ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DEL SISTEMA DE CONTROL DE COTIZACIONES EN PLATAFORMA WEB CON HERRAMIENTAS NO PROPIETARIAS PARA LA EMPRESA "LABORATORIO ECUAAMERICAN"

PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

BARRAGÁN LASLUISA BÁRBARA XIMENA
barby_fm@hotmail.es

TOAPANTA GUALPA MARCELO IVAN
marcelo.toapanta@hotmail.com

DIRECTOR: ING. SALVADOR PESANTES EDWIN GONZALO
edwin.salvador@epn.edu.ec

CO-DIRECTOR: ING. GALLARDO CARRERA CESAR EDUARDO
cesar.gallardo@epn.edu.ec

Quito, Marzo 2017

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Barragán Lasluisa Bárbara Ximena y Toapanta Gualpa Marcelo Iván, bajo nuestra supervisión.

Ing. Edwin Salvador	Ing. Cesar Gallardo
DIRECTOR DEL PROYECTO	CO-DIRECTOR DEL PROYECTO

DECLARACIÓN

Nosotros, Barragán Lasluisa Bárbara Ximena y Toapanta Gualpa Marcelo Iván declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Barragán Lasluisa Bárbara Ximena	Toapanta Gualpa Marcelo Iván

DEDICATORIA

A Dios, por haberme permitido la dicha de vivir, recorrer este gran camino dentro de mi universidad, por permitirme convertirme en un ser profesional, gracias a cada maestro que me hizo parte de este proceso de formación y así culminar la carrera con conocimientos que perduraran durante mi vida.

A mi angelito que está por venir, por quien seguiré luchando y superándome, este es uno de tantos pasos que haré por verlo feliz y con bien.

Mis hermanas y sobrina por su apoyo incondicional, que han sido mi principal fuerza e incentivo para superarme, mil gracias.

A mi abuelito, mi papi Cesar quien cuido de nosotras haciendo de padre toda mi vida, que con sus consejos supo guiar para no rendirme y culminar con mis proyectos.

A Marcelo, por tu tiempo, paciencia y explicaciones logramos culminar este proyecto que será el primero de muchos que falta por vivir, gracias por todo y ser un gran compañero y amigo durante este tiempo

A mi madre, por haberme dado la vida, por ser padre y madre a la vez, demostrar que nada es imposible en esta vida y que siempre pase lo que pase podré contar con ella como mi ángel guardián y mi mejor amiga y así superar cualquier obstáculo que se presente.

Barragán Lasluisa Bárbara Ximena

DEDICATORIA

En primer lugar a mis padres por ser mi ejemplo a seguir, por bríndame su amor y apoyo incondicional para lograr la meta tan ansiada, terminar satisfactoriamente mi carrera profesional.

A mí amada esposa Verónica y a mi hermosa hija Linda Mishelle por su amor, fortaleza y paciencia y por ser mi razón de vivir y seguir adelante.

A mi hermano Darío por todos sus consejos a nivel personal y profesional, por ser un excelente hermano mayor.

A Bárbara por su tiempo, esfuerzo y paciencia invertidos en este proyecto, por ser la mejor compañera y amiga que tuve a lo largo de la carrera.

Finalmente a todos los programadores del mundo por ser parte de una profesión que es más que análisis y líneas de código, es una filosofía de vida.

Marcelo Iván Toapanta Gualpa

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN 1
   1.1. Objetivos 1
      Objetivo General 1
      Objetivos Específicos 1
   1.2. Alcance 2
2. METODOLOGÍA 3
   2.1. Herramientas de desarrollo 3
      Herramientas *Back-End* 4
      Herramientas *Front-End* 6
   2.2. Metodología de desarrollo *Scrum* 7
      Roles de *Scrum* 7
      Componentes de *Scrum* 9
   2.3. Requerimientos del sistema 10
      Lista de historias de usuario 12
      Roles del sistema 14
      Mensajes de alerta 15
      Formatos de fecha y totales de cotización 15
      Funcionalidad *responsive* 16
      Gráficas estadísticas 16
      Rendimiento del Sistema 17
   2.4. Diseño de modelos de datos 17
   2.5. Diseño de interfaces de usuario 18
      Inicio de sesión 18
      Registrar usuario 19
      Formulario de usuario 20
      Menú principal 20
      Panel digital 21
      Lista de empresas 21
      Lista de cotizaciones 22
      Lista de exámenes 23
      Detalle de empresa 24
      Detalle de cotización 24

Detalle de examen 25
Formulario para empresa 26
Formulario para cotización 26
Formulario para examen 27
2.6. Planificación de *Sprints* 28
3. RESULTADOS Y DISCUSIÓN 30
3.1. Ejecución de *sprints* 30
Sprint 0: Generación de base de datos y administración de contenidos 30
Sprint 1: Módulo usuarios 34
Sprint 2: Módulo empresas 40
Sprint 3: Módulo exámenes 47
Sprint 4: Módulo cotizaciones 53
Sprint 5: Módulo panel digital 59
Sprint 6: Diseño *responsive* y navegabilidad 66
Sprint 7: Mejoras 75
4. CONCLUSIONES 81
5. REFERENCIAS BIBLIOGRÁFICAS 83
6. ANEXOS 87
6.1. Historias de Usuario 87
6.2. Pruebas de carga 96
Panel Digital 97
Empresas 99
Cotizaciones 101
Módulo: Exámenes 103

ÍNDICE DE FIGURAS

Figura 1 - Arquitectura MVC (IT VDELGADO) 5

Figura 2 – *Burndown* 8

Figura 3 - *Burndown* ejemplo 9

Figura 4 – Metodología *Scrum* (SANTIMACNET'S BLOG) 10

Figura 5 – Proceso de control de cotizaciones 11

Figura 6 – Proceso de control de cotizaciones optimizado 12

Figura 7 - Estadísticas en barra 16

Figura 8 - Iniciar Sesión 19

Figura 9 - Registrar Usuario 19

Figura 10 - Formulario Usuario 20

Figura 11 - Menú Principal 20

Figura 12 - Panel Digital 21

Figura 13 - Lista de Empresas 22

Figura 14 - Lista de Cotizaciones 23

Figura 15 - Lista de Exámenes 23

Figura 16 - Detalle de Empresa 24

Figura 17 - Detalle Cotización 25

Figura 18 - Detalle Examen 25

Figura 19 – Formulario de Empresa 26

Figura 20 – Formulario para la cotización 27

Figura 21 – Formulario para el Examen 28

Figura 22 – *Burndown* inicial *sprint* 0 31

Figura 23 - *Burndown* final *sprint* 0 32

Figura 24 – Base de datos 33

Figura 25 – Módulo gestor de contenido 34

Figura 26- *Burndown* inicial *sprint* 1 36

Figura 27 – *Burndown* control *sprint* 1 36
Figura 28- *Burndown* final *sprint* 1 37
Figura 29 - Ilustración inicio de sesión 38
Figura 30 - Ilustración editar perfil 38
Figura 31 – *Burndown* inicial *sprint* 2 42
Figura 32 - *Burndown* control *sprint* 2 42
Figura 33 – *Burndown* final *sprint* 2 44
Figura 34 - Ilustración formulario para edición de empresas y contactos 45
Figura 35 – Ilustración formulario para edición de empresas y contactos 45
Figura 36 – Ilustración lista de empresas 46
Figura 37 – *Burndown* inicial *sprint* 3 48
Figura 38 – *Burndown* control *sprint* 3 49
Figura 39 – *Burndown* final *sprint* 3 49
Figura 40 – Ilustración formulario para registro de exámenes 51
Figura 41 – Ilustración formulario para editar exámenes 51
Figura 42 – *Burndown* inicial *sprint* 4 54
Figura 43 – *Burndown* control *sprint* 4 54
Figura 44 – *Burndown* final *sprint* 4 55
Figura 45 – Ilustración formulario para registro de cotizaciones 56
Figura 46 – Ilustración formulario para editar cotizaciones 57
Figura 47 – Ilustración lista de cotizaciones 57
Figura 48 – *Burndown* inicial *sprint* 5 61
Figura 49 – *Burndown* control *sprint* 5 61
Figura 50 – *Burndown* final *sprint* 5 62
Figura 51 - Ilustración subtotales desactivados 63
Figura 52 – Ilustración subtotales activados 64
Figura 53 – Ilustración cambio de mensaje 64

Figura 54 – Ilustración panel digital 64

Figura 55 – Ilustración buscar cotización 65

Figura 56 – *Burndown* inicial *sprint* 6 68

Figura 57 – *Burndown* control *sprint* 6 69

Figura 58 – Burndown final sprint 6 70

Figura 59 – Ilustración menú de navegación 71

Figura 60 – Ilustración vistas para tablets horizontal 71

Figura 61 – Ilustración vistas para tablets vertical 72

Figura 62 – Ilustración vistas para teléfonos inteligentes horizontal 73

Figura 63 – Ilustración vistas para teléfonos inteligentes vertical 73

Figura 64 – *Burndown* inicial *sprint* 7 76

Figura 65 – *Burndown* control *sprint* 7 77

Figura 66 – Burndown final sprint 7 78

Figura 67 – Ilustración bloqueo de botón 79

Figura 68 – Ilustración mensaje de confirmación 79

Figura 69 – Ilustración cabios de valores de exámenes 80

Figura 70 – Ilustración no permitir editar cotizaciones aprobadas o rechazadas 80

Figura 71 – Tiempos de carga (Panel digital) 97

Figura 72 – Detalle de petición (Panel digital) 97

Figura 73 – Vista de conexión (Panel digital) 98

Figura 74 – Recursos (Panel digital) 99

Figura 75 – Tiempos de carga (Empresas) 99

Figura 76 – Detalle de petición (Empresas) 100

Figura 77 – Vista de conexión (Empresas) 100

Figura 78 – Recursos (Empresa) 100

Figura 79 – Tiempos de carga (Cotizaciones) 101

Figura 80 – Detalle de petición (Cotizaciones) 101

Figura 81 – Vista de conexión (Cotizaciones) 102
Figura 82 – Recursos (Cotizaciones) 103
Figura 83 – Tiempos de carga (exámenes)..... 103
Figura 84 – Detalle de petición (exámenes)..... 104
Figura 85 – Vista de conexión (exámenes) 104
Figura 86 – Recursos (exámenes) 105

ÍNDICE DE TABLAS

Tabla 1 – Modelo de historia de usuario 13

Tabla 2 - *Product Backlog* 14

Tabla 3 – Modelos 17

Tabla 4 – Lista de *sprints* 29

Tabla 5 – Tarea 1 de *sprint* 0 30

Tabla 6 – Tarea 2 de *sprint* 0 31

Tabla 7 – Tarea 3 de *sprint* 0 31

Tabla 8 - Tarea 1 de *sprint* 1 34

Tabla 9 - Tarea 2 de *sprint* 1 35

Tabla 10 - Tarea 3 de *sprint* 1 35

Tabla 11 - Tarea 4 de *sprint* 1 35

Tabla 12 – Incidente 1 de *sprint* 1 37

Tabla 13 - Tarea 1 de *sprint* 2 40

Tabla 14 - Tarea 2 de *sprint* 2 40

Tabla 15 - Tarea 3 de *sprint* 2 41

Tabla 16 - Tarea 4 de *sprint* 2 41

Tabla 17 - Tarea 5 de *sprint* 2 41

Tabla 18 - Incidente 1 de *sprint* 2 43

Tabla 19 - Incidente 2 de *sprint* 2 43

Tabla 20 - Incidente 3 de sprint 2 44

Tabla 21 - Tarea 1 de sprint 3 47

Tabla 22 - Tarea 2 de sprint 3 47

Tabla 23 - Tarea 3 de sprint 3 48

Tabla 24 - Incidente 1 de *sprint* 3 50

Tabla 25 - Tarea 1 de *sprint* 4 53

Tabla 26 - Tarea 2 de *sprint* 4 53

Tabla 27 - Tarea 3 de sprint 4 54
Tabla 28 - Incidente 1 de *sprint* 4 55
Tabla 29 – Tarea 1 *sprint* 5 59
Tabla 30 – Tarea 2 de *sprint* 5 59
Tabla 31 - Tarea 3 de *sprint* 5 60
Tabla 32 - Tarea 4 de *sprint* 5 60
Tabla 33 - Tarea 5 de *sprint* 5 60
Tabla 34 - Tarea 6 de *sprint* 5 61
Tabla 35 - Incidente 1 de *sprint* 5 62
Tabla 36 - Tarea 1 de *sprint* 6 66
Tabla 37 - Tarea 2 de *sprint* 6 66
Tabla 38 - Tarea 3 de *sprint* 6 66
Tabla 39 - Tarea 4 de *sprint* 6 67
Tabla 40 - Tarea 5 de *sprint* 6 67
Tabla 41 - Tarea 6 de sprint 6 68
Tabla 42 - Incidente 1 de *sprint* 6 69
Tabla 43 - Tarea 1 de *sprint* 7 75
Tabla 44 - Tarea 2 de *sprint* 7 75
Tabla 45 - Tarea 3 de *sprint* 7 76
Tabla 46 - Tarea 4 de *sprint* 7 76
Tabla 47 - Incidente 1 de *sprint* 7 77
Tabla 48 – Historia de Usuario 1 87
Tabla 49 – Historia de Usuario 2 87
Tabla 50 – Historia de Usuario 3 88
Tabla 51 – Historia de Usuario 4 88
Tabla 52 – Historia de Usuario 5 89
Tabla 53 – Historia de Usuario 6 90

Tabla 54 – Historia de Usuario 7 90
Tabla 55 – Historia de Usuario 8 91
Tabla 56 – Historia de Usuario 9 92
Tabla 57 – Historia de Usuario 10 92
Tabla 58 – Historia de Usuario 11 93
Tabla 59– Historia de Usuario 12 94
Tabla 60 – Historia de Usuario 13 94
Tabla 61 – Historia de Usuario 14 95
Tabla 62 – Historia de Usuario 15 95
Tabla 63 – Historia de Usuario 16 96
Tabla 64 – Característica de procesamiento herokuapp (Salesforce.com) 96

RESUMEN

Este proyecto busca automatizar el proceso de control de cotizaciones de la empresa “LABORATORIO ECUAAMERICAN” mediante la integración de elementos tecnológicos al proceso manual aplicando la investigación aplicada, el mencionado proceso presenta los siguientes problemas: falta de comunicación entre los vendedores, duplicidad en las cotizaciones, no existe un control histórico de las cotizaciones, pérdida de clientes nuevos por demora en aprobación de cotizaciones, no existe análisis o estadísticas basados en cotizaciones históricas, por consiguiente la empresa pierde tiempo, recursos y dinero. Se implementó un sistema web desarrollado con herramientas no propietarias que automatizó el proceso, gracias a esto los involucrados en el proceso administran la información de forma óptima, rápida y segura. Con esto la empresa pretende aumentar la producción de las cotizaciones generadas por los vendedores ya que se redujo el tiempo de aprobación de una cotizacion de una semana a un día, además de evitar la pérdida de clientes, gracias a que los usuarios del sistema pueden dar seguimiento a las cotizaciones y ser alertados vía correo electrónico en cada etapa del proceso.

Palabras clave: Investigación aplicada, automatización, desarrollo web, *software* libre, Scrum, Ruby on Rails.

ABSTRACT

This project seeks to automate the process control of quotations for the company "LABORATORIO ECUAAMERICAN" by integrating technological elements to the manual process applying applied research, this mentioned process presents the following problems: lacks of communication between sellers, duplication in quotations, absence of historical control of quotations, loss of new customers due to delay of approvals. There is not any analysis or statistics based on historical data; therefore, the company loses time, resources and money. It has been implemented a web system developed with non-proprietary tools that automates the process, thanks to the people who are involved in the process of the information manage in an optimal, fast and secure way. With this the company intends to increase the production of the quotes generated by the sellers since the time of approval was reduced from one week to one day, in addition to avoiding the loss of customers, thanks to that the users of the system can follow up quotations and be alerted via email at each stage of the process.

Keywords: Applied research, automation, web development, non-proprietary software, Scrum, Ruby on Rails.

1. INTRODUCCIÓN

“LABORATORIO ECUAAMERICAN” es una institución privada de salud, que brinda servicios de análisis de laboratorio clínico, histopatología, imagen diagnóstica y medicina ocupacional con un enfoque integral y absoluta precisión, en pro del paciente y para el médico, la empresa ofrece sus servicios mediante cotizaciones realizadas por vendedores comerciales.

El proceso de control de cotizaciones actual provoca los siguientes conflictos: falta de comunicación entre los vendedores, duplicidad en las cotizaciones, ausencia de control histórico de las cotizaciones, pérdida de clientes nuevos por demora en aprobación de cotizaciones, no existe análisis o estadísticas basados en cotizaciones históricas.

Por tal razón la empresa solicita desarrollar un sistema de control de cotizaciones integrando nuevos elementos tecnológicos al proceso actual, el mismo será un sistema web desarrollado con herramientas no propietarias, accesible a través de internet y será alojado en servidores propios de la empresa, de tal forma, la información de las cotizaciones estará disponible de manera inmediata para todos los involucrados en el proceso.

1.1. Objetivos

Objetivo General

Desarrollar un sistema web para optimizar la gestión y análisis de cotizaciones de la empresa “LABORATORIO ECUAMERICAN”.

Objetivos Específicos

- Determinar los pasos necesarios y los involucrados en el proceso de control de cotizaciones.
- Construir un módulo para gestionar la información de las empresas que han sido visitadas por los vendedores.

- Construir un módulo que permita gestionar las cuentas de acceso de los vendedores al sistema.
- Construir un módulo para gestionar los exámenes que serán cotizados.
- Generar graficas estadísticas de las cotizaciones.
- Construir un tablero digital de acceso rápido con la información de las etapas de las cotizaciones y alertas para los involucrados en el proceso de control de cotizaciones.
- Construir un módulo de administración de contenido (Back-End) disponible solo para el administrador del sistema.
- Evaluar el sistema en un ambiente de pruebas que garantice el correcto funcionamiento en situaciones críticas.

1.2. Alcance

El sistema web será desarrollado con herramientas no propietarias, accesible mediante internet y será alojado en servidores propios de la empresa, la plataforma web permitirá que en un futuro el sistema provea de más servicios y se adapte de acuerdo a las necesidades de la empresa.

El sistema se desarrollará con interfaces amigables y fáciles de usar, y este será accesible desde un navegador de internet instalado en cualquier dispositivo tales como celulares inteligentes, tablets, laptops y computadores de escritorio.

El sistema constará de los siguientes módulos: usuarios, empresas, cotizaciones, exámenes, panel digital, los cuales tendrán un link de acceso rápido desde todas las páginas del sistema. El sistema contará con un módulo de administración de contenido el cual estará disponible solo para el administrador del sistema, el mismo permitirá editar toda la información del sistema sin restricción alguna.

El sistema será probado en los servidores de la empresa para garantizar el correcto funcionamiento, las pruebas serán realizadas por el departamento de sistemas de “LABORATORIO ECUAAMERICAN”.

2. METODOLOGÍA

El proyecto se enmarca en la investigación aplicada la cual busca la generación de conocimiento con aplicación directa a los problemas de la sociedad o el sector productivo, ocupándose del proceso de enlace entre la teoría y el producto (Lozada, 2014). Esto es precisamente lo que se ha realizado debido a que se puso en práctica los conocimientos teóricos y prácticos obtenidos a lo largo de la carrera, para generar el sistema web que busca automatizar el proceso de control de cotizaciones de la empresa "LABORATORIO ECUAAMERICAN", y solucionar los problemas descritos del proceso actual, esto generará un bienestar común en la empresa, la misma pertenece al sector productivo del país.

2.1. Herramientas de desarrollo

Una vez definido el objetivo general y los objetivos específicos del proyecto, el equipo de trabajo se preguntó ¿Cuáles son las herramientas tecnológicas disponibles en el mercado que mejor se adaptan al desarrollo del sistema?

Las herramientas de desarrollo son aquellos programas o aplicaciones que se utilizan durante el desarrollo de otro programa. Pueden ser de importancia vital o de importancia secundaria, como un IDE (*Integrated Development Environment* - Entorno de Desarrollo Integrado). (AndrewHunt, 2010)

Según "Free Software Foundation" existen dos grandes clasificaciones del *software* según su licencia o forma de distribución:

- ***Software* propietario**.- se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo con o sin modificaciones, o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido (Free Software Foundation, 1998).

- ***Software* libre**.- hace referencia a la libertad para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software (Free Software Foundation, 1998).

El sistema fue desarrollado en su totalidad con *software* libre debido a las siguientes características (Culebro Juáarez, Gómez Herrera, & Torres Sánchez, 2006):

- Bajo costo de adquisición y libre uso
- Innovación tecnológica
- Requisitos de hardware menores y durabilidad de las soluciones
- Escrutinio público
- Independencia del proveedor
- Adaptación del software

Herramientas *Back-End*

Las principales razones por las que se utilizó a Ruby como lenguaje de programación y no otros como Java o PHP, en este proyecto son: su facilidad de aprendizaje, simplicidad en la escritura de código, reutilización, rapidez en el proceso de desarrollo gracias a ser un lenguaje interpretado y no compilado, bajos costos de inversión, y años de vigencia en el mercado.

Ruby es un lenguaje de equilibrio cuidadoso. Su creador, Yukihiro “Matz” Matsumoto, mezcló partes de sus lenguajes favoritos (Perl, Smalltalk, Eiffel, Ada y Lisp) para formar un nuevo lenguaje que incorporara tanto la programación funcional como la imperativa. A menudo ha manifestado que está “tratando de hacer que Ruby sea natural, no simple”, de una forma que se asemeje a la vida real. (Ruby Lang)

Existen varias aplicaciones desarrolladas con este lenguaje, que van desde el desarrollo de sitios web simples hasta la simulación de ambientes complejos como Twitter y Soundcloud. Así mismo una amplia comunidad de desarrolladores respaldan el uso de este lenguaje. Estas comunidades se enfocan en usar las mejores prácticas de programación sin perder usabilidad.

El *framework* de desarrollo *Ruby on Rails* (RoR), es un ambiente de desarrollo web apropiado para este proyecto, debido a las siguientes características: utiliza Ruby como lenguaje de programación del lado del servidor, es de código abierto y está optimizado para ser amigable con los programadores. Este entorno de desarrollo es flexible y adaptable, integra a nuevos componentes y funciones mediante la implementación de librerías independientes conocidas como gemas. RoR será

instalado en el sistema operativo Ubuntu que es una distribución de Linux (Ruby, S., Thomas, D. and Hansson, D, 2013)

RoR implementa la arquitectura MVC que separa la lógica del negocio en: Modelos, Vistas y Controladores. La Figura 1 ilustra la arquitectura y su interacción con los clientes web.

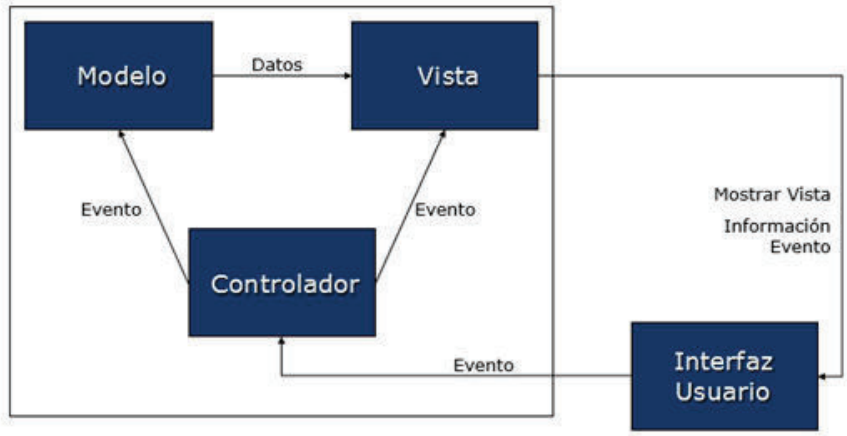


Figura 1 - Arquitectura MVC **(IT VDELGADO)**

El modelo es responsable de mantener el estado de la aplicación, algunas veces el estado es transitorio, finalizando solo con un par de interacciones con el usuario. Algunas veces el estado es permanente y será almacenado fuera de la aplicación, mayormente en una base de datos. Un modelo es más que simplemente datos, se enfoca en tocas las reglas de negocio que se aplican para dichos datos, por ejemplo: si un descuento no debería aplicar a órdenes de compra menores de 20 dólares, el modelo se enfocara en dicha restricción. (Sam Ruby, Dave Tomas, 2009, pág. 11)

La vista es responsable de generar las interfaces de usuario normalmente basado en los datos de un modelo, por ejemplo: una tienda en línea tendrá una lista de productos para ser mostrados en una pantalla de catálogo. Esta lista será accesible mediante el modelo, pero será la vista la cual acceda a dicha lista desde el modelo y de formato para mostrar al usuario, la vista en si nunca manipula la información de ingreso. (Sam Ruby, Dave Tomas, 2009, pág. 11)

Los controladores orquestan la aplicación. Los controladores reciben eventos desde el mundo exterior, interactúan con el modelo, y muestran una vista apropiada para el usuario. (Sam Ruby, Dave Tomas, 2009, pág. 12)

Para que el sistema esté disponible en internet se deberá instalar la librería Phusion Passenger y el servidor Apache. Passenger opera como un módulo el cual es transparente y permite a los programadores trabajar en ambos como un todo. (Sam Ruby, Dave Tomas, 2009, págs. 654-655)

Apache es un servidor web, que atiende y responde a las diversas peticiones de los navegadores (Riballo Arenas, 2008), proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). (Carles Mateu , 2004)

Los datos serán almacenados en PostgreSQL, el cual es un sistema de gestión de base de datos relacional que se distribuye libremente, se caracteriza por su solidez, capacidad, resistencia y facilidad de administración e implementación de estándares, lo que permite que se trabaje muy bien con grandes cantidades de datos y con una alta multitud de usuarios accediendo a la vez. (O'Reilly Media, 2014)

Herramientas *Front-End*

El objetivo general de este proyecto consiste en desarrollar un sistema web para optimizar la gestión y análisis de cotizaciones de la Empresa “LABORATORIO ECUAMERICAN”, para esto es necesario citar a HTML, el cual es un lenguaje de marcado de hipertexto y es el bloque más básico del entorno web, dicho texto describe el contenido de una página web y será interpretado por un navegador web (Duckett, HTML & CSS: Design and Build Web Sites, 2011)

CSS es el acrónimo de Hojas de estilo en cascada (*Cascading Style Sheets*) (Duckett, HTML & CSS: Design and Build Web Sites, 2011). Describe la forma que los elementos HTML serán mostrados mediante reglas de estilo, se entiende por cascada al hecho de que la última regla de estilo se impone a las anteriores.

Debido a que los usuarios del sistema pueden acceder mediante dispositivos móviles se hace necesario que las vistas se ajusten al ancho de cada pantalla de forma automática, esta funcionalidad se conoce como *responsive*. Foundation es un *framework* de CSS disponible para cualquier tipo de sistema que permite el fácil diseño de páginas web ajustables a cualquier pantalla, amigables con el usuario, el *framework* es legible, flexible y totalmente personalizable (ZURB FOUNDATION).

Para que el contenido del sistema sea dinámico e intuitivo con el usuario se usará JQuery, el cual es un *framework* de JavaScript, que permite implementar funcionalidad y efectos de forma rápida, legible y estructurada (Orós Cabello, 2012, pág. 257).

2.2. Metodología de desarrollo *Scrum*

"LABORATORIO ECUAMERICAN" solicitó que el sistema fuera desarrollado de forma rápida, eficiente, con calidad y en un periodo de tiempo relativamente corto (aproximadamente 4 meses), además tan solo muestra una visión general de los requerimientos y se corre el riesgo de que los mismos cambien durante la etapa de desarrollo.

En la ingeniería del software existen dos tendencias en las metodologías de desarrollo unas estructuradas y otras ágiles, siendo estas últimas las que mitigan los riesgos descritos previamente y han tomado gran importancia en estos últimos años por sus resultados positivos en proyectos altamente cambiantes. (Grupo ISSI, 2003)

La metodología ágil más ampliamente usada es *Scrum* que se caracteriza por la facilidad de reacción y mitigación de incidentes que podrían afectar al desarrollo del sistema y posteriormente al incumplimiento en la entrega del proyecto, esto requiere de un compromiso por parte del equipo de trabajo y del cliente (Kniberg, 2010). *Scrum* utiliza conceptos simples y comprobados para el desarrollo de productos, ciclos completos de retroalimentación cortos e iterativos y la reducción de costos en los cambios. (Palacio, Gestión de proyectos Scrum Manager, 2014). *Scrum* requiere que los involucrados en el proyecto cumplan funciones específicas, por lo cual son organizados en grupos conocidos como roles.

Roles de *Scrum*

- **Dueño del producto (*Product Owner*):** Es la persona que sabe del negocio del cliente y su perspectiva del producto.
- **Maestro Scrum *(Scrum Master)*:** Encargado de comprobar el modelo y la metodología, elimina los problemas que pueden hacer que el proceso no continúe e interactúa con el cliente y el equipo de trabajo.
- **Equipo de trabajo (*Team*):** organizan y toman decisiones para conseguir un objetivo, se involucran en la estimación del esfuerzo de las tareas.

- **Usuario:** cliente final que usará el producto (Casamayor, 2010).

