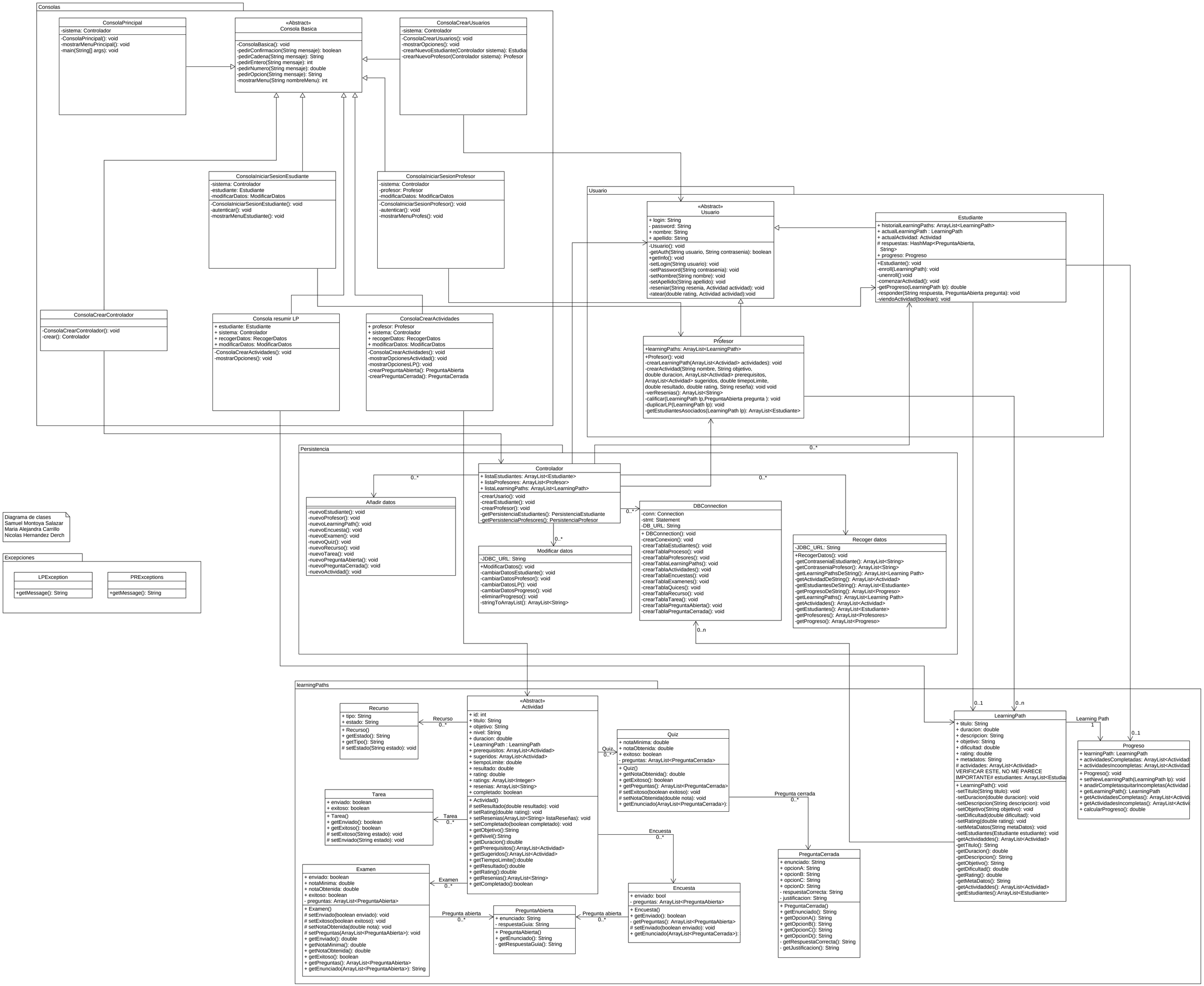
Consolas
ConsolaPrincipal
-sistema: Controlador
-ConsolaPrincipal(): void
-mostrarMenuPrincipal(): void
-main(String[] args): void
«Abstract»
Consola Basica
-ConsolaBasica(): void
-pedirConfirmacion(String mensaje): boolean
-pedirCadena(String mensaje): String
-pedirEntero(String mensaje): int
-pedirNumero(String mensaje): double
-pedirOpcion(String mensaje): String
-mostrarMenu(String nombreMenu): int
ConsolaCrearUsuarios
-sistema: Controlador
-ConsolaCrearUsuarios(): void
-mostrarOpciones(): void
-crearNuevoEstudiante(Controlador sistema): Estudia
-crearNuevoProfesor(Controlador sistema): Profesor
ConsolaIniciarSesionEsudiante
-sistema: Controlador
-estudiante: Estudiante
-modificarDatos: ModificarDatos
-ConsolaIniciarSesionEstudiante(): void
-autenticar(): void
-mostrarMenuEstudiante(): void
ConsolaIniciarSesionProfesor
-sistema: Controlador
-profesor: Profesor
-modificarDatos: ModificarDatos
-ConsolaIniciarSesionProfesor(): void
-autenticar(): void
-mostrarMenuProfes(): void
ConsolaCrearControlador
-ConsolaCrearControlador(): void
-crear(): Controlador
Consola resumir LP
+ estudiante: Estudiante
+ sistema: Controlador
+ recogerDatos: RecogerDatos
+ modificarDatos: ModificarDatos
-ConsolaCrearActividades(): void
-mostrarOpciones(): void
ConsolaCrearActividades
+ profesor: Profesor
+ sistema: Controlador
+ recogerDatos: RecogerDatos
+ modificarDatos: ModificarDatos
-ConsolaCrearActividades(): void
-mostrarOpcionesActividad(): void
-mostrarOpcionesLP(): void
-crearPreguntaAbierta(): PreguntaAbierta
-crearPreguntaCerrada(): PreguntaCerrada
Usuario
«Abstract»
Usuario
+ login: String
- password: String
+ nombre: String
+ apellido: String
-Usuario(): void
-getAuth(String usuario, String contrasenia): boolean
+getInfo(): void
-setLogin(String usuario): void
-setPassword(String contrasenia): void
-setNombre(String nombre): void
-setApellido(String apellido): void
-reseniar(String resenia, Actividad actividad): void
-ratear(double rating, Actividad actividad):void
Estudiante
+ historialLearningPaths: ArrayList<LearningPath>
+ actualLearningPath : LearningPath
+ actualActividad: Actividad
# respuestas: HashMap<PreguntaAbierta,
String>
+ progreso: Progreso
+Estudiante(): void
-enroll(LearningPath): void
-unenroll():void
-comenzarActividad(): void
-getProgreso(LearningPath lp): double
-responder(String respuesta, PreguntaAbierta pregunta): void
-viendoActividad(boolean): void
Profesor
+learningPaths: ArrayList<LearningPath>
+Profesor(): void
-crearLearningPath(ArrayList<Actividad> actividades): void
-crearActividad(String nombre, String objetivo,
double duracion, ArrayList<Actividad> prerequisitos,
ArrayList<Actividad> sugeridos, double timepoLimite,
double resultado, double rating, String reseña): void void
-verResenias(): ArrayList<String>
-calificar(LearningPath lp,PreguntaAbierta pregunta): void
-duplicarLP(LearningPath lp): void
-getEstudiantesAsociados(LearningPath lp): ArrayList<Estudiante>
Persistencia
0..*
Controlador
+ listaEstudiantes: ArrayList<Estudiante>
