### INTRODUCCIÓN

En los primeros tiempos, el gasto en equipos electrónicos de computación era principalmente de hardware y el software era gratuito, o era incluido en el precio del hardware. Con la evolución de la técnica, los costos del hardware han disminuido en forma continua, mientras que el costo del software se ha transformado en la parte más importante del gasto en tecnología informática.

El costo de producción de un software es insignificante, si se compara con el alto costo de su desarrollo. Las empresas de software compensan el costo de desarrollo del software con la venta de una gran cantidad de paquetes de estos. El fabricante que más venda dispondrá de mayor dinero para desarrollar más software, hacer campañas de marketing, distribución, etc. Es por ello que el mercado del software tiene tendencia al monopolio.

La importancia del software en el desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) sin embargo ha sido mucho mayor. El software es el que gestiona y agiliza con rapidez y calidad todos los procesos que se ejecutan en un ordenador.

El proceso de desarrollo de software abundante de metodologías y procesos como se conoce en la actualidad, ha sido el resultado de décadas de esfuerzos teóricos y prácticos. Las empresas que basan sus ingresos en el desarrollo de software han requerido de constante especialización y avance en la eficiencia de sus metodologías de desarrollo para poder mantenerse en el mercado con resultados económicos rentables y esto ha provocado por consiguiente una provechosa evolución para el software.

El desarrollo de software a nivel empresarial por tanto requiere de plataformas tecnológicas que garanticen mayor productividad con menor costo y esfuerzo. Los marcos de desarrollo son la base de las plataformas tecnológicas. En este aspecto en la actualidad se considera obligatorio que los desarrolladores utilicen marcos de trabajo ya que representan una estructura conceptual y tecnológica de soporte, normalmente con módulos de software concretos, que sirven de base para que otro proyecto de software pueda ser más fácilmente organizado y desarrollado. Típicamente, los marcos pueden incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un software. El uso de los marcos de trabajo beneficia a los equipos de desarrolladores con procesos automatizados para generar partes del software, ofrece excelentes prácticas y experiencias de programación, conexión y acceso a bases de datos y servicios distribuidos, mayor estandarización, entre otros aspectos. Con estas características se logra finalmente que los desarrolladores se preocupen de lo que realmente es importante para ellos: el software que pretenden desarrollar. Generalmente los marcos de trabajo son desarrollados por terceras empresas que garantizan su actualización, calidad y constante perfeccionamiento.

Con el avance constante del hardware y las redes de telecomunicaciones, ha aumentado también la complejidad del software y por tanto los requerimientos de los marcos de trabajo para crear estos programas o aplicaciones, son mucho mayores y difíciles de soportar. Entre los principales requisitos o requerimientos que deben soportar los marcos de trabajo para ser utilizados en empresas de desarrollo de software empresarial se encuentran:

* Creación de software multiplataforma.
* Generación automática de código.
* Soporte Modelo-Vista-Controlador o N capas.
* Gestión de conexiones a bases de datos o servicios distribuidos.
* Soporte para programación orientada a aspectos.
* Soporte para transacciones y programación en entornos concurrentes.
* Promoción de los mejores patrones de arquitectura y diseño.
* Altamente flexibles y configurables.
* Gestión de las pruebas.
* Gestión de la documentación.

El software sin embargo se desarrolla teniendo en cuenta la plataforma en la que será desplegado y utilizado por los usuarios finales. Las aplicaciones Web son un tipo específico de software que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador. Las aplicaciones Web son populares debido a lo práctico del navegador Web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones sin distribuir e instalar software a miles de usuarios potenciales.

Los marcos de trabajo que permiten crear aplicaciones Web que serán utilizadas en un entorno empresarial tienen además otros requerimientos claramente definidos que deben soportar:

* Generadores de plantillas para las interfaces Web de usuario.
* Almacenamiento de datos en caché.
* Gestión de la seguridad y validación de los datos.
* Gestión de la privacidad e integridad de las sesiones de los usuarios.
* Acceso y mapeo de bases de datos.
* Configuración automática.
* Gestión de servicios Web.
* Soporte AJAX y tecnologías PUSH.