De acuerdo a las recomendaciones de *Scrum*, en este proyecto los involucrados se organizaron de la siguiente forma:

- **Dueño del producto:** Gonzalo Molina (Departamento de sistemas ECUAAMERICAN)
- **Maestro Scrum:** Marcelo Toapanta
- **Equipo de trabajo:** Marcelo Toapanta (desarrollo), Bárbara Barragán (aseguramiento de calidad QA)
- **Cliente:** LABORATORIO ECUAMERICAN

Para mitigar posibles retrasos generados por el equipo de trabajo o por cambios en los requerimientos del cliente, *Scrum* brinda indicadores de resultados tales como reuniones diarias conocidos como *Standup daily metting* y el gráfico de avance conocido como *BurnDown*. (Grupo ISSI, 2003)

En las reuniones diarias se discutirá sobre los resultados obtenidos el día anterior y los problemas encontrados, se actualizará el gráfico *BurnDown* y se determinará si es posible completar el objetivo planteado, es posible que QA devuelva una tarea a la etapa de desarrollo o agregue nuevas tareas lo cual debe quedar reflejado también en el gráfico *BurnDown*. (Rendin, 2012)

BurnDown es una herramienta utilizada por el equipo de trabajo para mostrar el avance diario, detectar desviaciones y prevenir posibles retrasos.

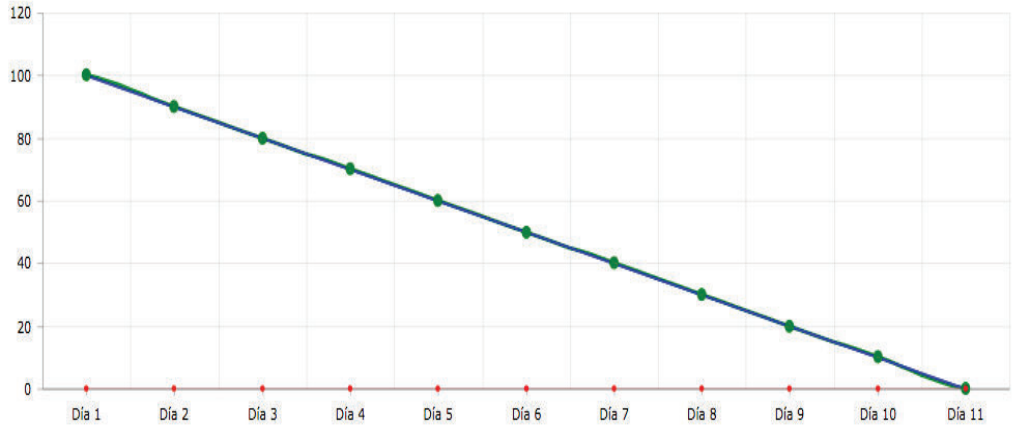


Figura 2 – *Burndown*

Como se puede observar en la Figura 2 en el eje horizontal se coloca el tiempo en días que tomará implementar las tareas del *sprint*, en el eje vertical se coloca el total de puntos del *sprint*, es decir la suma de todas las tareas que conforman el mismo.

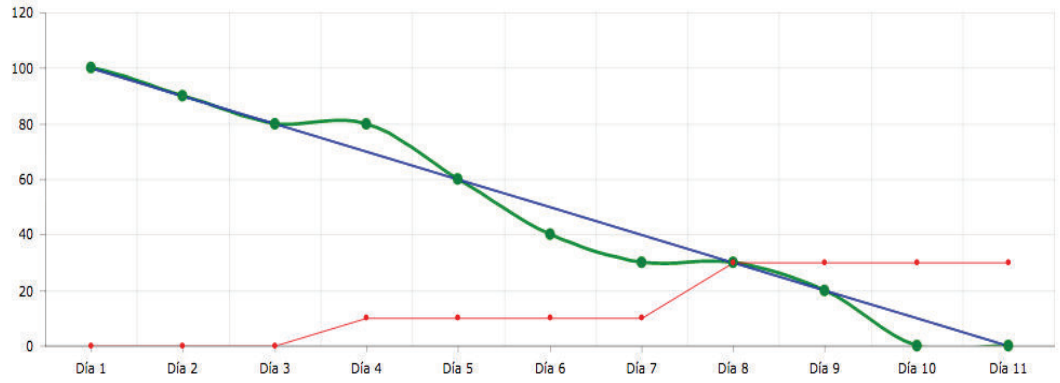


Figura 3 - *Burndown* ejemplo

En la Figura 3 se observa que la línea roja representa el trabajo realizado por el equipo de QA y la línea verde el trabajo del equipo de desarrollo, si la línea verde se ubica por encima de la línea azul, se corre el riesgo de no cumplir con a tiempo con la entrega, la línea roja, no será alterada con el avance del equipo de desarrollo, siendo que al terminar siempre mostrará el total agregado.

El equipo de trabajo está conformado por un equipo de desarrollo y un equipo de QA que trabajan de manera simultánea 4 horas al día, los puntos de trabajo tienen la siguiente relación, en un día laborable de 4 horas de trabajo se cubrirá 10 puntos de trabajo, es decir cada hora de trabajo cubrirá 2.5 puntos, con esta relación podremos calcular el valor real de los recursos humanos.

Componentes de *Scrum*

- **Pila del producto (*Product Backlog*):** indica las necesidades del cliente.
- **Iteración (*Sprint)*:** bloques de tareas a implementar, el producto final de un *sprint* es un demo parcial del producto.
- **Pila de Iteración *(Sprint Backlog)*:** indica las tareas que se realiza en un Sprint
- **Incremento:** parte desarrollada en un *sprint*, la parte terminada y operativa. (Casamayor, 2010)

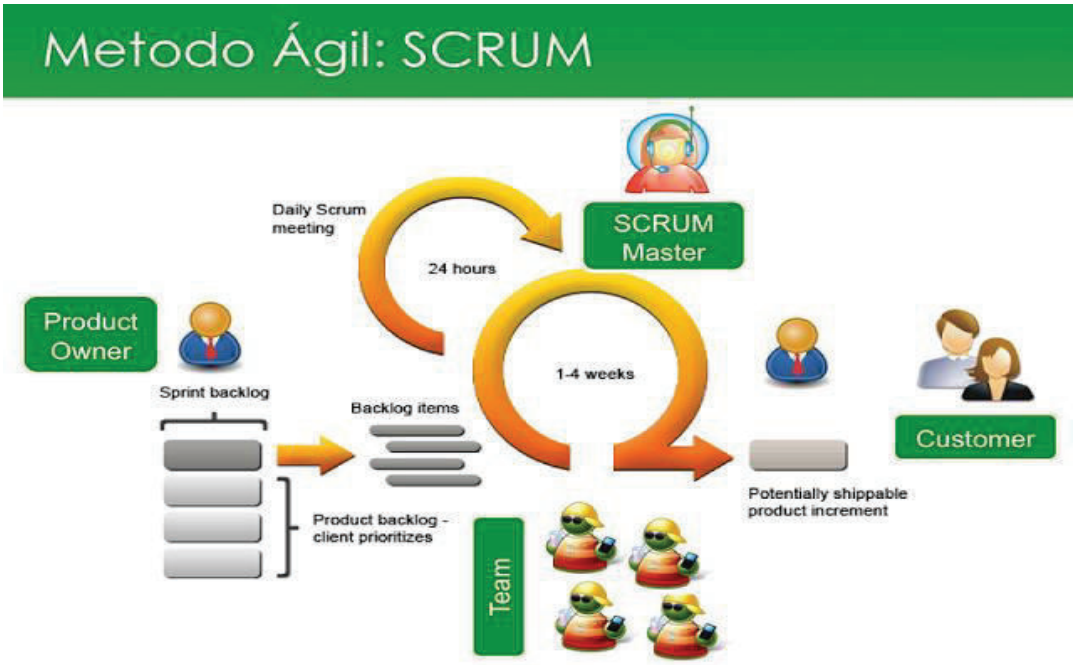


Figura 4 – Metodología *Scrum* **(SANTIMACNET'S BLOG)**

Los requerimientos del sistema conforman la línea base para el desarrollo, es vital que los mismos sean claramente definidos por el cliente y entendidos por el equipo de trabajo, esto ayuda a una óptima planificación de las iteraciones y reduce el número de incidencias reportadas por QA.

2.3. Requerimientos del sistema

Se mantuvo la reunión inicial con el cliente donde el mismo mostró el flujo del proceso para gestionar las cotizaciones y los problemas que presenta tal como se muestra en la Figura 5, el cliente indicó que el problema principal es la pérdida de clientes por la demora en la aprobación de una cotización, dicho proceso tarda en promedio una semana.

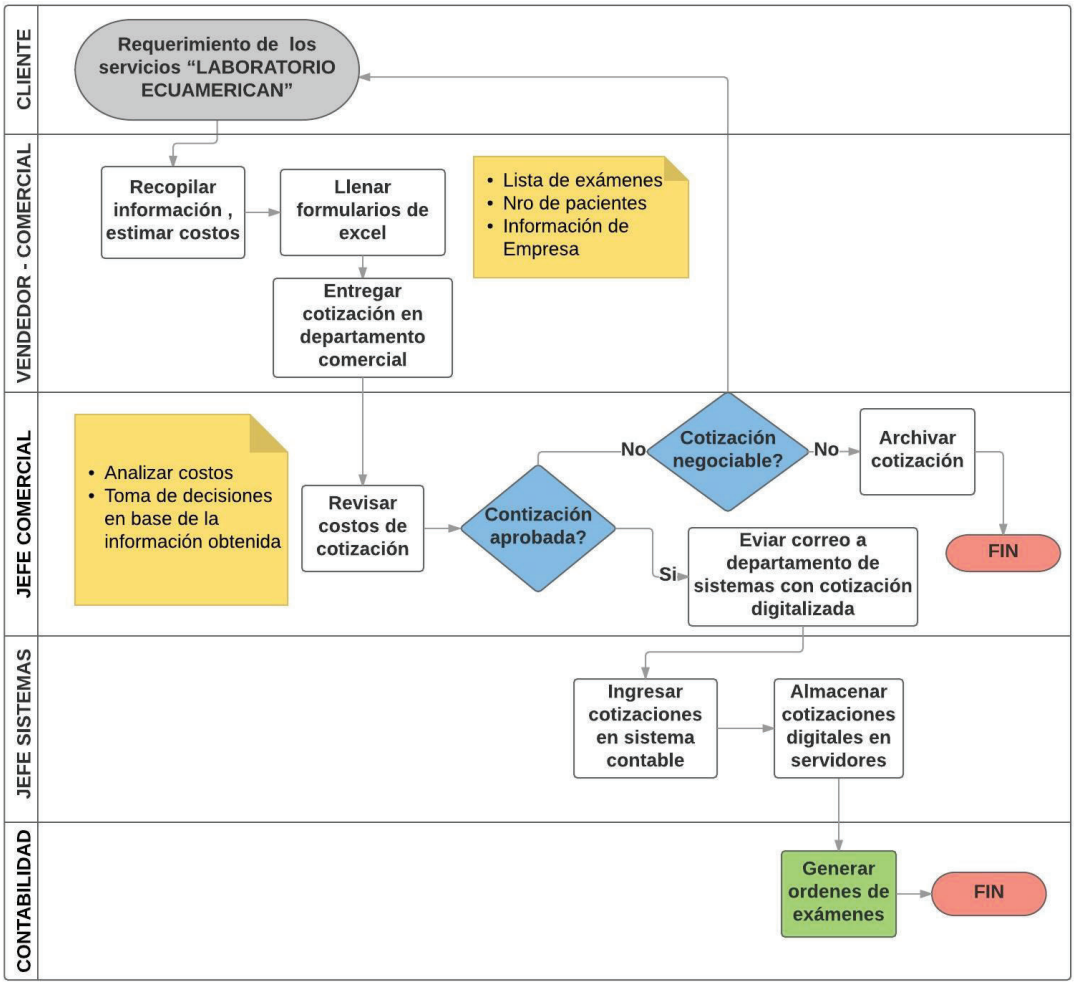


Figura 5 – Proceso de control de cotizaciones

Posteriormente el equipo de trabajo determinó que la mejor alternativa para reducir el tiempo de aprobación de una cotización es automatizar la información para que esté disponible de manera inmediata vía web para todos los participantes del proceso.

En la segunda reunión se discutió con el cliente los requerimientos del sistema de forma general, se presentó al cliente el flujo del proceso optimizado como se muestra en la Figura 6. El cliente solicitó un mecanismo que notifique inmediatamente a los usuarios del estado de sus cotizaciones, por este motivo el equipo de trabajo propone que el sistema enviará un correo electrónico, cuando una cotización sea aprobada o rechazada. Finalmente el cliente aprobó el flujo propuesto para la automatización del proceso.

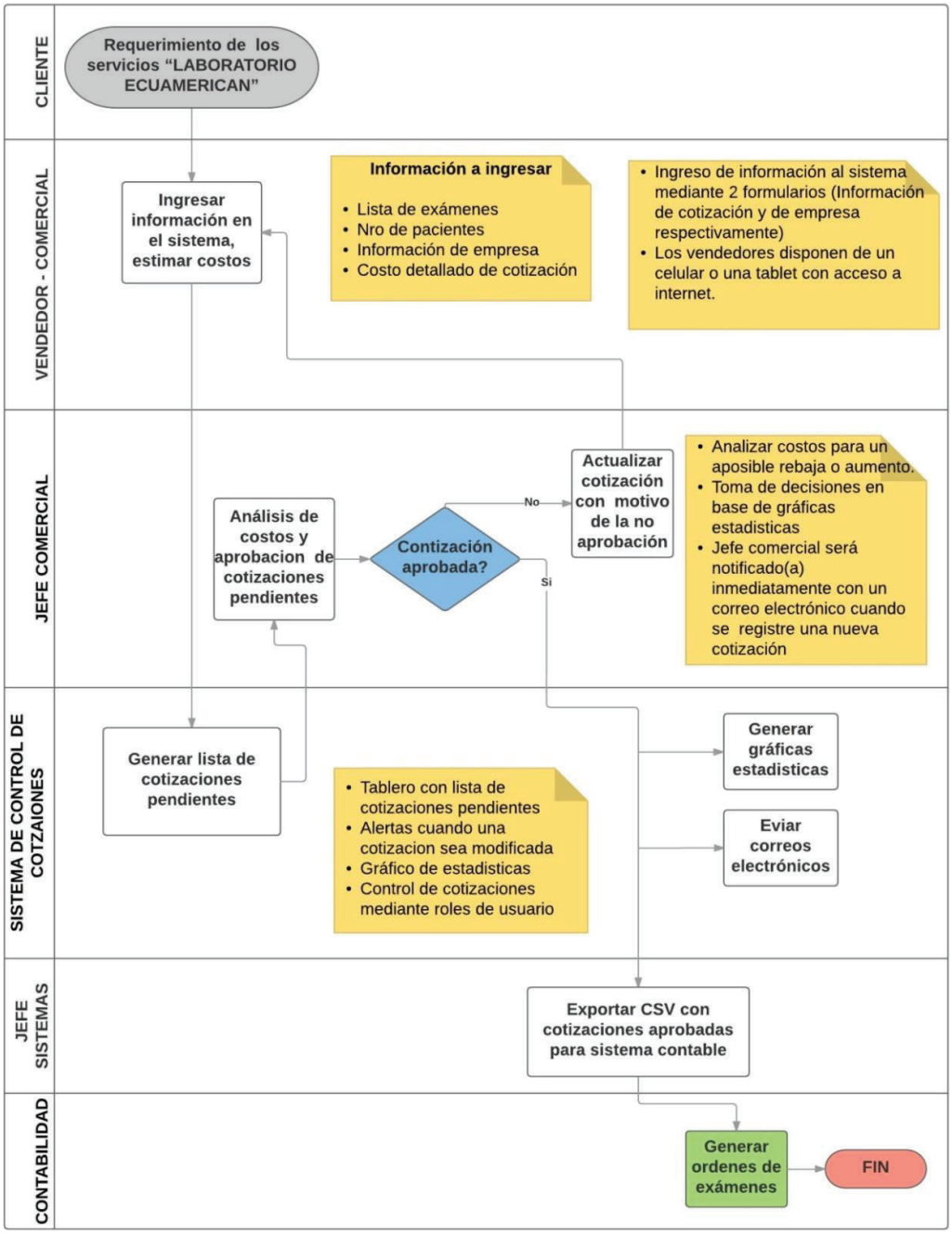


Figura 6 – Proceso de control de cotizaciones optimizado

Lista de historias de usuario

Para organizar los requerimientos del cliente, *Scrum* recomienda crear historias de usuario que contendrán los datos recopilados en las reuniones. La plantilla debe contener al menos los siguientes campos:

- **Número:** Entero identificador de la historia de usuario.
- **Tipo usuario:** Tipo o rol de usuario que interviene en la historia de usuario
- **Nombre de historia:** Nombre corto que permite identificar rápidamente a la historia de usuario.
- **Prioridad:** (alta, media, baja) nos permite identificar el grado de importancia de la historia de usuario con relación a los requerimientos del cliente.
- **Responsable:** Miembro del equipo de trabajo que se encargará de cumplir la historia de usuario.
- **Descripción:** Texto que detalla lo que debe cumplir el sistema, el mismo puede tener palabras técnicas para simplificar los párrafos.
- **Notas:** Observaciones de la historia de usuario, no es obligatorio llenar el campo sin embargo es de gran ayuda para determinar el alcance de la historia como tal. (Grupo ISSI, 2003)

En la Tabla 1 se puede observar un ejemplo de cómo se conforma la plantilla de una historia de usuario.

Tabla 1 – Modelo de historia de usuario

Historia de Usuario	
Número: #	**Tipo usuario:** Rol
Nombre Historia:	
Prioridad en el negocio: Alta, Media, Baja	**Riesgo en desarrollo:** Alta, Media, Baja
Puntos estimados: 1-5	**Iteración asignada:** Módulo
Programador responsable: Miembro del equipo	**Responsable QA:** Miembro del equipo
Descripción:	
Validación:	

Se denomina *Product Backlog* al conjunto de las historias de usuario recopiladas en las reuniones con el cliente, el mismo indica las necesidades del cliente, que el equipo de trabajo debe implementar en el sistema. (Schwaber Ken, 2011).

A continuación se lista las historias de usuario, las mismas se detallan en el anexo 6.1Historias de Usuario:

Tabla 2 - *Product Backlog*

Product Backlog		
Nro	**Historia de usuario**	**Puntos estimados**
1	Registrar empresas	60
2	Contactos de empresas	40
3	Lista de empresas	50
4	Registrar cotizaciones	50
5	Listar cotizaciones	30
6	Aprobar cotización	50
7	Inicio de sesión	70
8	Panel digital - Vendedor comercial	20
9	Panel digital - Aprobador	20
10	Panel digital - Administrador del sistema	20
11	Buscar cotización	30
12	Menú principal	20
13	Vistas responsive	100
14	Crear exámenes	30
15	Editar exámenes	30
16	Listar exámenes	20

Roles del sistema

Los involucrados en el proceso cumplen funciones específicas de acuerdo a su puesto de trabajo, dichas funciones serán agrupadas en roles que permiten al sistema identificar los niveles de acceso del usuario a los módulos, pantallas y acciones, los usuarios del sistema podrán tener uno de los siguientes roles:

- **Vendedor-Comercial**: Se encarga únicamente de generar las empresas y cotizaciones que alimentan al sistema.
- **Aprobador**: Su principal función es analizar las cotizaciones y aprobar o rechazar las mismas.
- **Administrador del sistema**: Usuario que puede gestionar toda la información del sistema sin restricción alguna.

Mensajes de alerta

El sistema deberá mostrar mensajes de alerta y notificación en los siguientes eventos:

- Inicio de sesión
- Creación, actualización y eliminación de empresas
- Creación, actualización y eliminación de cotizaciones
- Creación, actualización y eliminación de exámenes
- Usuario intente acceder a módulos a que no tiene acceso
- Lista de campos no válidos en él envió de un formulario

Los mensajes de alerta serán aquellos que representen un error por parte del usuario mientras que los mensajes de notificación serán aquellos que indique que una acción del usuario se realizó satisfactoriamente. Los mensajes tipo alerta serán de color verde, los mensajes tipo notificación serán de color rojo.

Formatos de fecha y totales de cotización

Con el fin de estandarizar el formato de las fechas mostradas en el sistema, el cliente solicitó que se utilice el siguiente formato en todos los formularios del sistema:

- Fecha: día-mes-año
- Fecha con tiempo: día-mes-año hora: minuto

Los valores totales generados en las cotizaciones representan un valor importante para la lógica del sistema, dicho montos serán tratados con el formato de moneda y una precisión de 2 dígitos como en el siguiente ejemplo:

- Total: $ 20.00

Funcionalidad *responsive*

Uno los requerimientos del sistema es que las vistas deben ajustarse a las diferentes pantallas de los dispositivos tales como: teléfonos inteligentes, *tablets*, *laptops* y computadores de escritorio. El principal problema a resolver del requerimiento mencionado es la variedad de tamaños que pueden tener las pantallas.

El diseño web *responsive* o adaptativo se denomina como una técnica de diseño web que apoya a la correcta visualización de una página web en diferentes tamaños de pantalla (Frain, 2012).

El diseño *responsive* es capaz de redimensionar y colocar los elementos de una página web de tal forma que se adapten al ancho y alto de los dispositivos, así los elementos se visualizarán correctamente para una mejor experiencia de usuario (Kadlec & Gustafson, 2012).

Gráficas estadísticas

El sistema mostrará gráficas estadísticas de barra que mostrarán la cantidad de cotizaciones en los últimos 6 meses, para obtener los datos requeridos se realiza una consulta SQL por cada estado de la cotización aprobado, pendiente y rechazado. La gráfica tiene el siguiente aspecto descrito en la Figura 7:

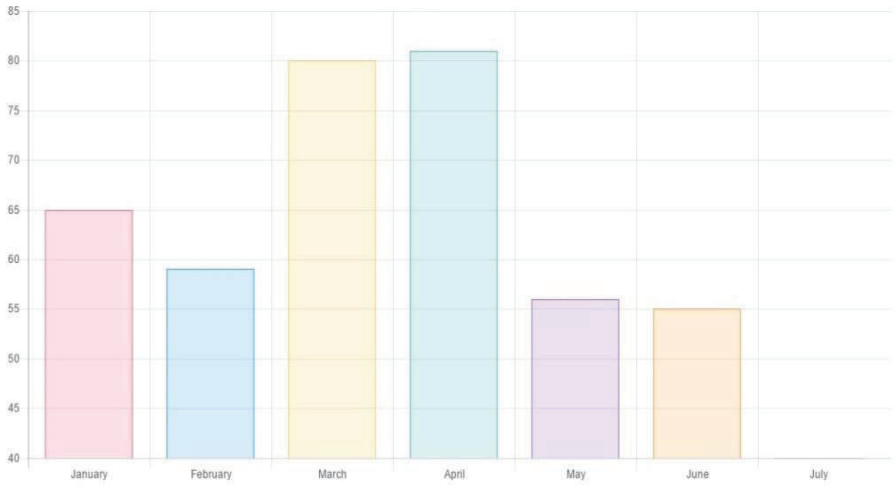


Figura 7 - Estadísticas en barra

Rendimiento del Sistema

El cliente solicitó que los tiempos de carga de cada módulo del sistema sean evaluados para garantizar el correcto funcionamiento, las pruebas serán realizadas por el equipo de trabajo en servidores de pruebas, para este fin se utilizará las plataforma web herokuapp.com. También indicó que el sistema debe responder con un tiempo máximo de 4 segundos por cada llamada y que actualmente existen 20 vendedores los mismos que utilizarán el sistema de lunes a viernes. Si el sistema no pudiera cumplir con el tiempo solicitado, el equipo de trabajo deberá analizar las posibles soluciones.

Los requerimientos mencionados anteriormente permiten al equipo de trabajo comenzar con el diseño del sistema, en el cual se generó la lista de entidades que conformarán la base de datos y prototipos de las vistas.

2.4. Diseño de modelos de datos

El modelado de datos del sistema definirá la estructura de las entidades y sus relaciones así como los campos de cada entidad y su tipo de dato.

Ruby permite generar modelos de las entidades para comunicarse con el gestor de base de datos mediante la herramienta “ActiveRecord”, esta herramienta permite al sistema acceder a la base de datos sin ejecutar código SQL. La principal ventaja es el manejo de entidades con objetos de Ruby reduciendo el uso de código SQL lo que implica cierta independencia del gestor de base de datos. (Sam Ruby, Dave Tomas, 2009, p. 17)

Debido a que RoR está escrito en idioma inglés, gestiona las relaciones de los objetos generados en singular y plural de acuerdo a este idioma, por ejemplo para usuarios cuando se habla de muchos como *has_many* y *has_and_belongs_to_many* se utiliza plural (Sam Ruby, Dave Tomas, 2009, p. 18)

Para gestionar los módulos requeridos se creó los siguientes modelos con su respectiva traducción:

Tabla 3 – Modelos

Modelos		
Nro.	Nombre	Traducción
1	Usuario	User
2	Compañía	Company
3	Cotización	Quotation
4	Item cotización	Item quotation
5	Examen	Exam

2.5. Diseño de interfaces de usuario

El diseño de prototipos muestra una clara idea de cómo se verán las pantallas del sistema, uno de los requerimientos del cliente es que los usuarios con el rol vendedor-comercial podrán ingresar cotizaciones desde dispositivos tales como teléfonos inteligentes, tabletas, laptops o computadores de escritorio, para cumplir con este requerimiento las interfaces deben ajustarse a las pantallas de dichos dispositivos.

En esta sección se definirá el tipo, tamaño y estilos de fuente, los colores que usará el sistema, imágenes que identifique a los módulos y rutas del sistema.

A continuación, se muestran los prototipos de las pantallas que fueron elaborados por el equipo de desarrollo de acuerdo a los requerimientos levantados.

Inicio de sesión

El usuario podrá iniciar sesión mediante la siguiente pantalla que consta con los siguientes campos: e-mail, clave y recodarme. También tiene los siguientes botones: iniciar sesión y registrarse, el último nos permite registrar una cuenta en el caso de no poseer una.

Figura 8 - Iniciar Sesión

Registrar usuario

Cuando un usuario no posea una cuenta, el mismo podrá registrarse en el sistema mediante la siguiente pantalla que contiene los siguientes campos: nombre, e-mail, clave y confirmar clave. Para enviar los datos existirá un botón llamado “registrarse”, adicional tendrá un botón llamado “iniciar sesión” en el caso de que el usuario tenga ya un cuenta.

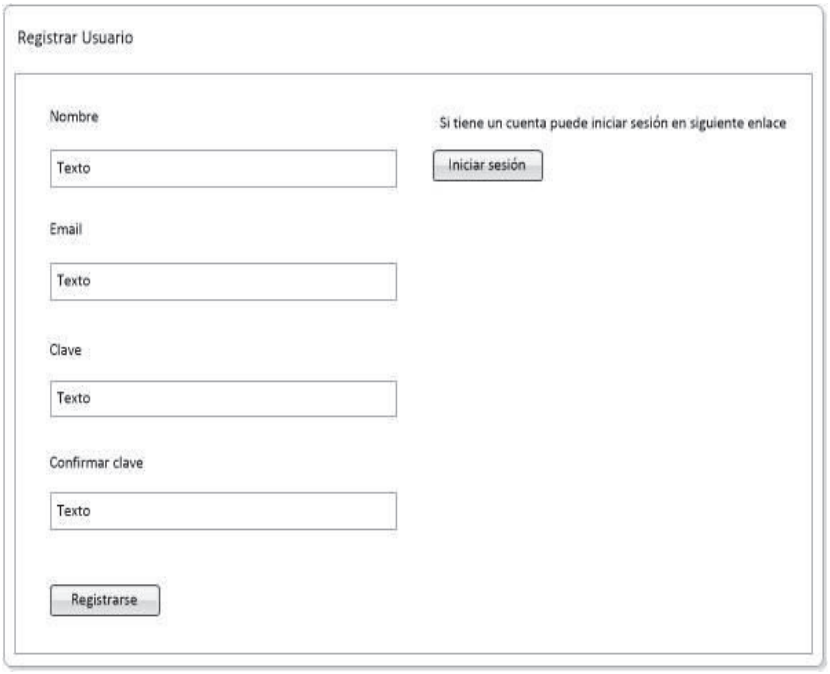


Figura 9 - Registrar Usuario

Formulario de usuario

El siguiente formulario nos permitirá editar la información de un usuario ya registrado en el sistema, para esto será necesario que los siguientes campos sean llenados: nombre, email. Si se desea cambiar la clave se debe llenar los siguientes campos: clave, confirmar clave y clave actual.

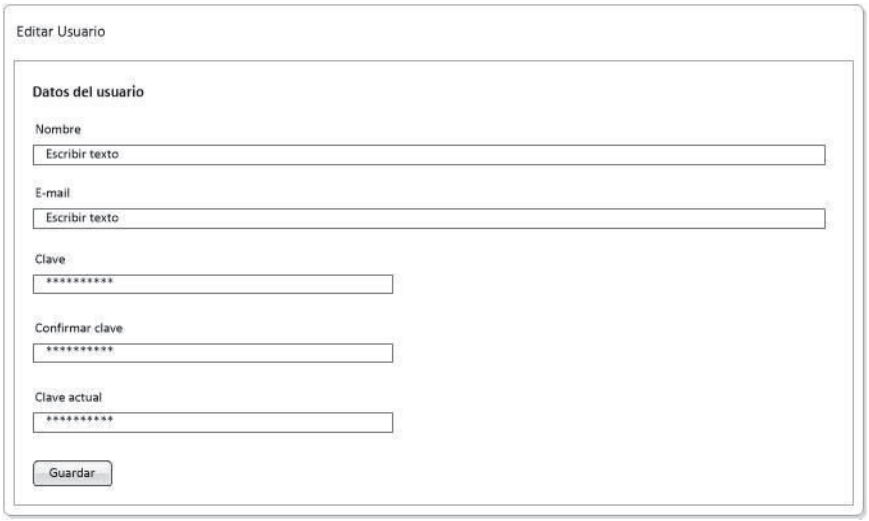


Figura 10 - Formulario Usuario

Menú principal

El menú principal estará disponible en todas las pantallas del sistema, cuando el usuario haya iniciado sesión, el mismo contiene los siguientes botones: Panel digital, Empresas, Cotizaciones, Exámenes y en el lado derecho una caja de texto que permite realizar una búsqueda en las cotizaciones registradas y finalmente en la parte inferior se ubica un título de la pantalla.

