



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN SISTEMAS INFORMÁTICOS

TEMA:

**DESARROLLO DEL SISTEMA PARA GESTIÓN DE EXÁMENES DE
LABORATORIO CON RUBYONRAILS PARA LA EMPRESA “ECUA-
AMERICAN”**

AUTOR:

GONZALO ALONSO MOLINA QUIMBIULCO

TUTOR:

MSc. Ing. HENRY RECALDE

QUITO, ECUADOR

2019

DECLARACIÓN DE AUTORÍA

El documento de tesis con título: “DESARROLLO DEL SISTEMA PARA GESTIÓN DE ÓRDENES DE EXÁMENES DE LABORATORIO CON RUBYONRAILS PARA LA EMPRESA ECUA-AMERICAN.”, ha sido desarrollado por el señor GONZALO ALONSO MOLINA QUIMBIULCO con C.C. No. 1726244526 persona que posee los derechos de autoría y responsabilidad, restringiéndose la copia o utilización de la información de esta tesis sin previa autorización.

GONZALO ALONSO MOLINA QUIMBIULCO

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación certifico:

Que el trabajo de titulación **“DESARROLLO DEL SISTEMA PARA GESTIÓN DE ÓRDENES DE EXÁMENES DE LABORATORIO CON RUBYONRAILS PARA LA EMPRESA “ECUA-AMERICAN”**, presentado por GONZALO ALONSO MOLINA QUIMBIULCO, estudiante de la Carrera Ingeniería en Sistemas Informáticos, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., agosto 2019

TUTOR

MSc. Ing. Henry Recalde.

AGRADECIMIENTOS

Agradezco este trabajo a mi familia quienes incondicionalmente me apoyaron en todo el proceso universitario, así también a los señores docentes quienes me guiaron año tras año para lograr el objetivo deseado.

A la Universidad Tecnológica de Israel por la apertura brindada a los estudiantes y permitimos cumplir un objetivo más en nuestra vida profesional.

DEDICATORIA

El presente trabajo está dedicado a las personas quienes me han apoyado y han influenciado en mi vida, dándome consejos y haciéndome una persona de bien, con todo el afecto se lo dedico a mi madre la señora María Esperanza

TABLA DE CONTENIDOS

Antecedentes de la situación objeto de estudio	1
Planteamiento del problema.....	2
Justificación	2
Objetivos.....	3
General.....	3
Objetivos específicos	3
Descripción de los capítulos	4
1 CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 Estado del arte.....	5
1.2 Lógica del negocio.....	7
1.3 Alternativas de solución.....	10
2 CAPÍTULO 2. MARCO METODOLÓGICO	11
2.1 Tipo de investigación.....	11
2.1.1 Metodología seleccionada.....	11
2.2 Recopilación de información	12
2.2.1 Técnicas de recopilación de información	12
3 CAPÍTULO 3. PROPUESTA	14
3.1 Diagramas de procesos	14
3.2 Factibilidad técnica.....	16
3.3 Factibilidad operacional.....	16
3.4 Factibilidad económica - financiera.....	17
Los costos detallados en la tabla anterior serán cubiertos por la empresa.....	17
3.5 Especificación de requerimientos	17
3.5.1 Ámbito del software.....	17
3.5.2 Funciones del producto	17

3.5.3	Características de los usuarios del sistema	20
3.5.4	Restricciones de desarrollo	21
3.5.5	Requisitos.....	21
4	CAPÍTULO 4. IMPLEMENTACIÓN	25
4.1	Diseño	25
4.2	Diagrama de base de datos.....	46
4.3	Diagrama de la arquitectura del sistema	47
4.4	Diseño de interfaces	47
4.5	Estándares de programación utilizados.....	54
4.6	Pruebas.....	55
4.6.1	Pruebas de funcionalidad (Aceptación de usuario).....	55
4.6.2	Pruebas de rendimiento (Aceptación técnica)	56
4.6.3	Pruebas de carga y estrés (Aceptación técnica)	56
4.7	Implementación	57
4.7.1	Plan de implementación.....	57
4.7.2	Requerimientos de implementación.....	57
4.7.3	Manual de usuario.....	58
4.7.4	Manual técnico.....	58
4.7.5	Plan de capacitación.....	58
	CONCLUSIONES	60
	RECOMENDACIONES.....	61
	REFERENCIAS BIBLIOGRÁFICAS	62
	ANEXOS	1

LISTA DE FIGURAS

<i>Figura 1.1.</i> Patrón de arquitectura MVC	8
<i>Figura 3.1.</i> Diagrama de proceso manual.....	14
<i>Figura 3.2.</i> Diagrama de proceso automatizado.....	15
<i>Figura 4.1.</i> Planificación inicial – Sprint 0	27
<i>Figura 4.2.</i> Resultados – Sprint 0.....	27
<i>Figura 4.9.</i> Planificación Inicial – Sprint 1	29
<i>Figura 4.10.</i> Resultados – Sprint 1	30
<i>Figura 4.11.</i> Resultados – Inicio de sesión.....	30
<i>Figura 4.12.</i> Resultados – Formulario usuario.....	31
<i>Figura 4.13.</i> Resultados – Lista usuarios	31
<i>Figura 4.17.</i> Planificación Inicial – Sprint 2	33
<i>Figura 4.18.</i> Resultados – Sprint 2.....	34
<i>Figura 4.19.</i> Resultados – Formulario de exámenes	34
<i>Figura 4.20.</i> Resultados – Lista de exámenes	35
<i>Figura 4.21.</i> Resultados – Formulario de Tubos	35
<i>Figura 4.22.</i> Resultados – Lista de Tubos	36
<i>Figura 4.25.</i> Planificación Inicial – Sprint 3	39
<i>Figura 4.26.</i> Resultados – Sprint 3	39
<i>Figura 4.27.</i> Resultados – Formulario de órdenes.....	40
<i>Figura 4.28.</i> Resultados – Lista de órdenes.....	40
<i>Figura 4.29.</i> Planificación Inicial – Sprint 4	41
<i>Figura 4.30.</i> Resultados – Sprint 4	42
<i>Figura 4.31.</i> Resultados – Lista de resultados de órdenes.....	42
<i>Figura 4.33.</i> Planificación Inicial – Sprint 5	44
<i>Figura 4.34.</i> Resultados – Sprint 5	44

<i>Figura 4.35.</i> Resultados – Menú de navegación	45
<i>Figura 4.36.</i> Diagrama base de datos	46
<i>Figura 4.37.</i> Diagrama cliente-servidor	47
<i>Figura 4.6.</i> Iniciar sesión	47
<i>Figura 4.7.</i> Usuario médico	48
<i>Figura 4.8.</i> Usuario recepcionista/laboratorista	49
<i>Figura 4.14.</i> Formulario examen	50
<i>Figura 4.15.</i> Formulario de tubos	50
<i>Figura 4.16.</i> Lista de exámenes	51
<i>Figura 4.23.</i> Formulario de órdenes	52
<i>Figura 4.24.</i> Lista de órdenes	53
<i>Figura 4.32.</i> Menú de navegación	53

LISTA DE TABLAS

Tabla 2.1. Preguntas para entrevista.....	12
Tabla 3.1. Detalle de costo	17
Tabla 3.2. Historia de usuario 1	17
Tabla 3.3. Historias de usuario 2	18
Tabla 3.4. Historias de usuario 3	18
Tabla 3.5. Historia de usuario 4	19
Tabla 3.6. Historia de usuario 5	19
Tabla 3.7. Funciones de usuario.....	20
Tabla 3.8. Historias de usuario	23
Tabla 3.9. Lista de Sprints	24
Tabla 4.1. Tarea - Diseñar diagrama entidad relación.....	25
Tabla 4.2. Tarea - Definir modelos relacionales.....	26
Tabla 4.3. Tarea - Generar migraciones	26
Tabla 4.4. Tarea - Instalar gema devise	28
Tabla 4.5. Tarea – Inicio de sesión.....	28
Tabla 4.6. Tarea – Perfil de usuario.....	28
Tabla 4.7. Tarea – Formulario de Tubos de laboratorio	32
Tabla 4.8. Tarea – Formulario de Exámenes	32
Tabla 4.9. Tarea - lista de Exámenes.....	32
Tabla 4.10. Tarea – Formulario de órdenes.....	36
Tabla 4.11. Tarea – Adjuntar resultados	37
Tabla 4.12. Tarea – Registrar pago de orden.....	37
Tabla 4.13. Tarea – Registrar Diagnostico	37
Tabla 4.14. Tarea – Lista de órdenes	38
Tabla 4.15. Tarea – Buscar órdenes.....	38

Tabla 4.16. <i>Tarea – Lista de resultados</i>	41
Tabla 4.17. <i>Tarea – Menú de navegación</i>	43
Tabla 4.18. <i>Tarea – Permisos de usuario</i>	43
Tabla 4.19. <i>Resumen de pruebas de funcionalidad</i>	55
Tabla 4.20. <i>Plan de implementación</i>	57
Tabla 4.21. <i>Plan de capacitación</i>	58

RESUMEN

Este documento de trabajo describe el “DESARROLLO DEL SISTEMA PARA GESTIÓN DE ÓRDENES DE EXÁMENES DE LABORATORIO CON HERRAMIENTAS NO PROPIETARIAS PARA LA EMPRESA ECUA-AMERICAN” como solución para optimizar el proceso de gestión de las órdenes, reduciendo costos y logrando una calidad aceptable.

Para la implementación de este proyecto se utilizó *Scrum*, esta metodología permite realizar controles periódicos al trabajo realizado, al final de cada periodo se realiza una demostración con el cliente para recibir una retroalimentación.

El sistema contará con gestión de usuarios por perfil los cuales son médico, laboratorista, recepcionista y administrador.

El flujo de trabajo del sistema abarcará con la gestión de órdenes de exámenes de laboratorio compuestas por tubos de muestra acorde a los exámenes requeridos por el paciente.

Es sistema fue probado en servidores locales de la empresa la cual está localizada en el D.M. Quito para cumplir con los parámetros de los procesos actuales y la confidencialidad de la información que el sistema almacenará.

PALABRAS CLAVES: Automatización, Proceso, Orden, Examen, Laboratorio, Ruby, Rails, PostgreSQL.

ABSTRACT

This work document describes the "DEVELOPMENT OF THE SYSTEM FOR MANAGEMENT OF LABORATORY EXAM ORDERS WITH NON-OWNING TOOLS FOR THE ECUA-AMERICAN COMPANY" as a solution to optimize the order management process, reducing costs and achieving acceptable quality.

For the implementation of this project, Scrum was used, this methodology allows periodic checks to be carried out, at the end of each period a demonstration is carried out with the client to receive feedback.

The system will have user management by profile which is doctor, laboratory, receptionist and administrator.

The system workflow will cover the management of laboratory test orders made up of sample tubes according to the exams required by the patient.

This system was tested on local servers of the company which is located in the D.M. Quito to comply with the parameters of the current processes and the confidentiality of the information that the system will store.

KEYWORDS: Automation, Process, Order, Exam, Laboratory, Ruby, Rails, PostgreSQL.

INTRODUCCIÓN

Antecedentes de la situación objeto de estudio

“LABORATORIO CLÍNICO ECUA-AMERICAN” 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. Ubicado en Av. América N33-42 Quito, Ecuador.

El departamento de laboratorio ubicado en la matriz de la empresa receipta las órdenes de exámenes de laboratorio las cuales indican el número de análisis y su descripción para que la persona con el perfil de recepcionista las ingrese al sistema. Existen aproximadamente 2000 análisis de laboratorio con nombres parecidos unos con otros, de igual manera existen exámenes que tienen que ser ingresados con otras pruebas de laboratorio ya que son complemento de las anteriores y el sistema de gestión principal no da alarma cuando suceden estos particulares.

Los pacientes suelen preguntar para qué sirve cada análisis a ellos enviados en su pedido, y aunque los recepcionistas no son personal médico no hay una fuente de consulta para dar información real a los pacientes en caso de pruebas que no son de frecuente ingreso situación la cual causa desconfianza del profesionalismo del personal recepcionista. Cada análisis de laboratorio requiere condiciones físicas (del paciente) y tecnológicas para llevarse a cabo y con un listado tan grande de exámenes hay una gran brecha para cometer errores y que los exámenes no sean ingresados de manera correcta causando incomodidad en los pacientes y por ende mala atención, tomando en cuenta que no entregar el resultado de un análisis de forma correcta puede incluso poner en riesgo la integridad física del paciente.

Planteamiento del problema

El proceso anteriormente mencionado genera los siguientes problemas:

Mala calidad de atención a los pacientes ya que su ingreso puede ser errado y no hay una fuente de consulta válida y veraz. Descuentos al personal de ingresos ya que no es permitido por normas internas de la institución llamar a cobrar al paciente valores extras a lo facturado en el momento de la atención si hubo algún error por parte del recepcionista o no ingresó algún examen. Altos tiempos de espera de los pacientes ya que tienen que esperar que alguna persona técnica conteste la llamada para aclarar cualquier duda con los exámenes a ingresarse.

Malestar en el personal técnico ya que tienen que responder varias llamadas en el día de consultas a veces repetidas, pero de distinto personal entorpeciendo su propio trabajo y así exponiéndose a perder la concentración que conlleven a reportar exámenes erradamente

El proceso de gestión de órdenes de exámenes de laboratorio actual seguirá provocando retrasos en la entrega de los resultados y por lo cual la empresa pierda confianza, esto podría causar la pérdida de nuevos clientes, al ser un proceso manual el tiempo de procesamiento de cada orden es mayor al igual que los pasos para completar dicho proceso y no es posible realizar análisis o estadísticas claras

Justificación

El desarrollo y la implementación de un sistema que gestione las pruebas de laboratorio y todos sus aspectos particulares como información de la misma, días de entrega, tubos a ser extraídos, etc. Minimizará notablemente el margen de error de los recepcionistas y proporcionará al paciente una mejor calidad de servicio y confianza en la institución.

El análisis estadístico de consultas permitirá a la administración realizar capacitaciones sobre las consultas que más tendencia han tenido durante un determinado lapso de tiempo y así mantener a su personal entrenado en las pruebas que al momento sean de mayor importancia

Se registrará los requerimientos y peticiones realizadas por personal del departamento técnico con el fin de que las interfaces a presentar tengan la información óptima de consulta, optimizando el proceso de verificación de que las pruebas ingresadas al paciente sean las correctas y se realizará el modelado de interfaces para ser aprobadas optimizando el proceso actual con procedimientos eficaces.

Objetivos

En esta sección se lista los objetivos planteados para llegar a la solución de la problemática.

General

Desarrollar un sistema para la gestión de órdenes de exámenes de laboratorio con Ruby on Rails para la empresa “Ecuamericana” reduciendo costos operativos.

Objetivos específicos

- Analizar y definir actores y responsables del proceso.
- Diseñar diagrama de procesos, interfaces del sistema, y base de datos
- Desarrollar e implementar los módulos del sistema dentro de los cuales se listan: gestión de usuarios, gestión de órdenes y exámenes, gestión de tubos de laboratorio, gestión de pacientes y gestión de resultados.
- Comprobar el correcto funcionamiento y disponibilidad del sistema con pruebas de carga y estrés.

Descripción de los capítulos

El capítulo 1. Fundamentación teórica, describe las normas, definiciones y documentos que ayudaron a la solución de la problemática de investigación.

El capítulo 2. Marco metodológico, resume la metodología seleccionada para el desarrollo del proyecto, debido a que el principal problema es el tiempo para en finalizar el proceso se detalla la investigación con enfoque cuantitativo.

En el capítulo 3. Propuesta, como solución a la problemática se pretende optimizar el proceso actual de órdenes de examen laboratorio, a través del desarrollo de un sistema web que permita gestionar la información de forma estructurada y sistemática con la seguridad que ofrecen las herramientas no propietarias disponibles en el mercado.

