

# Framework VAADIN, como herramienta para desarrollar aplicaciones de internet

## Vaadin framework, as tool to develop internet applications

Gabriel Gregorio Salvatierra Tumbaco<sup>1,\*</sup>, Carlos Alfredo Silva Villafuerte<sup>1,†</sup>, y Jorge Luis Veloz Zambrano<sup>1,‡</sup>

<sup>1</sup>Universidad Técnica de Manabí, Ecuador.

{gsalvatierra3575,csilva6721,jveloz}@utm.edu.ec

Fecha de recepción: 9 de agosto de 2016 — Fecha de aceptación: 16 de agosto de 2016

**Resumen**—Este trabajo de investigación demuestra los beneficios de utilizar el framework vaadin para el desarrollo de un sistema clínico en entorno web, el cual permite la gestión y administración de las consultas médicas realizadas en los diferentes consultorios, proporcionando a una empresa mayor rapidez y exactitud en el proceso de atenciones médicas, aumentando la productividad y satisfaciendo las necesidades de sus clientes de manera óptima. En el proceso investigativo se utilizó el enfoque cualitativo, que permitió identificar la problemática de los procesos administrativos a través de herramientas como las entrevistas y las fichas de observación. Para el desarrollo del software se utilizó la metodología del Proceso Ágil Unificado (PUA), que determinó el alcance, la viabilidad, el costo, los participantes, el modelado de requerimientos funcionales, la arquitectura del sistema y la estructura de la base de datos utilizados durante la fase de construcción del software.

**Palabras Clave**—Sistema clínico, Frameworks, Vaadin, PUA.

**Abstract**—The accelerated growth of Massive Open Online Course online courses has changed the way to acquire new knowledge, through multiple online learning platforms, universities now have the opportunity to expand the possibilities of granting A different education to the traditional one, due to the problematic that afflicts our society, to exist limitations in the learning by multiple factors that are unleashed product of the marginalization and the poverty. In many cases leads to an educational desertion, for this reason the desire to implement a new methodology of teaching learning that allows students access to electronic media such as the internet and different online platforms thus decreasing the gap Digital, thus diversifying the options of training and specialization. This paper documents the benefits of MOOCs and, on the other hand, studies aspects that allow students to learn from anywhere and at any time. In the new educational scenarios, university courses should not only meet the social-educational needs demanded by a globalized world, but universities must develop courses that include aspects of internationalization, where virtual mobility of students and teachers is promoted, as well as The international recognition.

**Keywords**—Clinical system, Frameworks, Vaadin, PUA.

### INTRODUCCIÓN

Los programas de escritorio, con los cuales muchos de los programadores aún se identifican, están siendo desplazados por las aplicaciones web, aquellas que brindan mayor flexibilidad, mejor rendimiento y reducción de costos finales, solventando los problemas de soporte y compatibilidad entre sistemas operativos presentados en los sistemas convencionales.

Los desarrolladores de software convencional poseen experiencia y dominio en entornos de desarrollo de escritorio; sin embargo, sienten temor al nuevo paradigma que establece el desarrollo web. Uno de los factores que influyen a este temor es la curva del aprendizaje de un nuevo lenguaje de programación y de las complicaciones que tendrían al migrar un software en producción.

Existen lenguajes de programación que facilitan adaptarse a los nuevos entornos de desarrollo web, sin mayores contratiempos o con resultados inesperados. Los lenguajes de programación cuentan con frameworks para agilizar el desarrollo,

garantizando funcionalidad y mejorando significativamente la apariencia visual.

Las aplicaciones RIA (Rich Internet Applications), logran integrar funcionalidades de las aplicaciones de escritorio, las tecnologías de telecomunicaciones y las aplicaciones web. En el mercado existe una gran cantidad de framework tipo RIA, tanto de códigos abiertos como privativos, que garantizan el desarrollo de aplicaciones web como si fueran de escritorio.

GWT es un conjunto de herramientas de desarrollo para crear y optimizar aplicaciones complejas basadas en el navegador. Su objetivo es permitir el desarrollo productivo de las aplicaciones web de alto rendimiento, sin que el desarrollador tenga que ser un experto en peculiaridades del navegador, XML, HttpRequest y JavaScript. GWT es utilizado por muchos productos en Google, incluido AdWords, AdSense, Flights, Hotel Finder, Offers, Wallet, Blogger. Es de código abierto, completamente libre, y utilizado por miles de desarrolladores de todo el mundo (Google, 2006).

