

ESTUDIO DE LA TÉCNICA TEST DRIVEN DEVELOPMENT (TDD) Y DESARROLLO DEL SISTEMA PARA LA ADMINISTRACIÓN DE CONSULTORIOS MÉDICOS

Luis Fernando Cuenca Girón, Andrés Veintimilla Pesantez, Ing. Jenny Alexandra Ruiz, Ing. Mario Bernabe Ron.

{lfcuenca, ajveintimilla, jaruiz, mbron}@espe.edu.ec

Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador.

RESUMEN

Con el continuo desarrollo de las aplicaciones distribuidas en el sector médico, poco a poco se presentan más a menudo los sistemas para automatizar los procesos internos de los consultorios médicos. El objetivo del presente trabajo es encontrar las ventajas de trabajar con la técnica de desarrollo guiado por pruebas frente al ciclo tradicional de testing dentro de la construcción de sistemas. Para llevar a cabo con el análisis, diseño e implementación de una aplicación Web basada en la Metodología SCRUM conjuntamente con TDD. Como parte de diseño, se aplica la Norma IEEE-830, que contempla los lineamientos para la recolección y análisis de la ingeniería de requerimientos. Para el análisis entre paradigmas de testing se implementan las pruebas unitarias y de integración como marco de referencia para la elaboración de un cuadro comparativo frente a TDD. Los resultados de la investigación muestran que la técnica de desarrollo guiado por pruebas en colaboración con el framework JUnit no simplemente abarca el testing de la aplicación, sino que conduce el diseño de la misma, mejorando el modelo de codificación de software. Además al presentarse un fallo sobre un test se mantiene un detalle con el cual se puede detectar y corregir el error a tiempo. Además se muestra la aceptación por parte de los usuarios que catalogan la aplicación como confiable, disponible y eficaz a la vez para gestionar los procesos para la administración del consultorio médico.

Palabra clave: Estándar IEEE-830.

ABSTRACT

With the continuous development of distributed applications in the medical sector, gradually systems are more often present to automate the internal processes of medical offices. The aim of this work is to find the advantages of working with the technique of test-driven development to the traditional cycle testing in building systems. To perform the analysis, design and implementation of a Web-based application SCRUM methodology in conjunction with TDD. As part of design, the IEEE-830 standard, which provides guidelines for the collection and analysis of requirements engineering is applied. For analysis between paradigms unit testing and integration testing as a framework for the development of a comparative table in front of TDD are implemented. The research results show that the technique of test-driven development in collaboration with the JUnit framework encompasses not just the testing of the application, but leads the design of it, improving the model coding software. In addition to a fault occurs on a detail with a test which can detect and correct the error in time is maintained. Further acceptance by users cataloging application as reliable, available and effective at the same time to manage the processes for managing medical office shown.

KeyWords: IEEE-830 Standard.

1. INTRODUCCIÓN

El progreso en el desarrollo del software ha tomado nuevas prácticas en los proyectos actuales, donde el entorno del sistema es muy variado, exige disminuir costos, reducir los tiempos de desarrollo y mantener una calidad de alto nivel, es por eso que emergen las metodologías ágiles como una posible respuesta a los problemas en el desarrollo de software. (Patricio Letelier, 2006)

En las metodologías ágiles la técnica de desarrollo de software dirigido mediante pruebas (TDD), es una técnica fundamental del ciclo de desarrollo que permite al grupo de desarrollo modificar y refactorizar el código existente sin miedo a romper funcionalidades, pues cada una de ellas tiene un test asociado que debe cumplirse.

TDD requiere disciplina para llevarse a cabo, pues para cada funcionalidad que se desee agregar se debe escribir, antes que nada, uno o varios test que comprueben que la funcionalidad está implementada correctamente. Esta necesidad de escribir las pruebas antes de la funcionalidad propiamente, lo cual obliga a escribir pruebas que no dependen del código que se haya estructurado. (ChileÁgil, 2010)

La obligación de escribir varios Test antes de implementar funcionalidades, obliga a considerar con más detenimiento las responsabilidades y diseño de éstas, lo que generalmente se traduce en código más robusto y entendible para otros desarrolladores.

