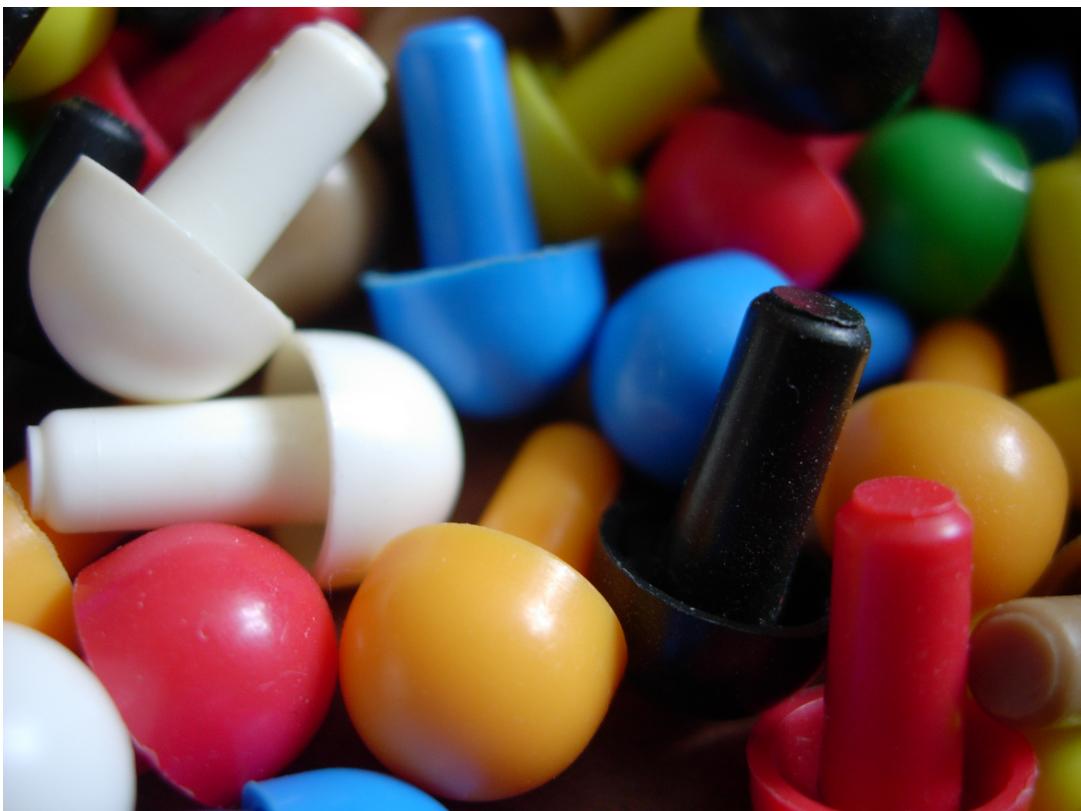


# MASTERMIND

---

## PROJECTE PROP



Versió 2.0

**Grup 24**

**Albert Carreira Muñoz** - albert.carreira  
**Ismael Haddad Ez-zarryaty** - ismael.haddad.ez-zarryaty  
**Jordi Nieto Maldonado** - jordi.nieto.maldonado

# Index:

<b>Index:</b>	<b>2</b>
<b>1. Manual d'usuari</b>	<b>3</b>
Modes:	3
Dificultats:	3
Normes:	3
Aplicació:	4
Com moure's pel menú	4
Nova Partida	4
Carregar Partida	4
Veure Rankings	4
<b>2. Diagrames complets:</b>	<b>6</b>
Domini i controladors del domini	6
Vistes i controladors de presentació	7
Gestors de disc	8
<b>3. Descripció d'estructures de dades i algorismes</b>	<b>9</b>
Classe Pair	9
Classe Codi (Abstracta)	9
Classes CodiFacil i CodiDificil	9
Classe Partida	10
Algorisme Five Guess	10
Classe ControladorPartida	11
Classe Jugada	11
Classe Ranking	11
Classe ControladorRanking	11
<b>4. Relació de les classes implementades per cada membre</b>	<b>12</b>

# 1. Manual d'usuari

Benvingut a **Mastermind**, un joc que posarà a prova les teves capacitats deductives! En aquesta aplicació podràs gaudir de dos modes de joc i dues dificultats diferents:

## Modes:

- *Code Breaker*: Modalitat clàssica de joc, l'ordinador et genera una solució i has d'aconseguir resoldre-la.
- *Code Maker*: Modalitat inversa. Introduceix la teva solució i observa si la màquina és capaç d'endevinar-la!

