

UNIVERSIDAD TECNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Sistemas Computacionales

**ESTUDIO DE LA INTEGRACIÓN DE YII2 Y REACT.JS PARA LA ELABORACIÓN
DE UN SISTEMA DE TIPIFICACIÓN PARA LA CRUZ ROJA ECUATORIANA JUNTA
PROVINCIAL DE IMBABURA.**

**Trabajo de grado presentado ante la Ilustre Universidad Técnica del Norte previo
a la obtención del título de Ingeniero en Sistemas Computacionales.**

Autor:

Carlos David Aguirre Benavides

Director:

Ing. Msc. Cosme McArthur Ortega Bustamante

Ibarra-Ecuador

Febrero 2019



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1.- IDENTIFICACIÓN DE LA OBRA

En cumplimiento de Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003770482		
APELLIDOS Y NOMBRES:	AGUIRRE BENAVIDES CARLOS DAVID		
DIRECCIÓN:	IBARRA, LOS CEIBOS RIO CHIMBO 2-18 Y RIO CURARAY		
EMAIL:	cdaguirreb@utn.edu.ec		
TELÉFONO FIJO:	062640503	TELÉFONO MOVIL	0996716563

DATOS DE LA OBRA	
TÍTULO:	ESTUDIO DE LA INTEGRACIÓN DE YII2 Y REACT.JS PARA LA ELABORACIÓN DE UN SISTEMA DE TIPIFICACIÓN PARA LA CRUZ ROJA ECUATORIANA JUNTA PROVINCIAL DE IMBABURA.
AUTOR (ES):	AGUIRRE BENAVIDES CARLOS DAVID
FECHA: DD/MM/AAAA	
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
DIRECTOR:	MSC. COSME MCARTHUR ORTEGA BUSTAMANTE

2.- CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de febrero de 2019

EL AUTOR:



Nombre: Carlos David Aguirre Benavides

CERTIFICACIÓN DIRECTOR

Por medio del presente certifico que el Sr. Carlos David Aguirre Benavides, portador de la cédula de identidad Nro. 100377048-2 ha trabajado en el desarrollo del proyecto de tesis: "ESTUDIO DE LA INTEGRACIÓN DE YII2 Y REACT.JS PARA LA ELABORACIÓN DE UN SISTEMA DE TIPIFICACIÓN PARA LA CRUZ ROJA ECUATORIANA JUNTA PROVINCIAL DE IMBABURA.", previo a la obtención del título de Ingeniero en Sistemas Computacionales, lo cual ha realizado en su totalidad con responsabilidad.

Es todo cuanto puede certificar en honor a la verdad.



MSc. Cosme Ortega Bustamante
DIRECTOR DE TESIS

EL SEÑOR ÁLVARO GUERRA VEGA, EN CALIDAD DE PRESIDENTE DE LA JUNTA PROVINCIAL CRUZ ROJA DE IMBABURA, A PETICIÓN VERBAL DEL INTERESADO TIENE A BIEN:

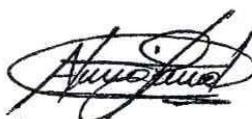
CERTIFICAR:

Que: El Señor Carlos David Aguirre Benavides, portador de la cédula de ciudadanía 100377048-2; culminó en la totalidad con el desarrollo de su tesis "ESTUDIO DE LA INTEGRACIÓN DE YII2 Y REACT.JS PARA LA ELABORACIÓN DE UN SISTEMA DE TIPIFICACIÓN PARA LA CRUZ ROJA ECUATORIANA JUNTA PROVINCIAL IMBABURA." Cumpliendo con todos los requisitos solicitados.

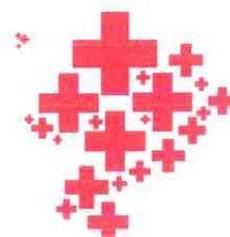
Es todo cuanto puedo certificar, facultando a la interesada hacer uso del presente documento como estime conveniente; excepto en trámites legales.

Ibarra, 18 de Febrero del 2019.

Atentamente,



Sr. Álvaro Guerra Vega
**PRESIDENTE JUNTA PROVINCIAL
CRUZ ROJA DE IMBABURA
CC. 100285342-0**



Dedicatoria

No sabría cómo empezar esto pero, en este tiempo tengo la motivación más grande para salir adelante y no es que no quiera dedicar este trabajo a nadie más pero a mi madre por su constante lucha y por enseñarme que el amor a la familia es la fuerza más grande que podemos poner; a ella que después de tanto tiempo que me guio, me cuido y protegió; creo que ha llegado mi turno de devolver un poco de todo eso que hizo por mí y que ahora ella es mi fuerza mi vida mi Maru mi todo y no quiere decir que se convirtió en una carga, más bien todo lo contrario se convirtió en pilar y la motivación más importante para culminar esta nueva etapa en mi vida.

A mi padre Juan que ha demostrado que el amor a la familia no debe terminar nunca que la promesa de vida que hizo cuando formo la familia sigue en pie y no ha decaído y que a lo largo de mi vida me ha enseñado y no solo engrandeciendo mis conocimientos sino también complementándome como persona formándome con los mejores valores.

A Joha y Ángel mis negros para ellos que no he sido el mejor ejemplo de vida, pero quiero que sepan que nunca he querido otra cosa que no sea lo mejor para ellos que traten de tomar lo mejor de mí y que eso sirva de ejemplo que la constancia y la dedicación puede hacer que cumplamos nuestras metas y que también entiendan que yo estaré para ellos en todo momento y lugar.

A toda mi familia y amigos que directa o indirectamente me apoyaron a lo largo de este largo camino y vaya que largo; para ellos que me apoyaron y ahora están celebrando conmigo este paso más en mi vida.

Sinceramente les dedico esto y más.

Carlos Aguirre Benavides

Agradecimientos

A la Universidad Técnica del Norte por abrirnos las puertas y la mente para poder obtener el título universitario

A todos los docentes que a lo largo de la carrera han sabido transmitir su sabiduría y con los cuales he formado grandes amistades; y a todas las personas que directa o indirectamente han estado apoyando y haciendo que esta meta ahora pueda ser cumplida.

A mis amigos que hombro a hombro hemos luchado por culminar esta etapa de la vida, por todas las experiencias vividas.

Tabla de Contenido

IDENTIFICACIÓN DE LA OBRA.....	II
CONSTANCIAS.....	III
CERTIFICACIÓN DIRECTOR	IV
Dedicatoria.....	VI
Agradecimientos	VII
Tabla de Contenido.....	VIII
Índice de Ilustraciones.....	X
Índice de Tablas.....	XII
Resumen.....	XIII
Abstract.....	XIV
INTRODUCCIÓN.....	1
Antecedentes	1
Situación Actual	2
Prospectiva	2
Justificación.....	2
Planteamiento del Problema	3
Objetivos.....	3
Objetivo General.....	3
Objetivos Específicos.....	3
Alcance	3
Funciones del aplicativo a desarrollar	4
CAPITULO I.....	6
1. Marco Teórico	6
1.1. Cruz Roja Ecuatoriana Junta Provincial de Imbabura.....	6
1.2. Framework PHP	8
1.3. Yii2.....	9
1.4. React.js.....	17
1.4. Servidor de aplicaciones XAMPP	21
1.5. Base de Datos MySQL.....	21
1.6. Composer	23
1.7. Editor de texto.....	24
1.8. Lenguajes de programación	25
1.9. Metodologías Agiles.....	27
CAPÍTULO II.....	33
2. Desarrollo.....	33

2.7.	Definición de roles y responsabilidades	33
2.8.	Definición de los integrantes del equipo de trabajo	34
2.9.	Historias de usuario y diseño	34
2.10.	Desarrollo de las historias de usuario	50
CAPITULO III.....		69
3.	Resultados	69
3.1.	Análisis Impacto	70
3.2.	Beneficio de la investigación.	72
3.3.	Beneficios del uso del sistema en CRE JPI.....	72
3.	Conclusiones y Recomendaciones.....	73
4.1.	Conclusiones.	73
4.2.	Recomendaciones.	74
5.	Bibliografía	75

Índice de Ilustraciones

Ilustración 1 Módulos sistema de tipificación	4
Ilustración 2: Flujograma proceso actual para tipificación.	7
Ilustración 3: Flujograma del nuevo proceso de tipificación.....	8
Ilustración 4 Estructura estática de aplicación Yii.....	11
Ilustración 5: diagrama de ciclo de vida de una petición	12
Ilustración 6: Log Instalación Yii2.....	13
Ilustración 7 Estructura de archivos de la plantilla avanzada	13
Ilustración 8: Configuración de Gii	14
Ilustración 9: Generador de código Gii.....	15
Ilustración 10: Generador de Código Gii / Generar Modelo.....	16
Ilustración 11 Estructura básica HTML	26
Ilustración 12 Ejemplo PHP	27
Ilustración 13: Comparativa ciclos de desarrollo entre metodologías	28
Ilustración 14: Modelo de datos - Tabla Usuario	35
Ilustración 15: Formulario de ingreso al sistema	36
Ilustración 16: Formulario administración de usuarios.....	37
Ilustración 17: Modelo de Datos - Tabla Cliente y relaciones.....	38
Ilustración 18: Listado de pacientes	38
Ilustración 19: Registro de pacientes	39
Ilustración 20: Edición de pacientes.....	40
Ilustración 21: Borrar Pacientes	41
Ilustración 22: Impresión de Carnet	41
Ilustración 23: Formulario de Búsquedas.....	43
Ilustración 24: Exportación de datos filtrados.....	43
Ilustración 25: Consulta de pacientes registrados.....	44
Ilustración 26: Caso de Uso / Usuario Administrador	45
Ilustración 27: Caso de Uso / Usuario Técnico.....	46
Ilustración 28: Caso de Uso / Usuario Final	47
Ilustración 29: Diagrama de arquitectura del sistema.....	49
Ilustración 30: Modelo entidad relación BDD	50
Ilustración 31: Configuración de Base de Datos en Yii.....	51
Ilustración 32: Edición Archivo Host de Windows.....	51
Ilustración 33: Edición del archivo httpd-vhost.config del servidor de aplicaciones	52
Ilustración 34: Generando los modelos con Gii.....	53
Ilustración 35: Creación de Controladores *.php.....	53
Ilustración 36: Formulario Login.....	54

Ilustración 37: Administración de Usuarios	54
Ilustración 38: Formulario para cambio de contraseña.....	54
Ilustración 39: Muestra de errores de campos incompletos en Login	56
Ilustración 40: Mensaje de error usuario inexistente.	56
Ilustración 41: Mensaje de error contraseña incorrecta.....	56
Ilustración 42: Mensaje de error no se puede guardar un usuario nuevo.	56
Ilustración 43: Pantalla de bienvenida.....	57
Ilustración 44: Lista de Usuarios registrados.....	57
Ilustración 45: Pantalla añadir clientes.....	58
Ilustración 46: Pantalla para editar cliente.....	59
Ilustración 47: Mensaje para confirmar eliminar cliente.....	59
Ilustración 48: Pantalla para impresión de carnet.....	59
Ilustración 49: Advertencia completa este campo añadir/editar clientes.....	61
Ilustración 50: Advertencia ingresar formato correcto de identificación.	61
Ilustración 51: Advertencia ingresar formato correcto de teléfono.	61
Ilustración 52: Advertencia ingresar formato correcto de email	62
Ilustración 53: Advertencia ingresar formato correcto de fecha.	62
Ilustración 54: Advertencia seleccionar un campo de lista.	62
Ilustración 55: Mensaje al añadir correctamente un cliente.....	63
Ilustración 56: Pantalla al añadir un nuevo cliente.	63
Ilustración 57: Mensaje edición correcta de clientes	63
Ilustración 58: Formulario de búsquedas	64
Ilustración 59: Pagina de búsqueda datos filtrados	65
Ilustración 60: Mensaje confirmación exportación registros filtrados.	65
Ilustración 61: Formulario consulta externa.....	66
Ilustración 62: completar campo en consulta externa.....	67
Ilustración 63: Mensaje de error no se encuentran coincidencias.	67
Ilustración 64: Pantalla de consulta con datos correctos.....	68
Ilustración 65: APIs consumidas en el Login.....	69
Ilustración 66: APIs consumidas en la gestión de usuarios.....	69
Ilustración 67: APIs consumidas en la gestión de clientes	70
Ilustración 68: APIs consumidas en la búsqueda.	70

Índice de Tablas

Tabla 1 Herramientas de Desarrollo	5
Tabla 2 Cuadro Comparativo de las características de los framework.	9
Tabla 3 Definición de Roles	33
Tabla 4 Equipos de Trabajo	34
Tabla 5: Historia de Usuario 1	34
Tabla 6: Tarea 1 - Historia de usuario 1	35
Tabla 7: Tarea 2 - Historia de Usuario 1	36
Tabla 8: Historia de usuario 2	37
Tabla 9: Tarea 1 - Historia de Usuario 2	38
Tabla 10: Tarea 2 – Historia de Usuario 2	39
Tabla 11: Tarea 3 - Historia de Usuario 2	40
Tabla 12: Tarea 4 - Historia de Usuario 2	40
Tabla 13: Tarea 5 - Historia de Usuario 2	41
Tabla 14: Historia de Usuario 3.....	41
Tabla 15: Tarea 1 – Historia de Usuario 3.....	42
Tabla 16: Tarea 2 - Historia de Usuario 3	43
Tabla 17: Historia de Usuario 4.....	44
Tabla 18: Tarea 1 - Historia de Usuario 4	44
Tabla 19: Descripción Caso de Uso / Usuario Administrador.....	45
Tabla 20: Descripción Caso de Uso / Usuario Técnico.	47
Tabla 21: Descripción Caso de Uso / Usuario Final	48
Tabla 22: Historial de revisiones - Historia de usuario 1.....	55
Tabla 23: Historial de revisiones - Historia de usuario 2.....	60
Tabla 24: Historial de revisiones - Historia de usuario 3.....	64
Tabla 25: Historial de revisiones - Historia de usuario 4.....	66
Tabla 26: Costo examen tipificación	71
Tabla 27: Costo canje carnet	71

Resumen

El avance constante que tiene la tecnología ha hecho que la sistematización de procesos tome fuerza, en la actualidad los programadores están obligados a tener una mejora constante en el desarrollo de sus aplicaciones y a optar por nuevas herramientas o a su vez combinarlas para obtener los resultados esperados.

El presente trabajo se ha desarrollado para poder obtener una respuesta ante la interrogante de si combinar un Framework PHP y una librería JavaScript es una apuesta oportuna para el desarrollo de un aplicativo que ayude a la sistematización del proceso de Tipificación de la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.

En el capítulo de introducción se detalla de manera rápida una descripción general del proyecto el cual contiene sus generalidades, objetivos problemas a resolver y su justificación.

En el Capítulo I, se plasma la investigación bibliográfica realizada acerca del framework y la librería a utilizarse, herramientas de desarrollo manejadas para la implementación del aplicativo, los procesos a sistematizar, y la metodología de programación.

En el Capítulo II, se muestra el desarrollo del aplicativo siguiendo los parámetros de la metodología eXtreme Programming, la implementación y pruebas del sistema.

En el Capítulo III, se detalla el impacto económico de la sistematización del proceso y los resultados obtenidos de la integración de Yii2 y React.js.

Abstract

The constant advance that technology has made the systematization of processes take force, at present the programmers are forced to have a constant improvement in the development of their applications and to opt for new tools or in turn combine them to obtain the expected results.

This work has been developed to get an answer to the question of whether combining a PHP Framework and a JavaScript library is a timely bet for the development of an application that helps the systematization of the process of Tipificación de la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.

In the introductory chapter, a general description of the project is briefly described, which contains its generalities, objectives, problems to be solved and its justification.

In Chapter I, the bibliographic research carried out on the framework and the library to be used, development tools used for the implementation of the application, the processes to be systematized, and the programming methodology are shown.

In Chapter II, the development of the application is shown following the parameters of the eXtreme Programming methodology, the implementation and testing of the system.

In Chapter III, the economic impact of the systematization of the process and the results obtained from the integration of Yii2 and React.Js are detailed

INTRODUCCIÓN

Antecedentes

En el anteproyecto de esta investigación se planteó el tema: **“ESTUDIO DE LA INTEGRACIÓN DE YII2 Y REACT.JS PARA LA ELABORACIÓN DE UN SISTEMA DE GESTION DE TALENTO HUMANO PARA LA CRUZ ROJA ECUATORIANA JUNTA PROVINCIAL DE IMBABURA”** pero la empresa auspiciadora adquirió un reloj biométrico el mismo que cuenta con un software que realiza las tareas que mi prototipo de tesis iba a realizar es por esto que en se solicitó un cambio de tema de tesis a: **“ESTUDIO DE LA INTEGRACIÓN DE YII2 Y REACT.JS PARA LA ELABORACIÓN DE UN SISTEMA DE TIPIFICACIÓN PARA LA CRUZ ROJA ECUATORIANA JUNTA PROVINCIAL DE IMBABURA”** el mismo que fue aprobado por el Honorable Consejo Universitario en la Resolución: **UTN-FICA-2018-0034** por tal: la situación actual de la empresa, los objetivos, el alcance, las funciones del aplicativo fueron cambiados.

Las aplicaciones web han optimizado el trabajo de las empresas ya que las mismas se adaptan a cualquier plataforma lo único que necesitamos es un navegador web y conexión a internet lo cual permite trabajar en cualquier lugar y a cualquier hora.

Las aplicaciones web no necesitan ser instaladas permitiendo así un acceso inmediato sin importar la configuración del equipo o su hardware y por otra parte permiten a múltiples usuarios utilizar la aplicación al mismo tiempo.