Luego del estudio de la citada técnica, fue aplicada en el desarrollo de un sistema para la administración de consultorios médicos, delineados por los requerimientos del centro médico del Dr. Wladimir Herrera.

Para los consultorios médicos la prioridad es brindar un servicio de calidad, íntegro y eficaz para salvaguardar la salud del paciente, sin embargo, se ha hecho de lado las potenciales ventajas que hoy en día se podría obtener de la tecnología, una de las mayores ventajas están en el tratamiento de las citas, consultas, historiales, pacientes y médicos a través de una página web la cual se encontrará disponible las 24 horas del día todos los días.

Frente a este escenario, se propone el análisis, diseño e implementación de un sistema Web que gestiona el proceso de reserva de citas médicas, historias clínicas, diagnósticos, recetas y curvas de crecimientos de los pacientes del consultorio médico.

Este documento describe el desarrollo de un sistema Web transaccional y la aplicación distribuida mediante la metodología SCRUM.

El resto del artículo ha sido estructurado como sigue: la sección 2 muestra la metodología de desarrollo de acuerdo a las fases de la misma. La sección 3 detalla el diseño e implementación de la propuesta. La sección 4 se detalla la evaluación y pruebas. En la sección 5 se analizan algunos trabajos relacionados. Finalmente sección 6 se presentan las conclusiones y líneas de trabajo futuro sobre la base de los resultados obtenidos.

2. METODOLOGÍA

En esta sección se expone los fundamentos teóricos para la implementación del sistema para la administración de consultorios médicos, que respaldaron cada una de las fases del ciclo de desarrollo del software. La norma IEEE-830 se aplica para el levantamiento de requerimientos, la metodología SCRUM para el desarrollo del sistema y TDD para el testing del sistema.

En la ilustración 1 se muestra las fases del proceso de desarrollo de software como son: Análisis de requerimientos, diseño y arquitectura, programación, integración y mantenimiento. Para la especificación de requerimientos se aplica la Norma IEEE-830 que aporta con el proceso sistemático y metodológico de levantamientos de requisitos.

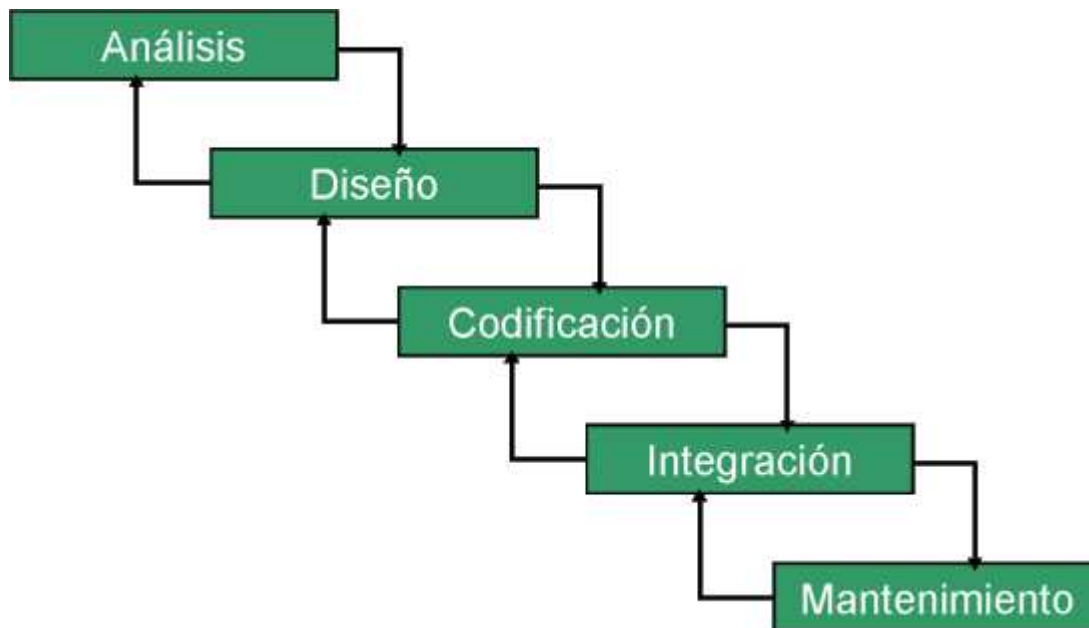


Ilustración 1: Fases del Proceso de Desarrollo del Software. (upm, Entornos de programación, 2006)

La ilustración 2 muestra el proceso de desarrollo de un sistema bajo los lineamientos de la metodología SCRUM. En una primera instancia se reúne los requerimientos del proyecto constituyéndolos en una lista de trabajo, para luego transformarlas en tareas de trabajo o Sprints, conjuntamente con las reuniones diarias se va desarrollando las tareas de iteración, para entregar un resultado que cubre el requisito priorizado por el cliente.



