PROJECTE DE PROP
KAKURO

PRIMERA ENTREGA

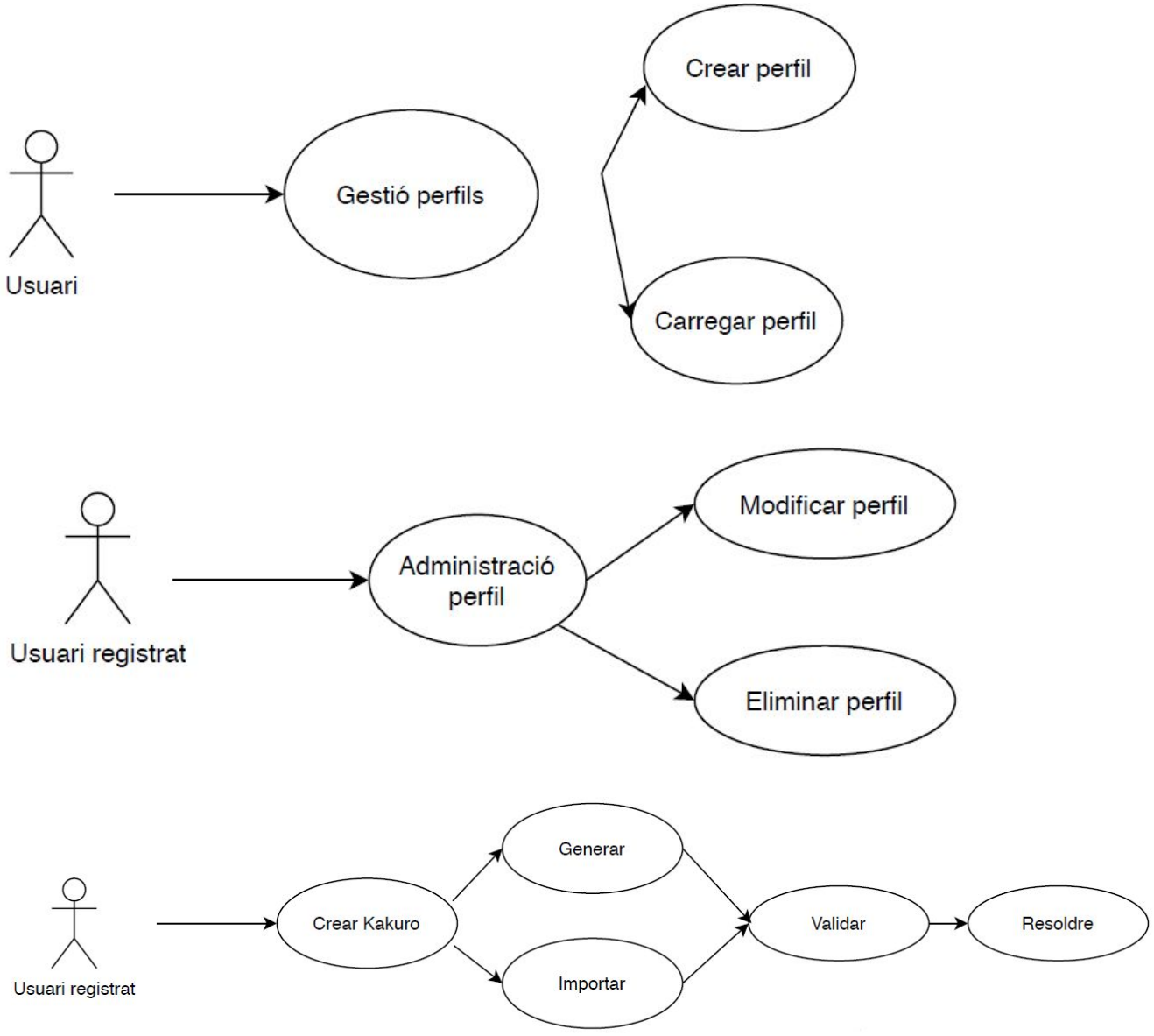
Grup 4.2

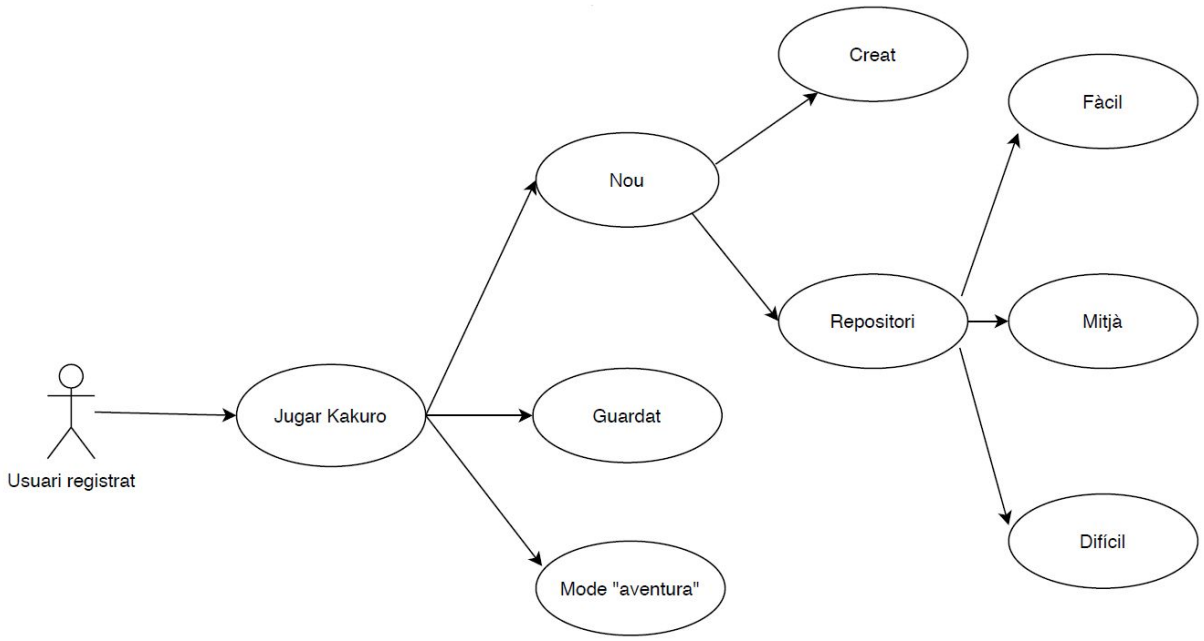
Alvaro Ariño Cabau, alvaro.arino@estudiantat.upc.edu
Albert Bertran Serrano, albert.bertran.serrano@estudiantat.upc.edu
Victor Manuel Delgado Padilla, victor.manuel.delgado@estudiantat.upc.edu
Victor Latorre López, victor.latorre@estudiantat.upc.edu
Versió de lliurament 1.0