## Dificultats:

- *Fàcil*: Les combinacions tenen 8 colors disponibles, dels quals s'utilitzen 4. No hi ha repetits.
- *Difícil*: Les combinacions tenen 8 colors disponibles, dels quals s'utilitzen 6. Pot haver-hi un repetit de cada color.

## Normes:

Les **normes** per a jugar al Mastermind són les següents:

1. Cada partida està composada d'un codi solució i altres codis que hom prova per a poder arribar a endevinar-lo. Si aconsegueixes arribar a la solució correcta, hauràs guanyat!
2. Per a saber com de prop o de lluny estàs de la solució, la màquina et donarà un conjunt de boles blanques i negres (B/N). Si al teu codi hi ha una bola d'un color que apareix a la solució l'ordinador et donarà una bola blanca. Si, a més, aquesta bola està en el lloc corresponent se't donarà una bola negra en comptes de la blanca!

El joc acaba quan totes les boles que introduceixes són del mateix color que les de la solució i estan al seu lloc, és a dir, si aconsegueixes obtenir 4 boles negres.

3. Segons la dificultat escollida es poden formar els codis d'una forma diferent. Tingues-ho en compte a l'hora de pensar qui introuiràs!

## Aplicació:

Per tal de poder jugar s'ha desenvolupat una aplicació que et permetrà iniciar una nova partida, guardar la partida actual i carregar-ne d'anteriors i veure els rècords actuals.

### Com moure's pel menú

Passar de pantalla a pantalla és fàcil, simplement clica al botó que indica la opció que desitges. A cada pantalla existeix un botó per a tornar a la pantalla anterior. Si encara tens més dubtes, al menú principal existeix un apartat d'informació on es tornen a explicar les instruccions!



### Nova Partida

A la pantalla de nova partida veuràs que tens totes les opcions per a iniciar una partida en el mode i en la dificultat que desitgis. Simplement has de clicar als botons per a seleccionar les opcions i a jugar!

*Nota: La dificultat difícil en el mode Code Maker és no recomenable a causa del temps de processament per tal de què l'aplicació sigui totalment estable.*

### Carregar Partida

Aquí podràs veure la llista de les 10 partides que has guardat anteriorment. De la mateixa forma que en nova partida, selecciona la partida que vols carregar clicant sobre el nom i a continuació prem el botó de carregar.



Si seleccions l'opció de *Code Maker* se't demanarà que introdueixis la solució a la que ha d'arribar l'ordinador. Per tal d'aconseguir-ho, fes clic a qualsevol de les boles disponibles tot tenint en compte les restriccions de la dificultat. Pots canviar de color qualsevol bola (sempre i quan no trenquis les restriccions) i canviar de bola seleccionada clicant sobre la bola del panell que vols introduir.

### Veure Rankings

L'aplicació també et permet mantenir un sistema de rànkings o puntuacions per a cada una de les dificultats. Per a veure el rànking d'una dificultat fes clic al seu botó.

*Nota: La puntuació de cada partida es calcula a partir del nombre de jugades que has necessitat per arribar a la solució. Si has demanat ajuda, la puntuació s'incrementa en 10.*

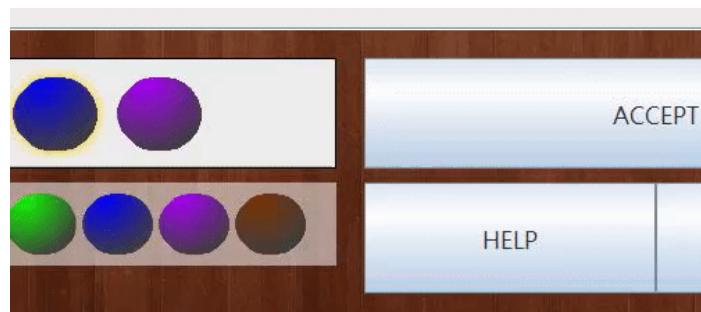
## En partida

Si has seleccionat la opció de *Code Maker* veuràs la llista de jugades que l'ordinador ha anat escollint fins arribar a la solució. Després només podràs tornar al menú principal.

