

**CLUBMAT: GESTOR DE CLUBES DE MATEMÁTICAS CONSTRUIDO CON
TECNOLOGÍAS JAVA EE 6 Y JAVAFX 2**

ANDRÉS EDUARDO SÁNCHEZ MENDIETA

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.**

2013

CLUBMAT: Gestor de clubes de matemáticas construido con tecnologías Java EE 6
y JavaFX 2

Autor:

Andrés Eduardo Sánchez Mendieta

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE INGENIERO DE
SISTEMAS

Directora

María Consuelo Franky de Toro

Jurados del Trabajo de Grado

Julio Ernesto Carreño Vargas

César Julio Bustacara Medina

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~CIS1310IS11/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.

Junio, 2013

Bogotá, 26 de noviembre de 2013

Señores

Comité de trabajos de grados

Facultad de Ingeniería de sistemas

Pontificia Universidad Javeriana

Estimados colegas:

Como directora de trabajo de grado del estudiante ANDRÉS EDUARDO SÁNCHEZ MENDIETA, titulado GESTOR DE CLUBES DE MATEMÁTICAS CONSTRUIDO CON TECNOLOGÍAS JAVA EE 6 Y JAVAFX 2, autorizo la entrega del documento final.

La sustentación del trabajo de grado se llevó a cabo el pasado jueves 10 de octubre de 2013 y contó con la presencia de los ingenieros Julio Ernesto Carreño Vargas y César Julio Bustacara Medina quienes actuaron como jurados en el proceso. El concepto final correspondiente al trabajo de grado es Aprobado.

Cordialmente,



María Consuelo Franky de Toro

Profesora de planta

Facultad de Ingeniería de sistemas

Pontificia Universidad Javeriana

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

En esta culminación de mi carrera profesional quisiera agradecerles a mis padres, pues sin ellos hubiera sido imposible alcanzar esta meta, ya que finalmente ellos son los que han estado ahí apoyándome a pesar de todas las dificultades presentadas en medio de este proceso.

También quisiera agradecer a la Ingeniera María Consuelo Franky, quien fue la directora de mi trabajo de grado, por todas sus ayudas, guías, asesorías para poder realizar el presente trabajo de grado.

Contenido

LISTA DE ILUSTRACIONES.....	15
LISTA DE TABLAS	16
INTRODUCCIÓN	22
1. OPORTUNIDAD, PROBLEMÁTICA, ANTECEDENTES	24
1.1. Descripción del contexto.....	24
1.2. Formulación del problema que se resolvió	24
1.3. Justificación	24
1.4. Impacto esperado	26
2. DESCRIPCIÓN DEL PROYECTO	27
2.1. Visión global.....	27
2.2. Objetivo general.....	27
2.3. Objetivos específicos	27
2.4. Método que se propuso para satisfacer cada objetivo	28
2.4.1. Programación extrema	28
2.4.2. Fase de documentación	31
2.4.3. Fase de integración	32
2.4.4. Fase de diseño	32
2.4.5. Fase de programación	33
2.4.6. Fase de evaluación	34
2.4.7. Fase de elaboración de guías.....	34
3. MARCO TEÓRICO	36
3.1. Marco Contextual: GESTIÓN DE CLUBES DE MATEMÁTICAS	36
3.1.1. Clubes de matemáticas.....	36
3.2. lista de reglas para crear o implementar un club de matemáticas	38

3.3.	Marco conceptual: Aplicaciones empresariales web	39
3.3.1.	Descripción	39
3.3.2.	Java EE 6.....	40
3.3.3.	JavaFX 2	42
3.3.4.	JavaFX Scene Builder.....	46
3.3.5.	JPA.....	47
3.3.6.	EJB.....	48
3.3.7.	Glassfish.....	48
3.3.8.	PostgreSQL.....	49
3.3.9.	Servicios web SOAP.....	50
4.	DESARROLLO DEL TRABAJO	52
4.1.	Problemática	52
4.1.1.	¿Qué es CLUBMAT?	52
4.2.	Análisis y diseño del sistema de información: CLUBMAT	53
4.2.1.	Requerimientos del sistema de información agrupado por módulos	53
4.2.2.	Módulos	56
4.2.3.	Casos de uso agrupados por módulos	57
4.2.4.	Descripción de los Roles.....	60
4.2.5.	Diagrama de entidades de negocio.....	61
4.3.	Arquitectura de la aplicación web CLUBMAT	64
4.3.1.	Integración de tecnologías basada en Servicios Web SOAP	68
4.3.2.	Descripción de la capa de presentación hecha con JavaFX 2	70
4.3.3.	Descripción de la capa de negocio	73
4.4.	Construcción del sistema de información	76
4.4.1.	Generación inicial de la aplicación CLUBMAT.....	82

4.4.2.	Módulo de seguridad.....	84
4.5.	Resumen de los pasos para construir un CRUD con tecnologías Java EE 6 y JavaFX	
2	85	
4.6.	Pruebas para la aplicación web CLUBMAT.....	88
4.6.1.	Gestión de Roles	88
4.6.2.	Gestión de usuarios	89
4.6.3.	Gestión de instituciones	90
4.6.4.	Gestión de clubes	90
5.	RESULTADOS	92
5.1.	Resultados en lo relativo a las nuevas tecnologías.....	92
5.2.	Resultados en lo relativo a la aplicación web CLUBMAT	92
5.3.	Cumplimiento de los objetivos generales y específicos.....	93
6.	CONCLUSIONES Y TRABAJOS FUTUROS.....	96
6.1.	Conclusiones	96
6.2.	Recomendaciones	97
6.3.	Trabajos futuros	97
7.	REFERENCIAS Y BIBLIOGRAFÍA	98
8.	ANEXOS	102
8.1.	Requerimientos del sistema (Hacer & Usos).....	102
8.2.	Inventario de módulos y casos de uso.....	102
8.3.	Manual del usuario.....	102
8.4.	Manual de mantenimiento.....	102
8.5.	Cronograma del proyecto.....	102
8.6.	Glosario.....	102

LISTA DE ILUSTRACIONES

Ilustración 1: Valores de la programación extrema [21]	29
Ilustración 2: Composición de un club de matemáticas	38
Ilustración 3: Distribución multi-capas de la plataforma Java EE 6 [1]	42
Ilustración 4: Gráficos estadísticos en JavaFX 2	44
Ilustración 5: Gráficos 3D en JavaFX 2	44
Ilustración 6: Captura de pantalla del software JavaFX Scene Builder	47
Ilustración 7: Modo de funcionamiento de un Servicio Web SOAP	50
Ilustración 8: Diagrama de casos de uso para la aplicación web: CLUBMAT	59
Ilustración 9: Diagrama de entidades de negocio	62
Ilustración 10: Arquitectura multicapa de la aplicación web CLUBMAT	66
Ilustración 11: Arquitectura de la aplicación web CLUBMAT	68
Ilustración 12: Diagrama para el CRUD Club	69
Ilustración 13: Diagrama de secuencia para crear un nuevo usuario	70
Ilustración 14: Vista lógica	72
Ilustración 15: Diagrama de clases para la capa de negocio	75
Ilustración 16: Llamar un método remoto de un servicio web SOAP en un controlador	79
Ilustración 17: Apariencia Visual 1	83
Ilustración 18: Apariencia Visual 2	84

LISTA DE TABLAS

Tabla 1: Objetivos específicos	28
Tabla 2: Fase de documentación.....	32
Tabla 3: Fase de integración	32
Tabla 4: Fase de diseño.....	33
Tabla 5: Fase de programación	33
Tabla 6: Fase de evaluación.....	34
Tabla 7: Fase de documentación.....	35
Tabla 8: Requerimientos agrupados por módulos.....	56
Tabla 9: Requerimientos no funcionales.....	56
Tabla 10: Módulos del sistema de información CLUBMAT.....	57
Tabla 11: Inventario de módulos y casos de uso	58
Tabla 12: Descripción de los roles.....	61
Tabla 13: Descripción de las entidades para la aplicación web CLUBMAT.....	64
Tabla 14: Análisis de los objetivos generales	93
Tabla 15: Análisis de los objetivos específicos	95

ABSTRACT

There are many programming platforms to build enterprise web applications. One of them is Java EE 6 [1], which is an architecture that integrates several frameworks such as JavaServer Faces (JSF), Enterprise JavaBeans (EJB) and Java Persistence API (JPA).

In this combination of technologies, the JSF framework causes several problems such as: security issues like XSS or CRSF [39], XHTML specialist is needed to build the GUI and there are no proprietary tools to generate 3D graphics, animations, among others. Given these problems and with the intention to explore new technologies, this project substituted JSF framework with JavaFX 2 technology. [7] This project explores a new framework like JavaFX 2 and seek advantage or contexts where this is more appropriate than JSF.

With JavaFX 2 and the other layers of Java EE 6 was built CLUBMAT application: Math Clubs Manager. This application is an information system that help to manage all the details about math clubs of PROSOFI social program of the Pontifical Xavierian University. [32]

This document describes how technological integration was done and how with these technologies was built CLUBMAT application.

RESUMEN

Para la construcción de aplicaciones web empresariales existen varias plataformas de programación. Una de ellas es Java EE 6 [1], la cual es una arquitectura que integra varios frameworks como lo son JavaServer Faces (JSF), Enterprise JavaBeans (EJB) y Java Persistence API (JPA).

En este conjunto de tecnologías, el framework JSF causa varios problemas como lo son: seguridad vulnerable como XSS o CRSF [39], se depende de especialistas en XHTML para construir la interfaz gráfica y se dispone de pocas herramientas propias para generación de animaciones, gráficos 3D, entre otros. Teniendo en cuenta estos problemas y con el ánimo de explorar nuevas tecnologías, en este proyecto se sustituyó el framework JSF por la tecnología JavaFX 2. [7]. En este proyecto se explora un nuevo framework como JavaFX 2 y busca ventajas o contextos en donde éste es más adecuado que JSF.

Con JavaFX 2 y las demás capas de Java EE 6 se construyó la aplicación CLUBMAT: Gestor de Clubes de Matemáticas. Esta aplicación consiste en un sistema de información que busca administrar todos los detalles alrededor de los clubes de matemáticas del programa social PROSOFI de la Pontificia Universidad Javeriana. [32]

Este documento describe cómo se realizó la integración de tecnologías y cómo con estas tecnologías se construyó la aplicación CLUBMAT.

RESUMEN EJECUTIVO

Las empresas de hoy en día viven en un mundo global competitivo, en el cual se necesitan aplicaciones robustas para satisfacer las necesidades de negocio, que son cada vez más complejas. Con el avance de las tecnologías web e Internet, se han abierto nuevas oportunidades para los desarrolladores de aplicaciones empresariales, permitiéndoles el uso de las nuevas tecnologías web en el desarrollo de aplicaciones mucho más complejas y escalables. Una de las plataformas que ha surgido es la edición empresarial de Java (Java Enterprise Edition, Java EE 6) [1] que es una plataforma de programación que proporciona un conjunto de tecnologías para el desarrollo de aplicaciones empresariales.

Las principales tecnologías que componen la plataforma Java EE 6 son: JavaBeans empresariales (Enterprise JavaBeans, EJB 3) [4], un API para el manejo de entidades persistentes (Java Persistence API, JPA 2) [3] sobre bases de datos relacionales y finalmente JavaServer Faces (JSF 2) [1] que es la tecnología estándar para la creación de interfaces de usuario en la web.

Esta última tecnología (JSF), a pesar de ser la tecnología oficial para crear la capa de presentación, genera algunos problemas como lo son: seguridad vulnerable como XSS o CRSF [39], se depende de especialistas en XHTML para construir la interfaz gráfica y que hay pocas herramientas para generación de animaciones, gráficos 3D, entre otros. Estos problemas se explican en la sección 1.3 Justificación.

Buscando una alternativa, en el presente proyecto se sustituyó la tecnología JSF por otra tecnología de interfaz gráfica como lo es JavaFX 2. [7] JavaFX 2 es un framework que, combinado con Java, permite crear y desplegar aplicaciones con un aspecto vanguardista y atractivo para el usuario final. Esta tecnología está pensada para el desarrollo de RIAs (Rich Internet Applications) [26], aplicaciones web que tienen características y capacidades similares a las aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas.

JavaFX 2 tiene la gran ventaja de que no genera páginas web HTML, pues se ejecuta como si fuera un applet en el navegador. La ventaja de los applets sobre las páginas web basadas en HTML es que son más difíciles de alterar en ejecución.

Los applets también tienen desventajas: son más lentos de procesar y tienen espacio muy delimitado en la página web donde se ejecutan, es decir, no se mezclan con todos los componentes de la página web ni tienen acceso a ellos.

El hecho de que los applets se descarguen en el cliente para ejecutarse, lo que se denomina cliente pesado, hace la aplicación web dependiente de la velocidad de procesamiento del usuario. [44]

En el presente trabajo de grado se buscó lograr una integración de tecnologías. En el cliente usar el framework gráfico Java FX 2 en vez de JSF. En el servidor usar los componentes estándares EJB 3 y JPA 2 de la arquitectura Java EE 6. [1]

Toda esta integración de tecnologías se usó en la aplicación CLUBMAT: Gestor de clubes de matemáticas. CLUBMAT es un sistema de información que permite la creación y la administración de clubes de matemáticas dentro de cualquier tipo de institución educativa en el contexto de PROSOFI.

Un club de matemáticas es un conjunto de clases en donde los estudiantes pueden adquirir práctica en el área de matemáticas. Es diferente a una clase convencional, pues en un club de matemáticas se pretende que el aprendizaje sea de forma lúdica en donde no exista ningún tipo de calificación al estudiante.

Si bien se busca que el producto final de este proyecto sea una porción del sistema de información propuesto, se pretende también que cuente con todo el rigor y la documentación necesaria para que a su vez pueda ser extendido y terminado a corto plazo por otro trabajo de grado de Ingeniería de Sistemas de la Pontificia Universidad Javeriana.

Previo a la construcción de la aplicación CLUBMAT se realizó un levantamiento de requerimientos donde se evaluaron las necesidades del cliente, que es la comunidad de Usme, y un alcance para la versión inicial del sistema de información.

Con base en los requerimientos se definieron cada uno de los módulos alcanzables que tiene la aplicación web, los cuales son: el módulo de seguridad y el módulo de clubes. Cada uno de estos módulos está compuesto por varios casos de uso.

El **módulo de seguridad** presta todos los servicios a los usuarios que controlan los permisos de ingreso a la aplicación. Allí se podrá definir a qué opciones del sistema tiene acceso un determinado usuario de acuerdo con su perfil de seguridad.

El **módulo de clubes** brinda todos los servicios asociados a los clubes de matemáticas. Estos servicios son: El registro, actualización, despliegue y eliminación de un club de matemáticas, un sistema de noticias, la gestión de olimpiadas de matemáticas en las cuales participan varios clubes, el seguimiento de las reglas de creación por cada club y un banco de preguntas.

Este documento pretende describir cómo fue el proceso de integración de tecnologías y cómo con estas tecnologías se construyó la aplicación CLUBMAT.

Primero se explica la oportunidad, problemática y antecedentes, allí se realiza la formulación del problema que se resolvió, la justificación y el impacto esperado del presente trabajo de grado. Luego se realiza la descripción del proyecto. Allí se especifican los objetivos y el método usado para realizar los mismos. Después se expone el marco teórico, en el cual se halla el marco contextual y el marco conceptual. Luego se explica cómo fue el desarrollo del presente trabajo de grado. Finalmente se precisan los resultados, conclusiones y trabajos futuros.

INTRODUCCIÓN

Para poder construir una interfaz gráfica en la plataforma Java EE 6 [1], la tecnología estándar es Java Server Faces. Esta tecnología presenta los siguientes problemas: seguridad vulnerable como XSS o CRSF [39], se depende de especialistas en XHTML para construir la interfaz gráfica y hay pocas herramientas para generación de animaciones, gráficos 3D, entre otros. Estos problemas se explican en la sección 1.3 Justificación.

Lo propuesto y realizado en el presente trabajo de grado fue la sustitución en la capa de presentación del framework JSF de la plataforma Java EE 6 [1], por otro framework de interfaz gráfica de usuario como lo es JavaFX 2. [7]

CLUBMAT (aplicación que dio como resultado este trabajo de grado) fue desarrollado utilizando los componentes EJB 3 y JPA 2 de la arquitectura Java EE 6 [1], y también haciendo uso del framework JavaFX 2 como tecnología para la capa de presentación. La mencionada aplicación tiene como principal objetivo ser un sistema de información que administre todos los detalles existentes alrededor de un conjunto de clubes de matemáticas bajo el marco del programa social PROSOFI.

El programa social PROSOFI es una iniciativa académica que nace en la facultad de ingeniería de la pontificia universidad javeriana en el 2009, y que hoy en día cuenta con la participación de 5 facultades más.

El objetivo de este programa es brindar a las comunidades segregadas de la ciudad, herramientas que puedan usar para incentivar su desarrollo. Actualmente se está llevando a cabo en la localidad de Usme al sur-oriente de Bogotá, con el apoyo y esfuerzo de todos sus habitantes. [32]

En este documento se describe el proceso detallado de todo lo que fue utilizado para el desarrollo del producto final. El producto final es la aplicación funcional la cual se anexa en un

CD junto con los manuales de cómo se realizó la integración de tecnologías y de cómo desarrollar en ese ambiente combinado.

Por último se muestran las conclusiones que se obtuvieron luego de la ejecución del presente trabajo de grado.

1. OPORTUNIDAD, PROBLEMÁTICA, ANTECEDENTES

1.1. DESCRIPCIÓN DEL CONTEXTO

El trabajo de grado: “CLUBMAT: Gestor de clubes de matemáticas construido con tecnologías Java EE 6 y JavaFX 2” tuvo como enfoque lograr una integración entre las tecnologías JavaFX 2 [7] y Java EE 6 [1], reemplazando el framework oficial: JSF.

Con lo anterior se logró disminuir los problemas presentes en el framework JSF. Estos problemas se explican en la sección 1.3 Justificación.

De igual forma se construyó una solución para la gestión de los Clubes de Matemáticas, esto a través de una aplicación web en donde se demuestra la integración de las tecnologías mencionadas.

1.2. FORMULACIÓN DEL PROBLEMA QUE SE RESOLVIÓ

¿Cómo pueden integrarse las tecnologías JavaFX 2 [7] y Java EE 6 [1] como alternativa al framework gráfico estándar JSF?

¿Cómo puede efectuarse la gestión de los clubes de matemáticas a través del uso de una aplicación web que integre las tecnologías JavaFX 2 y Java EE 6?

1.3. JUSTIFICACIÓN

Este trabajo de grado se justifica porque hay mucho interés en el framework JavaFX 2 [7] para construir interfaces de usuario. La razón del interés es porque con JavaFX 2 se encuentra una nueva alternativa tecnológica real para construir la capa de presentación de una aplicación web Java EE 6 [1].

Por otro lado, hoy en día no hay mucha documentación que exponga cómo integrar las interfaces JavaFX 2 [7] a los EJBs y demás elementos de la plataforma Java EE 6 [1].

Pero ¿qué se aportaría al realizar esta integración de tecnologías?, la respuesta es que al usar JavaFX 2 [7] como framework gráfico se podrían lograr ventajas respecto al framework JSF [1]. El framework JSF posee los siguientes problemas:

- ❖ **Problemas de seguridad:** En el framework JSF hay vulnerabilidades de seguridad conocidas como XSS o CSRF. [39] XSS (Cross-site scripting) es un agujero de seguridad con el cual se explota la confianza que tiene el usuario en una página web. Por otra parte, CSRF (Cross-site request forgery) busca explotar la confianza que tiene una página web en un usuario. Las mencionadas vulnerabilidades son problemas de seguridad típicos en aplicaciones web, que permite a un tercero inyectar código en una página web con el objetivo de atacar a los usuarios o a la aplicación web. [42] [43] Con JavaFX 2 [7] mejoran estos problemas porque es un applet y es más difícil de modificar en tiempo de ejecución.
- ❖ **Se depende de especialistas en XHTML:** Al ser JSF un framework basado en XHTML se requieren especialistas en ese lenguaje para poder construir la interfaz gráfica de la aplicación web. [41] Con el framework JavaFX 2 no es necesario tener conocimientos externos a Java, solo se necesita usar el software JavaFX Scene Builder para generar los componentes gráficos de la capa de presentación. [27]
- ❖ **No dispone de herramientas propias para una interfaz gráfica rica:** El framework JSF dispone de pocas herramientas para generar fácilmente: gráficos 3D, gráficos estadísticos, animaciones, efectos especiales, entre otros. En el framework JavaFX 2 hay herramientas propias para generar los elementos anteriormente mencionados. Eso es de gran utilidad para generar una interfaz gráfica de usuario rica. [7]

Otra razón por la que es justificable este proyecto es que se solucionó una necesidad presente en el programa social de la Pontificia Universidad Javeriana (PROSOFI), que consiste en desarrollar una aplicación web que permita realizar la gestión de los Clubes de Matemáticas existentes en las diversas instituciones educativas bajo el marco de PROSOFI.

