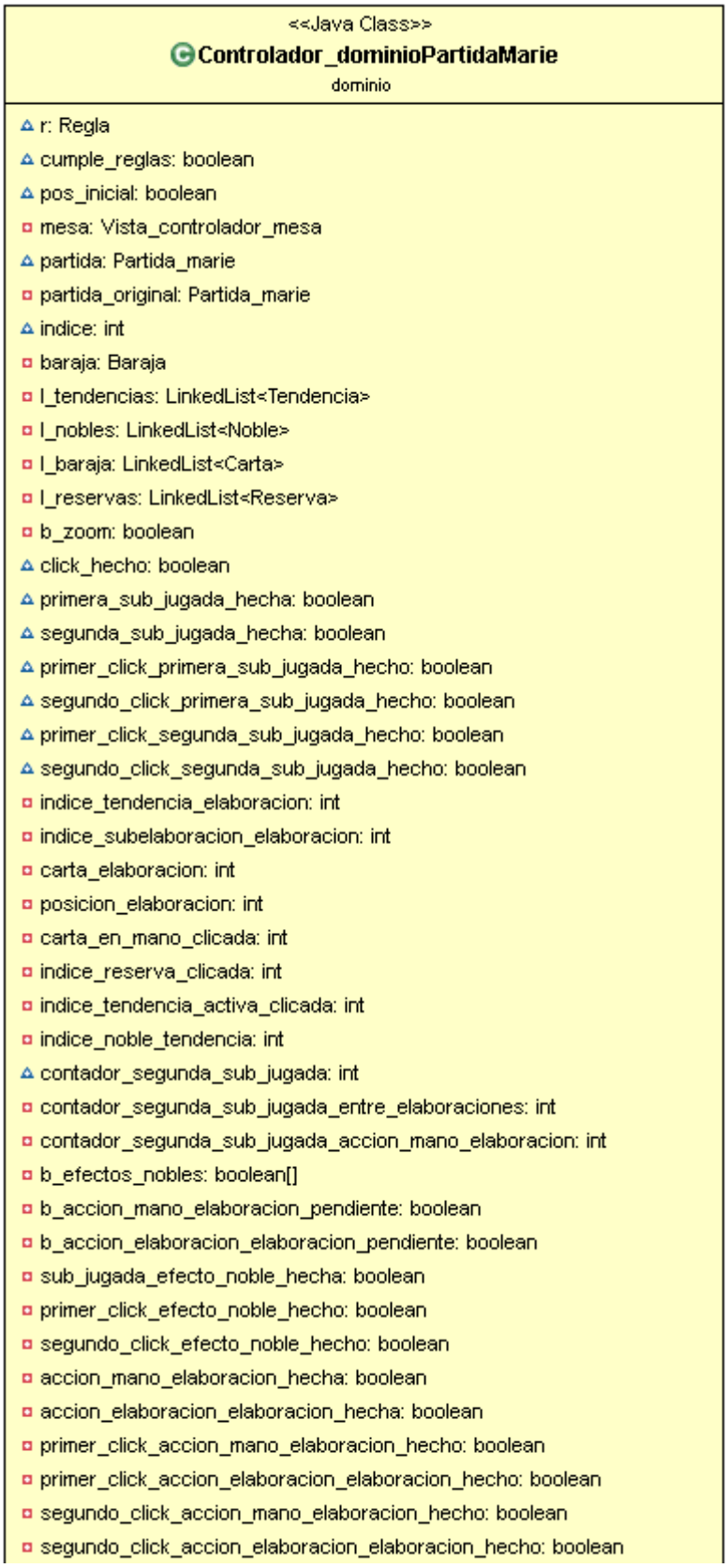


10.1.14 Controlador vistas

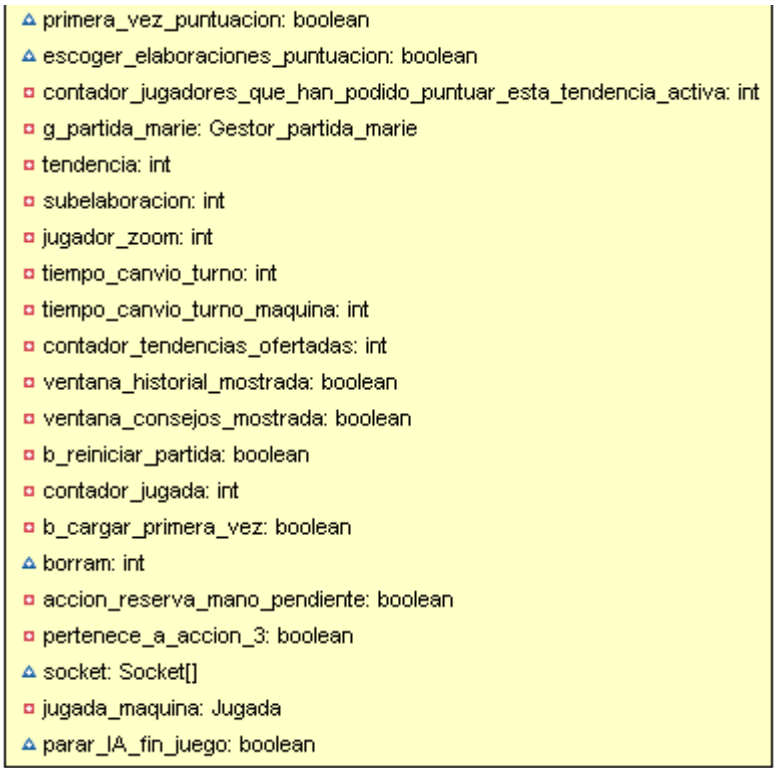
Se muestra a continuación la clase controlador vistas con sus funciones y variables. Esta clase es