Altrament, si has seleccionat la opció de *Code Breaker*, de forma igual a la introducció de la solució a la pantalla de nova partida, podràs anar formant les teves jugades fins arribar a la solució que s'ha generat aleatoriament.

Quan estiguis segur de què vols fer aquella jugada prem el botó d'acceptar, i l'ordinador t'ensenyarà quantes boles blanques i negres has aconseguit.

Si aconsegueixes obtenir 4 boles negres significarà que has guanyat i se't preguntarà el teu nom per a guardar el rècord en cas que tinguis una millor puntuació que els que n'hi ha actualment.



També pots demanar ajuda prement el botó corresponent, i l'ordinador et mostrerà una bola que segur que està a la solució i que no has introduït a la teva primera jugada amb millors resultats. Tingues present que demanar ajuda comportarà una penalització de +10 a la puntuació!

Finalment, si ho desitges pots abandonar la partida. Aleshores se't preguntarà si vols guardar la partida actual o no: si vols guardar-la hauràs d'introduir un nom però vigila! Perquè si ja tens 10 partides guardades se't borrerà la més antiga.

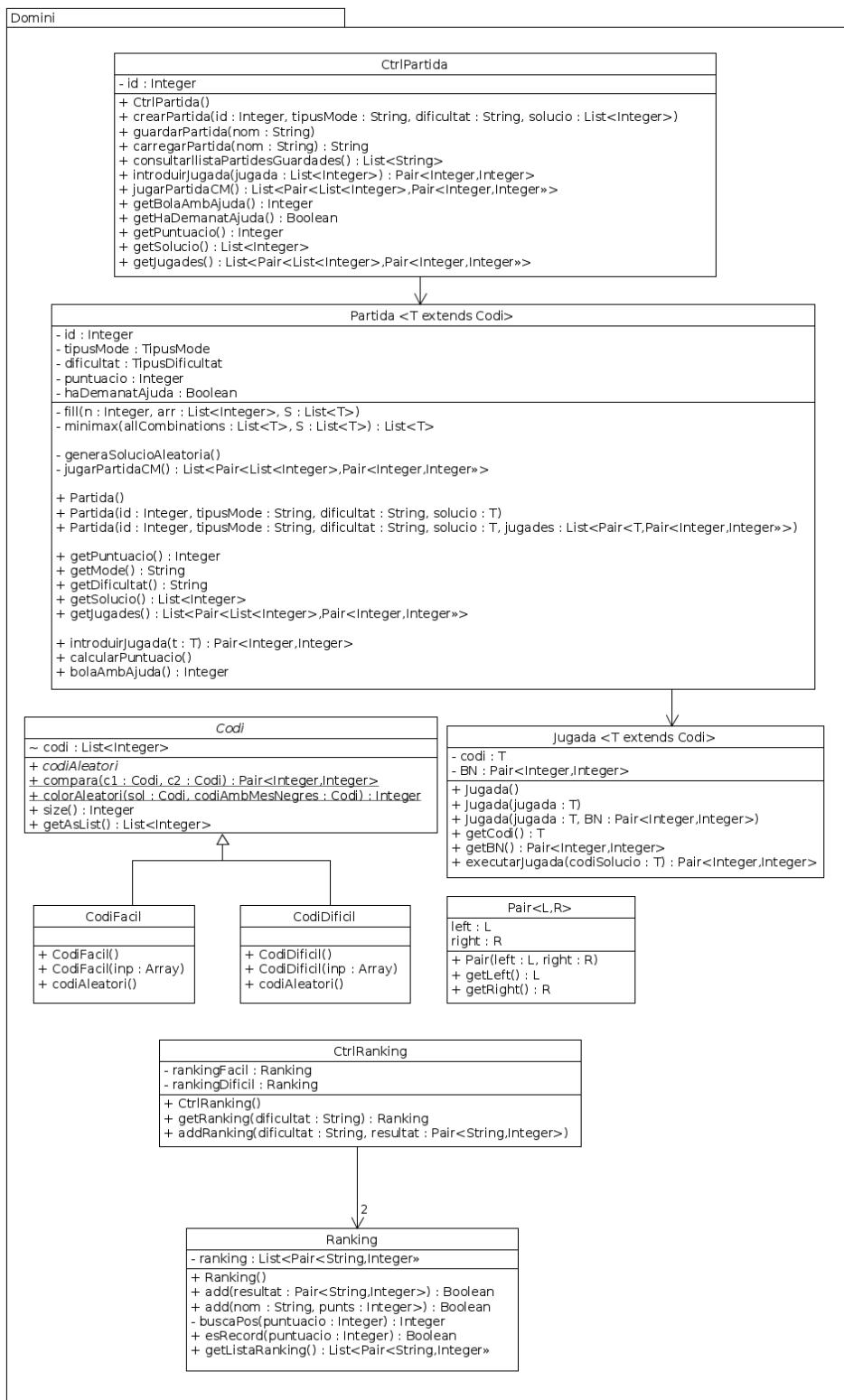
## About

Al menú principal també tens un botó que mostra els integrants que han desenvolupat l'aplicació.

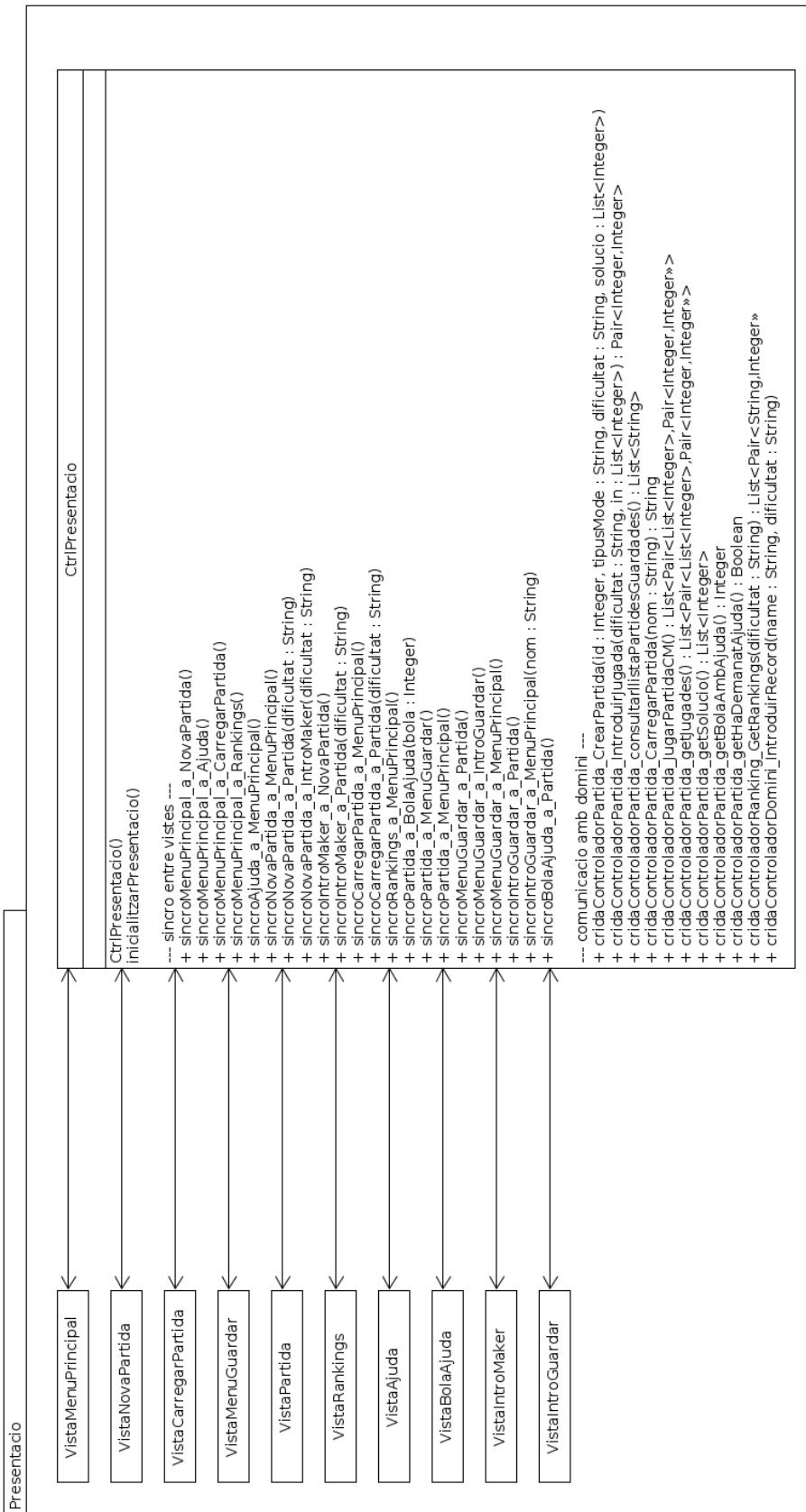
Res més, gaudet del joc!

