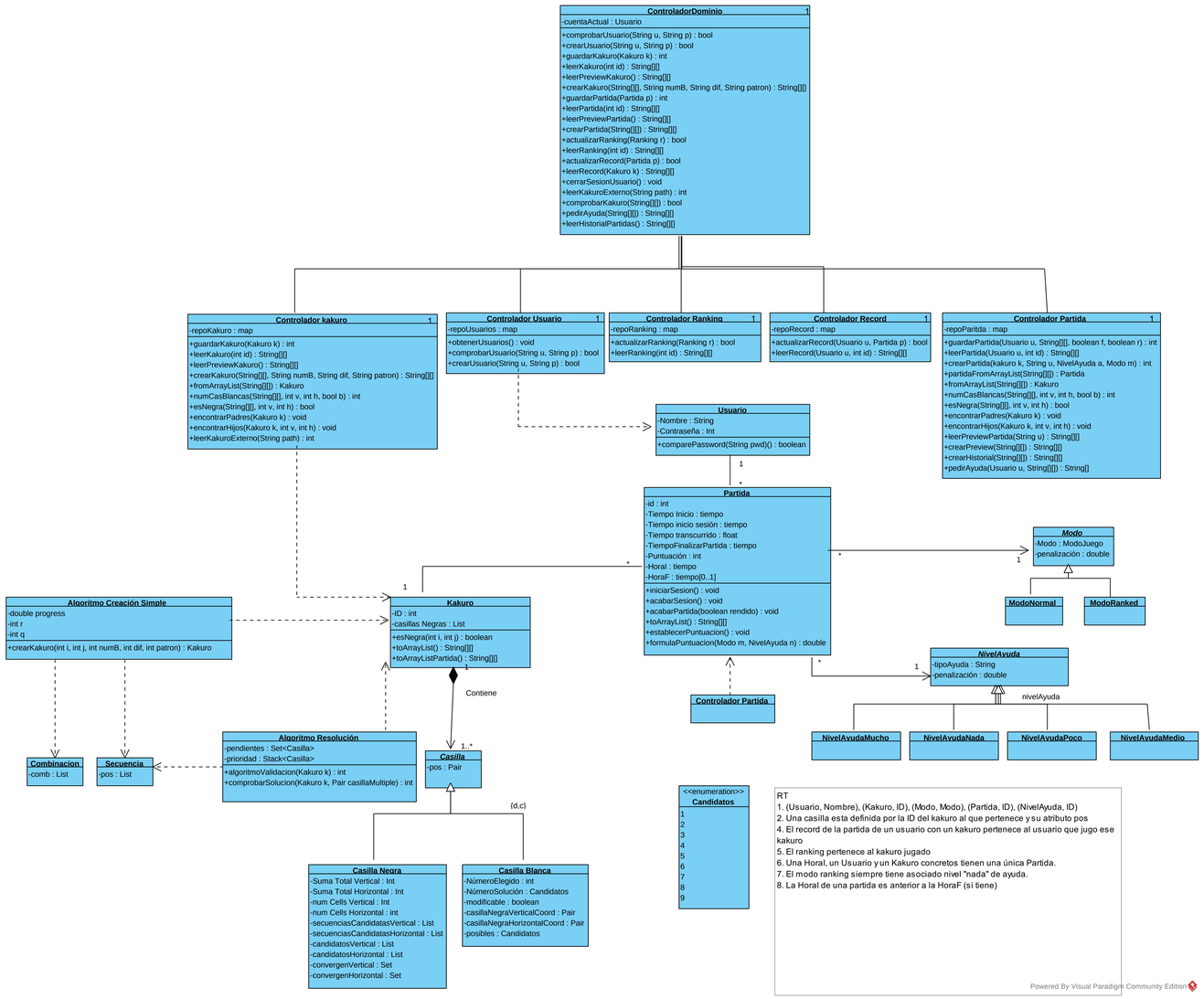
ControladorDominio
1
-cuentaActual : Usuario
+comprobarUsuario(String u, String p) : bool
+crearUsuario(String u, String p) : bool
+guardarKakuro(Kakuro k) : int
+leerKakuro(int id) : String[][]
+leerPreviewKakuro() : String[][]
+crearKakuro(String[][], String numB, String dif, String patron) : String[][]
+guardarPartida(Partida p) : int
+leerPartida(int id) : String[][]
+leerPreviewPartida() : String[][]
+crearPartida(String[][]) : String[][]
+actualizarRanking(Ranking r) : bool
+leerRanking(int id) : String[][]
+actualizarRecord(Partida p) : bool
+leerRecord(Kakuro k) : String[][]
+cerraSesionUsuario() : void
+leerKakuroExterno(String path) : int
+comprobarKakuro(String[][]) : bool
+pediryAyuda(String[][]) : String[][]
+leerHistorialPartidas() : String[][]
Controlador kakuro
1
-repoKakuro : map
+guardarKakuro(Kakuro k) : int
+leerKakuro(int id) : String[][]
+leerPreviewKakuro() : String[][]
+crearKakuro(String[][], String numB, String dif, String patron) : String[][]
+fromArrayList(String[][]) : Kakuro
+numCasBlancas(String[][], int v, int h, bool b) : int
+esNegra(String[][], int v, int h) : bool
+encontrarPadres(Kakuro k) : void
+encontrarHijos(Kakuro k, int v, int h) : void
+leerKakuroExterno(String path) : int
Controlador Usuario
1
-repoUsuarios : map
+obtenerUsuarios() : void
+comprobarUsuario(String u, String p) : bool
+crearUsuario(String u, String p) : bool
Controlador Ranking
1
-repoRanking : map
+actualizarRanking(Ranking r) : bool
+leerRanking(int id) : String[][]
Controlador Record
1
-repoRecord : map
+actualizarRecord(Usuario u, Partida p) : bool
+leerRecord(Usuario u, int id) : String[][]
Controlador Partida
1
-repoParitda : map
+guardarPartida(Usuario u, String[][], boolean f, boolean r) : int
+leerPartida(Usuario u, int id) : String[][]
+crearPartida(kakuro k, String u, NivelAyuda a, Modo m) : int
+partidaFromArrayList(String[][]) : Partida
+fromArrayList(String[][]) : Kakuro
+numCasBlancas(String[][], int v, int h, bool b) : int
+esNegra(String[][], int v, int h) : bool
+encontrarPadres(Kakuro k) : void
+encontrarHijos(Kakuro k, int v, int h) : void
+leerPreviewPartida(String u) : String[][]
+crearPreview(String[][]) : String[][]
+crearHistorial(String[][]) : String[][]
+pediryAyuda(Usuario u, String[][]) : String[]