Las necesidades de crear aplicaciones web llamativas y funcionales a lo largo del tiempo han hecho que los desarrolladores obtén por mezclar o integrar varias herramientas.

Los frameworks PHP son un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. (F. Sierra, 2013)

Javascript es un lenguaje interpretado usado para múltiples propósitos, pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudo a cambiar el modo que vemos Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades en ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confiar en el lenguaje Javascript como la mejor opción para la web. (Gauchat, 2012)

La Cruz Roja Ecuatoriana es Organización No Gubernamental (ONG) sin fines de lucro cuyos principios de ayuda humanitaria y a la comunidad son valores reconocidos en la sociedad. Su misión es ayudar a los sectores más vulnerables, movilizandoo el poder de la humanidad y mediante trabajos sin fines de lucro que generen auto sustentabilidad, llegar a todas las personas de diferentes etnias, nacionalidades y culturas. La Junta Provincial de la Cruz Roja de Imbabura, se suma a este esfuerzo colectivo con la finalidad de mejorar la salud de la población. (Vega, 2017)

Situación Actual

La necesidad de crear aplicaciones más robustas y más amigable para el usuario ha llevado a que el desarrollo web de pasos gigantes en avances; el uso de dispositivos conectados a una red cada vez es más frecuente y hasta se podría decir que indispensable por esto y más los desarrolladores han optado por combinar tecnologías y herramientas para poder satisfacer las exigencias del cliente y permitirle ejecutar sus sistemas desde cualquier dispositivo sin importar la configuración de hardware. Para esta investigación estudiaremos la integración de un Framework PHP (Yii2) y una librería Javascript (React.js).

En la actualidad la Cruz Roja Ecuatoriana Junta Provincial de Imbabura (CRE JPI) no cuenta con un sistema capaz de gestionar los datos que genera el servicio de tipificación del Laboratorio Clínico de la institución por esto que se ve la necesidad de crear un software capaz de generar un certificado de tipo de sangre y que permita al administrador generar reportes y de esta manera se automatizara un proceso tedioso y que conlleva mucho tiempo.

Prospectiva

La presente investigación servirá como un apoyo o base a futuras investigaciones, para desarrolladores en todos los niveles y que los mismos puedan aplicar de manera eficaz cada una de estas herramientas (Yii2 y React.js) ayudando así a que las aplicaciones web tradicionales se queden atrás.

Justificación

En la actualidad los estudios relacionados a la integración de Yii2 y React.js son muy pocos y mediante este estudio se pretende orientar a los desarrolladores a escoger las herramientas correctas; las mismas que ayudaran a realizar con mayor eficiencia las aplicaciones y cuenten con soportes a las exigencias de manera que el desarrollador o ingeniero de software pueda tener más tiempo de generar ideas y no se enfoque a la toma de decisión de las herramientas a utilizar para que de esta manera el software sea innovador.

Por esta razón se realiza el estudio de la integración del Framework Yii2 y la Librería React.js ya que estas herramientas actualmente son de las más utilizadas para el desarrollo web, destacando también que las dos herramientas son de código abierto, ofreciendo un soporte continuo, actualización y documentación en línea constantemente.

Planteamiento del Problema

La poca documentación a nivel global que hace referencia técnica, artículos científicos e investigaciones sobre la integración del Framework Yii2 y al Librería React.js para el desarrollo de aplicaciones web limita a los desarrolladores o ingenieros en software a construir aplicaciones más amigables al usuario y más veloces lo cual implica que el software desarrollado podría dejar de ser utilizado ya que no cumple con las exigencias y los estándares requeridos en determinado momento por los usuarios finales.

La falta de documentación sobre la integración de del Framework Yii2 y la librería de JavaScript React.js han obligado a los desarrolladores a elegir en la mayoría de las veces herramientas que no cumplen con sus expectativas.

Objetivos

Objetivo General

- Estudiar la integración del framework Yii2 y la librería React.Js para el desarrollo de una aplicación web del servicio de Tipificación de la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.

Objetivos Específicos

- Estudiar el Framework Yii2.
- Estudiar la librería React.js
- Integrar el Framework Yii2 y la librería React.js para la creación del Frontend y el Backend de la aplicación Web.

Alcance

Se estudiará la integración de del Framework Yii2 y la librería de JavaScript React.js. y se detallará de manera ordenada como funcionan dichas herramientas y todo acerca de sus características y por qué se debería escoger las herramientas.

El sistema de tipificación permitirá almacenar los datos de los clientes que acceden al servicio de tipificación en el laboratorio de la CRE JPI, así como también entregar un carnet de tipo de sangre.

El sistema web será accesible desde cualquier dispositivo conectado a la red Interna y externa (internet) de la CRE JPI.

Para el desarrollo del sistema de tipificación se utilizará una arquitectura MVC (modelo – vista - controlador).

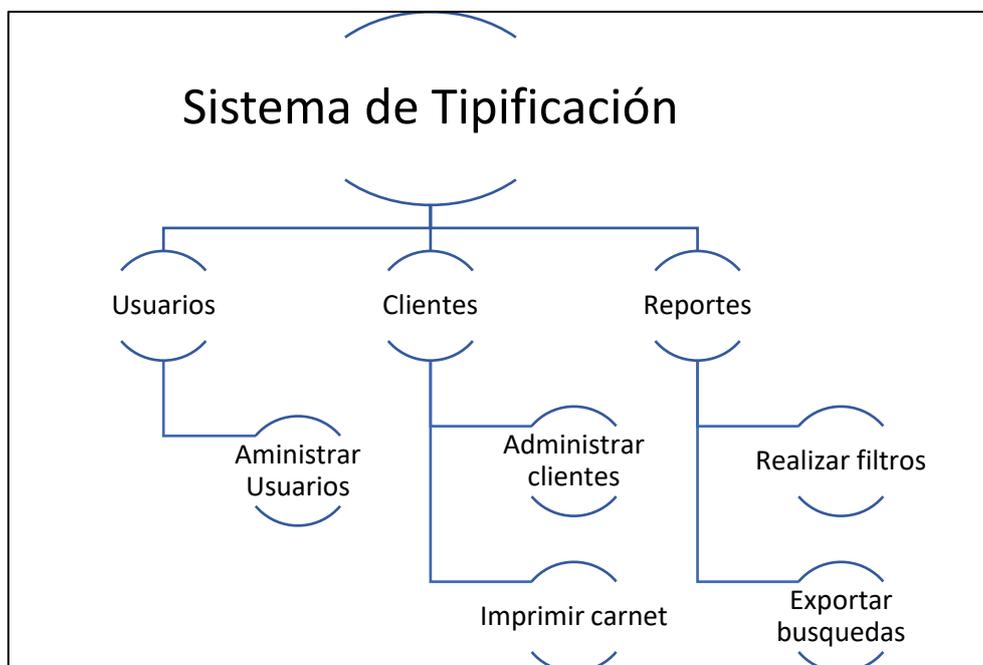
En general el alcance de la investigación es integrar el Framework Yii2 y la librería de JavaScript React.js para desarrollar el Sistema de Tipificación de la CRE JPI.

Funciones del aplicativo a desarrollar

El sistema manejará dos tipos de roles: administrador y técnico, cada uno de ellos tendrá acceso a áreas específicas del sistema dependiendo el rol.

El sistema almacenara la información en una base de datos MySQL.

Ilustración 1 Módulos sistema de tipificación



Fuente: Propia

- **Módulo Usuarios:** En este módulo podrá ser accedido solamente por el administrador del sistema y en este se añadirá, editará o se dará de baja usuarios técnicos y administradores.
- **Módulo Clientes:** Aquí se añadirá, editará o se eliminará clientes, se imprimirá el carnet de Tipo de Sangre y se buscaran los clientes por su número de identificación.

- **Modulo reportes:** Se podrá filtrar búsquedas por “Datos de cliente”, “Datos sanguíneos”, “Datos de auditoria”, “Datos geográficos” y se podrán exportar estos datos filtrados a un archivo “.CVS” a petición de la CRE JPI.

Se detallan las herramientas que se utilizaran en el desarrollo del sistema.

Tabla 1 Herramientas de Desarrollo

Base de Datos	MySQL
Lenguaje de desarrollo	PHP
Servidor de aplicaciones	XAMPP
Framework	Yii2
Librería JavaScript	React.js

Fuente: Propia

Xtreme Programminng (XP) es la metodología escogida para el desarrollo de este sistema, pues se adapta perfectamente en ambientes de trabajo pequeños y está enfocado a cumplir las tareas realmente necesarias.

CAPITULO I

1. Marco Teórico

El avance que ha sufrido el desarrollo web en los últimos años debido al incremento de dispositivos conectados a una red a producido que los desarrolladores o los ingenieros de software se vean en la necesidad de buscar nuevas herramientas e inclusive que integren herramientas para crear software que cumpla con las exigencias de los clientes.

El estudio del Framework Yii2 y la librería React.js ayudan a diseñar elaborar e implementar aplicaciones o sistemas web, lo cual abierto las puertas de la tecnología ágil ya que su fácil uso reduce el tiempo de la producción del software.

En este capítulo se definirá de forma teórica las herramientas y tecnologías a utilizar, para de esta manera tener un conocimiento amplio y previo de dichas herramientas y que esto sirva como referencia para el desarrollo del Sistema de Tipificación de la CRE JPI. Las herramientas esenciales que estudiaremos en este documento serán el Framework Yii2 y la librería React.js.

En capítulos siguientes se ahondará en el estudio de la integración de estas dos tecnologías y el uso correcto y oportuno de cada una de ellas de este modo se pretende que esta investigación sirva como referencia a futuros desarrollos que utilicen estas tecnologías y ayuden a los desarrolladores o ingenieros de software a obtener una amplia prospectiva sobre el manejo del Framework Yii2 y la librería React.js y así adquirir conocimientos sólidos.

Cabe mencionar que lo que a continuación se establece fue descrito en la introducción, el Framework Yii2 y la librería React.js y demás herramientas se emplearan en el transcurso del presente proyecto.

1.1. Cruz Roja Ecuatoriana Junta Provincial de Imbabura

(Perugachi, 2018) La CRE JPI es una ONG sin fines de lucro que se encuentra ubicada en Ecuador provincia de Imbabura cantón Ibarra en la dirección Eugenio Espejo 9-61 y Juan de Velazco, la cual brinda actualmente servicios orientados al mercado de la salud entre los cuales están:

- Laboratorio Clínico.
- Odontología
- Rayos X Dentales
- Rehabilitación Física

- Optometría
- Servicio de Ambulancia
- Servicio de enfermería y vacunas
- ADN

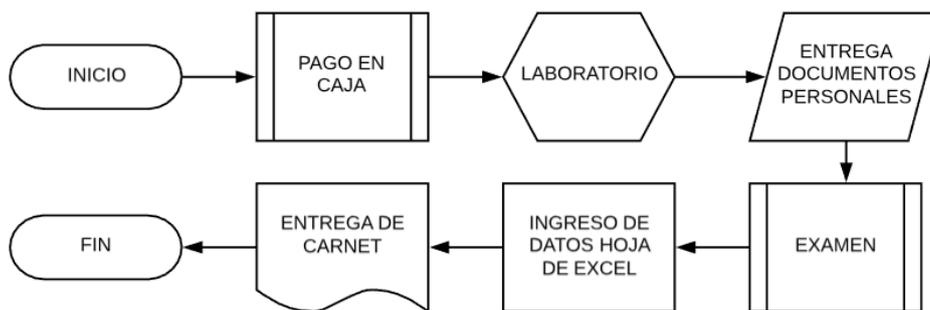
1.1.1. Servicio de laboratorio clínico

El laboratorio de la CRE JPI al momento ofrece 265 exámenes y uno de estos es el de tipificación sanguínea siendo este el de más demanda dentro del servicio con actualmente un promedio de 940 tipificaciones y 23 canjes al mes obteniendo como ingresos económicos un valor promedio mensual de \$4.412,83. (Cuaran, 2018)

Examen de Tipo de Sangre (proceso actual)

Para este examen en específico existe un proceso el cual se lo realiza de forma manual el mismo que comprende que el cliente cancela por el examen en caja, se acerca al laboratorio, entrega su factura y su cedula en el mediante un formato elaborado en Excel se ingresa solo su nombre completo, fecha actual y factor RH que corresponda, una vez realizado este proceso se imprime en un formato predeterminado de carnet impreso con anterioridad el cual contiene la información como el nombre de la institución el tipo de sangre (A, B, O, AB) y se procede a firmar por el técnico que realizo la prueba de tipo de sangre, para un nuevo cliente se repite el proceso sin guardar ningún registro de los clientes.

Ilustración 2: Flujograma proceso actual para tipificación.



Fuente: Propia

Examen de Tipo de Sangre (proceso nuevo)

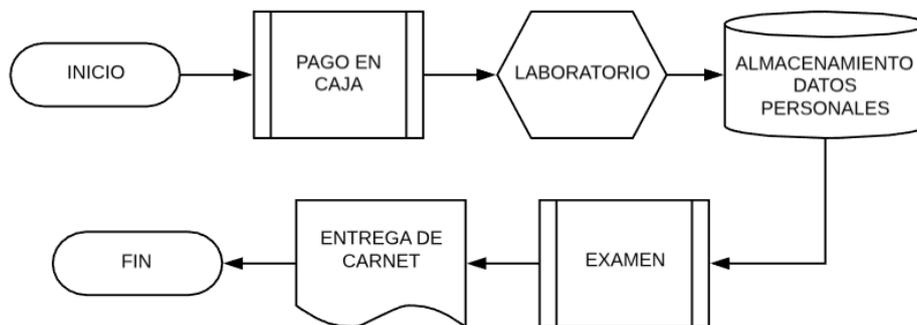
El proceso de pago será el mismo, lo que distingue este proceso del anterior es el almacenamiento de datos como:

- Cedula
- Nombres

- Apellidos
- Fecha de nacimiento
- Dirección
- Teléfono
- Email
- Ciudad de residencia
- Tipo de sangre
- Factor RH

Los mismos que serán almacenados en la base de datos y permitirán a la institución acceder a ellos para poder entregar canjes de certificado y determinar la cantidad de pacientes atendidos en rangos de fechas establecidas, los técnicos que realizaron la prueba y todos los reportes que permitan extraerse con los datos ingresados.

Ilustración 3: Flujograma del nuevo proceso de tipificación.



Fuente: Propia.

1.2. Framework PHP

Las plataformas para el desarrollo de aplicaciones web como herramientas facilitadoras para el desarrollador, brindan una base sólida para la construcción de esta. Los llamados frameworks para PHP, son un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Son un conjunto de archivos, en este caso PHP, que vienen preparados con toda la estructura necesaria para desarrollar varios tipos de proyectos. (F. Sierra, 2013)

1.2.1. Comparativa de Yii 2 con Otros Frameworks

(F. Sierra, 2013) A la hora de un desarrollo de software cualquiera; ¿Cómo saber que framework PHP escoger? Es por eso, que lo adecuado es que las personas que desarrollen PHP conozcan su proyecto para elegir el framework adecuado y así ahorrar una gran cantidad de trabajo.

Para poder elegir un framework frente a otro debemos conocer algunos de estos por lo que a continuación se detallan las características de Kumbia, Wasp, CodeIgniter, Seagull, BlueShoes, Qcodo, Akelos, PhpOpenbiz, Zoop, Ash.MVC, Diy, Wact, Zend Framework, Yii, Laravel, CakePHP y Symfony.

Tabla 2 Cuadro Comparativo de las características de los framework.

Framework	PHP4	PHP5	MVC	Múltiple BD	ORM	Plantillas	Cache	Ajax
KumbiaPHP		*	*	*	*	*		*
Wasp		*	*			*		*
CodeIgniter	*	*	*	*		*	*	
Seagull	*	*	*	*	*	*	*	*
BlueShoes	*	*	*	*		*		
Qcodo		*	*	*		*		*
Akelos	*	*	*	*	*	*	*	*
PhpOpenbiz		*	*	*	*	*		*
Zoop	*	*	*	*		*	*	*
Ash.MVC		*	*			*		
Diy		*	*		*	*	*	*
Wact	*	*	*	*		*		
Zend Framework		*	*	*		*		*
Yii	*	*	*		*	*		*
Laravel		*	*	*	*	*	*	*
CakePHP	*	*	*	*		*	*	*
Symfony		*	*	*		*		*

Fuente: (F. Sierra, 2013)

Como podemos apreciar en la Tabla 2 todos los Framework soportan PHP 5 o superior, cuentan con una arquitectura MVC y de la misma forma todos tienen a disposición del desarrollador plantillas. Para este estudio analizaremos únicamente el framework Yii en su segunda versión.

1.3. Yii2

Yii es un framework de PHP basado en componentes de alto rendimiento para el desarrollo rápido de aplicaciones web modernas. El nombre Yii (pronunciado Yee o [ji:]) significa "simple y evolutivo" en chino. También se puede considerar como un acrónimo de ¡Yes It Is!

Yii es un framework de programación web genérico, lo que significa que puede utilizarse para desarrollar todo tipo de aplicaciones web utilizando PHP. Debido a su arquitectura basada en componentes y su sofisticado soporte de caché, es especialmente adecuado para desarrollar aplicaciones a gran escala como portales, foros, sistemas de gestión de contenido (CMS), proyectos de comercio electrónico, servicios web RESTful, etc. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014)

1.3.1. Versiones de Yii

Yii actualmente tiene dos versiones principales disponibles: 1.1 y 2.0. La versión 1.1 es la generación anterior y ahora está en modo de mantenimiento. La versión 2.0 es una reescritura completa de Yii, que adopta las últimas tecnologías y protocolos, incluidos Composer, PSR, espacios de nombres, características, etc. La versión 2.0 representa la generación actual del framework y recibirá los principales esfuerzos de desarrollo en los próximos años. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014)

1.3.2. Requisitos

Yii 2.0 requiere PHP 5.4.0 o superior y funciona mejor con la última versión de PHP 7. El uso de Yii requiere conocimientos básicos de programación orientada a objetos (OOP), ya que Yii es un marco puro basado en OOP. Yii 2.0 también hace uso de las últimas características de PHP, como espacios de nombres y características. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014)

1.3.3. Características Yii

(Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014) nos dicen “si estás familiarizado con otros framework, puedes apreciar cómo se compara Yii con ellos.”

