



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE  
PUEBLA**

**FACULTAD CIENCIAS DE LA COMPUTACIÓN**

**“DESARROLLO DE UN OBJETO DE APRENDIZAJE  
BASADO EN EL CÁLCULO DE TRAYECTORIAS DE  
APRENDIZAJE UTILIZANDO ANSWER SET  
PLANNING”**

**TESIS**

**PARA OBTENER EL TÍTULO DE:**

**LICENCIATURA EN INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**PRESENTA:**

**BETZAIT ABURTO GUEVARA**

**ASESORES:**

**DRA. CLAUDIA ZEPEDA CORTÉS**

**DR. JOSÉ LUIS CARBALLIDO CARRANZA**

**PUEBLA, PUE.**

**OCTUBRE, 2019**

# ÍNDICE

<b>RESUMEN</b> .....	2
<b>INTRODUCCIÓN</b> .....	3
<b>CAPITULO 1: FUNDAMENTOS TEORICOS</b> .....	5
<b>1.1 ¿QUÉ ES UN OBJETO DE APRENDIZAJE?</b> .....	6
1.1.1 Características de un Objeto de Aprendizaje .....	6
1.1.2 Ventajas y desventajas de un Objeto de Aprendizaje .....	7
<b>1.2 DISEÑO DE UN OBJETO DE APRENDIZAJE</b> .....	8
1.2.1 Modelo ADDIE .....	8
1.2.2 Diseño de Interfaces .....	9
<b>1.3 ANSWER SET PLANNING</b> .....	10
<b>CAPITULO 2: ANALISIS Y DISEÑO</b> .....	11
<b>2.1 ANALISIS Y DISEÑO DEL OA PARA EL CURSO DE MATEMÁTICAS DISCRETAS</b> .	12
2.1.1 Principales objetivos de aprendizaje para el curso .....	12
2.1.2 Desarrollo de los temas a enseñar .....	13
2.1.3 Medios audiovisuales como estrategia de enseñanza .....	14
2.1.4 Estructura general del OA .....	15
2.1.5 Prototipos de interfaces .....	17
<b>2.2 HERRAMIENTAS DE DESARROLLO</b> .....	19
<b>CAPITULO 3: DESARROLLO E IMPLEMENTACION</b> .....	22
<b>3.1 PERSPECTIVA DEL OBJETO DE APRENDIZAJE</b> .....	23
3.1.1 Arquitectura del sistema .....	23
<b>3.2 FUNCIONALIDADES DEL SISTEMA</b> .....	25
3.2.1 Funcionalidad Java y DLV .....	26
3.2.2 Integración de DLV y Java .....	35
<b>CAPITULO 4: RESULTADOS DEL OBJETO DE APRENDIZAJE</b> .....	39
<b>4.1 INTERACCIÓN AMIGABLE CON EL USUARIO</b> .....	40
<b>CONCLUSIONES</b> .....	51
<b>REFERENCIAS</b> .....	52

# RESUMEN

Un Objeto de Aprendizaje (OA) es la unidad mínima de contenido, capaz de propiciar un proceso de enseñanza-aprendizaje, teniendo como base el diseño instruccional.

Un OA se organiza en base a una jerarquía composicional de niveles de granularidad que van desde los objetos multimedia y objetos de información (imágenes, video, textos planos, entre otros), hasta conjuntos de contenido educativo más complejos como secciones, unidades o módulos.

Su estructura permite construir contenidos educativos a partir de componentes o piezas que pueden ser ensambladas para el logro de los objetivos.

El contenido de un OA corresponde al conjunto de saberes orientados al desarrollo de una competencia. Estos deben ser representados a través de medios audiovisuales y auditivos.

Las actividades de aprendizaje corresponden a las propuestas de trabajo elaboradas con el fin de que sean desarrolladas por el usuario para adquirir o desarrollar una competencia, un conocimiento, destreza, actitud o valor con base en un contenido.

Precisamente el orden en que deben ser llevadas a cabo las actividades de aprendizaje y su evaluación es lo que nosotros llamamos trayectorias de aprendizaje. Estas trayectorias de aprendizaje serán calculadas utilizando técnicas de inteligencia artificial como lo es Answer Set Planning.

# INTRODUCCIÓN

La educación en México hoy en día se encuentra en estado alarmante, diferentes factores como condiciones estructurales, socioeconómicas o la deserción estudiantil por mencionar algunas son algunas de las causas que impiden el despliegue de calidad de nuestra educación. Enfocándonos directamente en la deserción estudiantil estadísticamente se reconoce que el 48% de los estudiantes mexicanos de 15 años tienen riesgo de abandonar sus estudios, una de las soluciones para esta problemática es la educación a través de medios virtuales debido a que flexibiliza más el acceso y los tiempos de aprendizaje. Precisamente aquí es donde manejamos el concepto de un Objeto de Aprendizaje (OA) ya que es parte de una filosofía de soporte digital orientado a la educación virtual, los beneficios de un OA pueden ser muy satisfactorios ya que promueve mayor autonomía y dedicación por parte de los alumnos.

Esta tesis describe el desarrollo de un Objeto de Aprendizaje basado en el cálculo de trayectorias de aprendizaje utilizando Answer Set Planning que tiene como propósito primordial generar un sistema de apoyo a la enseñanza en conceptos básicos de Matemáticas Discretas utilizando técnicas de inteligencia artificial como lo es Answer Set Planning.

Un Objeto de Aprendizaje (OA) según (Wiley, 2000), es cualquier recurso digital que puede ser reutilizado para apoyar el aprendizaje. La presencia de un Objeto de Aprendizaje genera una nueva forma de pensar sobre el contenido a aprender, pudiendo generar una mejor facilidad de abstracción sobre temas complejos para los estudiantes o bien fomentar la habilidad para trabajar de forma autónoma y adquirir nuevas estrategias de aprendizaje y desarrollo de competencias genéricas.

Tomando el panorama descrito anteriormente se denota la necesidad de integrar el proceso de enseñanza-aprendizaje para el OA precisando el orden jerárquico de los medios audiovisuales como las actividades de evaluación, teniendo como base el Answer Set Planning que calculara el orden instruccional de las trayectorias de aprendizaje.

La presente investigación se encuentra dividida en cuatro capítulos los cuales están distribuidos de la siguiente manera.

En el primer capítulo se estudiarán los fundamentos teóricos y de desarrollo de los OA con el fin de conocer las características particulares de estos sistemas de gestión de aprendizaje desde el punto de vista educativo.

Toda investigación necesita un marco teórico es por eso que en el segundo capítulo se elaborara el análisis, diseño e implementación del OA para el curso de Matemáticas discretas.

Posteriormente en el capítulo tres se implementará e integrará el OA utilizando DLV y Java con el objetivo de modular su funcionamiento a partir de las trayectorias calculadas con Answer Set Planning que nos permite llegar a una meta u objetivo final.

Finalmente, en el capítulo cuatro se presentan los resultados obtenidos tomando en cuenta las metodologías y análisis de desarrollo que fueron utilizadas para tener como resultado final un OA basado en el cálculo de trayectorias de aprendizaje utilizando Answer Set Planning.

# **CAPITULO 1:**

## **FUNDAMENTOS**

### **TEORICOS**



## 1.1 ¿QUÉ ES UN OBJETO DE APRENDIZAJE?

Hablar de un OA (Objeto De Aprendizaje) es un término relativamente reciente, existen múltiples definiciones por diversos autores, (Mason, Weller & Pegler, 2003), lo definen como “Una pieza digital de material de aprendizaje que direcciona a un tema claramente identificable o salida de aprendizaje y que tiene el potencial de ser reutilizado en diferentes contextos”.

Algunas otras pueden ser muy genéricas, como la de (Hodgins, 1992), quien define a estos como “cualquier recurso digital que puede ser usado como soporte para el aprendizaje”.

Podemos seguir enunciando otras definiciones, sin embargo, todas ellas coinciden en que los OA son recursos digitales utilizados como apoyo para el aprendizaje que pueden ser utilizados en diversos contextos educativos.

### 1.1.1 Características de un Objeto de Aprendizaje

Dicho esto, existen infinidad de recursos digitales que podrían ser catalogados como un OA tales como videos, animaciones, archivos de texto, audios, diapositivas etc. Sin embargo, para poder ser catalogado como un OA habría que tener en cuenta ciertas características que se deben de cumplir entre las cuales según (García, 2005) son:

**Reutilización:** objeto con capacidad para ser usado en contextos y propósitos educativos diferentes y para adaptarse y combinarse dentro de nuevas secuencias formativas.

**Educabilidad:** capacidad para generar aprendizaje.

**Interoperabilidad:** capacidad para poder integrarse en estructuras y sistemas (plataformas) diferentes.

**Accesibilidad:** facilidad para ser identificados, buscados y encontrados gracias al correspondiente etiquetado a través de diversos descriptores (metadatos) que permitirían la catalogación y almacenamiento en el correspondiente repositorio.

**Durabilidad:** vigencia de la información de los objetos, sin necesidad de nuevos diseños.

**Independencia y autonomía:** de los objetos con respecto de los sistemas desde los que fueron creados y con sentido propio.

**Generatividad:** capacidad para construir contenidos, objetos nuevos derivados de él. Capacidad para ser actualizados o modificados, aumentando sus potencialidades a través de la colaboración.

**Flexibilidad, versatilidad y funcionalidad:** capacidad para poder combinarse con diversas propuestas de áreas del saber diferente.

Todos estos atributos muestran a los OA como módulos que facilitan el desarrollo del aprendizaje. El tamaño de un OA es variable y esto se conoce como granularidad. La cual es la forma en la que los recursos están unidos entre sí, que van desde los objetos multimedia y objetos de información (imágenes, video, textos planos, entre otros), hasta conjuntos de contenido educativo más complejos como secciones, unidades o módulos. Podríamos decir que un OA es una unidad algo extensa, pero en realidad, podemos desarrollar un solo objetivo el cual contenga actividades y una evaluación final ya que entre más pequeña sea la extensión, más flexible y fácil de reutilizar será el OA.

### **1.1.2 Ventajas y desventajas de un Objeto de Aprendizaje**

Los Objetos de aprendizaje aportan muchos beneficios tanto para la parte usuario (alumno), como del formador (profesor), entre los cuales según (Sánchez, 2014) se desatacan los siguientes:

Para los Alumnos:

- Posibilita la autonomía del aprendizaje.
- Se integran en el proceso de aprendizaje.
- Acceso en cualquier momento.
- Se adaptan al ritmo de aprendizaje del alumno.

Para el Profesor:

- Adaptan los programas a las necesidades específicas de los alumnos.
- Ofrecen alternativas de aprendizaje.
- Crean contenidos que pueden ser rediseñados y adaptados a las nuevas tecnologías.

En términos generales podemos ver como los OA, ofrecen oportunidades para los profesores en la preparación de recursos de aprendizaje y por otro lado, una adaptación para fomentar la autonomía y desarrollo de habilidades para el alumno.

## 1.2 DISEÑO DE UN OBJETO DE APRENDIZAJE

El diseño instruccional es el proceso a través del cual se crea un ambiente de aprendizaje, así como los materiales necesarios, con el objetivo de ayudar al alumno a desarrollar la capacidad necesaria para lograr ciertas tareas (Broderick, 2001). Este proceso puede ser utilizado para estructurar un OA apto para las necesidades de los usuarios, se tiene que plantear el objetivo, actividades y la manera en que se evaluará el aprendizaje de los usuarios que utilicen el OA.

Desde esta perspectiva, existen diversas metodologías de desarrollo que intervienen en la construcción de Objetos de Aprendizaje, pero en esta ocasión abordaremos únicamente el modelo ADDIE.

### 1.2.1 Modelo ADDIE

El modelo ADDIE (Análisis, Diseño, Desarrollo, Implementación, Evaluación) es un diseño instruccional interactivo y está compuesto según (Belloch, 2013) por las siguientes fases que se describen a continuación:

- **Análisis:** El paso inicial es analizar el alumnado, el contenido y el entorno cuyo resultado será la descripción de una situación y sus necesidades formativas.
- **Diseño:** Se desarrolla un programa del curso deteniéndose especialmente en el enfoque pedagógico y en el modo de secuenciar y organizar el contenido.
- **Desarrollo:** La creación real (producción) de los contenidos y materiales de aprendizaje basados en la fase de diseño.
- **Implementación:** Ejecución y puesta en práctica de la acción formativa con la participación de los alumnos.
- **Evaluación:** Esta fase consiste en llevar a cabo la evaluación formativa de cada una de las etapas del proceso ADDIE y la evaluación sumativa a través de pruebas específicas para analizar los resultados de la acción formativa.