En el momento de realización del presente trabajo de grado no existía ninguna solución informática para administrar la información que gira en torno a los clubes de matemáticas bajo el marco de PROSOFI. Esta información se almacenaba en medios aislados como papeles o documentos de Microsoft Office.

1.4. IMPACTO ESPERADO

Junto con este trabajo de grado se escribieron varias guías: Una donde se documentó paso a paso cómo realizar la integración entre las tecnologías JavaFX 2 [7] y Java EE [1] y otra de cómo construir software en ese ambiente combinado. (Ver los anexos: Manual de mantenimiento y Manual del usuario)

Dichos documentos son un interesante aporte, pues hay pocas guías en donde se diga paso a paso como realizar una integración tecnológica de esas características. Un resumen de esos pasos está en la sección: 4 DESARROLLO DEL TRABAJO.

Además, se enseñó una alternativa real para remplazar la tecnología JSF por otra como JavaFX 2, [7] la cual trae ventajas importantes.

Otro impacto esperado es que CLUBMAT (aplicación que dio como resultado este trabajo de grado) es un primer prototipo estable de software, el cual es una aproximación inicial a un gestor de clubes de matemáticas.

2. DESCRIPCIÓN DEL PROYECTO

2.1. VISIÓN GLOBAL

CLUBMAT es una aplicación web en la cual se busca demostrar que se logró una integración estable entre las tecnologías JavaFX 2 [7] y Java EE 6. [1] Dicha integración se realizó a través de Servicios Web SOAP. [20]

Con CLUBMAT se podrán gestionar los clubes de apoyados por el programa social PROSOFI de la Pontificia Universidad Javeriana. Concretamente se lograrán manejar usuarios, roles, instituciones, clubes de matemáticas de cualquier tipo de institución educativa y olimpiadas.

2.2. OBJETIVO GENERAL

Desarrollar una primera versión de la aplicación CLUBMAT en donde se use la tecnología de interfaz gráfica JavaFX 2. [7]

La tecnología mencionada debe comunicarse con la capa de negocio (EJB) de la plataforma Java EE 6. [1]

2.3. OBJETIVOS ESPECÍFICOS

En la siguiente tabla se presentan los objetivos específicos.

ID	Objetivos específicos
1	Investigar, estudiar y comprender las herramientas tecnológicas alrededor del desarrollo del trabajo de grado: Java EE 6 [1] JavaFX 2 [7], EJB 3 [4], JPA 2 [3], GlassFish [3], entre otras. Entender los beneficios ofrecidos por ellas.
2	Apropiarse de los conocimientos obtenidos de las tecnologías alrededor del proyecto, realizando pruebas con pequeños prototipos. Se debe lograr la integración de todas las tecnologías.
3	Realizar un análisis sobre las reglas de negocio presentes en el proceso de la gestión de clubes de matemáticas bajo el marco de PROSOFI. Refinar que es lo que desea realmente el cliente en el software final.

4	Diseñar y modelar las reglas de negocio implícitas en el software de gestión de clubes de matemáticas, es decir, generar los artefactos UML que modelen dicho problema.
5	Realizar el desarrollo del prototipo cubriendo todos los requerimientos funcionales y no funcionales establecidos en la fase de diseño.
6	Realizar la evaluación del prototipo a entregar usando metodologías como las pruebas de aceptación, pruebas alfa y pruebas beta. [10]
7	Elaborar guías en donde explícitamente se diga cómo se realizó la integración de las tecnologías y como desarrollar software en ese ambiente integrado.

Tabla 1: Objetivos específicos

2.4. MÉTODO QUE SE PROPUSO PARA SATISFACER CADA OBJETIVO

Para asegurar la correcta realización de cada uno de los objetivos descritos anteriormente, fue necesario definir unas fases metodológicas. Algunas de estas fases fueron ejecutadas con la técnica de metodologías ágiles de software: la programación extrema. [21]

2.4.1. PROGRAMACIÓN EXTREMA

La Programación Extrema (XP) es posiblemente el método ágil más conocido y ampliamente utilizado. El nombre fue acuñado por Beck porque el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo, y con la participación del cliente. [22] Esta metodología se llama programación extrema porque dice que aquellas cosas que son buenas hay que llevarlas al extremo.

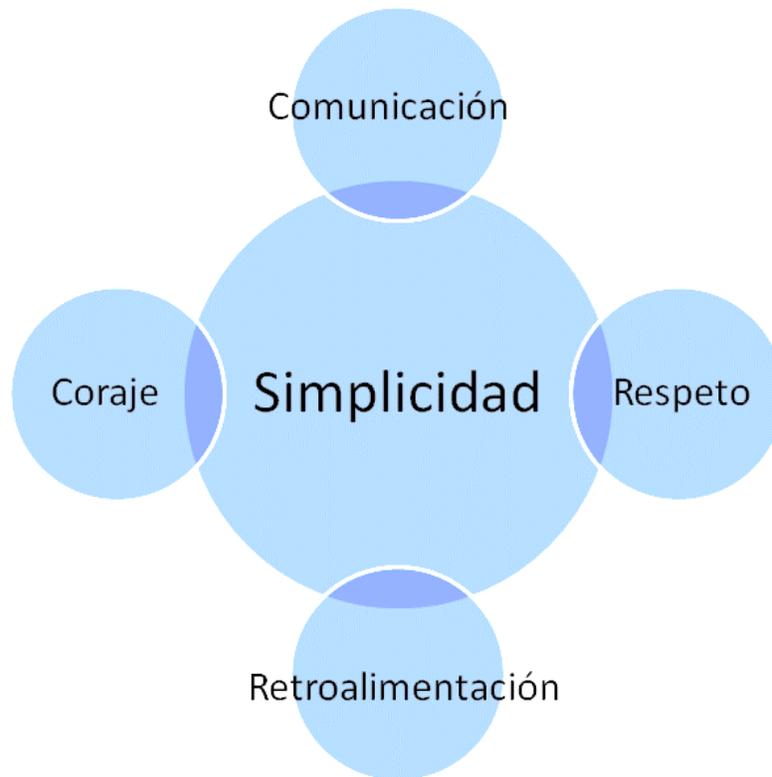


Ilustración 1: Valores de la programación extrema [21]

Esta metodología consiste en un proceso evolutivo de construcción de software, en el cual se deben tener en cuenta cuatro valores propios de este paradigma: [21]

- ❖ **Comunicación:** Se debe tener una alta comunicación entre desarrolladores para que haya buena coordinación en la construcción del software. También es vital para la resolución de problemas. Una importante forma de comunicación es generar código que se auto documente y se describa con comentarios el objetivo de un algoritmo cuando sea necesario.
- ❖ **Simplicidad:** Este valor dice que se debe programar de la forma más sencilla y clara posible. Esto implica, por ejemplo, el uso de nombres de variables que se auto documenten, para así aumentar la claridad.
- ❖ **Retroalimentación:** Debe haber retroalimentaciones del cliente principal del proyecto, y según estas, realizar una toma de decisiones sobre el software en proceso de construcción.

- ❖ **Coraje:** Este valor nos dice que debemos ser valientes a la hora de programar el software, así nos topemos con problemas. Con coraje también se refiere a enfrentar algo que no tenemos idea de cómo resolver. Se debe realizar una reestructuración del proyecto cada vez que sea necesario. La programación extrema es una metodología basada en la prueba y error.

Con base en los anteriores valores se deben ejecutar las siguientes prácticas: [48]

- ❖ **Diseño simple:** Hacer siempre lo mínimo imprescindible de la forma más sencilla posible. Mantener siempre sencillo el código.
- ❖ **Integración continua:** Deben tenerse siempre un ejecutable del proyecto que funcione y en cuanto se tenga una nueva pequeña funcionalidad, debe recompilarse y probarse. Es un error mantener una versión congelada dos meses mientras se hacen mejoras y luego integrarlas todas de golpe. Cuando falle algo, no se sabe qué es lo que falla de todo lo que hemos metido.
- ❖ **Ritmo sostenible:** Se debe trabajar a un ritmo que se pueda mantener indefinidamente. Esto quiere decir que no debe haber días muertos en que no se sabe qué hacer y no trabajar en exceso en otros días. Al tener claro semana a semana lo que debe hacerse, hay que trabajar duro en ello para conseguir el objetivo cercano de terminar una mini-versión.
- ❖ **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- ❖ **Requerimientos de software cambiantes:** Los requerimientos de software pueden cambiar en cualquier momento para adaptarse las necesidades del proyecto.

Teniendo en cuenta los anteriores valores y prácticas, se ejecutaron las fases de integración y de programación. Allí es donde los valores de simplicidad, coraje, retroalimentación y comunicación fueron útiles.

La razón es que en dichas fases es donde se realizaron pruebas para poder realizar la integración de tecnologías y donde se construyó la primera versión de la aplicación web CLUBMAT, es decir, se generaron artefactos en los cuales fue necesario programar y entregar código al cliente.

Para la generación de la aplicación web CLUBMAT fueron necesarias varias entregas incrementales al cliente, al igual que una alta comunicación, simplicidad en el código y coraje para la integración y construcción del producto final.

A continuación se presentan las fases que fueron definidas para el presente trabajo de grado:

2.4.2. FASE DE DOCUMENTACIÓN

Las etapas para lograr los objetivos asociados a la fase de documentación del presente proyecto fueron las siguientes:

Etapas	Descripción	Objetivo específico
❖ Búsqueda y obtención de bibliografía	En esta etapa se buscó información sobre todas las herramientas y teoría alrededor de este proyecto. Esta información fue consultada principalmente en documentos electrónicos y páginas web relevantes. También se usaron libros físicos y electrónicos.	1
❖ Filtro de información	En esta etapa se seleccionó solo la información más relevante de la obtenida en la etapa de búsqueda.	1
❖ Estudio de información	En esta etapa se realizó una lectura más profunda y un estudio de toda la información relevante obtenida.	1
❖ Redacción en documento	Cuando se realizó la redacción del trabajo de grado, se escribió toda la información y conocimiento de forma concreta.	1

Tabla 2: Fase de documentación

Entregable: En esta fase se generó información relevante acerca de todas las herramientas tecnológicas que giran alrededor del proyecto. Esta información se escribió en el capítulo de marco teórico de la memoria del trabajo de grado.

2.4.3. FASE DE INTEGRACIÓN

Las etapas para lograr los objetivos asociados a la fase de integración del presente proyecto son las siguientes:

Etapa	Descripción	Objetivo específico
❖ Integración de tecnologías	En esta etapa se realizó la integración de todas las tecnologías que envuelven el proyecto.	2
❖ Prototipos de prueba	En esta etapa se generaron pequeños prototipos en donde de manera práctica se logre la integración de tecnologías.	2
❖ Análisis de prototipos	En esta etapa se realizó un análisis sobre los prototipos, viendo si realmente se logró la integración de tecnologías o no.	2

Tabla 3: Fase de integración

Entregable: En esta fase se generaron prototipos en donde a pequeña escala se demuestre integración de tecnologías. Cabe aclarar que la mayoría de estos prototipos fueron fallidos como se explicará en el capítulo de desarrollo.

2.4.4. FASE DE DISEÑO

Las etapas para lograr los objetivos asociados a la fase de diseño del presente proyecto son las siguientes:

Etapa	Descripción	Objetivo específico
-------	-------------	---------------------

❖ Análisis del modelo de negocio del problema de manejo de clubes de matemáticas de PROSOFI	En esta etapa se realizó un estudio de las reglas del negocio del prototipo. Logrando entender el problema en su totalidad.	3
❖ Diseño del modelo de negocio del prototipo.	En esta etapa se realizó un análisis de las reglas del negocio del prototipo. Se busca generar los casos de uso y el levantamiento de requerimientos.	4

Tabla 4: Fase de diseño

Entregable: En esta fase se generaron diagramas y documentación estándar UML sobre el prototipo construido. Estos artefactos hacen parte del capítulo de desarrollo de la memoria del trabajo de grado.

2.4.5. FASE DE PROGRAMACIÓN

Las etapas para lograr los objetivos asociados a la fase de programación del presente proyecto son las siguientes:

Etapas	Descripción	Objetivo específico
❖ Desarrollo de la aplicación web CLUBMAT	En esta etapa se construyó el prototipo, cubriendo los requerimientos de alta prioridad.	5
❖ Retroalimentación con el cliente	En esta etapa se evaluó junto al cliente el correcto funcionamiento del prototipo.	6

Tabla 5: Fase de programación

Entregable: En esta fase se generó un prototipo funcional en el que se cubren los requerimientos alcanzables propuestos.

2.4.6. FASE DE EVALUACIÓN

Las etapas para lograr los objetivos asociados a la fase de evaluación del presente proyecto son las siguientes:

Etapa	Descripción	Objetivo específico
❖ Pruebas	En esta etapa el programador realiza en el prototipo pruebas de caja negra y de aceptación de requerimientos de la aplicación web CLUBMAT	6
❖ Pruebas con el cliente	En esta etapa con el cliente se realiza pruebas alfa y beta, asegurando el correcto funcionamiento del prototipo.	6

Tabla 6: Fase de evaluación

Entregable: En esta fase se generó un prototipo estable después de realizar las pruebas.

2.4.7. FASE DE ELABORACIÓN DE GUÍAS

Las etapas para lograr los objetivos asociados a la fase de elaboración de guías del presente proyecto son las siguientes:

Etapa	Descripción	Objetivo específico
❖ Redacción de las guías	En esta etapa se redactaron las guías de mantenimiento y usuario. En ellas se describe: 1) Como es el proceso de integración de tecnologías paso a paso. 2) Herramientas usadas para la generación de la	7

	<p>aplicación web CLUBMAT.</p> <p>3) cómo realizar mantenimiento al sistema entregado.</p> <p>4) Las acciones que puede hacer un usuario con la aplicación web CLUBMAT.</p>	
❖ Evaluación de las guías	En esta etapa se evaluó si las guías son precisas, concretas y correctas.	7

Tabla 7: Fase de documentación

Entregable: En esta fase se generaron unas guías de cómo lograr la integración y desarrollo en Java EE 6 [1] y JavaFX 2 [7] más prototipos. Igualmente, se construyó una página web en donde se publicó todo el material del presente trabajo de grado.

3. MARCO TEÓRICO

En el desarrollo de la aplicación web CLUBMAT se hizo uso de diversas tecnologías. Estas fueron integradas para dar como resultados un artefacto de software final. En este artefacto, que fue un aplicativo web Java EE 6 [1], se debe demostrar la correcta integración de todas las tecnologías propuestas.

Los componentes utilizados en la aplicación web CLUBMAT fueron: Java EE 6 [1], JavaFX 2 [7], EJB 3 [4], JPA 2 [3], GlassFish [3], entre otros.

3.1. MARCO CONTEXTUAL: GESTIÓN DE CLUBES DE MATEMÁTICAS

Los módulos que enmarcan el presente trabajo de grado son el módulo de seguridad y el módulo de clubes. Dichos módulos serán explicados en el capítulo 4 (DESARROLLO DEL TRABAJO) de este documento.

3.1.1. CLUBES DE MATEMÁTICAS

En nuestra sociedad que cada vez es más competitiva, es indispensable tener profesionales capaces de afrontar los diferentes retos que se puedan presentar en el día a día.

Si tenemos en cuenta que estos profesionales son el resultado de un proceso de formación que viene desde años atrás, entonces lo primordial es que desde temprana edad se formen cimientos fuertes de conocimiento que van a verse reflejados en el futuro.

Ahí es donde entra en juego un club de matemáticas. Un club de matemáticas es un espacio donde los estudiantes pueden afianzar sus conocimientos y a la vez aprender nuevos tópicos en matemáticas. Es diferente a una clase de un programa de estudios pues en un club de matemáticas se pretende que el aprendizaje sea de una forma lúdica y que no existan notas ni ningún tipo de evaluación al estudiante.

Luego de realizado un análisis de las comunidades con clubes de matemáticas, como la comunidad de Usme, se concluyó que estos se deben componer de las siguientes partes para un buen desarrollo y funcionamiento:

- ❖ **Interlocutor o profesor:** son aquellas personas que deben tener un nivel alto en conocimiento de matemáticas y en estrategias de enseñanza, deben ser personas lúdicas y que motiven a los estudiantes. Su función principal es la de ayudar a los estudiantes en temas en los que presenten dificultad o en temas en los que ellos indaguen. A nivel de la aplicación web, el interlocutor será quien se auto-registre; tomando como rol “Administrador club”.
- ❖ **Estudiantes:** son alumnos de la misma institución donde reside el club. Son el alma del club de matemáticas puesto que fue hecho para ellos. Los estudiantes de un club de matemáticas que participen en una olimpiada deberán ser registrados por el administrador del club.
- ❖ **Instituciones:** son aquellos sitios que se utilizan para el desarrollo de las sesiones del club. Una institución puede tener uno o varios lugares para el club de matemáticas según la sesión lo necesite, entre ellos encontramos:
 - ✓ Aulas de clase.
 - ✓ Salas de cómputo.
 - ✓ Lugares al aire libre.
 - ✓ Entre otros.

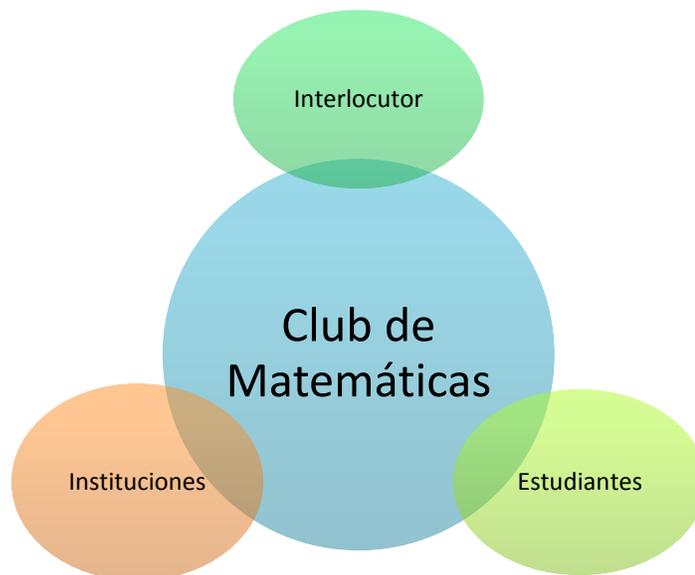


Ilustración 2: Composición de un club de matemáticas

Por otra parte, uno de los objetivos del programa social PROSOFI de la Pontificia Universidad Javeriana es ayudar a la localidad quinta de Usme, la cual se encuentra separada del casco urbano de la ciudad de Bogotá. Esta localidad incluye varios barrios del sur de la ciudad con extensas zonas rurales. Cuenta con población en situación de desplazamiento y las problemáticas que genera éste. [32]

En esta localidad es en donde PROSOFI está realizando las primeras implementaciones de clubes de matemáticas.

3.2. LISTA DE REGLAS PARA CREAR O IMPLEMENTAR UN CLUB DE MATEMÁTICAS

A continuación se enumeran los pasos necesarios para poder crear un club de matemáticas:

- 1) Proponer la inclusión del club de matemáticas por parte del administrador del mismo.
- 2) Aceptación del club de matemáticas por parte de las autoridades de la institución.
- 3) Definir roles dentro del club.

- 4) Establecer grupos de estudio.
- 5) Definir horarios y lugares de sesión.
- 6) Abrir convocatoria a estudiantes.
- 7) Definir nombre y logotipo de cada club.
- 8) Definir cronograma de actividades.
- 9) Definir presupuesto necesario.

