

MASTERMIND®

Versió 1.2 (2ª entrega)

Projecte de Programació (PROP)

Identificador de l'equip: 14.1

Marc Sardà Masriera → marc.sarda.masriera@estudiantat.upc.edu

Aglaya Khalipskaya → aglaya.khalipskaya@estudiantat.upc.edu

Pol Farré Burgos → pol.farre.burgos@estudiantat.upc.edu

Pol Kallai Raventós → pol.kallai@estudiantat.upc.edu

26/05/2023

Índex

1. Capa de domini.....	3
1.1 Diagrama del model conceptual de dades.....	3
1.2 Descripció detallada de les classes.....	4
1.2.1 Utils.....	4
1.2.2 Usuari.....	5
1.2.3 ControladorUsuaris.....	6
1.2.4 ControladorJoc.....	7
1.2.5 ControladorDomini.....	11
1.2.6 Partida.....	14
1.2.7 InfoPartida.....	16
1.2.8 ControladorPartides.....	17
1.2.9 Codemaker.....	18
1.2.10 Codebreaker.....	19
1.2.11 CodebreakerMaquina.....	20
1.2.12 Interfície Maquina.....	21
1.2.13 Genetic.....	21
1.2.14 AlgortimeFiveGuess.....	22
1.2.15 Enumeration Dificultat.....	24
1.2.16 Enumeration Rol.....	25
1.2.17 Answer.....	25
1.2.18 Suggestion.....	26
1.2.19 Settings.....	27
1.2.20 RandomCodeBreaker.....	28
1.2.21 AnsweredIterator.....	28
1.2.21 Exhaustive.....	29
1.2.22 NoDuplicatesIterator.....	29
1.2.23 FilteredIterator.....	30
2. Capa de persistència.....	31
2.1 Diagrama del model conceptual de dades.....	31
2.2 Descripció detallada de les classes.....	31
2.2.1 ControladorPersistencia.....	31
2.2.2 ExcepcioPersistencia.....	32
3. Capa de presentació.....	33
3.1 Diagrama del model conceptual de dades.....	33
3.2 Descripció detallada de les classes.....	33
3.2.1 ControladorPresentacio.....	33
3.2.2 Main.....	36
3.2.3 MenuUsuari.....	37
3.2.4 MenuRegistrarUsuari.....	38
3.2.5 MenuSeleccioPartidaGuardada.....	38
3.2.6 ConfirmacioEsborrarPartida.....	39

3.2.7 IniciarSessioContrasenya.....	39
3.2.8 NormativaJoc.....	40
3.2.9 MenuSeleccioUsuari.....	41
3.2.10 MenuPausa.....	42
3.2.11 EsborrarUsuariContrasenya.....	42
3.2.12 SeleccioCodiSecret.....	43
3.2.13 Joc.....	44
3.2.14 MenuPartida.....	46
3.2.15 Ranquing.....	47
3.2.16 MenuConfigPartida.....	48
3.2.17 AlertaPopUp.....	48
3.2.18 BotoCircular.....	49
3.2.19 ExcepcioPersistencia.....	51
3.2.20 ExcepcioPartida.....	51
3.2.21 ExcepcioUsuari.....	52
4. Diagrama del model conceptual de dades complet.....	53
5. Estructures de dades i algorismes utilitzats.....	54
5.1 Estructures de dades.....	54
5.1.1 List<InformacioUsuari>.....	54
5.1.2 List<InformacioPartida>.....	54
5.1.3 Pair<L,R>.....	54
5.1.4 PartidaEnJoc.....	54
5.1.5 Mapa d'usuaris (ControladorUsuaris).....	55
5.1.6 Mapa de partides (ControladorPartides).....	55
5.1.7 Answer.....	55
5.1.8 Settings.....	55
5.8.9 Suggestion.....	55
5.2 Algorismes.....	55
5.2.1 Five-Guess.....	55
5.2.2 Genetic.....	59

1.2 Descripció detallada de les classes

1.2.1 Utils

Nom de la classe	Utils
Breu descripció de la classe	Aquesta classe conté mètodes estàtics que hem considerat útils i utilitzem en diferents funcions d'altres classes del nostre projecte per a efectuar diverses operacions.
Descripció dels atributs	Aquesta classe no té cap atribut definit.
Descripció dels mètodes	<p>numeroAleatori(Integer min, Integer max): Genera un nombre enter aleatori entre els dos valors, inclusivament, enters <i>min</i> i <i>max</i> passats per paràmetre. Retorna l'enter generat que està dins d'aquest rang.</p> <p>cls(): Imprimeix 20 salts de línia a la terminal de l'usuari, creant un efecte de netejar la terminal en moure el contingut mostrat anteriorment fora de la vista de l'usuari.</p> <p>printTitol(String t): Imprimeix un títol de manera decorativa per la terminal. Agafa el paràmetre de tipus String <i>t</i> que representa el text del títol i l'imprimeix rodejat per un separador de guions per tal de crear un efecte visual de títol per la terminal de l'usuari.</p> <p>llegirNum(String hint): Llegeix una cadena de text introduïda per a l'usuari per la terminal, la converteix a enter i la retorna.</p> <p>llegirString(String hint): Llegeix una cadena de text de la terminal i la retorna en una variable String. A més a més, valida que la cadena introduïda per l'usuari per a la terminal tingui com a mínim 3 caràcters abans de retornar-la.</p> <p>llegirchar(String hint): Llegeix un caràcter de la terminal i el retorna.</p> <p>hashSHA256(String base): Crea un hash SHA-256 a partir</p>

	<p>del string <i>base</i> passat com a paràmetre. Utilitza la classe <code>MessageDigest</code> de Java per a realitzar la funció de hash i retorna el resultat com a string.</p> <p>generateUID(): Genera un identificador únic universal (UUID) i el retorna com una cadena de text (String) i utilitza la classe <code>UUID</code> de Java per a generar-lo.</p> <p>compararCombinacions(List<Integer> combinacio, List<Integer> codiAnterior): Valida una combinació comparant-la amb una segona segons la posició dels paràmetres que les formen. Rep dos paràmetres, <i>combinacio</i> que és la combinació que volem avaluar i <i>codiAnterior</i> que correspon a la segona combinació amb la qual volem avaluar l'altra. Retorna el resultat de la comparació on el primer component són les posicions correctes i el segon, les incorrectes.</p> <p>difToString(Dificultat d): Converteix l'enum de dificultat <i>d</i> rebut com a paràmetre a una variable de tipus String que retorna.</p> <p>AnswersCounts(Settings settings): Calcula el nombre de respostes possibles per a desxifrar el codi. Reb com a paràmetre <i>settings</i> la configuració de la partida i retorna el nombre de possibles respostes per a desxifrar el codi.</p>
--	---

1.2.2 Usuari

Nom de la classe	Usuari
Breu descripció de la classe	La classe <code>Usuari</code> representa un usuari del joc i, per tant, conté totes les dades que formen un perfil d'usuari.
Descripció dels atributs	<p>uid: identificador únic de l'usuari.</p> <p>nomUsuari: nom de l'usuari.</p> <p>hContrasenya: hash de la contrasenya de l'usuari.</p> <p>maxPuntuacio: màxima puntuació històrica de l'usuari.</p>

	<p>numPartidesGuanyades: nombre de partides guanyades per l'usuari.</p> <p>numPartidesPerdudes: nombre de partides perdudes per l'usuari.</p>
Descripció dels mètodes	<p>Usuari(): Constructor per defecte de la classe.</p> <p>Usuari(String nomUsuari, String hContrasenya): Constructor de la classe Usuari que reb com a paràmetres el nom d'usuari i la contrasenya de l'usuari per a crear un nou objecte Usuari amb els valors assignats als atributs corresponents.</p> <p>numPartidesTotals(): Calcula i retorna el nombre total de partides jugades per l'usuari (guanyades + perdudes).</p> <p>actualitzaEstadistiques(Boolean guanyat, Integer puntuacio): Actualitza el nombre de partides guanyades/perdudes i la màxima puntuació, d'acord amb els valors passats com a paràmetres.</p>

1.2.3 ControladorUsuaris

Nom de la classe	ControladorUsuaris
Breu descripció de la classe	El ControladorUsuaris fa una funció similar al ControladorPartides, però, com el seu nom indica, amb usuaris. S'encarrega de gestionar tots aquests, continguts en un mapa, i de forma similar a l'anterior controlador, proporciona un seguit de mètodes i funcions perquè el ControladorDomini pugui interactuar amb ells.
Descripció dels atributs	<p>controladorUsuaris: instància estàtica del mateix controlador per garantir l'existència d'una sola instància del controlador durant l'execució (patró singleton).</p> <p>usuaris: Map<K,V> que emmagatzema usuaris (V) usant com a clau (K) el seu atribut uid.</p>

	<p>controladorPersistencia: instància del ControladorPersistencia per tal de desar les dades en un fitxer</p> <p>path: ruta el fitxer on desar les dades.</p>
Descripció dels mètodes	<p>crearUsuari(String user, String pwd): Crea un nou usuari i el guarda tant a <i>usuaris</i> (l'afegeix) com al disc mitjançant la capa de persistència.</p> <p>esborrarUsuari(String uid, String pwd): Esborra un usuari d'<i>usuaris</i> i de disc.</p> <p>actualitzaUsuaris(): Actualitza l'atribut usuari amb la informació guardada a disc que retorna la capa de persistència.</p> <p>getInstance(): Retorna controladorUsuaris i instància un abans si és nul.</p> <p>existeixUsuari(String): Comprova si existeix un usuari mitjançant un identificador</p> <p>guardarUsuari(Usuari): Guarda un usuari</p> <p>llistaRanquingUsuaris(): Obte llistaUsuaris() ordenada segons la maxima puntuacio</p> <p>llistaUsuaris(): Obte informacio dels usuaris (UID, nom, maxPuntuacio i nombre de partides jugades)</p> <p>nomUsuariEnUs(String): Comprova si un nom d'usuari esta en us</p> <p>retornaUsuari(String): Obtenir usuari amb username</p> <p>retornaUsuariContrasenya(String, String): Comprova paramtres i obte un usuari amb uid i contrasenya</p>

1.2.4 ControladorJoc

Nom de la classe	ControladorJoc
------------------	----------------

Breu descripció de la classe	El ControladorJoc és l'encarregat de dur a terme la gestió de la partida que s'estigui jugant. Aquest controla un usuari i una partida i proporciona al ControladorDomini diferents mètodes perquè l'usuari pugui interactuar amb la partida i aquesta pugui també ser jugada per un dels algorismes mitjançant les instàncies de <i>codebreaker</i> i <i>codemaker</i> que aquest controla.
Descripció dels atributs	<p>codemaker: Jugador encarregat de crear el codi secret en una partida.</p> <p>codebreaker: Jugador encarregat d'intentar endevinar el codi secret en una partida.</p> <p>partidaCarregada: Instància de la partida que s'està jugant actualment.</p> <p>usuariCarregat: Instància de l'usuari carregat que està jugant la partida en aquest moment.</p> <p>controladorJoc: instància estàtica del mateix controlador per garantir l'existència d'una sola instància del controlador durant l'execució (patró singleton).</p>
Descripció dels mètodes	<p>esborrarUsuari(): Invalida l'usuari carregat, posant-lo a <i>null</i>.</p> <p>usuariCarregat(): Comprova si hi ha un usuari carregat i retorna <i>true</i> si n'hi ha un o <i>false</i> en cas contrari.</p> <p>uidUsuari(): Obté l'identificador de l'usuari carregat i llença l'excepció ExcepcioUsuari si no hi ha cap usuari carregat.</p> <p>usernameUsuari(): Obté el <i>username</i> de l'usuari carregat i llença l'excepció ExcepcioUsuari en el cas que no hi hagi cap usuari carregat.</p> <p>carregarUsuari(Usuari usuari): Carrega un usuari mitjançant l'objecte de la classe Usuari passat com a paràmetre.</p> <p>partidaCarregada(): Comprova si hi ha una partida carregada, retornant <i>true</i> si n'hi ha una o <i>false</i> en cas contrari.</p> <p>esMaquinaCodemaker(): Comprova si la màquina té el rol</p>