En el desarrollo de esta tesis, se tomará como guía este modelo debido a su versatilidad y fácil adaptación para la implementación del OA que se pretende conseguir como resultado de esta tesis.

### 1.2.2 Diseño de Interfaces

La etapa del diseño es de suma importancia, puede resultar ser todo un desafío ya que se debe elegir el contenido y con base en los contenidos se seleccionan los recursos multimedia y la manera en que estos recursos interactuarán con los usuarios.

Existen subdivisiones de diseño que nos serán útiles para el armado de nuestro OA las cuales según (Rosanigo, López & Bianchi, 2008) son:

- **Diseño de la información:** Significa clarificar los objetivos, definir, estructurar y presentar el conocimiento.
  
- **Diseño de la presentación:** Se define el estilo y el diseño de los elementos del contenido del OA, buscando lograr interfaces agradables y armónicas. Responde a ¿qué apariencia debe tener?
  
- **Diseño de la interacción:** Apunta al ¿cómo debe funcionar?, diseño de controles y de la interacción usuario-ordenador.

Para el diseño de nuestra interfaz nos enfocaremos principalmente en la subdivisión de diseño de interacción ya que el desarrollo de este OA es un prototipo que tiene como objetivo primordial una funcionalidad con la mínima interacción usuario-ordenador debido a que está basado en el paradigma Answer Set Planning el cual generará esta interacción basada en el conjunto de respuestas que correspondan a un plan en específico.

## 1.3 ANSWER SET PLANNING

La programación lógica es un tipo de programación que se basa principalmente en la lógica formal, este paradigma define una aplicación de reglas de la lógica para deducir conclusiones a partir de datos y alcanzar una solución automática. Todos los programas desarrollados en un lenguaje de programación lógica son un conjunto de oraciones que expresan reglas y hechos sobre algún problema en específico.

Una de las principales extensiones de la programación lógica es Answer Set Planning (ASP) un enfoque de la programación lógica desarrollado en los últimos años, su idea principal es programar conjuntos de respuestas cuyos conjuntos de respuestas corresponden a soluciones de un problema en específico.

ASP cuenta con diferentes implementaciones muy eficaces tales como DLV, ASPeRiX, Clasp, ASSAT, Cmodels, etc., que dan un conjunto de soluciones a problemas declarativos además de lograr el desarrollo de aplicaciones de inteligencia artificial [7], composición de música automática [7], planificación de movimientos de un robot [8], entre otros.

Particularmente se usará DLV un sistema de programación lógica disyuntiva que implementa la semántica del modelo estable bajo el paradigma de programación de conjuntos de respuestas. DLV es un programa que se conforma por el conocimiento de fondo y la descripción del dominio [9]. La descripción del dominio se estructura de fluentes, acciones y reglas. Los fluentes representan las propiedades básicas del sistema, así como el estado de este. Las acciones describen las operaciones que permiten realizar una transición de estados, la acción es aplicable solo si se cumple alguna precondition con la palabra reservada *executable* y especificando el efecto de la acción con la palabra reservada *caused*. Una vez iniciado el recorrido a través de los estados se plantea llegar al estado final, una planificación que se define como una secuencia de conjunto de acciones llamadas trayectorias.

Estructura de la descripción del dominio de DLV [10]:

```
fluents: <declaración de fluentes >  
actions: <declaración de acciones>  
always: <reglas>  
initially: <inicialización de reglas>  
noConcurrency.  
goal: <consulta>?
```

# CAPITULO 2: ANALISIS Y DISEÑO



## 2.1 ANALISIS Y DISEÑO DEL OA PARA EL CURSO DE MATEMÁTICAS DISCRETAS

Las Matemáticas discretas son un área de las Matemáticas encargadas del estudio de los conjuntos discretos: finitos o infinitos numerables.

En el desarrollo del prototipo de nuestro Objeto de Aprendizaje (OA) basado en el cálculo de trayectorias de aprendizaje utilizando Answer Set Planning nuestro tema será la enseñanza de esta área enfocándonos en sus conceptos básicos.

Para determinar la creación de nuestro OA fue necesario determinar una posible causa de algún problema de aprendizaje directamente relacionado con la materia de Matemáticas discretas y proponer una solución. Uno de los principales problemas es que los alumnos presentan dificultades para entender conceptos matemáticos abstractos, es debido a que son un tanto difíciles de entender y requieren un alto nivel de comprensión lectora y mental.

Algunas de las estrategias que usan los alumnos es memorizar muchas veces el concepto matemático, así como el proceso de desarrollo de algún problema en específico.

Debido a esto, el estudiante no es capaz de resolver y darle sentido a los diferentes problemas que se le presenten, ya que modificar alguna información, les resulta una tarea difícil y afecta de manera negativa todo su proceso de aprendizaje.

### 2.1.1 Principales objetivos de aprendizaje para el curso

- **Objetivo general:**

El estudiante repase ciertos conceptos de matemáticas discretas y así entienda y formalice una visión de las matemáticas discretas básicas y a partir de ello pueda construir mejores bases para matemáticas más complejas.

- **Objetivos en específico:**

Comprender de una manera más simple las definiciones expuestas en clase.

Aplicar conocimiento para el entendimiento y solución de algunos ejercicios.

Simplificar algunas definiciones y así el alumno entienda las propiedades de estas.

Tenemos clarificados nuestros objetivos generales y específicos del prototipo de nuestro OA, pero ¿Cuáles serán nuestros conceptos a enseñar, las estrategias y actividades? y ¿cómo estarán organizados en nuestro OA? Para ubicar estos elementos en la interfaz de nuestro OA debemos tener un diseño instruccional y una estructura lógica.

### 2.1.2 Desarrollo de los temas a enseñar

Las matemáticas discretas abarcan diferentes áreas, pero para el desarrollo del prototipo de nuestro Objeto de Aprendizaje nos enfocaremos únicamente en el tema de Teoría de conjuntos.

La Teoría de conjuntos es una parte de las matemáticas que tiene un objeto de estudio propio; con métodos propios, con ciertas relaciones con otras teorías matemáticas, en particular, con todas las teorías matemáticas tradicionales y a partir de sus principios se mantiene la existencia, estructura y relaciones mutuas entre ellos. Es decir, que el resto de la matemática puede expresarse en términos de conjuntos [11].

Entonces que temas de esta área tomaremos como base para la enseñanza en nuestro Objeto de Aprendizaje. Los temas serán los siguientes:

Nombre del tema	Características a enseñar del tema	Forma de presentar el contenido
<b>Conjuntos</b>	<ul style="list-style-type: none"> <li>▪ ¿Qué es un Conjunto?</li> <li>▪ ¿Qué es un elemento del Conjunto?</li> <li>▪ Formas de describir a un Conjunto.</li> <li>▪ Ejemplos de cómo describir a un Conjunto.</li> </ul>	El tema se enseñará mediante un contenido audiovisual ya que facilita la comprensión para el alumno debido a que se sienten atraídos por las explicaciones gráficas y animaciones interactivas.
<b>Subconjuntos</b>	<ul style="list-style-type: none"> <li>• ¿Qué es un Subconjunto?</li> <li>• Diferencias entre un Subconjunto y un no-Subconjunto.</li> <li>• Forma de describir un Subconjunto y no-Subconjunto</li> </ul>	El tema se enseñará mediante un contenido audiovisual ya que facilita la comprensión para el alumno debido a que se sienten atraídos por las explicaciones gráficas y animaciones interactivas.

<p><b>Conjunto universal</b></p>	<ul style="list-style-type: none"> <li>• ¿Qué es un Conjunto universal?</li> <li>• Manera de denotar al Conjunto universal.</li> <li>• Ejemplos de Conjuntos universales.</li> </ul>	<p>El tema se enseñará mediante un contenido audiovisual ya que facilita la comprensión para el alumno debido a que se sienten atraídos por las explicaciones gráficas y animaciones interactivas.</p>
<p><b>Conjunto potencia</b></p>	<ul style="list-style-type: none"> <li>• ¿Qué es un Conjunto potencia?</li> <li>• Manera de denotar al Conjunto potencia.</li> <li>• Cardinalidad del Conjunto potencia</li> <li>• Ejemplos de un Conjunto potencia</li> </ul>	<p>El tema se enseñará mediante un contenido audiovisual ya que facilita la comprensión para el alumno debido a que se sienten atraídos por las explicaciones gráficas y animaciones interactivas.</p>

Una vez que identificamos estos temas y teniendo en claro cuáles son las características que se desea que los alumnos comprendan se crearan los medios por los cuales se enseñara este contenido, en este caso se eligió a los medios audiovisuales, pero ¿Por qué medios audiovisuales como estrategia de enseñanza para nuestro OA?

### **2.1.3 Medios audiovisuales como estrategia de enseñanza**

Las estrategias de enseñanza se definen como los procedimientos o recursos utilizados por los docentes para lograr aprendizajes significativos en los alumnos.

Los medios audiovisuales son considerados como una estrategia de aprendizaje de mucha importancia que se han visto favorecidas por los avances tecnológicos de hoy en día. Son una excelente herramienta debido a que facilitan y agilizan la comprensión e interpretación de ideas tanto a corto como a mediano o largo plazo. Estos medios se pueden integrar de diferentes elementos que van desde imágenes, sonido, texto, gráficos entre muchos otros, particularmente poseen un beneficio significativo en el proceso de enseñanza-aprendizaje,

atendiendo eficazmente a las necesidades de una gran diversidad de alumnos a través de los sentidos.

Algunas ventajas de los medios audiovisuales son [12]:

- Permiten presentar los temas o conceptos de un tema de una manera objetiva, clara y accesible.
- El aprendizaje es más fluido y eficaz.
- Estimulan el interés y la motivación del alumno.
- Acercan a los estudiantes a la realidad y a darle significado a lo aprendido.
- Permiten facilitar la comunicación. Complementan las técnicas didácticas y economizan tiempo de aprendizaje.

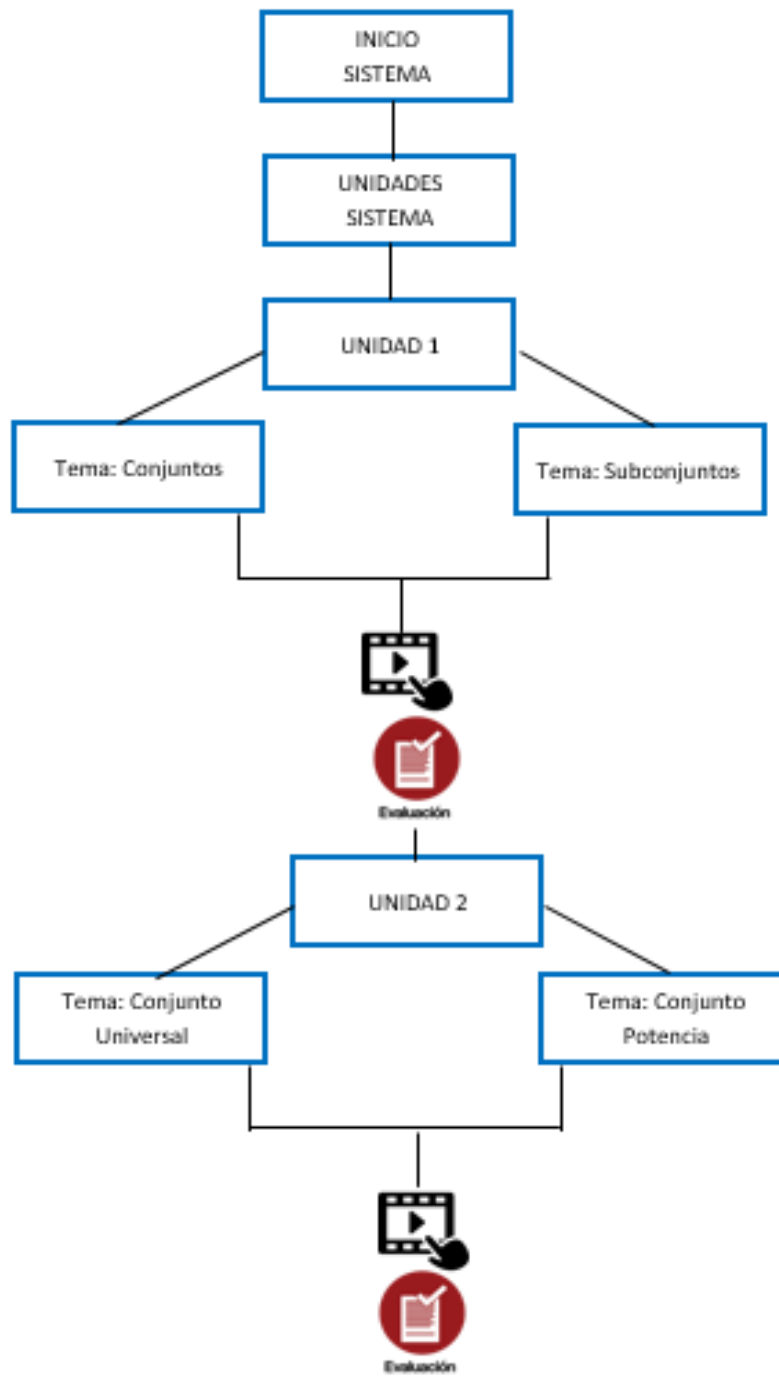
Usar medios audiovisuales nos proporcionarán una autonomía en cuanto a la creatividad de la creación de estos contenidos audiovisuales ya que se pretende que con la ayuda de un solo video bien estructurado con animaciones interactivas y gráficos muy explícitos estos permitan observar y visualizar de manera más sencilla las definiciones, procesos y diferentes elementos que conformen el tema a enseñar en nuestro OA.