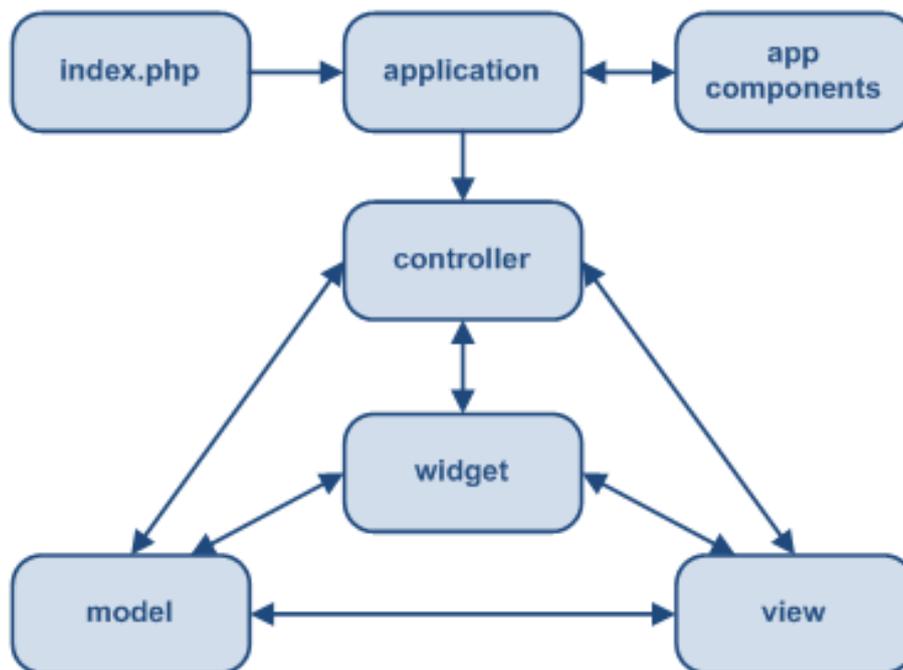
- Yii implementa el patrón de diseño MVC (Modelo-Vista-Controlador) y promueve la organización de código basada en este patrón.
- La filosofía de Yii consiste en escribir el código de manera simple y elegante, sin sobre diseñar nunca por el mero hecho de seguir un patrón de diseño determinado.
- Yii es un framework completo que provee muchas características probadas y listas para usar, como los constructores de consultas y la clase ActiveRecord para las bases de datos relacionales y NoSQL, la compatibilidad con la arquitectura REST para desarrollar API, la compatibilidad de caché en varios niveles y muchas más.

- Yii es extremadamente extensible. Puedes personalizar o reemplazar prácticamente cualquier pieza de código de base, como se puede también aprovechar su sólida arquitectura de extensiones para utilizar o desarrollar extensiones distribuibles.
- El alto rendimiento es siempre la meta principal de Yii.

1.3.4. Arquitectura de Yii

Las aplicaciones realizadas con Yii están organizadas de acuerdo con el patrón de diseño modelo-vista-controlador (MVC). Los modelos representan datos, la lógica de negocios y sus reglas; las vistas son la representación de salida de los modelos; y finalmente, los controladores que toman datos de entrada y los convierten en instrucciones para los modelos y vistas. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov, 2014)

Ilustración 4 Estructura estática de aplicación Yii.

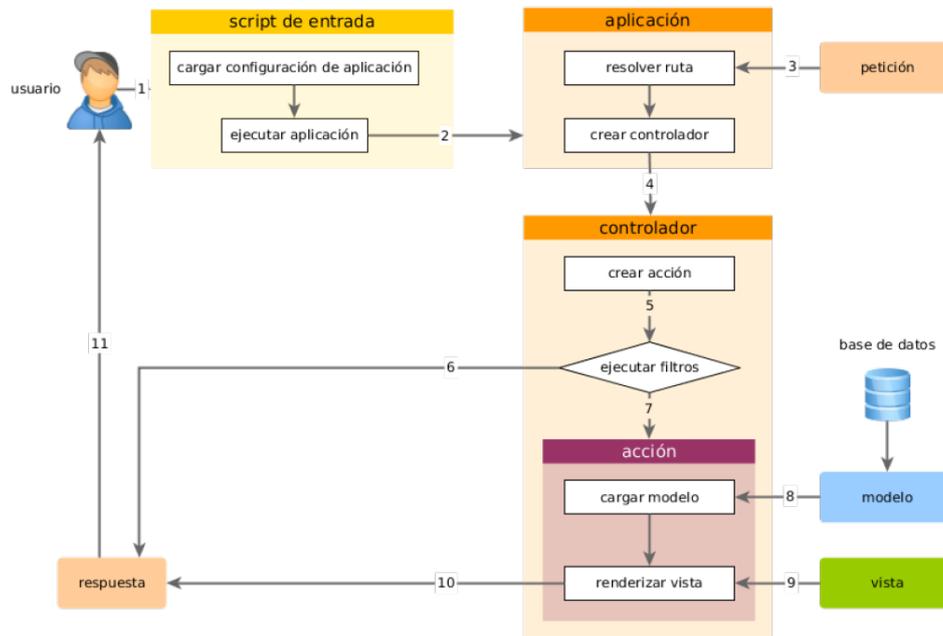


Fuente: www.yiiframework.com

1.3.5. Ciclo de Vida de una Petición (Request)

A continuación, se detalla el ciclo de vida de una petición en Yii como podemos ver en la ilustración 3 y una breve explicación de esta.

Ilustración 5: diagrama de ciclo de vida de una petición



Fuente: Propia.

Según (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014) detallan que el ciclo de petición se desarrolla de la siguiente manera.

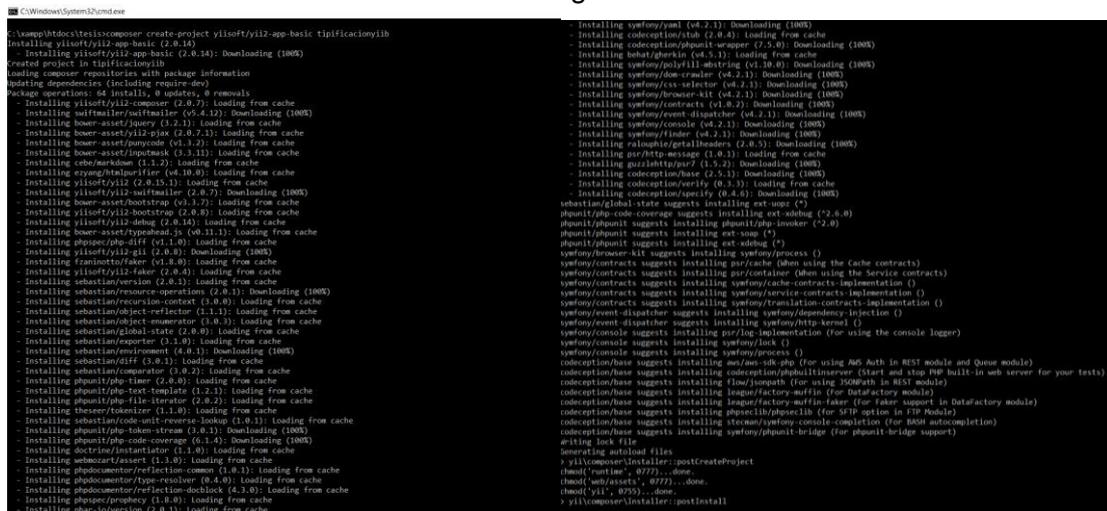
1. Un usuario realiza una petición al script de entrada web/index.php.
2. El script de entrada carga la configuración de la aplicación y crea una instancia de la aplicación para manejar la consulta.
3. La aplicación resuelve la ruta solicitada con la ayuda del componente request de la aplicación.
4. La aplicación crea una instancia de un controlador para manejar la petición.
5. El controlador crea una instancia de una acción y ejecuta los filtros de dicha acción.
6. Si alguno de los filtros falla, la acción es cancelada.
7. Si todos los filtros pasan, la acción es ejecutada.
8. La acción carga datos del modelo, posiblemente de la base de datos.
9. La acción renderiza una vista, pasándole los datos del modelo cargado.
10. El resultado de la renderización es pasado al componente response de la aplicación.
11. El componente response envía el resultado de la renderización al navegador del usuario.

1.3.6. Instalación

Yii 2.0 adopta íntegramente Composer, el administrador de paquetes de facto de PHP. Tanto la instalación del núcleo del framework como las extensiones se manejan a través de Composer. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov,; 2014)

Para el presente estudio se eligió instalar la plantilla avanzada de Yii 2 mediante el uso de Composer con la siguiente línea “*composer create-project --prefer-dist yiisoft/yii2-app-advanced yii-application*”. Reemplazamos “*yii-application*” por el nombre que queremos dar a nuestra aplicación.

Ilustración 6: Log Instalación Yii2



Fuente: Propia

1.3.7. Estructura de plantilla avanzada de Yii.

La estructura de la plantilla avanzada de Yii 2 una vez descargada se muestra en la ilustración 5.

Ilustración 7 Estructura de archivos de la plantilla avanzada

backend	17/12/2018 23:46	Carpeta de archivos	
common	17/12/2018 23:46	Carpeta de archivos	
console	17/12/2018 23:46	Carpeta de archivos	
environments	17/12/2018 23:46	Carpeta de archivos	
frontend	17/12/2018 23:46	Carpeta de archivos	
vagrant	17/12/2018 23:46	Carpeta de archivos	
vendor	17/12/2018 23:48	Carpeta de archivos	
.bowerrc	17/12/2018 23:45	Archivo BOWERRC	1 KB
.gitignore	17/12/2018 23:45	Documento de tex...	1 KB
codeception.yml	17/12/2018 23:46	Archivo YML	1 KB
composer.json	17/12/2018 23:46	Archivo JSON	2 KB
composer.lock	17/12/2018 23:48	Archivo LOCK	115 KB
init	17/12/2018 23:46	Archivo	9 KB
init.bat	17/12/2018 23:46	Archivo por lotes ...	1 KB
LICENSE.md	17/12/2018 23:45	Archivo MD	2 KB
README.md	17/12/2018 23:45	Archivo MD	3 KB
requirements.php	17/12/2018 23:46	Archivo PHP	5 KB
Vagrantfile	17/12/2018 23:45	Archivo	3 KB
yii.bat	17/12/2018 23:46	Archivo por lotes ...	1 KB

Fuente: Propia

1.3.8. Generador de código Gii

Gii es un generador de código que automáticamente implementa algunas de las características más comunes de una aplicación.

Características de Gii:

- Genera clases Active Record;
- Genera el código que implementa las operaciones ABM (CRUD) de una tabla de la base de datos.
- El código que genera Gii es editable.

Para poder utilizar el generador de código Gii es necesario configurarlo como modulo en nuestra aplicación y lo haremos en config/web.php de la siguiente manera:

Ilustración 8: Configuración de Gii

```
19
20     $config['bootstrap'][] = 'gii';
21     $config['modules']['gii'] = [
22         'class' => 'yii\gii\Module',
23     ];
24
25     $config['modules']['gii'] = [
26         'class' => 'yii\gii\Module',
27         'generators' => [
28             'crud' => [
29                 // 'class' => 'common\generators\crud\Generator',
30                 'class' => 'yii\gii\generators\crud\Generator',
31                 'templates' => ['adminlte_crud' => '@common/generators/
```

Fuente: Propia.

A continuación, se muestra un ejemplo de cómo el generador de código Gii trabaja

- Tabla: country
- Modelo: Country

Ilustración 9: Generador de código Gii

The screenshot shows the 'Model Generator' interface of the YII Code Generator. On the left is a sidebar menu with options: Model Generator (selected), CRUD Generator, Controller Generator, Form Generator, Module Generator, and Extension Generator. The main area is titled 'Model Generator' and contains the following fields and options:

- Table Name:** A text input field containing 'country'.
- Model Class:** A text input field containing 'Country|'.
- Namespace:** A text input field containing 'app\models'.
- Base Class:** A text input field containing 'yii\db\ActiveRecord'.
- Database Connection ID:** A text input field containing 'db'.
- Options:** A list of checkboxes: 'Use Table Prefix' (unchecked), 'Generate Relations' (checked), 'Generate Labels from DB Comments' (unchecked), and 'Enable I18N' (unchecked).
- Code Template:** A text area containing the default template path: 'default (/Users/qiang/Web/yii/basic2/vendor/yiisoft/yii2-gii/generators/model/default)'. A 'Preview' button is located below this field.

At the bottom of the page, there is a footer with the text 'A Product of YII Software LLC' on the left and 'Powered by YII Framework' on the right.

Fuente: (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014).

Al hacer clic en el botón "Preview". Verás que `models/Country.php` está mostrado listado como la clase resultante que ha de ser creada. Puedes hacer clic en el nombre de la clase para previsualizar su contenido. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014)

(Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014) también nos dicen que al utilizar Gii, si habías creado previamente el mismo archivo y puede ser sobrescrito, si haces clic en el botón diff cercano al nombre del archivo, verás las diferencias entre el código a ser generado y la versión existente del mismo.

Ilustración 10: Generador de Código Gii / Generar Modelo

The screenshot displays the 'Model Generator' interface of the Yii Code Generator. On the left, a sidebar lists various generators: Model Generator, CRUD Generator, Controller Generator, Form Generator, Module Generator, and Extension Generator. The main content area is titled 'Model Generator' and includes a description: 'This generator generates an ActiveRecord class for the specified database table.' The form contains the following fields and options:

- Table Name:** country
- Model Class:** Country
- Namespace:** app\models
- Base Class:** yii\db\ActiveRecord
- Database Connection ID:** db
- Use Table Prefix
- Generate Relations
- Generate Labels from DB Comments
- Enable I18N
- Code Template:** default (/Users/qiang/Web/yii/basic2/vendor/yiisoft/yii2-gii/generators/model/default)

At the bottom of the form are 'Preview' and 'Generate' buttons. Below these buttons, a message states: 'Click on the above Generate button to generate the files selected below:'. A table below this message shows the generated file 'models/Country.php' with an 'overwrite' checkbox checked.

Fuente: (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014).

Para sobrescribir un archivo existente, marca el checkbox que se encuentra al lado de “overwrite” y posteriormente haz clic en el botón “Generate”. Después, verás una página de confirmación indicando que el código ha sido generado correctamente y tu archivo `models/Country.php` ha sido sobrescrito con el nuevo código generado. (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014)

1.3.9. APIs REST

Según (Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;, 2014) nos dicen que Yii ofrece todo un conjunto de herramientas para simplificar la tarea de implementar un servicio web APIs RESTful. En particular, Yii soporta las siguientes características sobre APIs RESTful;

- Prototipado rápido con soporte para APIs comunes para Active Record;
- Formato de respuesta de negocio (soporta JSON y XML por defecto);
- Personalización de objetos serializados con soporte para campos de salida seleccionables;
- Formateo apropiado de colecciones de datos y validación de errores;
- Soporte para HATEOAS;
- Eficiente enrutamiento con una adecuada comprobación del verbo(verb) HTTP;

- Incorporado soporte para las OPTIONS y HEAD;
- Autenticación y autorización;
- Cacheo de datos y cacheo HTTP;
- Limitación de rango;

1.4. React.js

React es una librería Javascript focalizada en el desarrollo de interfaces de usuario. Esa es su principal área de trabajo, pero lo cierto es que, con todo el ecosistema de aplicaciones y herramientas y componentes, con React encontramos un excelente aliado para hacer todo tipo de aplicaciones web, SPA (Single Page Application) o incluso aplicaciones para móviles. (Alvares. Miguel, 2016).

De igual manera (Jimenez, 2015) nos dice que React, también conocido como React.js o ReactJS, es una librería JavaScript OpenSource diseñada para crear interfaces de usuario. Ha sido concebida para facilitar el desarrollo de SPA, Single Page Applications, obteniendo un gran rendimiento y ofreciendo una forma de desarrollo más cercana a la creación de videojuegos que a la de aplicaciones. Esta librería está mantenida por Facebook, Instagram y una gran comunidad de desarrolladores independientes y corporaciones.

1.4.1. Historia

(Alvares. Miguel, 2016) React es una librería que proviene de Facebook. Es software libre y a partir de su liberación una creciente comunidad de desarrolladores la está usando. Se crea en base a unas necesidades, generadas por el propio desarrollo de la web de la popular red social. En Facebook necesitaban herramientas para un desarrollo rápido pero focalizadas en un mayor rendimiento que otras alternativas existentes en el mercado.

Detectaron que el típico marco de binding y doble binding ralentizaba un poco su aplicación, debido a la cantidad de conexiones entre las vistas y los datos. Como respuesta crearon una nueva dinámica de funcionamiento, en la que optimizaron la forma como las vistas se renderizaban frente al cambio en los datos de la aplicación.

A partir de ahí la probaron en su red social con resultados positivos y luego en Instagram, también propiedad de Facebook, con éxito. Más adelante, alentados por los positivos resultados en el rendimiento de React, muchas otras aplicaciones web de primer nivel la fueron adoptando. BBC, Airbnb, Netflix, Dropbox y un largo etc.

