

**Ambiente visual para el aprendizaje de los  
conceptos básicos asociados a la  
recurrencia.**

**Presentado por:  
Carlos Jerson Angulo  
Diego Yesid Callejas**

**Director:  
Raúl Alfredo Chaparro**

**ESCUELA COLOMBIANA DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
Bogotá 2015**

## **Propósito**

Debe contener la información necesaria para darle al lector una idea precisa del Proyecto de Grado.

El proyecto de aprendizaje de conceptos básicos para la enseñanza de la recurrencia, quiere proveer a sus usuarios una herramienta que les permita aprender y enseñar los conceptos básicos asociados con el tema de la recurrencia. Constará de un ambiente interactivo dividido en varios módulos entre los cuales podrá conocer los conceptos básicos de recurrencia, jugar y aprender de manera didáctica cómo funciona la recurrencia para el diario vivir.

## **1. Contexto (Proyecto)**

### **1.1 Planteamiento del problema y pertinencia del mismo**

La tecnología se ha convertido en un instrumento primordial en la gestión creación de nuevos conocimientos. La enseñanza y el aprendizaje de la matemática atraviesa por una coyuntura, en la cual se hace cada día más importante ponerse a tono, de una forma más interactiva y autónoma. La enseñanza y el aprendizaje de la matemática exigen nuevas formas acordes con la actividad, cultura informática y tecnológica, en que están inmersos nuestros estudiantes. Que mejor que investigar en una simbiosis entre tecnología propia de la gestión del conocimiento matemático (ejemplo Wolfram-Mathematica) y tecnología de gestión, administración y motivación de entornos de aprendizaje y enseñanza (como la multimedia, JAVA, MOODLE etc.,)

### **1.2 Marco teórico y estado del arte.**

En la investigación en nuevas formas de representación del conocimiento de matemáticas para el aprendizaje y la didáctica., se han creado software muy valioso (por ejemplo (<http://www.cabri.com/> , <http://www.wolfram.com/> , <http://car-regla-y-compas.uptodown.com/> ) y ambientes de gestión como las tecnología de la multimedia y de la comunicación. Es pertinente usar este contexto para adaptar y crear propuestas que ayuden a motivar la didáctica y aprendizaje de las matemáticas básicas.

### **1.3 Objetivo**

Diseñar y construir un software, con una interfaz de alto componente visual, basado en una metáfora lúdica que permita la interacción fácil y motivante para la experimentación y conjetura, de conceptos relacionados con la recurrencia.

- Estudiar las principales técnicas de solución de problemas relacionados con la recurrencia.

- Diseñar un ambiente de software, basado en una interfaz con alto componente visual, que permita una fácil interacción
- Recopilación de problemas de distintos estilos
- Estudiar, el lenguaje apropiado, o la combinación de lenguajes para el desarrollo del proyecto
- Construcción de un ambiente, integrado con problemas y actividades tanto de uso del software como de temas relacionados con visualizaciones de problemas de naturaleza recurrente.
- Integración, del proyecto a la filosofía del proyecto escenarios de aprendizaje.

#### 1.4 Justificación

Generalmente para la conceptualización de los temas de matemática (útil en la ingeniería) se traduce a problemas clásicos que aparecen en los libros, situación que no permite la experimentación y la conjetura. Creemos que para estos temas es muy importante que el estudiante cuente con un laboratorio en el cual puedan encontrar los principios a través de situaciones dinámicas y motivadoras basadas en ambientes que permitan la experimentación, conjetura y la auto-regulación del aprendizaje. .

#### 1.5 Área de aplicación del producto resultado del proyecto. (Área empresarial o institucional en donde se va aplicar el producto.

En Colegios, universidades. En general en el contexto educativo.

#### 1.6 Cronograma de actividades

1 semana	Desarrollo y entendimiento de los requerimientos.
2-3 semanas	Desarrollo de problema interactivo con cubos.
4-9 semanas	Desarrollo de ambiente web, protipo de calculadora, diseño de programas de fractales.
10-12 semanas	Desarrollo preliminar de libro interactivo.
12-14 semanas	Pruebas, integración de objetivos al ambiente web.
15-16 semanas	desarrollo de presentación, presentación con jurados.
16-24 semanas	Desarrollo de problemas
24-28 semanas	Desarrollo de libro interactivo
28-30 semanas	Implimentación de modulos de ambiente web
30-31 semanas	Pruebas y aceptación de ambiente web
31-32 semanas	Desarrollo de documentos

## 2. Requerimientos

### 2.1. Descripción del sistema

Este es un ambiente dedicado al aprendizaje de la recursión, destinado a aquel estudiante que esté dispuesto a ampliar sus conocimientos, a través de esta herramienta interactiva que ofrece una serie de recursos para que el estudiante aprenda de manera didáctica. No obstante el objetivo de esta herramienta solo se logra sí, tanto como el estudiante y el pedagogo hacen un uso correcto de esta.