El capítulo 4. Implementación, recapitula los resultados obtenidos después de la aplicación de Scrum, se detalla el producto parcial obtenido en cada iteración, una retroalimentación del éxito o retraso acorde a la planificación inicial.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En esta sección se detalla las leyes, documentos y reglamentos en los cuales se apoya la propuesta de solución.

1.1 Estado del arte

En el mercado ecuatoriano existen trabajos similares realizados anteriormente a este, los cuales pretenden gestionar los procesos de un laboratorio clínico, dentro de los cuales encontramos:

Sistema integrado para automatización de un laboratorio clínico orientado a la web, Universidad Politécnica Salesiana, dicho sistema busca brindar todos los controles necesarios para administrar de forma eficaz y eficiente el laboratorio clínico y así alcanzar estándares de calidad y competitividad que exige la gestión clínica (Alvarez, Astudillo, & Zambrano, 2010).

Sistema web de reservación y consulta de exámenes médicos en el laboratorio clínico bacteriológico SEDILAB de la ciudad de Calceta, la implementación de un sistema informático web evitó la pérdida innecesaria de tiempo y recursos, tanto de empleados como de clientes, en la reservación y consulta de exámenes médicos en el laboratorio clínico bacteriológico SEDILAB de la ciudad de Calceta (Alcívar & Mera, 2013).

Los procesos del departamento de laboratorio clínico se fundamentan en la norma ISO-15189, la cual detalla para los laboratorios clínicos todos los puntos, y si siguen estos lineamientos son competentes técnicamente y capaces de producir resultados de calidad y válidos.

La norma ISO 15189 fue creada por el Comité Técnico ISO/TC 212 (*Clinical Laboratory Testing and In Vitro Diagnostic Systems*) basándose en las normas ISO / IEC 17025 y también ISO 9001.

A esta norma la podemos dividir en 2 grupos o partes, primero la parte de gestión que corresponde a requisitos para la certificación del sistema de calidad y como segundo grupo o parte el ámbito técnico que lista y describe los requisitos para el personal, procedimientos, equipos, informes, garantía de calidad e instalaciones. Es en la parte informes donde tiene la diferencia más grande de la norma en la que tiene sus bases, la Norma ISO 9001:2008.

La norma tiene además anexos de información que son 2, el primero que trata de las recomendaciones en cuanto al resguardo de los sistemas de información del laboratorio y el segundo trata de la ética aplicada en el laboratorio clínico.

El tratamiento de la norma da veracidad y comprueba objetivamente e independiente el sentido de compromiso del laboratorio clínico respecto a la calidad y la competencia técnica. Demostrándose de esta forma garantías acordes al funcionamiento del laboratorio clínico, el control que se ejerce sobre sus procesos, la capacidad para cubrir los requerimientos técnicos que se necesitan manteniendo información de suma importancia para el laboratorio clínico (Intedya, 2008).

Tomando toda la información antes descrita es necesario brindar al laboratorio las herramientas tecnológicas necesarias para cumplir los estándares de calidad necesarios manteniendo trazabilidad en las muestras e información clara de las mismas

1.2 Lógica del negocio

Ecuamerican ha establecido, implementado, mantiene y mejora continuamente un Sistema de Gestión de la Calidad, incluyendo los procesos necesarios para la realización del servicio y sus interacciones.

También se han determinado los procesos necesarios para llevar adelante el sistema de gestión de la calidad a través del mapa de procesos institucional.

Las normas que a continuación se mencionan contienen disposiciones que constituyen referencias de este Manual de Calidad.

- ISO 9000:2015, Sistemas de Gestión de la Calidad - Fundamentos y Vocabulario.
- ISO 9001: 2015, Sistemas de Gestión de la Calidad – Requisitos.

La Gerencia General de Ecuamerican asegura que se han identificado los requerimientos de los Clientes, con el objetivo de lograr la satisfacción de los mismos, así como los requerimientos de los clientes constan en los pedidos en el caso de pacientes particulares y domicilios, y en lo establecido en los contratos

Los requisitos legales y reglamentarios están definidos en el Marco Legal Aplicable. (ISO, 2015)

Glosario de los términos técnicos que serán utilizados posteriormente:

- Framework: Marco de Trabajo
- Ruby on Rails: Framework para aplicación con lenguaje Ruby
- Gem/Gema: Librerías externas para Ruby on Rails
- GUI: *Graphical User Interface* / Interfaz gráfica de usuario
- SQL: *Structured Query Language* / Lenguaje de consulta estructurada
- CSS: *Cascading Style Sheets* / Hojas de estilo en cascada
- JS: Abreviado de JavaScript
- IT: *Information Technology* / Tecnologías de la información
- Sprint: Iteración de metodología Scrum

En el mercado existen varias herramientas para el desarrollo de un sistema web, para este trabajo se seleccionó las siguientes:

El equipo de desarrollo después de analizar los lenguajes de programación disponibles en el mercado seleccionó a Ruby por ser un lenguaje de programación interpretado o llamado no compilado, dicha característica permite desplegar páginas web de forma fácil, rápida y estructurada, que es lo que la gerencia general busca para el proyecto, los médicos de la empresa solicitaron un sistema el cual no necesite instalación y se abra desde cualquier computador indistintamente del sistema operativo en uso

Para optimizar el tiempo del equipo de desarrollo se utilizó Rails o mejor conocido como RoR por las iniciales de Ruby on Rails que para nuestro proyecto nos permite desarrollar ágilmente todos los módulos planificados, integrar formularios HTML para fácil registro de pacientes y también es un *Framework* que usa estrictamente el patrón de arquitectura Modelo, Vista, Controlador (MVC) para el desarrollo de aplicaciones web

MVC separa la “lógica del dominio” (también llamada “lógica de negocio”) de la lógica de presentación y entrada asociada con una interfaz de usuario gráfica (GUI), esta característica permite al desarrollador concentrar su interés en la lógica del proceso y no en los problemas de desarrollo e implementación.

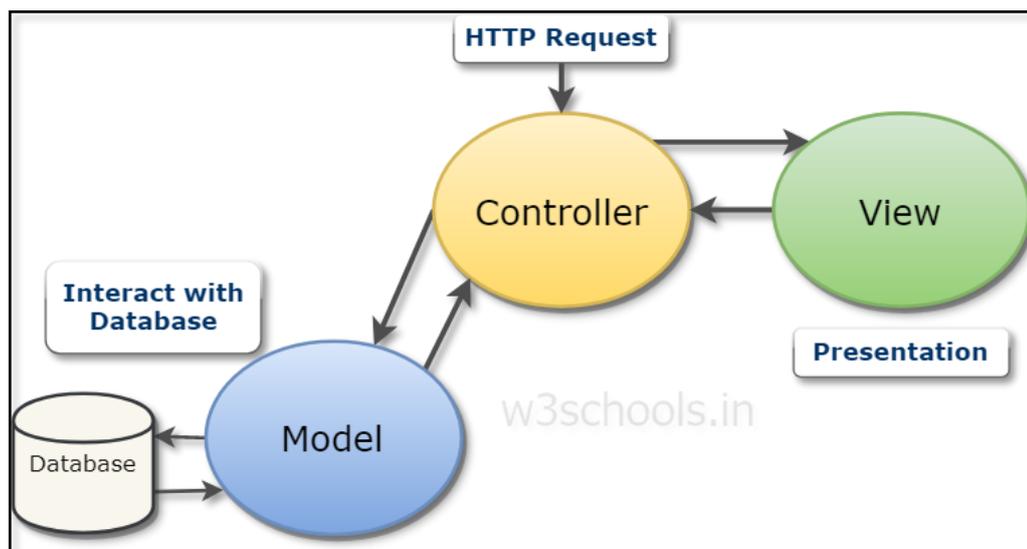


Figura 1.1. Patrón de arquitectura MVC

Las gemas de Ruby conocidas como *gems*, son librerías externas que añaden funcionalidad de forma modular, existe una gran comunidad de desarrolladores que aportan al desarrollo y mantenimiento de las mismas, el repositorio oficial de las gemas de Ruby es rubygems.org (García, 2017).

El sistema web necesita almacenar los datos generados a lo largo del tiempo para cumplir con este requisito se utilizó un sistema de gestión de base de datos permite la manipulación de datos basado en el álgebra relacional, dentro de las opciones compatibles con Ruby on Rails el equipo de trabajo optó por usar PostgreSQL.

PostgreSQL es un sistema de gestión de base de datos relacional de objetos, desarrollado en el Departamento de Informática de la Universidad de California en Berkeley, de código abierto de este código original de Berkeley. Es compatible con una gran parte del estándar SQL (The Group PostgreSQL Global Development, 2019).

El equipo de desarrollo utilizó el lenguaje HTML (del inglés *Hyper Text Markup Language*), para estructurar el contenido de una página web, a través de etiquetas y atributos, en su versión 5 permite la implementación de gráficos dinámicos, audio y video entre otras (Gallardo, 2008).

Para cambiar el tamaño, forma y color de los textos, imágenes el equipo de desarrollo utilizó archivos de estilos conocido como CSS, los cuales son un grupo de reglas de formato que ayudarán a cambiar la apariencia de una página web. JavaScript es un lenguaje interpretado que fue en la etapa de desarrollo para implementar interacción de las páginas con el usuario, es decir agrega dinámica a elementos estáticos de una página web (Gauchat, 2012).

1.3 Alternativas de solución

Existen laboratorios clínicos en Quito con problemas y soluciones similares al del cliente, dentro de las soluciones tenemos las siguientes:

Infinity Roche: Como proveedor de soluciones, Roche utiliza el poder de cobas infinity IT solutions para ayudar a los laboratorios en la integración del flujo de datos con el flujo de trabajo, las muestras, gestionando las actividades diarias relacionadas con la información. *Cobas infinity IT solutions* es una aplicación 100% web que permite acceder desde cualquier dispositivo, habilitado con cualquier navegador estándar, a toda la información del laboratorio.

NubeLab: Es un sistema web desarrollado en México el cual permite llevar el flujo de los análisis de laboratorio y las s de las mismas desde una interface Web en la cual se puede seleccionar varas alternativas y características informativas sobre las distintas pruebas que se pueden realizar dentro del laboratorio clínico.

Las soluciones listadas pueden cubrir el problema planteado sin embargo es deseo de la organización que el software a desarrollar sea exclusivo para el laboratorio y de desarrollo nacional para garantizar el soporte técnico y que los requerimientos sean específicos y no se use un software con características que no apliquen de acuerdo a su necesidad.

CAPÍTULO 2. MARCO METODOLÓGICO

Esta sección describe la metodología seleccionada para el desarrollo del proyecto.

2.1 Tipo de investigación

El tipo de investigación que se ajusta a este proyecto fue una investigación aplicada, que busca poner en práctica los conocimientos teóricos y prácticos obtenidos a lo largo de los niveles cursados por el autor de este trabajo para generar una solución que a los problemas descritos del proceso manual.

La solución genera beneficios a la empresa, la misma que apoya al sector productivo del país.

2.1.1 Metodología seleccionada

Método deductivo. - Se parte de aspectos generales para llegar a los específicos, con el cual fue de ayuda para la recopilación de la información general para poder plantear el problema del proyecto utilizando técnicas como la encuesta y la entrevista (Gutiérrez, 2006).

Acorde a la metodología seleccionada se procedió a crear una entrevista para con el fin de recopilar información de la problemática en la empresa, después de una análisis se procedió a realizar una propuesta.

2.2 Recopilación de información

Los métodos de recolección de información se detallan a continuación.

2.2.1 Técnicas de recopilación de información

De la entrevista

La entrevista es una técnica que permite recoger información, es muy utilizada en proceso de investigación, su objetivo principal es obtener información de forma oral y personalizada de acontecimientos, se la debe realizar mínimo entre dos personas (Coffey & Atkinson, 2003).

Objetivo

El objetivo de la encuesta realizada con la gerencia general busca aproximar al equipo de trabajo a la problemática de la organización, identificar a los involucrados en el proceso, problemas con el mismo y plantear una posible solución que se ajuste a los requerimientos de la empresa.

Construcción de preguntas

Las preguntas realizadas a la gerente general de la empresa fueron.

Tabla 2.1. *Preguntas para entrevista*

Preguntas para entrevista	
Nro.	Pregunta
1	¿Cuál es el principal problema en el proceso actual de gestión de órdenes de laboratorio?
2	¿Quiénes son los actores del proceso?
3	¿Cuántas órdenes de laboratorio se realizan por día?
4	¿La entrega de resultados se realiza de forma física, digital o ambas?
5	¿Cómo es gestionada la confidencialidad de la información?
6	¿La empresa cuenta con los recursos necesarios para la implementación de un sistema web?

Fuente: Elaboración propia

La entrevista realizada entre la gerencia general y el equipo de trabajo se detalla en el Anexo 5. Entrevista.

Evaluación

En la entrevista con la gerencia general se logró identificar los problemas relacionados al proceso de órdenes de exámenes, dichos problemas conciben en una mala organización y comunicación entre los involucrados, y por tal razón se pretende realizar una automatización al proceso para brindar una solución, como objetivo principal del sistema se manifestó que se desea reducir el costo del proceso con la automatización del mismo, y la reducción del personal que responsable, otro punto cubierto en la entrevista fue aumentar la cantidad de órdenes de laboratorio por día.

CAPÍTULO 3. PROPUESTA

Este capítulo describe la propuesta del equipo de trabajo para solucionar la problemática de la organización.

3.1 Diagramas de procesos

El proceso realizado de forma manual por el departamento de laboratorio clínico se describe en la Figura 3.1. Diagrama de proceso manual.

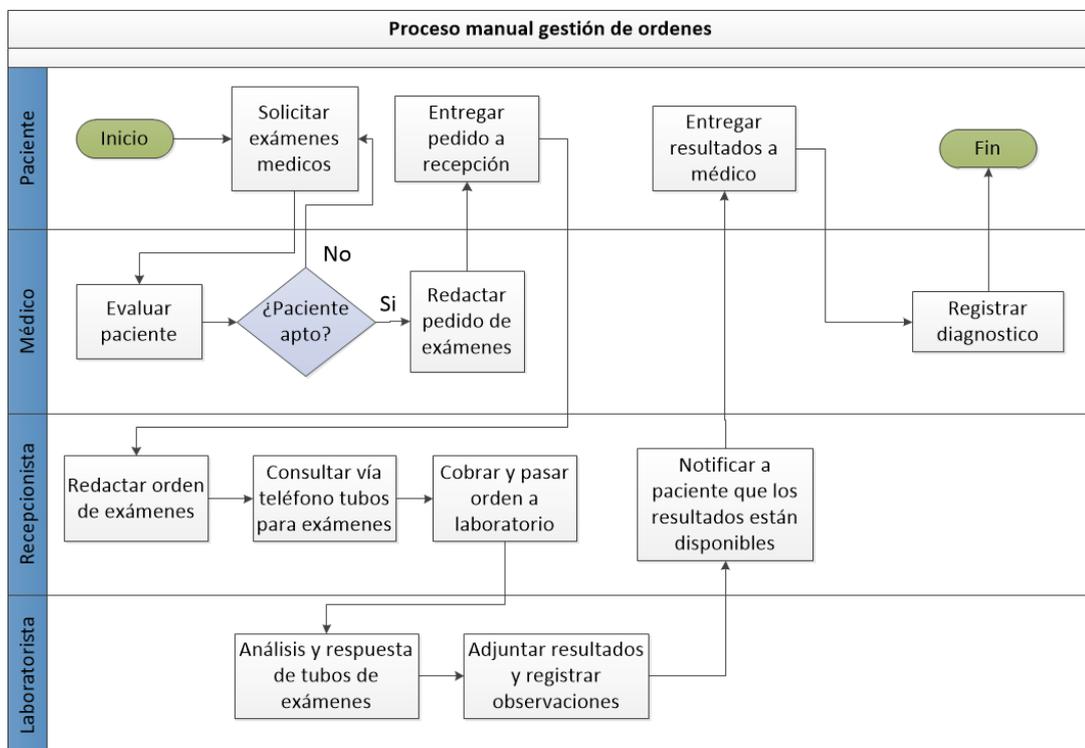


Figura 3.1. Diagrama de proceso manual

Fuente: Elaboración propia

El proceso de gestión de órdenes de exámenes de laboratorio actual provoca retrasos en la entrega de los resultados y por lo cual la empresa pierda confianza, esto podría causar la pérdida de nuevos clientes, al ser un proceso manual el tiempo de proceso de cada orden es mayor o igual que los pasos para completar dicho proceso, no es posible realizar análisis o estadísticas basadas en las órdenes generadas.