3.3. MARCO CONCEPTUAL: APLICACIONES EMPRESARIALES WEB

3.3.1. DESCRIPCIÓN

Una aplicación web es un software que se ejecuta en el entorno de internet y es accedido por el usuario a través de un navegador web. Dicho navegador permite que el usuario interactúe fácilmente con la aplicación. [23]

De igual forma, el navegador utiliza el protocolo de transferencia de información HTTP (Hyper text Transfer Protocol) que permite la conexión con el servidor. [23]

Para construir una aplicación web como la propuesta en este trabajo de grado, existen múltiples tecnologías, las cuales se dividen en dos grupos: tecnologías sin servidor de aplicaciones y tecnologías con servidor de aplicaciones.

Un servidor de aplicaciones es una tecnología que provee la infraestructura necesaria para que las aplicaciones web empresariales funcionen perfectamente. Esto quiere decir que los programadores van a poder dedicarse exclusivamente a programar la lógica de negocio, ya que servicios de uso común, como transacciones, seguridad, persistencia, etc. ya son proporcionados por el servidor de aplicaciones. Cabe aclarar que un servidor de aplicaciones también usa un servidor web y también un middleware que le permite comunicarse con los servicios que necesite. [46]

Dentro de las tecnologías sin servidor de aplicaciones encontramos herramientas como: PHP, Python y Perl. [45] Estas tecnologías dependen de un servidor web.

Por otro lado, en las tecnologías con servidor de aplicaciones encontramos las siguientes plataformas: Java EE 6 [1] y .NET 4.

Un servidor de aplicaciones tiene las siguientes ventajas: [24]

- ❖ La inclusión de un middleware que facilita la comunicación entre varios servicios. [25]
- ❖ Soporte a diferentes estándares como HTML, XHTML y XML permitiendo el funcionamiento en ambientes web.
- ❖ Alta disponibilidad de sus aplicaciones, además garantiza seguridad y confiabilidad de las mismas.
- ❖ Brinda escalabilidad y mantenibilidad en tal caso de que la carga de trabajo incremente. [25]
- ❖ Brinda un módulo de administración de transacciones cumpliendo con las propiedades ACID.
- ❖ Seguridad por roles.
- ❖ Persistencia automática de entidades de negocio.

Como puede observarse son muchas las alternativas tecnológicas a la hora de construir una aplicación web, pero para el presente proyecto se eligió la plataforma Java EE 6. [1] La razón de esta decisión es que el objetivo principal de este trabajo de grado es realizar una integración del framework gráfico JavaFX 2 [7] sobre la mencionada plataforma.

A continuación se realizará una descripción más detallada de las tecnologías que envuelven el presente trabajo de grado:

3.3.2. JAVA EE 6

El objetivo más importante de la plataforma Java EE 6 [1] es simplificar el desarrollo buscando cubrir una amplia gama de necesidades para las aplicaciones empresariales como lo son: transaccionalidad, seguridad, interoperabilidad, escalabilidad, persistencia, objetos distribuidos, entre otros. [47]

La plataforma Java EE 6 [1] mejora significativamente la productividad del programador pues esta arquitectura se caracteriza por la flexibilidad para desarrollar, extensibilidad y es de fácil reestructuración. También incluye las últimas versiones de tecnologías como JAX-RS 1.1 [1], JavaServer Faces (JSF) 2.0 [1], Enterprise JavaBeans (EJB) 3.1 [4], Java Persistence (App) 2.0 [3], Context and Dependency Injection (CDI) 1.0 [1] y mucho más.

La plataforma Java EE 6 incluye las siguientes características nuevas: [1]

- ❖ Perfiles: Para hacer más ligero el contenedor se introdujo el concepto de perfiles, donde podemos seleccionar el conjunto mínimo de tecnologías a utilizar. Por ejemplo si una aplicación Web necesita de EJB, pero no de Servicios Web se puede utilizar EJB Lite, el cual se enfoca en utilizar únicamente las características básicas de los EJB. [47]
- ❖ Nuevas tecnologías, incluyendo las siguientes: [9]
 - ✓ API Java para servicios web SOAP (JAX-WS) y para servicios web RESTful (JAX-RS) [9]
 - ✓ Gestión de Enterprise JavaBeans.
 - ✓ Contextos e inyección de dependencias para la plataforma Java EE (JSR 299), conocido informalmente como CDI [9]
 - ✓ Inyección de dependencias para Java (JSR 330)
 - ✓ Validación de Beans (JSR 303) [9]
 - ✓ Java Authentication Service Provider Interface for Containers (JASPIC).
- ❖ Facilidad de Uso: El uso de anotaciones simplificó en gran medida la configuración vía archivos xml, por lo que archivos como faces-config.xml, ejb-jar.xml y persistence.xml se redujeron a unas cuantas líneas, e incluso en algunos casos es opcional el utilizarlos. Además las clases ahora están orientadas a clases puras de Java (POJOs) e Interfaces, y en algunos casos, como los EJB's, el uso de interfaces es opcional. [47]

a. VENTAJAS:

El modelo de aplicación Java EE 6 define una arquitectura para implementar servicios como aplicaciones multicapa que proporcionan la escalabilidad, accesibilidad y manejabilidad

requeridas por parte de las aplicaciones empresariales. Este modelo particiona el trabajo requerido para implementar un servicio multicapa en las siguientes partes: [1]

- ❖ La lógica de negocio y presentación que debe de ser implementada por el desarrollador.
- ❖ Los servicios del sistema estándar proporcionados por la plataforma Java EE 6.

El desarrollador puede confiar en la plataforma para que le proporcione las soluciones a los problemas difíciles a nivel de sistema que aparecen al desarrollar un servicio multicapa.

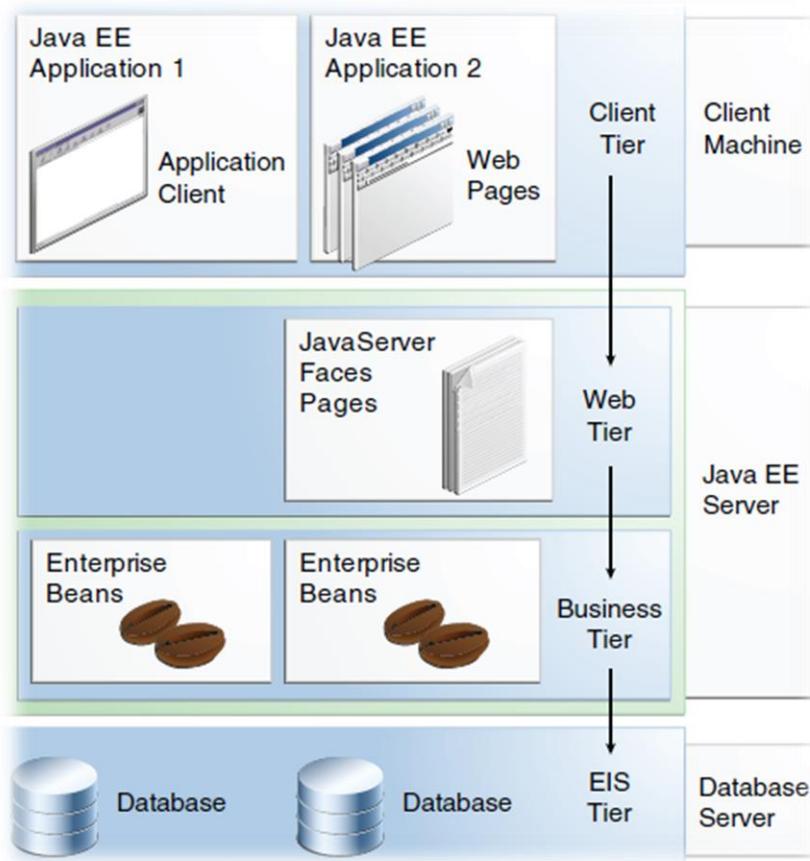


Ilustración 3: Distribución multi-capas de la plataforma Java EE 6 [1]

3.3.3. JAVAFX 2

JavaFX 2 [7] es el framework propuesto por Sun Microsystems, hoy Oracle, para el desarrollo de aplicaciones RIA basadas en el lenguaje de programación Java. Aunque Java ya había estado en el navegador a través de los Applets (clases Java que se ejecutan en el browser), JavaFX 2 es una tecnología diferente. [26]

Similar a Silverlight [26] (framework de Microsoft para construir una interfaz gráfica), las aplicaciones JavaFX 2 tienen una orientación hacia multimedia y podrán ser ejecutadas en el navegador, sin necesidad de cargar ningún plugin, gracias a que los navegadores soportan la máquina virtual Java. Lo que se requiere es el intérprete del bytecode de JavaFX, que es diferente a la sintaxis Java nativa, pero desde JavaFX 2 se puede utilizar cualquier clase o paquete Java. [26]

De esta forma, los desarrolladores pueden ampliar sus capacidades en Java y utilizar la revolucionaria tecnología de presentación que JavaFX 2 proporciona para crear atractivo visual. [9]

b. CARACTERÍSTICAS DE JAVAFX 2: [9]

- ❖ Permite a los usuarios ver aplicaciones de JavaFX en un navegador de internet.
- ❖ Amplía la tecnología Java permitiendo el uso de cualquier librería de Java en una aplicación JavaFX 2.
- ❖ Permite a los desarrolladores integrar gráficos vectoriales, animación, sonido y activos web de vídeo en una aplicación interactiva, completa y atractiva.

A continuación se colocan fotos que demuestran las características gráficas de JavaFX 2:

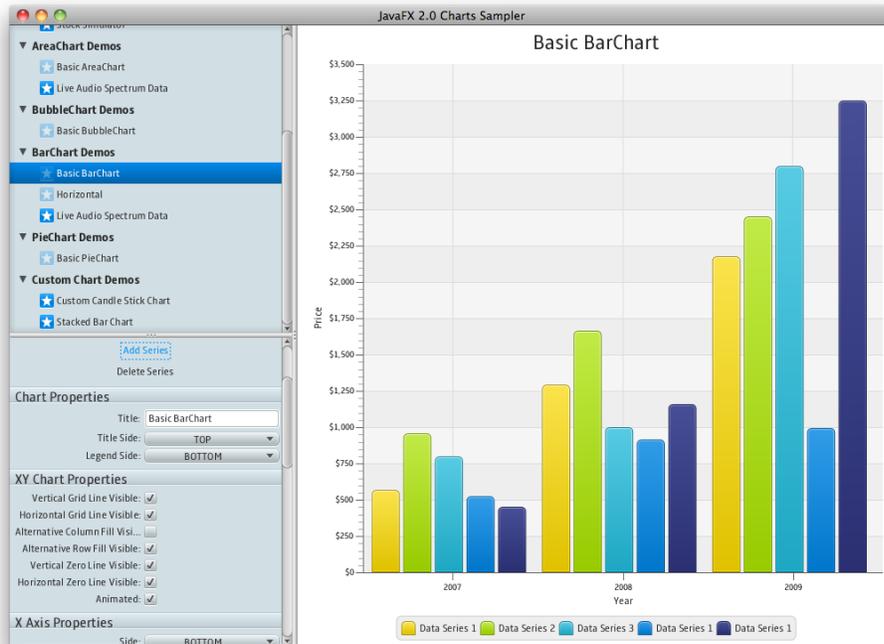


Ilustración 4: Gráficos estadísticos en JavaFX 2

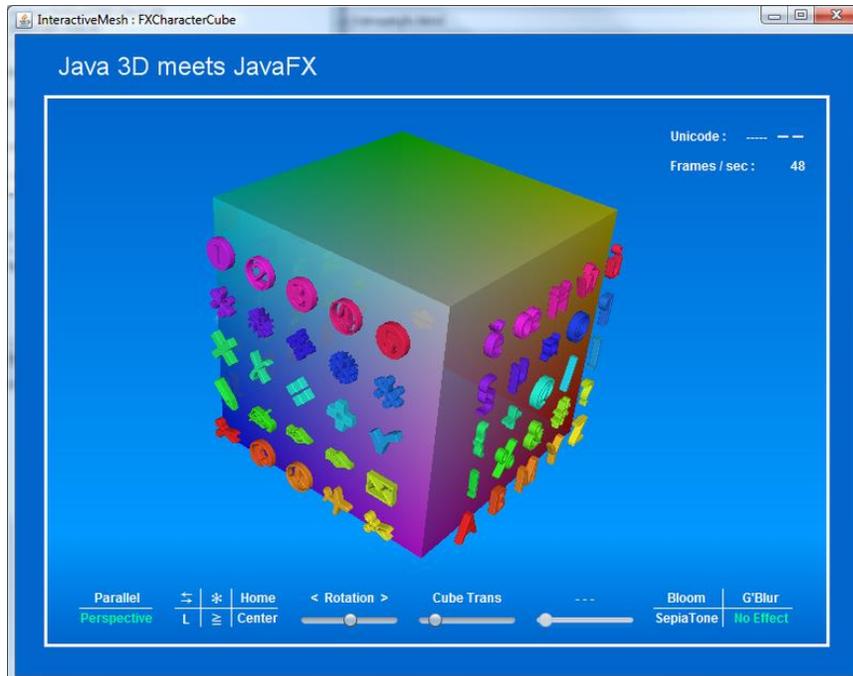


Ilustración 5: Gráficos 3D en JavaFX 2

Como se puede observar, JavaFX 2 es una tecnología gráfica aplicable en escenarios como el de la construcción de videojuegos, pues el framework contiene herramientas nativas para generación de gráficos 3D dinámicos.

También se usa en el escenario del desarrollo de aplicaciones web RIA [26] (Rich Internet Applications) el cual es software online que se comporta igual que una aplicación de escritorio.

Las ventajas de usar el framework JavaFX 2 respecto a JSF son las siguientes:

- ❖ **Menos problemas de seguridad:** En el framework JSF hay vulnerabilidades de seguridad conocidas como XSS o CRSF. [39] Las mencionadas vulnerabilidades son problemas de seguridad típicos en aplicaciones web, que permite a un tercero inyectar código en una página web y que el usuario la ejecute, explotando la confianza que éste tiene en un sitio web o por el contrario, la confianza que un sitio web tiene en un usuario en particular. [42] [43] Con JavaFX 2 [7] mejoran estos problemas porque es un applet el cual es más difícil de modificar en tiempo de ejecución.
- ❖ **No se depende de especialistas en XHTML:** Al ser JSF un framework basado en XHTML se requieren especialistas en ese lenguaje para poder construir la interfaz gráfica de la aplicación web. [41] Con el framework JavaFX 2 no es necesario tener conocimientos externos a Java, solo se necesita usar el software JavaFX Scene Builder para generar los componentes gráficos de la capa de presentación. [27]
- ❖ **Construcción de interfaces graficas ricas:** El framework JSF no dispone de herramientas propias para generar fácilmente: gráficos 3D, gráficos estadísticos, animaciones, efectos especiales, entre otros. En el framework JavaFX 2 se disponen de herramientas propias para generar los elementos anteriormente mencionados. Eso es de gran utilidad para generar una interfaz gráfica de usuario rica favoreciendo. [7]

Las desventajas de usar JavaFX 2 en la arquitectura Java EE 6 en vez de JSF son las siguientes:

- ❖ Al no ser la tecnología oficial para la capa de presentación no se disponen de generadores automáticos de código como los hay para JSF.

- ❖ El desempeño de una aplicación JavaEE 6 con interfaz gráfica JavaFX 2 integrada con servicios web SOAP es peor que el desempeño de una aplicación Java EE 6 con interfaz gráfica JSF. Esto es debido a que mientras que con JSF la integración con los demás componentes de la arquitectura es automática, con JavaFX 2 toca usar un método alternativo. Este método es explicado en la sección 4.3.1 Integración de tecnologías basada en Servicios Web SOAP.
- ❖ JavaFX 2 es una tecnología que requiere la máquina virtual de Java y se ejecuta como un applet en el navegador. [40] Los applets son más lentos de procesar y tienen espacio muy delimitado en la página web donde se ejecutan, es decir, no se mezclan con todos los componentes de la página web ni tienen acceso a ellos. El hecho de que los applets se descarguen en el cliente para ejecutarse, lo que se denomina cliente pesado, hace la aplicación web dependiente de la velocidad de procesamiento de la máquina del usuario. [44]

3.3.4. JAVA FX SCENE BUILDER

JavaFX Scene Builder es una herramienta bastante útil para los programadores de interfaces JavaFX 2 que construyen aplicaciones en Java. Esta utilidad se puede usar para generar la interfaz de una aplicación en forma visual, sin necesidad de escribir código.

Solo es necesario arrastrar y soltar los componentes deseados, editar sus propiedades y este software genera un archivo .FXML de manera automática en segundo plano. Dicho archivo contiene la descripción de la interfaz gráfica a través de lenguaje XML. [27]

Estas son algunas de las características principales de "JavaFX Scene Builder": [27]

- ❖ El diseño generado es fácil de usar.
- ❖ El software permite obtener una vista previa del diseño realizado.
- ❖ Admite formato CSS.

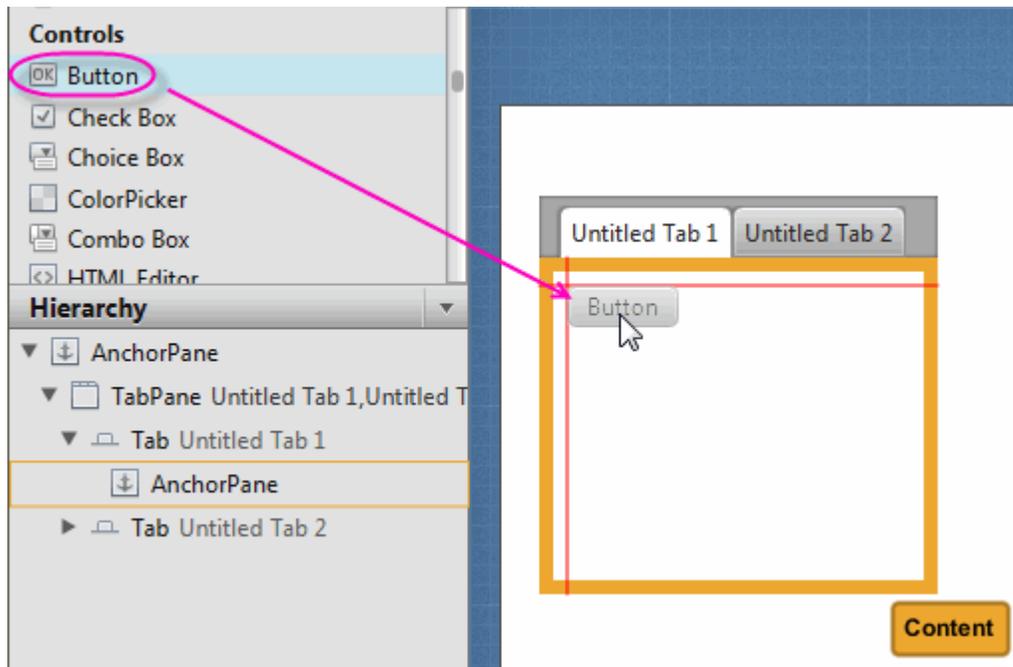


Ilustración 6: Captura de pantalla del software JavaFX Scene Builder

3.3.5. JPA

JPA 1.0 fue creado con Java EE 5 para lograr la persistencia automática de entidades de negocio. Proporciona un modelo de persistencia para transformar de objetos a tablas relacionales en bases de datos relacionales. [5]

En java EE 6 [1], JPA 2.0 sigue el mismo camino de simplicidad y robustez y agrega nuevas funcionalidades como las siguientes: [3]

- ❖ Propiedades de conexión con la base de datos más estandarizadas: algunas propiedades de acceso a la base de datos son usadas por todos los proveedores en el JDBC.
- ❖ Relaciones mejoradas: hay nuevos modos de relación entre dos entidades.
- ❖ JPQL mejorado.
- ❖ API Criteria que es el equivalente a JPQL en funcionalidad.

JPA 2 se puede utilizar para acceder y manipular datos relacionales desde Enterprise Java Beans (EJBs), componentes web y aplicaciones Java SE.