#### **2.1.4 Estructura general del OA**

Nuestro Objeto de Aprendizaje se compone de manera global con dos Unidades y cada una de ellas contiene dos subtemas. La Unidad 1 se compone de los subtemas: Conjuntos y Subconjuntos, mientras que la Unidad 2 de los subtemas: Conjunto Universal y Conjunto Potencia [2.1.2 ]. Cada uno de estos subtemas será enseñado mediante un video y finalmente cuando se enseñen ambos subtemas de la unidad se procederá a ser una pequeña evaluación general de la unidad.

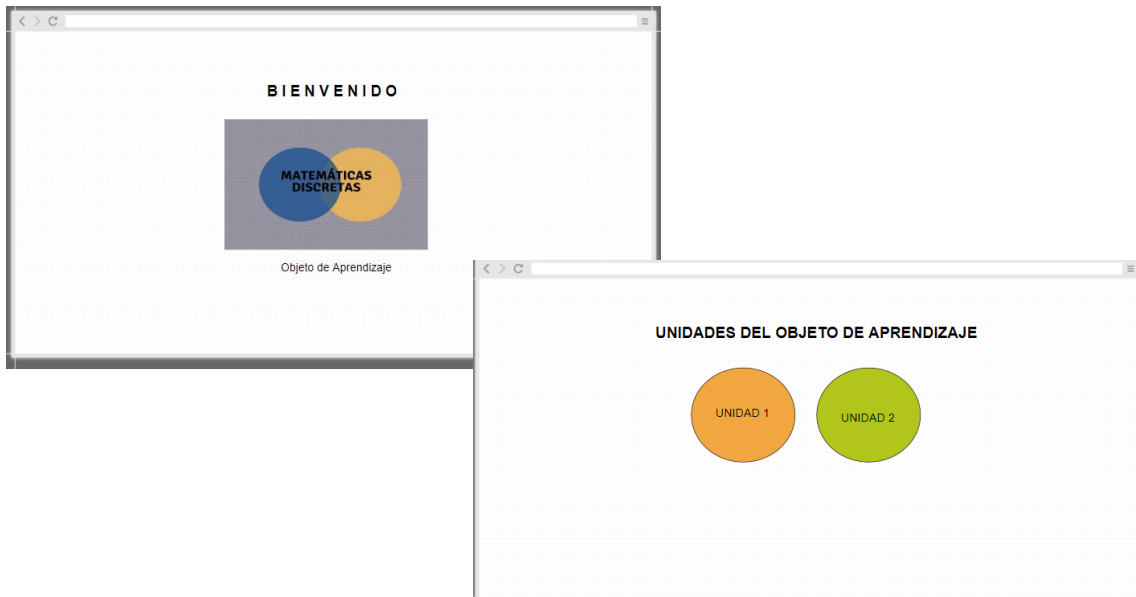
Si la unidad es aprobada, se procederá a enseñar la siguiente unidad, en caso de reprobar la unidad y dependiendo de los resultados se volverá a explicar y evaluar únicamente la sección donde fallo o bien toda la unidad hasta que el alumno la apruebe satisfactoriamente.

A continuación, se muestra un diagrama que presenta de manera general la estructura en la que se basa nuestro Objeto de Aprendizaje y que ha sido explicada hace un momento.



## 2.1.5 Prototipos de interfaces

### Inicio



## Unidades y subtemas



## Evaluaciones de los subtemas

A screenshot of an evaluation form titled "EVALUACIONES". It contains three questions, each with three radio button options:

**1.- Pregunta 1 relacionada con el tema a evaluar**  
 Respuesta a       Respuesta b       Respuesta c

**2.- Pregunta 2 relacionada con el tema a evaluar**  
 Respuesta a       Respuesta b       Respuesta c

**3.- Pregunta 3 relacionada con el tema a evaluar**  
 Respuesta a       Respuesta b       Respuesta c

At the bottom center is a button labeled "Verificar".

## 2.2 HERRAMIENTAS DE DESARROLLO

- Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados [13].



El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis proviene en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

Este lenguaje Java se creó con cinco objetivos principales [13]:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Hoy en día Java ha experimentado numerosos cambios desde la primera versión, JDK 1.0, así como un enorme incremento en el número de clases y paquetes que componen la biblioteca estándar.

- **Eclipse plataforma de desarrollo**

El lenguaje Java cuenta con diferentes plataformas de desarrollo una de ellas es Eclipse, Eclipse fue desarrollado originalmente por IMB como sucesor de su familia de herramientas para VisualAge, hoy en día es desarrollado por la fundación Eclipse, una organización independiente que promueve a la comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse es uno de los entornos de desarrollo integrado (IDE) más utilizados hoy en día. Los IDE son software que ayudan a crear aplicaciones de manera más cómoda, con su ayuda podemos escribir código, compilar, ejecutar y usar múltiples herramientas desde el propio entorno para poder desarrollar programas en el lenguaje Java.



Este software (Eclipse) nos ayudara para la creación de nuestro Objeto de Aprendizaje y para la creación de interfaces graficas en Java utilizaremos la ayuda del plug-in WindowsBuilder que puede ser instalado [14] en Eclipse y que nos facilitara crear una interfaz con swing arrastrando únicamente los componentes que creamos necesarios esta.

- **Integración JDLV**

Como el mismo título de esta tesis lo remarca tenemos como principal meta el Desarrollar un Objeto de Aprendizaje (OA) basado en el cálculo de trayectorias de aprendizaje utilizando Answer Set Planning, esto únicamente se logrará con la unión de Java y DLV y es aquí donde JDLV entra ya que es un nuevo marco de programación que combina justamente a DLV con programación en Java.



*“El marco se basa en JASP, un lenguaje híbrido que admite de manera transparente una interacción bilateral entre Answer Set Programming y Java. Un ingrediente clave de JASP es la asignación entre colecciones de objetos Java y hechos ASP. Los objetos Java se asignan a hechos lógicos y viceversa mediante la adopción de una estrategia de mapeo estructural similar a la empleada por las herramientas ORM para recuperar, guardar objetos persistentes en bases de datos relacionales. En particular, las especificaciones JASP cumplen con el estándar JPA para el Mapeo relacional de objetos para adaptarse perfectamente a las tecnologías de aplicación empresarial ampliamente adoptadas [15]”.*

El marco JDLV está diseñado para DLV e incluye:

- JDLV un compilador de JASP a Java
- JDLV-plugin un complemento para la plataforma Eclipse TM ([www.eclipse.org](http://www.eclipse.org)).

Teniendo en cuenta estos aspectos, la infraestructura de nuestro entorno de desarrollo tanto del software y plugins es la siguiente:

- Eclipse jee mars 2 en 32 bits.
- WindowsBuilder
- DLVwrapper-v4.2.jar
- jdlv\_executor.jar

# CAPITULO 3:

## DESARROLLO E IMPLEMENTACION



## **3.1 PERSPECTIVA DEL OBJETO DE APRENDIZAJE**

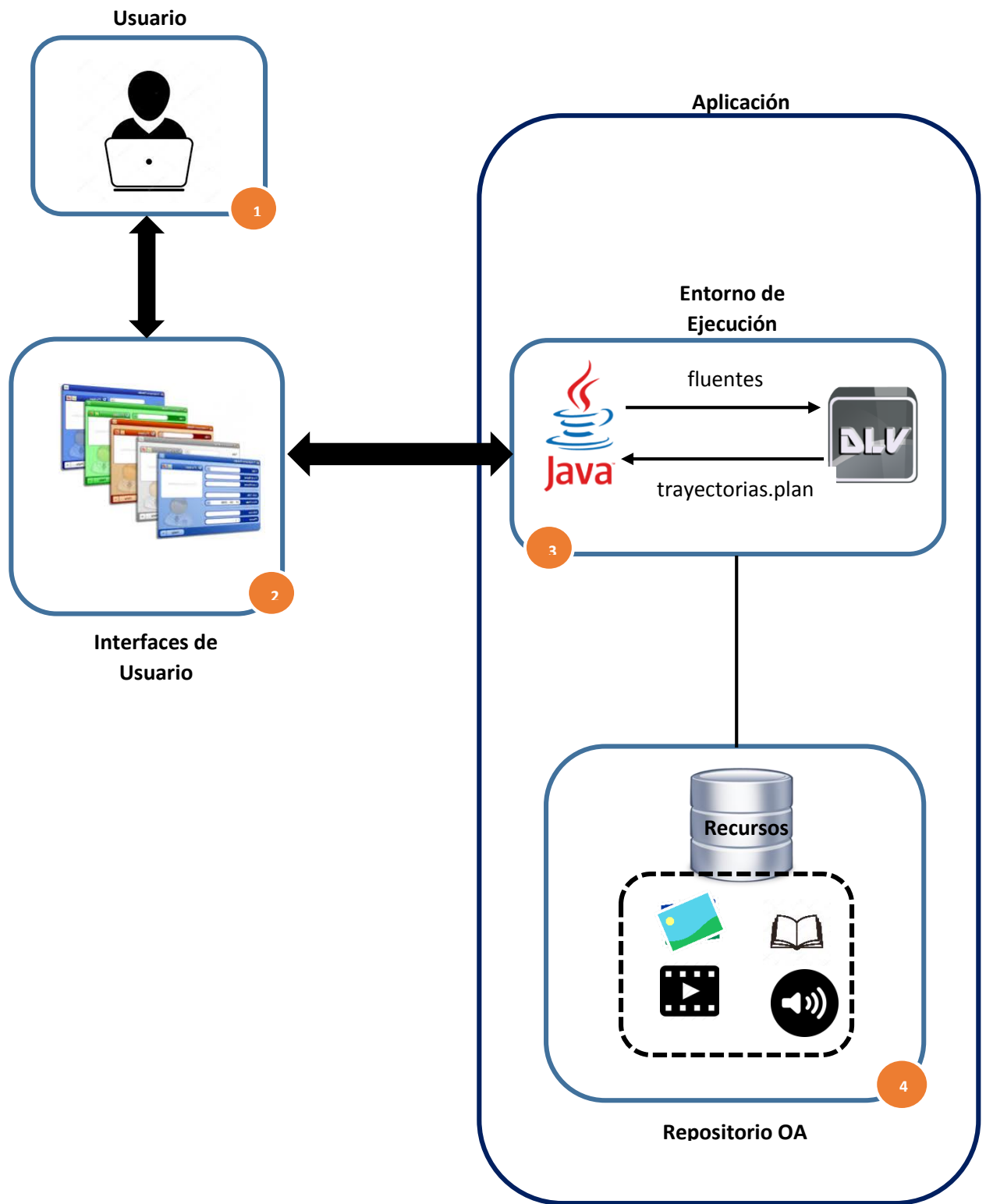
Esta tesis va dirigida a Desarrollar un prototipo de un Objeto de Aprendizaje (OA) basado en el cálculo de trayectorias de aprendizaje utilizando Answer Set Planning. Es por eso que en los capítulos anteriores se analizaron primero todos los Fundamentos Teóricos necesarios para el desarrollo de un Objeto de Aprendizaje, desde las características que lo conforman y metodologías de diseño, de igual manera se hizo una introducción a lo que es Answer Set Planning, nuestra principal herramienta de apoyo para el funcionamiento del prototipo de OA a implementar. Particularmente en el diseño del contenido educativo que se impartirá en el Objeto de Aprendizaje ya se ha explicado la estructura general, así como los temas y/o definiciones de Matemáticas Discretas que utilizaremos.