## 2. Diagrames complets:

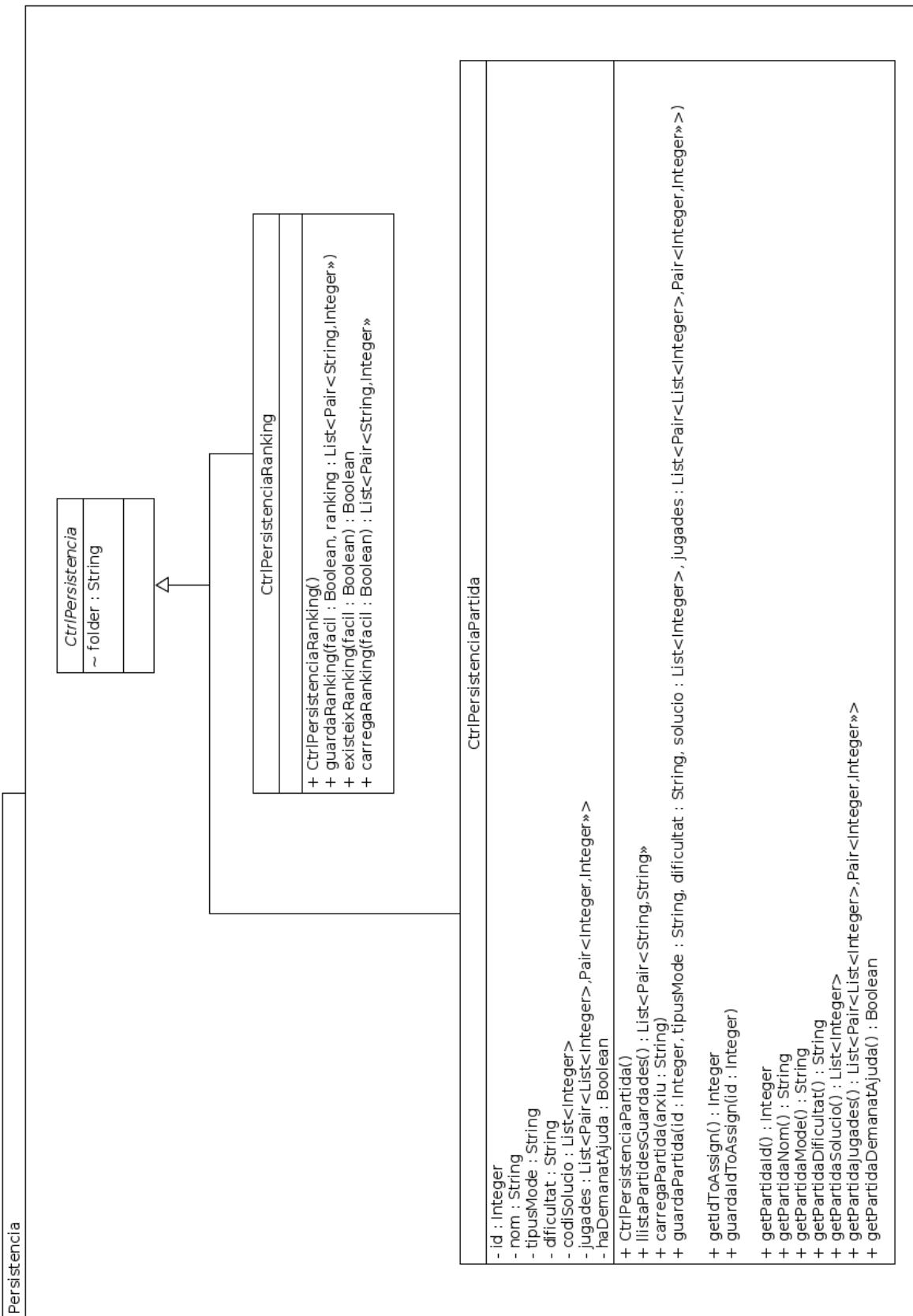
### Domini i controladors del domini



# Vistes i controladors de presentació



# Gestors de disc



### 3. Descripció d'estructures de dades i algorismes

#### Classe *Pair*

Hem implementat una classe d'ajuda genèrica *Pair* que, de forma similar a la que hi ha en C++, ens permet guardar parells d'objectes i treballar amb les dues parts (*left* i *right*). Necessitavem guardar el resultat de les jugades en una estructura de dades per a poder treballar fàcilment amb ells, i no ens convencia l'opció d'utilitzar una llista de només dos elements. També ens ha estat útil per solucionar la necessitat de guardar parells d'objectes en els registres del Ranking.

Així doncs, i donat que Java no implementa aquesta classe, vam decidir crear-la per tal d'utilitzar-la com si fos una classe genèrica que tingués el comportament que buscàvem i ens ajudés a complir el nostre objectiu.

#### Classe *Codi (Abstracta)*

Classe fonamental per a poder fer jugades de diferents dificultats i comparar-les. Implementa una sèrie de constants i una llista de nombres enters on es guarden els valors dels colors jugats (de llargària diferent segons la subclasse instanciada per dificultat), a més de un mètode abstracte que s'especificarà a cada subclasse. A més, en aquesta classe també s'implementa, un mètode estàtic de comparació de dos codis i dos *getters*: la primera ens retorna la llargària del codi (a partir de la qual podrem saber de manera fàcil en quina dificultat estem jugant) i la segona retorna el valor del Codi en Objectes Java Estàndards.

- L'algorisme utilitzat en el mètode de comparació de dos objectes Codi és un recorregut lineal del primer codi sobre el segon. Es compara utilitzant mateixa posició per trobar les boles negres i, en cas contrari, es fa un recorregut sobre el segon codi mirant si el color buscat està en una posició que no s'hagi visitat i que no hagi de ser negra. El resultat es retorna amb un objecte *Pair* amb les boles blanques i negres, respectivament.