### 2.2. Visión y alcance

#### Visión

El propósito del sistema que se va a desarrollar es que las personas que estén interesadas en el aprendizaje de la recursión dispongan de un ambiente interactivo, usable y para esto se va a desarrollar un ambiente WEB interactivo que se irá desarrollando y mejorando a medida que se realicen pruebas con los usuarios del sistema, para que estos hagan sugerencias y propuestas para este.

#### Alcance

Una página WEB interactiva en la que se podrán ver las ilustraciones visuales de problemas básicos llevados a la vida real con la temática de la recurrencia y los fractales.

### 2.3. Usuarios

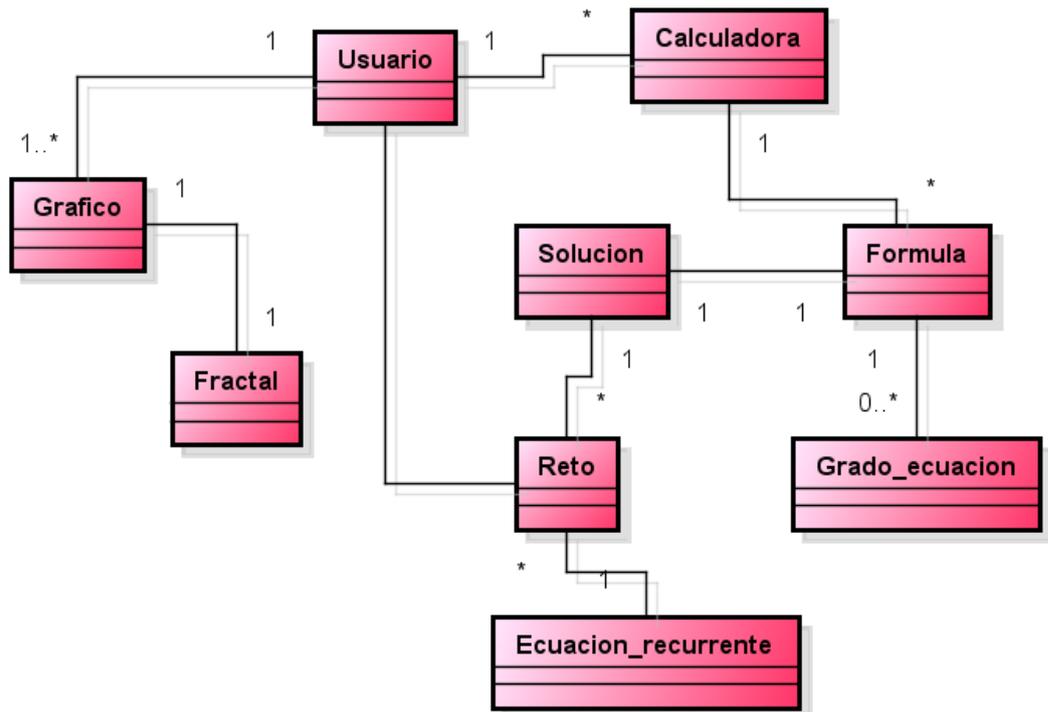
**Estudiante:** Es aquel que tiene un registro en el ambiente, y puede con una cuenta de estudiante acceder a todos los módulos que ofrece la aplicación. Excepto agregar nuevos módulos, retos o problemas gráficos.

**Administrador:** Es aquella persona que goza de privilegios administrativos sobre los usuarios del sistema. Además de tener los beneficios de un estudiante.