(Jimenez, 2015) aclara que React intenta ayudar a los desarrolladores a crear aplicaciones web complejas que utilizan un gran intercambio de datos. Su principio es

sencillo declarative and composable. React solo se preocupará de la interfaz de usuario de tu app; si piensas en el paradigma de software Modelo-Vista-Controlador (MVC) React es únicamente la V. Esto hace que puedas utilizarlo combinándolo con otras librerías JavaScript o con frameworks como AngularJS, Ember o Backbone.

1.4.2. Objetivo

(Alvares. Miguel, 2016) Sirve para desarrollar aplicaciones web de una manera más ordenada y con menos código que si usas Javascript puro o librerías como jQuery centradas en la manipulación del DOM. Permite que las vistas se asocien con los datos, de modo que, si cambian los datos, también cambian las vistas.

1.4.3. Propiedades

(Jimenez, 2015) Las propiedades son parámetros inmutables que se reciben siempre desde un componente Padre. El propio componente nunca podrá modificar sus propiedades, y la única manera de modificarlas será desde el padre cuando este ejecute una nueva renderización del componente Hijo el cual recibirá las nuevas propiedades. Podemos decir entonces que las propiedades siempre se establecen desde componentes superiores, Padres, a componentes inferiores, Hijos.

1.4.3.1. getDefaultProps

(Jimenez, 2015) La función getDefaultProps se llamará para definir valores por defecto en las propiedades de tus componentes. Éstas serán sobrescritas cuando vengan establecidas desde el componente Padre. Esto es realmente útil cuando en nuestro componente tenemos propiedades opcionales las cuales queremos que tengan un valor por defecto.

1.4.3.2. propTypes

(Jimenez, 2015) El atributo propTypes dentro de tus componentes establece un validador de tipo que generará warnings en tu consola de desarrollo. Es muy recomendable usar siempre la definición de propTypes en todos tus componentes, a pesar de que solo obtendrás un beneficio cuando estés desarrollando tu aplicación es una buena manera de generar consistencia en tus componentes. También te servirá como una vía de documentación implícita, puesto que quien colabore contigo sabrá muy bien qué propiedades espera tu componente y de qué tipo. Cuando se proporciona un valor no válido, una advertencia se mostrará en la consola JavaScript. Ten en cuenta que por razones de rendimiento propTypes solamente se comprueba en el modo de desarrollo.

1.4.3.3. Refs

(Jimenez, 2015) React soporta una propiedad especial que puedes añadir a cualquiera de tus componentes. Esta propiedad especial te permite crear una referencia, a modo de enlace simbólico, a la instancia generada una vez el componente ha sido renderizado en el DOM. Siempre garantizando que la referencia contenga la instancia apropiada, en cualquier punto del ciclo de vida del componente.

1.4.4. Atributos Soportados

(Jimenez, 2015) React soporta todos los atributos data-* y aria-* así como todos los atributos de la siguiente lista:

accept acceptCharset accessKey action allowFullScreen allowTransparency alt async autoComplete autoFocus autoPlay cellPadding cellSpacing charSet checked classID className colSpan cols content contentEditable contextMenu controls coords crossOrigin data dateTime defer dir disabled download draggable encType form formAction formEncType formMethod formNoValidate formTarget frameBorder headers height hidden high href hrefLang htmlFor httpEquiv icon id label lang list loop low manifest marginHeight marginWidth max maxLength media mediaGroup method min multiple muted name noValidate open optimum pattern placeholder poster preload radioGroup readOnly rel required role rowSpan rows sandbox scope scoped scrolling seamless selected shape size sizes span spellCheck src srcDoc srcSet start step style tabIndex target title type useMap value width wmode.

Hay que tener únicamente cuidado con el atributo class (utilizado para establecer clases CSS), ya que es una palabra reservada dentro de JavaScript, por tanto, tienes que utilizar la propiedad "className".

1.4.5. Ciclo de vida

Según (Jimenez, 2015) Todos los componentes de React tienen un ciclo de vida realmente complicado, la propia naturaleza de React hace que este sea de esta manera. Para facilitarnos la vida React nos provee de una serie de eventos para capturar y actuar en determinados puntos de este ciclo de vida. Realmente son solo 7 puntos de captura, los cuales podemos dividir por puntos de construcción, actualización y destrucción.

A continuación, podrás enlistar cuales son y para utilizarlos solo es necesario incluirlos dentro de la definición de tu componente.

1.4.5.1. Construcción: `componentWillMount`

componentWillMount: function() {}

Este método se ejecutará solo una vez, tanto si estás utilizando React en el Cliente como en el Servidor, y lo hará una vez el proceso de renderización haya concluido.

1.4.5.2. Construcción: `componentDidMount`

componentDidMount: function() {}

Al igual que `componentWillMount` se ejecuta una sola vez, pero esta vez solo en aplicaciones Cliente

1.4.5.3. Actualización: `componentWillReceiveProps`

componentWillReceiveProps: function(next_props) {}

Es un punto de captura de actualización y como su nombre indica este método se ejecutará cuando nuestro componente reciba nuevas propiedades

1.4.5.4. Actualización: `componentWillUpdate`

componentWillUpdate: function(next_props, next_state) {}

Este punto de captura de actualización se ejecutará inmediatamente antes de renderizar las nuevas propiedades o atributos de estado.

1.4.5.5. Actualización: `shouldComponentUpdate`

shouldComponentUpdate: function(next_props, next_state) { return true; }

Este punto de captura de actualización, al igual que el anterior, se ejecuta antes de renderizar las nuevas propiedades o estado, y nunca se ejecutará con la renderización inicial.

A diferencia del resto de puntos de captura de actualización puedes utilizar este método para bloquear la renderización retornando un Boolean con `false`.

1.4.5.6. Actualización: `componentDidUpdate`

componentDidUpdate: function(prev_props, prev_state) {}

Este punto de captura de actualización se ejecuta inmediatamente después de que el componente haya sido renderizado.

1.4.5.7. Destrucción: `componentWillUnmount`

componentWillUnmount: function() {}

Es el único punto de captura de destrucción que tenemos dentro del ciclo de vida de nuestro componente, y se ejecutará inmediatamente antes de que el componente se desmonte del DOM.

1.4. Servidor de aplicaciones XAMPP

Un programa que permite instalar Apache, PHP y MySQL en un solo paso y de manera sencilla en sistemas Linux, Windows, Mac OS X y Solaris y permite contar con estos programas correctamente configurados, sin tener que preocuparnos en los pormenores de cada uno.

Lo que realmente hace especial a XAMPP es que tiene distribuciones para distintos sistemas operativos: Linux, Windows, Mac OS X y Solaris. Así que podemos utilizarlo casi en cualquier sistema operativo que podamos tener. (Alvarez, 2007)

(Apache, 2018) XAMPP incluye:

- Apache 2.4.29
- MariaDB 10.1.30
- PHP 7.0.27
- phpMyAdmin 4.7.4
- OpenSSL 1.0.2
- XAMPP Control Panel 3.2.2
- Webalizer 2.23-04
- Mercury Mail Transport System 4.63
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.56
- Strawberry Perl 7.0.56 Portable

1.5. Base de Datos MySQL

MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio. (Oracle, 2014)

Según (Oracle, 2014).

- **MySQL es un sistema de gestión de bases de datos.**

Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas

cantidades de información en una red corporativa. Para añadir, acceder, y procesar los datos almacenados en una base de datos, necesita un sistema de gestión de base de datos como MySQL Server

- **MySQL es un sistema de gestión de bases de datos relacionales.**

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL.

- **MySQL software es Open Source.**

Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License).

- **El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.**

El servidor MySQL también tiene una serie de características prácticas desarrolladas en cooperación con los usuarios. MySQL Server se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. MySQL Server ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet

- **MySQL Server trabaja en entornos cliente/servidor o incrustados.**

El software de bases de datos MySQL es un sistema cliente/servidor que consiste en un servidor SQL multi-threaded que trabaja con diferentes bakends, programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones (APIs).

- **Una gran cantidad de software de contribuciones está disponible para MySQL.**

Es muy posible que su aplicación o lenguaje favorito soporte el servidor de base de datos MySQL.

1.5.5. Historia de MySQL

MySQL empezó con la intención de usar mSQL para conectar a nuestras tablas utilizando nuestras propias rutinas rápidas de bajo nivel (ISAM). Sin embargo y tras algunas pruebas, llegamos a la conclusión que mSQL no era lo suficientemente rápido

o flexible para nuestras necesidades. Esto provocó la creación de una nueva interfaz SQL para nuestra base de datos, pero casi con la misma interfaz API que mSQL. Esta API fue diseñada para permitir código de terceras partes que fue escrito para poder usarse con mSQL para ser fácilmente portado para el uso con MySQL.

1.5.6. Características de MySQL

La siguiente lista describe algunas de las características más importantes del software de base de datos MySQL (Oracle, 2014).

- Interioridades y portabilidad
- Tipos de columnas
- Sentencias y funciones
- Seguridad
- Escalabilidad y límites
- Conectividad
- Localización
- Clientes y herramientas

1.5.7. Estabilidad de MySQL

El código original se remonta a los principios de los años 80. En TcX, la predecesora de MySQL AB, el código MySQL ha funcionado en proyectos desde mediados de 1996 sin ningún problema. Cuando el software de base de datos MySQL fue distribuido entre un público más amplio, nuestros nuevos usuarios rápidamente encontraron trozos de código no probados. Cada nueva versión desde entonces ha tenido pocos problemas de portabilidad incluso considerando que cada nueva versión ha tenido muchas nuevas funcionalidades.

Cada versión de MySQL Server ha sido usable. Los problemas han ocurrido únicamente cuando los usuarios han probado código de las "zonas grises". Naturalmente, los nuevos usuarios no conocen cuáles son estas zonas; esta sección, por lo tanto, trata de documentar dichas áreas conocidas hoy en día. (Oracle, 2014)

1.6. Composer

Composer es una herramienta para la administración de dependencias en PHP. Le permite declarar las bibliotecas de las que depende su proyecto y las administrará (las instalará / actualizará) (Composer, 2013).

1.6.5. Gestión de dependencia

Según (Composer, 2013) Composer no es un administrador de paquetes en el mismo sentido que YUM o APT. Sí, se trata de "paquetes" o bibliotecas, pero los

gestiona por proyecto, instalándolos en un directorio (por ejemplo, vendor) dentro de su proyecto. Por defecto no instala nada globalmente. Por lo tanto, es un administrador de dependencia. Sin embargo, admite un proyecto "global" por conveniencia a través del comando global.

Esta idea no es nueva y Composer está fuertemente inspirada en node's npm y ruby's bundler.

1.6.6. Requisitos del sistema

Composer requiere PHP 5.3.2 o superior para ejecutarse. También se requieren algunas configuraciones sensibles de PHP y banderas de compilación, pero al usar el instalador se advertirá acerca de cualquier incompatibilidad.

Para instalar paquetes desde fuentes en lugar de simples archivos comprimidos, necesitará GIT, SVN, FOSSIL o HG dependiendo de cómo se controle la versión del paquete.

Composer es multiplataforma y se esfuerza para que funcione igualmente bien en Windows, Linux y OSX. (Composer, 2013).

1.7. Editor de texto

El editor de texto es el mejor compañero de los desarrolladores; es su herramienta fundamental para poder escribir las líneas de código que luego darán vida a una aplicación o a una página web. Un buen editor de texto puede ayudar mucho a los desarrolladores y, por ejemplo, nos puede ahorrar tiempo en la fase de depuración si el editor nos ofrece ayuda con la sintaxis del lenguaje de programación con el que estemos trabajando. (Velasco, 2014)

1.7.6. Atom

Atom no es un editor de texto complejo, Github buscaba desarrollar una herramienta que pudiese ser utilizada, por ejemplo, por alguien que estuviese dando sus primeros pasos en el mundo de la programación. Para hacer simples las cosas, Github se ha apoyado en el mundo de la web y Atom se ha construido usando los mismos componentes que se usarían para desarrollar una web: HTML y CSS. Además, como buena herramienta de trabajo, Atom se ha concebido como un editor flexible al que se le pueden añadir nuevas funcionalidades y complementos, además de temas, que nos permitan configurar un espacio de trabajo adaptado a nuestras necesidades. (Velasco, 2014)

1.8. Lenguajes de programación

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis.

1.8.6. HTML

Según (Mateu, 2004) el lenguaje HTML (hypertext markup language) se utiliza para crear documentos que muestren una estructura de hipertexto. Un documento de hipertexto es aquel que contiene información cruzada con otros documentos, lo cual nos permite pasar de un documento al referenciado desde la misma aplicación con la que lo estamos visualizando. HTML permite, además, crear documentos de tipo multimedia, es decir, que contengan información más allá de la simplemente textual, por ejemplo:

- Imágenes
- Video
- Sonido
- Subprogramas activos (plugin, applets)

El lenguaje HTML no es el único lenguaje existente para crear documentos hipertexto. Hay otros lenguajes anteriores o posteriores a HTML (SGML, XML, etc.), si bien HTML se ha convertido en el lenguaje estándar para la creación de contenido para Internet.

1.8.6.1. Estructura básica

Todos los documentos HTML siguen aproximadamente la misma estructura. Todo el documento debe ir contenido en una etiqueta HTML, dividiéndose en dos partes: la cabecera, contenida en una etiqueta HEAD y el cuerpo del documento (donde está la información del documento), que está envuelto por una etiqueta BODY. La cabecera contiene algunas definiciones sobre el documento: su título, marcas extra de formato, palabras clave, etc.

Un primer ejemplo sería:

Ilustración 11 Estructura básica HTML

```
<HTML>
  <HEAD>
    <TITLE>Título del documento</TITLE>
  </HEAD>
  <BODY>
    Texto del documento
  </BODY>
</HTML>
```

Fuente: (Mateu, 2004).

1.8.7. Javascript

Javascript es un lenguaje de programación interpretado (un lenguaje de tipo script). A pesar de que existen intérpretes no dependientes de ningún navegador, es un lenguaje de script que suele encontrarse vinculado a páginas web. Javascript y Java son dos lenguajes de programación distintos con filosofías muy diferentes. El único punto en común es la sintaxis, ya que cuando Netscape diseñó Javascript, se inspiró en la sintaxis de Java. (Mateu, 2004)

1.8.8. PHP

PHP, cuyas siglas responden a un acrónimo recursivo (PHP: hypertext preprocessor), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como Perl, C y C++. Es rápido, interpretado, orientado a objetos y multiplataforma. Para él se encuentra disponible una multitud de librerías. PHP es un lenguaje ideal tanto para aprender a desarrollar aplicaciones web como para desarrollar aplicaciones web complejas. PHP añade a todo eso la ventaja de que el intérprete de PHP, los diversos módulos y gran cantidad de librerías desarrolladas para PHP son de código libre, con lo que el programador de PHP dispone de un impresionante arsenal de herramientas libres para desarrollar aplicaciones. (Mateu, 2004)

1.8.8.1. Como funciona PHP y sintaxis

Si solicitamos a nuestro servidor una página PHP, éste envía dicha página al intérprete de PHP que la ejecuta (de hecho, no se trata más que de un programa) y devuelve el resultado (generalmente HTML) al servidor web, el cual, a su vez, se lo enviará al cliente. (Mateu, 2004)

Para empezar a comprender la sintaxis del lenguaje, analizaremos un programa mínimo de PHP:

Ilustración 12 Ejemplo PHP

```
<?php
    $MYVAR = "1234";
    $myvar = "4321";
    echo $MYVAR. "<br>\n";
    echo $myvar. "<br>\n";
?>
```

Fuente: (Mateu, 2004).

La ejecución de este programa (su visualización desde un navegador), dará como resultado:

1234

4321

El primer punto que debemos destacar es que los bloques de código de PHP están delimitados en HTML con "<?php" y "?>". Podemos, por tanto, escribir una página HTML e incluir en ella diversos bloques de instrucciones PHP. (Mateu, 2004)

1.9. Metodologías Ágiles

El término Ágil, aplicado a software nace en 2001, en una reunión celebrada en Utah, donde un grupo de 17 expertos de la industria de software tenían como objetivo, ofrecer una alternativa a los procesos de desarrollo de software tradicionales, que se caracterizaban por ser rígidos y dirigidos por I documentación. Tras esta reunión se creó la Alianza Agile. (Rodríguez, 2015).

Las metodologías ágiles según (Rodríguez, 2015) son técnicas que permiten incorporar cambios con rapidez en el desarrollo de software. En conclusión, las metodologías ágiles son una serie de técnicas empleadas para la gestión de proyectos y estas responden al cumplimiento del manifiesto ágil:

1. Individuos e iteraciones por encima de procesos y herramientas: Las personas son el principal factor de éxito, es más importante construir un buen equipo que un buen entorno.
2. Software funcionando contra documentación extensiva: Solo producir documentos cuando sea necesario, estos deben ser cortos y centrados en lo fundamental.
3. Colaboración con el cliente por encima del contrato de negociación: interactuar constantemente con el cliente y el equipo de desarrollo.
4. Respuesta ante el cambio sobre seguir un plan determinado: la planificación debe ser flexible y abierta, esto facilita la habilidad de responder ante posibles cambios.

