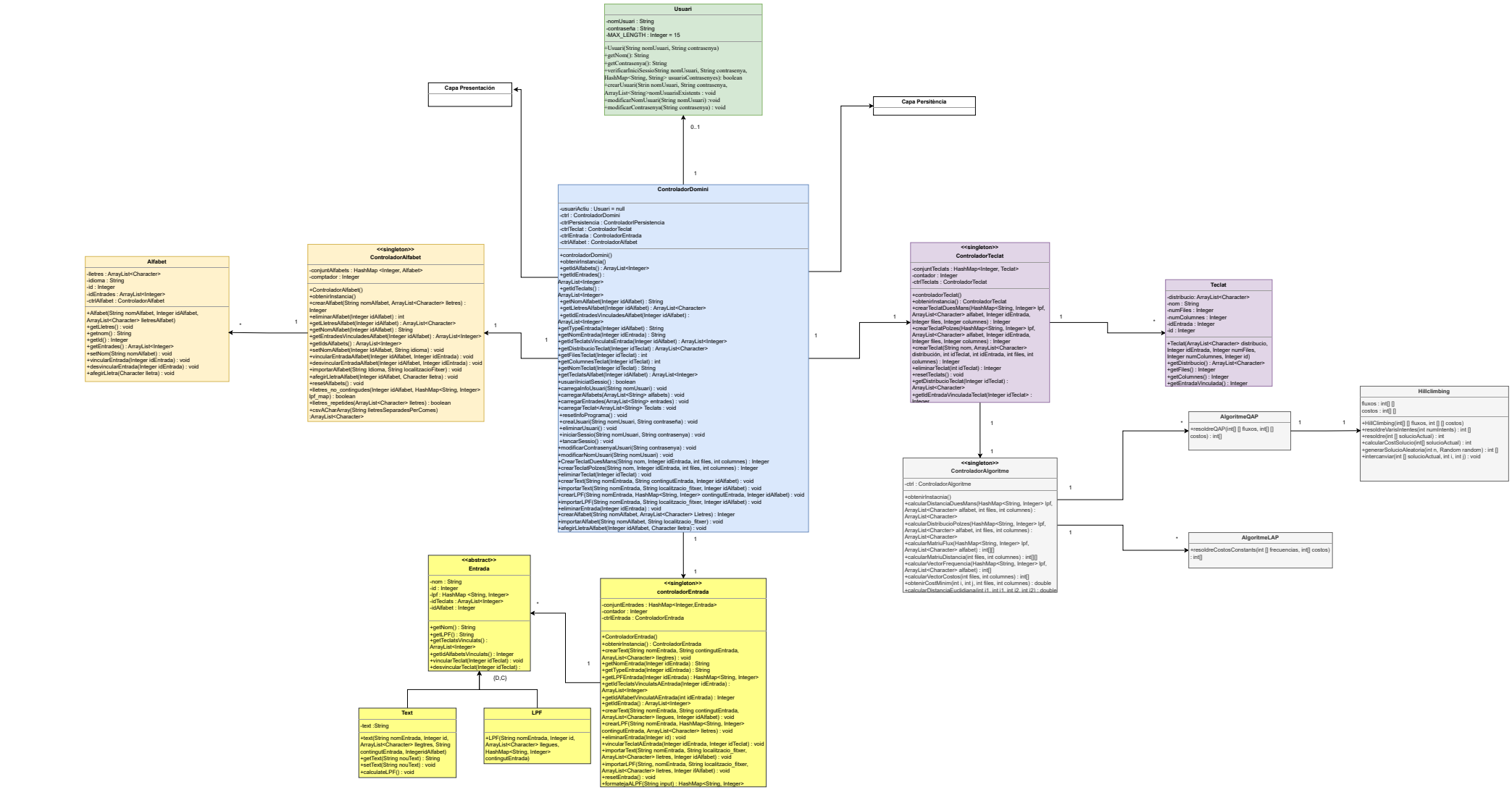
Usuari
-nomUsuari : String
-contraseña : String
-MAX_LENGTH : Integer = 15
+Usuari(String nomUsuari, String contrasenya)
+getNom(): String
+getContrasenya(): String
+verificarInicSessioString nomUsuari, String contrasenya, HashMap<String, String> usuarisContrasenyes): boolean
+crearUsuari(Strin nomUsuari, String contrasenya, ArrayList<String>nomUsuarisExistents : void
+modificarNomUsuari(String nomUsuari) :void
+modificarContrasenya(String contrasenya) : void
0..1
1
Capa Presentación
Capa Persitència
ControladorDomini
-usuariActiu : Usuari = null
-ctrl : ControladorDomini
-ctrlPersistencia : ControladorIPersistencia
-ctrlTeclat : ControladorTeclat
-ctrlEntrada : ControladorEntrada
-ctrlAlfabet : ControladorAlfabet
+controladorDomini()
+obtenirInstancia()
+getIdAlfabets() : ArrayList<Integer>
+getIdEntrades() : ArrayList<Integer>
+getIdTeclats() : ArrayList<Integer>
+getNomAlfabet(Integer idAlfabet) : String
+getLletresAlfabet(Integer idAlfabet) : ArrayList<Character>
+getIdEntradesVinculadesAlfabet(Integer idAlfabet) : ArrayList<Integer>
+getTypeEntrada(Integer idAlfabet) : String
+getNomEntrada(Integer idEntrada) : String
+getIdTeclatsVinculatsEntrada(Integer idAlfabet) : ArrayList<Integer>
+getDistribucioTeclat(Integer idTeclat) : ArrayList<Character>
+getFilesTeclat(Integer idTeclat) : int
+getColumnesTeclat(Integer idTeclat) : int
+getNomTeclat(Integer idTeclat) : String
+getTeclatsAlfabet(Integer idAlfabet) : ArrayList<Integer>
+usuariIniciatSessio() : boolean
+carregaInfoUsuari(String nomUsuari) : void
+carregarAlfabets(ArrayList<String> alfabets) : void
+carregarEntrades(ArrayList<String> entrades) : void
+carregarTeclat<ArrayList<String> Teclats : void
+resetInfoPrograma() : void
+creaUsuari(String nomUsuari, String contraseña) : void
+eliminarUsuari() : void
+iniciarSessio(String nomUsuari, String contrasenya) : void
+tancarSessio() : void
+modificarContrasenyaUsuari(String contrasenya) : void
+modificarNomUsuari(String nomUsuari) : void
+CrearTeclatDuesMans(String nom, Integer idEntrada, int files, int columnes) : Integer
+crearTeclatPolzes(String nom, Integer idEntrada, int files, int columnes) : Integer
+eliminarTeclat(Integer idTeclat) : void
+crearText(String nomEntrada, String contingutEntrada, Integer idAlfabet) : void
+importarText(String nomEntrada, String localitzacio_fitxer, Integer idAlfabet) : void
+crearLPF(String nomEntrada, HashMap<String, Integer> contingutEntrada, Integer idAlfabet) : void
+importarLPF(String nomEntrada, String localitzacio_fitxer, Integer idAlfabet) : void
+eliminarEntrada(Integer idEntrada) : void
+crearAlfabet(String nomAlfabet, ArrayList<Character> Lletres) : Integer
+importarAlfabet(String nomAlfabet, String localitzacio_fitxer) : void
+afegirLletraAlfabet(Integer idAlfabet, Character lletra) : void
1
1
Alfabet
-lletres : ArrayList<Character>
-idioma : String
-id : Integer