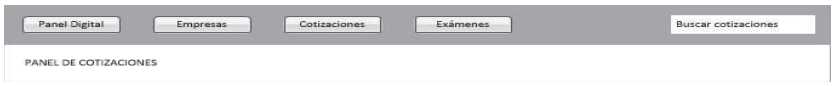


Figura 11 - Menú Principal

Panel digital

La pantalla del panel digital presentará los siguientes bloques: estadísticas, cotizaciones por aprobar y últimas empresas registradas.

Figura 12 - Panel Digital

Lista de empresas

La Figura 13 muestra la lista de empresas registradas en el sistema esta pantalla debe contener una tabla con las siguientes columnas: nombre, creado en, ruc, sector, nombre de contacto, email de contacto y acciones.

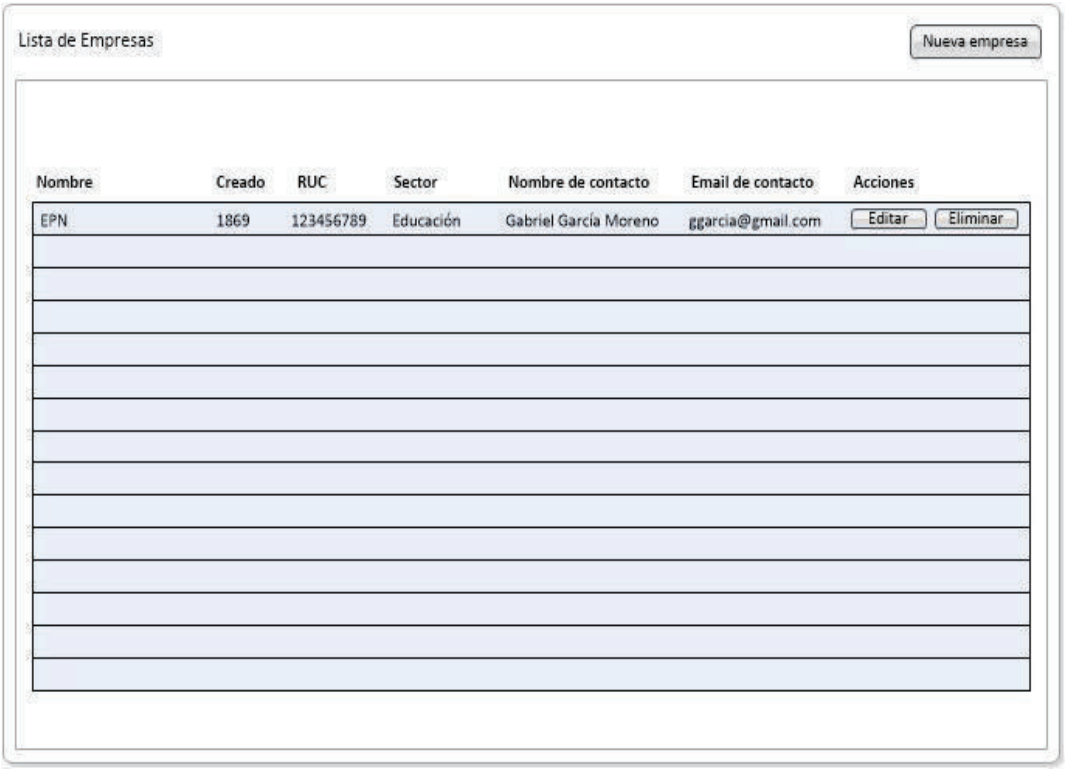


Figura 13 - Lista de Empresas

Lista de cotizaciones

La Figura 14 muestra la lista de cotizaciones registradas en el sistema esta pantalla debe contener una tabla con las siguientes columnas: descripción, creado en, total, aprobado, aprobado por, fecha de aprobación y acciones.

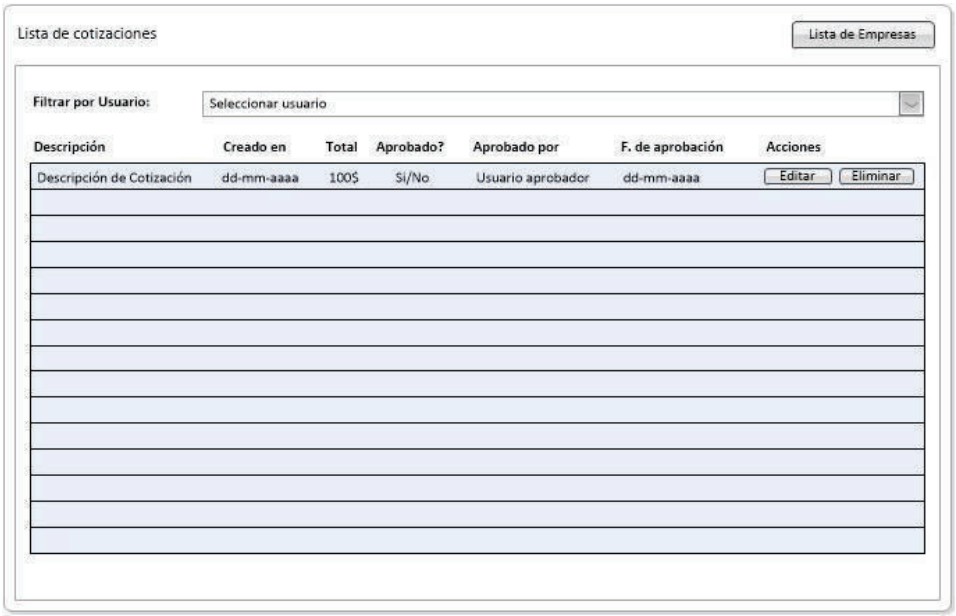


Figura 14 - Lista de Cotizaciones

Lista de exámenes

La Figura 15 muestra la lista de exámenes registrados en el sistema esta pantalla debe contener una tabla con las siguientes columnas: código, nombre, creado en, activado, valor unitario por ciudad, valor unitario por provincia, valor unitario por unidad móvil y acciones.

Figura 15 - Lista de Exámenes

Detalle de empresa

La Figura 16 muestra los detalle de una empresa, esta pantalla debe contener una lista con los siguientes datos: nombre, ruc, dirección, descripción, sector, nombre de contacto, email de contacto, nombre de doctor, email doctor, nombre responsable de facturación y email responsable de facturación.

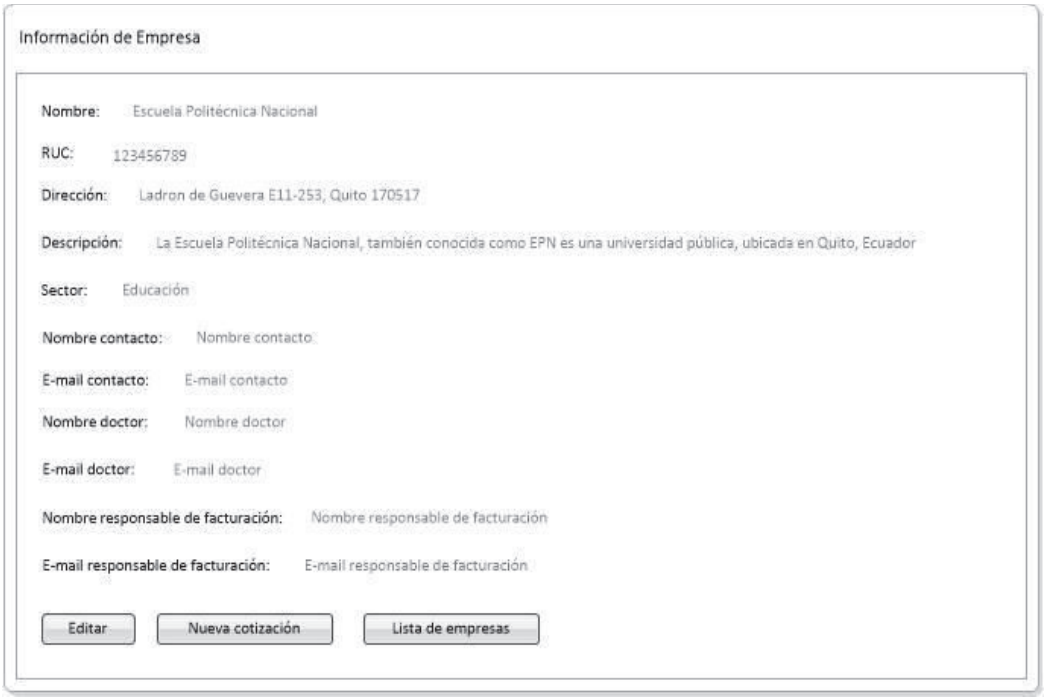


Figura 16 - Detalle de Empresa

Detalle de cotización

La Figura 17 muestra los detalles de una cotización, esta pantalla debe contener una lista con los siguientes datos: descripción, fecha de creación, aprobado, aprobado por, fecha de aprobación, lista de exámenes.

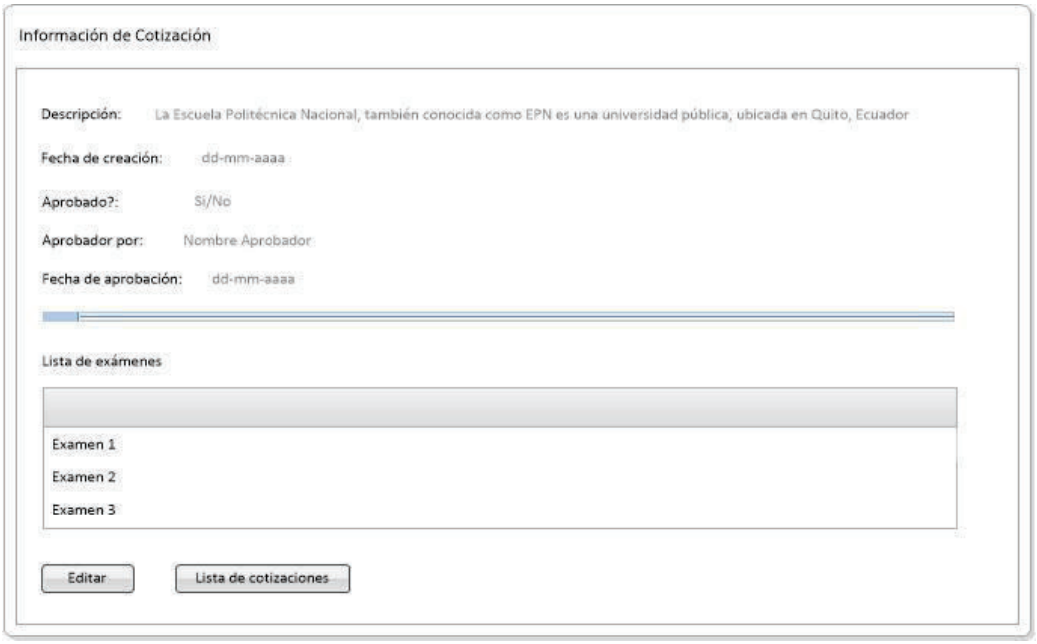


Figura 17 - Detalle Cotización

Detalle de examen

La Figura 18 muestra los detalles de un examen, esta pantalla debe contener una lista con los siguientes datos: nombre, ruc, dirección, descripción, sector, nombre de contacto, email de contacto, nombre de doctor, email doctor, nombre responsable de facturación y email responsable de facturación.

Figura 18 - Detalle Examen

Formulario para empresa

La Figura 19 muestra el formulario para crear o editar empresas debe contar de dos bloques, el primero contendrá información de empresa y el segundo información de contacto. Los datos del formulario serán: nombre de empresa, ruc, dirección, descripción, sector, nombre de contacto, email de contacto, nombre de doctor, email de doctor, nombre de responsable de facturación y email de responsable de facturación.

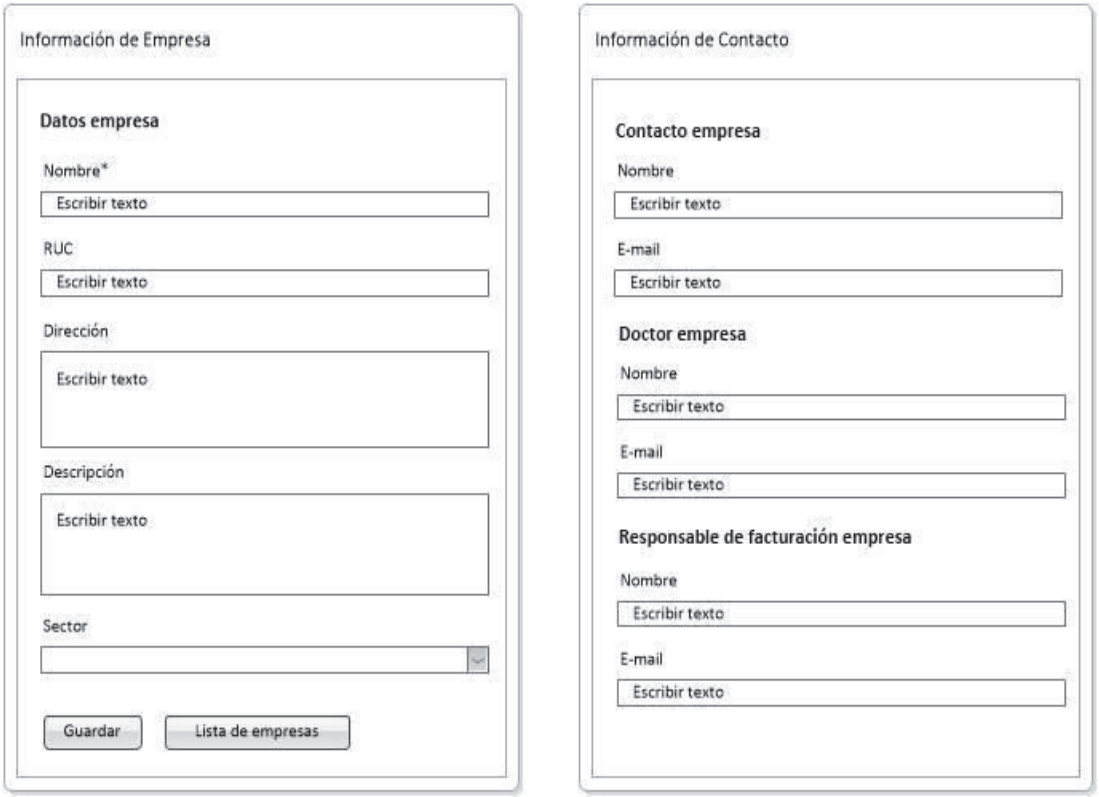


Figura 19 – Formulario de Empresa

Formulario para cotización

La Figura 20 muestra el formulario para crear o editar cotizaciones debe contar con tres bloques, el primero contendrá datos de la cotización, el segundo datos de la empresa el tercero un resumen de los totales de la cotización.

Los datos de la cotización serán: descripción, PDF adjunto, vigencia, total usuarios, aprobado y exámenes de laboratorio.

Los datos de la empresa serán: nombre, empresa nueva, nombre de contacto, email de contacto, sector y dirección.

Los datos del resumen de totales serán: subtotal por ciudad, subtotal por provincia, subtotal unidad móvil y total final.

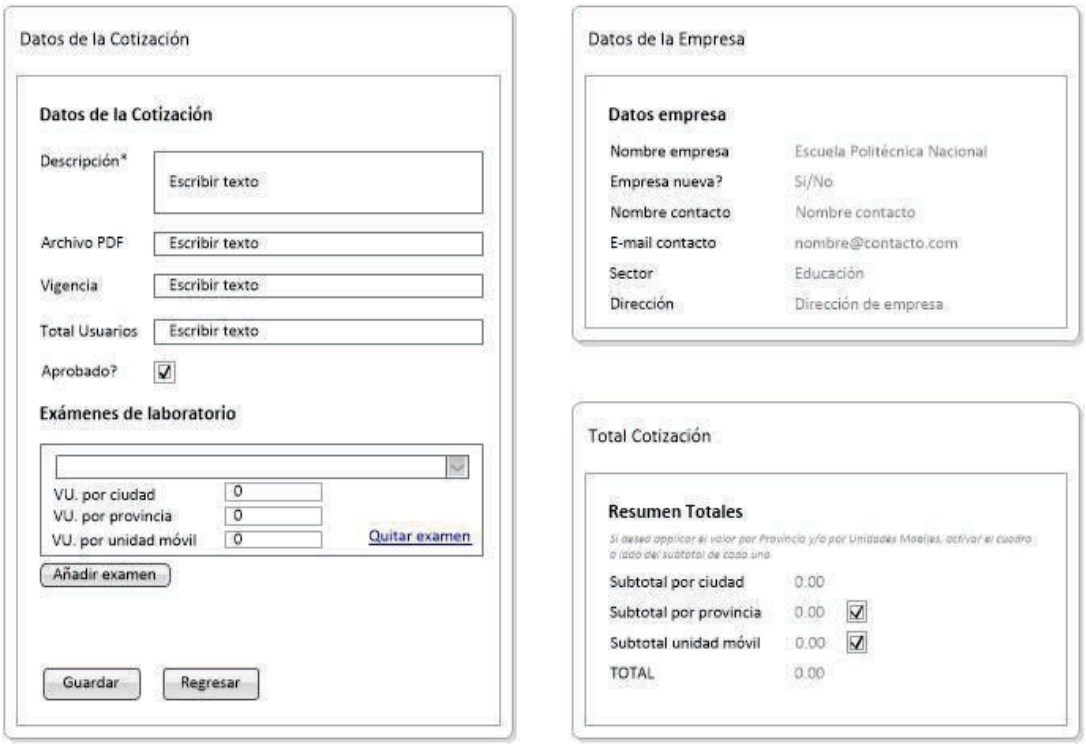


Figura 20 – Formulario para la cotización

Formulario para examen

La Figura 21 muestra el formulario para crear o editar, debe contener los siguientes campos: código, nombre, valor unitario por ciudad, valor unitario por provincia, valor unitario por unidades móviles.

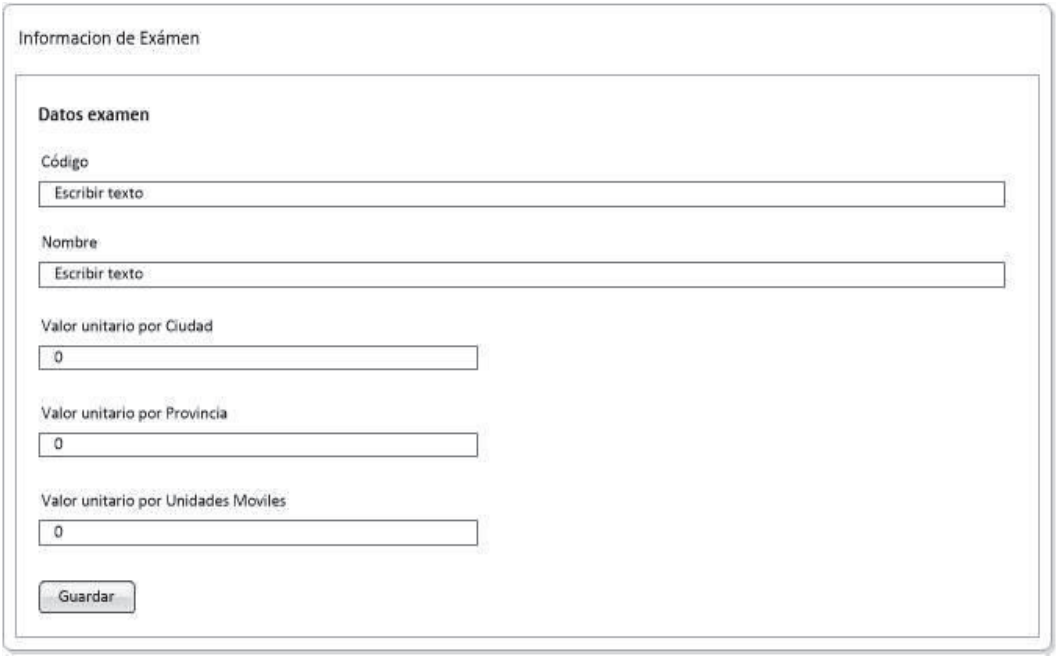


Figura 21 – Formulario para el Examen

En base de la información obtenida se procede a la planificación de *sprints*, este trabajo es necesario para generar una vista global de proceso de desarrollo de todo el sistema (Palacio, Gestión de proyectos Scrum Manager, 2014).

En *Scrum* un sprint es un bloque de trabajo temporal corto y tiene objetivos claros que se demostrarán al final de cada periodo (Palacio, Gestión de proyectos Scrum Manager, 2014).

2.6. Planificación de *Sprints*

Si bien en este punto se van a definir y planificar todos los *sprints*, se deben revisar al inicio de cada uno, de esta forma se puede mantener control sobre los cambios solicitados por el cliente, así como de los posibles retrasos que se puedan presentar.

A continuación se lista las historias de usuario, las mismas se detallan en el anexo Historias de Usuario:

Tabla 2 se agruparon en módulos de esta forma se facilitó la presentación del demo al cliente, cada módulo será implementado en un sprint, la lista de sprints se muestra a continuación en la Tabla 4.

Tabla 4 – Lista de *sprints*

Lista de *sprints*			
Nro	**Nombre *sprint***	**Historias de usuario**	**Duración días**
0	Generación de base de datos y administración de contenidos	---	4
1	Módulo usuarios	7.Inicio de sesión	10
2	Módulo empresas	1.Registrar empresas, 2.Contactos de empresas	15
3	Módulo exámenes	14. Crear exámenes, 15.Editar exámenes, 16.Listar exámenes	10
4	Módulo cotizaciones	4.Registrar cotizaciones, 5.Listar cotizaciones, 6.Aprobar cotización	10
5	Módulo panel digital	8.Panel digital - Vendedor comercial, 9.Panel digital - Aprobador, 10.Panel digital - Administrador del sistema, 11.Buscar cotización	9
6	Diseño responsive y navegabilidad	12.Menú principal, 13.Vistas responsive	12

Cada *sprint* fue cubierto de forma simultánea, cada historia de usuario fue descompuesta en tareas, de tal forma que se pudo controlar el avance de cada *sprint* o en su defecto el retraso, cuando una tarea no se completa es agregada al siguiente ciclo o sprint, el cliente fue informado de cada retraso y los cambios generados en la planificación inicial.

3. RESULTADOS Y DISCUSIÓN

3.1. Ejecución de *sprints*

Sprint 0: Generación de base de datos y administración de contenidos

Objetivo: Generar base de datos funcional
Fecha: 12/07/2016
Historias a implementar: N/A
Tiempo de ejecución: 4 días laborables
Puntos totales: 40

La gema *ActiveRecord* de RoR permite generar tablas, campos, relaciones y restricciones mediante migraciones, a partir de los modelos escritos en archivos de texto plano, es posible generar una base de datos sin necesidad de crear *scripts* SQL y ejecutarlos directamente. Por otro lado la gema *Rails Admin* permite generar automáticamente las pantallas para crear, listar, actualizar y eliminar los registros de todos los modelos.

Tabla 5 – Tarea 1 de *sprint* 0

Tarea: Generar modelos en RoR		
Número: 1	Historia: N/A	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Crear los siguientes modelos • Usuario • Examen • Empresa • Cotización • Item Cotización		

Tabla 6 – Tarea 2 de *sprint* 0

Tarea: Generar base de datos		
Número: 1	Historia: N/A	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Cargar modelos en la base de datos.		

Tabla 7 – Tarea 3 de *sprint* 0

Tarea: Generar pantallas para gestión de contenido		
Número: 3	Historia: N/A	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Este módulo permite crear, listar, actualizar y eliminar registros de cualquier modelo sin ninguna restricción, el mismo es accesible únicamente para el administrador del sistema.		

Burndown inicial

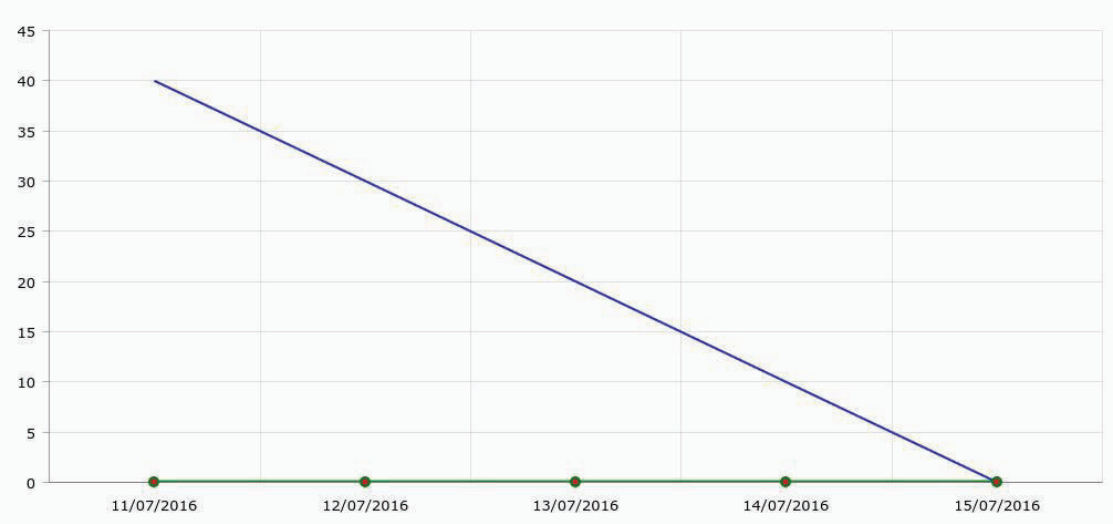


Figura 22 – *Burndown* inicial *sprint* 0

Control a los 4 días

Fecha: 15/07/2016

Puntos agregados QA: 0

Puntos avanzados: 40

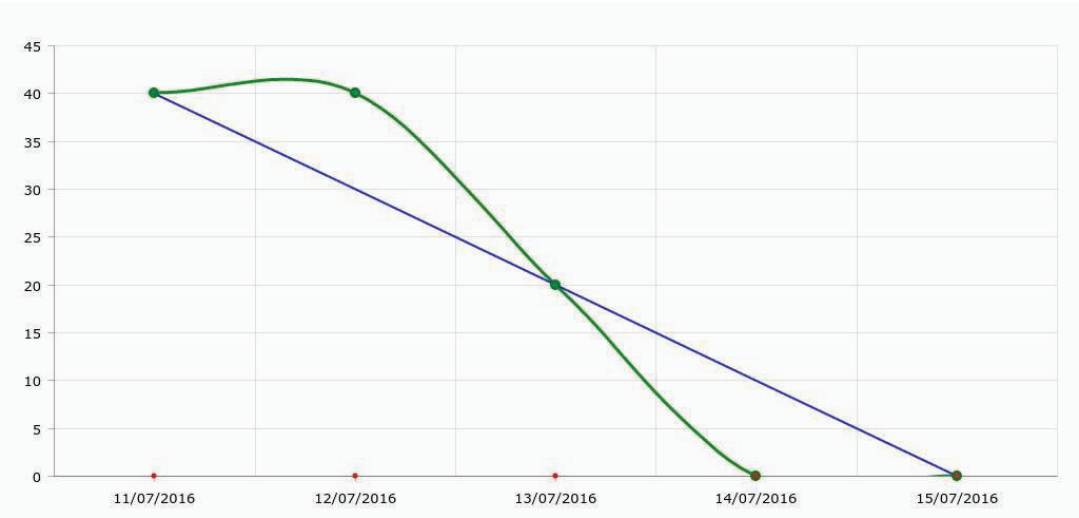


Figura 23 - *Burndown* final *sprint* 0

En la Figura 23 se puede observar el avance de todo el sprint, nótese que al inicio aparentemente no hubieron avances, esto es debido a que la lista de modelos, sus campos, relaciones y validaciones fue extensa y solo se pudo dar como terminada la primera tarea al tercer día.

ActiveRecord, se encargó de ejecutar los *scripts* necesarios en el gestor de base de datos, para crear las tablas, atributos y relaciones.

Debido a que la mayor parte del administrador de contenidos es generado mediante la gema *Rails Admin*, la tarea de gestión de contenido fue implementada sin mayores inconvenientes.