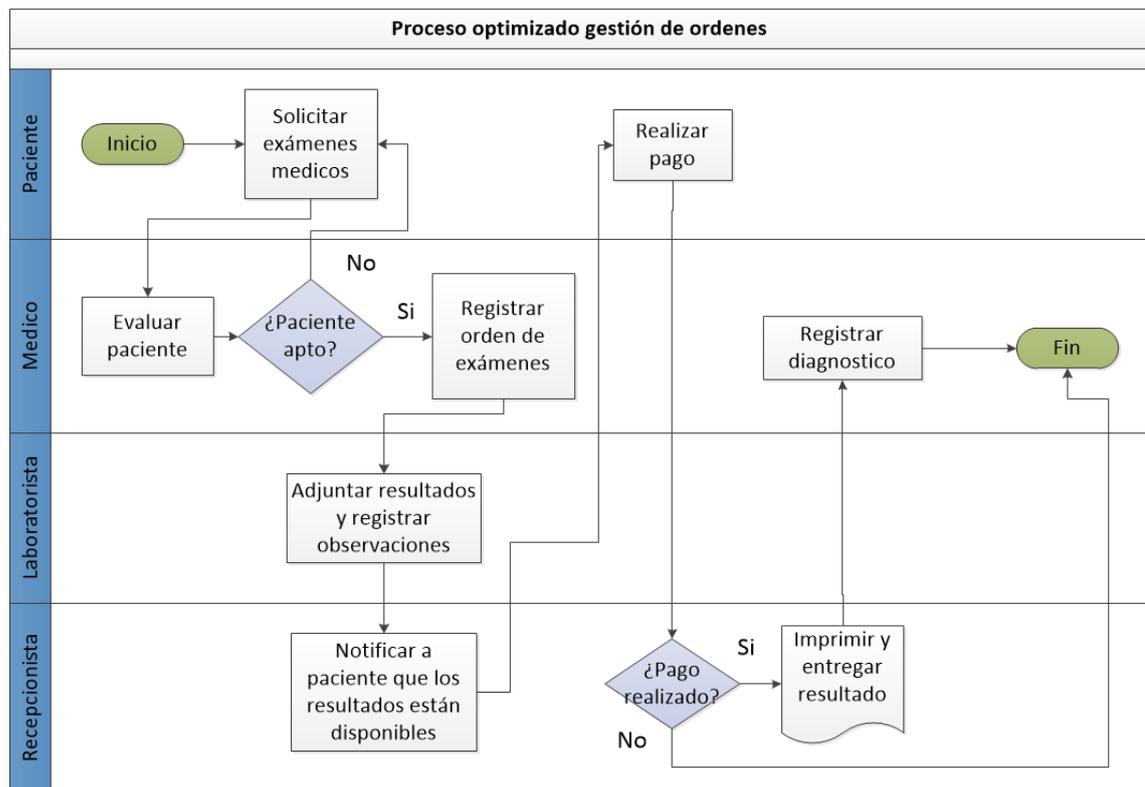


Figura 3.2. Diagrama de proceso automatizado

Fuente: Elaboración propia

La automatización del proceso busca acortar la cantidad de pasos y reducir el tiempo de ejecución de una orden, se definen los roles de usuario que permitirán controlar el flujo de una orden desde que es requerida por el paciente hasta que los resultados son entregados.

Los roles definidos en el proceso son: paciente, médico, laboratorio, recepcionista.

3.2 Factibilidad técnica

En esta sección se lista los recursos técnicos que fueron necesarios para el desarrollo de este proyecto.

Hardware

- Laptop Hp Envy, Intel Core I7, 8 RAM
- Computadora de escritorio Dell, Core I5, 8 RAM,

Software

- Lenguaje de programación Ruby
- Marco de trabajo Ruby on Rails
- Visual Studio Code, editor de código
- Sistema operativo Ubuntu
- Terminal de Linux
- PgAdmin3
- PostgreSQL
- SmartGit, gestor de repositorios Git

Todos los recursos listados anteriormente fueron proporcionados por el autor de este trabajo.

3.3 Factibilidad operacional

De acuerdo a la metodología Scrum se asignaron roles para el desarrollo del proyecto de la siguiente forma.

- **Dueño del producto:** María del Lourdes Ávila, gerente general Ecuamerican.
- **Maestro Scrum:** Gonzalo Molina, autor del documento.
- **Asegurador de la Calidad:** María del Lourdes Ávila, gerente general Ecuamerican.
- **Desarrollador de Software:** Gonzalo Molina, autor del documento.

3.4 Factibilidad económica - financiera

Laboratorio Ecu-american tiene la infraestructura necesaria para el desarrollo de la solución de este proyecto, a continuación, se lista los recursos que fueron utilizados.

Tabla 3.1. *Detalle de costo*

Detalle de costos			
Duración	Valor Unitario	Recurso	Total
3 meses	\$2700	Servidor	\$2.700
3 meses	\$6	Luz eléctrica	\$18
3 meses	\$800	Programador	\$2.400
3 meses	\$22	Internet	\$66
Total			\$5.184

Fuente: Elaboración propia

Los costos detallados en la tabla anterior serán cubiertos por la empresa.

3.5 Especificación de requerimientos

En esta sección se describen los requerimientos del sistema a implementar.

3.5.1 **Ámbito del software**

El Equipo de trabajo denominó al sistema web como EcuSys el cual pretende lograr los objetivos planificados inicialmente, los usuarios del sistema tendrán acceso fácil y seguro a la información almacenada, aplicando los controles según su rol asignado.

3.5.2 **Funciones del producto**

Acorde a la metodología Scrum, las funciones del producto son descritas por las historias de usuario, dicha herramienta permite crear una plantilla para registrar la información del negocio y estimar su prioridad en el desarrollo del proyecto, el equipo de trabajo recopiló la siguiente información a través de una reunión con la gerente general.

Tabla 3.2. *Historia de usuario 1*

HISTORIA DE USUARIO

Número: 1**Tipo de usuario:** Administrador**Nombre historia:** Gestión de usuarios**Prioridad en negocio:** Media**Prioridad en desarrollo:** Baja**Puntos estimados:** 40**Iteración asignada:** Módulo de usuarios

Descripción: El usuario administrador del sistema debe tener acceso a un módulo para gestionar los usuarios y roles para acceder a las funciones del proceso de acuerdo al diagrama de proceso optimizado.

Observación: Los usuarios podrán iniciar una sesión de forma segura en el sistema a través de una pantalla que solicite un nombre de usuario y clave.

Fuente: Elaboración propiaTabla 3.3. *Historias de usuario 2*

HISTORIA DE USUARIO

Número: 2**Tipo de usuario:** Administrador**Nombre historia:** Gestión de exámenes y tubos**Prioridad en negocio:** Media**Prioridad en desarrollo:** Media**Puntos estimados:** 60**Iteración asignada:** Módulo de exámenes

Descripción: El usuario administrador del sistema tendrá acceso para gestionar los exámenes que realizan la empresa y los tubos de laboratorio necesarios para llevar a cabo esta operación.

Observación: Los exámenes y tubos de laboratorio podrán ser creados, editados, listados y eliminados por el administrador del sistema, el formulario de los tubos de laboratorio incluye el precio con el fin de generar una orden con precios reales e informar el costo al paciente.

Fuente: Elaboración propiaTabla 3.4. *Historias de usuario 3*

HISTORIA DE USUARIO

Número: 3**Tipo de usuario:** Médico, Recepcionista, Laboratorista**Nombre historia:** Gestión de órdenes**Prioridad en negocio:** Alta**Prioridad en desarrollo:** Alta**Puntos estimados:** 80**Iteración asignada:** Módulo de órdenes

Descripción: El medico podrá registrar órdenes de exámenes, después de esta acción el laboratorista registrará los resultados y finalmente el recepcionista entregará los resultados después de recibir el pago por parte del paciente.

Observación: Las órdenes siguen un proceso secuencial, solo los médicos podrán registrar órdenes, después se adjuntará los resultados y finalmente la recepción entrega los resultados y envían a facturación.

Fuente: Elaboración propiaTabla 3.5. *Historia de usuario 4*

HISTORIA DE USUARIO

Número: 4**Tipo de usuario:** Recepcionista**Nombre historia:** Gestión de resultados**Prioridad en negocio:** Baja**Prioridad en desarrollo:** Baja**Puntos estimados:** 40**Iteración asignada:** Módulo de usuarios

Descripción: El usuario recepcionista tendrá acceso a una pantalla para imprimir los resultados del de los exámenes en cualquier momento.

Observación Los resultados de los exámenes se adjuntan en PDF a las órdenes.

Fuente: Elaboración propiaTabla 3.6. *Historia de usuario 5*

HISTORIA DE USUARIO

Número: 5

Tipo de usuario: Todos

Nombre historia: Navegación y permisos de usuario

Prioridad en negocio: Media

Prioridad en desarrollo: Media

Puntos estimados: 50

Iteración asignada: Navegación y permisos de usuario

Descripción: Los usuarios podrán navegar en entre los módulos y el administrador del sistema podrá gestionar los permisos por usuario.

Observación Si un usuario no tiene acceso a una pantalla el mensaje debe notificar con un mensaje que no tiene acceso.

Fuente: Elaboración propia

3.5.3 Características de los usuarios del sistema

A continuación, se describen las funciones de los roles definidos anteriormente para la gestión y control de la información del sistema.

Tabla 3.7. *Funciones de usuario*

Nombre de Rol	Tipo de Usuario	Área Funcional	Actividad
Administrador	Administrador del Sistema	Administración	Administrar toda la información del sistema
Médico	Generador de órdenes	Departamento medico	Crear/Modificar órdenes de exámenes Registrar diagnostico
Paciente	Solicitador de órdenes	Clientes	Solicitar órdenes de exámenes
Recepcionista	Registrador de órdenes	Recepción	Registrar pago de órdenes Imprimir órdenes Enviar órdenes a facturación
Laboratorista	Registrador de resultados	Laboratorio clínico	Adjuntar resultados

Fuente: Elaboración propia

3.5.4 Restricciones de desarrollo

El desarrollo de este proyecto tuvo como reestructuraciones:

- El sistema fue desarrollado como una solución a los problemas del proceso de gestión de órdenes del laboratorio Ecu-a-american ubicado en el D.M. de Quito
- Se utilizó herramientas de código abierto para el desarrollo e implementación del sistema, dentro de las cuales están: Ruby on Rails, PostgreSQL, Visual Studio Code, HTML, CSS y JavaScript.
- El sistema gestionará las órdenes a nivel de laboratorio clínico, no es un sistema de facturación o control de costos.

3.5.5 Requisitos

En esta sección se detalla los requisitos funcionales y no funcionales del proyecto.

Funcionales.

RF01: El sistema debe incluir un módulo para la gestión de usuarios, el cual permitirá a los usuarios iniciar sesión, modificar perfil, crear usuarios, listar usuarios.

RF02: El sistema debe incluir un módulo para la gestión de exámenes y tubos, el mismo podrá crear, editar, listar y eliminar exámenes y resultados.

RF03: El sistema debe incluir un módulo para la gestión de órdenes, el mismo podrá crear, editar, listar y eliminar ordenes, además la opción de buscar por cedula o nombre del paciente.

RF04: El sistema debe incluir un módulo para la gestión de resultados, el mismo estará compuesto por una página para con los resultados agrupados por el estado de la oren.

RF05: El sistema deberá incluir un menú de navegación para acceder a los módulos del sistema.

No funcionales.

Los requisitos no funcionales para este proyecto fueron los siguientes.

Eficiencia.

RNF01: Cuando un usuario genere información en el sistema como crear pacientes, exámenes, órdenes, la operación en la base de datos debe tomar como máximo 3 segundos en realizar.

RNF02: Las operaciones como buscar y listar tendrán hasta 5 segundos como máximo.

Seguridad, consistencia de datos.

RNF03: Los accesos para cada rol de usuario serán gestionados únicamente por el administrador del sistema.

RNF04: La seguridad del acceso al sistema será gestionado por una gema de Ruby on Rails llamada “devise”, la cual permite gestionar los las sesiones del usuario y los accesos a los módulos del sistema.

RNF05: El sistema será alojado en un servidor local de la organización bajo una intranet de la empresa.

RNF06: El servidor de la organización será el encargado de generar respaldos automáticos de la base de datos.

Usabilidad.

RNF07: El diseño del sistema y sus interfaces deben ser fáciles de entender y con títulos que describan su funcionamiento.

RNF08: Los usuarios deben acceder al sistema desde cualquier punto de red de la intranet de la organización por tal razón el sistema será desarrollado en plataforma web.

Lista de usuarios

Como recomendación de *Scrum* se definieron los requisitos funcionales mediante las historias de usuario recopiladas anteriormente.

Al conjunto de todas las historias se lo denomina *Product Backlog*, esta acción permite estimar en puntos cada historia y priorizar según el nivel de dificultad.

A continuación, se observa todas las historias de usuario y sus puntos estimados por el equipo de trabajo.

Tabla 3.8. *Historias de usuario*

Lista de historias de usuario		
Nro. Referencia	Historia de usuario	Puntos estimados
1	Gestión de usuarios	40
2	Gestión de exámenes y tubos	50
3	Gestión de órdenes	60
4	Gestión de resultados	40
5	Navegación y permisos de usuario	50

Fuente: Elaboración propia

Cada historia de usuario representó una iteración, es decir con la implementación de cada historia se obtuvo un producto demostrable, a este ciclo se lo conoce como Sprint, a continuación, se muestra la planificación de Sprints.

Tabla 3.9. *Lista de Sprints*

Lista de Sprints			
Numero	Nombre	Historias de Usuario	Días
0	Crear base de datos	No aplica	5
1	Módulo de usuario	1	5
2	Módulo de exámenes y tubos de laboratorio	2	10
3	Módulo de órdenes	3	15
4	Módulo de resultados	4	5
5	Navegación y permisos de usuario	5	5

Fuente: Elaboración propia

Los ciclos fueron implementados por el equipo de trabajo, y al final de cada uno de estos, se registró una retroalimentación que permite al equipo de trabajo controlar los incidentes que pudieran presentarse.

CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo se detalla los pasos seguidos sistemáticamente para implementar el sistema web.

4.1 Diseño

De acuerdo a la metodología Scrum, se procede a la ejecución de los Sprints.

Sprint 0

En el primer Sprint se consiguió como resultado la creación de la base de datos, Ruby on Rails permite gestionar una base de datos a través de migraciones gestionadas por una gema llamada “Active Record”, para nuestro caso se utilizó el gestor de base de datos PostgreSQL, las tareas necesarias para cubrir esta iteración con las siguientes:

Tabla 4.1. *Tarea - Diseñar diagrama entidad relación*

Tarea: Definir modelos relacionales		
Número: 1	Historia: No aplica	Puntos: 5
Responsable Desarrollo:	Gonzalo Molina	
Responsable Calidad:	Gerencia Ecuamericana	
Descripción: Diseñar un diagrama entidad relación, para posteriormente generar los modelos necesarios, los cuales son los siguientes: Usuario, Rol, Orden, Detalle de orden, Examen, Tubo.		