	<p>de <i>CODEMAKER</i> retornant <i>true</i> si és el cas o <i>false</i> altrament.</p> <p>abandonarPartida(): Abandona la partida carregada, invalidant-la i invalidant els jugadors.</p> <p>carregarPartida(Partida p): Carrega una partida i instància els jugadors segons el rol escollit i llença l'excepció <i>ExcepcioPartida</i> en el cas que no hi hagi cap partida carregada.</p> <p>codiDificultatValid(codi: List<Integer> codi, Dificultat dificultat): Comprova si el codi passat com a paràmetre <i>codi</i> compleix la dificultat de la partida, retorna <i>true</i> si compleix les normes de la dificultat o <i>false</i> en cas contrari.</p> <p>informacioUsuariCarregat(): Obté la informació de l'usuari carregat i la retorna. Si no hi ha cap usuari carregat, fa saltar l'excepció <i>ExcepcioUsuari</i>.</p> <p>iniciarPartida(): Comprova els paràmetres de la partida carregada i inicia la partida. En el cas que no hi hagi cap partida carregada, llença l'excepció <i>ExcepcioPartida</i>. A més a més, retorna un objecte del tipus <i>PartidaEnJoc</i> que conté la informació de la partida en joc.</p> <p>tornCodemakerHuma(codiSecret: List<Integer>): Comprova els paràmetres i inicialitza el codi secret proporcionat per l'usuari en el cas que aquest tingui el rol de <i>codemaker</i>. Si no és el cas o no hi ha cap partida carregada o el <i>codiSecret</i> passat com a paràmetre no és vàlid o no correspon a la dificultat de la partida, llença l'excepció <i>ExcepcioPartida</i>. A més a més, retorna un objecte del tipus <i>PartidaEnJoc</i> que conté la informació de la partida en joc.</p> <p>tornCodemakerMaquina(): En aquest cas, la màquina té el rol de <i>codemaker</i> en la partida carregada i aquest mètode inicialitza el codi secret proporcionat per la màquina (generat aleatòriament) en la partida en joc. En el cas que no hi hagi cap partida carregada, llença l'excepció <i>ExcepcioPartida</i>. A més a més, aquest mètode retorna un objecte del tipus</p>
--	--

	<p>PartidaEnJoc que conté la informació de la partida en joc.</p> <p>tornCodebreakerHuma(codi: List<Integer>): Comprova els paràmetres i realitza un intent per a desxifrar el codi secret per part de l'usuari passat com a paràmetre codi. Si no hi ha cap partida carregada, el jugador no té el rol de <i>codebreaker</i> en la partida carregada o el codi passat com a paràmetre no és vàlid, llença l'excepció ExcepcioPartida. A més a més, retorna un objecte del tipus PartidaEnJoc que conté la informació de la partida en joc.</p> <p>infoPartidaEnJoc(): Comprova si la partida actual ha finalitzat i obté la informació de la partida de joc actual i la retorna en forma d'un objecte de la classe PartidaEnJoc amb les dades de la partida, l'usuari carregar, el codemaker i el codebreaker. En el cas que no hi hagi cap partida carregada, llença l'excepció ExcepcioPartida.</p> <p>recuperarPartidaCarregada(): Obté la informació de la partida que es troba carregada i la retorna en un objecte de tipus Partida. Si no hi ha cap partida carregada, llença l'excepció ExcepcioPartida.</p> <p>recuperarUsuariCarregat(): Obté la informació del perfil de l'usuari carregat i la retorna en un objecte de tipus Usuari. En el cas que no hi hagi cap usuari carregat, llença l'excepció ExcepcioUsuari.</p> <p>getInstance(): Retorna controladorJoc i instància un abans si és nul.</p> <p>codiJoc(): Obté el codi secret de la partida carregada</p> <p>usaPista(): Descompta la puntuació en funció del nombre de pistes ha usat l'usuari</p>
--	--

1.2.5 ControladorDomini

Nom de la classe	ControladorDomini
Breu descripció de la classe	El ControladorDomini és el cervell de capa de domini, ja que és l'encarregat de comunicar-se amb la capa de presentació. Disposa de molts mètodes per tal de realitzar la comunicació entre l'usuari i l'aplicació, però no calcula res per ell mateix sinó que crida a altres mètodes de controladors especialitzats, realitzant una funció d'encaminador de peticions.
Descripció dels atributs	controladorDomini: instància estàtica del mateix controlador per garantir l'existència d'una sola instància del controlador durant l'execució (patró singleton).
Descripció dels mètodes	usuariCarregat(): Comprova si hi ha un usuari carregat. Retorna cert si hi ha un usuari carregat o fals si no. crearUsuari(String username, String pwd): Crea un usuari amb el nom d'usuari i contrasenya especificats com a paràmetres. Llança una excepció "ExcepcioUsuari" si hi ha un error en la creació de l'usuari carregarUsuari(String uid, String pwd): Carrega un usuari amb l'identificador d'usuari i contrasenya especificats com a paràmetres. Llança una excepció "ExcepcioUsuari" si hi ha un error en la càrrega de l'usuari. esborrarUsuari(String uid, String pwd): Esborra un usuari amb l'identificador d'usuari i contrasenya especificats com a

	<p>paràmetres. Llança una excepció "ExcepcioUsuari" si hi ha un error en l'esborrat de l'usuari.</p> <p>llistaUsuaris(): Obté i retorna una llista amb informació de tots els usuaris, incloent-hi l'identificador d'usuari, nom d'usuari i puntuació màxima. Llança una excepció "ExcepcioUsuari" si hi ha un error en l'obtenció de la llista d'usuaris.</p> <p>llistaRanquingUsuaris(): Obté i retorna una llista d'informació d'usuaris ordenada per la puntuació màxima. Llança una excepció "ExcepcioUsuari" si hi ha un error en l'obtenció de la llista d'usuaris.</p> <p>logout(): Esborra l'usuari carregat, tancant la sessió.</p> <p>guardarUsuari(): Guarda l'usuari carregat en memòria i persistència. Llança una excepció "ExcepcioUsuari" si hi ha un error en la desada de l'usuari.</p> <p>informacioUsuariCarregat(): Obté informació de l'usuari carregat, incloent-hi el seu identificador d'usuari, nom d'usuari i puntuació màxima. Llança una excepció ExcepcioUsuari si no hi ha cap usuari carregat.</p> <p>partidaActiva(): Comprova si hi ha una partida carregada i retorna cert en cas que sí que n'hi hagi una o fals en cas contrari.</p> <p>crearPartida(Dificultat d, Rol r): Comprova que els paràmetres passats són correctes i crea la partida configurant-la segons la dificultat i el rol de l'usuari indicats. A més a més, fa saltar l'excepció ExcepcioUsuari si no hi ha cap usuari carregat en aquest moment i si no salta, retorna la indormació de l'estat de la partida en un atribut de tipus PartidaEnJoc.</p> <p>carregarPartida(String uidPartida): Carrega una partida amb l'identificador de partida especificat. Llança una excepció ExcepcioPartida si no es pot carregar la partida.</p>
--	---

	<p>guardarPartida(): Desa l'estat de la partida carregada en memòria i persistència. Llança una excepció ExcepcioPartida si hi ha algun error en l'operació de desament de la partida.</p> <p>abandonarPartida(): Tanca la partida carregada, eliminant-la de la memòria. Llança una excepció ExcepcioPartida si no hi ha cap partida carregada.</p> <p>llistaPartidesUsuari(String uid): Retorna una llista que cada posició d'aquesta correspon a un atribut del tipus InformacioPartida que conté la informació de les partides no acabades d'un usuari. En concret, l'usuari amb uid passat com a paràmetre. A més a més, fa saltar l'excepció ExcepcioUsuari si l'identificador d'usuari passat com a paràmetre <i>uid</i> no és vàlid o l'usuari amb aquest <i>uid</i> no existeix.</p> <p>iniciarPartida(): Inicia la partida carregada i retorna la informació de l'estat d'aquesta en un atribut del tipus PartidaEnJoc.</p> <p>iniciarCodi(List<Integer> codiSecret): Inicialitza el codi secret de la partida amb el codi passat com a paràmetre. A més a més, retorna la informació de l'estat de la partida en un atribut del tipus PartidaEnJoc.</p> <p>intentarCodi(List<integer> codi): Realitza un intent de trobar el codi secret de la partida amb el codi passat com a paràmetre. A més a més, retorna la informació de l'estat de la partida en un atribut del tipus PartidaEnJoc.</p> <p>infoPartidaEnJoc(): Retorna la informació de l'estat de la partida en joc en un atribut del tipus PartidaEnJoc.</p> <p>getInstance(): Retorna controladorDomini i instància un abans si és nul.</p> <p>codiJoc(): Obté el codi secret de la partida carregada</p> <p>esborrarPartida(String): Esborra la partida amb el uidP indicat</p> <p>informacioUsuariCarregat():Obte informacio de l'usuari</p>
--	--

	carregat usaPista(): Descompta la puntuació en funció del nombre de pistes ha usat l'usuari
--	---

1.2.6 Partida

Nom de la classe	Partida
Breu descripció de la classe	<p>Aquesta classe permet gestionar i mantenir l'estat d'una partida, així com efectuar les operacions necessàries com inicialitzar el codi secret, gestionar els intents, correccions i puntuació de la partida. També proporciona mètodes per obtenir i establir els atributs de la partida com ara l'estat, el número de torn, la informació de partida, el temps de partida, la puntuació, els intents i les correccions.</p>
Descripció dels atributs	<p>acabada: Booleà que indica si la partida ha acabat o no.</p> <p>numTorn: Enter que representa el número de torn actual de la partida.</p> <p>infoPartida: Objecte de la classe <i>InfoPartida</i> que conté la informació de la partida com a l'identificador d'usuari que la juga, l'identificador de la partida, la dificultat i el rol de cada jugador.</p> <p>tempsPartida: Un long que emmagatzema el temps transcorregut de la partida des del seu inici, en milisegons.</p> <p>puntuacio: Enter que representa la puntuació de la partida de l'usuari.</p> <p>intents: Llista de llistes d'enters que emmagatzema els intents realitzats pel jugador amb rol <i>codebreaker</i>.</p> <p>correccions: Llista de parells d'enters que emmagatzema les correccions dels intents realitzats pel jugador amb rol <i>codebreaker</i>.</p>

Descripció dels mètodes	<p>Partida(String uidPartida, String uidUsuari, Dificultat dificultat, Rol rol): Constructor de la classe que rep com a paràmetres l'identificador de partida, l'identificador d'usuari, la dificultat i el rol del jugador, i inicialitza els atributs de la classe com ara l'estat de la partida, el número de torn, la informació de partida, la puntuació i el temps de partida.</p> <p>inicialitzarCodiSecret(List<Integer> codiSecret): Inicialitza el codi secret de la partida amb una llista d'enters passada com a paràmetre.</p> <p>incrementaTorn(): Incrementa en 1 el número de torn de la partida.</p> <p>restaPuntuacio(Integer i): Resta un valor enter passat com a paràmetre a la puntuació de la partida.</p> <p>fi(): Estableix l'estat de la partida com a acabada.</p> <p>actualitzaTemps(Long l): Actualitza el temps de partida amb el valor passat com a paràmetre en mil·lisegons.</p> <p>intentarCodi(List<Integer> codi): Afegeix un intent realitzat pel jugador a la llista d'intents de la partida.</p> <p>esMaquinaCodemaker(): Retorna un booleà que indica si el rol del jugador és <i>codebreaker</i> o no.</p> <p>corretgirCodi(Pair<Integer, Integer> correccio): Afegeix una correcció a la llista de correccions de la partida.</p>
--------------------------------	--

1.2.7 InfoPartida

Nom de la classe	InfoPartida
Breu descripció de la classe	La classe InfoPartida representa els valors constants d'una partida com poden ser la dificultat d'aquesta, el rol que pren l'usuari, la data de creació de la partida, el nombre màxim de torns, els identificadors tant de l'usuari com de la partida o el codi secret escollit pel <i>codemaker</i> .

