

CLUBMAT 1.1: Extensión de una aplicación web destinada al fortalecimiento de clubes escolares matemáticos integrando JavaFX2 y JavaEE6 con servicios web basados en REST



Autor:

Julián Camilo Ortega Muñoz

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.
2013

CLUBMAT 1.1: Extensión de una aplicación destinada al fortalecimiento de clubes escolares matemáticos acoplando JavaFX2 y Java-EE6 con web services basados en REST

Autor:

Julián Camilo Ortega Muñoz

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

Director

Ing: Maria Consuelo Franky

Jurados del Trabajo de Grado

Jaime Pavlich Mariscal

Blanca Elvira Oviedo Torres

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~CIS1330IS04>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.
10,2013

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS**

Rector Magnífico

Joaquín Emilio Sánchez García S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Decano del Medio Universitario Facultad de Ingeniería

Padre Sergio Bernal Restrepo S.J.

Director de la Carrera de Ingeniería de Sistemas

Ingeniero Germán Alberto Chavarro Flórez

Director Departamento de Ingeniería de Sistemas

Ingeniero Rafael Andrés González Rivera

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 primera instancia, quiero agradecer a Dios por permitirme llegar a esta etapa de mi vida, ya que gracias a ÈL, pude terminar satisfactoriamente mi reto profesional.

A mis padres y mi familia por su gran esfuerzo de brindarme la oportunidad de estudiar en esta universidad, gracias a ellos he logrado llegar al final y con el propósito de devolverles ese esfuerzo.

Tambièn a mi directora de trabajo de grado, María Consuelo Franky por su apoyo constante en el desarrollo del trabajo, sus consejos, retroalimentación y enseñanza permanente que me brindo en todo el proceso.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	13
I - DESCRIPCIÓN GENERAL DEL TRABAJO DE GRADO.....	14
1. Oportunidad, Problemática, Antecedentes	14
1.2 <i>Formulación del problema que se resolvió.....</i>	<i>15</i>
1.3 <i>Justificación.....</i>	<i>15</i>
1.4 <i>Impacto Esperado.....</i>	<i>17</i>
2. Descripción del Proyecto	18
2.1 <i>Visión global.....</i>	<i>18</i>
2.3 <i>Objetivo general.....</i>	<i>19</i>
2.4 <i>Objetivos específicos.....</i>	<i>19</i>
2.5 <i>Método que se propuso para satisfacer cada fase metodológica.....</i>	<i>19</i>
2.7 <i>Fases Metodologicas.....</i>	<i>20</i>
2.7.1 <i>Fase de Planeación.....</i>	<i>21</i>
2.7.2 <i>Fase de Preparación y Desarrollo.....</i>	<i>21</i>
2.7.3 <i>Fase de Entrega.....</i>	<i>22</i>
2.8 <i>Actividades y Entregables.....</i>	<i>23</i>
II - MARCO TEÓRICO.....	25
3. Marco Contextual	25
3.1 <i>Clubes de matemáticas.....</i>	<i>25</i>
3.2 <i>Clubmat Version 1.0.....</i>	<i>27</i>
4. Marco Conceptual	28
4.1 <i>Aplicaciones Empresariales Web.....</i>	<i>28</i>
4.2 <i>Herramientas RIA y Java FX2.....</i>	<i>32</i>
4.3 <i>PostgreSQL.....</i>	<i>35</i>
4.3 <i>Servicios Web.....</i>	<i>35</i>
III – DESARROLLO DEL TRABAJO.....	41
5. Fase de Planeación	41
5.1 <i>Integración JavaFX2 y JavaEE6 con servicios web basados en REST.....</i>	<i>41</i>
5.1.1 <i>REST vs SOAP.....</i>	<i>41</i>
5.1.3 <i>Pruebas Funcionales de Rendimiento con REST.....</i>	<i>43</i>
5.1.5 <i>Demo y Tutorial para Aplicaciones Futuras.....</i>	<i>46</i>
5.2 <i>Especificaciones del producto CLUBMAT.....</i>	<i>47</i>
5.2.1 <i>Modelo del Dominio.....</i>	<i>47</i>
5.2.2 <i>Requerimientos del Sistema.....</i>	<i>50</i>
5.2.3 <i>Casos de Uso CLUBMAT.....</i>	<i>56</i>
5.3 <i>Arquitectura del sistema.....</i>	<i>61</i>

6. Fases de Preparación, Diseño, Desarrollo y Pruebas	65
6.1 Estrategia general para desarrollar un caso de uso CRUD: Ejemplo CRUD de Clubes de Matemáticas.....	65
6.1.1 Preparación.....	65
6.1.2 Diseño.....	65
6.1.3 CREAR CLUB.....	67
6.1.4 EDITAR CLUB.....	71
6.1.5 ELIMINAR CLUB.....	76
6.1.6 VER CLUBES.....	80
6.2 Estrategia General para desarrollar un caso de uso de negocio: Ejemplo caso de uso “Ver opciones de preguntas”	85
6.2.1 Pruebas Funcionales	89
IV - RESULTADOS Y REFLEXIÓN SOBRE LOS MISMOS	92
7. Pruebas de Aceptación.....	92
V – CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....	93
8. Conclusiones	93
9. Recomendaciones	94
10. Trabajos Futuros	94
VI - REFERENCIAS Y BIBLIOGRAFÍA	96
VII – ANEXOS.....	100
Anexo 1. Tutorial y demo JavaFX2 y JavaEE6 con servicios web REST.....	100
Anexo 2. Pruebas funcionales y aceptación CLUBMAT	100
Anexo 3. Manual de usuario CLUBMAT	100
Anexo 4: Manual de mantenimiento CLUBMAT	101
Anexo 5. Prototipos no funcionales CLUBMAT	101
CARTA DE AUTORIZACIÓN DE LOS AUTORES (Licencia de uso).....	102
DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADOFORMULARIO	105

ILUSTRACIONES Y TABLAS

Ilustración 1: Datos estadísticos estudios de PROSOFI a la comunidad de Usme	17
Ilustración 2: Detalles de las fases metodológicas SCRUM y las actividades.....	20
Tabla 1: Actividades y Entregables	24
Ilustración 3: Composición de un club de matemáticas.....	26
Ilustración 4: CLUBMAT 1.0.....	27
Ilustración 5: Arquitectura JAVAEE6.....	30
Ilustración 6: Ejemplos aplicaciones javafx2.....	33
Ilustración 7: JavaFX Scene builder	35
Ilustración 8: Servicios web con protocolo SOAP.....	38
Ilustración 9: Ejemplo Arquitectura REST.....	40
Ilustración 10: REST vs SOAP.....	42
Ilustración 11: Ventana CRUD pruebas SOAP vs REST	43
Ilustración 12: Código prueba protocolo REST.....	44
Ilustración 13: Código Prueba Protocolo SOAP.....	45
Ilustración 14: Resultados pruebas REST VS SOAP	45
Ilustración 14: Roles y Usuarios	50
Tabla 2: Requerimientos Funcionales CLUBMAT	53
Tabla 3: Requerimientos No Funcionales CLUBMAT	55
Tabla 4: Priorización de Requerimientos.....	56
Ilustración 15: Diagrama de Casos de Uso CLUBMAT.....	57
Tabla 5: Casos de Uso CLUBMAT	59
Ilustración 16: Diagrama de Entidades de Negocio CLUBMAT	60
Ilustración 17: Arquitectura JAVAEE6- JAVAFX2- REST	61
Tabla 6: Objetivos arquitecturales	62
Ilustración 18: Arquitectura Global CLUBMAT.....	63
Ilustración 19: Control de Flujo Global CLUBMAT.....	64
Ilustración 20: Diagrama EBC Club de Matemáticas.....	66
Ilustración 21: Diagrama Proceso CRUD clubes de matemáticas.....	66
Ilustración 22: Diagrama de Secuencia Crear Club.....	67
Ilustración 23: Crearclub.fxml	68
Ilustración 24: Crearclub Controller	68
Ilustración 24: Crearclub ClienteRest.....	69
Ilustración 25: clubmatematicasFacadeRest.java	70
Ilustración 26: AbstractFacade.java.....	70
Ilustración 27: Clubmatematicas.java.....	71
Ilustración 28: Diagrama de Secuencia Editar Club	71
Ilustración 29: EditarClub.FXML.....	72
Ilustración 30: EditarClubController.java.....	73

Ilustración 31: ClubClienteRest.java	74
Ilustración 32: ClubmatematicasFacadeRest	75
Ilustración 33: AbstractFacade.java	76
Ilustración 34: Diagrama Secuencia Eliminar Club	77
Ilustración 35: ClubesInscritos.FXML	78
Ilustración 36: ClubesInscritosController.java	78
Ilustración 37: ClubClienteRest.java	79
Ilustración 38: clubmatematicasFacadeRest.java	80
Ilustración 39: AbstractFacade.java	80
Ilustración 40: Diagrama Secuencia Ver Clubes	81
Ilustración 41: ClubesInscritos.FXML	82
Ilustración 42: ClubesInscritosController.java	82
Ilustración 43: ClubesClienteRest.java	83
Ilustración 44: clubmatematicasFacadeREST.java	83
Ilustración 45: AbstractFacad.java	84
Ilustración 46: Diagrama Secuencia Ver Pregunta	85
Ilustración 47: VerPregunta.FXML	86
Ilustración 48: VerPreguntaController.java	87
Ilustración 49: PreguntasNegocioClienteRest.java	87
Ilustración 50: PreguntasNegocioClienteRest.java	89
Ilustración 51: Probar Servicios Rest	90
Ilustración 52: Pruebas Servicios Web Rest	91
Ilustración 51: Encuestas Aceptación Cliente	92

ABSTRACT

In the development of enterprise web applications there are a variety of technologies for its implementation. An important technology is JavaEE6 [2]. JavaEE6 architecture utilizes JSF (Java Server Faces) as standard framework for conducting user interfaces. This framework presents a several number of problems in security, interactivity, development time among others. (See Section 1.1 Justificación). [2] [11]

In the need to solve these problems we explored with the JavaFX2 a RIA framework (Rich Internet Applications) which by its nature it solved certain aspects of the user interaction in relation to JSF with the business application. (See Section 4. Marco Conceptual) [9]

Previously done a thesis where they made a coupling between these technologies (JavaFX2 and JavaEE6) using SOAP Web services . This work was based on a case study in the social program PROSOFI from Pontificia Universidad Javeriana about math clubs from Usme (Community of Bogotá) and they made part of the development of the system called CLUBMAT. [3] [4] [10]

CLUBMAT is a web information system that manages math clubs and was the case study work coupling between JavaEE6 and JavaFX2 .

The purpose of this work is to study the architecture left previously in the realization of this software to find improvements in coupling exploring technologies REST-based web services. Also, leave a standard integration of these technologies and complement the system CLUBMAT respect to customer requirements .

RESUMEN

En el desarrollo de aplicaciones web empresariales existe gran variedad de tecnologías para su realización. Una tecnología importante es JavaEE6 [2]. En la arquitectura JavaEE6 se trabaja como estándar el framework JSF(*Java Server Faces*) para la realización de interfaces de usuario. Este framework presenta una serie de problemáticas en cuestiones de seguridad, interactividad, tiempo de desarrollo, entre otras. (Ver Sección 1.1Justificación). [2] [11]

En la necesidad de resolver estas problemáticas se exploró JavaFX2, un framework RIA (*Rich Internet Applications*) que por sus características soluciona ciertos aspectos en la interacción del usuario con la aplicación empresarial y es menos susceptible de ser alterado en ejecución por usuarios mal intencionados. (Ver Sección: 4. Marco Conceptual). [9]

Previamente se realizó un trabajo de integración entre estas tecnologías (JavaFX2 y JavaEE6), utilizando servicios web basados en SOAP [30] . Este trabajo fue basado en un caso de estudio enmarcado en el programa social PROSOFI de la Pontificia Universidad Javeriana acerca de los clubes de matemáticas de la comunidad de Usme y se desarrolló parte del sistema de información denominado CLUBMAT. [3] [4] [10]

CLUBMAT es un sistema de información web que gestiona y administra clubes de matemáticas y fue el caso de estudio para el trabajo de integración entre JavaFX2 y JavaEE6.

El propósito de este trabajo de grado fue estudiar la arquitectura dejada previamente, en la realización de este software para encontrar mejoras en el acoplamiento de las tecnologías, explorando los servicios web basados en REST [30]. Así mismo, dejar un estándar de integración de estas tecnologías y complementar el sistema CLUBMAT en su totalidad respecto a los requerimientos del cliente.

Lo anterior favoreció para la formulación de conclusiones, con las cuales queremos ayudar a nuestro público objetivo, a obtener una visión general de las condiciones actuales de la aplicación de CLUBMAT, de manera que se pueda dar una implementación rápida y útil de la misma, y así mismo la permanencia y replicación de la aplicación a otras comunidades con una necesidad similar.

RESUMEN EJECUTIVO

En la actualidad existe gran variedad de plataformas y lenguajes de programación para el desarrollo de aplicaciones web empresariales. Una de las plataformas que ha surgido es la edición empresarial de Java (*Java Enterprise Edition*) JavaEE6[2][13]. Esta es una tecnología que presenta una arquitectura de tecnologías y frameworks para el desarrollo de estas aplicaciones.

Las principales tecnologías que componen la plataforma Java EE 6 son: JavaBeans empresariales (*Enterprise JavaBeans, EJB*) [15], un API para el manejo de entidades persistentes (*Java Persistence API, JPA*) [28], sobre bases de datos relacionales y finalmente JavaServer Faces (JSF) que es la tecnología estándar para la creación de interfaces de usuario en la web.

Esta última mencionada, funciona como framework para el desarrollo de interfaces de usuario de estas aplicaciones, pero tiene ciertos problemas como la seguridad, donde las páginas web pueden ser alteradas en ejecución, depende de especialistas en XHTML para su desarrollo, pocas herramientas gráficas visuales como gráficos 3D, animaciones haciendo que la interfaz sea poco interactiva (en relación a las aplicaciones de escritorio) entre otras problemáticas. (Ver Sección 1.1 Justificación).

Con la intención de buscar nuevas tecnologías y suplir estas necesidades, se exploró con JavaFX2 un framework RIA [1] (*Rich Internet Applications*) donde presenta aspectos de mejora con respecto a JSF tales como su eficiencia en tiempo de desarrollo, es una tecnología sofisticada en herramientas gráficas interactivas para el usuario y respecto al problema de seguridad en tiempo de ejecución se solventa ya que es código binario el que se envía al navegador actuando como un applet a diferencia de texto plano HTML como lo realiza JSF. (Ver Sección 1.1 Justificación). [9] [11]

Existe un trabajo previo de integración entre la tecnología JavaFX2 y JavaEE6 utilizando servicios web basados en SOAP para su comunicación, este trabajo está enmarcado en el programa social PROSOFI de la Pontificia Universidad Javeriana acerca de un gestor de clubes de matemáticas llamado CLUBMAT; el cual dejó una base importante en la integración de estas tecnologías y del desarrollo del sistema de información web CLUBMAT. [4] [3].

En este proyecto se realizó una comparación y un estudio acerca de los servicios web basados en SOAP aplicados en el trabajo previo de CLUBMAT y los servicios web basados en REST (*Representational State Transfer*) [40] donde se maneja un paradigma diferente de comunicación.

Se determinó que los servicios web basados en REST tienen ventajas respecto a SOAP en el desempeño y rendimiento, factores que influyeron en la decisión de adaptar la aplicación CLUBMAT pasando de utilizar servicios web SOAP a utilizar servicios web REST, para la

comunicación e integración de las tecnologías JavaFX2 y JavaEE6 para su desarrollo. (Ver Sección: 5.1 Integración JavaFX2 y JavaEE6 con servicios web basados en REST) [30] [38].

La aplicación CLUBMAT consiste en un sistema de información web que gestiona clubes de matemáticas para la comunidad de Usme.

Entre las funciones principales de este sistema se encuentra; administrar usuarios, clubes, instituciones, crear olimpiadas, realizar pruebas para los estudiantes, registrar el desempeño de los estudiantes en las pruebas, registrar el progreso de formalización de los clubes de matemáticas, publicar noticias a la comunidad y eventos, entre otros.(Ver Sección: 3. Marco Contextual).

En su primera versión CLUBMAT fue desarrollada en los módulos de seguridad y clubes de matemáticas, esta versión se realizó integrando JavaEE6 y JavaFX2 con servicios web basados en SOAP.

El presente trabajo de grado consiste en la extensión del trabajo de grado anterior, finalizando los módulos y requerimientos del cliente en su totalidad, de la aplicación CLUBMAT.

Así mismo, se realizó el estudio de los servicios web SOAP y REST para posteriormente realizar la adaptación y mejoras respecto a la arquitectura presentada en la primera versión, utilizando servicios web basados en REST para su comunicación debido a sus mejoras respecto a SOAP.

Este documento describe el estudio de las tecnologías JavaFX2 y JavaEE6 en su integración para aplicaciones empresariales web, el análisis entre los servicios web basados en REST y SOAP, para la mejora en la comunicación, prestación de servicios y el desarrollo de la aplicación CLUBMAT, en su segunda fase completando la totalidad de los requerimientos del sistema.

Se elaboró un tutorial de cómo realizar cualquier aplicación empresarial web en estas tecnologías, con servicios web basados en REST para el desarrollo futuro de aplicaciones empresariales aprovechando las bondades de estas tecnologías.

INTRODUCCIÓN

Con la intención de explorar nuevas tecnologías para el desarrollo de aplicaciones web empresariales, se realizó un trabajo de grado previo donde se integraron las tecnologías JavaFX2 y JavaEE6 [1][2] con servicios web basados en SOAP enfocados en un sistema de información web para la gestión de clubes de matemáticas de la comunidad de Usme enmarcados en el programa social PROSOFI de la Pontificia Universidad Javeriana. [4]

Esta intención surgió de buscar mejoras en la arquitectura JavaEE6 con respecto a su framework estándar de desarrollo de interfaces de usuario JSF [11].

En el presente, se realizó un estudio detallado entre JSF y JavaFX2 para identificar las bondades que tiene javaFX2, para la realización de aplicaciones empresariales robustas solventando problemáticas presentadas por el frameworkJSF. [14]Así mismo, se realizó un estudio comparativo de los servicios web basados con protocolo SOAP y los servicios web REST.

Este estudio se hizo con la finalidad de comparar estas tecnologías y sustentar la adaptación o la mejora de la arquitectura del sistema CLUBMAT.Como resultado del estudio, se decidió por la adaptación y cambio de la arquitectura previa del sistema de información CLUBMAT para integrar JavaFX2 y JavaEE6 con servicios web basados en REST.

En el presente documento se verá el desarrollo de estas tecnologías y los resultados obtenidos para constituir una arquitectura másrobusta que solventa objetivos arquitecturales como el desempeño, rendimiento, seguridad, entre otras. (Ver Sección: 5.1 Integración JavaFX2 y JavaEE6 con servicios web basados en REST)

Por lo cual , el presente trabajo de grado presenta el desarrollo de la aplicación empresarial web CLUBMAT en su totalidad respecto a los requerimientos del cliente adaptada a la nueva arquitectura. Finalmente se dejaun tutorial en la realización de aplicaciones empresariales web con tecnologías JavaFX2 y JavaEE6 con servicios web basados en REST.

Este documento esta dividido por las siguientes secciones:

- Descripción general del trabajo de grado, donde se explica la propuesta y la metodología que se propuso para cumplir con los objetivos.
- Marco Teorico: Donde se contextualiza al lector de los temas involucrados en el desarrollo del trabajo de grado.
- Desarrollo del trabajo: En esta sección se explica el paso a paso de los aportes y el desarrollo del trabajo de grado.
- Resultados y Conclusiones: se presentan los resultados adquiridos por el público objetivo y las conclusiones del trabajo realizado.

I - DESCRIPCIÓN GENERAL DEL TRABAJO DE GRADO

1. Oportunidad, Problemática, Antecedentes

La problemática, son los inconvenientes que tienen las páginas web tradicionales y el interés de buscar nuevas tecnologías para construir las interfaces usuario de las aplicaciones web. Hay importantes aspectos que podrían ser mejorados respecto a la interactividad, navegabilidad, el desempeño, entre otros (Ver Sección 1.1: Justificación).

Por esta razón, hoy en día, una opción para dar solución a estas problemáticas son las aplicaciones RIA (*Rich Internet Applications*) que constituyen la nueva generación de aplicaciones Web cuyo principal foco de atención es el usuario.

Es un nuevo tipo de aplicación con más ventajas que las tradicionales, entre ellas se puede destacar la mejora en la arquitectura, para lograr efectos eficientes y acortar la brecha entre las aplicaciones Web y las de escritorio [15].

En el Departamento de Ingeniería de Sistemas de la Pontificia Universidad Javeriana se realizó un proyecto de grado enfocado a la integración entre dos tecnologías (Java EE 6 y Java FX 2), para el desarrollo de aplicativos web de alto nivel y ricas a nivel de presentación para el cliente [4]. Java FX2 es una herramienta RIA que trae muchos beneficios y fue elegida para el desarrollo del proyecto (Ver Sección 1.1: Justificación).

De esta forma se dejó una solución para la integración de JavaFX2 y JavaEE6 utilizando servicios web con protocolo SOAP para el desarrollo de aplicaciones empresariales web.

Fue necesario retomar el estudio realizado por la Pontificia Universidad Javeriana y adaptarlo para mejorar su arquitectura [4]. Esto con la intención de dejar una base importante para el desarrollo de proyectos futuros en estas tecnologías que brindan grandes beneficios a nivel de presentación e infraestructura. Esta problemática vino acompañada del caso de estudio en el cual se desarrolla el proyecto, la oportunidad para aplicar los conocimientos, y acoplar las tecnologías y la necesidad de apoyar a los niños en el aprendizaje de las matemáticas.

El programa social PROSOFI, se han venido realizando estudios en la comunidad de Usme acerca de este tema [3] y, a través de la experiencia se ha determinado que en los niños entre 10 y 12 años existen temores, generados por colectivos imaginarios, tales como; que las matemáticas son difíciles, no poder aprobar las pruebas y así mismo los grados superiores, ni los exámenes del estado (Icfes). Arrojando resultados preocupantes y negativos respecto a esta problemática en la comunidad infantil [3].

La comunidad ha estado trabajando con clubes de matemáticas donde se intenta motivar y cautivar a los niños en el aprendizaje brindando un espacio agradable y atacando este tema desde perspectivas que habitualmente no están presentes en la práctica escolar: el experimental, lúdico, recreativo y cultural. [36]

Las instituciones de esta comunidad se han reunido con PROSOFI y ya se tiene el compromiso de las dos partes para realizar la gestión de estos clubes de matemáticas con herramientas informáticas para seguir motivando esta iniciativa, por lo cual se generó la necesidad de la construcción de un aplicativo para gestionar estos clubes y cumplir con los compromisos pactados, todo en pro del desarrollo de los niños en esta área de estudio. [3]

En conclusión este es el caso de estudio, donde se aplicó la problemática que se tiene del acoplamiento de las tecnologías y el desarrollo de una aplicación rica a nivel de infraestructura y presentación. Esta oportunidad fue ideal para aplicar el estudio de la integración entre JavaFX2 y JavaEE6. Así mismo, se dejó una base importante para trabajos futuros y se brindó un beneficio a la comunidad con la aplicación.