Fuente: Elaboración propia

Tabla 4.2. *Tarea - Definir modelos relacionales*

Tarea: Definir modelos relacionales		
Número: 2	Historia: No aplica	Puntos: 5
Responsable Desarrollo:	Gonzalo Molina	
Responsable Calidad:	Gerencia Ecuamericana	
Descripción: El desarrollador debe escribir los modelos mediante migraciones, que permiten genera la base de datos en PostgreSQL: Usuario, Rol, Orden, Detalle de orden, Examen, Tubo.		

Fuente: Elaboración propia

Tabla 4.3. *Tarea - Generar migraciones*

Tarea: Generar migraciones		
Número: 3	Historia: No aplica	Puntos: 5
Responsable Desarrollo:	Gonzalo Molina	
Responsable Calidad:	Gerencia Ecuamericana	
Descripción: Definir los modelos necesarios para cumplir con la lógica del sistema, dentro de los cuales se encuentran los siguientes: Usuario, Rol, Orden, Detalle de orden, Examen, Tubo.		

Fuente: Elaboración propia

Resultados

En la siguiente figura se observa la planificación que realizó el equipo de trabajo, para esta iteración se estimó 5 días de duración en los cuales se debe cubrir 15 puntos correspondientes a 3 tareas.

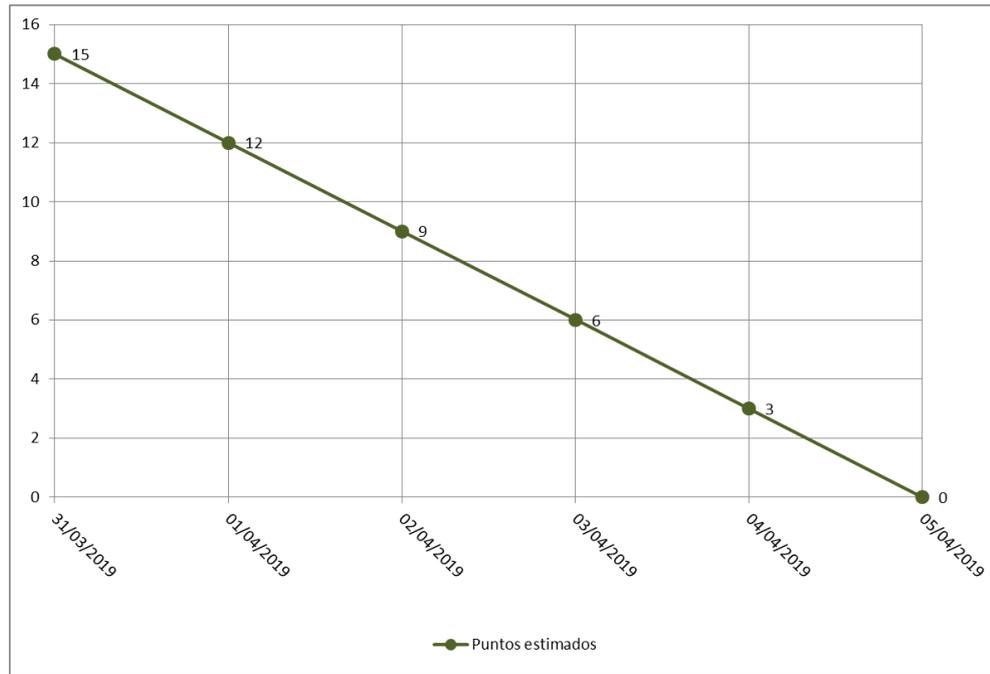


Figura 4.1. Planificación inicial – Sprint 0

Fuente: Elaboración propia

En la siguiente figura se observa los resultados que logró el equipo de trabajo.

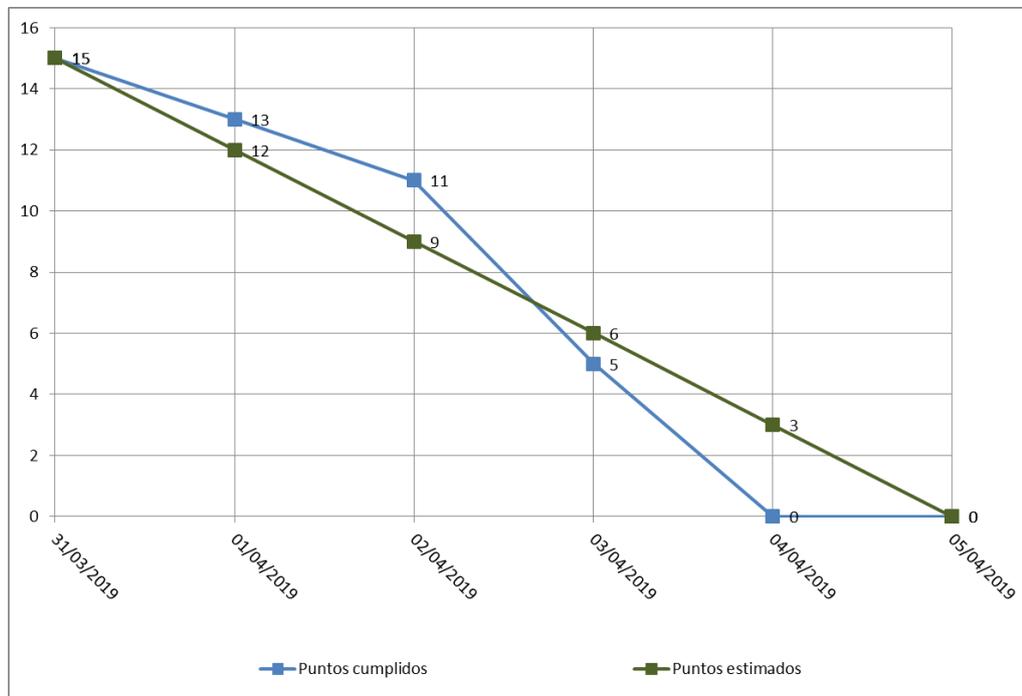


Figura 4.2. Resultados – Sprint 0

Fuente: Elaboración propia

Demostración

En la presentación con el cliente se mostró los resultados que se obtuvo en esta iteración, ya que la base de datos es acerca de un tema técnico, no se explicó a mayor detalle, solo se mostró al cliente un diagrama con la base de datos mostrada en la siguiente imagen.

Sprint 1

De acuerdo al *Sprint 1*, el equipo de trabajo definió las siguientes tareas para cumplir con la gestión de usuarios.

Tabla 4.4. *Tarea - Instalar gema devise*

Tarea: Instalar gema devise		
Número: 1	Historia: 1	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable Calidad:	Gerencia Ecu-a-american	
Descripción: Buscar e instalar la versión de la gema “devise” compatible con Ruby on Rails 5 y generar las migraciones necesarias.		

Fuente: Elaboración propia

Tabla 4.5. *Tarea – Inicio de sesión.*

Tarea: Iniciar sesión		
Número: 2	Historia: 1	Puntos: 15
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: Crear una pantalla con la que los usuarios puedan iniciar una sesión segura en el sistema, como datos requeridos se tendrá un nombre de usuario y una contraseña.		

Fuente: Elaboración propia

Tabla 4.6. *Tarea – Perfil de usuario*

Tarea: Perfil para usuarios

Número: 3**Historia:** 1**Puntos:**15**Responsable Desarrollo:**

Gonzalo Molina

Responsable de Calidad:

Gerencia Ecuo-american

Descripción: Crear una pantalla para administrar los datos de los usuarios del sistema, esta pantalla solo estará disponible para el administrador del sistema.

Fuente: Elaboración propia**Resultados**

La planificación inicial de esta iteración constó de 5 días para cubrir 40 puntos

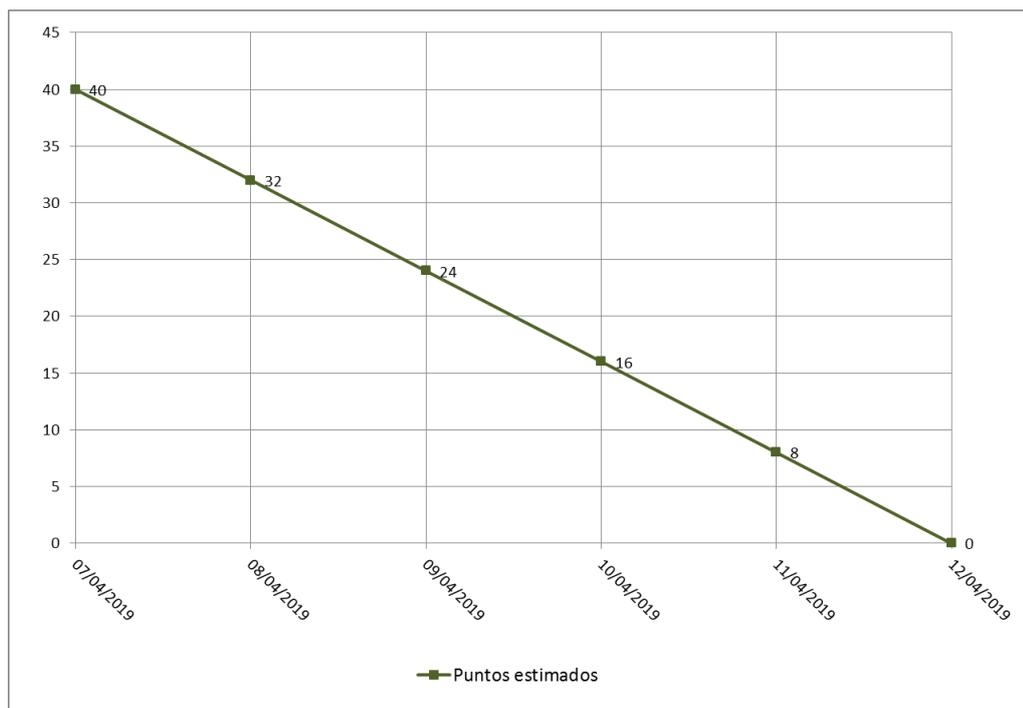


Figura 4.3. Planificación Inicial – Sprint 1

Fuente: Elaboración propia

Los resultados de esta iteración se muestran en la siguiente figura.

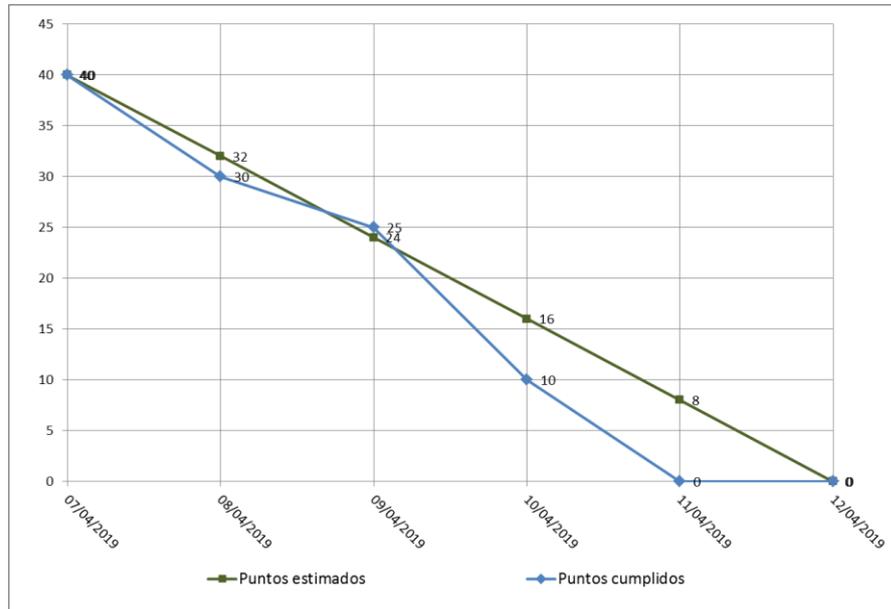


Figura 4.4. Resultados – Sprint 1

Fuente: Elaboración propia

El equipo de trabajo logró completar esta iteración un día antes de lo planificado como podemos observar en la Figura 4.4, la línea azul siempre se mantuvo bajo la línea verde.

Demostración

EcuaSys

Iniciar Sesión

Nombre

Clave

Recordarme?

Iniciar sesión

Si tienes problemas para iniciar sesión, contactate con el Administrador del Sistema: admin@cotizacion.com

Figura 4.5. Resultados – Inicio de sesión

Fuente: Elaboración propia

Figura 4.6. Resultados – Formulario usuario

Fuente: Elaboración propia

Nombre	Creado en	CI	Genero	Edad	Teléfono	Email	Acciones
Carlos Torres	12-08-2019	1721050290	Masculino	29	1234567	usuario1@ecuasys.com	Editar Eliminar
Julio Vasquez	12-08-2019	123456789	Masculino	33	1234567	usuario2@ecuasys.com	Editar Eliminar
Jairo Pineda	12-08-2019	123456789	Masculino	25	1234567	usuario3@ecuasys.com	Editar Eliminar
Lorena Rodríguez	12-08-2019	123456789	N/A	28	1234567	usuario4@ecuasys.com	Editar Eliminar

Total : 4

Figura 4.7. Resultados – Lista usuarios

Fuente: Elaboración propia

Sprint 2

En el *Sprint 2*, se logró implementar el módulo de exámenes y tubos de laboratorio, para lo cual se crearon las siguientes tareas.

Tabla 4.7. *Tarea – Formulario de Tubos de laboratorio*

Tarea: Formulario de Tubos de laboratorio		
Número: 1	Historia: 2	Puntos: 20
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El administrador debe tener acceso a una pantalla para crear tubos de laboratorio.		

Fuente: Elaboración propia

Tabla 4.8. *Tarea – Formulario de Exámenes*

Tarea: Formulario de Exámenes		
Número: 2	Historia: 2	Puntos: 20
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El usuario administrador debe tener acceso a una pantalla para crear exámenes.		

Fuente: Elaboración propia

Tabla 4.9. *Tarea - lista de Exámenes.*

Tarea: Lista de Exámenes		
Número: 3	Historia: 2	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El administrador debe tener acceso a una pantalla para listar todos los exámenes.		

Fuente: Elaboración propia

En la pantalla anterior se observa como los usuarios del sistema podrán listar las órdenes registradas en el sistema, esta pantalla consta de un cajón de texto para buscar órdenes, con consecuencia los usuarios tendrán la comodidad para buscar órdenes de forma fácil y sencilla.

Resultados

Para esta iteración se estimaron 10 días para cubrir 30 puntos correspondientes a 3 tareas, de tal forma la planificación se muestra en la siguiente figura.