## 3. Análisis

### 3.1. Descripción del subsistema

### 3.2. Diagrama conceptual

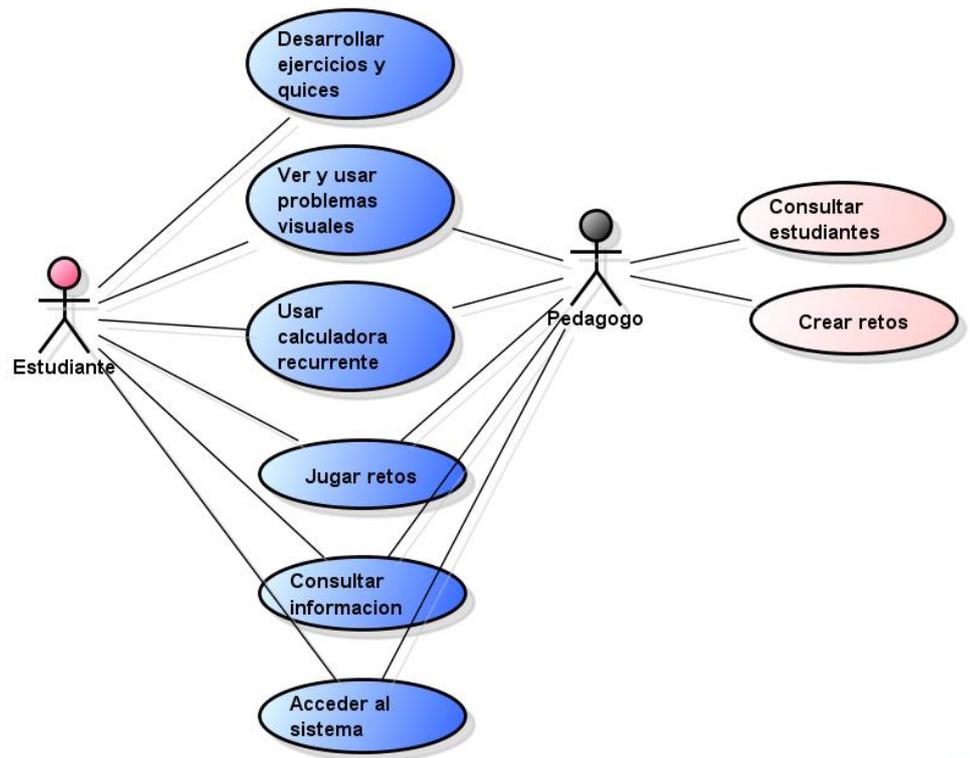


powered by Astah

#### Glosario:

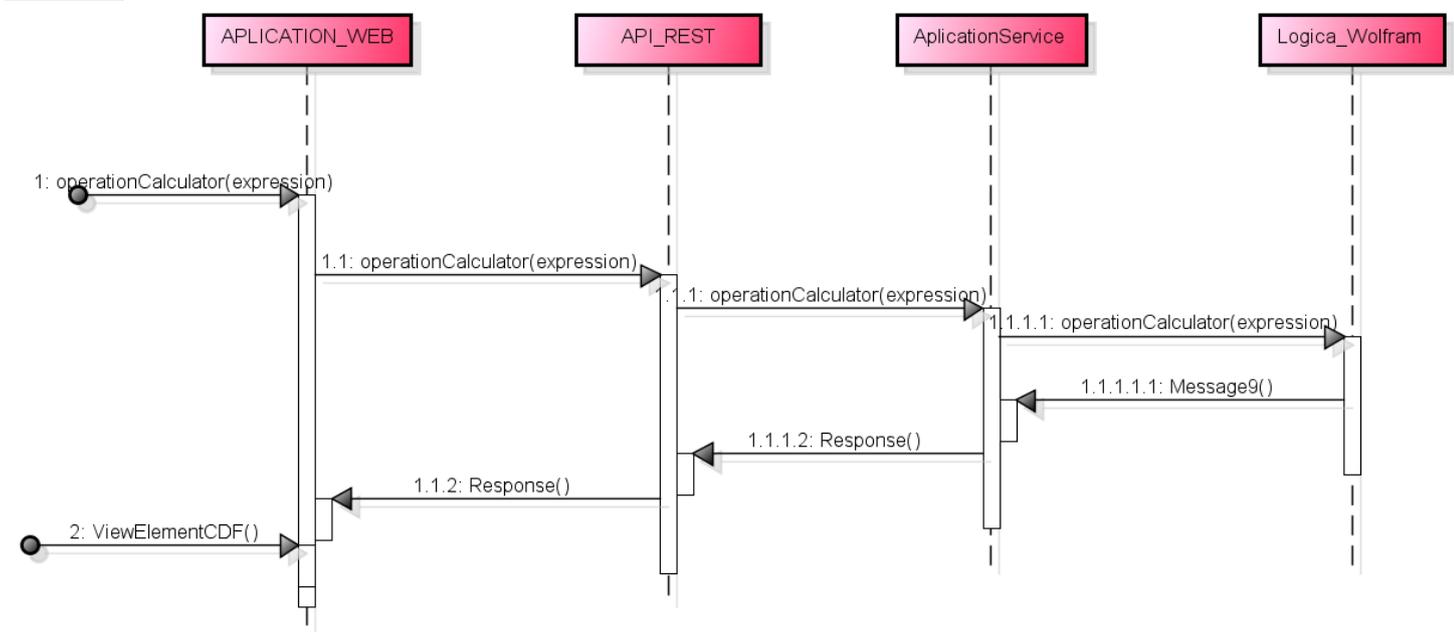
- KernelMathematica: Es el “cerebro” encargado de procesar las instrucciones lógicas de Wolfram.
- Lógica: Es la encargada de enviar las instrucciones a Wólfam y toda la parte lógica de la calculadora
- CalculadoraGUI: Es la implementación de la interfaz que utiliza la lógica de la calculadora para mostrar cómo funciona la calculadora.