Django[[1]](#footnote-2), Symfony[[2]](#footnote-3), Zend[[3]](#footnote-4) y Spring[[4]](#footnote-5)son algunos de los marcos de trabajo Web que soportan estos requerimientos.

jWebSocket es un servidor de aplicaciones y a la vez un marco de trabajo para desarrollar aplicaciones en tiempo real para la Web utilizando el protocolo WebSocket. Este utiliza el lenguaje de programación Java en el lado del servidor permitiendo que los clientes puedan ser desarrollados en múltiples lenguajes. La creación de aplicaciones en el lado del servidor se compone de filtros, extensiones y escuchadores. Los programadores pueden de manera muy simple crear aplicaciones básicas y comenzar a usarlas.

Debido a que jWebSocket es un marco de trabajo joven, este aún se encuentra en un proceso de maduración tecnológica que se deriva de tu corto tiempo de desarrollo (surgió a finales de 2009), como marco de trabajo aún carece de soporte a los requerimientos necesarios para desarrollar aplicaciones en un entorno empresarial dinámico y exigente de tecnologías eficientes. Utilizarlo provocaría poca productividad y alta propensión a errores, afectando negativamente la calidad. Los desarrolladores deben invertir mucho tiempo y esfuerzo en la creación de partes genéricas de software, carentes en el marco de trabajo, provocando finalmente poca estandarización.

De lo planteado se deriva la siguiente situación problemática: El marco de trabajo jWebSocket ofrece un mecanismo muy básico para desarrollar las aplicaciones en el lado del servidor. Los desarrolladores carecen de mecanismos en el marco de trabajo que faciliten el proceso de desarrollo de software, el manejo de aspectos y la carencia de programación orientada a objetos en el intercambio de mensajes entre los clientes y el servidor son algunas de las carencias.

La utilización del marco de trabajo jWebSocket provocaría poca productividad y alta propensión a errores disminuyendo la calidad, si es empleado en el desarrollo de aplicaciones empresariales complejas. Como resultado de lo anterior jWebSocket no se encuentra apto para ser utilizado como base tecnológica por empresas en el desarrollo de sus aplicaciones.

Debido a la situación anteriormente descrita surge el siguiente problema científico: ¿Cómo lograr mayor productividad y calidad con menor tiempo y esfuerzo en el desarrollo de aplicaciones en un entorno empresarial con el marco de trabajo jWebSocket?

La presente investigación centra su objeto de estudioen los requerimientos que deben soportar los marcos de trabajo para ser utilizados en el desarrollo de aplicaciones Web en un entorno empresarial y en especial el marco de trabajo jWebSocket.

Se define como objetivo general desarrollar una extensión en el marco de trabajo jWebSocket para soportar los requerimientos del desarrollo de software a nivel empresarial, logrando mayor productividad y calidad con menor tiempo y esfuerzo.

Para dar cumplimiento al objetivo general planteado se han derivado las siguientes preguntas científicas:

* ¿Cuáles son los fundamentos teóricos de la investigación?
* ¿Cuáles son los requerimientos soportados por los marcos de trabajo más utilizados en la actualidad para satisfacer las necesidades del desarrollo de aplicaciones Web a nivel empresarial?
* ¿Cómo desarrollar una extensión que aporte al marco de trabajo jWebSocket el soporte para satisfacer los requerimientos de aplicaciones empresariales, ofreciendo mayor productividad y calidad en menor tiempo y esfuerzo?

Para dar respuesta a los objetivos trazados se plantea el cumplimiento de las siguientes tareas de la investigación:

* Fundamentación teórica de la investigación.
* Definición de la metodología, herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.
* Realización del análisis y diseño para lograr una solución óptima.
* Implementación de la aplicación teniendo en cuenta buenas prácticas de programación.
* Realización de las pruebas unitarias y funcionales de la aplicación para verificar el cumplimiento de los requisitos definidos.
* Validación de los efectos de la aplicación en un entorno en tiempo real.

Los métodos científicos utilizados son los siguientes:

Histórico lógico: permite la compresión lógica del objeto de estudio haciendo un análisis riguroso de sus antecedentes y el proceso evolutivo por el cual han transitados todas las tecnologías relacionadas con las tarjetas inteligentes y los servicios en línea.