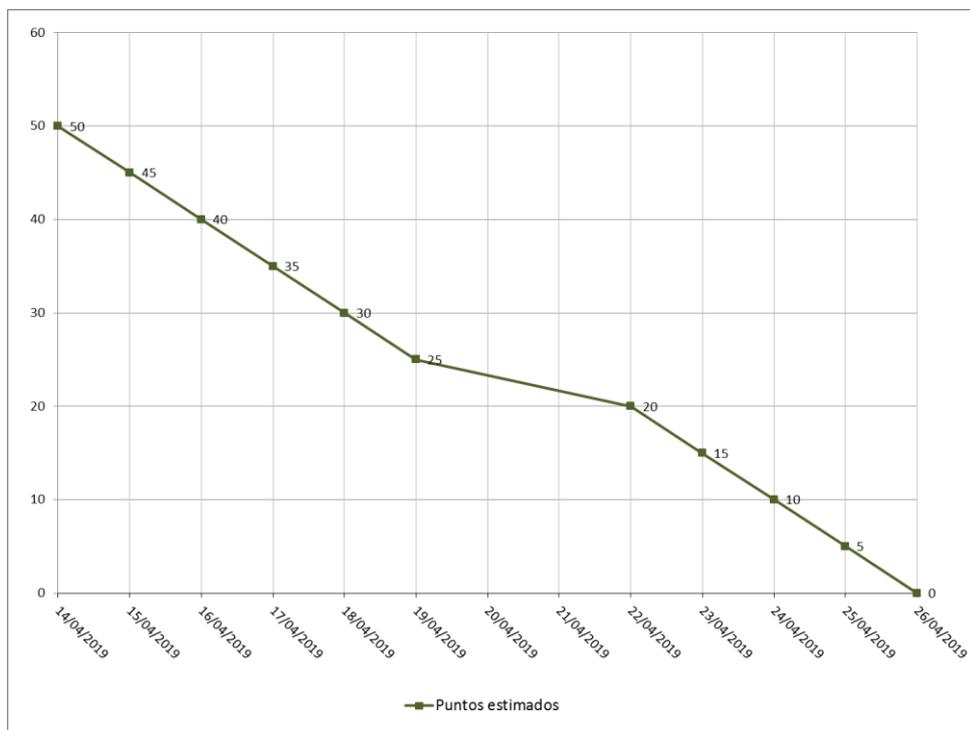


Figura 4.8. Planificación Inicial – Sprint 2

Fuente: Elaboración propia

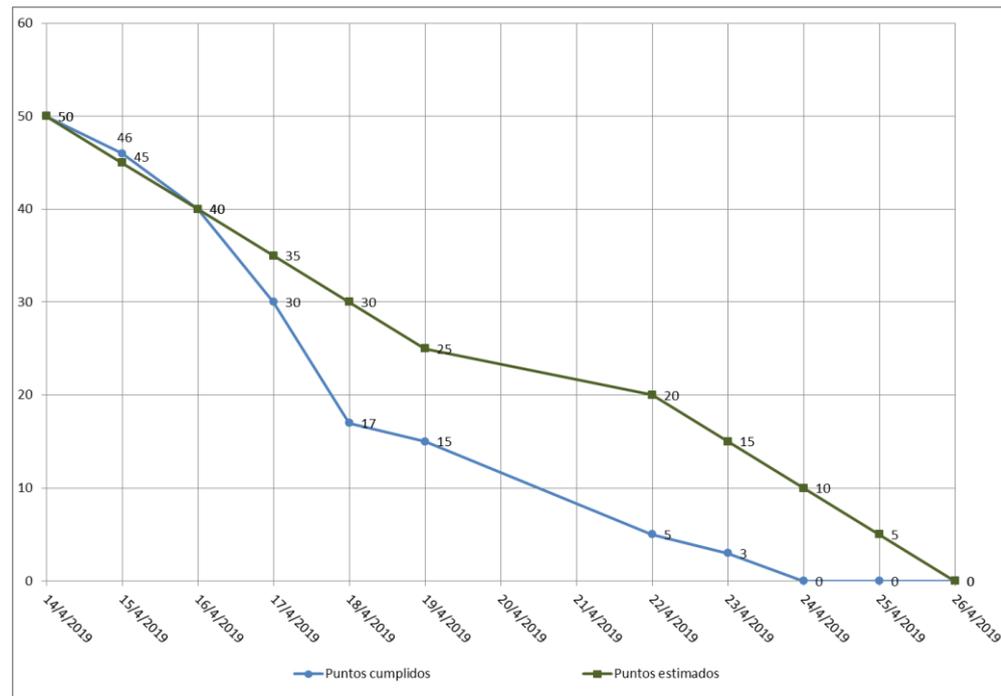


Figura 4.9. Resultados – Sprint 2

Fuente: Elaboración propia

Demostración

Resultados Pacientes Órdenes Exámenes Tubos Buscar cotizaciones

ECUAamerican NUEVO EXÁMEN

Información de examen

Código:

Nombre:

Activado?

Tubo:

Figura 4.10. Resultados – Formulario de exámenes

Fuente: Elaboración propia

Código	Nombre	Creado en	Tubo	Activado	Acciones
EXA001	Hemograma completo	12-08-2019	Tubo azul	Si	Editar Eliminar
EXA002	Urinálisis completo	12-08-2019	Tubo blanco	Si	Editar Eliminar
EXA003	Heces por parásito, sangre oculta	12-08-2019	Tubo Morado	Si	Editar Eliminar
EXA004	Perfil renal: Nitrógeno de urea, Creatinina, Ácido úrico, Proteína total, albúmina/globulina calcio, glucosa	12-08-2019	Tubo rojo	Si	Editar Eliminar
EXA005	Perfil lipídico: Colesterol, LDL; HDL; triglicérido	12-08-2019	Tubo amarillo	Si	Editar Eliminar

Total : 5

Figura 4.11. Resultados – Lista de exámenes

Fuente: Elaboración propia

Información del tubo

Código	<input type="text" value="TB001"/>
Nombre	<input type="text" value="Tubo azul"/>
Precio	<input type="text" value="10.0"/>

[Guardar](#)

Figura 4.12. Resultados – Formulario de Tubos

Fuente: Elaboración propia

Código	Nombre	Creado en	Precio	Acciones
TB001	Tubo azul	12-08-2019	10.0	Editar Eliminar
TB002	Tubo blanco	12-08-2019	20.0	Editar Eliminar
TB003	Tubo Morado	12-08-2019	10.0	Editar Eliminar
TB004	Tubo rojo	12-08-2019	12.0	Editar Eliminar
TB005	Tubo amarillo	12-08-2019	8.0	Editar Eliminar
Total : 5				

Figura 4.13. Resultados – Lista de Tubos

Fuente: Elaboración propia

Sprint 3

En esta iteración se logró implementar los módulos de órdenes, en especial este módulo es el más extenso ya que se pretende que sea el más utilizado por los usuarios del sistema. Las tareas creadas fueron las siguientes.

Tabla 4.10. *Tarea – Formulario de órdenes*

Tarea: Formulario de órdenes		
Número: 1	Historia: 3	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable Calidad:	Gerencia Ecu-a-american	
Descripción: El usuario médico debe tener acceso a una pantalla para crear órdenes en el sistema de acuerdo a los exámenes requeridos por el paciente.		

Fuente: Elaboración propia

Tabla 4.11. *Tarea – Adjuntar resultados*

Tarea: Adjuntar resultados		
Número: 2	Historia: 3	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El usuario laboratorista debe tener acceso a una orden para adjuntar un PDF con los resultados de los exámenes requeridos.		

Fuente: Elaboración propia

Tabla 4.12. *Tarea – Registrar pago de orden*

Tarea: Registrar pago de orden		
Número: 3	Historia: 3	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El usuario recepcionista debe tener acceso a una orden para registrar el pago de la misma cuando el paciente se acerque a la recepción y realice la cancelación.		

Fuente: Elaboración propia

Tabla 4.13. *Tarea – Registrar Diagnostico*

Tarea: Registrar Diagnostico		
Número: 4	Historia: 3	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El usuario médico debe tener acceso a una orden para registrar el diagnostico de acuerdo a los resultados adjuntos.		

Fuente: Elaboración propia

Tabla 4.14. *Tarea – Lista de órdenes*

Tarea: Lista de órdenes		
Número: 5	Historia: 3	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: Los usuarios involucrados en el proceso deben tener acceso a una pantalla para listar las órdenes registradas y analizar su información.		

Fuente: Elaboración propia

Tabla 4.15. *Tarea – Buscar órdenes*

Tarea: Buscar órdenes		
Número: 6	Historia: 3	Puntos: 10
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: Los usuarios involucrados en el proceso deben tener acceso a una pantalla para buscar órdenes.		

Fuente: Elaboración propia

De acuerdo con el diagrama de proceso el usuario laboratorista tendrá la opción de adjuntar un PDF con los resultados y el medico podrá registrar un diagnóstico.

Como pedido del cliente el equipo de desarrollo aumentó un campo para distinguir si los resultados son prioritarios, por esta razón podemos ver al campo llamado “Resultados emergentes”.

Resultados

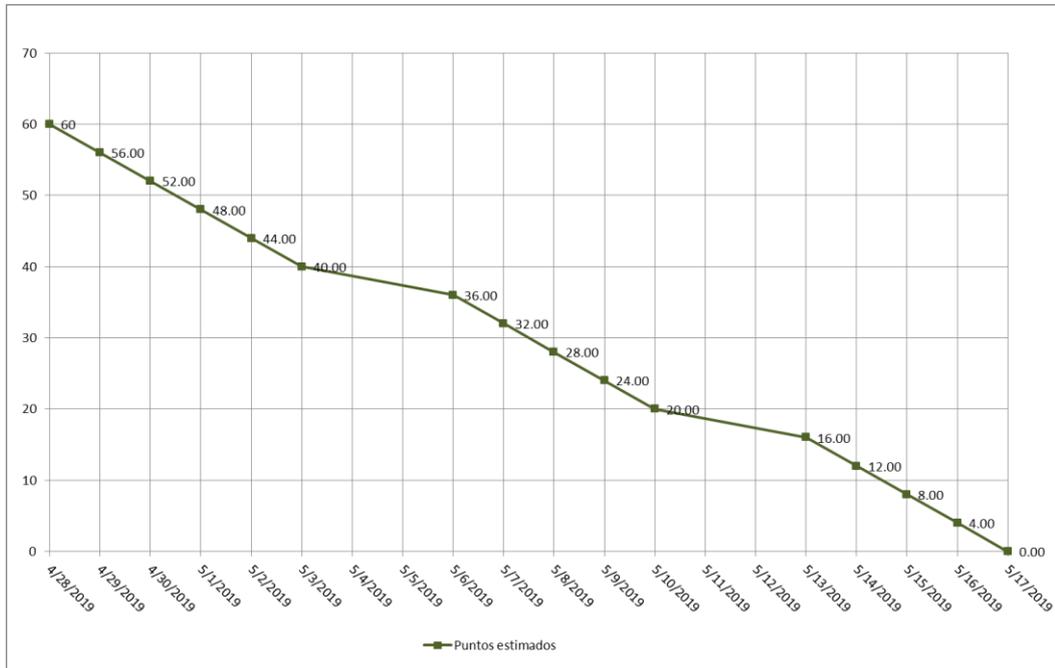


Figura 4.14. Planificación Inicial – Sprint 3

Fuente: Elaboración propia

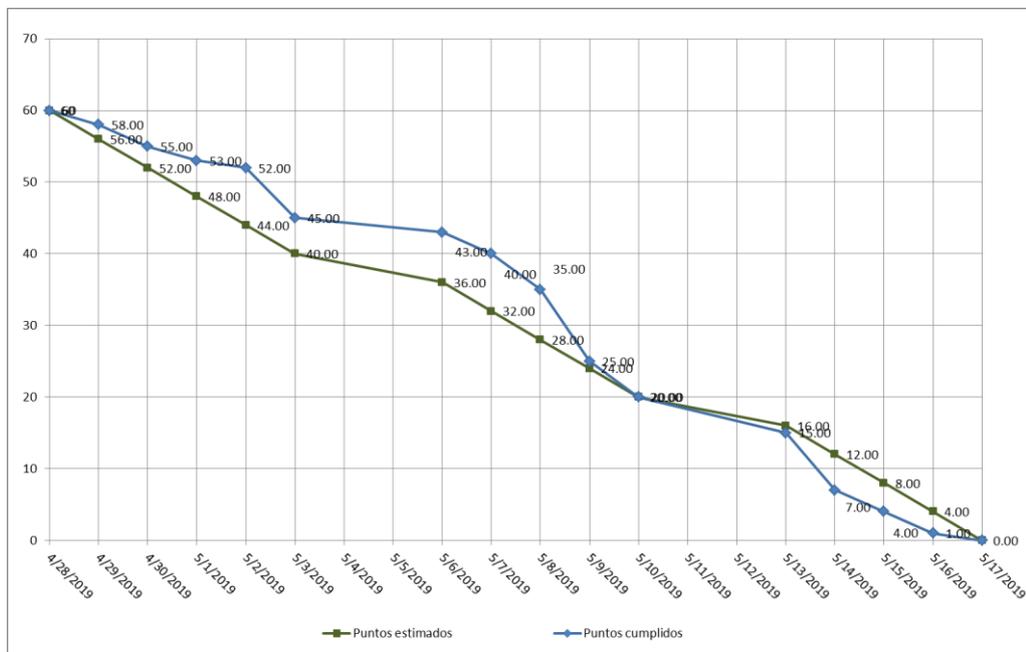


Figura 4.15. Resultados – Sprint 3

Fuente: Elaboración propia

Demostración

Figura 4.16. Resultados – Formulario de órdenes

Fuente: Elaboración propia

Descripción	Empresa	Creado en	Total	Etapa	Aprobado en	Aprobador por	Acciones
Cotización de prueba	Petroecuador	12-08-2019	\$35.00	Aprobada	12-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	12-08-2019	\$50.00	Rechazada			Editar Eliminar
Cotización de prueba	Petroecuador	11-08-2019	\$25.00	Aprobada	11-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	11-08-2019	\$25.00	Aprobada	11-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	10-08-2019	\$50.00	Aprobada	10-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	08-08-2019	\$26.00	Rechazada			Editar Eliminar
Cotización de prueba	Petroecuador	08-08-2019	\$35.00	Aprobada	08-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	08-08-2019	\$18.00	Aprobada	08-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	05-08-2019	\$18.00	Aprobada	05-08-2019	Marcelo Toapanta	Editar Eliminar
Cotización de prueba	Petroecuador	03-08-2019	\$25.00	Pendiente			Editar Eliminar
Cotización de prueba	Petroecuador	02-08-2019	\$35.00	Pendiente			Editar Eliminar
Cotización de prueba	Petroecuador	02-08-2019	\$55.00	Aprobada	02-08-2019	Marcelo Toapanta	Editar Eliminar

Figura 4.17. Resultados – Lista de órdenes

Fuente: Elaboración propia

Sprint 4

En esta iteración se logró implementar el módulo de resultados el cual se planificó para 5 días, el módulo de resultados permite a los usuarios buscar las órdenes con resultados y visualizar su información, la tarea necesaria para implementar este módulo es la siguiente.

Tabla 4.16. *Tarea – Lista de resultados*

Tarea: Lista de resultados		
Número: 1	Historia: 4	Puntos: 40
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecuamericana	
Descripción: Los usuarios involucrados en el proceso deben tener acceso a una pantalla para listar y filtrar los resultados de las órdenes de examen.		

