

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS COMPUTACIONALES

**ESTUDIO DE INTEGRACIÓN DE LOS FRAMEWORKS ANGULAR 4 Y YII2,  
ORIENTADO A SERVICIOS REST, QUE PERMITAN LA GESTIÓN Y CONTROL  
DE INVENTARIOS PARA MEJORAR LA PRODUCTIVIDAD EN LA EMPRESA  
INDUXION**

AUTOR:

ANDRÉS RUBÉN GUDIÑO QUINTEROS

DIRECTOR:

ING. DIEGO JAVIER TREJO ESPAÑA, MSC.

IBARRA, 2018



## 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 del 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			
<b>CÉDULA DE IDENTIDAD:</b>	091598471-0		
<b>APELLIDOS Y NOMBRES:</b>	GUDIÑO QUINTEROS ANDRÉS RUBÉN		
<b>DIRECCIÓN:</b>	IBARRA, AV. VÍCTOR MANUEL PEÑAHERRERA 2-74 Y ROBERTO POSSO		
<b>EMAIL:</b>	argudinioq@utn.edu.ec		
<b>TELÉFONO FIJO:</b>	062607138	<b>TELÉFONO MÓVIL:</b>	0987382877

DATOS DE LA OBRA	
<b>TÍTULO:</b>	ESTUDIO DE INTEGRACIÓN DE LOS FRAMEWORKS ANGULAR 4 Y YII2, ORIENTADO A SERVICIOS REST, QUE PERMITAN LA GESTIÓN Y CONTROL DE INVENTARIOS PARA MEJORAR LA PRODUCTIVIDAD EN LA EMPRESA INDUXION
<b>AUTOR (ES):</b>	GUDIÑO QUINTEROS ANDRÉS RUBÉN
<b>FECHA: DD/MM/AAAA</b>	21/11/2018
SOLO PARA TRABAJOS DE GRADO	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERO EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR:</b>	MSC. DIEGO JAVIER TREJO ESPAÑA

## 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 la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de noviembre de 2018

**EL AUTOR:**

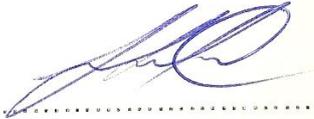
(Firma) .....  .....

Nombre: Andrés Rubén Gudiño Quinteros

## CERTIFICACIÓN DIRECTOR

Por medio del presente certifico que el Sr. Andrés Rubén Gudiño Quinteros, portador de la cédula de identidad Nro. 091598471-0 ha trabajado en el desarrollo del proyecto de tesis: **“ESTUDIO DE INTEGRACIÓN DE LOS FRAMEWORKS ANGULAR 4 Y YII2, ORIENTADO A SERVICIOS REST, QUE PERMITAN LA GESTIÓN Y CONTROL DE INVENTARIOS PARA MEJORAR LA PRODUCTIVIDAD EN LA EMPRESA INDUXION”**, 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.

A handwritten signature in blue ink, appearing to read 'Diego Trejo España', is written over a horizontal dotted line. The signature is contained within a light yellow rectangular background.

MSc. Diego Javier Trejo España

**DIRECTOR DE TESIS**

## CERTIFICACIÓN EMPRESA AUSPICIANTE



Ibarra, 26 de septiembre de 2018

### CERTIFICADO

La empresa INDUXION, certifica, que el Sr. Andrés Rubén Gudiño Quinteros, portador de la cédula de ciudadanía Nro. 0915984710, ha implementado de forma satisfactoria el sistema de **"GESTIÓN Y CONTROL DE INVENTARIOS PARA MEJORAR LA PRODUCTIVIDAD"**. Nos es grato informar que la aplicación ha sido desarrollada en su totalidad cumpliendo los requerimientos solicitados.

Por lo que se recibe el proyecto cómo culminado, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa.

El Sr. Andrés Rubén Gudiño Quinteros, puede hacer uso de este documento para fines pertinentes de la Universidad Técnica del Norte.

Atentamente.

  
Sr. Luis Eduardo Núñez



**Gerente Comercial Empresa INDUXION**

 [induxion@hotmail.com](mailto:induxion@hotmail.com)  
 099 878 0137 / 099 163 3065 / 062 615 057  
 Cdla. Los Ceibos Río Blanco 4-81 y Av. El Retorno



## DEDICATORIA

Esta presente tesis está dedicada a mi héroe de vida mi Padre Rubén Gudiño Albuja quien, con sus consejos y lucha constante en sus duros momentos, supo entregarme el más preciado regalo en vida; su valentía, su tenacidad y constancia los tomé como míos y paso a paso este sueño se hizo realidad, convertirme en un profesional a carta cabal. Sin duda mi dedicatoria será mirando al cielo en donde sé que tú estás junto a Dios. Gracias por ser mi compañero en esta gran etapa de mi vida.

A mi segunda fuerza y motor de vida mi Madre Marcia Quinteros Lucumí, mi valiente tesoro y compañera que en todo momento ha sido mi soporte y creyó en este sueño que ahora es nuestro. Gracias por dedicarme tu vida, tu esfuerzo y tus ganas de continuar acompañándome en mis éxitos y logros. Recuerda que tenemos muchos más por disfrutar luego de este logro que es nuestro. Te amo.

A mi hermana Vero quien me enseñó que todo esfuerzo lleva su tiempo, si pensamos en la recompensa que tiene, las largas noches y el duro caminar en la superación personal y profesional.

A Maye Núñez por su apoyo constante en este camino hacia mi vida profesional. Gracias por cada palabra otorgada cuando los caminos se ponían duros y motivarme para continuar.

**Andrés Rubén Gudiño Quinteros**

## **AGRADECIMIENTOS**

A la Universidad Técnica del Norte quien me abrió las puertas para lograr lo que ahora soy un profesional, proactivo e innovador. Con miras al aporte comunitario y al desarrollo de actividades que se enfoquen a la continua formación de profesionales.

A mis docentes en cada uno de mis años de formación quienes muchos entendieron mi esfuerzo a diario y otros se convirtieron en la energía necesaria para continuar con esta lucha de superación.

A la empresa INDUXION por permitir realizar mi trabajo de grado en la empresa y brindarme las facilidades y aplicación del sistema informático.

A mi tutor MSc. Diego Trejo España, por el apoyo constante en el término de mi trabajo. Sus indicaciones fueron fundamentales para desarrollar un programa útil que fomentará la aplicación de las tecnologías de la información en las entidades privadas, como elemento base de optimización en tiempos y servicios de calidad a los clientes.

A mis compañeros quienes fueron parte de cada aprendizaje, en las aulas que jamás olvidaré.

**Andrés Rubén Gudiño Quinteros**

## TABLA DE CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICACIÓN .....	ii
CERTIFICACIÓN DIRECTOR .....	iv
CERTIFICACIÓN EMPRESA AUSPICIANTE .....	v
DEDICATORIA .....	vi
AGRADECIMIENTOS .....	vii
TABLA DE CONTENIDO .....	viii
ÍNDICE DE TABLAS .....	xii
ÍNDICE DE FIGURAS .....	xiv
RESUMEN .....	xviii
ABSTRACT .....	xix
INTRODUCCIÓN .....	1
Antecedentes .....	1
Situación Actual .....	2
Prospectiva .....	2
Planteamiento del Problema .....	3
Objetivos .....	3
Objetivo General .....	3
Objetivos Específicos .....	3
Alcance .....	4
Arquitectura de la Aplicación .....	4
Justificación .....	5
CAPÍTULO 1 .....	7
1. Marco Teórico .....	7
1.1. Empresa INDUXION .....	7
1.1.1. Situación Actual .....	7
1.1.2. Proyección del sistema de gestión y control de inventarios .....	8
1.1.3. Procesos manuales en la empresa INDUXION .....	8

1.2.	Base de datos Relacional “SQL” .....	9
1.2.1.	Base de datos MariaDB .....	10
1.3.	HTML 5.....	10
1.4.	CSS 3.....	10
1.5.	PHP .....	10
1.6.	Lenguaje JavaScript .....	11
1.7.	Lenguaje TypeScript.....	11
1.8.	Node.js .....	11
1.8.1.	NPM.....	12
1.9.	JSON.....	12
1.10.	Protocolo HTTP.....	12
1.11.	Servicios Web REST .....	13
1.11.1.	RESTful API .....	15
1.12.	Framework Angular.....	16
1.12.1.	Arquitectura .....	16
1.12.2.	Arquitectura de una aplicación.....	17
1.12.3.	Angular CLI.....	17
1.12.4.	Estructura de archivos del Framework .....	19
1.12.5.	Características .....	19
1.13.	Yii2 Framework .....	20
1.13.1.	Arquitectura .....	20
1.13.2.	Ciclo de Vida de una petición en Yii2. ....	21
1.13.3.	Estructura de archivos del Framework. ....	22
1.13.4.	Características .....	23
1.14.	Integración de Servicios REST entre los Frameworks Angular y Yii2 .....	23
1.14.1.	Herramientas para el estudio de integración de servicios REST.....	24
1.14.2.	Arquitectura de Integración de servicios REST Frameworks Yii2 – Angular..	25
1.14.3.	Desarrollo de un API REST con el Framework Yii2 .....	26
1.14.4.	Consumo de servicios REST con el Framework Angular .....	36

1.15.	Metodología SCRUM .....	47
1.15.1.	Roles y responsabilidades .....	47
1.15.2.	Elementos de Scrum .....	48
CAPÍTULO 2 .....		51
2.	Desarrollo de la Metodología del Software .....	51
2.1.	Roles del Sistema .....	51
2.2.	Historias de Usuario y Criterios de Aceptación .....	51
2.3.	Pila de Productos .....	54
2.4.	Pila de Tareas .....	55
2.5.	Planificación del Proyecto .....	60
2.6.	Iteraciones .....	61
2.6.1.	Iteración 1. Análisis y estructuración del Proyecto .....	61
2.6.2.	Iteración 2. Creación del Módulo Clientes .....	68
2.6.3.	Iteración 3. Creación del Módulo Negocio .....	71
2.6.4.	Iteración 4. Creación del Módulo Productos .....	77
2.6.5.	Iteración 5. Creación del Módulo Impuestos .....	83
2.6.6.	Iteración 6. Creación del Módulo Mercadería .....	85
2.6.7.	Iteración 7. Creación del Módulo Stock .....	90
2.6.8.	Iteración 8. Creación del Módulo Ventas .....	92
2.6.9.	Iteración 9. Creación del Módulo Kardex .....	96
2.6.10.	Iteración 10. Módulo Seguridad y Control .....	97
2.6.11.	Iteración 11. Implementación del sistema .....	100
CAPÍTULO 3 .....		103
3.	RESULTADOS .....	103
3.1.	Beneficio adquirido con el estudio de integración del Framework Yii2 y Angular 103	
3.2.	Beneficios adquiridos con la implementación del sistema informático en la empresa INDUXION .....	103
3.3.	Beneficios de tiempo en operaciones realizadas en el sistema para mejorar la productividad en la empresa INDUXION .....	104

Conclusiones.....	108
Recomendaciones .....	109
Bibliografía .....	110
Anexos .....	113

## ÍNDICE DE TABLAS

Tabla 1. Registro de anotaciones diarias para reporte de Ventas .....	8
Tabla 2. Comandos para operaciones con Angular CLI .....	18
Tabla 3. Parámetros de conexión a la base de datos desde el Framework Yii2 .....	26
Tabla 4. Parámetros para crear migraciones en Yii2.....	27
Tabla 5. Parámetros para generar componentes en Angular.....	37
Tabla 6. Parámetros para generar un servicio en Angular .....	38
Tabla 7. Roles del Sistema.....	51
Tabla 8. Historias de usuarios y criterios de aceptación.....	52
Tabla 9. Pila de Productos.....	54
Tabla 10. Análisis y Estructuración del Proyecto .....	55
Tabla 11. Creación del Módulo Clientes .....	56
Tabla 12. Creación del Módulo Negocio .....	56
Tabla 13. Creación del Módulo Productos .....	57
Tabla 14. Creación del Módulo Impuestos.....	57
Tabla 15. Creación del Módulo Mercadería .....	58
Tabla 16. Creación del Módulo Stock .....	58
Tabla 17. Creación del Módulo Ventas .....	58
Tabla 18. Creación del Módulo Kardex.....	59
Tabla 19. Módulo de Seguridad y Control.....	59
Tabla 20. Implementación del Sistema .....	59
Tabla 21. Planificación del Proyecto .....	60
Tabla 22. Sprint 1 – Hoja Electrónica.....	67
Tabla 23. Sprint 1 – Pizarrón.....	67
Tabla 24. Sprint 2 – Hoja Electrónica.....	70
Tabla 25. Sprint 2 – Pizarrón.....	70
Tabla 26. Sprint 3 – Hoja Electrónica.....	76
Tabla 27. Sprint 3 – Pizarrón.....	76
Tabla 28. Sprint 4 – Hoja Electrónica.....	82
Tabla 29. Sprint 4 – Pizarrón.....	83
Tabla 30. Sprint 5 – Hoja Electrónica.....	84
Tabla 31. Sprint 5 – Pizarrón.....	84
Tabla 32. Sprint 6 – Hoja Electrónica.....	89
Tabla 33. Sprint 6 – Pizarrón.....	89
Tabla 34. Sprint 7 – Hoja Electrónica.....	91
Tabla 35. Sprint 7 – Pizarrón.....	91

Tabla 36. Sprint 8 – Hoja Electrónica.....	95
Tabla 37. Sprint 8 – Pizarrón.....	95
Tabla 38. Sprint 9 – Hoja Electrónica.....	97
Tabla 39. Sprint 9 – Pizarrón.....	97
Tabla 40. Sprint 10 – Hoja Electrónica.....	100
Tabla 41. Sprint 10 – Pizarrón.....	100
Tabla 42. Sprint 11 – Hoja Electrónica.....	101
Tabla 43. Sprint 11 – Pizarrón.....	101

## ÍNDICE DE FIGURAS

Fig. 1. Planteamiento del problema.....	3
Fig. 2. Arquitectura de la aplicación .....	4
Fig. 3. Flujo de trabajo de la metodología SCRUM .....	5
Fig. 4. Empresa INDUXION.....	7
Fig. 5. Inventario empresa .....	9
Fig. 6. Servicios REST - Cliente – Servidor .....	13
Fig. 7. Servicios REST - Sin estado .....	14
Fig. 8. Servicios REST - Cache.....	14
Fig. 9. Servicios REST – Servicios Uniformes .....	14
Fig. 10. Servicios REST – Arquitectura en capas.....	15
Fig. 11. Arquitectura del framework Angular.....	16
Fig. 12. Arquitectura de una aplicación en el Framework Angular .....	17
Fig. 13. Arquitectura MVC "Modelo – Vista – Controlador" .....	20
Fig. 14. Ciclo de vida de una petición en Yii2 .....	21
Fig. 15. Arquitectura de integración de servicios REST entre Yii2 – Angular .....	25
Fig. 16. Configuración de parámetros de conexión en Visual Studio Code.....	27
Fig. 17. Migración de tablas en Yii2 .....	28
Fig. 18. Verificación Migraciones en Yii2 .....	28
Fig. 19. Línea de comando para migración a la base de datos MariaDB .....	29
Fig. 20. Creación de migración a la base de datos MariaDB .....	29
Fig. 21. Base de datos tab_cliente en servidor MariaDB.....	29
Fig. 22. Generador de código en Yii2, generador de modelo.....	30
Fig. 23. Generador de nuevo modelo en Yii2. ....	30
Fig. 24. Generador del nuevo modelo Yii2.....	31
Fig. 25. Verificar modelo creado en el proyecto de Yii2 Framework .....	31
Fig. 26. Script de configuración ClienteController.php.....	32
Fig. 27. Script de configuración web.php .....	33
Fig. 28. Configuración de servicio REST en Yii2 .....	33
Fig. 29. Prueba del servicio REST para clientes .....	34
Fig. 30. Postman, prueba de servicio REST acción GET .....	34
Fig. 31. Postman, prueba de servicio REST acción PUT .....	35
Fig. 32. Postman, prueba de servicio REST acción POST .....	35
Fig. 33. Postman, prueba de servicio REST acción DELETE .....	36
Fig. 34. Ejecución de terminal integrado de Visual Code para proyecto en Angular .....	36
Fig. 35. Generación de nuevos componentes en Angular.....	37

Fig. 36. Generación de un nuevo servicio en Angular .....	38
Fig. 37. Creación de un nuevo archivo de rutas en Angular.....	38
Fig. 38. Script de configuración cliente.service.ts .....	40
Fig. 39. Script de configuración cliente-lista.component.ts.....	41
Fig. 40. Script de configuración cliente-lista.component.html.....	42
Fig. 41. Script de configuración cliente-actualizar.component.ts .....	43
Fig. 42. Script de configuración Cliente-actualizar.component.html.....	44
Fig. 43. Script de configuración app.routing.ts .....	45
Fig. 44. Importación de los módulos al name space.....	45
Fig. 45. Mostrar los datos del componente en Angular.....	46
Fig. 46. Librerías Bootstrap en Angular.....	46
Fig. 47. Consumo de REST entre Yii2 y Angular.....	47
Fig. 48. Organización en Scrum .....	48
Fig. 49. Elementos de la metodología Scrum .....	49
Fig. 50. Instalación de APACHE.....	61
Fig. 51. Base de datos MariaDB.....	61
Fig. 52. Instalación de Composer.....	62
Fig. 53. Instalación de Visual Studio Code .....	62
Fig. 54. Instalación de Node.js .....	63
Fig. 55. Instalación de TypeScript .....	63
Fig. 56. Framework Yii2.....	64
Fig. 57. Instalación del Framework Yii2 .....	64
Fig. 58. Vista inicial del proyecto instalado del Framework Yii2.....	64
Fig. 59. Postman.....	65
Fig. 60. Instalación de Angular .....	65
Fig. 61. Iniciar proyecto Angular comando ng serve, ruta http://localhost:4200/ .....	65
Fig. 62. Diagrama Entidad Relación Base de datos. ....	66
Fig. 63. Servicio REST cliente.....	68
Fig. 64. Crear cliente. ....	68
Fig. 65. Actualizar cliente.....	69
Fig. 66. Lista de clientes registrados.....	69
Fig. 67. Buscar clientes registrados. ....	69
Fig. 68. Reporte de clientes registrados.....	70
Fig. 69. Servicio REST representante negocio.....	71
Fig. 70. Servicio REST ciudad.....	71
Fig. 71. Servicio REST sucursal.....	71
Fig. 72. Crear representante legal.....	72

Fig. 73. Actualizar representante legal.....	72
Fig. 74. Lista del representante legal. ....	72
Fig. 75. Buscar representante legal. ....	73
Fig. 76. Crear ciudad. ....	73
Fig. 77. Actualizar ciudad. ....	73
Fig. 78. Lista de ciudades.....	74
Fig. 79. Buscar ciudades. ....	74
Fig. 80. Crear tienda. ....	74
Fig. 81. Actualizar tienda. ....	75
Fig. 82. Listar tienda. ....	75
Fig. 83. Listar tienda. ....	75
Fig. 84. Servicio REST categoría. ....	77
Fig. 85. Servicio REST marca. ....	77
Fig. 86. Servicio REST producto. ....	77
Fig. 87. Crear categoría de productos.....	78
Fig. 88. Actualizar categoría de productos.....	78
Fig. 89. Lista categoría de productos. ....	78
Fig. 90. Buscar una categoría de productos. ....	79
Fig. 91. Crear marca de productos.....	79
Fig. 92. Actualizar marca de productos.....	79
Fig. 93. Lista de marcas de productos. ....	80
Fig. 94. Buscar marcas de productos.....	80
Fig. 95. Crear productos. ....	80
Fig. 96. Actualizar productos. ....	81
Fig. 97. Lista de productos. ....	81
Fig. 98. Buscar productos.....	82
Fig. 99. Reporte del catálogo de productos. ....	82
Fig. 100. Servicio REST categoría. ....	83
Fig. 101. Editar impuesto IVA.....	83
Fig. 102. Lista de impuesto. ....	84
Fig. 103. Buscar impuesto.....	84
Fig. 104. Servicio REST proveedor.....	85
Fig. 105. Servicio REST compra. ....	85
Fig. 106. Crear proveedor.....	86
Fig. 107. Actualizar proveedor.....	86
Fig. 108. Lista de proveedor.....	86
Fig. 109. Buscar proveedor .....	87

Fig. 110. Ingreso de compras.....	87
Fig. 111. Listar y buscar compras para el ingreso de mercadería. ....	88
Fig. 112. Registro de una entrada de productos - compras. ....	88
Fig. 113. Registro de una entrada anulada de productos - compras. ....	88
Fig. 114. Servicio REST stock de productos.....	90
Fig. 115. Listar, buscar productos con su respectivo stock. ....	90
Fig. 116. Reporte de productos con su respectivo stock.....	91
Fig. 117. Servicio REST ventas.....	92
Fig. 118. Registro de salida de productos - ventas.....	92
Fig. 119. Listar ventas. ....	93
Fig. 120. Buscar salida de productos registradas - ventas.....	93
Fig. 121. Detalles salida de productos registrada - ventas.....	93
Fig. 122. Detalles salida de productos anuladas en su totalidad - venta. ....	94
Fig. 123. Reporte de salida de productos - ventas. ....	94
Fig. 124. Recaudaciones empresa.....	94
Fig. 125. Reporte del total recaudado. ....	95
Fig. 126. Servicio REST Kardex.....	96
Fig. 127. Listar, buscar entradas y salidas de productos ....	96
Fig. 128. Reporte de entradas y salidas de productos ....	97
Fig. 129. Servicio REST usuarios.....	98
Fig. 130. Gestión de usuarios para el ingreso al sistema.....	98
Fig. 131. Registro de usuarios para el ingreso al sistema.....	98
Fig. 132. Lista de usuarios del sistema.....	99
Fig. 133. Panel Administradores. ....	99
Fig. 134. Panel Vendedores.....	99
Fig. 135. Sistema INDUXION - 1.....	100
Fig. 136. Sistema INDUXION - 2.....	101
Fig. 137. Reporte de productos con sus respectivos valores de venta, stock he impuestos .....	104
Fig. 138. Verificación de Stock de productos en la empresa.....	105
Fig. 139. Reportes entradas y salidas de productos.....	105
Fig. 140. Registro de una venta ....	106
Fig. 141. Total, Recaudado a Diario.....	106
Fig. 142. Total, Recaudado por Fechas.....	107

## RESUMEN

El presente proyecto tiene como propósito realizar un estudio para la integración de los Frameworks Angular y Yii2 orientados a servicios REST, permitiendo comprender el funcionamiento y como relacionarse entre sí, para la creación de un sistema de gestión y control de inventarios que permitirá mejorar la productividad en la empresa INDUXION.

En la introducción se detallan los conceptos, aspectos referentes a la empresa INDUXION, dando a conocer los antecedentes y la necesidad de la implementación de un sistema informático, Además se menciona los objetivos para la realización de este proyecto.

El Capítulo uno, se detalla todo el marco teórico como son los conceptos, detalles de la empresa con sus procesos, tecnología a aplicar, herramientas, la metodología para el desarrollo del sistema como es la metodología SCRUM. También se encuentra presente el estudio de integración de los Frameworks Angular y Yii2 orientados a servicios REST mediante un ejemplo.

El Capítulo dos, se detalla todo el proceso de desarrollo del software mediante la metodología SCRUM y la implementación del sistema en la empresa INDUXION.

El Capítulo tres, se detalla los resultados con la implementación del sistema informático.

## **ABSTRACT**

The purpose of this project is to carry out a study for the integration of the Angular and Yii2 frameworks oriented to REST services, allowing us to understand how they work and how to relate to each other, for the creation of an inventory management and control system that will allow us to improve productivity in the company INDUXION.

In the introduction, the concepts and aspects related to the company INDUXION are detailed, making known the background and the need for the implementation of a computer system. In addition, the objectives for the realization of this project are mentioned.

Chapter one, details the entire theoretical framework such as the concepts, details of the company with its processes, technology to apply, tools, the methodology for the development of the system such as the SCRUM methodology, also present is the study of integration of the Angular and Yii2 frameworks oriented to REST services through an example.

Chapter two details the entire software development process using the SCRUM methodology and the implementation of the system in the company INDUXION.

Chapter three, details the results with the implementation of the computer system.

# INTRODUCCIÓN

## Antecedentes

Angular es un framework que se encuentra estructurada por el lenguaje TypeScript, que es un lenguaje de programación de código abierto, el principal beneficio del Framework Angular en el desarrollo de aplicaciones web, podremos lograr una interfaz fluida y una buena aceptación por parte de los clientes (Darío Blasco, 2016). En Angular la principal característica es la velocidad y rendimiento en plataformas web, propone un control de escalabilidad (Powered by Google, 2017).

Yii2 es un Framework basado en componentes para el desarrollo de aplicaciones de gran escala, se encuentra estructurado en el lenguaje PHP. Este Framework permite realizar un desarrollo de aplicaciones de forma ágil, presenta la arquitectura MVC – Modelo Vista Controlador (Yiiframework.com, 2016).

Yii2 presenta una configuración tanto Frontend para usuarios finales y Backend para la parte administrativa de todo el sistema con los diferentes módulos como configuración de la base de datos, conexión entre servicios (Ivan Alvarado Diaz, 2014).

Servicio REST en siglas “Representational State Transfer”, esta arquitectura de interfaces se basa en un protocolo de comunicación Http donde interactúan cliente / servidor, presentan operaciones definidas en donde los recursos estén identificados de forma única y establecidos por URLs. Los métodos más utilizados son los siguientes: Get, Post, Put, Delete (Adrián Peña, 2016).

TypeScript es un lenguaje de programación que permite traducir instrucciones de un lenguaje a otro. TypeScript es conocido como un pre-compilador esto quiere decir que realiza el trabajo de un compilador JavaScript, sumada a las funciones de un traductor de instrucciones. Este lenguaje permite crear aplicaciones web robustas (Emmanuel Valverde Ramos, 2014).

JavaScript es un lenguaje de programación, permite interactuar entre la página web por medio de un navegador. Con JavaScript se pueden crear páginas web dinámicas añadiendo efectos, para elaborar un programa se lo puede realizar únicamente en el código JavaScript o creando un programa a partir de una página estructurada en HTML. Este lenguaje es orientado a desarrolladores Web (Mohedano, Saiz, & Salazar Román, 2012) .

Angular CLI es un componente esencial para el funcionamiento de Angular, es el intérprete de línea de comando que permite dar inicio a proyectos, dar la estructura básica del mismo con sus diferentes componentes (Mikel Eizagirre, 2016). Presenta una amplia documentación para el desarrollo de aplicaciones (Powered by Google, 2016).

La empresa INDUXION fue creada en el año 2015 por el Sr. Eduardo Núñez y el Sr. Pablo Chuquin. El principal campo, es la venta de cocinas de inducción y ollas para las mismas cocinas, luego de un año esta empresa fue creciendo y adquiriendo más mercadería, esto ampliando el mercado a productos para el hogar, la ubicación de la empresa INDUXION se encuentra en Ibarra en la dirección Av. Mariano Acosta y Víctor Gómez Jurado - La Plaza Shopping.

El principal problema es no poseer ningún sistema de inventarios para la automatización de la gestión y control de mercadería. anterior y así mostrar la información real del inventario con sus existencias.

### **Situación Actual**

Actualmente en la empresa INDUXION no existe ningún sistema para la gestión y control de inventarios, por tal motivo causa demora en la respectiva verificación de existencias de productos en documentos físicos, esto provoca inconformidad entre los vendedores y dueños de la empresa al momento de entregar una información oportuna.

El proceso que realiza los empleados de la empresa consiste en registrar toda información de ventas realizadas en el día, usando hojas provisionales con el número de ítems vendidos, al finalizar el día las anotaciones de salida de mercadería que fueron utilizadas son requeridas para el respectivo cruce de información con el día anterior y así mostrar la información real del inventario con sus existencias.

### **Prospectiva**

Con este sistema se establecerá un soporte informático administrable, frente a documentos físicos que por el tiempo se pueden dañar, deteriorar, presentar inconsistencias al momento de redactar informes y causar la pérdida total o parcial de los mismos, el sistema contara con el módulo encargado de la gestión y control de inventarios.

Para el desarrollo de este sistema se implementará de herramientas de software libre como es el Framework Angular y Yii2, esto permitirá facilitar la gestión de la información por parte de los miembros de la empresa INDUXION. A un futuro se podrá incorporar módulos como: reserva de pedidos, facturación electrónica, seguimiento de envíos de productos a clientes. anterior y así mostrar la información real del inventario con sus existencias.

## Planteamiento del Problema

¿Por qué existe una deficiente administración de inventarios entre vendedores y dueños de la empresa INDUXION?

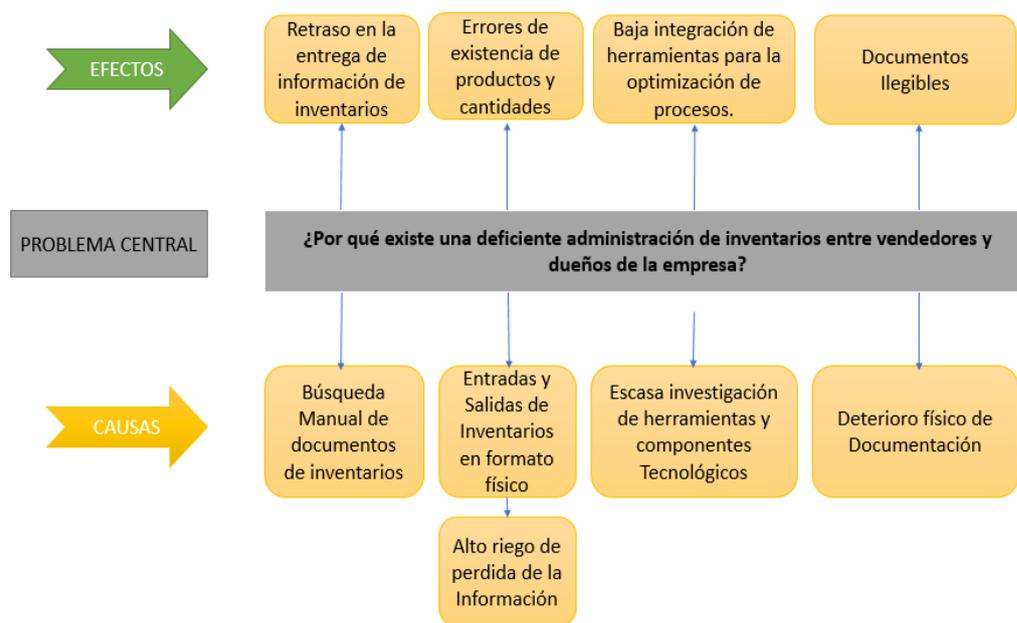


Fig. 1. Planteamiento del problema

Fuente: Propia

## Objetivos

### Objetivo General

Automatizar la gestión y control de inventarios para mejorar la productividad en la empresa INDUXION aplicando la integración de los Frameworks Angular 4 y Yii2 orientado a servicios REST.

### Objetivos Específicos

- Recopilar información acerca del proceso que se lleva acabo para el control de inventarios y presentación de informes de existencias de productos en la empresa "INDUXION".

- Realizar una investigación sobre el Framework Angular y Yii2.
- Realizar una investigación sobre la integración de servicios REST con Angular.
- Medir el impacto de la productividad en la Empresa.
- Implementar el sistema de inventarios en la Empresa.

## Alcance

La finalidad del proyecto consiste en realizar un estudio de la integración de dos Frameworks: Angular y Yii2 orientado a servicios REST, para el desarrollo de un sistema que permita la automatización, gestión y control de inventarios para la mejora de la productividad en la empresa INDUXION.

Para el estudio y desarrollo del sistema es necesario el uso de herramientas de software libre, con lenguajes de programación PHP y TypeScript. La principal ventaja para la implementación de esta tecnología es lograr una máxima velocidad de respuesta en el sistema y el manejo de las prestaciones de los servicios del lado del servidor.

Los módulos para implementar en el sistema son los siguientes:

- Roles de Usuarios: Ingreso al sistema por medio de permisos asignados a cada usuario.
- Registro de Personal: Detalles correspondientes a personas que trabajan en la empresa y determinado rol.
- Sucursales: Ubicación de tiendas.
- Marca de Productos: Nombre de las marcas de productos ofrecidos.
- Catálogo de Productos: Detalle toda la información de los productos existentes.
- Stock: Número determinado de existencias.
- Kardex: Presenta Entradas y Salidas de productos.
- Reportes: Información de productos por rango de fechas.

## Arquitectura de la Aplicación

Para el desarrollo de la aplicación es necesario realizar la siguiente arquitectura como se muestra en la Fig. 2.



Fig. 2. Arquitectura de la aplicación

Fuente: Propia

## Justificación

En el estudio surge la necesidad de dar a conocer dos Framework's a los desarrolladores y exponer sobre la tecnología, potencial y diferentes utilidades. Se realizará el estudio de los componentes y características necesarias de cada uno, y la integración de Frameworks Angular y Yii2 orientado a servicios REST para el desarrollo del sistema. Esto permitirá la optimización de procesos y agilidad en el sistema al momento de brindar una respuesta oportuna de la información requerida, por los usuarios.

Correspondiente a las tendencias a desarrollo orientado a servicios, la principal ventaja es que potencian la conectividad de diferentes frameworks, dispositivos, tecnologías lenguajes de programación, mediante archivos "xml, json", o protocolos simples "http Get, Post".

La metodología para implementar en el estudio y desarrollo es SCRUM, porque permite el desarrollo de software de forma ágil y flexible, los principales beneficios de la metodología SCRUM, es la adaptación a cambios requeridos, reducción de tiempos, reducción de riesgos, esto permite llevar a cabo funcionalidades de mayor valor como primer eje del proyecto como se detalla en la Fig. 3.