El Objeto de Aprendizaje (OA) será un sistema que permita en un futuro ser utilizado como herramienta de apoyo a estudiantes de la materia de Matemáticas discretas, el desarrollo de este OA es solamente una prueba piloto de un posible recurso educativo digital el cual debe funcionar utilizando la técnica de inteligencia artificial Answer Set Planning.

### **3.1.1 Arquitectura del sistema**

La arquitectura de un sistema indica la estructura, funcionamiento e interacción entre las partes del software, en este caso daremos una vista general de esta estructura y la interacción de Java y DLV.

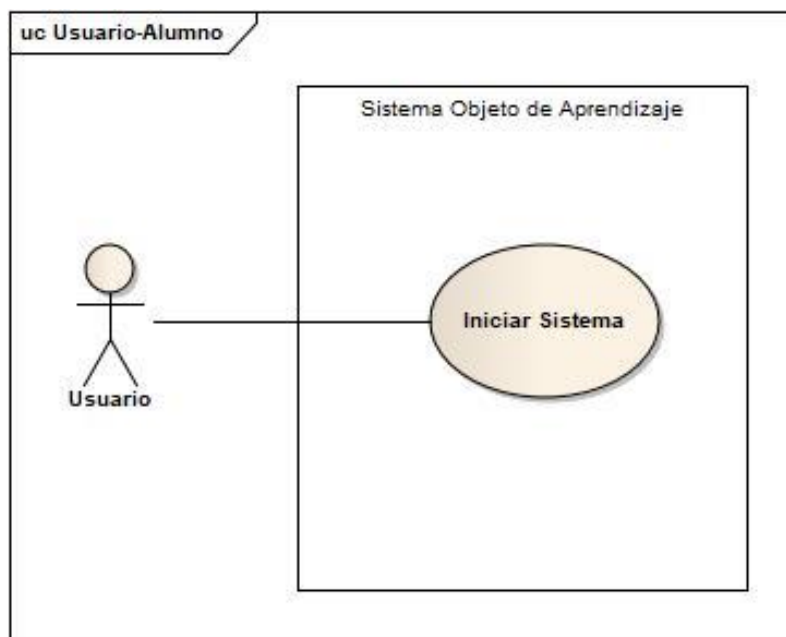
- 1)** El usuario tendrá una interacción directa únicamente con las interfaces creadas en Java.
- 2)** Las interfaces serán una vista general del sistema que estarán funcionando con la unión e integración de Java y DLV.
- 3)** La integración de Java y DLV funcionara de tal manera en que Java enviara fluentes a DLV y este generara planes, los cuales determinaran el funcionamiento e interacción del usuario y las interfaces.
- 4)** Son los recursos que se utilizaran en el OA para la enseñanza de los temas que van desde objetos multimedia y objetos de información (videos, imágenes, textos, entre otros).



## 3.2 FUNCIONALIDADES DEL SISTEMA

Siguiendo la documentación de UML (Lenguaje Unificado de Modelado) se identificó primero a los actores del sistema, así como las funciones que podrá hacer en el mismo, teniendo esto en cuenta debemos contar con un diagrama de casos de usos para así poder más adelante utilizar diagramas de secuencia y poder explicar a detalle la funcionalidad entre Java y DLV.

El diagrama de casos de uso de nuestro OA quedo de la siguiente manera:



En este caso tenemos a un único actor que es el Usuario, debido a que el Objeto de Aprendizaje no contara con niveles de funcionalidad es decir diferentes tipos de actores que tengan privilegios dentro del sistema que les permitan hacer acciones diferentes a otros actores del sistema.

La única funcionalidad que podrá hacer nuestro actor Usuario es la de Iniciar Sistema, es decir ingresar al OA y seguidamente de hacer esta función se desencadenara la interacción entre Java y DLV y serán estos los que determinen las trayectorias que seguirá el usuario dentro del sistema.

### 3.2.1 Funcionalidad Java y DLV

Utilizando la integración de DLV y Java, se presentará al OA como una interacción amigable con el usuario, definida con Answer Set Planning que calcula planes o bien trayectorias a partir de un estado final para llegar a una meta.

Para detallar toda esta interacción tenemos el apoyo de los diagramas de secuencia UML que son usados para modelar la comunicación de objetos y que mensajes dispararan esas comunicaciones.

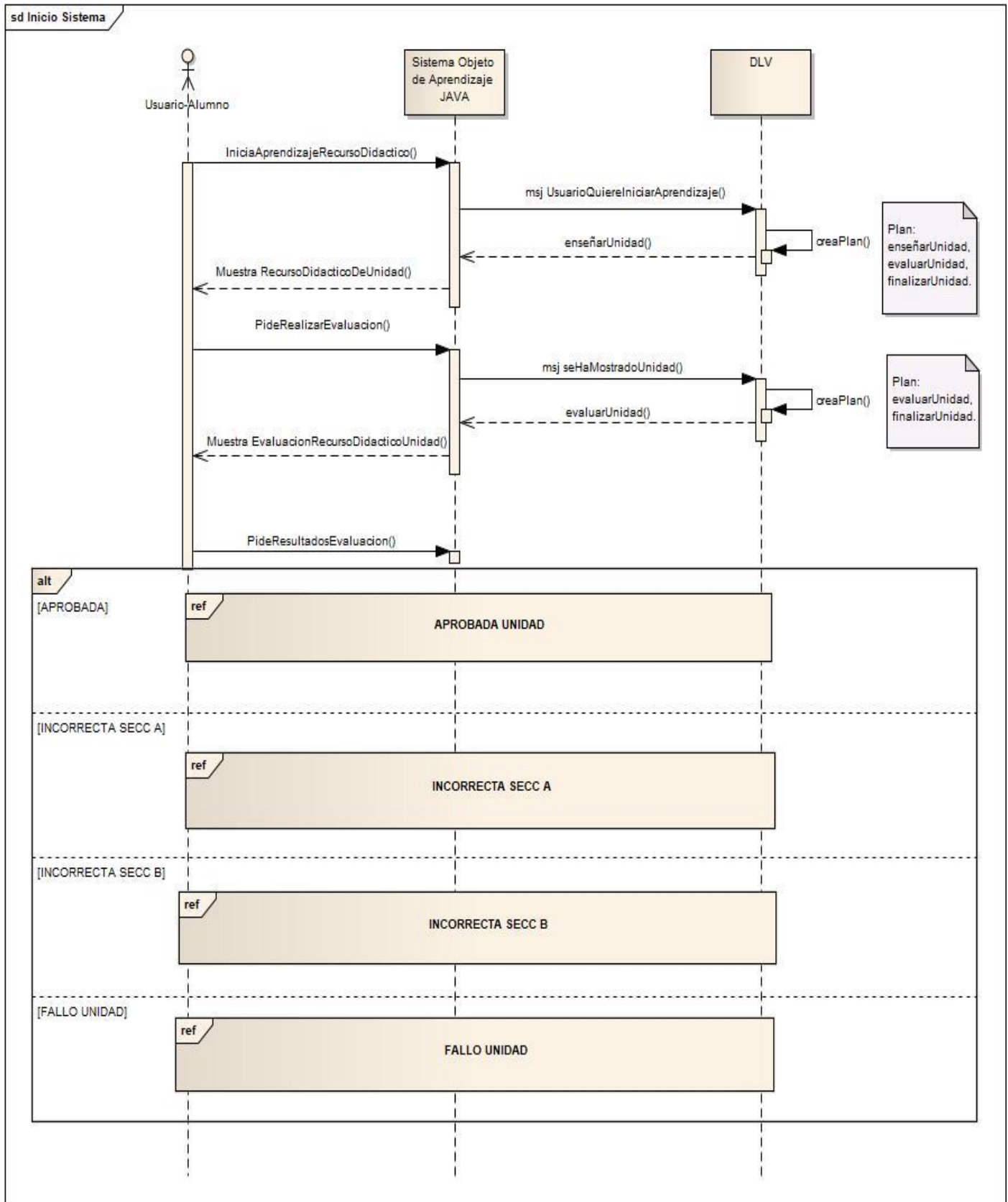
El siguiente diagrama de secuencia muestra las primeras interacciones del usuario con la interfaz, cuando el usuario ingrese al OA y pida iniciar el aprendizaje del recurso didáctico, Java se encargará de hacer la interacción con DLV y así este genere planes que dicten la siguiente interacción del usuario con la interfaz de Java.

Explicación de la situación del diagrama de secuencia:

**Tabla 1: Situación nominal Diagrama de Secuencia: Iniciar Sistema**

Título del diagrama: Iniciar Sistema	
Situación nominal Diagrama de Secuencia	
1.- Inicia aprendizaje del recurso didáctico	2. – Java envía mensaje a DLV que el usuario quiere iniciar aprendizaje. 3. – DLV inicia y crea el Plan: enseñarUnidad, evaluarUnidad, finalizarUnidad. 4.- DLV le dice a Java que enseñe la unidad, esto de acuerdo al estado 0 de su plan. 5.-Java Le muestra el recurso didáctico de la unidad al usuario.
6. – Usuario ve la interfaz donde se le muestra el recurso didáctico de la unidad. 7.- Pide realizar evaluación del recurso didáctico que se le acaba de mostrar.	8.- Java envía mensaje a DLV que el recurso didáctico ha sido mostrado. 9.- DLV crea Plan: evaluarUnidad, finalizarUnidad. 10.- DLV le dice a Java que evalúe la unidad, esto de acuerdo al estado 0 de su plan. 11.- Java le muestra la evaluación del recurso didáctico de la unidad al usuario.
12.-Usuario ve la interfaz donde se le muestra la evaluación del recurso didáctico de la unidad. 13.- Pide resultados de su evaluación una vez que la ha contestado.	14.- Java evaluara las respuestas de la evaluación, las cuales pueden ser: Aprobada Unidad, Incorrecta Sección A, Incorrecta Sección B o bien haber fallado toda la unidad.

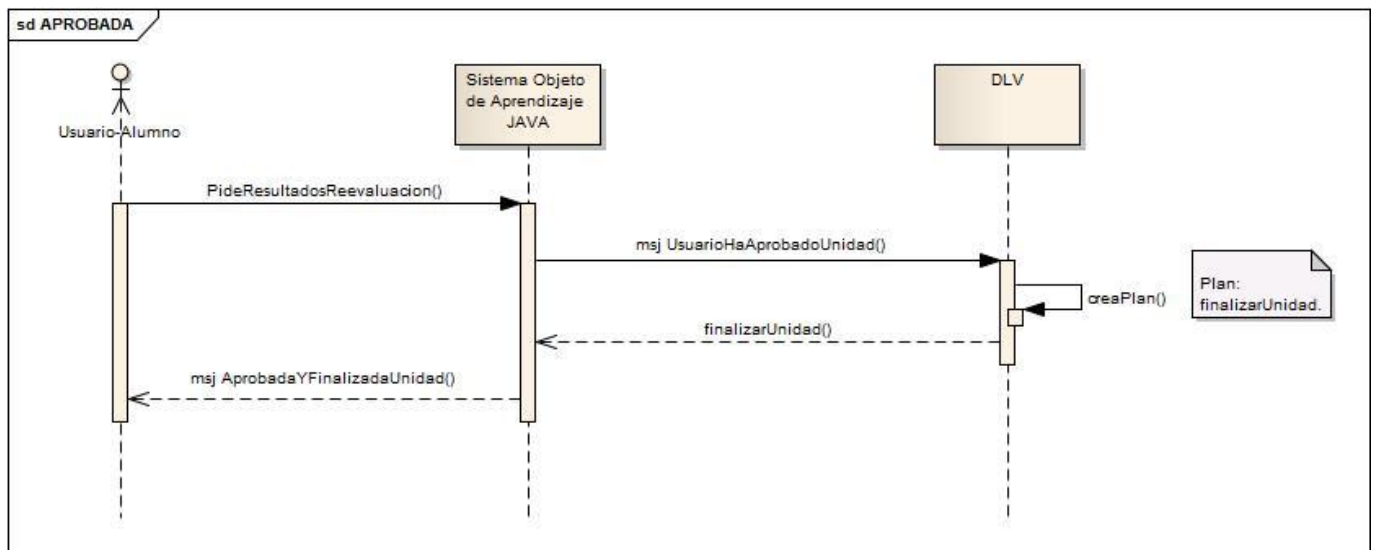
Diagrama de Secuencia: Inicio Sistema.



