HIDATO

PROJECTES DE PROGRAMACIÓ - FIB

Albert Figuera Pérez, Jacobo Moral Buendía, Jia Xiang Cheng

ÍNDEX

Diagrama de casos d'ús

Amb el diagrama de casos d'ús conseguim una simple representació de la interacció d'un usuari amb el sistema que mostra la relació entre l'usuari i els diferents casos d'ús en què l'usuari està involucrat.

Descripció casos d'ús

Cas d'us Crear Hidato

Primary Actor: Usuari.
Precondition: -.
Trigger: Un usuari vol crear un Hidato.
Main Success Scenario:

1. L'usuari, en el menú principal, selecciona l'opció '*crear Hidato'*.
2. El sistema mostra per pantalla diverses opcions: 'Autogenerar Hidato', 'Hidato personalitzat' i 'importat Hidato'.
3. L'usuari selecciona una de les opcions i el sistema crea una instància Hidato per tal de que l'usuari pugui jugar.
4. El sistema mostra per pantalla la matriu hidato amb la qual començarà el joc.

Cas d'us Verificar Hidato

Primary Actor: Usuari.
Precondition: S'ha creat una instància hidato.
Trigger: al crear-se la instancia hidato.
Main Success Scenario:

1. Si estem en el procés de crear un hidato nou, salta el pas 4.
2. L'usuari selecciona l'opció de verificar un hidato en el menú principal.
3. L'usuari pot enviar al sistema l'hidato que vol verificar de 3 maneres diferents, introduir les característiques o importar des d'un fitxer o importar des del clipboard.
4. El sistema rep l'hidato que s'ha de verificar.
5. El sistema comprava que el sintaxi sigui correcte i també que sigui resoluble.
6. Si l'hidato no compleix la sintaxi o que no es soluble llavors salta una excepció i mostrar un missatge d'error que indica al usuari. Sinó mostra un missatge indicant que l'hidato és correcte.

Cas d'us Consultar Rànking

Primary Actor: Usuari.
Precondition: -.
Trigger: quan un usuari vol consultar el ranking del joc.
Main Success Scenario:

1. L'usuari clica el botó de ranking en el meu principal.
2. El sistema mostra el ranking de totes les partides que han sigut resolts i acabats.

Cas d'us Jugar Hidato

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari tria en el menú principal l'opció de començar a jugar una partida.
Main Success Scenario:

1. L'usuari selecciona l'opció jugar una partida en el menú principal.
2. El sistema mostra una pantalla amb dos opcions que son: 'nova partida' o 'reprendre partida'.
3. Si l'usuari selecciona una nova partida, llavors ha de triar com vol generar un hidato nou per jugar o seleccionar un hidato que estigui en el bases de dades.
4. Si l'usuari selecciona reprendre una partida llavors ha de escollir de les partides guardes quin vol continuar.
5. Un cop acabat de decidir començarà la partida o es reprendrà la partida.

Subcas d'us Auto Generar Hidato

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari dona d'alta un hidato autogenerat.
Main Success Scenario:

1. L'usuari selecciona l'opció de crear un hidato de manera automàtica.
2. La maquina rep la petició, i pregunta al usuari de quin dificultat el vol.
3. L'usuari selecciona la dificultat.
4. El sistema mostra per la pantalla el hidato generat.

Subcas d'us Introduir Condicions

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona crear hidato personalitzat.
Main Success Scenario:

1. L'usuari selecciona l'opció introduir les condicions per crear un hidato nou.
2. El sistema pregunta al usuari com vol que sigui el hidato nou.
3. L'usuari introdueix les característiques.
4. El sistema rep les condicions introduïdes per l'usuari.
5. El sistema verifica si l'hidato és possible de generar o no.
6. Si no es pot generar, salta una excepció i juntament amb un missatge d'error, llavors tornem al pas 3.
7. El sistema genera un hidato a l'atzar amb les condicions del usuari.
8. El sistema mostra en la pantalla el hidato mostrat.

Subcas d'us Importar Des D'un Fitxer

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona importar hidato des d'un fitxer.
Main Success Scenario:

1. L'usuari selecciona l'opció d'importar un hidato en format de text per crear un hidato nou.
2. El sistema es carrega la pantalla d'adjuntar un fitxer local.
3. L'usuari selecciona un fitxer que conté l'hidato que vol crear i puja el fitxer
4. El sistema rep el fitxer.
5. El sistema llegeix i analitzar el fitxer.
6. El sistema verifica si l'hidato és possible de generar o no.
7. Si no es pot generar, salta una excepció i juntament amb un missatge d'error, llavors tornem al pas 3.
8. El sistema genera el hidato corresponent,
9. El sistema mostra per la pantalla el hidato importat.

