



# *Sistema de cómputo en la nube para el seguimiento y monitoreo de trayectorias del turismo en Zacatecas con dispositivos móviles Android*

**Derechos de autor**

Todos los derechos reservados

*Laboratorio de Innovación y Desarrollo Tecnológico en Inteligencia Artificial  
(LIDTIA)*

Prohibida su reproducción total o parcial,  
sin consentimiento expreso de LIDTIA

México MMXX





# Universidad Autónoma de Zacatecas

## Unidad Académica de Ingeniería Eléctrica

Laboratorio de Innovación y Desarrollo Tecnológico  
en Inteligencia Artificial (LIDTIA)

### *Sistema de Cómputo en la nube para el seguimiento y monitoreo de trayectorias del turismo en Zacatecas con dispositivos móviles Android*



Que en cumplimiento parcial de los requisitos para obtener el grado de  
MAESTRO EN INGENIERÍA Y TECNOLOGÍA APLICADA

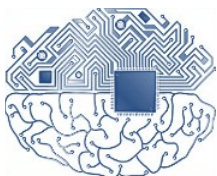
Presenta

**Jorge Armando Ortiz Murillo**

#### Directores de tesis

Dr. José Manuel Ortiz Rodríguez (UAZ - México)  
Dra. Ma. del Rosario Martínez Blanco (UAZ – México)  
Dr. Antonio del Rio de Santiago (UAZ – México)

Zacatecas, Zac., México, julio de 2020



LIDTIA

Oficios de aprobación



*Sistema de Computo en la nube para el seguimiento y monitoreo de trayectorias de dispositivos móviles Android*

**Aprobada por el jurado:**

---

Dr. José Manuel Ortiz Rodríguez

---

Dra. Ma. del Rosario Martínez Blanco

---

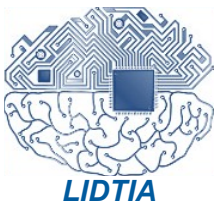
Dr. Antonio del Rio de Santiago

---

Dr. Héctor Alonso Guerrero Osuna

---

Dr. Luis Octavio Solís Sánchez



## Dedicatoria

---

Esta tesis es la conclusión de un trabajo realizado con mucho esfuerzo y dedicación con el fin de cumplir objetivos planteados a través de la vida diaria, la elaboración de este documento está dedicada a todas las personas que me apoyaron para obtener este logro: amigos, profesores, familiares y asesores. Quiero agradecer en primer lugar a mis padres los cuales siempre han estado para apoyarme, son un ejemplo de cómo quiero llegar a ser algún día, mi esposa e hijos quienes a pesar de los tropiezos que hemos tenido son mi motivación para seguir adelante, mis amigos y/o compañeros con los cuales compartí este proceso de formación y a mis profesores y asesorías que sin su conocimiento y disponibilidad este logro jamás hubiera sido posible.

## Agradecimientos

---

Este trabajo fue parcialmente apoyado por:

- Fondo Sectorial de Investigación para la Educación de CONACyT bajo el convenio 241771, con número de becario 496992.
- El Laboratorio de Innovación y Desarrollo Tecnológico en Inteligencia Artificial (LIDTIA) con sede en la Universidad Autónoma de Zacatecas.
- El Posgrado en Ingeniería y Tecnología Aplicada con sede en la Universidad Autónoma de Zacatecas.