Descripció dels atributs	<p>uidPartida: Cadena de caràcters (String) que representa l'identificador únic de la partida.</p> <p>uidUsuari: String que representa l'identificador únic de l'usuari que ha iniciat la partida.</p> <p>dificultat: Objecte de la classe Dificultat que representa la dificultat de la partida.</p> <p>totalTorns: Enter que representa el nombre total de torns de la partida.</p> <p>dataInici: Objecte de la classe LocalDateTime que representa la data i hora d'inici de la partida.</p> <p>codiSecret: Llista d'enters que representa el codi secret del joc.</p> <p>rol: Objecte de la classe Rol que representa el rol de l'usuari en la partida.</p>
Descripció dels mètodes	<p>InfoPartida(String uidUsuari, String uidPartida, Dificultat dificultat, Rol rol): Constructor de la classe que rep com a paràmetres els identificadors d'usuari i partida, la dificultat i el rol de l'usuari. Aquest constructor inicialitza els atributs de la classe, com ara el total de torns, en funció de la dificultat especificada.</p>

1.2.8 ControladorPartides

Nom de la classe	ControladorPartides
Breu descripció de la classe	El ControladorPartides és l'encarregat de gestionar les partides existents, conté totes les instàncies d'aquestes en un mapa i proporciona informació d'aquestes mitjançant mètodes usats pel ControladorDomini.
Descripció dels atributs	<p>Partides: Atribut privat de tipus <i>Map</i> que emmagatzema les partides del joc. La clau del mapa és l'identificador de la partida (de tipus String), i el valor és una instància de la</p>

	<p>classe <i>Partida</i>. Aquest atribut s'utilitza per emmagatzemar i gestionar les partides creades pel controlador.</p> <p>controladorPartides: instància estàtica del mateix controlador per garantir l'existència d'una sola instància del controlador durant l'execució (patró singleton).</p> <p>controladorPersistencia: instància del ControladorPersistencia per tal de desar les dades en un fitxer</p> <p>path: ruta el fitxer on desar les dades.</p>
Descripció dels mètodes	<p>crearPartida(String uid, Rol r, Dificultat d): Crea una nova partida amb els paràmetres d'entrada (identificador del jugador <i>uid</i>, rol del jugador <i>r</i> i dificultat de la partida <i>d</i>). Genera un identificador únic per la partida utilitzant el mètode <code>Utils.generateUID()</code>, crea una nova instància de la classe <i>Partida</i> amb aquest identificador i els altres paràmetres, i l'afegeix a l'atribut <i>partides</i> utilitzant l'identificador de la partida com a clau. Finalment, retorna la partida creada.</p> <p>llistaPartidesUsuari(String uid): Retorna una llista d'objectes de la classe <i>InformacioPartida</i> amb la informació de les partides no acabades d'un usuari amb l'identificador d'entrada. Recorre totes les entrades de l'atribut <i>partides</i> i compara l'identificador de l'usuari de cada partida amb l'identificador d'entrada. Si coincideixen, afegeix una nova instància de la classe <i>InformacioPartida</i> amb la informació de la partida a la llista. Si no es troben partides pendents d'aquest usuari, llança una excepció de la classe <i>ExcepcioPartida</i>.</p> <p>guardaPartida(Partida p): Guarda la partida <i>p</i> passada com a paràmetre a l'atribut <i>partides</i>.</p> <p>getInstance(): Retorna controladorPartides i instància un abans si és nul.</p>

	<p>actualitzaPartides(): Inicialitza les dades amb informació de la capa de persistència</p> <p>esborrarPartida(String): Esborra la partida amb l'uid indicat</p> <p>obtePartida(String): Retorna la partida amb l'uid indicat.</p>
--	--

1.2.9 Codemaker

Nom de la classe	Codemaker
Breu descripció de la classe	La classe Codemaker representa el jugador encarregar de configurar el codi secret és a dir, el jugador que té el rol de <i>codemaker</i> en la partida.
Descripció dels atributs	Aquesta classe no té cap atribut declarat explícitament.
Descripció dels mètodes	<p>generarCodi(Dificultat dificultat): Genera un codi secret aleatòriament d'acord amb la dificultat de la partida. Rep un paràmetre de tipus <i>Dificultat</i> que indica el nivell de dificultat de la partida. El mètode utilitza la classe <i>Utils</i> per generar nombres aleatoris i crear una llista d'enters amb 4 nombres aleatoris, basant-se en la dificultat especificada. Retorna la llista de nombres generada com a codi secret.</p> <p>validarCodi(List<Integer> codi, List<Integer> solucio): Aquest mètode valida el codi introduït pel jugador amb rol <i>codebreaker</i> comparant-lo amb el codi secret del codificador (<i>codemaker</i>). Rep dues llistes d'enters com a paràmetres, <i>codi</i> que representa el codi introduït pel jugador <i>codebreaker</i> i <i>solucio</i> que representa el codi secret del codificador <i>codemaker</i>. Utilitza la classe <i>Utils</i> per comparar les dues combinacions de nombres i determinar el nombre de punts blancs i negres que s'han de retornar com a resultat. Si el codi introduït no té la longitud esperada, llança una excepció <i>ExcepcioPartida</i>. Retorna un objecte de tipus <i>Pair<Integer,</i></p>

	<i>Integer</i> > amb els punts blancs i negres com a components <i>L</i> i <i>R</i> respectivament.
--	---

1.2.10 Codebreaker

Nom de la classe	Codebreaker
Breu descripció de la classe	La classe Codebreaker representa el jugador encarregat de desxifrar el codi secret. És la superclasse de les classes CodebreakerHuma i CodebreakerMaquina que hereten un mètode per resoldre les combinacions secretes.
Descripció dels atributs	Aquesta classe no té cap atribut.
Descripció dels mètodes	solve(List<Integer> solution): Obté les combinacions utilitzades per descobrir el codi secret utilitzant un algoritme específic. Rep una llista d'enters com a paràmetre, <i>solution</i> , que representa el codi secret a descobrir. Aquest mètode té com a finalitat ser sobreescrit per altres classes filles, com per exemple <i>CodebreakerMaquina</i> , ja que es declara com a "null" i no té cap implementació en la classe "Codebreaker" en si. Aquest mètode pot llançar una excepció "Exception" en cas que es produeixi algun error durant la seva execució.

1.2.11 CodebreakerMaquina

Nom de la classe	CodebreakerMaquina
Breu descripció de la classe	Classe que hereta de la classe <i>Codebreaker</i> i representa el jugador màquina amb rol <i>codebreaker</i> que utilitza el mètode solve d'un algoritme de resolució per intentar endevinar el codi secret creat per l'usuari que juga la partida amb rol <i>codemaker</i> .
Descripció dels	algoritme: Atribut privat de tipus <i>Maquina</i> que representa

atributs	l'algoritme utilitzat per a la resolució automàtica del codi.
Descripció dels mètodes	solve(List<Integer> solution): Aquest mètode sobreescriu el mètode <i>solve</i> de la classe mare <i>Codebreaker</i> i s'utilitza per a obtenir les combinacions utilitzades per a descobrir el codi secret mitjançant l'algoritme específic assignat a l'atribut <i>algoritme</i> de la classe. Rep una llista d'enters com a paràmetre, <i>solution</i> , que representa el codi secret a descobrir. A més a més, el mètode pot llançar una excepció de tipus <i>Exception</i> en el cas que es produeixi algun error durant la seva execució.

1.2.12 Interfície Maquina

Nom de la interfície	Maquina
Breu descripció de la interfície	Interfície proporcionada pel professorat de PROP que declara un sol mètode. Representa una màquina que pot resoldre la combinació de colors creada per al jugador de la partida amb rol <i>codemaker</i> . La interfície defineix una operació <i>solve</i> que pren una llista de codis com a paràmetre i retorna una llista de llistes d'enters.
Descripció dels atributs	La interfície no té cap atribut.
Descripció dels mètodes	solve(List<Integer> solution): Utilitza un dels algoritmes proposats (sigui l'algoritme genètic o el five guess) per crear una llista de codis que portin a la solució. Si l'algoritme no pot trobar la solució en menys de <i>maxSteps</i> passos, la llista retornada contindrà una llista de codis amb <i>maxSteps</i> elements. L'operació pot llençar una excepció si la solució del codi secret no és consistent amb els paràmetres del joc.

1.2.13 Genetic

Nom de la classe	Genetic
Breu descripció de la classe	Aquesta classe implementa la interfície <i>Maquina</i> . És una classe final, la qual cosa significa que no pot ser heretada per altres classes. Està dissenyada per a resoldre una partida utilitzant l'algorisme genètic plantejat.
Descripció dels atributs	settings: És un objecte de la classe <i>Settings</i> que emmagatzema les configuracions per la partida. Aquest atribut és públic i, per tant, és accessible des de fora de la classe.
Descripció dels mètodes	Genetic(InfoPartida info): Constructor de la classe que rep un objecte <i>info</i> de la classe <i>InfoPartida</i> com a paràmetre. Aquest constructor crea una nova instància de la classe <i>Settings</i> i l'assigna a l'atribut <i>settings</i> de la classe. A part, crida al mètode <i>fromInfo()</i> de l'objecte <i>settings</i> per assignar-li les configuracions des de l'objecte <i>info</i> . solve(List<Integer> solution): És l'algorisme principal per a resoldre la partida. Rep una llista d'enters com a paràmetre <i>solution</i> i retorna una llista de llistes d'enters. Utilitza un bucle <i>do-while</i> per realitzar iteracions i resoldre la partida. Dins del bucle, crida a altres mètodes com <i>GetSuggestion()</i> i <i>AnswerRecieved()</i> de les classes <i>RandomCodeBreaker</i> i <i>Answer</i> respectivament, per obtenir suggeriments i processar les respostes. La llista de suggeriments és afegida a la llista <i>list</i> i és retornada al final del mètode.

1.2.14 AlgortimeFiveGuess

Nom de la classe	AlgortimeFiveGuess
-------------------------	--------------------

Breu descripció de la classe	<p>La classe <code>AlgortimeFiveGuess</code> és un dels algorismes utilitzats per a resoldre el joc, aquesta endevina una combinació secreta a partir de provar i avaluar combinacions segons uns paràmetres, la seva funció principal és retornar totes les combinacions utilitzades fins a arribar a la solució (en els menys intents possibles).</p>
Descripció dels atributs	<p>colorsTotals: És una llista que emmagatzema els colors disponibles per generar combinacions. Conté els números del 0 al 5, que representen la totalitat de colors possibles amb els quals es pot jugar.</p> <p>possiblesCombinacions: És una llista que emmagatzema totes les possibles combinacions que podrien ser la solució del joc. S'utilitza per generar totes les combinacions possibles combinacions amb repetició o sense, i es va modificant a partir que es descarten combinacions.</p> <p>totalCombinacions: És una llista que emmagatzema totes les combinacions que s'han generat i s'han provat, el resultat final són tots els intents fins a arribar a la solució.</p> <p>codiAnterior: És una llista que emmagatzema el codi utilitzat anteriorment per intentar resoldre el joc. Es fa servir per comparar les possibles combinacions generades amb aquesta i per saber la resolució de colors correctes i incorrectes.</p> <p>tamanyCombinació: És enter que indica la mida de la combinació.</p> <p>repeticions: És una variable booleana que indica si hi ha o no repeticions de colors a la combinació.</p> <p>nColors: És un enter que indica el nombre de possibles colors que hi ha a la combinació i que podem utilitzar per resoldre-la.</p> <p>resolucio: És un array de dos enters que indica les posicions correctes i incorrectes de l'intent anterior.</p>

	intents: Guarda el nombre total d'intents que pot emprar.
Descripció dels mètodes	<p>AlgoritmeFiveGuess (InfoPartida ip): Creadora per defecte de la classe.</p> <p>solve(List<Integer> codiSecret): Itera fins a trobar la solució o acabar els torns i retorna una estructura de dades amb totes les combinacions usades fins a arribar a la correcta.</p> <p>combinaciolnicial(List<Integer> novaCombinacio): Crea la combinació inicial que pot eliminar més possibilitats.</p> <p>getPossiblesCombinacions(): Retorna totes les possibles combinacions (atribut de la classe).</p> <p>generarCombinacionsRepetides(): Crea les estructures de dades necessàries per a generar totes les possibles combinacions amb repeticions.</p> <p>gCR(int: i, List<Integer> resposta): Backtracking per a generar totes les possibles combinacions d'una mida en concret i amb repeticions.</p> <p>generarCombinacionsNoRepetides(): Crea les estructures de dades necessàries per a generar totes les possibles combinacions sense repeticions.</p> <p>gCnR(int i, boolean[] visitats, List<Integer> resposta): Backtracking per a generar totes les possibles combinacions d'una mida en concret i sense repeticions.</p> <p>minmax(List<Integer> proximIntent): Avaluació de totes les combinacions i tria de la millor d'elles.</p> <p>compararCombinacions(List<Integer> combinacio, List<Integer> codiAnterior): Validació d'una combinació comparant-lo amb una segona segons la posició dels paràmetres que les formen.</p>

1.2.15 Enumeration Dificultat

Nom de l'enumeration	Dificultat
-----------------------------	------------

Breu descripció de la classe	És una enumeració que defineix els diferents nivells de dificultat del joc. Aquesta classe enumera tres valors constants: <i>FACIL</i> , <i>INTERMIG</i> i <i>DIFICIL</i> que representen els diferents nivells de dificultat disponibles.
Descripció dels atributs	No té atributs.
Descripció dels mètodes	No té mètodes addicionals, ja que és una enumeració i només defineix els valors constants per als nivells de dificultat.