1.2 Formulación del problema que se resolvió

¿De qué manera se puede extender un sistema de información web realizado en tecnologías JavaFX2 Y JavaEE 6, para el fortalecimiento de clubes escolares de matemáticas en colegios de la localidad de Usme, enmarcado dentro del Programa Social PROSOFI de la Pontificia Universidad Javeriana?

1.3 Justificación

Respecto a las tecnologías para desarrollar el proyecto, Java EE 6 es la versión de Java Empresarial que se está utilizando actualmente para desarrollar y ejecutar software de aplicaciones web en el lenguaje de programación Java (aunque existe una nueva versión JavaEE7 de hace algunos meses que no se consideró en el presente proyecto por el riesgo que supone su reciente publicación). [17] [18]

Para la parte de presentación, como framework gráfico se utilizó JavaFX 2 [1], en lugar de JSF (*Java Server Faces*) que hace parte del estándar de Java EE 6 [2]. Con esta herramienta RIA se solucionaron varios problemas presentes en las interfaces JSF, de los cuales se pueden resaltar los siguientes:

- Dado que la plataforma JavaFX 2 está escrito en Java, los desarrolladores de Java pueden aprovechar sus habilidades existentes y las herramientas para desarrollar aplicaciones en JavaFX2 en un menor tiempo, permitiendo el uso de cualquier librería de Java en una aplicación JavaFX2. [23][24]
- Se caracteriza por hacer una separación limpia de aplicación de interfaz de usuario y lógica simplificando el mantenimiento del código. [20]
- Integra los contenidos web y medios de comunicación sin problemas en la aplicación Java. [6] [24]
- Permite crear interfaces de usuario sofisticadas y ricas en gráficos sin problemas de rendimiento. Las interfaces gráficas basadas en HTML (Como lo es JSF) muchas veces son muy planas y poco interesantes para los usuarios. Java FX2 [1] es rico en componentes gráficos haciendo la interacción con el cliente más amigable y visualmente atractiva [6] [19].
- Permite a los desarrolladores integrar gráficos vectoriales, animación, sonido y activos web de vídeo en una aplicación interactiva completa [1] [6].
- Amplía la tecnología Java y permite mantener un eficaz flujo de trabajo entre diseñador y desarrollador en el que los diseñadores pueden trabajar en las herramientas que deseen mientras colaboran con los desarrolladores [11] [6].
- En tiempo de ejecución, las páginas web con una interfaz gráfica basada en HTML como JSF se pueden alterar en su contenido. Con JavaFX2, se tendrá más control de seguridad ya que es el bytecode el que viaja al navegador.[1] [6] [11].

Estas características justifican el desarrollo de aplicaciones web con Java FX 2 en lugar de JSF.

El caso de estudio para implementar este proyecto, es decir el soporte a los clubes de Matemáticas, tiene su propósito y su justificación propia, es importante destacarla ya que este propósito es fundamental para encaminar el proyecto.

Se justifica realizar este proyecto por la parte social, ya que se utilizará como estrategia para facilitar el aprendizaje de las matemáticas en niños entre los 10 y 12 años.

Les brindará una plataforma de apoyo y abierta a las comunidades que tengan la misma necesidad. Es específicamente para el desarrollo de pruebas matemáticas a nivel interno y externo de las instituciones y el estado.

1.4 Impacto Esperado

Respecto al acoplamiento y desarrollo del proyecto en las tecnologías propuestas (Java EE6 y JavaFX2), el impacto se enfoca en la experiencia que dejará el uso de estas herramientas y la base que dejará el proyecto para otras aplicaciones futuras.

La apropiación de las herramientas va a permitir el desarrollo de aplicaciones de este tipo con más agilidad y facilidad teniendo en cuenta la base que se dejó en este proyecto.

La apropiación de la tecnología es el impacto principal que dejó la realización de este proyecto respecto a las tecnologías dado que tiene buena proyección a futuro y no se tiene un estándar en la realización de aplicaciones web combinando estas tecnologías. Existe mucha información dispersa acerca del tema y el ideal es unificar estos conocimientos en una aplicación y generar una base importante para desarrollos futuros, dejando una pauta y una base para la construcción de proyectos aprovechando las bondades de estas tecnologías.

Su realización proveerá gran experiencia en el manejo de tecnologías como Java FX2, Java EE6 entre otras, para el desarrollo de aplicaciones empresariales de negocio de alto desempeño y ricas a nivel de presentación para el usuario.

Este proyecto tendrá impacto no solamente en la parte tecnológica, sino en nuestro caso de estudio. Es importante destacar esto ya que este proyecto es una oportunidad para solucionar una problemática importante y tendrá impacto directo sobre una población.

El proyecto centrado en los clubes de matemáticas tendrá impacto directo sobre niñas y niños de tres instituciones educativas del sector Bolonia en Usme, Bogotá. [3]. La ilustración 1 refleja los datos de la población:

Rango de edad	Hombre	Mujer	No personas	Características
6 – 11 años	35	30	65	Niñas y niños de estratos 1 y 2.
12 – 17 años	5	5	10	
TOTAL	40	35	75	

Ilustración 1: Datos estadísticos estudios de PROSOFI a la comunidad de Usme

Los impactos esperados con la realización y éxito del proyecto son los siguientes:

- Tres Clubes de Matemáticas implementados sosteniblemente, uno en cada institución asociada.
- Una herramienta informática para la creación y gestión de clubes de matemáticas de 5° grado.

- Tres Clubes de Matemáticas administrados según la plataforma de Gestión implementada.
- Una herramienta informática que permita crear, alimentar y administrar un banco de preguntas para ser utilizadas en olimpiadas matemáticas de 5°.
- Realización ágil, organizada y estándar de las Olimpiadas de Matemáticas.
- Clubes de Matemáticas incluidos entre las actividades extracurriculares de los alumnos de 5°, o como alternativa de uso del tiempo postmeridiano previsto para los colegios públicos por el Plan Distrital de Educación “Jornada Extendida”.

Así mismo el proyecto **es replicable** sin restricción a otras poblaciones de niñas y niños con la misma escolaridad. Con esto se espera su aplicación en otras comunidades que presenten la misma necesidad.

2. Descripción del Proyecto

En esta sección se definirán la visión global del proyecto y su objetivo general y específicos como la respecta metodología utilizada para su desarrollo.

2.1 Visión global

En el presente trabajo, se realizó una aplicación web de un gestor de clubes de matemáticas para niños de 10 a 12 años de la comunidad de Usme CLUBMAT. Esta aplicación fue desarrollada acoplando la arquitectura JavaEE6 y el framework RIA JAVAFX2 [1] con servicios web basados en REST [30], dejando un estándar para aplicaciones empresariales futuras aprovechando los beneficios que brindan estas arquitecturas y frameworks.

En esta aplicación se puede gestionar todo lo que abarca un club de matemáticas respecto a usuarios, roles, instituciones, clubes, olimpiadas, pruebas, guías, noticias, etc. (Ver Sección: 3.Marco Contextual).

Para la realización de este trabajo de grado se retomo un trabajo de grado previo [4] donde se realizó la integración entre JavaEE6 y JavaFX2 utilizando servicios web con protocolo SOAP. Adicional a la integración de JavaFX2 y JavaEE6 cumplió con un 20% de los casos de uso planteados del sistema.

El trabajo previo fue de gran soporte para la realización de este trabajo y como complemento a los requerimientos ya realizados previamente.

2.3 Objetivo general

Extender la aplicación CLUBMAT como un sistema de información web, que permita la creación y administración de clubes de matemáticas para niños de 10 a 12 años utilizando nuevas tecnologías de alto desempeño (JavaFX2 y JavaEE6) a nivel de infraestructura y presentación.

2.4 Objetivos específicos

Los objetivos específicos que tendrá el proyecto son los siguientes:

1. Soporte de los requerimientos previamente obtenidos acerca de la aplicación CLUBMAT.
2. Aplicar servicios web basados en REST a la aplicación CLUBMAT para la conectividad de la misma, modificando la conectividad del proyecto previo, realizada con servicios web con protocolo SOAP.[4]
3. Diseñar la arquitectura e interfaz de usuario de la aplicación CLUBMAT como extensión o modificación del proyecto previo.[4]
4. Implementar todos los requerimientos funcionales y no funcionales de la aplicación CLUBMAT acordados previamente.
5. Realizar pruebas funcionales y de aceptación por parte de los *Stakeholders*.

2.5 Método que se propuso para satisfacer cada fase metodológica

Para la realización de este proyecto se aplicarán prácticas de metodologías de desarrollo ágil de software.

A grandes rasgos, estos son métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Así mismo enfatizan las comunicaciones cara a cara en vez de la documentación y el software desarrollado en una unidad de tiempo es llamado una iteración, el cual comprende los procesos de ciclo de vida determinados por los desarrolladores. [8]

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología es adaptada al contexto del proyecto y en este caso

particular se tomaron prácticas de la metodología SCRUM [8][11] como base para el desarrollo del proyecto adaptando las prácticas de la metodología al proyecto. [8]

SCRUM es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar colaborativamente, y obtener el mejor resultado posible de un proyecto. [8]

En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos y la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. [8]

Se decidió utilizar esta metodología ágil ya que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito y se hacen en el transcurso del tiempo. [8] Como se decía anteriormente, se tomaron las prácticas de la metodología pero no en su totalidad. Por esto, el único aspecto que no aplicó al proyecto y que se adaptó es el desarrollo en grupo, ya que este proyecto se realizó por una sola persona, las reuniones (“*DailyScrum Meeting*”) se dejaron a un lado y la retroalimentación se realizó con los *Stakeholders* en las fechas de entrega de cada Sprint (Ver Anexos Actas).

Basado en esto, la estructura de la metodología asociada a los objetivos específicos se indica a continuación.

2.7 Fases Metodologicas

Respecto a la metodología los procesos que se realizarán se muestran en la Ilustración 2:



Ilustración 2: Detalles de las fases metodológicas SCRUM y las actividades

2.7.1 Fase de Planeación

Esta fase está asociada a los objetivos específicos 1,2 donde se tuvo en cuenta la obtención de requerimientos de todo el sistema y el presente proyecto modifica la aplicación dejada por el trabajo previo.

utilizando servicios web basados en REST para el manejo de la conectividad de la aplicación [4]

En esta fase se adquirieron los requerimientos previamente obtenidos por PROSOFI, y se documentaron para priorizarlos posteriormente.

En términos de la metodología se le denomina establecer el **ProductBacklog**[8] que se refiere a la lista de requerimientos primarios que representan, la visión y expectativas del cliente respecto a los objetivos y entregas del proyecto.

Con esta lista se tendrán los requerimientos priorizados y así mismo los que de acuerdo al alcance se van a implementar.

2.7.2 Fase de Preparación y Desarrollo

Esta fase comprende los objetivos específicos 2, 3, 4 y 5 determinados anteriormente.

Involucra la aplicación de los servicios web basados en REST, la extensión y modificación del diseño, arquitectura e implementación inicial de CLUBMAT y las pruebas funcionales a cada Sprint que se realice. En esta fase se maneja el concepto de Sprint Backlog[8] [11] ,donde Sprint Backlog es un subconjunto del ProductBacklog que se acepta completar en un sprint o iteración (por tareas) y se determina de acuerdo a la prioridad de los requerimientos. En cada Sprint se realizaron los siguientes procesos:

- **Proceso de Preparación:**

Se determinó la lista de tareas que van a constituir el Sprint Backlog de acuerdo a la priorización de los requerimientos.

- **Proceso de Diseño:**

El diseño de la aplicación es el proceso mediante el cual se traducen las tareas del Sprint Backlog en una representación del software. El diseño es la base de todas las partes posteriores del desarrollo y de la fase de pruebas. La metodología Scrum sugiere que hay que conse-

guir diseños simples y sencillos. [11] El proceso de diseño se dividió en 4 partes fundamentales:

- **Diseño de Datos:** En esta división se determinaron las entidades de negocio asociadas a los requerimientos asociados al Sprint como una extensión del trabajo previo.
- **Diseño Arquitectónico:** Como su nombre lo indica, se definió la posible arquitectura del software, los componentes que la conforman y la relación entre ellos, extendiendo y modificando el trabajo previo.
- **Diseño Procedimental:** Esta parte del diseño se refiere a la secuencia de los procedimientos, el orden en el que se van a ejecutar y la interacción con las entidades de negocio y componentes asociados.
- **Diseño de Interfaz:** Finalmente se diseñaron las pantallas de usuario de la aplicación por cada Sprint.

- **Proceso de Desarrollo:**

Esta fase comprende el objetivo específico del desarrollo de la aplicación en términos de requerimientos funcionales y no funcionales, en la programación de sus componentes y la integración de los mismos.

Como su nombre lo indica es el paso del diseño al desarrollo y codificación. Esto se realiza respecto a cada requerimiento y lista de tareas del Sprint Backlog. Respecto a la integración del software, en esta metodología (SCRUM) [8] se ha de hacer una integración continua, es decir, cada vez se tienen que ir integrando pequeños fragmentos de código, para evitar que al finalizar el proyecto se tenga que invertir grandes esfuerzos en una integración final.[8] [11]

- **Proceso de Pruebas**

Este proceso se realizarón pruebas funcionales por cada Sprint realizado y se documentarón los resultados para integrarlo con las demás funcionalidades del sistema.

2.7.3 Fase de Entrega

Esta fase está asociada al objetivo específico 5, que determina las pruebas de aceptación por parte de los Stakeholders de todo el sistema.

En esta fase se maneja el concepto de ReleaseBacklog[8] que es el producto resultante del Sprint Backlog.

En esta fase se realizarón las pruebas de aceptación directamente con los Stakeholders buscando la aprobación del software de acuerdo a lo acordado previamente en el ProductBacklog.

Asi mismo, se continuocon el desarrollo del manual de usuario para el uso del software por parte de los clientes, manual de instalación y manual de mantenimiento por si en un futuro se desea continuar con el desarrollo y mejora del proyecto.

2.8 Actividades y Entregables

La siguiente tabla (Tabla 1) muestra los objetivos específicos asociados con sus actividades y los entregables que generarán cada uno de ellos.

OBJETIVO GENERAL	OBJETIVOS ESPECIFICOS	ACTIVIDADES	ENTREGABLES	
Extender la aplicación CLUBMAT como un sistema de información web, que permita la creación y administración de clubes de matemáticas para niños de 10 a 12 años utilizando nuevas tecnologías de alto desempeño	1. Obtener los requerimientos por parte de los stakeholders acerca de la aplicación deseada.	Adquirir los requerimientos de usuario	<ul style="list-style-type: none"> • Documentación requerimientos de cliente • Documentación de casos de uso. • Tutorial y Demo funcional de acoplamiento entre JavaEE6 y JavaFX2 con servicios web basados en REST. 	
	2. Aplicarservicios web basados en REST a la aplicación CLUBMAT para la conectividad de la aplicación modificando la conectividad del proyecto previo realizada con servicios web con protocolo SOAP.	Priorizar los requerimientos		
		Estudiar las tecnologías, herramientas y prácticas asociadas		Explorar y adaptar trabajo previo de acoplamiento entre las tecnologías [4]

(JavaFX2 y JavaEE6) a nivel de infraestructura y presentación.	3. Diseñar la arquitectura y la interfaz de usuario de la aplicación.	Diseño de datos (Entidades de negocio)	<ul style="list-style-type: none"> • Arquitectura de la aplicación CLUBMAT • Prototipos de interfaz de usuario
		Diseñar la arquitectura	
		Diseñar los procedimientos	
		Diseñar las interfaces de usuario	
	4. Implementar un subconjunto de requerimientos funcionales y no funcionales de la aplicación CLUBMAT	Desarrollar cada módulo independiente (Sprint Backlog)	<ul style="list-style-type: none"> • Implementación en código de cada módulo • Software unificado
		Integrar módulos	
	5. Realizar pruebas funcionales y de aceptación por parte de los Stakeholders	Realizar pruebas funcionales	<ul style="list-style-type: none"> • Release Backlog • Registro de resultados de pruebas funcionales por modulo • Registro de resultados de pruebas de aceptación del software integrado • Manual de usuario • Manual de mantenimiento
		Realizar pruebas de aceptación	

Tabla 1: Actividades y Entregables

Adicional a los entregables por cada objetivo específico enunciados anteriormente se entregarán los siguientes documentos:

- Memoria del trabajo de grado
- Actas de las iteraciones (Sprints) en las fases de preparación y desarrollo.

- Se debe aclarar que la validación de la aplicación se realizó en un colegio de los involucrados. Se entregó una carta de aceptación del producto por parte del Director de la institución como complemento a los entregables del proyecto de grado.

II - MARCO TEÓRICO

En esta sección se encuentra la información relacionada con el marco contextual y conceptual del proyecto, las bases y la introducción a todo el global del proyecto.

3. Marco Contextual

El marco contextual hace referencia como su nombre lo indica al contexto del proyecto y todo lo que enmarca su realización respecto al contexto.

3.1 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 se tiene 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.

Aquí 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[3], se concluyó que estos se deben componer de las siguientes partes para un buen desarrollo y funcionamiento:[3]

- **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.

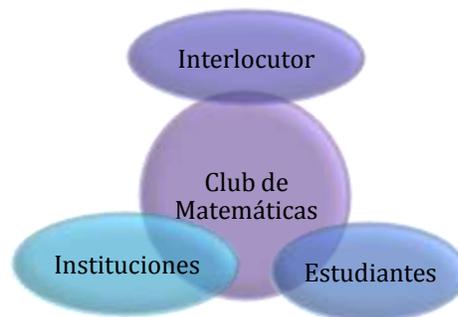


Ilustración 3: Composición de un club de matemáticas

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. [3][36]

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

3.1.1 Lista de reglas para formalizar un club de matemáticas.

A continuación se enumeran los pasos necesarios para poder crear un club de matemáticas [3]:

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.

3.2 Clubmat Version 1.0



Ilustración 4: CLUBMAT 1.0

Al iniciar este proyecto de grado se tenía la base y la experiencia de un trabajo de grado previo. Las actividades más representativas para este trabajo de grado, se apoyan en la labor realizada por el estudiante Andrés Sánchez llamado "Acoplar en la plataforma Java EE 6 el Framework JavaFX 2 como tecnología de interfaz gráfica". [4]. La ilustración 4 representa la primera versión del sistema CLUBMAT.

Este proyecto se realizó cumpliendo las mismas tecnologías pero con el protocolo de comunicación SOAP utilizando servicios web.

El trabajo previo dejó una parte del conocimiento en las tecnologías, pero se realizó un estudio posterior donde se determinó que la mejor opción para el desarrollo de esta aplicación sería utilizando servicios web basados en REST.

La contribución de este trabajo de grado fue importante ya que empezó con una base importante el desarrollo y el aprendizaje de estas herramientas. No obstante la segunda versión de CLUBMAT representada por el presente trabajo de grado, tiene un enfoque distinto debido a cambios en tecnologías y requerimientos por parte del cliente.

4. Marco Conceptual

En el marco conceptual se encuentran las definiciones de los conceptos relevantes para el proyecto.

4.1 Aplicaciones Empresariales Web

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. [29]

De igual forma, el navegador utiliza el protocolo de transferencia de información HTTP (Hypertext Transfer Protocol) que permite la conexión con el servidor. [29]

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 correctamente. 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. [13]

Dentro de las tecnologías sin servidor de aplicaciones encontramos herramientas como: PHP, Python y Perl. [29] Estas tecnologías dependen de un servidor web.

Por otro lado, en las tecnologías con servidor de aplicaciones encontramos las siguientes plataformas: Java EE 6 [2] y .NET 4.

Un servidor de aplicaciones tiene las siguientes ventajas: [14] [15]

- La inclusión de un middleware que facilita la comunicación entre varios servicios. [14]
- Soporte a diferentes estándares como HTML, XHTML y XML permitiendo el funcionamiento en ambientes web.[15]
- Alta disponibilidad de sus aplicaciones, además garantiza seguridad y confiabilidad de las mismas.[14]
- Brinda escalabilidad y mantenibilidad en tal caso de que la carga de trabajo incremente. [15]
- 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. [2] La razón de esta decisión es que el objetivo principal de este trabajo de grado es realizar una integración del framework gráfico JavaFX 2 [1] sobre la mencionada plataforma.

4.1.1 JavaEE6

El objetivo más importante de la plataforma Java EE 6 [2] es simplificar el desarrollo cubriendo una amplia gama de necesidades para las aplicaciones empresariales como lo son: transaccionalidad, seguridad, interoperabilidad, escalabilidad, persistencia, objetos distribuidos, entre otros. [5] [17]

La plataforma Java EE 6 [2] 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 [5], JavaServer Faces (JSF) 2.0 [5], Enterprise JavaBeans (EJB) 3.1 [5], Java Persistence (App) 2.0 [28], Context and DependencyInjection (CDI) 1.0 [5] y mucho más.

La plataforma Java EE 6 incluye las siguientes características nuevas: [5]

- 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. [29]
- Nuevas tecnologías, incluyendo las siguientes: [29]

- ✓ 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
 - ✓ Inyección de dependencias para Java (JSR 330)
 - ✓ Validación de Beans (JSR 303)
 - ✓ Java AuthenticationServiceProvidier Interface forContainers (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. [13]

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: [2]

- 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. La ilustración 5 da una idea de la arquitectura general de JavaEE6.

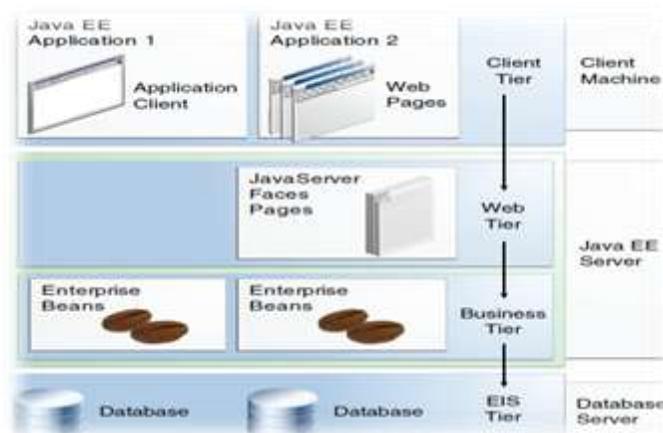


Ilustración 5: Arquitectura JAVAEE6

4.1.2 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. [28]

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

- 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 [28].