## Resumen

---

En la actualidad el uso de tecnológicas está impulsando el crecimiento de la industria, gracias a esto se creó el concepto de industria 4.0, dicho concepto se utilizó por primera vez en la hannover fair y consiste principalmente en el uso de las siguientes tecnologías: big data, could computing, robótica, internet de las cosas, manufactura aditiva y realidad aumentada, esto lleva a la creación de redes inteligentes con el uso de servicios de comunicación inalámbrica y sensores, un incremento en la movilidad de los sistemas, mayor integración con los clientes y la necesidad de la creación de nuevos modelos de negocio. El concepto se aplica a varios sectores de la industria, este proyecto está relacionado con la industria del turismo o turismo inteligente. El desarrollo de nuevas tecnologías de información y comunicación está cambiando la esencia del turismo y esto da la oportunidad de crear nuevos conceptos y tecnologías en la industria del turismo. El turismo es la industria más grande del mundo y provee mayor contribución al desarrollo de la economía y desarrollo de los países. El turismo inteligente es la información e infraestructura turística existente con el uso de la información digital y el internet de las cosas. En este proyecto se presenta la implementación de una aplicación móvil, en la cual se utilizaron varias API's de Android basadas en la tecnología Java para la recolección de datos de geo localización GPS y preferencias de usuarios, la conexión a una base de datos remota utilizando un servidor web REST y el uso de listados y mapas en una página web desarrollada en Java con el patrón de diseño Modelo Vista Controlador. La base de datos está desarrollada con la tecnología MySQL, la cual cuenta con 12 tablas para el registro de todas las acciones realizadas por los usuarios de la aplicación Android. La aplicación web se utiliza para el análisis de la información recolectada por los dispositivos móviles y la administración de los

catálogos del sistema tales como Usuarios y Edificios Históricos. Con los datos obtenidos por este proyecto se abre la posibilidad de la implementación de nuevos proyectos para utilizar el modelo de análisis de información que ofrece Big Data debido a las características de los datos recolectado, la velocidad a la que se generan y la gran cantidad de información recolectada para realizar la predicción de las preferencias de los usuarios de la aplicación en conjunto con los sitios más visitados y de esta manera apoyar al turismo en el estado de Zacatecas. Los resultados obtenidos muestran que el sistema recolecta las preferencias y los datos de posicionamiento geográfico de los usuarios que utilizaron la aplicación móvil y dichos datos muestran la posición geográfica de los edificios históricos más visitados, así como las trayectorias realizadas por los usuarios.

Palabras clave: seguimiento de trayectorias, cómputo en la nube, dispositivos móviles.



## Abstract

---

Currently the use of technology is driving the growth of the industry, thanks to this the concept of industry 4.0 was created, this concept was used for the first time in the hannover fair and consists mainly of the use of the following technologies: big data , cloud computing, robotics, internet of things, additive manufacturing and augmented reality, this leads to the creation of intelligent networks with the use of wireless communication services and sensors, an increase in the mobility of systems, greater integration with customers and the need to creation of new business models. The concept is applied to several sectors of the industry. This project is related to the tourism or smart tourism industry. The development of new information and communication technologies is changing the essence of tourism and this gives the opportunity to create new concepts and technologies in the tourism industry. Tourism is the largest industry in the world and provides the greatest contribution to the development of the economy and development of the countries. Smart tourism is information and existing tourism infrastructure with the use of digital information and internet of things. This project presents the implementation of a mobile application, in which several Android APIs based on Java technology were used for the collection of geographical positioning data and user preferences data, the connection to a remote database using a REST web server and the use of data lists and Google maps in a web page developed in Java with the design pattern Model Vista Controller. The database was developed with MySQL technology, which contains 12 tables for the registration of all actions performed by users of the Android application. The web application is used for the analysis of collected data by mobile devices and the administration of system catalogs such as Users and Historic Buildings. With the data obtained by this project, it opens the possibility of the implementation of new projects

to use the information analysis model offered by Big Data due to the characteristics of the data collected, the speed at which they are generated and the large number of information collected to make the prediction of the users preferences in the application in conjunction with the most visited sites and in this way support tourism in the state of Zacatecas. The results obtained show that the system collects the preferences and geographical positioning data of the users who used the mobile application and these data show the geographical position of the most visited historic buildings, as well as the trajectories made by the users.