Subcas d'us Importar Des De Clipboard

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona importar hidato des d'un clipboard.
Main Success Scenario:

1. L'usuari selecciona l'opció d' importar l'hidato des del clipboard.
2. L'usuari enganxa l'hidato copiat i enviar al sistema.
3. El sistema rep l'hidato i l'analitza.
4. El sistema verifica si l'hidato és possible de generar o no.
5. Si no es pot generar, salta una excepció i juntament amb un missatge d'error, llavors tornem al pas 2.
6. El sistema genera un hidato amb el format corresponent.
7. El sistema mostra una pantalla l'hidato generat al usuari.

Subcas d'us Escollir Dificultat

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona autogenerar hidato.
Main Success Scenario:

1. L'usuari selecciona l'opció de auto generar per crear un hidato.
2. El sistema mostra un pantalla amb els dificultats que hi han.
3. L'usuari selecciona una dificultat.
4. El sistema genera un hidato amb la dificultat triada.
5. El sistema mostra un hidato generat per la pantalla.

Subcas d'us Resoldre Hidato

Primary Actor: Usuari.
Precondition: Instància hidato creada i te solució.
Trigger: quan l'usuari selecciona solucionar o resoldre hidato.
Main Success Scenario:

1. L'usuari en el menú principal selecciona l'opció de crear un hidato.
2. El sistema mostra una pantalla on l'usuari pot adjuntar el seu fitxer o enganxa l´hidato des del clipboard.
3. L'usuari adjunta l'hidato que té o enganxa l'hidato des del clipboard.
4. El sistema rep el fitxer o l'hidato de forma directe.
5. El sistema el analitzar.
6. El sistema verifica si l'hidato és possible de generar o no.
7. Si no es pot generar, salta una excepció i juntament amb un missatge d'error, llavors tornem al pas 3.
8. El sistema resolt l'hidato i presenta la solució en la pantalla.

Subcas d'us Demanar Pista

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona demanar pista.
Main Success Scenario:

1. L'usuari està jugant en una partida.
2. L'usuari vol demanar ajuda al sistema per obtenir el pas següent.
3. El sistema comprova la situació actual del hidato.
4. Si troba que algun dels números posats estan malament, indicarà al usuari quins números estan mal posats i l'usuari ha de corregir-los. Un cop corregit i no hi ha cap error, el sistema auto completa el pas següent del hidato.
5. El sistema auto completa el pas següent.

Subcas d'us Guardar Partida

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona guardar partida.
Main Success Scenario:

1. L'usuari està jugant una partida.
2. L'usuari clica el botó de guarda la partida.
3. El sistema guarda la partida actual de l'usuari.
4. El sistema mostra una finestra perquè l'usuari pugui posar un nom el que vulgui a la partida.
5. El sistema guarda la partida en un fitxer.
6. El sistema mostra un missatge d'èxit i torna al menú principal.

Subcas d'us Guardar Partida

Primary Actor: Usuari.
Precondition: -.
Trigger: quan l'usuari selecciona guardar partida.
Main Success Scenario:

1. L'usuari està jugant una partida.
2. L'usuari clica el botó de guarda la partida.
3. El sistema guarda la partida actual de l'usuari.
4. El sistema mostra una finestra perquè l'usuari pugui posar un nom el que vulgui a la partida.
5. El sistema guarda la partida en un fitxer.
6. El sistema mostra un missatge d'èxit i torna al menú principal.

Subcas d'us Reprendre Partida

Primary Actor: Usuari.
Precondition: l'Usuari té una partida per reprendre.
Trigger: l'usuari selecciona reprendre la partida.
Main Success Scenario:

1. L'usuari clica el botó de reprendre una partida.
2. El sistema mostra una pantalla on hi ha una llista on hi ha totes les partides guardades.
3. L'usuari selecciona la partida que vol continuar jugant.
4. El sistema carrega la partida seleccionada.

Subcas d'us Guardar Resultat

Primary Actor: Usuari.
Precondition: Hidato finalitzat.
Trigger: l'usuari resol l'hidato correctament.
Main Success Scenario:

1. L'usuari resol completament un hidato.
2. El sistema mostra un missatge de victòria al usuari.
3. El sistema demana al usuari que posi un nickname per la puntuació de la partida a través d'una finestra.
4. L'usuari introdueix el seu nickaname i accepte.
5. El sistema guarda el resultat amb el nom escollit en el bases de dades.