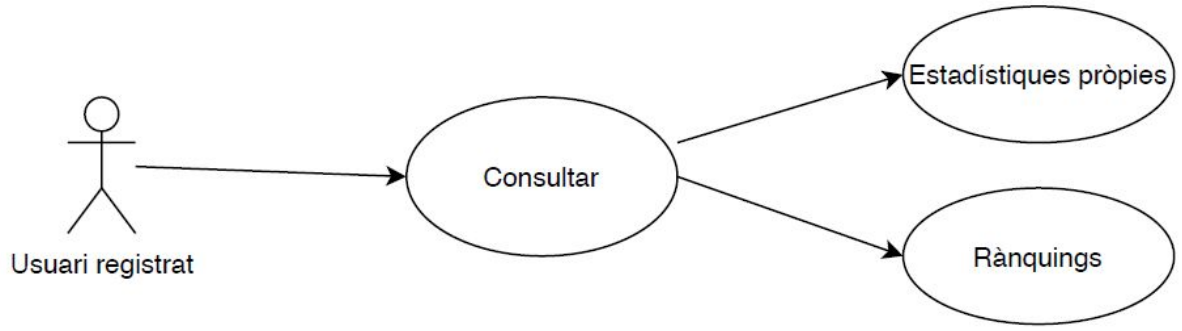
Índex

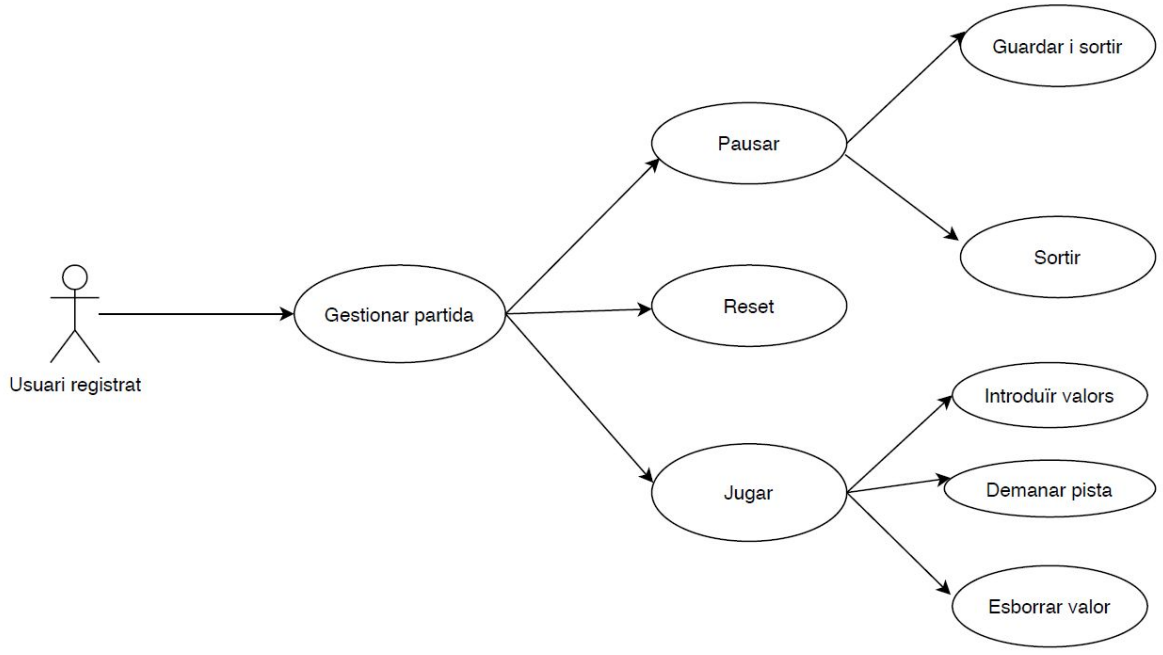
1.Diagrama de casos d'ús i la seva descripció

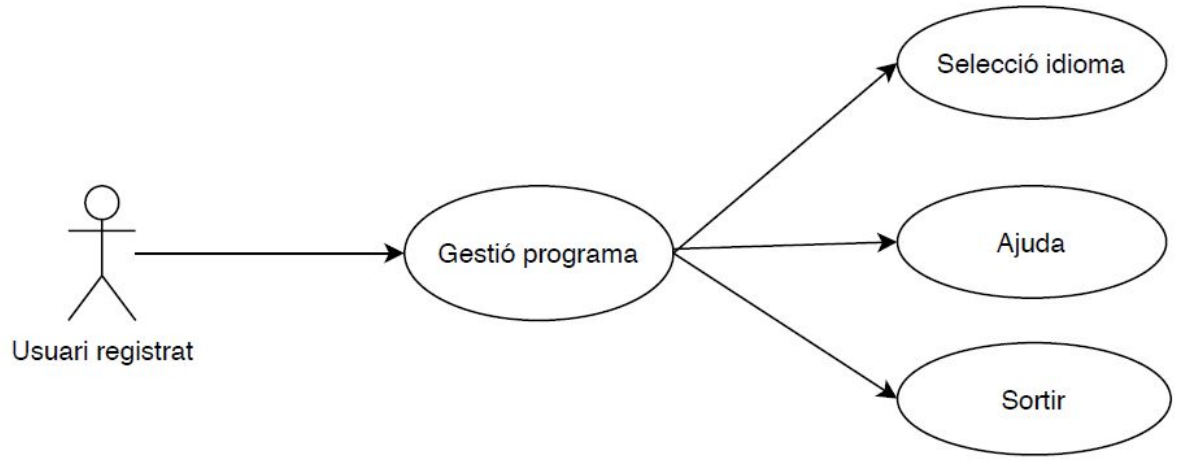
El nostre diagrama de casos d'ús és el següent:




Creat
Fàcil
Nou
Repositori
Mitjà
Usuari registrat
Jugar Kakuro
Guardat
Difícil
Mode "aventura"


Estadístiques pròpies
Consultar
Usuari registrat
Rànquings


Guardar i sortir
Pausar
Sortir
Usuari registrat
Gestionar partida
Reset
Introduïr valors
Jugar
Demanar pista
Esborrar valor



Descripció dels casos d'ús que presentem al diagrama:

Gestió perfils

Nom: Crear perfil
Actor: Usuari
Comportament:
-L'usuari es crea un perfil informant del seu identificador de perfil, nom, contrasenya i icona si escau.
-El sistema valida els valors introduïts i la coherència d'aquests.
Errors possibles i cursos alternatius: Si existeix un perfil amb el mateix identificador, s'informa a l'usuari que faci servir un altre.

Nom: Carregar perfil
Actor: Usuari
Comportament:
-L'usuari inicia la sessió amb el seu identificador de perfil i contrasenya
-El sistema valida els valors introduïts i proporciona accés.
Errors possibles i cursos alternatius:
-Si l'identificador del perfil no figura al sistema, s'informa a l'usuari.
-Si existeix l'identificador del perfil però la contrasenya no coincideix, s'informa a l'usuari.
-Si l'usuari no recorda la contrasenya, el sistema ofereix canviar-la.

Administració perfil

Nom: Modificar perfil
Actor: Usuari Registrat
-Comportament: L'usuari desitja canviar nom, identificador de perfil, contrasenya o icona
Errors possibles i cursos alternatius:
-Si el nou identificador de perfil que introdueix l'usuari ja existeix, s'informa de que introdueixi un de nou.

Nom: Eliminar perfil

Actor: Usuari Registrat
Comportament:
-L'usuari desitja eliminar un Perfil el qual ha accedit prèviament.
Errors possibles i cursos alternatius: Es pregunta a l'Usuari si està segur d'eliminar el perfil i si és així s'elimina.

Crear Kakuro

Nom: Generar
Actor: Usuari Registrat
Comportament:
-L'usuari desitja generar un Kakuro amb les dades que ell mateix introdueix: dimensions, nombre de caselles blanques i negres i nombre de xifres col·locades.
Errors possibles i cursos alternatius: Si amb les dades introduïdes el Kakuro no permet una solució, s'informa a l'usuari. En el cas que tingui solució, s'informa de si aquesta és única.

Nom: Importar
Actor: Usuari Registrat
Comportament:
-L'Usuari desitja jugar un Kakuro importat d'un fitxer
Errors possibles i cursos alternatius: És possible que el Kakuro no segueixi el format establert, que no tingui solució o que en tingui una o moltes, en tot cas s'informarà a l'Usuari en la fase de validació.

Nom: Validar
Actor: Usuari Registrat
Comportament:
-L'Usuari pren el botó validar per tal de comprovar si un Kakuro que ha penjat en un fitxer o que vol crear amb els seus paràmetres, és correcte o no. Si és correcte el sistema informa de si té solució única.
Errors possibles i cursos alternatius: És possible que el fitxer no segueixi el format establert o que els paràmetres no facin possible la creació d'un Kakuro. En aquests casos el sistema informa dels errors.

Nom: Resoldre
Actor: Usuari Registrat
Comportament:
-L'Usuari desitja resoldre veure la resolució del Kakuro. El sistema la mostra.
Errors possibles i cursos alternatius: Cap.

Jugar Kakuro

Nom: Nou
Actor: Usuari Registrat
Comportament: L'Usuari desitja jugar a un Kakuro Nou, ja sigui creat per ell o del repositori de Kakuros.
Errors possibles i cursos alternatius: Cap.

Nom: Creat
Actor: Usuari Registrat
Comportament:
-L'Usuari juga a un Kakuro nou que ha generat ell (ha estat validat prèviament).
Errors possibles i cursos alternatius: Cap.