Keywords: path tracking, cloud computing, mobile devices

# Índice general

---

<b>DEDICATORIA.....</b>	<b>5</b>
<b>AGRADECIMIENTOS.....</b>	<b>6</b>
<b>RESUMEN.....</b>	<b>7</b>
<b>ABSTRACT.....</b>	<b>9</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
INTRODUCCIÓN.....	1
JUSTIFICACIÓN.....	2
DEFINICIÓN DEL PROBLEMA.....	3
PREGUNTAS DE INVESTIGACIÓN.....	3
HIPÓTESIS.....	4
OBJETIVO GENERAL.....	4
<i>Objetivos particulares.....</i>	<i>4</i>
<b>REVISIÓN DE LITERATURA.....</b>	<b>5</b>
2.1. INDUSTRIA 4.0.....	5
2.1.1. <i>Ciudades inteligentes.....</i>	<i>7</i>
2.1.2. <i>Turismo o turismo inteligente.....</i>	<i>10</i>
2.1.3. <i>Dispositivos Móviles.....</i>	<i>10</i>
2.1.4. <i>Seguimiento de Trayectorias.....</i>	<i>12</i>
2.1.4.1. <i>Existen tres tipos principales de rastreo terrestre: [6].....</i>	<i>13</i>
2.2. TÉCNICAS DE PROGRAMACIÓN.....	14
2.2.1. <i>La programación orientada a objetos.....</i>	<i>14</i>
2.2.1.1. <i>Poliformismo.....</i>	<i>15</i>
2.2.1.2. <i>Clases.....</i>	<i>15</i>
2.2.1.3. <i>Herencia.....</i>	<i>15</i>
2.2.2. <i>Programación orientada a aspectos.....</i>	<i>16</i>
2.2.2.1. <i>Consideraciones al utilizar la técnica de programación orientada a aspectos.....</i>	<i>17</i>
2.2.3. <i>Tipos de datos en java.....</i>	<i>17</i>
2.3. BASES DE DATOS.....	18
2.3.1. <i>Modelo en red.....</i>	<i>20</i>

2.3.2. Modelo entidad relación.....	20
2.3.2.1. Correspondencia de cardinalidades.....	21
2.3.2.2. Restricciones de participación.....	21
2.4. ARQUITECTURA DE SOFTWARE.....	23
2.4.1. Patrones de diseño.....	23
2.5. FRAMEWORKS.....	29
2.5.1. Hibérnate.....	29
2.5.2. Spring.....	30
2.5.3. Spring Security.....	34
2.5.4. Servidores Web.....	34
2.6. BIG DATA.....	39
<b>MATERIALES Y MÉTODOS.....</b>	<b>50</b>
3.1. MATERIALES.....	50
3.1.1. Software.....	50
3.1.2. Hardware.....	52
3.2. METODOLOGÍA.....	52
3.2.1. Instalación y configuración de paquetes de software.....	52
3.2.1.1. MySQLServer.....	53
3.2.2. Desarrollo de prototipos para la aplicación Android y Web.....	103
3.3. CREACIÓN DE MAPAS CONCEPTUALES PARA EL SERVIDOR WEB Y LAS APLICACIONES ANDROID Y WEB:.....	112
3.4. DISEÑO DE LA BASE DE DATOS EN EL PROGRAMA MYSQL WORKBENCH:.....	117
3.5. DESARROLLO DE APLICACIONES.....	117
<b>RESULTADOS Y DISCUSIÓN.....</b>	<b>119</b>
4.1. APLICACIONES.....	119
4.1.1. Aplicación Móvil.....	119
4.1.2. Servidor Web.....	121
4.1.3. Aplicación Web.....	122
4.2. PARTICIPACIÓN EN CONGRESOS.....	126
4.2.1. 17vo Seminario de Investigación.....	126
4.2.2. XVI International Symposium on Solid State Dosimetry.....	126
4.3. CERTIFICACIONES.....	127
4.3.1. Big Data Integration and Processing.....	127
4.3.2. Big Data Modeling and Management Systems.....	127
4.3.3. Clasificación de Imágenes.....	127
4.3.4. Fundamentals of Digital Image and Video Processing.....	127
4.3.5. Introducción a Big Data.....	127