### 3.3. Diagramas de casos de uso



powered by Astah

#### 3.3.1. Diagrama de secuencias del sistema



powered by Astah

### 3.3.2. Contratos de operación

<b>Nombre</b>	<i>Calculadora Recurrente</i>
<b>Identificador</b>	CR1
<b>Responsabilidades</b>	<i>Interacción con la calculadora recurrente</i>
<b>Tipo</b>	<i>Usuario</i>
<b>Referencias Casos de Uso</b>	-
<b>Referencias Requisitos</b>	-
<b>PRECONDICIONES</b>	
<i>De Instancia</i>	
<b>POSCONDICIONES</b>	
<i>Se genera aleatoriamente una secuencia y se ocultan algunos términos</i>	
<b>SALIDAS PANTALLA</b>	
<i>Se presenta una secuencia con términos faltantes y se solicita al estudiante que las complete</i>	

<b>Nombre</b>	<i>Completar secuencia</i>
<b>Identificador</b>	CS1
<b>Responsabilidades</b>	<i>Interacción con la aplicación de completar las secuencias propuestas</i>
<b>Tipo</b>	<i>Usuario</i>
<b>Referencias Casos de Uso</b>	-

<b>Referencias</b>	-
<b>Requisitos</b>	
<b>PRECONDICIONES</b>	
<p><i>De Instancia</i></p> <p>Casos base (a[1],a[2],a[3]), Fórmula recurrente a(n)</p>	
<b>POSCONDICIONES</b>	
<b>SALIDAS PANTALLA</b>	
<p><i>En el frame de la aplicación, se muestra la secuencia generada de la fórmula dada con los casos base dados.</i></p>	

### 3.3.3. Especificación de casos de uso

#### 4.

<b>Nombre</b>	Hallar ecuación recurrente (JAVA)
<b>Identificador</b>	HER
<b>Responsabilidades</b>	Módulo para mostrar una secuencia y pedir al usuario la ecuación que genera esa secuencia.
<b>Tipo</b>	Usuario
<b>Referencias</b>	-
<b>Casos de Uso</b>	
<b>Referencias</b>	-
<b>Requisitos</b>	
<b>PRECONDICIONES</b>	