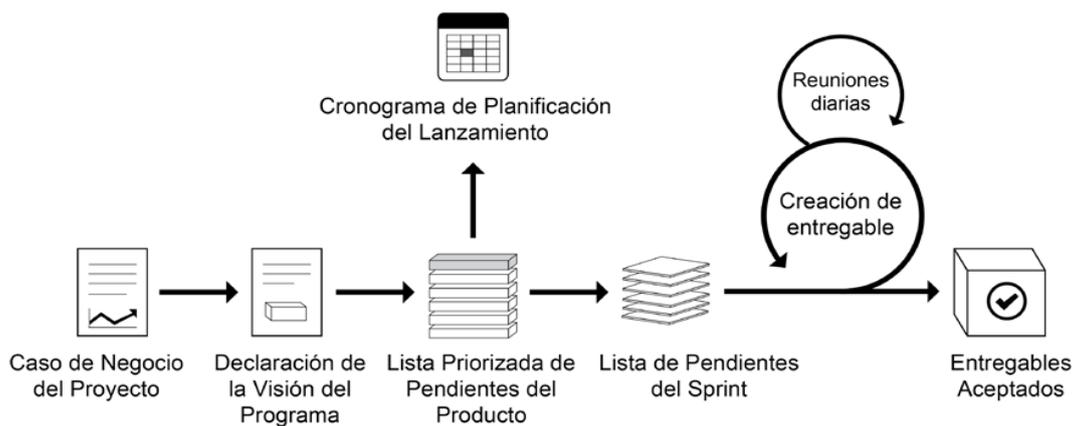


Fig. 3. Flujo de trabajo de la metodología SCRUM

Fuente: (Satpathy, 2016)



# CAPÍTULO 1

## 1. Marco Teórico

### 1.1. Empresa INDUXION

La empresa INDUXION fue creada en el año 2015 por el Sr. Eduardo Núñez y el Sr. Pablo Chuquín por la necesidad de incorporarse en el mercado ecuatoriano. El principal campo de la empresa es la venta de cocinas y ollas de inducción, luego de un año la empresa fue creciendo, adquiriendo más mercadería y estableciendo una mayor variedad de productos, esto ampliando el mercado a productos para el hogar actualmente se encuentra ubicada en Ibarra en la dirección Av. Mariano Acosta y Víctor Gómez Jurado - La Plaza Shopping, Ibarra – Ecuador.



Fig. 4. Empresa INDUXION

Fuente: Propia

#### 1.1.1. Situación Actual

Actualmente en la empresa INDUXION no existe ningún sistema informático que permita la gestión y control de inventarios, esto provoca un problema al momento de verificar el stock disponible de productos.

### 1.1.2. Proyección del sistema de gestión y control de inventarios

El sistema permitirá registrar todos los documentos correspondientes a entradas y salidas de productos, de esta manera se optimizarán los procesos de la empresa y se facilitarán la búsqueda de productos existentes con sus respectivas cantidades. Entre los principales beneficios tenemos:

- Registro de entradas y salidas de productos.
- Información de existencias de productos actualizada.
- Optimización de procesos.
- Documentos legibles.

### 1.1.3. Procesos manuales en la empresa INDUXION

Ingreso de mercadería: Es el proceso de abastecimiento de productos, esto lo realiza los dueños de la empresa al momento de realizar sus compras a los proveedores.

Atención al cliente: Se detalla toda la información correspondiente a los productos, dando a conocer las características y precios.

Salida de productos: Para este proceso es necesario tener un listado actualizado de productos, esto permitirá a los vendedores ver las existencias y ofrecer productos a los clientes. Una vez seleccionado el producto se llena una nota de venta para el respectivo desglose de productos, se establece el método de pago, y se realiza el cobro, el proceso se lo realiza de forma manual. Una vez realizada una venta se procede a guardar la información en un papel de anotaciones por parte de gente de ventas como se detalla en la Tabla 1.

Tabla 1. Registro de anotaciones diarias para reporte de Ventas

Código	Nombre Producto	Día /Mes / Año	Cantidad	Precio
UM101	Dispensador de Agua	13 – 12 -2017	1	35.00

Fuente: Propia

Fin de la jornada: Se verifican las ventas realizadas en el día y se procede a cuadra las existencias de productos para el siguiente día en el inventario físico.

Stock de productos: En esta sección se encuentra todo el inventario de productos, detallado con entradas y salidas detalladas en la Fig. 5.

**INVENTARIO PUNTO DE VENTA 2017**

	RESPON.	FECHA RESPON.	FECHA						
		22/05/17		23/05/17		24/05/17		25/05/17	26/05/17

CODIGO	PRECIO	SALDO	ING.	VENTA									
<b>UMCO</b>													
UM02	JUEGO CALDEROS ACERO INOX. UTIL. 16-20-24	27,50		2		2							
UM03	SET UMCOINOX 6 PIEZAS	37,00											
UM04	SET UMCOINOX 7 PIEZAS	44,00											
UM05	BATERIA COCINA AC. INOX. 7 PIEZAS GOURMET	43,50											
UM06	SET BRILLIANCE 8 PIEZAS INDUCCION	52,00											

Fig. 5. Inventario empresa

Fuente: Propia

Observación: Todos los procesos anteriormente citados la empresa lo realiza de forma manual, por este motivo es necesario la implementación de un sistema informático que permita la gestión y control de inventarios.

### 1.2. Base de datos Relacional “SQL”

Es una base de datos que permite crear relaciones entre una o varias tablas que contenga información de forma ordenada. Este tipo de base de datos es comúnmente utilizado para resolver problemas habituales que presenta una persona, empresa u organización. Las tablas para estas bases de datos deben presentar una estructura de cadena de atributos para su perfecta planificación, gestión y control (Ester Chicano Tejada, 2013).

El principal objetivo es resolver la redundancia de los datos e ingresar toda la información de forma ordenada, entre las principales características tenemos las siguientes (Pablo Valderrey Sanz, 2014).

- Consultas complejas entre varias tablas.
- Manejo fácil de datos.
- Interacción de tablas.

Para este tipo de base de datos es debe cumplir condiciones para el buen funcionamiento de la misma (Pablo Valderrey Sanz, 2014).

- Los registros de una tabla deben tener el mismo número de campos ingresados.
- Una base de datos debe estar formada por varias tablas.
- Cada tabla debe tener su respectivo nombre.
- Una tabla no debe tener campos repetidos.
- Los campos de registro de una tabla no poseen un orden establecido.
- Creación de nuevas tablas en base a tablas ya existentes.
- Cada campo de una tabla está definido por rangos.

### **1.2.1. Base de datos MariaDB**

MariaDB es un gestor de base de datos relacional SQL, entre las principales características tenemos la mejora en la velocidad de almacenamiento de datos, mejora la velocidad en las consultas complejas y soporte con bases de datos MySQL. Presenta un múltiple soporte correspondiente a documentación en páginas oficiales y comunidad libre (MariaDB, 2017).

### **1.3. HTML 5**

HTML conocido como Lenguaje de marcas de hipertexto, en inglés (Hyper Text Markup Language), es un lenguaje que permite estructurar una página web con contenido para ser interpretadas por un navegador web. HTML permite publicar contenido como es texto, fotos, listas, interfaces para formularios, permite abrir nuevos enlaces de hipertexto. El SGML (Standard Generalized Markup Language) es un estándar con el que se creó HTML, esto permite organizar las etiquetas o marcas con sus respectivos contenidos de los documentos (Javier Zofío Jiménez, 2013).

HTML5 es una nueva versión que propone nuevas mejoras para el desarrollo web, como son nuevas etiquetas para la estructura de documentos HTML, nuevos elementos multimedia como son audio y video que permiten explotar su potencial.

### **1.4. CSS 3**

Es un lenguaje que propone separa la estructura de una página web para establecer su diseño y formato creado en HTML. Su principal función es mejora la experiencia de uso y es esencial para el diseño gráfico de una página web agregando colores, bordes, sobras paralelas y creación de objetos interactivos. (José María Tomás Zafra, 2014).

CSS3 mejora el control sobre los estilos propuestos en una página web sin recurrir a medios de terceros que no son recomendables y complique la visualización de navegadores web por no cumplir estándares (José María Tomás Zafra, 2014).

### **1.5. PHP**

PHP es un lenguaje de programación de código abierto, adecuado para el desarrollo web y que puede ser incrustado en HTML, lo que significa que se puede combinar el código PHP, es muy utilizado para generar páginas web dinámicas. Lo que distingue a PHP es que el código es ejecutado en el servidor y que por medio de una página en HTML envía respuestas de forma visual al cliente, (The PHP Group, 2018).

El campo para el lenguaje PHP es el desarrollo web, el principal propósito es implementar soluciones web simples, eficientes y rápidas. Las principales características son:

- Es de código abierto, no requiere de licencias para su uso.
- Multiplataforma, funciona en cualquier sistema operativo.
- Amplio soporte en documentación de forma oficial o por medio de la comunidad.
- Soporte para servidores remotos o servidores local como son las herramientas libres: XAMPP, WAMP que simulan una máquina virtual para la ejecución de ficheros PHP.

### **1.6. Lenguaje JavaScript**

Es un lenguaje de programación para la creación de páginas web, JavaScript proporciona una capacidad de interacción con los usuarios, es decir permite a los usuarios navegar por nuevos enlaces disponibles, añade acciones como una respuesta a las peticiones de usuario que realice con el servidor, Javascript es un lenguaje interpretado, todas las instrucciones son analizadas por el navegador web para su interpretación de forma inmediata. Este lenguaje debe ser integrado a una página web para activar la ejecución del código por medio de un flujo esto quiere decir si un usuario da clic en un botón (Mohedano et al., 2012).

### **1.7. Lenguaje TypeScript**

Es un lenguaje de programación de código abierto creado por Microsoft, compuesto por un super-conjunto de JavaScript. Ofrece soporte para las diferentes funciones presentadas por JavaScript para el desarrollo de componentes sólidos y entrega de desarrollo de aplicaciones de óptimas condiciones y de alta confianza. TypeScript una vez compilado de forma limpia y simple JavaScript puede ser ejecutado desde cualquier navegador web (Microsoft, 2017).

TypeScript es un compilador encargado de traducir instrucciones de un lenguaje a otro también, llamado como un pre-compilador que su acción final será convertir JavaScript limpio y sin errores (Emmanuel Valverde Ramos, 2014).

### **1.8. Node.js**

Según la página oficial de Node.js refiere: "Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El

ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo” (Node.js Foundation, 2017).

Node.js añade un entorno de ejecución, especificando el lenguaje de programación, librerías y módulos de forma navita para el desarrollo de aplicaciones orientadas a la comunicación, esto permitirá el manejo de información de entrada/salida (Arturo Muñoz de la Torre Monzón, 2013).

### **1.8.1. NPM**

Es un administrador de paquetes para el lenguaje de programación JavaScript, la principal ventaja es compartir y reutilizar código que pueda facilitar el desarrollo de aplicaciones, permite verificar y actualizar paquetes en sus últimas versiones por parte de sus desarrolladores (Isaac Z. Schlueter, Voss, & Silverio, 2014).

Los paquetes también conocidos como módulos son directorios que almacenan varios activos que contienen metadatos, que permitirán la resolución de un problema a partir de bloques de construcción. Un beneficio a partir de la instalación de módulos de terceros es beneficiarse de la experiencia de desarrolladores que se han enfocado en una determinada área de trabajo.

### **1.9. JSON**

Es un formato para el envío y recepción de una determinada información, sus siglas en ingles significa “JavaScript Object Notation”, es un formato muy ligero que es utilizado para el intercambio de información, presenta dos estructuras: lista ordenada de valores y una agrupación de nombre/valor (Juan Manuel Vara Mesa, Marcos López Sanz, 2014).

### **1.10. Protocolo HTTP**

El protocolo de transferencia de hipertexto en inglés (Hypertext Trasfer Protocol), es un protocolo que permite compartir y propagar información de los sistemas mediante las páginas web, fue desarrollado por Sir Timothy Berners-Lee, en 1990 el protocolo HTTP se convirtió en la red global mundial conocida como la www. HTTP presenta un modelo cliente/servidor, esto quiere decir que el cliente es un navegador web y envía peticiones al servidor, el servidor recepta estas peticiones y devuelve una respuesta (Román Carceller Cheza, Carlos Campos Saborido, 2013).

### 1.11. Servicios Web REST

Transferencia de estado representacional, en inglés (Representational State Transfer), presenta una arquitectura basada en el protocolo Http en la que está presente el cliente/servidor, todos los recursos se encuentran de forma única por medio de URLS. La arquitectura REST presenta cuatro tipos de operaciones para compartir y manipular recursos (Antonio Luis Cardador Cabello, 2014).

- Read (Leer): El cliente envía una petición al servidor para leer una determinada información de un recurso.
- Create (Crear): El cliente envía una petición al servidor para la creación de un nuevo recurso en el sistema.
- Delete (Eliminar): El cliente envía una petición al servidor, esta acción permite eliminar un recurso del sistema o del servidor.
- Update (Actualizar): El cliente puede realizar una petición al servidor para modificar un recurso en el sistema.

Para realizar cualquier tipo de operaciones necesaria se ejecutarán por el protocolo HTTP que permitirá operaciones como: GET (leer y consultar), PUT (editar), POST (crear), DELETE (eliminar) (Antonio Luis Cardador Cabello, 2014).

Para considerar un servicio REST (Representational State Transfer), presenta un estilo de arquitectura para desarrollar servicios web que deben respetar las siguientes condiciones:

- **Cliente – Servidor:** Consiste en la separación de los dos agentes de intercambio de información como es el cliente y el servidor. Los dos agentes deben ser independientes, permiten una flexibilidad de comunicación y el uso de múltiples plataformas como se detalla en la Fig. 6.



Fig. 6. Servicios REST - Cliente – Servidor

Fuente: (Cecilio Álvarez Caules, 2013)

- **Sin estado:** Cada petición que recibe el servidor debe ser independiente, no mantiene un estado asociado al cliente como se indica en la Fig. 7.

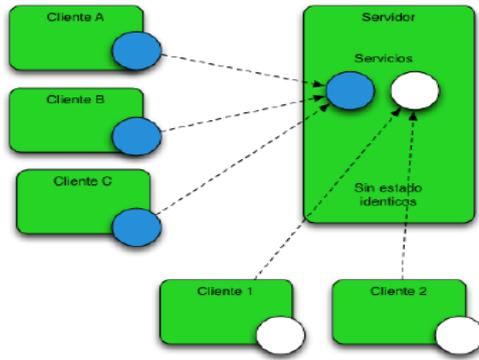


Fig. 7. Servicios REST - Sin estado

Fuente: (Cecilio Álvarez Caules, 2013)

- **Cache:** Si se realiza por primera vez una petición en los servicios REST, esta petición se puede almacenar en cache para apoyarse con futuras peticiones si fuera necesario, evitando repetir la misma petición al servidor como se indica en la Fig. 8.

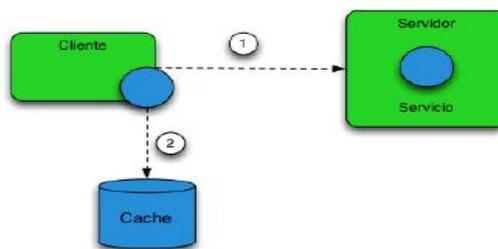


Fig. 8. Servicios REST - Cache

Fuente: (Cecilio Álvarez Caules, 2013)

- **Servicios Uniformes:** En la Fig. 9, en los servicios REST permite compartir una forma de invocación y métodos como:
  - GET: Obtener una información del servidor.
  - POST: Crear un nuevo registro en el servidor.
  - PUT: Editar un archivo o registro en el servidor.
  - DELETE: Borrar un determinado archivo o registro en el servidor.

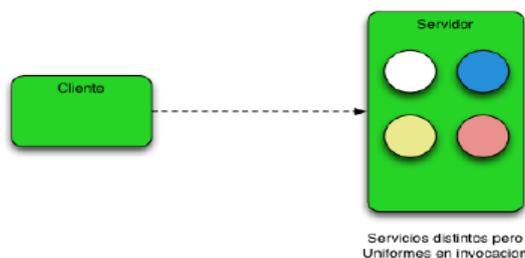


Fig. 9. Servicios REST – Servicios Uniformes

Fuente: (Cecilio Álvarez Caules, 2013)

- **Arquitectura en Capas:** Todos los servicios REST están compuestos por numerosas capas, en este proceso el cliente no puede saber si está relacionando con una capa final o intermedia, esta restricción facilita la escalabilidad y permite establecer políticas de seguridad (José Luis Berenguel Gómez, 2016). Un cliente REST no será capaz de distinguir entre si está realizando una petición directamente al servidor, o se lo está devolviendo un sistema de cache intermedio, un ejemplo puede ser cuando existe un balanceador que se encarga de redirigirlo a otro servidor como se detalla en la Fig. 10, (Cecilio Álvarez Caules, 2013).

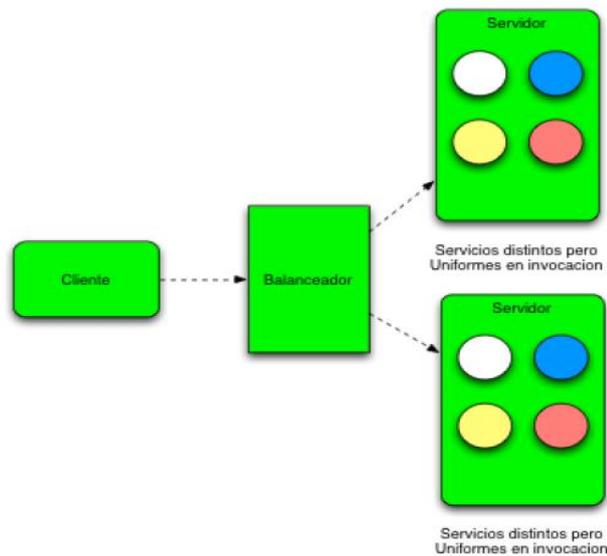


Fig. 10. Servicios REST – Arquitectura en capas

Fuente: (Cecilio Álvarez Caules, 2013)

Si un sistema cumple todas las restricciones citadas anteriormente se puede establecer que es un sistema RESTful. El patrón de diseño RESTful se utiliza en combinación con HTTP para ejecutar las cuatro operaciones de los métodos, GET, POST, PUT y DELETE.

### 1.11.1. RESTful API

API es la interfaz de programación de aplicaciones, en inglés (Application Programming Interface), permite utilizar recursos de software por otro, es decir permite consumir servicios en cualquier lenguaje de programación o plataforma. Un RESTful API permite consumir servicios REST para la obtención de información por medio del protocolo HTTP, ejecutando los principales métodos como GET, POST, PUT. DELETE, un componente primordial es el HTTP Status Code que significa el estado de la respuesta recibido por parte del cliente para representar un resultado (Federico Guzman, 2016).

## 1.12. Framework Angular

Angular es un Framework desarrollado por la empresa Google, que permite la creación de aplicaciones web, especialmente un cliente Front-End esto quiere decir que es un sitio web que interactúa con el cliente. Se encuentra estructurada por el lenguaje HTML, JavaScript y TypeScript. Es una plataforma para la creación de aplicaciones web orientado a proyectos de mediana a gran escala.

### 1.12.1. Arquitectura

En la Fig. 11, Angular presenta una arquitectura compuesta por módulos, que permita la estructura y que el código sea organizado por funcionalidad, puede ser reutilizable para el desarrollo de aplicaciones web.

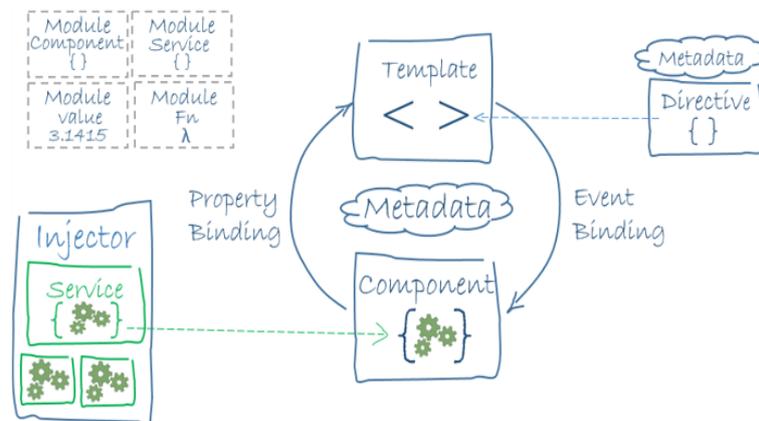


Fig. 11. Arquitectura del framework Angular

Fuente: (Powered by Google, 2017).

Para un mayor entendimiento de la arquitectura a continuación, se detalla cada uno:

- **NgModule:** Es un contenedor de código dedicado a un flujo de trabajo estrechamente relacionada, como son los servicios globales, componentes, y otros archivos necesarios para el correcto funcionamiento. Una aplicación generalmente presenta muchos módulos esto permite gestionar el desarrollo de aplicaciones y el diseño para la reutilización del código.
- **Component – Componentes:** Es el encargado de definir una clase que contiene datos y la lógica de la aplicación, se encuentra asociada a una plantilla HTML que se define como una vista para la visualización de la aplicación.
- **Directives – Directivas:** Es una clase con un decorador orientada a las plantillas HTML que se encuentran alojadas en los componentes de cada módulo.

- **Sevices – Servicios:** Se establece las funcionalidades o características que una aplicación necesita. Un servicio es la lógica de la aplicación.
- **Injector – Inyección de dependencias:** permite proveer la información de un servicio a un componente dándole acceso inmediato.
- **Router – Enrutamiento:** Permite la navegación de una vista seleccionada por el cliente, es el acceso URL a las páginas.
- **Metadata – Metadatos de Componentes:** Son los datos asociados a una plantilla Html con el componente, estos datos son mostrados por una vista.
- **Templates – Plantillas y Vistas:** Una plantilla formada por Html indica al framework como renderizar o visualizar el componente. La vista permite mostrar, modificar, ocultar secciones o páginas.

### 1.12.2. Arquitectura de una aplicación

Una aplicación desarrollada en angular se encuentra conformada de la siguiente manera:

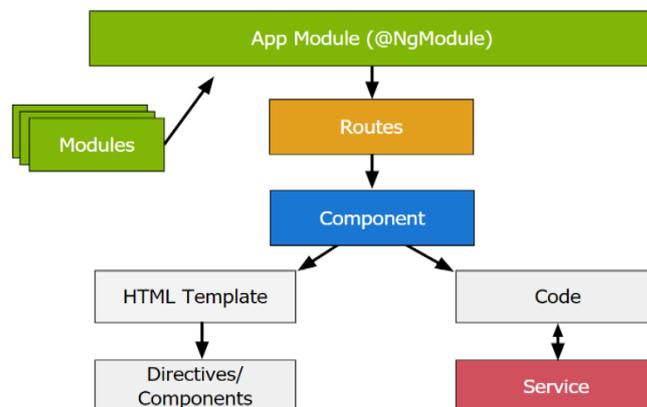


Fig. 12. Arquitectura de una aplicación en el Framework Angular

Fuente: Presentación

Una aplicación de angular debe poseer un contenedor de código de un flujo de datos como se puede observar en la Fig. 12, presenta rutas de acceso para los diferentes componentes de una aplicación que se encuentra estructurados por plantillas HTML, directivas y servicios que hacen posible la ejecución de un proyecto.

### 1.12.3. Angular CLI

Es un conjunto de instrucciones de comandos que permite interactuar con el Framework Angular, es el eje primordial en la creación y ejecución de proyectos para esta tecnología. En la Tabla 2, se presenta los siguientes comandos de interfaz utilizadas para el desarrollo de aplicaciones:

Tabla 2. Comandos para operaciones con Angular CLI

Nombre	Descripción	Código de Generación
Instalación	Línea de comando por instalación de dependencia Angular CLI	<code>npm install -g @angular/cli</code>
Generación de Proyecto	Línea de comando para generar un nuevo proyecto de Angular	<code>ng new PROJECT-NAME</code>
Prueba de Proyecto	Línea de comando para ejecutar el proyecto por medio de un servidor local interno. "http://localhost:4200/"	<code>ng serve</code>
	Component - Componentes: Presenta una lógica que permite soportar una vista dentro de una clase y que pueda interactuar por medio de una API.	<code>ng generate component my-new-component</code>
Generación de Componentes, Directivas, Pipes y Servicios	Directive - Directivas: Son condiciones necesarias para cada proyecto.	<code>ng g directive my-new-directive</code>
	Pipe: Transforma un dato de entrada a un dato de salida deseada. Por ejemplo: minúsculas a mayúsculas.	<code>ng g pipe my-new-pipecomponent</code>
	Service - Servicios: Se encarga de presentar y delegar datos de acceso que presentan los componentes, los servicios son necesarios para compartir información entre las clases de una aplicación.	<code>ng g service my-new-service</code>
	Class - Clase: especificación de datos por medio de una estructura.	<code>ng g class my-new-class</code>
	Interface - Interfaz: Permiten verificar datos recibidos y verificar que sigan una estructura.	<code>ng g interface my-new-interface</code>
Actualizar Proyectos	Module - Módulos: Se caracterizan por compartir el mismo inyector de dependencias, esto se refiere a que todos los servicios proporcionados de un módulo están disponibles.	<code>ng g module my-module</code>
	Línea de comando para actualizar las dependencias Angular CLI	<code>npm uninstall -g @angular/cli</code> <code>npm install -g @angular/cli@latest</code>

Fuente: (Powered by Google, 2016)

#### 1.12.4. Estructura de archivos del Framework

Para la estructura de archivos para el Framework Angular son las siguientes:

- Directorio e2e: contiene los archivos de configuración del Framework.
- Directorio node\_module: contiene las dependencias necesarias para el correcto funcionamiento.
- Directorio src: directorio para el desarrollo de los módulos de la aplicación detalladas a continuación:
  - Carpeta app: es la carpeta para la implementación de componentes.
  - Carpeta assets: es la carpeta que almacena los estilos y archivos adicionales.
  - Carpeta environments: en esta carpeta se encuentra las configuraciones y variables cuando un proyecto es ejecutado en modo producción.
  - Archivo favicon: es el icono del proyecto.
  - Archivo index.html: es el archivo inicial del proyecto.
  - Archivo main.ts: es un archivo TypeScript en donde se encuentran las configuraciones globales del proyecto.
- Archivo angular-cli.json: este archivo presenta la estructura y la descripción del proyecto. Se ejecutará con los comandos de Angular CLI.
- Archivo .editorconfig: configuración del texto y su codificación.
- Archivo .gitignore: parámetros para subir cambios a nuestro repositorio.
- Archivo karma.conf.js: configuraciones para pruebas unitarias
- Archivo package.json: descripción de dependencias del proyecto
- Archivo protactor.conf.js: es un marco de pruebas
- Archivo tsconfig.json: configuración para la compilación de archivos.
- Archivo tslint.json: características de compilación de archivos.

#### 1.12.5. Características

Entre las principales características ofrecidas por el Framework son las siguientes:

- Multiplataforma: Creación de aplicaciones: Web, Web Móvil, Dispositivos Móviles nativos, Aplicaciones de escritorio PC.
- Presenta una licencia de software libre llamada MIT, permite operar el software sin restricciones, incluidos, los derechos de uso, copia, modificación, publicar y distribuir (Powered by Google, 2017).
- Velocidad y Rendimiento: Máxima velocidad por medio de la plataforma web y representación de información del lado del servidor.

- Herramientas CLI: Uso de componentes y plantillas existentes.

### 1.13. Yii2 Framework

Yii2 es un Framework que se encuentra basado en el lenguaje de programación PHP, presenta componentes de alto desempeño para la creación de aplicaciones web modernas de una forma ágil. El significado de las iniciales en chino Yii es: “simple y evolutivo”. Es un Framework genérico orientado al desarrollo de cualquier tipo de aplicaciones web de gran escala (Yiiframework.com, 2016).

Yii2 se encarga de dar una estructura de código limpio y reutilizable en una aplicación, separa la lógica del servidor, la regla de negocio y por ultimo las interfaces de usuario. Presenta un alto soporte en herramientas y permite la automatización de tareas comunes al momento del desarrollo de aplicaciones web.

#### 1.13.1. Arquitectura

Yii2 está establecido con el patrón de diseño web como es la arquitectura MVC “Modelo – Vista – Controlador”, es elegido considerablemente en la programación web, el principal objetivo de la arquitectura es separar la lógica del negocio de la interfaz de usuario, para que los desarrolladores puedan realizar modificaciones en cada parte del proyecto de forma cómoda y sin afectar otra, se encuentra conformador por los siguientes niveles representados en la Fig. 13:

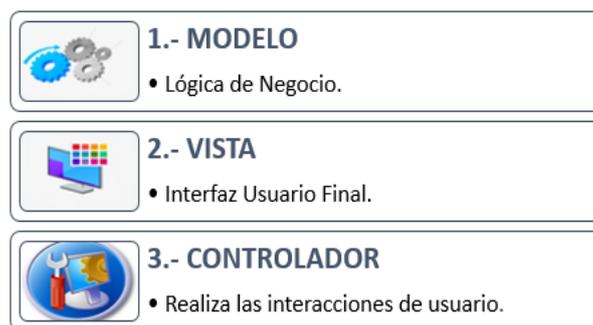


Fig. 13. Arquitectura MVC “Modelo – Vista – Controlador”

Fuente: (Yiiframework.com, 2016)

**Modelo:** Es conocida como la lógica de negocio, es la representación de la información del sistema, permite la gestión de accesos de la información por medio de consultas de actualizaciones que sean establecidas en la aplicación. Todas las peticiones de manipulación o actualización de la información son invocadas por el modelo (Vicente Javier Eslava Muñoz, 2013).

**Vista:** Presenta la información y la lógica del negocio, principalmente es una interface orientada al usuario final, esta interface permite realizar múltiples peticiones al sistema para recibir un resultado (Vicente Javier Eslava Muñoz, 2013).

**Controlador:** Responde las acciones del usuario común mente llamadas peticiones, estas peticiones se las realiza al modelo para la solicitud de información como actualizar o ingresar un nuevo registro a la base de datos, el controlador es conocido como un intermediario entre la vista y el modelo (Vicente Javier Eslava Muñoz, 2013).

### 1.13.2. Ciclo de Vida de una petición en Yii2.

En la Fig. 14, se detalla cuando el usuario final realiza una petición en el sistema, en Yii2 se realiza un flujo del proceso para obtener una respuesta apropiada a la solicitud requerida por parte del usuario.

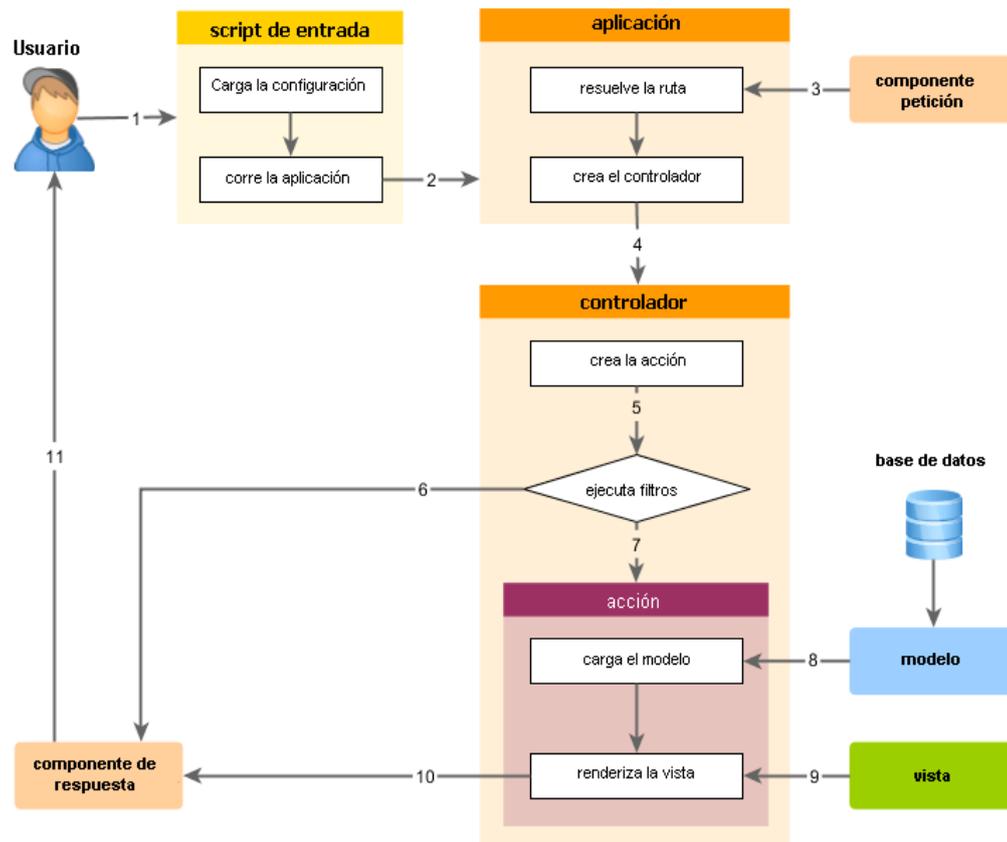


Fig. 14. Ciclo de vida de una petición en Yii2

Fuente: (Yiiframework.com, 2016)

- **Paso 1:** El usuario realiza una petición por medio de un script de entrada comúnmente es el archivo de inicio de la aplicación web llamado index.php
- **Paso 2:** Se crea una instancia para manejar la petición de la aplicación.

- **Paso 3:** La aplicación por medio del componente de petición resuelve la ruta con su respectivo controlador necesario.
- **Paso 4:** La aplicación por medio del controlador crea la acción a la petición solicitada.
- **Paso 5:** Se ejecuta los filtros necesarios para dicha acción.
- **Paso 6:** Si el filtro presenta una falla en la acción esta es cancelada de forma inmediata.
- **Paso 7:** Si el filtro es correcto la acción se puede ejecutar.
- **Paso 8:** La acción carga un modelo que presenta una conexión con la base de datos.
- **Paso 9:** La acción es retornada por medio de una vista con su respectivo modelo.
- **Paso 10:** El resultado obtenido de la acción es retornado al componente de la aplicación de respuesta.
- **Paso 11:** El componente de respuesta envía los resultados al usuario final por medio del navegador web.