1.2.16 Enumeration Rol

Nom de l'enumeration	Rol
Breu descripció de la classe	És una enumeració que defineix dos possibles valors: <i>CODEMAKER</i> i <i>CODEBREAKER</i> que representen els rols que poden tenir els jugadors del joc.
Descripció dels atributs	No té atributs, ja que és una enumeració, que és una llista fixa de valors constants.
Descripció dels mètodes	No té cap mètode, ja que és una enumeració, que només conté constants predefinides i no permet definir nous mètodes.

1.2.17 Answer

Nom de la classe	Answer
-------------------------	--------

Breu descripció de la classe	<p>És una estructura que salva els valors d'una resposta de codemaker i els paràmetres del joc. A més a més, té molts mètodes de conversió de valors per a comunicar amb el rest del sistema.</p>
Descripció dels atributs	<p>Settings: paràmetres del joc</p> <p>value: el valor de la resposta de codeMaker</p> <p>alfabeth: llista de caràcters per a la transformació del valor a String</p> <p>rand: objecte Random per a la generació dels valors aleatoris</p>
Descripció dels mètodes	<p>Answer(Settings): creadora per defecte de la classe</p> <p>Check(): comprova <i>value</i> perquè la seva mida i els valors siguin d'acord amb els Settings</p> <p>equalTo(Answer): compara el valor de l'Answer a <i>value</i></p> <p>fromAnswer(Answer): copia els valors de l'Answer a <i>value</i></p> <p>fromArray(Integer[]): copia els valors de l'array a <i>value</i></p> <p>FromPair(Pair<Integer, Integer>): copia els valors del Pair a <i>value</i></p> <p>fromString(String): copia els valors del String a <i>value</i>, passant-los a enters</p> <p>gameOver(): retorna <i>true</i> si <i>value</i> és d'un torn guanyador</p> <p>toArray(): retorna un array que conté els valors de <i>value</i></p> <p>toString(): retorna un String que conté els caràcters que representen els valors de <i>value</i></p>

1.2.18 Suggestion

Nom de la classe	Suggestion
-------------------------	------------

Breu descripció de la classe	És una estructura que salva els valors d'una combinació de colors i els paràmetres del joc. A més a més, té molts mètodes de conversió de valors per a comunicar amb la resta del sistema.
Descripció dels atributs	Settings: paràmetres del joc value: valors de la combinació alfabeth: llista de caràcters per a la transformació del valor a String rand: objecte Random per a la generació dels valors aleatoris
Descripció dels mètodes	Suggestion(Settings): creadora per defecte de la classe Check(): comprova <i>value</i> perquè la seva mida i els valors siguin d'acord amb els Settings equalTo(Suggestion): compara el valor del suggestion amb el <i>value</i> fromSuggestion(Suggestion): copia els valors de suggestion al <i>value</i> fromString(String): copia els valors del String al <i>value</i> , passant-los a enters fromList(List<Integer>): copia els valors del List a <i>value</i> toArray(): retorna un array que conté els valors de <i>value</i> toString(): retorna un String que conté els caràcters que representen els valors de <i>value</i> hasDuplicates(): retorna si el <i>value</i> té valors repetits increment(): incrementa el <i>value</i> un pas randomize(): genera el <i>value</i> aleatori toList(): retorna un List que conté els valors de <i>value</i>

1.2.19 Settings

Nom de la classe	Settings
-------------------------	----------

Breu descripció de la classe	Conté els paràmetres del joc per l'ús en l'algorisme Genètic
Descripció dels atributs	combinationLength: mida de combinació duplicationsAllowed: booleà per saber si estan permeses les repeticions de valors maxMoves: el nombre màxim de torns numLetters: el nombre de colors usats
Descripció dels mètodes	Settings(): creadora per defecte de la classe fromInfo(InfoPartida): copia els valors d'InfoPartida a Settings

1.2.20 RandomCodeBreaker

Nom de la classe	RandomCodeBreaker
Breu descripció de la classe	Conté tots els atributs i mètodes necessaris per a l'aplicació de l'algorisme genètic
Descripció dels atributs	settings: salva els paràmetres del joc nextSuggestion: salva la combinació per al torn següent variants: salva l'iterator corrent rand: objecte Random per a la generació dels valors aleatoris

Descripció dels mètodes	<p>RandomCodeBreaker(): creadora per defecte de la classe</p> <p>Cardinality(): retorna el nombre de les combinacions possibles en el moment</p> <p>AnswerRecieved(answer): instància una nova iteradora Answeredlterator i li passa els paràmetres necessaris per al càlcul de les combinacions possibles</p> <p>GetSuggestion(): retorna nextSuggestion</p> <p>PeekRandom(): retorna una combinació aleatòria de la llista de les combinacions possibles</p>
--------------------------------	---

1.2.21 Answeredlterator

Nom de la classe	Answeredlterator
Breu descripció de la classe	Itera per la llista de les combinacions possibles i aplica el filtre de comparació de respostes i exten Filteredlterator
Descripció dels atributs	<p>settings: salva els paràmetres del joc</p> <p>suggestion: salva la combinació corrent</p> <p>(underlay: salva l'iterador anterior)</p>
Descripció dels mètodes	<p>Answeredlterator(Suggestionslterator, Suggestion, Answer, Settings): creadora per defecte de la classe</p> <p>valid(): retorna si el valor de suggestion és vàlid</p>

1.2.21 Exhaustive

Nom de la classe	Exhaustive
-------------------------	------------

Breu descripció de la classe	Itera per la llista de totes les combinacions possibles i implementa SuggestionsIterator
Descripció dels atributs	settings: salva els paràmetres del joc current: salva la combinació corrent _eof: salva si l'iterador ha arribat al seu final
Descripció dels mètodes	Exhaustive(Settings): creadora per defecte de la classe currentValue(): retorna la combinació corrent eof(): retorna si l'iterador ha arribat al seu final next(): substitueix el corrent amb el suggestion següent reset(): reinicia l'iterador a l'estat inicial

1.2.22 NoDuplicatesIterator

Nom de la classe	NoDuplicatesIterator
Breu descripció de la classe	Itera per la llista de totes les combinacions possibles que no tenen valors repetits i exten FilteredIterator
Descripció dels atributs	No té atributs propis que no siguin de la classe pare
Descripció dels mètodes	NoDuplicatesIterator(Settings): creadora per defecte de la classe valid(): retorna si el valor de suggestion és vàlid

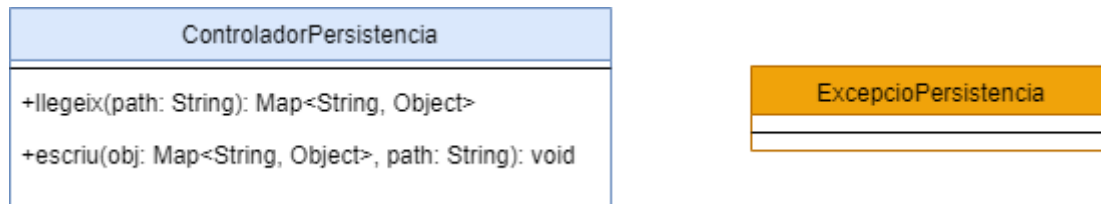
1.2.23 FilteredIterator

Nom de la classe	FilteredIterator
Breu descripció de la classe	Itera per la llista de totes les combinacions possibles que no tenen valors repetits i implementa SuggestionsIterator
Descripció dels atributs	underlay: salva un altre iterador
Descripció dels mètodes	FilteredIterator(SuggestionsIterator): creadora per defecte de la classe valid(): retorna si el valor de suggestion és vàlid

2. Capa de persistència

2.1 Diagrama del model conceptual de dades

Podeu trobar aquest diagrama al directori *DOCS* de l'entrega. En color cian es marquen els controladors i en color taronja les excepcions que s'usen en aquesta capa.



2.2 Descripció detallada de les classes

2.2.1 ControladorPersistencia

Nom de la classe	ControladorPersistencia
Breu descripció de la classe	Classe responsable de llegir i escriure objectes en el disc.
Descripció dels atributs	Aquesta classe no té cap mètode.
Descripció dels mètodes	<p>llegeix (path: String): Aquest mètode s'utilitza per a llegir un objecte des d'un fitxer especificat en el "path". Primer, verifica si l'arxiu existeix. Si no existeix, crea un arxiu nou i escriu un objecte buit en ell.</p> <p>escriu (obj: Map<String, Object>, path: String): Mètode que guarda un objecte en un arxiu especificat per la ruta "path". Utilitza fluxos de sortida per a escriure l'objecte en el fitxer, assegurant-se que les dades es guardin correctament. Si hi ha algun error, es llença una excepció "ExcepcioPersistencia" amb un missatge descriptiu.</p>

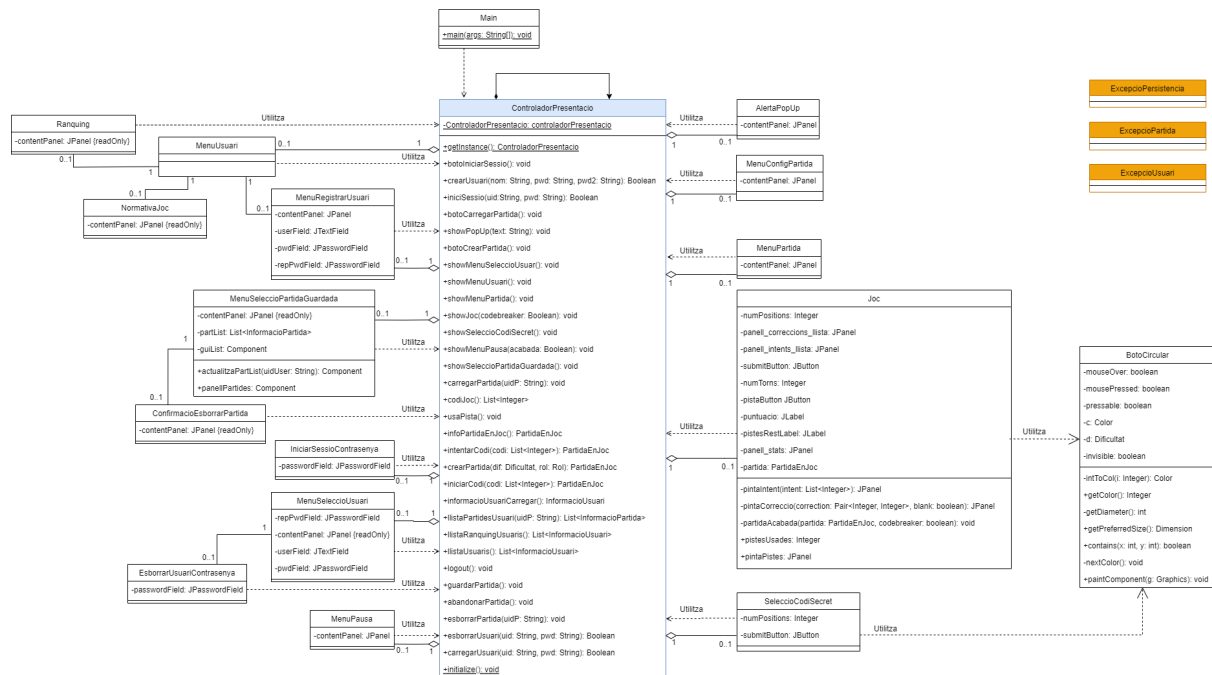
2.2.2 ExcepcioPersistencia

Nom de la classe	ExcepcioPersistencia
Breu descripció de la classe	La classe “ExcepcioPersistencia” és una extensió de la classe “Exception”. Es orientada a gestionar situacions inesperades o errors durant l’execució d’un programa. Forma part d’un mecanisme per capturar i notificar de l’errata. Conté informació rellevant sobre l’error, com un missatge descriptiu.
Descripció dels atributs	Aquesta classe no té atributs.
Descripció dels mètodes	Té dos constructors, un sense arguments i un altre que rep una cadena “s” com a missatge d’error. Aquests dos constructors permeten llançar i propagar excepcions personalitzades amb missatges descriptius relacionats amb problemes de persistencia en el joc.

