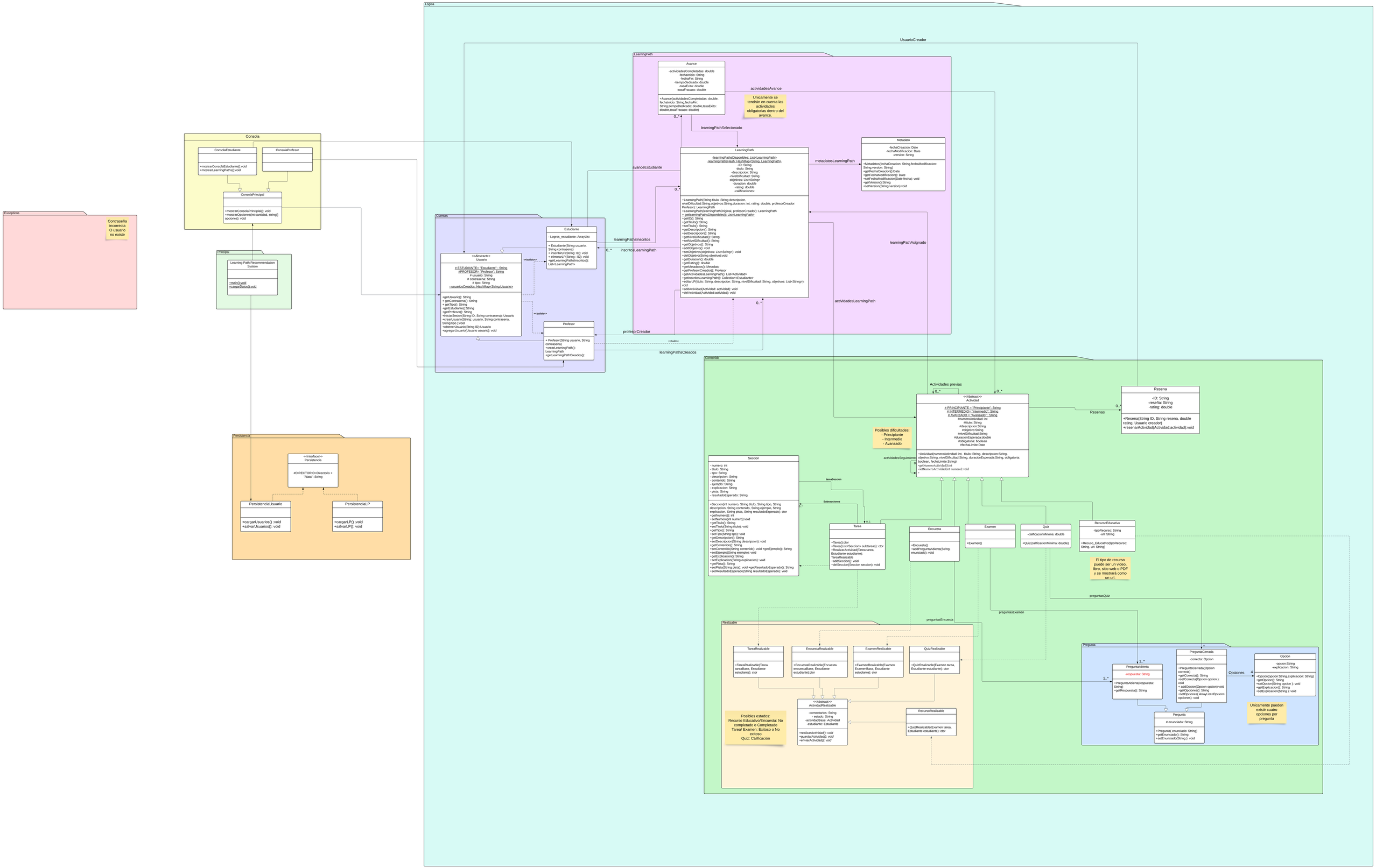
Logica
UsuarioCreador
LearningPath
Avance
-actividadesCompletadas: double
-fechainicio: String
-fechaFin: String
-tiempoDedicado: double
-tasaExito: double
-tasaFracaso: double
+Avance(actividadesCompletadas: double, fechainicio: String,fechaFin: String,tiempoDedicado: double,tasaExito: double,tasaFracaso: double)
actividadesAvance
Unicamente se tendrán en cuenta las actividades obligatorias dentro del avance.