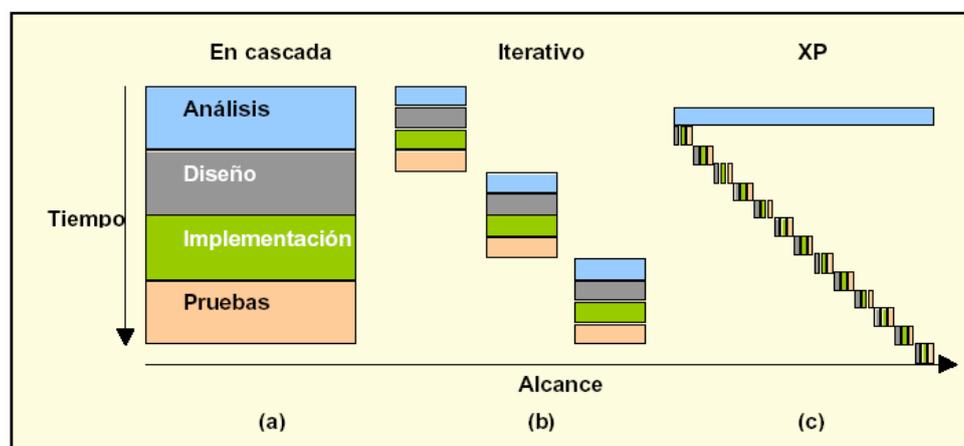
1.9.6. Metodología XP

(Joskowicz. José, 2008) Nos dice que la metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. Además, se especifica que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores. El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres.

XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto.

Típicamente un proyecto con XP lleva 10 a 15 ciclos o iteraciones. La siguiente figura esquematiza los ciclos de desarrollo en cascada e iterativos tradicionales (por ejemplo, incremental o espiral), comparados con el de XP.

Ilustración 13: Comparativa ciclos de desarrollo entre metodologías



Fuente: (Joskowicz. José, 2008)

1.9.6.1. Fases

Un proyecto XP puede tornarse muy dinámico es por esto por lo que se puede separar en fases las cuales son:

- **Fase de exploración**

En esta fase se decidirán las generalidades del proyecto y para esto con el cliente se desarrollarán sencillas “historias de usuarios”, y ya con estos datos el tiempo de desarrollo será determinado por los programadores.

- **Fase de planificación**

Normalmente esta es la fase más corta, en la que entre los clientes y desarrolladores determinan el orden en que serán implantadas las historias de usuario y obviamente cuando serán entregadas.

- **Fase de iteraciones**

Esta es una fase muy importante ya que en esta fase se desarrollarán las historias de usuario y al final de cada iteración se entregará un producto ya funcional.

- **Fase de puesta en producción**

Si en la anterior fase se entregan productos funcionales en esta fase es donde se pone a producción el software final tomando en cuenta que se podrían presentar tareas de ajuste.

1.9.6.2. Reglas y Practicas

(Joskowicz. José, 2008) La metodología XP tiene un conjunto importante de reglas y prácticas. En forma genérica, se pueden agrupar en:

- **Reglas y prácticas para la Planificación**

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes.

Los conceptos básicos de esta planificación son los siguientes:

Historias de usuarios: Las “Historias de usuarios” (“User stories”) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar.

Plan de entregas (“Release Plan”): El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de estas.

Plan de iteraciones (“Iteration Plan”): Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo con el orden preestablecido.

Reuniones diarias de seguimiento (“Stand-up mitin”): El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones.

- **Reglas y prácticas para el Diseño**

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

Simplicidad: Un diseño simple se implementa más rápidamente que uno complejo.

Soluciones “spike”: Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba, para explorar diferentes soluciones.

Recodificación: La recodificación (“refactoring”) consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible.

Metáforas: Una “metáfora” es algo que todos entienden, sin necesidad de mayores explicaciones. La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura de este.

- **Reglas y prácticas para el Desarrollo**

Disponibilidad del cliente: Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP.

Uso de estándares: Si bien esto no es una idea nueva, XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo, y que facilite la recodificación.

Programación dirigida por las pruebas (“Test-driven programming”): La metodología XP propone un modelo, en el que, lo primero que se escribe son las

pruebas que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas.

Programación en pares: XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas.

Integraciones permanentes: Todos los desarrolladores necesitan trabajar siempre con la “última versión”. Pero XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores.

Propiedad colectiva del código: En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, cualquier pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o recodificar.

Ritmo sostenido: El concepto que se desea establecer con esta práctica es el de planificar el trabajo de manera de mantener un ritmo constante y razonable, sin sobrecargar al equipo.

- **Reglas y prácticas para las Pruebas**

Pruebas unitarias: Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados.

Detección y corrección de errores: Cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir.

Pruebas de aceptación: Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada.

1.9.6.3. Valores en XP

XP se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito

- **Comunicación**

La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación.

- **Simplicidad**

XP, como metodología ágil, apuesta a la sencillez, en su máxima expresión. Sencillez en el diseño, en el código, en los procesos, etc.

- **Retroalimentación**

El cliente debe brindar retroalimentación de las funciones desarrolladas, de manera de poder tomar sus comentarios para la próxima iteración, y para comprender, cada vez más, sus necesidades.

- **Coraje**

Cuando se encuentran problemas serios en el diseño, o en cualquier otro aspecto, se debe tener el coraje suficiente como para encarar su solución, sin importar que tan difícil sea. Si es necesario cambiar completamente parte del código, hay que hacerlo, sin importar cuanto tiempo se ha invertido previamente en el mismo.

CAPÍTULO II

2. Desarrollo

En este capítulo se documentará el desarrollo del aplicativo de acuerdo a la metodología elegida que es eXtreme Programing (XP)

2.7. Definición de roles y responsabilidades

Para lograr un buen desarrollo del proyecto es necesario organizar a las personas involucradas con la asignación de roles y responsabilidades.

Tabla 3 Definición de Roles

Rol	Descripción	Responsabilidades
Programador	Se encargará de: <ul style="list-style-type: none">• Escribir el código del prototipo• Diseñar el prototipo• Probar las funciones del sistema.	<ul style="list-style-type: none">• Define las tareas y asigna un tiempo para cada una.• Realiza historias de usuario.• Planifica y demuestra el funcionamiento del prototipo.• Capacita al usuario para la utilización del prototipo.
Usuario	Persona que conoce el proceso de entrega del carnet de tipificación sanguínea.	<ul style="list-style-type: none">• Solicita el desarrollo del prototipo.• Ayuda con información para la creación de las historias de usuario.• Define las funciones prioritarias para la implementación.• Revisa que se esté cumpliendo los requerimientos del prototipo
Encargado de pruebas	Persona encargada de mostrar al usuario las funciones del prototipo en todas las fases del desarrollo.	<ul style="list-style-type: none">• Realiza las pruebas al prototipo.

		<ul style="list-style-type: none"> • Informa al usuario los resultados de las pruebas del prototipo.
Tutor	Encargado de asesorar durante el desarrollo del prototipo.	Apoya en todas las fases del desarrollo del prototipo.

Fuente: Propia

2.8. Definición de los integrantes del equipo de trabajo

La creación de equipos de trabajo es una parte esencial para el desarrollo con Metodología XP; a continuación, se detalla el equipo de trabajo.

Tabla 4 Equipos de Trabajo

Nombre	Descripción	Rol
Ing. MSc, Cosme Ortega	Revisa y verifica periódicamente que se desarrolle el prototipo apegándose a la planificación establecida.	Director de trabajo de grado.
Lic. Claudio Medina	Encargado de facilitar los requerimientos del prototipo.	Usuario final del prototipo
Sr. Carlos Aguirre	Se encargará del desarrollo del prototipo y la entrega de este.	Programador / Tesista

Fuente: Propia

2.9. Historias de usuario y diseño

A continuación, se desarrollan las historias de usuario que se establecen para el desarrollo del prototipo.

2.9.6. Ingreso al Sistema y Usuarios

Tabla 5: Historia de Usuario 1

HISTORIA DE USUARIO	
Desarrollo de un sistema de tipificación para la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.	
Numero: 1	Usuario: Ing. Cosme Ortega
Nombre de la historia: Ingreso al sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja

Estimación en horas: 5	Iteraciones asignadas: 2
Programador responsable: Carlos Aguirre Benavides	
Descripción: Para el ingreso al sistema es necesario de contar con un usuario y contraseña, existiendo dos (2) roles, cada usuario tendrá asignadas funciones del sistema.	
Observaciones: Los roles son: Administrador y Técnico	
Fecha: 3/12/2018	
Firma:	

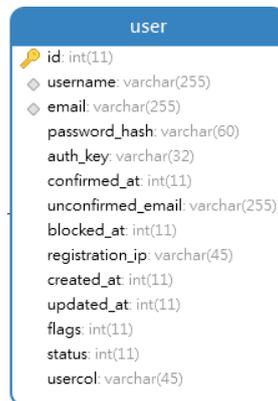
Fuente: Propia

Tabla 6: Tarea 1 - Historia de usuario 1

Tarea	
Numero de tarea: 1	Numero de historia: 1
Nombre de tarea: Creación del formulario Login.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 3/12/2018	Fecha de finalización: 3/12/2018
Programador responsable: Carlos Aguirre Benavides.	
Descripción: Diseño y desarrollo del formulario que permite el ingreso al sistema mediante el uso de usuario y contraseña.	

Fuente: Propia

Ilustración 14: Modelo de datos - Tabla Usuario



Fuente: Propia

Ilustración 15: Formulario de ingreso al sistema

Formulario de ingreso al sistema con los siguientes elementos:

- Campo de texto etiquetado "Usuario" con un placeholder "Usuario".
- Campo de texto etiquetado "Contraseña" con un placeholder "Password".
- Botón "INGRESAR" de color verde.
- Botón "CONSULTAR" de color azul.

Fuente: Propia

Tabla 7: Tarea 2 - Historia de Usuario 1

Tarea	
Numero de tarea: 2	Numero de historia: 1
Nombre de tarea: Administrador de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 4/12/2018	Fecha de finalización: 4/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo del formulario que permite administrar los usuarios del sistema este formulario permitirá:	
<ul style="list-style-type: none">• Acceso a crear usuarios.• Modificar contraseñas.• Dar de baja el usuario.• Asignar el rol	

Fuente: Propia

Ilustración 16: Formulario administración de usuarios

Administración de Usuario

Usuario	Rol	Estado	Acciones
Admin	Administrador	Activo	Cambio Clave Eliminar Rol

Fuente: Propia.

2.9.7. Pacientes

Tabla 8: Historia de usuario 2

HISTORIA DE USUARIO	
Desarrollo de un sistema de tipificación para la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.	
Numero: 2	Usuario: Ing. Cosme Ortega
Nombre de la historia: Gestión de pacientes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Estimación en horas: 15	Iteraciones asignadas: 5
Programador responsable: Carlos Aguirre Benavides	
Descripción:	
El sistema debe permitir:	
<ul style="list-style-type: none">• Listar pacientes.• Registrar pacientes con los siguientes datos:<ul style="list-style-type: none">✓ Cedula✓ Nombres✓ Apellidos✓ Teléfono✓ Email✓ Fecha de Nacimiento✓ Factor RH✓ Tipo Sanguíneo✓ Ciudad✓ Dirección	

- **Editar pacientes.**
- **Borrar pacientes.**
- **Editar pacientes.**
- **Exportar una lista de pacientes filtrada por identificación**

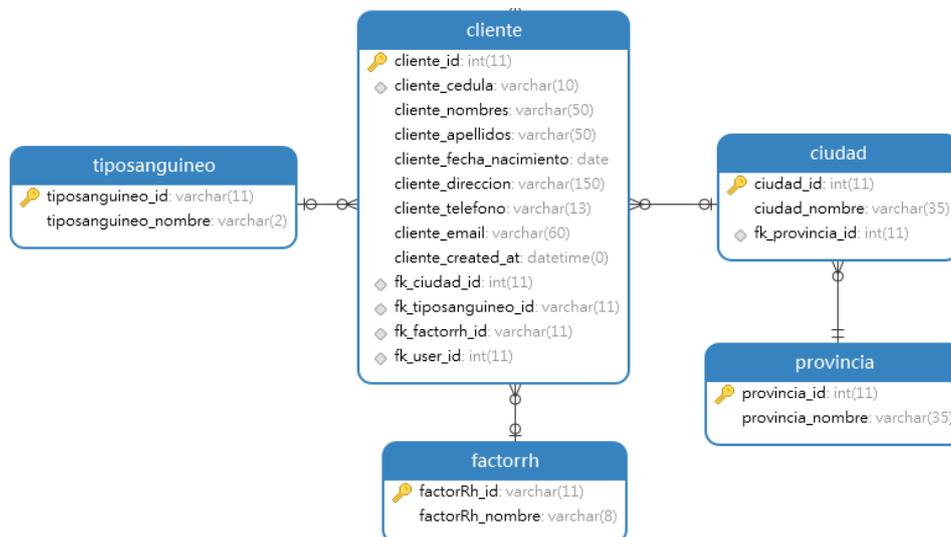
Observaciones: **Adicional se almacenará la provincia automáticamente, la fecha de creación y el usuario que registro el paciente.**

Fecha: 10/12/2018

Firma:

Fuente: Propia

Ilustración 17: Modelo de Datos - Tabla Cliente y relaciones.



Fuente: Propia

Tabla 9: Tarea 1 - Historia de Usuario 2

Tarea	
Numero de tarea: 1	Numero de historia: 2
Nombre de tarea: Listado de Clientes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10/12/2018	Fecha de finalización: 10/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo del formulario que permite el listado de los pacientes registrados en el sistema.	

Fuente: Propia.

Ilustración 18: Listado de pacientes

Cientes

Buscar por cédula

Nombre	Apellido	Cédula	Acciones
Carlos	Aguirre	1003770482	Editar Eliminar Imprimir

Fuente: Propia.

Tabla 10: Tarea 2 – Historia de Usuario 2

Tarea	
Numero de tarea: 2	Numero de historia: 2
Nombre de tarea: Registro de pacientes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 11/12/2018	Fecha de finalización: 11/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo del formulario que permite el registro de los pacientes registrados en el sistema.	

Fuente: Propia.

Ilustración 19: Registro de pacientes

Añadir Clientes

Cédula:	Nombre:
<input type="text"/>	<input type="text"/>
Apellidos:	Teléfono:
<input type="text"/>	<input type="text"/>
E-mail:	Fecha de nacimiento:
<input type="text"/>	<input type="text"/>
Factor RH:	Tipo sanguíneo:
<input type="text"/>	<input type="text"/>
Ciudad:	Dirección:
<input type="text"/>	<input type="text"/>

[GUARDAR](#) [CANCELAR](#)

Fuente: Propia.

Tabla 11: Tarea 3 - Historia de Usuario 2

Tarea	
Numero de tarea: 3	Numero de historia: 2
Nombre de tarea: Edición de pacientes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 12/12/2018	Fecha de finalización: 12/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo del formulario que permite la edición de los pacientes registrados en el sistema.	

Fuente: Propia

Ilustración 20: Edición de pacientes

Editar Clientes

Cédula:	Nombre:
<input type="text" value="1003770482"/>	<input type="text" value="Carlos David"/>
Apellidos:	Teléfono:
<input type="text" value="Aguirre Benavides"/>	<input type="text" value="0996716563"/>
E-mail:	Fecha de nacimiento:
<input type="text" value="tato.cre@gmail.com"/>	<input type="text" value="17/02/1992"/> 
Factor RH:	Tipo sanguíneo:
<input type="text" value="Positivo"/>	<input type="text" value="B"/>
Ciudad:	Dirección:
<input type="text" value="Ibarra"/>	<input type="text" value="Los Ceibos"/>

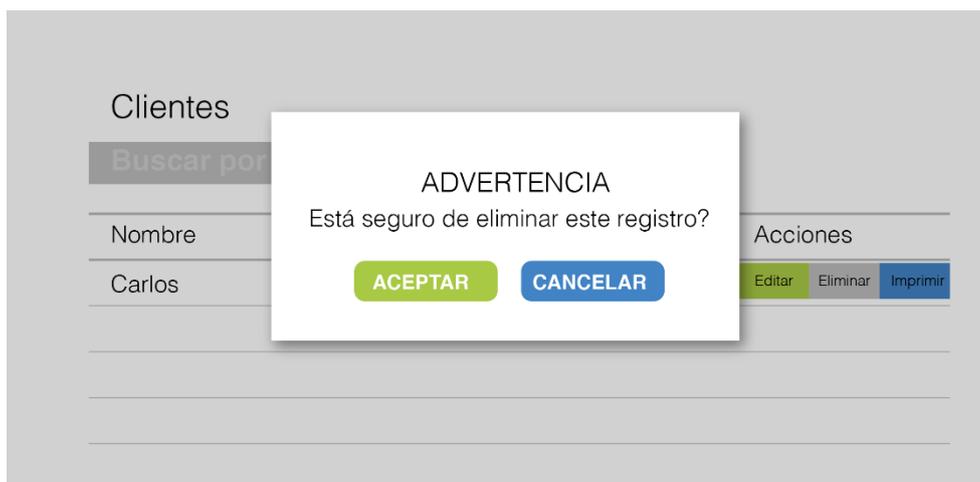
Fuente: Propia

Tabla 12: Tarea 4 - Historia de Usuario 2

Tarea	
Numero de tarea: 4	Numero de historia: 2
Nombre de tarea: Borrar pacientes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 13/12/2018	Fecha de finalización: 13/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo del formulario que permite borrar los pacientes registrados en el sistema.	

Fuente: Propia.

Ilustración 21: Borrar Pacientes



Fuente: Propia.

Tabla 13: Tarea 5 - Historia de Usuario 2

Tarea	
Numero de tarea: 5	Numero de historia: 2
Nombre de tarea: Imprimir Carnet.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 14/12/2018	Fecha de finalización: 14/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo del formulario que permite imprimir carnets de los pacientes registrados en el sistema.	

Fuente: Propia.

Ilustración 22: Impresión de Carnet



Carlos David Aguirre Benavides
19/02/2019
POSITIVO

Fuente: Propia

2.9.8. Búsqueda y Exportaciones

Tabla 14: Historia de Usuario 3.

HISTORIA DE USUARIO

Desarrollo de un sistema de tipificación para la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.