Analítico-sintético: permite la consulta de diversas fuentes bibliográficas y la extracción de los elementos más importantes que se relacionan con el objeto de estudio. Será de gran importancia en el estudio del estado del arte.

Modelación: permite la abstracción de diversas situaciones y la representación de sus características fundamentales desde determinada perspectiva siendo de gran importancia en el desarrollo de la investigación, muestra de ello serán todos los diagramas y modelos que se producirán a lo largo de la misma.

Bibliográfico: permite realizar un estudio de la documentación referente a los marcos de trabajo en el desarrollo de software y a las herramientas utilizadas para desarrollar la solución esperada.

Se derivan las siguientes variables de investigación:

Variable Independiente:

Extensión del marco de trabajo jWebSocket para permitir el desarrollo de aplicaciones Web empresariales.

Variables Dependientes:

Productividad en el desarrollo de aplicaciones Web.

Calidad en el proceso de desarrollo de software.

Los posibles resultados a obtener con el desarrollo de la investigación son:

* Extensión del marco de trabajo jWebSocket para soportar los requerimientos del desarrollo de aplicaciones empresariales.

Para una mejor comprensión de la investigación, el contenido ha sido estructurado de la siguiente manera:

Capítulo 1. Fundamentación Teórica:Se realiza la fundamentación teórica de la investigación. Se expone un estudio del estado del arte del proceso de desarrollo de software con la utilización de marcos de trabajo a nivel empresarial. jWebSocket como marco de desarrollo para realizar aplicaciones Web.

Capítulo 2. Características, Análisis y Diseño del Sistema: Brinda una fundamentación de la solución propuesta, a partir de la cual se describen las actividades de análisis de la solución, seguidas por la descripción de los procesos del sistema y de la etapa de diseño.

Capítulo 3. Implementación, Pruebas y Validación del Sistema: Se describe la etapa de implementación que conlleva a la obtención del software. Se elaboran y documentan las pruebas realizadas a la solución propuesta para demostrar el correcto funcionamiento de los requerimientos de la misma. Se realiza un análisis de los resultados de la aplicación en un entorno real, comparando indicadores antes y luego de la solución.

### CAPITULO 1. FUNDAMENTACIÓN TEÓRICA

#### Marco teórico

El costo de producción de un software es insignificante, si se compara con el alto costo de su desarrollo. Las empresas de software compensan el costo de desarrollo del software con la venta de una gran cantidad de paquetes de estos. El fabricante que más venda dispondrá de mayor dinero para desarrollar más software, hacer campañas de marketing, distribución, etc. Es por ello que el mercado del software tiene tendencia al monopolio.

El software se define como el conjunto de los programas informáticos, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Considerando esta definición, el concepto de software va más allá de los programas de cómputo en sus distintos estados: código fuente, binario o ejecutable; también su documentación, datos a procesar e información de usuario forman parte del software: es decir, abarca todo lo intangible, todo lo no físico relacionado.

El software de aplicación conocido como aplicación, es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre otros:

* Aplicaciones para Control de sistemas y automatización industrial (Los conocidos sistemas de Supervisión, Control y Adquisición de Datos, SCADA)
* Aplicaciones ofimáticas
* Software educativo
* Software empresarial
* Bases de datos
* Telecomunicaciones (Internet y toda su estructura lógica)
* Videojuegos
* Software médico

Las aplicaciones Web a su vez son programas que utilizan un navegador de internet para visualizar su interfaz. Son soluciones informáticas que los usuarios utilizan accediendo a un servidor a través de Internet o su red interna (intranet). Como interfaz con la aplicación se utiliza un navegador de Internet.

Las ventajas son múltiples:

* Basadas en arquitectura cliente/servidor.
* Los datos y el procesamiento están centralizados en el servidor (no requiere hardware adicional en las terminales).
* No hay límite en el número de terminales.
* Compatible con todos los sistemas operativos.
* Las actualizaciones son inmediatas, ya que no requieren instalación.

La arquitectura cliente/servidor utilizada por las aplicaciones Web, es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

* En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.
* La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores Web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Las aplicaciones Web desde su surgimiento han evolucionado desde contenidos estáticos hasta mega sistemas que gestionan altos volúmenes de información generada por usuarios u otros sistemas informáticos, como son las redes sociales en Internet. En la actualidad se habla de aplicaciones Web más dinámicas, sociales y que sean capaces de gestionar información en tiempo real.