JPA 2 es una abstracción que está por encima de JDBC lo que permite ser independiente de SQL. [5]

3.3.6. EJB

Los JavaBeans empresariales (Enterprise JavaBeans EJB) están diseñados para desarrollo y despliegue de aplicaciones (distribuidas) de negocio basadas en componentes del lado del servidor. [5]

Con esta tecnología es posible desarrollar aplicaciones empresariales sin tener que crear de nuevo los servicios de transacción, seguridad, persistencia, concurrencia y lo que se pueda necesitar en el proceso de creación de una aplicación; permitiendo a los desarrolladores enfocarse en la implementación de la lógica de negocio. [5]

3.3.7. GLASSFISH

GlassFish Server es un servidor de aplicaciones de referencia para los proveedores de Java EE 6 [1] y listo para la producción, que se ha desarrollado con una licencia de código abierto.

GlassFish 3 proporciona una base con todas las funciones para la implementación de Java EE 6. [1]

Otras de las características que hacen que Glassfish 3 haya sido el servidor de aplicaciones elegido para el presente proyecto son las siguientes: [28]

- ❖ GlassFish es un proyecto Open Source modular que permite incluir sus librerías como parte de otros frameworks, toolkits y productos.
- ❖ Tiene soporte comercial disponible.
- ❖ Glassfish es de fácil uso y no consume demasiados recursos.
- ❖ Soporta aplicaciones web desarrolladas bajo código Java de gran complejidad, en las que se necesitan alta escalabilidad y disponibilidad, haciendo uso exhaustivo de las tecnologías de Java EE.
- ❖ Alto desempeño.
- ❖ Orientado a arquitectura de servicios: en el servidor de Glassfish se puede implementar servicios web.
- ❖ GlassFish es la implementación de referencia (RI) de Java EE.

- ❖ GlassFish es la base de código de las distribuciones estables, certificadas y con opción de contratar soporte y mantenimiento del Servidor de Aplicaciones de Oracle.
- ❖ Administración centralizada, permite gestionar clúster de servidores de aplicaciones desde una consola centralizada.
- ❖ GlassFish tiene asociada una comunidad que contribuye mediante código, detección de bugs, foros de discusión, feedback, documentación, wikis, blogs y otros medios a la difusión y éxito del proyecto. La comunidad GlassFish colabora en la compatibilidad de otros frameworks JAVA Open Source.

3.3.8. POSTGRESQL

PostgreSQL es un sistema de base de datos relacional. Fue pionera en muchos conceptos implementados en algunas soluciones comerciales. Por su licencia libre, PostgreSQL puede ser utilizado, modificado, y distribuido gratuitamente para cualquier propósito, sea privado, comercial, o académico. [38]

PostgreSQL es un intérprete del lenguaje SQL y compatible en mayor parte del estándar SQL.

Ofrece las siguientes características: [38]

- ❖ Consultas complejas
- ❖ Integridad referencial
- ❖ Triggers
- ❖ Vistas
- ❖ Control de concurrencia
- ❖ Propiedades ACID

Tiene a sus espaldas más de 10 años de desarrollo, convirtiéndose así en una de las soluciones de código abierto más avanzadas del mundo.

En el presente proyecto se podía usar cualquier otro motor de base de datos. Hay flexibilidad para cambiarlo en cualquier momento gracias a la estandarización de JPA 2.

3.3.9. SERVICIOS WEB SOAP

Los Servicios Web SOAP fue la tecnología elegida en este proyecto para integrar JavaFX 2 con la capa de negocio de la arquitectura Java EE 6, EJB.

¿Qué es un Servicio Web? Un Servicio Web es un componente de software que presta servicios a otras aplicaciones codificando los mensaje en lenguaje XML y enviando estos mensaje a través de protocolos estándares de Internet tales como el HTTP. [31]

En el siguiente gráfico puede observar el modo de operación de un servicio web basado en SOAP:

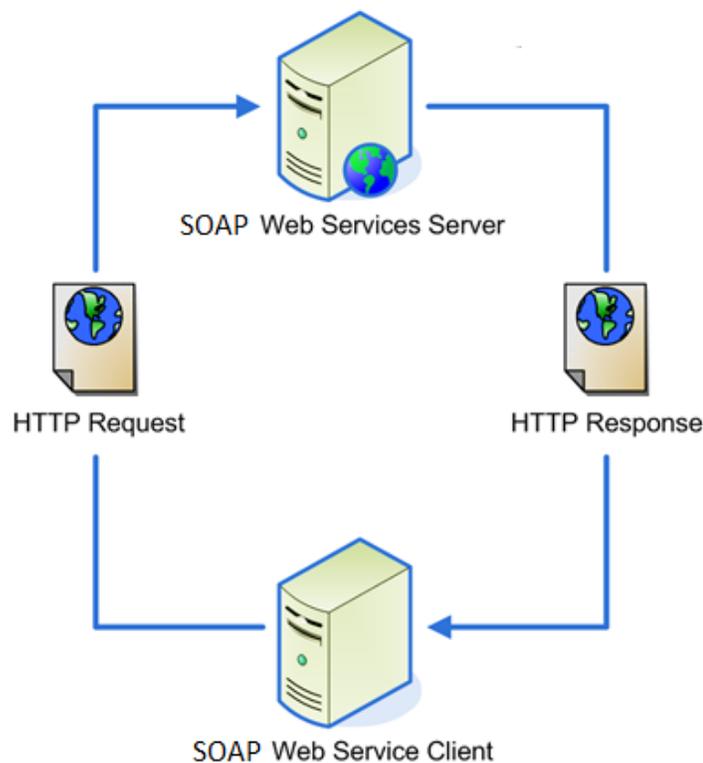


Ilustración 7: Modo de funcionamiento de un Servicio Web SOAP

Un servicio web es similar a un sitio web que no cuenta con una interfaz de usuario y que da servicios a las aplicaciones en vez de a las personas. Un servicio web, en vez de obtener solicitudes desde el navegador y retornar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en lenguaje XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en lenguaje XML. [31]

SOAP es uno de los protocolos de comunicación existentes para el intercambio de mensajes en los Servicios Web.

Un mensaje SOAP es similar a una carta: es un sobre que contiene una cabecera con la dirección del receptor del mensaje, un conjunto de opciones de entrega (la información de encriptación), y un cuerpo o body con la información o data del mensaje. [31]

Los servicios Web SOAP proporcionan una forma de describir las interfaces con el suficiente detalle para permitir a los usuarios construir una aplicación que se comunique con ellos. Esta descripción se proporciona normalmente en un documento XML llamado WSDL (Web Services Description Language) [31]

4. DESARROLLO DEL TRABAJO

En esta sección fueron plasmados los artefactos generados luego del análisis y el diseño de las reglas de negocio que se estimaron en el gestor de clubes de matemáticas: CLUBMAT.

4.1. PROBLEMÁTICA

La aplicación web construida en el presente trabajo de grado busca resolver un problema presente en el programa social PROSOFI de la Pontificia Universidad Javeriana. El problema consiste en que no existe un sistema de información que administre toda la información que gira en torno a los clubes de matemáticas. La aplicación web CLUBMAT es un sistema de información de los clubes de matemáticas bajo el marco de PROSOFI.

4.1.1. ¿QUÉ ES CLUBMAT?

CLUBMAT es una aplicación web que no enseña matemáticas a los estudiantes. CLUBMAT consiste en un sistema de información en el cual se administrará toda la información que gira entorno a los clubes de matemáticas.

A continuación encontrara un listado de las funciones principales de la aplicación web CLUBMAT:

- ❖ En CLUBMAT se registrarán los datos de los clubes de matemáticas con su respectiva institución e interlocutor.
- ❖ CLUBMAT contará con un sistema de noticias en donde un administrador de comunidades podrá escribir novedades acerca de los clubes de matemáticas. Estas noticias saldrán listadas en la interfaz gráfica inicial de la aplicación web.
- ❖ En CLUBMAT un administrador de comunidades podrá controlar el cumplimiento de las reglas de creación de un club de matemáticas.
- ❖ En CLUBMAT un administrador de un club podrá declarar una regla de creación de clubes como satisfecha. Quien valida si un club cumple realmente una regla de creación es el administrador de comunidades.
- ❖ En CLUBMAT un administrador de un club podrá registrar los estudiantes que participarán en una olimpiada matemática.

- ❖ En CLUBMAT un administrador de comunidades puede crear una olimpiada matemática, en donde participan uno o más clubes.
- ❖ En CLUBMAT habrá un banco de preguntas. Estas preguntas las deberán resolver los estudiantes en una olimpiada matemática.

4.2. ANÁLISIS Y DISEÑO DEL SISTEMA DE INFORMACIÓN: CLUBMAT

En esta sección están plasmados los artefactos generados luego de hacer el análisis y el diseño de la aplicación web CLUBMAT: Gestor de clubes de matemáticas.

En la presente sección se describen:

- ❖ Los requerimientos del sistema agrupados por módulos.
- ❖ La descripción de cada uno de los módulos de la aplicación web.
- ❖ Los casos de uso de la aplicación agrupados por módulos.
- ❖ Descripción de los roles planteados para la aplicación web.
- ❖ Diagrama de entidades de negociación para la aplicación web.

4.2.1. REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN AGRUPADO POR MÓDULOS

A continuación se listan todos los requerimientos que fueron identificados para el gestor de club de matemáticas: CLUBMAT.

Los requerimientos se encuentran agrupados por módulos.

La siguiente tabla es un resumen de los requerimientos funcionales del formato Hacer & Usos: (Para ver más detalles ver anexo Requerimientos del sistema (Hacer & Usos))

Modulo	Id Requerimiento	Especificación del requerimiento
--------	------------------	----------------------------------

Módulo de Seguridad	R01	El sistema debe proveer un mecanismo de gestión de usuarios (CRUD de Usuarios), que permita ver los usuarios, ingresar uno nuevo, editar o eliminar un usuario seleccionado.
	R02	El sistema debe proveer un mecanismo de gestión de roles (CRUD de Roles), que permita ver rol por usuario, ingresar un nuevo rol a un usuario, editar o eliminar un rol a un usuario seleccionado.
	R04	El sistema mostrará al usuario en su menú solo las opciones que el mismo tiene derecho según su rol
	R15	El sistema debe permitir a un usuario registrado iniciar sesión en el sistema
	R06	El sistema debe permitir que un usuario cierre su sesión.
	R21	El sistema debe permitir que un usuario invitado pueda auto registrarse y ser un nuevo administrador de club de matemáticas
	R20	El sistema debe permitir que un usuario "Líder" pueda aprobar el auto registro de un usuario invitado
Módulo de Clubes	R05	El sistema debe proveer un mecanismo de gestión de clubes de matemáticas (CRUD de los clubes de matemáticas), que permita listar los clubes existentes, ingresar uno nuevo, editar o eliminar un club seleccionado.
	R07	El sistema debe proveer un mecanismo de gestión de noticias de clubes de matemáticas (CRUD de noticias), que permita listar las noticias existentes, ingresar una nueva, editar o eliminar una noticia seleccionada.
	R08	El sistema debe proveer un mecanismo de gestión de pasos (reglas) para la creación de clubes de matemáticas (CRUD de pasos), que permita listar los pasos de creación existentes, ingresar uno nuevo, editar o eliminar un paso seleccionado.

	R09	El sistema debe proveer un mecanismo de gestión de olimpiadas (CRUD de olimpiadas), que permita listar las olimpiadas existentes, ingresar una nueva, editar o eliminar una olimpiada seleccionada.
	R10	El sistema debe proveer un mecanismo de gestión de estudiantes en olimpiadas (CRUD de estudiantes), que permita listar los estudiantes en olimpiada existentes, ingresar uno nuevo, editar o eliminar un estudiante de una olimpiada seleccionada.
	R13	El sistema debe proveer un mecanismo de gestión de instituciones (CRUD de instituciones), que permita listar las instituciones existentes, ingresar una nueva, editar o eliminar una institución seleccionada.
	R14	El sistema debe proveer un mecanismo de gestión de preguntas para olimpiadas (CRUD de preguntas), que permita listar las preguntas existentes, ingresar una nueva, editar o eliminar una pregunta seleccionada.
	R18	El sistema debe proveer un mecanismo de gestión de responsables de una olimpiada (CRUD de responsables), que permita listar los responsables existentes, ingresar un nuevo, editar o eliminar un responsable seleccionado.
	R11	El sistema debe desplegar las noticias de los clubes de matemáticas en la pantalla principal
	R12	El sistema debe calcular la evolución del club de matemáticas
	R16	El sistema debe permitir a un "Administrador Club" ver una lista de chequeo con los pasos para crear su club de matemáticas
	R17	Permitir seleccionar como ya realizado un paso para la creación de un club de matemáticas
	R19	Un administrador debe poder inscribir un club de matemáticas a una olimpiada

	R23	El sistema debe permitir generar una prueba para una olimpiada matemática
	R24	El sistema debe permitir calificar una prueba de una olimpiada matemática

Tabla 8: Requerimientos agrupados por módulos

A continuación se listan los requerimientos no funcionales identificados para la aplicación web CLUBMAT:

Id Requerimiento	Especificación del requerimiento
R20	El sistema debe tener un manual de usuario y un manual de mantenimiento
R21	El sistema debe desplegar la información en idioma español
R22	El sistema debe poder ejecutarse en cualquier navegador de internet
R03	El sistema debe garantizar las propiedades ACID en todas las transacciones de datos.
R25	El sistema debe ser fácil de usar
R26	El sistema debe ser escalable

Tabla 9: Requerimientos no funcionales

Para ver los requerimientos de software funcionales y no funcionales con un mayor detalle (justificación, alcance, prioridad, etc.) debe ir al anexo de 8.1 Requerimientos del sistema.

4.2.2. MÓDULOS

Los módulos del sistema de información para la gestión de clubes de matemáticas son:

MÓDULO	DESCRIPCIÓN
Módulo de seguridad	El módulo de seguridad presta todos los servicios a los usuarios que controlan cuales usuarios pueden acceder a la aplicación web, la gestión de roles y la autenticación de usuarios.

Módulo de clubes	Brinda todos los servicios asociados a los clubes de matemáticas. Estos servicios son: El registro, actualización, despliegue y eliminación de un club de matemáticas, un sistema de noticias, la gestión de las olimpiadas, el seguimiento de las reglas de creación por cada club, un banco de preguntas y generación y calificación de pruebas.
------------------	--

Tabla 10: Módulos del sistema de información CLUBMAT

4.2.3. CASOS DE USO AGRUPADOS POR MÓDULOS

A continuación encuentra todos los casos de uso del sistema: CLUBMAT, agrupados por módulos.

Modulo	Caso de uso	Nombre del Caso de Uso
Módulo de Seguridad	CU08	CRUD de roles
	CU18	CRUD de usuarios
	CU16	Identificarse como un usuario
	CU14	Cerrar la sesión del sistema
	CU01	Auto-registrarse como administrador de un club de matemáticas
	CU17	Aprobar el auto-registro de un nuevo administrador
Módulo de Clubes	CU03	CRUD de clubes de matemáticas
	CU13	CRUD de noticias de los clubes de matemáticas

	CU05	CRUD de reglas para crear un club de matemáticas
	CU20	CRUD de olimpiadas
	CU12	CRUD de estudiantes en una olimpiada
	CU02	CRUD de instituciones
	CU07	CRUD de preguntas para olimpiadas
	CU06	CRUD de los responsables de una olimpiada
	CU15	Ver las noticias de los clubes de matemáticas en la página principal de la aplicación.
	CU04	Ver el estado de avance de creación del club de matemáticas
	CU09	Validar la realización de una regla
	CU10	Inscribir mi club de matemáticas a una olimpiada
	CU19	Generar una prueba para una olimpiada
	CU21	Calificar la prueba de una olimpiada

Tabla 11: Inventario de módulos y casos de uso

El siguiente es el diagrama de casos de uso generado para la aplicación web: CLUBMAT. En este puede observar los actores asociados a cada caso de uso.

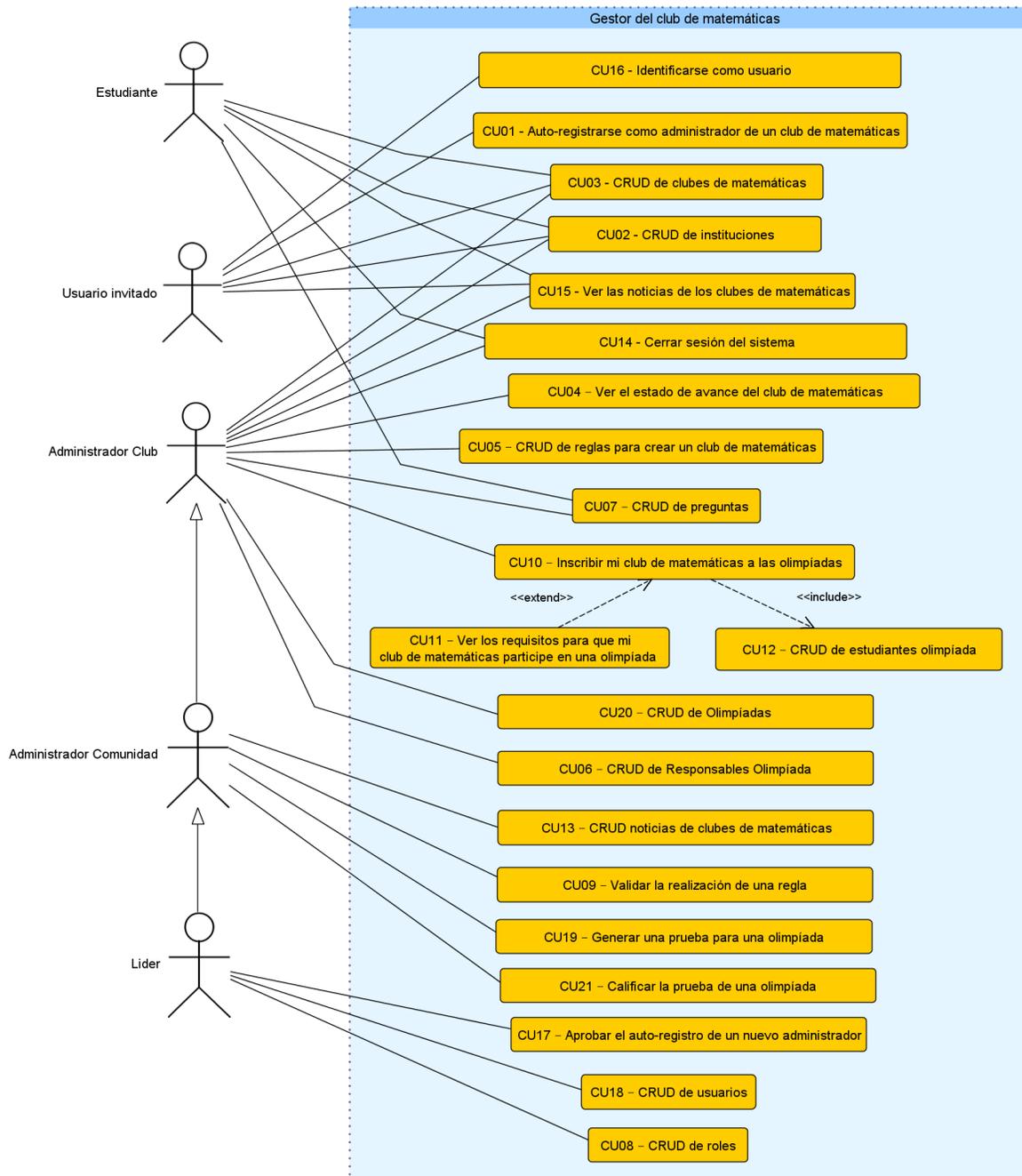


Ilustración 8: Diagrama de casos de uso para la aplicación web: CLUBMAT

Si desea observar con mayor detalle los casos de uso agrupados por modulo (requerimientos asociados y complejidad), por favor vea el anexo 8.2 Inventario de módulos y casos de uso.