-idEntrades : ArrayList<Integer>
-ctrlAlfabet : ControladorAlfabet
+Alfabet(String nomAlfabet, Integer idAlfabet, ArrayList<Character> lletresAlfabet)
+getLletres() : void
+getnom() : String
+getId() : Integer
+getEntrades() : ArrayList<Integer>
+setNom(String nomAlfabet) : void
+vincularEntrada(Integer idEntrada) : void
+desvincularEntrada(Integer idEntrada) : void
+afegirLletra(Character lletra) : void
*
1
<<singleton>>
ControladorAlfabet
-conjuntAlfabets : HashMap <Integer, Alfabet>
-comptador : Integer
+ControladorAlfabet()
+obtenirInstancia()
+crearAlfabet(String nomAlfabet, ArrayList<Character> lletres) : Integer
+eliminarAlfabet(Integer idAlfabet) : int
+getLletresAlfabet(Integer idAlfabet) : ArrayList<Character>
+getNomAlfabet(Integer idAlfabet) : String
+getEntradesVinculadesAlfabet(Integer idAlfabet) : ArrayList<Integer>
+getIdsAlfabets() : ArrayList<Integer>
+setNomAlfabet(Integer idAlfabet, String idioma) : void
+vincularEntradaAlfabet(Integer idAlfabet, Integer idEntrada) : void
+desvincularEntradaAlfabet(Integer idAlfabet, Integer idEntrada) : void
+importarAlfabet(String Idioma, String localitzacioFitxer) : void
+afegirLletraAlfabet(Integer idAlfabet, Character lletra) : void
+resetAlfabets() : void
+lletres_no_continguades(Integer idAlfabet, HashMap<String, Integer> lpf, map) : boolean
+lletres_repetides(ArrayList<Character> lletres) : boolean
+csvACharArray(String lletresSeparadesPerComes) :ArrayList<Character>
1
1
<<singleton>>
ControladorTeclat
-conjuntTeclats : HashMap<Integer, Teclat>
-contador : Integer
-ctrlTeclats : ControladorTeclat
+controladorTeclat()
+obtenirInstancia() : ControladorTeclat
+crearTeclatDuesMans(HashMap<String, Integer> lpf, ArrayList<Character> alfabet, Integer idEntrada, Integer files, Integer columnes) : Integer
+crearTeclatPolzes(HashMap<String, Integer> lpf, ArrayList<Character> alfabet, Integer idEntrada, Integer files, Integer columnes) : Integer
+crearTeclat(String nom, ArrayList<Character> distribución, int idTeclat, int idEntrada, int files, int columnes) : Integer
+eliminarTeclat(int idTeclat) : Integer
+resetTeclats() : void
+getDistribucioTeclat(Integer idTeclat) : ArrayList<Character>
+getIdEntradaVinculadaTeclat(Integer idTeclat) : Integer
1
*
Teclat
-distribucio: ArrayList<Character>
-nom : String
-numFiles : Integer
-numColumnes : Integer
-idEntrada : Integer
-id : Integer
+Teclat(ArrayList<Character> distribucio, Integer idEntrada, Integer numFiles, Integer numColumnes, Integer id)
+getDistribucio() : ArrayList<Character>
+getFiles() : Integer
+getColumnes() : Integer
+getEntradaVinculada() : Integer
1
1
<<singleton>>
ControladorAlgoritme
-ctrl : ControladorAlgoritme
+obtenirInstacnia()
+calcularDistanciaDuesMans(HashMap<String, Integer> lpf, ArrayList<Character> alfabet, int files, int columnes) : ArrayList<Character>
+calcularDistribucioPolzes(HashMap<String, Integer> lpf, ArrayList<Charcter> alfabet, int files, int columnes) : ArrayList<Character>