### 1.13.3. Estructura de archivos del Framework.

La estructura más importante de archivos del Framework Yii2 son las siguientes:

- Directorio assets: se almacena la configuración que hace referencia a una página web como es los estilos CSS.
- Directorio commands: contiene las clases de comandos de consola.
- Directorio config: archivos para la configuración del proyecto y conexión a la base de datos.
- Directorio controllers: almacenan las clases que son responsables de procesar las solicitudes y generar respuestas.
- Directorio filters: se almacenan los objetos que se ejecutan antes y / o después de las acciones del controlador.
- Directorio migrations: almacena la estructura de las tablas para una base de datos.
- Directorio models: se almacenan las clases que representan datos de negocio, reglas y lógica.
- Directorio modules: se almacenan los modelos, vistas, controladores y otros componentes de soporte.
- Directorio runtime: contiene archivos generados por Yii2 en tiempo de ejecución, como archivos de log y cache.
- Directorio vendor: contiene los paquetes y librerías instalados por Composer, incluyendo el propio núcleo de Yii2.
- Directorio views: almacena los archivos de vistas (templates).

- Directorio web: contiene los archivos accesibles vía web.
- Archivo composer.json: es un archivo utilizado por Composer, detalla todas las librerías y paquetes.
- Archivo yii: script para la ejecución de comandos por medio de la consola.

#### **1.13.4. Características**

Las principales características que distinguen al Framework Yii2 son las siguiente:

- Presenta una licencia BSD, para el desarrollo de aplicaciones web de código abierto, que permite utilizar libremente para fines comerciales en obras derivadas, modificar, distribuir, otorgar derechos parciales de producción / distribución a terceros no incluidos en la licencia (Yiiframework.com, 2016).
- Es Independiente de gestores de base de datos.
- Presenta un múltiple soporte para el desarrollo de API – REST, esto quiere decir que es un conjunto de reglas que las aplicaciones necesitan seguir para la comunicación entre ellas.
- Iteración con base de datos relacionales y no relacionales.
- Preparado para proyectos de gran escala como: Proyectos Empresariales, Sistemas de gestión de contenidos, comercio electrónico, servicios Web REST.
- Amplia configuración de reglas y políticas para cada empresa
- Lectura de Código de forma entendible, integración de complementos y librerías por medio de la web oficial y terceros.

#### **1.14. Integración de Servicios REST entre los Frameworks Angular y Yii2**

Para el respectivo estudio de integración, es necesario realizar la instalación de las herramientas adecuadas que permitan el correcto funcionamiento entre los Frameworks seleccionados, para esto es necesario revisar los procedimientos de instalación de las herramientas adjuntos en (Anexos A). El estudio presenta cuatro partes detalladas a continuación.

- Herramientas para el estudio de integración de servicios REST.
- Arquitectura de Integración de servicios REST Frameworks Yii2 – Angular.
- Desarrollo de un API REST con el Framework Yii2.
- Consumo de servicios REST con el Framework Angular.

### 1.14.1. Herramientas para el estudio de integración de servicios REST

Para el correcto funcionamiento en el desarrollo del estudio y del sistema es necesario dar a conocer las herramientas utilizadas.

- **Visual Studio Code:** Es un editor de código fuente, presenta un soporte integrado para TypeScript, Node.js y presenta soporte para lenguajes de programación por medio de extensiones como es: PHP, Java, Python etc.
- **Postman – Developing APIs:** Postman es un cliente que permite realizar pruebas de servicios web por medio de los métodos Http: Get, Post, Put, Delete y así comprobar el correcto funcionamiento de nuestra aplicación en construcción.
- **XAMPP:** Es un servidor independiente de forma local, es una distribución de apache gratuita, presenta servidores de base de datos como es MariaDB.

Añade el intérprete del lenguaje de programación PHP, permite realizar proyectos web y probarlos en nuestro equipo sin necesidad de internet.

- **Composer:** Es un manejador de dependencias PHP que permite descargar librerías necesarias para cada proyecto de desarrollo.
- **Node.js:** Servidor para el entorno de ejecución para JavaScript.
- **Framework Yii2:** Framework PHP para la construcción del servicio REST.
- **Framework Angular:** Framework encargado de la visualización del cliente final, su lenguaje de programación es TypeScript.

### 1.14.2. Arquitectura de Integración de servicios REST Frameworks Yii2 – Angular

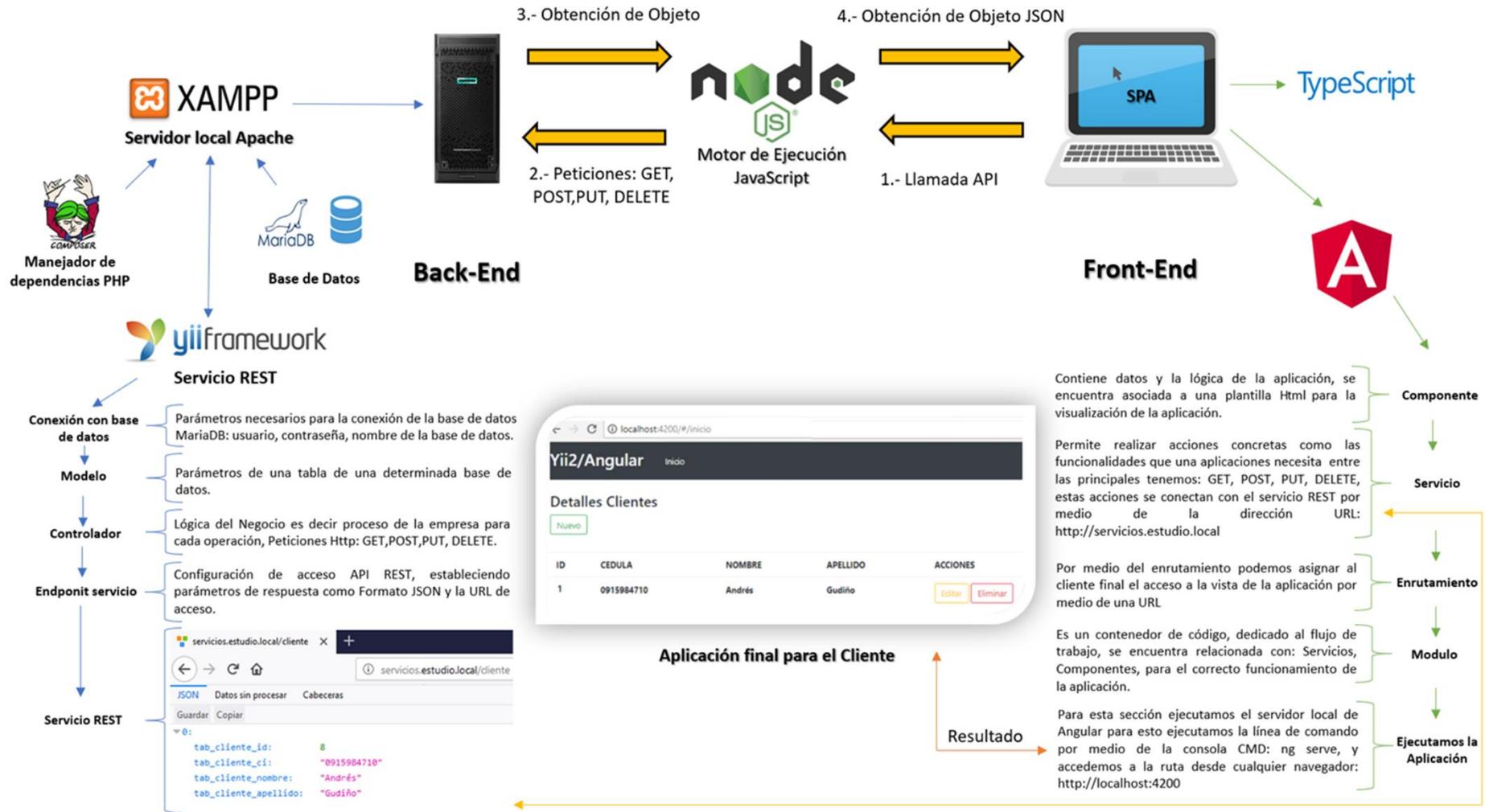


Fig. 15. Arquitectura de integración de servicios REST entre Yii2 – Angular

Fuente: Propia

### 1.14.3. Desarrollo de un API REST con el Framework Yii2

Yii2 Framework presenta el lenguaje de programación PHP, proporciona un conjunto de herramientas para la implementación de un API de servicios Web REST y permite el correcto funcionamiento del enrutamiento para las peticiones Http como son: GET, POST, PUT, DELETE, gracias a la negociación de respuestas compatibles como es el formato de archivos JSON.

Para crear un servicio web REST es necesario tener una base de datos creada en MariaDB para el registro de datos. Lo primero a realizar es la instalación de XAMPP Server y Yii2 Framework y la configuración del servidor local virtual, los pasos de la instalación están presentes en (Anexos A: Instalaciones, Anexo B: Configuración de un host Virtual).

Realizada la instalación de Yii2 Framework en nuestro servidor local en la siguiente dirección: C:\xampp\htdocs\ estudio\_yii2angular4\servicios, se establecer la conexión a la base de datos.

Para este proceso debemos acceder a la carpeta del proyecto en la siguiente ruta, config\db.php y agregar los parámetros de conexión esto permitirán la comunicación entre el Framework y la base de datos. Los parámetros de configuración para la comunicación son detallados en la Tabla 3.

Tabla 3. Parámetros de conexión a la base de datos desde el Framework Yii2

Parámetros de conexión a la Base de Datos MariaDB		
'class'	'yii\db\Connection'	Clase con el controlador para Mysql, preestablecido por Yii2.
'dsn'	'mysql:host=localhost;dbname=yii2_angular4',	Aplicación de Forma local y se especifica el nombre de la base de datos.
'username'	'root',	Nombre de usuario para autenticación en base de datos MariaDB.
'password'	'',	Contraseña de usuario para autenticación en base de datos MariaDB.
'charset'	'utf8'	Formato de la base de datos.

Fuente: Propia

En el editor de código Visual Studio Code Fig. 16, ingresamos a la ruta antes mencionada para asignar los parámetros de conexión a la base de datos.

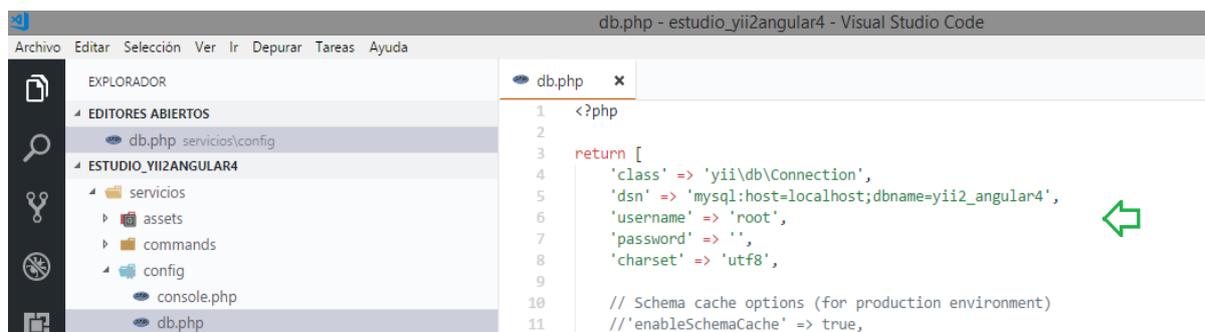


Fig. 16. Configuración de parámetros de conexión en Visual Studio Code

Fuente: Propio

Realizada la configuración de la conexión entre el Framework y la Base de datos MariaDB, se procede a crear las migraciones con los parámetros de las tablas a trabajar. Las migraciones, son esquemas de la base de datos: nombre de la tabla, definición de la columna y tipo de dato.

La operación se realizar de forma manual a partir de una base de datos o automática. Para crear una migración de forma instantánea es necesario establecer los parámetros detallados a continuación en la Tabla 4.

Tabla 4. Parámetros para crear migraciones en Yii2

<code>yii migrate/create &lt;name&gt;</code>	<name> es el argumento requerido para una descripción de la nueva migración.
<code>yii migrate/create create_news_table</code>	Especificación de los detalles de la migración a crear.
<code>yii migrate/create create_post_table --fields="title:string,body:text"</code>	Para crear campos de tabla de inmediato, especifíquelos mediante --fields.

#### Creación de migración - Cliente

```

yii migrate/create create_tab_cliente_table --
fields="tab_cliente_id:primaryKey,tab_cliente_ci:string(10):notNull:unique,tab_cliente_nombre:string(30):notNull,tab_cliente_apellido:string(30):notNull"

```

Fuente: Propia.

Para crear la migración de la tabla cliente en el Framework Yii2 ingresamos al terminar integrado de Visual Studio Code en la siguiente ruta del proyecto, C:\xampp\htdocs\estudio\_yii2angular4, se procede a ejecutar la siguiente línea de comando como se muestra en la Fig. 17.

- yii migrate/create create\_tab\_cliente\_table -  
fields="tab\_cliente\_id:primaryKey,tab\_cliente\_ci:string(10):notNull:unique,tab\_cliente\_nombre:string(30):notNull,tab\_cliente\_apellido:string(30):notNull"

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
1: cmd
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

c:\xampp\htdocs\estudio_yii2angular4\servicios>yii migrate/create create_tab_cliente_table --fields="tab_cliente_id:primaryKey,t
ab_cliente_ci:string(10):notNull:unique,tab_cliente_nombre:string(30):notNull,tab_cliente_apellido:string(30):notNull"

```

Fig. 17. Migración de tablas en Yii2

Fuente: Propia

Creada la migración en el proyecto, procedemos a verificar el archivo con todos los parámetros asignados como se indica en la Fig. 18.

```

m171120_213952_create_tab_cliente_table.php - estudio_yii2angular4 - Visual Studio Code
Archivo Editar Selección Ver Ir Depurar Tareas Ayuda
EXPLORADOR
EDITORES ABIERTOS
m171120_213952_create_tab_cliente_table.php servicios...
ESTUDIO_YII2ANGULAR4
servicios
assets
commands
config
controllers
mail
migrations
m171120_213952_create_tab_cliente_table.php
models
runtime
tests
vendor
views
web
widgets
.bowerrc
.gitignore
codeception.yml
composer.json
composer.lock
LICENSE.md
README.md
requirements.php
yii

1 <?php
2
3 use yii\db\Migration;
4
5 /**
6  * Handles the creation of table `tab_cliente`.
7  */
8 class m171120_213952_create_tab_cliente_table extends Migration
9 {
10     /**
11      * @inheritdoc
12      */
13     public function up()
14     {
15         $this->createTable('tab_cliente', [
16             'tab_cliente_id' => $this->primaryKey(),
17             'tab_cliente_ci' => $this->string(10)->notNull()->unique(),
18             'tab_cliente_nombre' => $this->string(30)->notNull(),
19             'tab_cliente_apellido' => $this->string(30)->notNull(),
20         ]);
21     }
22
23     /**
24      * @inheritdoc
25      */
26     public function down()
27     {
28         $this->dropTable('tab_cliente');
29     }
30 }
31

```

Fig. 18. Verificación Migraciones en Yii2

Fuente: Propia

El siguiente paso es cargar la migración de la tabla cliente en la base de datos MariaDB, es necesario ingresar la siguiente línea de comando yii migrate/up en el terminal integrado de Visual Studio Code: c:\xampp\htdocs\estudio\_yii2angular4\servicios>yii migrate/up, como se indica en la Fig. 19 y Fig. 20.

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
2: cmd
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

c:\xampp\htdocs\estudio_yii2angular4\servicios>yii migrate/up

```

Fig. 19. Línea de comando para migración a la base de datos MariaDB

Fuente: Propia

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
2: cmd
m171120_213952_create_tab_cliente_table
Apply the above migration? (yes|no) [no]:yes
*** applying m171120_213952_create_tab_cliente_table
> create table tab_cliente ... done (time: 0.032s)
*** applied m171120_213952_create_tab_cliente_table (time: 0.055s)

1 migration was applied.
Migrated up successfully.

```

Fig. 20. Creación de migración a la base de datos MariaDB

Fuente: Propia

Verificamos la creación de la base de datos, con todos los parámetros asignados como se puede observar en la Fig. 21.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	tab_cliente_id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	tab_cliente_ci	varchar(10)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
3	tab_cliente_nombre	varchar(30)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
4	tab_cliente_apellido	varchar(30)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

Fig. 21. Base de datos tab\_cliente en servidor MariaDB

Fuente: Propia

En el proyecto es necesario crear un modelo a nuestra tabla a partir de la base de datos MariaDB, para esto generaremos un modelo por parte de la herramienta de Yii2 para la generación de código, ingresamos a la siguiente dirección: <http://servicios.estudio.local/gii> y generaremos el modelo como se indica en la Fig. 22.

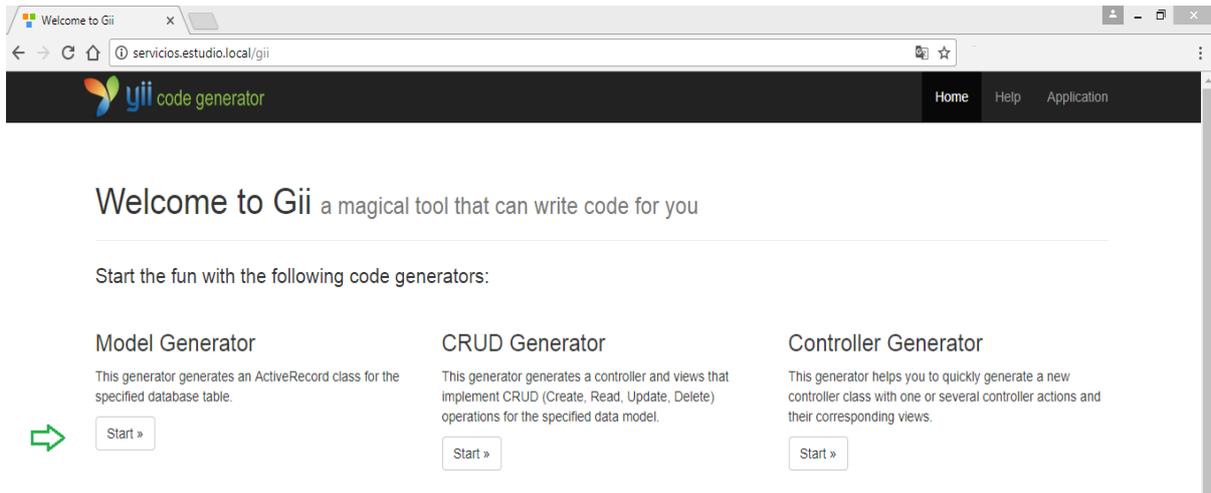


Fig. 22. Generador de código en Yii2, generador de modelo

Fuente: Propia

En la Fig. 23, Asignamos los datos correspondientes al nombre de la tabla en la base de datos y nombre del modelo a generar el respectivo código en el proyecto Yii2.

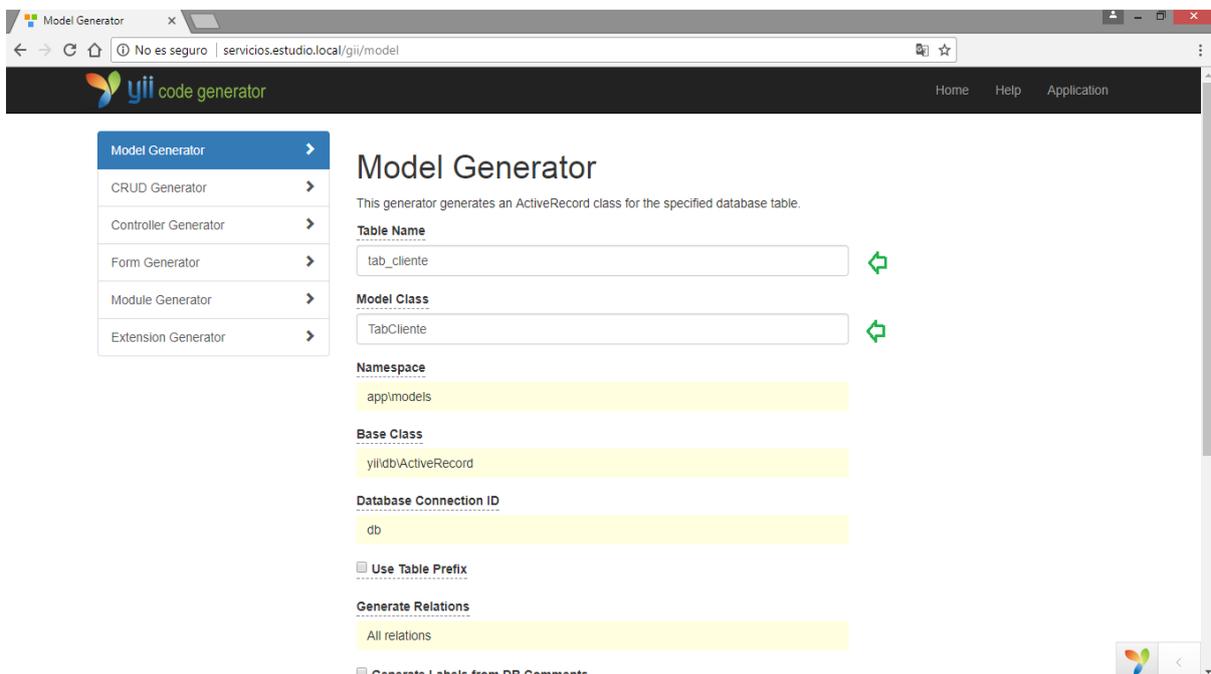


Fig. 23. Generador de nuevo modelo en Yii2.

Fuente: Propia

Dar clic en “Preview”, y para finalizar clic en Generar “Generate” y verificamos el modelo generado como se indica en la Fig. 24.

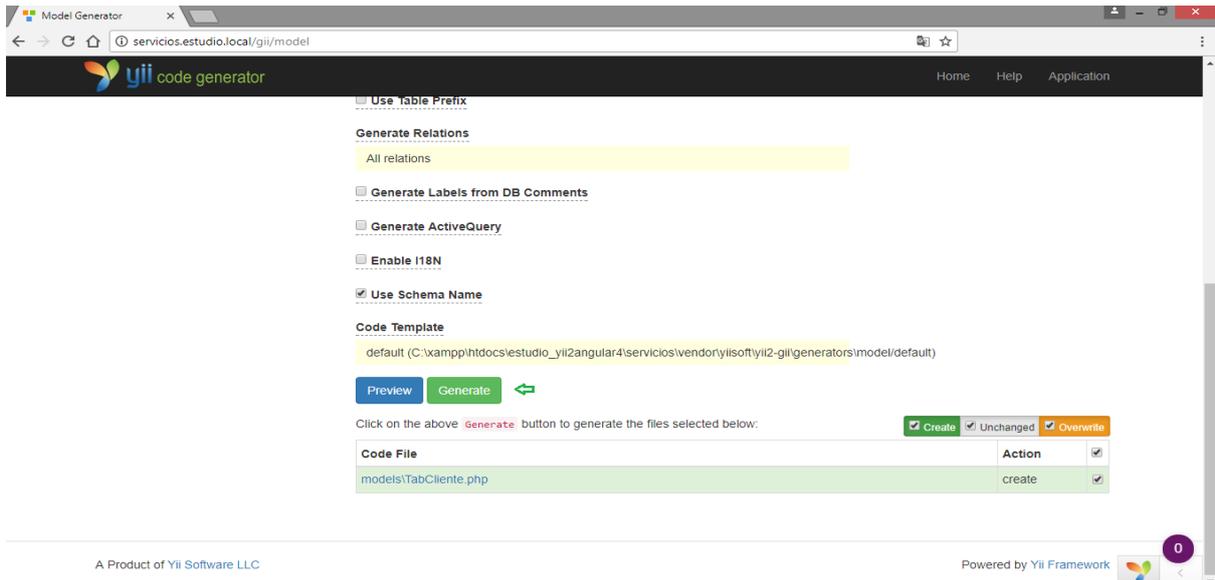


Fig. 24. Generador del nuevo modelo Yii2

Fuente: Propia

En la Fig. 25, se procede a la revisión del modelo creado por parte de Yii2 Framework.

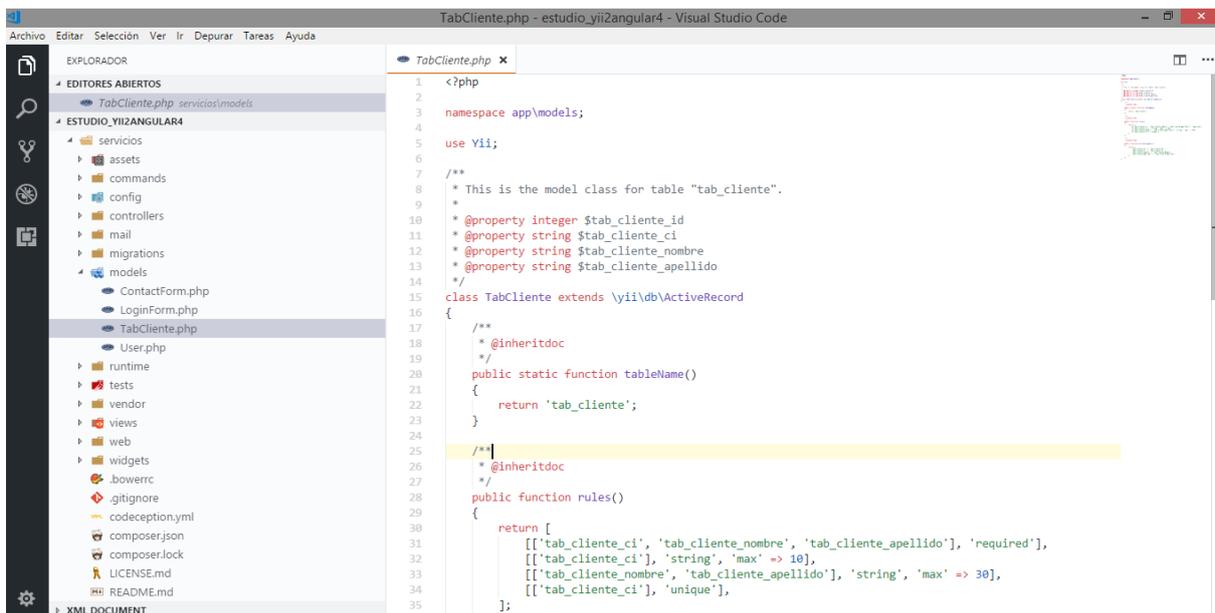


Fig. 25. Verificar modelo creado en el proyecto de Yii2 Framework

Fuente: Propia

Creado el modelo por parte del generador de código del Framework Yii2, se debe crear un controlador que permita interactuar las acciones REST, con los modelos creados a partir de la base de datos MariaDB, el beneficio de este proceso es realizar llamadas a la base de datos para obtener todos los datos de la tabla en formato JSON.

Creamos un nuevo archivo en la siguiente dirección: C:\xampp\htdocs\estudio\_yii2angular4\servicios\controllers\ClienteController.php y configuramos el archivo con las siguientes instrucciones como se muestra en la Fig. 26.

```
<?php
namespace app\controllers;

use yii\rest\ActiveController;
use yii\web\Response;

class ClienteController extends ActiveController
{

public $modelClass = 'app\models\TabCliente';
public function behaviors()
{

$behaviors = parent::behaviors();

// Habilita el acceso cuando se trabaja en dominios diferentes

$behaviors['corsFilter'] = [
'class' => \yii\filters\Cors::className(),
'cors' => [
'Origin' => ['*'],
'Access-Control-Request-Method' => ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS'],
'Access-Control-Request-Headers' => ['*'],
],
];

// Negociador, formato de respuesta, lenguaje de la aplicación
$behaviors['contentNegotiator'] = [
'class' => 'yii\filters\ContentNegotiator',
'formats' => [
'text/html' => Response::FORMAT_JSON,
'application/json' => Response::FORMAT_JSON,
'application/xml' => Response::FORMAT_XML,
],
];
return $behaviors;
}}
```

Fig. 26. Script de configuración ClienteController.php

Fuente: Propia

Es necesario establecer las reglas de Urls amigables y realizar la configuración del urlManager, para el acceso REST a clientes.

Ingresamos a: C:\xampp\htdocs\estudio\_yii2angular4\servicios\config\web.php y configuramos con la siguiente instrucción como se indica en las Fig. 27 y Fig. 28.

```
'urlManager' => [  
    'enablePrettyUrl' => true,  
    'showScriptName' => false,  
    'rules' => [  
  
        // Configuración Endpoint "Servicio" Cliente  
        [  
            'class'          => 'yii\rest\UrlRule',  
            'controller'     => 'cliente',  
            'pluralize'      => false,  
        ],  
    ],  
],
```

Fig. 27. Script de configuración web.php

Fuente: Propia

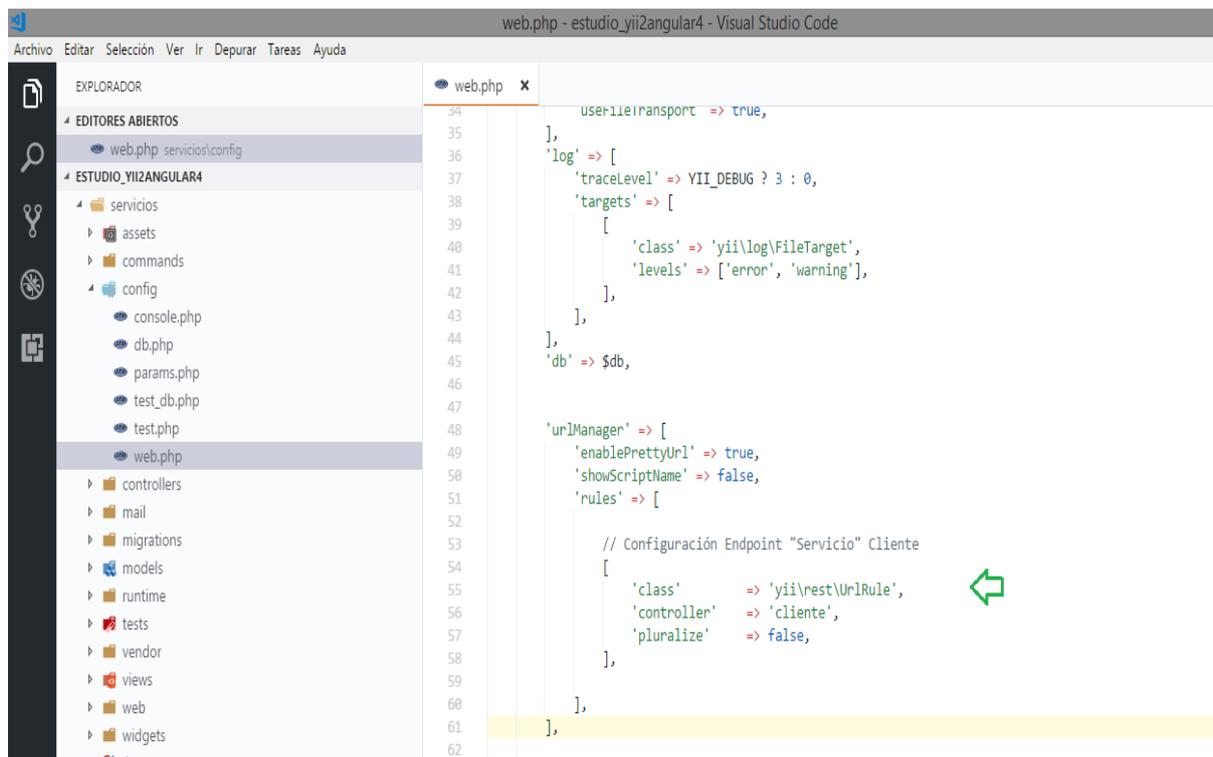


Fig. 28. Configuración de servicio REST en Yii2

Fuente: Propia

Finalmente probamos el servicio REST creado. Ingresamos la dirección citada desde cualquier navegador en la siguiente ruta `http://servicios.estudio.local/cliente`, este proceso se indica en la Fig. 29.