Usuario
-Nombre : String
-Contraseña : Int
+comparePassword(String pwd)() : boolean
1
*
Partida
-id : int
-Tiempo Inicio : tiempo
-Tiempo inicio sesión : tiempo
-Tiempo transcurrido : float
-TiempoFinalizarPartida : tiempo
-Puntuación : int
-Horal : tiempo
-HoraF : tiempo[0..1]
+iniciarSesion() : void
+acabarSesion() : void
+acabarPartida(boolean rendido) : void
+toArrayList() : String[][]
+establecerPuntuacion() : void
+formulaPuntuacion(Modo m, NivelAyuda n) : double
*
1
Modo
-Modo : ModoJuego
-penalización : double
ModoNormal
ModoRanked
1
*
NivelAyuda
-tipoAyuda : String
-penalización : double
nivelAyuda
NivelAyudaMucho
NivelAyudaNada
NivelAyudaPoco
NivelAyudaMedio
Controlador Partida
1
*
Kakuro
-ID : int
-casillas Negras : List
+esNegra(int i, int j) : boolean
+toArrayList() : String[][]
+toArrayListPartida() : String[][]
Contiene
1..*
Algoritmo Creación Simple
-double progress
-int r
-int q
+crearKakuro(int i, int j, int numB, int dif, int patron) : Kakuro
Algoritmo Resolución
-pendientes : Set<Casilla>
-prioridad : Stack<Casilla>
+algoritmoValidacion(Kakuro k) : int
+comprobarSolucion(Kakuro k, Pair casillaMultiple) : int
Combinacion
-comb : List
Secuencia
-pos : List
Casilla
-pos : Pair
{d,c}
Casilla Negra
-Suma Total Vertical : Int
-Suma Total Horizontal : Int
-num Cells Vertical : Int
-num Cells Horizontal : int
-secuenciasCandidatasVertical : List
-secuenciasCandidatasHorizontal : List
-candidatosVertical : List
-candidatosHorizontal : List
-convergenVertical : Set
-convergenHorizontal : Set
Casilla Blanca
-NúmeroElegido : int
-NúmeroSolución : Candidatos
-modificable : boolean
-casillaNegraVerticalCoord : Pair
-casillaNegraHorizontalCoord : Pair
-posibles : Candidatos
<<enumeration>>
Candidatos
1
2
3
4
5
6
7
8
9
RT
1. (Usuario, Nombre), (Kakuro, ID), (Modo, Modo), (Partida, ID), (NivelAyuda, ID)
2. Una casilla esta definida por la ID del kakuro al que pertenece y su atributo pos
4. El record de la partida de un usuario con un kakuro pertenece al usuario que jugo ese kakuro
5. El ranking pertenece al kakuro jugado
6. Una Horal, un Usuario y un Kakuro concretos tienen una única Partida.
7. El modo ranking siempre tiene asociado nivel "nada" de ayuda.
8. La Horal de una partida es anterior a la HoraF (si tiene)
Powered By Visual Paradigm Community Edition