Fuente: Elaboración propia

Resultados

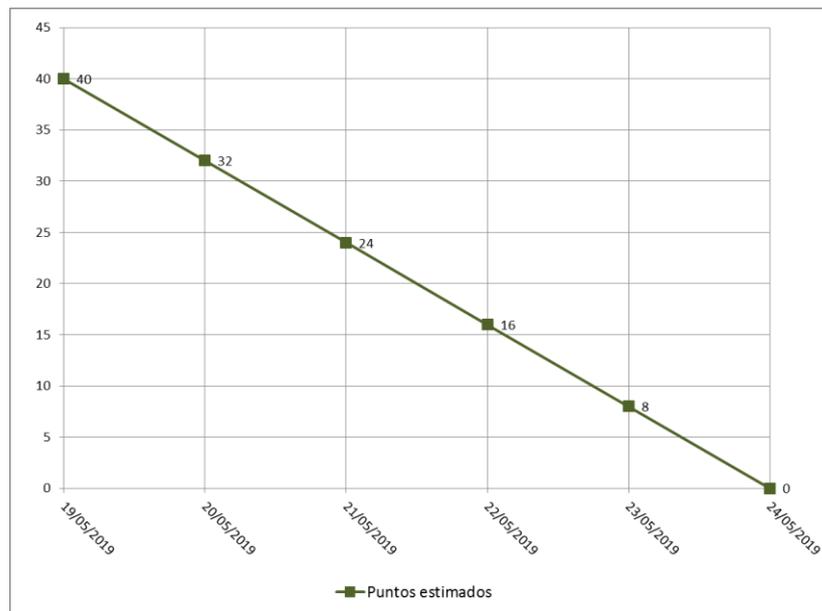


Figura 4.18. Planificación Inicial – Sprint 4

Fuente: Elaboración propia

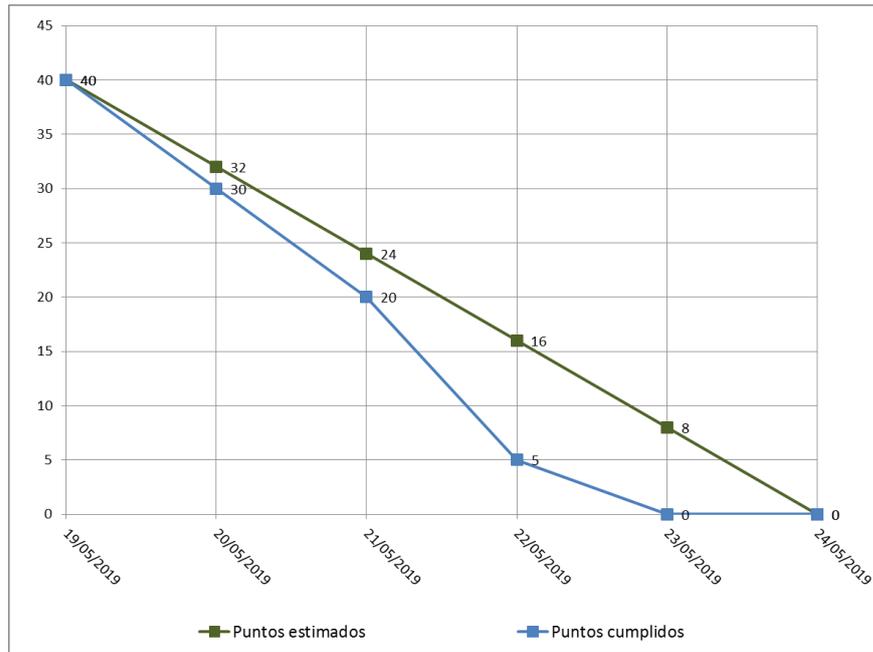


Figura 4.19. Resultados – Sprint 4

Fuente: Elaboración propia

Demostración

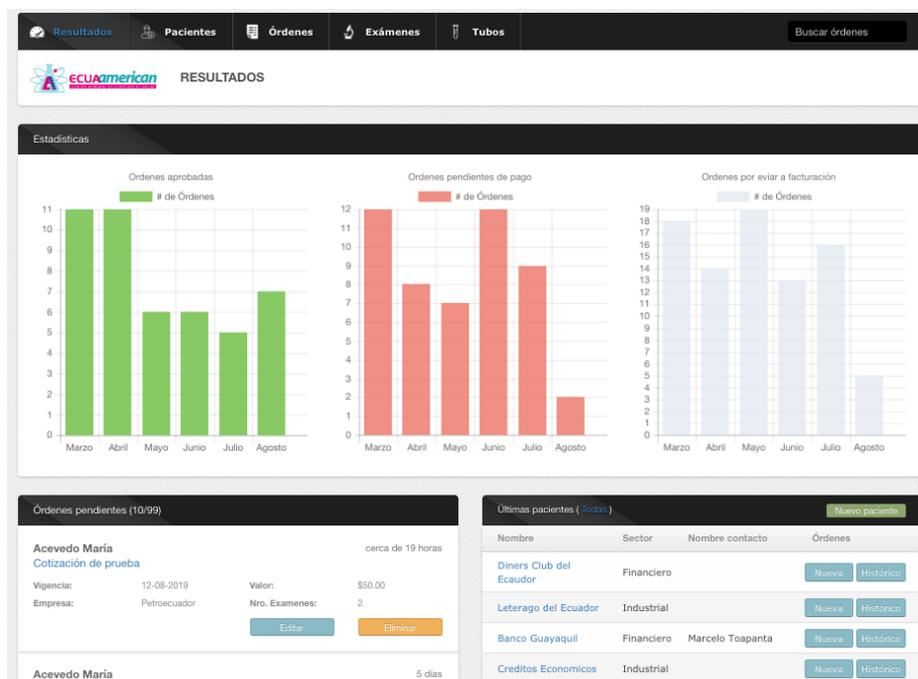


Figura 4.20. Resultados – Lista de resultados de órdenes

Fuente: Elaboración propia

Sprint 5

En el Sprint 5 se logró implementar el menú de navegación y los permisos de usuario, esta iteración se planificó para 5 días y las tareas necesarias para esto fueron las siguientes.

Tabla 4.17. *Tarea – Menú de navegación*

Tarea: Menú de navegación		
Número: 1	Historia: 5	Puntos: 25
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: Los usuarios deben acceder a los diferentes módulos del sistema y para esto es necesario implementar un menú de navegación con enlaces a cada módulo según su rol de usuario.		

Fuente: Elaboración propia

Tabla 4.18. *Tarea – Permisos de usuario*

Tarea: Permisos de usuario		
Número: 2	Historia: 5	Puntos: 25
Responsable Desarrollo:	Gonzalo Molina	
Responsable de Calidad:	Gerencia Ecu-a-american	
Descripción: El administrador del sistema debe ser capaz de gestionar los permisos de acceso a los módulos según el rol.		

Fuente: Elaboración propia

La navegación del usuario por las funciones está atada al módulo de permisos de usuario, por tal razón los menús deben ajustarse a cada rol, es decir solo se mostrarán los enlaces hacia los módulos los cuales el administrador permita.

Resultados

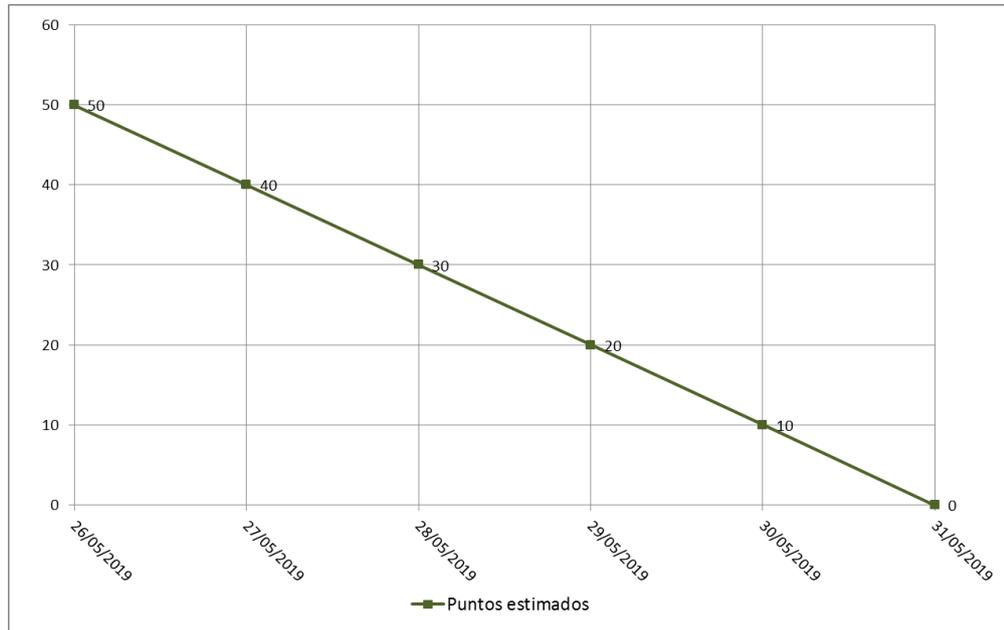


Figura 4.21. Planificación Inicial – Sprint 5

Fuente: Elaboración propia

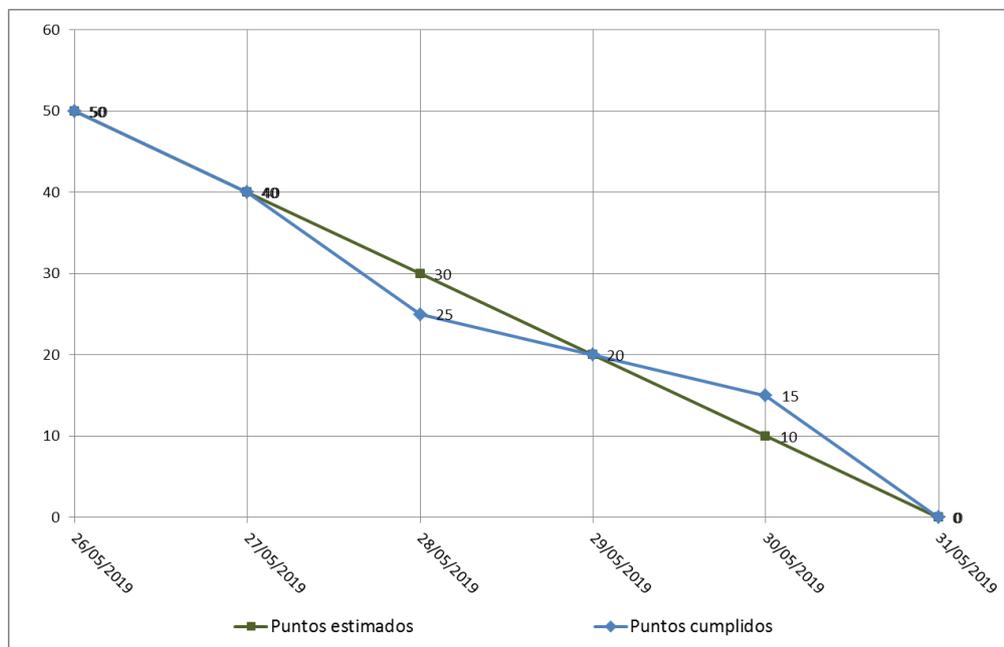


Figura 4.22. Resultados – Sprint 5

Fuente: Elaboración propia

Demostración



Figura 4.23. Resultados – Menú de navegación

Fuente: Elaboración propia

4.2 Diagrama de base de datos



Figura 4.24. Diagrama base de datos

Fuente: Elaboración propia

4.3 Diagrama de la arquitectura del sistema

La arquitectura para el sistema que el equipo de trabajo utilizó para la implementación fue cliente-servidor, el cual consiste en que el cliente realiza una solicitud al servidor a través de un navegador web, el servidor responde con un bloque caracteres acorde al formato de la solicitud y el navegador interpreta la respuesta y muestra en un formato legible para el usuario final.

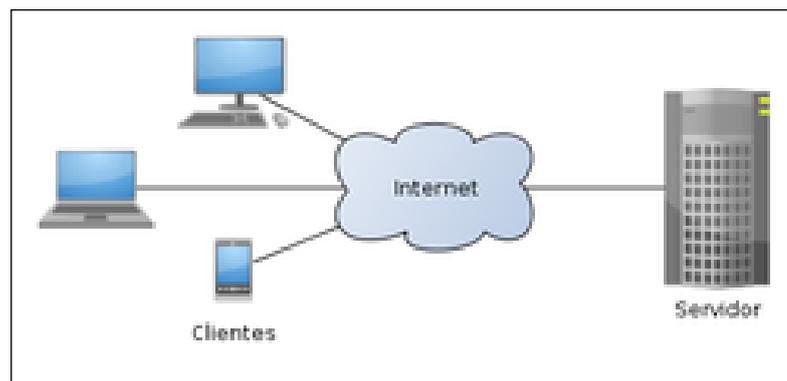


Figura 4.25. Diagrama cliente-servidor

Fuente: (Neves, 2013).

4.4 Diseño de interfaces

Prototipo de pantalla – Modulo de usuarios

Iniciar sesión

*Si tienes problemas para iniciar sesión
contactar con el Administrador del Sistema*

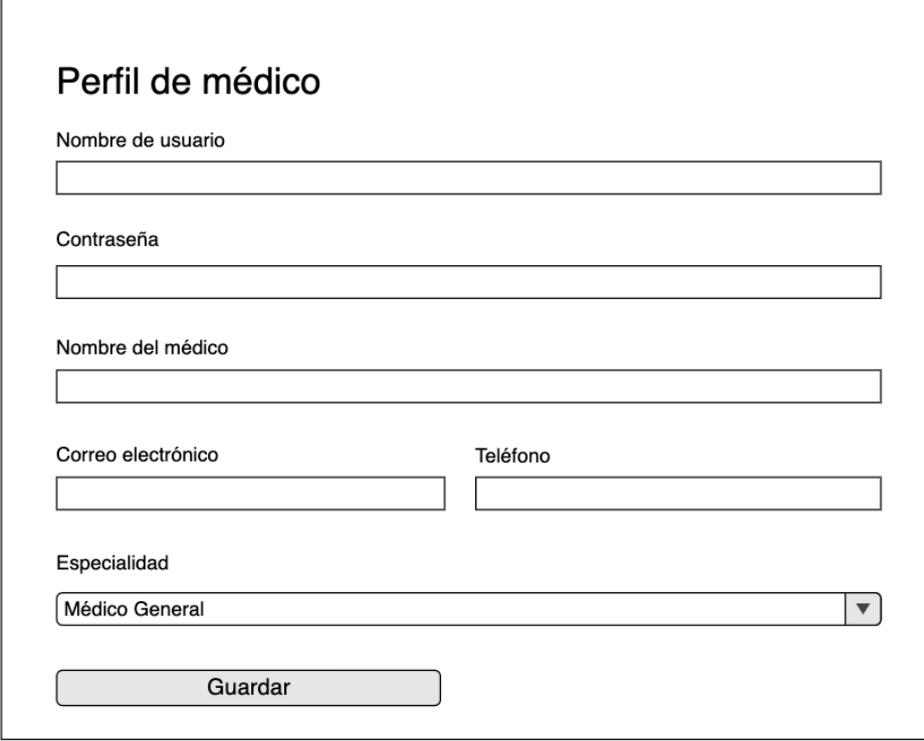
Nombre

Contraseña

Figura 4.26. Iniciar sesión

Fuente: Elaboración propia

La pantalla para iniciar sesión tiene como campos requeridos el nombre de usuario y contraseña, estos datos permiten ingresar a los módulos del sistema de forma fácil y segura.



Perfil de médico

Nombre de usuario

Contraseña

Nombre del médico

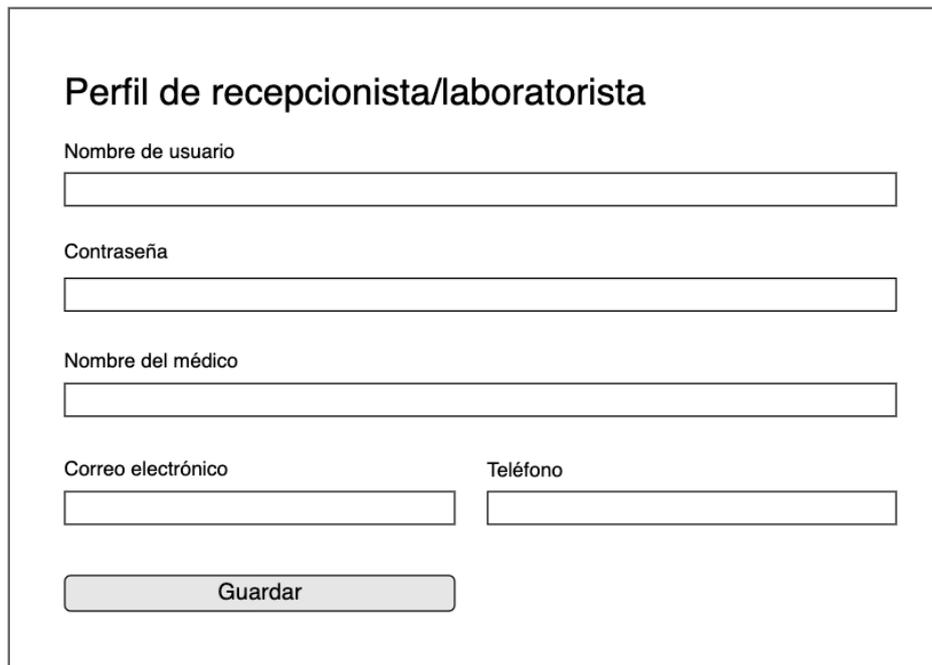
Correo electrónico Teléfono

Especialidad
Médico General ▼

Figura 4.27. Usuario médico

Fuente: Elaboración propia

La pantalla para gestionar los datos de los médicos consta de los siguientes campos: nombre de usuario, contraseña, nombre médico, correo electrónico, teléfono y especialidad.