Numero: **3** **Usuario:** Ing. Cosme Ortega

Nombre de la historia: **Creación de formulario de búsquedas y exportaciones**

Prioridad en negocio: **Alta** **Riesgo en desarrollo:** Baja

Estimación en horas: **20** **Iteraciones asignadas:** 2

Programador responsable: **Carlos Aguirre Benavides**

Descripción:

El sistema debe permitir la búsqueda y filtración de datos de los clientes estos datos filtrados se podrán exportar.

Observaciones: **el formato a exportar es .CVS**

Fecha: 17/12/2018

Firma:

Fuente: Propia.

Tabla 15: Tarea 1 – Historia de Usuario 3

Tarea

Numero de tarea: **1** **Numero de historia:** 3

Nombre de tarea: **Creación del formulario de Búsquedas**

Tipo de tarea: **Desarrollo** **Puntos estimados:** 1

Fecha de inicio: **17/12/2018** **Fecha de finalización:** 17/12/2018

Programador responsable: **Carlos Aguirre Benavides**

Descripción: **Diseño y desarrollo del formulario que permite la filtración o búsqueda de datos de los pacientes registrados en el sistema.**

Fuente: Propia.

Ilustración 23: Formulario de Búsquedas.

BÚSQUEDAS

Nombre	Apellido	Cédula	Factor RH	Tipo sanguíneo:	Fecha de Creación:	Usuario	Ciudad	Provincia
<input type="text"/>	<input type="text"/>	<input type="text"/>	TODOS ▾	TODOS ▾	dd/mm/aaaa	<input type="text"/>	<input type="text"/>	<input type="text"/>
Carlos	Aguirre	1003770482	Positivo	B	19/02/2019	Admin	Ibarra	Imbabura

Exportar

Fuente: Propia.

Tabla 16: Tarea 2 - Historia de Usuario 3

Tarea	
Numero de tarea: 2	Numero de historia: 3
Nombre de tarea: Exportar datos filtrados.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 18/12/2018	Fecha de finalización: 18/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: Diseño y desarrollo de la exportación de los datos filtrados del sistema.	

Fuente: Propia.

Ilustración 24: Exportación de datos filtrados.

BÚSQUEDAS

Nombre	Apellido	Cédula	Factor RH	Tipo sanguíneo:	Fecha de Creación:	Usuario	Ciudad	Provincia
<input type="text"/>	<input type="text"/>	<input type="text"/>	TODOS ▾	TODOS ▾	dd/mm/aaaa	<input type="text"/>	<input type="text"/>	<input type="text"/>
Carlos	Aguirre	1003770482	Positivo	B	19/02/2019	Admin	Ibarra	Imbabura

Exportar

EXPORTAR

Atención ! Se exportarán los registros

EXPORTAR **CANCELAR**

Fuente: Propia.

2.9.9. Consulta pacientes registrados

Tabla 17: Historia de Usuario 4

HISTORIA DE USUARIO	
Desarrollo de un sistema de tipificación para la Cruz Roja Ecuatoriana Junta Provincial de Imbabura.	
Numero: 4	Usuario: Ing. Cosme Ortega
Nombre de la historia: Creación de formulario de consulta de pacientes registrados	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Estimación en horas: 5	Iteraciones asignadas: 1
Programador responsable: Carlos Aguirre Benavides	
Descripción: El sistema debe permitir la búsqueda de pacientes registrados en el sistema	
Observaciones: A este formulario se podrá acceder tanto registrado en sistema como no.	
Fecha: 27/12/2018	
Firma:	

Fuente: Propia.

Tabla 18: Tarea 1 - Historia de Usuario 4

Tarea	
Numero de tarea: 1	Numero de historia: 4
Nombre de tarea: Consulta de pacientes registrados.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 27/12/2018	Fecha de finalización: 27/12/2018
Programador responsable: Carlos Aguirre Benavides	
Descripción: El sistema debe permitir la búsqueda de pacientes registrados en el sistema.	

Fuente: Propia.

Ilustración 25: Consulta de pacientes registrados.

CONSULTA

Identificación

CONSULTAR LIMPIAR

Carlos David Aguirre Benavides
B POSITIVO

Fuente: Propia.

2.9.10. Diagramas de caso de uso.

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Ivar Jacobson e incorporada a UML. Modela la funcionalidad del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular. (García. Francisco, 2017 -2018).

2.9.10.1. Caso de uso administración del sistema.

A continuación, se muestra el caso de uso para el usuario administrador.

Ilustración 26: Caso de Uso / Usuario Administrador



Fuente: Propia.

Tabla 19: Descripción Caso de Uso / Usuario Administrador.

Caso de Uso	Usuario administrador
-------------	-----------------------

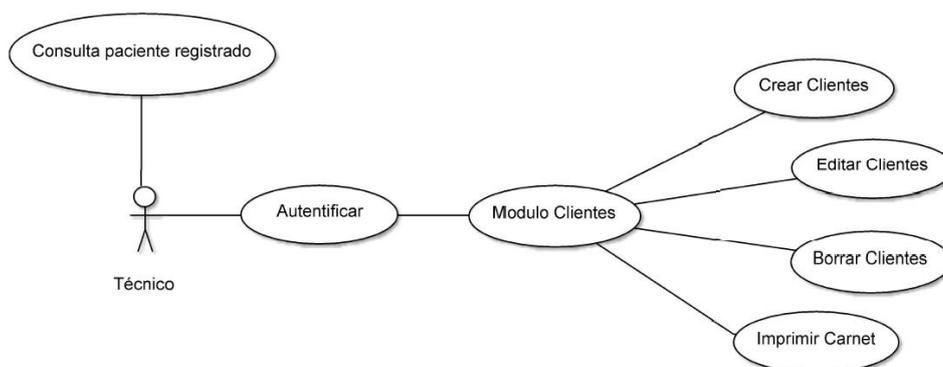
Descripción	El administrador del sistema tiene acceso a todas las funciones como: administrar usuarios, administrar pacientes, realizar búsquedas y exportar las mismas.
Actor	Administrador.
Condiciones previas	Tener acceso con usuario y contraseña y rol de administrador.
Flujo básico eventos	Registrar, cambiar contraseñas, inactivar y asignar usuarios. Registrar, editar, borrar e imprimir carnets. Realizar búsquedas, filtrar y exportar los datos filtrados.
Flujos alternativos	Mensajes de error: <ul style="list-style-type: none"> • Intenta ingresar al sistema con una contraseña incorrecta. • Intenta ingresar al sistema con usuario inactivo. • Intenta registrar un usuario repetido. • Intenta registrar un usuario con un correo repetido. • Intenta registrar un paciente con un número de identificación repetido. • Intenta registrar un paciente sin rellenar todos los datos del formulario. • Si se consulta un paciente que no esté registrado.

Fuente: Propia.

2.9.10.2. Caso de Uso Técnico

A continuación, se muestra el caso de uso para el usuario Técnico.

Ilustración 27: Caso de Uso / Usuario Técnico



Fuente: Propia.

Tabla 20: Descripción Caso de Uso / Usuario Técnico.

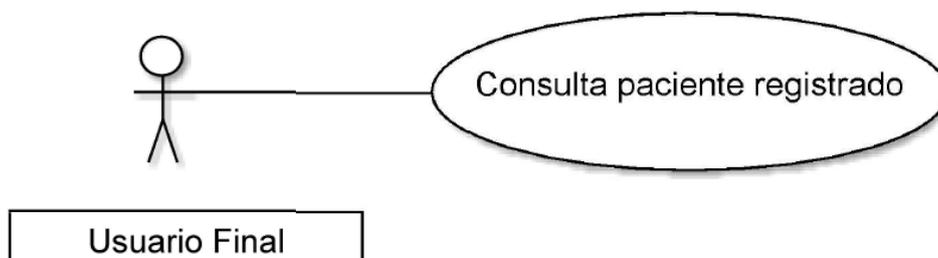
Caso de Uso	Usuario técnico.
Descripción	El usuario técnico del sistema tiene acceso para administrar pacientes.
Actor	Técnico.
Condiciones previas	Tener acceso con usuario y contraseña y rol de técnico.
Flujo básico eventos	Registrar, editar, borrar e imprimir carnets de los pacientes.
Flujos alternativos	<p>Mensajes de error:</p> <ul style="list-style-type: none"> • Intenta ingresar al sistema con una contraseña incorrecta. • Intenta ingresar al sistema con usuario inactivo. • Intenta registrar un paciente con un número de identificación repetido. • Intenta registrar un paciente sin rellenar todos los datos del formulario. • Si se consulta un paciente que no esté registrado.

Fuente: Propia.

2.9.10.3. Caso de Uso Usuario Final

A continuación, se muestra el caso de uso para el usuario final.

Ilustración 28: Caso de Uso / Usuario Final



Fuente: Propia.

Tabla 21: Descripción Caso de Uso / Usuario Final

Caso de Uso	Usuario final.
Descripción	El usuario final del sistema tiene acceso para consultar si un paciente está registrado o no en el sistema.
Actor	Final.
Condiciones previas	Tener URL de acceso al formulario de consulta.
Flujo básico eventos	Consultar con un número de identificación.
Flujos alternativos	Mensajes de error si se consulta un paciente que no esté registrado.

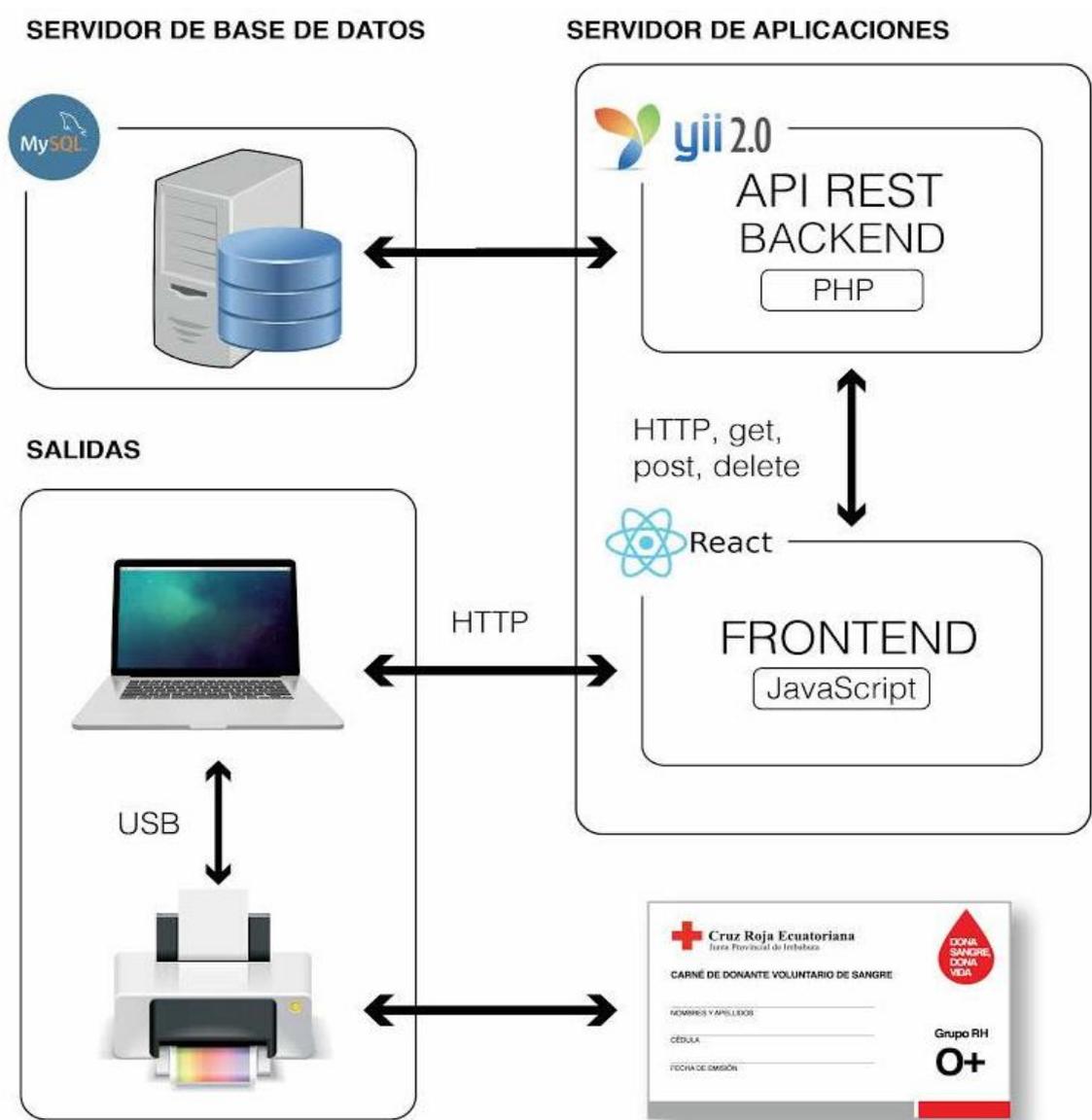
Fuente: Propia.

2.9.11. Arquitectura del Sistema

El prototipo se desarrollará con PHP y JavaScript, con el Framework Yii2 para el desarrollo de las APIs RESTfull el Backend y la tecnología React.JS para el Frontend. El servidor de aplicaciones que aloja el sistema será APACHE y los datos los almacenará en una base de datos MySQL para poder utilizar el sistema es necesario acceder por un navegador web.

Para el acceso al sistema se lo realizara desde cualquier dispositivo conectado al internet o red.

Ilustración 29: Diagrama de arquitectura del sistema.

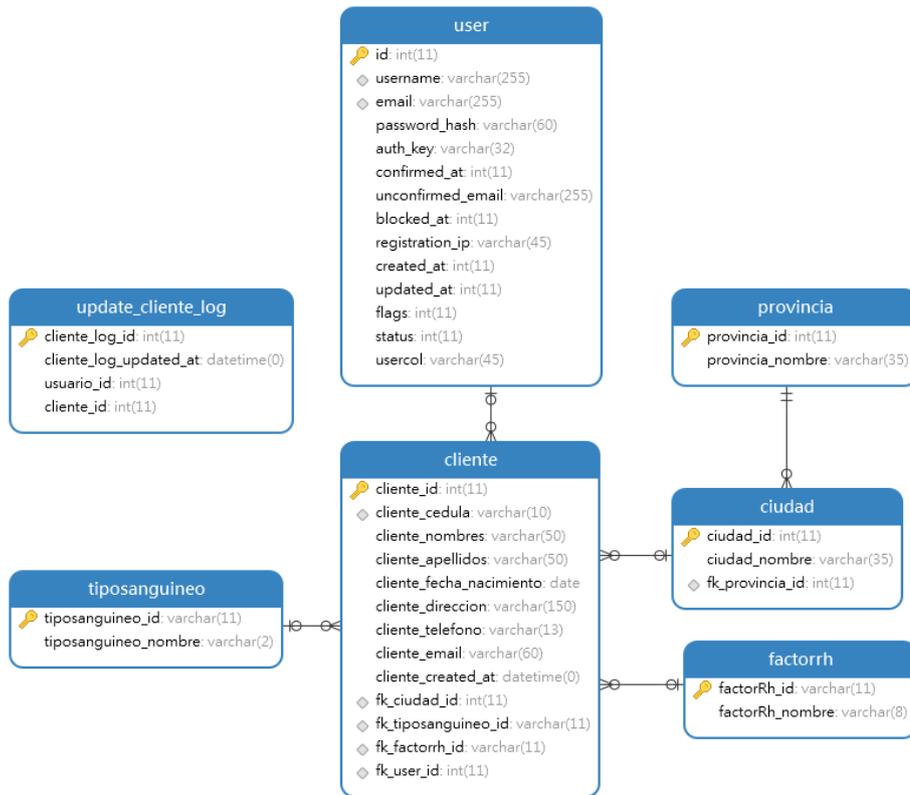


Fuente: Propia.

2.9.12. Diagrama entidad relación de la Base de Datos.

En la siguiente figura podemos apreciar el modelo entidad relación de la base de datos que se usará para el desarrollo del aplicativo

Ilustración 30: Modelo entidad relación BDD



Fuente: Propia.

Se debe tomar en cuenta que la tabla “update_cliente_log” guarda información acerca de las actualizaciones que se realizan a los clientes y servirá a futuro para implementar un reporte en el que se aprecie cuando un cliente fue modificado.

2.10. Desarrollo de las historias de usuario

2.10.1. Historia de Usuario 1: Ingreso al Sistema.

Tareas:

- Configurar la plantilla de Yii2 (plantilla avanzada) para el desarrollo de API’s RESTful.
- Creación de la ventana de Login.
- Crear el formulario para administrar los usuarios.

En esta tarea se realizará la configuración de la plantilla del Framework y se desarrollara los formularios para ingreso y administrar los usuarios, los mismos que contaran con validaciones pertinentes.

En la siguiente ilustración configuraremos el archivo bd.php para poder utilizar la base de datos MySQL de nuestro servidor de aplicaciones.

Ilustración 31: Configuración de Base de Datos en Yii

```
db.php
1 <?php
2
3 return [
4     'class' => 'yii\db\Connection',
5     'dsn' => 'mysql:host=localhost;dbname=tipificacion_db',
6     'username' => 'root',
7     'password' => '',
8     'charset' => 'utf8',
9
10    // Schema cache options (for production environment)
11    //'enableSchemaCache' => true,
12    //'schemaCacheDuration' => 60,
13    //'schemaCache' => 'cache',
14 ];
```

Fuente: Propia.