4.2.4. DESCRIPCIÓN DE LOS ROLES

A continuación se explica detalladamente los roles definidos en el sistema CLUBMAT:

NOMBRE DEL ACTOR	DESCRIPCIÓN
Usuario invitado	Representa un usuario "desconocido" del gestor de clubes de matemáticas. Es decir, es una persona que acaba de ingresar a la aplicación web sin registrarse y el sistema no lo puede identificar.
Administrador de club de matemáticas (Administrador Club)	Representa un usuario del gestor de clubes de matemáticas registrado. Se identifica con un nombre de usuario y contraseña. Al registrarse el usuario toma el rol de "Administrador de un club de matemáticas" y puede registrar, crear o eliminar su propio club de matemáticas, junto a su respectiva institución. Este usuario será el interlocutor del club de matemáticas que registre. También puede usar los CRUD de preguntas y estudiantes-olimpiada.
Administrador de las comunidades de clubes de matemáticas (Administrador Comunidad)	Representa un usuario que puede realizar todas las acciones que puede hacer un Administrador de clubes y también puede: <ul style="list-style-type: none"> ❖ Usar el CRUD de noticias. ❖ Validar el estado de una regla para la creación un club de matemáticas. ❖ Usar el CRUD de reglas de creación de clubes de matemáticas completamente. ❖ Usar el CRUD de olimpiadas completamente. ❖ Usar el CRUD de responsables olimpiada completamente.

Líder	<p>Es un súper-administrador. Representa un usuario que puede realizar todo lo que puede hacer un Administrador de comunidades y también puede:</p> <ul style="list-style-type: none">❖ Usar los CRUD de usuarios y roles completamente❖ Usar los CRUD de instituciones y clubes completamente.❖ Aprobar el auto-registro de nuevos usuarios.
Estudiante	<p>Representa un estudiante de un club de matemáticas que va participar en una olimpiada en donde responderá un conjunto de preguntas.</p>

Tabla 12: Descripción de los roles

4.2.5. DIAGRAMA DE ENTIDADES DE NEGOCIO

Para cumplir con los casos de uso y requerimientos del “Gestor de clubes de matemáticas: CLUBMAT”, se realizó el siguiente modelamiento de entidades:

Rol	Representa los roles registrados en el sistema.
Usuario	Es el interlocutor de un club de matemáticas. Él es la raíz del sistema pues registra un club de matemáticas. El usuario solo puede tomar un rol de los descritos en la sección 4.2.4 Descripción de los Roles.
Noticia	Representa las noticias del gestor de clubes de matemáticas. Si el usuario tiene asignado el rol de “Administrador comunidad” puede escribir una o más noticias sobre los clubes, las cuales serán desplegadas en la pantalla principal de la aplicación web.
Institución	Representa las instituciones registradas en la aplicación web. Un usuario puede registrar una sola institución.
ClubMatematicas	Representa los clubes de matemáticas registrados en la aplicación web. Un usuario puede registrar un club de matemáticas, asociado a la institución que él haya registrado, convirtiéndose en el administrador de ese club de matemáticas.
ReglaCreacion	Representa un paso o una regla que deben cumplir los clubes de matemáticas para crearse.
Olimpiada	Representa una olimpiada matemática registrada en la aplicación web.
Estudiante	Representa los estudiantes de un club de matemáticas que van a participar en una olimpiada.

Responsable	Representa los responsables de una olimpiada matemática. Cada uno tiene una responsabilidad que debe ejecutarse en orden para que la olimpiada se realice exitosamente.
ReglamentoOlimpiada	Reglamento que debe cumplir una o más olimpiadas para que puedan realizarse.
Pregunta	Se diseñó un banco de preguntas, en donde puede haber una o más preguntas creadas por cada usuario. Esas preguntas serán realizadas a los estudiantes en una o más olimpiadas.

Tabla 13: Descripción de las entidades para la aplicación web CLUBMAT

4.3. ARQUITECTURA DE LA APLICACIÓN WEB CLUBMAT

CLUBMAT fue una aplicación web diseñada y construida teniendo en cuenta la arquitectura multicapa de la arquitectura Java EE 6 [1], la cual se compone de una capa de presentación (JavaFX 2) [7], capa de servicios de negocio (EJBs), capa de dominio (Entidades persistentes JPA) y capa de persistencia (Bases de datos).

Para la integración de las capas mencionadas hay distintas alternativas posibles que fueron analizadas en el presente proyecto:

- ❖ **RedFX:** Es una librería que permite a las aplicaciones JavaFX 2 intercambiar datos con un servidor Java EE 6 sin necesidad de escribir grandes cantidades de código para lograrlo. Esta librería se basa en el protocolo REST para la comunicación. Lamentablemente no se usó en el presente proyecto porque no se encontró documentación suficiente de la misma.

- ❖ **GraniteDS:** Es un conjunto de librerías que tienen como objetivo el intercambio de datos entre un cliente JavaFX 2 y La arquitectura Java EE 6. Ofrece un alto rendimiento, integración con distintos tipos de servidores de aplicaciones (Glassfish, JBoss, Weblogic) y distintas herramientas de mapeo de persistencia como EclipseLink, Hibernate, OpenJPA, entre otros. También ofrece herramientas para generación automática de código. Para la integración de tecnologías se basa en Web Sockets. No se usó en el presente proyecto porque en el momento de la realización del mismo, la pila de tecnologías estándar (JavaFX 2, EJB 3, JPA 2, EclipseLink y Glassfish 3) no tenía la documentación suficiente.

- ❖ **Servicios web SOAP:** tecnología usada en el presente proyecto para la integración de JavaFX 2 con la capa de negocio EJB. Se decidió usar porque fue la única tecnología que realmente permitió una integración estable entre la capa de presentación JavaFX 2 y la capa de negocio de la arquitectura Java EE 6.

- ❖ **Servicios web REST:** Usando servicios web REST también se puede integrar el framework JavaFX 2 con la capa de negocio EJB de la arquitectura Java EE 6. En la aplicación web CLUBMAT no se usó esta tecnología porque el cronograma del proyecto no ofrecía el tiempo necesario para su estudio.

En el siguiente diagrama puede observar la arquitectura multicapa de la aplicación web CLUBMAT:

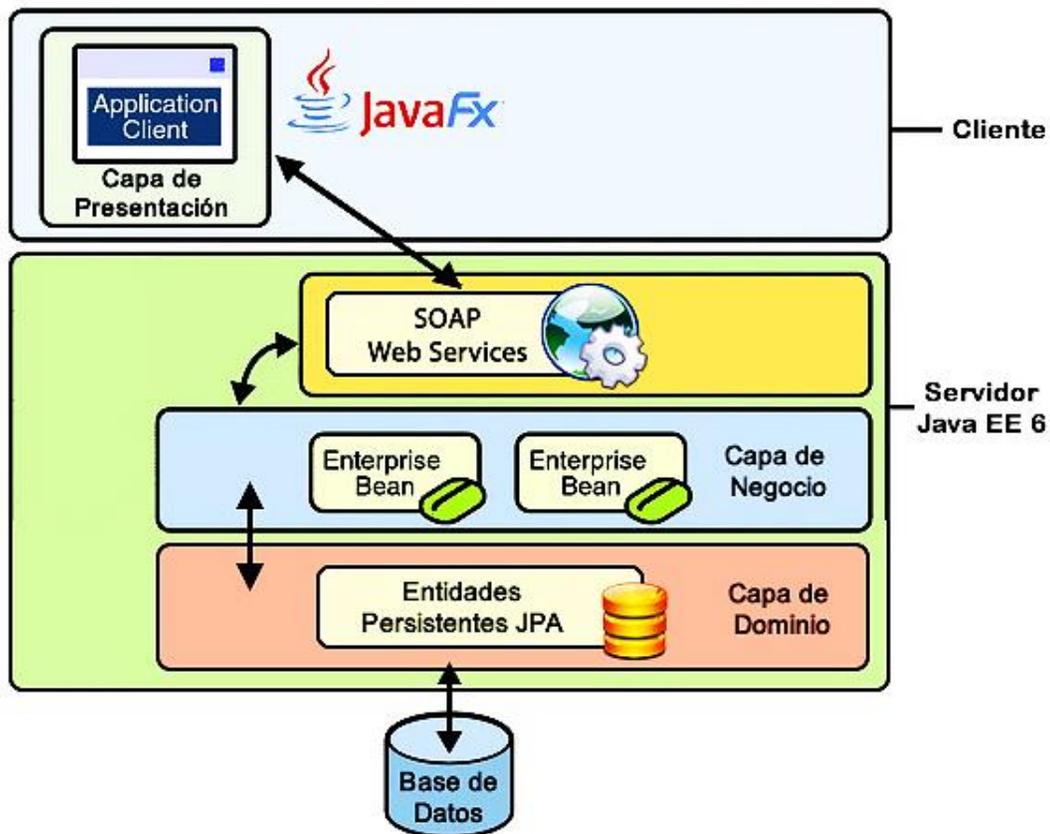


Ilustración 10: Arquitectura multicapa de la aplicación web CLUBMAT

Como se puede observar hay una capa de presentación, la cual está en el cliente usando el framework JavaFX 2 [7] en vez del framework JSF que es la tecnología estándar.

Es importante aclarar que al usar JavaFX 2 junto con la herramienta JavaFX Scene Builder 1.1 se debe usar un patrón Modelo Vista Controlador (MVC). [27] El modelo corresponde a la capa de negocio (EJB). La vista son todos los archivos FXML que definen la interfaz gráfica por medio de lenguaje XML. Estos archivos son generados automáticamente por JavaFX Scene Builder 1.1.

Finalmente, el controlador corresponde a clases que permiten manipular los elementos gráficos descritos en el archivo FXML, buscando realizar verificaciones, leer o escribir datos.

A continuación se encuentra un diagrama adicional para describir la arquitectura de la aplicación web CLUBMAT. En el se muestra los componentes del Modelo Vista Controlador usado.

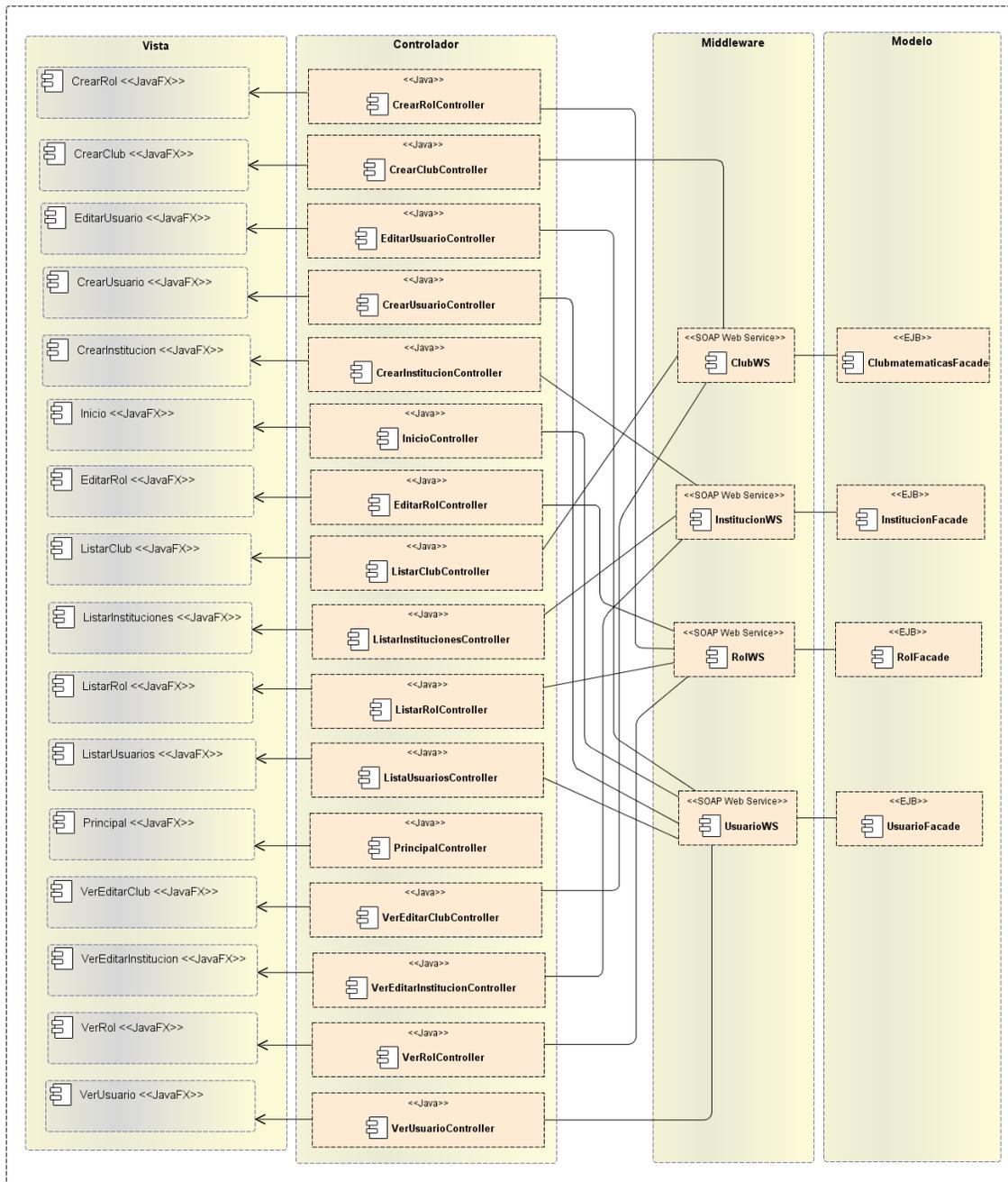


Ilustración 11: Arquitectura de la aplicación web CLUBMAT**4.3.1. INTEGRACIÓN DE TECNOLOGÍAS BASADA EN SERVICIOS WEB SOAP**

Los Servicios web SOAP (ver descripción de este concepto en la sección 3.3.9 Servicios web SOAP) fue la herramienta intermediaria usada para poder hacer la integración entre la capa de presentación (ubicada en el cliente) y la capa de negocio (ubicada en el servidor Java EE 6), es decir los Enterprise JavaBeans.

Por cada Enterprise JavaBean fue generado un Servicio Web SOAP el cual expone cada uno de los métodos del EJB para que sean accesibles desde el exterior. En el caso de este proyecto, para que el cliente JavaFX 2 pueda invocar los métodos de un EJB de manera remota.

A continuación se encuentra un diagrama que muestra todas las clases que intervienen en las distintas capas para el CRUD Club.

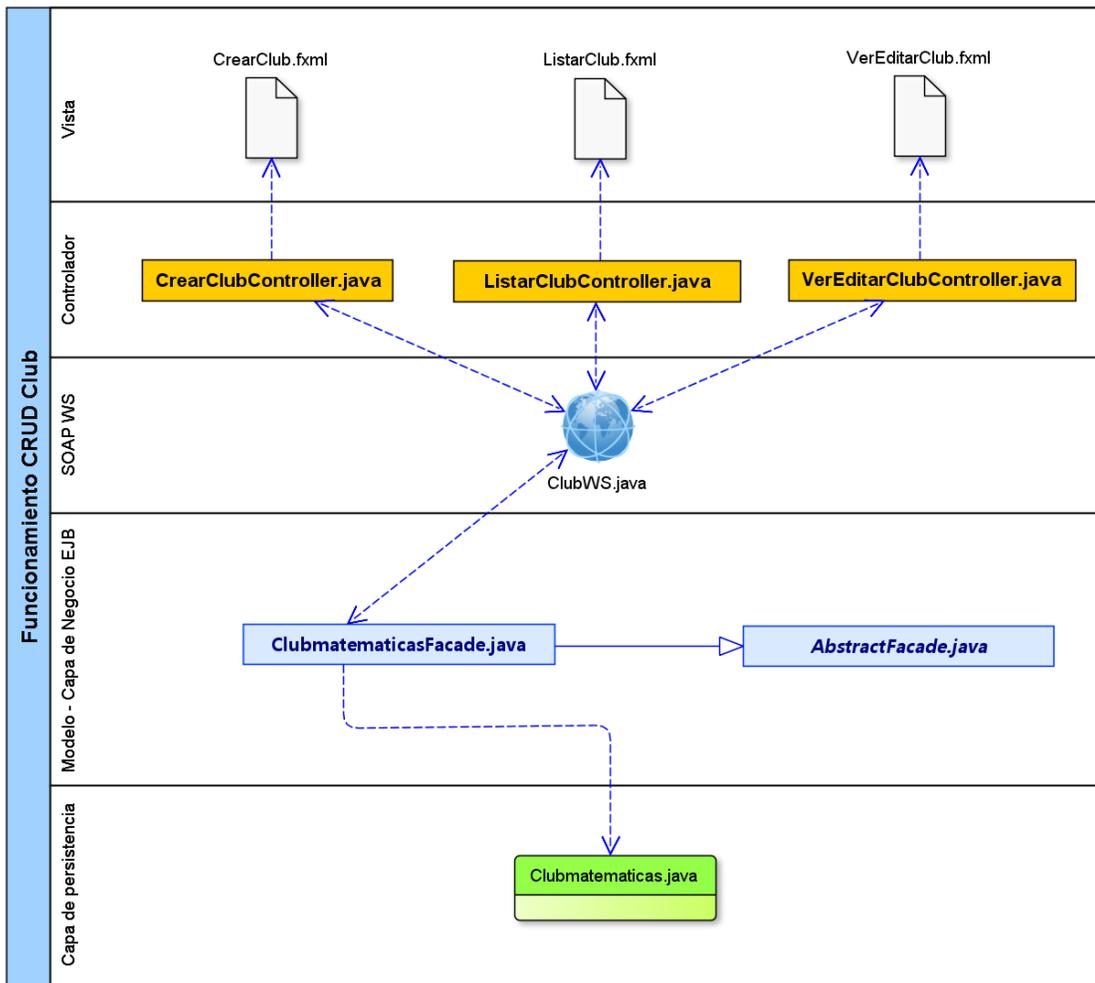


Ilustración 12: Diagrama para el CRUD Club

A continuación se encuentra un diagrama de secuencia en donde se detalla la acción de registrar un nuevo usuario por parte de un usuario con rol líder.