+calcularMatriuFlux(HashMap<String, Integer> lpf, ArrayList<Character> alfabet) : int[][]
+calcularMatriuDistancia(int files, int columnes) : int[][]
+calcularVectorFrequencia(HashMap<String, Integer> lpf, ArrayList<Character> alfabet) : int[]
+calcularVectorCostos(int files, int columnes) : int[]
+obteninrCostMinim(int i, int j, int files, int columnes) : double
+calcularDistanciaEuclidianaIjd (1, int j 1, int j2, int j2) : double
1
1
*
AlgoritmeQAP
+resoldreQAP(int[] [] fluxos, int[] [] costos) : int[]
1
1
Hillclimbing
fluxos : int[] []
costos : int[] []
+HillClimbing(int[] [] fluxos, int [] [] costos)
+resoldreVarisIntentes(int numIntents) : int []
+resoldre(int [] solucioActual) : int
+calcularCostSolucio(int[] solucioActual) : int
+generarSolucioAleatoria(int n, Random random) : int []
+intercanviar(int [] solucioActual, int i, int j) : void
*
AlgoritmeLAP
+resoldreCostosConstants(int [] frecuencias, int[] costos) : int[]
<<abstract>>
Entrada
-nom : String
-id : Integer
-lpf : HashMap <String, Integer>
-idTeclats : ArrayList<Integer>
-idAlfabet : Integer
+getNom() : String
+getLPF() : String
+getTeclatsVinculats() : ArrayList<Integer>
+getIdAlfabetsVinculats() : Integer
+vincularTeclat(Integer idTeclat) : void
+desvincularTeclat(Integer idTeclat) :
*
1
{D,C}
Text
-text :String
+text(String nomEntrada, Integer id, ArrayList<Character> llegtres, String contingutEntrada, IntegerIdAlfabet)
+getText(String nouText) : String
+setText(String nouText) : void
+calculateLPF() : void
LPF
+LPF(String nomEntrada, Integer id, ArrayList<Character> llegues, HashMap<String, Integer> contingutEntrada)
<<singleton>>
controladorEntrada
-conjuntEntrades : HashMap<Integer,Entrada>
-contador : Integer
-ctrlEntrada : ControladorEntrada
+ControladorEntrada()
+obteniirInstancia() : ControladorEntrada
+crearText(String nomEntrada, String contingutEntrada, ArrayList<Character> llegtres) : void
+getNomEntrada(Integer idEntrada) : String
+getTypeEntrada(Integer idEntrada) : String
+getLPFEntrada(Integer idEntrada) : HashMap<String, Integer>
+getIdTeclatsVinculatsAEntrada(Integer idEntrada) : ArrayList<Integer>
+getIdAlfabetVinculatAEntrada(int idEntrada) : Integer
+getIdEntrada() : ArrayList<Integer>
+crearText(String nomEntrada, String contingutEntrada, ArrayList<Character> llegues, Integer idAlfabet) : void
+crearLPF(String nomEntrada, HashMap<String, Integer> contingutEntrada, ArrayList<Character> lletres) : void
+eliminarEntrada(Integer id) : void
+vincularTeclatAEntrada(Integer idEntrada, Integer idTeclat) : void
+importarText(String nomEntrada, String localitzacio_fitxer, ArrayList<Character> lletres, Integer idAlfabet) : void
+importarLPF(String, nomEntrada, String localitzacio_fitxer, ArrayList<Character> lletres, Integer idAlfabet) : void
+resetEntrada() : void
+formatejaALPF(String input) : HashMap<String, Integer>