3. Capa de presentació

3.1 Diagrama del model conceptual de dades

Podeu trobar aquest diagrama al directori *DOCS* de l'entrega. En color cian es marquen els controladors i en color taronja les excepcions que s'usen en aquesta capa.



3.2 Descripció detallada de les classes

3.2.1 ControladorPresentacio

Nom de la classe	ControladorPresentacio
Breu descripció de la classe	Aquesta classe representa el controlador de la capa de presentació. És la responsable de coordinar les diferents interaccions entre la interfície d'usuari i el controlador de domini.
Descripció dels atributs	controladorPresentacio: Atribut estàtic, que representa una instància única de la classe ControladorPresentacio.

**Descripció dels
mètodes**

getInstance(): Mètode estàtic que retorna la instància única de la classe ControladorPresentacio.

botonIniciarSessió(): Aquest mètode mostra el diàleg de selecció d'un usuari en fer clic en el botó d'iniciar sessió.

crearUsuari(nom: String, pwd: String, pwd2: String): Crea un nou usuari amb el nom, la contrasenya i la confirmació de la contrasenya (repetició) proporcionats. Retorna un valor booleà que indica si la creació de l'usuari ha sigut exitosa o no. A més a més, realitza les comprovacions necessàries com ara, la verificació de si el nom d'usuari proporcionat ja està en ús i si les contrasenyes coincideixen.

iniciSessio(uid: String, pwd: String): Inicia sessió amb l'UID d'usuari i la contrasenya proporcionats i retorna cert si s'ha iniciat sessió correctament o fals en cas contrari. Verifica si les credencials són vàlides i actualitza l'estat d'inici de sessió del sistema.

botoCarregarPartida(): Mostra el diàleg de selecció d'una partida guardada en fer clic al botó "Carregar partida" de la interfície d'usuari. Permet a l'usuari seleccionar una partida guardada prèviament per a continuar jugant-la.

showPopUp(text: String): Mostra una finestra emergent d'alerta amb el text proporcionat per la classe/funció que la crida. S'utilitza per a mostrar missatges informatius o d'error a l'usuari.

botoCrearPartida(): Mostra el diàleg de configuració d'una partida en fer clic en el botó "Crear partida". Permet a l'usuari configurar els paràmetres d'una nova partida, com la dificultat i el rol.

showMenuSeleccioUsuari(): Mostra el menú de selecció d'usuari. És cridat en prémer el botó "Carregar Perfil" i permet a l'usuari seleccionar el perfil d'usuari amb el qual vol iniciar sessió.

showMenuUsuari(): Mostra el menú d'usuari. És cridat

	<p>després d'iniciar sessió exitosament. Proporciona opcions com carregar una partida, veure el rànquing d'usuaris o eliminar l'usuari actual.</p> <p>showMenuPartida(): Mostra el menú de la partida. És cridat després de carregar o crear una partida. Proporciona opcions com intentar endevinar la combinació, utilitzar pistes o guardar/abandonar la partida.</p> <p>showJoc(codebreaker: Boolean): Mostra el joc amb el rol de Codebreaker si es proporciona el valor booleà <i>true</i>. És cridat en iniciar una partida i mostra la interfície del joc corresponent al rol del jugador.</p> <p>showSeleccioCodiSecret(): Mostra el diàleg de selecció de combinació secreta. Permet al jugador que assumeixi el rol de Codemaker seleccionar una combinació secreta per tal que el Codebreaker l'endevini.</p> <p>showMenuPausa(): Mostra el menú de pausa durant una partida. S'utilitza quan es pausa una partida i es mostren les opcions de guardar, continuar o abandonar la partida.</p> <p>showSeleccioPartidaGuardada(): Mostra el diàleg de selecció d'una partida. Permet a l'usuari seleccionar una partida guardada prèviament.</p> <p>carregarPartida(uidP: String): Mètode encarregat de carregar una partida guardada prèviament. Recupera les dades de la partida guardada i restaura l'estat del joc per a continuar en el mateix punt on es va deixar.</p> <p>intentarCodi(codi: List<Integer>): Utilitzat per a realitzar un intent d'endevinar el codi en el joc.</p> <p>crearPartida(dif: Dificultat, rol: Rol): Crea una nova partida amb la dificultat i rol proporcionats i retorna la informació de la partida en joc.</p> <p>iniciarCodi(codi: List<Integer>): Conté la lògica necessària per a configurar l'entorn del joc, carregar els recursos necessaris, mostrar la interfície d'usuari i preparar tot el</p>
--	---

	<p>necessari perquè l'usuari pugui començar a interactuar amb el joc.</p> <p>informacioUsuariCarregat(): Retorna la informació de l'usuari que té sessió iniciada actualment en el sistema (del perfil d'usuari que s'ha carregat).</p> <p>llistaPartidesUsuari(uidP: String): Retorna una llista d'informació de les partides de l'usuari amb uid proporcionat.</p> <p>llistaRanquingUsuaris(): Retorna una llista d'informació dels usuaris classificats segons la seva millor puntuació obtinguda una partida jugada.</p> <p>logout(): Tanca la sessió actual de l'usuari.</p> <p>guardarPartida(): Guarda la partida actual.</p> <p>abandonarPartida(): Abandona la partida actual que està jugant l'usuari en aquell moment.</p> <p>esborrarPartida(uidP: String): Esborra la partida amb l'identificador de partida proporcionat.</p> <p>llistaUsuaris(): Retorna una llista d'informació de tots els usuaris.</p> <p>esborrarUsuari(udi: String, pwd: String): Borra l'usuari amb l'UID d'usuari i la contrasenya proporcionats. Retorna un valor booleà que indica si s'ha eliminat l'usuari correctament.</p> <p>carregarUsuari(uid: String, pwd: String): Carrega l'usuari amb l'identificador UID i contrasenya pwd proporcionats. Retorna un valor booleà que indica si s'ha carregat correctament l'usuari o no.</p> <p>initialize(): Mètode d'inicialització que mostra el menú d'usuari a l'inici del programa.</p>
--	---

3.2.2 Main

Nom de la classe	Main
Breu descripció de la classe	Classe principal del programa que conté el mètode "main".

Descripció dels atributs	Aquesta classe no té cap atribut.
Descripció dels mètodes	main(): Mètode constructor de la classe responsable d'iniciar el programa i cridar al mètode d'inicialització del controlador de presentació per tal de començar l'execució de l'aplicació.

3.2.3 MenuUsuari

Nom de la classe	MenuUsuari
Breu descripció de la classe	Aquesta classe proporciona una interfície gràfica d'usuari on els usuaris poden carregar perfils existents, crear-ne de nous, accedir a la normativa del joc i veure el rànding de posicions dels usuaris registrats que han jugat alguna partida. La finestra té un disseny senzill i utilitza colors personalitzats per al fons i els components que la formen. A més a més, també mostra un missatge de benvinguda al joc a l'usuari i els noms dels membres de l'equip que han desenvolupat l'aplicació de joc.
Descripció dels atributs	Aquesta classe no té cap atribut.
Descripció dels mètodes	MenuUsuari(): Aquest mètode defineix la constructora de la classe "MenuUsuari" i s'encarrega d'inicialitzar la interfície gràfica del menú principal del joc Mastermind. Configura el títol de la finestra, el color de fons, el disseny dels components i crea els botons per a carregar un perfil d'usuari, crear un nou perfil d'usuari, veure les normes del joc i accedir al rànding. A més a més, gestiona els botons per a realitzar les accions corresponents a aquests com ara, carregar un perfil, crear un nou perfil o sortir del joc. També mostra informació sobre el projecte i els noms dels desenvolupadors d'aquest.

3.2.4 MenuRegistrarUsuari

Nom de la classe	MenuRegistrarUsuari
Breu descripció de la classe	Aquesta classe és un diàleg de registre d'usuari en una interfície d'usuari. Permet a l'usuari introduir un nom d'usuari, una contrasenya i repetir la contrasenya per tal de registrar-se en el sistema. En fer clic en el botó "Registrar usuari", es crida al controlador de presentació per a crear un nou usuari amb les dades proporcionades. Si el registre és exitós, el diàleg es tanca si no, salta una finestra emergent informant a l'usuari de l'error produït.
Descripció dels atributs	contentPanel: Panell que conté el contingut principal del diàleg. userField: Camp de text per introduir el nom d'usuari. pwdField: Camp de text per a introduir la contrasenya. repPwdField: Camp de text per a introduir la repetició de la contrasenya.
Descripció dels mètodes	MenuRegistrarUsuari(): Constructor de la classe. Crea una instància de diàleg de registre d'usuari. actionPerformed(e): Mètode que s'executa quan es realitza una acció en el diàleg. En aquest cas, es crida quan es prem el botó "Registrar usuari". Registra un nou usuari si els camps de nom d'usuari i contrasenya indicats són vàlids. cancelButton(): Mètode que s'executa quan es fa clic en el botó "Cancelar". Tanca el diàleg sense realitzar cap acció.

3.2.5 MenuSeleccioPartidaGuardada

Nom de la classe	MenuSeleccioPartidaGuardada
Breu descripció de la classe	Aquesta classe representa un diàleg per a seleccionar i carregar una partida guardada en la interfície de l'usuari.

Descripció dels atributs	<p>contentPanel: Panell de contingut de diàleg que conté els components visuals.</p> <p>partList: Llista d'objectes "InformacioPartida" que representa les partides guardades de l'usuari.</p> <p>guiList: Component visual que mostra la llista de partides guardades en el diàleg.</p>
Descripció dels mètodes	<p>actualitzarPartList (uidUser: String): Actualitza la llista de partides guardades de l'usuari especificat i retorna un component que representa la llista actualitzada.</p> <p>panellPartides(): Crea i retorna un component que mostra una llista de partides guardades amb la seva informació, com la dificultat, la puntuació i la data de creació de la partida.</p>

3.2.6 ConfirmacioEsborrarPartida

Nom de la classe	ConfirmacioEsborrarPartida
Breu descripció de la classe	Aquesta classe és una finestra de diàleg que es mostra per a confirmar l'eliminació d'una partida.
Descripció dels atributs	<p>contentPanel: Panell de contingut de diàleg que conté els components visuals.</p>
Descripció dels mètodes	<p>ConfirmacioEsborrarPartida (String: uidPartida, Integer: n): És la constructora de la classe. Rep dos paràmetres: uidPartida, que és l'identificador de la partida a eliminar, i n, que representa la quantitat de partides guardades. En aquest mètode es configura l'aspecte visual de la finestra de diàleg, s'afegeixen els components necessaris i es defineixen els listeners dels botons "Sí" i "No".</p>

3.2.7 IniciarSessioContrasenya

Nom de la classe	IniciarSessioContrasenya
-------------------------	--------------------------

Breu descripció de la classe	Aquesta classe representa la finestra d'inici de sessió d'un perfil d'usuari seleccionat a l'hora de carregar un perfil d'usuari existent. En ella, es demana a l'usuari que aquest introdueixi la contrasenya associada al perfil seleccionat a carregar.
Descripció dels atributs	passwordField: És un camp de contrasenya on l'usuari pot introduir la contrasenya associada al perfil d'usuari a carregar. Té associat un color de fons i està centrat horitzontalment.
Descripció dels mètodes	IniciarSessioContrasenya (username: String, uid: String): Aquest mètode és la constructora de la classe. Rep els paràmetres "username", que correspon al nom d'usuari seleccionat a l'hora de carregar un perfil d'usuari i pel qual se sol·licita la contrasenya i el paràmetre uid, que és l'identificador únic de l'usuari en el sistema. En aquest mètode, es configura l'aspecte visual de la finestra d'inici de sessió a un perfil, s'afegeixen els components necessaris per a aquesta i es defineixen els listeners dels botons "Acceptar" i "Cancel·lar".