4.3.6. <i>Machine Learning</i> .....	128
4.4. DISCUSIÓN.....	128
<b>CONCLUSIONES</b> .....	<b>129</b>
5.1. PROSPECTIVAS.....	129
<b>BIBLIOGRAFÍA</b> .....	<b>130</b>
<b>CURRÍCULUM VITAE</b> .....	<b>134</b>

# Índice de figuras

---

FIGURA 1.1 INDUSTRIA 4.0.....	1
FIGURA 1.2 GPS.....	3
FIGURA 2.1 ARQUITECTURA DE UNA CIUDAD INTELIGENTE.....	9
FIGURA 2.2 DISPOSITIVOS MÓVILES EN LA NUBE.....	12
FIGURA 2.3 MODELO JERÁRQUICO.....	20
FIGURA 2.4 EJEMPLO ENVIÓ DE MENSAJES MVC.....	27
FIGURA 2.5 PATRÓN DE PUENTE.....	28
FIGURA 2.6 PATRÓN DE ESTRATEGIA.....	28
FIGURA 2.7 SPRING FRAMEWORK RUNTIME.....	32
FIGURA 2.8 HDFS.....	44
FIGURA 2.9 COMPUTO EN LA NUBE.....	48
FIGURA 3.1 METODOLOGÍA.....	53
FIGURA 3.2 MYSQL COMMUNITY DOWNLOAD.....	54
FIGURA 3.3 MYSQL VERSIÓN.....	55
FIGURA 3.4 CONFIRMACIÓN DESCARGA MYSQL.....	55
FIGURA 3.5 ARCHIVO DESCARGADO MYSQL.....	56
FIGURA 3.6 INSTALADOR MYSQL.....	56
FIGURA 3.7 LICENCIA MYSQL.....	57
FIGURA 3.8 PRODUCTOS MYSQL.....	58
FIGURA 3.9 DIRECTORIO DE INSTALACIÓN MYSQL.....	58
FIGURA 3.10 CONFIRMACIÓN DE INSTALACIÓN MYSQL.....	59
FIGURA 3.11 CONFIRMACIÓN DE INSTALACIÓN MYSQL.....	59
FIGURA 3.12 FINALIZACIÓN DE INSTALACIÓN MYSQL.....	60
FIGURA 3.13 REPLICACIÓN MYSQL.....	61
FIGURA 3.14 RED MYSQL.....	61
FIGURA 3.15 AUTENTICACIÓN MYSQL.....	62
FIGURA 3.16 CUENTAS Y ROLES MYSQL.....	63
FIGURA 3.17 SERVICIO MYSQL.....	64
FIGURA 3.18 APLICAR CONFIGURACIÓN MYSQL.....	64
FIGURA 3.19 CONFIGURACIÓN COMPLETA MYSQL.....	65
FIGURA 3.20 DESCARGA JAVA.....	66
FIGURA 3.21 TÉRMINOS Y CONDICIONES JAVA.....	66
FIGURA 3.22 CONFIRMAR DESCARGA JAVA.....	66
FIGURA 3.23 DIRECTORIO DE DESCARGA JAVA.....	67
FIGURA 3.24 CARPETA DE INSTALACIÓN JAVA.....	67
FIGURA 3.25 INSTALACIÓN JAVA.....	68
FIGURA 3.26 INSTALACIÓN COMPLETA JAVA.....	68
FIGURA 3.27 DESCARGA TOMCAT.....	69