0..*
learningPathSelecionado
LearningPath
-learningPathsDisponibles: List<LearningPath>
-learningPathsHash: HashMap<String, LearningPath>
-ID: String
-titulo: String
-descripcion: String
-nivelDificultad: String
-objetivos: List<String>
-duracion: double
-rating: double
-calificaciones:
+LearningPath(String titulo ,String descripcion, nivelDificultad:String,objetivos:String,duracion: int, rating: double, profesorCreador: Profesor): LearningPath
+LearningPath(learningPathOriginal, profesorCreador): LearningPath
+ getlearningPathsDisponibles(): List<LearningPath>
+getID(): String
+getTitulo(): String
+setTitulo(): String
+getDescripcion(): String
+setDescripcion(): String
+getNivelDificultad(): String
+setNivelDificultad(): String
+getObjetivos(): String
+addObjetivo(): void
+setObjetivos(objetivos: List<String>): void
+delObjetivo(String objetivo):void
+getDuracion(): double
+getRating(): double
+getMetadatos(): Metadato
+getProfesorCreador(): Profesor
+getActividadesLearningPath(): List<Actividad>
+getInscritosLearningPath(): Collection<Estudiante>
+editarLP(titulo: String, descripcion: String, nivelDificultad: String, objetivos: List<String>): void
+addActividad(Actividad: actividad): void
+delActividad(Actividad:actividad): void
avanceEstudiante
0..*
metadatosLearningPath
Metadato
-fechaCreacion: Date
-fechaModificacion: Date
-version: String
+Metadatos(fechaCreacion: String,fechaModificacion: String,version: String)
+getFechaCreacion():Date
+getFechaModificacion(): Date
+setFechaModificacion(Date fecha): void
+getVersion():String
+setVersion(String version):void
learningPathAsignado
actividadesLearningPath
0..*
Consola
ConsolaEstudiante
+mostrarConsolaEstudiante():void
+mostrarLearningPaths():void
ConsolaProfesor
ConsolaPrincipal
+mostrarConsolaPrincipal(): void
+mostrarOpciones(int cantidad, string[] opciones): void
Exceptions
Contraseña incorrecta
O usuario no existe
Principal
Learning Path Recommendation System
+main():void
+cargaDatos():void
Cuentas
Estudiante
- Logros_estudiante: ArrayList
+ Estudiante(String usuario, String contrasena)
+ inscribirLP(String: ID): void
+ eliminarLP(String : ID): void
+getLearningPathsInscritos():List<LearningPath>
learningPathsInscritos
0..*
inscritosLearningPath
<<Abstract>>
Usuario
ESTUDIANTE= "Estudiante" : String
#PROFESOR= "Profesor": String
usuario: String
contrasena: String
tipo: String
- usuariosCreados: HashMap<String,Usuario>
+getUsuario(): String
+ getContrasena(): String
+ getTipo(): String
+getEstudiante():String
+getProfesor(): String
+iniciarSesion(String ID, String contrasena): Usuario
+crearUsuario(String: usuario, String contrasena, String:tipo):void
+obtenerUsuario(String ID):Usuario
+agregarUsuario(Usuario usuario): void
<<builds>>
<<builds>>
Profesor
+ Profesor(String usuario, String contrasena)
+crearLearningPath():LearningPath
+getLearningPathCreados():
profesorCreador
<<build>>
learningPathsCreados
Persistencia
<<interface>>
Persistencia
#DIRECTORIO=Directorio + "data/": String
PersistenciaUsuario
+cargarUsuarios() :void
+salvarUsuarios(): void
PersistenciaLP
+cargarLP() :void
+salvarLP(): void
Contenido
Actividades previas
0..*
0..*
<<Abstract>>
Actividad
PRINCIPIANTE = "Principiante": String
INTERMEDIO= "Intermedio": String
AVANZADO = "Avanzado": String
#numeroActividad: int
#titulo: String
#descripcion:String
#objetivo:String
#nivelDificultad:String
#duracionEsperada:double
#obligatoria: boolean
#fechaLimite:Date