Ilustración 2: Ciclo del desarrollo Scrum. (reserv, 2010)

TDD permite evaluar el sistema de forma modular, expresando la funcionalidad que necesitaba ser implementada a continuación dentro del diseño del software.

La ilustración 3 muestra las 6 faces que contempla TDD: (Araújo, 2007)

- Agregar una prueba.
- Correr las pruebas.
- Hacer un pequeño cambio en el código de producción.
- Correr las pruebas.
- Reconstruir.
- Correr las pruebas.

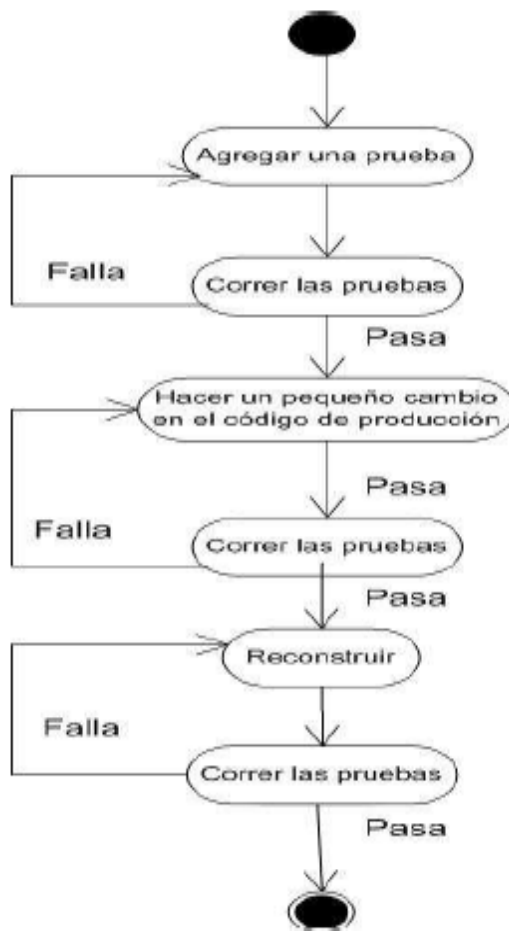


Ilustración 3: Ciclo TDD. (Araújo, 2007)

3. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA

3.1. Diseño

La ilustración 4 muestra la implementación del sistema basado en el modelo MVC (modelo, vista, controlador). La capa Web es del cliente quien interactúa a través de un browser o navegador Web. Para el desarrollo de este proyecto se utiliza la tecnología Java Server Faces (JSF). Para la capa de Negocio se implementan los Web Service SOAP. Esta arquitectura ayuda a crear proyectos empresariales robustos. La capa de Datos donde se implementa la tecnología JPA (Java Persistence API) para acceder a la base de datos MySQL.