FIGURA 3.28 DISTRIBUCIÓN TOMCAT.....	70
FIGURA 3.29 DIRECTORIO DESCARGA TOMCAT.....	70
FIGURA 3.30 DESCOMPRIMIR TOMCAT.....	71
FIGURA 3.31 DIRECTORIO DE ARCHIVOS TOMCAT.....	72
FIGURA 3.32 DESCARGA MAVEN.....	73
FIGURA 3.33 DISTRIBUCIÓN MAVEN.....	73
FIGURA 3.34 DIRECTORIO DESCARGA MAVEN.....	73
FIGURA 3.35 DESCOMPRIMIR MAVEN.....	74
FIGURA 3.36 DIRECTORIO DE ARCHIVOS MAVEN.....	75
FIGURA 3.37 DESCARGA ANDROID STUDIO.....	75
FIGURA 3.38 TÉRMINOS Y CONDICIONES ANDROID STUDIO.....	76
FIGURA 3.39 CONFIRMAR DESCARGA ANDROID STUDIO.....	76
FIGURA 3.40 INSTALACIÓN ANDROID STUDIO.....	76
FIGURA 3.41 RECOMENDACIONES ANDROID STUDIO.....	77
FIGURA 3.42 DIRECTORIO DE INSTALACIÓN ANDROID STUDIO.....	78
FIGURA 3.43 INSTALACIÓN COMPLETA ANDROID STUDIO.....	79
FIGURA 3.44 INSTALACIÓN COMPLETA ANDROID STUDIO.....	79
FIGURA 3.45 DESCARGA VIRTUALBOX.....	80
FIGURA 3.46 DISTRIBUCIÓN VIRTUALBOX.....	81
FIGURA 3.47 CONFIRMAR DESCARGA VIRTUALBOX.....	81
FIGURA 3.48 DIRECTORIO DESCARGA VIRTUALBOX.....	82
FIGURA 3.49 INSTALACIÓN VIRTUALBOX.....	82
FIGURA 3.50 PRODUCTOS VIRTUALBOX.....	83
FIGURA 3.51 RED VIRTUALBOX.....	84
FIGURA 3.52 INICIAR CONFIGURACIÓN VIRTUALBOX.....	85
FIGURA 3.53 INTERFAZ VIRTUALBOX.....	86
FIGURA 3.54 DESCARGA GENYMOTION.....	87
FIGURA 3.55 DISTRIBUCIÓN GENYMOTION.....	88
FIGURA 3.56 DESCARGA SIN VIRTUALBOX GENYMOTION.....	88
FIGURA 3.57 CONFIRMAR DESCARGA GENYMOTION.....	88
FIGURA 3.58 DIRECTORIO DE DESCARGA GENYMOTION.....	89
FIGURA 3.59 IDIOMA INSTALACIÓN GENYMOTION.....	89
FIGURA 3.60 DIRECTORIO INSTALACIÓN GENYMOTION.....	90
FIGURA 3.61 ACCESO GENYMOTION.....	91
FIGURA 3.62 CONFIRMACIÓN DE INSTALACIÓN GENYMOTION.....	92
FIGURA 3.63 FINALIZAR INSTALACIÓN GENYMOTION.....	93
FIGURA 3.64 LICENCIA GENYMOTION.....	94
FIGURA 3.65 TÉRMINOS Y CONDICIONES GENYMOTION.....	95
FIGURA 3.66 INTERFAZ GENYMOTION.....	96
FIGURA 3.67 INTERFAZ SPRING SOURCE TOOL SUIT 4.....	97
FIGURA 3.68 DIRECTORIO DE DESCARGA SPRING SOURCE TOOL SUIT 4.....	97
FIGURA 3.69 DESCOMPRIMIR SPRING SOURCE TOOL SUIT 4.....	98
FIGURA 3.70 DIRECTORIO DE INSTALACIÓN SPRING SOURCE TOOL SUIT 4.....	99
FIGURA 3.71 DIRECTORIO SPRING SOURCE TOOL SUIT 4.....	99
FIGURA 3.72 DIRECTORIO SPRING SOURCE TOOL SUIT 4.....	100
FIGURA 3.73 DESCARGA MYSQL WORKBENCH.....	101
FIGURA 3.74 DISTRIBUCIÓN MYSQL WORKBENCH.....	101
FIGURA 3.75 CONFIRMAR DESCARGA MYSQL WORKBENCH.....	102
FIGURA 3.76 DIRECTORIO DE DESCARGA MYSQL WORKBENCH.....	102