Ilustración diagramas de base de datos

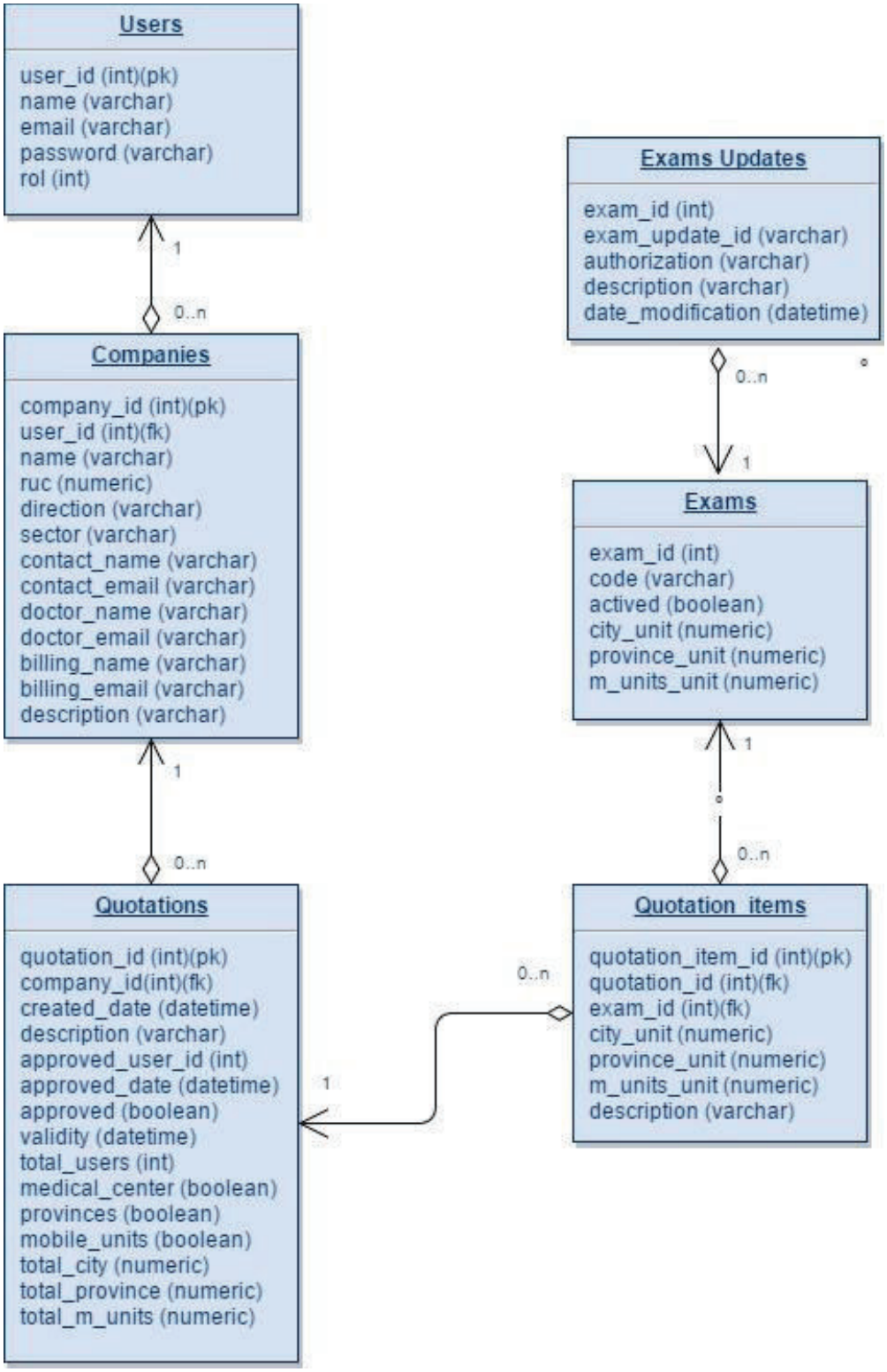


Figura 24 – Base de datos

Módulo gestor de contenido

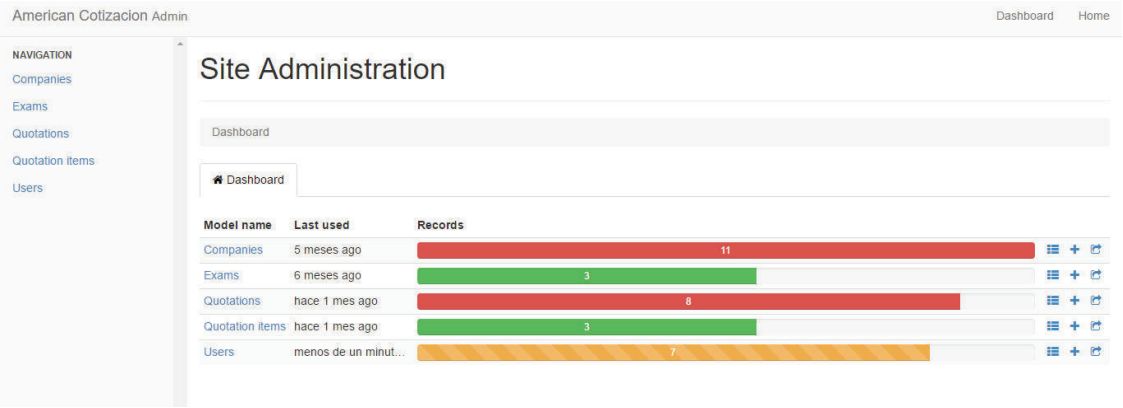


Figura 25 – Módulo gestor de contenido

Sprint 1: Módulo usuarios

Objetivo: Implementar gema "devise" para gestión de usuarios.

Fecha: 18/07/2016

Historias a implementar: #7

Tiempo de ejecución: 10 días laborables

Puntos totales: 70

El equipo de desarrollo listó las tareas a ejecutar para implementar esta historia de usuario.

Tabla 8 - Tarea 1 de *sprint* 1

Tarea: Implementar pantalla "inicio de sesión"		
Número: 1	Historia: 7	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Esta pantalla permite el ingreso al sistema del usuario.		

Tabla 9 - Tarea 2 de *sprint* 1

Tarea: Crear Administrador de sistema		
Número: 2	Historia: 7	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Se debe crear un método para crear al Administrador de sistema de forma automática. El método se ejecutará cuando se acceda al sitio por primera vez y no exista un administrador previamente creado.		

Tabla 10 - Tarea 3 de *sprint* 1

Tarea: Crear pantalla “perfil de usuario”		
Número: 3	Historia: 7	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Los campos requeridos en este formulario son; email, en éste mismo formulario podrá cambiar la clave. Para registrar una nueva clave debe ingresar la clave actual.		

Tabla 11 - Tarea 4 de *sprint* 1

Tarea: Control de acceso de usuarios		
Número: 4	Historia: 7	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Un usuario tiene control únicamente de los registros que crea y no puede administrar la		

información de otros usuarios.
Si el usuario trata de ingresar al perfil de otro cambiando el valor del ID en la URL, deberá ser redirigido su perfil personal.

Burndown inicial

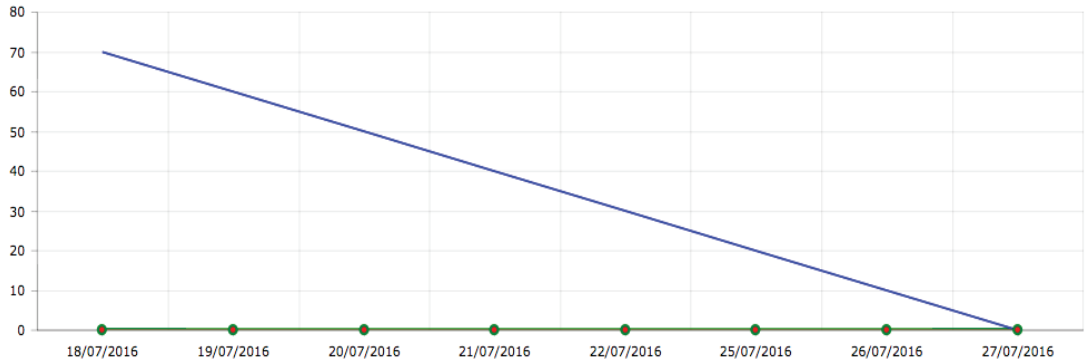


Figura 26- *Burndown* inicial *sprint* 1

Control a los 4 días

Fecha: 22-07-2016

Puntos agregados QA: 10

Puntos avanzados: 50

En éste momento es importante revisar el avance del equipo y determinar si puede cumplir con lo planificado y las acciones a tomar para evitar el un posible atraso.

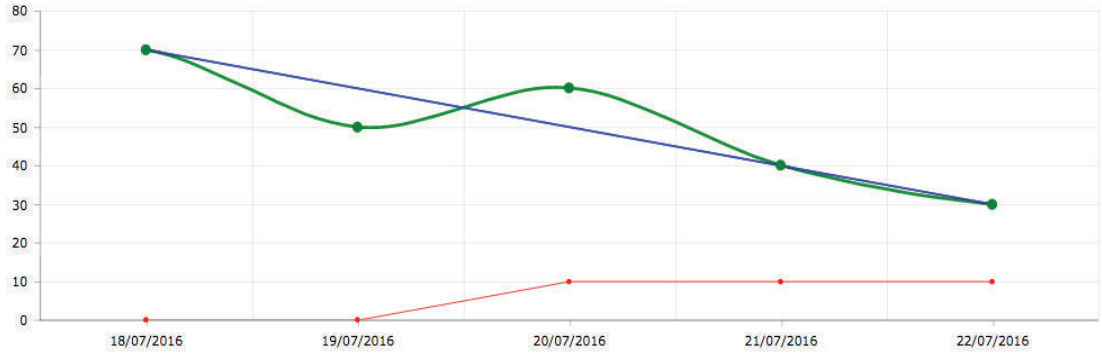


Figura 27 – *Burndown* control *sprint* 1

En la Figura 27 se puede observar que en el primer día el equipo de desarrollo avanzó 20 puntos cuando el objetivo eran 10, para el segundo día el equipo de QA reportó la

incidencia detallada en la Tabla 12, agregando 10 puntos más al trabajo que debe completar el equipo.

Tabla 12 – Incidente 1 de *sprint* 1

Incidente: Campos requeridos		
Número: 1	Historia: 7	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El formulario no muestra que campos son obligatorios, se ha definido colocar asteriscos de color rojo en las etiquetas de los campos que sean requeridos.		

Hacia el tercer día el equipo de desarrollo logró solucionar el incidente 1, agregando los asteriscos de color rojo requeridos, esto se implementa como una regla de estilo que será reutilizado en todo el sistema. En el cuarto día se cumplen satisfactoriamente los puntos planificados para el mismo, lo que indica que se va a cumplir con la fecha de entrega del *sprint*.

Control a los 7 días

Fecha: 27-07-2016

Puntos agregados QA: 10

Puntos avanzados: 80

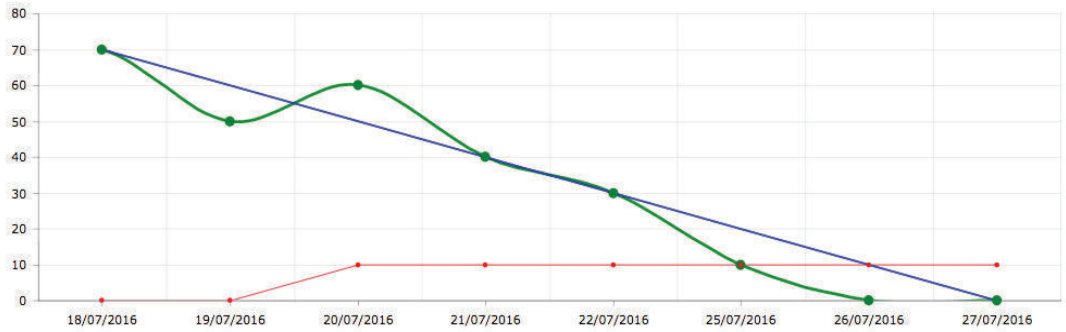


Figura 28- *Burndown* final *sprint* 1

Las Figura 28 muestra que en el día quinto se logró avanzar con 20 puntos de los 10 planificados, esto significa un adelanto en el trabajo. En el día sexto se logró completar satisfactoriamente con todos los puntos del *sprint*.

Demo con el cliente

Fecha: 27/07/2016

Al final del sprint el equipo de trabajo se reunió con el cliente para presentar el demo correspondiente al módulo de usuarios, el mismo mostró su satisfacción con la entrega del parcial. A continuación se muestran ilustraciones de las pantallas implementadas en este sprint.

Inicio de sesión

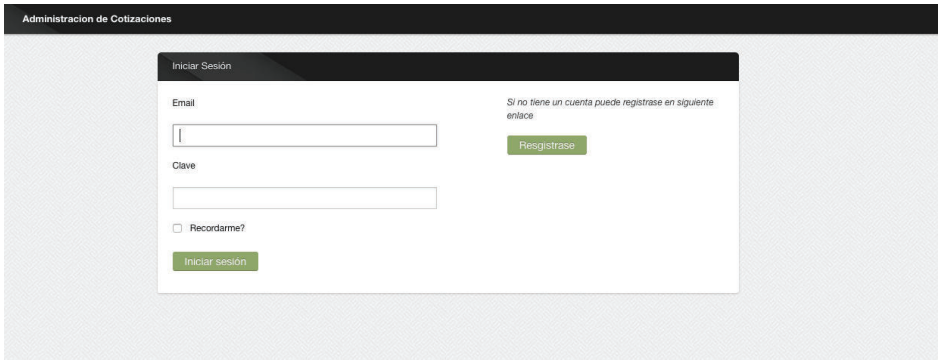


Figura 29 - Ilustración inicio de sesión

Editar perfil

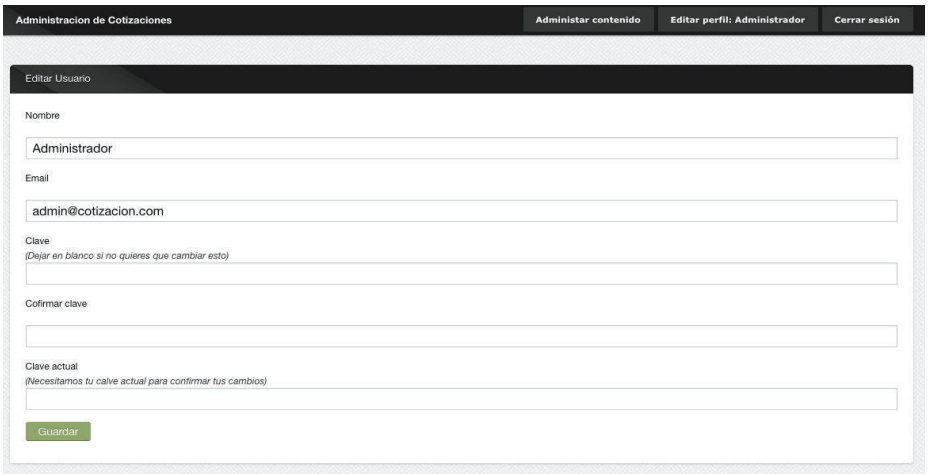


Figura 30 - Ilustración editar perfil

Retroalimentación:

Se presentó un riesgo para el cumplimiento generando un posible atraso en la entrega, se puede observar en la Figura 27 que en los días 3 y 4 la línea verde está sobre la línea azul, sin embargo, el equipo de desarrollo supo reaccionar a tiempo para mitigar dicho atraso.

Por su parte el cliente pudo probar las pantallas de inicio de sesión, accedió con la cuenta de Administrador y pudo editar su perfil. Además pudo acceder al gestor de contenidos y se le explicó cómo crear nuevos usuarios. El cliente mostró su inquietud por probar estas funcionalidades en dispositivos móviles pero se le supo indicar que es parte de un futuro sprint.

Sprint 2: Módulo empresas

Objetivo: Implementar el módulo empresas.
Fecha: 29/07/2016
Historias a implementar: #1, #2, #3
Tiempo de ejecución: 15 días laborables
Puntos totales: 150

En la siguiente lista se muestran las tareas fueron ejecutadas para implementar las historias de usuario de este *sprint*.

Tabla 13 - Tarea 1 de *sprint* 2

Tarea: Implementar formulario para registro empresas		
Número: 1	Historia: 1	Puntos: 30
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El formulario contendrá los siguientes campos: nombre, ruc, dirección, descripción y sector. Sector puede tomar los siguientes valores: educación, financiero, industrial y ninguno.		

Tabla 14 - Tarea 2 de *sprint* 2

Tarea: Implementar formulario para edición de empresas		
Número: 2	Historia: 1	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El formulario para la edición es el mismo para registrar una empresa con la diferencia que los campos serán cargados previamente.		

Tabla 15 - Tarea 3 de *sprint* 2

Tarea: Implementar método para evitar duplicidad de RUC y nombre		
Número: 3	Historia: 1	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones:		

Tabla 16 - Tarea 4 de *sprint* 2

Tarea: Implementar formulario para Contactos de empresa		
Número: 4	Historia: 2	Puntos: 40
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Los correos electrónicos deben ser válidos.		

Tabla 17 - Tarea 5 de *sprint* 2

Tarea: Implementar pantalla “lista de empresas”		
Número: 5	Historia: 3	Puntos: 50
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Las empresas serán ordenadas por fecha de creación en forma ascendente.		

Burndown inicial

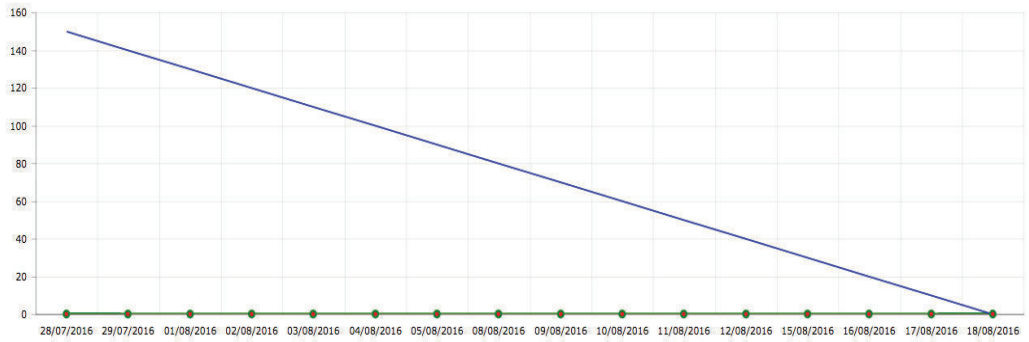


Figura 31 – *Burndown* inicial *sprint* 2

Control a los 8 días

Fecha: 11/08/2016

Puntos agregados QA: 20

Puntos avanzados: 90

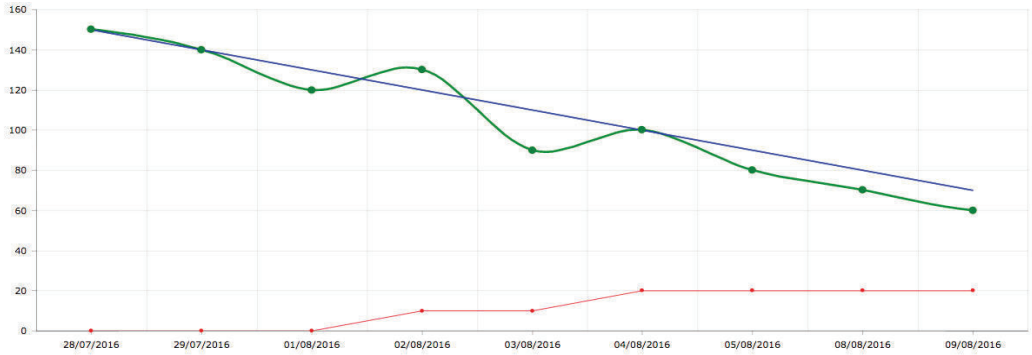


Figura 32 - *Burndown* control *sprint* 2

La Figura 32 muestra que en el día 3 el equipo de QA reportó una incidencia estimada con 10 puntos, detalla en la Tabla 18, la misma que se solucionó agregando el código necesario para mostrar la lista de errores en cada formulario, esta porción de código será replicada en cada formulario en el futuro para evitar repetir el incidente.

Tabla 18 - Incidente 1 de *sprint* 2

Incidente: Formulario no muestra mensajes de errores en el envío		
Número: 1	Historia: 2	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El formulario debe mostrar una lista de errores cuando el usuario no llene los campos obligatorios o los datos ingresados no cumplan con los formatos establecidos.		

En el día 6 el equipo QA reportó el incidente de la Tabla 19 valorado en 10 puntos, el mismo se solucionó agregando un mensaje de error en el controlador empresa para que se liste con los otros posibles errores. Gracias al progreso en días previos, el último incidente no representa un riesgo para la entrega a tiempo del demo.

Tabla 19 - Incidente 2 de *sprint* 2

Incidente: Mostrar mensaje cuando se ingresó RUC o nombre existente		
Número: 5	Historia: 1	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El método para evitar duplicidad funciona, sin embargo el sistema debe mostrar un mensaje que se encontró una empresa ya registrada para no generar confusiones en el usuario.		

Control a los 15 días

Fecha: 18/08/2016

Puntos agregados QA: 30

Puntos avanzados: 180

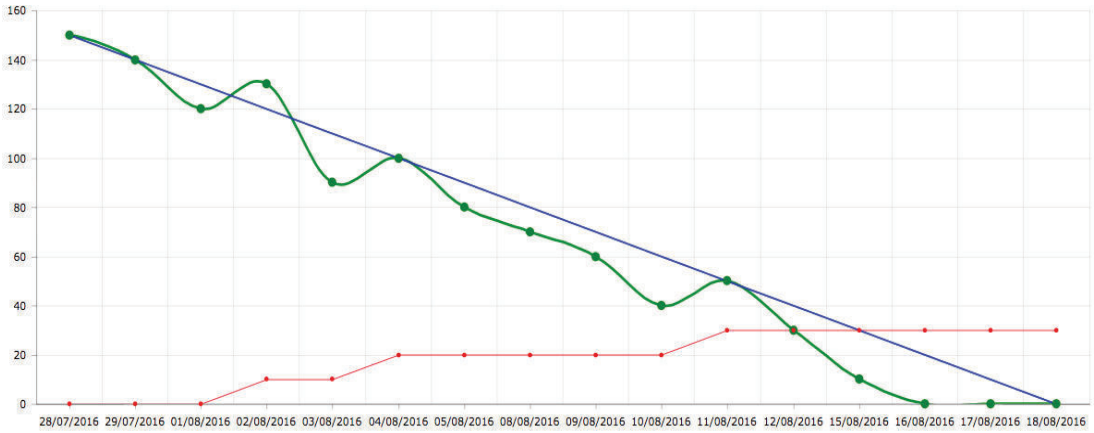


Figura 33 – *Burndown* final *sprint* 2

En el día 10 se reporta el incidente detallado en la Tabla 20, este fue solucionado agregando reglas de estilo de tal forma que sean globales para todas las paginaciones del resto del sistema.

Tabla 20 - Incidente 3 de sprint 2

Incidente: Estilos para paginación		
Número: 3	Historia: 3	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Los números de la paginación se muestran más grandes que el resto de letras en la lista.		

Demo con el cliente

Fecha: 19/08/2016

El equipo de trabajo mantuvo una reunión con el cliente para presentar el módulo de empresas, las pantallas resultantes de este sprint son:

Formulario para registro de empresas y contactos

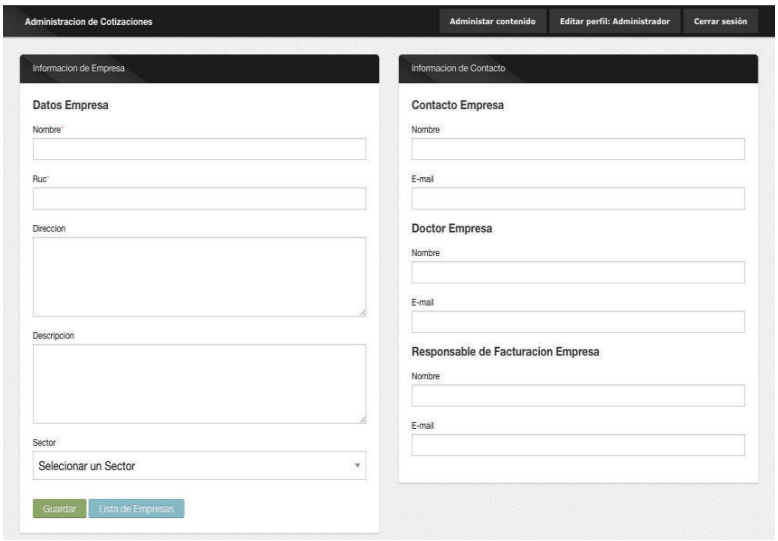


Figura 34 - Ilustración formulario para edición de empresas y contactos

Formulario para edición de empresas y contactos

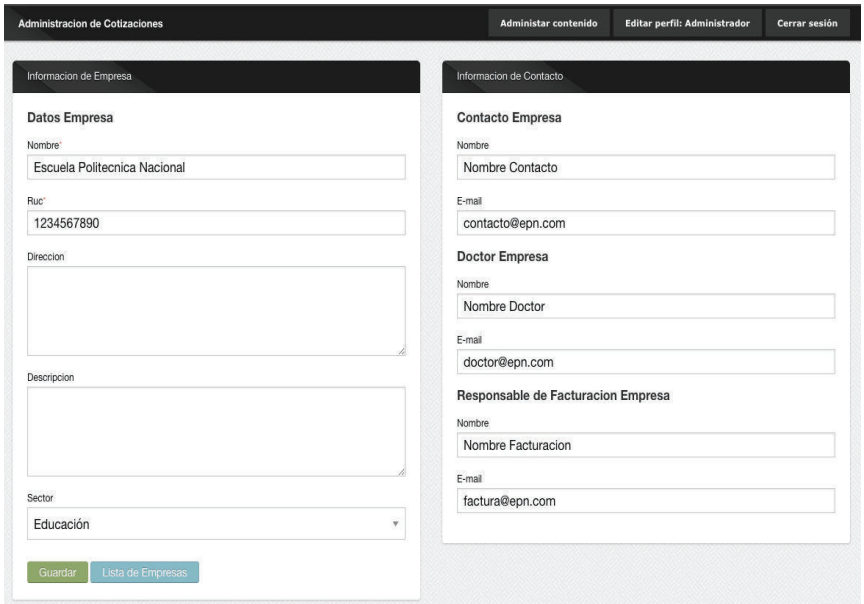


Figura 35 – Ilustración formulario para edición de empresas y contactos

Lista de empresas

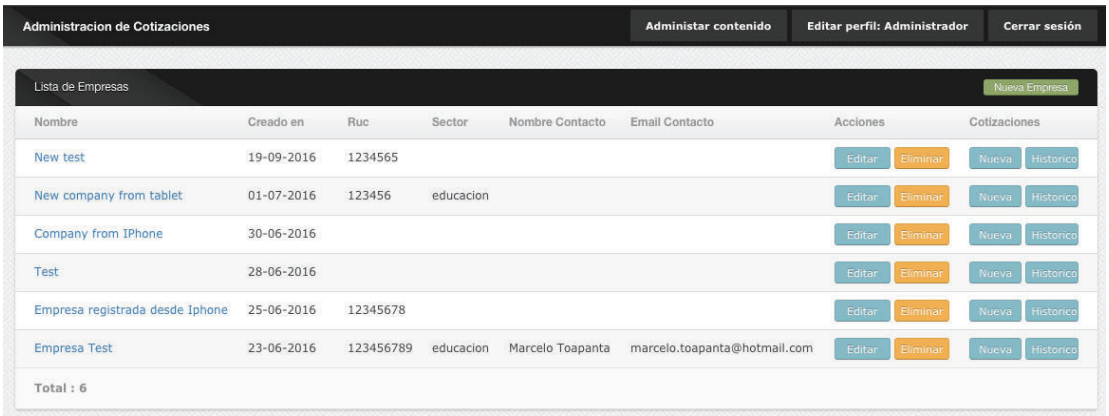


Figura 36 – Ilustración lista de empresas

Retroalimentación:

El cliente registró una empresa sin complicaciones, en ese momento el mismo preguntó ¿El sistema detecta empresas con el nombre o RUC duplicados?, el equipo de trabajo supo indicar que la validación por nombre estaba implementada pero no la validación por RUC. Para cubrir con el requerimiento del cliente se creó una tarea nueva en siguiente *sprint.*

El cliente solicitó que las columnas de “nombre” y “creado en” sean ordenables, se tomó nota de su requerimiento pero indicando que no es parte de lo solicitado y que se analizará la viabilidad luego de completar lo ya definido y planificado.

Sprint 3: Módulo exámenes

Objetivo: Implementar módulo de exámenes.
Historias a implementar: Historia #14, #15
Tiempo de ejecución: 10 días laborables
Puntos totales: 80

A continuación, se lista las tareas que fueron ejecutadas para implementar las historias que conforman el sprint 3.

Tabla 21 - Tarea 1 de sprint 3

Tarea: Implementar formulario para registro de exámenes		
Número: 1	Historia: 14	Puntos: 30
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Este formulario estará disponible únicamente para el Administrador del Sistema. El campo código y nombre son requeridos, y con valores únicos. Si un usuario ingresa la “url” de este formulario el sistema mostrará un mensaje con la siguiente leyenda: “El usuario actual no es autorizado para crear exámenes” y será re direccionado al directorio raíz.		

Tabla 22 - Tarea 2 de sprint 3

Tarea: Implementar formulario para edición de exámenes		
Número: 2	Historia: 15	Puntos: 30
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Este formulario estará disponible únicamente para el Administrador del Sistema.		

El administrador del sistema puede modificar los valores de los exámenes en cualquier momento para satisfacer las necesidades de los clientes. Un examen no puede ser eliminado.

Tabla 23 - Tarea 3 de sprint 3

Tarea: Implementar pantalla “lista de exámenes”		
Número: 3	Historia: 16	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Lista de exámenes ordenada por fecha de creación.		