Vaadin es un framework basado en GWT, para el desarrollo de aplicaciones web AJAX, que permite a los desarrolladores de software construir interfaces de usuario de gran calidad con Java como si fuese una aplicación de escritorio pero más fácilmente, tanto en el lado del servidor como en el lado del cliente, el cual proporciona un conjunto de librerías de

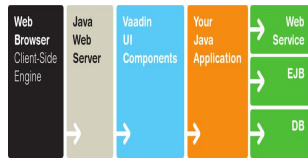
\*Ingeniero en Sistemas Informáticos.

†Ingeniero en Sistemas Informáticos.

‡Ingeniero en Sistemas Computacionales, Magister en Telecomunicaciones.

componentes de interfaz de usuario listos para ser usados y un marco limpio para crear componentes propios. Está enfocado en brindar facilidad de uso, reusabilidad, escalabilidad y cumple con los requisitos de las aplicaciones que demandan las empresas (Vaadin Ltd., 2016).

En la figura 1 se observa la estructura del framework Vaadin.

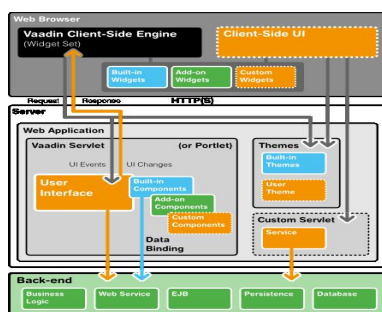


**Figura 1.** Vaadin Web Framework.  
**Fuente:** (Isola, 2013).

El modelo de programación web tradicional invierte mucho tiempo en el aprendizaje de nuevas tecnologías, éstas podrían mejorar su productividad concentrándose en la lógica de la aplicación. Desde el lado del servidor, Vaadin se encarga de manejar la interfaz de usuario en el navegador y de las comunicaciones AJAX entre cliente y servidor. Con el enfoque de Vaadin, no es necesario aprender ni tratar directamente con las tecnologías del navegador, como son HTML o JavaScript (Vaadin Ltd., 2016).

El marco Vaadin consiste de un API del lado del servidor (server-side API), un API del lado del cliente (client-side API), una multitud de componentes/Controles de interfaz de usuario (user interface components/widgets) en ambos lados, temas (themes) para controlar la apariencia, y un modelo de datos (data model) que permite vincular los componentes del lado del servidor directamente a los datos. Para el desarrollo del lado del cliente, se incluye el compilador de Vaadin, que permite compilar Java a JavaScript (Vaadin Ltd., 2016).

En la figura 2 se observa la arquitectura de ejecución del framework vaadin.



**Figura 2.** Arquitectura de ejecución de Vaadin.  
**Fuente:** (Isola, 2013).

La institución solicitante no contaba con un software para gestionar centralizadamente la información de historiales clínicos de los pacientes en los diferentes consultorios médicos del país, para lo cual se desarrolló un sistema clínico web utilizando la herramienta Vaadin, reduciendo costos y tiempo de desarrollo; una vez implementado el software, permitió optimizar los servicios médicos que la empresa ofrece.

## MATERIALES Y MÉTODOS

El software se desarrolló en el lenguaje de programación **Java**, utilizando el Framework **Vaadin**, como entorno de desarrollo integrado se utilizó **Eclipse Mars** y como motor de base de datos **MariaDB**, todas estas herramientas son de código abierto.

La metodología empleada para el desarrollo de esta investigación fué la del Proceso Ágil Unificado (PUA), el cual brinda un revestimiento en serie (por ejemplo, una secuencia lineal de actividades de ingeniería de software) que permite al equipo visualizar el flujo general del proceso de un proyecto de software, y se resume en las siguientes fases: concepción, elaboración, construcción y transición (Pressman et al., 2010).

En la fase de **concepción** del PUA, se estableció la comunicación con el cliente, para identificar a los participantes y requerimientos del sistema para el desarrollo.

En la fase de **elaboración**, se incluyó las actividades de comunicación y esquema del modelo general del proceso. La elaboración mejoró y amplió los casos de uso preliminares desarrollados y aumentó la representación de la arquitectura del software.

En la fase de **construcción**, se desarrolló los componentes del software (código) que hizo que cada caso de uso sea operativo para los usuarios finales. Después se implementaron en código fuente, todas las características y funciones necesarias para el incremento del software. A medida que se implementaron los componentes, se diseñaron y efectuaron pruebas unitarias para cada uno; además, se realizaron actividades de integración.

En la fase de **transición**, Se entregó el software y la documentación necesaria (manuales de usuario, guías de solución de problemas, procedimientos de instalación, etc.) a los usuarios finales para las pruebas beta, quienes reportaron tanto los defectos como los cambios necesarios.