Para un correcto uso de las URL es necesario la creación de un host virtual para lo cual necesitamos editar archivos como Host de Windows y httpd-vhost.config del servidor de aplicaciones Apache como se muestran en las siguientes ilustraciones

Ilustración 32: Edición Archivo Host de Windows

```
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #      102.54.94.97      rhino.acme.com      # source server
17 #      38.25.63.10     x.acme.com         # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 # 127.0.0.1      localhost
21 # ::1           localhost
22 127.0.0.1      activate.navicat.com
23 127.0.0.1      tato.local
24 127.0.0.1      tipificacion.local
25 127.0.0.1      vistatipificacion.local
26 127.0.0.1      tipificacionB.local
27 127.0.0.1      live.virtualdj.com
28
```

Fuente: Propia

Ilustración 33: Edición del archivo httpd-vhost.config del servidor de aplicaciones

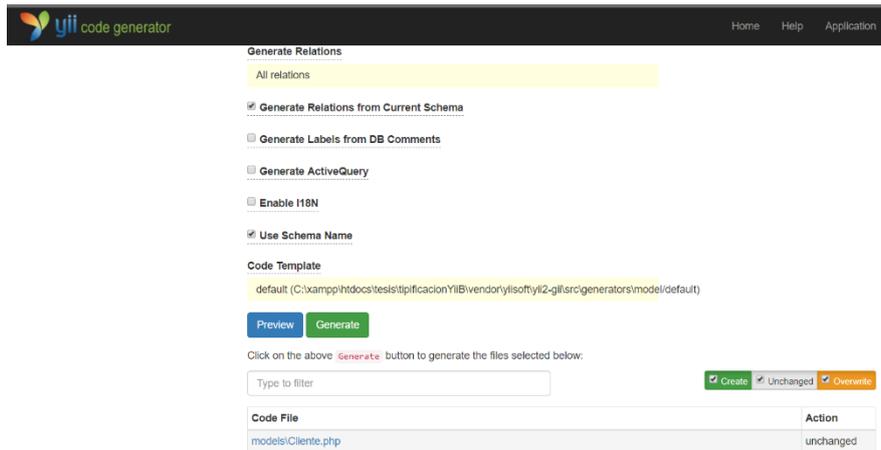
```
httpd-vhosts.conf x
33  ##CustomLog "logs/dummy-host.example.com-access.log" common
34  ##</VirtualHost>
35
36  ##<VirtualHost *:80>
37  ##ServerAdmin webmaster@dummy-host2.example.com
38  ##DocumentRoot "C:/xampp/htdocs/dummy-host2.example.com"
39  ##ServerName dummy-host2.example.com
40  ##ErrorLog "logs/dummy-host2.example.com-error.log"
41  ##CustomLog "logs/dummy-host2.example.com-access.log" common
42  ##</VirtualHost>
43
44  <VirtualHost *:80>
45  ServerAdmin tato.cre@gmail.com
46  DocumentRoot "C:/xampp/htdocs/react"
47  ServerName vistatipificacion.local
48  </VirtualHost>
49
50  <VirtualHost *:80>
51  ServerAdmin tato.cre@gmail.com
52  DocumentRoot "C:/xampp/htdocs/tesis/tipificacion"
53  ServerName tipificacion.local
54  </VirtualHost>
55
56  <VirtualHost *:80>
57  ServerAdmin tato.cre@gmail.com
58  DocumentRoot "C:/xampp/htdocs/tato"
59  ServerName tato.local
60  </VirtualHost>
61
62  <VirtualHost *:80>
63  ServerAdmin tato.cre@gmail.com
64  DocumentRoot "C:/xampp/htdocs/tesis/tipificacionYiib"
65  ServerName tipificacionb.local
66  </VirtualHost>
67
68
```

Fuente: Propia.

Con el generador de código de Yii crearemos los modelos de las tablas de nuestra base de datos las cuales son:

- Cliente
- User
- Factor Rh
- Tipo sanguíneo
- Ciudad
- Provincia
- Singup

Ilustración 34: Generando los modelos con Gii



Fuente: Propia

A continuación, crearemos los controladores en los cuales desarrollaremos nuestras APIs Restfull los controladores a crear son:

- Login
- User
- Cliente
- Seguridad
- Tipo Sanguíneo
- Ciudad

Ilustración 35: Creación de Controladores *.php



Fuente: Propia

Una configurada la plantilla de Yii y después de haber escrito el código para la creación de los formularios de ingreso y el mantenimiento de los usuarios los resultados obtenidos se muestran en las siguientes ilustraciones.

Ilustración 36: Formulario Login.

The screenshot shows a login form on the left and a red warning box on the right. The form has two input fields: 'Usuario' with the value '@ ADMIN' and 'Contraseña' with masked characters. Below the fields are two buttons: 'INGRESAR' (red) and 'Consultar sin registro' (grey). The red box contains the text 'Leer importante...' followed by a paragraph: 'Para poder acceder al sistema de Tipificación de la Cruz Roja Ecuatoriana Junta Provincial de Imbabura es necesario tener un usuario de acceso en el caso de no disponer de uno consulte con el administrador del sistema'.

Fuente: Propia.

Ilustración 37: Administración de Usuarios

Crear Usuario

The 'Crear Usuario' form includes four input fields: 'Usuario', 'Email', 'Contraseña', and 'Perfil'. The 'Perfil' field is a dropdown menu currently showing '-- Seleccione --'. At the bottom are two buttons: 'Guardar' (blue) and 'Cancelar' (grey).

Administración de Usuario

Usuario	Correo	Estado	Acciones
admin	tato.cre@gmail.com	Activo	Cambio Clave Dar Baja Desactivar Admin

Fuente: Propia.

Ilustración 38: Formulario para cambio de contraseña

Cambio de Clave

The 'Cambio de Clave' form has a 'Usuario' field with the value 'ADMIN'. Below it are three input fields: 'Nueva Contraseña', 'Repetir nueva Contraseña', and a third empty field. At the bottom are two buttons: 'Guardar' (blue) and 'Cancelar' (grey).

Fuente: Propia

2.10.1.1. Especificación de pruebas: Ingreso al sistema

Historial de revisiones

Tabla 22: Historial de revisiones - Historia de usuario 1.

Fecha	Versión	Descripción	Autor
05/12/2018	1.0	Ingreso	Carlos Aguirre Benavides

Fuente: Propia.

Descripción

Mediante este documento se lleva un control detallado de las diferentes revisiones realizadas a la Historia de Usuario 1 (Ingreso al Sistema).

En la historia de usuario 1 se controla el acceso a usuarios no autorizados al sistema, para lo cual se debe crear usuario usuarios y verificar los datos al momento de la autenticación, además de realizar las validaciones correspondientes en los diferentes campos del formulario.

Registro de datos incorrectos

- **Descripción**

Para poder acceder al sistema el usuario debe hacerlo mediante el uso de un nombre de usuario y una contraseña; en caso de ser el primer usuario que accede al sistema deberá acceder como administrador, y posteriormente crear los usuarios que podrán acceder al sistema ingresando datos válidos caso contrario el sistema le mostrará un mensaje de error y no podrá crear el usuario.

- **Condiciones de ejecución**

Ninguna

- **Entrada**

No se introducen todos los datos

Se introducen datos incorrectos.

- **Resultado esperado**

En la siguiente ilustración se muestra el mensaje de error cuando no se introduce el usuario y la contraseña.

Ilustración 39: Muestra de errores de campos incompletos en Login

The image shows two side-by-side screenshots of a login form. The left screenshot shows the 'Usuario' field with an '@' symbol and a tooltip saying 'Completa este campo'. The 'Contraseña' field is filled with dots. Below the fields are buttons for 'INGRESAR' and 'Consultar sin registro'. The right screenshot shows the 'Usuario' field with '@ ADMIN' and the 'Contraseña' field with a tooltip saying 'Completa este campo'. Below the fields are buttons for 'INGRESAR' and 'Consultar sin registro'.

Fuente: Propia.

En la siguiente ilustración se muestra el mensaje de error cuando un usuario no existe.

Ilustración 40: Mensaje de error usuario inexistente.



Fuente: Propia.

En la siguiente ilustración se muestra el mensaje de error cuando la contraseña ingresada no es la correcta.

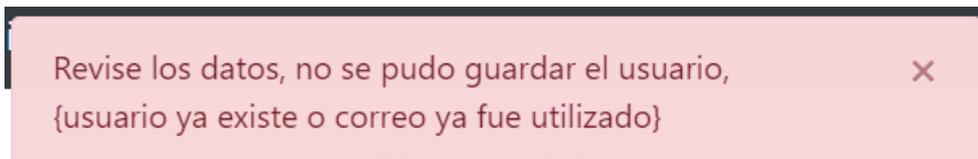
Ilustración 41: Mensaje de error contraseña incorrecta.



Fuente: Propia.

En la siguiente ilustración se muestra el mensaje de error cuando se intenta introducir datos repetidos de usuarios ya registrados

Ilustración 42: Mensaje de error no se puede guardar un usuario nuevo.



Fuente: Propia.

Registro datos correctos

- **Descripción**

El ingreso al sistema se debe realizar con los datos de un usuario registrado en el sistema, al registrar un nuevo usuario se debe ingresar datos reales.

- **Condiciones de ejecución**

Ninguna.

- **Entrada**

El usuario introduce datos correctos.

El usuario ingresa todos los campos requeridos.

- **Resultado esperado**

En la siguiente ilustración se muestra la pantalla de bienvenida si los datos ingresados en el formulario de ingreso fueron correctos.

Ilustración 43: Pantalla de bienvenida



Fuente: Propia.

En la siguiente ilustración se muestra la pantalla que se muestra al guardar un usuario correctamente (TECNICO).

Ilustración 44: Lista de Usuarios registrados.

Administración de Usuario			Crea usuario
Usuario	Correo	Estado	Acciones
admin	tato.cre@gmail.com	Activo	Cambio Clave Dar Baja Desactivar Admin
TECNICO	mail1@mail.com	Activo	Cambio Clave Dar Baja Activar Admin

Fuente: Propia.

2.10.2. Historia de Usuario 2: Gestión de pacientes

Tareas:

- Crear formulario administración de clientes
- Crear vista de listado de clientes

En esta tarea se realizará la creación de los formularios para la administración de clientes tanto el de crear, modificar y eliminar clientes, así como también la página que mostrara el listado de clientes registrados y la impresión de su carnet de tipo de sangre en el sistema como se muestran en las siguientes ilustraciones.

Ilustración 45: Pantalla añadir clientes

Añadir Cliente

Cédula

Nombre

Apellidos

Teléfono

Email

Fecha Nacimiento

Factor RH

Tipo Sanguíneo

Ciudad

Dirección

Fuente: Propia.

Ilustración 46: Pantalla para editar cliente.

Editar Cliente

Cédula

1003770482

Nombre

Carlos David

Apellidos

Aguirre Benavides

Teléfono

0996716563

Email

tato.cre@gmail.com

Fecha Nacimiento

17/02/1992

Factor RH

POSITIVO

Tipo Sanguíneo

B

Ciudad

Ibarra

Dirección

Los Ceibos

Guardar

Cancelar

Fuente: Propia.

Ilustración 47: Mensaje para confirmar eliminar cliente.

localhost:3000 dice

Estas seguro de eliminar este registro?

Aceptar Cancelar

Fuente: Propia.

Ilustración 48: Pantalla para impresión de carnet.

Imprimir

Carlos David Aguirre Benavides

2019-2-25

POSITIVO

Fuente: Propia.

2.10.2.1. Especificación de pruebas: Gestión de pacientes.

Historial de revisiones

Tabla 23: Historial de revisiones - Historia de usuario 2

Fecha	Versión	Descripción	Autor
15/12/2018	1.0	Gestión de pacientes	Carlos Aguirre Benavides

Fuente: Propia.

Descripción

Mediante este documento se lleva un control detallado de las diferentes revisiones realizadas a la Historia de Usuario 2 (Gestión de pacientes).

En la historia de usuario 2 se generará la administración de los clientes en el ingreso y edición se registrarán datos personales, además de realizar las validaciones correspondientes en los diferentes campos del formulario.

Registro de datos incorrectos

- **Descripción**

Para poder gestionar los clientes es necesaria tener datos correctos los mismos que serán ingresados al sistema por el técnico de laboratorio los datos a ingresar son

- Cedula
- Nombres
- Apellidos
- Teléfono
- Email
- Fecha de Nacimiento
- Factor RH
- Tipo Sanguíneo
- Ciudad
- Dirección

- **Condiciones de ejecución**

Ninguna

- **Entrada**

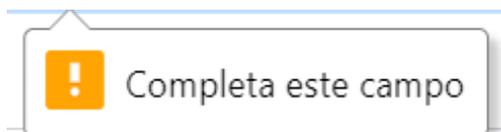
No se introducen todos los datos

Se introducen datos incorrectos.

- **Resultado esperado**

La siguiente ilustración nos muestra la advertencia de completar el campo dicha advertencia aparece en todos los campos al ingresar y editar un cliente

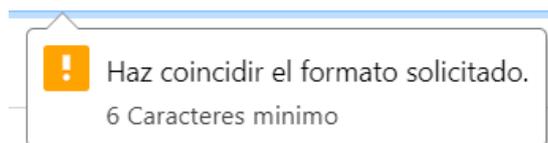
Ilustración 49: Advertencia completa este campo añadir/editar clientes.



Fuente: Propia.

La siguiente ilustración nos muestra la advertencia nos indica que debemos ingresar el campo con el formato requerido en este caso nos dice que deben ser 6 caracteres mínimos porque los pasaportes cuentan con 6 dígitos.

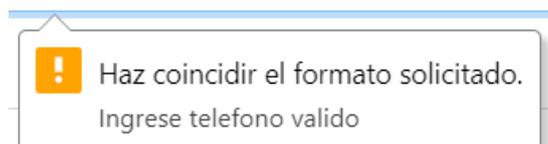
Ilustración 50: Advertencia ingresar formato correcto de identificación.



Fuente: Propia.

La siguiente ilustración nos muestra la advertencia para ingresar correctamente el número de teléfono.

Ilustración 51: Advertencia ingresar formato correcto de teléfono.



Fuente: Propia.

La siguiente ilustración nos muestra la advertencia para ingresar correctamente el correo electrónico.

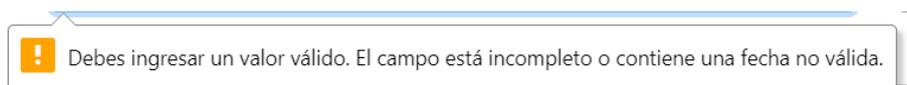
Ilustración 52: Advertencia ingresar formato correcto de email



Fuente: Propia.

La siguiente ilustración nos muestra la advertencia para ingresar correctamente la fecha de nacimiento.

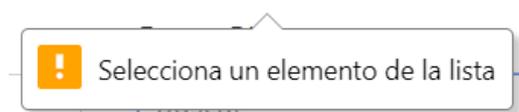
Ilustración 53: Advertencia ingresar formato correcto de fecha.



Fuente: Propia.

La siguiente ilustración nos muestra la advertencia para seleccionar un elemento de la lista de los campos: tipo sanguíneo, factor Rh y ciudad

Ilustración 54: Advertencia seleccionar un campo de lista.



Fuente: Propia.

Registro datos correctos

- **Descripción**

El usuario ingresa todos los datos correctos del cliente y ejecuta las acciones permitidas en el sistema

- **Condiciones de ejecución**

Ninguna.

- **Entrada**

El usuario introduce datos correctos.

El usuario ingresa todos los campos requeridos.

- **Resultado esperado**

La siguiente ilustración nos muestra el mensaje al momento de ingresar un nuevo cliente.

Ilustración 55: Mensaje al añadir correctamente un cliente.



Fuente: Propia.

La siguiente ilustración nos muestra la lista de clientes después de haber ingresado un nuevo cliente.

Ilustración 56: Pantalla al añadir un nuevo cliente.

Una captura de pantalla de una interfaz de usuario para la gestión de clientes. Incluye un botón "Nuevo Cliente", un campo de búsqueda "Buscar por Identificación", un botón "Exportar csv" y una tabla con columnas: Nombres, Apellidos, Identificación, Ciudad y Acciones. Las acciones incluyen "Editar", "Borrar" y "Imprimir".

Nombres	Apellidos	Identificación	Ciudad	Acciones
Juan	Aguirre	1002042818	Ibarra	Editar Borrar Imprimir
Johanna	Aguirre Benavides	1003849278	Ibarra	Editar Borrar Imprimir
Carlos David	Aguirre Benavides	1003770482	Ibarra	Editar Borrar Imprimir

Fuente: Propia.

La siguiente ilustración nos muestra el mensaje al momento de editar un cliente.

Ilustración 57: Mensaje edición correcta de clientes



Fuente: Propia.

2.10.3. Historia de Usuario 3: Creación de formulario de búsquedas y exportaciones

Tareas:

- Implementar el formulario para las búsquedas.
- Implementar el botón de exportación de las búsquedas.

La implementación del formulario y el botón de exportación ayudara a los usuarios administradores a generar los reportes necesarios para un correcto control del servicio.