```
1 // 20171120172750
2 // http://servicios.estudio.local/cliente
3
4 [
5   {
6     "tab_cliente_id": 1,
7     "tab_cliente_ci": "0915984710",
8     "tab_cliente_nombre": "Andrés",
9     "tab_cliente_apellido": "G."
10  }
11 ]
```

Fig. 29. Prueba del servicio REST para clientes

Fuente: Propia

Para realizar las pruebas del funcionamiento de las peticiones Http, se debe instalar el programa POSTMAN y asignar la dirección del servicio REST `http://servicios.estudio.local/cliente`, para probar las respectivas acciones Http como: GET, POST, PUT, DELETE como se muestra en las: Fig. 30, Fig. 31, Fig. 32, Fig. 33.

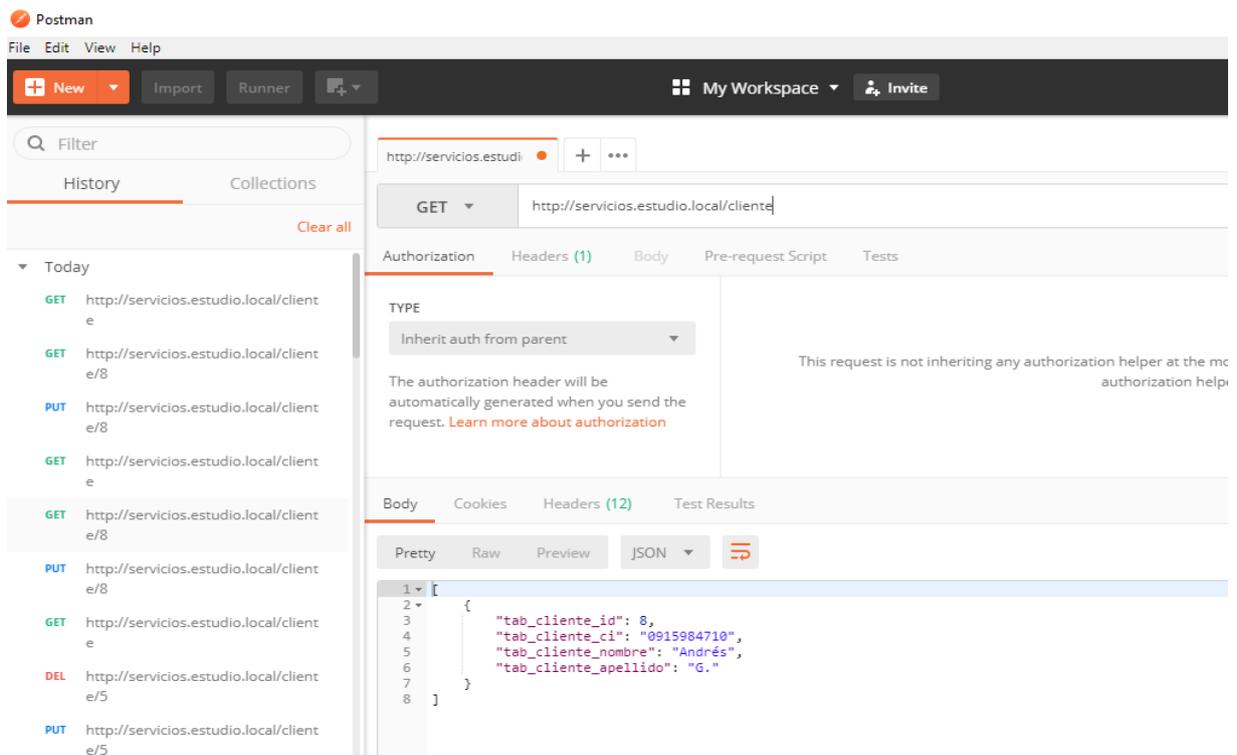


Fig. 30. Postman, prueba de servicio REST acción GET

Fuente: Propia

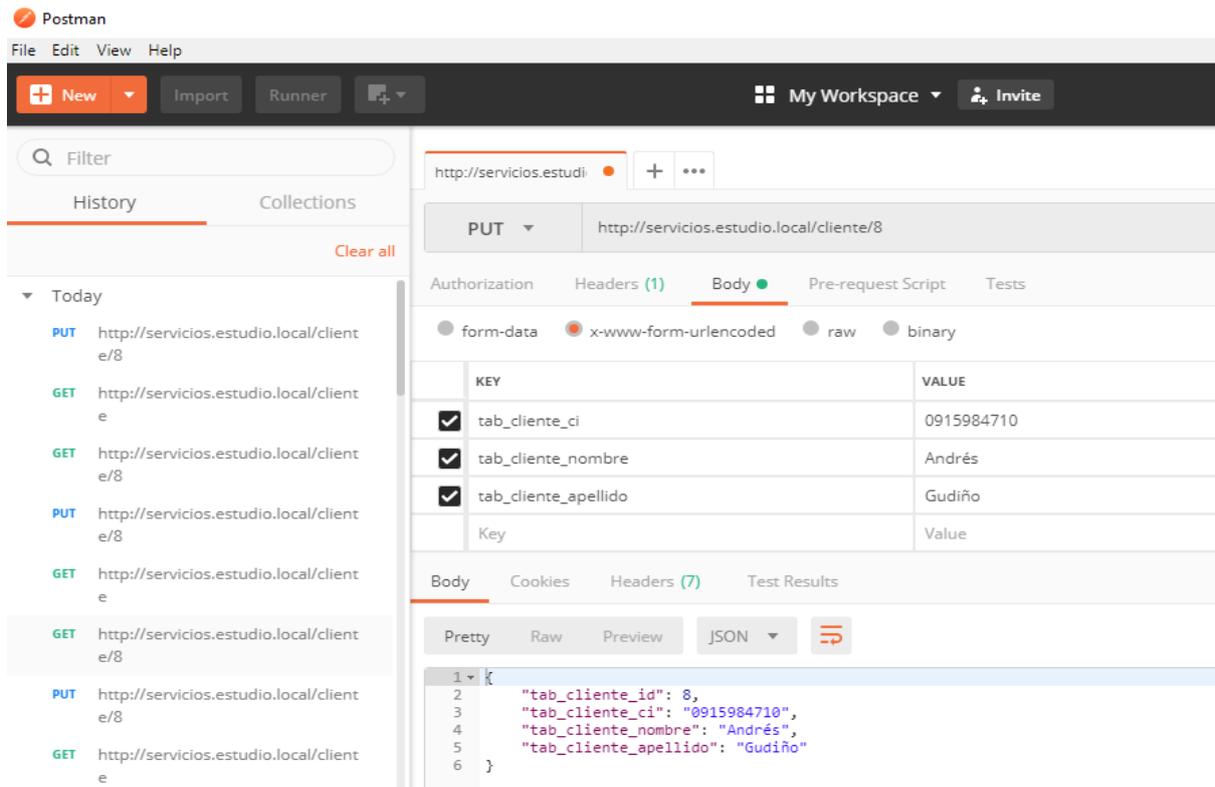


Fig. 31. Postman, prueba de servicio REST acción PUT

Fuente: Propia

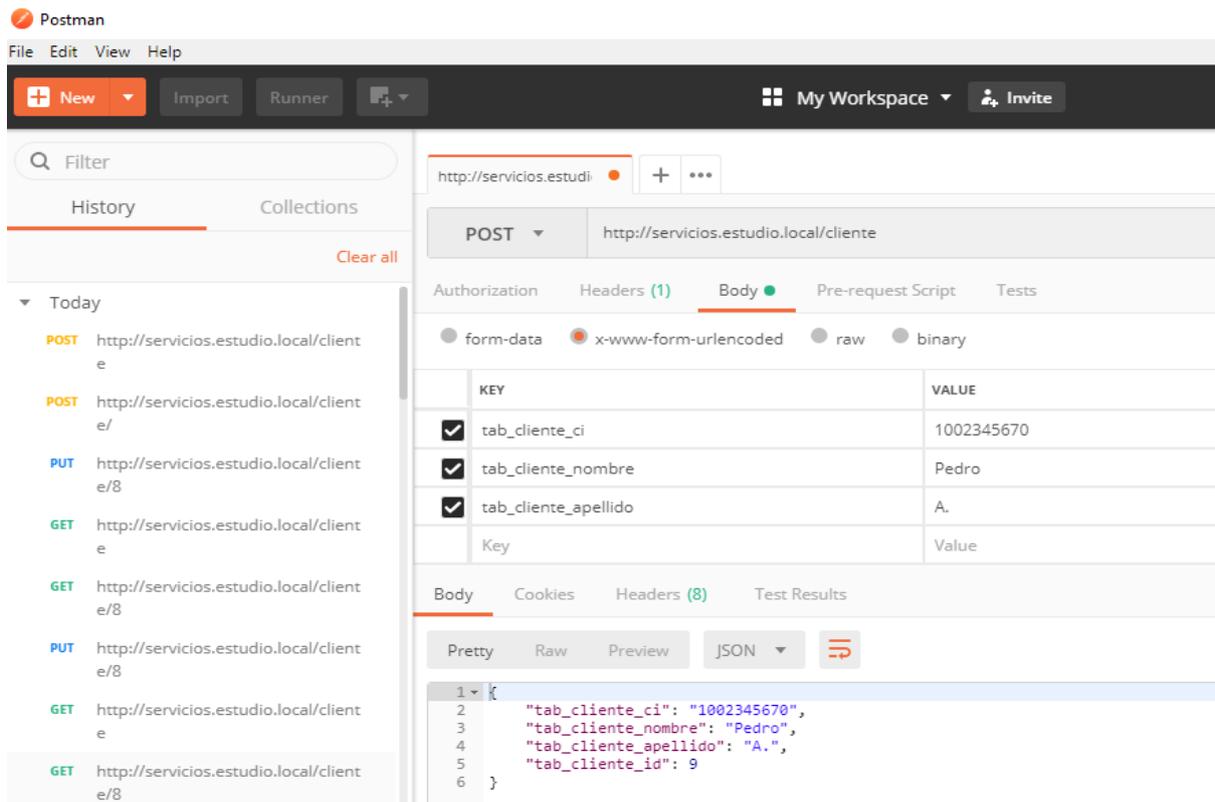


Fig. 32. Postman, prueba de servicio REST acción POST

Fuente: Propia

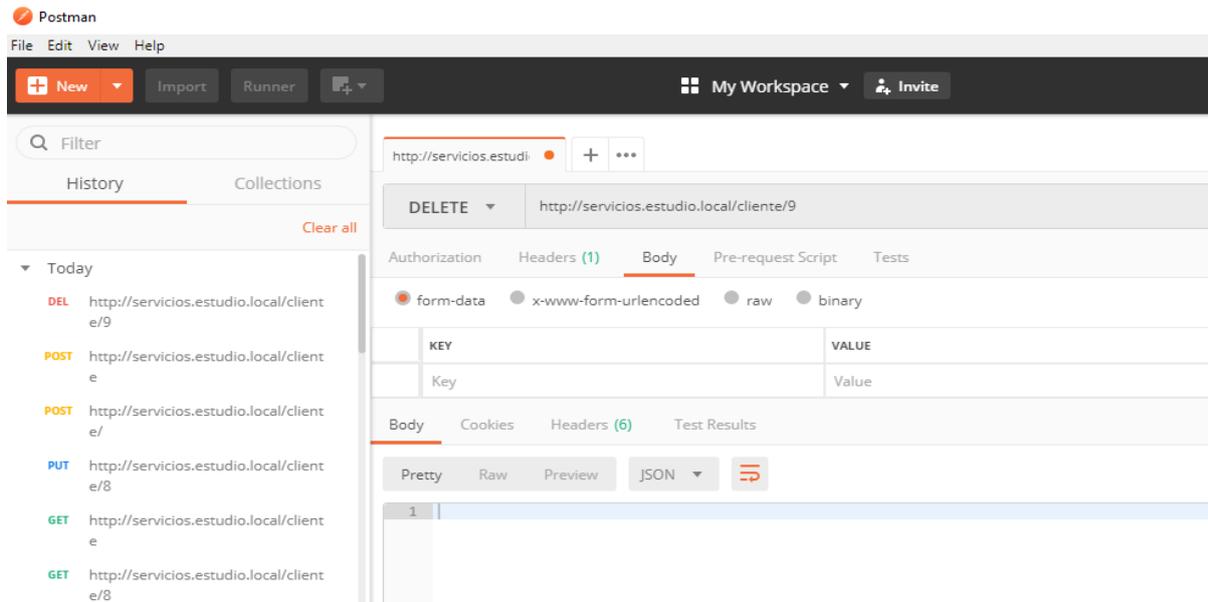


Fig. 33. Postman, prueba de servicio REST acción DELETE

Fuente: Propia

#### 1.14.4. Consumo de servicios REST con el Framework Angular

Para el respectivo consumo de un servicio REST es necesario la instalación del Framework Angular, la instalación se presenta en (Anexos 1). Una vez realizado los pasos de la instalación y configuración procederemos al desarrollo de la aplicación.

Lo primero que debemos hacer es abrir el proyecto con el editor de código Visual Studio Code y generar los componentes necesarios para el consumo del servicio REST creado en Yii2 Framework por medio de las peticiones Http: GET, POST, PUT, DELETE.

En Visual Studio Code presenta el terminal integrado "CMD" como se indica en la Fig. 34, esto permite interactuar las líneas de comando de Angular CLI con el proyecto de forma ágil.



Fig. 34. Ejecución de terminal integrado de Visual Code para proyecto en Angular

Fuente: Propia.

En la consola integrada de Visual Studio Code se puede ejecutar las instrucciones para la creación de componentes vitales para el proyecto, los componentes son encargados de controlar un espacio de pantalla o comúnmente llamado vista. Lo que se visualizara en pantalla está controlado y gestionado por un componente. Es necesario seguir los siguientes parámetros establecidos en la Tabla 5.

Tabla 5. Parámetros para generar componentes en Angular

Nombre	Código	Observación
Component - Componente	ng g component my-new-component	Generación de Componentes
Cliente-Lista	ng g c components/admin/cliente-lista --spec=false	Generación de Componentes requeridos
Cliente-Crear	ng g c components/admin/cliente-crear --spec=false	
Cliente-Actualizar	ng g c components/admin/cliente-actualizar --spec=false	

Fuente: Propia

Lo primero que se debe realizar es la generación automática de componentes citadas anteriormente en la Tabla 5, y visualizar los componentes creados para el cliente como se muestra en la Fig. 35.

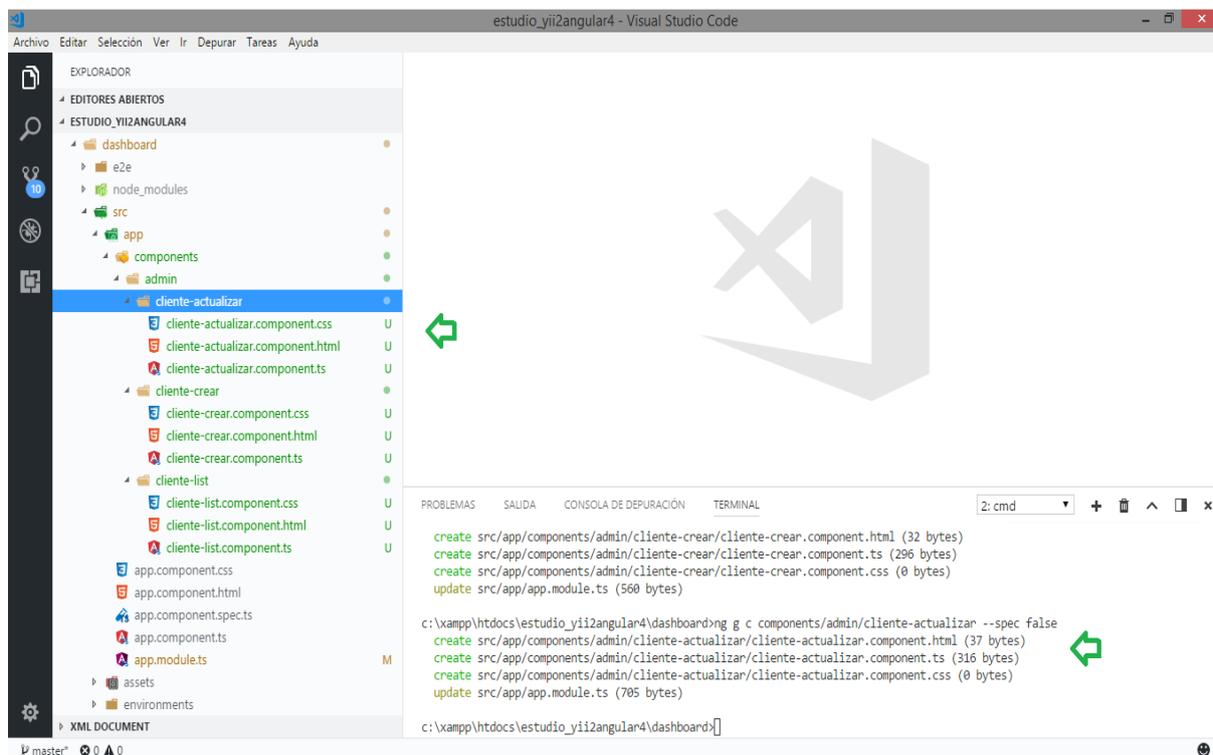


Fig. 35. Generación de nuevos componentes en Angular.

Fuente: Propia

Finalizada la generación de cada componente requerido, se debe crear los servicios. Los servicios son clases que nos permiten realizar acciones concretas, como son peticiones al

servidor o la lógica fuera de un componente. Es necesario seguir los siguientes parámetros establecidos en la Tabla 6.

Tabla 6. Parámetros para generar un servicio en Angular

Nombre	Código	Observación
Service - Servicio	<code>ng g service my-new-service</code>	Generación de Servicio
	Generación de Componentes requeridos	
Servicio Cliente	<code>ng g s servicios/cliente --spec=false</code>	

Fuente: Propia

Se debe establecer las instrucciones citadas anteriormente en la Tabla 6 para el proyecto y observar el servicio creado para cliente como se indica en la Fig. 36.



Fig. 36. Generación de un nuevo servicio en Angular

Fuente: Propia

El siguiente paso es crear un archivo de rutas, esto permite redireccionar los componentes entre sí para la comunicación, accedemos a la siguiente dirección: `C:\xampp\htdocs\estudio_yii2angular4\dashboard\src\app`, y crear un nuevo archivo llamado: `app.routing.ts` como se indica en la Fig. 37.

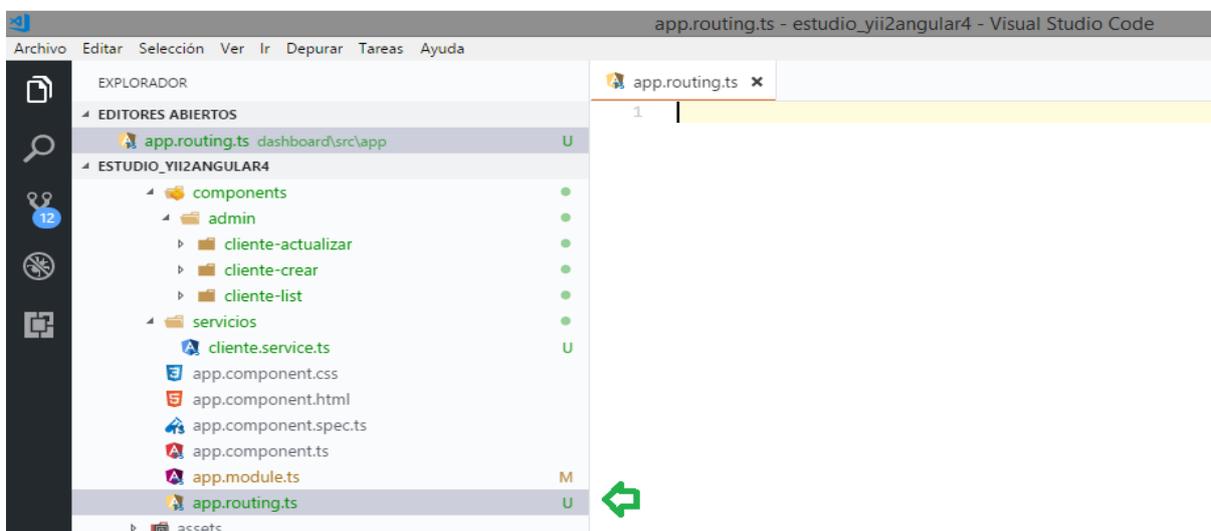


Fig. 37. Creación de un nuevo archivo de rutas en Angular

Fuente: Propia

Para el consumo de las petición mediante los Métodos GET, POST, PUT, DELET, se debe crear los servicios, para esto accedemos a: cliente.service.ts y se debe modificar el archivo con la siguiente estructura como se muestra en la Fig. 38.

```
// Inyector de dependencias, que son necesarios para los componentes,
// para acceder a los servicios.
import { Injectable } from '@angular/core';
//Permiten recibir respuestas para las solicitudes HTTP.
import { Http, Headers, RequestOptions, Response } from '@angular/http';
// Módulo HttpClient: Son métodos para realizar solicitudes HTTP.
import { HttpClient } from '@angular/common/http';
// Operadores de respuesta de valores.
import 'rxjs/add/operator/map';
import 'rxjs/add/operator/do';
import 'rxjs/add/operator/catch';
// Manejador de múltiples eventos, transformar, filtrar, consultar flujos de
datos.
import { Observable } from 'rxjs/Rx';

@Injectable()
export class ClienteService {

  // Declaramos el Servicio REST de YII2 asignando una variable.
  _apiurl: String = "http://servicios.estudio.local";

  // Creación de un constructor declarando los módulos para las respectivas
  peticiones, en este caso utilizaremos los módulos: Http, HttpClient
  constructor(private _http: Http, private _httpClient: HttpClient) { }
  // Metodo GET - Consumo de Servicio YII2.
  getAllClientes() {
    // Realiza una solicitud con el método GET http.
    return this._http.get(
      this._apiurl + '/cliente'
    )
    // Recibir datos en formato JSON.
    .map(response => response.json())
    // Captura de errores.
    .catch(this.handleError);
  }
  // Método DELETE - Petición a Servicio YII2.
  deleteCliente(id) {
    // Realiza una solicitud con el Método DELETE Http con el parámetro id, el
    identificador del objeto.
    this._httpClient.delete(this._apiurl + "/cliente/" + id)
    // Recibe como argumento un puntero a la respuesta Http.
    .subscribe(
      (data: any) => {
        console.log(data);
      }
    )
  }
}
```

```

}
// Método POST - Petición a Servicio YII2.
postCliente(formulario: any) {
  // Realiza una solicitud con el Método POST Http con los datos a ingresar
  // Por medio de un formulario.
  console.log("datos", formulario.value);
  this._httpClient.post(this._apiurl + `/cliente`, formulario)
    // Recibe como argumento un puntero a la respuesta Http.
    .subscribe(
      (data: any) => {
        console.log(data);
      }
    )
}
// Método PUT - Petición a Servicio YII2.
putCliente(id, formulario: any) {
  // Realiza una solicitud del Método PUT Http para retornar los datos creados
  y así actualizar, por medio de un formulario.
  console.log("datos", formulario.value);
  this._httpClient.put(this._apiurl + `/cliente/` + id, formulario)
    // Recibe como argumento un puntero a la respuesta Http.
    .subscribe(
      (data: any) => {
        console.log(data);
      }
    )
}
//Imprime mensajes de error que se produzcan y estos errores se muestren en la
consola.Para interceptar el manejo de errores.
private handleError(error: Response | any) {
  let errorMessage: any = {};
  // Connection error
  if (error.status == 0) {
    errorMessage = {
      success: false,
      status: 0,
      data: "Se produjo un error de conexión. Vuelve a intentarlo.",
    };
  }else {
    errorMessage = error.json();
  }
  return Observable.throw(errorMessage);
}
}
}

```

Fig. 38. Script de configuración cliente.service.ts

Fuente: Propia

Creado el servicio para los métodos Http: GET, POST, PUT, DELETE, realizamos la configuración de los componentes del cliente: Lista, Crear, Actualizar y los modificaremos:

- Para el componente Cliente Lista: cliente-lista.component.ts se debe modificar el código como se muestra en la Fig. 39.

```
//Directiva para verificar propiedades ligadas a datos.
import { Component, OnInit } from '@angular/core';
// Importamos el servicio creado con las peticiones "GET,POST,PUT,DELETE"
import { ClienteService } from '../../servicios/cliente.service';
//Proporciona la navegación y las capacidades de manipulación de URL.
import { Router } from '@angular/router';
@Component({
  selector: 'app-cliente-lista',
  templateUrl: './cliente-lista.component.html',
  styleUrls: ['./cliente-lista.component.css']
})
export class ClientelistaComponent implements OnInit {
  // Declaración de parámetros de datos.
  public cliente: any;
  public errorMessage: any;
  // Declaraciones del Servicio y URL asignadas.
  constructor(public _servcliente: ClienteService, private _router: Router) { }
  // Datos ligados al método.
  ngOnInit() {
    this.getAllClientesVista();
  }
  // Método para retornar Datos por medio del Servicio "GET"
  getAllClientesVista() {
    this._servcliente.getAllClientes().subscribe(
      result => {
        this.cliente = result;
        // Impresión de datos obtenidos en la consola.
        console.log("cliente", this.cliente);
      }
    ),
    //Imprime mensajes de error que se produzcan y estos errores se muestren
    en la consola. Para interceptar el manejo de errores.
    error => {
      this.errorMessage = <any>error;
      if (this.errorMessage != null) {
        console.log(this.errorMessage);
      }
    }
  }
  // Método para Eliminar un Dato por medio del Servicio "DELETE"
  eliminarCliente(id){
    // Se invoca al servicio con el parámetro id para ejecutar la acción DELETE
    this._servcliente.deleteCliente(id);
    this._router.navigate(['/']);
  }
}
```

Fig. 39. Script de configuración cliente-lista.component.ts

Fuente: Propia

- Para el componente Cliente Lista: cliente-lista.component.html, modificamos los parámetros de visualización de datos en la pantalla principal como se muestra en la Fig. 40.

```

<br>
<div class="container">
  <h3 center>Detalles Clientes </h3>
</div>
<div class="container">
  <h3 ><button type="button" class="btn btn-outline-success"
routerLink="/crear-cliente">Nuevo</button> </h3>
</div>
<br>
<div class="container">
  <table class="table">
    <thead>
      <tr>
        <th>ID</th>
        <th>CEDULA</th>
        <th>NOMBRE</th>
        <th>APELLIDO</th>
        <th>ACCIONES</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let item of cliente">
        <th>{{item.tab_cliente_id}}</th>
        <th>{{item.tab_cliente_ci}}</th>
        <th>{{item.tab_cliente_nombre}}</th>
        <th>{{item.tab_cliente_apellido}}</th>
        <th>
          <button type="button" class="btn btn-outline-warning"
routerLink="/actualizar-cliente/{{item.tab_cliente_id}}">Editar</button>
          <button type="button" class="btn btn-outline-danger"
(click)="eliminarCliente(item.tab_cliente_id)">Eliminar</button>
        </th>
      </tr>
    </tbody>
  </table>
</div>

```

Fig. 40. Script de configuración cliente-lista.component.html

Fuente: Propia

- Para el componente Cliente Actualizar: cliente-actualizar.component.ts es necesario realizar las siguientes instrucciones como se muestra en la Fig. 41.

```

// Permiten recibir respuestas para las solicitudes HTTP.
import { Http } from '@angular/http';
// Propiedades de una directiva ligadas a datos
import { Component, OnInit } from '@angular/core';
// Invocación del servicio
import { ClienteService } from '../../servicios/cliente.service';
// Vincula a un formulario para rastrear con los atributos de cada ítem del
formulario.
import { NgForm, NgModel } from '@angular/forms';
import { HttpClient } from '@angular/common/http';
//Proporciona la navegación y las capacidades de manipulación de URL.
import { Router } from '@angular/router';

@Component({
  selector: 'app-cliente-actualizar', templateUrl: './cliente-
actualizar.component.html', styleUrls: ['./cliente-actualizar.component.css']
})
export class ClienteActualizarComponent implements OnInit {
  // Declaración de parámetros de datos.
  public cliente: any;
  public errorMessage: any;
  // Parametros para ingreso de datos.
  _putCliente: Object = {
    tab_cliente_ci: "",
    tab_cliente_nombre: "",
    tab_cliente_apellido: ""
  }
  // Creación de un constructor declarando los módulos y servicios para las
respectivas peticiones.
  // En este caso utilizaremos los módulos: HttpClient y el servicio:
ClienteService.
  // Router proporciona la navegación y las capacidades de manipulación de URL.
  constructor(public _servcliente: ClienteService, private httpClient:
HttpClient, private _router: Router) { }
  // Método para ingresar Datos por medio del Servicio "PUT"
  actualizarCliente(id,formulario: any) {
    console.log("datos", formulario.value);
    // Se invoca al servicio con los parámetros ingresados por el
// formulario para ejecutar la acción PUT
    this._servcliente.putCliente(id, formulario.value );
    this._router.navigate(['']);
  }ngOnInit() {
  }
}

```

Fig. 41. Script de configuración cliente-actualizar.component.ts

Fuente: Propia

- Para el archivo cliente-actualizar.component.html, es necesario la creación del formulario para actualización de registros como se indica en la Fig. 42.

```

<br>
<br>
<form (ngSubmit)="actualizarCliente(3,formulario)" #formulario="ngForm"
novalidate="">
  <div class="container">
    <div class="form-group row">
      <label class="col-2 col-form-label">Cédula</label>
      <div class="col-md-8">
        <input class="form-control" type="text" placeholder="Cédula"
name="tab_cliente_ci" [(ngModel)]="_putCliente.tab_cliente_ci" required>
      </div>
    </div>
    <div class="form-group row">
      <label class="col-2 col-form-label">Nombre</label>
      <div class="col-8">
        <input class="form-control" type="text" placeholder="Nombre"
name="tab_cliente_nombre" [(ngModel)]="_putCliente.tab_cliente_nombre" required>
      </div>
    </div>
    <div class="form-group row">
      <label class="col-2 col-form-label">Apellido</label>
      <div class="col-8">
        <input class="form-control" type="text" placeholder="Apellido"
name="tab_cliente_apellido" [(ngModel)]="_putCliente.tab_cliente_apellido"
required>
      </div>
    </div>
    <div class="form-group row">
      <label class="col-2 col-form-label">&nbsp;</label>
      <div class="input-group col-md-8">
        <button type="submit" class="btn btn-outline-primary" >
          Actualizar
        </button>
      </div>
    </div>
  </form>

```

Fig. 42. Script de configuración Cliente-actualizar.component.html

Fuente: Propia

Terminado los pasos anteriores, se necesita crear las rutas para cada componente con su respectiva vista, para esto modificamos el archivo: C:\xampp\htdocs\estudio\_yii2angular4\dashboard\src\app\app.routing.ts como se muestra en la Fig. 43.

```
// Importamos el Componente
import { ClienteListaComponent } from './components/admin/cliente-lista/cliente-lista.component';
import { ClienteCrearComponent } from './components/admin/cliente-crear/cliente-crear.component';
import { ClienteActualizarComponent } from './components/admin/cliente-actualizar/cliente-actualizar.component';
// Rutas
import { RouterModule, Routes } from '@angular/router';

// Asignamos las rutas para los componentes.
const app_routes: Routes = [
  { path: 'inicio', component: ClienteListaComponent },
  { path: 'eliminar-cliente/:id', component: ClienteListaComponent },
  { path: 'crear-cliente', component: ClienteCrearComponent },
  { path: 'actualizar-cliente/:id', component: ClienteActualizarComponent },
  { path: '**', pathMatch: 'full', redirectTo: 'inicio' }
];
export const app_routing = RouterModule.forRoot(app_routes, { useHash: true });
```

Fig. 43. Script de configuración app.routing.ts

Fuente: Propia

Comprobamos el name space de los módulos en la sección de imports en el archivo app.module.ts como se muestra en la Fig. 44.

```
app.module.ts
1 import { ClienteService } from './servicios/cliente.service';
2 import { app_routing } from './app.routing';
3 import { RouterModule, Routes } from '@angular/router';
4 import { BrowserModule } from '@angular/platform-browser';
5 import { NgModule } from '@angular/core';
6 import { HttpClientModule } from '@angular/http';
7 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
8 import { HttpClientModule } from '@angular/common/http';
9 import { AppComponent } from './app.component';
10 import { ClienteListaComponent } from './components/admin/cliente-lista/cliente-lista.component';
11 import { ClienteCrearComponent } from './components/admin/cliente-crear/cliente-crear.component';
12 import { ClienteActualizarComponent } from './components/admin/cliente-actualizar/cliente-actualizar.component';
13
14 @NgModule({
15   declarations: [
16     AppComponent,
17     ClienteListaComponent,
18     ClienteCrearComponent,
19     ClienteActualizarComponent
20   ],
21   imports: [
22     BrowserModule,
23     RouterModule,
24     app_routing,
25     HttpClientModule,
26     FormsModule,
27     ReactiveFormsModule,
28   ],
29   providers: [ClienteService],
30   bootstrap: [AppComponent]
31 })
32 export class AppModule { }
33
```

Fig. 44. Importación de los módulos al name space

Fuente: Propia

Modificados el archivo app.component.html, para mostrar la página principal de la aplicación, para esto ingresamos a la siguiente ruta: C:\xampp\htdocs\estudio\_yii2angular4\dashboard\src\app\app.component.html, este archivo es necesario para mostrar la data o información del componente como se indica en la Fig. 45.

```

1 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
2   <div class="container">
3     <a class="navbar-brand" href="#"><h2>Yii2/Angular4</h2></a>
4     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExample07"
5       aria-controls="navbarsExample07" aria-expanded="false" aria-label="Toggle navigation">
6       <span class="navbar-toggler-icon"></span>
7     </button>
8     <div class="collapse navbar-collapse" id="navbarsExample07">
9       <ul class="navbar-nav mr-auto">
10        <li class="nav-item active">
11          <a class="nav-link" href="#">Inicio <span class="sr-only">(current)</span></a>
12        </li>
13      </ul>
14    </div>
15  </div>
16 </nav>
17 <router-outlet></router-outlet>
18

```

Fig. 45. Mostrar los datos del componente en Angular

Fuente: Propia.

Agregaremos las librerías de Bootstrap para los estilos de las plantillas de la aplicación, para esto se debe acceder al index.html que se encuentra en la ruta: C:\xampp\htdocs\estudio\_yii2angular4\dashboard\src\index.html y modificar con las siguientes instrucciones como se indica en la Fig. 46.

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Dashboard</title>
6   <base href="/">
7
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="icon" type="image/x-icon" href="favicon.ico">
10  <link rel="stylesheet" href="/assets/css/bootstrap.min.css">
11 </head>
12 <body>
13   <app-root></app-root>
14   <script src="/assets/js/bootstrap.min.js"></script>
15   <script src="/assets/js/jquery-3.2.1.slim.min.js"></script>
16   <script src="/assets/js/popper.min.js"></script>
17 </body>
18 </html>
19

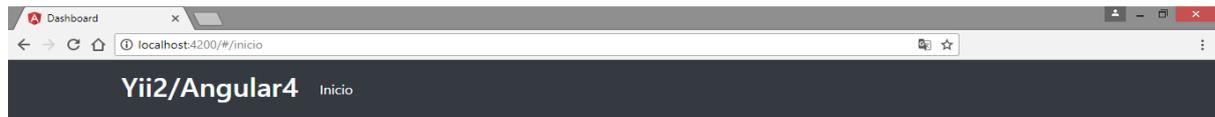
```

Fig. 46. Librerías Bootstrap en Angular

Fuente: Propia

Como parte final del estudio, es necesario ejecutar el proyecto con los diferentes métodos creados para los protocolos de comunicación Http: GET, POST, PUT, DELETE. Para esto

iniciamos el proyecto de Angular, ejecutando en “CMD” la siguiente línea de código `ng serve`. Ingresamos en el navegador web, para esto utilizamos Google Chrome para ejecutar el proyecto en la siguiente dirección: `http://localhost:4200` como se indica en la Fig. 47.



ID	CEDULA	NOMBRE	APELLIDO	ACCIONES
1	0915984710	Andrés	Gudiño	<button>Editar</button> <button>Eliminar</button>
3	1005653211	Marco	Almeida	<button>Editar</button> <button>Eliminar</button>

Fig. 47. Consumo de REST entre Yii2 y Angular

Fuente: Propia

Como se puede observar en la pantalla principal de la aplicación el cliente puede realizar las opciones como: listar, crear, editar y eliminar un cliente. La documentación de forma más detallada se encuentra presente en (Anexo B).