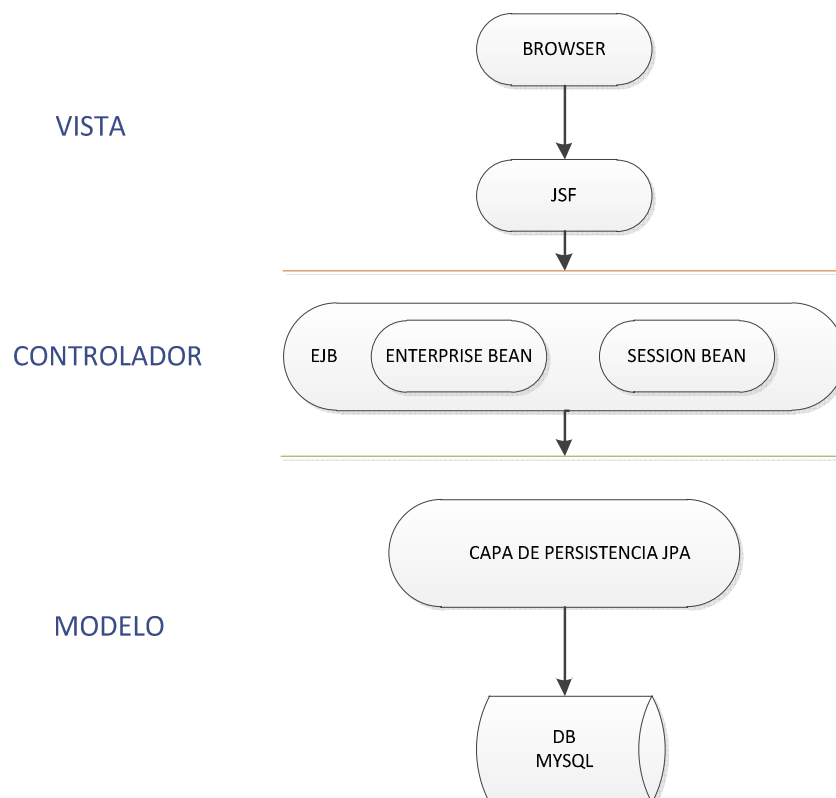


Ilustración 4: Arquitectura MVC. (Cuenca Luis, 2014)

3.2. Implementación

Para este desarrollo se aplica la metodología SCRUM la cual contribuye con las fases del ciclo de desarrollo de software basándose en tareas iterativas. Esta metodología permite tener un control y una distribución adecuada de las actividades de trabajo y a la vez centrarse en actividades prioritarias, logrando una aplicación distribuida y sencilla. A continuación se describen los diagramas y procesos que generan más valor en el proyecto como son: Diagrama de casos de uso y modelo lógico.

3.2.1. Diagrama de casos de uso.

En la ilustración 5 se muestra la interacción del usuario hacia el sistema, accediendo dependiendo del perfil a los distintos módulos.