- **Resultado esperado**

En la siguiente ilustración se muestra la página de búsquedas y en este caso filtrado el apellido “Aguirre” y el tipo sanguíneo “O”

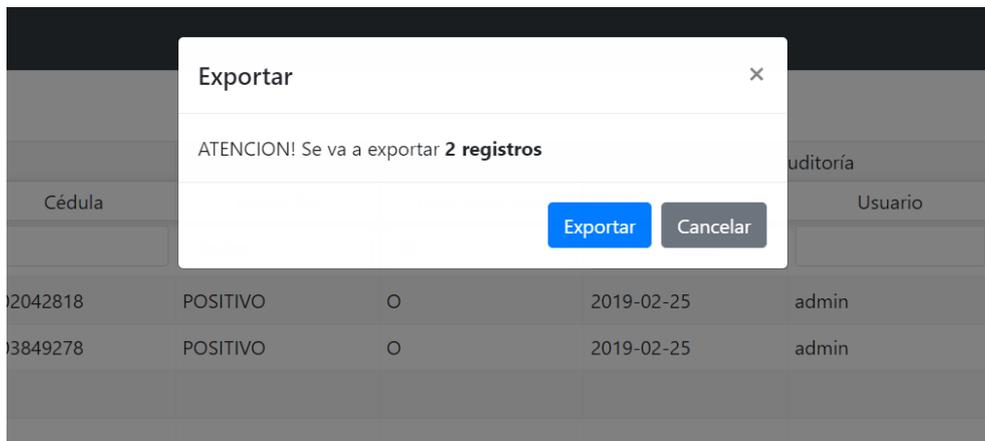
Ilustración 59: Pagina de búsqueda datos filtrados

Datos Clientes			Datos sanguíneos		Datos auditoría		Datos geográficos	
Nombres	Apellidos	Cédula	Factor RH	Tipo Sanguíneo	Fecha creación	Usuario	Ciudad	Provincia
	Aguirre		Todos	O	dd/mm/aaaa			
Juan	Aguirre	1002042818	POSITIVO	O	2019-02-25	admin	Ibarra	Imbabura
Johanna	Aguirre Benavides	1003849278	POSITIVO	O	2019-02-25	admin	Ibarra	Imbabura

Fuente: Propia.

La siguiente ilustración nos muestra el mensaje para confirmar la exportación de los datos filtrados.

Ilustración 60: Mensaje confirmación exportación registros filtrados.



Fuente: Propia.

2.10.4. Historia de Usuario 4: Creación de formulario de consulta de pacientes registrados

Tareas:

- Generar un formulario para consulta de pacientes registrados.

Sin la necesidad de estar registrados en el sistema cualquier persona podrá consultar si se encuentra o no registrado en la base de datos de la CRE JPI en el servicio de laboratorio.

Ilustración 61: Formulario consulta externa.



Fuente: Propia.

2.10.4.1. Especificación de pruebas: Consulta para pacientes.

Historial de revisiones

Tabla 25: Historial de revisiones - Historia de usuario 4

Fecha	Versión	Descripción	Autor
28/12/2018	1.0	Consulta Ext	Carlos Aguirre Benavides

Fuente: Propia.

Descripción

Mediante este documento se lleva un control detallado de las diferentes revisiones realizadas a la Historia de Usuario 4 (Creación de formulario de consulta de pacientes registrados).

En la historia de usuario 4 se generará el formulario para consulta de los pacientes registrados, además de realizar las validaciones correspondientes en los diferentes campos del formulario.

Registro de datos incorrectos

- **Descripción**

Para realizar la búsqueda deberá tener el número de identificación del paciente a buscar en el sistema

- **Condiciones de ejecución**

Ninguna

- **Entrada**

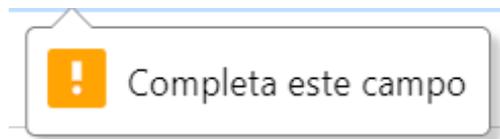
No se introducen todos los datos

Se introducen datos incorrectos.

- **Resultado esperado**

En la siguiente ilustración se muestra la advertencia de completar el campo Identificación.

Ilustración 62: completar campo en consulta externa.



Fuente: Propia.

La siguiente ilustración nos muestra el mensaje de error cuando no se encuentran datos para mostrar en la consulta.

Ilustración 63: Mensaje de error no se encuentran coincidencias.



Fuente: Propia.

Registro datos correctos

- **Descripción**

El usuario ingresa los datos correctos al sistema

- **Condiciones de ejecución**

Ninguna.

- **Entrada**

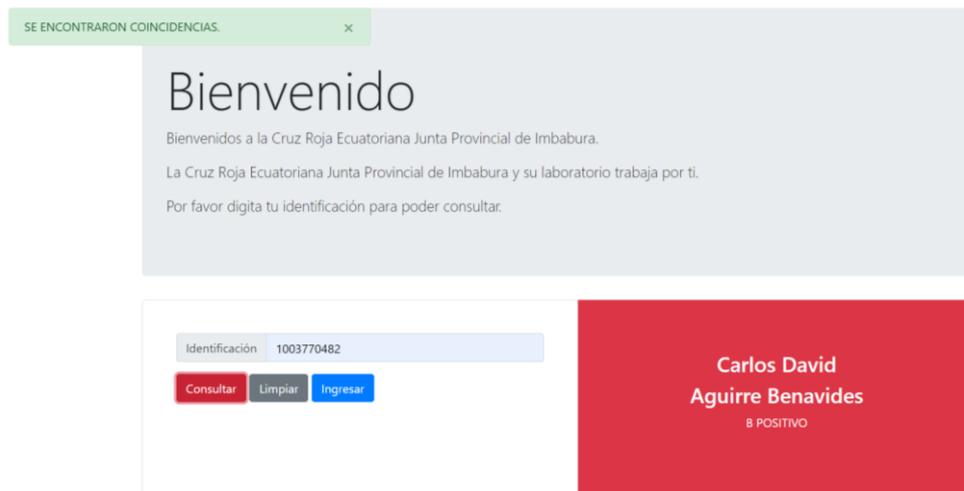
El usuario introduce datos correctos.

El usuario ingresa todos los campos requeridos.

- **Resultado esperado**

En la siguiente ilustración se muestra el resultado de la búsqueda con datos correctos.

Ilustración 64: Pantalla de consulta con datos correctos.



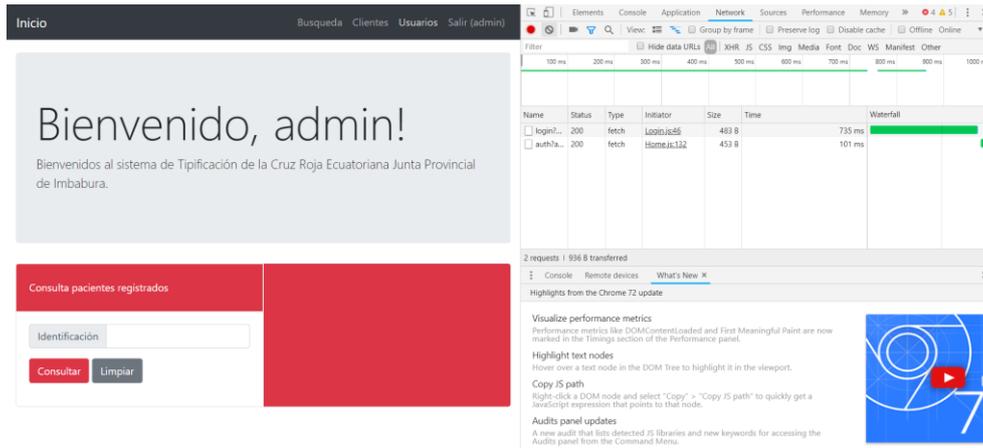
Fuente: Propia.

CAPITULO III

3. Resultados

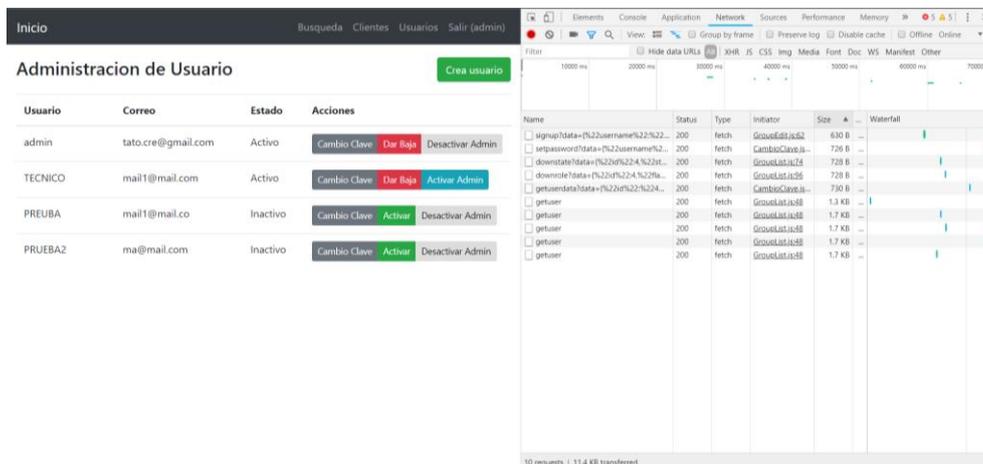
La presente investigación tiene como principal propósito la integración del framework PHP Yii2 y la librería JavaScript React.Js es por esto que en las siguientes ilustraciones se muestran los consumos de las APIs de Yii en las vistas creadas en React.Js

Ilustración 65: APIs consumidas en el Login.



Fuente: Propia.

Ilustración 66: APIs consumidas en la gestión de usuarios.



Fuente: Propia.

Ilustración 67: APIs consumidas en la gestión de clientes

The screenshot displays a web application interface for managing clients. On the left, there is a search bar labeled 'Buscar por identificación' and a table with columns: Nombres, Apellidos, Identificación, Ciudad, and Acciones. The table contains one entry for 'Carlos David Aguirre Benavides' with identification '1003770482' and city 'Ibarra'. On the right, the browser's network panel shows 15 requests, including 'getcliente' and 'getprovincias'.

Fuente: Propia.

Ilustración 68: APIs consumidas en la búsqueda.

The screenshot displays a search interface with an 'Exportar' button and a table with columns: Nombres, Apellidos, Cédula, Factor RH, Tipo Sang..., Fecha crea..., Usuario, and Ciuda... The table contains one entry for 'Carlos David Aguirre Be...' with identification '1003770482', blood type 'POSITIVO', and city 'Ibarra'. The network panel shows 1 request for 'getclientes'.

Fuente: Propia.

3.1. Análisis Impacto

3.1.1. Económico.

Mediante la implementación de este sistema la CRE JPI podrá realizar canjes de carnets de tipo de sangre ya que actualmente no cuenta con una base de datos con la cual se puedan realizar consultas de si un paciente ya se ha realizado una prueba de tipificación sanguínea; actualmente se realizan canjes de carnets de tipo de sangre solo si el paciente presenta una copia del carnet que se le entrego anteriormente.

Mediante un análisis de costos se obtiene que una tipificación sanguínea cuesta \$1,55 y un canje \$0,73

Tabla 26: Costo examen tipificación

EXAMEN	DETALLE	CANT	VALOR	COSTO	VALOR VENTA
	LAMINA PORTA OBJETO	1	0,01		
	ANTI-A	100	0,05		
	ANTI-B	100	0,05		
	ANTI DRH	100	0,09		
	LANCETA DESCARTABLES X 100	1	0,04		
TIPIFICACION	CARNTÓN DE TIPIFICACIÓN	1	0,06	1,55	5,00
IB	IMPRESIÓN DE HOJA/ SOBRE Y FACTURA	1	0,10		
	PORCIÓN DE ALGODÓN	1	0,01		
	ALCOHOL	1	0,08		
	SERVICIOS BASICOS	1	0,10		
	HONORARIOS TECNICOS	1	0,75		
	MATERIAL DE LAVADO	1	0,22		

Fuente: (Cuaran, 2018)

Tabla 27: Costo canje carnet

EXAMEN	DETALLE	CANT.	VALOR UNI.	COSTO	VALOR VENTA
CANJE CARNE	CARTÓN DE TIPIFICACIÓN	1	0,06	0,73	3,00
	IMPRESIÓN FACTURA	1	0,01		
	SERVICIOS BASICOS	1	0,10		
	HONORARIOS TECNICOS	1	0,50		
	IMPRESIÓN CARNET	1	0,06		

Fuente: (Cuaran, 2018).

Haciendo un análisis de utilidad tenemos que realizar una prueba de tipificación se genera una ganancia de 322% mientras que si se realiza un canje la ganancia es de 411%.

Con estos datos podemos determinar que al utilizar el sistema y poder ofrecer canje de carnets la CRE JPI tendrá mayores ganancias y se reducirá el tiempo de entrega de carnet ya que solamente se reimprimirá el carnet sin necesidad de ingresar los datos nuevamente al formato de Excel que se ocupa actualmente.

3.2. Beneficio de la investigación.

- La presente investigación muestra a los desarrolladores la capacidad que tiene tanto Yii2 y React.Js para combinarse y para ser exactos el poder consumir las APIs Restfull de Yii2 en un Frontend desarrollado con React.Js

3.3. Beneficios del uso del sistema en CRE JPI

- Facilita un adecuado registro de los pacientes que se realizan el examen de tipificación.
- Conserva seguros los datos de los pacientes entregados a la CRE JPI los cuales serán utilizados únicamente con el propósito de entregar un carnet de tipo de sangre.
- Permite a los administradores del sistema conocer con exactitud las pruebas de tipificación que se realizaron en el laboratorio con la extracción de los reportes.
- Asegura que los clientes puedan contar con el servicio de canje de carnet.
- Facilita a los clientes un formulario para poder consultar si sus datos se encuentran en la base de datos de la CRE JPI

3. Conclusiones y Recomendaciones

4.1. Conclusiones.

- El aplicativo web que se desarrolló permite a la CRE JPI tener un registro de los pacientes que se realizan el examen de tipificación sanguínea y con estos registros se podrá extraer informes acerca de los pacientes y también se abre la posibilidad a la entrega de canjes de carnets de tipo de sangre.
- Yii 2 es un framework robusto que permite a los desarrolladores implementar cualquier aplicativo web, teniendo como respaldo una gran comunidad que aporta a los desarrolladores documentación y ejemplos del uso de este.
- La librería React.JS es una librería que ha tomado fuerza en los últimos años ya que esta nos permite tener desarrollos ágiles, ordenados y con una arquitectura mantenible haciendo que los aplicativos sean más rápidos con interfaces atractivas a los usuarios.
- La integración que se logró entre el Framework Yii2 y la librería React.Js fue optima ya que con Yii se desarrolló las APIs Restfull que fueron consumidas por la librería React.Js sin ninguna complicación.
- El uso de la metodología XP apporto al desarrollo del aplicativo en gran medida ya que esta metodología se adaptó perfectamente al ritmo de trabajo y a las necesidades para el desarrollo del aplicativo.
- Es importante recalcar que al momento de iniciar este proyecto la documentación sobre React.js era escasa y se logró desarrollar el aplicativo gracias a que al ser una librería de JavaScript era muy similar a otras como la de AngularJS.

4.2. Recomendaciones.

- Al momento de la elección de una herramienta es necesario verificar primero que dicha herramienta cuente con soporte e información para su uso ya que como se pudo evidenciar en este proyecto la librería React.JS no contaba con suficiente información para el desarrollo de un aplicativo web.
- Para implementar un proyecto con Yii2 es necesario conocer el alcance del proyecto ya que este framework propone dos plantillas una básica y una avanzada y es indispensable saber elegir cual de las dos se utilizará en este proyecto se ocupo la plantilla avanzada lo que permite escalabilidad al momento de pretender implementar más módulos.
- La implementación del aplicativo en la CRE JPI esta pensado para resolver procesos de la mejor manera tomando en cuenta que el diseño esta orientado a que este sea intuitivo y fácil de manejar.
- Se debe tomar en cuenta que este aplicativo no solo es funcional para la CRE JPI ya que este podrá ser implementado a cualquier filial de la Cruz Roja Ecuatoriana en la cual se realice este examen.
- La metodología escogida para el desarrollo del aplicativo cumplió con lo esperado ya que la misma apunta a un correcto trabajo en equipo permite que el desarrollo no se trunque es por esto que se recomienda una buena comunicación entre los integrantes del equipo de desarrollo.

5. Bibliografía

- Alvares. Miguel, D. M. (2016). Manual de React JS. *desarrolloweb.com*.
- Alvarez, M. A. (18 de Julio de 2007). *desarrolloweb.com*. Obtenido de <https://desarrolloweb.com/articulos/xampp.html>
- Apache, F. (2018). *Apache Friends*. Obtenido de https://www.apachefriends.org/es/download_success.html
- Chacon, S., & Straub, B. (2015). *Pro Git*. Apress.
- Composer, C. (2013). *Composer*. Obtenido de <https://getcomposer.org/doc/00-intro.md>
- Cuaran, R. (1 de Septiembre de 2018). Gerente Comercial CRE JPI. (C. Aguirre, Entrevistador)
- F. Sierra, J. A. (2013). *Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web*. unisimon.
- Garcia. Francisco, H. A. (2017 -2018). *FUNDAMENTOS DE LA VISTA DE CASOS DE USO*. Salamanca: Universidad de Salamanca.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Marcombo.
- Jimenez, J. (2015). *DESCUBRE REACT*. leanpub.
- Joskowicz. José. (2008). *Reglas y Prácticas en eXtreme Programming*.
- Mateu, C. (2004). *Desarrollo de aplicaciones web*. Barcelona: Eureka Valencia Alzaga.
- Oracle, a. i. (2014). *MySQL 5.0 Reference Manual*.
- Perugachi, V. (1 de Septiembre de 2018). Administradora CRE JPI. (C. Aguirre, Entrevistador)

Rodriguez, C. (20 de Mayo de 2015). *Metodologías Ágiles*. Obtenido de <https://comunidad.iebschool.com/metodologiasagiles/general/concepto-metodologias-agiles/>

Vega, A. G. (20 de Julio de 2017). Quienes Somos. (C. A. Benavides, Entrevistador)

Velasco, J. (27 de 02 de 2014). *Hipertextual*. Obtenido de <https://hipertextual.com/archivo/2014/02/atom-editor-texto-desarrollo-github/>

Xue, Qiang; Makarov, Alexander; Brandt, Carsten; Paul, Klimov;. (2014). *La Guía Definitiva de Yii2* . Yii Software LLC.