## 1.15. Metodología SCRUM

Scrum es una metodología de desarrollo de software ágil, cada proyecto creado a partir de la metodología tiene como objetivo controlar y planificar todas las etapas de un proyecto para ajustar todos los resultados y exigencias de un cliente.

Scrum permite minimizar los riesgos en el desarrollo de un sistema y mejorando la productividad con el equipo de trabajo.

### 1.15.1. Roles y responsabilidades

Para el proceso de desarrollo de software es necesario asignar roles que determinan la responsabilidad para el equipo de trabajo como se indica en la Fig. 48.

- **Propietario del Producto “Product Owner”:** Representa la voz del cliente, es el delegado de transmitir la visión del producto. Su principal función es recabar todas las ideas formuladas por el cliente, para luego escribir de forma ordenada asignando prioridades en la pila de producto “Product Backlog” y determinar tareas para cada sprint para proporcionar un producto de calidad (Satpathy, 2016).

- **Director del proyecto “Scrum Master”:** Es el líder del equipo de desarrollo Scrum, asegura que el equipo este dotado de un ambiente de trabajo completo, es el encargado de garantizar el cumplimiento de procesos, funciones y de la metodología empleada en el proyecto. Resuelve complicaciones entre el equipo de trabajo (Satpathy, 2016).
- **Equipo Scrum “Scrum Team”:** Es un equipo de personas profesionales y responsables de entregar avances del producto correspondiente a cada tarea “Sprint” (Ken Schwaber y Jeff Sutherland, 2014).
- **Stakeholders:** Son usuarios pertenecientes a la empresa del cliente, que interactúan con el propietario del producto, son personas ajenas al equipo de desarrollo Scrum. Son observadores durante las revisiones de las tareas “Sprint” (Satpathy, 2016).

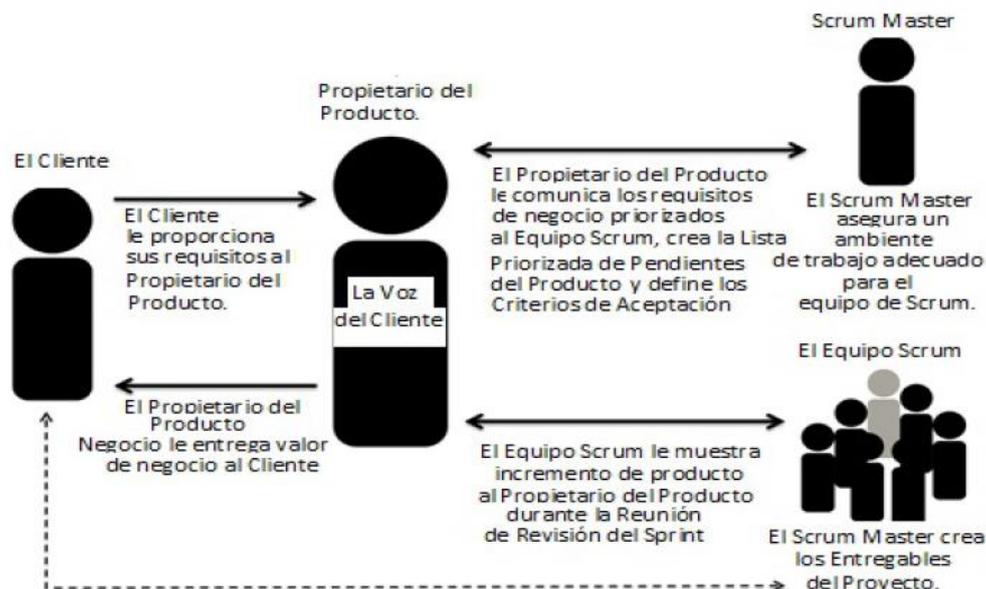


Fig. 48. Organización en Scrum

Fuente: (Satpathy, 2016)

### 1.15.2. Elementos de Scrum

- **Lista de Producto “Product Backlog:** Es una lista ordenada con todas las necesidades del cliente, sirve para realizar cualquier cambio en el producto. La lista debe ser dinámica y está sujeta a cambios constantes para identificar que el producto sea adecuado y competitivo (Ken Schwaber y Jeff Sutherland, 2014).

- **Pila de Sprint “Sprint Backlog”**: Es la lista de tareas que deben completarse en los tiempos acordados para que un incremento del producto sea entregado y cumplir el objetivo de cada Sprint. Las listas pendientes del Sprint son funcionalidades que formara parte de un próximo incremento (Ken Schwaber y Jeff Sutherland, 2014).
- **Reuniones Scrum “Scrum Daily”**: Son reuniones realizadas todos los días por parte de cada miembro del equipo, en donde se detalla los avances realizados de cada sprint, objetivos alcanzados y plantear problemas que puedan ocurrir para establecer una solución (Ken Schwaber y Jeff Sutherland, 2014).
- **Reunión de planificación Sprint “Scrum Daily”**: En esta reunión se analiza el objetivo del Sprint a realizar asignando un sinnúmero de tareas y estableciendo tiempos. (Ken Schwaber y Jeff Sutherland, 2014).
- **Revisión Sprint**: Son reuniones informales para revisar dificultades para lograr las tareas encomendadas (Satpathy, 2016).
- **Retrospectiva Sprint**: Es la revisión de problemas ocurridos en el Sprint anterior. (Satpathy, 2016).



Fig. 49. Elementos de la metodología Scrum

Fuente: (Islavisual.com, 2014)



## CAPÍTULO 2

### 2. Desarrollo de la Metodología del Software

Para el desarrollo del software se ha optado por la implementación de la metodología SCRUM, para esto es necesario la estructuración correcta del proyecto, así como la información recabada del proyecto para su respectiva construcción.

#### 2.1. Roles del Sistema

Para el desarrollo de la aplicación se debe asignar roles al equipo de trabajo para un correcto desempeño. En la Tabla 7, se detalla el equipo de trabajo para la metodología SCRUM:

Tabla 7. Roles del Sistema

Rol	Nombre	Descripción
Propietario del Proyecto (Product Owner).	Sr. Eduardo Núñez Gerente Comercial empresa INDUXION	Entidad Auspiciante
Jefe del Proyecto (Scrum Master).	MSc. Diego Trejo	Director de Tesis
Equipo de desarrolladores (Team Members).	Andrés Gudiño	Desarrollador del Sistema - Tesista

Fuente: Propia

#### 2.2. Historias de Usuario y Criterios de Aceptación

En la metodología de desarrollo ágil SCRUM, la estructura óptima para levantar los requerimientos es mediante las historias de usuario, esto se enfoca en las necesidades que el usuario necesita hacer. La estructura para el desarrollo de las historias de usuario presenta los siguientes campos de información:

- Identificador de la historia (ID): Identificador que se asigna a la historia de usuario.
- Rol: Muestra el rol de un usuario con sus respectivas tareas de forma detallada.
- Funcionalidad: Se detalla que va a poder hacer en la aplicación.
- Número de escenarios: Son los números que pueden estar presentes para cada funcionalidad.
- Criterios: Descripción de los escenarios con lo que pueden presentar.
- Resultados: Es el comportamiento que presenta el sistema en cada criterio.

Tabla 8. Historias de usuarios y criterios de aceptación

Enunciado de la Historia					Criterios de Aceptación	
Id Historia	Solicitante	Rol	Funcionalidad	Escenario	Criterio	Resultados
001	Sr. Eduardo Núñez	Como Administrador y Vendedor.	Registrar a los clientes que realizan compras en la empresa.	1	Puede crear y actualizar la información de clientes.	Presentar una página para registrar la información necesaria del cliente.
002	Sr. Eduardo Núñez	Como Administrador y Vendedor.	Información de clientes registrados.	1	Los trabajadores de la empresa pueden buscar a un cliente por cédula o por nombre.	Mostrar los datos por medio de una búsqueda en la página.
				2	Sin Resultados	Mostrar un mensaje.
003	Sr. Eduardo Núñez	Como Administrador	Registro de los datos de la empresa	1	Puede crear y actualizar la información del negocio.	Se creará un módulo correspondiente a los datos del negocio como: Representante legal, ciudad, información de la sucursal.
004	Sr. Eduardo Núñez	Como Administrador	Registro: catálogo de productos de proveedores.	1	Puede crear y actualizar la información correspondiente a productos.	Se creará un módulo de productos con la información sobre: categoría, marca, productos.
				2	Puedo crear y actualizar la información del proveedor.	Creación del módulo Compras: Registro de proveedor.

005	Sr. Eduardo Núñez	Como Administrador	Registro de compras realizadas a proveedores con las respectivas cantidades y su valor.	1	Registrar o anular en su totalidad una compra realizada a un proveedor, seleccionar productos y establecer los valores.	Sección Compras: Registrar información correspondiente a las entradas de productos - compras. Adicional crear el módulo de Impuesto IVA.
				2	Lista de compras realizadas	Sección compras: Lista de compras con sus respectivos detalles.
006	Sr. Eduardo Núñez	Como Administrador y Vendedor.	Información del stock de productos disponible en la empresa.	1	Visualizar las cantidades disponibles de productos en la empresa.	Módulo: Stock - Inventario de productos.
007	Sr. Eduardo Núñez	Como Administrador y Vendedor.	Registrar las salidas de Productos.	1	Registrar o anular en su totalidad una salida de productos que se ha realizado en la empresa.	Creación de un módulo ventas para el respectivo descargo de productos.
				2	Listar las ventas realizadas	Sección ventas: Listar un registro con las ventas realizadas con sus respectivos detalles.
008	Sr. Eduardo Núñez	Como Administrador y Vendedor.	Información de entradas y salida de productos.	1	Información detallada de las entradas y salidas correspondiente a productos.	Creación de un módulo Kardex o Movimientos para detallar las entradas y salidas de productos.
009	Sr. Eduardo Núñez	Como Administrador	Registro de usuarios para el ingreso al sistema.	1	Registrar usuarios con su rol: Administrador y Vendedor.	Creación del Módulo Administración del Sistema.

Fuente: Propia

### 2.3. Pila de Productos

Se establecen los requerimientos para el diseño, desarrollo y descripción de funcionalidades del sistema que son programadas en la fase de iteraciones como se indica en la Tabla 9.

Tabla 9. Pila de Productos

Id	Nombre	Observación
R1	Análisis y estructuración del Proyecto.	Descripción detallada con todos los parámetros necesarios para la creación del proyecto.
R2	Creación del Módulo Clientes.	Registro de información de clientes, acciones a realizar: crear, actualiza, listar, buscar registros.
R3	Creación del Módulo Negocio.	Creación de secciones para el módulo: representante legal, ciudad, información sobre la ubicación. Acciones para realizar para cada sección: crear, actualizar, buscar registros.
R4	Creación del Módulo Productos.	Creación de secciones para el módulo como: categoría, marca, producto. Acciones para realizar para cada sección: crear, actualizar, buscar registros.
R5	Creación del Módulo Impuestos.	Lista de impuesto con la opción de actualizar el valor de impuesto IVA.
R6	Creación del Módulo Mercadería.	Creación de secciones para el módulo como: proveedores, compras, lista de compras realizadas. Acciones para realizar en proveedores: crear, actualizar y buscar registros. Acciones para realizar en compras: registrar una compra o anular en su totalidad una compra. Acciones para realizar en lista de compras: buscar registro y visualizar los detalles de cada compra realizada.
R7	Creación del Módulo Stock.	Información detallada de las existencias de productos en la empresa, las acciones a realizar en este módulo son: listar productos con toda la información y su valor de venta.
R8	Creación del Módulo Ventas.	Creación de secciones para el módulo como: ventas, lista de ventas. Acciones para realizar una venta: registra una venta o anulación en su totalidad. Acciones para realizar en listar ventas: buscar registros, visualizar los detalles de cada venta realizada.

R9	Creación del Módulo Kardex.	Información detallada de las entradas y salidas de productos.
R10	Módulo de Seguridad y Control.	Autenticación para el ingreso al sistema con su respectivo rol de usuario. Roles del sistema: Administrador, Vendedor.
R11	Implementación del Sistema.	Implementación del sistema para la respectiva empresa.

Fuente: Propia

## 2.4. Pila de Tareas

Pila encargada de realizar detenidamente el análisis de requerimientos, asignando a cada una en pequeñas tareas que los desarrolladores deben cumplir, en un tiempo corto de desarrollo.

Tabla 10. Análisis y Estructuración del Proyecto

Id	Nombre	Observación
T1	Instalación de Servidor Apache XAMPP.	Configuración e instalación del servidor web local y del motor de base de datos MariaDB.
T2	Instalación de Composer.	Instalación y configuración del manejador de dependencias PHP.
T3	Instalación de Visual Studio Code.	Instalación del editor de código de Visual Studio Code.
T4	Instalación de Node.js	Instalación y configuración del motor de entorno para JavaScript.
T5	Instalación de TypeScript.	Instalación de TypeScript.
T6	Instalación del Framework Yii2 – Proyecto.	Proyecto: Instalación y configuración del Framework PHP asignado como Back-End.
T7	Instalación de Postman.	Instalación, programa para realizar pruebas de los servicios REST.
T8	Instalación del Framework Angular – Proyecto.	Proyecto: Instalación y configuración del Framework TypeScript asignado como Front-End
T9	Creación de la Base de datos.	Modelo entidad relación y creación de la base de datos.

Fuente: Propia

Tabla 11. Creación del Módulo Clientes

Id	Nombre	Observación
T10	Creación del servicio REST para el Módulo clientes.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End.
T11	Creación de la interfaz módulo cliente.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear y editar.
T12	Lista de clientes.	Buscador de clientes por cédula o nombres.
T13	Reporte de clientes.	Exportación de un archivo en formato Excel con todos los registros ingresados en el sistema.

Fuente: Propia

Tabla 12. Creación del Módulo Negocio

Id	Nombre	Observación
T14	Creación del servicio REST para el Módulo Negocio.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End. Se crea sus respectivas secciones: representante legal, ciudad, información de tienda.
T15	Creación de la interfaz representante legal.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End. con las opciones de crear, editar.
T16	Listar y buscar el representante legal registrado.	Listar, buscador de información del representante legal, por cédula o nombre.
T17	Creación de la interfaz ciudad.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear, editar.
T18	Listar, buscar ciudades registradas.	Buscador de información por nombre.
T19	Creación de la interfaz tienda.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear, editar.
T20	Listar, buscar.	Buscador de información por nombre.

Fuente: Propia

Tabla 13. Creación del Módulo Productos

Id	Nombre	Observación
T21	Creación del servicio REST para el Módulo Productos.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End. Se crea sus respectivas secciones: categoría, marca, productos.
T22	Creación de la interfaz categoría.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear, editar.
T23	Listar y buscar registros para categoría.	Listar y buscar de información por nombre.
T24	Creación de la interfaz marca.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear, editar.
T25	Listar y buscar registros para marcas.	Listar, buscar información por nombre.
T26	Creación de la interfaz producto.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear, editar.
T27	Listar buscar y reportes de productos.	Listar, buscar información de productos.

Fuente: Propia

Tabla 14. Creación del Módulo Impuestos

Id	Nombre	Observación
T28	Creación del servicio REST para el Módulo Impuestos.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End.
T29	Creación de la interfaz impuestos IVA.	Se crea el formulario con sus parámetros desde el Framework Angular encargado del Front-End, con la operación editar.
T30	Listar, buscar registros correspondientes a impuestos.	Buscar impuestos por nombre.

Fuente: Propia

Tabla 15. Creación del Módulo Mercadería

Id	Nombre	Observación
T31	Creación del servicio REST para el Módulo Mercadería.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End. Se crea sus respectivas secciones: proveedores, compras.
T32	Creación de la interfaz proveedores.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones de crear, editar.
T33	Listar, buscar registros de proveedores.	Listar y buscar de información por nombre.
T34	Crear interfaz de compras de productos.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con la opción de crear una compra seleccionando a un proveedor, listar compras, también se puede anular la compra en su totalidad.
T35	Listar, buscar registros de compras ingresadas en la empresa.	Listar compras registradas o anuladas.

Fuente: Propia

Tabla 16. Creación del Módulo Stock

Id	Nombre	Observación
T36	Creación del servicio REST para el Módulo Stock.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End.
T37	Creación de la interfaz Stock.	Se crea el formulario con sus parámetros desde el Framework Angular encargado del Front-End. Lista de productos con su respectiva información y un buscador de productos con sus existencias.

Fuente: Propia

Tabla 17. Creación del Módulo Ventas

Id	Nombre	Observación
T38	Creación del servicio REST para el Módulo Ventas.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End.

T39	Crear interfaz de ventas de productos.	Se crea los formularios con sus parámetros desde el Framework Angular encargado del Front-End, con la opción de crear una venta, con su respectiva información de productos, impuestos y cliente. Adicional se puede realizar la anulación en su totalidad una venta y un reporte del total recaudado en ventas.
T40	Listar, buscar registros de ventas registradas o anuladas.	Listar las ventas realizadas con sus respectivos detalles y con la opción de buscar anulación de venta.

Fuente: Propia

Tabla 18. Creación del Módulo Kardex

Id	Nombre	Observación
T41	Creación del servicio REST para el módulo Kardex.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End.
T42	Creación de la interfaz Kardex.	Se crea el formulario con sus parámetros desde el Framework Angular encargado del Front-End. Lista de movimientos de productos con su respectiva información y un buscador para entradas y salidas de productos

Fuente: Propia

Tabla 19. Módulo de Seguridad y Control

Id	Nombre	Observación
T43	Creación del servicio REST para el módulo de ingreso al sistema.	Se crea el servicio REST desde el Framework Yii2 PHP que es el encargado del Back-End.
T44	Creación de la interfaz para la administración de usuarios con su rol.	Se crea el formulario con sus parámetros desde el Framework Angular encargado del Front-End, con las opciones crear, editar y buscar usuarios.

Fuente: Propia

Tabla 20. Implementación del Sistema

Id	Nombre	Observación
T45	Implementación del sistema en la empresa, en un servidor Hosting.	Almacenar toda la información de la empresa y comprobar su funcionalidad del sistema.

Fuente: Propia

## 2.5. Planificación del Proyecto

En esta sección del proyecto se planifica cada iteración o sprint, con el que se define qué tareas y requisitos se debe cumplir en cada iteración, además se determina la fecha de inicio y fin para la ejecución como se indica en la Tabla 21.

Tabla 21. Planificación del Proyecto

Id	Nombre	Inicio	Fin	Requerimientos
I1	Análisis y estructuración del Proyecto	02/04/2018	30/04/2018	R1, T1, T2, T3, T4, T5, T6, T7, T8, T9
I2	Creación del Módulo Clientes	01/05/2018	15/05/2018	R2, T10, T11, T12, T13
I3	Creación del Módulo Negocio	16/05/2018	01/06/2018	R3, T14, T15, T16, T17, T18, T19, T20
I4	Creación del Módulo Productos	04/06/2018	22/06/2018	R4, T21, T22, T23, T24, T25, T26, T27
I5	Creación del Módulo Impuestos	25/06/2018	29/06/2018	R5, T28, T29, T30
I6	Creación del Módulo Mercadería	02/07/2018	24/07/2018	R6, T31, T32, T33, T34, T35
I7	Creación del Módulo Stock	25/07/2018	29/07/2018	R7, T36, T37
I8	Creación del Módulo Ventas	30/07/2018	25/08/2018	R7, T38, T39, T40
I9	Creación del Módulo Kardex	27/08/2018	30/08/2018	R9, T41, T42
I10	Módulo de Seguridad y Control	31/08/2018	20/09/2018	R10, T43, T44
I11	Implementación del Sistema	21/09/2018	26/09/2018	R11, T45

Fuente: Propia

## 2.6. Iteraciones

### 2.6.1. Iteración 1. Análisis y estructuración del Proyecto

Para la presente sección se detalla los procedimientos como es la instalación y configuración de las herramientas necesarias para el desarrollo del software y la creación de la base de datos base de datos. Las instalaciones de programas paso a paso se encuentran en (Anexos A).

**Tarea Nro. 1.-** Instalación del servidor local Apache XAMPP.

El panel de control XAMPP Fig. 50, permite iniciar los servicios del servidor apache y MariaDB como el motor de base de datos y como la configuración de puertos para cada uno. Para la correcta administración de la base de datos MariaDB Fig. 51, para la empresa INDUXION se presenta la siguiente interfaz.

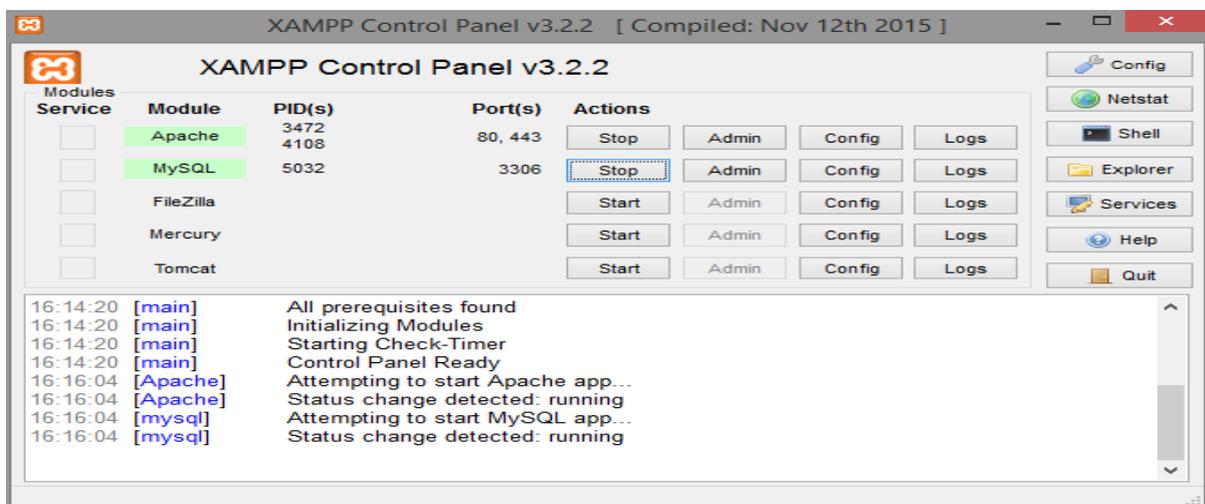


Fig. 50. Instalación de APACHE

Fuente: Propia

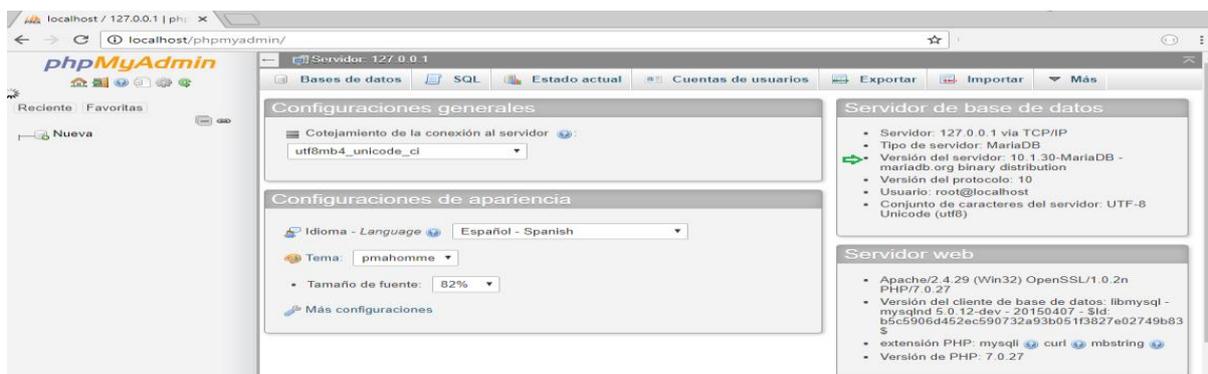
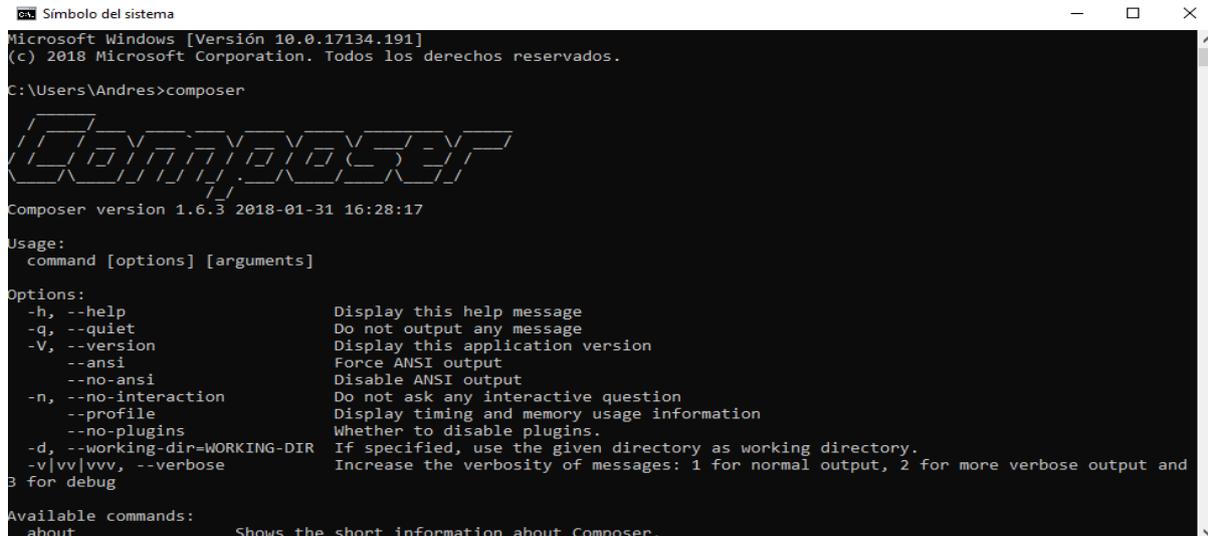


Fig. 51. Base de datos MariaDB

Fuente: Propia

## Tarea Nro. 2.- Instalación de Composer.

Composer es el manejador de dependencias del lenguaje PHP Fig. 52, permite descargar repositorios requeridos a los proyectos para el respectivo desarrollo.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17134.191]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Andres>composer

Composer version 1.6.3 2018-01-31 16:28:17

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet                Do not output any message
  -V, --version              Display this application version
  --ansi                    Force ANSI output
  --no-ansi                 Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                  Display timing and memory usage information
  --no-plugins               Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                      Shows the short information about Composer.
```

Fig. 52. Instalación de Composer

Fuente: Propia

## Tarea Nro. 3.- Instalación de Visual Studio Code.

En la Fig. 53, se muestra el software Visual Studio Code que es un editor de código desarrollado por Microsoft, presenta un soporte para los diferentes lenguajes de programación por medio de extensiones.

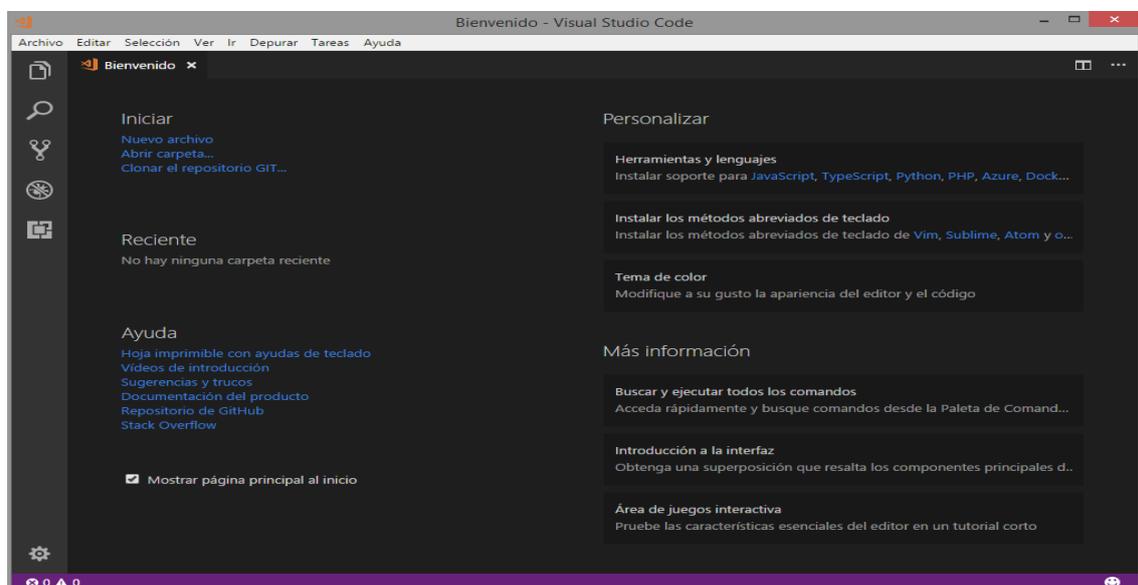
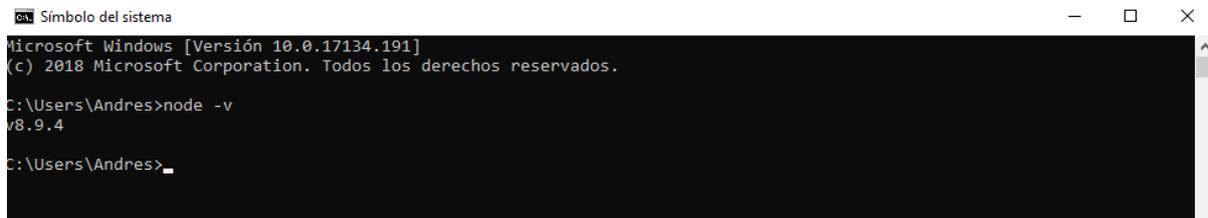


Fig. 53. Instalación de Visual Studio Code

Fuente: Propia

#### Tarea Nro. 4.- Instalación de Node.js.

Node.js es un entorno de ejecución para JavaScript, que se presenta del lado del servidor para la ejecución de instrucciones, para verificar la versión presente en el sistema se puede verificar con la siguiente instrucción por medio de la consola CMD: `node -v` Fig. 54.



```
Microsoft Windows [Versión 10.0.17134.191]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Andres>node -v
v8.9.4

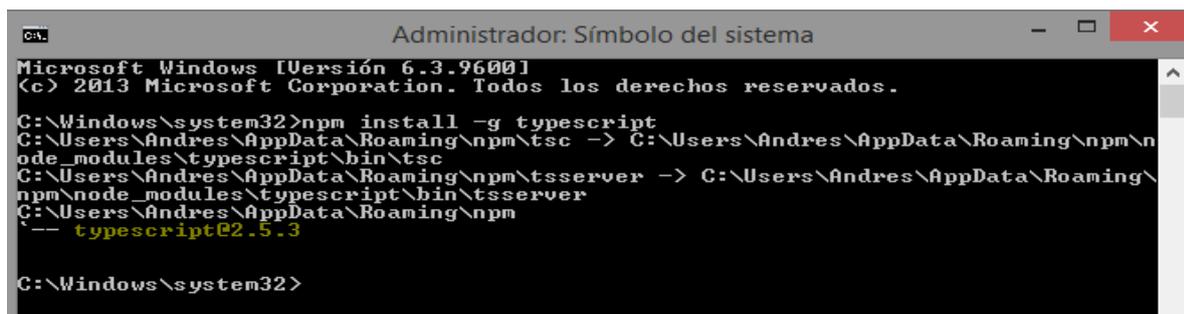
C:\Users\Andres>
```

Fig. 54. Instalación de Node.js

Fuente: Propia

#### Tarea Nro. 5.- Instalación de TypeScript.

Para el proceso de instalación se debe ingresar la siguiente línea de código por la consola CMD: `npm install -g typescript` como se indica en la Fig. 55. TypeScript es el lenguaje de programación a trabajar en el desarrollo del software con el Framework Angular.



```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>npm install -g typescript
C:\Users\Andres\AppData\Roaming\npm\tsc -> C:\Users\Andres\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\Andres\AppData\Roaming\npm\tsserver -> C:\Users\Andres\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
C:\Users\Andres\AppData\Roaming\npm
-- typescript@2.5.3

C:\Windows\system32>
```

Fig. 55. Instalación de TypeScript

Fuente: Propia

#### Tarea Nro. 6.- Instalación del Framework Yii2.

La instalación de Yii2 se realiza en la carpeta del servidor apache “XAMPP” para el correcto funcionamiento del Framework, esto quiere decir que todos los proyectos debemos instalar en la siguiente ruta: `C:\xampp\htdocs`.

Para crear un proyecto nuevo, ejecutamos un terminal “CMD” la ruta citada anteriormente, ingresamos la siguiente línea de código en la consola: `composer create-project yiisoft/yii2-app-basic “nombre_proyecto”` como se puede apreciar en la Fig. 56 y Fig. 57.