Tiempo real en el área de la informática y las telecomunicaciones se define el conjunto de hardware y software que están sujetos a estrictas restricciones de tiempo desde la ocurrencia de un evento hasta la respuesta del sistema al mismo. Generalmente los tiempos de respuesta en tiempo real son entendidos en el orden de los milisegundos y algunas veces en microsegundos. Un sistema informático en tiempo real debe soportar las restricciones de tiempo asociadas a las respuestas del sistema aunque el mismo se encuentre altamente sobrecargado.

El tiempo real en la Web se rige por el concepto anterior y por tanto se convierte enun paradigma en el que los usuarios reciben la información tan pronto como esta se encuentra disponible, en lugar de tener que consultar periódicamente la fuente de información por actualizaciones.

La Web en tiempo real es rápida, fluida, torrencial y sin fronteras. Rápida significa que la visión del mundo alrededor es siempre fresca. Fluida significa que la información viene a gotas en lugar de a cubos. Torrencial ya que el volumen de cambios es abrumador, sin filtros y sin porteros. Sin fronteras para que la información, los amigos y la experiencia van con el usuario a todas partes de internet. Web en tiempo real cambia el modo de sentir la Web, se hace más inmediato, interpersonal, completo y humano.

La base tecnológica estándar que permite que las aplicaciones Web puedan presumir de comportamientos en tiempo real es el protocolo llamado WebSocket. Este protocolo permite realizar conexiones bidireccionales entre un cliente y un servidor. Consiste en un mecanismo de handshake[[5]](#footnote-6) seguido de intercambios de mensajes sobre la capa TCP[[6]](#footnote-7). El objetivo de esta tecnología es proveer un mecanismo para aplicaciones basadas en navegadores que necesitan comunicación bidireccional con el servidor en vez de tener que realizar múltiples conexiones HTTP.

Las conexiones entre un navegador Web y un servidor basadas en el protocolo WebSocket, producen un consumo de ancho de banda de 1/50 comparado con el consumo del protocolo HTTP, reduce la latencia en el orden de 1/3.

WebSocket es la tecnología estándar para la Web que permite que se puedan establecer conexiones en tiempo real entre un cliente y un servidor.

La plataforma donde surgieron las primeras aplicaciones con soporte para el protocolo WebSocket fue la plataforma Java. Creada por la empresa Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.

La tecnología Java es versátil, eficiente, portable, y su alto grado de seguridad resulta ideal la computación en redes de comunicaciones. Desde laptops hasta grandes centros de datos, desde consolas de juegos hasta supercomputadoras científicas, desde dispositivos móviles hasta internet. Java está en cualquier lugar.

La presente investigación aborda el proceso de desarrollo de software, también conocido como Ingeniería de Software, es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy día es cada vez más frecuente la consideración de la Ingeniería de Software como una nueva área de la ingeniería. La Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos.

El desarrollo de software a nivel empresarial requiere de plataformas tecnológicas que garanticen mayor productividad con menor costo y esfuerzo. Los marcos de trabajo representan la base de las plataformas tecnológicas y en la actualidad han llegado a automatizar todos los requerimientos tecnológicos del desarrollo de software, exceptuando claro está las peculiaridades de los requerimientos.

Los marcos de trabajo representan una estructura conceptual y tecnológica de soporte, normalmente con módulos de software concretos, que sirven de base para que otro proyecto de software pueda ser más fácilmente organizado y desarrollado. Típicamente, los marcos pueden incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un software.

Entre los principales requisitos o requerimientos que deben soportar los marcos de trabajo para ser utilizados en empresas de desarrollo de software empresarial se encuentran:

* Creación de software multiplataforma.
* Generación automática de código.
* Soporte Modelo-Vista-Controlador o N capas.
* Gestión de conexiones a bases de datos o servicios distribuidos.
* Soporte para programación orientada a aspectos.
* Soporte para transacciones y programación en entornos concurrentes.
* Promoción de los mejores patrones de arquitectura y diseño.
* Altamente flexibles y configurables.
* Gestión de las pruebas.
* Gestión de la documentación.