Burndown inicial

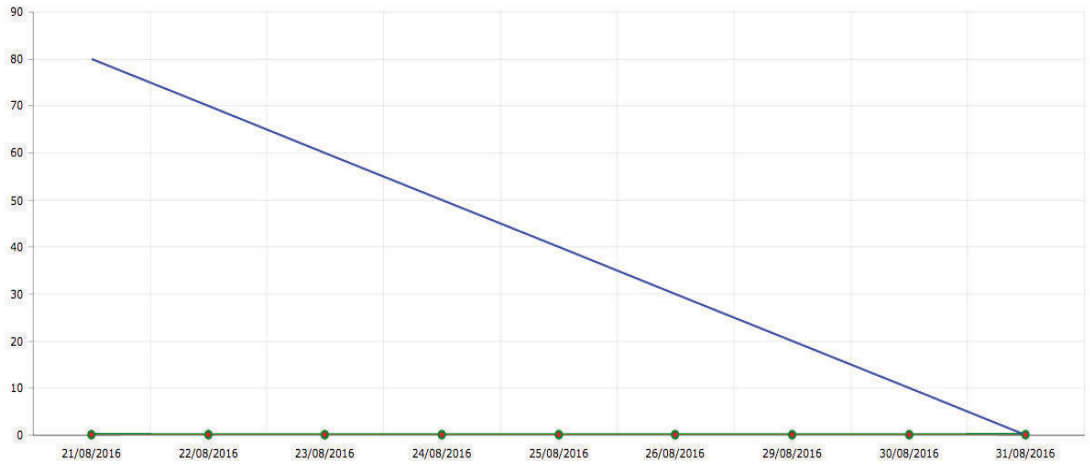


Figura 37 – *Burndown* inicial *sprint* 3

Control a los 4 días

Fecha: 25/08/2016

Puntos agregados QA: 0

Puntos avanzados: 40

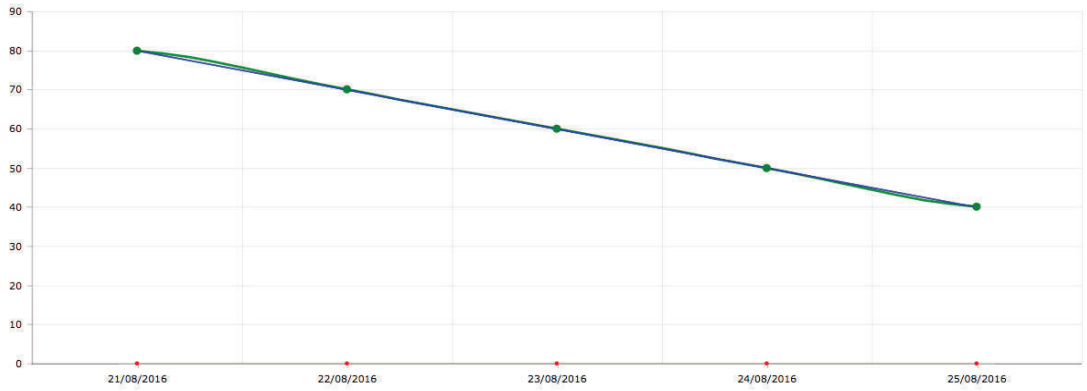


Figura 38 – *Burndown* control *sprint* 3

Al 4 día de trabajo, la Figura 38Figura 1 muestra que las tareas se han cumplido de acuerdo al cronograma planificado. El equipo de QA en estos 4 días no ha reportado incidencias por lo cual no se agregaron puntos al trabajo de este sprint.

Control a los 8 días

Fecha: 31/08/2016

Puntos agregados QA: 10

Puntos avanzados: 90

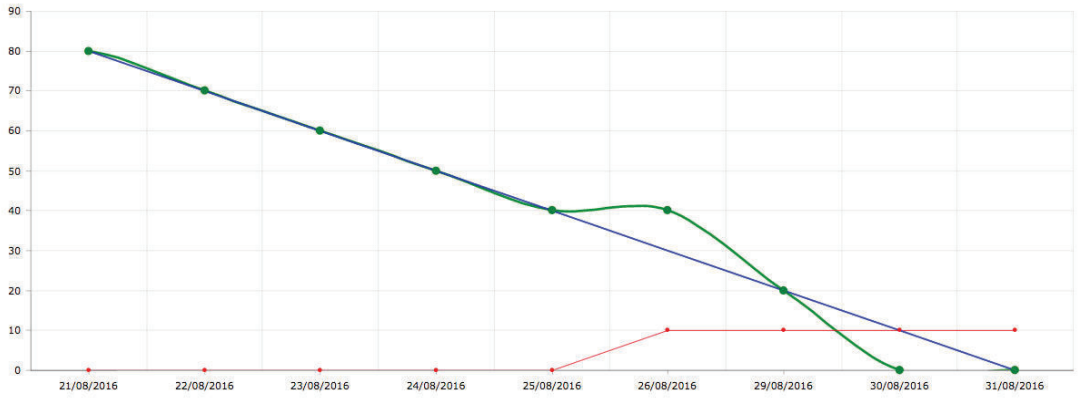


Figura 39 – *Burndown* final *sprint* 3

Al finalizar el sprint se analizaron los resultados obtenidos, como principal novedad la Figura 39 detalla que al quinto día se registró un incidente estimado con 10 puntos, el mismo representó un posible retraso para la entrega.

Tabla 24 - Incidente 1 de *sprint* 3

Incidente: Mostrar el total de empresas		
Número: 1	Historia: 16	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Mostrar el total de las empresas al final de la lista.		

El equipo de desarrollo añadió el total de empresas al final de la tabla como una fila, de esta manera se mitigó el incidente reportado por QA, y en adición a este cumplió los 10 puntos planificados para este día, la Figura 39 muestra que el equipo de trabajo logró superar los contratiempos.

Al llegar el séptimo día el equipo de trabajo completó el total de puntos calculados para este sprint, por consecuencia se logró terminar un día antes con todo el trabajo.

Demo con el cliente
Fecha: 01/09/2016

El equipo de trabajo presentó el demo correspondiente al módulo de exámenes, se recomienda ser rápidos en la presentación de los resultados, el cliente indicó que no necesita introducciones al contenido de las pantallas ya que conoce muy bien la lógica del proceso.

Los resultados obtenidos al final de este sprint son:

Formulario para registro de exámenes

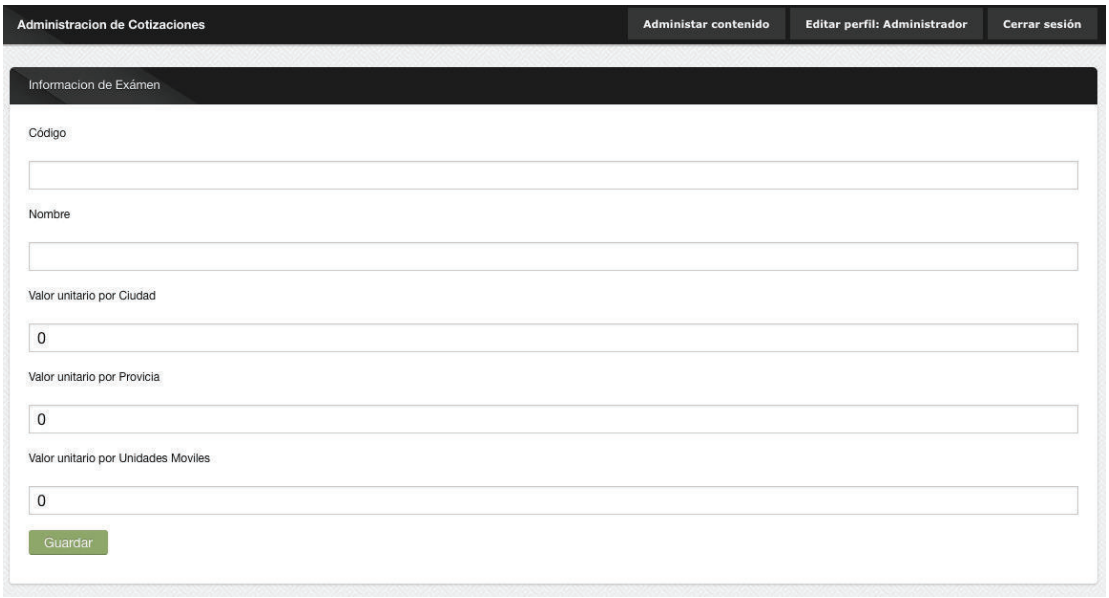


Figura 40 – Ilustración formulario para registro de exámenes

Formulario para editar exámenes

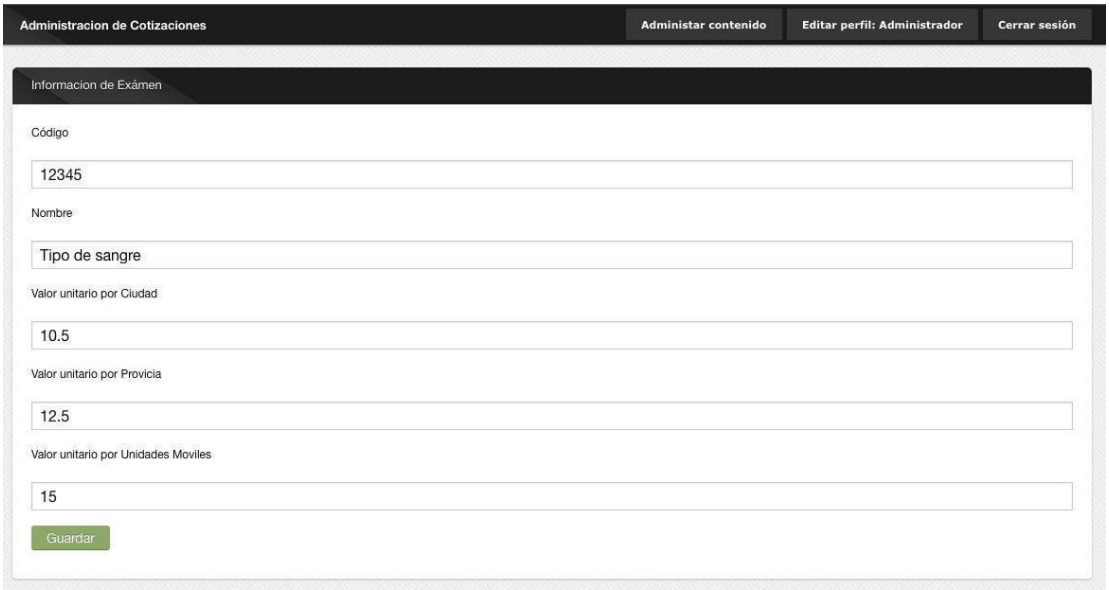


Figura 41 – Ilustración formulario para editar exámenes

Lista de exámenes

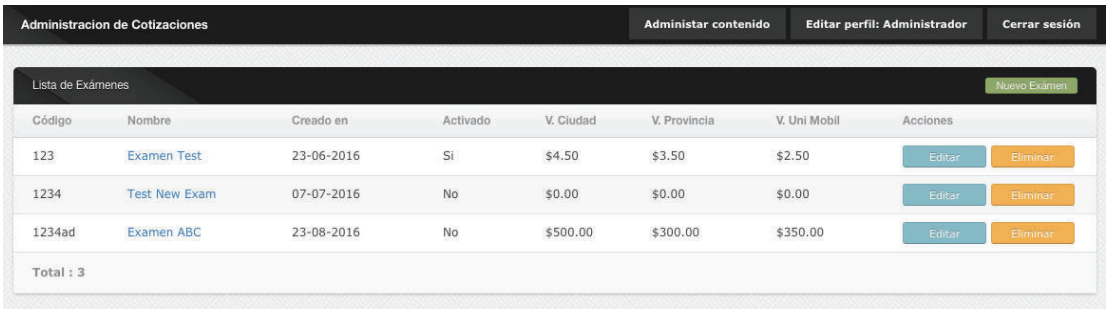


Figura 38 – Ilustración lista de exámenes

Retroalimentación:

El cliente mostró su inquietud ante el mecanismo para evitar que los usuarios que no son el administrador del sistema, puedan acceder a este módulo, el equipo de desarrollo supo explicar que mediante el rol del usuario se puede definir a que módulos podrá acceder, incluso si se ingresa directamente la “url” del módulo, el sistema re direccionará al directorio raíz del sistema y mostrará un mensaje indicando que no puede acceder debido a que no tiene los permisos necesarios.

Sprint 4: Módulo cotizaciones

Objetivo: Implementar módulo de cotización.
Historias a implementar: Historia #4, #5, #6
Tiempo de ejecución: 10 días laborables
Puntos totales: 130

El sprint 4 implementó las historias de usuario correspondientes al módulo de cotizaciones, en particular este módulo tuvo un alto grado de complejidad debido a que es el encargado de registrar las cotizaciones, sus montos parciales y totales.

Tabla 25 - Tarea 1 de *sprint* 4

Tarea: Implementar formulario para registrar cotizaciones		
Número: 1	Historia: 4	Puntos: 50
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Para registrar una cotización es necesario especificar qué empresa requiere los servicios de ECUAMERICAN, una vez seleccionada la empresa la información de la misma se mostrará en este formulario.		

Tabla 26 - Tarea 2 de *sprint* 4

Tarea: Implementar pantalla “lista de cotizaciones”		
Número: 2	Historia: 5	Puntos: 30
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones:		

Tabla 27 - Tarea 3 de sprint 4

Tarea: Implementar formulario para aprobar cotizaciones		
Número: 3	Historia: 6	Puntos: 50
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Las cotizaciones que sean aprobadas no se podrán editar más. Se deberá registrar la fecha de aprobación y el usuario aprobador de forma automática.		

Burndown inicial

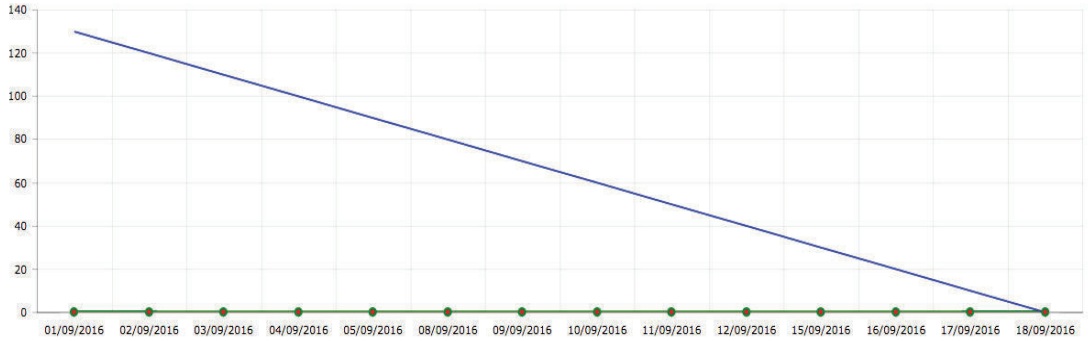


Figura 42 – *Burndown* inicial *sprint* 4

Control a los 7 días

Fecha: 10/09/2016

Puntos agregados QA: 0

Puntos avanzados: 80

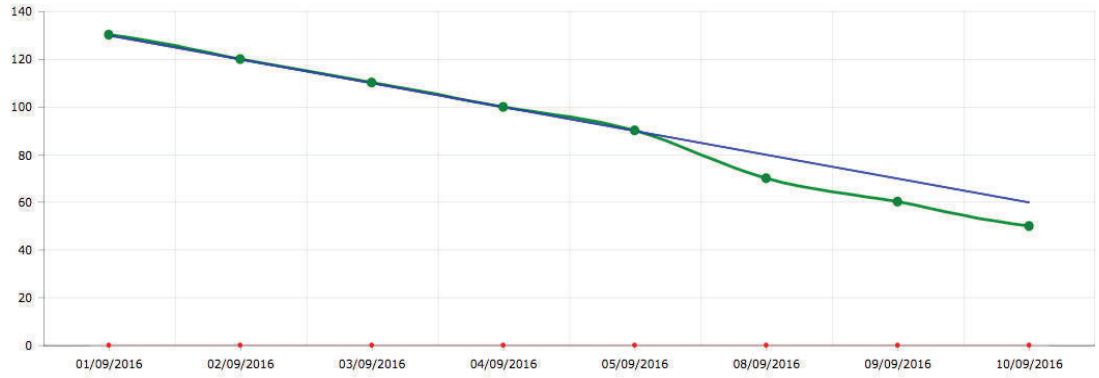


Figura 43 – *Burndown* control *sprint* 4

Llegado el día quinto la Figura 43 indica un trabajo óptimo, sin embargo, se presentaron errores al momento que el equipo de QA registró una cotización, en las pruebas realizadas anteriormente se registró una empresa sin nombre, la misma provocó un error al momento de mostrar los detalles de la nueva cotización. El equipo de desarrollo eliminó la empresa antes mencionada y el error fue solucionado.

Otro error que fue detectado al momento de realizar pruebas, fue que el sistema permitía registrar una cotización sin exámenes, lo cual no debería ser posible. El equipo de desarrollo aumentó una validación a nivel de modelo para mitigar este incidente. De tal forma que una cotización no puede ser registrada si no se ha agregado al menos un examen.

Control a los 13 días

Fecha: 18/09/2016

Puntos agregados QA: 10

Puntos avanzados: 140

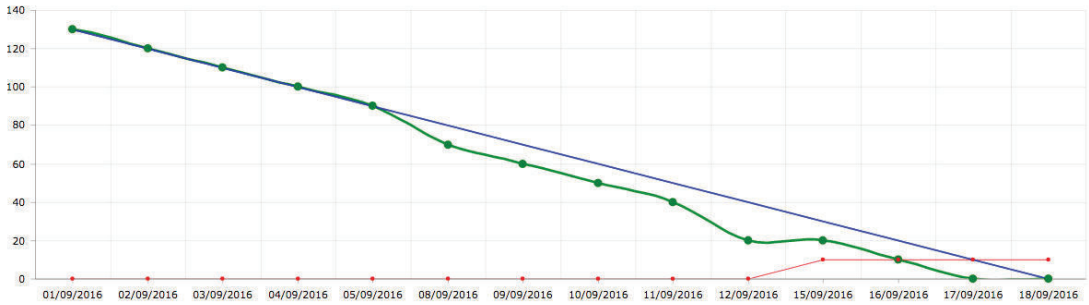


Figura 44 – *Burndown* final *sprint* 4

La Figura 44 En el noveno día se presentó otro avance que superó en 10 puntos lo planificado inicialmente, sin embargo, en el día décimo el equipo de QA registra un incidente.

Tabla 28 - Incidente 1 de *sprint* 4

Incidente: Bloquear opción de aprobar cotización		
Número: 1	Historia: 6	Puntos: 50
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán

Observaciones: Cuando el usuario aprobó una cotización, el formulario del mismo debe bloquear el campo de aprobar, de tal manera no se podrá cambiar el valor de este campo.

El incidente detallado en la Tabla 28 no representa un riesgo para la fecha de entrega debido a que el equipo de trabajo logró dos grandes avances mencionados anteriormente. El equipo de desarrollo implementó una validación a nivel de código que detecta si la cotización fue aprobada anteriormente, si este es el caso el formulario se mostrará con el campo aprobar boqueado para el usuario. Finalmente y pese a que existió un incidente el equipo de trabajo logró cumplir con la meta del sprint un día antes de lo planeado.

Demo con el cliente

Fecha: 19/09/2016

A continuación se muestran los resultados obtenidos al final de este *sprint*.

Formulario para registro de cotizaciones

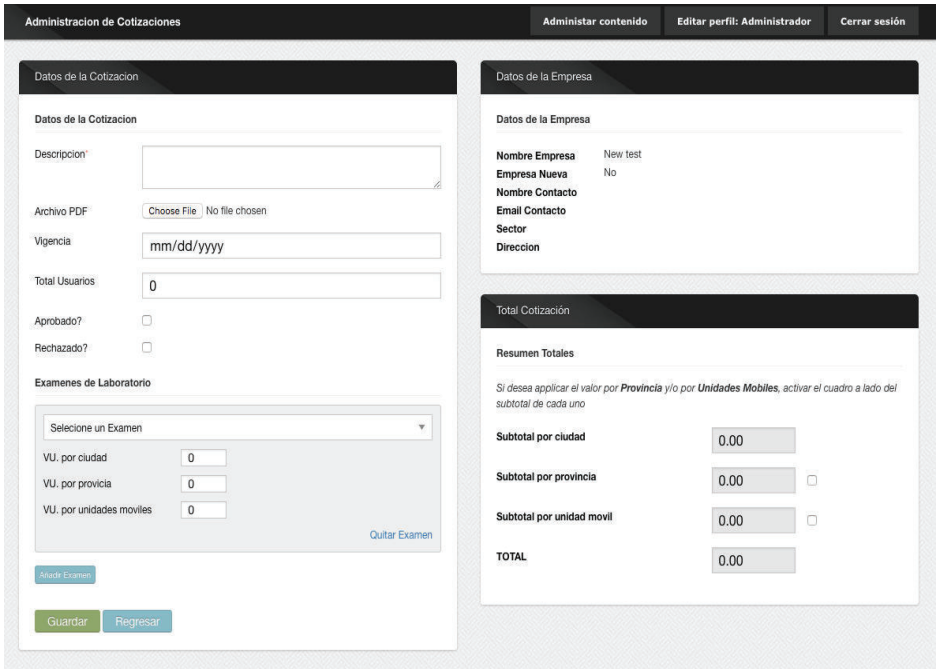


Figura 45 – Ilustración formulario para registro de cotizaciones

Formulario para editar cotizaciones

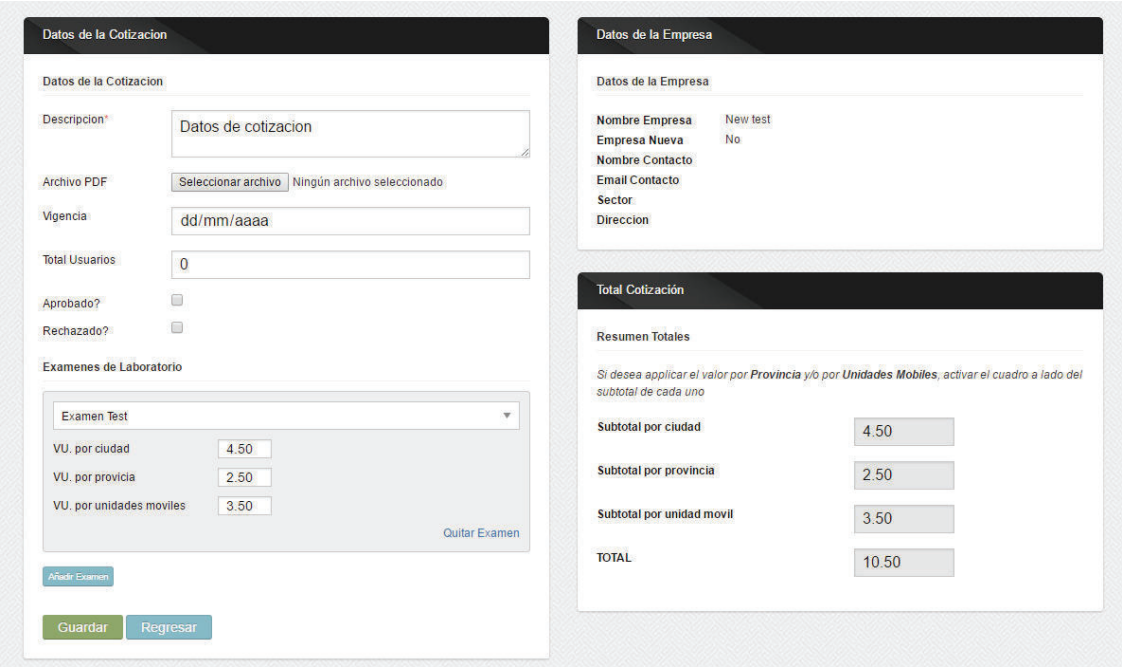


Figura 46 – Ilustración formulario para editar cotizaciones

Lista de cotizaciones

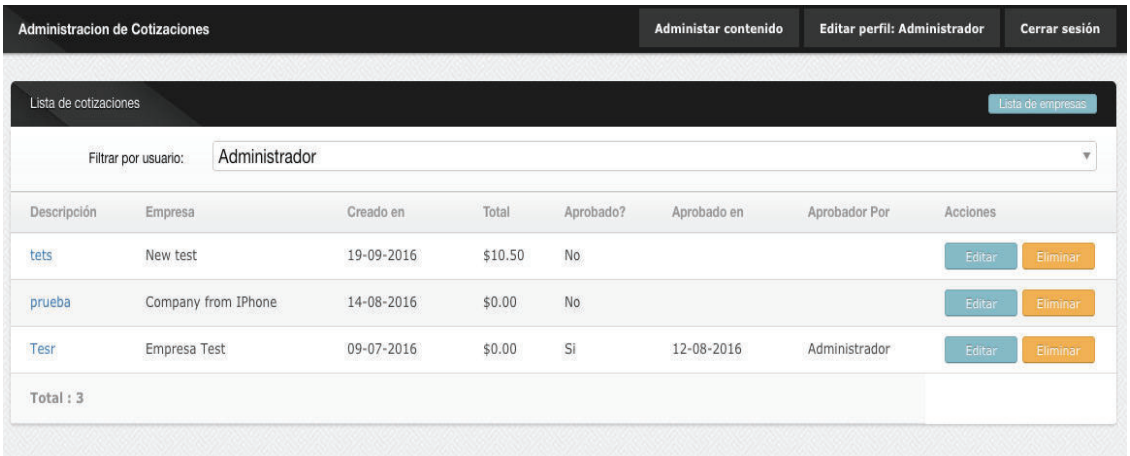


Figura 47 – Ilustración lista de cotizaciones

Retroalimentación:

Los requerirnos del cliente fueron cubiertos satisfactoriamente, sin embargo en la demostración el cliente notó que es necesario implementar una nueva funcionalidad que permita activar/desactivar los subtotales, de tal manera el total se verá afectado cuando el usuario lo crea necesario sin perder los datos.

Adicionalmente el cliente solicitó un cambio de mensaje en alerta para crear cotizaciones, cambiar de “Cotización creada exitosamente” a “Nueva cotización creada exitosamente para (nombre de empresa)”.

Debido a que el sprint 4 finalizó, las tareas solicitadas por el cliente serán cubiertas en el sprint 5, esto modificará el cronograma dependiendo de la dificultad que representen las nuevas tareas.

Sprint 5: Módulo panel digital

Objetivo: Implementar módulo panel digital.
Historias a implementar: Historia #8, #9, #10, #11
Tiempo de ejecución: 9 días laborables
Puntos totales: 90

El equipo de desarrollo lista las tareas que fueron ejecutadas para implementar las historias del módulo panel digital y las tareas resultantes del *sprint* 4.

Tabla 29 – Tarea 1 *sprint* 5

Tarea: Implementar funcionalidad para activar/desactivar los subtotales		
Número: 1	Historia: 6	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El usuario debe tener la capacidad de activar o desactivar los subtotales de tal manera que se reflejen el total. Existen 3 subtotales para una cotización: por ciudad, por provincia, por unidad móvil, de las cuales “por ciudad” y “por provincia” serán las tomadas en cuenta para esta funcionalidad.		

Tabla 30 – Tarea 2 de *sprint* 5

Tarea: Cambio de mensaje en alerta para crear cotizaciones		
Número: 2	Historia: 6	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Cambiar de “Cotización creada exitosamente” a “Nueva cotización creada exitosamente para (nombre de empresa)”.		

Tabla 31 - Tarea 3 de *sprint* 5

Tarea: Implementar panel digital para Administrador del Sistema		
Número: 3	Historia: 10	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El usuario administrador del sistema podrá ver la información de todos los usuarios.		

Tabla 32 - Tarea 4 de *sprint* 5

Tarea: Implementar panel digital para usuario aprobador		
Número: 4	Historia: 9	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Los aprobadores verán la información registrada únicamente por ellos.		

Tabla 33 - Tarea 5 de *sprint* 5

Tarea: Implementar panel digital para usuario vendedor - comercial		
Número: 5	Historia: 8	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Los vendedores podrán ver la información registrada únicamente por ellos		

Tabla 34 - Tarea 6 de *sprint* 5

Tarea: Implementar pantalla “buscar cotización”		
Número: 6	Historia: 11	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones:		

Burndown inicial

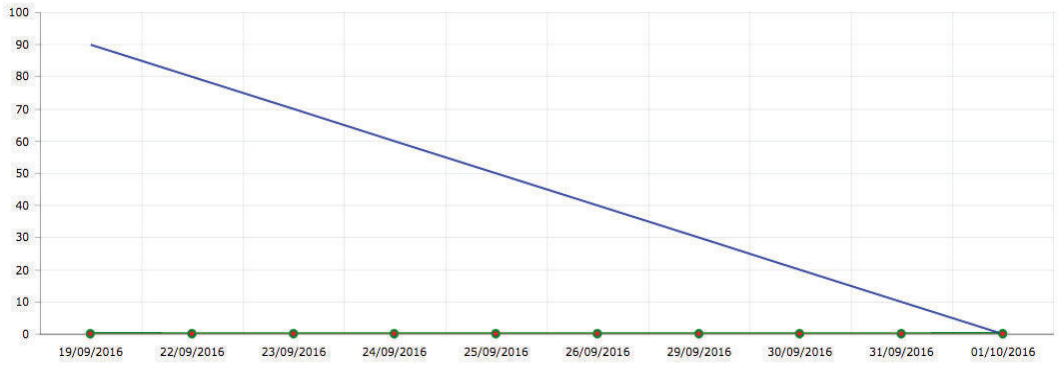


Figura 48 – *Burndown* inicial *sprint* 5

Control a los 4 días

Fecha: 26/09/2016

Puntos agregados QA: 10

Puntos avanzados: 60

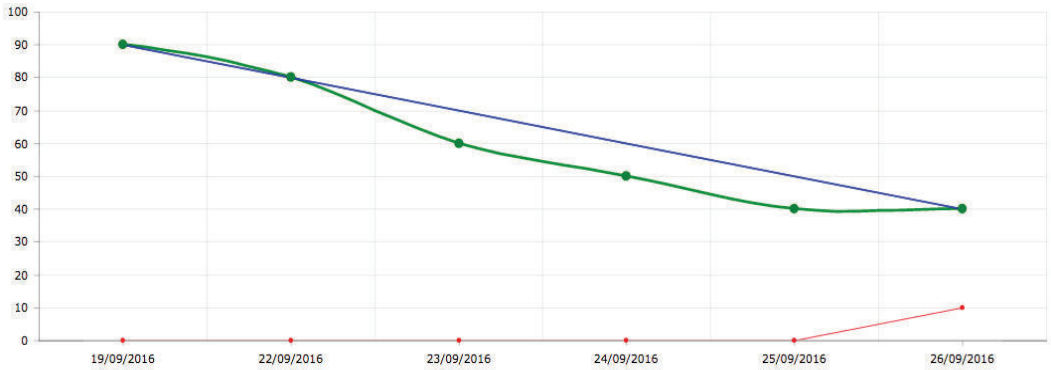


Figura 49 – *Burndown* control *sprint* 5

Se puede observar en la Figura 49 que al segundo día el equipo de trabajo completó 20 puntos cuando se requerían 10, esto representó un adelanto de un día de trabajo. El equipo de QA registró un incidente estimado con 10 puntos en el día quinto, el cual es cubierto con el adelanto generado en el día dos, esto no representó un riesgo para la fecha de entrega.