UNIVERSIDAD AUTÓNOMA DE ZACATECAS

“FRANCISCO GARCÍA SALINAS”  
Unidad Académica de Ingeniería Eléctrica



**Dra. Samanta Deciré Bernal Ayala.**  
**Departamento de Servicios Escolares.**  
**Presente.**

El suscrito **Dr. José Manuel Ortiz Rodríguez** docente de la Maestría en Ingeniería y Tecnología Aplicada, dependiente de la Universidad Autónoma de Zacatecas:

#### DICTAMINA

Que después de haber revisado la Tesis titulada “Análisis, diseño e implementación de tecnología basada en redes neuronales de regresión generalizada para la solución de problemas inversos, aplicada a la espectrometría y dosimetría neutrónica” Presentada por la **C. Ma. Del Rosario Martínez Blanco**, egresada de este programa educativo con matrícula **97501856**, cumple con todos los requisitos que el reglamento de titulación de los posgrados exige para este tipo de trabajos, por lo que el mencionado egresado puede pasar a reproducirlo para presentar su Examen Profesional.

Se extiende la presente a solicitud del interesado, para los fines legales que a el mismo convenga, en la ciudad de Zacatecas, Zac. A los 10 días del mes de febrero del año dos mil diecisiete.

**Asesor de Tesis**  
**Dr. José Manuel Ortiz Rodríguez**



## Índice de tablas

---

TABLA 2.1 CATEGORÍAS DE LOS MODELOS DE NEGOCIO EN LAS CIUDADES INTELIGENTES.....	10
TABLA 2.2. PALABRAS RESERVADAS EN JAVA.....	19
TABLA 2.3. DATOS PRIMITIVOS EN JAVA.....	19
TABLA 2.4 SPRING JDBC.....	35

# Introducción

---

## Introducción

El término Industria 4.0 fue mencionado por primera vez en la “Hannover Fair” con la presentación de la iniciativa “Industry 4.0” [1]. El principio básico de industria 4.0 es el uso del internet de las cosas y la manufactura inteligente, lo cual provoca el cambio en los sistemas centralizados de las fábricas a una inteligencia descentralizada [2].

La Figura 1.1 muestra los elementos que conforman a la industria 4.0:

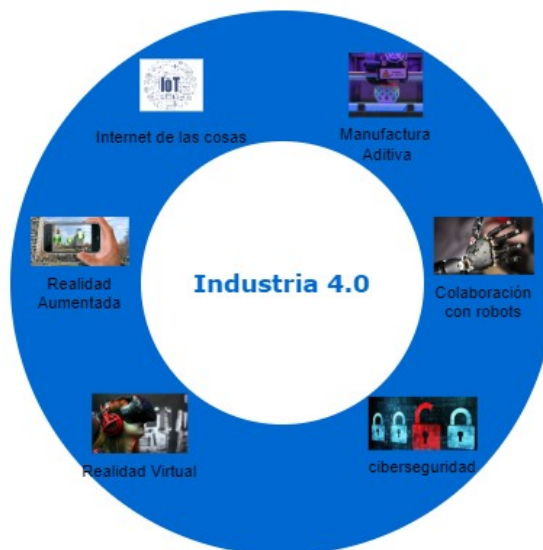


Figura 1.1 Industria 4.0

El 50% de la población mundial vive en una ciudad en vez de un área rural, más aun las ciudades van a continuar creciendo y se prevé que para el 2050 el 70% de la población mundial viva en ciudades, aunado a esto nos enfrentamos a crecientes limitaciones de recursos naturales, aumento de la población y una reestructuración de la economía mundial [3].

Debido a esto, las ciudades existentes y nuevas deben manejar los siguientes conceptos: impacto ambiental, crecimiento económico y evolución social, las tecnologías digitales se siguieren para el desarrollo de las ciudades inteligentes, las cuales integran infraestructura digital con su ciudad física para reducir el impacto ambiental mientras se mejora la calidad de vida y la prosperidad económica [3].