+ listaProfesores: ArrayList<Profesor>
+ listaLearningPaths: ArrayList<LearningPath>
-crearUsario(): void
-crearEstudiante(): void
-crearProfesor(): void
-getPersistenciaEstudiantes(): PersistenciaEstudiante
-getPersistenciaProfesores(): PersistenciaProfesor
0..*
0..*
0..*
0..*
Añadir datos
-nuevoEstudiante(): void
-nuevoProfesor(): void
-nuevoLearningPath(): void
-nuevoEncuesta(): void
-nuevoExamen(): void
-nuevoQuiz(): void
-nuevoRecurso(): void
-nuevoTarea(): void
-nuevoPreguntaAbierta(): void
-nuevoPreguntaCerrada(): void
-nuevoActividad(): void
DBConnection
-conn: Connection
-stmt: Statement
-DB_URL: String
+ DBConnection(): void
-crearConexion(): void
-crearTablaEstudiantes(): void
-crearTablaProceso(): void
-crearTablaProfesores(): void
-crearTablaLearningPaths(): void
-crearTablaActividades(): void
-crearTablaEncuestas(): void
-crearTablaExamenes(): void
-crearTablaQuices(): void
-crearTablaRecurso(): void
-crearTablaTarea(): void
-crearTablaPreguntaAbierta(): void
-crearTablaPreguntaCerrada(): void
Modificar datos
-JDBC_URL: String
+ModificarDatos(): void
-cambiarDatosEstudiante(): void
-cambiarDatosProfesor(): void
-cambiarDatosLP(): void
-cambiarDatosProgreso(): void
-eliminarProgreso(): void
-stringToArrayList(): ArrayList<String>
Recoger datos
-JDBC_URL: String
+RecogerDatos(): void
-getContraseniaEstudiante(): ArrayList<String>
-getContraseniaProfesor(): ArrayList<String>
-getLearningPathsDeString(): ArrayList<Learning Path>
-getActividadDeString(): ArrayList<Actividad>
-getEstudiantesDeString(): ArrayList<Estudiante>
-getProgresoDeString(): ArrayList<Progreso>
-getLearningPaths(): ArrayList<Learning Path>
-getActividades(): ArrayList<Actividad>
-getEstudiantes(): ArrayList<Estudiante>
-getProfesores(): ArrayList<Profesores>
-getProgreso(): ArrayList<Progreso>
0..n
Diagrama de clases
Samuel Montoya Salazar
Maria Alejandra Carrillo
Nicolas Hernandez Derch
Excepciones
LPException
+getMessage(): String
PRExceptions
+getMessage(): String
learningPaths
Recurso
+ tipo: String
+ estado: String
+ Recurso()
+ getEstado(): String
+ getTipo(): String
# setEstado(String estado): void
Recurso
0..*
«Abstract»
Actividad
+ id: int
+ titulo: String
+ objetivo: String
+ nivel: String
+ duracion: double
+ LearningPath : LearningPath
+ prerequisitos: ArrayList<Actividad>
+ sugeridos: ArrayList<Actividad>
+ tiempoLimite: double
+ resultado: double
+ rating: double
+ ratings: ArrayList<Integer>
+ resenias: ArrayList<String>
+ completado: boolean
+ Actividad()
# setResultado(double resultado): void
# setRating(double rating): void
+ setResenias(ArrayList<String> listaReseñas): void
+ setCompletado(boolean completado): void
+ getObjetivo():String
+ getNivel():String
+ getDuracion():double
+ getPrerequisitos():ArrayList<Actividad>
+ getSugeridos():ArrayList<Actividad>
+ getTiempoLimite():double
+ getResultado():double
+ getRating():double
+ getResenias():ArrayList<String>
+ getCompletado():boolean
Quiz
0..*
Quiz
+ notaMinima: double
+ notaObtenida: double
+ exitoso: boolean