*De Instancia*

*String con la fórmula propuesta por el usuario*

*Casos base propuestos por el usuario*

**POSCONDICIONES**

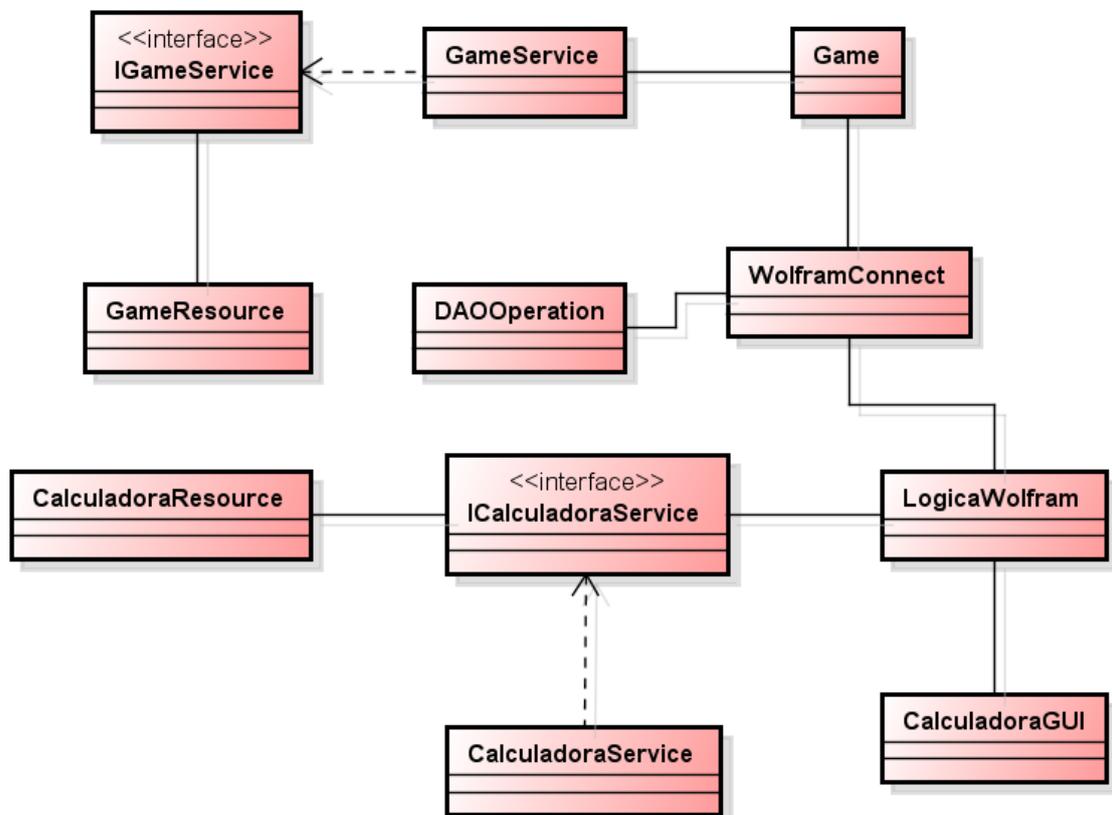
*Intentos se suma en 1*

**SALIDAS PANTALLA**

*Le muestra al usuario si la fórmula es correcta a través de un mensaje de JAVA.*

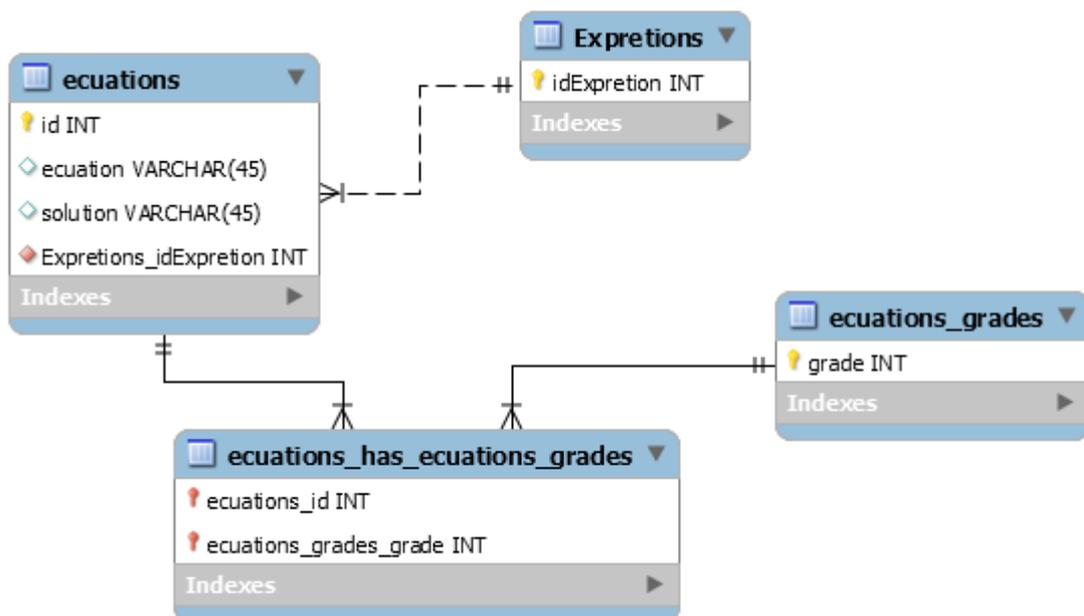
## 5. Diseño

### 5.1. Diagrama de clases



- **IGameService:** interfaz que provee los métodos para implementar un juego que se exporta mediante un servicio
- **GameService:** Implementación de la interfaz “IGameService” que se exporta como un servicio.
- **GameResource:** Recurso REST que provee servicios Web para la implementación de los juegos de la aplicación.
- **Game:** Clase que contiene la lógica y se conecta con Wolfram para el manejo de instrucciones para la implementación de los juegos.
- **WolframConnect:** Clase para la conexión y manejo de mensajes entre java y el kernel de Wolfram Mathematica
- **LogicaWolfram:** Clase encargada del procesamiento de instrucciones de wólfram mediante java.
- **CalculadoraGUI:** Interfaz swing que maneja la calculadora recurrente usando wólfram mathematica.
- **DAOOperation:** Clase que se encarga del manejo de la persistencia de la aplicación y la conexión con la base d datos.
- **CalculadoraResource:** Recurso REST que proporciona servicios web para el manejo de la calculadora.