la encargada de controlar todos los menús antes de empezar la partida. También se encarga de empezar la partida.



10.1.15 Controlador Partida Marie



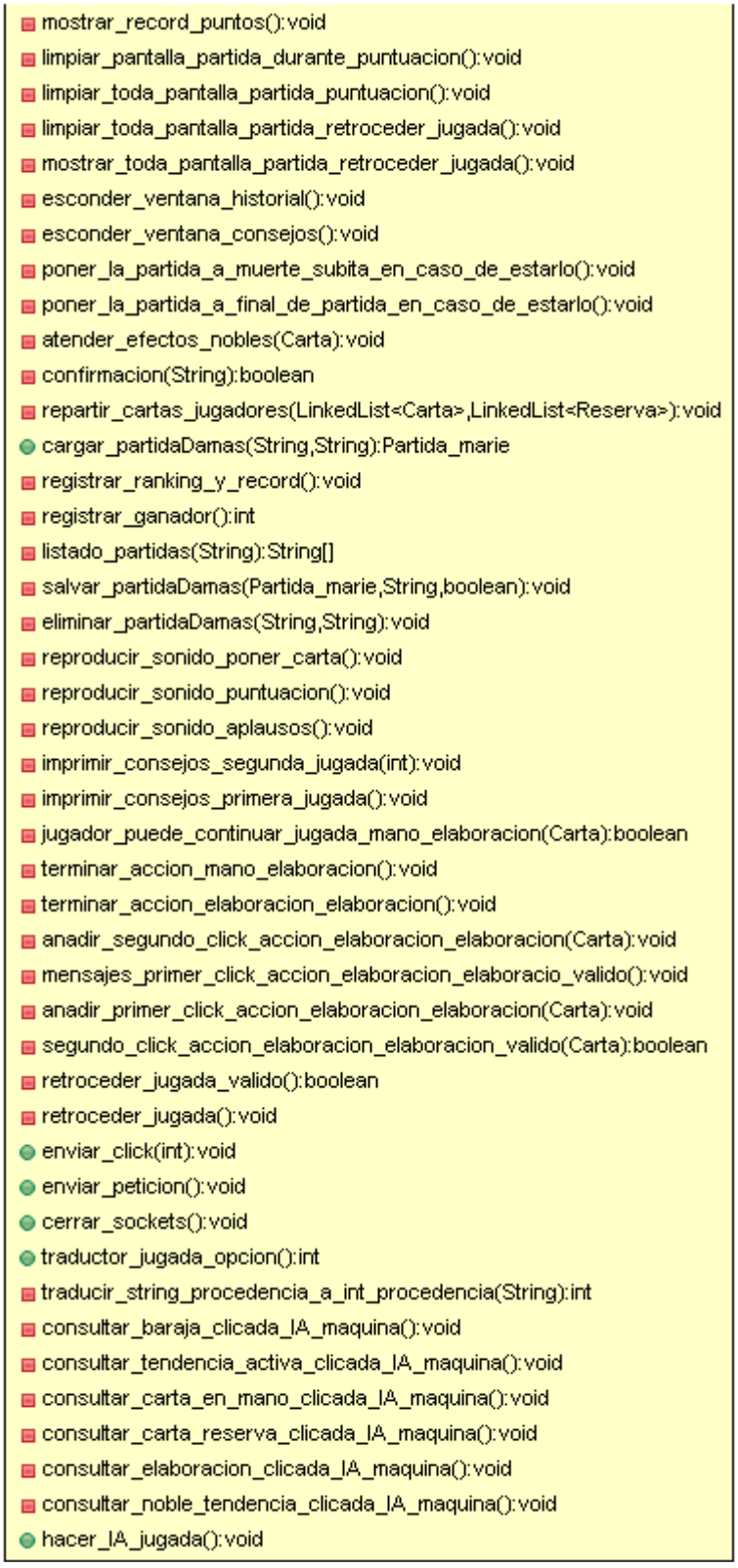
(Variables)