**Tabla 2: Situación nominal Diagrama de Secuencia: Aprobada Unidad**

Título del diagrama: <b>Aprobada Unidad</b>	
Situación nominal Diagrama de Secuencia	
<p><b>1.-</b> Pide resultados de su evaluación una vez que la ha contestado.</p>	<p><b>2.-</b> Java evalúa las respuestas y en este caso el Usuario ha aprobado la unidad, sabiendo esto java le manda un mensaje a DLV con estos resultados.</p> <p><b>3.-</b> DLV crea Plan: finalizarUnidad.</p> <p><b>4.-</b> DLV le dice a Java que finalice la unidad ya que el usuario la ha aprobado correctamente, esto de acuerdo al estado 0 de su plan.</p> <p><b>5.-</b> Java le muestra un mensaje al usuario con sus resultados e indicándole que su aprendizaje ha finalizado.</p>
<p><b>6. –</b> Usuario ve mensaje que ha aprobado la unidad y por lo tanto finalizado su aprendizaje.</p>	

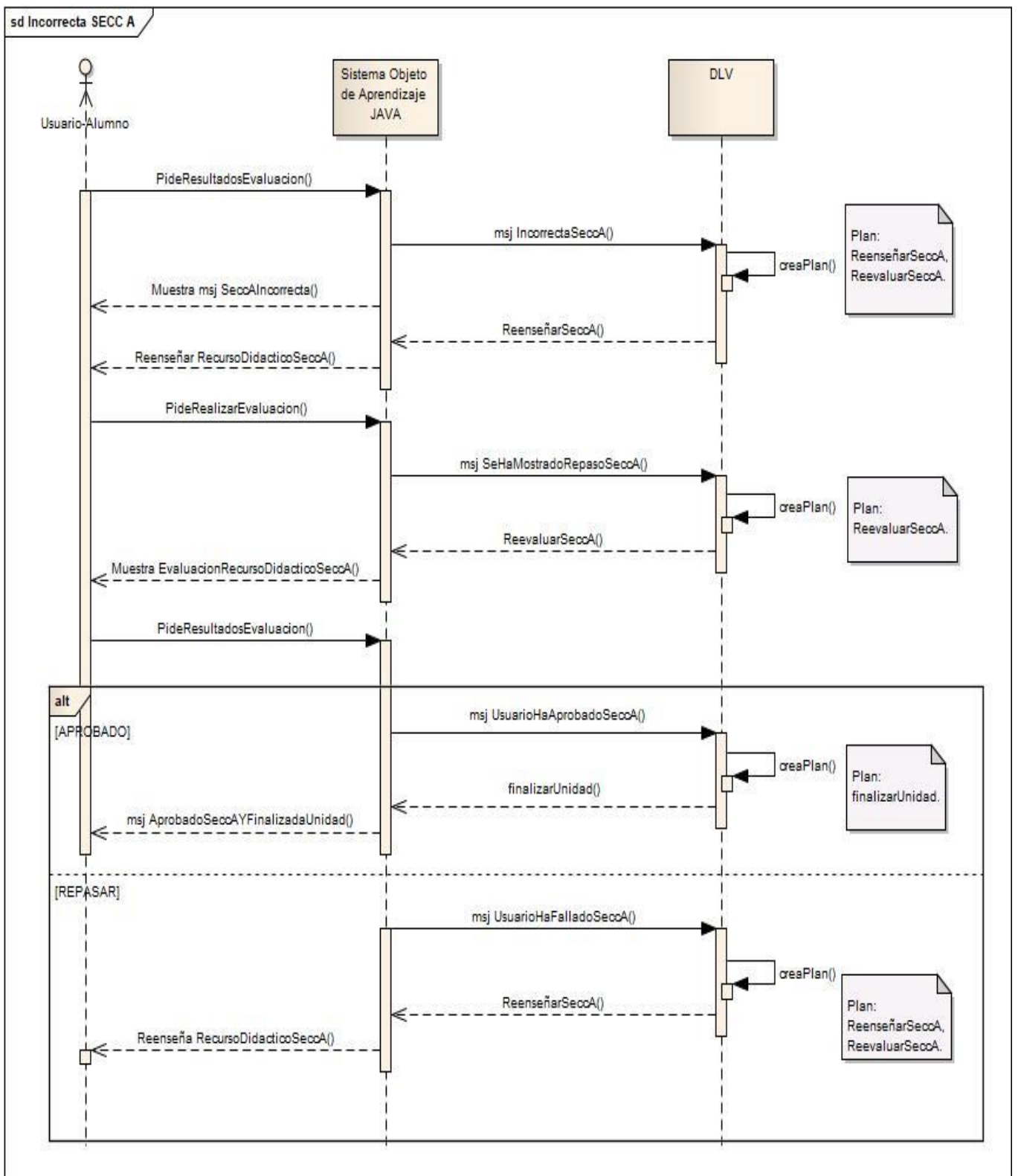
Diagrama de Secuencia: Aprobada Unidad.



**Tabla 3: Situación nominal Diagrama de Secuencia: Incorrecta Sección A**

Título del diagrama Incorrecta Sección A	
Situación nominal Diagrama de Secuencia	
<p><b>1.-</b> Pide resultados de su evaluación una vez que la ha contestado.</p>	<p><b>2.-</b> Java evalúa las respuestas y en este caso el Usuario ha tenido incorrecta la sección A de la unidad, sabiendo esto java le manda un mensaje a DLV con estos resultados.</p> <p><b>3.-</b> DLV crea Plan: ReenseñarSeccA, ReevaluarSeccA.</p> <p><b>4.-</b> DLV le dice a Java que reenseñe la sección A, esto de acuerdo al estado 0 de su plan.</p> <p><b>5.-</b> Java le muestra el recurso didáctico al usuario de la sección A que se va a reenseñar.</p>
<p><b>6. –</b> Usuario ve la interfaz donde se le muestra el recurso didáctico de la sección A de la unidad.</p> <p><b>7.-</b> Pide realizar evaluación del recurso didáctico de la sección A.</p>	<p><b>8.-</b> Java evaluara las respuestas correspondientes a la sección A, y generara los siguientes encadenamientos alternativos.</p>
Encadenamientos Alternativos	
<b>A1: Aprobada sección A de la unidad</b> El encadenamiento A1 inicia en el punto nominal 8.	
<p><b>12. –</b> Usuario ve mensaje que ha aprobado la sección A y por lo tanto finalizado su aprendizaje.</p>	<p><b>8.-</b> Los resultados del Usuario son satisfactorios es decir, ha aprobado la sección A de la unidad y envía un mensaje a DLV de los resultados.</p> <p><b>9.-</b> DLV crea Plan: finalizarUnidad.</p> <p><b>10.-</b> DLV le dice a Java que finalice la unidad ya que el usuario la ha aprobado correctamente, esto de acuerdo al estado 0 de su plan.</p> <p><b>11.-</b> Java le muestra un mensaje al usuario con sus resultados e indicándole que su aprendizaje ha finalizado.</p>
<b>A2: Incorrecta sección A de la unidad</b> El encadenamiento A2 inicia en el punto nominal 8.	
	<p><b>8.-</b> Los resultados del Usuario no son satisfactorios es decir, ha fallado la sección A de la unidad y envía un mensaje a DLV de los resultados.</p> <p><b>9.-</b> La situación nominal se retorna al punto 3.</p>

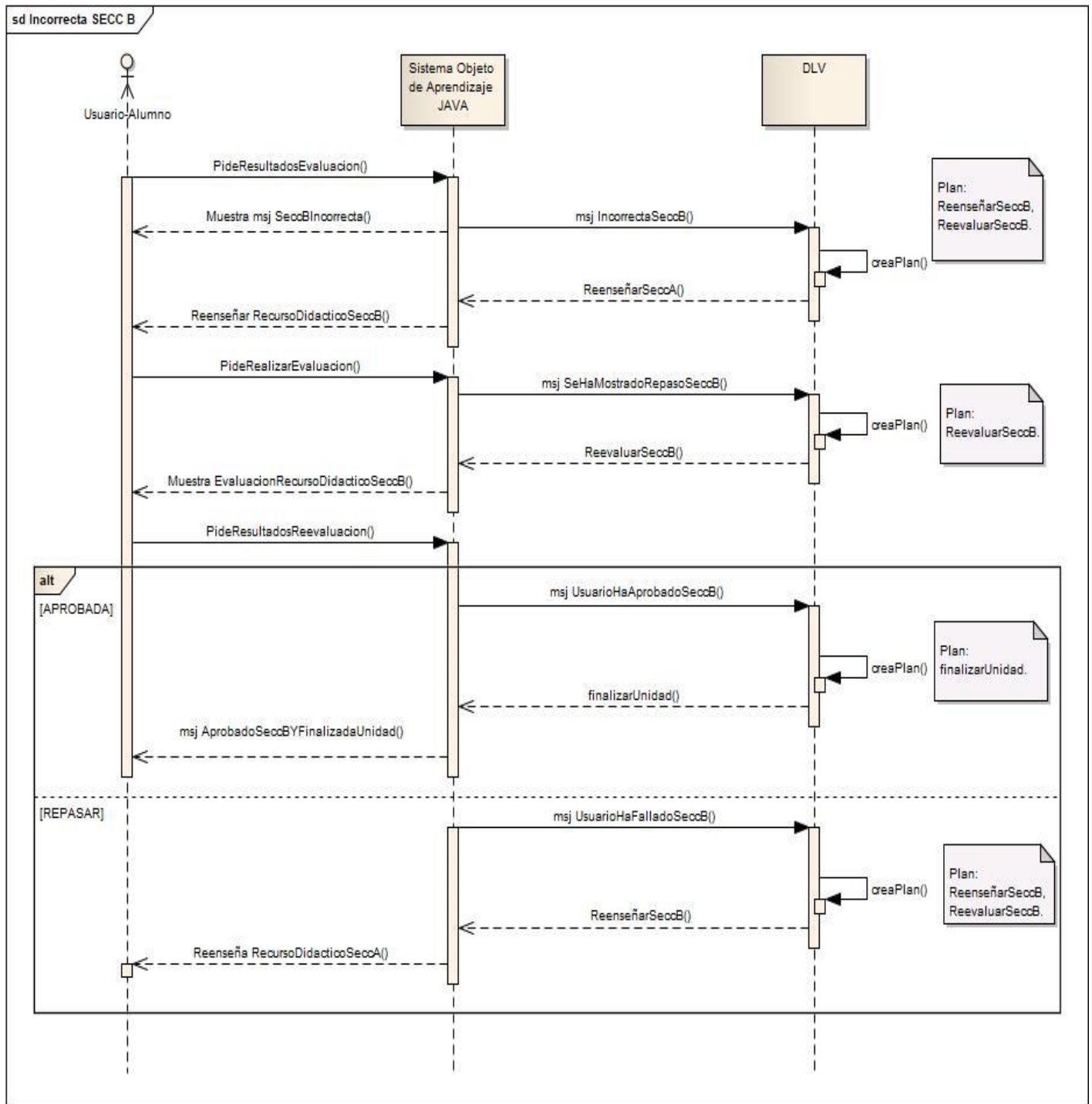
Diagrama de Secuencia: Incorrecta sección A.