## RESULTADOS Y DISCUSIÓN

Este framework permitió desarrollar un sistema clínico en entorno web, a través de la metodología PUA, obteniendo los resultados, en cada una de las fases, como se menciona a continuación:

### Fase de Iniciación

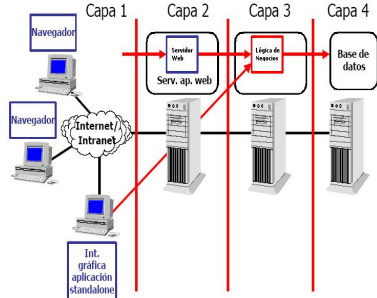
El sistema clínico permitió llevar el control de los historiales clínicos de pacientes tales como los registros de datos personales, signos vitales, exámenes físicos, diagnóstico, prescripción y órdenes de exámenes de laboratorio.

Respecto a la administración y parametrización del sistema, el administrador pudo hacer el registro de usuarios (tipo, privilegios), registros de empresas, sucursales, seguros e importaciones de vademécums. Además; pudieron visualizar la auditoría de todos los procesos realizados dentro del sistema.

Los reportes estuvieron realizados de acuerdo a los requerimientos propios del cliente (partes diarios, consolidados mensuales, entre otros) e incluyeron reportes estadísticos tales como, índices de atenciones de pacientes por sucursal, consultorio y por médico, índice de las enfermedades más comunes, todos ellos parametrizables.

**Fase de Elaboración**

Se utilizó la arquitectura del modelo de cuatro capas ya que es el estándar de las aplicaciones web desarrolladas con Java como lenguaje de programación, como se observa en la figura 3.



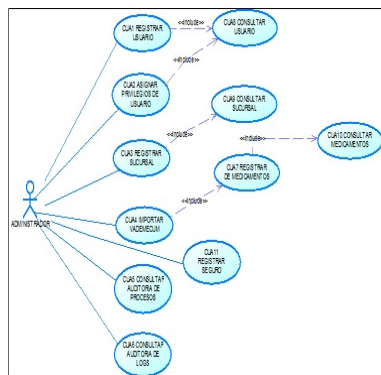
**Figura 3.** Arquitectura del modelo de cuatro capas.  
**Fuente:** (Log, 2013).

Cabe mencionar que la lógica de los datos estuvo a cargo de la base de datos mediante procedimientos almacenados.

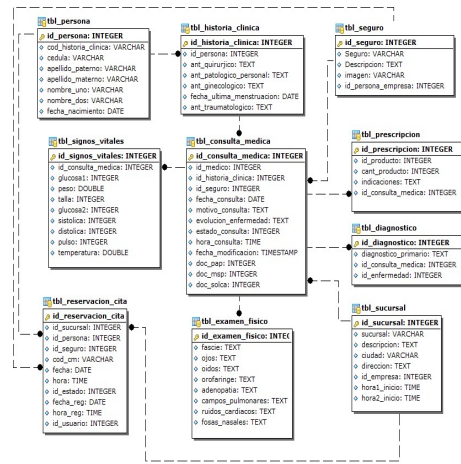
La capa de negocio actúa como interfaz entre el servidor web y la base de datos; además, si en un futuro se necesita expandir el sistema, la base de datos puede residir en el mismo servidor web o en otro servidor que lleve la carga de procesamiento de datos por separado.

**Fase de Construcción**

Para el modelado del sistema clínico se utilizó el lenguaje unificado de modelado (UML), con diagramas de casos de uso, tal como se muestra en la figura 4 y secuencia para la definición de módulos y objetos a usar. Así mismo; el modelo físico de la base de datos fue diseñado, como se muestra en las figuras 5.

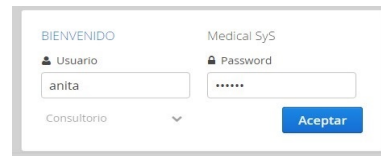


**Figura 4.** Arquitectura del modelo de cuatro capas.  
**Fuente:** (Log, 2013).



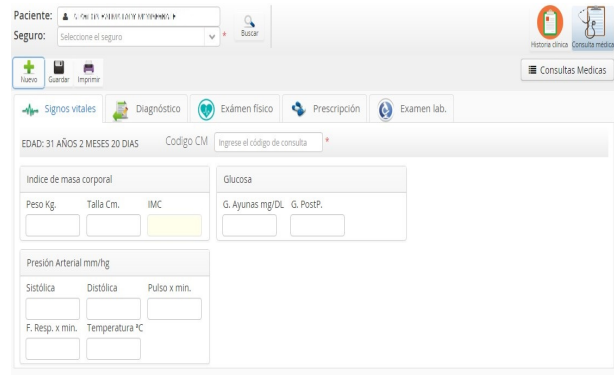
**Figura 5.** Tablas principales de la base de datos.  
**Fuente:** Elaboración propia.