Tabla 35 - Incidente 1 de *sprint* 5

Incidente: Estilos de paginación en pantalla de buscar cotizaciones		
Número: 1	Historia: 11	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Los números se muestran más grandes que el resto de letras, además debe mostrarse alineado horizontalmente.		

Control a los 7 días

Fecha: 01/10/2016

Puntos agregados QA: 10

Puntos avanzados: 100

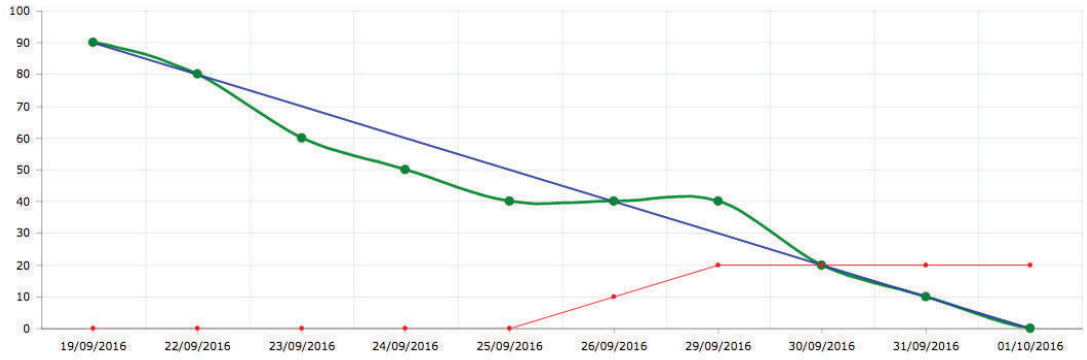


Figura 50 – *Burndown* final *sprint* 5

La Figura 50 muestra que, al cumplir el sexto día, el equipo de QA registró un nuevo incidente estimado con 10 puntos, como es de conocimiento esto representa un riesgo para la fecha de entrega. El equipo de trabajo se reúne para solucionar este incidente,

la solución es definir reglas de estilo *css* independientes para la paginación, de tal manera que no afecte al resto del sistema.

Al momento que el equipo de QA registro una cotización como vendedor, para comprobar que las gráficas estadísticas del panel digital contaban correctamente todas ellas, se dieron cuenta que los datos mostrados eran de todos los usuarios, el cliente indicó que un vendedor podrá ver las estadísticas únicamente de sus cotizaciones. El equipo de desarrollo revisó la lógica de las gráficas, identificó el problema y agregó el filtro por usuario requerido.

Demo con el cliente
Fecha: 02/10/2016

Una vez terminado el sprint, las tareas implementadas, tanto las resultantes en el sprint anterior como las que conforman este sprint, se obtuvieron las siguientes pantallas.

Funcionalidad para activar/desactivar subtotales

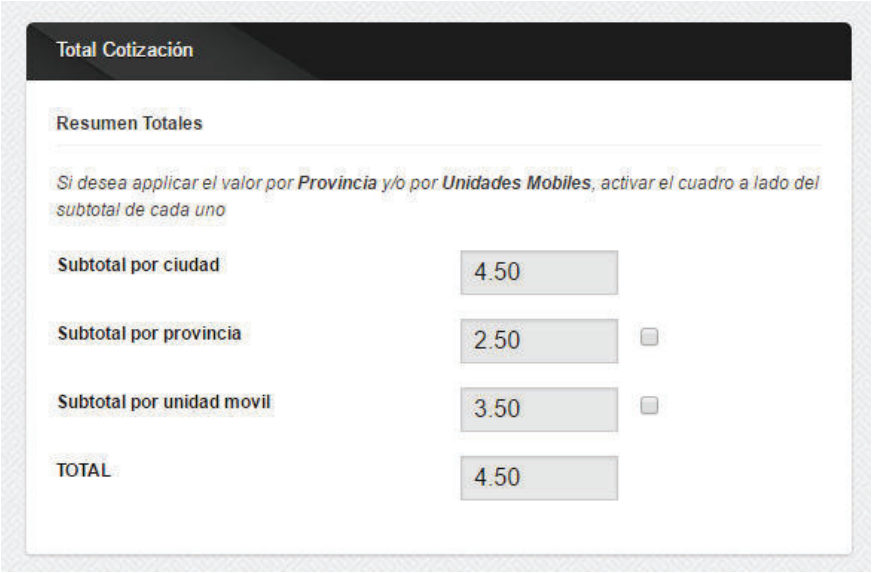


Figura 51 - Ilustración subtotales desactivados

Figura 52 – Ilustración subtotales activados

Cambio de mensaje en alerta para crear cotización

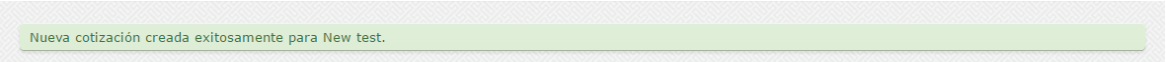


Figura 53 – Ilustración cambio de mensaje

Panel digital

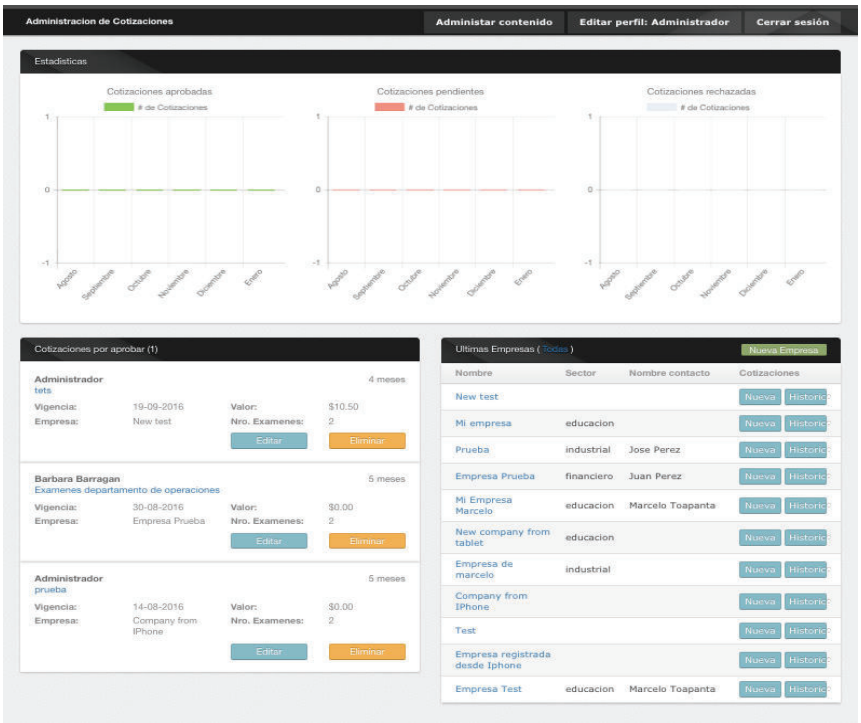


Figura 54 – Ilustración panel digital

Buscar cotización

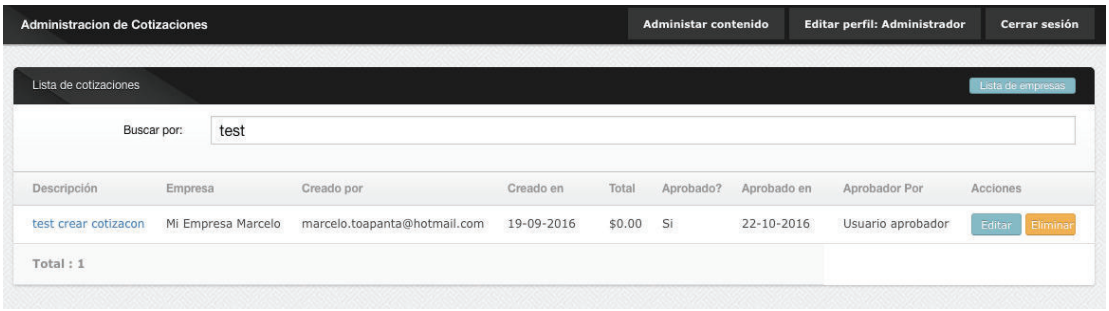


Figura 55 – Ilustración buscar cotización

Retroalimentación:

Particularmente en este sprint se puede observar como *Scrum* permitió cubrir con las tareas nuevas resultado de la presentación con el cliente, se agregó 2 tareas resultantes del sprint 4, que fueron cubiertas en este ciclo.

El equipo de trabajo mostró al cliente los resultados, en la reunión se probó la nueva funcionalidad para activar y desactivar los totales de cada cotización. El cliente registro varias cotizaciones como un usuario vendedor para comprobar el correcto funcionamiento de las gráficas estadísticas. Adicional a esto mostró la inquietud sobre qué pasaría, si, 100 vendedores registran una cotización respectivamente, ¿Cómo las tablas se comportarían? El equipo de desarrollo indicó que las tablas realizan cálculos internos para mostrar los totales en escalas, de acuerdo al ancho y alto de las pantallas en las que el usuario este accediendo al sistema.

Sprint 6: Diseño *responsive* y navegabilidad

Objetivo: Implementar menú de navegación y funcionalidad *responsive*.

Historias a implementar: Historia #12, #13

Tiempo de ejecución: 12 días laborables

Puntos totales: 120

Tabla 36 - Tarea 1 de *sprint* 6

Tarea: Implementar menú de navegación		
Número: 1	Historia: 12	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones:		

Tabla 37 - Tarea 2 de *sprint* 6

Tarea: Implementar funcionalidad responsive para módulo usuarios		
Número: 2	Historia: 13	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: La meta de esta tarea es implementar la funcionalidad responsive para las pantallas del módulo usuarios. No se implementará nuevos modelos ni opciones.		

Tabla 38 - Tarea 3 de *sprint* 6

Tarea: Implementar funcionalidad responsive para módulo empresas		
Número: 3	Historia: 13	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta

Responsable QA:	Bárbara Barragán
Observaciones: La meta de esta tarea es implementar la funcionalidad responsive para las pantallas del módulo empresas. No se implementará nuevos modelos ni opciones.	

Tabla 39 - Tarea 4 de *sprint* 6

Tarea: Implementar funcionalidad responsive para módulo exámenes		
Número: 4	Historia: 13	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: La meta de esta tarea es implementar la funcionalidad responsive para las pantallas del módulo exámenes. No se implementará nuevos modelos ni opciones.		

Tabla 40 - Tarea 5 de *sprint* 6

Tarea: Implementar funcionalidad responsive para módulo cotizaciones		
Número: 5	Historia: 13	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: La meta de esta tarea es implementar la funcionalidad responsive para las pantallas del módulo cotizaciones. No se implementará nuevos modelos ni opciones.		

Tabla 41 - Tarea 6 de sprint 6

Tarea: Implementar funcionalidad responsive para módulo panel digital		
Número: 6	Historia: 13	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: La meta de esta tarea es implementar la funcionalidad responsive para las pantallas del módulo panel digital. No se implementará nuevos modelos ni opciones.		

***Burndown* inicial**

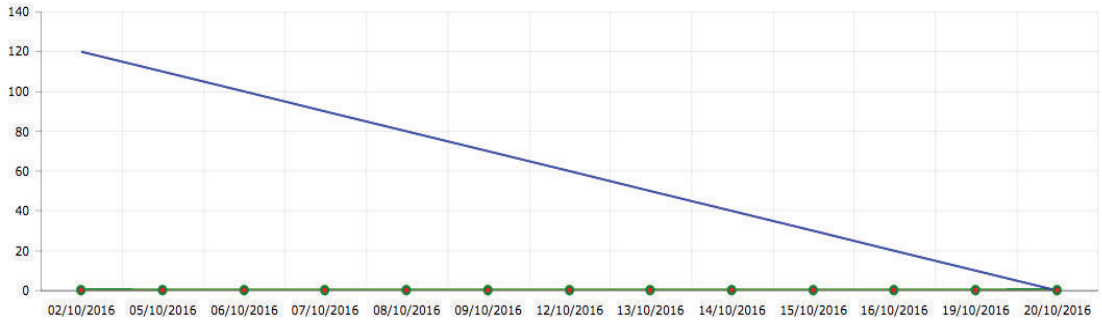


Figura 56 – *Burndown* inicial *sprint* 6

Control a los 6 días

Fecha: 12-10-2016

Puntos agregados QA: 10

Puntos avanzados: 80

Se realizó un control del avance del equipo de trabajo. En este sprint no se aumentó lógica al sistema, solo se agregó la funcionalidad para que el diseño existente se ajuste a las pantallas de los diferentes dispositivos ya mencionados previamente.

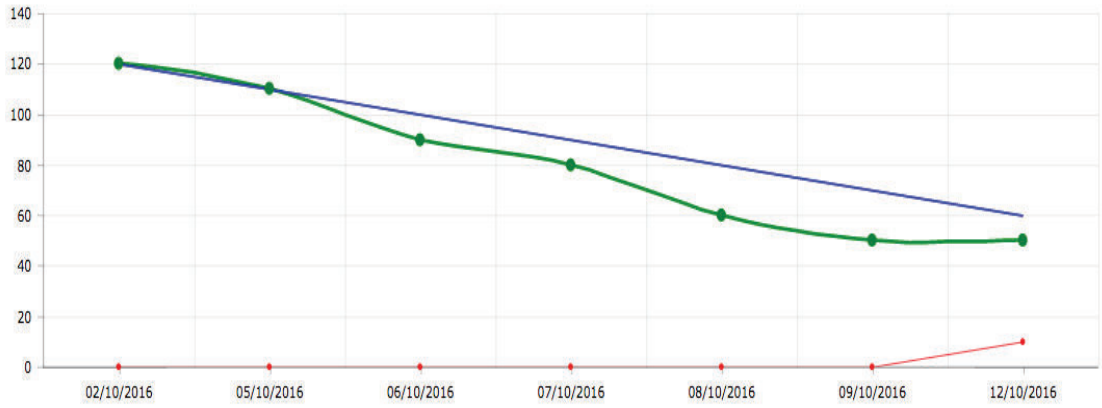


Figura 57 – *Burndown* control *sprint* 6

En el segundo día, se avanzó 10 puntos más sobre de planeado al igual en el cuarto día, se avanzó otro 10 puntos más sobre lo planeado, estos dos avances generados por el equipo ayudará a cumplir con la meta antes de la fecha.

Al llegar el sexto día, el equipo de QA registró un incidente estimado en 10 puntos, sin embargo no representa un riesgo gracias a los avances previos del equipo.

Tabla 42 - Incidente 1 de *sprint* 6

Incidente: Hacer link nombre de empresa		
Número: 1	Historia: 13	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: En versiones par mobile y tablets el nombre de la empresa debe ser un link para que el usuario pueda visualizar los detalles de la empresa.		

Control a los 12 días

Fecha: 20-10-2016

Puntos agregados QA: 10

Puntos avanzados: 110

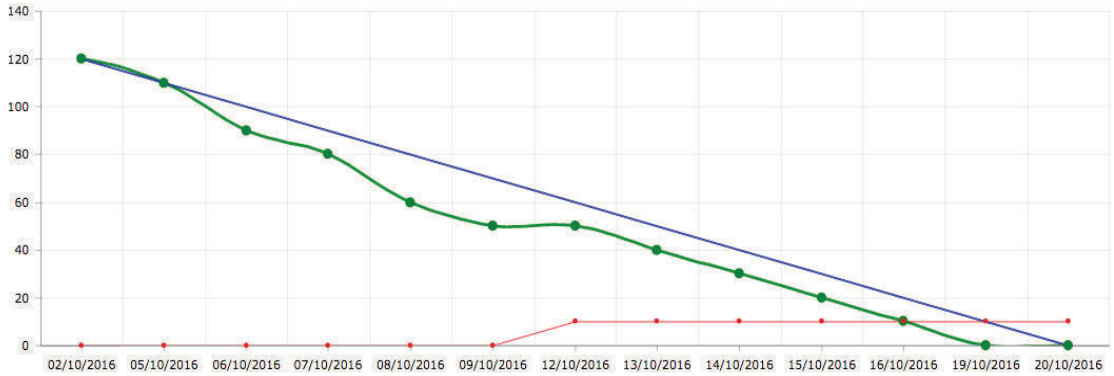


Figura 58 – Burndown final sprint 6

El equipo de QA realizo varias pruebas para comprobar el correcto funcionamiento del sistema en dispositivos móviles, para esto instaló simuladores de IOS y Android. Los navegadores actuales como Chrome, Firefox, IE, Safari, contienen simuladores para realizar este tipo de pruebas, sin embargo existen reglas CSS que no son compatibles con todos los navegadores. El equipo de desarrollo utilizó únicamente reglas CSS del *framework* Foundation, para evitar dichos problemas de compatibilidad.

Otro problema registrado por el equipo de QA es que las fuentes no son mostradas de igual manera en todos los dispositivos, a pesar de llevar el mismo nombre. Este inconveniente no afecta con la lógica del sistema, el mismo será notificado al cliente en la siguiente reunión.

Demo con el cliente

Fecha: 21-10-2016

Se puede observar el resultado de la implementación de la funcionalidad responsive en las pantallas del sistema en las siguientes figuras:

Menú de navegación

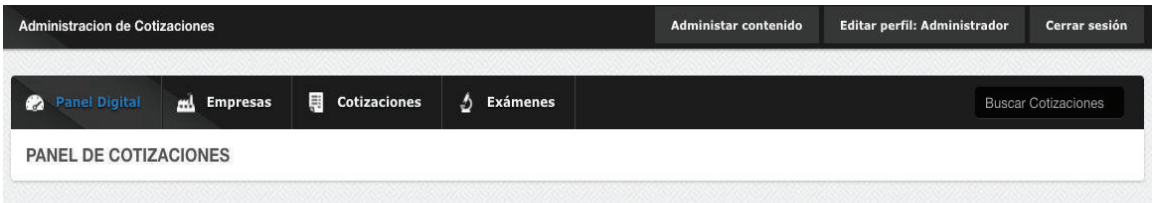


Figura 59 – Ilustración menú de navegación

Vistas para tablets

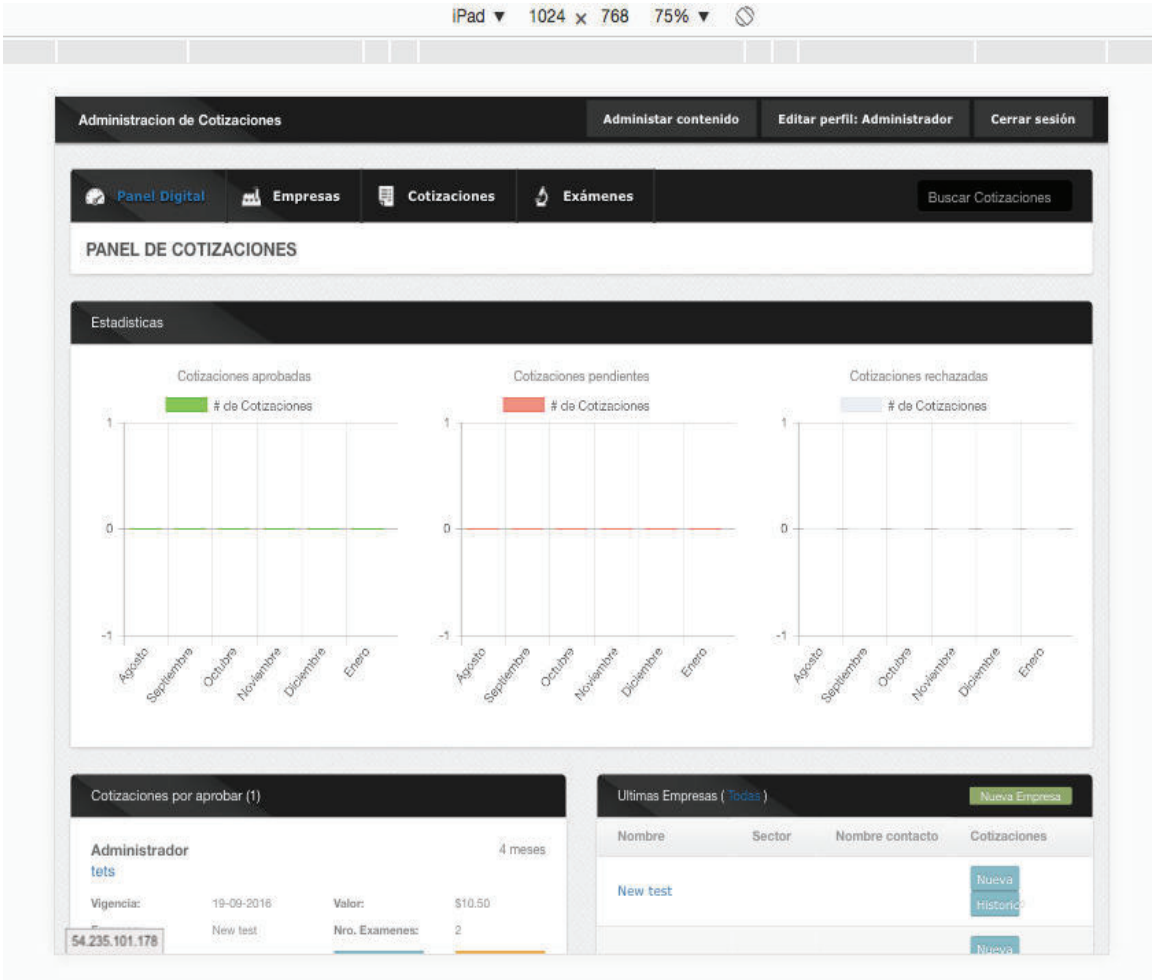


Figura 60 – Ilustración vistas para tablets horizontal

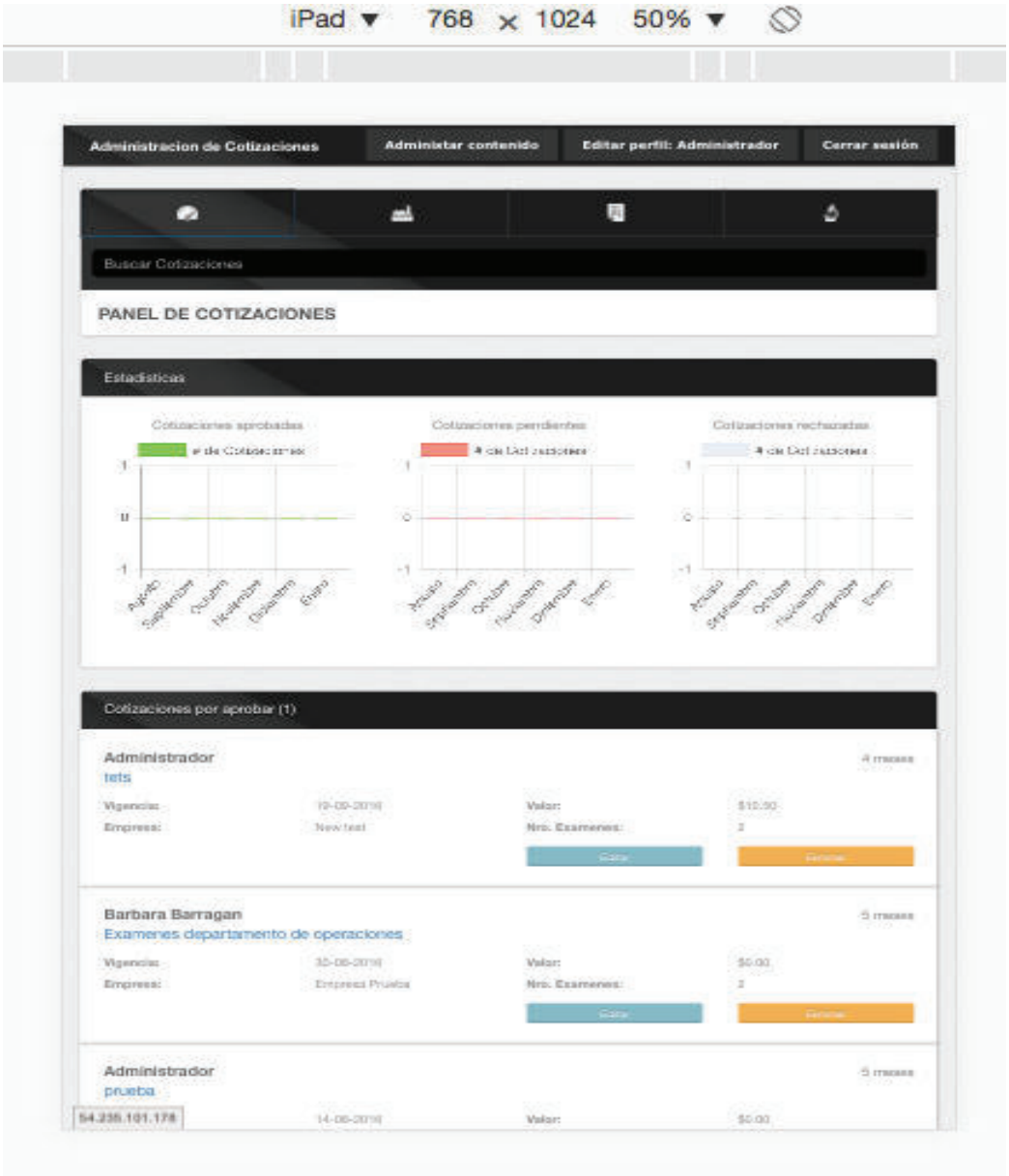


Figura 61 – Ilustración vistas para tablets vertical

Vistas para teléfonos inteligentes

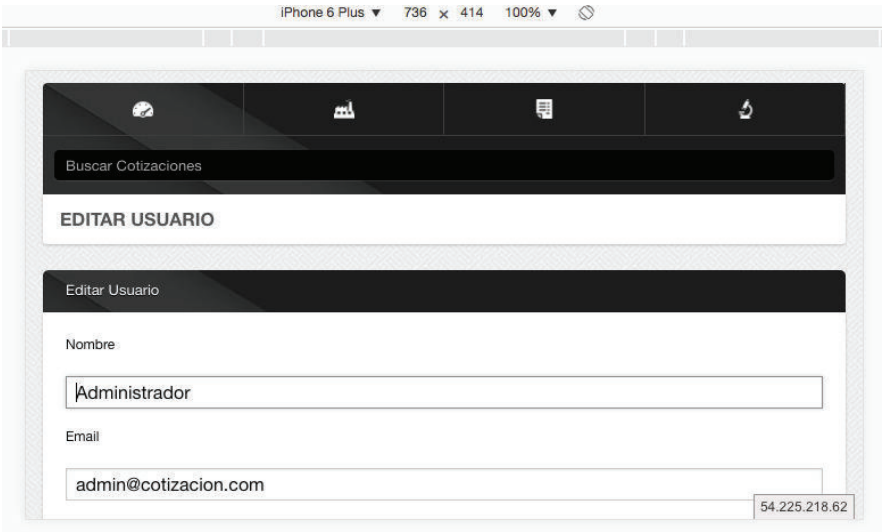


Figura 62 – Ilustración vistas para teléfonos inteligentes horizontal

Figura 63 – Ilustración vistas para teléfonos inteligentes vertical

Retroalimentación:

El equipo de trabajo notifica de las complicaciones que existieron en este ciclo, la principal relacionado con las fuentes en cada dispositivo, se indicó al cliente que cada sistema operativo y navegador gestiona sus fuentes, por ejemplo: la fuente “Arial”, mostrada en Windows no será exactamente igual a la mostrada en Android. El cliente indico que las diferencias encontradas no son motivo de preocupación debido a que son totalmente legibles y no causan problemas con la lógica del sistema.

Sprint 7: Mejoras

Objetivo: Implementar recomendaciones del equipo de desarrollo
Historias a implementar: N/A
Tiempo de ejecución: 10 días laborables
Puntos totales: 120

Tabla 43 - Tarea 1 de *sprint* 7

Tarea: Bloquear botón de envió de formulario		
Número: 1	Historia: N/A	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: El usuario debe ser capaz de enviar solo una vez, los datos de un formulario, cuando el usuario envíe el formulario el botón se bloqueará y no permitirá otro envió hasta que procese por completo el primer envío.		

Tabla 44 - Tarea 2 de *sprint* 7

Tarea: Mensaje de confirmación para formulario de cotizaciones		
Número: 2	Historia: N/A	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Al momento de registrar una cotizacion el usuario debe recibir un mensaje de confirmación con el fin de prevenir errores en la información enviada, otra razón para esto es que las cotizaciones ya no pueden ser editadas cuando son enviadas a la etapa de aprobación. De tal forma se reducen el riesgo de enviar información incompleta por error.		

Tabla 45 - Tarea 3 de *sprint* 7

Tarea: Cambio de valores de exámenes mediante número de circular correspondiente		
Número: 3	Historia: N/A	Puntos: 40
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Cuando se cambie el valor de un examen, el mismo debe ser respaldado por el número de circular que donde conste los detalles del cambio y los responsables de dicho cambio.		