Subcas d'us Iniciar Temps

Primary Actor: Usuari.
Precondition: Hidato creat i amb solució.
Trigger: l'usuari comença a resoldre l'hidato.
Main Success Scenario:

1. L'usuari vol jugar una patida.
2. L'usuari selecciona la manera de crear un hidato.
3. El sistema el crear i mostra al usuari.
4. L'usuari dona un clic al botó començar.
5. El sistema posa en marxa el comptador de temps.

Subcas d'us Filtrar Rànking

Primary Actor: Usuari.
Precondition: -.
Trigger: l'usuari, en el ranking, filtra.
Main Success Scenario:

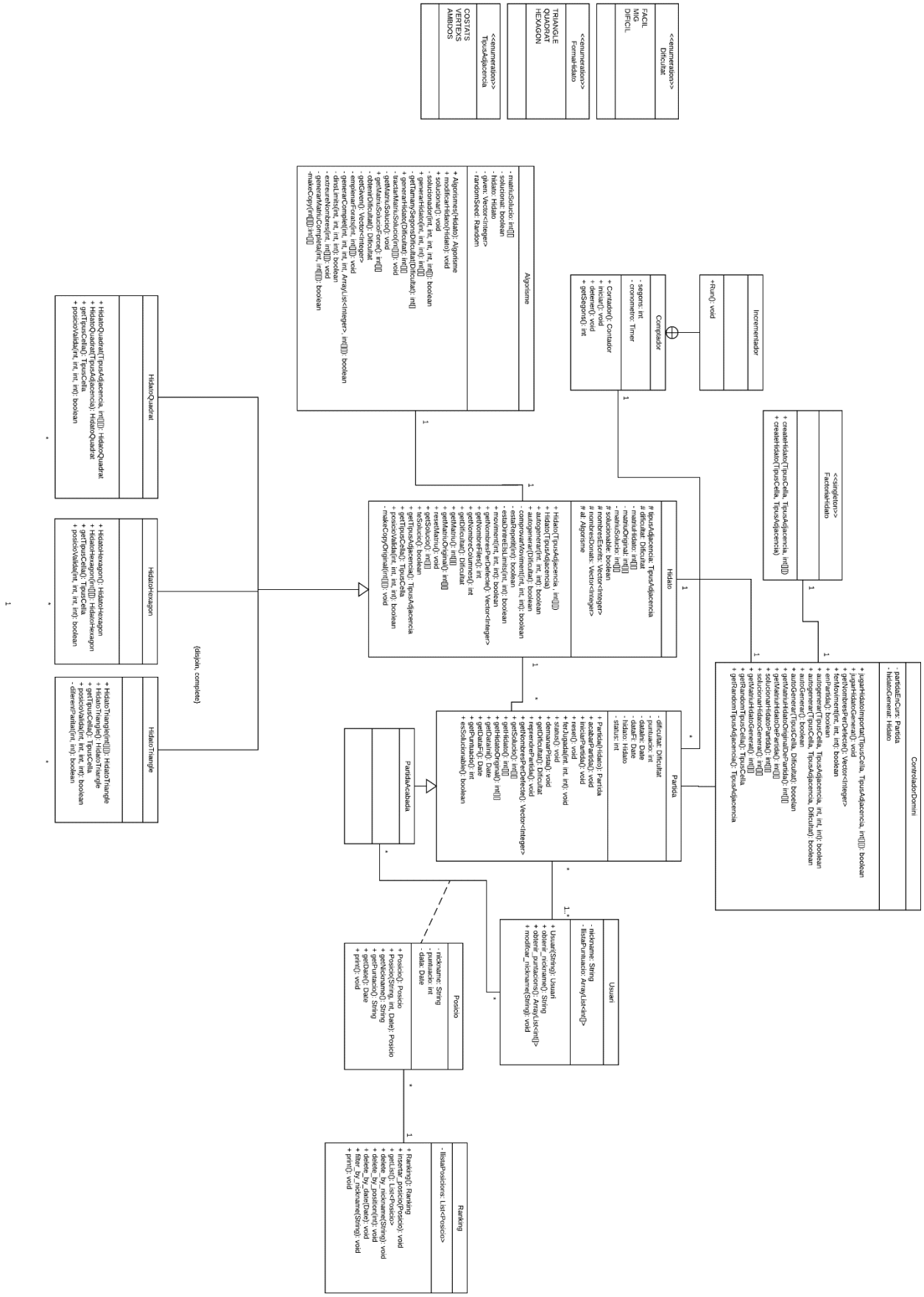
1. L'usuari consulta el ranking.
2. L'usuari indica si vol filtrar el ranking segons el nom, el tipus de hidato, dificultat, el temps ...
3. El sistema filtrar el ranking segons el criteri de l'usuari.
4. El sistema mostra el ranking filtrat.