**Tabla 4: Situación nominal Diagrama de Secuencia: Incorrecta Sección B**

Título del diagrama: Incorrecta Sección B	
Situación nominal Diagrama de Secuencia	
<p><b>1.-</b> Pide resultados de su evaluación una vez que la ha contestado.</p>	<p><b>2.-</b> Java evalúa las respuestas y en este caso el Usuario ha tenido incorrecta la sección B de la unidad, sabiendo esto java le manda un mensaje a DLV con estos resultados.</p> <p><b>3.-</b> DLV crea Plan: ReenseñarSeccB, ReevaluarSeccB.</p> <p><b>4.-</b> DLV le dice a Java que reenseñe la sección B, esto de acuerdo al estado 0 de su plan.</p> <p><b>5.-</b> Java le muestra el recurso didáctico al usuario de la sección B que se va a reenseñar.</p>
<p><b>6. –</b> Usuario ve la interfaz donde se le muestra el recurso didáctico de la sección A de la unidad.</p> <p><b>7.-</b> Pide realizar evaluación del recurso didáctico de la sección B.</p>	<p><b>8.-</b> Java evaluara las respuestas correspondientes a la sección B, y generara los siguientes encadenamientos alternativos.</p>
Encadenamientos Alternativos	
<b>A1: Aprobada sección B de la unidad</b> El encadenamiento A1 inicia en el punto nominal 8.	
<p><b>12. –</b> Usuario ve mensaje que ha aprobado la sección B y por lo tanto finalizado su aprendizaje.</p>	<p><b>8.-</b> Los resultados del Usuario son satisfactorios, es decir, ha aprobado la sección B de la unidad y envía un mensaje a DLV de los resultados.</p> <p><b>9.-</b> DLV crea Plan: finalizarUnidad.</p> <p><b>10.-</b> DLV le dice a Java que finalice la unidad ya que el usuario la ha aprobado correctamente, esto de acuerdo al estado 0 de su plan.</p> <p><b>11.-</b> Java le muestra un mensaje al usuario con sus resultados e indicándole que su aprendizaje ha finalizado.</p>
<b>A2: Incorrecta sección B de la unidad</b> El encadenamiento A2 inicia en el punto nominal 8.	
	<p><b>8.-</b> Los resultados del Usuario no son satisfactorios, es decir, ha fallado la sección B de la unidad y envía un mensaje a DLV de los resultados.</p> <p><b>9.-</b> La situación nominal se retorna al punto 3.</p>

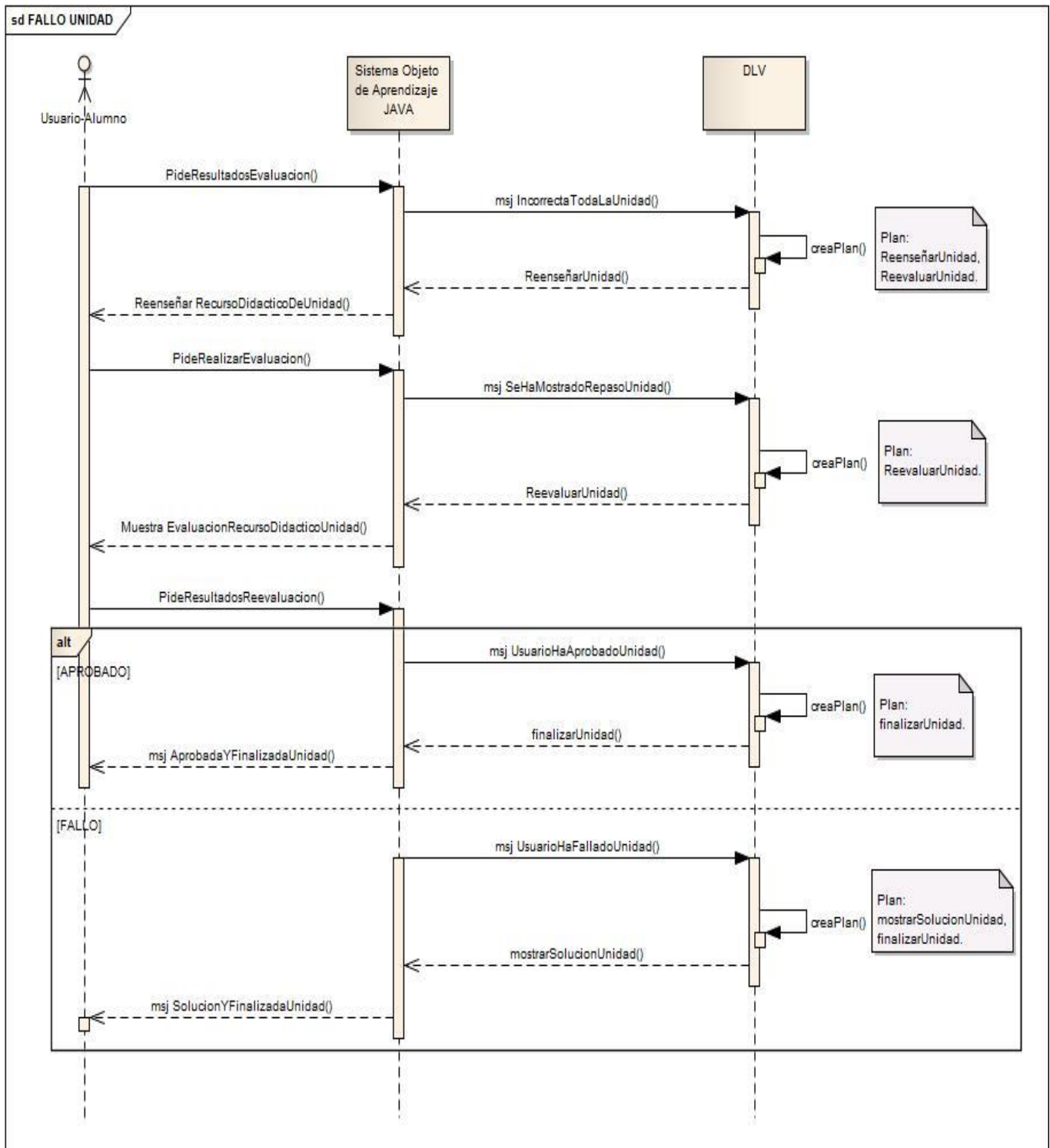
## Diagrama de Secuencia: Incorrecta sección B.



**Tabla 5: Situación nominal Diagrama de Secuencia: Fallo Unidad**

Título del diagrama: Fallo Unidad	
Situación nominal Diagrama de Secuencia	
<p><b>1.-</b> Pide resultados de su evaluación una vez que la ha contestado.</p>	<p><b>2.-</b> Java evalúa las respuestas y en este caso el Usuario ha fallado toda de la unidad, sabiendo esto java le manda un mensaje a DLV con estos resultados.</p> <p><b>3.-</b> DLV crea Plan: ReenseñarUnidad, ReevaluarUnidad.</p> <p><b>4.-</b> DLV le dice a Java que reenseñe toda la unidad, esto de acuerdo al estado 0 de su plan.</p> <p><b>5.-</b>Java le muestra el recurso didáctico al usuario de toda la unidad qué se va a reenseñar.</p>
<p><b>6. –</b> Usuario ve la interfaz donde se le muestra el recurso didáctico de toda la unidad.</p> <p><b>7.-</b> Pide realizar evaluación del recurso didáctico de la unidad.</p>	<p><b>8.-</b> Java evaluara las respuestas correspondientes de toda la unidad, y generara los siguientes encadenamientos alternativos.</p>
Encadenamientos Alternativos	
<b>A1: Aprobada Unidad</b> El encadenamiento A1 inicia en el punto nominal 8.	
<p><b>12. –</b> Usuario ve mensaje que ha aprobado la sección B y por lo tanto finalizado su aprendizaje.</p>	<p><b>8.-</b> Los resultados del Usuario son satisfactorios, es decir, ha aprobado la unidad y envía un mensaje a DLV de los resultados.</p> <p><b>9.-</b> DLV crea Plan: finalizarUnidad.</p> <p><b>10.-</b> DLV le dice a Java que finalice la unidad ya que el usuario la ha aprobado correctamente, esto de acuerdo al estado 0 de su plan.</p> <p><b>11.-</b> Java le muestra un mensaje al usuario con sus resultados e indicándole que su aprendizaje ha finalizado.</p>
<b>A2: Fallo Unidad</b> El encadenamiento A2 inicia en el punto nominal 8.	
<p><b>12. –</b> Usuario ve mensaje de la solución de la unidad y la finalización de esta.</p>	<p><b>8.-</b>Los resultados del Usuario no son satisfactorios, es decir, ha fallado de nuevo la unidad y envía un mensaje a DLV de los resultados.</p> <p><b>9.-</b> DLV crea plan: mostrarSolucionUnidad, finalizarUnidad.</p> <p><b>10.-</b> DLV le dice a Java que muestre la solución y finalice la unidad.</p> <p><b>11.-</b> Java le muestra un mensaje al usuario con sus resultados e indicándole la finalización de la unidad.</p>

## Diagrama de Secuencia: Fallo Unidad.



### 3.2.2 Integración de DLV y Java

Una vez explicado la funcionalidad del DLV con Java daremos paso a explicar cómo se implementó e integró la unión entre ambos lenguajes para poder presentar el OA como una interfaz amigable con el usuario.

DLV es un lenguaje que generalmente corre mediante una consola proporcionada cuando uno descarga el software de DLV [15], posteriormente se crea un archivo con la extensión .plan en donde se describen los flujos, acciones y reglas que nos permitirán realizar una transición de estados. La principal meta de DLV será generar una planificación de acciones llamadas trayectorias o planes.

Para la unión de DLV en Java se creó la función correDlv() donde su principal función es ejecutar DLV en Java y ya no desde consola.

A continuación, se muestra un fragmento de esta función y el resultado de ejecutarla.

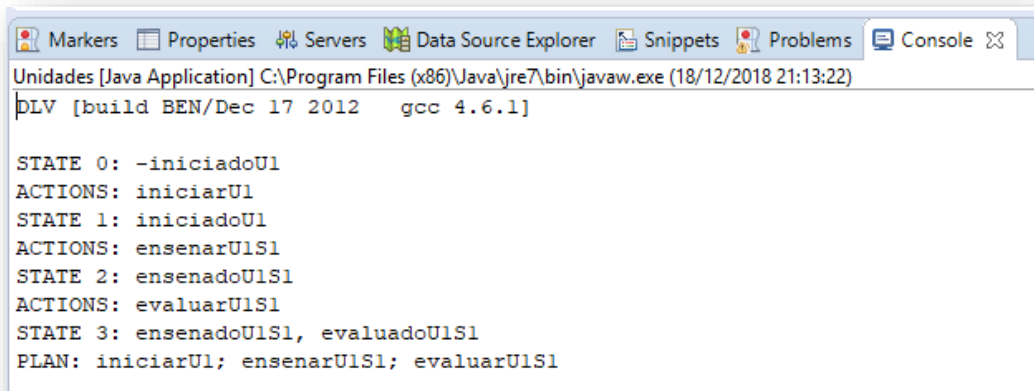
```
public void correDlv()
{
    try {
        // Determinar en qué SO estamos
        String so = System.getProperty("os.name");

        String comando;

        // Comando para Linux
        if (so.equals("Linux"))
            comando = "dlv entradaF.txt -nofinitecheck -filter=modelP <salida.txt";

        // Comando para Windows
        else
            comando = "cmd /c dlv trayectorias.plan trayectorias.dl -FPopt -N=3";

        // Ejecutamos el comando
        Process p = Runtime.getRuntime().exec(comando);
    }
}
```



```
Unidades [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (18/12/2018 21:13:22)
DLV [build BEN/Dec 17 2012 gcc 4.6.1]

STATE 0: -iniciadoU1
ACTIONS: iniciarU1
STATE 1: iniciadoU1
ACTIONS: enseñarU1S1
STATE 2: enseñadoU1S1
ACTIONS: evaluarU1S1
STATE 3: enseñadoU1S1, evaluadoU1S1
PLAN: iniciarU1; enseñarU1S1; evaluarU1S1
```

Como podemos notar ahora el archivo **trayectorias.plan** está siendo ejecutado en java de manera correcta, generando así un plan inicial que nos permitirá definir y seguir las trayectorias trazadas por DLV.

El resultado que se muestra en la imagen es guardado en un archivo txt con la siguiente función:

```
public void escribeFichero (String fichero)
{
    File f;
    f = new File(fichero);

    try{
        FileWriter w = new FileWriter(f);
        BufferedWriter bw = new BufferedWriter(w);
        PrintWriter wr = new PrintWriter(bw);
        wr.write(salidaF);

        wr.close();
        bw.close();
    }
    catch(IOException e){};
}
```

Una vez que DLV está corriendo de forma correcta y su resultado siendo guardado en un archivo externo para más adelante ser utilizado se creó una función llamada `BuscarReemplazar()` que nos permite hacer cambios al archivo **trayectorias.plan**, los cambios se harán exclusivamente en los estados y metas logrando así que cada cambio realizado logre en su conjunción nuevos planes a seguir para el funcionamiento del Objeto de Aprendizaje.