```
C:\Windows\System32\cmd.exe - composer create-project yiisoft/yii2-app-basic servicios
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\xampp\htdocs\yii2-angular4>composer create-project yiisoft/yii2-app-basic servicios
Installing yiisoft/yii2-app-basic (2.0.13)
- Installing yiisoft/yii2-app-basic (2.0.13): Downloading (connecting...)
```

Fig. 56. Framework Yii2

Fuente: Propia

```
C:\Windows\System32\cmd.exe
symfony/console suggests installing symfony/filesystem (<)
symfony/console suggests installing symfony/process (<)
symfony/browser-kit suggests installing symfony/process (<)
symfony/event-dispatcher suggests installing symfony/dependency-injection (<)
symfony/event-dispatcher suggests installing symfony/http-kernel (<)
phpunit/phpunit-mock-objects suggests installing ext-soap (<*)
phpunit/php-code-coverage suggests installing ext-xdebug (<^2.5.5)
sebastian/global-state suggests installing ext-uopz (<*)
phpunit/phpunit suggests installing phpunit/php-invoker (<^1.1)
phpunit/phpunit suggests installing ext-xdebug (<*)
codeception/base suggests installing flow/jsonpath (For using JSONPath in REST module)
codeception/base suggests installing phpseclib/phpseclib (for SFTP option in FTP Module)
codeception/base suggests installing league/factory-muffin (For DataFactory module)
codeception/base suggests installing league/factory-muffin-faker (For Faker support in DataFactory module)
codeception/base suggests installing symfony/phpunit-bridge (For phpunit-bridge support)

Writing lock file
Generating autoload files
> yii\composer\Installer::postCreateProject
chmod('runtime', 0777)...done.
chmod('web/assets', 0777)...done.
chmod('yii', 0755)...done.
> yii\composer\Installer::postInstall
C:\xampp\htdocs\yii2-angular4>
```

Fig. 57. Instalación del Framework Yii2

Fuente: Propia

Finalizada la instalación del Framework, debemos ejecutar los servicios de XAMPP y ejecutar en un navegador web la siguiente dirección: <http://127.0.0.1/projects/basic/web/> como se indica en la Fig. 58.

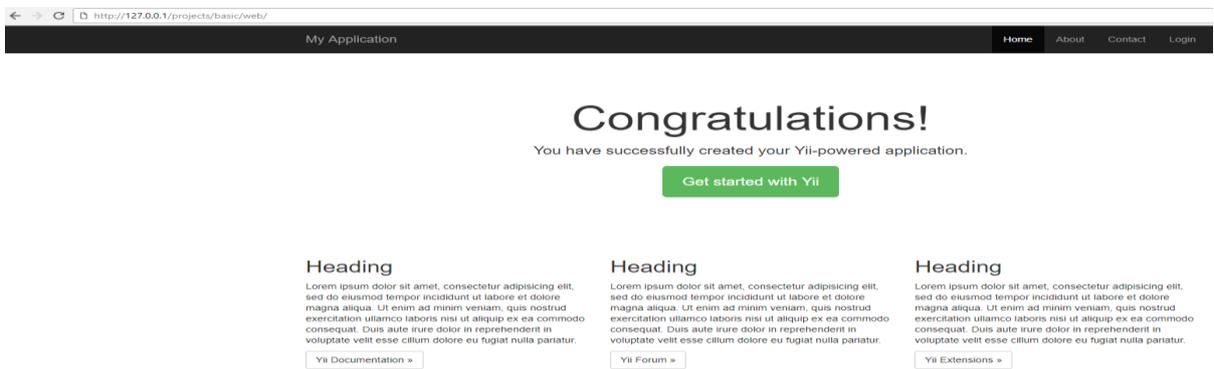


Fig. 58. Vista inicial del proyecto instalado del Framework Yii2

Fuente: Propia

## Tarea Nro. 7.- Instalación de Postman.

Postman es un software que permite realizar pruebas para comprobar las peticiones realizadas desde Http REST, la interfaz del software se presenta en la Fig. 59.

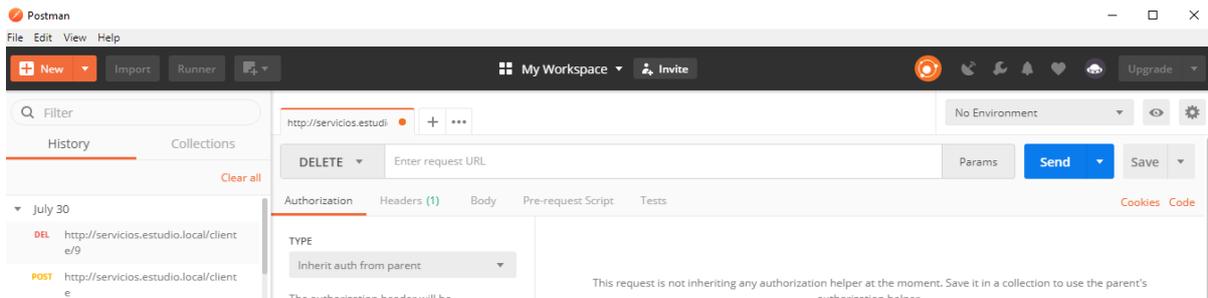


Fig. 59. Postman

Fuente: Propia

### Tarea Nro. 8.- Instalación del Framework Angular.

Para la correcta instalación del Framework Angular, es necesario establecer una dirección de instalación del nuevo proyecto, por ejemplo: C:\Users\Andres\Desktop\Instalación Angular, en la misma dirección ejecutamos un terminal CMD para la instalación del Framework por la siguiente línea de comando: ng new "nombre del proyecto", una vez procesada la instrucción se debe esperar a que finalice la instalación Fig. 60. Para realizar la respectiva prueba del proyecto, por consola se debe ingresar la línea de comando ng serve y probar el proyecto desde el navegador web en la siguiente dirección http://localhost:4200/ Fig. 61.

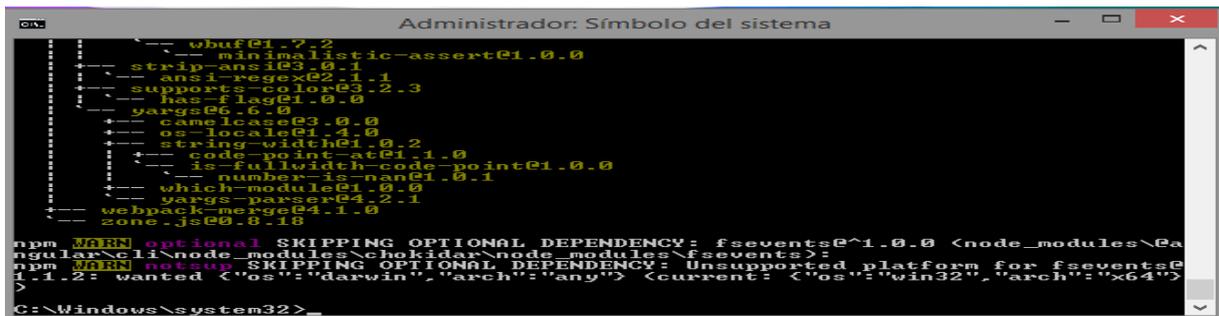


Fig. 60. Instalación de Angular

Fuente: Propia

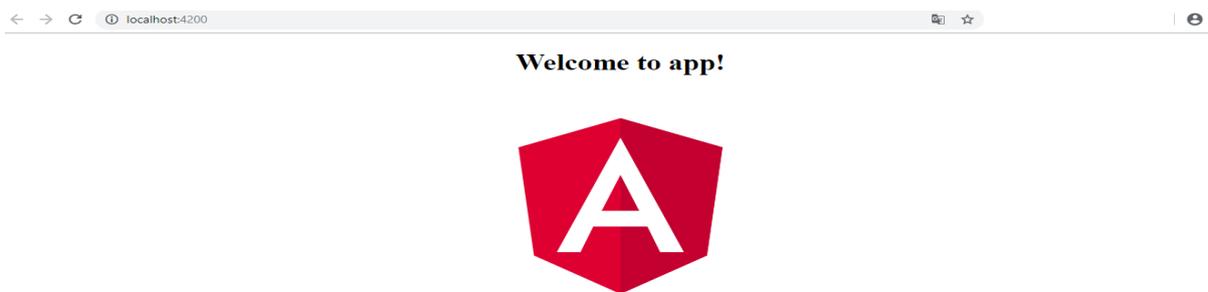


Fig. 61. Iniciar proyecto Angular comando ng serve, ruta http://localhost:4200/

Fuente: Propia

### Tarea Nro. 9.- Creación de la base de datos entidad relación.

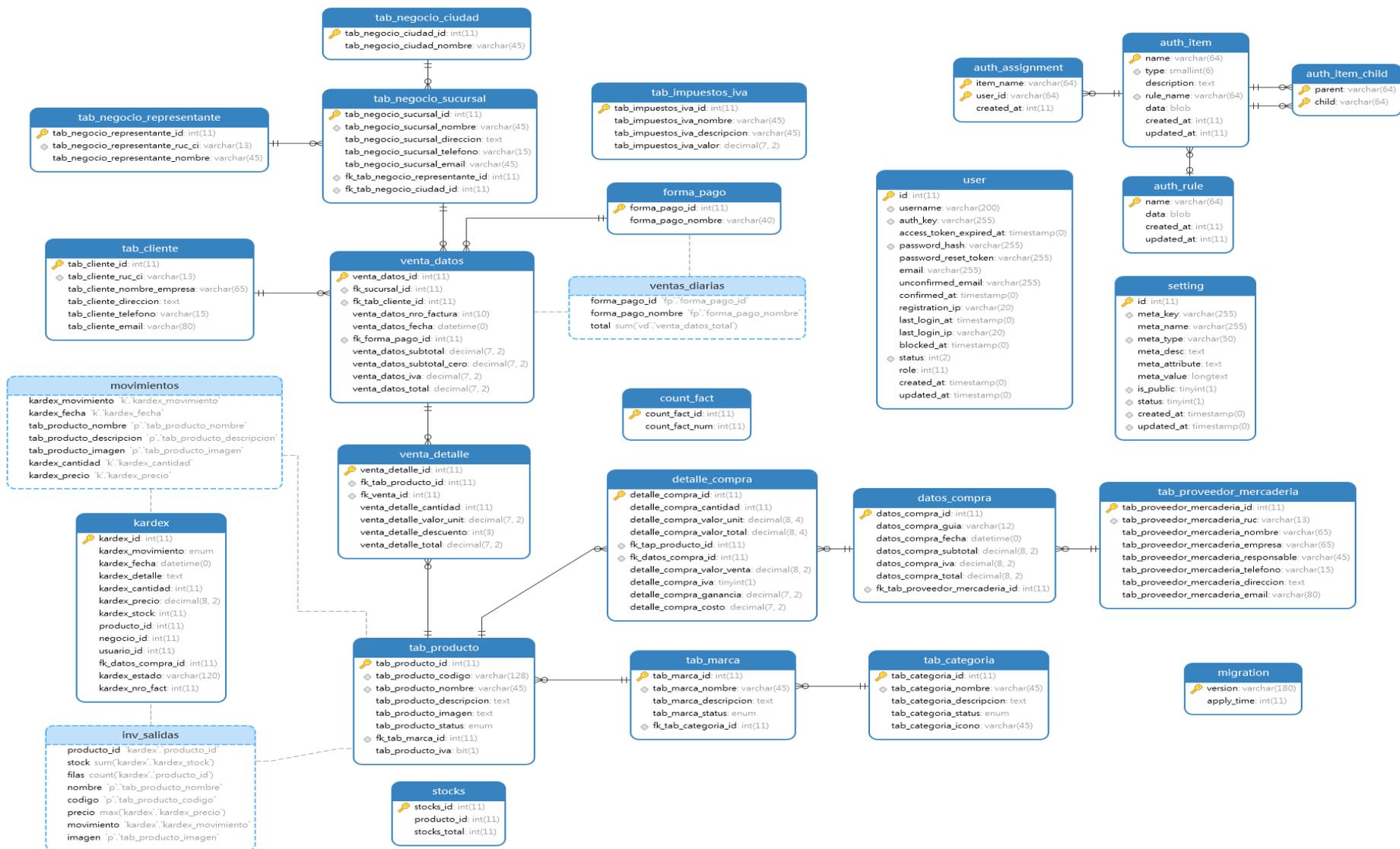


Fig. 62. Diagrama Entidad Relación Base de datos.

Fuente: Propia

Tabla 22. Sprint 1 – Hoja Electrónica

Sprint: 1			
Fecha Inicio: 02 de abril de 2018			
			02 abril
			0
Requerimiento	Tarea	Responsable	Estado
Análisis y Estructuración Del Proyecto	Instalación de Servidor Apache XAMPP	Andrés Gudiño	Terminado
	Instalación de Composer.	Andrés Gudiño	Terminado
	Instalación de Visual Studio Code	Andrés Gudiño	Terminado
	Instalación de Node.js	Andrés Gudiño	Terminado
	Instalación de TypeScript	Andrés Gudiño	Terminado
	Instalación del Framework Yii2 – Proyecto.	Andrés Gudiño	Terminado
	Instalación de Postman	Andrés Gudiño	Terminado
	Instalación del Framework Angular - Proyecto	Andrés Gudiño	Terminado
	Creación de la Base de datos.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 23. Sprint 1 – Pizarrón

Pendiente	En Proceso	Terminada
		Instalación de Servidor Apache XAMPP
		Instalación de Composer.
		Instalación de Visual Studio Code
		Instalación de Node.js
		Instalación de TypeScript
		Instalación del Framework Yii2 – Proyecto.
		Instalación de Postman
		Instalación del Framework Angular - Proyecto
		Creación de la Base de datos.

Fuente: Propia

## 2.6.2. Iteración 2. Creación del Módulo Clientes

En el presente módulo es el encargado de la administración de la información de clientes compradores de la empresa.

**Requerimiento Nro. 2.-** Creación del Módulo Clientes.

**Tarea Nro. 10.-** Creación del servicio REST para el Módulo Clientes Fig. 63.



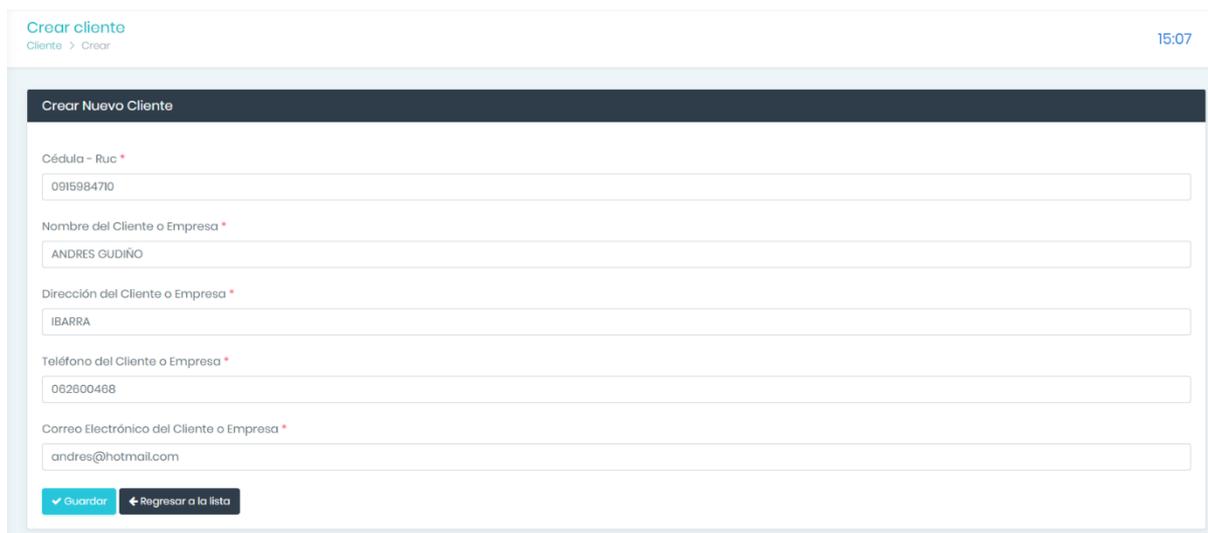
```
servicios.induxion.local/v x
servicios.induxion.local/v1/cliente
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_cliente_id": 1,
      "tab_cliente_ruc_ci": "1000002589",
      "tab_cliente_nombre_empresa": "Pedro",
      "tab_cliente_direccion": "Ibarra",
      "tab_cliente_telefono": "062123456",
      "tab_cliente_email": "mail@mail.com"
    }
  ]
}
```

Fig. 63. Servicio REST cliente.

Fuente: Propia

**Tarea Nro. 11.-** Creación de la interfaz del módulo cliente.

- Para crear un nuevo cliente se presenta la Interfaz necesaria para el registro como se muestra en la Fig. 64.



Crear cliente  
Cliente > Crear 15:07

Crear Nuevo Cliente

Cédula - Ruc \*  
0915984710

Nombre del Cliente o Empresa \*  
ANDRES GUDIÑO

Dirección del Cliente o Empresa \*  
IBARRA

Teléfono del Cliente o Empresa \*  
062800488

Correo Electrónico del Cliente o Empresa \*  
andres@hotmail.com

Guardar Regresar a la lista

Fig. 64. Crear cliente.

Fuente: Propia

- Para actualizar a un cliente es necesario seleccionar a un registro y editar la información como se muestra en la Fig. 65.

Actualizar Cliente

Cédula - Ruc \*

Nombre del Cliente o Empresa \*

Dirección del Cliente o Empresa \*

Teléfono del Cliente o Empresa \*

Correo Electrónico del Cliente o Empresa \*

Fig. 65. Actualizar cliente.

Fuente: Propia

### Tarea Nro. 12.- Lista de clientes.

- En la Fig. 66, se muestra la interfaz para listar todos los registros de clientes almacenados en el sistema.

Cientes [Exportar](#)

[Crear Cliente](#)

Cédula	Nombre	Dirección	Email	Teléfono	Acciones
1000000000	ANDRES GUDIÑO	TRAS DE LA IGLESIA DE SANTO DOMINGO - IBARRA	andresg@gmail.com	062600688	
1003620802	EDUARDO NUÑEZ	IBARRA RIO BLANCO	le456@hotmail.com	062654967	
1025698741	PEDRO R	IBARRA	pedro@hotmail.com	062589741	

Fig. 66. Lista de clientes registrados.

Fuente: Propia

- En la Fig. 67, se muestra Interfaz para buscar un cliente por medio de su cedula o nombre.

Cientes [Exportar](#)

[Crear Cliente](#)

Cédula	Nombre	Dirección	Email	Teléfono	Acciones
<input type="text" value="091598"/>	<input type="text" value="Nombre"/>	<input type="text" value="Dirección"/>	<input type="text" value="Email"/>	<input type="text" value="Teléfono"/>	<input type="text" value="Acciones"/>
0915984710	ANDRES GUDIÑO	IBARRA	andros@hotmail.com	062600468	

Fig. 67. Buscar clientes registrados.

Fuente: Propia

**Tarea Nro. 13.-** Reporte de clientes.

- Para exportar los registros de clientes de la empresa en un formulario de Excel, es necesario dar clic en la opción “Exportar” para tener un reporte de los datos ingresados en el sistema Fig. 68.

	A	B	C	D	E	F	G
2	1	1000000000	ANDRES GUDIÑO	TRAS DE LA IGLESIA DE SANTO DOMINGO - IBARRA	062600688	andresg@gmail.com	
3	2	1003620802	EDUARDO NUÑEZ	IBARRA RIO BLANCO	062654987	le456@hotmail.com	
4	3	1025698741	PEDRO R	IBARRA	062589741	pedro@hotmail.com	
5	4	1003153366	SANTIAGO ROSERO	MOSQUERA NARVÁEZ	062601724	santy12@hotmail.com	

Fig. 68. Reporte de clientes registrados.

Fuente: Propia

Tabla 24. Sprint 2 – Hoja Electrónica

Sprint: 2			
Fecha Inicio: 01 de mayo de 2018			
		Tareas Pendientes	01 mayo
Requerimiento	Tarea	Responsable	Estado
Creación del Módulo Clientes	Creación del servicio REST para el Módulo Clientes.	Andrés Gudiño	Terminado
	Creación de la interfaz Módulo Cliente.	Andrés Gudiño	Terminado
	Lista de clientes.	Andrés Gudiño	Terminado
	Reporte de clientes.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 25. Sprint 2 – Pizarrón

Pendiente	En Proceso	Terminada
		Creación del servicio REST para el módulo clientes.
		Creación de la interfaz módulo cliente
		Lista de Clientes.
		Reporte de Clientes.

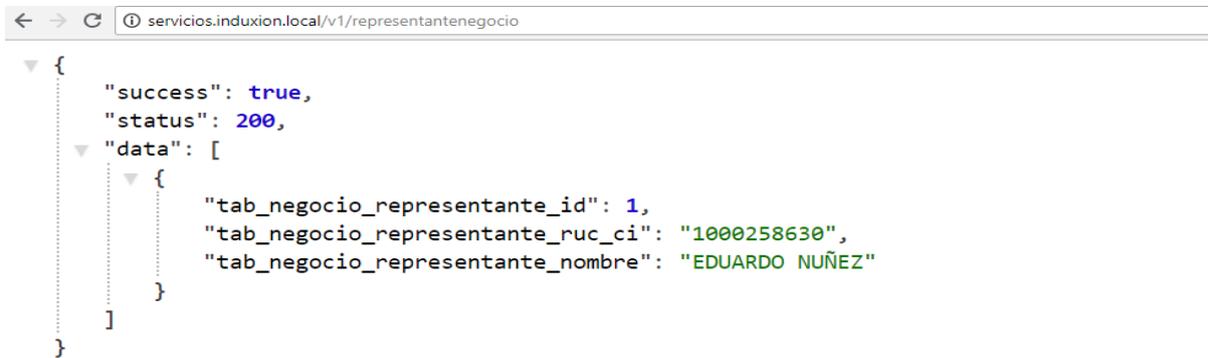
Fuente: Propia

### 2.6.3. Iteración 3. Creación del Módulo Negocio

En el presente módulo es el encargado de la administración de los datos de la empresa con todos sus detalles.

**Requerimiento Nro. 3.-** Creación del Módulo Negocio.

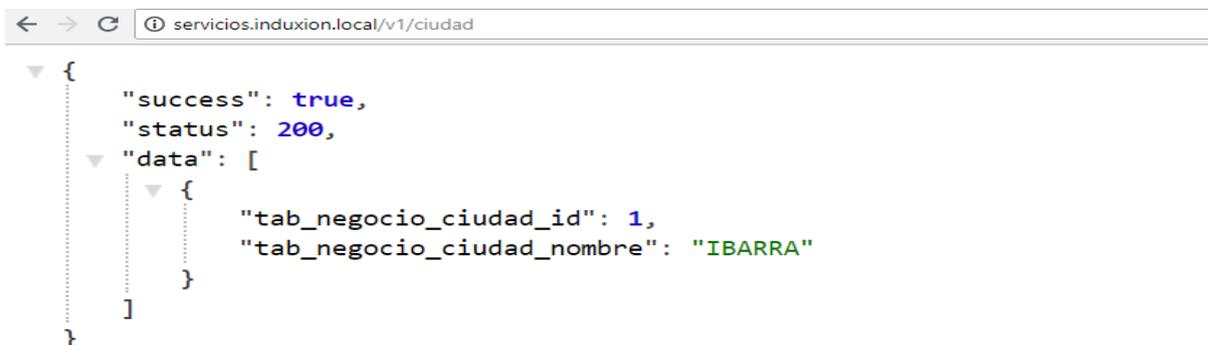
**Tarea Nro. 14.-** Creación del servicio REST para el Módulo Negocio Fig. 69, Fig.70, Fig. 71.



```
← → ↻ servicios.induxion.local/v1/representantenegocio
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_negocio_representante_id": 1,
      "tab_negocio_representante_ruc_ci": "1000258630",
      "tab_negocio_representante_nombre": "EDUARDO NUÑEZ"
    }
  ]
}
```

Fig. 69. Servicio REST representante negocio.

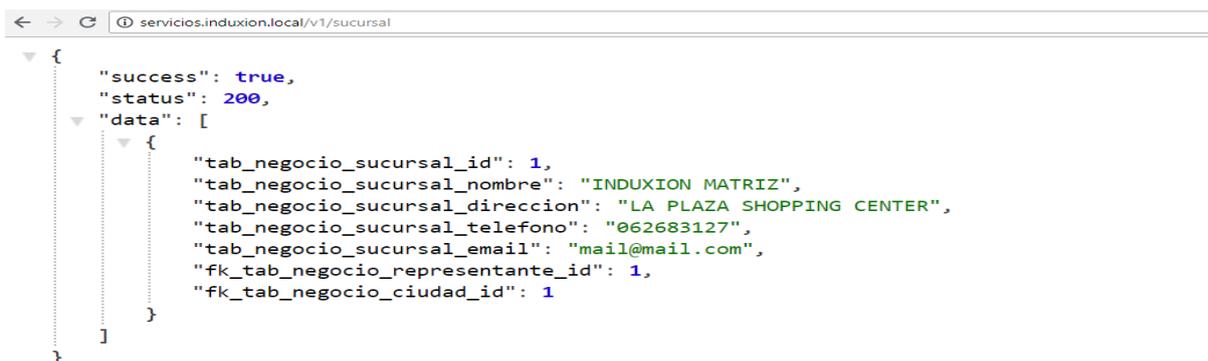
Fuente: Propia



```
← → ↻ servicios.induxion.local/v1/ciudad
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_negocio_ciudad_id": 1,
      "tab_negocio_ciudad_nombre": "IBARRA"
    }
  ]
}
```

Fig. 70. Servicio REST ciudad.

Fuente: Propia



```
← → ↻ servicios.induxion.local/v1/sucursal
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_negocio_sucursal_id": 1,
      "tab_negocio_sucursal_nombre": "INDUXION MATRIZ",
      "tab_negocio_sucursal_direccion": "LA PLAZA SHOPPING CENTER",
      "tab_negocio_sucursal_telefono": "062683127",
      "tab_negocio_sucursal_email": "mail@mail.com",
      "fk_tab_negocio_representante_id": 1,
      "fk_tab_negocio_ciudad_id": 1
    }
  ]
}
```

Fig. 71. Servicio REST sucursal.

Fuente: Propia

### Tarea Nro. 15.- Creación de la interfaz representante legal.

- Para crear un nuevo representante legal de la empresa se presenta la Interfaz necesaria para el registro como se muestra en la Fig. 72.

Crear representante  
Representante > Crear

**Crear Nuevo Representante**

Cédula - Ruc \*  
Ingrese el numerod e cédula o ruc...  
Campo requerido

Nombre del Representante \*  
Ingrese el nombre del Representante...  
Campo requerido

✓ Guardar   ← Regresar a la lista

Fig. 72. Crear representante legal.

Fuente: Propia

- Para actualizar a un representante legal de la empresa es necesario seleccionar a un registro y editar la información como se muestra en la Fig. 73.

Actualizar representante  
Representante > Actualizar

**Actualizar Representante**

Cédula - Ruc \*  
1000258630  
Campo requerido

Nombre del Representante \*  
EDUARDO NUÑEZ  
Campo requerido

✓ Guardar   ← Regresar a la lista

Fig. 73. Actualizar representante legal.

Fuente: Propia

### Tarea Nro. 16.- Listar y buscar el representante legal registrado.

- Para listar los registros de los representantes, se debe situar en la pantalla principal de la sección como muestra en la Fig. 74.

Representantes

+ Crear Representante

Cedula	Nombre	Acciones
Cedula	Nombre	Acciones
1000258630	EDUARDO NUÑEZ	

Fig. 74. Lista del representante legal.

Fuente: Propia

- Para buscar un determinado registro correspondiente a un representante legal, se debe buscar por su cédula o nombre como lo indica la Fig. 75.

Representantes

[+ Crear Representante](#)

Cedula	Nombre	Acciones
<input type="text" value="Cedula"/>	<input type="text" value="eduardo"/>	<input type="text" value="Acciones"/>
1000258630	EDUARDO NUÑEZ	

Fig. 75. Buscar representante legal.

Fuente: Propia

### Tarea Nro. 17.- Creación de la interfaz ciudad.

- Para crear una ciudad se presenta la Interfaz necesaria para el registro como se muestra en la Fig. 76.

Crear ciudad

Ciudad > Crear

**Crear Nueva Ciudad**

Nombre \*

Campo requerido

Fig. 76. Crear ciudad.

Fuente: Propia

- Para actualizar una ciudad es necesario seleccionar un registro y editar la información como se muestra en la Fig. 77.

Actualizar ciudad

Ciudad > Actualizar

**Actualizar Ciudad**

Nombre \*

Fig. 77. Actualizar ciudad.

Fuente: Propia

### Tarea Nro. 18.- Listar, buscar ciudades registradas.

- Para listar los registros de las ciudades, se debe situar en la pantalla principal de la sección como muestra en la Fig. 78.



Ciudad

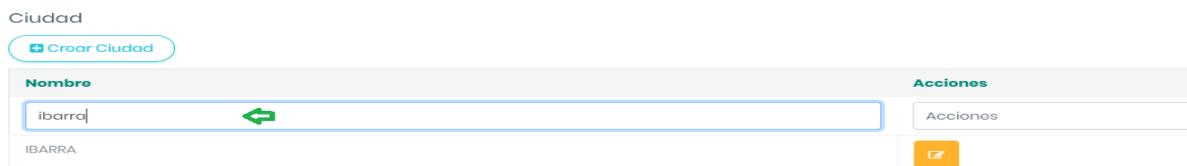
[Crear Ciudad](#)

Nombre	Acciones
Nombre	Acciones
IBARRA	

Fig. 78. Lista de ciudades.

Fuente: Propia

- Para buscar un determinado registro correspondiente a ciudad, se debe buscar por su nombre como lo indica la Fig. 79.



Ciudad

[Crear Ciudad](#)

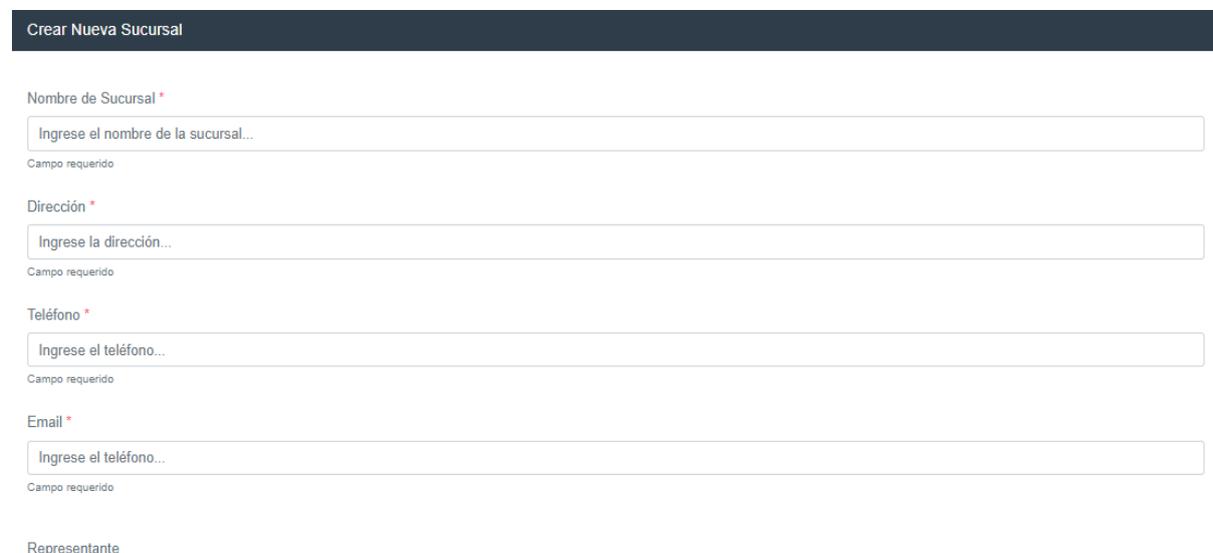
Nombre	Acciones
<input type="text" value="ibarra"/>	Acciones
IBARRA	

Fig. 79. Buscar ciudades.

Fuente: Propia

### Tarea Nro. 19.- Creación de la interfaz tienda.

- Para crear una tienda – sucursal, se presenta la Interfaz necesaria para el registro como se muestra en la Fig. 80.



Crear Nueva Sucursal

Nombre de Sucursal \*

Campo requerido

Dirección \*

Campo requerido

Teléfono \*

Campo requerido

Email \*

Campo requerido

Representante

Fig. 80. Crear tienda.

Fuente: Propia

- Para actualizar la información de una tienda -sucursal, es necesario seleccionar un registro y editar la información como se muestra en la Fig. 81.

**Actualizar Sucursal**

Nombre de Sucursal \*

Dirección \*

Teléfono \*

Email \*

Representante

Ciudad

Fig. 81. Actualizar tienda.

Fuente: Propia

**Tarea Nro. 20.- Listar, buscar.**

- Para listar los registros de las tiendas, se debe situar en la pantalla principal de la sección como muestra en la Fig. 82.

Sucursal

[+ Crear sucursal](#)

Representante	Ciudad	Dirección	Email	Nombre	Teléfono	Acciones
<input type="text" value="Representante"/>	<input type="text" value="Ciudad"/>	<input type="text" value="Dirección"/>	<input type="text" value="Email"/>	<input type="text" value="Nombre"/>	<input type="text" value="Teléfono"/>	<input type="text" value="Acciones"/>
EDUARDO NUÑEZ	IBARRA	AV. MARIANO ACOSTA Y VÍCTOR GOMEZ JURADO	eduardonunez@induxionecuador.com	INDUXION	0998780137	

Fig. 82. Listar tienda.

Fuente: Propia

- Para buscar un determinado registro correspondiente a tienda, se debe buscar por su nombre como lo indica la Fig. 83.

Representante	Ciudad	Dirección	Email	Nombre	Teléfono	Acciones
<input type="text" value="Representante"/>	<input type="text" value="Ciudad"/>	<input type="text" value="Dirección"/>	<input type="text" value="Email"/>	<input type="text" value="INDUXION"/>	<input type="text" value="Teléfono"/>	<input type="text" value="Acciones"/>
EDUARDO NUÑEZ	IBARRA	AV. MARIANO ACOSTA Y VÍCTOR GOMEZ JURADO	eduardonunez@induxionecuador.com	INDUXION	0998780137	

Fig. 83. Listar tienda.

Fuente: Propia

Tabla 26. Sprint 3 – Hoja Electrónica

Sprint: 3			
Fecha Inicio: 16 de mayo de 2018			
Tareas Pendientes			16 de mayo 0
Requerimiento	Tarea	Responsable	Estado
	Creación del servicio REST para el Módulo Negocio.	Andrés Gudiño	Terminado
	Creación de la interfaz representante legal	Andrés Gudiño	Terminado
Creación del Módulo Negocio	Lista y buscar el representante legal registrado.	Andrés Gudiño	Terminado
	Creación de la interfaz ciudad	Andrés Gudiño	Terminado
	Listar, buscar ciudades registradas.	Andrés Gudiño	Terminado
	Creación de la interfaz tienda.	Andrés Gudiño	Terminado
	Listar, buscar.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 27. Sprint 3 – Pizarrón

Pendiente	En Proceso	Terminada
_____	_____	Creación del servicio REST para el módulo Negocio.
_____	_____	Creación de la interfaz representante legal.
_____	_____	Listar y buscar el representante legal registrado.
_____	_____	Creación de la interfaz ciudad.
_____	_____	Listar, buscar ciudades registradas.
_____	_____	Creación de la interfaz tienda.
_____	_____	Listar, buscar.

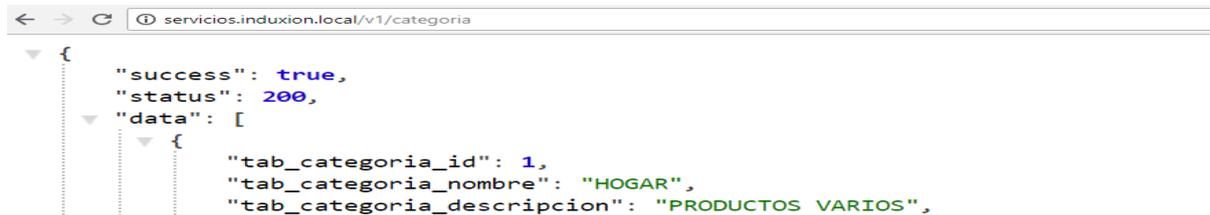
Fuente: Propia

#### 2.6.4. Iteración 4. Creación del Módulo Productos

En el presente módulo es el encargado de la administración de los registros de los productos de la empresa con todos sus detalles.

**Requerimiento Nro. 4.-** Creación del Módulo Productos.

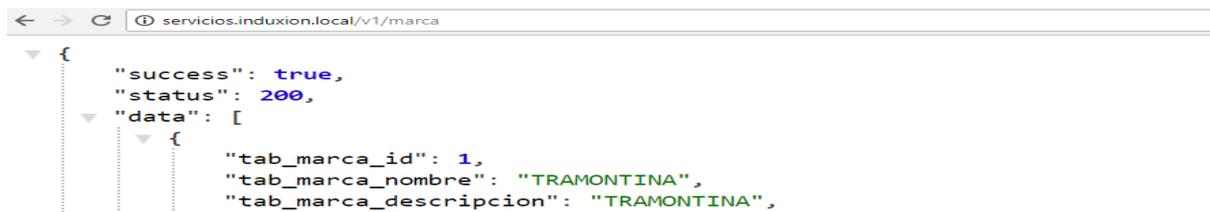
**Tarea Nro. 21.-** Creación del servicio REST para el Módulo Productos Fig.84, Fig.85, Fig. 86.