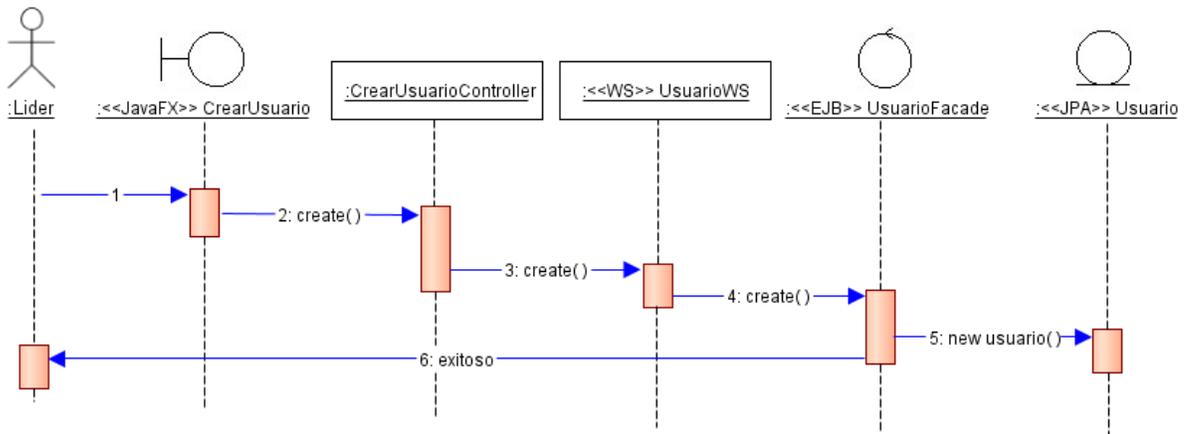
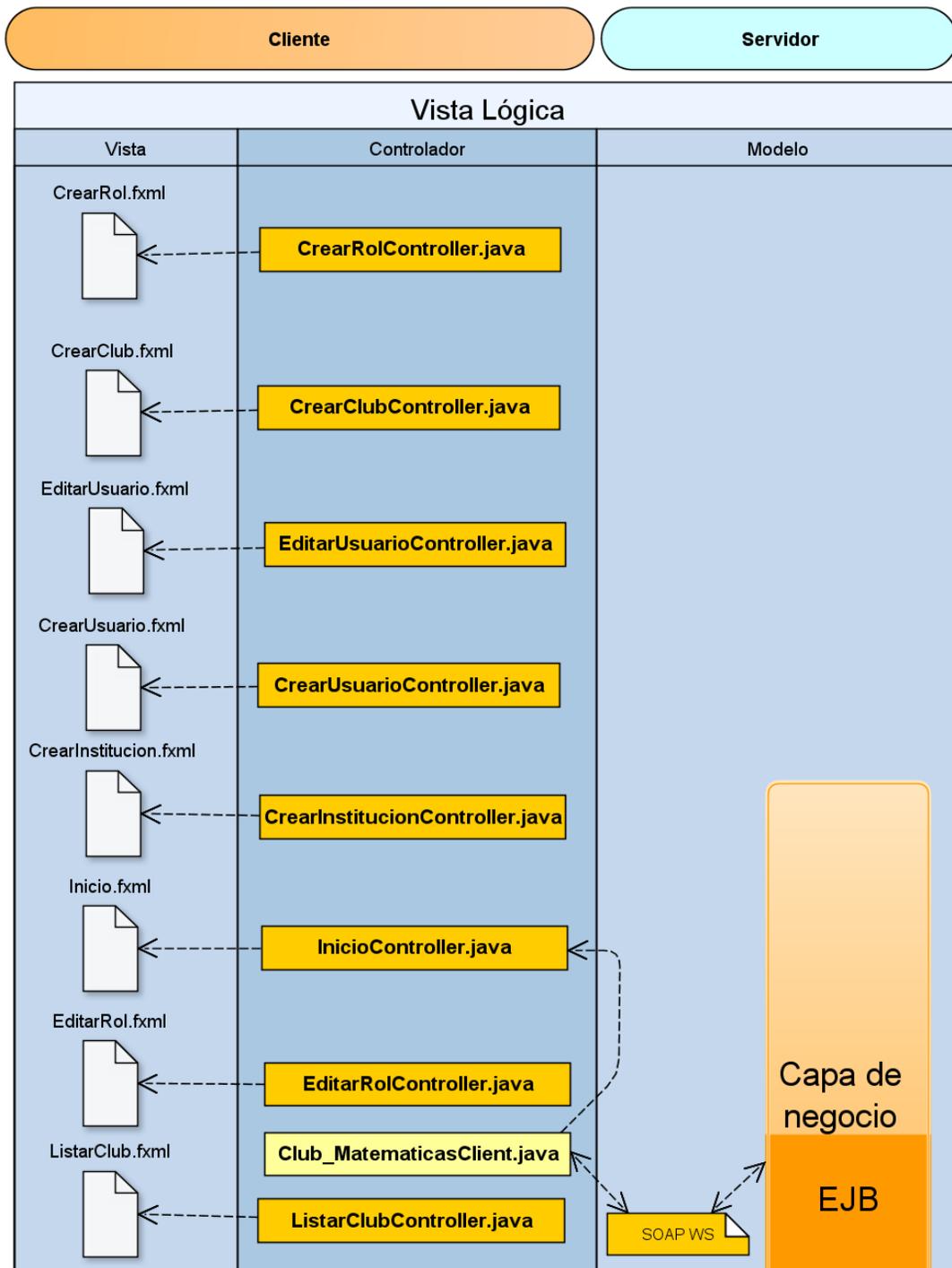


Ilustración 13: Diagrama de secuencia para crear un nuevo usuario

4.3.2. DESCRIPCIÓN DE LA CAPA DE PRESENTACIÓN HECHA CON JAVA FX 2

A continuación se presenta la vista lógica. Con ella se explica en detalle el Modelo Vista Controlador. También se detalla la capa de presentación hecha con JavaFX 2, la cual se conforma de una vista y un controlador.



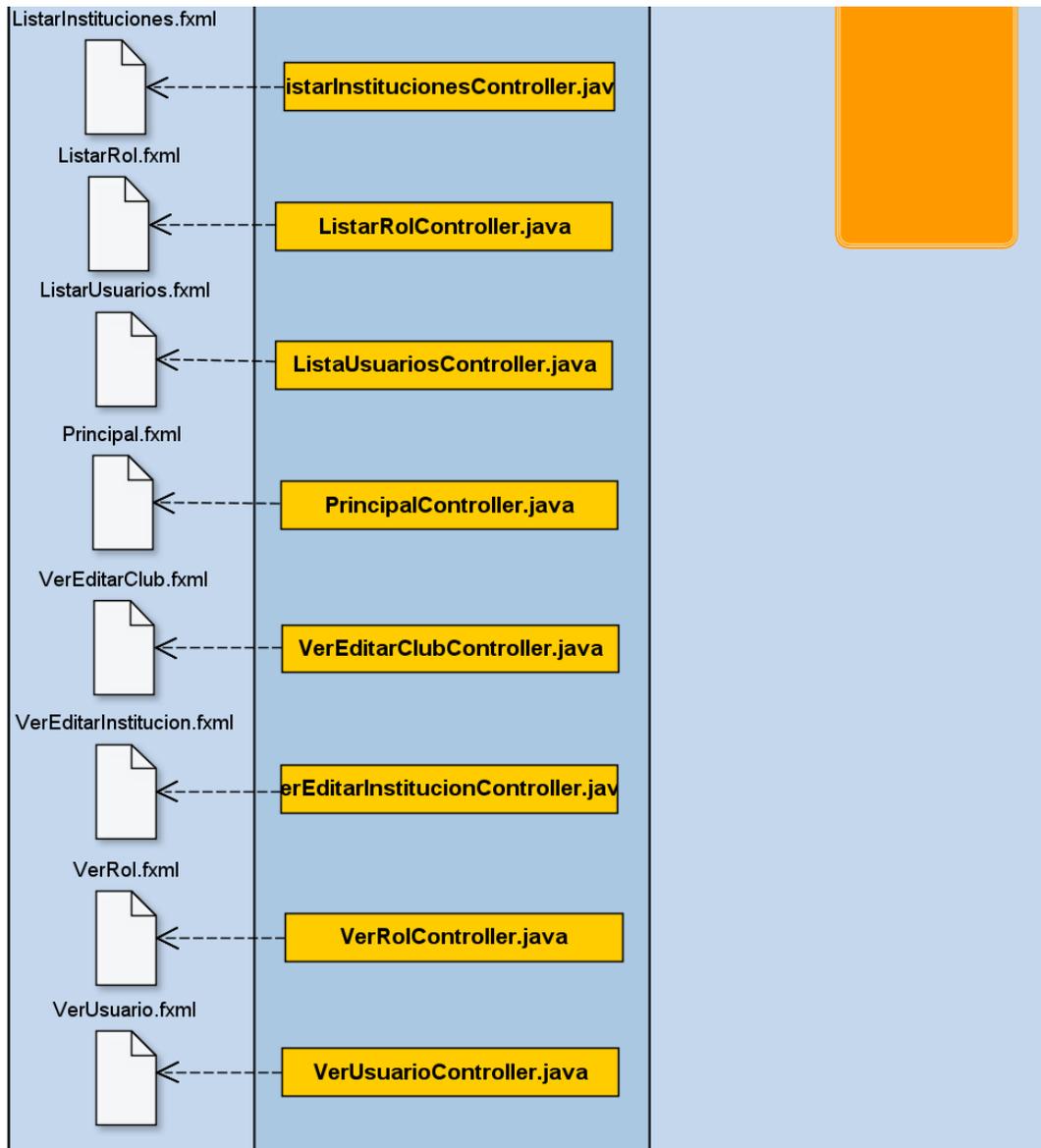


Ilustración 14: Vista lógica

Como se puede observar en el anterior diagrama, la vista y el controlador se encuentran ubicados en el cliente y constituyen la capa de presentación. Mientras que el modelo se encuentra en el servidor y es la capa de negocio (EJB).

La clase Club_MatematicasClient.java que hace parte del controlador es la que invoca los demás controladores. Cada controlador maneja una vista, por ejemplo, la clase CrearRolController.java maneja la interfaz gráfica descrita en la vista CrearRol.fxml.

El controlador se comunica con la capa de negocio (EJBs) de la arquitectura Java EE 6 a través de servicios web SOAP.

Para la gestión de las vistas (pantallas) fue necesario usar un framework que permitió manejarlas a través de un Modelo Vista Controlador.

Para utilizar el framework mencionado en este proyecto, fue necesario seguir paso a paso el manual del mismo, publicado en el blog de la ingeniera Ángela Caicedo en:

https://blogs.oracle.com/acaicedo/entry/manejando_multiples_pantallas_en_javafx.

También fue necesario ayudarse con un video tutorial sobre este framework, también realizado por la ingeniera Ángela Caicedo en la URL:

<http://www.youtube.com/watch?v=5GsdaZWDcdY>.

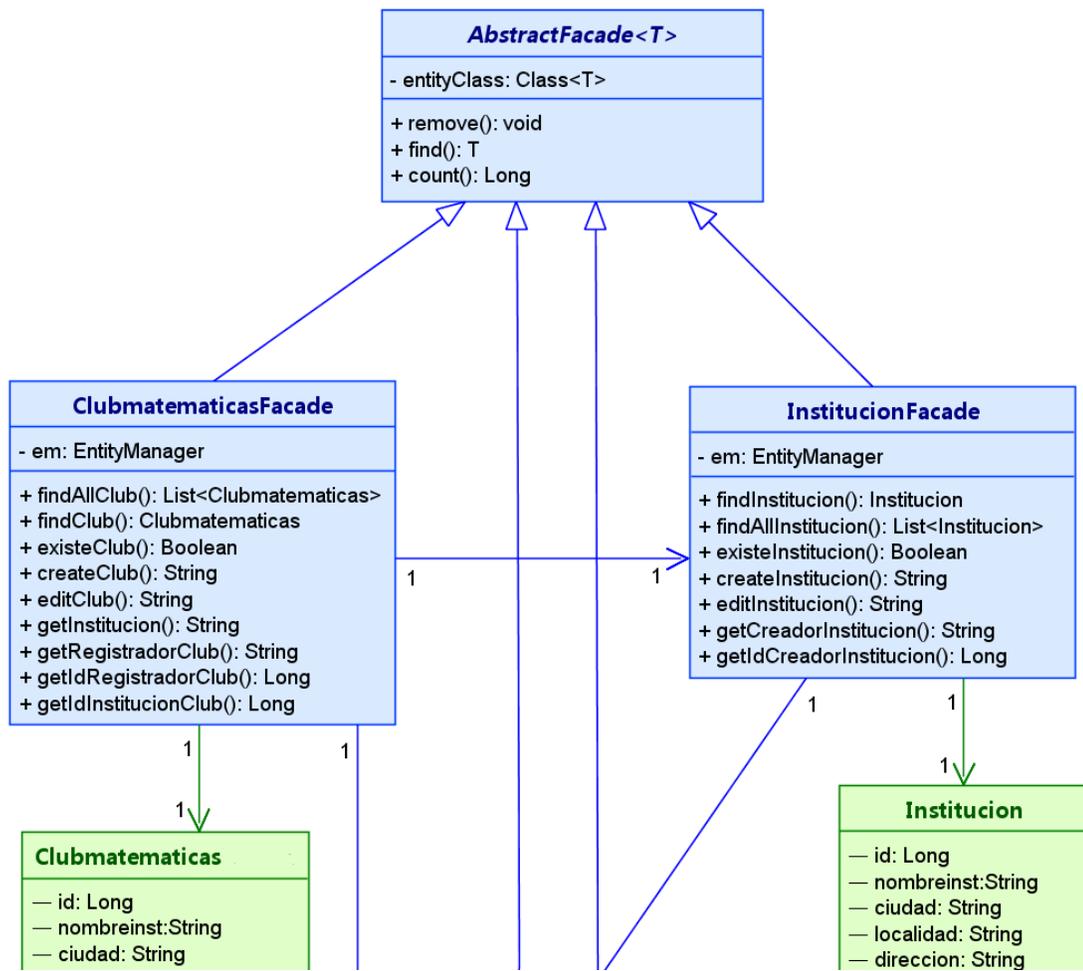
El framework de pantallas es de gran utilidad pues permite que por cada pantalla (vista FXML), exista un controlador. De esta manera tenemos la estructura de nuestro proyecto organizada y mantenible.

El framework de pantallas también nos permite navegar entre las vistas fácilmente. Navegación entre vistas significa que se pueda ir de una pantalla a otra, devolverse, actualizarla o mantenerla desactualizada de una manera fácil.

Para obtener un mayor detalle acerca del framework de pantallas ver el anexo 8.4 Manual de mantenimiento.

4.3.3. DESCRIPCIÓN DE LA CAPA DE NEGOCIO

A continuación se encuentra el diagrama de clases para la capa de negocio:



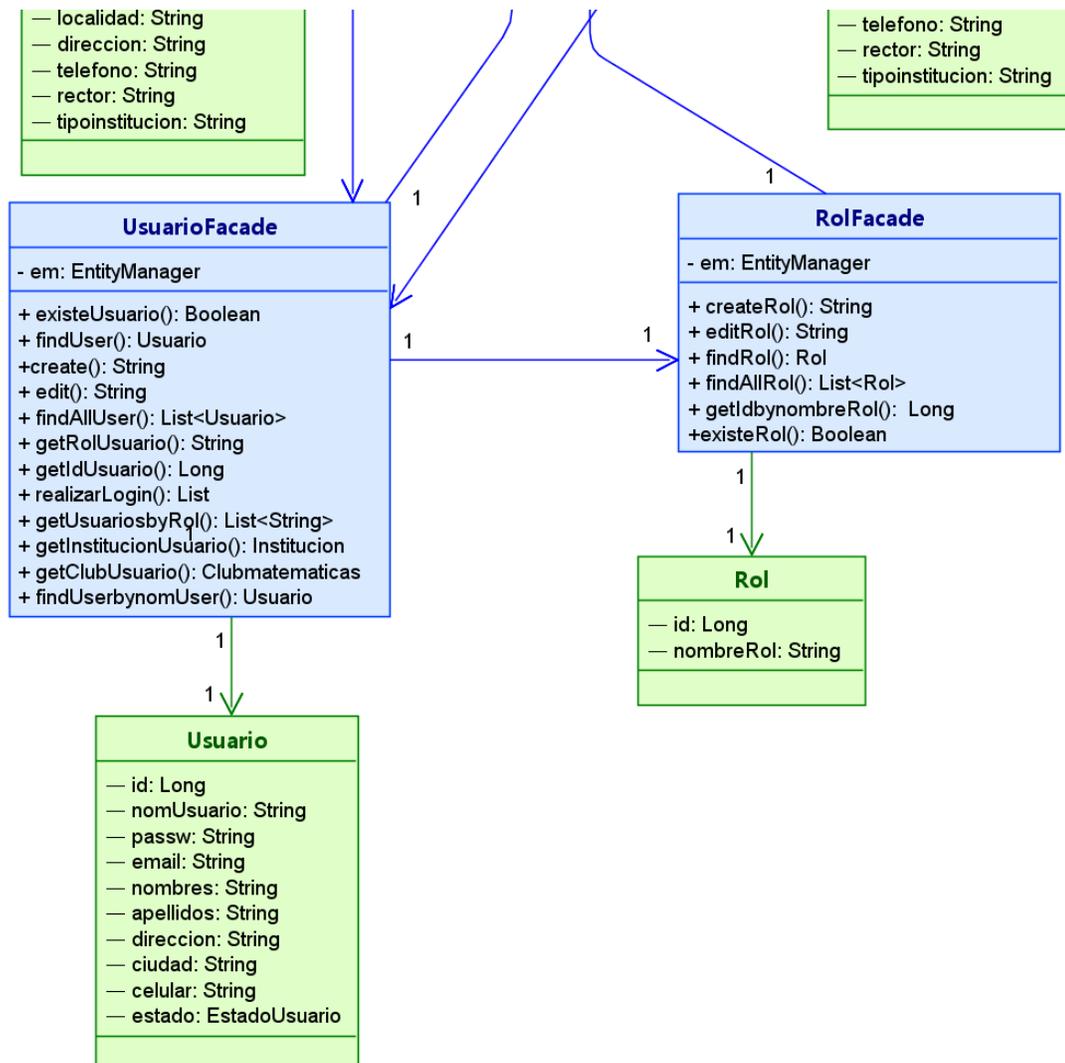


Ilustración 15: Diagrama de clases para la capa de negocio

Como puede observarse en el anterior diagrama, en la capa de negocio se hallan los Enterprise JavaBeans. Allí es donde se realizó la implementación de las reglas de negocio para las entidades: Rol, Usuario, Institución y Clubmaticas. Todos los EJB heredan de una clase abstracta llamada “AbstractFacade” que implementa algunos métodos genéricos funcionales para cualquier entidad.

Finalmente, hay una capa de dominio la cual está compuesta de las entidades persistentes JPA. Estas entidades facilitan la comunicación con la base de datos PostgreSQL. Es decir que la capa

de dominio es la que ayuda a la capa de negocio a extraer y colocar información en la base de datos. El diagrama de entidades de negocio contiene todas las entidades persistentes con sus atributos y relaciones.

4.4. CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

En esta sección se describe el proceso seguido para la construcción de la aplicación web CLUBMAT: Gestor de clubes de matemáticas y también un resumen de los pasos para desarrollar una aplicación JavaFX 2 integrada a los componentes de la arquitectura JavaEE 6.

a. PASOS PARA DESARROLLAR UNA APLICACIÓN JAVA FX 2 INTEGRADA A COMPONENTES DE JAVA EE 6

A continuación se encuentra los pasos resumidos para poder integrar JavaFX 2 con la capa de negocio de la arquitectura Java EE 6.

- 1) Crear una base de datos en postreSQL con pgAdmin III
- 2) Configurar Netbeans 7.3 para que pueda conectarse a la base de datos PostgreSQL creada en el paso anterior.
- 3) En Netbeans 7.3 generar un descriptor para la base de datos PostgreSQL que acabo de construir.
- 4) Crear un proyecto con una entidad JPA “Rol”.
- 5) La entidad Rol debe tener el siguiente código:

```
package co.edu.javeriana.clubmat.jpa;

import java.io.Serializable;
import javax.persistence.*;
import javax.validation.constraints.Size;

@Entity
@NamedQueries({
    @NamedQuery(name = "Rol.existRol", query = "SELECT
COUNT(r.nombrerol) FROM Rol r WHERE r.nombrerol = :nombrerol")})
public class Rol implements Serializable {
```

```
private static final long serialVersionUID = 1L;

@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private Long id;

@Size(min = 3)
@Column(unique = true, nullable = false, length = 40)
private String nombrerol;

public Rol() { }

public Rol( String rol ) {
    this.nombrerol = rol;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public String getNombrerol() {
    return nombrerol;
}

public void setNombrerol(String nombrerol) {
    this.nombrerol = nombrerol;
}

@Override
public String toString() {
    return "JPA.Rol[ id=" + id + " nombrerol=" + nombrerol + " ]";
}
}
```

- 6) Generar un EJB automáticamente a partir la entidad Rol creada. Se debe hacer click derecho sobre el modulo web del proyecto y usar la opcion “Sessions Beans for Entity Classes”.

- 7) El EJB “RolFacade” generado debe tener el siguiente código:

```
public Boolean existeRol( String rol )
{
    Query q = em.createNamedQuery( "Rol.existRol" );
    q.setParameter( "nombrerol", rol );
    Long exist = ( Long ) q.getSingleResult( );

    if( exist > 0 )
        return true;

    return false;
}

public String crearRol( String rol )
{
    if( existeRol( rol ) )
        return "El rol ya existe en el sistema, elija otro";

    Rol r = new Rol( rol );
    em.persist( r );

    return "exitoso";
}
```

- 8) Usar el asistente “Create Web Service from Existing Session Bean” para generar un servicio web a partir de un EJB existente, en este caso RolFacade.
- 9) Desplegar el proyecto Java EE 6 en Netbeans.
- 10) Crear un nuevo proyecto JavaFX 2 en Netbeans.
- 11) Luego de creado, hacer click derecho sobre el proyecto JavaFX 2 >> New >> Web Service Client, ingresar la URL del WDSL. Netbeans automáticamente generara un cliente JavaFX 2 para el servicio web SOAP.

- 12) Usar el asistente “Call Web Service Operation” haciendo click derecho en el controlador de la vista. Este asistente permite llamar un método remoto del servicio web SOAP.