El siguiente es un fragmento de esta función, donde se puede observar cómo cambia el estado y la meta reemplazando estos por nueva información.

```

case 1:
String buscar = "-iniciadoU1. % estado 1";
String remplaza="iniciadoU1. % estado 2";
String buscar1 = "evaluadoU1S1 ? (3) % estado 1";
String remplazal="evaluadoU1S1 ? (2) % estado 2";

String newContent = content.replace(buscar,remplaza).replace(buscar1,
remplazal);

FileWriter writer = new FileWriter("trayectorias.plan");
writer.write(newContent);

writer.close();
break;

```

La ejecución de esta función genera los siguientes cambios dentro del archivo *trayectorias.plan*

```

151
152 initially: ANTES
153
154
155 -iniciadoU1. % estado 1
156
157
158
159 noConcurrency.
160
161
162 goal:
163 evaluadoU1S1 ? (3) % estado 1
164
165
166
167

```

```

151
152 initially: DESPUES
153
154
155 iniciadoU1. % estado 2
156
157
158
159 noConcurrency.
160
161
162 goal:
163 evaluadoU1S1 ? (2) % estado 2
164
165
166
167

```

Estos nuevos cambios producen un nuevo plan, el cual es una nueva trayectoria que precisara el orden en que deben ser mostradas las interfaces o bien las actividades del objeto de aprendizaje.

Finalmente con la función BuscarPlan() utilizando el archivo txt que se creó con la función escribeFichero() se busca la primera acción generada por DLV como se muestra a continuación

```
public void BuscarPlan() {  
  
    scanner = new Scanner("iniciarU1");  
    scanner1 = new Scanner("enseñarU1S1");  
    scanner2 = new Scanner("evaluarU1S1");  
    scanner3 = new Scanner("reenseñarSecAU1S1");  
    scanner4 = new Scanner("reevaluarSecAU1S1");  
    scanner5 = new Scanner("reenseñarSecBU1S1");  
    scanner6 = new Scanner("reevaluarSecBU1S1");  
    scanner7 = new Scanner("reenseñarU1S1");  
    scanner8 = new Scanner("reevaluarU1S1");  
    scanner9 = new Scanner("resolverReposSecAU1S1");  
    scanner10 = new Scanner("resolverReposSecBU1S1");  
    scanner11 = new Scanner("resolverReposU1S1");  
    scanner12 = new Scanner("enseñarU1S2");  
  
}
```

Finalmente, dentro de la misma función una vez encontrada la primera acción se guarda en una variable y se compara con una nueva cadena, esto para poder determinar que sean iguales y seguir el trayecto correcto del Objeto de Aprendizaje como se percibe en el siguiente fragmento de código.

```
try {  
    final BufferedReader reader = new BufferedReader(new  
    FileReader("C:/Users/karen/workspace1/TesisSistema/plan.txt"));  
    String line = "";  
    String plan="ACTIONS: iniciarU1";  
    String plan1="ACTIONS: enseñarU1S1";  
    String plan2="ACTIONS: evaluarU1S1";  
    String plan3="ACTIONS: reenseñarSecAU1S1";  
    String plan4="ACTIONS: reevaluarSecAU1S1";  
    String plan5="ACTIONS: reenseñarSecBU1S1";  
    String plan6="ACTIONS: reevaluarSecBU1S1";  
    String plan7="ACTIONS: reenseñarU1S1";  
    String plan8="ACTIONS: reevaluarU1S1";  
    String plan9="ACTIONS: resolverReposSecAU1S1";  
    String plan10="ACTIONS: resolverReposSecBU1S1";  
    String plan11="ACTIONS: resolverReposU1S1";  
    String plan12="ACTIONS: enseñarU1S2";  
  
    while ((line = reader.readLine()) != null) {  
        if (line.indexOf(request) != -1) {  
            if (line.equals(plan))  
            {  
                InicioUnidad1 window = new InicioUnidad1();  
                window.frame.setVisible(true);  
                break;  
            }  
        }  
    }  
  
}
```

# **CAPITULO 4:**

## **RESULTADOS DEL**

### **OBJETO DE**

#### **APRENDIZAJE**



## 4.1 INTERACCIÓN AMIGABLE CON EL USUARIO

Inicialmente se mostrará la siguiente interfaz de Bienvenida, una simple interfaz en donde aún no se inicia la ejecución de DLV y Java. Únicamente servirá como presentación del Objeto de Aprendizaje.



La siguiente interfaz muestra las Unidades que contiene nuestro OA, finalmente es en esta interfaz es donde inicia nuestra conexión DLV y Java. En el botón INICIAR de la unidad 1 al dar click se ejecutarán las funciones que anteriormente fueron explicadas, mismas que nos permitirán calcular y seguir las trayectorias para llegar a un estado o meta final.



```

JButton btnNewButton = new JButton("INICIAR");
    btnNewButton.setForeground(Color.BLUE);
    btnNewButton.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e) {
            ejecuta g = new ejecuta();
            g.correDlv();
            g.escribeFichero("plan.txt");
            g.BuscarPlan();
            g.BuscarReemplazar(1);
            frame.dispose();
        }
    });

```

A continuación, se muestran todos los planes generados por DLV, así como las interfaces correspondientes a cada plan.

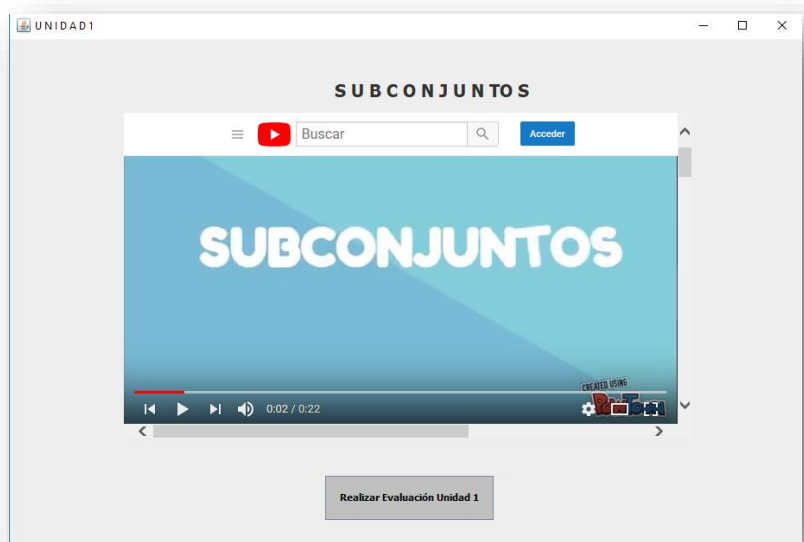
Plan

```
iniciarU1  
enseñarU1S1  
evaluarU1S1
```



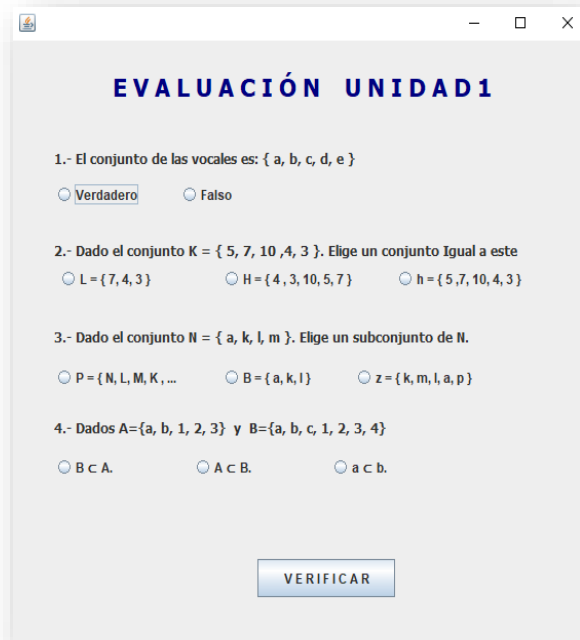
Plan

enseñarU1S1  
evaluarU1S1

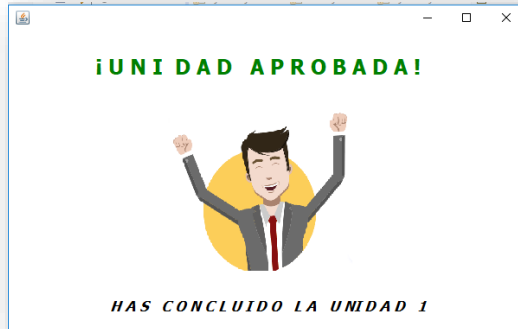


Plan

evaluarU1S1



Las trayectorias que tomará nuestro OA dependerá de los resultados obtenidos en la evaluación, las cuales serán las siguientes:



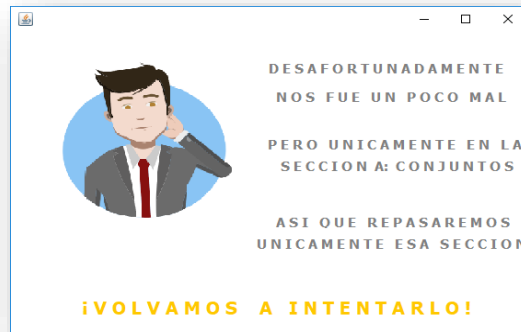
- **ÉXITO: Aprobó la Evaluación**

Plan

enseñarU1S2



● FALLO A: Fallo en sección Conjuntos.



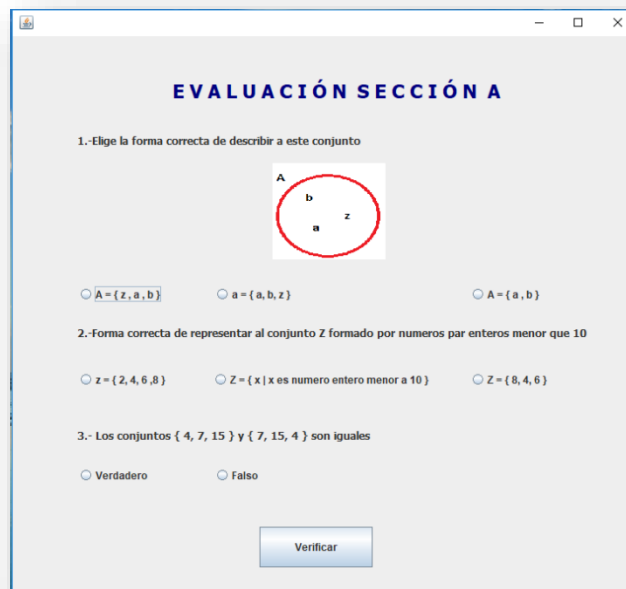
Plan

reenselarSecAU1S1  
reevaluarSecAU1S1



Plan

reevaluarSecAU1S1

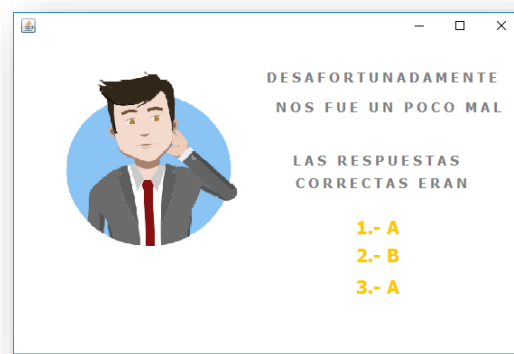


El usuario puede aprobar o bien fallar de nuevo la sección de conjuntos.

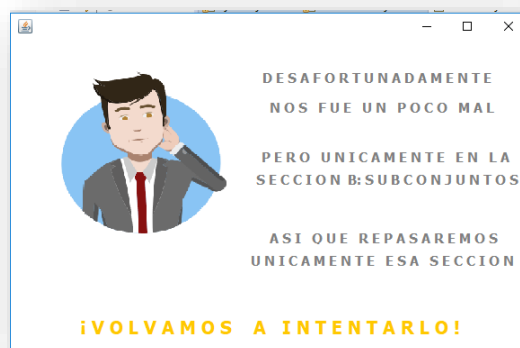


Plan

```
resolverReposSecAU1S1
```



- **FALLO B: Fallo en sección Subconjuntos**



Plan

reenselarSecBU1S1  
reevaluarSecBU1S1

Unidad 1 Sección B SubConjuntos

Si cada elemento de A es también elemento de B, se dice que A es un subconjunto de



Reabzar Evaluación

Plan

reevaluarSecBU1S1

EVALUACIÓN SECCIÓN B

1.- Dado el Conjunto  $K = \{ 1, 5, 8, 0, 10 \}$ , Elige un subconjunto de este conjunto:

$A = \{ 1, 0, 5, 8 \}$         $A = \{ 10, 0, 7, 9 \}$         $A = \{ 1, 5, 8, 0, 10, 11 \}$

2.- Dados  $A = \{ a, b \}$  y  $B = \{ 1, a, b \}$ , se puede decir que  $A \subseteq B$

Verdadero       Falso

3.- Si todos los elementos del conjunto F pertenecen a otro conjunto B. Como se describe esto?