Perfil de recepcionista/laboratorista

Nombre de usuario

Contraseña

Nombre del médico

Correo electrónico Teléfono

Figura 4.28. Usuario recepcionista/laboratorista

Fuente: Elaboración propia

La pantalla para gestionar los datos de los médicos consta de los siguientes campos: nombre de usuario, contraseña, nombre recepcionista o laboratorista, correo electrónico y teléfono.

Las pantallas anteriores describen como el usuario administrador podrá gestionar los usuarios del sistema, los cuales tiene un rol específico que asigna funciones específicas para gestionar la información generada en el sistema.

Prototipo de pantalla - Módulo de exámenes / tubos

Nuevo examen

Código

Nombre

Tubo para examen

Disponible

Figura 4.29. Formulario examen

Fuente: Elaboración propia

Las órdenes pueden tener varios exámenes y los exámenes a su vez se les asignaran un tubo de laboratorio, si un examen no está disponible físicamente por alguna razón, el usuario administrador podrá activarlos o desactivarlos con la opción llamada disponible en el formulario.

Nuevo tubo

Código

Nombre

Valor unitario

Disponible

Figura 4.30. Formulario de tubos

Fuente: Elaboración propia

Para registrar un tubo necesitamos los siguientes campos: código, nombre y valor unitario, al igual los exámenes el usuario administrador tendrá la posibilidad de marcarlos como disponible o en su defecto como no disponible.

Lista de exámenes

▼ Código	▼ Nombre del examen	▼ Activo	▼ Opciones
			<input checked="" type="checkbox"/>

Figura 4.31. Lista de exámenes

Fuente: Elaboración propia

Prototipo de pantalla - Módulo de órdenes

Nueva Orden

C.I paciente

Nombre paciente

Notaciones para laboratorista

Lista de exámenes

Nuevo examen

- Examen 1 (8.00)
- Examen 2 (6.00)
- Examen 3 (18.00)

Total Pagado

Resultados [Imprimir resultados](#) Resultados emergentes

Diagnostico

Figura 4.32. Formulario de órdenes

Fuente: Elaboración propia

Para registrar órdenes los médicos dispondrán de la pantalla anterior, los campos requeridos para ingresar una orden son: datos del paciente, notaciones para laboratorista y los exámenes requeridos por el paciente.

Lista de órdenes

Q C.I o Nombre de paciente

▼ C.I	▼ Nombre del paciente	▼ Fecha	▼ Valor	▼ Opciones
				<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>

Nueva orden

Figura 4.33. Lista de órdenes

Fuente: Elaboración propia

La lista de órdenes debe contener información útil para los usuarios de acuerdo a su función, por tal razón las columnas de esta lista variaran según el usuario conectado, esta pantalla dispone de una caja de texto para realizar búsquedas, los criterios descritos por el cliente para la búsqueda fueron cedula de identidad y nombre del paciente

Prototipo de pantalla - Menú de navegación

[Pacientes](#)
[Doctores](#)
[Recepcionistas](#)
[Laboratoristas](#)
[Salir](#)

Ecuasys

Figura 4.34. Menú de navegación

Fuente: Elaboración propia

4.5 Estándares de programación utilizados

Los estándares de programación usados en este proyecto fueron proporcionados por Ruby, los cuales son listados a continuación (Ponce Moreno, 2013):

Estándares de Ruby para definición

- **Snake case:** para nombrar métodos, todas las letras en minúscula y las palabras separadas por barra baja, ejemplo: “snake_case”.
- **Camel case:** para nombrar clases y módulos, primera letra de cada palabra en mayúscula y todas las palabras juntas, ejemplo: “CamelCase”.
- **Screaming Snake case:** para nombrar constantes, todas las letras en mayúsculas y las palabras separadas por barra baja, ejemplo: “SCREAMING_SNAKE_CASE”.

Estándares de Ruby para formato

- Usar indentación o sangría de 2 espacios (No usar Tabs).
- Usar espacios alrededor de los operadores, después de las comas, después de dos puntos, después de punto y coma, alrededor de {y antes de}.
- No usar espacios después de (, [ni antes de],).

Estándares de Ruby para sintaxis

- Usar **def** con paréntesis cuando tenga argumentos.
- Usar **&&** y **||** para expresiones booleanas.
- Usar **and** y **or** para flujos de control.

Estándares de Ruby para base de datos

Active Record utiliza algunas convenciones de nomenclatura para descubrir cómo se debe crear la asignación entre modelos y tablas de bases de datos. Ruby on Rails pluralizará los nombres de sus clases para encontrar la tabla de base de datos respectiva. Por lo tanto, para una clase **Libro**, debe tener una tabla llamada **libros**.

- **Modelo/Clase:** Article, **Tabla:** articles
- **Modelo/Clase:** LineItem, **Tabla:** line_items

Active Record usa convenciones de nomenclatura para las columnas en las tablas de la base de datos, dependiendo del propósito de estas columnas.

Claves foráneas: estos campos deben nombrarse siguiendo el patrón **singularized_table_name_id** (por ejemplo, **item_id**, **order_id**). Estos son los campos que buscará *Active Record* cuando cree asociaciones entre sus modelos.

Claves primarias: de forma predeterminada, *Active Record* utilizará una columna entera llamada **id** como clave principal de la tabla (**bigint** para PostgreSQL). Cuando use Migraciones de registros activos para crear sus tablas, esta columna se creará automáticamente (RailsGuides).

4.6 Pruebas

En esta sección se detalla las pruebas realizadas al sistema web.

4.6.1 Pruebas de funcionalidad (Aceptación de usuario)

La aceptación del cliente se evidencia en los resultados de cada iteración ejecutada y la retroalimentación, en las demostraciones periódicas que realizó el equipo de trabajo se capacitó al cliente del uso de cada módulo, las mismas se detallan en la siguiente tabla.

Tabla 4.19. *Resumen de pruebas de funcionalidad*

Resumen de pruebas de funcionalidad

Numero	Nombre del módulo	Resume prueba
1	Módulo de usuario	El equipo de trabajo mostró el funcionamiento del módulo de usuario, el cliente logró crear un usuario nuevo tipo doctor, inicio sesión con dicho usuario y cambio los datos de su perfil.
2	Módulo de exámenes y tubos de laboratorio	Las pantallas del módulo de tubos permitieron al cliente crear, editar y listar los tubos de muestra, al momento de crear exámenes el cliente logró asignar tubos y marcar como disponibles para posteriormente crear órdenes.
3	Módulo de órdenes	El cliente logró ingresar al formulario como un usuario tipo doctor para crear órdenes, en el cual seleccionó dos exámenes y un paciente, luego como un laboratorista ajuntó un archivo PDF con los resultados, como usuario recepcionista registro el pago y finalmente como usuario tipo doctor registro el diagnostico.
4	Módulo de resultados	El cliente ingresó al sistema y visualizó la pantalla de resultados.
5	Navegación y permisos de usuario	El cliente logró navegar por los diferentes módulos del sistema con enlaces de acceso directo.

Fuente: Elaboración propia

4.6.2 Pruebas de rendimiento (Aceptación técnica)

Las pruebas de rendimiento se detallan en el Anexo 1.

4.6.3 Pruebas de carga y estrés (Aceptación técnica)

Las pruebas de carga y estrés se detallan en el Anexo 2.

4.7 Implementación

En esta sección se detalla la implementación en el servidor de la empresa.

4.7.1 Plan de implementación

Los pasos a seguir para la implementación se detallan en la siguiente tabla.

Tabla 4.20. *Plan de implementación*

Plan de implementación		
Tarea	Fecha	Recursos
Instalar Ubuntu 14 Server	15/Jun/2019	Cliente Ubuntu, Servidor Web
Instalar PostgreSQL	15/Jun/2019	Cliente Ubuntu, Servidor Web
Configurar credenciales	15/Jun/2019	Cliente Ubuntu, Servidor Web
Desplegar código	16/Jun/2019	Cliente Ubuntu, Servidor Web
Ejecutar migraciones	16/Jun/2019	Cliente Ubuntu, Servidor Web
Comprobar acceso al sistema	16/Jun/2019	Cliente Ubuntu, Servidor Web

Fuente: Elaboración propia

4.7.2 Requerimientos de implementación

Software para servidor

- Ubuntu server 14
- PostgreSQL 10
- Servidor Web Nginx
- Gemas y dependencias de Ruby on Rails

Software para usuario.

- Sistema operativo Windows, MacOS, Ubuntu
- Navegador web que soporte HTML y JavaScript

Hardware para servidor

- RAM al menos de 8Gb.
- Procesador Core I5 al menos 7ma generación.
- Interfaz de Red 1Gbps.
- Disco duro de 1Tb

Hardware para usuario

- RAM al menos de 2Gb.
- Interfaz de Red 100 Mbps.

4.7.3 Manual de usuario

El manual de usuario se detalla en el Anexo 3.

4.7.4 Manual técnico

El manual técnico se detalla en el Anexo 4.

4.7.5 Plan de capacitación

El plan de capacitación elaborado por el equipo de trabajo se detalla a continuación.

Plan de capacitación

Tarea	Fecha	Personal
Socialización de la mejora del proceso	18/Jun/2019	Doctores, recepcionistas, laboratoristas, equipo de trabajo
Presentación del manual de usuario	18/Jun/2019	Doctores, recepcionistas, laboratoristas, equipo de trabajo
Demostración del flujo del proceso	19/Jun/2019	Doctores, recepcionistas, laboratoristas, equipo de trabajo
Demostración de configuraciones	20/Jun/2019	Administrador del sistema
Presentación del manual técnico	20/Jun/2019	Administrador del sistema

Fuente: Elaboración propia

CONCLUSIONES

El costo para gestionar las órdenes de laboratorio se redujo de la siguiente forma: se redujo de 5 recepcionistas a 3, ahorrando \$1400 mensuales por el sueldo de 2 recepcionistas, también se eliminaron las horas extras para laboratoristas debido a que los recepcionistas consultan los valores de los tubos en el sistema y no con llamadas a los laboratoristas, finalmente se eliminó el costo de las impresiones de órdenes.

La metodología Scrum facilita el control de incidentes en la etapa de desarrollo e implementación de un proyecto, además de entregar productos funcionales al cliente de forma periódica (Gandarillas, 2017).

La implementación del sistema Ecuasys proporciona a los involucrados una herramienta de gestión fácil, rápida y segura de las órdenes, por tal razón los pacientes de la empresa se ven beneficiados por la reducción de tiempos en la espera de resultados como se observa en el Anexo 1 y Anexo 2.

Ruby on Rails proporciona la facilidad de desarrollar e implementar sistemas web de forma modular y funcional en periodos cortos de tiempo debido a su gran cantidad de gemas disponibles y el soporte técnico de una comunidad de desarrolladores muy grande (RubyGems.org).

RECOMENDACIONES

Es importante continuar con la optimización de procesos como parte de una mejora continua en toda la empresa bajo estándares normalizados.

Se recomienda desarrollar una interfaz de programación de aplicaciones para crear una comunicación directa con los datos de sistemas ya implementados en la empresa.

Implementar un diseño adaptable en las interfaces del sistema para que los usuarios puedan utilizar el mismo en sus dispositivos móviles.

Es fundamental capacitar a los trabajadores sobre el uso de tecnología en los procesos para acortar los pasos y el trabajo en sí, de tal forma se optimizan los recursos tanto físicos como lógicos de la organización.

REFERENCIAS BIBLIOGRÁFICAS

- Coffey, A., & Atkinson, P. (2003). *Encontrar sentido a los datos cualitativos*. Colombia: Contus.
- Cuello, J., & Vittone, J. (2013). *Diseñando apps*. Barcelona: Catalina Duque Giraldo.
- Ecuared. (20 de Septiembre de 2011). *Ecuared*. Obtenido de https://www.ecured.cu/Agile_Unified_Process
- Gallardo, C. G. (2008). *Cómo funciona la Web*. Santiago de Chile: Centro de Investigación de la Web.
- Gandarillas, A. (23 de Julio de 2017). *Metodologías para el desarrollo y mantenimiento de software y sistemas de información*. Recuperado el 12 de Febrero de 2019, de <https://metodologia.es/aup/>
- García, J. (20 de Octubre de 2017). *Openwebinars*. Recuperado el 3 de Noviembre de 2018, de <https://openwebinars.net/blog/que-es-ruby/>
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Barcelona: MARCOMBO.
- Gutiérrez, R. (2006). *Introducción al Método científico*. México: Esfinge.
- Hartl, M. (2016). *Ruby on Rails Tutorial*. New York: Derek Sivers.

Intedya. (2008). *Sistema de Gestión de la Calidad en Laboratorios Clínicos ISO 15189*. Recuperado el 01 de Noviembre de 2018, de <https://www.intedya.com/internacional/73/consultoria-sistema-de-gestion-de-la-calidad-en-laboratorios-clinicos-iso-15189.html>

Krall, C. (2006). *JavaScript desde cero*. Barcelona: Aprender a programar.

Ponce Moreno, S. (2013). *Desarrollo práctico de aplicaciones web*. Madrid: RC Libros.

RailsGuides. (s.f.). *RailsGuides*. Recuperado el 15 de Marzo de 2019, de https://edgeguides.rubyonrails.org/active_record_basics.html

RubyGems.org. (s.f.). *RubyGems*. Obtenido de <https://rubygems.org/>

Sabariego, M., Dorio, I., & Massot, M. (2004). *Características generales de la investigación cualitativa*. Madrid: La Muralla.

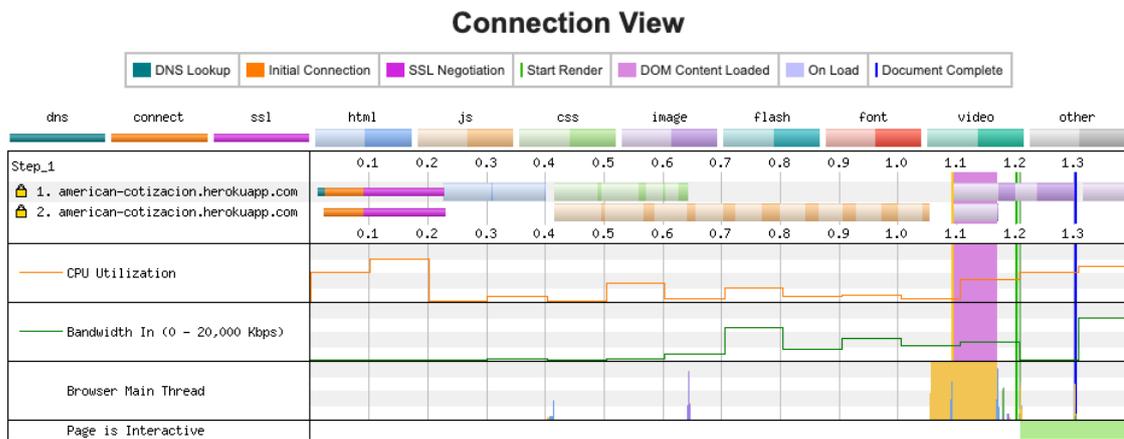
The Group PostgreSQL Global Development. (20 de Junio de 2019). *PostgreSQL*. Recuperado el 1 de Junio de 2019, de <https://www.postgresql.org/about/>

ANEXOS

Anexo 1. Pruebas de rendimiento

Las herramientas utilizadas para las pruebas de rendimiento fue *WebPageTest*, la cual es de uso libre, para esto fue necesario desplegar el sistema web en un servidor de pruebas con el dominio *american-cotizacion.herokuapp.com*.

Las pruebas de rendimiento realizadas por el equipo de trabajo se detallan continuación



En la figura anterior se muestra que la carga de las páginas tiene un máximo de 1.3 segundos y se ajusta a los requerimientos definidos inicialmente, los recursos que tomó mayor tiempo en cargar son los archivos *JS*.