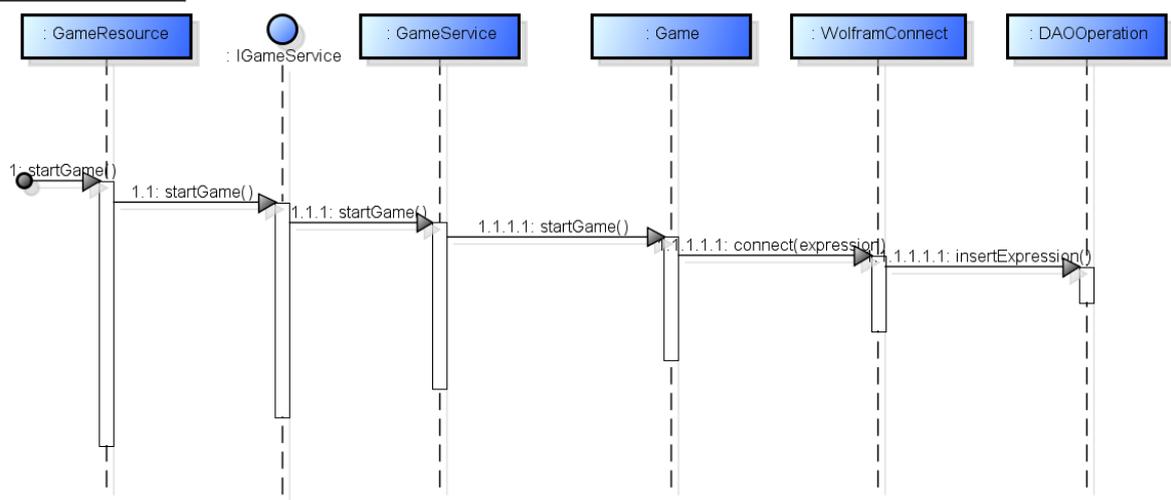
## 5.2. Modelo entidad-relación



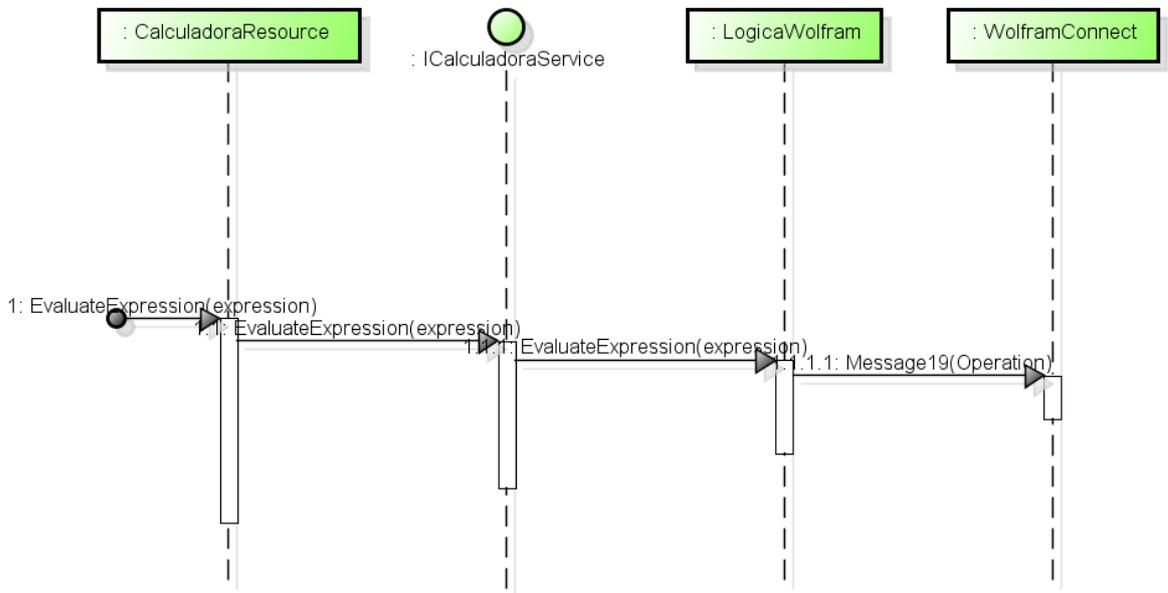
Este modelo ER muestra la persistencia de la aplicación en donde se almacena la principal información de la aplicación.