JPA 2 es una abstracción que está por encima de JDBC lo que permite ser independiente de SQL. [28]

4.1.3 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. [29]

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. [29] [2].

4.1.4 Glassfish

GlassFish Server es un servidor de aplicaciones de referencia para los proveedores de Java EE 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.[14]

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

- GlassFish es un proyecto Open Source modular que permite incluir sus librerías como parte de otros frameworks, toolkits y productos. [14]
- Tiene soporte comercial disponible.
- Glassfish es de fácil uso y no consume demasiados recursos. [14]
- 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. [14]
- 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. [14]
- 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. [15]
- Administración centralizada, permite gestionar clúster de servidores de aplicaciones desde una consola centralizada. [15]
- 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. [15]

4.2 Herramientas RIA y Java FX2

En la actualidad existen frameworks y herramientas RIA (Rich Internet Applications) similares a JavaFX2 como Silverlight y Flash que comparten características ricas en interfaz de usuario y todas buscan mejorar la experiencia del usuario, y aquí es donde es importante justificar por qué se estableció el uso de Java FX 2 para este proyecto por encima de otras herramientas RIA. [9]

JavaFX2 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. [1]

Similar a Silverlight [7] (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. Desde JavaFX 2 se puede utilizar cualquier clase o paquete Java. [6]. 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.

4.2.1 Características de JavaFX 2

- 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.

En la ilustración 6 se pueden ver fotos que demuestran las características gráficas de JavaFX 2:

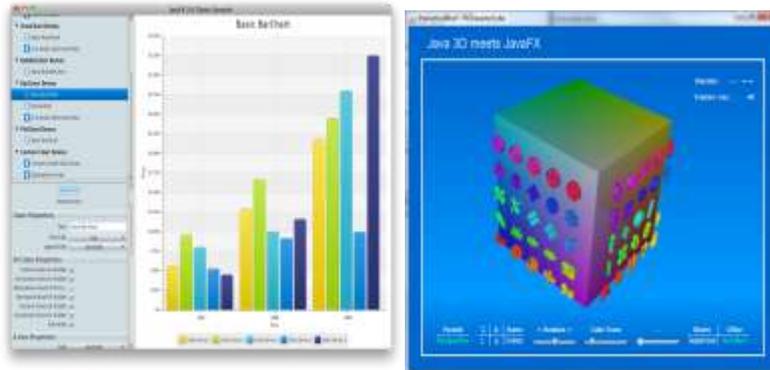


Ilustración 6: Ejemplos aplicaciones javafx2

Las razones fundamentales están dadas por 3 ventajas que se pueden destacar de esta herramienta en comparación con las demás herramientas RIA (interacción con el cliente, mantenibilidad y seguridad) que dan un valor agregado a las aplicaciones que se realicen en ella. [9]:

Interacción con el cliente: Java FX2 presenta propiedades adicionales a sus propiedades multimedia y de presentación visualmente atractiva. La agilidad, el rendimiento y la interacción con el cliente.[9]

Mantenibilidad: Debido a su proyección y siendo una herramienta competitiva a nivel de herramientas RIA no requiere gran cantidad de esfuerzo para conservar su funcionamiento normal ya que se mantendrá en el tiempo y no se requerirán de grandes esfuerzos para sostenerse o restituirse.[9] [11].

Seguridad: La modificación en tiempo de ejecución de las páginas web con una interfaz gráfica basada en HTML como JSF es posible el contenido de una página web. Con JavaFX 2, se logra más control de seguridad ya que es el bytecode el que viaja al navegador y no el texto HTML. [9] [11].

4.2.2 JavaFX Scene Builder

JavaFXSceneBuilder[23] 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.

Estas son algunas de las características principales de "JavaFXSceneBuilder":

- El diseño generado es fácil de usar ya que maneja una paleta de componentes donde se arrastran los componentes que se quieren utilizar en la pantalla y ponerlas en la posición deseada a diferencia de JSF. (Ver Ilustración 7)
- El software permite obtener una vista previa del diseño realizado.
- Admite formato CSS.

La ilustración 7 hace referencia al editor JavaFX2 con sus componentes para construir la interfaz grafica de usuario.



Ilustración 7: JavaFXScenebuilder

4.3 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.[39]

PostgreSQL es un intérprete del lenguaje SQL y compatible en mayor parte del estándar SQL. [37]

Ofrece las siguientes características:

- 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. [37]

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[28].

4.3 Servicios Web

4.3.1 ¿Qué es un servicio web?

Es importante aclarar el concepto de servicio web para entender los conceptos de SOAP y REST. El consorcio W3C define los Servicios Web como sistemas software diseñados para soportar una interacción interoperable maquina a maquina sobre una red. [30]

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.[30]

Los Servicios Web suelen ser APIs Web que pueden ser accedidas dentro de una red (principalmente Internet) y son ejecutados en el sistema que los aloja. [30]

A continuación se listan los estilos de usos más comunes:

- SOAP (Simple Object Access Protocol). Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambios de datos XML, el punto identificativo de SOAP es que las operaciones son definidas como puertos WSDL (Web ServicesDescriptionLanguage) [30].
- RemoteProcedureCalls (RPC, Llamadas a Procedimientos Remotos): Los Servicios Web basados en RPC presentan una interfaz de llamada a procedimientos y funciones distribuidas, lo cual es familiar a muchos desarrolladores.[30]
- Arquitectura Orientada a Servicios (Service-orientedArchitecture, SOA). Los Servicios Web pueden también ser implementados siguiendo los conceptos de la arquitectura SOA, donde la unidad básica de comunicación es el mensaje , más que la operación . Esto es típicamente referenciado como servicios orientados a mensajes.[30]
- REST (REpresentationState Transfer). Los Servicios Web basados en REST intentan emular al protocolo HTTP o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones estándar (por ejemplo GET, PUT,...). Por tanto, este estilo se centra más en interactuar con recursos con estado, que con mensajes y operaciones. [10]

Hoy en día se manejan servicios Web basados en SOAP, pero se trabajará para la conectividad el uso de servicios web basados en REST. Muchos diseñadores de Servicios Web están llegando a la conclusión que SOAP puede ser simple de implementar pero implica overhead de tiempo de ejecución. [10] Por tanto, están comenzando a utilizar Servicios Web basados en REST para mostrar cantidades de datos masivos.

Este es el caso de grandes empresas como eBay y Google [9]. Se pueden identificar características importantes de este estilo de servicio web en comparación con SOAP como las siguientes [11]:

- Bajo consumo de recursos.
- Las instancias del proceso son creadas explícitamente.
- El cliente no necesita información de enrutamiento a partir de la URI inicial.
- Los clientes pueden tener una interfaz “listener” (escuchadora) genérica para las notificaciones.
- Generalmente fácil de construir y adoptar. [10]

En la sección de desarrollo del trabajo se presentará una comparación formal entre SOAP Y REST donde no solo teóricamente se justificara el uso de REST en este proyecto, también se presentaran pruebas funcionales de los protocolos para su demostración.(Ver Sección Desarrollo del trabajo).

A continuación se complementaran los servicios más relevantes para este proyecto en detalle (SOAP y REST).

4.3.2 ¿Qué es SOAP?

SOAP es uno de los protocolos de comunicación existentes para el intercambio de mensajes en los Servicios Web. [10]

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. [10]

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 ServicesDescriptionLanguage) [10]

Es por esto que será aconsejable utilizar este protocolo en entornos donde se establecerá un contrato formal y donde se describirán todas las funciones de la interfaz así como el tipo de datos utilizados tanto de entrada como de salida. El lenguaje WSDL nos permitirá definir claramente cualquier detalle de las funciones de nuestro WS. La ilustración 8 nos muestra la comunicación dada por los servicios web SOAP.

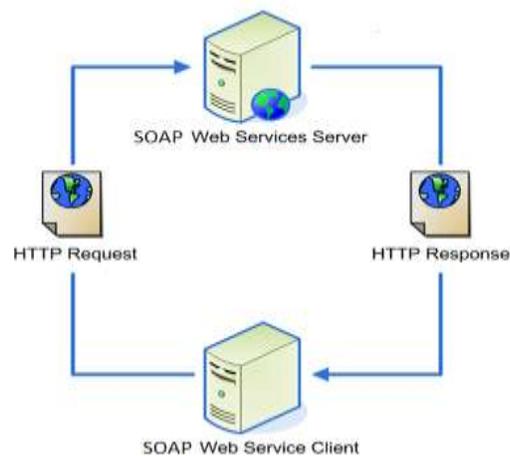


Ilustración 8: Servicios web con protocolo SOAP

4.3.3 ¿Qué es REST?

Las siglas REST significan (RepresentationalState Transfer). Es un estilo de arquitectura de software para sistemas distribuidos tales como la web. A diferencia de SOAP, se centra en el uso de los estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional. [40]

Las operaciones(o funciones) se solicitarán mediante GET, POST, PUT y DELETE, por lo que no requiere de implementaciones especiales para consumir estos servicios. Además se podrá utilizar JSON en vez de XML [40] como contenedor de la información, por lo que será aconsejable utilizar este protocolo cuando busquemos mejorar el rendimiento, o cuando disponemos de escasos recursos, como sería el caso de los dispositivos móviles. [40]

Cada uno de estos métodos determina la acción que hará el REST sobre nuestra aplicación. No debe haber más de un GET o POST o DELETE o PUT, solo tiene que haber uno de cada método. Cada uno tiene una tareaespecífica:

1. GET: Para obtener un valor. Puede ser un listado de objetos
2. POST: Para guardar un nuevo objeto (instancia de identidad) en la aplicación
3. DELETE: Para eliminar un objeto (instancia de identidad)
4. PUT: Para actualizar un objeto.

JSON (JavaScript ObjectNotation), es un formato ligero para el intercambio de datos y XML (extensible MarkupLanguage) ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en

forma legible. Estos formatos son los que manejan estos servicios REST para la transmisión de datos. [40]

En realidad, REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen cómo los recursos son definidos y diseccionados. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional, como hace SOAP. Estos dos significados pueden chocar o incluso solaparse. Es posible diseñar un sistema software de gran tamaño de acuerdo con la arquitectura propuesta por Fielding sin utilizar HTTP o sin interactuar con la Web. Así como también es posible diseñar una simple interfaz XML+HTTP que no sigue los principios REST, y en cambio seguir un modelo RPC [30].

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave: [38]

- **Un protocolo cliente/servidor sin estado:** cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)
- **Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información:** HTTP define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE. Con frecuencia estas operaciones se equiparan a las operaciones CRUD que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.
- **Una sintaxis universal para identificar los recursos.** En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- **El uso de hipertextos**, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

La ilustración 9 nos da un ejemplo de cómo funciona la arquitectura de los servicios REST

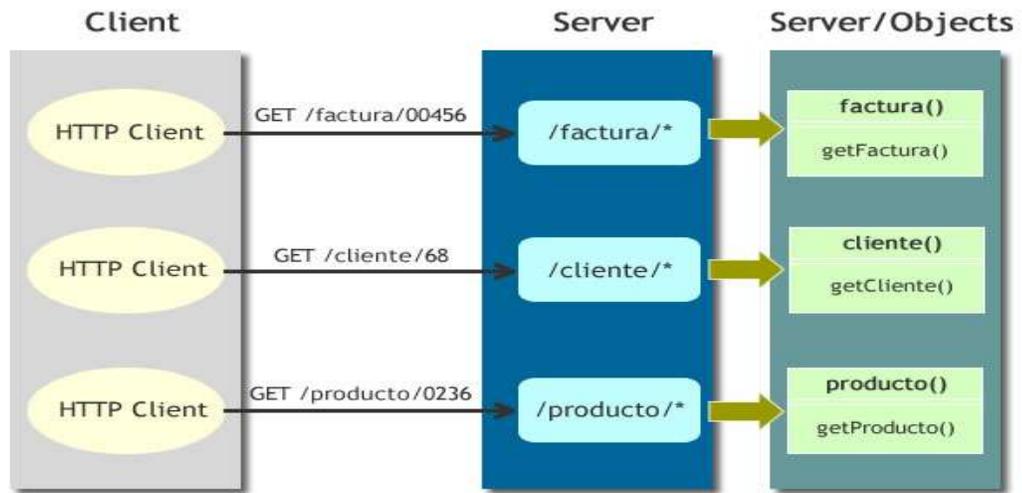


Ilustración 9: Ejemplo Arquitectura REST

III – DESARROLLO DEL TRABAJO

En este capítulo se explicará el proceso que se llevó a cabo para cumplir con el objetivo general propuesto y cada uno de los objetivos específicos desarrollados en las fases metodológicas planteadas anteriormente.

5. Fase de Planeación

En esta fase se realizó la planeación para el desarrollo del sistema CLUBMAT, los requerimientos del sistema y la definición de su arquitectura.

5.1 Integración JavaFX2 y JavaEE6 con servicios web basados en REST

El propósito de esta sección es mostrar el trabajo de investigación acerca de los servicios web REST y dar las razones tanto teóricas como funcionales del porqué se decidió utilizar estos servicios para el sistema CLUBMAT.

En la versión anterior de CLUBMAT, el sistema utilizó servicios web SOAP para la integración entre JavaFX2 y JavaEE6. En el presente proyecto se realizó una comparación teórica y práctica acerca de estos servicios web y se determinó el uso de servicios web basado en REST para el desarrollo de la segunda versión de CLUBMAT.

Los resultados se muestran a continuación:

5.1.1 REST vs SOAP

A menudo se tienden a confundir los conceptos ya que REST es un estilo, mientras que los servicios Web son sistemas software. Por tanto, no es posible la comparación de ambos conceptos. [38]

Por otra parte , popularmente se generaliza el concepto de servicio Web con el de servicio Web basado en SOAP . Como hemos visto en apartados anteriores , es posible diseñar servicios Web basados en REST, es decir tomando REST como estilo de diseño. [10]

El principal beneficio de SOAP recae en ser fuertemente acoplado, lo que permite poder ser probado y depurado antes de poner en marcha la aplicación . En cambio , las ventajas de la aproximación basada en REST recaen en la potencial escalabilidad de este tipo de sistemas , así como el acceso con escaso consumo de recursos a sus operaciones debido al limitado número de operaciones y el esquema de direccionamiento unificado [10]

La ilustración 10 se ve una comparación general de REST Y SOAP [10].

	REST	SOAP
Características	Las operaciones se definen en los mensajes. Una dirección única para cada instancia del proceso. Cada objeto soporta las operaciones estándares definidas. Componentes débilmente acoplados.	Las operaciones son definidas como puertos WSDL. Dirección única para todas las operaciones. Múltiple instancias del proceso comparten la misma operación. Componentes fuertemente acoplados.
Ventajas declaradas	Bajo consumo de recursos. Las instancias del proceso son creadas explícitamente. El cliente no necesita información de enrutamiento a partir de la URI inicial. Los clientes pueden tener una interfaz "listener" (escuchadora) genérica para las notificaciones. Generalmente fácil de construir y adoptar.	Fácil (generalmente) de utilizar. La depuración es posible. Las operaciones complejas pueden ser escondidas detrás de una fachada. Envolver APIs existentes es sencillo Incrementa la privacidad. Herramientas de desarrollo.
Posibles desventajas	Gran número de objetos. Manejar el espacio de nombres (URIs) puede ser engorroso. La descripción sintáctica/semántica muy informal (orientada al usuario). Pocas herramientas de desarrollo.	Los clientes necesitan saber las operaciones y su semántica antes del uso. Los clientes necesitan puertos dedicados para diferentes tipos de notificaciones. Las instancias del proceso son creadas implícitamente.

Ilustración 10: REST vs SOAP (TOMADA DE [40])

5.1.2. Clubmat y Comunicación basada en REST

Para el presente proyecto, se ha decidido trabajar con el estilo arquitectural REST, ya que hablando en términos de requerimientos no funcionales se da prioridad al rendimiento de la aplicación.

Como vimos anteriormente, podemos concluir que REST es muy ligero, sus respuestas contienen exactamente la información que necesitamos. Para los nosotros los humanos es muy fácil y simple de interpretar. Es sencillo de desarrollar y no se necesita mucho código extra. Es flexible en cuanto al tipo de respuesta que se necesita, ya que puede ser XML o JSON.

De acuerdo a esto se tomó la decisión de implementar servicios web basados en REST para el desarrollo de la aplicación del presente trabajo de grado teniendo en cuenta los beneficios que brinda ese estilo arquitectural dados los pocos recursos que se tienen para el uso de la aplicación.

A continuación se presentaran unas pruebas funcionales donde se verifica prácticamente lo dicho anteriormente y se apoya la decisión de realizar la aplicación de esta forma.

5.1.3 Pruebas Funcionales de Rendimiento con REST

Adjunto a este documento se encuentra un proyecto DEMO con un CRUD sobre una tabla de datos donde se realizan funciones utilizando SOAP y REST para la comunicación sobre los datos.

Utilizando JavaFX2 para la interfaz gráfica se realizó el acoplamiento con Java EE para realizar el CRUD y realizar pruebas de rendimiento para comprobar el funcionamiento de estos conceptos de comunicación en una aplicación real.

La aplicación muestra la siguiente interfaz de usuario (Ver Ilustración 11):



Ilustración 11: Ventana CRUD pruebas SOAP vs REST

Esta aplicación realiza las acciones de crear, editar y eliminar datos sobre una tabla utilizando las dos estrategias mencionadas anteriormente.

Se realizó una prueba para cargar los datos de la tabla por los dos métodos al iniciar la aplicación y se midieron los tiempos de demora de cada método para comprobar su rendimiento.

La ilustración 12 muestra un llamado a un método del servicio web REST donde trae todas las instancias de la base de datos. El código crea una instancia del servicio Rest e invoca el servicio de **findAll_XML** donde trae todas las instancias de la base de datos en un formato XML y calcula el tiempo que demora desde que se pida la petición hasta que carga todos los datos en este caso haciendo uso del protocolo REST. Finalmente se imprime el tiempo calculado.

```
//LISTAR PROTOCOLO REST

ClienteRest teamClient = new ClienteRest();

long tiempoInicio = System.currentTimeMillis();
ClienteResponse response = teamClient.findAll_XML(ClienteResponse.class);
long totalTiempo = System.currentTimeMillis() - tiempoInicio;
System.out.println("El tiempo de demora REST es : " + totalTiempo + " miliseg");

GenericType<List<Team>> genericType = new GenericType<List<Team>>() {};
List<Team> data = new ArrayList<Team>();
data = (response.getEntity(genericType));
teams = FXCollections.observableArrayList( );
teams.addAll(data);
System.out.println("TEAMS : "+teams.size());
tablaTeams.setItems(teams);
```

Ilustración 12: Código prueba protocolo REST

La ilustración 13 muestra un llamado al método de un servicio web SOAP que carga a memoria todas las instancias igual que el anterior y se cuenta el tiempo desde la petición hasta que carga los datos, El código muestra como se invoca el método **findAll()** (referenciado en el descriptor WSDL), trae todos los datos relacionados y se calcula el tiempo de demora en esta operación haciendo uso del protocolo SOAP. Finalmente se imprime el tiempo calculado.

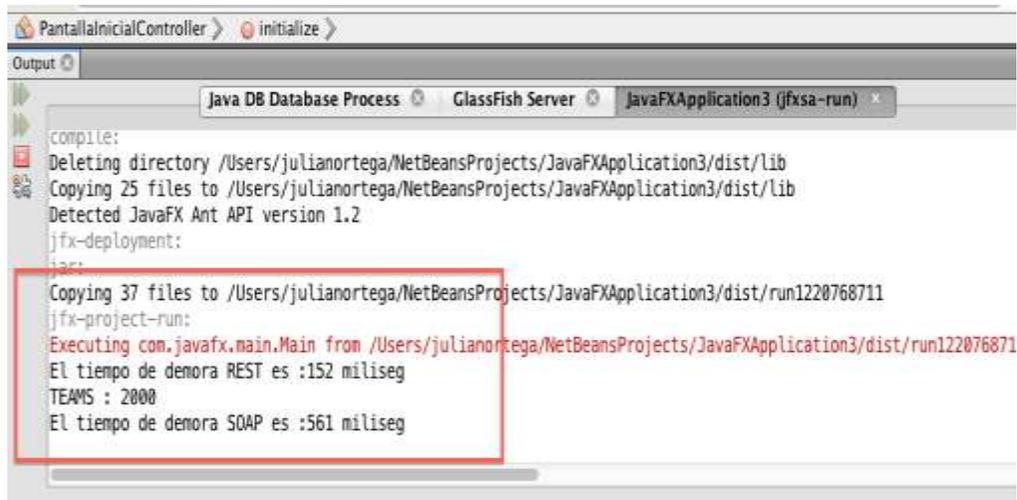
```
//LISTAR PROTOCOLO SOAP

this.columnaIdsoap.setCellValueFactory( new PropertyValueFactory< Team, Integer >( "teamId" ) );
this.columnaNombresoap.setCellValueFactory( new PropertyValueFactory< Team, String >( "teanname" ) );
this.columnaPaisssoap.setCellValueFactory( new PropertyValueFactory< Team, String >( "country" ) );
teamsSOAP = FXCollections.observableArrayList( );

long tiempoInicio2 = System.currentTimeMillis();
teamsSOAP.addAll(findAll());
long totalTiempo2 = System.currentTimeMillis() - tiempoInicio2;
System.out.println("El tiempo de demora SOAP es : " + totalTiempo2 + " miliseg");
tablaTeamsSOAP.setItems(teamsSOAP);
```

Ilustración 13: Código Prueba Protocolo SOAP

La base de datos se cargo con 2000 datos y se cargaron en memoria utilizando los dos procedimientos. Los resultados de rendimiento fueron los presentados en la ilustración 14:



```
Output
Java DB Database Process  GlassFish Server  JavaFXApplication3 (jfxsa-run)
compile:
Deleting directory /Users/julianortega/NetBeansProjects/JavaFXApplication3/dist/lib
Copying 25 files to /Users/julianortega/NetBeansProjects/JavaFXApplication3/dist/lib
Detected JavaFX Ant API version 1.2
jfx-deployment:
jar:
Copying 37 files to /Users/julianortega/NetBeansProjects/JavaFXApplication3/dist/run1220768711
jfx-project-run:
Executing com.javaafx.main.Main from /Users/julianortega/NetBeansProjects/JavaFXApplication3/dist/run122076871
El tiempo de demora REST es :152 miliseg
TEAMS : 2000
El tiempo de demora SOAP es :561 miliseg
```

Ilustración 14: Resultados pruebas REST VS SOAP

Los resultados que se obtuvieron al cargar los 2000 datos con los servicios web fueron los siguientes:

- **Tiempo de demora REST: 152 milisegundos**
- **Tiempo de demora SOAP: 561 milisegundos**

Podemos ver que el rendimiento de REST es casi 5 veces más rápido en esta prueba en comparación a SOAP.