+Actividad(numeroActividad: int, titulo: String, descripcion:String, objetivo:String, nivelDificultad:String, duracionEsperada:String, obligatoria: boolean, fechaLimite:String)
+getNumeroActividad():int
+setNumeroActividad(int numero): void
+
Posibles dificultades:
- Principiante
- Intermedio
- Avanzado
actividadesSeguimiento
Resena
-ID: String
-reseña: String
-rating: double
+Resena(String ID, String resena, double rating, Usuario creador)
+resenarActividad(Actividad:actividad):void
0..*
Resenas
Seccion
- numero: int
- titulo: String
- tipo: String
- descripcion: String
- contenido: String
- ejemplo: String
- explicacion: String
- pista: String
- resultadoEsperado: String
+Seccion(int numero, String titulo, String tipo, String descripcion, String contenido, String ejemplo, String explicacion, String pista, String resultadoEsperado): ctor
+getNumero():int
+setNumero(int numero):void
+getTitulo(): String
+setTitulo(String titulo): void
+getTipo(): String
+setTipo(String tipo): void
+getDescripcion(): String
+setDescripcion(String descripcion): void
+getContenido(): String
+setContenido(String contenido): void +getEjemplo(): String
+setEjemplo(String ejemplo): void
+getExplicacion(): String
+setExplicacion(String explicacion): void
+getPista(): String
+setPista(String pista): void +getResultadoEsperado(): String
+setResultadoEsperado(String resultadoEsperado): void
tareaSeccion
Subsecciones
1..*
0..1
Tarea
+Tarea():ctor
+Tarea(List<Seccion> subtareas): ctor
+RealizarActividad(Tarea tarea, Estudiante estudiante): TareaRealizable
+addSeccion(): void
+delSeccion(Seccion seccion): void
Encuesta
+Encuesta()
+addPreguntaAbierta(String enunciado): void
Examen
+Examen()
Quiz
-calificacionMinima: double
+Quiz(calificacionMinima: double)
RecursoEducativo
-tipoRecurso: String
-url: String
+Recuso_Educativo(tipoRecurso: String, url: String)
El tipo de recurso puede ser un video, libro, sitio web o PDF y se mostrará como un url.
preguntasQuiz
preguntasExamen
preguntasEncuesta
Realizable
TareaRealizable
+TareaRealizable(Tarea tareaBase, Estudiante estudiante): ctor
EncuestaRealizable
+EncuestaRealizable(Encuesta encuestaBase, Estudiante estudiante):ctor
ExamenRealizable
+ExamenRealizable(Examen ExamenBase, Estudiante estudiante): ctor
QuizRealizable
+QuizRealizable(Examen tarea, Estudiante estudiante): ctor
<<Abstract>>
ActividadRealizable
-comentarios: String
- estado: String
-actividadBase: Actividad
-estudiante: Estudiante
+realizarActividad(): void
+guardarActividad(): void
+enviarActividad(): void
RecursoRealizable
+QuizRealizable(Examen tarea, Estudiante estudiante): ctor
Posibles estados:
Recurso Educativo/Encuesta: No completado o Completado
Tarea/ Examen: Exitoso o No exitoso
Quiz: Calificación
Pregunta
PreguntaAbierta
-respuesta: String
+PreguntaAbierta(respuesta: String)
+getRespuesta(): String
1..*
1..*
PreguntaCerrada
-correcta: Opcion
+PreguntaCerrada(Opcion correcta)
+getCorrecta(): String
+setCorrecta(Opcion opcion): void
+ addOpcion(Opcion opcion):void
+getOpciones(): String
+setOpciones(ArrayList<Opcion> opciones): void
Opciones
4
Opcion
-opcion:String
-explicacion: String
+Opcion(opcion:String,explicacion: String)
+getOpcion(): String
+setOpcion(String opcion): void
+getExplicacion(): String
+setExplicacion(String): void
Unicamente pueden existir cuatro opciones por pregunta
Pregunta
enunciado: String
+Pregunta(enunciado: String)
+getEnunciado():String
+setEnunciado(String): void