## 5.3. Vistas arquitectónicas

### 5.3.1. Vista lógica

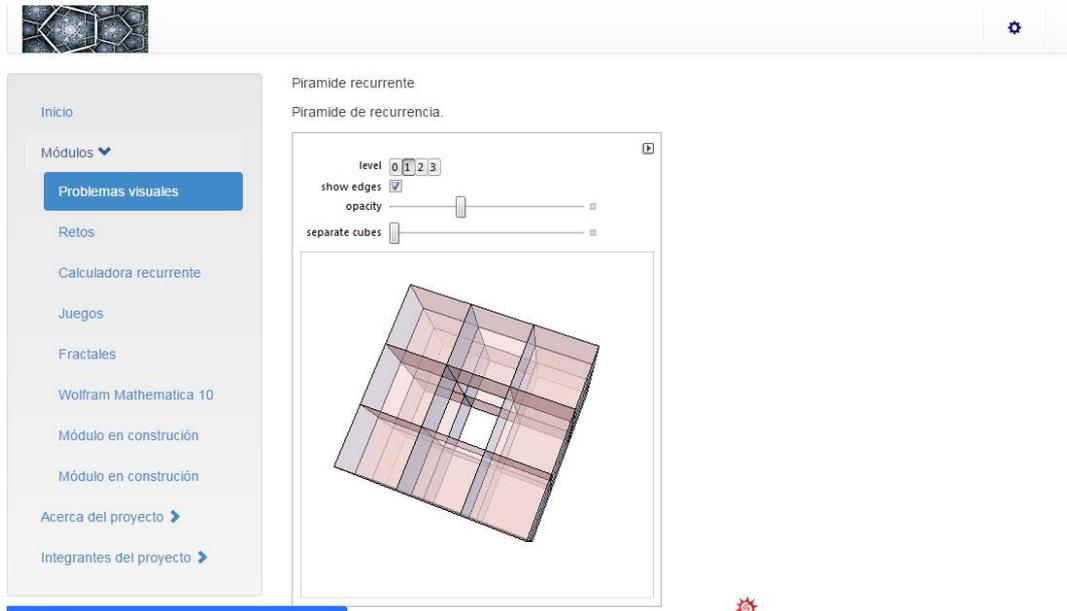


powered by Astah



powered by Astah

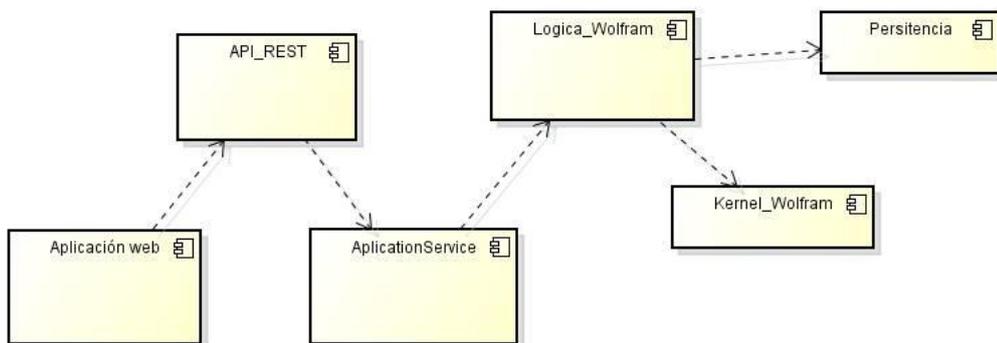
### 5.3.2. Vista física



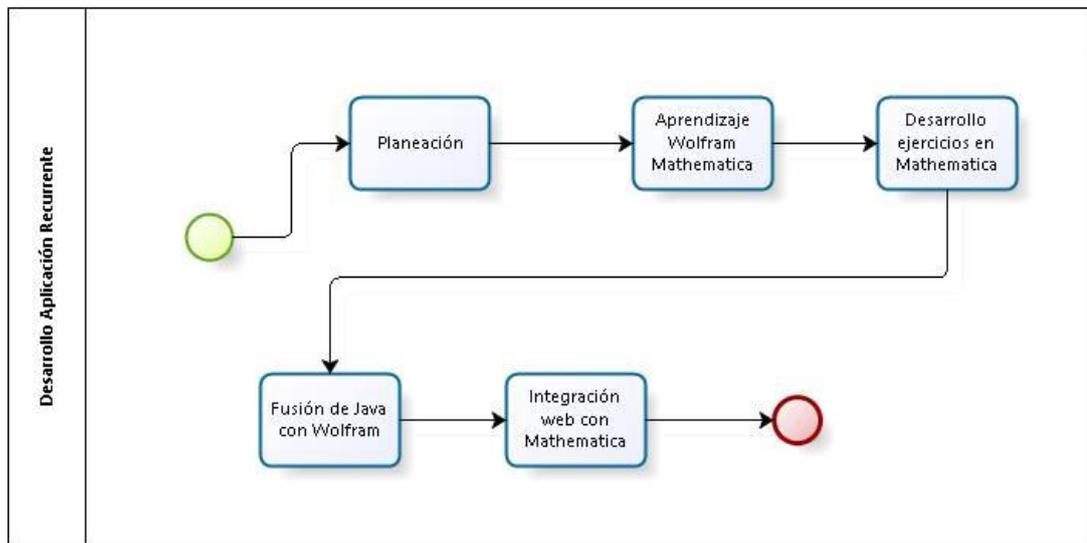
El uso de la aplicación consta de un login con un usuario y una contraseña, donde sí se accede de manera exitosa, la aplicación lo redirigirá a la vista principal donde se encuentran en una barra lateral los módulos que tiene disponibles, (retos, juegos, fractales, calculadora recurrente y información de la aplicación)