Los marcos de trabajo para desarrollar aplicaciones Web son aquellos que son utilizados por desarrolladores para crear aplicaciones Web. En la actualidad existen cientos de estos marcos debido al uso masivo de la Web e Internet.

Los marcos de trabajo que permiten crear aplicaciones Web que serán utilizadas en un entorno empresarial tienen además otros requerimientos claramente definidos que deben soportar:

* Generadores de plantillas para las interfaces Web de usuario.
* Almacenamiento de datos en caché.
* Gestión de la seguridad y validación de los datos.
* Gestión de la privacidad e integridad de las sesiones de los usuarios.
* Acceso y mapeo de bases de datos.
* Configuración automática.
* Gestión de servicios Web.
* Soporte AJAX y tecnologías PUSH.

#### Tecnologías utilizadas para el desarrollo de la solución

#### Marcos de trabajo

#### jWebSocket

jWebSocket es un marco de trabajo de código abierto para el desarrollo de aplicaciones Web estacionarias y móviles basado en Java en el lado del servidor y en JavaScript del lado del cliente. jWebSocket establece un modelo de comunicación basado en tokens, estos son datos abstractos que a través de una estructura jerárquica y una API proporcionan métodos de acceso a los contenidos. Con el objetivo de realizar una abstracción en la manipulación de los diferentes formatos, el marco de trabajo convierte los paquetes de datos entrantes y salientes en tokens.

El servidor jWebSocket está diseñado para funcionar como servidor de comunicaciones o como servidor Web, brindando total flexibilidad. Ofrece la ventaja de ejecutarse fácilmente desde una línea de comandos o integrarse a la biblioteca de una aplicación existente de Java.

jWebSocket como servidor Web proporciona un conjunto importante de funcionalidades y su arquitectura extensible permite añadir fácilmente características adicionales a un sistema independiente. Por otra parte los administradores pueden configurar el servidor exactamente como sea necesario y dejar a un lado todos los módulos que no necesiten. En un clúster las extensiones se pueden utilizar como servicios, por lo que jWebSocket perfectamente es compatible con SOA (Service Oriented Architectures) en un entorno totalmente basado en eventos. Estas características muestran la fortaleza y flexibilidad del marco de trabajo para el desarrollo de aplicaciones Web estacionarias y móviles, multiplataforma, multisectorial y compatible con todos los navegadores.

jWebSocket es el marco de trabajo y la vez el servidor de aplicaciones con licencia libre más robusto y flexible para la plataforma Java, está respaldado por excelentes resultados en cuanto a rendimiento y escalabilidad. Su pequeña curva de aprendizaje le permite a los desarrolladores crear y desplegar nuevas aplicaciones en corto tiempo. Además jWebSocket es un proyecto comunitario en el cual un equipo de estudiantes de la Facultad Regional de la UCI “Mártires de Artemisa”, constituye alrededor de un 50% de los integrantes del equipo de desarrollo.

La presente tesis está enfocada a mejorar el soporte tecnológico del marco de trabajo jWebSocket y por tanto es el marco escogido.

#### Spring

Spring es el marco de trabajo más popular para el desarrollo de aplicaciones empresariales sobre la plataforma Java. Millones de desarrolladores usan Spring para crear aplicaciones con alto rendimiento, fácilmente probables y código reutilizable sin contratiempos.

La licencia de Spring es Apache License[[7]](#footnote-8), considerada una licencia de software libre.

Entre las características más importantes de Spring destacan:

* El contenedor de dependencias más completo, que proporciona una configuración centralizada, automatizada y facilita cableado de los objetos de su aplicación. El contenedor no es invasivo y es capaz de ensamblar un sistema complejo partiendo de un conjunto de componentes débilmente acoplados en una forma coherente y transparente. El contenedor brinda agilidad y mejora la capacidad de pruebas de aplicaciones y la escalabilidad al permitir que los componentes de software puedan ser primero desarrollados y probados de forma aislada, y luego ampliarse para el despliegue en cualquier entorno (Java 2 Segunda Edición (J2SE) o Java 2 Edición Empresarial (J2EE)).
* Una capa de abstracción común para la gestión de transacciones, permitiendo gestores de transacción enchufables, lo que permite más fácilmente delimitar las transacciones sin tener que lidiar con los problemas de bajo nivel. Las estrategias genéricas de la API para transacciones de Java (JTA) y un único origen de datos JDBC (Conexión a bases de datos de Java) están incluidos. El soporte de transacciones de Spring no está atado a entornos J2EE.
* Una capa de abstracción JDBC que ofrece una jerarquía de excepción significativa, simplifica el manejo de errores y reduce enormemente la cantidad de código que se necesita escribir. No es necesario volver a escribir otro bloque “finally” para utilizar JDBC de nuevo.
* Integración con TopLink[[8]](#footnote-9), Hibernate[[9]](#footnote-10), JDO[[10]](#footnote-11), y Mapas iBATIS SQL[[11]](#footnote-12).
* Soporte para Programación Orientada a Aspectos (AOP), plenamente integrado en la gestión de la configuración de Spring.
* Un flexible marco de trabajo Modelo Vista Controlador para aplicaciones Web. Este marco es altamente configurable a través de interfaces de estrategia, y tiene integraciones para múltiples tecnologías de interfaces de usuario, como JSP[[12]](#footnote-13), Velocity[[13]](#footnote-14), Tiles[[14]](#footnote-15) y iText[[15]](#footnote-16).

Se pueden reutilizar todos los componentes de Spring sin ninguna necesidad de adquirir dependencias adicionales. Esto ofrece enorme flexibilidad ya que las aplicaciones solo usan de Spring lo que necesitan.

Spring ha sido seleccionado como marco de trabajo a reutilizar debido a que incorpora la experiencia de más de 10 años en el desarrollo de aplicaciones empresariales. Es a la vez la base de referencia sobre la cual se han construido otros marcos de desarrollo. Su arquitectura flexible y desacoplada permite que se puedan utilizar en la solución solamente los módulos requeridos.

Spring es la tecnología no oficial de la plataforma Java más utilizada en entornos empresariales y promueve el uso de los mejores patrones y estándares de desarrollo de software.

### CAPITULO 2. CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA

#### Introducción

En el presente capítulo se interpretan las necesidades del sistema especificándolas mediante los requerimientos funcionales y no funcionales. Se realiza el modelado del dominio con el objetivo de comprender el contexto, identificando para esto las entidades principales y las relaciones entre ellas. Además se detallan brevemente los artefactos de la metodología de desarrollo SXP, propuesta en el capítulo anterior. Describiéndose las historias de usuario y las tareas de ingeniería asociadas a las mismas.

#### Propuesta de solución

La solución propuesta para la problemática que se aborda en la investigación consiste en una extensión para el marco de trabajo jWebSocket que permite desarrollar aplicaciones utilizando programación orientada a eventos en el lado del cliente y el servidor. Esta característica permite a los desarrolladores que están familiarizados con este tipo de programación, tener mucho más productividad, al permitir diseños totalmente desacoplados, se benefician más aquellos equipos compuestos por muchos desarrolladores, al poder desarrollar muchos componentes de modo concurrente. La integración entre el cliente y el servidor permite que la capa del controlador en las aplicaciones del lado del servidor se autogenere en el cliente sin que los desarrolladores tengan necesidad de programar estos componentes, disminuyendo considerablemente el tiempo de desarrollo.

La extensión eleva la utilización de la programación orientada a objetos en el lado del servidor, reemplazando la estructura de datos abstracta Token por eventos que son representados mediante clases.

La solución ofrece soporte para los más diversos aspectos relacionados con la comunicación cliente – servidor. El control de la seguridad, almacenamiento de datos en caché, tratamiento de excepciones, validación de los argumentos de entrada y argumentos de salida, son algunos de los aspectos que son manejados por la extensión, de este modo los desarrolladores tiene que programar aún menos.

La extensión reutiliza el marco de trabajo Spring para permitir el desarrollo de las aplicaciones del lado del servidor. El marco de trabajo Spring aporta a la extensión el soporte para un amplio grupo de requerimientos necesarios en el desarrollo de software.

#### Modelo de Dominio

Dentro de los artefactos que genera la metodología SXP se encuentra la plantilla del Modelo Historias de Usuario del Negocio, donde se definen las características específicas del negocio, así como la forma en que interactúa el sistema con los clientes y viceversa. Pero si dicho negocio no está bien definido entre los clientes y los ejecutores del proyecto; entonces es generado el llamado Modelo de Dominio, a continuación se presenta el del sistema que se propone:

Aplicación Web cliente: Aplicación Web cliente que posee las interfaces gráficas de usuario mediante las cuales estos pueden comunicarse con la aplicación del lado del servidor.