Nom: Repositori
Actor: Usuari Registrat
Comportament:
-L'usuari escull un Kakuro dins un repositori de Kakuros amb una dificultat determinada (fàcil, mitjà o difícil).
Errors possibles i cursos alternatius: És possible que l'Usuari ja tingui el Kakuro resolt, en aquest cas el sistema l'informa.

Nom: Guardat
Actor: Usuari Registrat
Comportament:
-L'Usuari vol jugar a un joc que havia pausat.
Errors possibles i cursos alternatius: Cap.

Nom: Mode "aventura"
Actor: Usuari Registrat
Comportament: L'Usuari desitja jugar l'anomenat mode aventura. Anirà rebent per pantalla certs Kakuros amb diferent dificultat.
Errors possibles i cursos alternatius: Cap

<u>Consultar</u>

Nom: Estadístiques pròpies
Actor: Usuari Registrat
Comportament: L'usuari consulta les seves estadístiques en la resolució dels kakuros.
Errors possibles i cursos alternatius: Si l'usuari no ha resolt cap kakuro, se l'informa.

Nom: Rànquings
Actor: Usuari Registrat
Comportament: L'usuari consulta les estadístiques de resolució de kakuros globals, de tots els usuaris
Errors possibles i cursos alternatius: No hi ha cap usuari que hagi resolt almenys un kakuro, s'informa a l'usuari.

Gestionar partida

Nom: Pausar: Guardar i sortir
Actor: Usuari Registrat
Comportament: L'usuari pausa la partida actual, la guarda per continuar-la en un moment posterior i surt de la partida.
Errors possibles i cursos alternatius: Cap

Nom: Pausar i Sortir
Actor: Usuari Registrat
Comportament: L'usuari pausa la partida actual, i abandona la partida sense guardar els valors introduïts.
Errors possibles i cursos alternatius: Cap

Nom: Reset
Actor: Usuari Registrat
Comportament: El sistema esborra tots els valors entrats en les cel·les blanques per tornar a començar la resolució.
Errors possibles i cursos alternatius: Cap

Nom: Jugar: Introduir valors
Actor: Usuari Registrat
Comportament: L'usuari selecciona una cel·la i marca un valor per introduir-lo.
Errors possibles i cursos alternatius: El valor introduït no es troba en el rang 1-9, s'avisa al usuari. La cel·la seleccionada es correspon amb una cel·la negra, s'avisa al usuari.

Nom: Jugar: Demanar pista
Actor: Usuari Registrat
Comportament: L'usuari obté una pista per ajudar-lo en la resolució del kakuro.
Errors possibles i cursos alternatius: Cap

Nom: Jugar: Esborrar valor
Actor: Usuari Registrat
Comportament: L'usuari selecciona una cel·la i esborra el valor introduït prèviament.
Errors possibles i cursos alternatius: La cel·la seleccionada es correspon amb una cel·la negra, s'avisa al usuari.

Gestió programa

Nom: Selecció idioma
Actor: Usuari Registrat
Comportament: L'usuari selecciona un idioma dels oferts pel sistema
Errors possibles i cursos alternatius: Cap

Nom: Ajuda
Actor: Usuari Registrat
Comportament: L'usuari obté ajuda del sistema sobre com utilitzar l'aplicació i com resoldre un kakuro.
Errors possibles i cursos alternatius: Cap

Nom: Sortir
Actor: Usuari Registrat
Comportament: L'usuari surt de l'aplicació.
Errors possibles i cursos alternatius: Cap

2. Diagrama estàtic complet del model conceptual de dades en versió disseny i breu descripció de cada classe.

El diagrama de classes en versió disseny es pot trobar a

Descripció de cada classe:

Classe Kakuro:
Aquesta classe representa un Kakuro amb el seu Tauler (multiplicitat 1) i un número aleatori aleat. Pot formar part de diverses partides, o de diverses aventures. Les funcions que conté aquesta classe són:

1) **Kakuro(int n,int m):** Constructora amb les dimensions n i m.
2) **Kakuro():** Constructora buida.
3) **generar():** El mètode genera un Kakuro.
4) **generar_usuario(int n, int m, int negras, int blancas):** Es genera un Kakuro amb les especificacions de l'Usuari.
5) **getBoard():** Retorna un Tauler.