NOTA: Las especificaciones técnicas de la maquina donde se realizaron las pruebas funcionales son las siguientes:

MacBook Pro 13 Pulgadas

Procesador	Doble núcleo a 2,5 GHz
	Core i5 de Intel (Turbo Boost de hasta 3,1 GHz) con 3 MB de caché de nivel 3
	Opción de configuración con Core i7 de Intel de doble núcleo a 2,9 GHz (Turbo Boost de hasta 3,6 GHz) y 4 MB de caché de nivel 3 compartida

Memoria	4 GB de memoria DDR3 a 1.600 MHz
	Opción de configuración con 8 GB

Tabla 7: Características Técnicas maquina de desarrollo

Realizando estas pruebas funcionales se pudo determinar que respecto a rendimiento, funcionalmente con la arquitectura REST se obtiene un beneficio respecto al protocolo SOAP.

Por esta razón se verifican las ventajas mencionadas anteriormente de manera práctica y se decidió continuar con esta arquitectura para el desarrollo del proyecto CLUBMAT ya que las condiciones de rendimiento son fundamentales contando con los recursos que se tienen para el desarrollo y uso de la aplicación.

5.1.4 Conclusiones REST vs SOAP

Es claro que cada concepto tiene sus ventajas y desventajas, por esta razón no se puede decir cual es mejor. Por otro lado, se puede identificar como en este caso cuál es mas conveniente para la realización de aplicaciones de acuerdo a sus características y a los beneficios que nos puedan brindar [30].

Cuando se trata de aplicaciones públicas y servicios de uso masivo es mucho mejor utilizar REST por la sencillez en su implementación y respuestas, inclusive en clientes móviles es mucho más fácil utilizar REST.

Por otro lado si se está pensando en servicios web para corporaciones donde se manejan datos complejos y se necesita una precisión detallada en las respuestas se puede utilizar SOAP. [30]

En este caso se decidió trabajar con REST porque se adapta mejor a las necesidades de la aplicación CLUBMAT y en general estos conceptos de comunicación nos brindan ciertas posibilidades que podemos adaptar y es importante buscar las necesidades que se puedan suplir con las características que cada concepto nos brinda.

5.1.5 Demo y Tutorial para Aplicaciones Futuras.

El propósito de este trabajo de grado no era únicamente realizar la aplicación para el beneficio de los estudiantes de CLUBMAT, también se tenía como propósito dejar una base importante para trabajos futuros y cualquier tipo de aplicación en estas tecnologías.

Como anexo a este trabajo de grado y como resultado del estudio de la integración entre estas tecnologías, se entrega un tutorial funcional de como hacer una aplicación web empresarial con JavaFX2 y JavaEE6 utilizando servicios web REST. (Ver Anexo Tutorial y Demo JavaFX2 y JavaEE6 con servicios web REST)

Este tutorial enseña desde la misma instalación de los programas y sus versiones, hasta la interacción del cliente con la interfaz gráfica hasta la base de datos. Este tutorial será muy útil para las personas que desarrollen este tipo de aplicaciones empresariales web y se deja un aprendizaje importante en esta área.

5.2 Especificaciones del producto CLUBMAT

En este punto se estableció el producto a realizar de acuerdo a los requerimientos y se determinó la arquitectura del software adecuado de acuerdo al estudio de las tecnologías propuestas.

Así mismo, se permitió definir todos los parámetros arquitecturalmente necesarios para desarrollar una aplicación de software apropiada. [11]

El objetivo a cumplir fue definir las características arquitecturales para desarrollar un gestor de clubes de matemáticas CLUBMAT, a partir de los requerimientos establecidos por el cliente y por el entorno.

La definición de las características de la arquitectura se realizará a partir de los requerimientos determinados por el cliente. De este modo, se podrán realizar las vistas lógicas y físicas, para que finalmente se establezcan las tácticas arquitecturales que serán implementadas en el desarrollo de la aplicación.

El propósito de esta fase fue establecer una visión más clara del sistema, definiendo vistas a diferentes niveles, iniciando desde un punto muy general, hasta llegar al detalle y la respectiva especificación.

El alcance de esta fase fue la descripción y el modelado a nivel arquitectural del sistema de gestor de clubes matemáticos CLUBMAT a partir del levantamiento de requerimientos realizados con los casos de uso que la compañía ha puesto a disposición para el desarrollo del sistema.

5.2.1 Modelo del Dominio

En esta sección se explica en que consiste la aplicación CLUBMAT y sus respectivos componentes.

A. ¿Qué es CLUBMAT?

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:

- Publicar noticias de la comunidad CLUBMAT
- Inscribir instituciones y clubes de matematicas
- Controlar el progreso de un club de matematicas.
- Pruebas de olimpiadas, test y practica a estudiantes.
- Olimpiadas e inscribir estudiantes.
- Usuarios con roles para ingresar al sistema
- Banco de preguntas para pruebas y olimpiadas
- Guías de estudio para los estudiantes

Los siguientes componentes hacen parte del dominio de la aplicación.

- Usuarios y roles: Cada usuario tendrá un rol con un diferente alcance dentro de la aplicación (Ver sección 4.1.7.1 MODOS DE OPERACIÓN DE USUARIOS).
- Instituciones: Se refiere a colegios e institutos que se inscriban al sistema y de los cuales se derivan los clubes de matemáticas.
- Clubes: Se refiere a los clubes de matemáticas inscritos a CLUBMAT, están asociados a una institución pero pueden participar niños de diferentes instituciones. Estos contienen información acerca de profesores, estudiantes, guías para docentes, ejercicios, noticias. Para estos clubes se manejan unas reglas de creación para realizar su formalización como club de matemáticas.
- Olimpiadas: Son eventos que programan los clubes de matemáticas y donde pueden participar más de un club en ellos, en estas olimpiadas se especifica el evento donde se realizara y promover la participación de otros clubes.
- Pruebas: Los estudiantes podrán realizar ejercicios que sus profesores dejen en la aplicación, se utilizara un banco de preguntas donde se obtendrán los ejercicios propios de la aplicación.

- Guías: Son guías de los temas para los docentes, con estas ellos se podrán apoyar para sus distintas actividades o ejercicios que planeen con sus estudiantes.
- Noticias: CLUBMAT será una comunidad de clubes de matemáticas, por esto cada club podrá estar informado de lo que pasa en la comunidad y así mismo publicar noticias para los demás interesados.
- Preguntas: Será una banco de preguntas por tema, grado, nivel de dificultad donde serán utilizadas para las pruebas y para olimpiadas. Estas preguntas van asociadas a unas opciones de respuesta asociadas.

La Ilustración 14 refleja el diagrama de modelo de dominio de la aplicación.

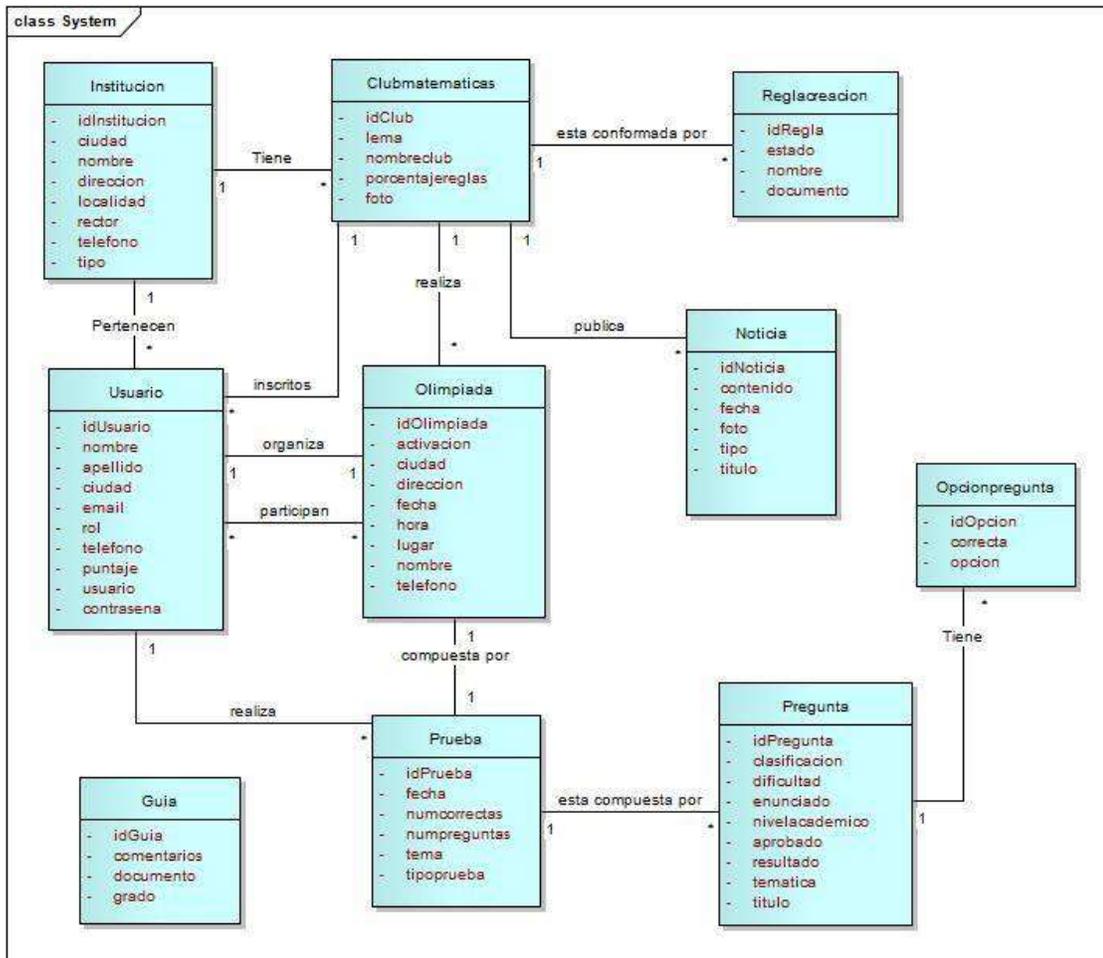


Ilustración 14: Diagrama Modelo de Dominio

B. Roles y Usuarios

La aplicación tendrá distintos tipos de usuario con alcances diferentes como se observa en la ilustración 14.

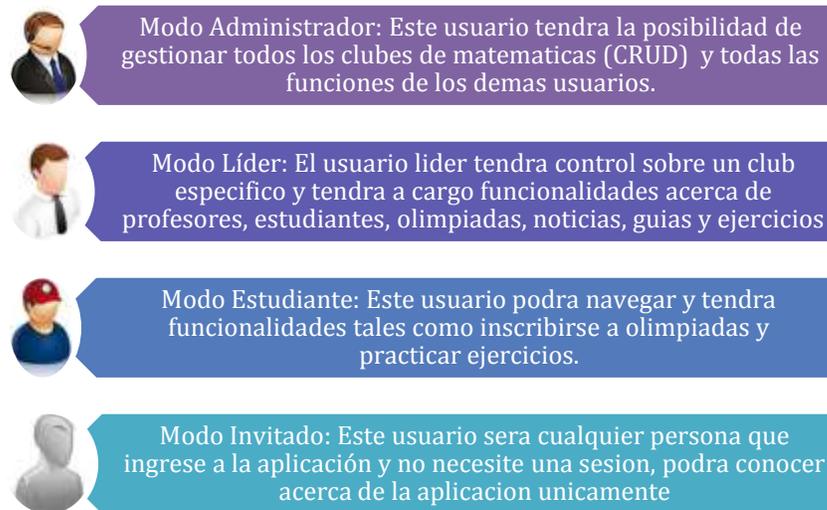


Ilustración 14: Roles y Usuarios

5.2.2 Requerimientos del Sistema

En esta sección se podrán ver los requerimientos del sistema (Funcionales y no funcionales), restricciones que tiene el sistema y el listado de los requerimientos dividido por módulos.

A. REQUERIMIENTOS FUNCIONALES

La distribución de los requerimientos se realizó por módulos definidos por los Stakeholders y son los siguientes:

- **Módulo de Seguridad:**

Este módulo comprende requerimientos acerca de los usuarios, roles, accesos y permisos que tienen los usuarios para administrar y utilizar la aplicación.

- **Módulo de Administración:**

Este módulo abarca los requerimientos asociados a los CRUD de clubes de matemáticas, olimpiadas, reglas, noticias.

- **Módulo de Negocio:**

Este módulo comprende los requerimientos asociados a las reglas de negocio, accesos y métodos adicionales como cálculos, sesiones, alcance de los usuarios, etc.

- **Módulo de Reportes:**

Este módulo abarca los requerimientos asociados con información mostrada al usuario como, número de estudiantes, resultados de pruebas, etc.

La Tabla 2 muestra el listado de requerimientos funcionales agrupados por modulo y con los respectivos usuarios que tienen acceso al requerimiento.

Modulo	Id	Especificación del requerimiento	Roles asociados
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.	Administrador, Lider
	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.	Administrador, Lider
	R03	El sistema debe garantizar las propiedades ACID en todas las transacciones de datos.	Administrador, Lider, Estudiante, Invitado
	R04	El sistema mostrará al usuario en su menú solo las opciones que el mismo tiene derecho según su rol	Administrador, Lider, Estudiante, Invitado
	R05	El sistema debe permitir a un usuario registrado iniciar sesión en el sistema	Administrador, Lider, Estudiante
	R06	El sistema debe permitir que un usuario cierre su sesión.	Administrador, Lider, Estudiante
RACION ADMINISTRATIVO	R08	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	Administrador

	seleccionado.	
R09	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.	Administrador, Lider
R10	El sistema debe proveer un mecanismo de gestión de guias para docentes (CRUD de guias) , que permita buscar por tema, grado y nivel de dificultad.	Administrador, Lider
R11	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.	Administrador, Lider
R12	El sistema debe proveer un mecanismo de gestión de estudiantes (CRUD de usuarios) , que permita listar los estudiantes en existentes, ingresar uno nuevo, editar o eliminar un estudiante.	Administrador, Lider
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.	Administrador, Lider
R14	El sistema debe proveer un mecanismo de gestión de pruebas (CRUD de pruebas) , que permita listar las instituciones existentes, ingresar una nueva, editar o eliminar una prueba seleccionada.	Administrador, Lider, Estudiante
R15	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.	Administrador, Lider
R16	El sistema debe permitir a un "Administrador" ver una lista de pasos para crear su club de matemáticas	Administrador
R17	Permitir seleccionar como ya realizado un paso para la creación de un club de matemáticas	Administrador

	R18	El sistema debe mostrar los estudiantes inscritos a una olimpiada	Administrador, Lider Estudiante,
NEGOCIO	R19	El sistema debe permitir publicar noticias en el componente de noticias	Administrador, Lider
	R20	El sistema debe permitir a un estudiante inscribirse y desinscribirse a una olimpiada existente	Estudiante
	R21	El sistema debe permitir al administrador modificar la evolución del club de matematicas	Administrador
	R22	El sistema debe permitir cambiar la contraseña del usuario que lo solicite	Administrador, Lider, Estudiante
	R23	El sistema debe permitir buscar las guias para los docentes de acuerdo al grado	Administrador, Lider
	R24	El sistema debe permitir cerrar la sesión de un usuario en cualquier momento de la aplicación	Administrador, Lider, Estudiante
	R25	El sistema debe permitir ver la información de CLUBMAT a un usuario invitado sin posibilidad de modificar ni cambiar información	Administrador, Lider, Estudiante, Invitado
	R26	El sistema debe permitir ver la información del usuario que esta en esa sesión	Administrador, Lider Estudiante,
REPORTES	R27	El sistema debe desplegar las noticias de los clubes de matemáticas en la pantalla inicial	Administrador, Lider, Estudiante, Invitado
	R28	El sistema debe mostrar la información de ¿Quiénes somos? Para cualquier usuario	Administrador, Lider, Estudiante, Invitado
	R29	El sistema debe mostrar la información de los clubes inscritos en CLUBMAT para cualquier usuario	Administrador, Lider, Estudiante, Invitado
	R30	El sistema debe mostrar la información de las instituciones inscritas a CLUBMAT para cualquier usuario	Administrador, Lider, Estudiante, Invitado
	R31	El sistema debe mostrar la información de contacto de CLUBMAT para cualquier usuario	Administrador, Lider, Estudiante, Invitado
	R32	El sistema debe calcular la evolución del club de matemáticas	Administrador

tabla 2: Requerimientos Funcionales CLUBMAT

B. RESTRICCIONES

- **Restricciones Generales**

La aplicación se desarrollará en idioma español.

El proyecto debe cumplir las siguientes especificaciones dadas por el cliente:

- ✓ Implementación Cliente – Servidor
- ✓ Manejo de Persistencia
- ✓ Manejo de GUI fuerte
- ✓ Deberá desarrollarse en el ciclo lectivo 2013 – 3, entre el 23 de julio y el 30 de noviembre.

- **Restricciones de Software**

Al desarrollar el sistema en lenguaje Java es de vital importancia tener instalado la máquina virtual (JVM) sin importar la plataforma en la cual se ejecute el software. Se maneja una licencia de software libre.

- **Restricciones de Hardware**

Se recomienda ejecutar el sistema en un computador con al menos 1 GB de memoria RAM, procesador de 1.6 GHz, tarjeta de video de 100 MB en una red con velocidad de al menos 1 Gb de transferencia. [3].

C. REQUERIMIENTOS NO FUNCIONALES

- **Requerimientos de Persistencia**

Estos requerimientos están dados por la base de datos donde va a persistir la información de CLUBMAT.

- **Requerimientos de Persistencia**

Estos requerimientos se refieren a las autorizaciones y autenticación de los usuarios

- **Requerimientos de Comunicación**

Estos requerimientos hacen referencia al protocolo de comunicación que va a tener la aplicación CLUBMAT

- **Requerimientos de Disponibilidad**

Estos requerimientos se refieren al tiempo de disponibilidad del sistema para los usuarios

- **Requerimientos de Concurrencia**

Estos requerimientos están dados al uso constante y paralelo de los usuarios al sistema.

- **Requerimientos de Interfaz de Usuario**

Estos requerimientos están dados por la capa de presentación o interfaz de usuario, y los componentes que va a tener la aplicación y la herramienta donde va a ser desarrollada.

- **Requerimientos de Documentación**

Requerimiento asociado a los manuales de instalación y manuales para el usuario relacionados con el sistema.

La tabla 3 muestra el listado de requerimientos No funcionales del sistema CLUBMAT.

Modulo	Id	Especificación del requerimiento
NO FUNCIONALES	R33	PERSISTENCIA: El sistema debe almacenar la información de instituciones, clubes, usuarios, reglas, pruebas, preguntas, olimpiadas, guías y noticias y debe persistir en una base de datos
	R34	SEGURIDAD: El sistema debe autenticar a los usuarios con usuario y contraseña
	R35	SEGURIDAD: El sistema debe autorizar el ingreso a las funciones del usuario dependiendo de su rol
	R36	DISPONIBILIDAD: La aplicación debe estar en un servidor activo 24 horas, el uso del sistema va a ser via web
	R37	CONCURRENCIA: El sistema debe manejar usuarios en paralelo evitando errores de concurrencia
	R38	COMUNICACIÓN: El sistema debe comunicarse via web basado en servicios web REST
	R39	INTERFAZ USUARIO: El sistema debe estar realizado en JavaFX2 como interfaz gráfica de usuario.
	R40	DOCUMENTACION: Deben generarse manuales para usuario y mantenimiento del software

TABLA 3: Requerimientos No Funcionales CLUBMAT

D. PRIORIZACION DE REQUERIMIENTOS

Cada requerimiento de cada módulo (Ver sección 5) tendrá su nivel de prioridad respecto a los otros, se hará uso del siguiente esquema de prioridad de requerimientos a partir de la importancia que tengan en el desarrollo final del producto (Tabla 4).

IDENTIFICADOR DE LA PRIORIDAD	DESCRIPCIÓN
Alta	Requerimiento que debe ser implementado en su totalidad.
Media	Requerimiento importante para ser implementado que sirve como soporte para los requerimientos de alta prioridad.
Baja	Requerimiento no vital para la entrega del producto.

TABLA 4: Priorización de Requerimientos

Se optó por utilizar el método de Wiegers[11], Que consiste en calcular una prioridad para cada requerimiento en base:

- ✓ Beneficio para el cliente.
- ✓ Costo de implementación.
- ✓ Riesgo técnico.

Asignando a cada atributo un valor entre 1 y 9, antes de calcular prioridades se determina que todos los requerimientos tengan el mismo nivel de abstracción.

1 – 3: Baja , 4 – 6: Media , 7 – 9: Alta

La prioridad de los requerimientos estan detallados en la lista de requerimientos anexa (Ver Anexo: Lista de Requerimientos CLUBMAT) con el detalle de cada requerimiento.