#### Classes *CodiFacil* i *CodiDificil*

Ambdues són subclasses de *Codi*. Cada una assigna a la seva llista de colors una llargària diferent (4 per fàcil i 6 per difícil). En cas que no es compleixi la restricció de llargària, el codi es retalla en cas que la llargària sigui major o es posa un valor per defecte en cas que la llargària sigui menor. També s'implementa el mètode abstracte definit a la superclasse per tal de generar un codi amb valors aleatoris, assegurant que es compleixen les restriccions determinades per dificultat: a *CodiFacil* no pot haver-hi repetits però a *CodiDificil* n'hi pot haver un de cada color (són restriccions diferents).

## Classe *Partida*

Aquesta classe implementa totes les funcionalitats necessàries per a poder jugar una partida. Cal remarcar que és una classe genèrica que s'implementarà de forma diferent segons si volem fer jugades amb codis fàcils o difícils. Guarda els atributs indispensables per a poder jugar (identificador de la partida, dificultat, puntuació, mode de joc, codi solució, conjunt de jugades i si s'ha demanat ajuda) i té mètodes que permeten generar un codi aleatori i introduir una jugada. La funcionalitat més important, però, és la implementació de una IA, un algorisme capaç de solucionar el joc per a qualsevol solució (mode *CodeMaker*).

## Algorisme *Five Guess*

Algorisme implementat a la classe *Partida* que permet arribar a la solució proposada en com a molt 5 conjectures. Per a poder portar a terme aquest algorisme s'han codificat diversos mètodes: *fill*, que omple una llista amb totes les combinacions possibles segons el nombre de colors i la quantitat que n'utilitzem i les restriccions de repetits; *minimax*, que escolleix les millors combinacions per a continuar amb l'algorisme i arribar a la solució i *jugarPartidaCM*, que conté les inicialitzacions necessàries i el cos del bucle principal.