- preguntas: ArrayList<PreguntaCerrada>
+ Quiz()
+ getNotaObtenida(): double
+ getExitoso(): boolean
+ getPreguntas(): ArrayList<PreguntaCerrada>
# setExitoso(boolean exitoso): void
# setNotaObtenida(double nota): void
+ getEnunciado(ArrayList<PreguntaCerrada>):
Pregunta cerrada
0..*
Tarea
+ enviado: boolean
+ exitoso: boolean
+ Tarea()
+ getEnviado(): boolean
+ getExitoso(): boolean
# setExitoso(String estado): void
# setEnviado(String estado): void
Tarea
0..*
Encuesta
0..*
Examen
+ enviado: boolean
+ notaMinima: double
+ notaObtenida: double
+ exitoso: boolean
- preguntas: ArrayList<PreguntaAbierta>
+ Examen()
# setEnviado(boolean enviado): void
# setExitoso(boolean exitoso): void
# setNotaObtenida(double nota): void
# setPreguntas(ArrayList<PreguntaAbierta>): void
+ getEnviado(): double
+ getNotaMinima(): double
+ getNotaObtenida(): double
+ getExitoso(): boolean
+ getPreguntas(): ArrayList<PreguntaAbierta>
+ getEnunciado(ArrayList<PreguntaAbierta>): String
Examen
0..*
Pregunta abierta
0..*
PreguntaAbierta
+ enunciado: String
- respuestaGuia: String
+ PreguntaAbierta()
+ getEnunciado(): String
- getRespuestaGuia(): String
Pregunta abierta
0..*
Encuesta
+ enviado: bool
- preguntas: ArrayList<PreguntaAbierta>
+ Encuesta()
+ getEnviado(): boolean
- getPreguntas(): ArrayList<PreguntaAbierta>
setEnviado(boolean enviado): void
+ getEnunciado(ArrayList<PreguntaCerrada>):
PreguntaCerrada
+ enunciado: String
+ opcionA: String
+ opcionB: String
+ opcionC: String
+ opcionD: String
- respuestaCorrecta: String
- justificacion: String
+ PreguntaCerrada()
+ getEnunciado(): String
+ getOpcionA(): String
+ getOpcionB(): String
+ getOpcionC(): String
+ getOpcionD(): String
- getRespuestaCorrecta(): String
- getJustificacion(): String
0..1
0..n
LearningPath
+ titulo: String
+ duracion: double
+ descripcion: String
+ objetivo: String
+ dificultad: double
+ rating: double
+ metadatos: String
# actividades: ArrayList<Actividad>
VERIFICAR ESTE, NO ME PARECE
IMPORTANTE# estudiantes: ArrayList<Estudia
+ LearningPath(): void
-setTitulo(String titulo): void
-setDuracion(double duracion): void
-setDescripcion(String descripcion): void
-setObjetivo(String objetivo): void
-setDificultad(double dificultad): void
-setRating(double rating): void
-setMetaDatos(String metaDatos): void
-setEstudiantes(Estudiante estudiante): void
-getActividaddes(): ArrayList<Actividad>
-getTitulo(): String
-getDuracion(): double
-getDescripcion(): String
-getObjetivo(): String
-getDificultad(): double
-getRating(): double
-getMetaDatos(): String
-getActividaddes(): ArrayList<Actividad>
-getEstudiantes():ArrayList<Estudiante>
Learning Path
1
0..1
Progreso
+ learningPath: LearningPath
+ actividadesCompletadas: ArrayList<Actividad
+ actividadesIncoompletas: ArrayList<Actividad
+ Progreso(): void
+ setNewLearningPath(LearningPath lp): void
+ anadirCompletasquitarIncompletas(Actividad
+ getLearningPath(): LearningPath
+ getActividadesCompletas(): ArrayList<Activid
+ getActividadesIncompletas(): ArrayList<Activi
+ calcularProgreso(): double