5.2.3 Casos de Uso CLUBMAT

Esta sección define los casos de uso por medio de los cuales se describen las actividades que deben llevarse a cabo para realizar un determinado proceso dentro del sistema, así como los actores que participan en el desarrollo de estos casos de uso. (Ver ilustración 15)

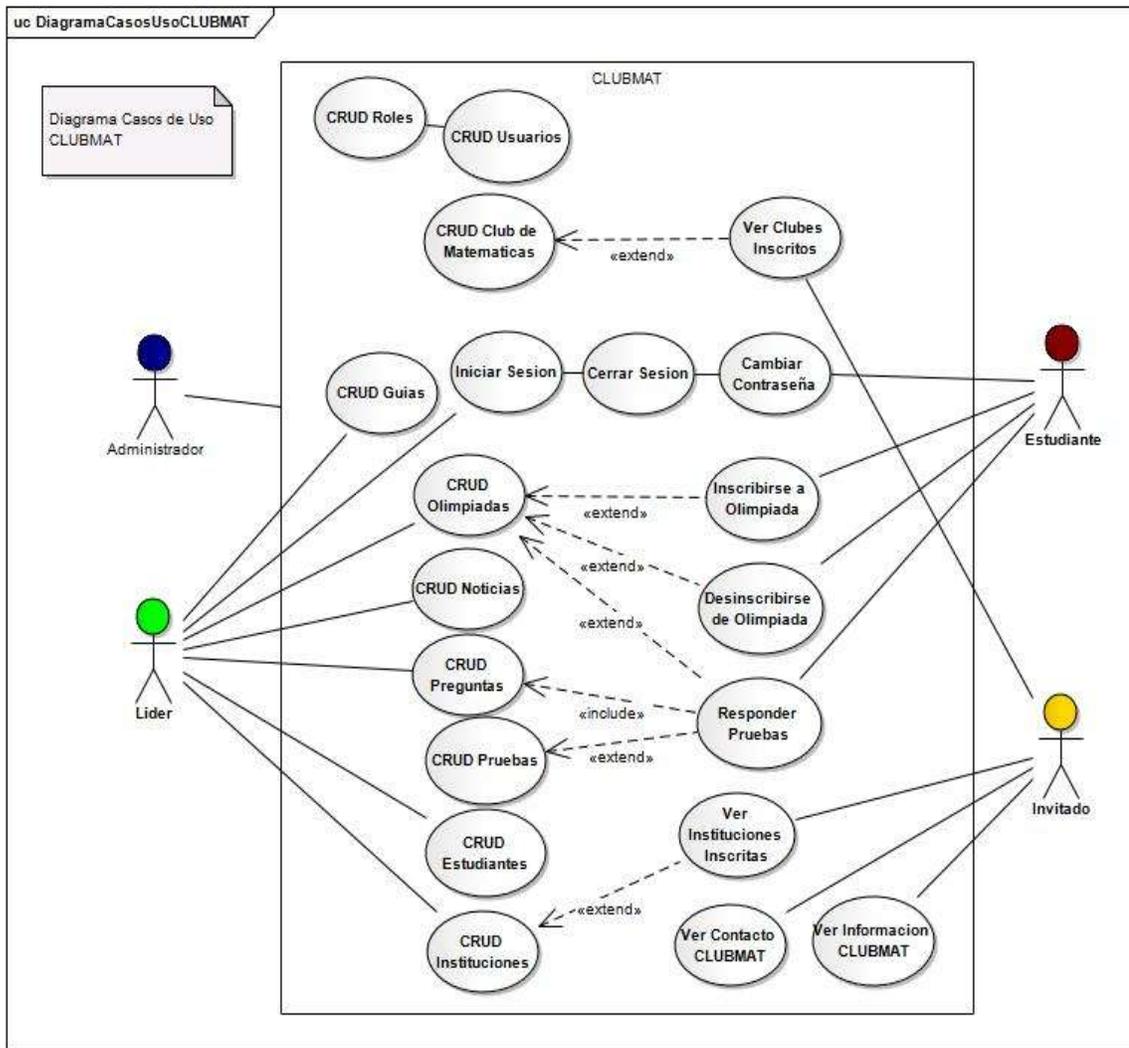


Ilustración 15: Diagrama de Casos de Uso CLUBMAT

En este diagrama de casos de uso es importante aclarar que el Rol Administrador tiene acceso a todas las funcionalidades del sistema, el Líder cumple con ciertas funcionalidades como señala el diagrama al igual que el Estudiante y el Invitado.

La descripción de cada caso de uso se encuentra a continuación (Ver tabla 5).

Modulo	Caso de uso	Nombre del Caso de Uso	Descripción del Caso de Uso
--------	-------------	------------------------	-----------------------------

SEGURIDAD	CU01	CRUD de roles	Esta funcionalidad permite crear, ver, actualizar y eliminar roles.
	CU02	CRUD de usuarios	Esta funcionalidad permite crear, ver, actualizar y eliminar usuarios.
	CU03	Iniciar Sesión	Esta funcionalidad permite ingresar a la aplicación web por medio de un nombre de usuario y contraseña.
	CU04	Cerrar la sesión del sistema	Esta funcionalidad permite que un usuario autenticado cierre sesión de la aplicación web.
ADMINISTRACION	CU05	CRUD de clubes de matemáticas	Esta funcionalidad permite crear, ver, actualizar y eliminar clubes de matemáticas
	CU06	CRUD de noticias de los clubes de matemáticas	Esta funcionalidad permite crear, ver, actualizar y eliminar noticias.
	CU07	CRUD de guías para docentes	Esta funcionalidad permite crear, ver, actualizar y eliminar guías.
	CU08	CRUD de olimpiadas	Esta funcionalidad permite crear, ver, actualizar y eliminar olimpiadas
	CU09	CRUD de estudiantes	Esta funcionalidad permite crear, ver, actualizar y eliminar estudiantes
	CU10	CRUD de instituciones	Esta funcionalidad permite crear, ver, actualizar y eliminar instituciones.
	CU11	CRUD de preguntas	Esta funcionalidad permite crear, ver, actualizar y eliminar preguntas.
	CU12	CRUD de instituciones	Esta funcionalidad permite crear, ver, actualizar y eliminar instituciones.
	CU13	CRUD de pruebas	Esta funcionalidad permite crear, ver, actualizar y eliminar pruebas.

NEGOCIO	CU14	Inscribirse a olimpiada	Permite a cada estudiante inscribirse a una olimpiada existente
	CU15	Desinscribirse de olimpiada	Permite a cada estudiante desinscribirse de una olimpiada existente
	CU21	Responder Pruebas	
	CU16	Cambiar Contraseña	Permite a cada usuario en su sesión cambiar su contraseña
REPORTES	CU17	Ver Información de CLUBMAT	El sistema muestra la información de ¿Quiénes Somos? Para todos los usuarios para dar a conocer CLUBMAT
	CU18	Ver Clubes Inscritos	Permite ver los clubes inscritos a toda la comunidad y usuarios invitados
	CU19	Ver Instituciones Inscritas	Permite ver las instituciones inscritas a toda la comunidad y usuarios invitados
	CU20	Ver contacto CLUBMAT	Permite ver la información del contacto de CLUBMAT a toda la comunidad y usuarios invitados para comunicarse con los administradores de la aplicación y poder pertenecer a la comunidad

TABLA 5: Casos de Uso CLUBMAT

5.2.4 Modelo de Entidades de Negocio

Esta sección muestra la estructura del sistema de forma general y que representa la arquitectura lógica del mismo.

La ilustración 16 define el modelo relacional del sistema para ir del nivel más profundo de la arquitectura del sistema hasta la parte global y física del sistema.

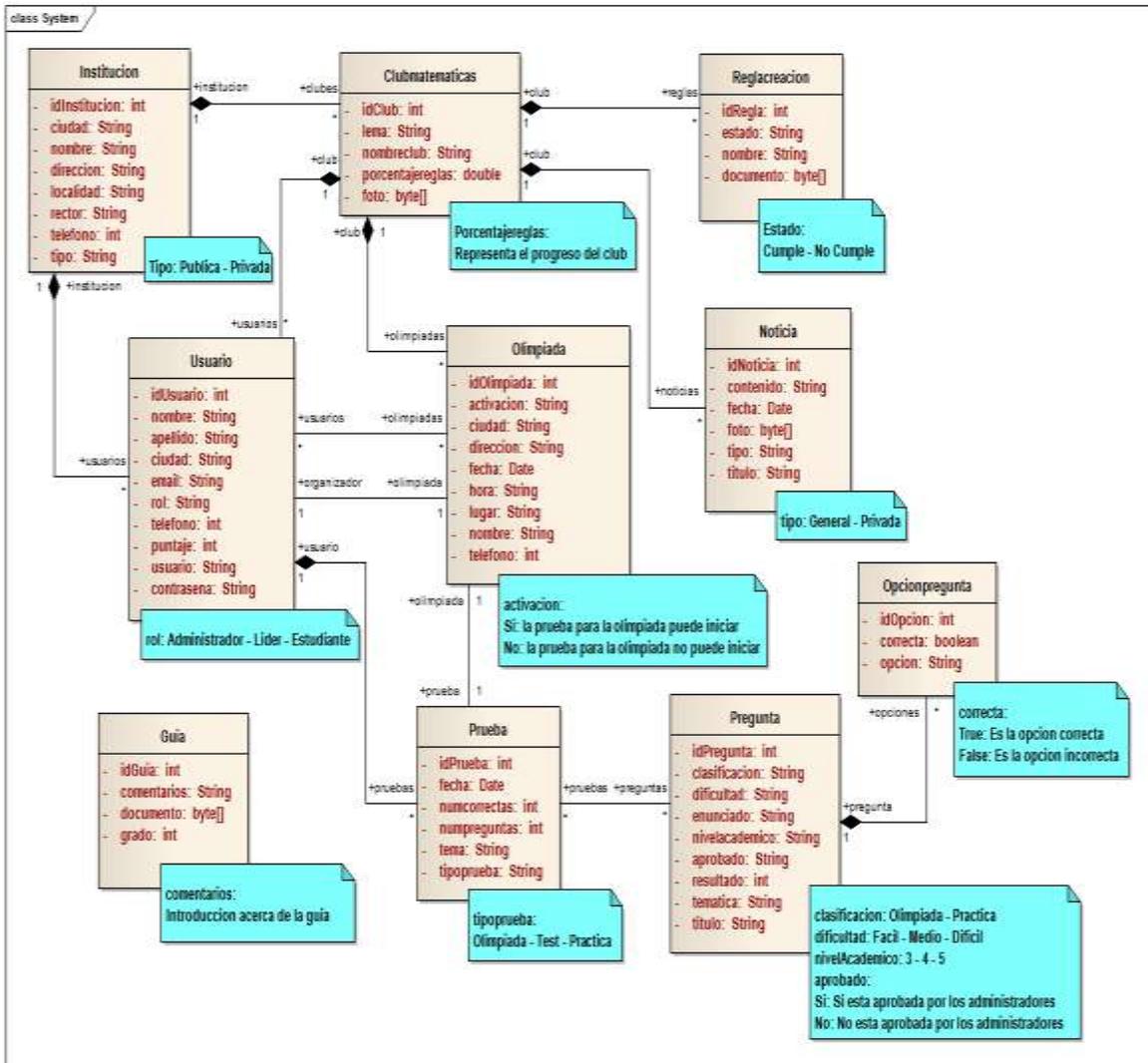


Ilustración 16: Diagrama de Entidades de Negocio CLUBMAT

El diagrama anterior se puede ver con detalle en los archivos adjuntos a este documento (Ver Anexos: DiagramaModeloRelacional).

Para entender las entidades y su significado revisar la sección (Modelo de dominio).

5.3 Arquitectura del sistema

La arquitectura JavaEE6 acoplada con el framework JavaFX2 utilizando servicios web basados en REST general para cualquier tipo de proyecto tendría la forma que se ve en la ilustración 17 [2].

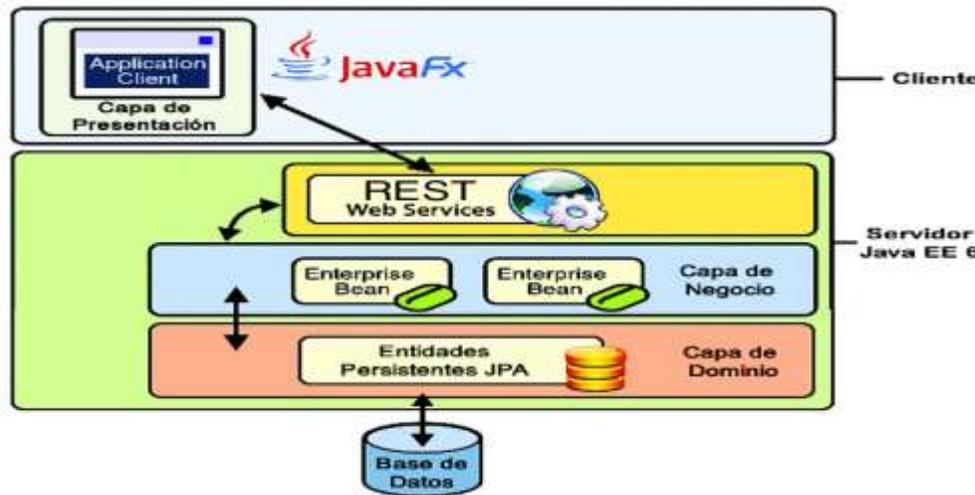


Ilustración 17: Arquitectura JAVAFX2- JAVAAEE6- REST

Esta arquitectura se aplicó al contexto de este proyecto y así mismo se determinaron los componentes que la conforman y su proceso de interacción.

5.3.1 Objetivos Arquitecturales y Restricciones

OBJETIVOS ARQUITECTURALES Y RESTRICCIONES	TECNICAS ARQUITECTURALES
<p>PERSISTENCIA: El sistema debe almacenar la información de instituciones, clubes, usuarios, reglas, pruebas, preguntas, olimpiadas, guías y noticias y debe persistir en una base de datos</p>	<p>Se conectó el sistema a la base de datos POSTGRESQL , haciendo uso del JPA (Java Persistence API) que nos brinda el uso de la arquitectura JavaEE6 y el cual se encarga del manejo de datos relacionales desde Java a la base de datos.</p>

SEGURIDAD: El sistema debe autenticar a los usuarios con usuario y contraseña y autorizar el ingreso a las funciones del usuario dependiendo de su rol	La validación se realizó desde el mismo sistema validando y autenticando la información con la base de datos haciendo uso de la arquitectura JavaEE6.
DISPONIBILIDAD: La aplicación debe estar en un servidor activo 24 horas, el uso del sistema va a ser via web	Se instalará la aplicación empresarial en un servidor permanente de la universidad Javeriana para el uso permanente via web de los usuarios
CONCURRENCIA: El sistema debe manejar usuarios en paralelo evitando errores de concurrencia	La arquitectura JavaEE6 nos permite librarnos del manejo de concurrencia ya que los EJB se encargan de manejar la concurrencia y las transacciones que se tengan de la aplicación.
DESEMPEÑO: El tiempo de respuesta de la aplicación no debe tardarse mas de 10 segundos	Se utilizó para la comunicación de los datos servicios web basados en REST mejorando el desempeño del sistema y el tiempo de respuesta. (Ver Marco Conceptual)

TABLA 6: Objetivos arquitecturales

A continuación se muestra cómo el sistema será desplegado a través de la identificación de ubicaciones y los nodos lógicos en cada una de las ubicaciones independientes de la tecnología.

En la ilustración 18 se evidencian los componentes de la arquitectura JavaEE6 integrada con JavaFX2 haciendo uso de servicios web REST y solventando los objetivos arquitecturales y las restricciones dadas por los requerimientos no funcionales presentados previamente.

5.3.2 Diagrama Vista Fisica CLUBMAT

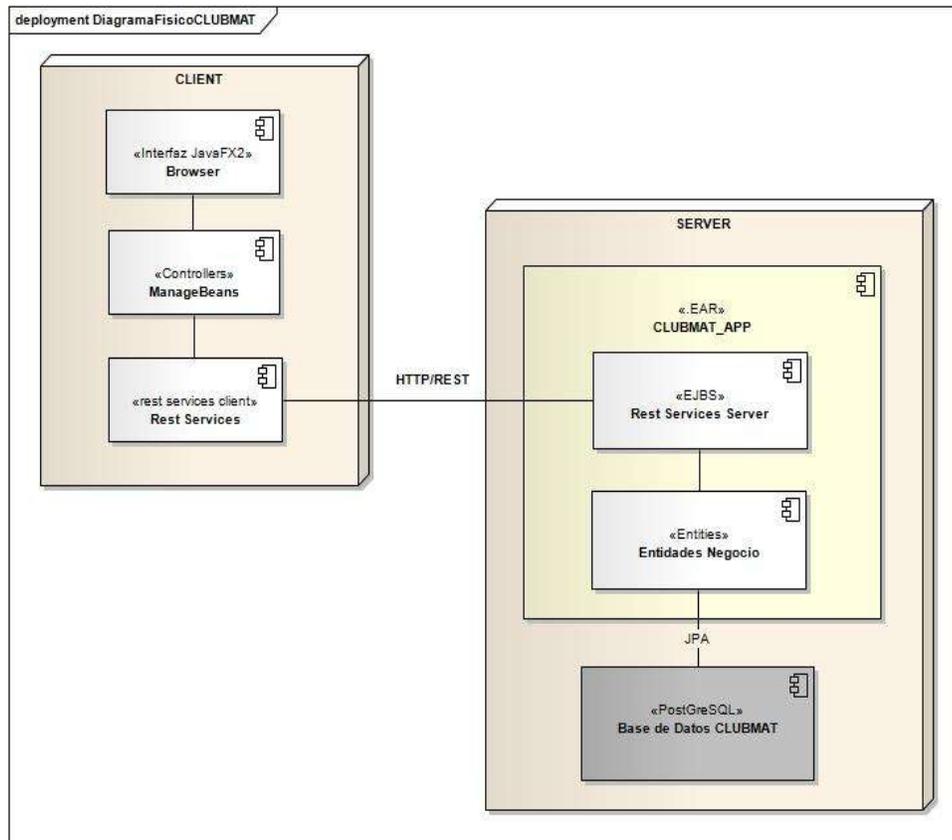


Ilustración 18: Arquitectura Global CLUBMAT

5.3.3 Componentes Vista Física

- **Componente BROWSER**

Este componente es la interfaz gráfica de usuario, esta interfaz va a ser web por lo que un browser es el necesario para utilizarla, es importante especificar que la interfaz de usuario está en JavaFX2 [5].

- **Componente MANAGEBEANS**

Este componente contiene los controladores que capturan la información de las acciones y peticiones del cliente a través de la interfaz gráfica y hacen llamado a los servicios web con protocolo REST [4] para realizar las peticiones del cliente.

- **Componente REST SERVICES CLIENT**

Este componente contiene las referencias de los servicios web REST y los servicios que estos prestan por parte del cliente. Estos contienen la firma de los servicios que remotamente los servicios web proveen. Los controladores hacen uso de estas referencias para obtener servicios prestados por parte del servidor.

- **Componente REST SERVICES SERVER**

Este componente contiene la lógica de negocio de la aplicación y los EJB [3]. Así mismo estos EJB son los servicios web REST que proveen servicios al cliente y se encargan de tareas como seguridad, concurrencia, persistencia, etc.

- **Componente ENTIDADES NEGOCIO**

Este componente es donde se alojan las entidades de negocio del sistema CLUBMAT, esta la estructura de la base de datos. A partir de estas se crea la base de datos y se realiza su configuración.

- **Componente BASE DE DATOS CLUBMAT**

Este componente es donde esta alojada la base de datos al lado del servidor, este componente guardara toda la información acerca de la aplicación CLUBMAT y será la única, es importante aclarar que se hará con la herramienta PostgreSQL [6].

5.3.4 Control de Flujo de Componentes

En la ilustración 19 se puede ver la interacción de los componentes y su comportamiento secuencial.

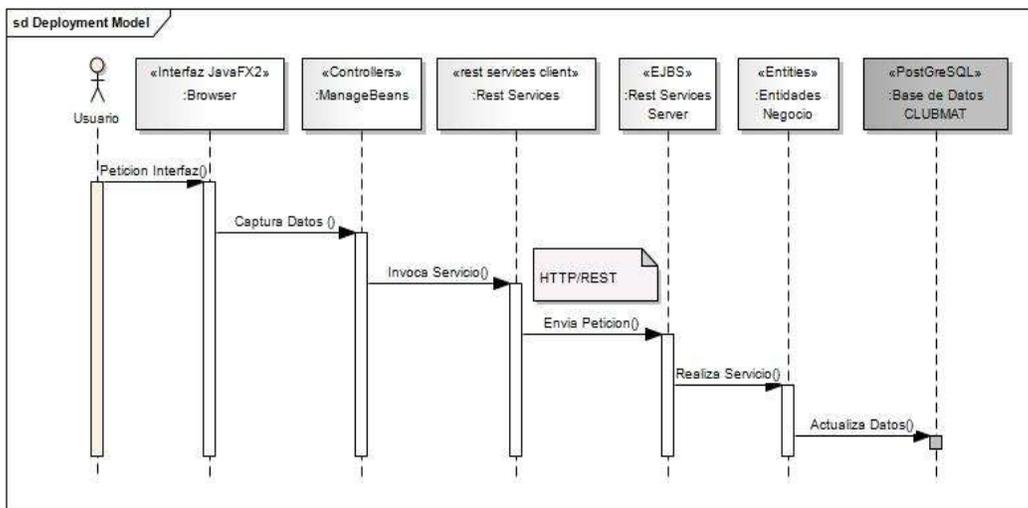


Ilustración 19: Control de Flujo Global CLUBMAT

6. Fases de Preparación, Diseño, Desarrollo y Pruebas

De acuerdo a la priorización de los requerimientos y dada la arquitectura del software se determinaron los pasos a seguir de acuerdo a la metodología propuesta para pasar a la implementación del software.

Las siguientes fases se realizarán por cada iteración (Sprint) [8] de acuerdo a la priorización de los requerimientos pero en este documento se presentará un caso de uso prioritario para explicar cada fase en detalle. La documentación de cada iteración se encuentra en los anexos a este documento (Ver Anexo: Actas Iteraciones).

6.1 Estrategia general para desarrollar un caso de uso CRUD: Ejemplo CRUD de Clubes de Matemáticas

En esta sección se presentará el desarrollo de un CRUD específico integrando JavaEE6 y JavaFX2 con servicios web REST para ilustrar la interacción detallada de los componentes que intervienen en su desarrollo.

6.1.1 Preparación

En esta iteración se desarrollaron los requerimientos asociados a los CLUBES DE MATEMATICOS. Los requerimientos asociados a la creación, edición, eliminación y muestra de información respecto a esta entidad del sistema. Para ver el detalle de esta entidad en el sistema ver (Sección 6.5.1 Modelo Relacional).

6.1.2 Diseño

A continuación se pueden ver los componentes que intervienen en el desarrollo de estos casos de uso CRUD y sus respectivas relaciones se ven a continuación en el diagrama EBC (Entity, Boundary, Control). [28] Así mismo, el comportamiento de estos componentes de acuerdo a la secuencia se ven en los respectivos diagramas de secuencia por cada caso de uso. (Ver ilustración 20)

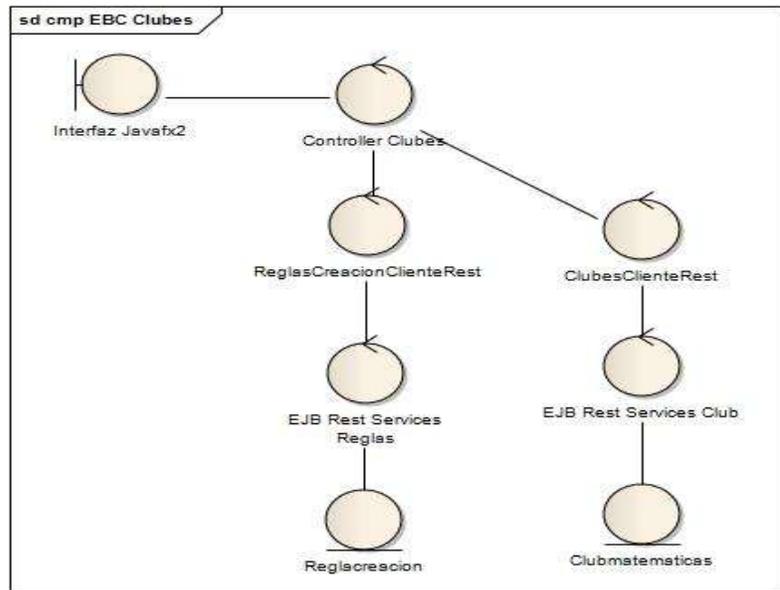


Ilustración 20: Diagrama EBC Club de Matematicas

Estos componentes muestran el global de la comunicación que existe entre componentes respecto a este CRUD específico.