Se esta mostrando la clase controlador partida marie con sus funciones y variables (en varias paginas). Esta clase es la encargada de controlar todo el juego una vez esta puesto en marcha. Es las que tiene más complejidad de todo el proyecto pues controla el cumplimiento de todas las normas así como de controlar todo lo referente a la pantalla del juego una vez iniciado. El proyectista ha trabajado un 80% de todo el tiempo en esta clase.

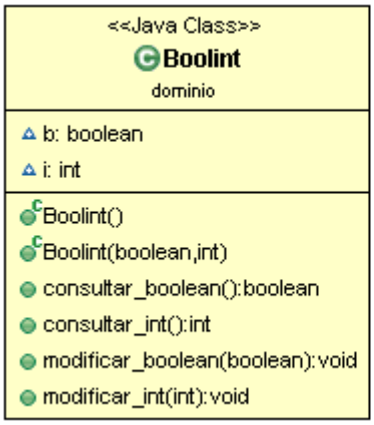
(Funciones)

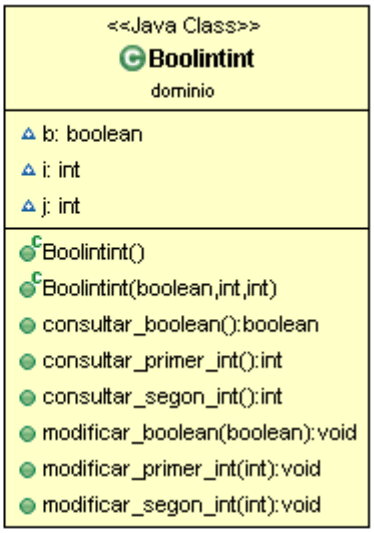

<<Java Class>>
Controlador_dominioPartidaMarie
dominio
Controlador_dominioPartidaMarie(Partida_marie,Socket[])
Controlador_dominioPartidaMarie(Partida_marie)
inicializar_variables_clase():void
inicializar_mesa():void
inicializar_variables_partida(Partida_marie):void
inicializar_todo(Partida_marie):void
juego():void
atender_click_carta(Carta):void
atender_accion_mano_elaboracion_pendiente(Carta):void
atender_accion_elaboracion_elaboracion_pendiente(Carta):void
anadir_segundo_click_accion_mano_elaboracion(Carta):void
anadir_primer_click_accion_mano_elaboracion(Carta):void
segundo_click_accion_mano_elaboracion_valido(Carta):boolean
primer_click_accion_mano_elaboracion_valido(Carta):boolean
primer_click_accion_elaboracion_elaboracion_valido(Carta):boolean
pasar_de_la_segunda_sub_jugada_a_la_tercera():void
canviar_turno():void
anadir_primera_sub_jugada_segundo_click(Carta):void
analisis_carta_historial(Carta):void
anadir_segundo_click_efecto_noble_el_sibarita(Carta):void
anadir_segundo_click_efecto_noble_el_experimentado(Carta):void
anadir_segundo_click_efecto_noble_el_introvertido(Carta):void
anadir_segunda_sub_jugada_segundo_click(Carta):void
primer_click_primera_sub_jugada_hecho_valido(Carta):boolean
primer_click_segunda_sub_jugada_hecho_valido(Carta):boolean
primer_click_efecto_noble_el_experimentado_valido(Carta):boolean
primer_click_efecto_noble_el_sibarita_valido(Carta):boolean
primer_click_efecto_noble_el_diablo_valido(Carta):boolean
primer_click_efecto_noble_el_introvertido_valido(Carta):boolean
anadir_primer_click_efecto_noble(Carta):void
anadir_primera_sub_jugada_primer_click(Carta):void
anadir_segunda_sub_jugada_primer_click(Carta):void
segundo_click_primera_sub_jugada_hecho_valido(Carta):boolean
segundo_click_efecto_noble_el_sibarita_valido(Carta):boolean
segundo_click_efecto_noble_el_experimentado_valido(Carta):boolean
segundo_click_efecto_noble_el_introvertido_valido(Carta):boolean
segundo_click_segunda_sub_jugada_hecho_valido(Carta):boolean
mostrar_ventana_historial():void
mostrar_ventana_consejos():void
mostrar_mensaje_2_puntuacion():void
mostrar_ayuda():void
mostrar_pantalla_partida_durante_puntuacion():void
mostrar_ranking():void