3.2.8 NormativaJoc

Nom de la classe	NormativaJoc
Breu descripció de la classe	Aquesta classe representa una finestra de diàleg que mostra la normativa del joc Mastermind en la nostra aplicació.
Descripció dels atributs	contentPanel: Panell que actua com a contenidor principal dels elements de la finestra.
Descripció dels mètodes	NormativaJoc(): Mètode constructor sense paràmetres de la classe. En aquest, es configura l'aspecte visual de la finestra de normativa i s'afegeixen els components necessaris

	d'aquesta. S'estableix la posició i la mida de la finestra utilitzant la classe "Toolkit" per a obtenir la mida de la pantalla. Després es configuren el disseny i el contenidor principal utilitzant "BorderLayout" i "contentPanel". S'afegeix una vora buida al "contentpanel" i s'estableix un disseny nul per a permetre un posicionament absolut dels components.
--	---

3.2.9 MenuSeleccioUsuari

Nom de la classe	MenuSeleccioUsuari
Breu descripció de la classe	Finestra de selecció d'usuari en la interfície gràfica del joc. La finestra mostra una etiqueta gran amb el text "Selecciona l'usuari per a carregar" i obté la llista d'usuaris. Per a cada usuari, es mostra el seu nom, el nombre de partides jugades i la puntuació màxima en un panell separat. També s'afegeixen els botons "Seleccionar" i "Esborrar" per a interactuar amb cada usuari. Si se selecciona "Seleccionar", s'obre una finestra per a iniciar sessió amb la contrasenya de l'usuari. Si se selecciona "Esborrar", s'obre una finestra de confirmació per a esborrar l'usuari. Finalment, hi ha un botó "Enrere" per a retornar al menú principal de l'usuari.
Descripció dels atributs	contentPanel: Panell de contingut de diàleg que conté els components visuals de la finestra.
Descripció dels mètodes	MenuSeleccioUsuari (): Constructor de la classe sense paràmetres. En aquest, es configura l'aparença i els comportaments de la finestra de selecció d'usuari. S'estableix el fons de la finestra i s'obté la mida de la pantalla utilitzant la classe "Toolkit". Després, es configuren el disseny i el contenidor principal de la finestra utilitzant "BoxLayout". Es crea un "contentPanel" que actua com a panell principal de la

	finestra i s'estableix el seu disseny i vora. Dins del "contentPanel", s'afegeixen diversos components i panells per a mostrar la informació dels usuaris disponibles.
--	--

3.2.10 MenuPausa

Nom de la classe	MenuPausa
Breu descripció de la classe	Aquesta classe és una finestra emergent (JDialog) que representa el menú de pausa d'una partida. Té diversos components visuals emmagatzemats en un panell principal (contentPanel). Implementa una finestra de menú de pausa en el joc i ofereix opcions a l'usuari per a continuar, guardar i sortir o abandonar la partida actual.
Descripció dels atributs	contentPanel: Panell de contingut de diàleg que conté els components visuals.
Descripció dels mètodes	MenuPausa (): Mètode constructor de la classe encarregat d'inicialitzar i configurar la finestra de diàleg del menú de pausa del joc.

3.2.11 EsborrarUsuariContrasenya

Nom de la classe	EsborrarUsuariContrasenya
Breu descripció de la classe	És una finestra de diàleg que sol·licita a l'usuari confirmar l'eliminació d'un perfil d'usuari seleccionat. Mostra un missatge d'advertència i un camp de contrasenya per a confirmar l'acció. També inclou botons per a realitzar l'eliminació o cancel·lar-la.
Descripció dels atributs	passwordField: Objecte de la classe JPasswordField que representa un camp de text emmascarat utilitzat per introduir a contrasenyes. En la classe "EsborrarUsuariContrasenya",

	aquest camp s'utilitza perquè l'usuari introdueixi la contrasenya i confirmi l'eliminació de l'usuari seleccionat.
Descripció dels mètodes	EsborrarUsuariContrasenya(): Aquest mètode és la constructora de la classe, per tant, inicialitza una instància d'aquesta. Configura l'aparença i la disposició de la finestra de diàleg. Crea i agrega components com ara etiquetes, panells i botons al contingut de la finestra. A més a més, estableix els gestionadors d'esdeveniments per als botons "Esborrar" i "Cancelar". Permet a l'usuari introduir la contrasenya i realitzar l'acció corresponent com borrar el perfil d'usuari seleccionat o cancel·lar l'operació.

3.2.12 SeleccioCodiSecret

Nom de la classe	SeleccioCodiSecret
Breu descripció de la classe	La classe "SeleccioCodiSecret" és una finestra de diàleg que permet a l'usuari seleccionar una combinació secreta. La finestra té una aparença específica i una disposició de components com etiquetes, panells i botons. L'usuari pot seleccionar els colors desitjats mitjançant botons circulars i, després, pot confirmar la selecció amb el botó "Escollir combinació". Hi ha també un botó "Enrere" per abandonar la selecció i tornar enrere.
Descripció dels atributs	submitButton: Aquest atribut és un objecte de la classe "JButton" que representa el botó "Escollir combinació" a la finestra de diàleg. Aquest botó permet a l'usuari confirmar la selecció de la combinació secreta. S'hi associa un gestor d'esdeveniments per a gestionar l'acció de l'usuari en prémer el botó.

Descripció dels mètodes	<p>SeleccioCodiSecret (): Aquest mètode és el creador de la classe i com a tal, inicialitza una instància d'aquesta. Configura el títol de la finestra de diàleg i l'aparença de la interfície gràfica. Crea i afegeix components com panells, botons i objectes de la classe "BotoCircular" al contingut de la finestra. Estableix els gestadors d'esdeveniments dels botons "Escollir combinació" i "Enrere". Permet a l'usuari seleccionar una combinació de colors i realitzar l'acció corresponent com ara iniciar el joc o retornar a la pantalla anterior.</p>
--------------------------------	--

3.2.13 Joc

Nom de la classe	Joc
Breu descripció de la classe	<p>Aquesta classe que s'estén de "JFrame" i representa la finestra de joc, conté diversos components gràfics com ara botons, etiquetes i panells per a mostrar la informació relacionada amb la partida en curs i fer accions relacionades amb el joc.</p>
Descripció dels atributs	<p>numPositions: És un atribut de tipus enter que s'inicialitza a 4 en totes les partides del joc, ja que determina el nombre de colors que pot tenir la combinació que escull l'usuari en fer un torn.</p> <p>panell_correccions_llista: És un panell de la pantalla de joc de la interfície d'usuari que mostra les correccions de cada intent realitzat pel codebreaker en la resolució de la combinació de colors.</p> <p>panell_intents_llista: Forma part de la pantalla de joc de la interfície d'usuari. Mostra els intents de les combinacions de colors que el codebreaker fa en cada torn d'una partida.</p>

	<p>submitButton: És un botó que forma part de la pantalla de joc d'una partida que serveix per enviar la combinació de colors realitzada pel codebreaker en un intent.</p> <p>numTorns: És un enter que emmagatzema el nombre de torns que té el codebreaker per a endevinar la combinació de colors escollida pel codemaker en una partida. Aquest número depèn de la dificultat de la partida escollida per l'usuari en crear-ne una de nova. Si la dificultat és fàcil val 12, si és intermitja val 10 i si és difícil, 8.</p> <p>pistaButton: Botó de la pantalla de joc d'una partida que serveix perquè l'usuari pugui demanar una pista per a la resolució de la combinació de colors.</p> <p>puntuació: Etiqueta que mostra en tot moment, la puntuació de l'usuari en la realització de la partida.</p> <p>pistesRestLabel: Etiqueta que mostra en tot moment, el nombre de pistes restants que li queden a l'usuari per a demanar.</p> <p>panell_stats: Panell d'estat de la partida, que conté les etiquetes de puntuació i pistes restants de l'usuari descrites anteriorment.</p> <p>partida: Objecte de la classe "PartidaEnJoc" que representa i conté tota la informació de la partida en joc.</p>
Descripció dels mètodes	<p>pintaIntent(intent: List<Integer>): Mètode que crea i retorna un panell que representa un intent de l'usuari en la partida en joc. Aquest, mostra els botons circulars corresponents als colors seleccionats per a l'usuari amb el rol codebreaker en l'intent.</p> <p>pintaCorreccio(correction: Pair<Integer, Integer>, blank: boolean): Mètode privat que crea i retorna un panell que representa la correcció de la combinació de colors realitzada pel codebreaker en cada intent de la partida. El panell mostra botons circulars que indiquen la quantitat de posicions de la</p>

	<p>combinació correctes. Els botons que són de color groc indiquen que el color i la posició d'aquest són correctes i els de color blanc, indiquen que el color és correcte però la seva posició respecte a la combinació escollida pel codemaker no.</p> <p>partidaAcabada(partida: PartidaEnJoc, codebreaker: boolean): Mètode privat que mostra una finestra emergent de notificació per a indicar el final de la partida. Depenent del valor del paràmetre "codebreaker" i l'estat de la partida, es mostra un missatge adequat al rol de jugador de l'usuari.</p> <p>pistesUsades: Mètode privat que calcula i retorna el nombre de pistes usades durant la partida en funció de la puntuació i el nombre de torns d'aquesta.</p> <p>pintaPistes: Crea i retorna un panell que representa les pistes demanades per a l'usuari durant la partida. Aquest panell mostra els botons circulars corresponents a les fitxes de la combinació secreta revelada com a pistes.</p>
--	---

3.2.14 MenuPartida

Nom de la classe	MenuPartida
Breu descripció de la classe	<p>Aquesta classe és una finestra de la interfície gràfica que mostra un menú principal per a l'usuari. Configura l'aparença i la disposició de la finestra. Conté etiquetes de títol i missatges de benvinguda per a l'usuari. A més a més, presenta botons que permeten a l'usuari realitzar accions com ara, iniciar una nova partida, carregar una partida existent, tancar la sessió i tancar l'aplicació. Els botons estan associats a un gestor d'esdeveniments que realitzen les accions corresponents quan s'activen en prémer un botó.</p>
Descripció dels atributs	contentPanel: Panell de contingut de diàleg que conté els components visuals.