Para ver en detalle los diagramas EBC de toda la interacción del sistema ver los anexos (Diagramas CLUBMAT).

La ilustración 21 muestra en detalle los componentes que intervienen en este CRUD de clubes de matemáticas.

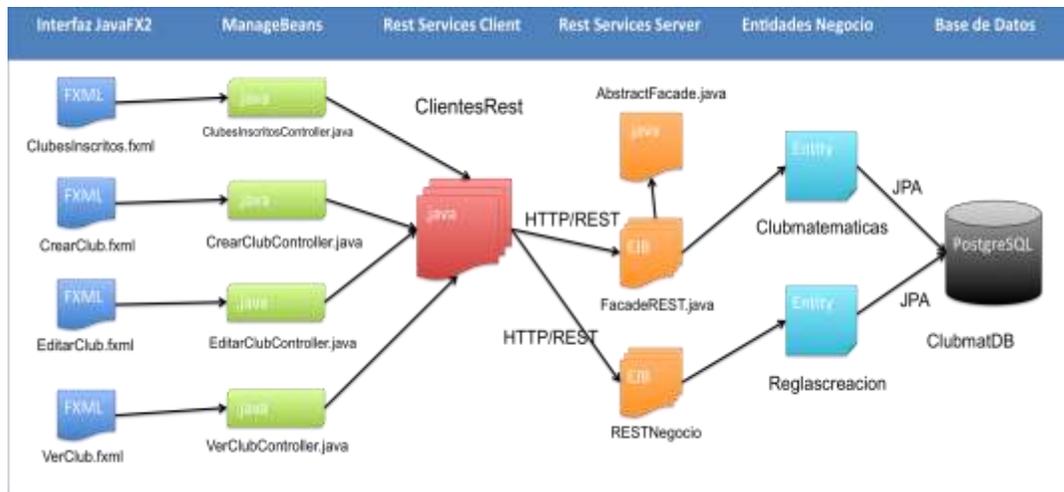


Ilustración 21: Diagrama Proceso CRUD clubes de matematicas

A continuación se verá en detalle el diseño de cada funcionalidad de este CRUD y su desarrollo.

6.1.3 CREAR CLUB

En esta sección se verá en detalle el paso a paso para crear un club de matemáticas de acuerdo a la arquitectura planteada anteriormente. (Ver ilustración 22)

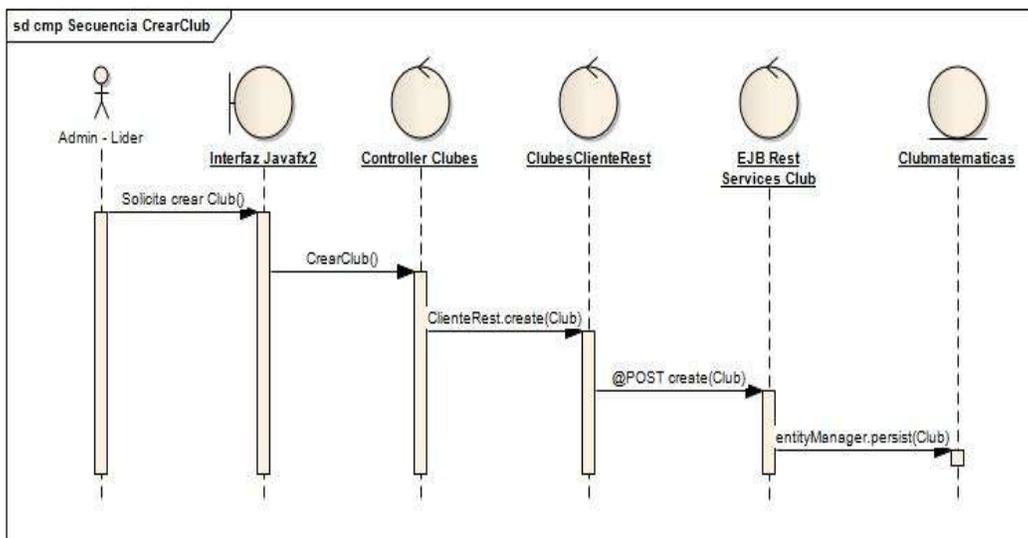


Ilustración 22: Diagrama de Secuencia Crear Club

Este diagrama muestra el global del comportamiento de los componentes en secuencia para la creación de un club de matemáticas.

1. Interfaz JavaFX2 (CrearClub.fxml)

- Esta interfaz representa el componente *Interfaz Javafx2 (Boundary)*. El usuario llena los datos del club en los respectivos espacios y da acción al botón crear club. (Ver ilustración 23).



Ilustración 23: Crearclub.fxml

2. Controller Clubes (CrearClubController.java)

- El botón tiene asociado un método del controlador de esta pantalla *CrearClubController.java* (*Control*) la cual activa la petición.
- Crea una instancia del clienteRest para realizar la petición del servicio web de crear un club.
- Crea una instancia de un club de matemáticas con los datos ingresados
- Invoca al servicio de crear club y le envía la instancia creada. (Ver ilustración 24)

```
@FXML
private void crearClub(ActionEvent event){
    try{
        ClubClienteRest ccr = new ClubClienteRest(); //Instancia ClienteRest

        Clubmatematicas club = new Clubmatematicas(); //Instancia Club de Matematicas
        club.setNombreclub(textNombre.getText());
        club.setLema(textLema.getText());

        ccr.create_JSON(club); //Se invoca al servicio web
        ccr.close(); //se destruye la instancia ClienteRest
```

Ilustración 24: CrearclubController

3. ClubesClienteRest (ClubClienteRest.java)

- Esta clase se encarga de recibir la petición del cliente y generar una URI para enviar la petición al servidor.
- Crea el path de la petición y teniendo como base la referencia del servidor donde está ubicado el servicio web.
- Envía la información de la petición en una URI al servidor de acuerdo al método que se invoca. Es importante que el método para crear una instancia en rest es POST. (Ver ilustración 25).

```
public class ClubClienteRest {
    private WebResource webResource;
    private Client client;
    private static final String BASE_URI = "http://localhost:8080/Clubmat-war/webresources";

    public ClubClienteRest() {
        com.sun.jersey.api.client.config.ClientConfig config = new com.sun.jersey.api.client.config.DefaultClientConfig();
        client = Client.create(config);
        webResource = client.resource(BASE_URI).path("co.edu.javeriana.clubmat.entidades.clubmaticas");
    }

    public void create_JSON(Object requestEntity) throws UniformInterfaceException {
        webResource.type(javax.ws.rs.core.MediaType.APPLICATION_JSON).post(requestEntity);
    }
}
```

Ilustración 25: CrearclubClienteRest

4. EJB Rest Services Club (clubmaticasFacadeREST.java)

- Al lado del servidor, los EJB hacen la función de servicios web REST, estos tienen los métodos que llama el cliente.
- En este caso esta clase tiene el *entityManager* que se encarga de los métodos que se utilizan para interactuar con el contexto de persistencia, y en este caso, permite almacenar la instancia del club en la base de datos.
- Como decíamos anteriormente, en este caso de creación la anotación REST utilizada es *@POST*.
- La anotación *@Consumes* se refiere al formato que recibe de la entidad. (Ver ilustración 26)
- En este caso esta clase hereda de la clase *AbstractFacade* que contiene los métodos del *entityManager* para las demás clases. (Ver ilustración 27)

```

@Stateless
@Path("co.edu.javeriana.clubmat.entidades.clubmatematicas")
public class ClubmatematicasFacadeREST extends AbstractFacade<Clubmatematicas> {
    @PersistenceContext(unitName = "Clubmat-warPU")
    private EntityManager em;

    public ClubmatematicasFacadeREST() {
        super(Clubmatematicas.class);
    }

    @POST
    @Override
    @Consumes({"application/xml", "application/json"})
    public void create(Clubmatematicas entity) {
        super.create(entity);
    }
}

```

Ilustración 26: clubmatematicasFacadeRest.java

```

public abstract class AbstractFacade<T> {
    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public void create(T entity) {
        getEntityManager().persist(entity);
    }
}

```

Ilustración 27: AbstractFacade.java

5. Clubmatematicas (Clubmatematicas.java)

- Esta clase es la entidad del club de matemáticas que es un objeto de dominio de persistencia. Normalmente, una entidad representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla.
- En este caso particular es la del club de matemáticas. Se almacena una instancia de esta entidad (un registro) en la base de datos.
- La anotación *@Entity* se refiere a que esa clase java es una entidad.
- El nombre que tiene la anotación *@Table* es el que va a tener la tabla en la base de datos. (Ver ilustración 28).

- Esta clase esta compuesta por los atributos de la entidad y los queries para hacer peticiones mas especificas. (Ver Seccion: Ver Reglas)

```
@Entity
@Table(name = "clubmaticas")
```

Ilustración 28: Clubmaticas.java

Este proceso da por finalizado la creación del club de matemáticas y se ve paso a paso como se almacena desde el cliente hasta la base de datos.

6.1.4 EDITAR CLUB

En esta sección se vera en detalle el paso a paso para editar un club de matematicas de acuerdo a la arquitectura planteada anteriormente. (Ver ilustración 29)

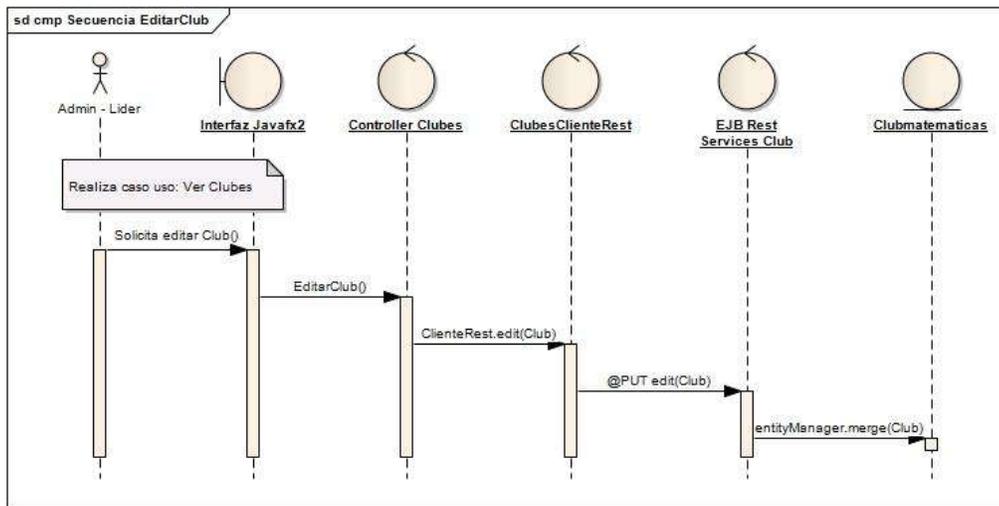


Ilustración 29: Diagrama de Secuencia Editar Club

Este diagrama muestra el global del comportamiento de los componentes en secuencia para la edición de un club de matemáticas.

1. Interfaz JavaFX2 (EditarClub.fxml)

- Esta interfaz representa el componente **Interfaz Javafx2 (Boundary)**. El usuario llena los datos del club en los respectivos espacios y da acción al botón editar club. (Ver ilustración 30)
- Previamente el usuario debio haber seleccionado un club existente en la base de datos para posteriormente editarlo (Ver Seccion: Ver Clubes).

The screenshot shows a web application interface for editing a club. At the top, there is a navigation bar with the 'CLUB MAT' logo, the text 'Gestor de Clubes de Matemáticas', the Pontificia Universidad JAVERIANA logo, and links for '¿Quiénes Somos?', 'Contáctanos', 'Administrador Julian', and 'Cerrar Sesión'. Below the navigation bar is a colorful bar with the word 'CLUBES'. The main content area is titled 'EDITAR CLUB' and contains a form with the following fields:

- * Nombre :** Clubmat3
- * Lema :** Amo las matematicas
- Logo :** Exammat (with a note: Si no seleccionas un archivo, el logo sera el actual)
- * Institución :** Colegio Rafael Uribe Uribe

At the bottom of the form are two buttons: 'Cancelar' and 'Editar Club'. The 'Editar Club' button is highlighted with a red box.

Ilustración 30: EditarClub.FXML

2. Controller Clubes (EditarClubController.java)

- El botón tiene asociado un método del controlador de esta pantalla **EditarClubController.java (Control)** la cual activa la petición.
- Se crea una instancia del ClienteRest que tiene los servicios.
- Solicita la instancia del club seleccionado en la base de datos.
- Se editan los campos de la instancia con los datos nuevos del cliente.

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

    try{
        ClubClienteRest clubCliente = new ClubClienteRest();
        ClientResponse response = clubCliente.find_JSON(ClientResponse.class, ClubesInscritosController.clubSeleccionado.getIdClub());
        GenericType<Clubmatematicas> genericType = new GenericType<Clubmatematicas>() {};
        Clubmatematicas miClub = response.getEntity(genericType);

        miClub.setNombreclub(textNombre.getText());
        miClub.setLema(textLema.getText());
```

```
        clubCliente.edit_JSON(miClub);
        clubCliente.close();
```

Ilustración 31: EditarClubController.java

- Invoca al servicio de editar club y le envía la instancia editada. (Ver ilustración 31).
- El formato en que se envía la información es JSON. Aunque el servidor en este caso recibe JSON y XML.

3. ClubesClienteRest (ClubClienteRest.java)

- Esta clase se encarga de recibir la petición del cliente y generar una URI para enviar la petición al servidor.
- Crea el path de la petición y teniendo como base la referencia del servidor donde está ubicado el servicio web (Ver ilustración 32).

```
public class ClubClienteRest {
    private WebResource webResource;
    private Client client;
    private static final String BASE_URI = "http://localhost:8080/Clubmat-war/webresources";

    public ClubClienteRest() {
        com.sun.jersey.api.client.config.ClientConfig config = new com.sun.jersey.api.client.config.DefaultClientConfig();
        client = Client.create(config);
        webResource = client.resource(BASE_URI).path("co.edu.javeriana.clubmat.entidades.clubmatematicas");
    }
}
```

```
public <T> T find_JSON(Class<T> responseType, String id) throws UniformInterfaceException {
    WebResource resource = webResource;
    resource = resource.path(java.text.MessageFormat.format("{0}", new Object[]{id}));
    return resource.accept(javax.ws.rs.core.MediaType.APPLICATION_JSON).get(responseType);
}

public void edit_JSON(Object requestEntity) throws UniformInterfaceException {
    webResource.type(javax.ws.rs.core.MediaType.APPLICATION_JSON).put(requestEntity);
}
```

Ilustración 32: ClubClienteRest.java

- Envía la información de la petición en una URI al servidor de acuerdo al método que se invoco.
- Es importante que el método para editar una instancia en REST es PUT.
- Para la búsqueda de la instancia que se quiere editar se utiliza el método GET que es de búsqueda.

4. EJB Rest Services Club (clubmatematicasFacadeREST.java)

- Al lado del servidor, los EJB hacen la función de servicios web REST, estos tienen los métodos que llama el cliente.
- En este caso esta clase tiene el *entityManager* que se encarga de los métodos que se utilizan para interactuar con el contexto de persistencia, y en este caso, permite almacenar la instancia del club en la base de datos.
- Como decíamos anteriormente, en este caso de edición la anotación REST utilizada es *@PUT*.
- Para la búsqueda de la instancia que se quiere editar la anotación del método es *@GET*
- La anotación *@Consumes* se refiere al formato que recibe de la entidad en este caso JSON (Ver ilustración 33).
- En este caso esta clase hereda de la clase *AbstractFacade* que contiene los métodos del *entityManager* para las demás clases (Ver ilustración 34).

```
@Stateless
@Path("co.edu.javeriana.clubmat.entidades.clubmaticas")
public class ClubmaticasFacadeREST extends AbstractFacade<Clubmaticas> {
    @PersistenceContext(unitName = "Clubmat-warPU")
    private EntityManager em;

    public ClubmaticasFacadeREST() {
        super(Clubmaticas.class);
    }

    @PUT
    @Override
    @Consumes({"application/xml", "application/json"})
    public void edit(Clubmaticas entity) {
        super.edit(entity);
    }

    @GET
    @Path("{id}")
    @Produces({"application/xml", "application/json"})
    public Clubmaticas find(@PathParam("id") Integer id) {
        return super.find(id);
    }
}
```

Ilustración 33: ClubmaticasFacadeRest

- La anotación **@Path** se refiere al parámetro que ingresa, en este caso el id del club para la búsqueda del registro. Como se decía anteriormente lo que se envía es una URI con la petición al servidor, este path adiciona el parámetro a la URI para enviar la petición con el parámetro de búsqueda.

```
public abstract class AbstractFacade<T> {
    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public T find(Object id) {
        return getEntityManager().find(entityClass, id);
    }

    public void edit(T entity) {
        getEntityManager().merge(entity);
    }
}
```

Ilustración 34: AbstractFacade.java

5. Clubmatematicas (Clubmatematicas.java)

El proceso con la entidad es el mismo para cualquier caso de uso, este funcionamiento fue explicado en el desarrollo de la creación de un club de matemáticas (Ver sección 6.1.3 Crear Club).

6.1.5 ELIMINAR CLUB

En esta sección se verá en detalle el paso a paso para eliminar un club de matemáticas de acuerdo a la arquitectura planteada anteriormente. (Ver ilustración 35)

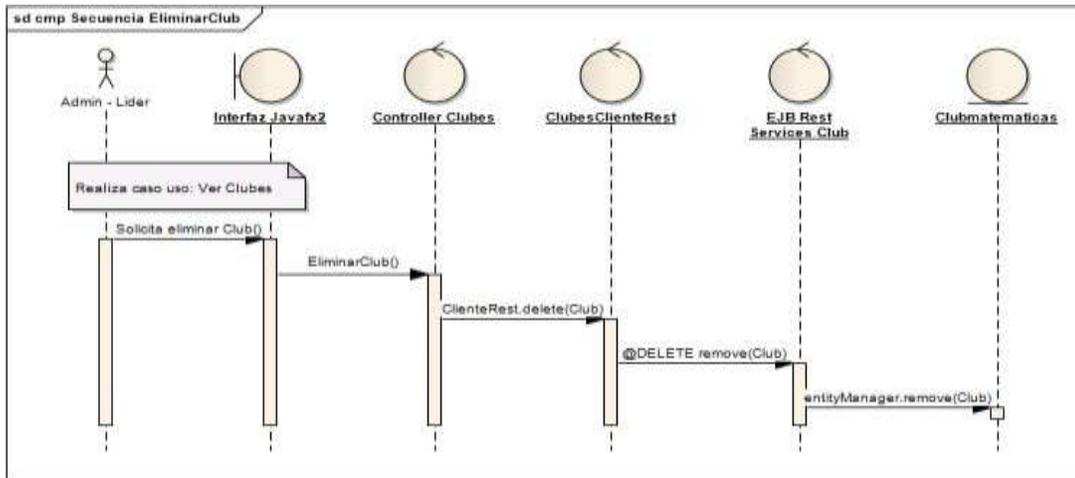


Ilustración 35: Diagrama Secuencia Eliminar Club

Este diagrama muestra el global del comportamiento de los componentes en secuencia para la eliminación de un club de matemáticas.

DESARROLLO DEL CASO DE USO ELIMINAR CLUB

1. Interfaz JavaFX2 (ClubesInscritos.fxml)

- Esta interfaz representa el componente **Interfaz Javafx2 (Boundary)**. El usuario da acción al botón eliminar club (Ver ilustración 36).
- Previamente el usuario debio haber seleccionado un club existente en la base de datos para posteriormente eliminarlo (Ver Seccion: 6.1.4 Ver Clubes).



Ilustración 36: ClubesInscritos.FXML

2. Controller Clubes (ClubesInscritosController.java)

- El botón tiene asociado un método del controlador de esta pantalla *ClubesInscritosController.java (Control)* la cual activa la petición.
- Se crea una instancia del ClienteRest que tiene los servicios.
- Solicita la instancia del club seleccionado en la base de datos.
- Se elimina la instancia (Ver ilustración 37).
- Invoca al servicio de eliminar club y le envía la instancia que se quiere eliminar.

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

    try{
        List< Clubmaticas > filaSeleccionada = this.tablaClubes.getSelectionModel().getSelectedItem();
        clubSeleccionado = filaSeleccionada.get(0);
        DialogResponse response = Dialogs.showConfirmDialog(ClubmatCliente.stage, "¿Estas seguro de eliminar este
        System.out.println("RESPONSE : "+response);

        if(response.toString().equals("YES")){
            ClubClienteRest clubCliente = new ClubClienteRest();
            clubCliente.remove(clubSeleccionado.getIdclub().toString());
            Dialogs.showInformationDialog(ClubmatCliente.stage, "El Club se ha eliminado con exito!", "CONFIRMAC
```

Ilustración 37: ClubesInscritosController.java

3. ClubesClienteRest (ClubClienteRest.java)

- Esta clase se encarga de recibir la petición del cliente y generar una URI para enviar la petición al servidor.
- Crea el path de la petición y teniendo como base la referencia del servidor donde esta ubicado el servicio web (Ver ilustración 38).

```

public class ClubClienteRest {
    private WebResource webResource;
    private Client client;
    private static final String BASE_URI = "http://localhost:8080/Clubmat-war/webresources";

    public ClubClienteRest() {
        com.sun.jersey.api.client.config.ClientConfig config = new com.sun.jersey.api.client.config.DefaultClientConfig();
        client = Client.create(config);
        webResource = client.resource(BASE_URI).path("co.edu.javeriana.clubmat.entidades.clubmatematicas");
    }

    public void remove(String id) throws UniformInterfaceException {
        webResource.path(java.text.MessageFormat.format("{0}", new Object[]{id})).delete();
    }
}

```

Ilustración 38: ClubClienteRest.java

- Envía la información de la petición en una URI al servidor de acuerdo al método que se invoco.
- Es importante que el método para eliminar una instancia en REST es DELETE.
- Para la búsqueda de la instancia que se quiere editar se utiliza el método GET que es de búsqueda.