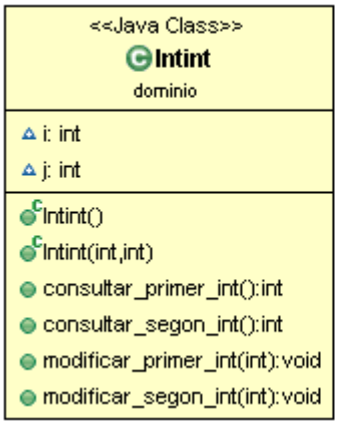

mostrar_record_puntos():void
limpiar_pantalla_partida_durante_puntuacion():void
limpiar_toda_pantalla_partida_puntuacion():void
limpiar_toda_pantalla_partida_retroceder_jugada():void
mostrar_toda_pantalla_partida_retroceder_jugada():void
esconder_ventana_historial():void
esconder_ventana_consejos():void
poner_la_partida_a_muerte_subita_en_caso_de_estarlo():void
poner_la_partida_a_final_de_partida_en_caso_de_estarlo():void
atender_efectos_nobles(Carta):void
confirmacion(String):boolean
repartir_cartas_jugadores(LinkedList<Carta>,LinkedList<Reserva>):void
cargar_partidaDamas(String,String):Partida_marie
registrar_ranking_y_record():void
registrar_ganador():int
listado_partidas(String):String[]
salvar_partidaDamas(Partida_marie,String,boolean):void
eliminar_partidaDamas(String,String):void
reproducir_sonido_poner_carta():void
reproducir_sonido_puntuacion():void
reproducir_sonido_aplausos():void
imprimir_consejos_segunda_jugada(int):void
imprimir_consejos_primera_jugada():void
jugador_puede_continuar_jugada_mano_elaboracion(Carta):boolean
terminar_accion_mano_elaboracion():void
terminar_accion_elaboracion_elaboracion():void
anadir_segundo_click_accion_elaboracion_elaboracion(Carta):void
mensajes_primer_click_accion_elaboracion_elaboracio_valido():void
anadir_primer_click_accion_elaboracion_elaboracion(Carta):void
segundo_click_accion_elaboracion_elaboracion_valido(Carta):boolean
retroceder_jugada_valido():boolean
retroceder_jugada():void
enviar_click(int):void
enviar_peticion():void
cerrar_sockets():void
traductor_jugada_opcion():int
traducir_string_procedencia_a_int_procedencia(String):int
consultar_baraja_clicada_IA_maquina():void
consultar_tendencia_activa_clicada_IA_maquina():void
consultar_carta_en_mano_clicada_IA_maquina():void
consultar_carta_reserva_clicada_IA_maquina():void
consultar_elaboracion_clicada_IA_maquina():void
consultar_noble_tendencia_clicada_IA_maquina():void
hacer_IA_jugada():void

10.1.16 Tuplas

Se muestran a continuación varias clases tupla creadas durante el transcurso del proyecto para facilitar el retorno de variables en funciones y así hacer mas eficiente el juego. De este modo no se repetía código pues una misma función era suficiente para realizar una tarea y no hacia falta crear varias funciones similares.

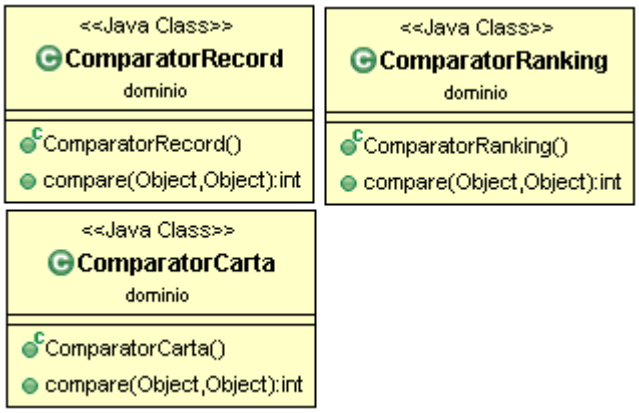






10.1.17 Comparators

Estas clases fueron creadas para la facilitar la ordenación aprovechando la API de java. Especialmente util a la hora de elaborar el ranking y el récord.