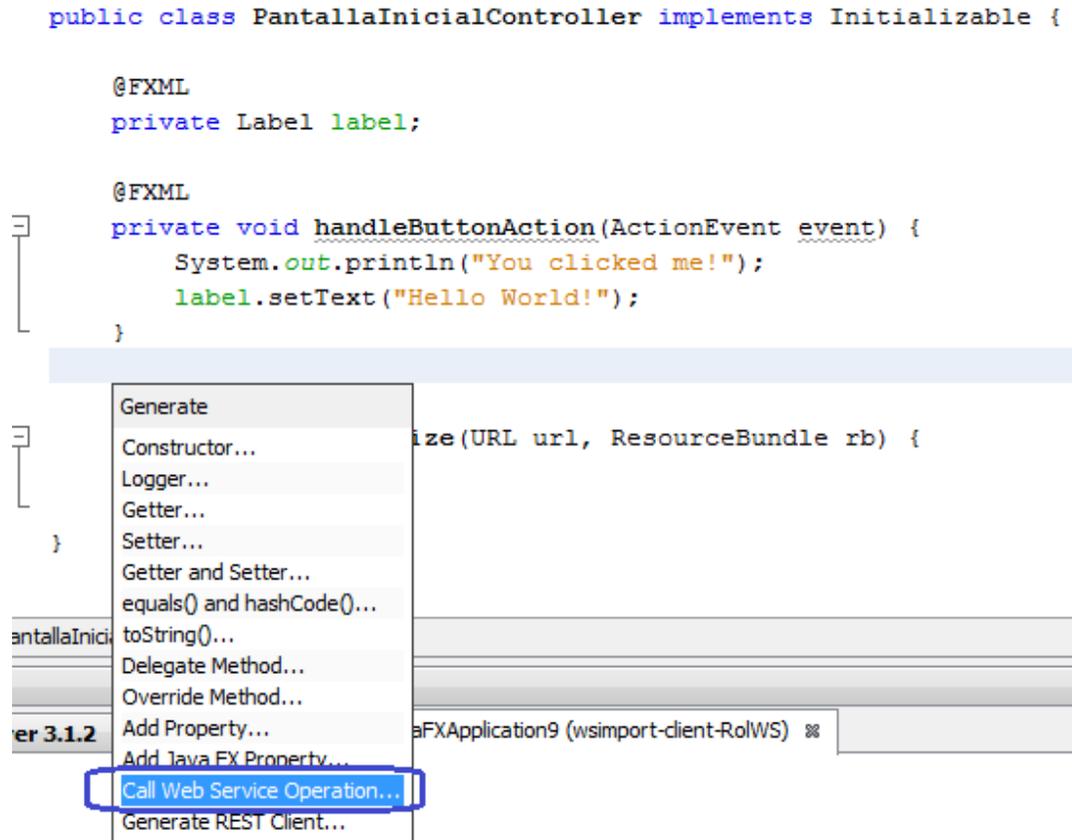
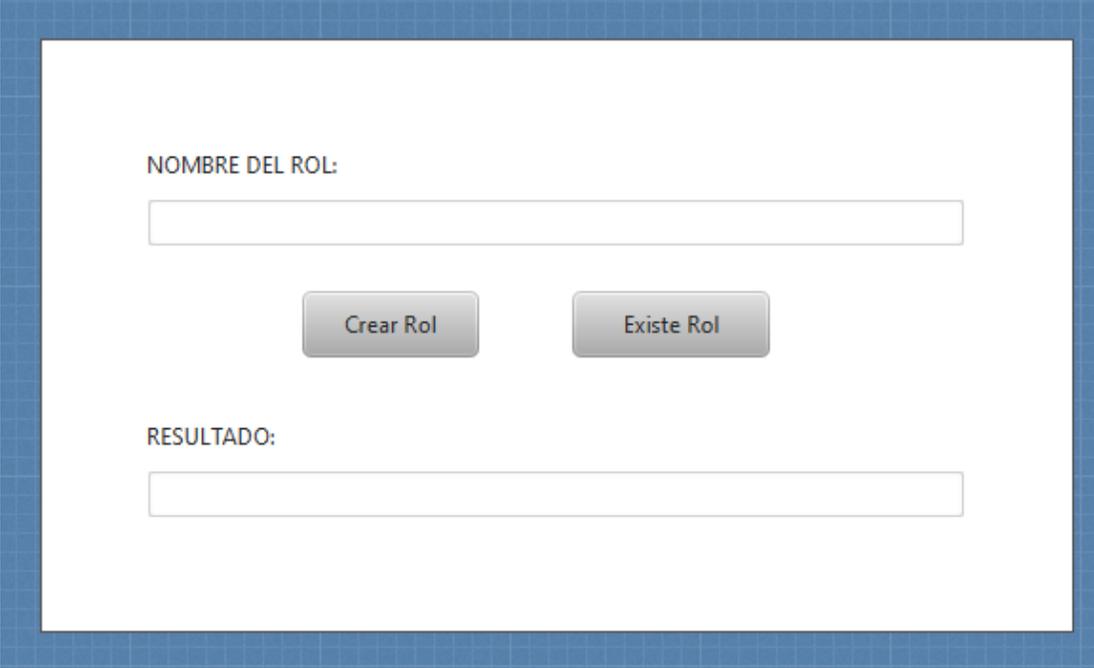


Ilustración 16: Llamar un método remoto de un servicio web SOAP en un controlador

- 13) Construir una interfaz gráfica usando JavaFX Scene Builder. Hacer doble click en la vista FXML.
- 14) Luego de arrastrar y soltar elementos gráficos se debe obtener una interfaz gráfica así:



The image shows a Java Swing window with a blue border. Inside the window, there is a text input field labeled "NOMBRE DEL ROL:". Below this field are two buttons: "Crear Rol" and "Existe Rol". Below the buttons is another text input field labeled "RESULTADO:".

- 15) Asígnele un id a los elementos de la interfaz gráfica para poder manejarlos desde la clase controladora.
- 16) Asígnele un id a los eventos a usar.
- 17) Inyectar los elementos gráficos a manipular en la clase controladora.
- 18) Inyectar los eventos a usar en la clase controladora.
- 19) En la clase controladora: Reemplace:

```
@FXML
private void handleButtonAction(ActionEvent event) {
    System.out.println("You clicked me!");
    label.setText("Hello World!");
}
```

Por:

```
@FXML
private void crearRol(ActionEvent event) {
```

```
String resultado = crearRol( this.textfieldNombrerol.getText(
) );
this.textfieldResultado.setText( resultado );
}

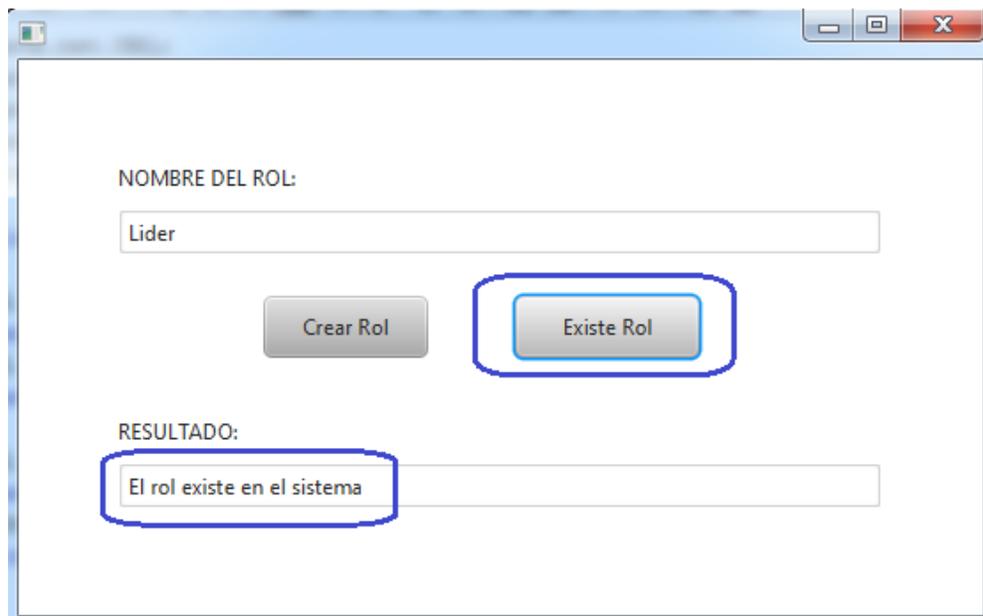
@FXML
private void existeRol(ActionEvent event) {
    Boolean resultado = existeRol( this.textfieldNombrerol.getText(
) );

    if( resultado )
        this.textfieldResultado.setText( "El rol existe en el
sistema" );
    else this.textfieldResultado.setText( "El rol no existe en el
sistema" );
}
```

20) Hacer click derecho sobre su proyecto JavaFX >> Clean and Build.

Hacer click derecho sobre su proyecto JavaFX >> Run

Probar la aplicación con diversos datos:



Para ver los pasos en detalle se debe leer el manual de mantenimiento. (Anexo: 8.4 Manual de mantenimiento)

4.4.1. GENERACIÓN INICIAL DE LA APLICACIÓN CLUBMAT

Con las herramienta yEd Graph Editor [33], la cual es gratuita (freeware), se generaron todos los diagramas UML del presente trabajo de grado.

yEd es una herramienta que genera muchos tipos de diagramas, entre ellos los UML, es flexible y genera diagramas con una apariencia atractiva.

Luego de tener el diagrama de entidades de negocio se escribieron las entidades JPA (Entity Beans). Estas entidades JPA no fueron generadas, fueron escritas manualmente usando la IDE Netbeans 7.3. [35] Todo esto en un proyecto de aplicación empresarial de Java EE 6, configurado para que se ejecutara sobre un servidor de aplicaciones GlassFish 3 y con un pool de conexiones a una base de datos PostgreSQL. [36]

Los Enterprise JavaBeans fueron generados por medio de un asistente de la IDE Netbeans 7.3 llamado “Session Beans for Entity Classes”, el cual a partir de cada Entity Bean, genera un EJB. En estos EJB es donde se implementaron los métodos de negocio necesarios para la construcción de la aplicación web CLUBMAT.

Después de tener implementados los Enterprise JavaBeans de este proyecto, se generó por cada EJB un Servicio Web SOAP, usando el asistente de Netbeans: “Create Web Service from Existing Session Bean”.

Finalmente, para la construcción de la capa de presentación se usó un proyecto de JavaFX 2 [7] en la IDE Netbeans 7.3. Este proyecto se configuro de tal forma que funcionara como un cliente de un Servicio Web SOAP.

Con ayuda de la herramienta JavaFX Scene Builder 1.1 [34] se diseñó la interfaz gráfica de usuario de la aplicación web, separando la vista de la lógica por medio de un patrón MVC (Modelo Vista Controlador).

A continuación podrá encontrar capturas de pantalla de la aplicación web: CLUBMAT.

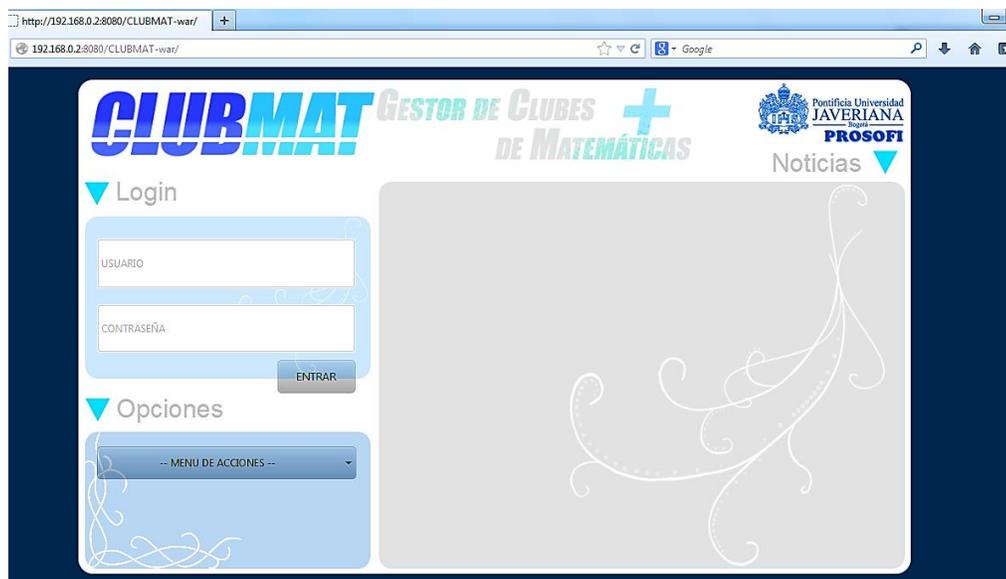


Ilustración 17: Apariencia Visual 1

La anterior foto muestra la apariencia visual de la pantalla principal de la aplicación web CLUBMAT. Como se puede observar, hay un formulario para realizar la autenticación de usuarios. También se puede observar un menú de acciones en donde los usuarios pueden acceder a determinadas funciones según su rol en el sistema.

- ❖ Identificarse como un usuario
- ❖ Cerrar la sesión del sistema
- ❖ Auto-registrarse como administrador de un club de matemáticas
- ❖ Aprobar el registro de un nuevo administrador.

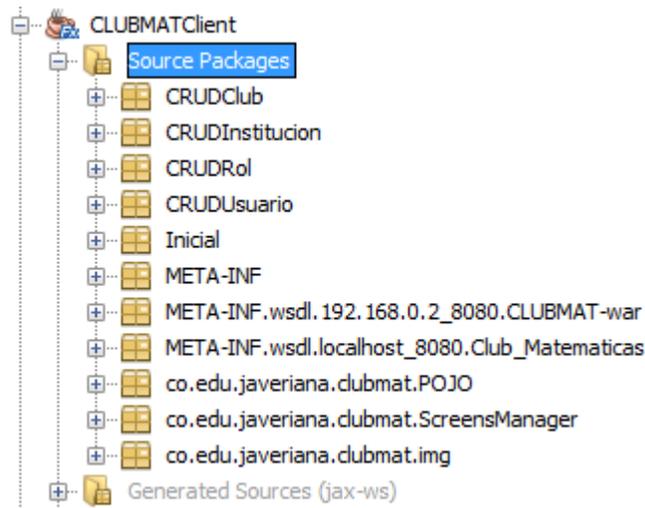
De estos seis casos de uso se implementaron solo aquellos que tenían asociados requerimientos de sistema con prioridad alta. Es decir, se realizaron todos excepto el caso de uso auto-registrarse como administrador de un club de matemáticas.

La metodología usada para la autenticación de los usuarios se realizó sobre la capa de presentación, es decir sobre el cliente JavaFX 2. [7] Este cliente es quien tiene la responsabilidad de activar o desactivar las operaciones a las que puede acceder un usuario según su rol. Esto se logra refrescando la interfaz gráfica teniendo en cuenta el rol del usuario autenticado. Esta autenticación se realiza consultando la base de datos.

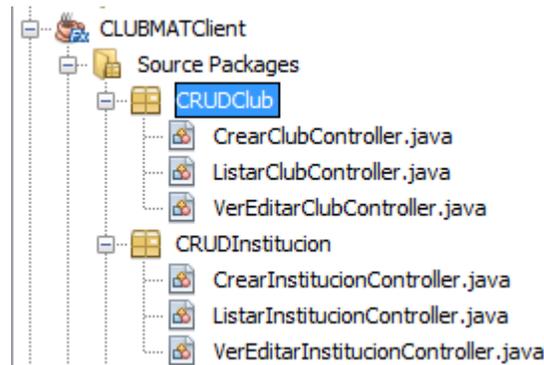
4.5. RESUMEN DE LOS PASOS PARA CONSTRUIR UN CRUD CON TECNOLOGÍAS JAVA EE 6 Y JAVAFX 2

Los pasos presentados a continuación son un resumen de los que se encuentran en el manual de mantenimiento. Leer el anexo 8.4 Manual de mantenimiento en donde se encuentran los pasos con mayor detalle.

- 1) Abrir la IDE Netbeans 7.3 y vaya a: File >> Open Project... Seleccione el proyecto empresarial CLUBMAT >> Clic en Open Project.
- 2) File >> Open Project... >> CLUBMATClient >> Click en Open Project
- 3) Por cada CRUD se debe crear un paquete.



- 4) Desplegar los paquetes CRUDClub y CRUDInstitucion.



- 5) Para crear un CRUD en JavaFX 2 es necesario crear mínimo tres clases controladores, cada una de ellas controla una vista .fxml.

Esto quiere decir que por cada CRUD se deben diseñar mínimo tres vistas.

- 6) Crear una vista FXML vacía para la acción de Listar (Clic derecho sobre CLUBMATClient >> New >> Empty FXML...). Usar una clase controladora para el FXML a crear.
- 7) La clase controladora de la acción listar debe contener elementos gráficos como los siguientes:

```
@FXML
private TableView< Club > tablaClub;
@FXML
private TableColumn columnaIdClub;
@FXML
private TableColumn columnaNombreClub;
@FXML
private TableColumn columnaLema;
@FXML
private TableColumn columnaInstitucion;
@FXML
private TableColumn columnaRegistrador;

@FXML
private Button botonVer;
@FXML
private Button botonEditar;
@FXML
private Button botonEliminar;
@FXML
private Button botonRegistrarClub;

ObservableList< Club > clubes;

public static Club clubSeleccionado;

public static Boolean editarClub;

private final ListChangeListener< Club > selectorTablaInstitucion =
```

- 8) Es necesario crear un POJO por cada CRUD, es decir una clase que mapea uno a uno los atributos de una entidad y los relaciona con cada una de las columnas de la tabla.
- 9) realice las acciones “Ver” y “Editar”, las cuales son dos operaciones diseñadas en una sola pantalla. Basarse en los archivos VerEditarClub.fxml y VerEditarClubController.java
- 10) Realice la acción “Crear” de su CRUD. Examine los archivos: CrearClub.fxml y CrearClubController.java a profundidad para entender cómo construir esta acción. Después de terminar la operación “Crear” habrá terminado satisfactoriamente la construcción de su CRUD.

- 11) Para que un CRUD diseñado en el framework JavaFX 2 logre conectarse con componentes de la plataforma Java EE 6, deben usarse Servicios Web SOAP.

En cada clase controladora de un CRUD debe llamar remotamente los métodos de negocio implementados en un EJB que necesite.

4.6. PRUEBAS PARA LA APLICACIÓN WEB CLUBMAT

Luego de realizar la construcción de la aplicación web CLUBMAT se realizó un test de prueba sobre CRUDs. Este test es realizado sobre los casos de uso que son CRUDS en la aplicación web CLUBMAT.

4.6.1. GESTIÓN DE ROLES

Test Crear un nuevo Rol:

nombreRol	Acción	Observaciones
Lider	Se ejecutó con normalidad	
Lider	No se ejecutó	El rol líder ya existe
Administrador Club	Se ejecutó con normalidad	

Test Modificar Rol:

Se realizaron diferentes pruebas sobre los roles Lider y Administrador Club ya existentes. Se modificó el nombre de rol “Lider” por “Lider2”. Luego se modificó “Lider2” por “Administrador Club” y no se ejecutó correctamente pues ya existía un rol con ese nombre.

nombreRol	Acción	Observaciones
Lider2	Se ejecutó con normalidad	
Lider2 -> Administrador Club	No se ejecutó	El rol Administrador Club ya existe

Test Eliminar Rol:

Se eliminó el rol Lider2 correctamente. El sistema impidió eliminar el rol “Administrador Club” pues es obligatorio para el funcionamiento de la aplicación web.