Descripció dels mètodes	<p>MenuPartida(): Mètode constructor de la classe que inicialitza una instància de la finestra del menú principal. Configura l'aparença i la disposició de la finestra tenint en compte la mida i la ubicació d'aquesta. Crea i afegeix components com etiquetes i botons al contingut de la finestra. Estableix els controladors d'esdeveniments per als botons que permeten a l'usuari interactuar amb el menú i realitzar accions com ara iniciar una nova partida, carregar una partida existent, tancar la sessió del perfil d'usuari carregat prèviament i tancar l'aplicació. També mostra informació rellevant com el nom d'usuari i la posició d'aquest en el rànquing segons la seva puntuació màxima.</p>
--------------------------------	---

3.2.15 Ranquing

Nom de la classe	Ranquing
Breu descripció de la classe	<p>Aquesta classe és un diàleg que mostra el rànquing dels millors jugadors del joc. La finestra té un fons de color i una disposició de contingut establerta. Mostra una etiqueta amb el títol del rànquing i crea panells per a cada usuari amb la seva posició, el nom, el nombre de partides jugades i la màxima puntuació aconseguida. Si hi ha usuaris en el rànquing, es mostra un panell desplegable. Finalment, inclou un botó "Enrere" que permet a l'usuari tancar el diàleg.</p>
Descripció dels atributs	<p>contentPanel: Panell de contingut de diàleg que conté els components visuals.</p>
Descripció dels mètodes	<p>Ranquing (): El mètode constructor d'aquesta classe inicialitza i configura la finestra del diàleg. Estableix la mida i la disposició de la finestra així com el color de fons. A més a més, crea i organitza els components de la interfície d'usuari (etiquetes i panells) per a mostrar el rànquing dels millors jugadors.</p>

3.2.16 MenuConfigPartida

Nom de la classe	MenuConfigPartida
Breu descripció de la classe	És una finestra de diàleg que permet a l'usuari configurar una nova partida. La finestra té un disseny i color de fons específics. Conté les etiquetes i panells de seleccionar la dificultat i el rol del jugador en la partida. Utilitza "JComboBox" per a mostrar les opcions de dificultat i rol disponibles. A més a més, hi ha dos botons: "iniciar partida" i "Cancelar". Aquest primer botó crea una partida amb la configuració establerta per a l'usuari i mostra la interfície de joc corresponent a aquesta. En canvi, el botó "Cancelar" tanca la finestra i retorna al menú de partida anterior.
Descripció dels atributs	contentPanel: Panell de contingut de diàleg que conté els components visuals.
Descripció dels mètodes	MenuConfigPartida (): Configura la finestra de diàleg, establint el color de fons i les dimensions d'aquesta. Afegeix i configura elements visuals com etiquetes i quadres desplegable. També defineix els "ActionListener" pels botons "Iniciar partida" i "Cancelar" és a dir, aquest mètode crea i configura tots els components necessaris per a la finestra de configuració de partida.

3.2.17 AlertaPopUp

Nom de la classe	AlertaPopUp
Breu descripció de la classe	Aquesta classe llança una finestra emergent tipus "pop up" mostrant un missatge concret per pantalla.
Descripció dels atributs	contentPanel: Panell de contingut de diàleg que conté els components visuals.

Descripció dels mètodes	<p>AlertaPopUp (text: String): Crea una finestra emergent de diàleg que mostra un missatge d'alerta. Estableix el títol de la finestra com "Atenció!" i el tipus com a "POPUP". Defineix les dimensions de la finestra tenint en compte la mida de la pantalla. Crea un panell de contingut amb un color de fons específic i una vora buida. Afegeix un component "JLabel" amb el text rebut com a paràmetre, el qual es mostra en format HTML per permetre salts de línia. En resum, el mètode constructor crea una finestra emergent amb un missatge d'alerta personalitzat.</p>
--------------------------------	---

3.2.18 BotoCircular

Nom de la classe	BotoCircular
Breu descripció de la classe	<p>Subclasse de JButton que representa un botó circular amb funcionalitat interactiva. Permet canviar de color al ser pressionat. Té mètodes per a gestionar el comportament del botó, obtenir i establir el color actual d'aquest i determinar si es pot pressionar o invisible. Controla la seva aparença visual mitjançant mètodes de dibuix utilitzant la classe Graphics.</p>
Descripció dels atributs	<p>mouseOver: Atribut booleà que indica si el mouse (ratolí) es troba sobre el botó. Aquest atribut s'utilitza per a controlar l'aparença visual del botó quan el ratolí està sobre aquest.</p> <p>mousePressed: Atribut booleà que indica si el botó està sent pressionat. S'utilitza per a canviar de color el botó quan aquest es prem.</p> <p>pressable: Atribut de tipus booleà que indica si el botó es pot pressionar o no és a dir, si l'usuari pot fer clic en ell. Si el valor de l'atribut és true, cert, el botó pot canviar de color en ser pressionat. Per altra band, quan el seu valor és false el botó no respondrà a les interaccions del mouse.</p>

	<p>c: Objecte de la classe “Color” que representa el color actual del botó quan aquest es dibuixa a la pantalla de la interfície d’usuari.</p> <p>d: Objecte de l’enumeració “Dificultat” que representa la dificultat associada al botó. Aquest atribut s’utilitza per a determinar com canviarà el color del botó quan aquest es pressioni, depenent de la dificultat establerta podrà ser d’un color o un altre.</p> <p>invisible: Atribut booleà que indica si el botó és visible o no. Si el valor d’aquest atribut és “true”, cert, el botó no serà visible i no respondrà a les interaccions del mouse que faci l’usuari.</p>
Descripció dels mètodes	<p>intToCol(i: Integer): Mètode privat que rep un valor enter “i” i retorna un objecte de la classe “Color” corresponent al valor enter proporcionat. S’utilitza per a mappejar enters a colors predefinits.</p> <p>getColor(): Mètode que retorna un valor enter que representa el color actual del botó. El valor enter s’assigna segons la correspondència establerta en el mètode “intToCol()”.</p> <p>getDiameter(): Mètode privat que calcula i retorna el diàmetre del botó circular. El diàmetre es calcula com el valor mínim entre l’amplada i l’altura del botó.</p> <p>getPreferredSize(): Mètode que anul·la el mètode “getPreferredSize()” de la classe “JButton” i retorna les dimensions predefinides del botó. Les dimensions es calculen d’acord amb la mida del text que es mostrarà en el botó.</p> <p>contains(x: Int, y: Int): Aquest mètode anul·la el “contains(int x, int y)” de la classe base “JButton” i verifica si les coordenades (x, y) estan dins de l’àrea circular del botó. Retorna true, cert, si les coordenades estan dins del botó i fals en cas contrari.</p>

	<p>nextColor(): Aquest mètode privat canvia de color el botó al següent color en funció de la dificultat establerta per a l'usuari en crear la partida.</p> <p>paintComponent(g: Graphics): S'encarrega de dibuixar/pintar el botó en el component gràfic "g" passat com a paràmetre. Controla el color de fons, el contorn i el text del botó segons els estats i atributs actuals de la posició del ratolí, si el botó es pot pressionar o és invisible, i del color actual del botó.</p>
--	---

3.2.19 ExcepcioPersistencia

Nom de la classe	ExcepcioPersistencia
Breu descripció de la classe	Classe orientada a gestionar situacions inesperades o errors durant l'execució d'un programa. Forma part d'un mecanisme per capturar i notificar de l'errada. Conté informació rellevant sobre l'error, com un missatge descriptiu. Aquesta classe va orientada a excepcions de persistència i crea una finestra emergent "pop up" mostrant el missatge corresponent.
Descripció dels atributs	Aquesta classe no té atributs.
Descripció dels mètodes	Consta només de la creadora.

3.2.20 ExcepcioPartida

Nom de la classe	ExcepcioPartida
-------------------------	-----------------

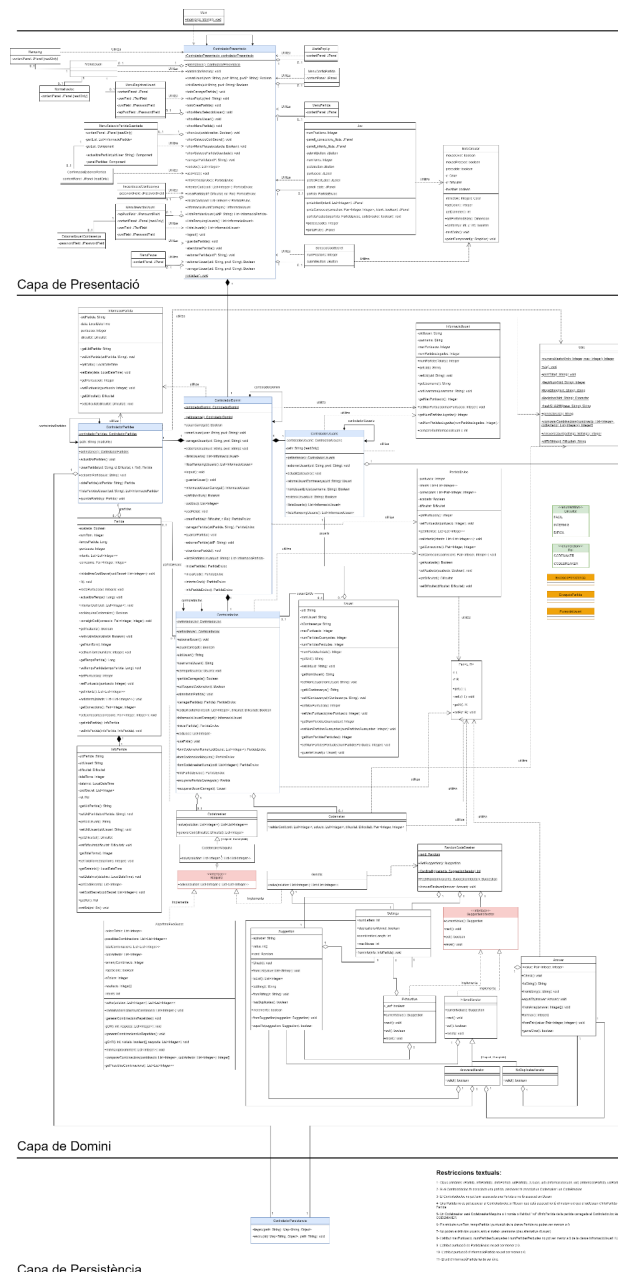
Breu descripció de la classe	Classe orientada a gestionar situacions inesperades o errors durant l'execució d'un programa. Forma part d'un mecanisme per capturar i notificar de l'errada. Conté informació rellevant sobre l'error, com un missatge descriptiu. Aquesta classe va orientada a excepcions de partides i crea una finestra emergent "pop up" mostrant el missatge corresponent.
Descripció dels atributs	Aquesta classe no té atributs.
Descripció dels mètodes	Consta només de la creadora.

3.2.21 ExcepcioUsuari

Nom de la classe	ExcepcioUsuari
Breu descripció de la classe	Classe orientada a gestionar situacions inesperades o errors durant l'execució d'un programa. Forma part d'un mecanisme per capturar i notificar de l'errada. Conté informació rellevant sobre l'error, com un missatge descriptiu. Aquesta classe va orientada a excepcions d'usuari i crea una finestra emergent "pop up" mostrant el missatge corresponent.
Descripció dels atributs	Aquesta classe no té atributs.
Descripció dels mètodes	Consta només de la creadora.

4. Diagrama del model conceptual de dades complet

Podeu trobar aquest diagrama al directori *DOCS* de l'entrega. En color cian es marquen els controladors, en color verd els *enums*, en color rosat les interfícies i en color taronja les excepcions que s'usen en aquesta capa. S'han usat separadors per a diferenciar les diverses capes. La comunicació entre capes es du a terme mitjançant els controladors, de manera que la capa de presentació es comunica amb la capa de domini a través del `ControladorPresentacio` que crida al `ControladorDomini` i la comunicació entre la capa de domini i la capa de persistència es du a terme a través del `ControladorPartides` i el `ControladorUsuaris`, creant una instància de `ControladorPersistencia` cada un (ja que no té estat) en ser inicialitzats.



5. Estructures de dades i algorismes utilitzats

5.1 Estructures de dades

5.1.1 List<InformacioUsuari>

Hem decidit crear aquesta estructura de dades per, en un futur, passar informació d'un usuari de la capa de domini a la capa de presentació sense comprometre cap camp sensible com pot ser el *hash* de la contrasenya (present a la classe *Usuari*). D'aquesta manera podem enviar només la informació necessària per, per exemple, imprimir els usuaris registrats per pantalla passant els valors de l'identificador, nom d'usuari i puntuació.

5.1.2 List<InformacioPartida>

De forma molt similar a l'anterior estructura de dades descrita, aquesta també s'usa per a la mateixa raó: enviar de la capa de domini a la capa de presentació només la informació necessària per a operar amb la partida, és a dir, camps com l'identificador de la partida, la data de creació o la dificultat, entre d'altres (i evitar enviar un objecte *Partida*).