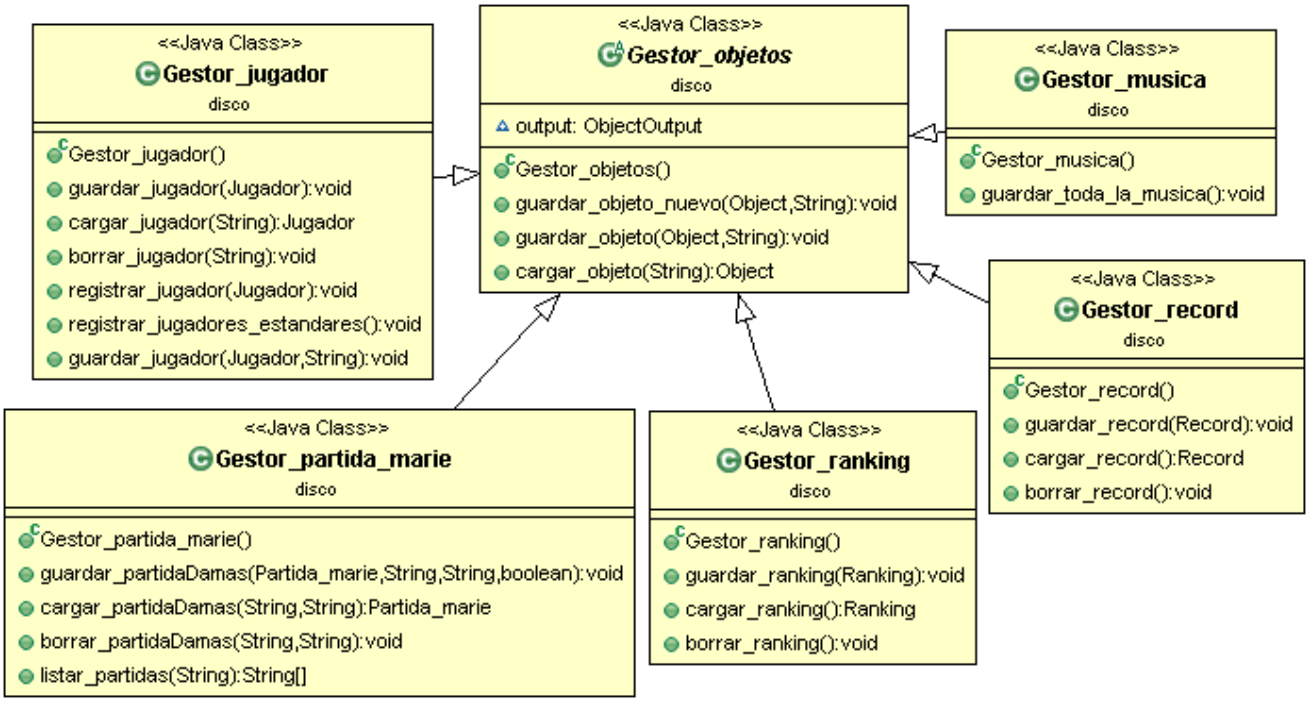


10.2 Capa de Disco

Una vez mostrada la capa de dominio, se procede a mostrar la capa de disco. Luego más adelante se muestra la capa de vistas. Procedemos pues ha mostrar la capa de disco en este apartado empezando en los gestores de disco. Esta capa es la que menos clases tiene de las tres.

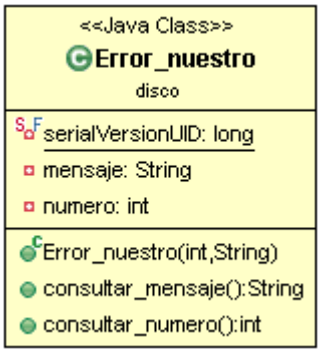
10.2.1 Gestores de disco

Los gestores de disco son los encargados de guardar datos es decir de la persistencia. Todo lo que se guarda pasa por estos gestores.



10.2.2 Error nuestro

Esta clase gestiona los errores. Quizás es mas un controlador y pertenece a la capa de dominio pues no



guarda nada a disco. Fue creada durante el desarrollo de la antigua asignatura de Prop

10.3 Capa de Presentación

La última capa que queda por mostrar es la capa de presentación. Dicha capa tiene todas las vistas y menús del juego. Estos menús y vistas son controlados por los controladores de la capa de dominio. Esta capa es la encargada de mostrar al usuario vistas para que el usuario sepa fácilmente como interactuar con la capa de dominio.

10.3.1 Menús

10.3.2 Vistas

Anteriormente se mostraron los menús, ahora se procede a mostrar las vistas. Observad que tanto los menús como las vistas tienen funciones muy similares que coinciden.

A la pagina siguiente se continuaran mostrando las vistas que restan .