```
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_categoria_id": 1,
      "tab_categoria_nombre": "HOGAR",
      "tab_categoria_descripcion": "PRODUCTOS VARIOS",
    }
  ]
}
```

Fig. 84. Servicio REST categoría.

Fuente: Propia



```
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_marca_id": 1,
      "tab_marca_nombre": "TRAMONTINA",
      "tab_marca_descripcion": "TRAMONTINA",
    }
  ]
}
```

Fig. 85. Servicio REST marca.

Fuente: Propia



```
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_producto_id": 1,
      "tab_producto_codigo": "TR001",
      "tab_producto_nombre": "OLLA ARROCERA",
    }
  ]
}
```

Fig. 86. Servicio REST producto.

Fuente: Propia

**Tarea Nro. 22.-** Creación de la interfaz categoría.

- Para crear nuevas categorías para las marcas de productos se presenta la Interfaz necesaria como se muestra en la Fig. 87.

**Crear Nueva Categoría de Productos**

Nombre de Categoría \*

Ingrese el nombre de la categoría...

Campo requerido

Descripción \*

Ingrese el nombre de la categoría...

Campo requerido

Estado

Activo

✓ Guardar   ← Regresar a la lista

Fig. 87. Crear categoría de productos.

Fuente: Propia

- Para actualizar una categoría es necesario seleccionar un registro y editar la información como se muestra en la Fig. 88.

**Actualizar Categoría de Productos**

Nombre de Categoría \*

HOGAR

Descripción \*

PRODUCTOS VARIOS

Estado

Activo

✓ Guardar   ← Regresar a la lista

Fig. 88. Actualizar categoría de productos.

Fuente: Propia

**Tarea Nro. 23.-** Listar y buscar registros para categoría.

- Para listar los registros de las categorías, se debe situar en la pantalla principal de la sección como muestra en la Fig. 89.

Categoría Productos

Crear Categoría   Exportar

Nombre	Descripción	Estado	Acciones
Nombre	Descripción	Seleccionar...	Acciones
HOGAR	PRODUCTOS VARIOS	Activo	
DEPORTES	DEPORTES EXTREMOS	Activo	

Fig. 89. Lista categoría de productos.

Fuente: Propia

- Para buscar un determinado registro correspondiente a categorías, se debe buscar por su nombre o descripción como lo indica la Fig. 90.

Categoría Productos

[Crear Categoría](#)

Nombre	Descripción	Estado	Acciones
hogar	Descripción	Seleccionar...	Acciones
HOGAR	PRODUCTOS VARIOS	activo	

Fig. 90. Buscar una categoría de productos.

Fuente: Propia

### Tarea Nro. 24.- Creación de la interfaz marca.

- Para crear una nueva marca de productos se presenta la Interfaz necesaria como se muestra en la Fig. 91.

Crear Nueva Marca de Producto

Marca \*

Campo requerido

Descripción de la Marca \*

Campo requerido

Estado

Categoría

Fig. 91. Crear marca de productos.

Fuente: Propia

- Para actualizar una marca de producto, se debe seleccionar un registro y editar la información como se muestra en la Fig. 92.

Actualizar Marca

Marca \*

Descripción de la Marca \*

Estado

Categoría

Fig. 92. Actualizar marca de productos.

Fuente: Propia

## Tarea Nro. 25.- Listar y buscar registros para marcas.

- Para listar los registros de las marcas de los productos, se debe situar en la pantalla principal de la sección como muestra en la Fig. 93.

Marca Productos

[Crear Marca](#) [Exportar](#)

Marca	Descripción	Estado	Acciones
<input type="text" value="Marca"/>	<input type="text" value="Descripción"/>	Seleccionar ▼	Acciones
TRAMONTINA	MÁS QUE NÚMEROS, LO QUE DEFINE A TRAMONTINA ES EL ESFUERZO PERMANENTE PARA MEJORAR CADA VEZ MÁS LA VIDA DE LA GENTE.	Activo	
ECUACYCLO	DEPORTES EXTREMOS	Activo	

Fig. 93. Lista de marcas de productos.

Fuente: Propia

- Para buscar un determinado registro correspondiente a marcas de productos, se debe buscar por su nombre o descripción como lo indica la Fig. 94.

Marca Productos

[Crear Marca](#) [Exportar](#)

Marca	Descripción	Estado	Acciones
<input type="text" value="Tramontina"/>	<input type="text" value="Descripción"/>	Seleccionar ▼	Acciones
TRAMONTINA	MÁS QUE NÚMEROS, LO QUE DEFINE A TRAMONTINA ES EL ESFUERZO PERMANENTE PARA MEJORAR CADA VEZ MÁS LA VIDA DE LA GENTE.	Activo	

Fig. 94. Buscar marcas de productos.

Fuente: Propia

## Tarea Nro. 26.- Creación de la interfaz producto.

- Para crear un nuevo producto con sus respectivos detalles, se presenta la Interfaz necesaria como se muestra en la Fig. 95.

Crear Nuevo Producto

Código del Producto \*  
  
Campo requerido

Nombre del Producto \*  
  
Campo requerido

Descripción del Producto \*  
  
Campo requerido

Imagen del Producto

Estado

Iva

Marca del Producto

[Guardar](#) [Regresar a la lista](#)

Fig. 95. Crear productos.

Fuente: Propia

- Para actualizar un producto, se debe seleccionar un registro y editar la información como se muestra en la Fig. 96.

Actualizar Producto

Código del Producto \*

Nombre del Producto \*

Descripción del Producto \*

Imagen del Producto

Estado

Iva

Marca del Producto

Fig. 96. Actualizar productos.

Fuente: Propia

**Tarea Nro. 27.-** Listar buscar y reportes de productos.

- Para listar los registros del catálogo de productos, se debe situar en la pantalla principal de la sección como muestra en la Fig. 97.

Productos

[+ Crear Producto](#) [Exportar](#)

Código -	Nombre	Descripción	Marca	Catgoría	Estado	IVA	Imagen	Acciones
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Selecci ▼	Selecci ▼	<input type="text"/>	<input type="text"/>
BI01	BICICLETA KIDS RIDE ARO 12	SIN PEDAL	ECUACYCLO	DEPORTES	Activo	SI		<input type="button" value="✎"/>
TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>
TR02	CUBIERTOS NEW COLOR	42 PZ	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>
TR03	CUBIERTOS BUZIOS	24 PZ	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>
TR04	CUBIERTOS LEME	24 PZ	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>
TR05	CUBIERTOS MALIBU	24 PIEZAS	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>
TR06	CUBIERTOS CARMEL	24 PZ	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>
TR07	CUBIERTOS FAQUEIRO TURQUESA	24 PZ	TRAMONTINA	HOGAR	Activo	SI		<input type="button" value="✎"/>

Fig. 97. Lista de productos.

Fuente: Propia

- Para buscar un determinado producto correspondiente al catálogo, se debe buscar por su código, nombre, descripción, marca o categoría como lo indica la Fig. 98.

Código	Nombre	Descripción	Marca	Categoría	Estado	IVA	Imagen	Acciones
TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	TRAMONTINA	HOGAR	Activo	SI		

Fig. 98. Buscar productos.

Fuente: Propia

- Para exportar los registros del catálogo de productos de la empresa en un formulario de Excel, es necesario dar clic en la opción “Exportar” para tener un reporte de los datos ingresados en el sistema Fig. 99.

	A	B	C	D	E	F
2	BI01	BICICLETA KIDS RIDE ARO 12	SIN PEDAL	ECUACYCLO	DEPORTES	CON IVA data:i
3	TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	TRAMONTINA	HOGAR	CON IVA https:
4	TR02	CUBIERTOS NEW COLOR	42 PZ	TRAMONTINA	HOGAR	CON IVA https:
5	TR03	CUBIERTOS BUZIOS	24 PZ	TRAMONTINA	HOGAR	CON IVA data:i
6	TR04	CUBIERTOS LEME	24 PZ	TRAMONTINA	HOGAR	CON IVA https:
7	TR05	CUBIERTOS MALIBU	24 PIEZAS	TRAMONTINA	HOGAR	CON IVA https:
8	TR06	CUBIERTOS CARMEL	24 PZ	TRAMONTINA	HOGAR	CON IVA https:
9	TR07	CUBIERTOS FAQUEIRO TURQUESA	24 PZ	TRAMONTINA	HOGAR	CON IVA https:
10	TR08	CUBIERTOS FAQUEIRO TOMATE - VERDE	24 PZ	TRAMONTINA	HOGAR	CON IVA https:
11	TR09	CUBIERTOS FAQUEIRO TURQUESA 2	42 PZ	TRAMONTINA	HOGAR	CON IVA https:
12	TR10	CUBIERTOS FAQUEIRO TOMATE	42 PZ	TRAMONTINA	HOGAR	CON IVA https:
13	TR11	CUBIERTOS ANGRA	36 PZ	TRAMONTINA	HOGAR	CON IVA data:i
14	TR12	CUBIERTOS CAJA CUADRADA	16 PZ	TRAMONTINA	HOGAR	CON IVA http://

Fig. 99. Reporte del catálogo de productos.

Fuente: Propia

Tabla 28. Sprint 4 – Hoja Electrónica

Sprint: 4				04 de junio
Fecha Inicio: 04 de junio de 2018				0
Requerimiento	Tarea	Responsable	Estado	
Creación del Módulo Productos	Creación del servicio REST para el Módulo Productos.	Andrés Gudiño	Terminado	
	Creación de la interfaz categoría	Andrés Gudiño	Terminado	
	Listar y buscar registros para categoría.	Andrés Gudiño	Terminado	
	Creación de la interfaz marca.	Andrés Gudiño	Terminado	
	Listar y buscar registros para marcas.	Andrés Gudiño	Terminado	
	Creación de la interfaz producto.	Andrés Gudiño	Terminado	
	Listar buscar y reportes de productos.	Andrés Gudiño	Terminado	

Fuente: Propia

Tabla 29. Sprint 4 – Pizarrón

Pendiente	En Proceso	Terminada
		Creación del servicio REST para el Módulo Productos.
		Creación de la interfaz categoría
		Listar y buscar registros para categoría.
		Creación de la interfaz marca.
		Listar y buscar registros para marcas.
		Creación de la interfaz producto.
		Listar buscar y reportes de productos.

Fuente: Propia

### 2.6.5. Iteración 5. Creación del Módulo Impuestos

El presente módulo es el encargado de la administración de impuestos para productos denominado IVA.

**Requerimiento Nro. 5.-** Creación del Módulo Impuestos.

**Tarea Nro. 28.-** Creación del servicio REST para el Módulo Impuestos Fig. 100.

```

{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_impuestos_iva_id": 1,
      "tab_impuestos_iva_nombre": "IVA 12%"
    }
  ]
}
    
```

Fig. 100. Servicio REST categoría.

Fuente: Propia

**Tarea Nro. 29.-** Crear interfaz impuestos.

- Para actualizar los valores de impuesto IVA, se debe seleccionar un registro y editar la información como se muestra en la Fig. 101.

Fig. 101. Editar impuesto IVA.

Fuente: Propia

**Tarea Nro. 30.-** Listar, buscar registros sobre impuestos IVA.

- Para listar los registros correspondientes a impuestos, se debe situar en la pantalla principal de la sección como muestra en la Fig. 102.



Fig. 102. Lista de impuesto.

Fuente: Propia

- Para buscar un impuesto, se debe buscar por su nombre, descripción o valor como lo indica la Fig. 103.



Fig. 103. Buscar impuesto.

Fuente: Propia

Tabla 30. Sprint 5 – Hoja Electrónica

Sprint: 5			
Fecha Inicio: 25 de junio de 2018			
Requerimiento	Tarea	Tareas Pendientes	25 de junio
		Responsable	0
			Estado
	Creación del servicio REST para el Módulo Impuestos.	Andrés Gudiño	Terminado
Creación del Módulo Impuestos	Creación de la interfaz impuestos IVA.	Andrés Gudiño	Terminado
	Listar, buscar registros sobre impuestos.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 31. Sprint 5 – Pizarrón

Pendiente	En Proceso	Terminada
_____	_____	Creación del servicio REST para el módulo Impuestos.
_____	_____	Creación de la interfaz impuestos
_____	_____	Listar, buscar registros sobre impuestos.

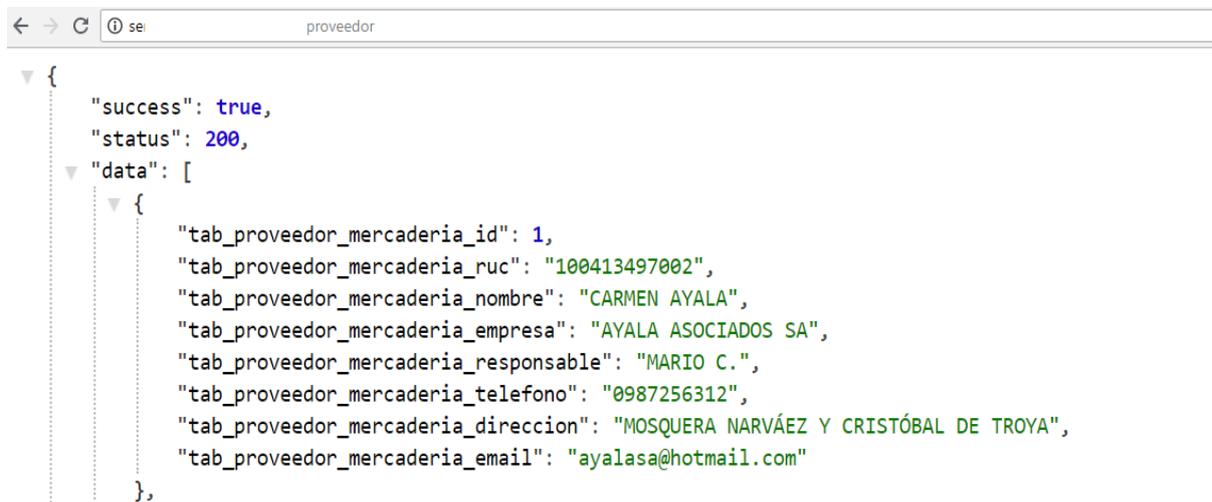
Fuente: Propia

## 2.6.6. Iteración 6. Creación del Módulo Mercadería

El presente módulo es el encargado de la administración del ingreso de mercadería para la empresa.

**Requerimiento Nro. 6.-** Creación del Módulo Mercadería.

**Tarea Nro. 31.-** Creación del servicio REST para el Módulo Mercadería Fig. 104, Fig. 105.



```
{
  "success": true,
  "status": 200,
  "data": [
    {
      "tab_proveedor_mercaderia_id": 1,
      "tab_proveedor_mercaderia_ruc": "100413497002",
      "tab_proveedor_mercaderia_nombre": "CARMEN AYALA",
      "tab_proveedor_mercaderia_empresa": "AYALA ASOCIADOS SA",
      "tab_proveedor_mercaderia_responsable": "MARIO C.",
      "tab_proveedor_mercaderia_telefono": "0987256312",
      "tab_proveedor_mercaderia_direccion": "MOSQUERA NARVÁEZ Y CRISTÓBAL DE TROYA",
      "tab_proveedor_mercaderia_email": "ayalasa@hotmail.com"
    }
  ]
}
```

Fig. 104. Servicio REST proveedor.

Fuente: Propia



```
{
  "success": true,
  "status": 200,
  "data": {
    "error": false,
    "code": "200",
    "compras": [
      {
        "datos_compra_id": "58",
        "datos_compra_guia": "12222",
        "datos_compra_fecha": "2018-08-01 10:10:18",
      }
    ]
  }
}
```

Fig. 105. Servicio REST compra.

Fuente: Propia

**Tarea Nro. 32.-** Creación de la interfaz proveedores.

- Para crear un nuevo proveedor de mercadería con sus respectivos detalles, se presenta la Interfaz necesaria como se muestra en la Fig. 106.

**Crear Nuevo Proveedor**

Ruc Proveedor \*  
Ingrese el RUC del Proveedor...  
Campo requerido

Nombre Proveedor \*  
Ingrese el Nombre del Proveedor...  
Campo requerido

Empresa \*  
Ingrese la empresa...  
Campo requerido

Responsable \*  
Ingrese el responsable...  
Campo requerido

Telefono \*  
Ingrese el numero de telefono...  
Campo requerido

Dirección \*  
Ingrese el numero de telefono...  
Campo requerido

Email \*  
Ingrese el numero de telefono...  
Campo requerido

[✓ Guardar](#) [← Regresar a la lista](#)

Fig. 106. Crear proveedor.

Fuente: Propia

- Para actualizar un proveedor, se debe seleccionar un registro y editar la información como se muestra en la Fig. 107.

**Actualizar Proveedor**

Ruc Proveedor \*  
100413497002

Nombre Proveedor \*  
CARMEN AYALA

Empresa \*  
AYALA ASOCIADOS SA

Responsable \*  
MARIO C.

Telefono \*  
0987256312

Dirección \*  
MOSQUERA NARVÁEZ Y CRISTÓBAL DE TROYA

Email \*  
ayalasa@hotmail.com

[✓ Guardar](#) [← Regresar a la lista](#)

Fig. 107. Actualizar proveedor.

Fuente: Propia

### Tarea Nro. 33.- Listar, buscar registros de proveedores.

- Para listar los registros correspondientes a proveedores, se debe situar en la pantalla principal de la sección como muestra en la Fig. 108.

Proveedores [Crear Proveedor](#) [Export](#)

Ruc Proveedor	Nombre Dueño	Empresa	Responsable	Teléfono	Dirección	Email	Acciones
Ruc Proveedor	Nombre Dueño	Empresa	Responsable	Teléfono	Dirección	Email	Acciones
100413497002	CARMEN AYALA	AYALA ASOCIADOS SA	MARIO C.	0987256312	MOSQUERA NARVÁEZ Y CRISTÓBAL DE TROYA	ayalasa@hotmail.com	
1002385400	IBERO SA	IBERO SA	CARLOS A.	026987236	IBARRA	iberosa@hotmail.com	

Fig. 108. Lista de proveedor.

Fuente: Propia

- Para buscar un proveedor, se debe buscar por su ruc, nombre, empresa, responsable, teléfono, dirección o email como lo indica la Fig. 109.

Proveedores

[Crear Proveedor](#) [Export](#)

Ruc Proveedor	Nombre Dueño	Empresa	Responsable	Teléfono	Dirección	Email	Acciones
<input type="text" value="Ruc Proveedor"/>	<input type="text" value="PRO"/>	<input type="text" value="Empresa"/>	<input type="text" value="Responsable"/>	<input type="text" value="Teléfono"/>	<input type="text" value="Dirección"/>	<input type="text" value="Email"/>	<input type="text" value="Acciones"/>
1000002200001	PRO	PRODIS PO	JAIRO	022365412	IBARRA	prodispo@hotmail.com	

Fig. 109. Buscar proveedor

Fuente: Propia

### Tarea Nro. 34.- Crear interfaz de compras de productos.

- Para el registro de las compras realizadas por la empresa para el respectivo ingreso de mercadería con sus detalles, se presenta la Interfaz necesaria como se indica en la Fig. 110.

Compras > Proceso de adquisición de productos

Proveedores

Seleccione el proveedor.

PRODIST - 1000002200001

Detalle de compras a Proveedores

tr01

Código: 456998 Fecha: dd/mm/aaaa

Cédula: 1000002200001 Nombres: PRODIST Email: prodispo@hotmail.com

#	Producto	Cantidad	Valor Unit	IVA	Costo	Utilidad %	Valor Venta	Total	Acciones
1	TR01	1	15	SI	16.8	200	33.6	16.8	

TR01 - CUBIERTOS NEW COLOR 24 PZ  
15,00 \$  
IVA CON IVA

Fig. 110. Ingreso de compras.

Fuente: Propia

### Tarea Nro. 35.- Listar, buscar registros de compras ingresadas en la empresa.

- Para listar y buscar los registros almacenados en el sistema pertenecientes a las compras registradas, se presenta muestra la interfaz Fig. 111.

Compras

[Exportar](#)

Nro Guía	Fecha Ingreso	Proveedor	Total	Estado -	Usuario Ingreso	Fecha Actualización	Usuario Actualización	Acciones
<input type="text"/>	<input type="text" value="Fecha Ingreso"/>	<input type="text" value="Proveedor"/>	<input type="text" value="Total"/>	<input type="text" value="Selecci"/>	<input type="text" value="Usuario Ingreso"/>	<input type="text" value="Fecha Actualiza"/>	<input type="text" value="Usuario Actualiz"/>	<input type="text" value="Acciones"/>
104	2018-10-21 15:54:02	PRO	\$1,657.85	REGISTRADA	eduardo	2018-10-21 15:54:03		
103	2018-10-21 15:47:53	PRO	\$101.31	REGISTRADA	eduardo	2018-10-21 15:47:53		
101	2018-10-20 20:07:51	PRO	\$115.55	REGISTRADA	eduardo	2018-10-20 20:07:51		

Fig. 111. Listar y buscar compras para el ingreso de mercadería.

Fuente: Propia

- Los detalles de una entrada de productos – compras muestra en la Fig. 112.

Compra Detalle  
Compras > Detalles

### REGISTRADA

**PROVEEDOR:** PRODIST **NRO GUIA:** 101  
**RUC:** 1000002200001 **FECHA:** 2018-10-20 20:07:51  
**TELEFONO:** 022365412 **COMPRA POR:** eduardo  
**EMAIL:** prodispo@hotmail.com

Detalles:

#	Código	Nombre	Cant	Valor Unit	IVA	Costo	Ganancia	Precio Venta	Total
1	TR01	CUBIERTOS NEW COLOR 24 PZ	15	6.8780	Si	7.7034	194.7000 %	15.0000	115,55 \$
Total									115,55 \$

[Anular compra](#) [Imprimir](#)

Fig. 112. Registro de una entrada de productos - compras.

Fuente: Propia

- Los detalles de una entrada anulada en su totalidad – compras se muestra en la Fig. 113.

Compra Detalle  
Compras > Detalles

### ANULADA

**PROVEEDOR:** PRODIST **NRO GUIA:** 102  
**RUC:** 1000002200001 **FECHA:** 2018-10-20 20:13:54  
**TELEFONO:** 022365412 **COMPRA POR:** eduardo  
**EMAIL:** prodispo@hotmail.com

Detalles:

#	Código	Nombre	Cant	Valor Unit	IVA	Costo	Ganancia	Precio Venta	Total
1	TR02	CUBIERTOS NEW COLOR	2	9.0458	Si	10.1313	207.3000 %	21.0000	20,26 \$
Total									20,26 \$

Fig. 113. Registro de una entrada anulada de productos - compras.

Fuente: Propia

Tabla 32. Sprint 6 – Hoja Electrónica

Sprint: 6 Fecha Inicio: 02 de julio de 2018			
Requerimiento	Tarea	Tareas Pendientes Responsable	02 de julio 0 Estado
	Creación del servicio REST para el Módulo Mercadería.	Andrés Gudiño	Terminado
	Creación de la interfaz proveedores.	Andrés Gudiño	Terminado
Creación del Módulo Mercadería	Listar, buscar registros de proveedores.	Andrés Gudiño	Terminado
	Crear interfaz de compras de productos.	Andrés Gudiño	Terminado
	Listar, buscar registros de compras ingresadas en la empresa.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 33. Sprint 6 – Pizarrón

Pendiente	En Proceso	Terminada
_____	_____	Creación del servicio REST para el módulo Mercadería.
_____	_____	Creación de la interfaz proveedores.
_____	_____	Listar, buscar registros de proveedores.
_____	_____	Crear interfaz de compras de productos.
_____	_____	Listar, buscar registros de compras ingresadas en la empresa.
_____	_____	Creación del servicio REST para el módulo Mercadería.
_____	_____	Creación de la interfaz proveedores.

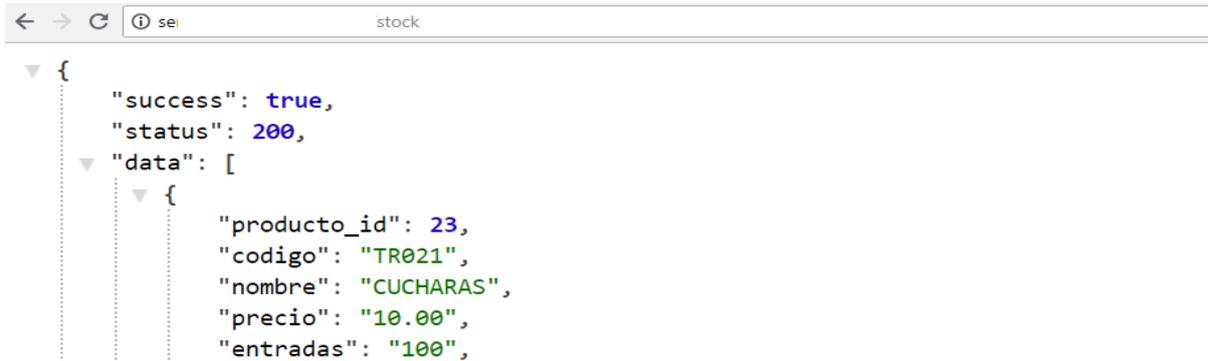
Fuente: Propia

### 2.6.7. Iteración 7. Creación del Módulo Stock

El presente módulo es el encargado de informar las existencias de productos.

**Requerimiento Nro. 7.-** Creación del Módulo Stock.

**Tarea Nro. 36.-** Creación del servicio REST para el Módulo Stock Fig. 114.



```
{
  "success": true,
  "status": 200,
  "data": [
    {
      "producto_id": 23,
      "codigo": "TR021",
      "nombre": "CUCHARAS",
      "precio": "10.00",
      "entradas": "100",
    }
  ]
}
```

Fig. 114. Servicio REST stock de productos.

Fuente: Propia

**Tarea Nro. 37.-** Creación de la interfaz Stock.

- Para listar, buscar y reportes de stock de productos se presenta la Interfaz para verificar las existencias de productos que presenta la empresa, buscar productos por código, nombre como lo indica en la Fig. 115. Adicionalmente se puede exportar en un archivo de Excel para adquirir la información en digital correspondiente al stock disponible por la empresa como se indica en la Fig. 116.

Stock de Productos



Código	Nombre	Descripción	Marca	Categoría	IVA	Precio de Venta	Stock	Imagen
BI01	BICICLETA KIDS RIDE ARO 12	SIN PEDAL	ECUACYCLO	DEPORTES	SI	\$50.00	7	
TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	TRAMONTINA	HOGAR	SI	\$15.00	6	
TR02	CUBIERTOS NEW COLOR	42 PZ	TRAMONTINA	HOGAR	SI	\$21.00	8	
TR03	CUBIERTOS BUZIOS	24 PZ	TRAMONTINA	HOGAR	SI	\$20.50	6	
TR04	CUBIERTOS LEME	24 PZ	TRAMONTINA	HOGAR	SI	\$19.00	6	
TR05	CUBIERTOS MALIBU	24 PIEZAS	TRAMONTINA	HOGAR	SI	\$20.00	7	

Fig. 115. Listar, buscar productos con su respectivo stock.

Fuente: Propia

1	codigo	nombre	descripcion	marca	categoria	precio	iva	entradas	salidas	stock
2	TR13	CUBIERTOS PROFESIONAL	24 PZ	TRAMONTINA	HOGAR	41.00	SI	19	2	17
3	TR12	CUBIERTOS CAJA CUADRADA	16 PZ	TRAMONTINA	HOGAR	27.00	SI	11	0	11
4	TR02	CUBIERTOS NEW COLOR	42 PZ	TRAMONTINA	HOGAR	21.00	SI	10	2	8
5	TR05	CUBIERTOS MALIBU	24 PIEZAS	TRAMONTINA	HOGAR	20.00	SI	11	4	7
6	TR06	CUBIERTOS CARMEL	24 PZ	TRAMONTINA	HOGAR	26.00	SI	9	2	7

Fig. 116. Reporte de productos con su respectivo stock.

Fuente: Propia

Tabla 34. Sprint 7 – Hoja Electrónica

Sprint: 7				25 de julio
Fecha Inicio: 25 de julio de 2018				0
Requerimiento	Tarea	Responsable	Estado	Tareas Pendientes
Creación del Módulo Stock	Creación del servicio REST para el Módulo Stock.	Andrés Gudiño	Terminado	
	Creación de la interfaz Stock.	Andrés Gudiño	Terminado	

Fuente: Propia

Tabla 35. Sprint 7 – Pizarrón

Pendiente	En Proceso	Terminada
_____	_____	Creación del servicio REST para el módulo Stock.
_____	_____	Creación de la interfaz Stock.

Fuente: Propia

### 2.6.8. Iteración 8. Creación del Módulo Ventas

El presente módulo es el encargado de registrar las todas las salidas de mercadería, ventas realizadas por la empresa asignando los productos y clientes.

**Requerimiento Nro. 8.-** Creación del Módulo Ventas.

**Tarea Nro. 38.-** Creación del servicio REST para el Módulo Ventas Fig. 117.