4. EJB Rest Services Club (clubmatematicasFacadeREST.java)

- Al lado del servidor, los EJB hacen la función de servicios web REST, estos tienen los métodos que llama el cliente.
- En este caso esta clase tiene el *entityManager* que se encarga de los métodos que se utilizan para interactuar con el contexto de persistencia, y en este caso, permite almacenar la instancia del club en la base de datos.
- Como decíamos anteriormente, en este caso de eliminación la anotación REST utilizada es *@DELETE*.
- Para la búsqueda de la instancia que se quiere editar la anotación del método es *@GET* (Ver ilustración 39)
- En este caso esta clase hereda de la clase *AbstractFacade* que contiene los métodos del *entityManager* para las demás clases. (Ver ilustración 40)

```

@Stateless
@Path("co.edu.javeriana.clubmat.entidades.clubmatematicas")
public class ClubmatematicasFacadeREST extends AbstractFacade<Clubmatematicas> {
    @PersistenceContext(unitName = "Clubmat-warPU")
    private EntityManager em;

    public ClubmatematicasFacadeREST() {
        super(Clubmatematicas.class);
    }
}

```

```
@DELETE
@Path("/{id}")
public void remove(@PathParam("id") Integer id) {
    super.remove(super.find(id));
}
```

Ilustración 39: clubmatematicasFacadeRest.java

- La anotación *@Path* se refiere al parámetro que ingresa, en este caso el id del club para la búsqueda del registro. Como se decía anteriormente lo que se envía es una URI con la petición al servidor, este path adiciona el parámetro a la URI para enviar la petición con el parámetro de búsqueda.

```
public abstract class AbstractFacade<T> {
    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public void remove(T entity) {
        getEntityManager().remove(getEntityManager().merge(entity));
    }
}
```

Ilustración 40: AbstractFacade.java

5. Clubmatematicas (Clubmatematicas.java)

El proceso con la entidad es el mismo para cualquier caso de uso, este funcionamiento fue explicado en el desarrollo de la creación de un club de matemáticas (Ver sección 6.1.3 Crear Club).

6.1.6 VER CLUBES

En esta sección se vera en detalle el paso a paso para ver los clubes de matematicas de acuerdo a la arquitectura planteada anteriormente (Ver ilustración 41)

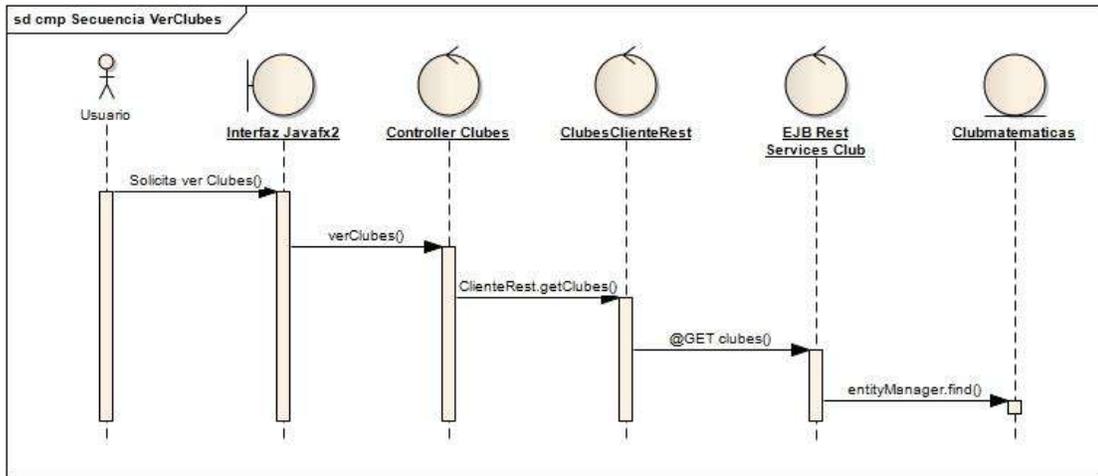


Ilustración 41: Diagrama Secuencia Ver Clubes

Este diagrama muestra el global del comportamiento de los componentes en secuencia para ver los clubes de matemáticas almacenados.

DESARROLLO DEL CASO DE USO VER CLUBES

1. Interfaz JavaFX2 (ClubesInscritos.fxml)

- Esta interfaz representa el componente *Interfaz Javafx2 (Boundary)*. El usuario selecciono previamente la opción de ver los clubes en las pestañas como señala la imagen (Ver ilustración 42)



Ilustración 42: ClubesInscritos.FXML

2. Controller Clubes (ClubesInscritosController.java)

- El botón tiene asociado un método del controlador de esta pantalla **ClubesInscritosController.java (Control)** la cual activa la petición.
- Se crea una instancia del ClienteRest que tiene los servicios.
- Se invoca el método que trae los clubes inscritos. (Ver ilustración 43)

```
@Override
public void initialize(URL url, ResourceBundle rb) {
    // CARGAR TABLA CLUBES (REST)
    this.columnaid.setCellValueFactory( new PropertyValueFactory< Clubmatematicas, Integer >( "idclub" ) );
    this.columnanombre.setCellValueFactory( new PropertyValueFactory< Clubmatematicas, String >( "nombreclub" ) );
    this.columnalema.setCellValueFactory( new PropertyValueFactory< Clubmatematicas, String >( "lema" ) );
    this.columnainstitucion.setCellValueFactory( new PropertyValueFactory< Clubmatematicas, String >( "idinstitucion" ) );

    ClubClienteRest clubClient = new ClubClienteRest(); //Instancia cliente REST
    ClientResponse response = clubClient.findAll_XML(ClientResponse.class); //Se invoca el servicio
    GenericType<List<Clubmatematicas>> genericType = new GenericType<List<Clubmatematicas>>() {};

    List<Clubmatematicas> data = new ArrayList<Clubmatematicas>();
    data = (response.getEntity(genericType));
    clubes = FXCollections.observableArrayList( );
    clubes.addAll(data);
    tablaClubes.setItems(clubres);
    clubClient.close();
}
```

Ilustración 43: ClubesInscritosController.java

3. ClubesClienteRest (ClubClienteRest.java)

- Esta clase se encarga de recibir la petición del cliente y generar una URI para enviar la petición al servidor.
- Crea el path de la petición y teniendo como base la referencia del servidor donde esta ubicado el servicio web.
- Envía la información de la petición en una URI al servidor de acuerdo al método que se invoco. (Ver ilustración 44)
- Es importante que el método para obtener datos en REST es GET.

```
public class ClubClienteRest {
    private WebResource webResource;
    private Client client;
    private static final String BASE_URI = "http://localhost:8080/Clubmat-war/webresources";

    public ClubClienteRest() {
        com.sun.jersey.api.client.config.ClientConfig config = new com.sun.jersey.api.client.config.DefaultClientConfig();
        client = Client.create(config);
        webResource = client.resource(BASE_URI).path("co.edu.javeriana.clubmat.entidades.clubmatematicas");
    }
}
```

```

public <T> T findAll_JSON(Class<T> responseType) throws UniformInterfaceException {
    WebResource resource = webResource;
    return resource.accept(javax.ws.rs.core.MediaType.APPLICATION_JSON).get(responseType);
}

```

Ilustración 44: ClubesClienteRest.java

4. EJB Rest Services Club (clubmatematicasFacadeREST.java)

- Al lado del servidor, los EJB hacen la función de servicios web REST, estos tienen los métodos que llama el cliente.
- En este caso esta clase tiene el *entityManager* que se encarga de los métodos que se utilizan para interactuar con el contexto de persistencia, y en este caso, permite almacenar la instancia del club en la base de datos.
- Como decíamos anteriormente, en este caso de búsqueda la anotación REST utilizada es *@GET* (Ver ilustración 45).
- En este caso esta clase hereda de la clase *AbstractFacade* que contiene los métodos del *entityManager* para las demás clases (Ver ilustración 46).

```

@Stateless
@Path("co.edu.javeriana.clubmat.entidades.clubmatematicas")
public class ClubmatematicasFacadeREST extends AbstractFacade<Clubmatematicas> {
    @PersistenceContext(unitName = "Clubmat-warPU")
    private EntityManager em;

    public ClubmatematicasFacadeREST() {
        super(Clubmatematicas.class);
    }

    @GET
    @Override
    @Produces({"application/xml", "application/json"})
    public List<Clubmatematicas> findAll() {
        return super.findAll();
    }
}

```

Ilustración 45: clubmatematicasFacadeREST.java

```
public abstract class AbstractFacade<T> {
    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public List<T> findAll() {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        return getEntityManager().createQuery(cq).getResultList();
    }
}
```

Ilustración 46: AbstractFacade.java

5. Clubmaticas (Clubmaticas.java)

El proceso con la entidad es el mismo para cualquier caso de uso, este funcionamiento fue explicado en el desarrollo de la creación de un club de matemáticas (Ver sección 6.1.3 Crear Club).

6.2 Estrategia General para desarrollar un caso de uso de negocio: Ejemplo caso de uso “Ver opciones de preguntas”

Se ilustrará ahora un caso de uso de negocio. Este caso de uso se encarga de mostrar las opciones de la pregunta para que el usuario pueda responder la pregunta. Este caso de uso es de negocio ya que se necesita el ID de la pregunta para acceder a las preguntas y es importante realizar los métodos de negocio para realizar esta solicitud.

Cada pregunta tiene varias opciones de respuesta, en este caso específico 4 opciones, donde una opción es verdadera y el resto son falsas.

Esta búsqueda de las opciones por es un método de negocio que es independiente de los servicios que nos brindan los servicios web REST que son generados a partir de las entidades.

En este caso particular se necesitan dos métodos de negocio para completar ese caso de uso, uno de ellos es la búsqueda de las preguntas por el grado y luego de obtener estas preguntas se selecciona la pregunta a la cual se quiere ver en detalle, en este caso sus opciones de respuesta. En la ilustración 47 se ve el desarrollo de este caso de uso de negocio paso a paso.

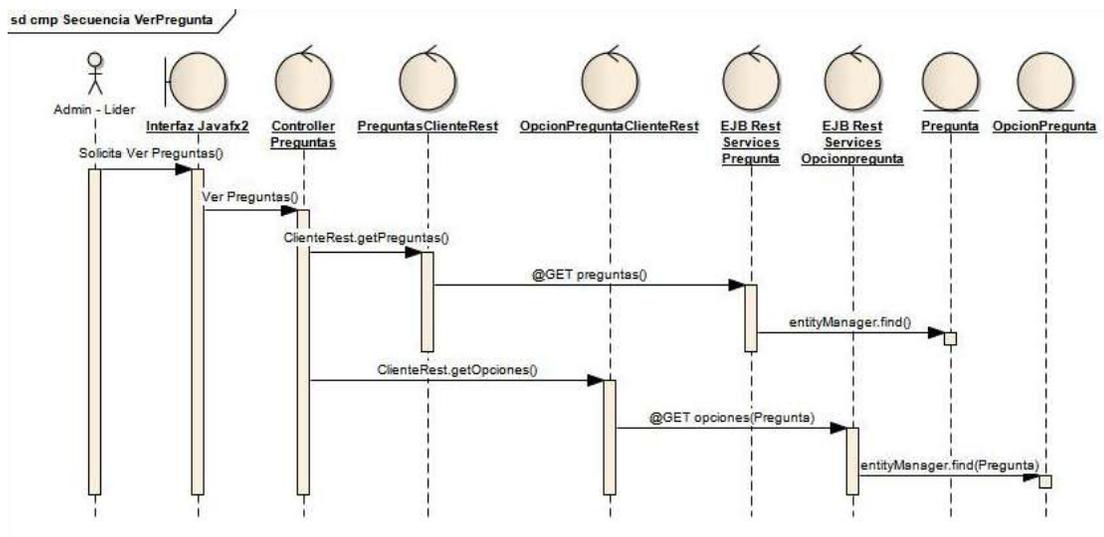


Ilustración 47: Diagrama Secuencia Ver Pregunta

Este diagrama muestra el global del comportamiento de los componentes en secuencia para ver las opciones de la pregunta.

1. Interfaz JavaFX2 (VerPregunta.fxml)

- Esta interfaz representa el componente *Interfaz Javafx2 (Boundary)*. El usuario selecciono previamente la opción Banco de Preguntas (Ver ilustración 48)



Ilustración 48: VerPregunta.FXML

2. ControllerPreguntas (VerPreguntaController.java)

- El botón tiene asociado un método del controlador de esta pantalla *VerPreguntaController.java (Control)* la cual activa la petición.
- Se crea una instancia del ClienteRest que tiene los servicios, en este caso particular es uno creado manualmente que maneja este caso de uso.
- Este realiza el llamado de dos métodos de negocio que pertenecen a ese servicio REST que permiten completar el caso de uso de negocio (Ver ilustración 49).

```
//CARGAR OPCIONES
PreguntasNegocioClienteRest opcl = new PreguntasNegocioClienteRest();
ClientResponse response = opcl.getOpcionesPregunta_XML(ClientResponse.class, BancoPreguntasController.preguntaSeleccionada,
GenericTypesList<Opcionpregunta>> genericType = new GenericTypesList<Opcionpregunta>>() {});
System.out.println("Response : "+response);
data = new ArrayList<Opcionpregunta>();
data = (response.getEntity(genericType));
opcl.close();
```

Ilustración 49: VerPreguntaController.java

3. PreguntasNegocioClienteRest (PreguntasNegocioClienteRest.java)

- Esta clase se encarga de recibir la petición del cliente y generar una URI para enviar la petición al servidor.
- Crea el path de la petición y teniendo como base la referencia del servidor donde esta ubicado el servicio web.

```
public class PreguntasNegocioClienteRest {
    private WebResource webResource;
    private Client client;
    private static final String BASE_URI = "http://localhost:8080/Clubmat-war/webresources";

    public PreguntasNegocioClienteRest() {
        com.sun.jersey.api.client.config.ClientConfig config = new com.sun.jersey.api.client.config.DefaultCli
        client = Client.create(config);
        webResource = client.resource(BASE_URI).path("PreguntasNegocio");
    }

    public <T> T getPreguntasGrado_JSON(Class<T> responseType, String nivelacademico) throws UniformInterface
    WebResource resource = webResource;
    if (nivelacademico != null) {
        resource = resource.queryParam("nivelacademico", nivelacademico);
    }
    resource = resource.path("grado");
    return resource.accept(javax.ws.rs.core.MediaType.APPLICATION_JSON).get(responseType);
}

    public <T> T getOpcionesPregunta_XML(Class<T> responseType, String idpregunta) throws UniformInterfaceExc
    WebResource resource = webResource;
    if (idpregunta != null) {
        resource = resource.queryParam("idpregunta", idpregunta);
    }
    resource = resource.path("opciones");
    return resource.accept(javax.ws.rs.core.MediaType.APPLICATION_XML).get(responseType);
}
```

Ilustración 50: PreguntasNegocioClienteRest.java

- Envía la información de la petición en una URI al servidor de acuerdo al método que se invoco(Ver ilustración 50).
- Es importante que el método para obtener datos en REST es GET, en este caso particular se pide por una búsqueda con un parámetro.

4. EJB Rest Services Preguntas (PreguntasNegocioResource.java)

- Para este método de negocio se crean manualmente una clase EJB con el metodo de búsqueda de las reglas por club.
- Al lado del servidor, los EJB hacen la función de servicios web REST, estos tienen los métodos que llama el cliente.
- En este caso esta clase tiene el *entityManager* que se encarga de los métodos que se utilizan para interactuar con el contexto de persistencia, y en este caso, permite almacenar la instancia del club en la base de datos.
- Como decíamos anteriormente, en este caso de búsqueda la anotación REST utilizada es *@GET*.
- La anotación *@Path* es fundamental ya que determina que la clase es un servicio REST con el nombre de **ClubNegocio**, esta anotación se encuentra en el método de negocio y es importante para diferenciar ese método de los demás en el llamado del servicio web.
- En este método se ha llamado a un *Query* que esta declarado en la Entidad y que permite buscar las reglas del club.
- Se puede ver en la ilustración 51 los dos métodos de negocio que se utilizan para la búsqueda de preguntas y opciones. Esta es la forma para realizar métodos de negocio manualmente.
- Es importante tener en cuenta las anotaciones para la creación, edición y eliminación descritas en el caso de uso CRUD.

```

@GET
@Path("/opciones")
@Produces({"application/xml", "application/json"})
public List<Opcionpregunta> getOpcionesPregunta(@QueryParam("idpregunta") Integer idpregunta) {
    Query query = em.createNamedQuery("Opcionpregunta.findByPregunta");
    query.setParameter("idpregunta", idpregunta);
    List resultList = query.getResultList();
    return resultList;
}

@GET
@Path("/grado")
@Produces({"application/xml", "application/json"})
public List<Pregunta> getPreguntasGrado(@QueryParam("nivelacademico") String nivelacademico) {
    Query query = em.createNamedQuery("Pregunta.findByNivelacademico");
    query.setParameter("nivelacademico", nivelacademico);
    List resultList = query.getResultList();
    return resultList;
}

```

Ilustración 51: PreguntasNegocioClienteRest.java

5. Pregunta (Pregunta.java)

El proceso con la entidad es el mismo para cualquier caso de uso, este funcionamiento fue explicado en el desarrollo de la creación de un club de matemáticas (Ver sección 6.1.3 Crear Club).

Este proceso da por finalizado la búsqueda de las opciones de las preguntas y se ve paso a paso como se cargan desde el cliente hasta la base de datos.

Para ver en detalle los demás casos de uso y su secuencia ver anexo acerca de los diagramas de secuencia de cada caso de uso (Ver Anexo: Diagramas CLUBMAT).

6.2.1 Pruebas Funcionales

Estas pruebas fueron realizadas a los servicios REST creados para el proyecto y validar su funcionamiento.

Para probar los servicios web y su correcto funcionamiento se realiza lo siguiente:

- En modulo web del proyecto buscar el folder RESTful Web Services y hacer Click derecho en el. Luego hacer Click en Test RESTful Web Services, tal como muestra la ilustración 52.

- En la configuración de la prueba elegir que sea local y luego dar ok.

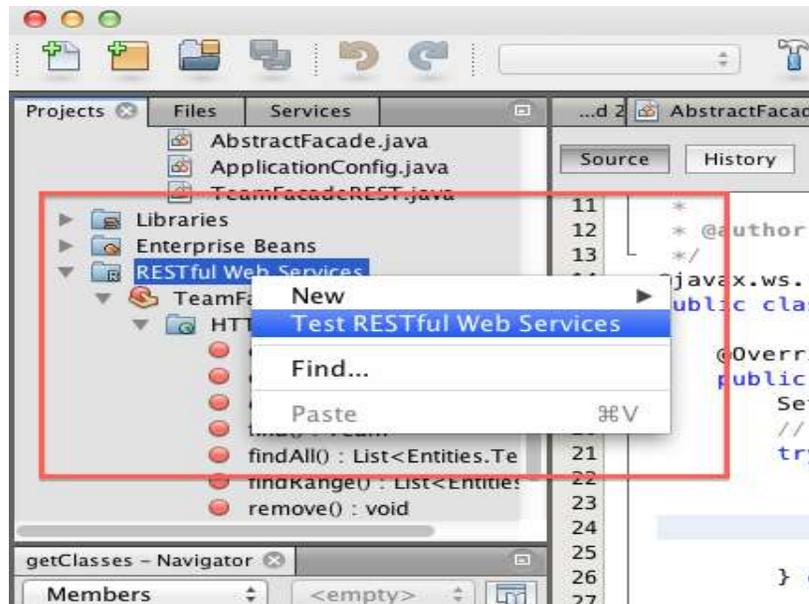


Ilustración 52: Probar Servicios Rest

- Luego la aplicación va a abrir un browser para realizar las pruebas del Servicio Web.
- Elegir el servicio web elegido y probar los métodos http mencionados anteriormente en el formato elegido (XML, JSON) para probar que efectivamente funciona con la base de datos.
- En la ilustración 53 se muestra un ejemplo de la prueba al Servicio Web

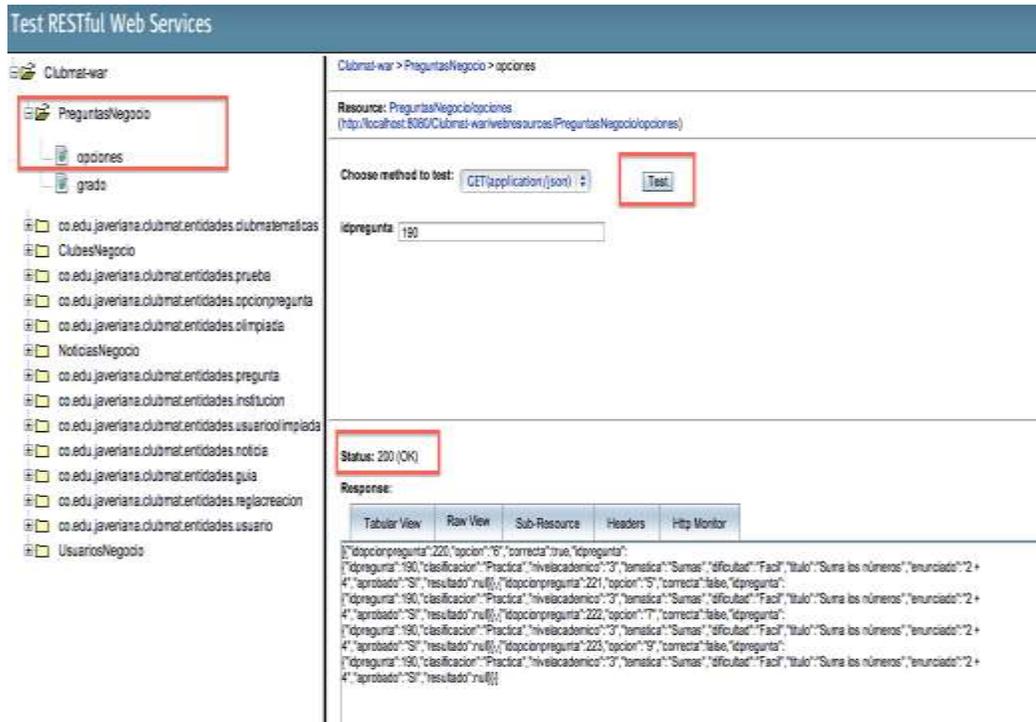


Ilustración 53: Pruebas Servicios Web Rest

De esta forma se realizan las pruebas funcionales de los servicios web REST generados. Las pruebas respectivas de los servicios REST implementados en la aplicación CLUBMAT ver archivo anexo. (Ver Anexo: Pruebas Funcionales y de Aceptacion CLUBMAT) donde se valida la funcionalidad de los servicios web REST.

IV - RESULTADOS Y REFLEXIÓN SOBRE LOS MISMOS

7. Pruebas de Aceptación

Se realizaron las pruebas de aceptación de aceptación por parte del cliente acerca de la versión actual de CLUBMAT. Se realizó una encuesta de aceptación para que los clientes dejaran comentarios y sus opiniones acerca de la aplicación.

Un ejemplo de las encuestas realizadas a los clientes se muestra a continuación (Ver ilustración 51).

ENCUESTA SISTEMA CLUBMAT

Nombre: Elana Duabo
 Institución: P.U.S

Responda las preguntas marcando la casilla correspondiente, siendo 5 la mejor calificación y 1 la mas baja calificación.