El creciente desarrollo de tecnología de información y comunicación está cambiando la esencia del turismo, los métodos, herramientas de administración y el marketing, esto brinda la oportunidad de renovar conceptos y metodologías utilizados en turismo[4]. El turismo es la industria más grande del mundo y hace una mejor contribución al desarrollo de la economía y el desarrollo de los países [5].

## **Justificación**

En la actualidad el método más común de recolección de trayectorias en las ciencias sociales en general o en el turismo en particular, es con el uso de las tecnologías, es un método no observador. [6]

Uno de estos métodos, es el Sistema de Posicionamiento Global o GPS por sus siglas en ingles que es básicamente una serie de satélites que orbitan las señales terrestres de radio difusión capturadas por un sistema receptor, triangula la información recibida de al menos 4 satélites como se ilustra en la figura 1.2. La tecnología GPS es tomada como una herramienta para rastrear la actividad de los peatones donde su principal ventaja es ser global. [6]



*Figura 1.2 GPS*

Otro método es el sistema de rastreo Terrestre o Land-Based Tracking Systems, esta tecnología se basa en el principio de que las señales electromagnéticas trabajan a una velocidad conocida, por lo general son redes locales ya que se distribuyen los detectores de radio frecuencia en una zona determinada. [6]

Considerando los métodos anteriormente mencionados, es posible la elaboración de una herramienta que utilice el método de sistema de posicionamiento global, para obtener la ubicación del turismo en el estado de Zacatecas, el uso de los dispositivos móviles Android; para realizar el envío de dicha información y la creación de una aplicación web; para el procesamiento y monitoreo de la información.

## **Definición del problema**

Entre los retos más importantes para la creación de ciudades inteligentes está la recolección y análisis de información significativa, además de la necesidad de construcción de un turismo inteligente, mediante el traslado de la información turística existente y el internet de las cosas. [7],[8]

## **Preguntas de Investigación**

¿Se puede hacer el seguimiento y monitoreo de rutas mediante el uso de dispositivos móviles, Servidores Web y Aplicación Web?

¿Se pueden crear aplicaciones móviles compatibles con la mayoría de dispositivos Android?

¿Se pueden conectar aplicaciones móviles a bases de datos remotas?

¿Es compatible la implementación de aplicaciones web, servidores web y aplicación Android?

## **Hipótesis**

Usando la información generada por los dispositivos móviles y el modelo de manejo de información de los servidores web es posible hacer el seguimiento y monitoreo de trayectorias.

## **Objetivo general**

Realizar el seguimiento y monitoreo de trayectorias utilizando dispositivos móviles, Servidor y Aplicación Web.

Para alcanzar el objetivo general, se proponen los siguientes objetivos particulares:

### ***Objetivos particulares***

- Diseño e Implementación de Base de datos para el almacenamiento de trayectorias.
- Creación de Aplicación Android para el seguimiento de rutas.
- Creación de Aplicación Web para el monitoreo de rutas.
- Creación de Servidor Web para interfaz entre base de datos y aplicaciones móviles.
- Implementación de Aplicación Web, Servidor Web y Base de Datos.

# Revisión de literatura

---

## 2.1. Industria 4.0

El término Industria 4.0 fue mencionado por primera vez en la “Hannover Fair” con la presentación de la iniciativa “Industry 4.0” [1]. La primera revolución industrial “Mecanización” como resultado de la invención de la máquina de vapor, la segunda “Producción en masa” con la ayuda de la electricidad, la tercera “Digitalización” con el uso la electrónica y las tecnologías de información, marca la venida de la cuarta revolución industrial con el uso de los sistemas físicos cibernéticos por sus siglas en ingles CPS y el internet de las cosas y servicios [1]. El objetivo de la Industria 4.0 es la aparición de fábricas digitales con las siguientes características [1]:

*Creación de redes inteligentes:* sistemas interconectados con cyber tecnología, como servicios de comunicación inalámbrica, actores y sensores inteligentes y tecnologías de telecomunicación.