Capa de Eventos: Bibliotecas de clases que permite el manejo de eventos. Esta capa se comunica con el cliente jWebSocket para poder transmitir los mensajes al servidor.

Cliente jWebSocket: Establece la conexión con el cliente, envía y recibe paquetes de información hacia y desde el servidor.

Servidor jWebSocket: Representa los componentes del marco de trabajo jWebSocket gestiona la comunicación de bajo nivel con el cliente.

Aplicación Servidora: Representa la aplicación en el lado del servidor creada utilizando la extensión EventsPlugIn.

#### Descripción de la Arquitectura

La arquitectura de una aplicación es la vista conceptual de su estructura. Toda aplicación contiene código de presentación, procesamiento y almacenamiento de datos. La arquitectura de las aplicaciones difiere según como este distribuido su código.

Para el desarrollo de la solución se hará uso de la arquitectura en capas, dicha arquitectura tiene como objetivo primordial separar la lógica de negocios de la lógica de diseño. La principal ventaja de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que ocurra algún cambio sólo afectará dicho nivel, logrando obviar las demás capas del sistema. Permite además distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles.

#### Diseño con Metáforas

La metodología SXP asume las fortalezas tanto de SCRUM como de XP. Esta última metodología define el término metáforas, el cual es considerado una historia compartida que describe cómo debería funcionar el sistema. La práctica de las metáforas consiste en formar un conjunto de nombre que actúen como vocabulario para hablar sobre el dominio del problema.

El diseño con metáforas es sencillamente el diseño de la solución más simple que pueda funcionar y ser implementado en un momento dado del proyecto; lo cual genera el artefacto conocido como Modelo de Diseño, que a su vez está compuesto por un diagrama de paquetes.

Los diagramas de paquetes describen los elementos físicos del sistema y sus relaciones. Muestra las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados. Estos muestran además la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables.

A continuación se representa el diagrama de paquetes para el sistema que se propone.

Ilustración Diagrama de paquetes de la solución

#### Diagrama de Componentes

En el diagrama de componentes se describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de softwareque entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros.

A continuación se presenta el diagrama de componentes para el sistema que se propone.

Ilustración Vista global del diagrama de componentes

Ilustración Diagrama de componentes del servidor

Ilustración Diagrama de componentes del cliente

**Descripción**:

* Los diagramas representan una solución basada en la arquitectura cliente – servidor.
* La aplicación en el lado del cliente utiliza una capa encargada del soporte al modelo de eventos.
* La aplicación del lado del cliente están compuesta por extensiones, filtros y escuchadores.
* La aplicación en el lado del servidor utiliza varios componentes del marco de trabajo Spring.
* La aplicación del lado del servidor están compuesta por extensiones, filtros y escuchadores.
* La aplicación en el lado del cliente envía eventos a la aplicación del lado del servidor. La aplicación del lado del servidor recibe eventos del cliente y les da soporte mediante extensiones y escuchadores que los procesan para devolver un resultado al cliente.
* La aplicación del lado del servidor envía eventos a la aplicación del lado del cliente. La aplicación en el lado del cliente recibe y ofrece soporte a los eventos del servidor mediante extensiones y escuchadores que los procesan para devolver un resultado al servidor.

### CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

#### Diagrama de Despliegue

El diagrama de despliegue es un artefacto que modela la arquitectura en tiempo de ejecución de un sistema. En él se indica la situación física de los componentes lógicos desarrollados, lo que significa que sitúa el software en el hardware que lo contiene. A continuación se muestra el diagrama de despliegue, que representa la distribución física del sistema en términos de cómo se distribuirán las funcionalidades entre los nodos, donde cada nodo representa un recurso de cómputo, siendo estos procesadores o dispositivos hardware que se necesitan para el despliegue del sistema:

Ilustración Diagrama de despliegue de la solución

Cliente: Aplicación cliente que se conecta por medio del protocolo WebSocket al Servidor para lo cual solo necesitará un navegador que soporte WebSocket sobre cualquier sistema operativo.