Tabla 46 - Tarea 4 de *sprint* 7

Tarea: Bloquear cotizaciones que sean aprobadas o rechazadas		
Número: 4	Historia: N/A	Puntos: 20
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones:		

Burndown inicial

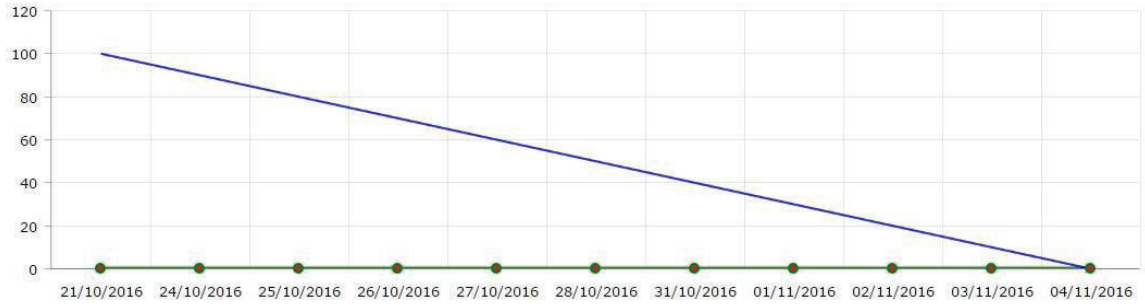


Figura 64 – *Burndown* inicial *sprint* 7

Control a los 5 días

Fecha: 28-10-2016

Puntos agregados QA: 10

Puntos avanzados: 60

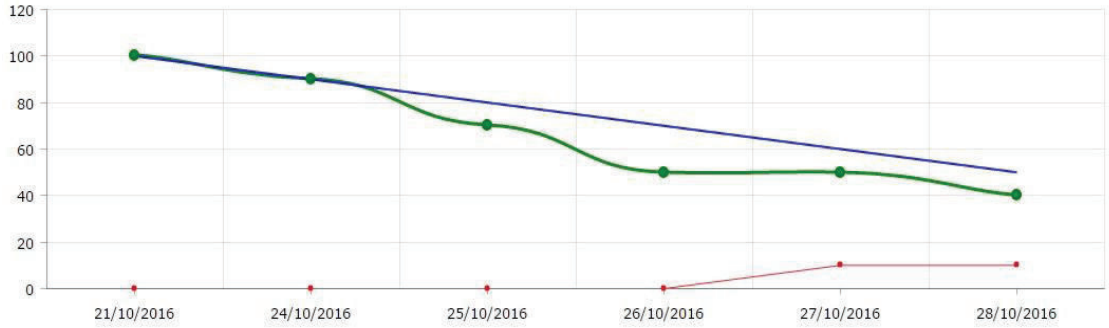


Figura 65 – *Burndown* control *sprint* 7

La Figura 65 muestra el progreso del equipo de trabajo, donde el mismo implemento el bloqueo de los botones de envió para todos los formularios del sistema, esto evita que los usuarios envíen múltiples y repetidas peticiones al servidor, por otro lado se logró agregar un mensaje de confirmación para el formulario de cotizaciones, el mismo evita que los usuario envíen un cotización por accidente.

Se inició con la tarea 3, la misma presentó un incidente, el cual consiste en mostrar un mensaje de alerta si el usuario modifica los valores de un examen pero no ingresa el número de circular. Ya que si no muestra dicho mensaje, el usuario podría interpretar que si se actualizaron los valores.

Tabla 47 - Incidente 1 de *sprint* 7

Incidente: Mensaje de alerta en formulario de exámenes		
Número: 1	Historia: N/A	Puntos: 10
Responsables Desarrollo:		Marcelo Toapanta
Responsable QA:		Bárbara Barragán
Observaciones: Cuando el administrador del sistema modifique los valores de un examen, dicho cambio debe ser respaldado por el número de circular correspondiente, caso contrario se descartaran los cambios realizados.		

Control a los 10 días

Fecha: 20-10-2016

Puntos agregados QA: 10

Puntos avanzados: 110

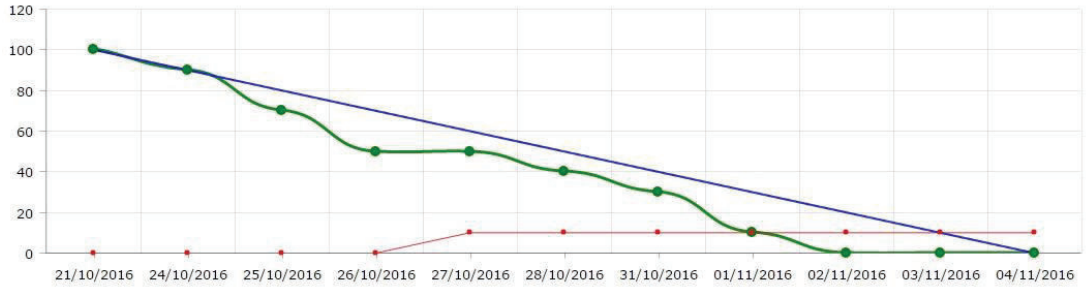


Figura 66 – Burndown final sprint 7

La Figura 66 muestra el progreso del equipo de trabajo a lo largo del sprint 7, donde se observa que se finalizó con las tareas 3 días antes de lo planificado, esto permite adelantar la fecha del demo con el cliente.

Se implementó un registro de los cambios en los valores de un examen, eso permite observar rápidamente en qué fecha hubo un cambio de valores y el número de circular que lo respalda.

La última tarea implementa consiste en bloquear los botones que permitan editar una cotización, si la misma ya fue aprobada o rechazada, es decir las cotizaciones puede ser editadas únicamente en la etapa pendiente.

Demo con el cliente

Fecha: 04-11-2016

A continuación se muestran las figuras resultantes de la implementación del sprint 7:

Bloqueo de botón una vez enviado el formulario

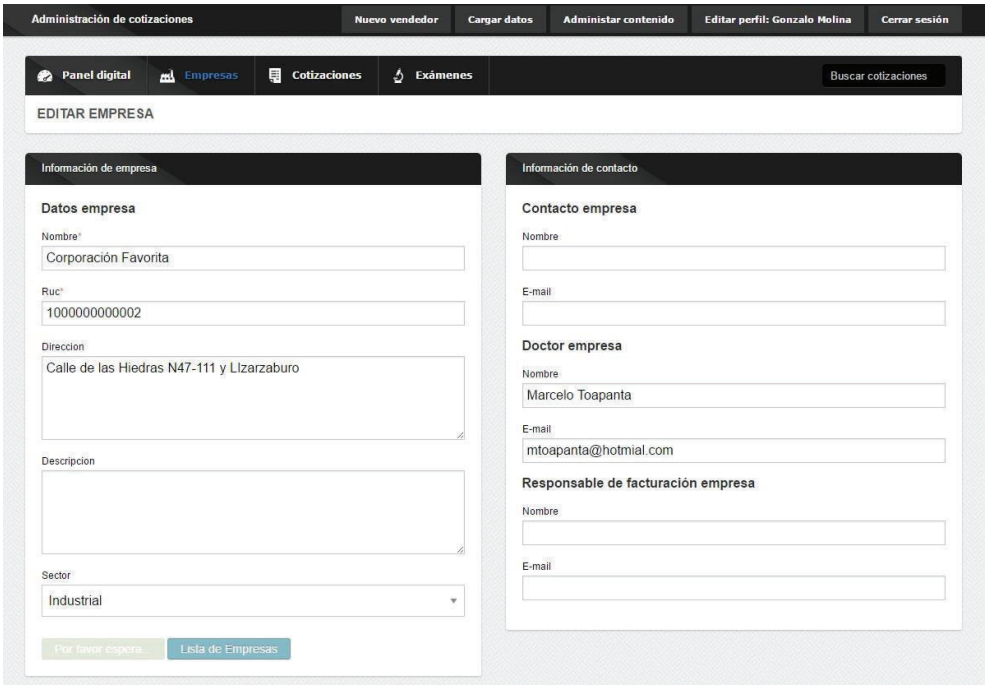


Figura 67 – Ilustración bloqueo de botón

Mensaje de confirmación para cotizaciones

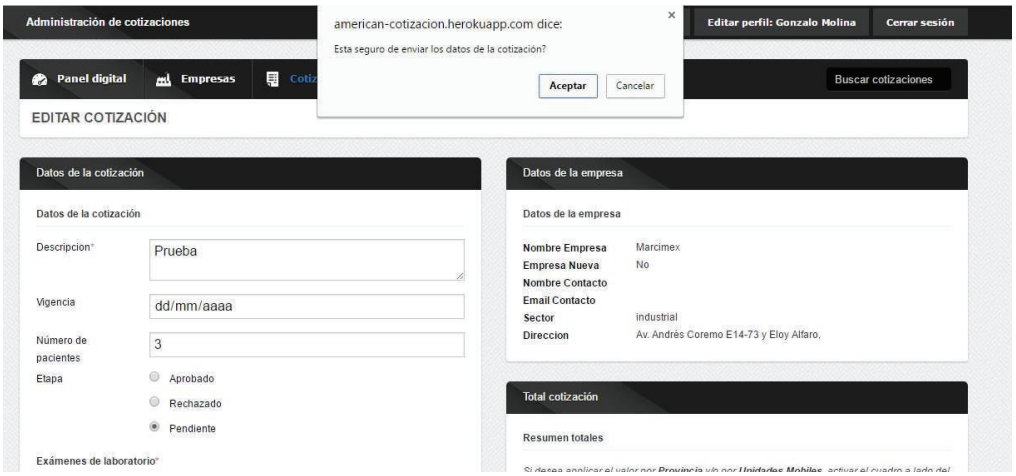


Figura 68 – Ilustración mensaje de confirmación

Cabios de valores de exámenes

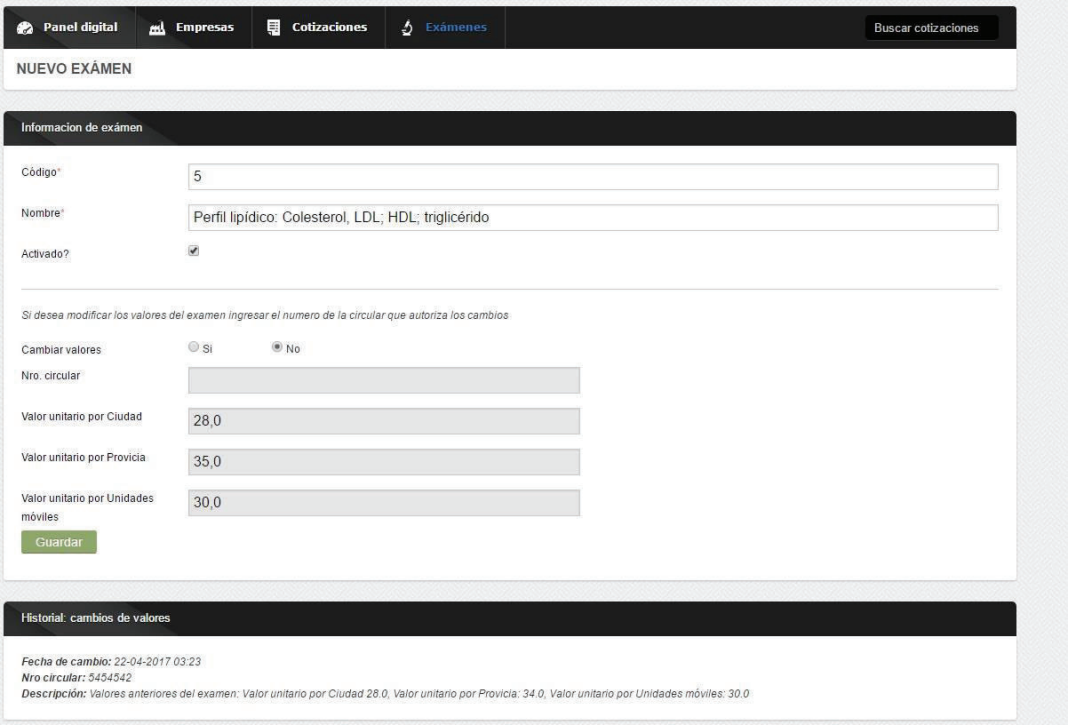


Figura 69 – Ilustración cabios de valores de exámenes

No permitir editar cotizaciones aprobadas o rechazadas

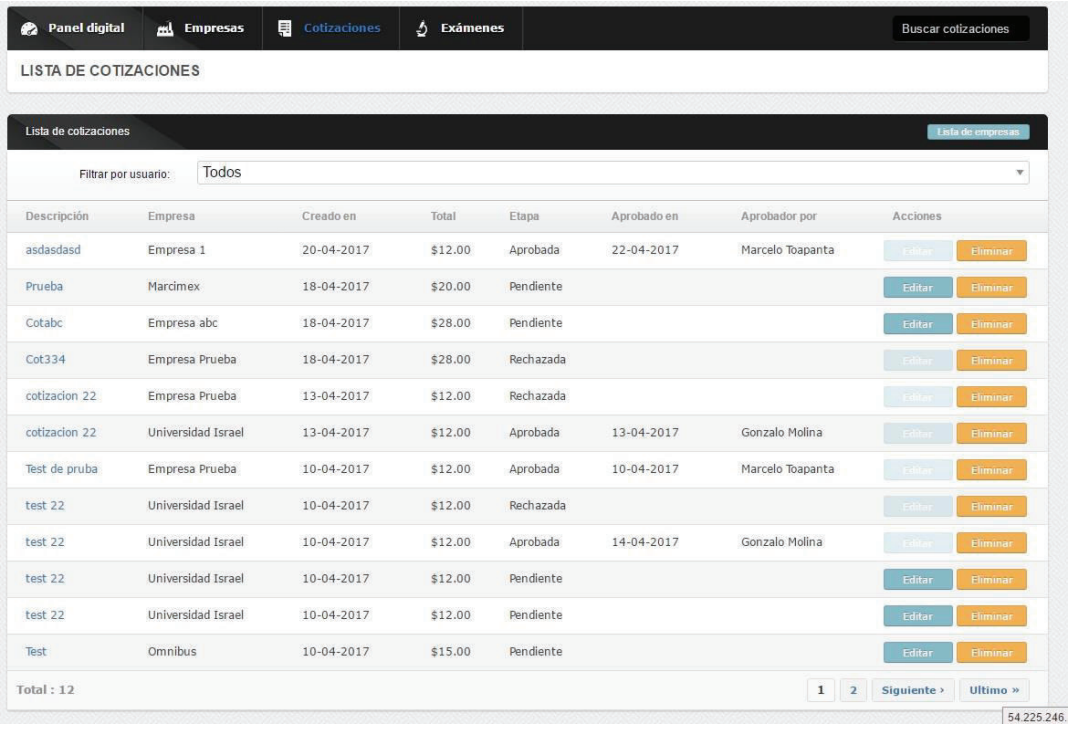


Figura 70 – Ilustración no permitir editar cotizaciones aprobadas o rechazadas

Retroalimentación:

Las mejoras sugeridas por el equipo de desarrollo fueron aceptadas por el cliente, se indicó al cliente que las mismas ayudaran al mantenimiento del sistema así como a la consistencia de los datos. El cliente realizó las pruebas en el sistema las cuales fueron exitosas, con esto los sprints planificados para este sistema están completos.

4. CONCLUSIONES

El sistema web implementado para el control de cotizaciones provee a los usuarios la facilidad de consultar la información de forma inmediata, segura y amigable desde cualquier dispositivo con acceso a internet.

Con la automatización del proceso, el tiempo de aprobación se redujo de una semana a un día, esto representa un ahorro de dinero para la empresa y la posibilidad de generar un mayor número de cotizaciones por vendedor. Las validaciones impuestas en el proceso e implementadas en el sistema han evitado duplicaciones de clientes y cotizaciones consecuentemente la credibilidad de la empresa se ha visto mejorada.

A lo largo del desarrollo de este sistema los requerimientos cambiaron, incluso algunos fueron desaprobados por el cliente. La aplicación de la metodología *Scrum*, permitió realizar revisiones planificadas, calendarizadas y constantes al trabajo realizado por los involucrados en el proyecto, gracias a esto se pudieron prevenir retrasos en las entregas. Las demostraciones y retroalimentaciones efectuadas al cumplir cada iteración ayudaron a mejorar el trabajo en términos de tiempo y calidad. Por este motivo, se concluye que la elección de la metodología *Scrum* fue correcta ya que esta se ajustó a los propósitos formulados en este proyecto.

Ruby y RoR son herramientas vigentes y posicionadas en el mercado de desarrollo web, gracias a la información existente para dichas herramientas el equipo de desarrollo logró implementar satisfactoriamente los requerimientos del sistema con soluciones prácticas y automatizadas, centrándose principalmente en la lógica del negocio.

Las gráficas estadísticas separadas por etapas del tablero digital y el historial de cotizaciones muestran datos que ayudan al usuario aprobador a proyectar el mejor costo de los exámenes en cada cotización, evitando pérdidas de dinero y malos entendidos con los clientes.

Las alertas enviadas por el sistema a los usuarios vía correo electrónico, informan inmediatamente cuando una cotización es aprobada o rechazada, esto permite una comunicación rápida y fluida entre los participantes del proceso de tal forma que puedan aplicar los correctivos necesarios de manera oportuna.

El diseño de interfaces amigables y ajustables proporcionan al usuario la comodidad de acceder al sistema desde cualquier dispositivo móvil, gracias a esto, los vendedores pueden realizar cotizaciones puerta a puerta, sin la necesidad de solicitar una reunión y depender de un ordenador de escritorio.

El módulo “Administrador de contenido” provee al usuario administrador del sistema una interfaz amigable para listar, crear, editar y eliminar los registros de la base de datos sin restricción alguna, gracias a esto es posible solucionar rápidamente errores de contenido como: faltas ortográficas o espacios en blanco innecesarios. Además permite modificar cotizaciones en ausencia del vendedor que la creó, de tal forma se mantiene el seguimiento al cliente.

Las pruebas de carga fueron realizadas por el equipo de trabajo, en un servidor de pruebas alojado en la plataforma web herokuapp.com, las mismas se detallan en el anexo 6.2 Pruebas de carga, dichas pruebas arrojaron los siguientes resultados: el promedio del tiempo de carga fue 1.912 segundos, el mayor tiempo de carga fue 2.425 segundos (módulo de empresas), el mínimo tiempo de carga fue 1.447 segundos (módulo de exámenes). Los resultados cumplen con lo solicitado por el cliente con respecto al rendimiento del sistema detallado en la página 17. Por este motivo se concluye que el sistema fue desarrollado con las técnicas adecuadas como: paginación de resultados, reducción y minimización de archivos *CSS* y *JavaScript*, optimización de imágenes, correcta estructura HTML y por consiguiente cumple con los criterios de rendimiento

5. REFERENCIAS BIBLIOGRÁFICAS

O'Reilly Media. (2014). *PostgreSQL Up and Running.*

El Sistema Operativo GNU. (01 de 01 de 2014). Recuperado el 28 de 12 de 2016, de https://www.gnu.org/philosophy/free-sw.es.html

El pensante. (7 de Abril de 2016). Obtenido de https://educacion.elpensante.com/la-investigacion-aplicada-que-es-y-en-que-consiste/

40deFiebre. (2016). *¿Qué es el Diseño Responsive?.* Obtenido de https://www.40defiebre.com/que-es/diseno-responsive/

AndrewHunt, T. D. (20 de 10 de 2010). *The Pragmatic Programmer .* Stoughton, Massachusetts: Addison-Wesley.

Carles Mateu . (2004). Desarrollo de aplicaciones web. Barcelona: Eureca Media, SL.

Casamayor, A. (Febrero de 2010). Mejora de la Metodologia Scrum. *Proyecto de Titulacion*. Buenos Aires.

Culebro Juáarez, M., Gómez Herrera, W. G., & Torres Sánchez, S. (2006). Software libre vs software propietario. México.

Duckett, J. (2011). *HTML & CSS: Design and Build Web Sites.* Indianapolis: John Wiley & Sons, Inc.

Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development.* Indianapolis: John Wiley & Sons, Inc.

EcuaRed. (15 de 06 de 2016). *Software libre*. Recuperado el 15 de 06 de 2016, de https://www.ecured.cu/Software_libre

EcuaRed. (15 de 06 de 2016). *Software propietario*. Obtenido de https://www.ecured.cu/Software_propietario

es.wikipedia.org. (s.f.). Recuperado el 16 de 04 de 2016, de https://es.wikipedia.org/wiki/PostgreSQL

Foundation for Site. (s.f.). Recuperado el 10 de 05 de 2016, de http://foundation.zurb.com/sites.html

Frain, B. (2012). *Responsive Web Design with HTML5 and CSS3.* Mumbai: Packt Publishing.

Free Software Foundation. (05 de 1998). *Categorías de Software Libre y No Libre.*

Grupo ISSI. (12 de 09 de 2003). *Metodologías Ágiles en el Desarrollo de Software.* Alicante – España: Grupo ISSI.

Hansson, S. D. (2013). *Ruby Agile web development with Rails 4.* United States of America: Pragmatic Bookshelf.

IT VDELGADO. (s.f.). Recuperado el 15 de 08 de 2016, de https://victdelr.wordpress.com/2013/02/19/patron-modelo-vista-controladormodelo-de-4-capas/

Kadlec, T., & Gustafson, A. (2012). *Implementing Responsive Design: Building sites for an anywhere, everywhere web.* Berkeley: New Riders.

Kniberg, H. (2010). *Kanban and Scrum - Making the Most of Both.* Estados Unidos: C4Media Inc.

Lozada, J. (3 de Diciembre de 2014). Investigación Aplicada: Definición, Propiedad Intelectual e Industria. Quito, Pichincha, Ecuador.

Orós Cabello, J. (2012). Guia práctica de XMTML, JavaScript y CSS. Mexico: Alfaomega Grupo Editor.

Palacio, J. (2008). Flexibilidad con Scrum: Principios de diseño e implantacion de campos de Scrum. Autoedicion .

Palacio, J. (2014). *Gestión de proyectos Scrum Manager.* Iubaris Info 4 Media S.L.

Rails Guides. (s.f.). Recuperado el 14 de 07 de 2016, de http://guides.rubyonrails.org/layouts_and_rendering.html

Raphaël Hertzog y Roland Mas. (2015). El manual del Administrador de Debian.

Recursos.cepindalo.es. (s.f.). *Sistema operativo servidor (Ubuntu Server)*. Recuperado el 15 de 04 de 2016, de http://recursos.cepindalo.es/mod/book/tool/print/index.php?id=548

Rendin, R. (06 de 2012). Aplicacion de Scrum al desarrollo del sistema para la calificacion y asignacion de montos y plazos de credito. *Proyecto de Titulacion*. Quito.

Riballo Arenas, M. (2008). *Guía Didáctica. Implantación de aplicaciones informáticas de gestión.* España: RA-MA S.A. Editorial y Publicaciones.

Richard M. Stallman. (2004). *Software libre para una sociedad libre.* Madrid: Traficantes de Sueños.

Ruby Lang, C. (s.f.). *Ruby, el mejor amigo de un desarrollador*. Obtenido de https://www.ruby-lang.org/es/about/

Ruby, S., Thomas, D. and Hansson, D. (2013). *Agile web development with Rails 4.* United States of America: Pragmatic Bookshelf.

Salesforce.com, ©. (s.f.). *Heroku dev center*. Recuperado el 15 de 08 de 2016, de https://devcenter.heroku.com/categories/ruby

Sam Ruby, Dave Tomas. (2009). Agile web development with Rails 3rd edition. En *The architecture of Rails applications* (pág. 11). Dallas, Texas: The pragmatic programers.

Santimacnet, B. (4 de Noviembre de 2010). *Curso Gratis Scrum Dia a Dia*. Obtenido de https://santimacnet.wordpress.com/2010/11/04/curso-gratis-scrum-dia-a-dia

SANTIMACNET'S BLOG. (s.f.). Recuperado el 20 de 09 de 2016, de https://santimacnet.wordpress.com/2010/11/04/curso-gratis-scrum-dia-a-dia/

Schwaber Ken, S. J. (Octubre de 2011). *Scrum.org.* Obtenido de La guia Definitiva de Scrum: Las Reglas del Juego: http://www.scrumguides.org/scrum-guide.html#artifacts-productbacklog

Thomas, D. (2005). *Programming Ruby.* Dallas, Texas: The Pragmatic Bookshelf.

Thomas, D. (2014). *Programming Elixir.* United States of America: Lynn Beighley.

Valim, J. (2014). *Crafting Rails 4 applications.* Frisco, TX: Pragmatic Programmers.

VDelagado, I. (1 de Febrero de 2013). *IT VDelagado*. Obtenido de https://victdelr.wordpress.com/

WebPageTest. (s.f.). Recuperado el 15 de 08 de 2016, de https://www.webpagetest.org/about

Wikilibros. (s.f.). *Wikilibros*. Recuperado el 01 de 06 de 2016, de https://es.wikibooks.org/wiki/Fundamentos_de_programaci%C3%B3n/Herramientas_de_desarrollo

ZURB FOUNDATION. (s.f.). *ZURB Foundation*. Recuperado el 15 de Junio de 2016, de http://foundation.zurb.com/

6. ANEXOS

6.1. Historias de Usuario

Tabla 48 – Historia de Usuario 1

Historia de Usuario	
Número: 1	**Rol:** Vendedor-Comercial, Aprobador, Administrador del Sistema
Nombre Historia: Registrar empresas	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 60	**Iteración asignada:** Módulo Empresas
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los clientes potenciales serán registrados en el sistema con los siguientes campos: nombre, ruc, dirección, descripción y sector.	
Validación: El nombre de la empresa y su RUC serán únicos para evitar duplicidad de datos. El campo sector será una lista desplegable con las siguientes opciones: educación, financiero, industrial y ninguno. Los vendedores no podrán eliminar las empresas ya registradas. Al editar la información de una empresa, no podrán modificar ni el RUC ni el nombre.	

Tabla 49 – Historia de Usuario 2

Historia de Usuario	
Número: 2	**Rol:** Vendedor-Comercial, Aprobador, Administrador del Sistema
Nombre Historia: Contactos de empresas	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 40	**Iteración asignada:** Módulo Empresas
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Para facilitar la comunicación con las empresas los vendedores solicitarán el nombre y correo electrónico de las 3 personas siguientes: Contacto de empresa	

Doctor de empresa Responsable de Facturación en empresa Es posible que las empresas en un primer contacto se nieguen a brindar esta información por esta razón estos campos serán opcionales.
Validación: Los correos electrónicos deberán ser válidos, si no cumplen con el requerimiento, serán descartados.

Tabla 50 – Historia de Usuario 3

Historia de Usuario	
Número: 3	**Rol:** Vendedor-Comercial, Aprobador, Administrador del Sistema
Nombre Historia: Lista de empresas	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 50	**Iteración asignada:** Módulo Empresas
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los vendedores podrán acceder a la lista de empresas que han registrado en el sistema, las cuales serán ordenadas en función de la fecha de creación en forma ascendente.	
Validación: Un vendedor podrá únicamente leer la información de las empresas registradas por otros vendedores mas no podrá editar dicha información.	

Tabla 51 – Historia de Usuario 4

Historia de Usuario	
Número: 4	**Rol:** Vendedor-Comercial, Aprobador, Administrador del Sistema
Nombre Historia: Registrar cotizaciones	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 50	**Iteración asignada:** Módulo Cotizaciones
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los vendedores podrán registrar sus cotizaciones en un formulario que permitirá	

seleccionar los exámenes médicos que el cliente solicite. La cotización tendrá la posibilidad de calcular el precio de cada examen por tres niveles: ciudad (obligatorio), provincia (opcional), unidades móviles (opcional), el valor unitario del examen será multiplicado por el total de empleados. Las cotizaciones tendrán fecha de vigencia y descripción como campos obligatorios. La cotización presentará subtotales por cada uno de los niveles antes mencionados y un gran total de la cotización. El cálculo de los subtotales y el gran total debe ser instantáneo, de tal forma que el vendedor comunique al cliente el valor exacto antes de registrar la cotización.
Validación: Para registrar una cotización es necesario especificar la empresa que requiere los servicios de ECUAMERICAN. En el formulario de registro se mostrará la información de la empresa. Cada examen solicitado por el cliente mostrará su valor unitario para cada nivel: ciudad, provincia, unidades móviles. El formulario deberá ajustarse a los diferentes tamaños de pantalla de los dispositivos que usarán los usuarios del sistema los cuales pueden ser: computadores de escritorio, laptops, tabletas y celulares inteligentes.

Tabla 52 – Historia de Usuario 5

Historia de Usuario	
Número: 5	**Rol:** Vendedor-Comercial, Aprobador, Administrador del Sistema
Nombre Historia: Listar cotizaciones	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 30	**Iteración asignada:** Módulo Cotizaciones
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los vendedores podrán acceder a la lista de cotizaciones registradas por ellos en el sistema.	
Validación: Los vendedores y el aprobador no podrán ver las cotizaciones registradas por el administrador del sistema.	

Tabla 53 – Historia de Usuario 6

Historia de Usuario	
Número: 6	**Rol:** Aprobador
Nombre Historia: Aprobar cotización	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Mediana
Puntos estimados: 50	**Iteración asignada:** Módulo Cotizaciones
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los usuarios: aprobador y administrador del sistema están encargados de aprobar las cotizaciones, podrán acceder a todas las cotizaciones registradas en el sistema ordenadas por fecha de creación de forma descendente. Una vez que el usuario aprobador verifique que una cotización cumple con los requerimientos de la empresa podrá cambiar el estado de dicha cotización a aprobado. Una vez aprobada una cotización se enviará una notificación al usuario que creó la misma.	
Validación: Se deberá registrar la fecha de aprobación y el usuario aprobador de forma automática.	