Se accedió a la aplicación web a través de la dirección IP del servidor, mostrando como resultado el formulario de ingreso al sistema, como se observa en la figura 6.



**Figura 6.** Formulario de login sistema web.  
**Fuente:** Elaboración propia.

Se realizó una consulta médica, y posteriormente el registro de los datos correspondientes en el formulario de consulta médica, como se detalla en la figura 7, posteriormente se emitió la ficha de prescripción, ver figura 8.

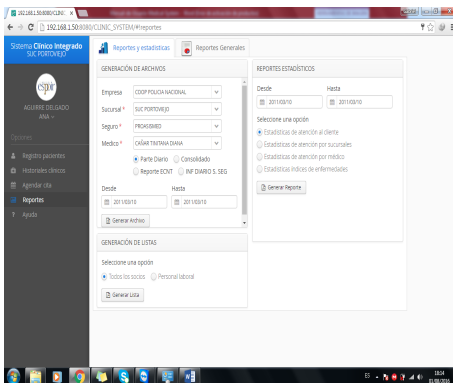


**Figura 7.** Formulario de registro de signos vitales.  
**Fuente:** Elaboración propia.



**Figura 8.** Ficha de prescripción.  
**Fuente:** Elaboración propia.

Se generó el reporte de consultas médicas realizadas por cada médico a través del formulario de reportes, como se muestra en la figura 9.



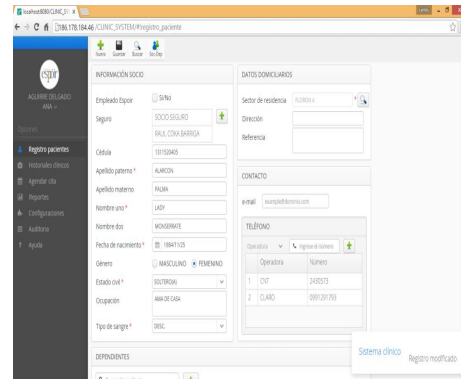
**Figura 9.** Ventana de generación de reportes y estadísticas.  
**Fuente:** Elaboración propia.

En base a la descripción de los casos de uso y las pruebas realizadas al final de cada iteración en la fase de construcción, se determinó el cumplimiento con los requerimientos del sistema clínico. Las pruebas iniciales fueron realizadas por el equipo de desarrollo y posteriormente por el cliente, obteniendo como resultado correcciones y nuevas especificaciones en el software.

**Fase de Transición**

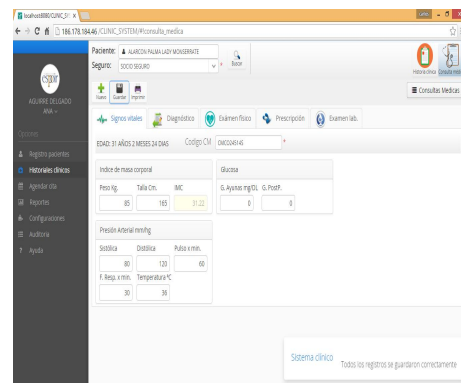
En las pruebas de caja negra realizadas por cada caso de uso, que indica un requisito funcional, se evaluó dicho caso de uso y se estableció la evaluación de la prueba como positiva o negativa. Ciertos casos de uso no fueron considerados para pruebas puesto que son procesos que por lógica el sistema debe realizar.

Las pruebas en el módulo de registro de pacientes, no encontraron error alguno, permitiendo realizar el registro de la información. Figura 10.



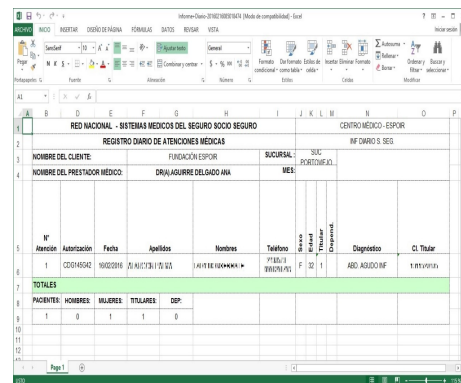
**Figura 10.** Pruebas del módulo de registro de pacientes.  
**Fuente:** Elaboración propia.

Las pruebas realizadas en el módulo de consulta médica, dieron éxito en el registro de los datos, como se muestra en la figura 11.