```

{
  "success": true,
  "status": 200,
  "data": {
    "error": false,
    "code": "200",
    "compras": [
      {
        "datos_compra_id": "46",
        "datos_compra_guia": "100",
        "datos_compra_fecha": "2018-07-29 14:42:19",
      }
    ]
  }
}

```

Fig. 117. Servicio REST ventas.

Fuente: Propia

**Tarea Nro. 39.-** Crear interfaz de ventas de productos.

- Para el registro de salidas de productos como son las ventas de la empresa, se presenta la Interfaz necesaria como se indica en la Fig. 118.

**Cientes**

Seleccione el cliente.

1003620802 - EDUARDO NUÑEZ      1025698741 - PEDRO R

---

**Detalle de Ventas a Clientes**

tr13

**TR13 - CUBIERTOS PROFESIONAL**  
41,00 \$  
Stock: 17  
IVA: SI

Nro Fact: 1002556      Fecha: 13/11/2018 11:07      Forma Pago: Efectivo

Cédula: 1025698741      Nombres: PEDRO R      Dirección: IBARRA

#	Producto	Cantidad	Valor Unit	IVA	Total	Acciones
1	TR13	1	41	SI	41	✖ + -

Subtotal \$: 36,61  
Subtotal 0 %: 0  
IVA 12%: 4,39  
Total: 41

Guardar

Fig. 118. Registro de salida de productos - ventas.

Fuente: Propia



- Los detalles de una salida de producto anulada en su totalidad – ventas, se muestra en la Fig. 122.

Venta detalle  
Ventas > Detalles

### ANULADA

CLIENTE: ANDRES GUDIÑO      NRO: 10003  
 RUC: 1000000000      FECHA: 2018-10-20 20:16:00  
 TELEFONO: 062600688      VENTA POR: eduardo  
 DIRECCION: TRAS DE LA IGLESIA DE SANTO DOMINGO - IBARRA

**Detalles:**

#	Código	Nombre	Cant	Valor Unit	Total	
1	TR01	CUBIERTOS NEW COLOR 24 PZ	1	15.00	15,00 S	
					SubTotal 12%	13,39 S
					SubTotal 0%	0,00 S
					Iva	1,61 S
					Total	15,00 S

Fig. 122. Detalles salida de productos anuladas en su totalidad - venta.

Fuente: Propia

- La Fig. 123, indica el reporte adquirido desde el sistema correspondiente al reporte de salida de productos.

Autoguardado lista\_venta\_export\_1542036516016.xlsx - Excel      ANDRES RUBEN GUDIÑO QUINTEROS

Archivo Inicio Insertar Diseño de página Fórmulas Datos Revisar Vista Ayuda Equipo ¿Qué desea hacer?

Portapapeles Fuente Alineación Número Estilos Celdas Edición

	A	B	C	D	E	F	G	H	I	J
1	inro	fecha	cedula	nombre	tipo	total	estado	username	kardex_fecha_actualizacion	usuario_modificacion
2	658311	2018-11-07 12:58:27	1003153366	SANTIAGO ROSERO	Efectivo	15.00	REGISTRADA	eduardo	2018-11-07 11:58:27	
3	4556	2018-10-28 19:23:58	1025698741	PEDRO R	Tarjetas	19.00	REGISTRADA	vendedor	2018-10-28 19:23:58	
4	653289	2018-10-28 06:27:14	1025698741	PEDRO R	Efectivo	30.50	REGISTRADA	vendedor	2018-10-28 06:27:14	
5	4566699	2018-10-27 21:10:41	1003620802	EDUARDO NUÑEZ	Efectivo	41.00	REGISTRADA	vendedor	2018-10-27 21:10:41	
6	885555	2018-10-27 15:15:56	1000000000	ANDRES GUDIÑO	Tarjetas	20.00	ANULADA	vendedor	2018-10-27 21:11:23	vendedor

Fig. 123. Reporte de salida de productos - ventas.

Fuente: Propia

- La Fig. 124 y Fig. 125, muestran las recaudaciones y reportes procesador por las ventas realizadas.

Total Recaudado Por Fechas:

[Exportar](#)

Fecha	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
Fecha	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
2018-11-21	0.00	26.00	0.00	0.00	\$26.00
2018-11-19	20.00	36.60	0.00	0.00	\$56.60
2018-11-13	0.00	106.40	0.00	0.00	\$106.40

Fig. 124. Recaudaciones empresa.

Fuente: Propia

Autoguardado recaudado\_export\_1542039098336.xlsx - Excel

Archivo Inicio Insertar Diseño de página Fórmulas Datos Revisar Vista Ayuda Equipo ¿Qué desea hacer?

Calibri 12 A<sup>+</sup> A<sup>-</sup> Ajustar texto General

Pegar Fuente Alineación Número Estilos

	A	B	C	D	E	F	G	H	I	J
1	mes	efectivo	tarjetas	dinero_ele	otros	total				
2	2018-11-07	15.00	0.00	0.00	0.00	15.00				
3	2018-10-28	30.50	19.00	0.00	0.00	49.50				
4	2018-10-27	41.00	0.00	0.00	0.00	41.00				
5	2018-10-26	0.00	19.00	61.50	0.00	80.50				
6	2018-10-25	0.00	57.00	45.00	0.00	102.00				
7	2018-10-24	92.00	0.00	35.00	15.00	142.00				
8	2018-10-23	0.00	0.00	54.50	80.00	134.50				
9	2018-10-22	30.50	21.00	0.00	0.00	51.50				
10	2018-10-21	15.00	196.00	0.00	0.00	211.00				
11	2018-10-20	15.00	30.00	0.00	0.00	45.00				

Fig. 125. Reporte del total recaudado.

Fuente: Propia

Tabla 36. Sprint 8 – Hoja Electrónica

Sprint: 8				
Fecha Inicio: 30 de julio de 2018				
				30 de julio
				Tareas Pendientes
				0
Requerimiento	Tarea	Responsable	Estado	
	Creación del servicio REST para el Módulo Ventas.	Andrés Gudiño	Terminado	
Creación del Módulo Ventas	Crear interfaz de ventas de productos.	Andrés Gudiño	Terminado	
	Listar, buscar registros de ventas registradas o anuladas.	Andrés Gudiño	Terminado	

Fuente: Propia

Tabla 37. Sprint 8 – Pizarrón

Pendiente	En Proceso	Terminada
_____	_____	Creación del servicio REST para el módulo Ventas.
_____	_____	Crear interfaz de ventas de productos.
_____	_____	Listar, buscar registros de ventas registradas o anuladas en su totalidad por la empresa.

Fuente: Propia

### 2.6.9. Iteración 9. Creación del Módulo Kardex

El presente módulo es el encargado de informar a la empresa las entradas y salidas de productos.

**Requerimiento Nro. 9.-** Creación del Módulo Kardex.

**Tarea Nro. 41.-** Creación del servicio REST para el Módulo Kardex Fig. 126.

```

< --> C @ sei /movimientos
{
  "success": true,
  "status": 200,
  "data": [
    {
      "kardex_movimiento": "SALIDA",
      "kardex_fecha": "2018-08-01 10:13:55",
      "tab_producto_nombre": "OLLA ARROCERA",
      "tab_producto_descripcion": "TAPA DE ALUMINIO CON REVESTIMIENTO INTERNO 26CM",
    }
  ]
}

```

Fig. 126. Servicio REST Kardex.

Fuente: Propia

**Tarea Nro. 42.-** Creación de la interfaz Kardex.

- El siguiente módulo presenta las opciones como: listar, buscar y reportes de movimientos de productos, adicionalmente se puede verificar las entradas y salidas de productos de la empresa, buscar productos por código, nombre y exportar en un archivo de Excel como se muestran en la Fig. 127 y Fig. 128.

Movimientos [Exportar](#)

Nro	Fecha Ingreso	Código	Nombre	Descripción	Cantidad	Movimiento	Imagen	Estado	Usuario Ingreso	Fecha Actualización	Usuario Actualización
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	2018-10-21 15:54:03	TR09	CUBIERTOS FAQUEIRO TURQUESA 2	42 PZ	8	ENTRADA		Entrada Compra	eduardo	2018-10-21 15:54:03	
3	2018-10-21 15:47:53	TR02	CUBIERTOS NEW COLOR	42 PZ	10	ENTRADA		Entrada Compra	eduardo	2018-10-21 15:47:53	
4	2018-10-21 00:42:15	TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	1	SALIDA		Salida Venta	eduardo	2018-10-21 00:42:15	
3	2018-10-20 20:16:00	TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	0	SALIDA		Anulación Venta	eduardo	2018-10-21 00:09:39	eduardo
2	2018-10-20 20:13:54	TR02	CUBIERTOS NEW COLOR	42 PZ	0	ENTRADA		Anulación Compra	eduardo	2018-10-20 20:14:09	eduardo
2	2018-10-20 20:11:33	TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	2	SALIDA		Salida Venta	eduardo	2018-10-20 20:11:33	
1	2018-10-20 20:09:01	TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	1	SALIDA		Salida Venta	eduardo	2018-10-20 20:09:01	
1	2018-10-20 20:07:51	TR01	CUBIERTOS NEW COLOR 24 PZ	23199/162	15	ENTRADA		Entrada Compra	eduardo	2018-10-20 20:07:51	

Fig. 127. Listar, buscar entradas y salidas de productos

Fuente: Propia

	A	B	C	D	E	F	G	H	I	J	K	L
2	SALIDA	2018-11-07 11:58:27	TR01	CUBIERTOS NEW C23199/162	https://http	1	15.00	Salida Compra	eduardo			2018-11-07 11:58:27
3	SALIDA	2018-10-28 19:23:58	TR04	CUBIERTOS LEME 24 PZ	https://tecl	1	19.00	Salida Compra	vendedor			2018-10-28 19:23:58
4	SALIDA	2018-10-28 06:27:14	TR08	CUBIERTOS FAQUE24 PZ	https://http	1	30.50	Salida Compra	vendedor			2018-10-28 06:27:14
5	SALIDA	2018-10-27 21:10:41	TR13	CUBIERTOS PROFE24 PZ	http://imgc	1	41.00	Salida Compra	vendedor			2018-10-27 21:10:41
6	ENTRADA	2018-10-27 15:19:01	TR17	CUCHARAS MEDID 5 PZAS	http://suip.	0	0.00	Anulacion compra	eduardo	eduardo		2018-10-27 15:20:06
7	SALIDA	2018-10-27 15:15:56	TR05	CUBIERTOS MALIB 24 PIEZAS	https://http	0	0.00	Anulacion venta	vendedor	vendedor		2018-10-27 21:11:23
8	SALIDA	2018-10-27 15:14:39	TR02	CUBIERTOS NEW C42 PZ	https://ww	0	0.00	Anulacion venta	vendedor	eduardo		2018-10-27 15:15:07
9	SALIDA	2018-10-27 15:12:03	TR02	CUBIERTOS NEW C42 PZ	https://ww	0	0.00	Anulacion venta	eduardo	eduardo		2018-10-27 15:12:47
10	SALIDA	2018-10-27 14:47:21	TR05	CUBIERTOS MALIB 24 PIEZAS	https://http	0	0.00	Anulacion venta	eduardo	eduardo		2018-10-27 14:54:21
11	ENTRADA	2018-10-27 14:35:29	TR05	CUBIERTOS MALIB 24 PIEZAS	https://http	0	0.00	Anulacion compra	eduardo	eduardo		2018-10-27 14:45:16
12	SALIDA	2018-10-26 04:19:52	TR02	CUBIERTOS NEW C42 PZ	https://ww	0	0.00	Anulacion venta	eduardo	eduardo		2018-10-26 04:24:33

Fig. 128. Reporte de entradas y salidas de productos

Fuente: Propia

Tabla 38. Sprint 9 – Hoja Electrónica

Sprint: 9			
Fecha Inicio: 27 de agosto de 2018			
Requerimiento	Tarea	Responsable	Estado
			27 de agosto
		Tareas Pendientes	0
Creación del Módulo Kardex	Creación del servicio REST para el módulo Kardex.	Andrés Gudiño	Terminado
	Creación de la interfaz Kardex.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 39. Sprint 9 – Pizarrón

Pendiente	En Proceso	Terminada
		Creación del servicio REST para el módulo Kardex.
		Creación de la interfaz Kardex.

Fuente: Propia

### 2.6.10. Iteración 10. Módulo Seguridad y Control

El presente módulo es el encargado de la administración de usuarios para el ingreso al sistema especificando su rol.

**Requerimiento Nro. 10.-** Creación del Módulo Seguridad y Control.

**Tarea Nro. 43.-** Creación del servicio REST para el Módulo de Seguridad y Control Fig. 129.



```
{
  "success": true,
  "status": 200,
  "data": [
    {
      "id": 3,
      "username": "user",
      "email": "user@.com",
      "unconfirmed_email": "user@.com",
    }
  ]
}
```

Fig. 129. Servicio REST usuarios

Fuente: Propia

**Tarea Nro. 44.-** Creación de la interfaz para la administración de usuarios con su respectivo rol.

- La Fig. 130, es la Interfaz de usuarios para crear, actualizar, asignar roles a usuarios y buscar.

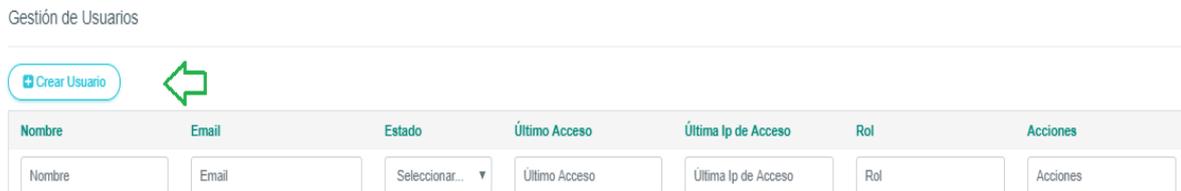


Fig. 130. Gestión de usuarios para el ingreso al sistema

Fuente: Propia

- Crear usuarios del sistema es necesario registrar toda la información como lo indica la Fig. 131.



**Crear Nuevo Usuario**

Username \*

\* El nombre de usuario solo permite caracteres alfanuméricos, guiones bajos y paneles entre 3 y 15 caracteres. es decir, ejemplousername

Email \*

\* El correo electrónico debe ser una dirección de correo electrónico válida. es decir, ejemplo@dominio.com.

Contraseña

\* La contraseña debe tener más de 6 caracteres.

Fig. 131. Registro de usuarios para el ingreso al sistema

Fuente: Propia

- Para listar los registros correspondientes a usuarios del sistema, se debe situar en la pantalla principal de la sección como muestra en la Fig. 132.

Gestión de Usuarios

[+ Crear Usuario](#)

Nombre	Email	Estado	Último Acceso	Última Ip de Acceso	Rol	Acciones
<input type="text" value="Nombre"/>	<input type="text" value="Email"/>	Selección ▾	<input type="text" value="Último Acceso"/>	<input type="text" value="Última Ip de Acceso"/>	<input type="text" value="Rol"/>	<input type="text" value="Acciones"/>
eduardo	edu12@hotmail.com	Activo	2018-11-12 10:00:38	186. . . . .	Administrador	
vendedor	vendedor@induxionecuador.com	Activo	2018-11-07 13:02:18	181. . . . .	Vendedor	
vendedor1	vendedor1@induxionecuador.com	Bloqueado	2018-10-21 16:15:10		Vendedor	

Fig. 132. Lista de usuarios del sistema.

Fuente: Propia

- La Fig. 133, indica el panel principal para los Administradores.

**Total Recaudado Por Fechas:**

Fecha -	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
Fecha	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
2018-10-20	15.00	30.00	0.00	0.00	\$45.00
2018-10-21	15.00	196.00	0.00	0.00	\$211.00
2018-10-22	30.50	21.00	0.00	0.00	\$51.50

Fig. 133. Panel Administradores.

Fuente: Propia

- La Fig. 134, indica el panel principal para los Vendedores.

**Total Recaudado Diario**

Fecha	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
Fecha	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
2018-11-12					\$0.00

**Total Recaudado por Fechas**

Fecha -	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
Fecha	Efectivo	Tarjetas	Dinero Electrónico	Otros	Total
2018-10-20	15.00	30.00	0.00	0.00	\$45.00
2018-10-21	15.00	196.00	0.00	0.00	\$211.00

Fig. 134. Panel Vendedores.

Fuente: Propia

Tabla 40. Sprint 10 – Hoja Electrónica

Sprint: 10			
Fecha Inicio: 31 de agosto de 2018			
			31 de agosto
Requerimiento	Tarea	Tareas Pendientes Responsable	0 Estado
Módulo Seguridad y Control.	Creación del servicio REST para el Módulo de Seguridad y Control.	Andrés Gudiño	Terminado
	Creación de la interfaz para la administración de usuarios con su rol.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 41. Sprint 10 – Pizarrón

Pendiente	En Proceso	Terminada
		Creación del servicio REST para el módulo de ingreso al sistema.
		Creación de la interfaz para la administración de usuarios con su rol.

Fuente: Propia

### 2.6.11. Iteración 11. Implementación del sistema

El presente sistema se encuentra almacenado en un servidor en la nube (Hosting), esto permite el acceso al sistema desde cualquier parte del mundo, así mismo solventar el acceso al sistema por parte del personal de la empresa.

**Requerimiento Nro. 11.-** Implementación del sistema.

**Tarea Nro. 45.-** Implementación del sistema.

- En la Fig. 135 se muestra el acceso online para los Administradores.



Fig. 135. Sistema INDUXION - 1

Fuente: Propia

- En la Fig. 136 se muestra el acceso online para los Asesores - Vendedores.

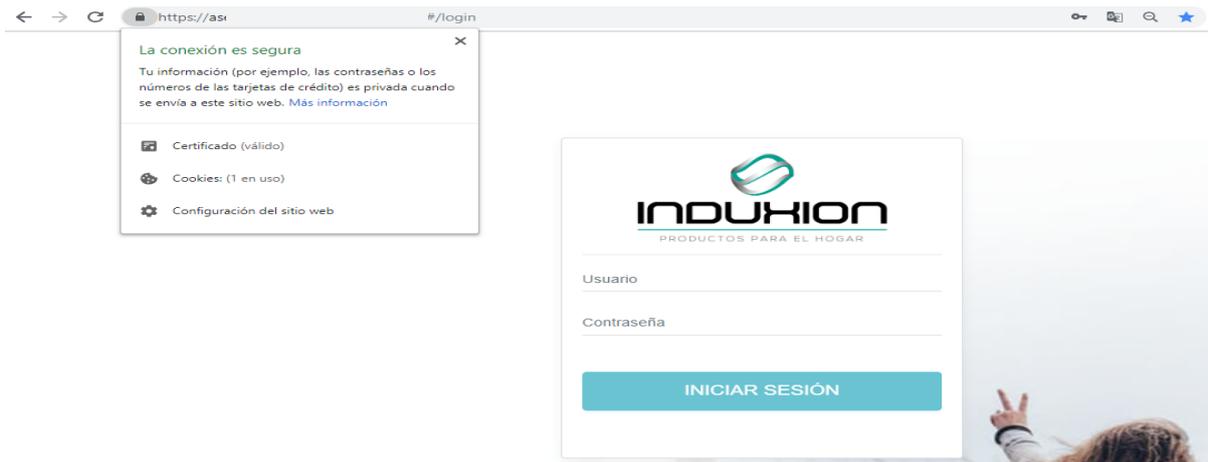


Fig. 136. Sistema INDUXION - 2

Fuente: Propia

Tabla 42. Sprint 11 – Hoja Electrónica

Sprint: 11			
Fecha Inicio: 21 de septiembre de 2018			
Requerimiento	Tarea	Tareas Pendientes	21 de septiembre
		Responsable	0
			Estado
Implementación del Sistema	Implementación del sistema en la empresa en un servidor Hosting.	Andrés Gudiño	Terminado

Fuente: Propia

Tabla 43. Sprint 11 – Pizarrón

Pendiente	En Proceso	Terminada
_____	_____	Implementación del sistema en la empresa en un servidor Hosting

Fuente: Propia



## CAPÍTULO 3

### 3. RESULTADOS

#### 3.1. Beneficio adquirido con el estudio de integración del Framework Yii2 y Angular

- Permite a los desarrolladores optar por nuevas herramientas de desarrollo, implementando diferentes formas de aplicar la lógica de negocio y consumo de servicios desde otras plataformas de multilinguaje de programación.

#### 3.2. Beneficios adquiridos con la implementación del sistema informático en la empresa INDUXION

- Facilita un control ordenado de los productos, manteniendo un registro de todas las transacciones realizadas.
- Permite a los usuarios del sistema ingresar la información de forma válida y ordenada.
- Asegura que toda la información se encuentre actualizada.
- Evita la pérdida de información, en el sistema se ha descartado la opción de eliminar, todo registro de entradas y salida puede ser anulado.
- Garantiza el acceso al sistema y sus funciones estableciendo su rol como: Administradores y Vendedores.
- Facilita a la empresa registrar los productos estableciendo sus categorías, marcas y descripción.
- Facilita registrar las compras realizadas en la empresa, para el abastecimiento de productos con sus márgenes de ganancia y estableciendo impuestos para cada uno de sus productos.
- Acelera el proceso de información relacionada a los productos como es el caso de detalles de productos, valores de venta, stock, descripción de impuestos como es el IVA, este proceso tiene una duración de 4 segundos en comparación al proceso manual que toma alrededor de 5 minutos.
- Acelera el proceso de verificación y reportes del stock de productos disponibles por la empresa, el proceso tiene una duración de 4 segundos, en comparación al proceso manual el tiempo estimado de 24 horas.

- Presenta un registro detallado de entradas y salidas de productos.
- Facilita al personal encargado de ventas registrar una salida de mercadería de forma eficiente y rápida, este proceso tiene una duración de 1 minuto a comparación del proceso manual que dura alrededor de 5 minutos.
- Optimiza los reportes de las ventas realizadas por día y por rango de fechas de forma eficiente, visualizando los tipos de pago asignados en cada una de las ventas realizadas, este proceso toma un tiempo de 4 segundos a comparación del proceso manual que toma alrededor de 30 minutos, correspondiente al proceso de reportes de ventas por rango de fechas toma un tiempo de 4 segundos a comparación del proceso manual toma una duración de 5 horas.

### 3.3. Beneficios de tiempo en operaciones realizadas en el sistema para mejorar la productividad en la empresa INDUXION.

En el proceso de la implementación del sistema informático en la empresa, se ha logrado la optimización en el tiempo de respuesta en la ejecución de operaciones que se realizan de forma cotidiana, a continuación, se muestra el antes y el después de la implementación del sistema.

- **Detalles de productos con sus respectivos valores de venta, stock y si posee impuestos**



Fig. 137. Reporte de productos con sus respectivos valores de venta, stock he impuestos

Fuente: Propia

La obtención de la información requerida por el agente de ventas para cada uno de los productos tomaba un tiempo de 5 minutos de forma manual, el sistema completa la tarea en un rango máximo 4 segundos como se indica en la Fig. 137.

- **Verificación de Stock de productos en la empresa**



Fig. 138. Verificación de Stock de productos en la empresa

Fuente: Propia

En el proceso de verificación de stock disponible por la empresa previo a la implementación del sistema, el personal realizaba tareas demorosas para obtener un registro de productos con una duración de 24 horas, el sistema realiza la operación en máximo 4 segundos como se indica en la Fig. 138.

- **Entradas y Salidas de Productos**



Fig. 139. Reportes entradas y salidas de productos

Fuente: Propia

Previo a la implementación del sistema, la empresa no contaba con un reporte de entradas y salidas de productos, por tal motivo se ha implementado el módulo para el registro de toda operación realizada en el sistema, esta operación tiene una duración máxima de 4 segundos como se indica en la Fig. 139.

- **Registro de Ventas**

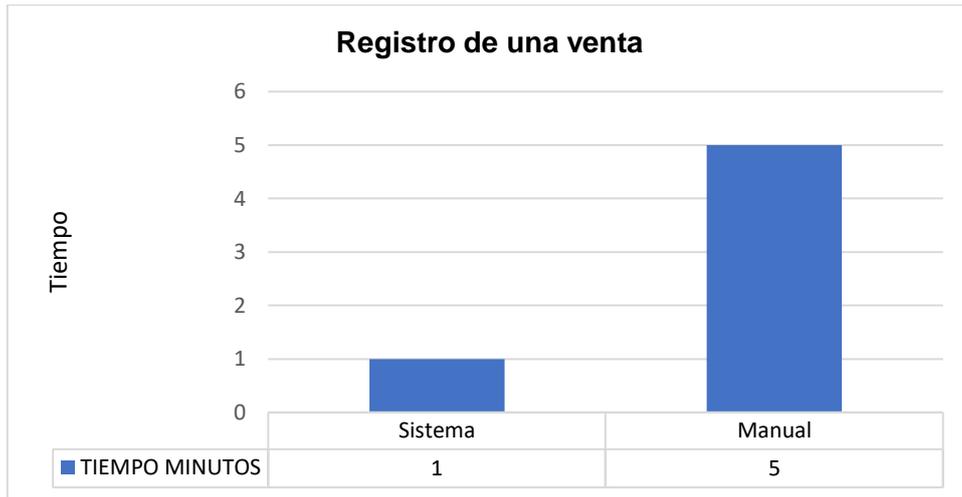


Fig. 140. Registro de una venta

Fuente: Propia

Como se puede observar el proceso de registro de una venta se ha optimizado a 1 minuto a diferencia del proceso manual que tenía una duración de 5 minutos como se muestra en la Fig. 140.

- **Reportes del total recaudado en ventas a diario**

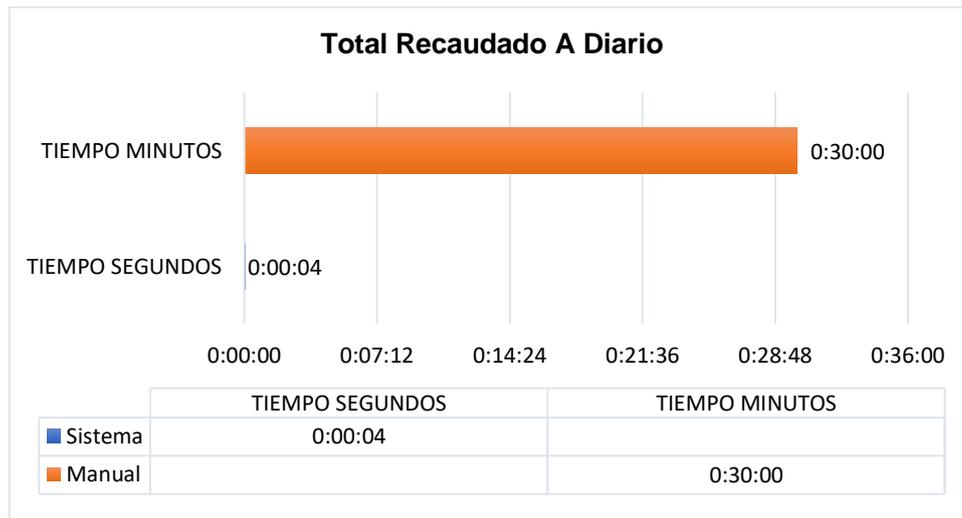


Fig. 141. Total, Recaudado a Diario

Fuente: Propia

Como se puede observar se ha optimizado el proceso de información del total de ventas recaudado a diario con una duración de 30 minutos de forma manual a diferencia del sistema que lo realiza en 4 segundos como se indica en la Fig. 141.

- **Reportes del total recaudado en ventas por fechas**

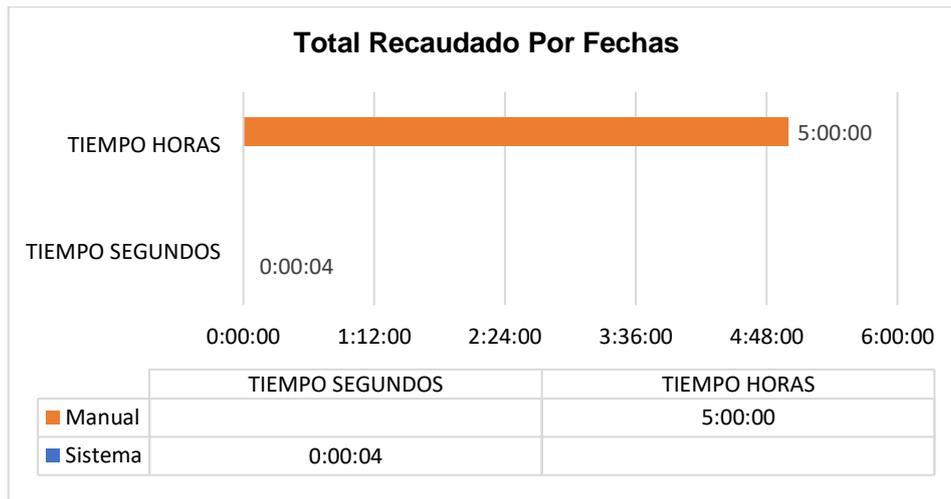


Fig. 142. Total, Recaudado por Fechas

Fuente: Propia

Como se puede observar se ha optimizado el proceso de información del total de ventas recaudadas por fechas, con una duración de 5 horas de forma manual a diferencia del sistema que lo realiza en 4 segundos como se indica en la Fig. 142.

## Conclusiones

- Se comprobó que todos los procesos realizados por la empresa INDUXION para el control de inventarios y reportes de existencias, antes de la implementación de un sistema informático, tomaba una gran parte de tiempo y responsabilidad en cada uno de los integrantes cuando se requería que la información permanezca actualizada, para la emisión de informes correspondientes al inventario de productos, toda la información se almacenaba de forma física a expensas de posibles daños de los documentos e inconsistencias por parte del personal de ventas y dueños.
- Yii2 es un Framework PHP para el desarrollo de sistemas web de forma ágil en un tiempo mínimo, permite la generación de código y la adaptación de características genéricas por parte de terceros. Correspondiente a Angular es un Framework orientado al desarrollo de aplicaciones web conocida como aplicación de página única SPA, permitiendo interactuar con el servidor gracias al consumo de un API REST para lograr una velocidad máxima de respuesta en la aplicación web.
- El uso de nuevas tecnologías, demanda nuevos métodos de comunicación entre aplicaciones, para esto es necesario la implementación de servicios web RESTful permitiendo la obtención de datos y manipulación de los mismo en formato JSON.
- Para mejorar la productividad de la empresa INDUXION, se ha eliminado los procesos manuales para reducir el uso excesivo de papel para el registro de la información, se optimizó los tiempos de respuesta en registros y la obtención de información vital de la empresa como es stock de productos, registro de cada proceso realizado en la empresa como es el ingreso y salida de productos, evitando retrasos en la entrega de información.
- La implementación del sistema informático para la empresa INDUXION permite optimizar todos los procesos manuales efectuados por cada uno de los responsables, ejecutando operaciones vitales para el buen desempeño del negocio; permite a los usuarios desempeñar sus funciones de forma ágil y presenta un control de todas las acciones realizadas.

## Recomendaciones

- Es necesario que toda empresa elimine los procesos manuales y el uso excesivo de papel. Gracias a la presencia de sistemas informáticos se puede realizar las mismas tareas de una forma ágil y sin pérdida de registros, esto es muy beneficioso al momento de optimizar el tiempo y los recursos de una empresa.
- Se recomienda mantener una actualización de contenidos correspondiente a los Frameworks Yii2 y Angular para el correcto funcionamiento de los mismo, al ser software de distribución libre presentan actualizaciones constantes para mejorar el rendimiento y optimización al momento del desarrollo de una aplicación. Al ser dos herramientas muy potentes para el desarrollo de aplicaciones es necesario realizar un estudio amplio para profundizar en el contenido de cada Framework.
- Se recomienda que todos los desarrolladores de software implementen nuevas formas de comunicación entre los sistemas al momento de la obtención de datos requeridos, como son los servicios web REST y así mejorar el tiempo de respuesta al solicitar la información requerida.
- Es necesario que la empresa siempre registre toda la información realizada durante las jornadas de trabajo, esto permitirá a los responsables de la empresa presentar informes actualizados y sin retrasos.
- El uso de software de distribución libre es muy recomendado para los desarrolladores de sistemas informáticos, permitiendo realizar proyectos de baja o alta escalabilidad requeridas por una persona o empresa, sin la preocupación de tener una limitación.

## Bibliografía

- Adrián Peña. (2016). Qué es un servicio RESTFUL. *Http://Www.I2factory.Com*. Retrieved from <http://www.i2factory.com/es/integracion/qué-es-un-servicio-restful>
- Antonio Luis Cardador Cabello. (2014). *Desarrollo de aplicaciones web distribuidas (UF1846)*. (IC Editorial, Ed.) (1º Edición). ProQuest Ebook Central. Retrieved from <https://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=4184019>
- Arturo Muñoz de la Torre Monzón. (2013). *INTRODUCCIÓN A NODE.JS A TRAVÉS DE KOANS*.
- Cecilio Álvarez Caules. (2013). Introducción a Servicios REST. Retrieved from <https://www.arquitecturajava.com/servicios-rest/>
- Darío Blasco. (2016). Ventajas y desventajas de programar en AngularJS. Retrieved from <https://www.atraura.com/angularjs/>
- Emmanuel Valverde Ramos, P. H. (2014). TypeScript, 112.
- Ester Chicano Tejada. (2013). *Utilización de las bases de datos relacionales en el sistema de gestión y almacenamiento de datos*. (IC Editorial, Ed.). ProQuest Ebook Central. Retrieved from <https://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=4499234>
- Federico Guzman. (2016). RESTful API: ¿Qué es y para que sirve? Retrieved from <http://www.weblantropia.com/2016/05/24/restful-api-que-es/>
- Isaac Z. Schlueter, Voss, L., & Silverio, C. (2014). NPM.
- Islavisual.com. (2014). DIFERENCIAS ENTRE SCRUM Y XP. Retrieved from [http://www.islavisual.com/articulos/desarrollo\\_web/diferencias-entre-scrum-y-xp.php](http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php)
- Ivan Alvarado Diaz. (2014). ¿Qué es FrontEnd Y Backend en la programación web? Retrieved from <https://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>
- Javier Zofío Jiménez. (2013). *Aplicaciones web*. (S. A. Macmillan Iberia, Ed.). ProQuest Ebook Central. Retrieved from <https://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=3217129>
- José Luis Berenguel Gómez. (2016). *Desarrollo de aplicaciones web distribuidas*. (S. A. Ediciones Paraninfo, Ed.). Retrieved from

[https://books.google.com.ec/books?id=AZ3gDAAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=AZ3gDAAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

José María Tomás Zafra. (2014). *Elaboración de hojas de estilo (UF1303)*. (IC Editorial, Ed.) (1ª Edición). ProQuest Ebook Central. Retrieved from

<https://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=4184032>

Juan Manuel Vara Mesa, Marcos López Sanz, and D. G. (2014). *Desarrollo web en entorno cliente*. (RA-MA Editorial, Ed.). Madrid, España: ProQuest Ebook Central. Retrieved from <https://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=3229683>

Ken Schwaber y Jeff Sutherland. (2014). *La Guía Definitiva de Scrum: Las Reglas del Juego*. (Vol. 1.0, p. 21).

MariaDB. (2017). MariaDB. Retrieved from <https://mariadb.com/kb/es/about-mariadb/>

Microsoft. (2017). TypeScript. Retrieved from <https://www.typescriptlang.org>

Mikel Eizagirre. (2016). *Introducción a Angular Cli*. *Txoko Tecnológico*. Retrieved from <https://blog.irontec.com/introduccion-a-angular-cli/>

Mohedano, J., Saiz, J. M., & Salazar Román, P. (2012). *Iniciación a javascript*. (Ministerio de Educación de España, Ed.). España.

Node.js Foundation. (2017). Node.js. Retrieved from <https://nodejs.org/es/>

Pablo Valderrey Sanz. (2014). *Gestión de bases de datos*. (R.-M. Editorial, Ed.). ProQuest Ebook Central. Retrieved from <https://ebookcentral.proquest.com/lib/utnortesp/reader.action?docID=3229010#>

Powered by Google. (2016). ANGULAR CLI. Retrieved from <https://cli.angular.io/>

Powered by Google. (2017). ANGULAR - One framework. Mobile & desktop. Retrieved from <https://angular.io/>

Román Carceller Cheza, Carlos Campos Saborido, and C. J. G. M. (2013). *Servicios en red*. (S. A. Macmillan Iberia, Ed.). ProQuest Ebook Central. Retrieved from <https://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=3217117>

Satpathy, T. (2016). *CONOCIMIENTO DE SCRUM (Guía SBOK™)*. SCRUMstudy™.

The PHP Group. (2018). PHP. Retrieved from <https://secure.php.net/>

Vicente Javier Eslava Muñoz. (2013). *El nuevo PHP. Conceptos avanzados*. (B. P. S.L, Ed.). Retrieved from

[https://books.google.com.ec/books?id=NSj3AQAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=NSj3AQAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

Yiiframework.com. (2016). ¿Qué es Yii? Retrieved from

<http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii#sec-1>

## Anexos

- **Anexo A:**
  - Manual Técnico Instalación de Herramientas (En CD).
  - Instaladores (En CD).
  
- **Anexo B:**
  - Manual Técnico Integración Frameworks: Yii2 - Angular (En CD).