$F \notin B$         $F \subset B$         $B \subset F$

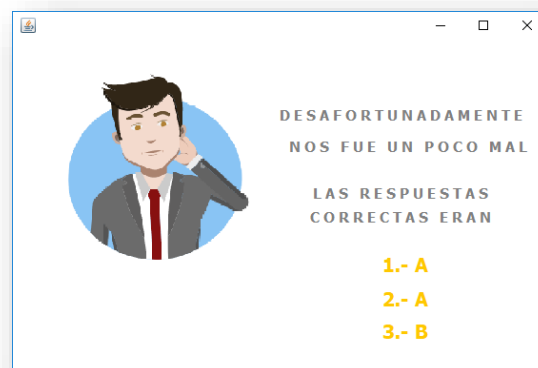
Verificar

El usuario puede aprobar o bien fallar de nuevo la sección de Subconjuntos.

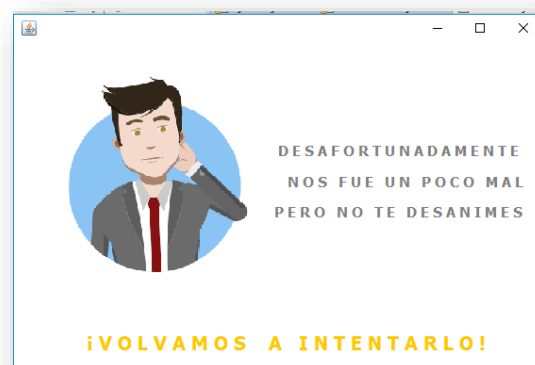


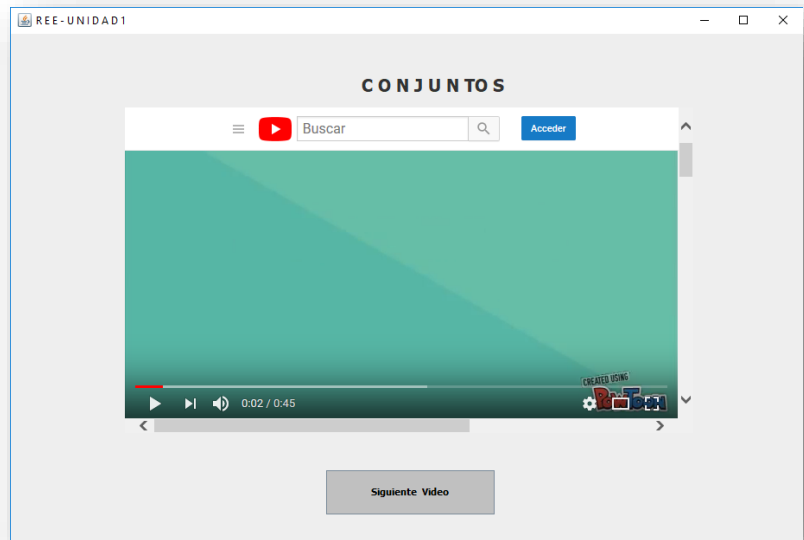
Plan

```
resolverReposSecBU1S1
```



- **FALLO UNIDAD:** Fallo en toda la Unidad 1.

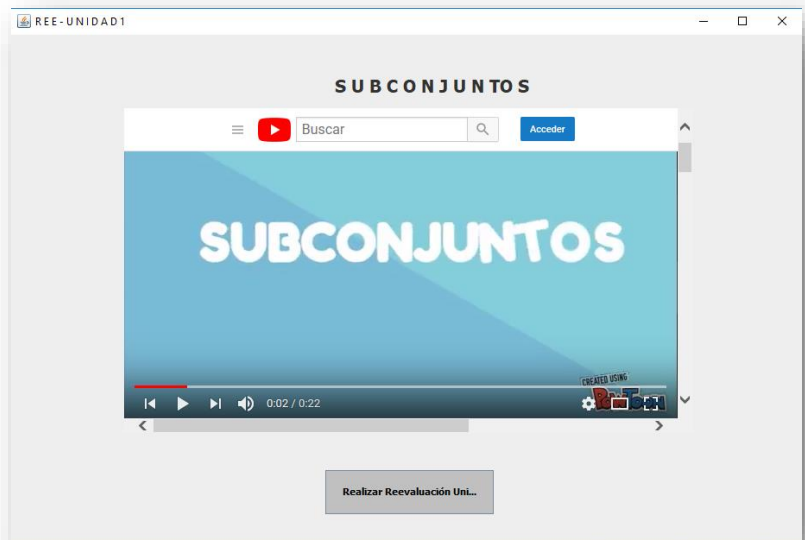




Plan

reenselarU1S1

reevaluarU1S1



Plan

reevaluarU1S1

**REEVALUACIÓN UNIDAD 1**

1.- El conjunto de las vocales es: { a, b, c, d, e }

Verdadero     Falso

2.- Dado el conjunto  $K = \{ 5, 7, 10, 4, 3 \}$ . Elige un conjunto Igual a este

L = { 7, 4, 3 }     H = { 4, 3, 10, 5, 7 }     h = { 5, 7, 10, 4, 3 }

3.- Dado el conjunto  $N = \{ a, k, l, m \}$ . Elige un subconjunto de N.

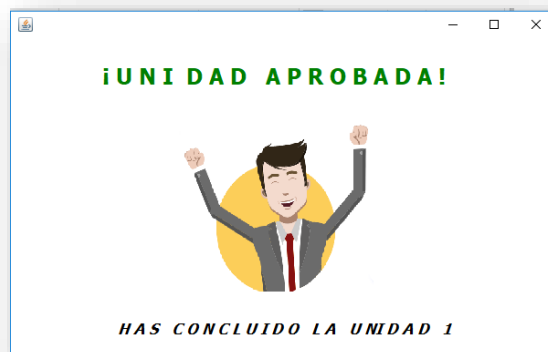
P = { N, L, M, K, ... }     B = { a, k, l }     z = { k, m, l, a, p }

4.- Dados  $A = \{ a, b, 1, 2, 3 \}$  y  $B = \{ a, b, c, 1, 2, 3, 4 \}$

$B \subset A$ .      $A \subset B$ .      $a \subset b$ .

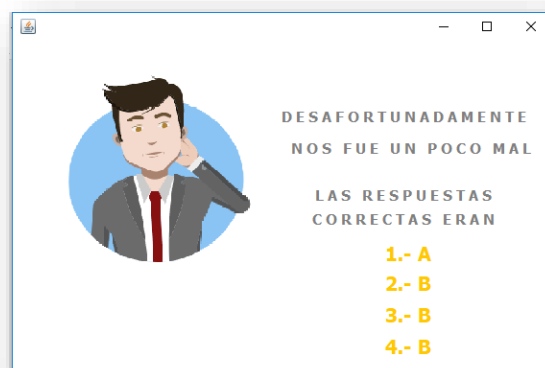
**VERIFICAR**

El usuario puede aprobar o bien fallar de nuevo toda la unidad 1.



Plan

resolverReposU1S1



# CONCLUSIONES

Desarrollar un Objeto de Aprendizaje (OA) como un soporte digital orientado a la educación virtual represento un gran desafío. Por tal motivo al estudiar los fundamentos teóricos y de desarrollo de los OA fue de gran importancia, debido a que se plantearon de manera correcta tanto objetivos, como actividades y la manera en que se debían presentar las evaluaciones a los usuarios.

Por otra parte, se determinó que la implementación del Objeto de Aprendizaje (OA) influyo en la automatización de todo el sistema porque se diseñó utilizando el lenguaje Unificado de Modelado UML obteniendo una visión detallada de todo su funcionamiento de acuerdo al estudio realizado. La arquitectura del sistema se propuso acorde a las necesidades de la interacción entre Java y DLV quienes determinaron las trayectorias que el usuario seguiría dentro del sistema, dándonos finalmente resultados alentadores.

Finalmente, al concluir con el análisis de los resultados obtenidos, se puede decir que todas las pruebas hechas comprobaron el funcionamiento de nuestro Objeto de Aprendizaje (OA) basado en el cálculo de trayectorias de aprendizaje utilizando Answer Set Planning. Estas pruebas nos aseguraron que precisamente el orden en que deben ser llevadas a cabo las actividades de aprendizaje y evaluaciones son determinadas por las trayectorias generadas por DLV.

No obstante, como continuación de este trabajo de tesis y como cualquier otro proyecto de investigación, existen diversas líneas de desarrollo que pueden ser mejoradas, una de ellas es que se pueden agregar más módulos y/o actividades de aprendizaje, así como plantear un mejor diseño de las mismas interfaces que componen a nuestro OA; estas mejoras pueden servir para retomarlas posteriormente en trabajos futuros para otros investigadores.

# REFERENCIAS

1. David A. Wiley. (2002). *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*. Utah State University Recuperado de <http://www.reusability.org/read/>
2. Mason R., Weller, M., Pegler, C.(2003). *Learning in the Connected Economy*. The Open. University course team, IET, Open University.
3. Garcia, Aretio. (2005). *Objetos de Aprendizaje, Características y repositorios*. BENED, Recuperado de [http://www.tecnoeducativos.com/descargas/objetos\\_virtuales\\_deapredizaje.pdf](http://www.tecnoeducativos.com/descargas/objetos_virtuales_deapredizaje.pdf)
4. Consuelo Belloch. (2013). *Diseño instruccional*. Unidad de Tecnología Educativa Recuperado de <https://www.uv.es/~bellochc/pedagogia/EVA4.pdf>
5. Rosanigo, Z., Saenz Lopez, M. and Bianchi, G. (2008). *Diseño de objetos de aprendizaje*. Facultad de Ingeniería, Sede Trelew. Recuperado de <http://sedici.unlp.edu.ar/bitstream/handle/10915/19081/1783-DISE%20DE+OBJETOS+DE+APRENDIZAJE.pdf?sequence=1>
6. Rivera Sánchez, C. (2014). *Objetos de aprendizaje: una primera mirada*. Recuperado de <http://www.infotecarios.com/objetos-de-aprendizaje-una-primeramirada/#.W4XZyegzblU>
7. Erdem, Michael Gelfond, Nicola Leone. (2016). *Applications of Answer Set Programming*. AI Magazine. Recuperado de [http://pages.suddenlink.net/ykahl/SLIDES/defining\\_asp.pdf](http://pages.suddenlink.net/ykahl/SLIDES/defining_asp.pdf)
8. Julián J. Portillo, Carmen L. García Mata, Pedro R. Márquez Gutiérrez y Rogelio Baray-Arana A. (2011). *Robot Platform Motion Planning using Answer Set Programming*, Instituto Tecnológico de Chihuahua, México. Recuperado de [http://ceur-ws.org/Vol-804/04\\_LANMR11.pdf](http://ceur-ws.org/Vol-804/04_LANMR11.pdf)
9. Thomas Either, Wolfgang Faber, Nicola Leone, Gerald Pfeifer, Axel Polleres. (2001). *System Description: The DLV<sup>K</sup> Planning System*. Recuperado de [https://link.springer.com/chapter/10.1007/3-540-45402-0\\_37](https://link.springer.com/chapter/10.1007/3-540-45402-0_37)
10. K Planning System. Recuperado de <http://www.dlvsystem.com/k-planning-system/>

11. (2017). Capítulo 7: TEORIA DE CONJUNTOS. Recuperado de <https://medium.com/@matematicasdiscretaslibro/cap%C3%ADtulo-7-teoria-de-conjuntos-5ef84ea70025>
12. (2007). Importancia de los medios audiovisuales. Recuperado de <https://pbox.wordpress.com/2007/11/23/importancia-de-los-medios-audiovisuales/>
13. Base de conocimientos: ¿Qué es el lenguaje Java? Recuperado de <http://cs.ictea.com/knowledgebase.php?action=displayarticle&id=8790>
14. WindowsBuilder Pro User Guide Installation. Recuperado de <http://help.eclipse.org/kepler/index.jsp?topic=/org.eclipse.wb.doc.user/html/installation/index.html>
15. DLVSYSTEM S.R.L Recuperado de <http://www.dlvsystem.com/jdlv/>