Las pruebas de rendimiento demuestran que los elementos que conforman las páginas web del sistema fueron distribuidos correctamente para que los navegadores web para nuestro caso *Google Chrome* los desplieguen de forma óptima.

Request Details

Before Start Render	Before On Load	After On Load
---------------------	----------------	---------------

Request Details												
# Resource	Content Type	Request Start	DNS Lookup	Initial Connection	SSL Negotiation	Time to First Byte	Content Download	Bytes Downloaded	Certificates	Error/Status Code	IP	
1	https://american-cot...acion.herokuapp.com/	0.226 s	8 ms	65 ms	138 ms	80 ms	-	-	-	302	35.153.122.21	
2	https://american-cot...pp.com/users/sign_in	0.308 s	-	-	-	86 ms	1 ms	4.2 KB	-	200	35.153.122.21	
3	https://american-cot...85b58823c9b5507d.css	0.415 s	-	-	-	73 ms	151 ms	77.3 KB	-	200	35.153.122.21	
4	https://american-cot...e04d25e955f046b2b.js	0.416 s	-	67 ms	138 ms	77 ms	555 ms	368.5 KB	-	200	35.153.122.21	
5	https://american-cot...m/images/body-bg.png	1.084 s	-	-	-	78 ms	-	0.6 KB	-	200	35.153.122.21	
6	https://american-cot...images/header-bg.png	1.085 s	-	-	-	88 ms	123 ms	203.4 KB	-	200	35.153.122.21	
7	https://american-cot...m/images/favicon.ico	1.311 s	-	-	-	74 ms	-	0.6 KB	-	200	35.153.122.21	

En el gráfico anterior se muestra la distribución de tiempos que tomó en desplegar todas las llamadas realizadas en las páginas web del sistema, dentro de los cuales se encuentran los bloques HTML, archivos JavaScript, archivos CSS y fotos.

Anexo 2. Pruebas de carga y estrés

Las pruebas realizadas en la herramienta *LoadImpact* se detallan a continuación.

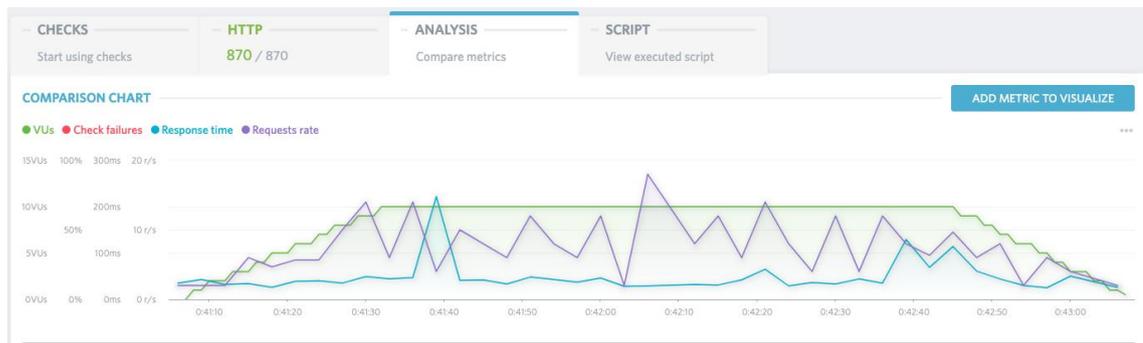
Tester: i-04e26b0b61347d142
First View only
Authenticated: admin

[Export HTTP Archive \(.har\)](#)
[Custom Metrics](#)

Load Time	First Byte	Start Render	First Contentful Paint	Visually Complete	Speed Index	First CPU Idle	Result (error code)	Document Complete			Fully Loaded		
								Time	Requests	Bytes In	Time	Requests	Bytes In
1.301s	0.399s	1.200s	1.207s	1.200s	1.200s	> 1.206s	0	1.301s	6	654 KB	1.391s	7	655 KB

Images	Colordepth	domInteractive	domContentLoaded	loadEvent
0	24	1.093s	1.093s - 1.169s (0.076s)	1.301s - 1.302s (0.001s)

En la figura anterior se muestra que el tiempo de carga promedio de las páginas son 1.301 segundos, la carga del documento HTML se completó en 1.301 segundos, las cargas posteriores y llamadas externas se completaron en 1.391 segundos, lo cual encaja en los parámetros requeridos.

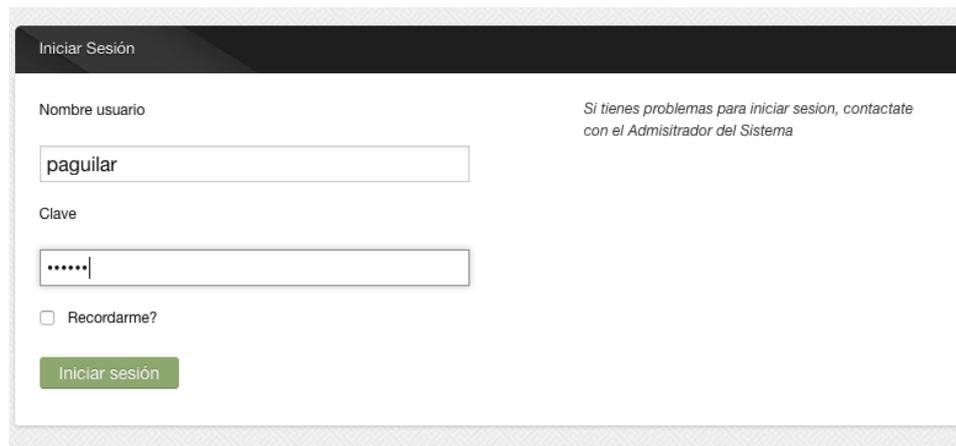


La figura anterior muestra las pruebas de estrés realizadas al sistema web, la misma consistió en generar tráfico de 10 sesiones activas durante 2 minutos de forma escalada.

Anexo 3. Manual de usuario

Inicio de sesión

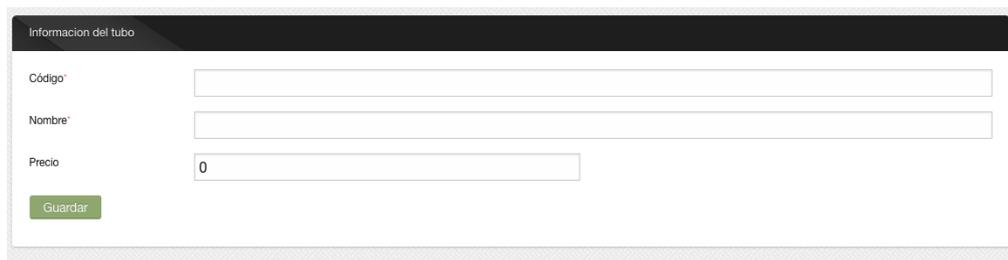
El usuario doctor y todos los usuarios pueden iniciar sesión en el siguiente formulario, que se despliega cuando se accede al sistema, el usuario debe ingresar su usuario y contraseña, que son proporcionados por el administrador del sistema



Formulario de inicio de sesión con el título "Iniciar Sesión". Incluye un campo de texto para el nombre de usuario con el valor "paguilar", un campo de contraseña con caracteres ocultos por puntos, un checkbox "Recordarme?" desactivado, y un botón "Iniciar sesión". Un mensaje de ayuda indica: "Si tienes problemas para iniciar sesión, contactate con el Administrador del Sistema".

Crear, editar tubos de muestra

Para crear tubos de muestra debe iniciar sesión como administrador del sistema y dar clic en el botón nuevo **Nuevo Tubo**, el usuario tendrá acceso al siguiente formulario.



Formulario de "Información del tubo" con campos para "Código", "Nombre" y "Precio" (valor predeterminado: 0). Incluye un botón "Guardar".

Después clic en **Guardar** para registrar el tubo de muestra en el sistema. El usuario será redirigido a una pantalla para ver el detalle del tubo y podrá editar o listar los tubos de muestra.

Información de tubo

Código: 1

Nombre: Tubo azul

Precio: \$20.00

[Editar](#) [Lista de tubos](#)

Crear, editar exámenes

Para crear exámenes debe iniciar sesión como administrador del sistema y dar clic en el botón nuevo [Nuevo Exámen](#), el usuario tendrá acceso al siguiente formulario.

Información de exámen

Código*

Nombre*

Activado?

Tubo

[Guardar](#)

Después clic en [Guardar](#) para registrar examen en el sistema. El usuario será redirigido a una pantalla para ver el detalle del examen y podrá editar o listar los exámenes.

Información de empresa

Código: 2

Nombre: Urinálisis completo

Tubo: Tubo rojo (\$ 18.00)

Activado?: SI

[Editar](#) [Lista de exámenes](#)

Editar perfiles

Para editar el perfil del usuario hacer clic en el menú de navegación en el siguiente botón [Editar perfil: Gonzalo Molina\(Administrador\)](#), se desplegará el siguiente formulario con los datos del usuario

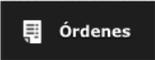
Para guardar los cambios del perfil clic en , debe ingresar la clave para poder registrar los cambios.

Usuario doctor para crear ordenes

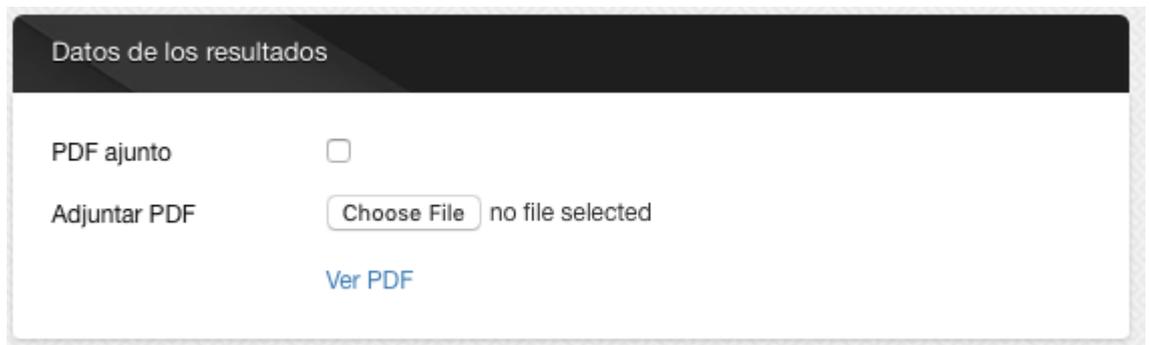
Los usuarios con el rol Doctor tendrán acceso a la lista de órdenes mediante el siguiente botón en el menú de navegación , en esta pantalla se muestra el siguiente botón para crear órdenes , cuando el usuario haga clic, se mostrará el formulario para registrar la información de órdenes.

El formulario de la orden permite seleccionar el paciente y los exámenes necesarios para crear una orden, finalmente se clic en el botón  para registrar la orden.

Usuario laboratorista para adjuntar PDF

El usuario laboratorista tendrá acceso a la lista de órdenes puede acceder con el siguiente botón , para adjuntar un PDF con los resultados de los exámenes podrá acceder al formulario con el botón editar .

El usuario podrá adjuntar el PDF con el siguiente bloque.



Datos de los resultados

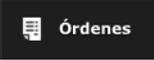
PDF ajunto

Adjuntar PDF no file selected

[Ver PDF](#)

Finalmente deberá guardar los cambios con el siguiente botón .

Usuario recepcionista para registrar pago

El usuario recepcionista tendrá acceso a la lista de órdenes para registrar el pago con el siguiente botón , para editar la orden podrá dar clic en .

El formulario para editar la orden mostrará un bloque llamado “Total orden” para registrar el pago, y podrá visualizar el monto a cancelar, el bloque tendrá el siguiente aspecto.

Total orden

Resumen totales

El total es la suma de los valores de cada examen seleccionado.

Pagado

TOTAL

Finalmente, clic en [Guardar](#) para registrar los cambios.

Acceso a módulos

Para acceder a los módulos del sistema se dispone del siguiente menú de navegación, el cual se mostrará en todas la paginas para un fácil acceso.

Ecuasys Recepcionistas Laboraristas Doctores Editar perfil: Gonzalo Molina(Administrador) Cerrar sesión

[Resultados](#) [Pacientes](#) [Órdenes](#) [Exámenes](#) [Tubos](#)

 LISTA DE DOCTORES

Anexo 4. Manual técnico

Subir código a un repositorio Git

Transferir el código de nuestra aplicación al servidor. La forma más fácil de hacerlo es a través de Git. Si ya ha configurado un repositorio Git, inserte el código de su aplicación en ese repositorio ejecutándolo en su computadora local:

```
$gitpush
```

Extracción de código

Elegir una ubicación en la que almacenar permanentemente el código de su aplicación. Una buena ubicación es `/var/www/ecua-sys`. Vamos a crear ese directorio

Instalar dependencias de la aplicación

Una aplicación Ruby tiene varias dependencias. Deben estar instalados. La mayoría de estas dependencias son gemas en su *Gemfile*.

```
$cd /var/www/ecua-sys/code  
$ bundleinstall --deployment --withoutdevelopment test
```

Compile activos de Rails y ejecute migraciones de bases de datos

Ejecutar el siguiente comando para compilar activos para la canalización de activos de Rails y para ejecutar migraciones de bases de datos:

```
$bundleexecrakeassets:precompiledb:migrate RAILS_ENV=production
```

Editar archivo de configuración de Nginx

Crear un archivo de configuración de Nginx y configurar una entrada de host virtual que apunte a su aplicación. Esta entrada de host virtual le dice a Nginx (y Passenger) dónde se encuentra su aplicación.

```
$sudo nano /etc/nginx/sites-enabled/ecua-sys.conf
```

```
server {  
listen 80;  
server_nameyourserver.com;  
  
# Tell Nginx and Passenger whereyourapp's 'public' directoryis  
root/var/www/ecua-sys/code/public;  
  
# Turn on Passenger  
passenger_enabled on;  
passenger_ruby/path-to-ruby;  
}
```

```
$sudo servicenginxrestart
```

Prueba de funcionamiento

Debería poder acceder a su aplicación a través del nombre de host del servidor, ejecutar esto desde tu computadora local.

```
$curl http://ecua-sys.com/
```

Anexo 5. Entrevista

Equipo de trabajo	Buenos días, para iniciar con la entrevista ¿Cuál es el principal problema en el proceso actual de gestión de órdenes de laboratorio?
Gerente general	El departamento de laboratorio ubicado en la matriz de la empresa receipta las órdenes de exámenes de laboratorio las cuales indican el número de análisis y su descripción para que la persona con el perfil de recepcionista de igual manera existen exámenes que tienen que ser ingresados con otras pruebas de laboratorio ya que son complemento de las anteriores y el sistema de gestión principal no da alarma cuando suceden estos particulares
Equipo de trabajo	¿Quiénes son los actores del proceso?
Gerente general	En el proceso intervienen 5 recepcionistas con un sueldo mensual \$700, 6 médicos con un sueldo mensual de \$2000 y 4 laboratoristas con un sueldo mensual de \$1100
Equipo de trabajo	¿Cuántas órdenes de laboratorio se realizan por día?
Gerente general	En un día laboral el laboratorio clínico gestiona un promedio de 100 órdenes.
Equipo de trabajo	¿La entrega de resultados se realiza de forma física, digital o ambas?
Gerente general	Los resultados son entregados de forma física y de forma digital si lo solicita el paciente,
Equipo de trabajo	¿Cómo es gestionada la confidencialidad de la información?
Gerente general	La información de los pacientes es administrada únicamente por los médicos de planta, los cuales se acogen al reglamento interno de la empresa que impide la divulgación de datos.
Equipo de trabajo	¿La empresa cuenta con los recursos necesarios para la implementación de un sistema web?
Gerente general	Si, la empresa posee un departamento de sistemas con profesionales capaces de gestionar sistemas web tanto en el ámbito de software y hardware.