Subcas d'us Seleccionar Hidato

Primary Actor: Usuari.
Precondition: -.
Trigger: l'usuari selecciona un hidato per jugar-lo.
Main Success Scenario:

1. L'usuari selecciona l'opció de jugar una partida a partir dels hidatos guardats en el bases de dades.
2. El sistema mostra una llista de hidatos que pots escollir.
3. L'usuari selecciona un en concret.
4. El sistema comença un nova partida amb el hidato seleccionat.

Diagrama estàtic complet del model conceptual de dades



A més de les classes del diagrama de classes, hem fet algunes més:

HidatoIO: Ens serveix per dur a terme funcions que hem fet servir molt habitualment, com llegir un hidato des de consola o escriure un hidato a consola.

InteraccioTerminal: Ens servia inicialment, abans de tenir els drivers, per provar la correctesa del nostre programa. Ens feia com una mena de "main".

Descripció del diagrama del model conceptual

A partir del diagrama anterior podem observar de quines classes formen la nostra capa de domini.

Primerament, tenim un controlador de domini que és l'encarregat de respondre els esdeveniments del la capa de la presentació quan volen demanar informacions sobre els models. També serveixen per comunicar a la capa de presentació de que hi hagut actualitzacions en els models per tal de mantenir la persistència de la informació de la vista corresponent. En el nostre cas, el controlador de domini només té relacions amb la classe Hidato i la Partida, els altres es relacionen a partir d'aquests dos.

Pel que fa les factories, nosaltres tenim una factoria de hidatos que serveix per donar alta hidatos al sistema.

Pel que fa les enumeracions, el fem servir per representar els diferents tipus d'atributs que només tindran uns valors determinats.

A més a més, tenim un private nested class en el comptador perquè tenim una classe privada dins de la classer Comptador.

Finalment, hem utilitzat el polimorfisme en la classe hidato pels diferents tipologies que podrem definir, perquè d'aquesta manera podem aprofitar les característiques de l'herència i també utilitzar les operacions abstractes. En aquest cas és de tipus disjoin perquè un hidato només pot tenir una forma durant una partida i completa perquè només els hidatos en el nostre projecte obligatòriament han de ser aquest tipus.

Relació de les classes implementades per cada membre del grup

No tenim en compte els stubs, que s'han creat pel mateix membre que la classe driver que el necessita.

Jacobo:

- Algorisme
- ControladorDomini
- DriverControladorDomini
- HidatoIO
- TestAlgorismes (JUnit)

Albert:

- Hidato
- HidatoFactory
- HidatoQuadrat, HidatoTriangle
- DriverHidato
- DriverHidatoTriangle, DriverHidatoQuadat
- InteraccioTerminal

Jia:

- Hidato
- Partida
- DriverPartida
- DriverAlgorismes
- HidatoHexagon
- Main
- DriverHidatoHexagon

Breu descripció de les estructures de dades i algorismes utilitzats per a implementar les funcionalitats principals

Pel que fa a l'estructura dades que utilitzem per als hidatos només fem servir els ararys, és a dir, int[][]. Perquè és una estructura simple, eficient (ja que un cop constituïda a memòria no es pot modificar, donant-li robustesa) i fàcil de gestionar amb poc recurs. Aquesta estructura és molt clara a l'hora de fer accessos, no és tan bona quan es vol ordenar l'estructura, ja que no consta de funcions com ordenar, però no ho necessitem.

Pel que fa a l'algorisme de resoldre fem servir un backtracking on, partint de la posició on es troba l'1, intentem arribar, passant per les cel·les que li són adjacents, fins a l'últim número. És a dir, fem servir depth first search (dfs), perquè només ens quedem amb la primera de les solucions que trobem

Pel que fa a l'algorisme generar un hidato, fem servir un algorisme molt similar que per resoldre'l. La gran diferencia està en el fet que en aquest cas, l'ordre de les posicions adjacents que explorem és aleatori, per aconseguir hidatos aleatoris. Ara en comptes d'acabar quan ha passat per totes les posicions de la matriu, acaba quan arriba a totes menys un nombre, que serà el nombre de forats (*) que volem ficar. Aprofitem que l'hidato creat és aleatori per posar als forats en blanc que el mateix algorisme ha deixat per convertir-los en forats (*).

Només hem utilitzat una llibreria externa, jUnit4, per poder fer els tests unitaris del nostre programa.