Classe Tauler:
Aquesta classe representa el tauler d'un Kakuro. Està representat per un conjunt de cel·les dins un Array i una dimensió en n i en m. És per aquesta raó que Tauler i cel·les tenen aquesta multiplicitat, un tauler pot tenir moltes cel·les però una cel·la només pertany a un tauler. A més a més, té relació amb la classe Kakuro perquè un Kakuro té un tauler i un tauler pertany a un Kakuro.
Els mètodes implementats en aquesta classe són els següents:

1) **Tauler(int n, int m):** Constructora de la classe Tauler amb els valors de n i m (amplada i alçada del tauler).
2) **getDimn():** Retorna l'alçada del tauler.
3) **getDimm():** Retorna l'amplada del tauler.
4) **getCella(int i, int j):** Retorna una cel·la en la posició i,j.
5) **posValida(int i, int j):** Aquesta funció comproba (retornant true) si una posició és vàlida per assingar-li una cel·la negra.
6) **pintar_celda(int i, int j, int cantidad):** El mètode assigna a una cel·la el tipus cel·la negra.
7) **pintar_negras(int cantidad):** Assigna la quantitat "cantidad" de negres al tauler.
8) **generar_aleatorios():** Retorna un número aleatori entre 1 i 9.
9) **noPresente(int valor, int i, int j):** retorna true si el valor està present en la mateixa fila i o columna j.
10) **rellenar_blancas():** Omple les caselles blanques de números aleatoris entre 1 i 9.
11) **hacer_sumas():** Realitza les sumes de les caselles blanques per tal d'assignar-les a les caselles negres.
12) **borrar_blancas():** Deixa sense valor les caselles blanques.
13) **solucionar():** Soluciona el Kakuro implícit.
14) **SolBacktracking(Cella[][] board, int fila, int col):** Operació recursiva que soluciona el Kakuro.

15) **valorValid(Cella[][] board, int fila, int col, int valor):** Retorna true si ValorValidfila i valorValidCol retornan true.
16) **ValorValidfila(Cella[][] board, int fila, int col, int valor):** Comprova si valor és vàlid a la fila delimitada per fila, col.
17) **valorValidCol(Cella[][] board, int fila, int col, int valor):** Comprova si valor és vàlid a la fila delimitada per col, fila.
18) **print ():** Imprimeix el Kakuro.

Classe Cella:
Classe abstracta representant una casella del tauler. Diverses cel·les conformen un tauler i n'hi ha de dos tipus: blanques i negres. Funcions que conté la classe Cella:

1) **Cella():** Crea una cella buida.
2) **intro_valor_blanca(int z):** Funció abstracta.
3) **getValor_blanca():** Funció abstracta.
4) **acumular_valor_derecha(int s):** Funció abstracta.
5) **acumular_valor_izquierda(int s):** Funció abstracta.
6) **color():** Funció abstracta.
7) **getValorDret():** Funció abstracta.
8) **getValorEsquerre():** Funció abstracta.

Classe CellaNegra: Cella del Kakuro que té com atributs valorDret i valorEsquerre.
1) **CellaNegra():** Crea una casella negra amb valor dret i valor esquerra amb valor null.
2) **CellaNegra(int dreta, int esquerra):** inicialita la cel·la negra amb valor dreta i esquerra.
3) **setValorFila(int val):** assigna el valor val a valorDret.
4) **setValorColumna(int val):** assigna el valor val a ValorEsquerre.
5) **acumular_valor_derecha(int s):** Si s'invoca aquesta funció per primer cop (la casella tenia valor null) llavors inicialitzem el valor a 0. Després anem sumant el valor que tenia la nova casella blanca s que entra.
6) **acumular_valor_izquierda(int s):** Si s'invoca aquesta funció per primer cop (la casella tenia valor null) llavors inicialitzem el valor a 0. Després anem sumant el valor que tenia la nova casella blanca s que entra.
7) **getValorDret():** Retorna el valor dret de la casella negra en cas de tenir, sinó retorna null.
8) **getValorEsquerre():** Retorna el valor esquerre de la casella negra en cas de tenir, sinó retorna null.
9) **color():** Retorna blanc(dos valors d'una enumeració).