Servidor:Representa la aplicación servidora a la cual una aplicación cliente se conecta para obtener datos que son necesarios para generar las interfaces gráficas al usuario.

#### Estrategia de Validación

EventsPlugIn es una extensión para el marco de trabajo jWebSocket que permite desarrollar aplicaciones utilizando programación orientada a eventos. Eleva la utilización de la programación orientada a objetos y ofrece soporte para los más diversos aspectos relacionados con la comunicación entre el cliente y el servidor. Promueve la utilización del marco de trabajo Spring en el desarrollo de aplicaciones del lado del servidor.

Al EventsPlugIn ser una solución tecnológica para realizar software, la mejor forma de probar y validar la solución es desarrollando con ella aplicaciones complejas en dimensión y requerimientos. Sin embargo existen otros mecanismos enfocados a la validación por separado de los componentes que integran la solución con son las pruebas de unidad y pruebas funcionales.

Para la validación de la presente solución se han realizado aplicaciones de prueba, pruebas de unidad y pruebas funcionales, en las que se han obtenido después de un arduo proceso de depuración, resultados satisfactorios.

#### Resultados Obtenidos

Por el resultado satisfactorio de los casos de pruebas realizados a la extensión del marco de trabajo jWebSocket para el desarrollo de aplicaciones Web empresariales queda disponible su versión 1.0. Se obtuvo un sistema que cumple con todas las especificaciones para permitir la gestión de tarjetas inteligentes en tiempo real.

#### Funcionalidades Obtenidas

Entre las principales funcionalidades que posee la extensión del marco de trabajo jWebSocket para el desarrollo de aplicaciones Web empresariales en su versión 1.0 se pueden mencionar:

* Soporte para la programación orientada a eventos en el lado del cliente y el servidor.
* Integración con el marco de trabajo Spring, especialmente con el módulo de validación, seguridad y el contenedor de IOC.
* Soporte para la creación de extensiones, filtros y escuchadores en el lado del servidor para atender notificaciones de eventos desde el cliente y soportar el chequeo de aspectos en la notificación de estos eventos.
* Soporte para un mecanismo de sesiones que permite a los usuarios almacenar persistentemente sus datos de sesión en el servidor.
* Soporte para un mecanismo de control global de excepciones globales con posibilidad de notificaciones al administrador del sistema u otros roles definidos.
* Soporte para un mecanismo de ejecución de pruebas automatizadas a la aplicación del lado del servidor.

#### Aporte Social y Económico

La extensión del marco de trabajo jWebSocket para el desarrollo de aplicaciones Web empresariales representa un innovación tecnológica que aumenta la productividad y eficiencia de las empresas o equipos de desarrollo que utilizan el marco de trabajo jWebSocket para desarrollar sus aplicaciones.

El ahorro de tiempo y mejores y eficientes prácticas de desarrollo se traducen directamente en impactos económicos positivos para las empresas.

En el ámbito social, la extensión forma parte de una solución llamada jWebSocket que posee licencia de software libre, permitiendo así que pueda ser utilizada sin costo alguno por los desarrolladores interesados. La extensión ha sido desarrollada en la UCI y por tanto se benefician de soporte directo y capacitación los estudiantes, profesores y trabajadores de esta institución.

1. https://www.djangoproject.com/ [↑](#footnote-ref-2)
2. http://www.symfony.es/ [↑](#footnote-ref-3)
3. http://framework.zend.com/ [↑](#footnote-ref-4)
4. http://www.springsource.org/ [↑](#footnote-ref-5)
5. Es un término del idioma inglés que se refiere a la conversación protocolar que realizan los sistemas en arquitecturas cliente/servidor antes de establecer una conexión. [↑](#footnote-ref-6)
6. [↑](#footnote-ref-7)
7. [↑](#footnote-ref-8)
8. [↑](#footnote-ref-9)
9. hibernate.org/ [↑](#footnote-ref-10)
10. [↑](#footnote-ref-11)
11. [↑](#footnote-ref-12)
12. [↑](#footnote-ref-13)
13. [↑](#footnote-ref-14)
14. [↑](#footnote-ref-15)
15. [↑](#footnote-ref-16)