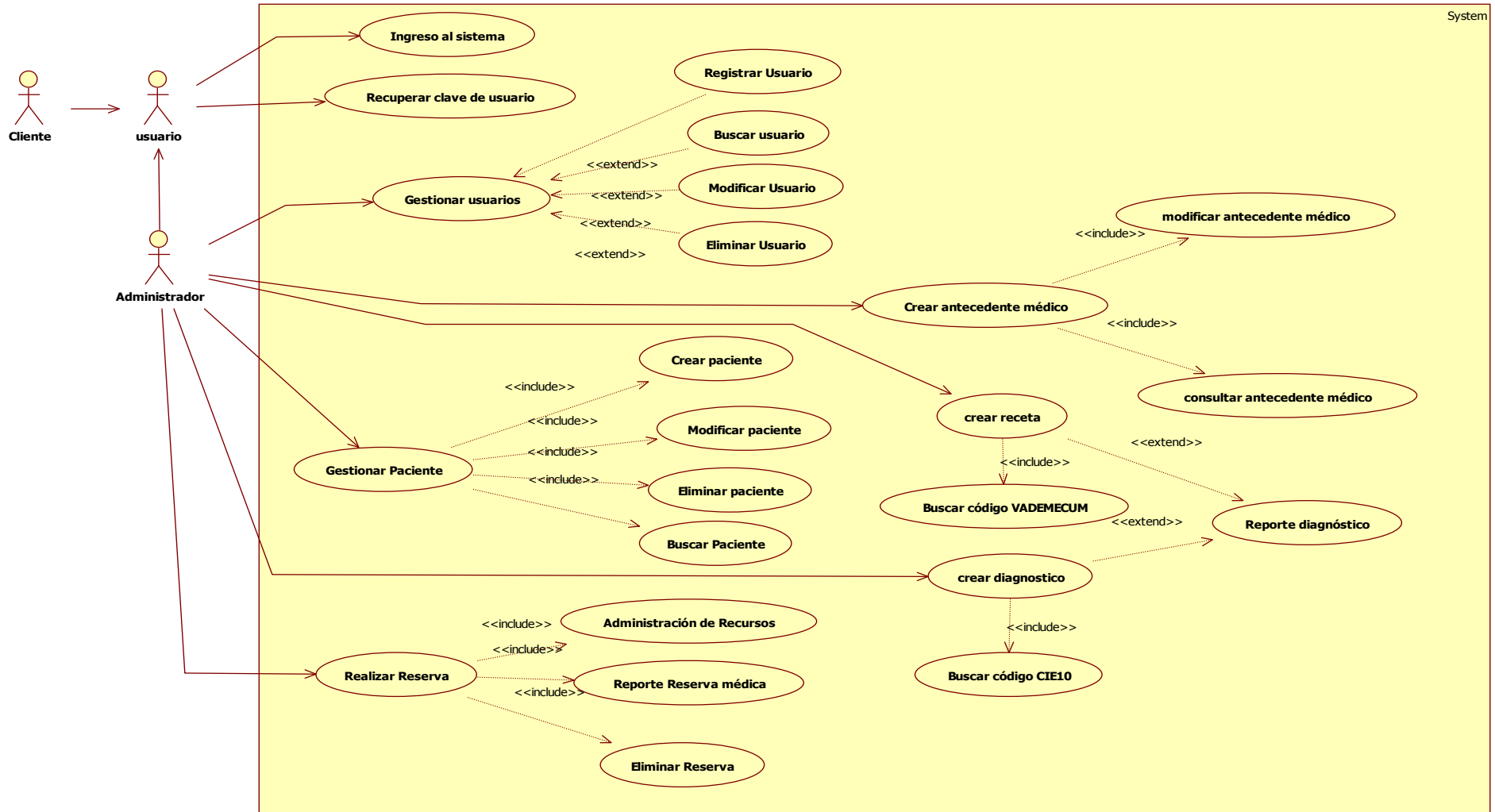


Ilustración 5: Diagrama de casos de uso. (Cuenca Luis, 2014)

3.2.2. Modelo lógico

El modelo lógico aporta con el diseño y análisis de la reglas de negocio para posteriormente crear el script de la base de datos que se implementa sobre MySQL. El modelo presenta las siguientes relaciones: un paciente tiene varios antecedentes médicos, un antecedente médico tiene varias historias clínicas, un paciente tiene varias citas médicas, un horario está puede estar asignado a varias citas médicas, una cita médica tiene varios diagnósticos, un código CIE 10 puede estar en varios diagnósticos, un código VADEMECUM puede estar en varias recetas.

En la ilustración 6 se muestra el modelo lógico de la base de datos del sistema.