Para finalizar la sesión se presión el botón logout y se cierra la sesión activa.

### 5.3.3. Vista de desarrollo



### 5.3.4. Vista de procesos



## 6. Implementación

### 6.1. Especificación de estándares utilizados

- JavaScript
- Html5
- Java\Spring
- Wolfram Development

## 7. Liberación

### 7.1. Configuración ambiente mínima/ideal

El ambiente requiere tener instalado:

- Wolfram cdf player
- Mozilla o Chrome
- Java 1.6 o una versión superior
- Apache tomcat 7 o una versión superior
- Configurar la variable JAVA\_HOME en las variables de entorno del sistema

### 7.2. Manual de instalación

- Es importante primero aclarar que este sistema SOLAMENTE funciona plenamente en los siguientes sistemas operativos:
  - Linux
  - Windows
  - Mac
- Se debe tener instalado en el ordenador que se quiera usar el ambiente previamente **wólfram cdf player** el cual se puede encontrar en este link <http://www.wolfram.com/cdf-player/>
- Luego se debe tener uno de estos dos navegadores:
  - <https://www.mozilla.org/es-ES/firefox/new/>

- <https://www.google.com/intl/es/chrome/browser/desktop/index.html>
- En uno de los dos navegadores anteriormente instalados ingrese al siguiente link: <https://localhost:8080/recurrence/app/#>
- Luego de ingresar al link y tener instalado correctamente el **wólfram cdf player** tiene la plena posibilidad de interactuar con el ambiente.

### 7.3. Manual de usuario

- Estudiante
  - Se debe registrar al ambiente ingresando solo unos pocos datos y el docente asociado.
  - Teniendo un usuario con rol de estudiante, este puede aplicar a todas las funcionalidades del sistema ya sean: retos, juegos, conceptos, aprendizaje de fractales, una calculadora especializada en recurrencia y aplicar a pruebas formuladas por el docente asociado.
- Pedagogo
  - Se debe registrar al ambiente ingresando solo unos pocos datos y títulos que se posean y esperar una aprobación por el administrador.
  - Luego de ser aprobado tiene la posibilidad de vincular estudiantes a sus pruebas, ejercicios propuestos y acceder a todos los servicios que ofrece.
  - Además de llevar un registro de desempeño de sus estudiantes.

### 7.4. Herramientas

- Wolfram Mathematica 10
- Java 1.7
- Jersey
- Astah Comunity
- Html5
- JavaScript
- Bootstrap
- AngularJS
- JAX RS
- DB visualizer

## **7.5. Guías para desarrolladores**

La aplicación está hecha bajo un arquitectura REST, si se desea continuar con el desarrollo se debe implementar la clase interfaz de juegos y montar nuevos servicios REST consumidos mediante un api RESTful. La aplicación está hecha con AngularJS siguiendo la metodología RESTful y la parte visual con Bootstrap y HTML5. Las vistas de la aplicación están la carpeta raíz recurrence/app subdividido en carpetas.

## **7.6. Plan para continuar el proyecto**

Esperamos que el proyecto aumente la gama de problemas visuales y retos más avanzados de los que se presentan actualmente. En el futuro el ambiente web esté integrado a todos los laboratorios de clases que se encuentran en la universidad, además que logre integrarse a otras plataformas virtuales (por ejemplo Moodle) para que los estudiantes accedan al ambiente desde cualquier lugar ajeno a la universidad.

Esperamos se continúe con el desarrollo y usando la misma filosofía y arquitectura, continuar el desarrollo