<<Java Class>>
Vista_record_puntos
vistas
serialVersionUID: long
contenedor: Container
volver: JButton
jug: JLabel[]
punt: JLabel[]
con_rec: Controlador_dominioRecord
pulsado: boolean
Vista_record_puntos(JFrame)
visualizar():void
pulsado():boolean
cerrar():void
actionPerformed(ActionEvent):void
titulos():void
botones():void
montar():void
sonar_efecto():void
<<Java Class>>
Vista_ranking_ganadores
vistas
serialVersionUID: long
contenedor: Container
volver: JButton
jug: JLabel[]
punt: JLabel[]
con_rank: Controlador_dominioRanking
pulsado: boolean
Vista_ranking_ganadores(JFrame)
visualizar():void
pulsado():boolean
cerrar():void
actionPerformed(ActionEvent):void
titulos():void
botones():void
montar():void
sonar_efecto():void
<<Java Class>>
Vista_unirse_partida_red
vistas
contenedor: Container
nueva: JButton
salir: JButton
l_poner_ip: JLabel
t_ip: JTextField
opcion: int
ip: String
Directorio_imagenes: String
label: JLabel
ck_login: JCheckBox
Vista_unirse_partida_red(JFrame)
visualizar():void
preguntar():int
cerrar():void
actionPerformed(ActionEvent):void
titulos():void
botones():void
montar():void
mousePressed(MouseEvent):void
mouseReleased(MouseEvent):void
mouseEntered(MouseEvent):void
mouseExited(MouseEvent):void
mouseClicked(MouseEvent):void
consultar_ip():String
validar_ip(String):boolean
b_hacer_login():boolean
reproducir_sonido_poner_carta():void
<<Java Class>>
Vista_salvarPartida
vistas
partida: int
opcion: int
l_partidas: DefaultListModel<String>
lista_partidas: JList<String>
b_ok: JButton
b_cancel: JButton
b_borrar: JButton
l_nombrepartida: JLabel
t_nombrepartida: JTextField
list_p: String[]
contenedor: Container
Vista_salvarPartida(JFrame,String[])
iniciar_componentes():void
valueChanged(ListSelectionEvent):void
actionPerformed(ActionEvent):void
cerrar():void
consultar_opcion():int
consultar_partida():int
consultar_nombre():String
sonar_efecto():void
<<Java Class>>
Vista_registrar
vistas
contenedor: Container
b_ok: JButton
b_cancelar: JButton
l_contrasenya1: JLabel
l_contrasenya2: JLabel
l_nom: JLabel
t_nom: JTextField
t_contrasenya1: JPasswordField
t_contrasenya2: JPasswordField
opcio: int
p_nombre: String
p_contrasenya1: char[]
p_contrasenya2: char[]
Vista_registrar(JFrame)
visualizar():void
consultar_opcion():int
consultar_nombre():String
consultar_contrasenya1():char[]
consultar_contrasenya2():char[]
tancar():void
actionPerformed(ActionEvent):void
titols():void
botons():void
montar():void
sonar_efecto():void

10.3.3 Vista mesa del juego

Esta vista es la más compleja del juego y es la que muestra la pantalla mientras estas jugando una partida. Dicha clase fue programada a mano sin ningún editor tal y como en su momento se recomendaba en Prop.