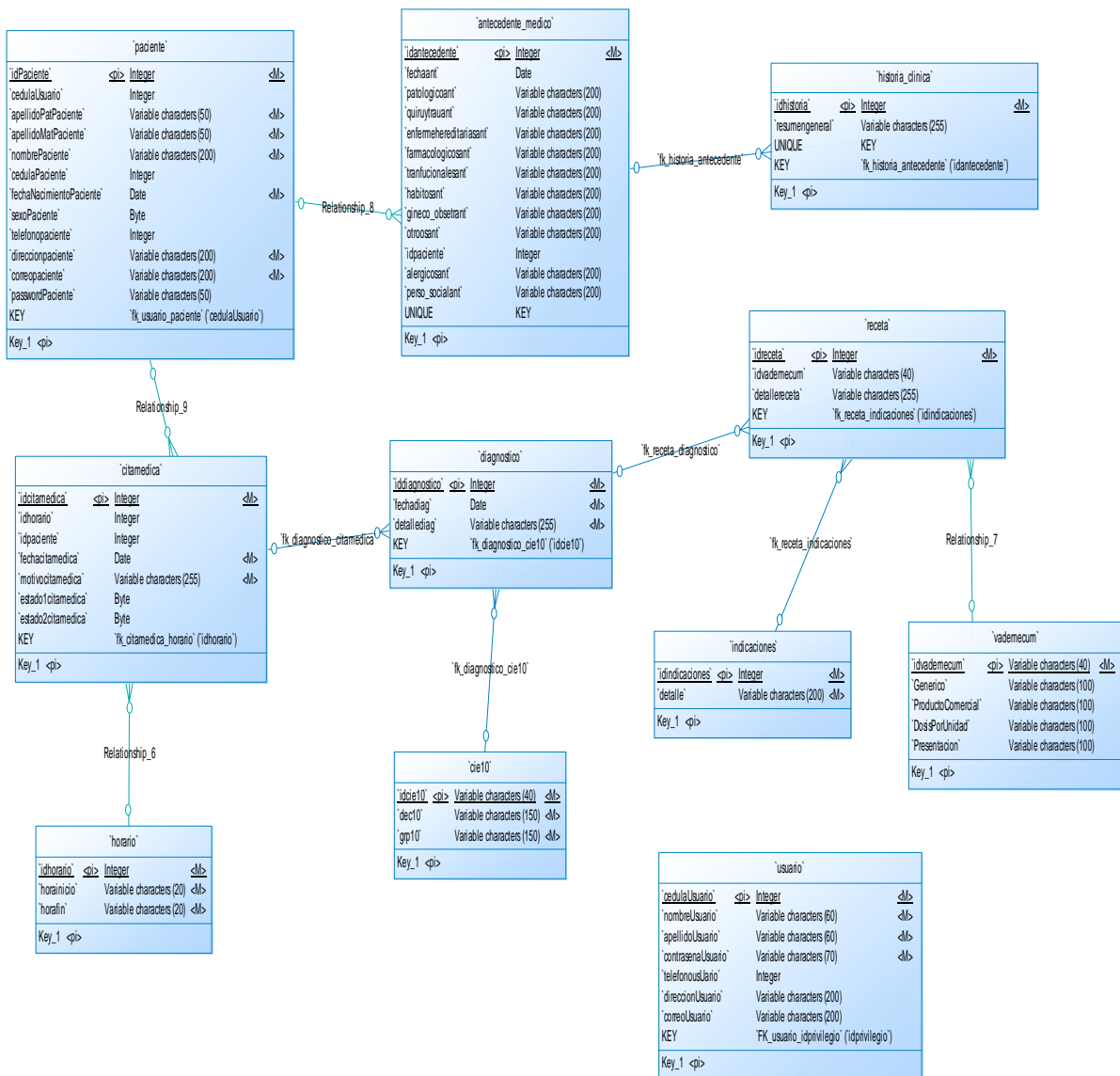


Ilustración 6: Modelo Lógico de la base de datos. (Cuenca Luis, 2014)

4. EVALUACIÓN Y PRUEBAS

En esta sección se exponen las pruebas que se utilizan en el sistema de administración de consultorios médicos. Se presentan las pruebas de aceptación, pruebas del sistema y las pruebas unitarias.

4.1. Pruebas de aceptación

Este tipo de pruebas están sujetas a las historias de usuario. Ayudan a comprobar los requisitos que el usuario solicitaba en caso de presentarse una funcionalidad distinta a la normal.

A continuación se muestra la prueba de aceptación de la siguiente acción: administración de usuarios. Se comprueba el proceso que se sigue en caso de presentarse una situación adversa.

Administración de Usuarios

- **¿Qué pasaría si se ingresa un nombre de usuario y contraseñas inválidas?**

El sistema no permitirá el ingreso al aplicativo web, indicando un mensaje de error “usuario o contraseña incorrectos”.

- **¿Qué pasaría si se registra un usuario con datos incorrectos?**

El sistema valida cada entrada de la información de registro, indicando el campo errado con un mensaje “Error en el campo”.

- **Ingresar al sistema con usuario estándar y visualizar únicamente la opción de la agenda electrónica.**
- **Ingresar al sistema con usuario administrador y visualizar todas las opciones del sistema.**

4.2. Pruebas del Sistema

Este tipo de pruebas aportan a verificar el ingreso, procesamiento y salida de datos para obtener los resultados esperados. Se encamina a los requerimientos del sistema ayudando a prevenir errores en: rendimiento, funcionalidad, resultados de interfaz de usuario, persistencia a la base de datos.

En la tabla 1 se muestra la prueba del sistema para las operaciones de: recupera clave de acceso, nuevo paciente, ingreso al sistema con los distintos perfiles de usuario.

Tabla 1: Pruebas del sistema: recuperar clave de acceso, nuevo paciente, ingreso al sistema con los distintos perfiles de usuario (Cuenca Luis, 2014).