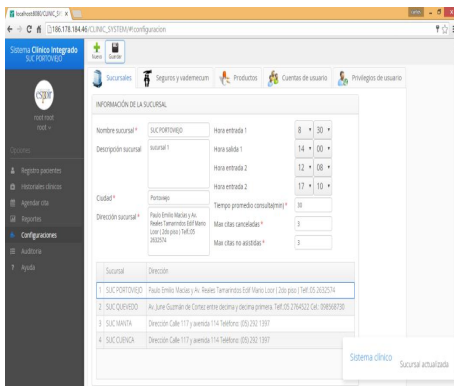
**Figura 11.** Pruebas del módulo de consultas de médicas.  
**Fuente:** Elaboración propia.

Las pruebas del módulo de reportes y estadísticas fueron realizadas en el módulo de reportes, en el cual no existió problema alguno, en la generación de archivos de Excel y Pdf. Figura 12.



**Figura 12.** Archivo de Excel generado a través del módulo de reporte.  
**Fuente:** Elaboración propia.

El módulo de configuración, no mostró la presencia de error alguno. Figura 13.



**Figura 13.** Pruebas del módulo de configuraciones.  
**Fuente:** Elaboración propia.

En el módulo de auditoria se descartó error alguno al momento de la implementación de la aplicación web. Figura 14.

Fecha	Hora	Usuario	Modulo y acción	Descripción	Información
31	2016-05-26	vanessa	HISTORIA CLINICA- ANTECEDENTES MODIFICAR	CHILUINGA OÑA ALEXANDRA JACO... 120548323	P: 192.168.7.82 I: IVEGADOR: Mozilla Firefox OS: Windows
32	2016-05-26	vanessa	HISTORIA CLINICA- ANTECEDENTES MODIFICAR	CHILUINGA OÑA ALEXANDRA JACO... 120548323	P: 192.168.7.82 I: IVEGADOR: Mozilla Firefox OS: Windows
33	2016-06-28	carolina	PACIENTE SOCIO REGISTRAR	VERA CERVILLOS MA... ESPERANZA 1305...	P: 192.168.2.71 I: IVEGADOR: Mozilla Firefox OS: Windows
34	2016-06-28	carolina	HISTORIA CLINICA- ANTECEDENTES REGISTRAR	VERA CERVILLOS MA... ESPERANZA 1305...	P: 192.168.2.71 I: IVEGADOR: Mozilla Firefox OS: Windows
35	2016-06-13	eddy/ce	CONSULTA MEDICA- SIGNOS VITALES MODIFICAR	CORONEL POVICI STEVEN RODRIGUE... 262520262 - FE...	P: 192.168.7.81 I: IVEGADOR: Mozilla Firefox OS: Windows

**Figura 14.** Pruebas del módulo de auditoria.  
**Fuente:** Elaboración propia.

**CONCLUSIONES**

Una vez finalizado el trabajo de investigación se concluye lo siguiente:

- La implementación del Proceso Ágil Unificado (PUA) como metodología de desarrollo, facilitó la toma de decisión en relación a la arquitectura y el modelado del software; además, permitió llevar un mejor control de calidad del mismo.
- La utilización del Framework Vaadin, proporcionó facilidad en el diseño de las interfaces de la aplicación web sin necesidad de utilizar lenguajes de diseño web como CSS, HTML, brindando facilidad en las tecnologías asíncronas como AJAX, que permite la carga de información en segundo plano, sin interferir con el diseño y visualización de la página.
- El uso de la aplicación web optimizó en gran nivel la gestión administrativa centralizada, permitiendo generar reportes de las consultas médicas y las estadísticas que contribuyen favorablemente la toma de decisiones a nivel jerárquico.
- Se logró reducir notablemente los costos de implementación; así también, la existencia de reportes de auditoria,

permiten a los administradores mantener la supervisión de los procesos realizados por los usuarios del sistema.

**AGRADECIMIENTOS**

A la Fundación Espear por darnos la confianza y facilidades en la implementación del Sistema Médico en su red de consultorios.

**REFERENCIAS BIBLIOGRÁFICAS**

Google (2006). Google Web Toolkit.  
Isola, L. (2013). Area Desarrollo TopGroup S.A.: Introducción a Vaadin.  
Log, A. (2013). Desarrollo de Aplicaciones Web: 1.2.1 Aplicaciones de 2,3 y n capas.  
Pressman, R., Martín, R., and Aguilar, L. (2010). *Ingeniería del software: un enfoque práctico*. Área de informática y computación. McGraw-Hill.  
Vaadin Ltd. (2016). Vaadin – User Interface Components for business apps - vaadin.com.