4.6.2. GESTIÓN DE USUARIOS

Test crear Usuario:

Se crearon tres nuevos usuarios.

nomUsuario	passw	email	no mb res	apellidos	dirección	ciudad	celular	estado	Acción	Observaciones
andres	123	l@l.com	andres	sanchez	1234456	Bogotá	1234567	activo	Se ejecutó con normalidad	
mercedes	123	l@l.com	mercedes	dfdsfsdfs	1234456	Bogotá	65456646	activo	No se ejecutó	El correo electrónico ya existe
mercedes	123	m@l.com	mercedes	dfdsfsdfs	1234456	Medellín	65456646	activo	Se ejecutó correctamente	
andres	543	f@g.com	43rer	wrwr	Re454342	Bogotá	676767	inactivo	No se ejecutó	El nombre de usuario ya existe
admin	123	h@g.com	admin	admin	12345	Bogotá	876543	activo	Se ejecutó correctamente	

Test modificar un usuario:

Se modificó el usuario “admin” en varias oportunidades exitosamente

nomUsuario	passw	email	nombre	apellidos	dirección	ciudad	celular	estado	Acción	Observaciones
admin	453	h@g.com	admin	admin	12345	Bogotá	876543	activo	Se ejecutó correctamente	
admin	453	h@g.com	admin	admin	12345	Bogotá	876543	inactivo	Se ejecutó correctamente	
admin2	453	h@g.com	admin	admin	12345	Bogotá	876543	inactivo	Se ejecutó correctamente	
admin2	453	p@g.com	admin	admin	12345	Bogotá	876543	inactivo	Se ejecutó correctamente	

4.6.3. GESTIÓN DE INSTITUCIONES

Test de registrar institución:

nombreinst	ciudad	localidad	dirección	telefono	rector	tipoinstitucion	Acción	Observaciones
Colegio A	Bogotá	Usme	123	1211334	weer	publica	Se ejecutó correctamente	
Colegio B	Bogotá	Usme	456	4342432434	werwerwer	privada	Se ejecutó correctamente	
Colegio A	Bogotá	Usme	123	1211334	weer	publica	No se ejecutó	El nombre de la institución ya existe

Test de modificar institución:

Se realizaron distintas pruebas sobre el colegio A registrado anteriormente.

nombreinst	ciudad	localidad	dirección	telefono	rector	tipoinstitucion	Acción	Observaciones
Colegio A	Bogotá	Usme	345445	1211334	weer	publica	Se ejecutó correctamente	
Colegio A	Bogotá	Usme	345445	1211334	dídfdf	publica	Se ejecutó correctamente	
Colegio A	Bogotá	Usme	345445	43343432	dídfdf	publica	Se ejecutó correctamente	

4.6.4. GESTIÓN DE CLUBES

Test de registrar un club:

Se registraron dos clubes: Club A y Club B.

nombre	lema	Acción	Observaciones
Club A	asasasas	Se ejecutó correctamente	
Club A	5445454	No se ejecutó	El Club A ya existe.
Club B	5rtretert	Se ejecutó correctamente	

Test de modificar un club:

Se realizaron distintas pruebas sobre el club A registrado anteriormente.

nombre	lema	Acción	Observaciones
Club A	hola	Se ejecutó correctamente	
Club A -> Club B	hola	No se ejecutó	El Club B ya existe.
Club A -> Club C	hola	Se ejecutó correctamente	

5. RESULTADOS

5.1. RESULTADOS EN LO RELATIVO A LAS NUEVAS TECNOLOGÍAS

La aplicación web construida generó los siguientes resultados en cuanto a las tecnologías:

- ❖ **Integración de nuevas tecnologías:** se realizó satisfactoriamente la integración del framework JavaFX 2 [7] y la plataforma Java EE 6. [1] Con esto se reemplazó el framework estándar JSF, tecnología en la cual se identificaron los siguientes problemas: agujeros de seguridad como XSS o CRSF [39], se depende de especialistas en XHTML para construir la interfaz gráfica y hay pocas herramientas para generación de animaciones, gráficos 3D, entre otros.
- ❖ **JavaFX2 y la interfaz gráfica de usuario:** Con el framework JavaFX 2 [7] se puede construir una interfaz gráfica de usuario rica e interactiva usando herramientas propias de JavaFX 2.

5.2. RESULTADOS EN LO RELATIVO A LA APLICACIÓN WEB CLUBMAT

Como resultado del presente proyecto se generó una aplicación web que facilita la gestión de clubes de matemáticas bajo el marco del programa social PROSOFI de la Pontificia Universidad Javeriana.

Por otra parte, junto a la aplicación web se generaron diseños estándar UML, levantamiento de requerimientos y casos de uso para que así esta pueda ser extendida a futuro.

Las siguientes son las mejoras logradas con este proyecto para la gestión de clubes de matemáticas bajo el marco de PROSOFI:

- ❖ La aplicación web CLUBMAT puede ser accedida desde cualquier parte del mundo, por medio de un computador con conexión a internet.

- ❖ Cada usuario tiene acceso al sistema por medio de un usuario y contraseña.
- ❖ La aplicación web CLUBMAT es un primer prototipo para la construcción de un sistema de información robusto que permite gestionar un club de matemáticas.
- ❖ La aplicación web CLUBMAT es un sistema más seguro que una aplicación construida con el framework JSF.
- ❖ La aplicación web CLUBMAT posee una interfaz gráfica de usuario limpia y atractiva para los usuarios, usando el framework JavaFX 2. [7]

5.3. CUMPLIMIENTO DE LOS OBJETIVOS GENERALES Y ESPECÍFICOS

A continuación se realiza un análisis del cumplimiento de los objetivos generales y los objetivos específicos del presente trabajo de grado.

Objetivos generales	Análisis
Desarrollar una primera versión de la aplicación CLUBMAT en donde se use la tecnología de interfaz gráfica JavaFX 2. [7]	Si se cumplió porque se construyó una primera versión de la aplicación web CLUBMAT que usa interfaz gráfica de usuario construida con tecnología JavaFX 2.
La tecnología JavaFX 2 debe comunicarse con la capa de negocio (EJB) de la plataforma Java EE 6. [1]	Si se cumplió porque se logró integrar el framework JavaFX 2 con la capa de negocio (EJB) de la arquitectura Java EE 6.

Tabla 14: Análisis de los objetivos generales

ID	Objetivos específicos	Análisis
1	Investigar, estudiar y comprender las herramientas tecnológicas alrededor del desarrollo del trabajo de grado: Java EE 6 [1] JavaFX 2 [7], EJB 3 [4], JPA 2 [3], GlassFish [3], entre	Se cumplió porque se investigó, estudio y comprendió todas las herramientas usadas en el presente trabajo de grado, identificando los beneficios ofrecidos por cada una de ellas.

	otras. Entender los beneficios ofrecidos por ellas.	
2	Apropiarse de los conocimientos obtenidos de las tecnologías alrededor del proyecto, realizando pruebas con pequeños prototipos. Se debe lograr la integración de todas las tecnologías.	Se cumplió porque con el conocimiento obtenido se generaron prototipos y luego se integraron todas las tecnologías.
3	Realizar un análisis sobre las reglas de negocio presentes en el proceso de la gestión de clubes de matemáticas bajo el marco de PROSOFI. Refinar que es lo que desea realmente el cliente en el software final.	Se cumplió porque se realizó un análisis de las reglas de negocio usando técnicas de análisis y diseño orientado a objetos e ingeniería de software.
4	Diseñar y modelar las reglas de negocio implícitas en el software de gestión de clubes de matemáticas, es decir, generar los artefactos UML que modelen dicho problema.	Se cumplió porque a lo largo del proyecto se generaron diversos artefactos UML que modelaron el problema a resolver.
5	Realizar el desarrollo del prototipo cubriendo todos los requerimientos funcionales y no funcionales establecidos en la fase de diseño.	Se cumplió porque se construyó un prototipo que cubrió los requerimientos funcionales y no funcionales que se estimaron alcanzables.
6	Realizar la evaluación del prototipo a entregar usando metodologías como las pruebas de aceptación, pruebas alfa y pruebas beta. [10]	Se cumplió porque se realizaron las pruebas mencionadas sobre el prototipo entregado.
7	Elaborar guías en donde explícitamente se diga cómo se realizó la integración de las tecnologías y cómo desarrollar software en ese ambiente integrado.	Se cumplió porque se generaron dos guías: Manual de usuario y manual de mantenimiento. Allí se explicó cómo fue la integración de las tecnologías y cómo

		desarrollar software en ese ambiente integrado.
--	--	---

Tabla 15: Análisis de los objetivos específicos

6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1. CONCLUSIONES

Las conclusiones del presente trabajo de grado son las siguientes:

- ❖ Fueron encontradas las ventajas y desventajas que se obtienen al sustituir el framework JSF por el framework Java FX 2 en la arquitectura Java EE 6.
- ❖ JavaFX 2 es una tecnología gráfica que aún no ha sido explotada lo suficiente, la cual es una alternativa real al framework JSF a la hora de construir una capa de presentación para aplicaciones Java EE 6.
- ❖ El uso de un servidor de aplicaciones en la construcción de una aplicación web permite que el programador se concentre en programar la lógica de negocio y dejar en un segundo plano el manejo transaccional, el cumplimiento de las propiedades ACID, objetos distribuidos, etc.
- ❖ JavaFX 2 cuenta con herramientas propias para generar una interfaz gráfica rica y permite a los desarrolladores programar todo en el lenguaje Java, sin necesidad de tener conocimientos de otras tecnologías externas.
- ❖ El uso del patrón Modelo Vista Controlador ayuda a que un software sea mantenible y que la estructura del mismo sea organizada, facilitando el desarrollo.
- ❖ Técnicamente este proyecto aportó la construcción de un primer prototipo de software de CLUBMAT sobre internet. Con esta primera versión de la aplicación web se puede gestionar usuarios, roles, instituciones y clubes de matemáticas.

- ❖ Socialmente este proyecto aportó que toda la información de los clubes de matemáticas se administre en una aplicación web y no en otros métodos aislados como puede ser papel o documentos de Microsoft Office.

6.2. RECOMENDACIONES

Para este tipo de proyectos que buscan una integración novedosa de tecnologías, es importante que el programador tenga algo de conocimiento de cómo realizar la integración que desea. De lo contrario puede que se sufran retrasos y por esto la construcción del software sea más lento.

También es importante tener muy claro el proyecto a realizar desde un principio y que el cliente este comprometido y muestre interés en que la construcción del software salga adelante.

6.3. TRABAJOS FUTUROS

La aplicación web CLUBMAT: Gestor de clubes de matemáticas, puede extenderse y mejorarse en el futuro. Se generaron artefactos de diseño y conocimiento clave sobre el framework JavaFX 2 [7] y su integración con la plataforma Java EE 6. [1]

Esta aplicación web puede mejorarse (menor sobrecarga y mejor tiempo de respuesta) si se usan Servicios Web REST para realizar la integración entre las tecnologías.

Esta mejora consiste en un mejor rendimiento de la aplicación en comparación de la tecnología SOAP. REST se caracteriza por ser una tecnología que consume bajos recursos y que se puede utilizar en los Servicios Web. [37]

7. REFERENCIAS Y BIBLIOGRAFÍA

- [1] Oracle Corporation, “The Java EE 6 Tutorial”, 2013. [En línea]. Available: <http://docs.oracle.com/javaee/6/tutorial/doc/jvaeetutorial6.pdf>
- [2] A. Goncalves, “Beginning Java EE6 plataform with glassfish 3”. de Beginning Java EE6 plataform with glassfish 3, Apress, 2010.
- [3] M. Keith and M. Schincariol. “Pro JPA 2 Mastering the Java Persistence API.”, Apress, 2009.
- [4] L. Rubinger y E. Bill, “Enterprise JavaBeans 3.1.” Safari, 6 Ed, 2012.
- [5] F. Pech, M. A. Gomez, y L. A. de la Cruz Diaz, “Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces”, Instituto Tecnológico Superior de los Ríos. 2012 [En línea] Available: <http://www.tamps.cinvestav.mx/~fpech/sd/files/paper001.pdf>
- [6] D. Carl. “JavaFX 2.0: Introduction by Example.”, Apress, 2011.
- [7] J. Weaver y S. Chin, “Pro JavaFX 2: A Definitive Guide to Rich Clients with Java Technology”, Apress, 2012.
- [8] D. Heffelfinger, “Java EE 6 Development with NetBeans 7”, Packt Publishing, 2011.
- [9] Oracle Corporation. [En línea]. Available: <http://www.java.com/es/about/>
- [10] I. Sommerville, “Ingeniería del software”, Madrid, Pearson Educación, 2005.
- [11] H. Böck, “The Definitive Guide to NetBeans Platform 7”, Apress, 2011.
- [12] D. Heffelfinger, “Java EE 6 with GlassFish 3 Application Server”, Packt Publishing, 2010.
- [13] M. Heinrichs, “JavaOne 2012: JavaFX for Business Application Developers.” [En línea]. Available: https://oracleus.activeevents.com/connect/sessionDetail.ww?SESSION_ID=5434
- [14] A. Bien, “JavaOne 2012: Building Serious JavaFX 2 Applications”. [En línea]. Available: https://oracleus.activeevents.com/connect/sessionDetail.ww?SESSION_ID=3908

- [15] A, Bien. “The Enterprise side of JavaFX”. [En línea]. Available: <http://www.oracle.com/technetwork/articles/java/enterprisefx-1651840.html>
- [16] A, Bien. “The Enterprise side of JavaFX Part Two”. [En línea]. Available: <http://www.oracle.com/technetwork/articles/java/enterprisefxpt2-1676394.html>
- [17] A, Bien. “The Enterprise side of JavaFX Part Three”. [En línea]. Available: <http://www.oracle.com/technetwork/articles/java/enterprisefxpt3-1735081.html>
- [18] K. Wähler, “When to use JavaFX 2 instead of HTML5 for a Rich Internet Application (RIA)?” Erlangen, 2012, [En línea]. Available: <http://www.kai-waehner.de/blog/2012/04/18/when-to-use-javafx-2-instead-of-html5-for-a-rich-internet-application-ria/>
- [19] J. L. Weaver, “Best Practices for JavaFX 2.0 Enterprise Applications”. [En línea]. Available: <http://www.oracle.com/technetwork/articles/java/fxbest-1583679.html>
- [20] M. Debnath, "Combining JavaFX and Java Enterprise Modules". 2013. [En línea]. Available: <http://www.developer.com/java/ent/combining-javafx-and-java-enterprise-modules.html>
- [21] K. E. Kendall, "Análisis y diseño de sistemas", Monterrey, Pearson Education, 2005.
- [22] L. Cao, K. Mohan y P. X. Balasubramaniam, “How Extreme does Extreme Programming Have to be? Adapting XP Practices to Large-scale Projects,” de International Conference on System Sciences, Hawaii, 2004.
- [23] A. Ramos, “Internet, características y evolución de Aplicaciones web”, S. Quality, Ed., Madrid, Paraninfo, 2011, p. 5.
- [24] IBM, “Application server purpose”, de WebSphere Application, Redbooks, 2009, pp. 4-5.
- [25] J. A. Garcias y M. S. Lancheros. "Sistema de información para el ingreso, mantenimiento y salida de estibas", Bogotá, 2013.
- [26] J.M. Calvo, "Web 2.0: ¿algún avance tecnológico o sólo Web Social?" [En línea]. Available: http://www.acis.org.co/fileadmin/Revista_116/Cinco.pdf
- [27] Oracle Corporation, “JavaFX Scene Builder Overview”, 2012 [En línea]. Available: <http://docs.oracle.com/javafx/scenebuilder/1/overview/jsbpub-overview.htm>

- [28] Y. A. Velasco y M. F. Villacrés, "Desarrollo del sistema control biométrico de docentes de la universidad central del ecuador", Quito, 2012. [En línea]. Available: <http://www.dspace.uce.edu.ec/bitstream/25000/220/1/T-UCE-0011-10.pdf>
- [29] I.Ramos y M. D. Lozano, "Ingeniería Del Software Y Bases de Datos: Tendencias Actuales", Ed. de la Universidad de Castilla, La mancha, 2000.
- [30] S. Mitchell, "Aprendiendo Active Server Pages 3.0 en 21 días", Ed. Pearson, Naucalpan de Juárez, 2001.
- [31] V. P. Madrid y J. F. De Paz, "Servicios Web", Salamanca. [En línea]. Available: <http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/ServiciosWeb.pdf>
- [32] PROSOFI, Pontificia Universidad Javeriana, Bogotá, 2012, [En línea]. Available: http://puj-portal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/plt_facultad/Prosofi
- [33] yWorks, "yEd Graph Editor", 2013. [En línea]. Available: http://www.yworks.com/en/products_yed_about.html
- [34] Oracle Corporation, "JavaFX Scene Builder: A Visual Layout Tool for JavaFX Applications", 2013. [En línea]. Available: <http://www.oracle.com/technetwork/java/javafx/tools/index.html>
- [35] Oracle Corporation, "NetBeans IDE 7.3.1 Release Information", 2013. [En línea]. Available: <http://www.netbeans.org/community/releases/73/>
- [36] The PostgreSQL Global Development Group, 2013. [En línea]. Available: <http://www.postgresql.org>
- [37] R. Navarro, "Modelado, Diseño e Implementación de Servicios Web: REST vs Web Services", Valencia, 2007. [En línea]. Available: <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>
- [38] The PostgreSQL Global Development Group, "PostgreSQL 9.2.4 Documentation", 2013. [En línea]. Available: <http://www.postgresql.org/files/documentation/pdf/9.2/postgresql-9.2-US.pdf>
- [39] A. Sood y K. Raja, "Dissecting Java Server Faces for Penetration Testing". 2011. [En línea]. Available: http://www.secniche.org/jsf/dissecting_jsf_pt_aks_kr.pdf

- [40] M. Katz, "Rich Internet Applications Tools: JSF, Flex, and JavaFX". 2011. [En línea]. Available: http://ajaxexperience.techtarget.com/assets/documents/Max_Katz_RIA_Tools.pdf
- [41] INCO: Facultad de Ingeniería, "Java Server Faces 2.0", 2010. [En línea]. Available: <http://www.fing.edu.uy/inco/cursos/tsi/TSI2/2010/teorico/09b-JSF.pdf>
- [42] R. Auger, "The Cross-Site Request Forgery (CSRF/XSRF) FAQ". 2010. [En línea]. Available: <http://www.cgisecurity.com/csrf-faq.html>
- [43] INTECO: Instituto Nacional de Tecnologías de Comunicación, "Tipos de Vulnerabilidades". [En línea]. Available: http://www.inteco.es/Formacion/Amenazas/Vulnerabilidades/Tipos_Vulnerabilidades
- [44] I. García y C. García, "Aplicación web para el conocimiento y conversión de unidades", 2006. [En línea]. Available: http://meteo.ieec.uned.es/www_Usumeteo2/Memoria/Capitulo3.pdf
- [45] A. Bustamante y C. Charum, "SIPROSOFI: Sistema de Información para la Gestión de Proyectos Sociales Construido mediante las tecnologías JEE 6, GXT, ERRAI", Bogotá, 2012. [En línea]. Available: <http://pegasus.javeriana.edu.co/~CIS1230IS05/documents/Memoria%20de%20Trabajo%20de%20Grado.pdf>
- [46] C. Fernández, "Servidores de aplicaciones", Universidad de Oviedo, Oviedo. [En línea]. Available: http://156.35.98.1/ficheros/apuntes/comercio_electronico/Tema%206%20-%20Servidores%20de%20aplicaciones.pdf
- [47] U. Acosta, "Introducción a Java Empresarial", Ciudad de México, 2012. [En línea]. Available: <http://www.globalmentoring.com.mx/curso-javaee/leccion1/PDFs/Curso%20Java%20EE%20-%202001%20Leccion%2001%20-%20Teoria.pdf>
- [48] J. Bautista, "Programación extrema", Universidad Unión Bolivariana, La Paz, 2012. [En línea]. Available: <http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACION%20EXTREMA.pdf>

8. ANEXOS

Los siguientes documentos forman parte de los anexos y se encuentran disponibles en la página del presente trabajo de grado:

8.1. REQUERIMIENTOS DEL SISTEMA (HACER & USOS)

Este anexo contiene la plantilla HACER & USOS. Esta plantilla contiene los conceptos, usuarios, y requerimientos del sistema de información de una manera detallada.

8.2. INVENTARIO DE MÓDULOS Y CASOS DE USO

Esta plantilla contiene el inventario de módulos y casos de uso de una manera detallada.

8.3. MANUAL DEL USUARIO

Este anexo contiene el manual del usuario para la aplicación web CLUBMAT: Gestor de clubes de matemáticas. Incluye indicaciones para la instalación del software.

8.4. MANUAL DE MANTENIMIENTO

Este anexo contiene el manual de mantenimiento para la aplicación web CLUBMAT: Gestor de clubes de matemáticas. Incluye indicaciones para la instalación del ambiente de desarrollo.

8.5. CRONOGRAMA DEL PROYECTO

Este anexo contiene el cronograma que fue seguido para la ejecución del presente trabajo de grado.

8.6. GLOSARIO

Este anexo contiene el glosario del proyecto de grado “CLUBMAT: Gestor de clubes de matemáticas construido con tecnologías Java EE 6 y JavaFX 2”.