ESTADO	DESCRIPCIÓN DEL TEST	RESULTADO	TIEMPO RESPUESRA
Culminado	Permitir al Usuario Administrador y al Usuario Estándar recuperar su clave de acceso al sistema.	Pasa, no se ha informado de ningún error.	8 segundos
Culminado	Permitir registrar un nuevo usuario estándar al sistema.	Pasa, no se ha informado de ningún error.	2 segundos
Culminado	Permitir ingresar al sistema como usuario administrador y	Pasa, no se ha informado de ningún	2 segundos

	usuario estándar.	error.	
--	-------------------	--------	--

4.3. Pruebas Unitarias

Este tipo de pruebas contribuyen al estudio de la técnica de desarrollo guiado por pruebas (TDD), apoyado por el framework JUnit. Este framework se usa con la finalidad de verificar, manejar y ejecutar conjuntos de pruebas automatizadas.

El Framework JUnit proporciona las clases y métodos necesarios con el objetivo de facilitar la tarea de desarrollar pruebas en el sistema y así asegurar su consistencia y funcionalidad.

A continuación se describe la prueba unitaria de administración de usuarios.

Administración de Usuarios

Test 1

```
@Test
public void testInsertar() {
    System.out.println("insertar");
    String cedula="1718384215";
    String nombre="Dario";
    String apellido="Gualoto";
    String contrasenia="darios1234";
    String telefono="23660188";
    String direccion="Ejercito";
    String correo="dario12@hotmail.com";
    String privilegio="1";
    int result=port.insertar(cedula, nombre, apellido, contrasenia,
telefono, direccion, correo, privilegio);
    assertEquals(1, result);
}
```

Ilustración 7: Test del método insertar.

```
@Test
public void testEliminar() {
    System.out.println("eliminar");
    String cedula="1718384215";
    int result=port.eliminar(cedula);
    assertEquals(1, result);
    //fail("The test case is a prototype.");
}
```

Ilustración 8: Test del método eliminar.

Resultado de la prueba

Al enviar datos correctamente al método descrito, se puede garantizar que el caso de prueba ha pasado exitosamente como se puede observar en la ilustración 9.

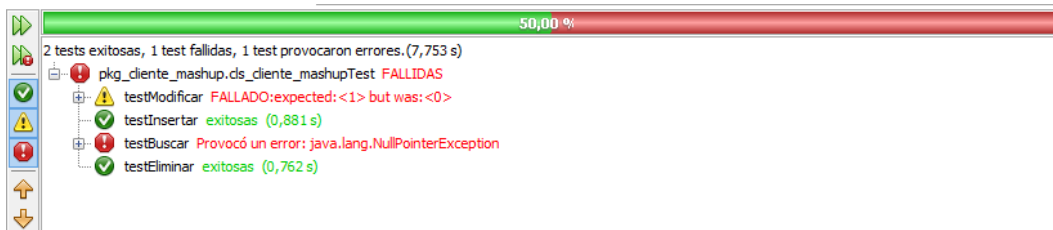


Ilustración 9: Caso de prueba exitoso.

Test 2

```
@Test
public void testModificar() {
    System.out.println("modificar");
    String cedula="1718384215";
    String nombre="Dario Santiago";
    String apellido="Gualoto Alvarez";
    String contrasenia="darios1";
    String telefono="23660183";
    String direccion="Ejercito 2";
    String correo="dario11@hotmail.com";
    String privilegio="1";
    int result=port.modificar(cedula, nombre, apellido, contrasenia,
telefono, direccion, correo, privilegio);
    assertEquals(1, result);
}
```

Ilustración 10: Test del método modificar.

Resultado de la prueba

Al ejecutar las pruebas conjuntamente el test pasa los requerimientos como se puede observar en la ilustración 11.

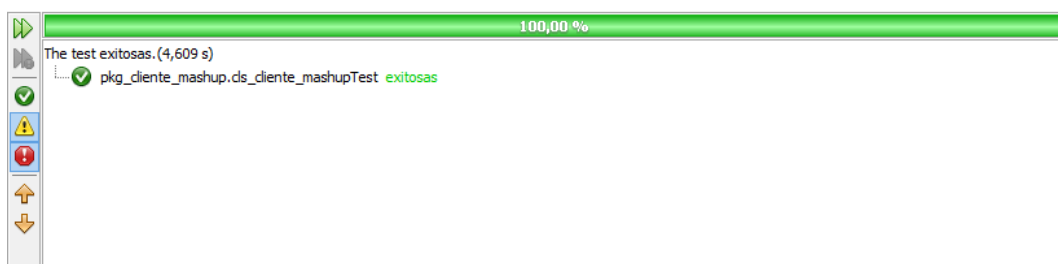


Ilustración 11: Test exitoso.