5.1.3 Pair<L,R>

La classe *Pair* s'ha creat per a casos en els quals era necessari el retorn de dos valors, un exemple que es pot trobar en el codi són les correccions d'un codi, que estan formades pel nombre de "fitxes blanques" (indiquen la presència de colors que es troben en el codi però en una posició diferent) i les "fitxes negres" (indiquen la presència de colors correctes en la posició correcta).

5.1.4 PartidaEnJoc

Per a mantenir la capa de presentació informada de l'estat de la partida fem ús d'aquesta classe per emmagatzemar dades com els intents del *codemaker*, les correccions del *codebreaker* (per imprimir-los per pantalla) o la puntuació, així evitem fer ús de classes de domini en altres capes i també el pas de dades innecessàries.

5.1.5 Mapa d'usuaris (ControladorUsuaris)

Per a emmagatzemar els usuaris registrats en l'aplicació hem decidit fer ús d'un `HashMap<K, V>` en el `ControladorUsuaris`, on hem posat l'identificador com a clau (K) i la instància d'`Usuari` com a valor, de manera que podem accedir eficientment quan necessitem carregar, modificar o realitzar alguna acció sobre aquest.

5.1.6 Mapa de partides (ControladorPartides)

Similarment a l'anterior punt, per a emmagatzemar les partides del sistema en memòria fem ús d'un `HashMap<K, V>` en el `ControladorPartides`, usant també com a clau l'identificador (aquest cop el de la partida) i com a valor la instància de l'objecte `Partida`. D'aquesta manera podem accedir també eficientment en cas de necessitar realitzar qualsevol acció.

5.1.7 Answer

Aquesta estructura guarda el valor `Pair<Integer, Integer>` que guarda la relació entre dues combinacions (vegeu `Genetic` per a la justificació) i els paràmetres del joc. A part d'això, conté diversos mètodes auxiliars per transformar valors de tipus de variables a l'`Answer` i viceversa.

5.1.8 Settings

Guarda els paràmetres del joc: la mida de les combinacions, el nombre de colors, el nombre de torns i si estan permets els valors duplicats. Aquests valors s'usen en les generacions de les combinacions.

5.8.9 Suggestion

Aquesta estructura guarda el valor `int[]` que guarda una combinació possible (vegeu `Genetic` per a la justificació) i els paràmetres del joc. A part d'això, conté diversos mètodes auxiliars per transformar valors de tipus de variables a `Suggestion` i al revers i també `increment()` per a incrementar el valor a 1 pas.

5.2 Algorismes

5.2.1 Five-Guess

Cost algorítmic del codi:

El cost algorítmic de l'algorisme és causat per la funció principal `getCombinacio`, aquesta es pot descompondre en el cost de les funcions que invoca i en el cost de les operacions realitzades en ella, per tant, el cost algorítmic és el següent:

La funció **`getCombinació`** invoca les funcions següents:

- **`generarCombinacionsRepetides` o `generarCombinacionsNoRepetides`**: El cost de les dues funcions és diferent. El cost de la generació amb repeticions és $O(nColors^{tamanyCombinació})$. En aquestes s'utilitza un backtracking per a generar totes les possibles combinacions, en el cas de no repeticions tenim un array de booleans per indicar si s'ha utilitzat o no (no varia el cost de la funció). La variable `tamanyCombinacio` en el nostre codi és una constant, però `nColors` pot anar de 4 a 8 depenent de la dificultat, per tant el cost final seria $O(nColors^4)$. En el cas que no hi hagin repeticions fem servir un backtracking amb poda que determinaria un cost algorítmic de $O(N(N-1))$ que simplificadament seria $O(N!)$, on `N` es el `tamanyCombinacio` que és constant, per tant, $O(4!)$, que resulta en $O(1)$.
- **`combinacioInicial`**: El cost d'aquesta funció és de $O(N)$. Si hi ha repeticions el bucle recorre la llista `novaCombinacio` i hi agrega dos elements diferents a cada iteració, la qual cosa fa que es recorri $n/2$ vegades. En canvi, si no hi ha repeticions el bucle agrega un element per iteració que té un cost de $O(N)$. La variable `N` representa `tamanyCombinacio`, que en el nostre codi és una constant, per tant: $O(N^2)$ passa a ser $O(4^2)$, que resulta en $O(1)$.
- **`compararCombinacions`**: La primera part del codi on tenim un bucle i dins d'aquest funcionalitats varies, el cost és de $O(N)$. Ja que n'es recorrerà tota la llista de mida "`N`" fent les comparacions pertinents. En la segona part del codi la complexitat és de $O(N^2)$ perquè hi ha un doble bucle que recorre les dues llistes, i en cada iteració es compara l'element *i*-èsim de la combinació amb tots els elements de `codiAnterior` i es marca l'índex com a utilitzat si es troba una coincidència. En el pitjor dels casos, s'han de recórrer `N` elements al bucle exterior i `N` elements al bucle interior, cosa que resulta en N^2 comparacions. La variable `N` representa `tamanyCombinacio`, que en el nostre

codi és una constant, per tant: $O(N^2)$ passa a ser $O(4^2)$, que resulta en $O(1)$.

- **minmax:** El cost algorítmic de la funció depèn de la mida de la llista possiblesCombinacions que en aquest cas li donarem el valor de N . A la primera part de la funció, es realitza un doble bucle que recorre totes les combinacions possibles de la llista, la qual cosa té un cost d' $O(N^2)$. Dins aquest bucle, s'efectua una operació de cerca en una taula hash, que té un cost mitjà d' $O(1)$, i en el pitjor dels casos $O(N)$, també tenim la funció per comparar dues combinacions que retorna la variable feedback, explicada anteriorment ($O(\text{tamanyCombinacio}^2)$). A la segona part de la funció, es realitza un altre bucle que recorre totes les possibles combinacions i, per a cada combinació, es busca a la taula hash per recuperar una llista de combinacions que tenen el mateix resultat de comparació amb la combinació actual. El cost d'aquesta operació en el pitjor dels casos és $O(N^2)$, en el cas que totes les combinacions tenen el mateix resultat de comparació, tot i que hem de tenir en compte per segona vegada el cost de la funció per comparar dues combinacions. El cost total de la funció seria $O(N^2 * \text{tamanyCombinacio}^2)$ en el pitjor dels casos, però en el nostre codi la variable tamanyCombinacio és una constant que sempre té el mateix valor, per tant, podem afirmar que el cost de la funció és $O(N^2)$.

A més de les operacions realitzades en les funcions invocades, la funció getCombinació efectua les operacions següents:

- Afegeix una combinació a totalCombinacions: el cost d'afegir una combinació a un ArrayList és constant, $O(1)$.
- Elimina elements de possiblesCombinacions amb removelf: el cost d'aquesta operació és $O(N)$, on N és la mida de possiblesCombinacions.
- Actualitza les variables codiAnterior i resolució: el cost d'aquestes operacions és constant, $O(1)$.

El cost algorítmic de la funció `getCombinació` en el pitjor dels casos és repeteixen colors i el nombre de `nColors` és 8 (que representa la màxima dificultat del joc). Per tant tenim en compte la següent taula:

Funció	Cost
<code>generarCombinacionsRpeteides</code>	$O(8^4)$
<code>combianciInicial</code>	$O(1)$
<code>compararCombinacions</code>	$O(1)$
<code>minmax</code>	$O(N^2)$
<code>generarCombinacionsNoRepetides</code> (no entra en el pitjor dels casos)	$O(1)$
Altres operacions realitzades	$O(1)$

L'operació més costosa a la funció principal és el bucle `while`, on s'efectuen operacions amb un cost $O(1)$ i un de cost $O(N^2)$. A cada iteració del bucle, es realitza una eliminació d'elements en `possiblesCombinacions`, que pot tenir un cost màxim d' $O(N)$ si totes les combinacions possibles són a la llista, però en general serà molt menys. La funció `minmax` es crida una vegada per iteració, per tant, en total tindrà un cost d' $O(N^2 * k)$, on k és el nombre d'iteracions del bucle `while` i N la mida de la llista `possiblesCombinacions`.

Per tant, podem afirmar que el cost algorítmic de la funció `getCombinació` serà de $O(nColors^4 + N^2 * k)$, on N és la mida de la llista `possiblesCombinacions` i k és el nombre d'iteracions del bucle `while`. En general, el valor de N , k i `nColors` dependrà de la dificultat utilitzada per endevinar el codi secret, per la qual cosa pot variar.

$O(N^2 * k)$

Justificació de la funcionalitat de l'algorisme:

L'algorisme five-guess es basa a donar una puntuació a la combinació amb més probabilitats d'eliminar-ne més de la llista de possibles combinacions. Aquesta seria la funcionalitat principal i és producte de la funció "minmax" que dona un valor a les combinacions depenent del punt esmentat anteriorment.

- Funció proposada:

La raó per la qual la nostra funció "minmax" és més eficient que la funció original de l'algorisme five-guess és perquè utilitzem una estratègia per reduir la quantitat de possibles combinacions que han de ser avaluades. En lloc de provar totes les combinacions possibles de colors i després eliminar les que no s'ajusten als resultats proporcionats per l'usuari, la funció "minmax" fa servir una estratègia, que consisteix a buscar la combinació que minimitza el nombre màxim de possibilitats restants, però només entre les possibles combinacions. Per aconseguir això, la funció crea una taula de puntuacions que "mappeja" els resultats de la comparació entre les combinacions possibles i la combinació a intentar (cal recalcar que la utilització d'estructures de dades que fan servir hash són més eficients a l'hora de buscar-ne un element). Després, utilitza aquesta taula per dividir el conjunt de possibles combinacions en particions de combinacions que produeixen el mateix resultat de comparació amb la combinació a intentar. Finalment, selecciona la partició amb el menor nombre màxim de possibilitats restants i tria una combinació d'aquesta partició com a proper intent.

5.2.2 Genetic

Cost algorítmic del codi:

El cost algorítmic de l'algorisme es compon de pocs elements i depèn en general dels costos dels **iteradors**. Els iteradors recorren totes les possibles combinacions amb el cost $O(n)$. Per a cada combinació es decideix si és vàlida per a la situació o no. Cap de les combinacions es guarda enlloc, sinó que es generen usant el mètode **increment** de la classe **Suggestion**. Hi ha 2 tipus d'iteradors: 2 iteradors inicials (**Exhaustive** i **NoDuplicatesIterator**) que s'iteren per totes les combinacions possibles o totes les que no tenen valors duplicats respectivament, i una iteradora filtrada (**FilteredIterator** que s'expandeix amb l'**AnsweredIterator**). Després, l'

iterador **AnsweredIterator** valida les combinacions a base de la comparació de la combinació corrent amb totes les altres via el mètode **compararCombinacions()** d'**Utils**, i si el resultat no coincideix amb el rebut per paràmetre la combinació es descarta. A més a més, aquest iterador guarda un iterador previ. D'aquesta manera, cada combinació passa per una llista de tots els filtres aplicats que es creix amb cada torn nou. Aquesta funcionalitat també permet recórrer les combinacions ja descartades, garantint el cost de $O(kn)$, on k és el nombre de torns i n és el nombre de totes les combinacions possibles.

Justificació de la funcionalitat de l'algorisme:

Quan parlem de les combinacions possibles s'ha de parlar primer sobre la relació entre combinacions. Aquesta relació és la possible resposta obtinguda si una combinació seria el codi i l'altre seria el torn. Està demostrat per Donald Knuth que per a qualsevol combinació hi ha una llista no nul·la de combinacions que es relacionen amb aquesta de la mateixa manera que la combinació amb el codi secret, i a més a més, el codi secret està dins d'aquesta llista. Aquesta llista és la llista de combinacions possibles. Si escollim una d'aquestes en aleatori i en cada torn reduïm la llista de combinacions possibles de la mateixa manera, o endevinarem a combinació amb la sort, o ens quedarem amb només una opció que serà la guanyadora.

El procés de la formació de la llista de les possibles combinacions està implementada via **iterators** que recorren totes les possibles combinacions i descarten totes que no són vàlides segons el filtre determinat.