*Sempre es comença amb la jugada 1122 perquè està demostrat que es minimitza el nombre de conjectures necessàries per a arribar a la solució.*

L'algorisme segueix com a es detalla continuació:

1. Primerament s'inicialitzen totes les variables i llistes necessàries: el nombre de colors i els utilitzats, dos vectors (un pels filtrats i un altre que sempre contindrà totes les possibilitats) que inicialment contenen totes les possibles combinacions.
2. Seguidament es dona una jugada inicial per defecte, 1122(3344) com s'ha esmentat anteriorment.
3. Es borra la jugada del vector de filtrats i de totes les combinacions. Es compara la jugada amb la solució, si traiem 4 boles negres s'acaba
4. Si no, eliminem del vector de filtrats totes les possibilitats que no donarien la mateixa resposta si la jugada fos la solució: filtrem les combinacions.
5. Apliquem la tècnica *minimax* al vector de filtrats. Donada una combinació d'entre totes les possibles, la comparem amb cadascuna dels filtrats i anotem per cada resultat quantes vegades ha sortit. Aleshores agafem el resultat que més vegades s'ha repetit i apuntem la combinació amb el número de vegades que ha sortit. Per a cada codi tindrem una puntuació.

Repetim aquest pas per a totes les combinacions possibles.

Quan ja hem acabat agafem el nombre mínim d'entre totes les puntuacions i seleccionem totes aquelles combinacions que tenen aquest mínim com a puntuació.

6. De totes aquestes combinacions candidates seleccionem una, prioritant les que es troben en el vector de filtrats.
7. Tornar al pas 3 amb la jugada seleccionada.

## *Classe ControladorPartida*

Classe més exterior de la capa de domini que s'encarrega de comunicar la classe Partida amb les capes de Presentació i Persistència. Ens permet manipular la partida en sí, tot creant una de nova, guardant-la o carregant-la des del sistema de fitxers, o introduint noves jugades que ens donaran les Boles Blanques/Negres com a resultat (la qual cosa ens permet jugar).

## *Classe Jugada*

És la responsable de guardar la combinació introduïda pel jugador (CodiFacil / CodiDificil) i les boles Blanques i Negres resultants de comparar-ho amb la solució de la partida.

## *Classe Ranking*

Ens permet crear una llista amb el nom del jugador i la seva puntuació en la partida. Està ordenada per puntuacions en ordre creixent ja que, com la puntuació es el nº de jugades fetes a la partida, com menys se n'hagin fet millor serà el rècord. Els nostres rànquings tenen la restricció que només guarden els 10 millors registres.

Per implementar-la hem aprofitat que havíem creat la classe genèrica Pair per guardar-nos les dades en una llista de parells.

## *Classe ControladorRanking*

Classe que permet comunicar la classe de domini Ranking amb les capes de Presentació i Persistència. Permet administrar correctament la creació d'un rànquing per a cada dificultat i assegurar la correcta introducció i lectura de les dades en cadascun d'aquests. També s'encarrega de guardar i carregar les dades dels Rankings mitjançant persistència.

## Vista Intro Maker i Partida

Per a poder introduir la jugada desitjada s'ha implementat un petit algorisme que compara quina és la bola que està seleccionada per mitjà del camp ImageIcon i canvia limatge actual per una altra amb la bola del mateix color un pèl més gran. A continuació es canvia limatge de la següent bola per una del mateix color però amb un cèrcol brillant al voltant. S'aplica la mateixa acció si es selecciona amb el ratolí qualsevol bola de les de la solució. Després es canvien l'antiga bola destacada al seu estat normal.

## 4. Relació de les classes implementades per cada membre

<b>Albert</b>		
<b>Vistes</b>	<b>Domini</b>	<b>Tests</b>
Ajuda	Partida	JUnit
BolaAjuda	ControladorRanking * 1a Entrega	Driver
IntroGuardar		
IntroMaker		
MenuGuardar		
MenuPrincipal		
NovaPartida		
VistaRankings		
ControladorPresentacio		Driver (transicions) Driver (crides)

<b>Jordi</b>		
<b>Persistència</b>	<b>Domini</b>	<b>Tests</b>
	Pair	-----
	Codi	Driver
	CodiFacil	Driver
	CodiDificil	Driver
	Ranking	Driver
	Controlador Ranking * 2a Entrega	Driver
ControladorPersistència		-----
ControladorPersPartida		Driver
ControladorPersRanking		Driver

<b>Ismael</b>		
<b>Vistes</b>	<b>Domini</b>	<b>Tests</b>
	Jugada	Driver
	ControladorPartida	Driver
VistaCarregarPartida		
VistaPartida		
ControladorPresentacio		