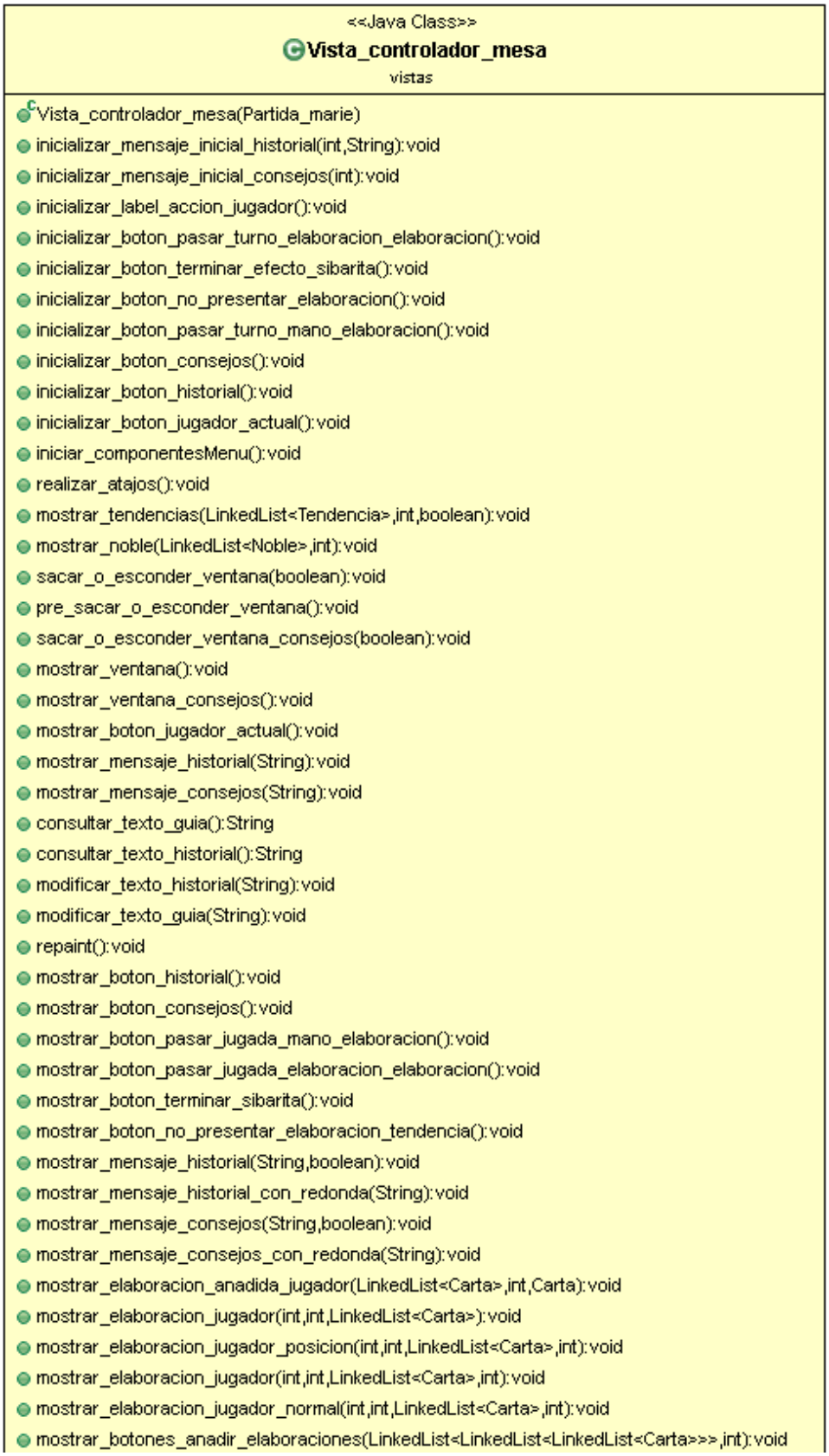
Esta clase dispone de numerosas funciones que son llamados por el controlador partida marie el cual se encarga de ordenar cuando mostrar o no las cosas. Podéis apreciar bastantes funciones que se llaman mostrar, esconder, quitar que se encargan de mostrar o quitar elementos según el controlador le ordene.

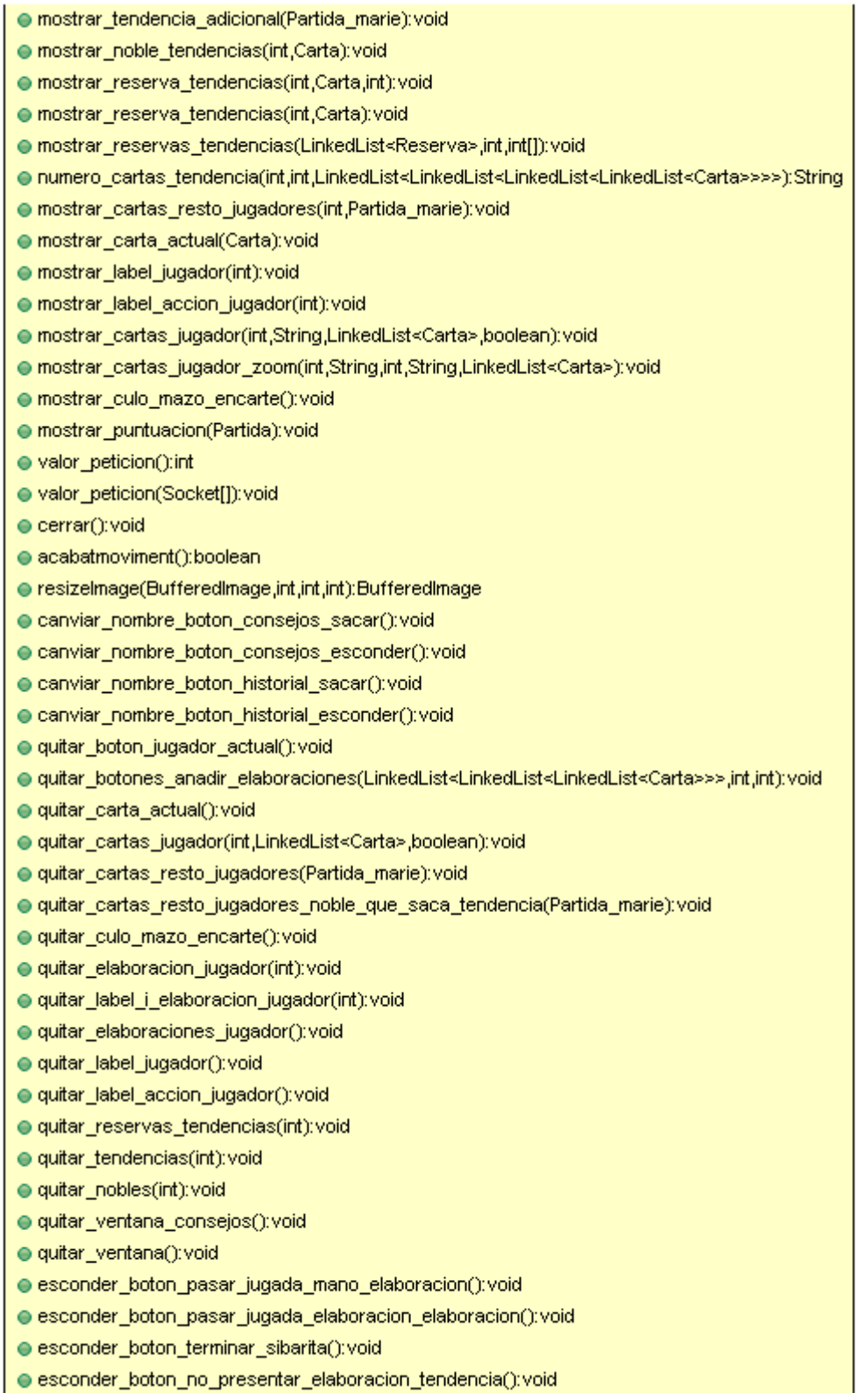
Primero se muestran las variables y luego las funciones. **(Variables)**