PREGUNTAS	1	2	3	4	5
¿Es claro el propósito del programa?					<input checked="" type="checkbox"/>
¿Le parece atractivo visualmente?					<input checked="" type="checkbox"/>
¿Es fácil su navegación?					<input checked="" type="checkbox"/>
¿Este programa aporta a la gestión de los clubes matemáticos?					<input checked="" type="checkbox"/>
¿Cree que este tendrá impacto en la juventud?				<input checked="" type="checkbox"/>	
¿Su funcionamiento es apropiado?				<input checked="" type="checkbox"/>	
¿El sistema es tolerante a fallos?			<input checked="" type="checkbox"/>		
¿Cumple con las expectativas de gestor de clubes de matemáticas en su 1ra versión?					<input checked="" type="checkbox"/>
¿Que calificación le da al sistema CLUBMAT en su 1ra versión.					<input checked="" type="checkbox"/>

COMENTARIOS

Mejoras en grafías e validaciones

ENCUESTA SISTEMA CLUBMAT

Nombre: Clara C. Vera
 Institución: U. Paisajero

Responda las preguntas marcando la casilla correspondiente, siendo 5 la mejor calificación y 1 la mas baja calificación.

PREGUNTAS	1	2	3	4	5
¿Es claro el propósito del programa?					<input checked="" type="checkbox"/>
¿Le parece atractivo visualmente?				<input checked="" type="checkbox"/>	
¿Es fácil su navegación?				<input checked="" type="checkbox"/>	
¿Este programa aporta a la gestión de los clubes matemáticos?				<input checked="" type="checkbox"/>	
¿Cree que este tendrá impacto en la juventud?				<input checked="" type="checkbox"/>	
¿Su funcionamiento es apropiado?					<input checked="" type="checkbox"/>
¿El sistema es tolerante a fallos?			<input checked="" type="checkbox"/>		
¿Cumple con las expectativas de gestor de clubes de matemáticas en su 1ra versión?					<input checked="" type="checkbox"/>
¿Que calificación le da al sistema CLUBMAT en su 1ra versión.					<input checked="" type="checkbox"/>

COMENTARIOS

Gracias Jolán,
 agradeceremos que tengas cuenta las observaciones para las futuras versiones.
 Eritas.

Ilustración 51: Encuestas Aceptación Cliente

Los resultados determinaron la aceptación por parte del cliente, se aceptó el producto y se realizó retroalimentación para mejoras futuras.

Los documentos asociados a las encuestas y las pruebas de aceptación se encuentran en los anexos (Ver Anexo: Pruebas funcionales y de aceptación CLUBMAT)

V – CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

8. Conclusiones

Las conclusiones destacadas del presente trabajo de grado fueron las siguientes:

- En cuestión de desempeño es mejor utilizar comunicación basada en la arquitectura REST, para este proyecto presentó mejores beneficios en contraste a la anterior arquitectura propuesta [4].
- Es importante seguir explorando acerca del framework JavaFx2 en cuestiones de seguridad (Manejo desde el servidor) y de su funcionamiento correcto en los distintos sistemas operativos y browsers. (Ver Manual Mantenimiento).
- Respecto a los servicios web Cuando se trata de aplicaciones públicas y servicios de uso masivo es mucho mejor utilizar REST por la sencillez en su implementación y respuestas, inclusive en clientes móviles es mucho más fácil utilizar REST. Por otro lado si se está pensando en servicios web para corporaciones donde se manejan datos complejos y se necesita una precisión detallada en las respuestas se puede utilizar SOAP. [30]
- Se obtuvo experiencia en la obtención y trato con el cliente, el levantamiento de requerimientos, es importante brindar solución con el trabajo propio a los clientes para los cuales va dirigido el sistema.
- El desarrollo en el framework JavaFX2 se facilitó debido a los conocimientos previos en Java siendo este framework basados en este lenguaje de programación, la curva de aprendizaje fue más corta y se desarrolló en un tiempo prudente.
- Se comprendió el valor de estas herramientas y que se tiene mucho campo por explorar y extender respecto a la integración de las tecnologías JavaFX2 y JavaEE6.
- Respecto a los servicios web se comprendieron los pros y contras de cada tipo de servicio web SOAP y REST, dejando un panorama amplio de posibilidades para el desarrollo de aplicaciones web.
- Se aprendió a trabajar bajo la guía de una metodología y entender la importancia de la planeación de un proyecto paso a paso.
- Mantener un constante repositorio del trabajo acumulado, el manejo de riesgos es importante en estos proyectos donde puede ocurrir cualquier problema y se pierde el trabajo realizado.
- Respecto a la comunidad de Usme de acuerdo a las pruebas de aceptación realizadas el sistema va a tener un impacto grande y es importante realizar estos proyectos sociales para ayudar a la gente que lo necesita.

9. Recomendaciones

- Realizar el levantamiento de requerimientos con anticipación y mantener constante contacto con el cliente ya que realizar cambios al final es complicado, es importante mantener un lineamiento en el desarrollo y seguir la metodología propuesta para no tener inconvenientes de atrasos en el cronograma. Ir dos pasos adelante del cronograma, el tiempo siempre falta.
- Documentar el trabajo realizado en detalle, es importante dejar un trabajo mantenible en el tiempo y dejar la posibilidad abierta para que otras personas retomen el trabajo realizado y se haga de la mejor forma, uno nunca sabe en que momento necesita retomar proyectos o trabajos previos en nuestra vida cotidiana.
- Seguir los manuales y tutoriales para el mantenimiento del sistema así como las versiones de los programas para hacer uso del mismo.

10. Trabajos Futuros

Este trabajo de grado se deja documentado en detalle para que a futuro se realicen actualizaciones y mejoras en las funcionalidades, diseño y arquitectura del sistema. Siendo este un software para una población de niños, el potencial que se tiene de acción es muy grande y el framework JavaFX2 brinda muchas posibilidades y componentes ricos gráficamente para explotar el sistema y realizar cosas más interesantes.

Extensiones propuestas por el cliente para CLUBMAT

En relación a las tecnologías se deben tener en cuenta para el futuro:

- Manejo metódico o automático para establecer los permisos del usuario sobre los elementos de navegación.
- Seguridad de la arquitectura donde debe ser manejada desde el servidor y no desde el cliente como se maneja actualmente en el sistema CLUBMAT.
- Explorar herramientas y plugins para el correcto funcionamiento del Applet generado por el framework JavaFX2 para los diferentes sistemas operativos y browsers.

Luego de las pruebas de aceptación los clientes realizaron unas peticiones para realizar a futuro en la aplicación CLUBMAT y es importante tener en cuenta para una posterior actualización del sistema y son las siguientes:

- Reportes más detallados de las olimpiadas respecto a ganadores y estadísticas.
- Manejar tiempo en las pruebas de olimpiadas y test.

- Asignar un puntaje global para cada estudiante y asignar puntos cuando realice pruebas y olimpiadas, esto con el propósito de fomentar el uso de CLUBMAT en los niños.
- Música y redes sociales de la aplicación.

Estas son algunas peticiones del cliente a futuro y a medida del uso de la aplicación saldrán mas requerimientos para posibles actualizaciones.

VI - REFERENCIAS Y BIBLIOGRAFÍA

En esta bibliografía se encuentran las referencias, autores, empresas y artículos más representativos acerca del tema objetivo para el desarrollo del trabajo de grado.

[1]	Jim Clarke, Jim Connors, and Eric J. Bruno, <i>JavaFX2: Developing Rich Internet Applications</i> . 2009.
[2]	S. Bodoff, <i>The Java EE 6 Tutorial</i> . Boston: Addison-Wesley, 2012.
[3]	Blanca Elvira Oviedo Torres, “CONVOCATORIA INTERNA APOYO A PROYECTOS DE PRESUPUESTO SOCIAL ‘San Francisco Javier’.”2012.
[4]	Andrés Eduardo Sánchez Mendieta, Trabajo de Grado: “Acoplar en la plataforma Java EE 6 el Framework JavaFX 2 como tecnología de interfaz gráfica.” Pontificia Universidad Javeriana. 28-Nov-2012.
[5]	Oracle, “Tutorial de Java EE 6.” [Online]. Available: http://docs.oracle.com/javaee/6/tutorial/doc/ .
[6]	Oracle, “JavaFX - The Rich Client Platform.” [Online]. Available: http://www.oracle.com/technetwork/java/javafx/overview/index.html .
[7]	“¿Qué es Silverlight?,” 2013. [Online]. Available: http://msdn.microsoft.com/es-es/silverlight/bb187401.aspx .
[8]	Craig Larman, <i>Agile and iterative development: a manager's guide</i> . Addison-Wesley Professional, 2004.
[9]	Eliseo Palacios, “Aplicaciones ricas en Internet (RIA). Un enfoque de refactorización.” UNPL Facultad de Informática, Diciembre de-2010.
[10]	Rafael Navarro Maset, “Rest VS Web Services,” 2006. [Online]. Available: http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf .
[11]	Gagliardi-Morales, “RIA JavaFX - Silverlight - Flash,” 2010. [Online]. Available: http://www.sicuma.uma.es/sicuma/independientes/argentina08/Gagliardi-Morales/javafx.htm .
[12]	SommervilleIan, <i>Ingeniería del Software</i> , 7ma ed. Madrid: Pearson Educación,

	2005.
[13]	Ian Gorton, Essential Software Architecture. Springer, 2006.
[14]	A. Goncalves, Beginning Java EE 6 Platform with GlassFish 3: From Novice to Professional, 1st ed. Berkely, CA, USA: Apress, 2009.
[15]	D. Heffelfinger, Java EE 6 with GlassFish 3 Application Server. Packt Publishing, 2010.
[16]	S. Shah and V. Shah, Java EE 6 for Beginners, 1st ed. Maharashtra, India, India: Shroff Publishers and Distributors Pvt. Ltd., 2010.
[17]	S. Shah and V. Shah, Java EE 6 Server Programming for Professionals, 1st ed. Maharashtra, India, India: Shroff Publishers and Distributors Pvt. Ltd., 2010.
[18]	E. Jendrock, I. Evans, D. Gollapudi, K. Haase, and C. Srivathsa, The Java EE 6 Tutorial: Basic Concepts, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.
[19]	Weaver, WeiqiGao, and Stephen Chin, Pro JavaFX 2: A Definitive Guide to Rich Clients with Java Technology. 2012.
[20]	J.F. DiMarzio, JavaFX A Beginners Guide. 2011.
[21]	R. T. Fielding. Architectural styles and the design of network-based software architectures. PhD thesis, University of California, 2000.
[22]	P. Prescod. Common rest mistakes. http://www.prescod.net/rest/mistakes/ , 2002.
[23]	Kim Topley, JavaFX Developer's Guide. 2010.
[24]	J. Weaver, "Best Practices for JavaFX 2.0 Enterprise Applications (Part One)." Apr-2012.
[25]	IEEE Computer Society, "SWEBOK: Guide to the Software Engineering Body of Knowledge." IEEE Computer Society, 2004.
[26]	PRESSMAN, R, Ingeniería de Software: Un enfoque práctico., Quinta Edición. McGraw-Hill,, 2002.
[27]	Peter Eeles and Peter Cripps, The process of Software Architecting. Addison Wesley, 2010.

[28]	M. Keith and M. Schincariol. "Pro JPA 2 Mastering the Java Persistence API.", Apress, 2009.
[29]	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
[30]	C. Pautasso. Soap vs. rest - bringing the web back into web services. http://www.iks.inf.ethz.ch/education/ss07/ws_soa/slides/SOAPvsREST_ETH.pdf , 2007.
[31]	A. Bien, "JavaOne 2012: Building Serious JavaFX 2 Applications". [En línea]. Available: https://oracleus.activeevents.com/connect/sessionDetail.wv?SESSION_ID=3908
[32]	A, Bien. "The Enterprise side of JavaFX". [En línea]. Available: http://www.oracle.com/technetwork/articles/java/enterprisefx-1651840.html
[33]	A, Bien. "The Enterprise side of JavaFX Part Two". [En línea]. Available: http://www.oracle.com/technetwork/articles/java/enterprisefxpt2-1676394.html
[34]	A, Bien. "The Enterprise side of JavaFX Part Three". [En línea]. Available: http://www.oracle.com/technetwork/articles/java/enterprisefxpt3-1735081.html
[35]	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
[36]	PROSOFI, Pontificia Universidad Javeriana, Bogotá, 2012, [En línea]. Available: http://portal.puj-por-tal.javeriana.edu.co/portal/page/portal/Facultad%20de%20Ingenieria/pla_facultad/Prosofi
[37]	The PostgreSQL Global Development Group, 2013. [En línea]. Available: http://www.postgresql.org
[38]	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
[39]	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
[40]	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
[41]	L. Richardson and S. Ruby. RESTful Web Services. O'Reilly, May 2007. ISBN 0596529260.

VII –ANEXOS

Anexo 1. Tutorial y demo JavaFX2 y JavaEE6 con servicios web REST

Este anexo es un documento tutorial donde se explica la realización de una aplicación empresarial web desarrollada en las tecnologías JavaFX2 y JavaEE6 utilizando servicios web basados en REST. Explica desde la instalación de los programas hasta la interacción del cliente y servidor.

[Anexos/TUTORIAL-JAVAFX2-JAVAAEE6-REST.pdf](#)

Anexo 2. Pruebas funcionales y aceptación CLUBMAT

Este documento trae anexo las pruebas funcionales que se realizarán al software CLUBMAT y el formato de aceptación por parte del cliente

[Anexos/PruebasFuncionalesAceptaciónCLUBMAT.pdf](#)

Anexo 3. Manual de usuario CLUBMAT

Este manual de usuario está diseñado para explicar el funcionamiento del programa CLUBMAT a los usuarios que lo utilicen, explica todas las funciones que pueden hacer respecto a su rol en el sistema.

[Anexos/ManualUsuarioClubmat.pdf](#)

Anexo 4: Manual de mantenimiento CLUBMAT

Este manual esta diseñado para el mantenimiento del sistema a futuro por otros desarrolladores y se quiera extender el sistema y mantener en el tiempo. Este manual explica en detalle como fue construido el sistema, sus componentes e interacción entre ellos.

[Anexos/ManualMantenimientoClubmat.pdf](#)

Anexo 5. Prototipos no funcionales CLUBMAT

Este documento muestra las pantallas no funcionales realizadas en principio para el cliente de cómo iba a llegar a ser el software y fue una base importante para modificaciones y refinamiento de requerimientos del software.

[Anexos/PrototiposNoFuncionales.pdf](#)

ANEXO 2

CARTA DE AUTORIZACIÓN DE LOS AUTORES(Licencia de uso)

Bogotá, D.C., 4 de Febrero de 2014

Señores
Biblioteca Alfonso Borrero Cabal S.J.
Pontificia Universidad Javeriana
Ciudad

Los suscritos:

JULIÁN CAMILO ORTEGA MUÑOZ

, con C.C. No

1018430270

En mi (nuestra) calidad de autor (es) exclusivo (s) de la obra titulada:

CLUBMAT 1.1: Extensión de una aplicación web destinada al fortalecimiento de clubes escolares matemáticos integrando JavaFX2 y JavaEE6 con servicios web basados en REST

(por favor señale con una "x" las opciones que apliquen)

Tesis doctoral Trabajo de grado Premio o distinción: Si No
cual: Nominación Mención de Honor

presentado y aprobado en el año 2013, por medio del presente escrito autorizo (autorizamos) a la Pontificia Universidad Javeriana para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mi (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autorizan a la Pontificia Universidad Javeriana, a los usuarios de la Biblioteca Alfonso Borrero Cabal S.J., así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado un convenio, son:

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La conservación de los ejemplares necesarios en la sala de tesis y trabajos de grado de la Biblioteca.	X	
2. La consulta física (sólo en las instalaciones de la Biblioteca)	X	
3. La consulta electrónica - on line (a través del catálogo Biblos y el Repositorio Institucional)	X	
4. La reproducción por cualquier formato conocido o por conocer	X	
5. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet	X	
6. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previo convenio perfeccionado con la Pontificia Universidad Javeriana para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

De manera complementaria, garantizo (garantizamos) en mi (nuestra) calidad de estudiante (s) y por ende autor (es) exclusivo (s), que la Tesis o Trabajo de Grado en cuestión, es producto de mi (nuestra) plena autoría, de mi (nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy (somos) el (los) único (s) titular (es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Pontificia Universidad Javeriana por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Pontificia Universidad Javeriana está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: Información Confidencial:

Esta Tesis o Trabajo de Grado contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de una investigación que se adelanta y cuyos resultados finales no se han publicado. Si No

En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta, tal situación con el fin de que se mantenga la restricción de acceso.

NOMBRE COMPLETO	No. del documento de identidad	FIRMA
JULIÁN CAMILO ORTEGA MUÑOZ	1018430270	

FACULTAD: Facultad de IngenieríaPROGRAMA ACADÉMICO: Ingeniería de Sistemas

ANEXO 3

BIBLIOTECA ALFONSO BARRERO CABAL, S.J.

DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADO

FORMULARIO

TÍTULO COMPLETO DE LA TESIS DOCTORAL O TRABAJO DE GRADO						
CLUBMAT 1.1: Extensión de una aplicación web destinada al fortalecimiento de clubes escolares matemáticos integrando JavaFX2 y JavaEE6 con servicios web basados en REST						
SUBTÍTULO, SI LO TIENE						
AUTOR O AUTORES						
Apellidos Completos			Nombres Completos			
ORTEGA MUÑOZ			JULIÁN CAMILO			
DIRECTOR (ES) TESIS O DEL TRABAJO DE GRADO						
Apellidos Completos			Nombres Completos			
FRANKY			MARIA CONSUELO			
FACULTAD						
FACULTAD DE INGENIERIA						
PROGRAMA ACADÉMICO						
Tipo de programa (seleccione con "x")						
Pregrado	Especialización	Maestría	Doctorado			
X						
Nombre del programa académico						
Ingeniería de Sistemas						
Nombres y apellidos del director del programa académico						
German Alberto ChavarroFlorez						
TRABAJO PARA OPTAR AL TÍTULO DE:						
Ingeniero de Sistemas						
PREMIO O DISTINCIÓN(En caso de ser LAUREADAS o tener una mención especial):						
CIUDAD		AÑO DE PRESENTACIÓN DE LA TESIS O DEL TRABAJO DE GRADO			NÚMERO DE PÁGINAS	
Bogotá		2013			80	
TIPO DE ILUSTRACIONES (seleccione con "x")						
Dibujos	Pinturas	Tablas, gráficos y diagramas	Planos	Mapas	Fotografías	Partituras
		X				
SOFTWARE REQUERIDO O ESPECIALIZADO PARA LA LECTURA DEL DOCUMENTO						

Nota: En caso de que el software (programa especializado requerido) no se encuentre licenciado por la Universidad a través de la Biblioteca (previa consulta al estudiante), el texto de la Tesis o Trabajo de Grado quedará solamente en formato PDF.

MATERIAL ACOMPAÑANTE					
TIPO	DURACIÓN (minutos)	CANTIDAD	FORMATO		
			CD	DVD	Otro ¿Cuál?
Vídeo					
Audio					
Multimedia					
Producción electrónica					
Otro ¿Cuál?					
DESCRIPTORES O PALABRAS CLAVE EN ESPAÑOL E INGLÉS					
Son los términos que definen los temas que identifican el contenido. (En caso de duda para designar estos descriptores, se recomienda consultar con la Sección de Desarrollo de Colecciones de la Biblioteca Alfonso Borrero Cabal S.J en el correo biblioteca@javeriana.edu.co , donde se les orientará).					
ESPAÑOL			INGLÉS		
Aplicación Web			Web Application		
Framework			Framework		
Aplicaciones de Internet Ricas			Rich Internet Applications		
RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS					
(Máximo 250 palabras - 1530 caracteres)					
<p>En el desarrollo de aplicaciones web empresariales existe gran variedad de tecnologías para su realización. Una tecnología importante es JavaEE6 [2]. En la arquitectura JavaEE6 se trabaja como estándar el framework JSF(<i>Java Server Faces</i>) para la realización de interfaces de usuario. Este framework presenta una serie de problemáticas en cuestiones de seguridad, interactividad, tiempo de desarrollo, entre otras. (Ver Sección 1.1Justificación). [2] [11]</p> <p>En la necesidad de resolver estas problemáticas se exploró JavaFX2, un framework RIA (<i>Rich Internet Applications</i>) que por sus características soluciona ciertos aspectos en la interacción del usuario con la aplicación empresarial y es menos susceptible de ser alterado en ejecución</p>					

por usuarios mal intencionados. (Ver Sección: 4. Marco Conceptual). [9]

Previamente se realizó un trabajo de integración entre estas tecnologías (JavaFX2 y JavaEE6), utilizando servicios web basados en SOAP [30]. Este trabajo fue basado en un caso de estudio enmarcado en el programa social PROSOFI de la Pontificia Universidad Javeriana acerca de los clubes de matemáticas de la comunidad de Usme y se desarrolló parte del sistema de información denominado CLUBMAT. [3] [4] [10]

CLUBMAT es un sistema de información web que gestiona y administra clubes de matemáticas y fue el caso de estudio para el trabajo de integración entre JavaFX2 y JavaEE6.

El propósito de este trabajo de grado fue estudiar la arquitectura dejada previamente, en la realización de este software para encontrar mejoras en el acoplamiento de las tecnologías, explorando los servicios web basados en REST [30]. Así mismo, dejar un estándar de integración de estas tecnologías y complementar el sistema CLUBMAT en su totalidad respecto a los requerimientos del cliente.

Lo anterior favoreció para la formulación de conclusiones, con las cuales queremos ayudar a nuestro público objetivo, a obtener una visión general de las condiciones actuales de la aplicación de CLUBMAT, de manera que se pueda dar una implementación rápida y útil de la misma, y así mismo la permanencia y replica de la aplicación a otras comunidades con una necesidad similar.

In the development of enterprise web applications there are a variety of technologies for its implementation. An important technology is JavaEE6 [2]. JavaEE6 architecture utilizes JSF (Java Server Faces) as standard framework for conducting user interfaces. This framework presents a several number of problems in security, interactivity, development time among others. (See Section 1.1 Justificación). [2] [11]

In the need to solve these problems we explored with the JavaFX2 a RIA framework (Rich Internet Applications) which by its nature it solved certain aspects of the user interaction in relation to JSF with the business application. (See Section 4. Marco Conceptual) [9]

Previously done a thesis where they made a coupling between these technologies (JavaFX2 and JavaEE6) using SOAP Web services. This work was based on a case study in the social program PROSOFI from Pontificia Universidad Javeriana about math clubs from Usme (Community of Bogotá) and they made part of the development of the system called CLUBMAT. [3] [4] [10]

CLUBMAT is a web information system that manages math clubs and was the case study work coupling between JavaEE6 and JavaFX2.

The purpose of this work is to study the architecture left previously in the realization of this software to find improvements in coupling exploring technologies REST-based web services. Also, leave a standard integration of these technologies and complement the system CLUBMAT respect to customer requirements.