5. TRABAJOS RELACIONADOS

En el sistema de Ricachi Carranza, Klever Eduardo titulado “Sistema para el control médico y odontológico de los empleados y trabajadores del H.C.P.T.” (Ricachi Carranza, 2006), se puede constatar que si existe software para la administración de consultorios médicos, pero a diferencia del presente trabajo, este está encaminado a solventar necesidades puntuales dictadas por el usuario final. Dentro del sistema para la administración de consultorios médicos se integró varios módulos que gestionan: pacientes, historias clínicas, curvas de crecimiento, perfiles de usuario, recetas, diagnósticos, agenda electrónica para el médico y agenda electrónica para el paciente, reportes y la integración de códigos CIE 10 y VADEMECUM.

Consultorios médicos particulares utilizan sistemas web como Ecuasanitas que presta servicios como: resultados de laboratorios, servicio hospitalario, historias clínicas, pacientes, farmacéutica. Tomando en cuenta que el sistema propuesto también procesa información clínica de pacientes y analiza diagnósticos, este trabajo podría ser similar pero desconocemos el alcance técnico de cómo fue desarrollado Ecuasanitas.

6. CONCLUSIONES

El sistema de administración de consultorios médicos puede gestionar el proceso de reserva de citas médicas y la prescripción tanto de diagnósticos como de recetas a los distintos pacientes registrados en el sistema, reduciendo las actividades a los gestores encargados de las mencionadas tareas, brindando un servicio fiable y disponible.

El levantamiento de requisitos y análisis de los mismos son los procesos más complejos del proyecto y los que más tiempo toman; en base a los lineamientos de la metodología SCRUM se realiza reuniones periódicas con cada uno de los involucrados en el desarrollo del sistema, tanto programadores como los usuarios finales, para priorizar y unificar los procesos más relevantes. Los resultados que se obtienen son los esperados por el cliente logrando unificar esfuerzos y procesos de manera eficaz.

Como trabajo futuro se plantea utilizar una solución que abarque la integración del sistema sobre dispositivos móviles con tecnología Android, para lograr potenciales beneficios, facilitar y optimizar los recursos que se utiliza dentro del consultorio médico; de esta forma se dispone de la información y se realiza un seguimiento constante a los pacientes.

7. REFERENCIAS BIBLIOGRÁFICAS

- Araújo, A. (s.f.). *Test Driven Development Fortalezas y Debilidades*. Montevideo, Uruguay .
- ChileÁgil. (19/11/2010 de 11 de 2010). *Introducción al Desarrollo Ágil: Test Driven Development*. Recuperado el 04 de 04 de 2014, de <http://www.chileagil.cl/blog/2010/11/19/introduccion-al-test-driven-development/>
- Cuenca Luis, A. V. (2014). Desarrollo del sistema para la administración de consultorios médicos.
- Gómez, M. Á. (s.f.). *mifergo*. Obtenido de <http://www.mifergo.es/tag/scrum/>
- Hut, P. (16 de Junio de 2009). <http://www.pmhut.com/the-chaos-report-2009-on-it-project-failure>.
- Jurado, C. B. (2010). *Diseño Ágil con TDD*.
- Patricio Letelier, M. C. (15 de 01 de 2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Recuperado el 02 de 03 de 2014, de http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
- reserv. (2010). *reserv*. Recuperado el 12 de septiembre de 2013, de <http://www.reserv.com.ar/metodologia.php>
- Ricachi Carranza, K. E. (s.f.). <http://repo.uta.edu.ec/>. Obtenido de <http://repo.uta.edu.ec/handle/123456789/378>
- Romo, A. (17 de Marzo de 2010). <http://mundobyte-x.blogspot.com/2010/03/ultima-revision-17-de-marzo-2010.html>.
- upm. (s.f.). *entornos*. Obtenido de <http://lml.ls.fi.upm.es/ep/0607/entornos.html>