Una vez se han mostrado las variables de la vista se procede a mostrar las funciones.

(Funciones)

Esta clase dispone de numerosas funciones que son llamados por el controlador partida marie el cual se encarga de ordenar cuando mostrar o no las cosas. Podéis apreciar bastantes funciones que se llaman mostrar, esconder, quitar que se encargan de mostrar o quitar elementos según el controlador le ordene.




mostrar_tendencia_adicional(Partida_marie):void
mostrar_noble_tendencias(int,Carta):void
mostrar_reserva_tendencias(int,Carta,int):void
mostrar_reserva_tendencias(int,Carta):void
mostrar_reservas_tendencias(LinkedList<Reserva>,int,int[]):void
numero_cartas_tendencia(int,int,LinkedList<LinkedList<LinkedList<LinkedList<Carta>>>>):String
mostrar_cartas_resto_jugadores(int,Partida_marie):void
mostrar_carta_actual(Carta):void
mostrar_label_jugador(int):void
mostrar_label_accion_jugador(int):void
mostrar_cartas_jugador(int,String,LinkedList<Carta>,boolean):void
mostrar_cartas_jugador_zoom(int,String,int,String,LinkedList<Carta>):void
mostrar_culo_mazo_encarte():void
mostrar_puntuacion(Partida):void
valor_peticion():int
valor_peticion(Socket[]):void
cerrar():void
acabatmoviment():boolean
resizeImage(BufferedImage,int,int,int):BufferedImage
canviar_nombre_boton_consejos_sacar():void
canviar_nombre_boton_consejos_esconder():void
canviar_nombre_boton_historial_sacar():void
canviar_nombre_boton_historial_esconder():void
quitar_boton_jugador_actual():void
quitar_botones_anadir_elaboraciones(LinkedList<LinkedList<LinkedList<Carta>>>,int,int):void
quitar_carta_actual():void
quitar_cartas_jugador(int,LinkedList<Carta>,boolean):void
quitar_cartas_resto_jugadores(Partida_marie):void
quitar_cartas_resto_jugadores_noble_que_saca_tendencia(Partida_marie):void
quitar_culo_mazo_encarte():void
quitar_elaboracion_jugador(int):void
quitar_label_i_elaboracion_jugador(int):void
quitar_elaboraciones_jugador():void
quitar_label_jugador():void
quitar_label_accion_jugador():void
quitar_reservas_tendencias(int):void
quitar_tendencias(int):void
quitar_nobles(int):void
quitar_ventana_consejos():void
quitar_ventana():void
esconder_boton_pasar_jugada_mano_elaboracion():void
esconder_boton_pasar_jugada_elaboracion_elaboracion():void
esconder_boton_terminar_sibarita():void
esconder_boton_no_presentar_elaboracion_tendencia():void

Se pueden observar también varias consultoras que son las encargadas de informar al controlador partida marie donde se ha clicada. A partir de esa información el controlador piensa como actuar e informa lo que debe mostrar o quitar esta vista.