Classe CellaBlanca: És una cel·la blanca amb un valor que pot ser -1 o entre 1 i 9.
1) **CellaBlanca():** Crea una casella blanca amb valor null.
2) **getValor_blanca():** Retorna el valor entre 1 i 9 de la casella blanca en cas de tenir, sinó retorna null.
3) **intro_valor_blanca(int z):** Assignem el valor a una casella blanca (números entre 1 i 9).
4) **color():** Retorna blanc(dos valors d'una enumeració).

Classe CjtPartides:

Aquesta classe representa un Conjunt de les partides al Kakuro. Les funcions que conté aquesta classe són:

1) **CjtPartides():** Constructora buida.
2) **PartidesTotals():** Retorna el nombre de partides totals que hi ha al Conjunt.

Classe Partida:

Aquesta classe representa una partida al Kakuro, amb el seu temps de joc i estat en el que es troba. Les funcions que conté aquesta classe són:

1. **Partida():** Constructora buida.
2. **getEstat():** retorna l'estat en el que es troba:
    i. **0:** Jugant, estat per defecte.
    ii. **1:** Pausat, estat en el que no corre el temps i en el que es posa la partida si polsem algun botó per tal de mirar les opcions.
3. **PausaPart():** Canvia l'estat a 1.
4. **ContinuaPart():** Canvia l'estat a 0.
5. **getTemps():** Retorna el temps de joc.

Classe Usuari

Representa a l'usuari del programa, permet emmagatzemar tots els seu perfils.

L'ID es un UUID que es generar en el moment de la crida del constructor.

1. **Usuari():** Constructora buida
2. **Usuari(String nom, String username):** crea un usuari amb un nom i un username.
3. **getID()**: permet obtenir l'ID
4. **getNom():** permet obtenir el nom de l'usuari.
5. **getUsername():** permet obtenir el username de l'usuari
6. **getNumPerfils()**: retorna el nombre de perfils assignats a l'usuari
7. **setNom():** permet introduir el nom de l'usuari
8. **setPass():** permet afegir una contrasenya a l'usuari
9. **addProfile()**: crea un perfil per l'usuari amb el seu nom

Classe Perfil

Representa un dels perfils de l'usuari, el supòsit d'aquesta classe es poder utilitzar perfils diferents, cadascun amb diferents estadístiques, partides jugades…

L'ID es un UUID que es generar en el moment de la crida del constructor.

1. **Perfil():** Constructora buida
2. **Perfil(String nom):** crea un perfil amb un nom.
3. **setNom():** permet introduir un nom de perfil.
4. **getNom():** permet obtenir el nom del perfil.

Classe Aventura

Aquesta classe representa un conjunt de Kakuros que formen el anomenat “Mode Aventura” on es juguen una determinada quantitat de Kakuros aleatoris. Les funcions que conté aquesta classe són:

1. **Aventura(int n):** Constructora amb quantitat de Kakuros n.
2. **Aventura():** Constructora buida amb quantitat de Kakuros aleatòria.
3. **getTemps():** Retorna el temps de joc a la Aventura.
4. **KakurosTotals():** Retorna el nombre de Kakuros de l’aventura.

Classe Estadística:

Aquesta classe representa un conjunt a on consta l’id d’un kakuro i el temps empleat per l’usuari a resoldre’l.
*Aquesta classe per ara no consta d’implementació, però com a atributs principals tindrà una array amb id kakuro + temps empleat i un enter a on constarà el temps total en resoldre tots els kakuros.

Classe RegistreKakuro:

Aquesta classe representa una associació entre un perfil i un kakuro, a on hi consta el temps empleat en resoldre el kakuro, aquest temps serà utilitzat per la classe estadística per formar la taula.
*Aquesta classe per ara no consta d’implementació.

Classe Ranking:

Aquesta classe representa un conjunt amb tots els id dels diferents perfils i el temps total en resoldre tots els kakuros associat a cada perfil.
*Aquesta classe per ara no consta d’implementació i el seu funcionament pot cambiar.