Tabla 54 – Historia de Usuario 7

Historia de Usuario	
Número: 7	**Rol:** Usuario
Nombre Historia: Inicio de sesión	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Mediana
Puntos estimados: 70	**Iteración asignada:** Módulo Usuarios
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: El sistema deberá mostrar una pantalla en la cual los usuarios podrán iniciar sesión en el sistema ingresando los siguientes datos: email, clave. Si el usuario ingresa la información incorrecta, se mostrará un mensaje de alerta informando del error.	

Cuando el usuario haya ingresado al sistema, el mismo podrá acceder a un link donde podrá editar su perfil de usuario, también podrá acceder desde cualquier pantalla del sistema a un link para cerrar o terminar la sesión.
Validación: Los usuarios vendedor-comercial y aprobador podrán únicamente modificar sus datos como email, clave y nombre de usuario, pero no podrán crear otro usuario ni eliminar. Las cuentas de usuario serán creadas por el administrador del sistema.

Tabla 55 – Historia de Usuario 8

Historia de Usuario	
Número: 8	**Rol:** Vendedor comercial
Nombre Historia: Panel digital	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Mediana
Puntos estimados: 20	**Iteración asignada:** Módulo panel digital
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: La pantalla panel digital será mostrada en la ruta raíz del sistema, es decir que será la primera pantalla que podrá ver el vendedor después de iniciar sesión. La pantalla de panel digital mostrará tres bloques los cuales serán: **Estadísticas:** Tres gráficos estadísticos tipo barras de los últimos 6 meses; cotizaciones aprobadas, cotizaciones por aprobar y cotizaciones rechazadas pertenecientes al vendedor. **Cotizaciones por aprobar:** Lista de las 10 últimas cotizaciones con estado no aprobado pertenecientes al vendedor. **Últimas empresas registradas:** Lista de las 10 últimas empresas creadas por el vendedor.	
Validación: Los vendedores podrán ver la información registrada únicamente por ellos.	

Tabla 56 – Historia de Usuario 9

Historia de Usuario	
Número: 9	**Rol:** Aprobador
Nombre Historia: Panel digital	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Mediana
Puntos estimados: 20	**Iteración asignada:** Módulo panel digital
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: La pantalla panel digital será mostrada en la ruta raíz del sistema, es decir que será la primera pantalla que podrá ver el aprobador después de iniciar sesión. La pantalla de panel digital mostrará tres bloques los cuales serán: **Estadísticas:** Tres gráficos estadísticos tipo barras de los últimos 6 meses; cotizaciones aprobadas, cotizaciones por aprobar y cotizaciones rechazadas pertenecientes al aprobador. **Cotizaciones por aprobar:** Lista de las 10 últimas cotizaciones con estado no aprobado pertenecientes al aprobador. **Últimas empresas registradas:** Lista de las 10 últimas empresas creadas por el aprobador.	
Validación: Los aprobadores verán la información registrada únicamente por ellos.	

Tabla 57 – Historia de Usuario 10

Historia de Usuario	
Número: 10	**Rol:** Administrador del Sistema
Nombre Historia: Panel digital	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Mediana
Puntos estimados: 20	**Iteración asignada:** Módulo panel digital
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: La pantalla panel digital será mostrada en la ruta raíz del sistema, es decir que será la primera pantalla que podrá ver el aprobador después de iniciar sesión. La pantalla de panel digital mostrará tres bloques los cuales serán:	

Estadísticas: Tres gráficos estadísticos tipo barras de los últimos 6 meses; cotizaciones aprobadas, cotizaciones por aprobar y cotizaciones rechazadas de todos los usuarios. **Cotizaciones por aprobar:** Lista de las 10 últimas cotizaciones con estado no aprobado de todos los usuarios. **Últimas empresas registradas:** Lista de las 10 últimas empresas creadas por todos los usuarios.
Validación: El usuario administrador del sistema podrá ver la información de todos los usuarios.

Tabla 58 – Historia de Usuario 11

Historia de Usuario	
Número: 11	**Rol:** Vendedor Comercial, Aprobador, Administrador del Sistema
Nombre Historia: Buscar cotización	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Alta
Puntos estimados: 30	**Iteración asignada:** Módulo panel digital
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los usuarios podrán buscar entre todas las cotizaciones registradas en el sistema, esta opción estará disponible en el menú principal.	
Validación: El campo de búsqueda será “Descripción”. Los botones en la columna “Acciones” de los resultados de búsqueda estarán disponibles únicamente para los propietarios de cada registro, el administrador del sistema podrá ver dichos botones para todos los registros.	

Tabla 59– Historia de Usuario 12

Historia de Usuario	
Número: 12	**Rol:** Administrador del Sistema
Nombre Historia: Menú principal	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Media
Puntos estimados: 20	**Iteración asignada:** Diseño y navegabilidad
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Con el fin de acceder de forma rápida a las pantallas del sistema existirá un menú principal que mostrará los links de los módulos del sistema en el siguiente orden: Panel Digital, Empresas, Cotizaciones y Exámenes.	
Validación:	

Tabla 60 – Historia de Usuario 13

Historia de Usuario	
Número: 13	**Rol:** Vendedor Comercial, Aprobador
Nombre Historia: Vistas responsive	
Prioridad en el negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 100	**Iteración asignada:** Diseño y navegabilidad
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Todas las pantallas del sistema deberán adaptarse a los tamaños de los dispositivos tales como: Ordenadores de escritorio, laptops, tablets y teléfonos inteligentes.	
Validación:	

Tabla 61 – Historia de Usuario 14

Historia de Usuario	
Número: 14	**Rol:** Administrador del sistema
Nombre Historia: Crear exámenes	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 30	**Iteración asignada:** Módulo Exámenes
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los exámenes son registros que permiten calcular el valor total de la cotización.	
Validación: Código y un nombre son campos requeridos con valores únicos.	

Tabla 62 – Historia de Usuario 15

Historia de Usuario	
Número: 15	**Rol:** Administrador del sistema
Nombre Historia: Editar exámenes	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 30	**Iteración asignada:** Módulo Exámenes
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Los costos de los exámenes podrán ser modificados por el Administrador del sistema de acuerdo a las necesidades de Laboratorio EcuaAmerican respaldado por número identificador de la circular correspondiente. La circular es enviada por el departamento de contabilidad.	
Validación: Los exámenes no podrán ser eliminados, existirá un campo que indique si el examen está habilitado.	

Tabla 63 – Historia de Usuario 16

Historia de Usuario	
Número: 16	**Rol:** Administrador del sistema
Nombre Historia: Listar exámenes	
Prioridad en el negocio: Media	**Riesgo en desarrollo:** Baja
Puntos estimados: 20	**Iteración asignada:** Módulo Exámenes
Programador responsable: Marcelo Toapanta	**Responsable QA:** Bárbara Barragán
Descripción: Lista ordenada por fecha de creación, mostrará los 3 valores de cada examen.	
Validación: Los valores se mostrarán en formato de moneda con precisión de dos decimales, ejemplo: $10.50	

6.2. Pruebas de carga

El equipo de trabajo realizó las pruebas de tiempos de carga con un servidor de pruebas alojado en Heroku y con la ayuda del sitio WebPageTest.

Heroku es una plataforma que brinda el servicio de computación en la nube que facilita el despliegue y escalado de aplicaciones Ruby entre otras. Ya sea que prefiera marcos como Sinatra o Rails, el servicio es escalable de acuerdo a nuestras necesidades. Para nuestras pruebas se utilizó un servidor con una capacidad de almacenamiento de 500MB y con las siguientes características procesamiento: (Salesforce.com)

Dyno Type	Sleeps	Professional Features	Memory (RAM)	CPU Share	Dedicated	Compute (1)
free	yes	no	512MB	1x	no	1x-4x

Tabla 64 – Característica de procesamiento herokuapp (Salesforce.com)

Al ser un servicio gratuito, el mismo permite hasta 20 conexiones simultáneas y una base de datos Postgres limitado a un máximo de 10.00 registros. Por otro lado permite la sincronización de repositorios remotos como GitHub y GitLab sin límenes de despliegue de cogido hacia el servidor. (Salesforce.com)

WebPagetest es un proyecto de código abierto que es desarrollado y apoyado principalmente por Google, es una herramienta que fue desarrollada originalmente por AOL para su uso interno y fue de código abierto en 2008 bajo una licencia BSD. La versión en línea en www.webpagetest.org se ejecuta para el beneficio de la comunidad de rendimiento con varias empresas y personas que proporcionan la infraestructura de pruebas en todo el mundo, es una herramienta de gran ayuda ya que permite mostrar las pruebas realizadas en tablas, gráficas de barras, gráficas de pastel. (WebPageTest)

Panel Digital

								Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Visually Complete	Speed Index	Interactive (beta)	Result (error code)	Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 2)	1.722s	0.293s	1.788s	2.900s	1984	> 4.381s	0	1.722s	8	483 KB	1.995s	9	484 KB

Interactive (beta)	RUM First Paint	domInteractive	domContentLoaded	loadEvent
> 4.381s	1.490s	1.610s	1.610s - 1.678s (0.068s)	1.704s - 1.717s (0.013s)

Figura 71 – Tiempos de carga (Panel digital)

En la Figura 71 podemos observar que el tiempo de carga (*load time*) para la página del panel digital es 1.722 segundos, es decir el tiempo en cargar todos los elementos de la página, sin embargo tomó 1.995 segundos en cargar, procesar y mostrar completamente la página, la diferencia 0.273 es el tiempo que tomó en procesar los elementos dinámicos, es decir reglas *CSS* y funciones *JavaScript*.

Before Start Render | Before On Load | After On Load

Request Details

#	Resource	Content Type	Request Start	DNS Lookup	Initial Connection	Time to First Byte	Content Download	Bytes Downloaded	Error/Status Code	IP
1	http://american-coti...acion.herokuapp.com/	text/html	0.134 s	31 ms	33 ms	159 ms	63 ms	32.6 KB	200	23.23.109.78
2	http://american-coti...ce17cc24c11332f4.css	text/css	0.314 s	-	32 ms	44 ms	212 ms	77.2 KB	200	23.23.109.78
3	http://american-coti...22e5d62db85420fd3.js	application/javascript	0.357 s	-	42 ms	38 ms	730 ms	368.6 KB	200	23.23.109.78
4	http://american-coti...m/images/body-bg.png	image/png	1.292 s	-	-	43 ms	-	0.8 KB	200	23.23.109.78
5	http://american-coti...images/dashboard.png	image/png	1.298 s	-	-	39 ms	-	0.9 KB	200	23.23.109.78
6	http://american-coti...m/images/company.png	image/png	1.3 s	-	-	48 ms	-	0.9 KB	200	23.23.109.78
7	http://american-coti...images/quotation.png	image/png	1.334 s	-	34 ms	45 ms	-	0.9 KB	200	23.23.109.78
8	http://american-coti....com/images/exam.png	image/png	1.335 s	-	35 ms	50 ms	-	0.9 KB	200	23.23.109.78
9	http://american-coti...m/images/favicon.ico	image/vnd.microsoft.icon	1.866 s	-	-	129 ms	-	0.8 KB	200	23.23.109.78

Figura 72 – Detalle de petición (Panel digital)

Como se puede observar en la Figura 72 para la petición del panel digital, se obtuvo 9 recursos en los cuales se encuentran las imágenes, archivos *CSS*, *JavaScript* y texto *HTML* necesarios para mostrar la página.

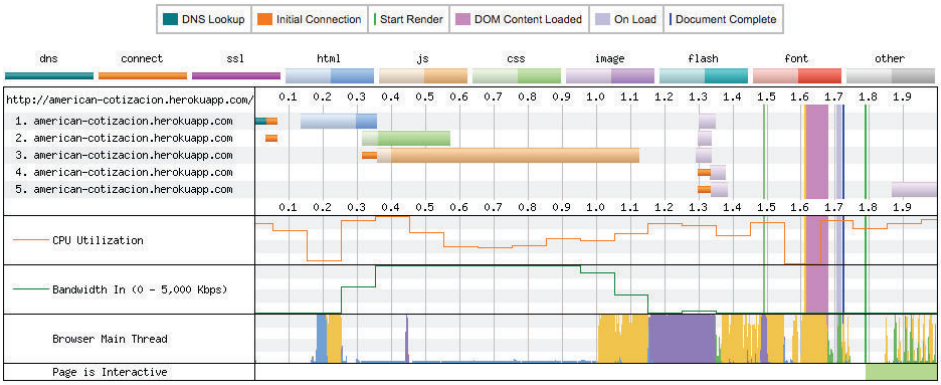
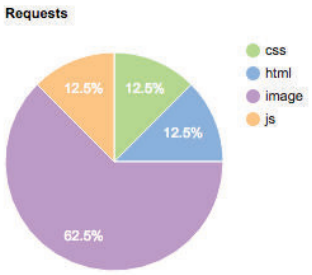
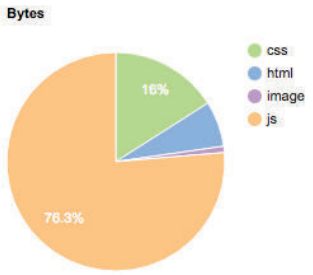


Figura 73 – Vista de conexión (Panel digital)

La Figura 73 muestra en orden cronológico la carga de los elementos con sus tipos y tiempos, los mismo completan la petición.





MIME Type	Requests
image	5
css	1
html	1
js	1
flash	0
font	0
other	0

MIME Type	Bytes	Uncompressed
js	377,424	377,172
css	79,083	78,846
html	33,348	32,418
image	4,565	3,385
flash	0	0
font	0	0
other	0	0

Figura 74 – Recursos (Panel digital)

La Figura 74 muestra en porcentajes los tipos de recurso con sus respectivos tamaños en bytes. Podemos observar que existen 5 imágenes para esta petición, sin embargo el archivo *JavaScript* es el que contiene más *Bytes*.

Empresas

	Load Time	First Byte	Start Render	Visually Complete	Speed Index	Interactive (beta)	Result (error code)	Document Complete			Fully Loaded		
								Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 1)	2.425s	0.640s	2.771s	2.800s	2800	> 4.850s	0	2.425s	9	475 KB	2.451s	9	476 KB

Interactive (beta)	RUM First Paint	domInteractive	domContentLoaded	loadEvent
> 4.850s	2.357s	2.077s	2.077s - 2.306s (0.229s)	2.375s - 2.389s (0.014s)

Figura 75 – Tiempos de carga (Empresas)

En la Figura 75 al igual que las pruebas del módulo anterior, podemos observar que el tiempo de carga (load time) para la página de empresas es 2.425 segundos, es decir el tiempo en cargar todos los elementos de la página, sin embargo tomó 2.451 segundos en cargar, procesar y mostrar completamente la página, la diferencia 0.026 es el tiempo que tomó en procesar los elementos dinámicos, es decir reglas *CSS* y funciones *JavaScript*.

Before Start Render | Before On Load | After On Load

Request Details

#	Resource	Content Type	Request Start	DNS Lookup	Initial Connection	Time to First Byte	Content Download	Bytes Downloaded	Error/Status Code	IP
1	http://american-coti...okuapp.com/companies	text/html	0.155 s	27 ms	33 ms	485 ms	65 ms	24.5 KB	200	50.17.235.124
2	http://american-coti...ce17cc24c11332f4.css	text/css	0.666 s	-	33 ms	460 ms	144 ms	77.2 KB	200	50.17.235.124
3	http://american-coti...22e5d62db85420fd3.js	application/javascript	0.707 s	-	-	566 ms	632 ms	368.6 KB	200	50.17.235.124
4	http://american-coti...m/images/body-bg.png	image/png	2.088 s	-	-	42 ms	-	0.8 KB	200	50.17.235.124
5	http://american-coti...images/dashboard.png	image/png	2.091 s	-	-	57 ms	-	0.9 KB	200	50.17.235.124
6	http://american-coti...m/images/company.png	image/png	2.093 s	-	1105 ms	62 ms	-	0.9 KB	200	50.17.235.124
7	http://american-coti...images/quotation.png	image/png	2.129 s	-	33 ms	41 ms	-	0.9 KB	200	50.17.235.124
8	http://american-coti....com/images/exam.png	image/png	2.13 s	-	35 ms	45 ms	-	0.9 KB	200	50.17.235.124
9	http://american-coti...m/images/favicon.ico	image/vnd.microsoft.icon	2.404 s	-	-	47 ms	-	0.8 KB	200	50.17.235.124

Figura 76 – Detalle de petición (Empresas)

La Figura 76 muestra que la petición retornó 9 recursos en los cuales se encuentran las imágenes, archivos *CSS*, *JavaScript* y texto *HTML* necesarios para mostrar la página.

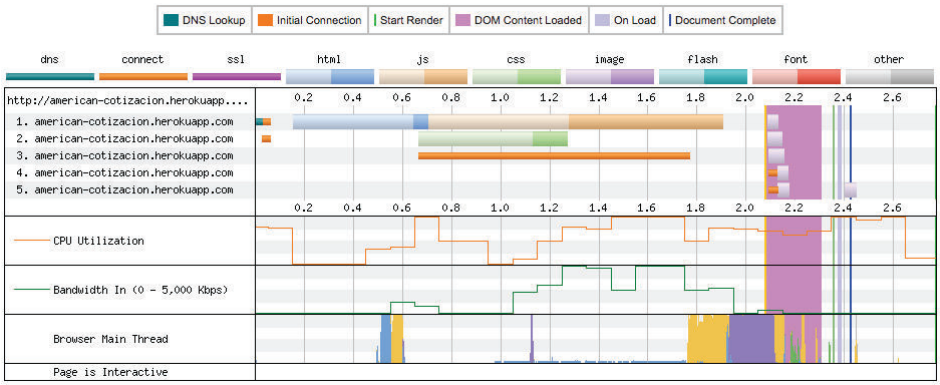


Figura 77 – Vista de conexión (Empresas)

La Figura 77 muestra los elementos cargados en esta página con sus tipos y tiempos ordenados cronológicamente.

MIME Type	Requests
image	5
css	1
html	1
js	1
flash	0
font	0
other	0

MIME Type	Bytes	Uncompressed
js	377,424	377,172
css	79,083	78,846
html	25,082	24,152
image	4,565	3,385
flash	0	0
font	0	0
other	0	0

Figura 78 – Recursos (Empresa)

La Figura 78 muestra en porcentajes los tipos de recurso con sus respectivos tamaños en bytes al igual que en el módulo anterior el archivo JavaScript es el que contiene más *Bytes*.

Cotizaciones

								Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Visually Complete	Speed Index	Interactive (beta)	Result (error code)	Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 1)	1.664s	0.282s	2.067s	2.100s	2100	> 3.830s	0	1.664s	9	467 KB	1.752s	9	467 KB

Interactive (beta)	RUM First Paint	domInteractive	domContentLoaded	loadEvent
> 3.830s	1.605s	1.281s	1.281s - 1.536s (0.255s)	1.624s - 1.637s (0.013s)

Figura 79 – Tiempos de carga (Cotizaciones)

En la Figura 79 podemos observar que el tiempo de carga (*load time*) para la página de cotizaciones es 1.664 segundos, es decir el tiempo en cargar todos los elementos de la página, sin embargo tomó 1.752 segundos en cargar, procesar y mostrar completamente la página, la diferencia 0.088 es el tiempo que tomó en procesar los elementos dinámicos, es decir reglas *CSS* y funciones *JavaScript*.

Before Start Render | Before On Load | After On Load

Request Details

#	Resource	Content Type	Request Start	DNS Lookup	Initial Connection	Time to First Byte	Content Download	Bytes Downloaded	Error/Status Code	IP
1	http://american-coti.../quotations?user=all	text/html	0.168 s	29 ms	56 ms	114 ms	35 ms	16.4 KB	200	54.243.91.79
2	http://american-coti...ce17cc24c11332f4.css	text/css	0.316 s	-	57 ms	41 ms	267 ms	77.2 KB	200	54.243.91.79
3	http://american-coti...22e5d62db85420fd3.js	application/javascript	0.319 s	-	-	58 ms	731 ms	368.6 KB	200	54.243.91.79
4	http://american-coti...m/images/body-bg.png	image/png	1.294 s	-	-	42 ms	-	0.8 KB	200	54.243.91.79
5	http://american-coti...images/dashboard.png	image/png	1.295 s	-	-	44 ms	-	0.9 KB	200	54.243.91.79
6	http://american-coti...m/images/company.png	image/png	1.332 s	-	34 ms	49 ms	-	0.9 KB	200	54.243.91.79
7	http://american-coti...images/quotation.png	image/png	1.335 s	-	34 ms	56 ms	-	0.9 KB	200	54.243.91.79
8	http://american-coti....com/images/exam.png	image/png	1.335 s	-	37 ms	68 ms	-	0.9 KB	200	54.243.91.79
9	http://american-coti...m/images/favicon.ico	image/vnd.microsoft.icon	1.657 s	-	-	95 ms	-	0.8 KB	200	54.243.91.79

Figura 80 – Detalle de petición (Cotizaciones)

La Figura 80 muestra que la petición retornó 9 recursos en los cuales se encuentran las imágenes, archivos *CSS* y *JavaScript* y texto *HTML* necesarios para mostrar la página.

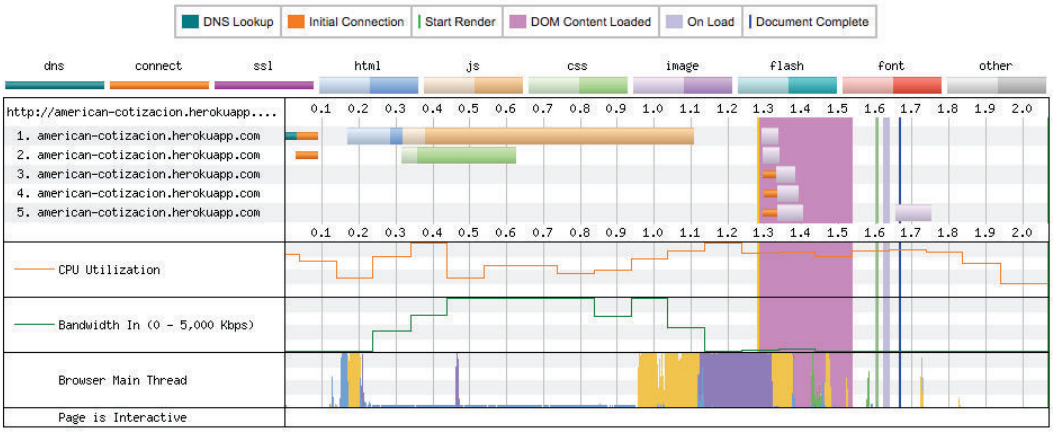
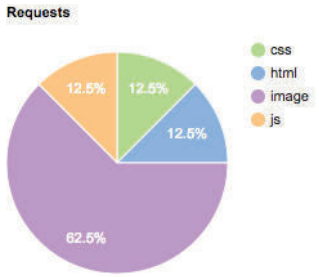
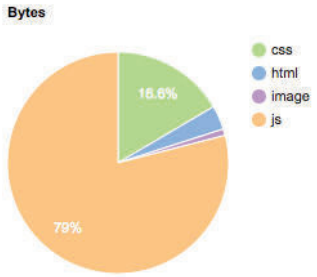


Figura 81 – Vista de conexión (Cotizaciones)

La Figura 81 muestra el orden cronológico de carga de los elementos con sus tipos y tiempos para completar la petición.





MIME Type	Requests
image	5
css	1
html	1
js	1
flash	0
font	0
other	0

MIME Type	Bytes	Uncompressed
js	377,424	377,172
css	79,083	78,846
html	16,749	15,819
image	4,565	3,385
flash	0	0
font	0	0
other	0	0

Figura 82 – Recursos (Cotizaciones)

La Figura 82 muestra en porcentajes los tipos de recurso que utiliza la página con sus respectivos tamaños en bytes. Como en los módulos anteriores la petición retorna 5 imágenes, pero el archivo *JavaScript* es el que contiene más *Bytes*.

Módulo: Exámenes

								Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Visually Complete	Speed Index	Interactive (beta)	Result (error code)	Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 1)	1.439s	0.244s	1.638s	1.800s	1618	> 3.845s	0	1.439s	9	460 KB	1.477s	9	461 KB

Interactive (beta)	RUM First Paint	domInteractive	domContentLoaded	loadEvent
> 3.845s	1.388s	1.190s	1.190s - 1.363s (0.173s)	1.402s - 1.414s (0.012s)

Figura 83 – Tiempos de carga (exámenes)

En la Figura 83 podemos observar que el tiempo de carga (*load time*) para la página de cotizaciones es 1.439 segundos, es decir el tiempo en cargar todos los elementos de la página, sin embargo tomó 1.477 segundos en cargar, procesar y mostrar completamente la página, la diferencia 0.038 es el tiempo que tomó en procesar los elementos dinámicos, es decir reglas *CSS* y funciones *JavaScript*.

Before Start Render | Before On Load | After On Load

Request Details

#	Resource	Content Type	Request Start	DNS Lookup	Initial Connection	Time to First Byte	Content Download	Bytes Downloaded	Error/Status Code	IP
1	http://american-coti....herokuapp.com/exams	text/html	0.164 s	27 ms	33 ms	80 ms	4 ms	9.6 KB	200	54.225.246.159
2	http://american-coti...ce17cc24c11332f4.css	text/css	0.264 s	-	-	67 ms	269 ms	77.2 KB	200	54.225.246.159
3	http://american-coti...22e5d62db85420fd3.js	application/javascript	0.268 s	-	36 ms	42 ms	750 ms	368.6 KB	200	54.225.246.159
4	http://american-coti...m/images/body-bg.png	image/png	1.2 s	-	-	41 ms	-	0.8 KB	200	54.225.246.159
5	http://american-coti...images/dashboard.png	image/png	1.204 s	-	-	40 ms	-	0.9 KB	200	54.225.246.159
6	http://american-coti...m/images/company.png	image/png	1.24 s	-	34 ms	39 ms	-	0.9 KB	200	54.225.246.159
7	http://american-coti...images/quotation.png	image/png	1.241 s	-	35 ms	45 ms	-	0.9 KB	200	54.225.246.159
8	http://american-coti....com/images/exam.png	image/png	1.241 s	-	35 ms	53 ms	-	0.9 KB	200	54.225.246.159
9	http://american-coti...m/images/favicon.ico	image/vnd.microsoft.icon	1.437 s	-	-	40 ms	-	0.8 KB	200	54.225.246.159

Figura 84 – Detalle de petición (exámenes)

De igual forma que los anteriores módulos, se pude observar en la Figura 84 que la petición retornó 9 recursos en los cuales se encuentra las imágenes, archivos *CSS*, *JavaScript* y texto *HTML* necesario para mostrar la página.

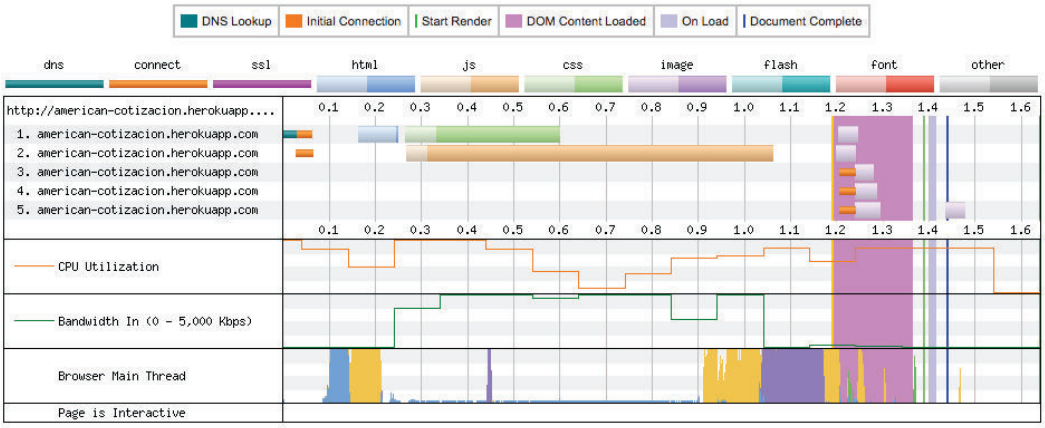
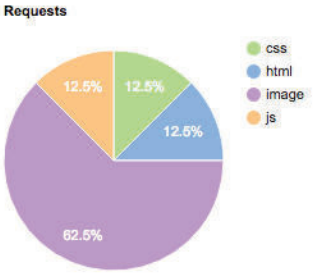
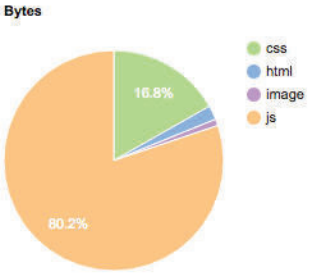


Figura 85 – Vista de conexión (exámenes)

La Figura 85 muestra en orden cronológico los elementos necesarios para cargar la página, cada elemento tiene un color dependiendo del tipo de elemento, adicionalmente podemos observar los tiempos en los que se carga cada uno.





MIME Type	Requests
image	5
css	1
html	1
js	1
flash	0
font	0
other	0

MIME Type	Bytes	Uncompressed
js	377,424	377,172
css	79,083	78,846
html	9,788	8,859
image	4,565	3,385
flash	0	0
font	0	0
other	0	0

Figura 86 – Recursos (exámenes)

En la Figura 86 podemos observar en porcentaje los tipos de elementos cargados en la página, de igual forma que los anteriores módulos existe un mayor número de imágenes cargadas, sin embargo el archivo *JavaScript* es el que tiene mayor cantidad de *Bytes*