3. Distribució de l'arquitectura en 3 capes

Controlador de presentació
És l'encarregat de gestionar tot el que te a veure amb la UI del programa.
En aquest trobem diferents funcions per accedir al Controlador de domini i obtenir els objecte necesaris.
Aquestes funcions fan ús de diferents vistes associades a diferents classes del domini.

- Vista perfils
- Vista Kakuros
- Vista Partida
- Vista Estadística
- Vista Gestió partides
- Vista programa gestió
- Vista Aventura

Controlador de domini
És l'encarregat de comunicar-se amb el controlador de dades per obtenir el contingut de la capa de persistencia i alhora també es comunica amb el controlador de presentació per mostrar a la pantalla els objectes
Dintre del controlador trobem els objectes del programa ja descrits a l'apartat anterior

Controlador de dades
És l'encarregat de llegir i escriure les dades als i des dels fitxers. Es comunica amb la capa de domini per enviar-li aquestes dades i que el domini les pugui utilitzar.
Dintre del controlador trobem diferents funcions per les parts del programa que necessiten aquesta persistència en forma de fitxers.

- Partides
- Usuaris
- Estadístiques

4. Explicació dels algorismes

- Generar kakuros:

  Per generar el kakuro primer de tot assignem les cel·les negres pel tauler de manera aleatoria, si el kakuro té dimensions més petites de 9x9 això es tot, si pel contrari alguna de les dues coordenades és més gran comprovem a continuació que no hi hagin formacions de més de 9 cel·les blanques seguides, en cas de ser així, posem una cel·la negra.
  Un cop assignades les cel·les negres assignem valors aleatoris a les cel·les blanques, comprovant que no es repeteixen els valors en cap fila o columna.
  Quan totes les cel·les blanques tenen valor, recorrem el kakuro i realitzem les sumes pertinents dels valors de les cel·les blanques, per acumular el valor a les cel·les negres.
  Per acabar borrem els valors de les cel·les blanques per així poder resoldre el kakuro.

  El generador no acaba de funcionar bé, ja que a vegades fa formacions de cel·les tipus [Negra,blanca,negra], el qual no hauria de ser valid en un kakuro ben format. Per altre costat, amb valors grans (>15) a vegades no genera el kakuro.

  Per la següent entrega intentarem millorar aquests dos errors i millorar l'algorisme.

- Solucionar kakuros:

  Per solucionar un kakuro utilitzem un algorisme de backtracking, un cop realitzades algunes comprovacions, comencem amb l'algorisme. L'algorisme es basa en un backtracking tradicional, en el qual abans d'afegir una cel·la a la solució comprovem que estem afegint un valor valid a la solució tant a la fila com a la columna. Ho fem anant assignant valors a les cel·les blanques i comprovant que la suma de la fila o columna no sobrepasa el valor de la cel·la negra corresponent, i comprovant també que no es repeteixi el valor de la cel·la blanca en aquella fila o columna.

  A priori el solucionador funciona bé amb les proves que hem fet, tot i que no gestiona el problema de si un kakuro té més d'una solució.

  Per la propera entrega millorarem l'algorisme perquè gestioni també aquest cas.

Reparto de trabajo

Leyenda:

ALVARO

VICTOR L

ALBERT

VICTOR D

Quién?	Clase	Controlador de dominio	Capa de dades	Capa de presentación
ALVARO	Usuari	Usuarios	Ctrl Dades Usuari	Vista Gestió d'usuaris
	Perfil			
	CjtPerfils			
VICTOR L	CjtPartides	Partidas		Vista Partida
	Partida			
TODOS	Kakuro	Kakuros	Ctrl Dades Kakuros	Vista Kakuros
ALVARO I ALBERT	Registre Kakuro			
VICTOR D	Tauler*			
	Cel·la			
	Cel·la blanca			
	Cel·la negra			
ALBERT	Estadística	Estadísticas	Ctrl Dades Estadistica	Vista Estadisticas
	Rànquing			
VICTORS	Aventura	Aventura	Ctrl Dades Aventura	Vista Aventura

* A la classe Tauler, a on per ara hi ha el generador i el solucionador hem participat els quatre per fer els algorismes. Les funcions que no estan relacionades amb el generador i el solucionador les ha fet en Victor Delgado