*Movilidad:* dispositivos móviles como celulares y tabletas, se han vuelto indispensables en la automatización de la industria. Que proveen un acceso temporal e independiente a los procesos y servicios de los sistemas automatizados.

*Flexibilidad:* la industria 4.0 permite una gran flexibilidad tanto en desarrollo, diagnóstico y mantenimiento, así como en la operación de los sistemas

automatizados. En el desarrollo de estos sistemas puedes escoger entre un amplio rango de componentes, módulos y servicios. El diagnóstico puede ser realizado en parte por el usuario. El uso de “Big Data” ayuda a la automatización. La información puede ser obtenida bajo demanda, inteligentemente y relacionada de forma que el diagnóstico automatizado puede ser alcanzado.

*Integración de clientes:* con la industria 4.0 es posible personalizar los productos acorde a las necesidades de los clientes. Los sistemas automatizados del siglo XXI pueden adaptarse a las habilidades de los usuarios y dar soporte en todas las situaciones, así como asistirlos en diferentes etapas de la vida.

*Nuevos modelos de negocio innovadores:* la producción en el futuro va a ser distribuida y flexible. Nuevos procesos de desarrollo, infraestructura y servicios deben surgir. Los productos se volverán modulares y configurables y el producto pueda ser adaptado al requerimiento.

El principio básico de industria 4.0 es el uso del internet de las cosas y la manufactura inteligente y esto provoca el cambio en los sistemas centralizados de las fábricas a una inteligencia descentralizada. Las características de las fábricas inteligentes: [2]

*Producción personalizada:* los procesos de producción deben cumplir con los diferentes requisitos en las ordenes de producción, es posible tener un volumen bajo de producción y aun así teniendo ganancias.

*Flexibilidad:* los procesos de producción inteligente deben considerar varios aspectos, como el tiempo, calidad, precio y aspectos ecológicos.

*Visibilidad de la fábrica y la optimización en la toma de decisiones:* tomar decisiones en cualquier momento es fundamental para tener éxito en el mercado, el internet de

las cosas provee la información en tiempo real, permitiendo la optimización de la fábrica y mejorando su eficiencia.

*Obtener valor de la información recolectada por big data:* nuevas mejoras y valor puede ser proporcionado por el análisis de datos.

*Cadena de producción conectada:* el uso del internet de las cosas ayuda en la obtención de la información en tiempo real de la cadena de producción.

*Administración de la energía:* la implementación de la eficiencia de la energía requiere del comportamiento en la línea de producción, basándose en el uso del internet de las cosas se pueden adaptar prácticas para mejorar la eficiencia a un nivel de producción. Creación de nuevos servicios, Nuevos métodos de planeación, monitoreo remoto, automatización y cambiar el rol del hombre y mantenimiento proactivo.

### **2.1.1. Ciudades inteligentes**

El 50% de la población mundial vive en una ciudad en vez de un área rural, más aun las ciudades van a continuar creciendo y se prevé que para el 2050 el 70% de la población mundial viva en ciudades, aunado a esto nos enfrentamos a crecientes limitaciones de recursos naturales, aumento de la población y una reestructuración de la economía mundial. Debido a esto, las ciudades existentes y nuevas deben manejar los siguientes conceptos: impacto ambiental, crecimiento económico y evolución social, las tecnologías digitales se sugieren para el desarrollo de las ciudades inteligentes, las cuales integran infraestructura digital con ciudad física para reducir el impacto ambiental mientras se mejora la calidad de vida y la prosperidad económica. [3]

IBM definió el concepto de ciudad inteligente como el uso de las tecnologías de información y comunicación para censar, analizar e integrar la información clave de



los sistemas fundamentales que corren en las ciudades, así mismo las ciudades inteligentes pueden hacer responsable a la inteligencia de diferentes tipos de necesidades como el estilo de vida, protección ambiental, seguridad pública y servicios que ofrece la ciudad y actividades industriales y comerciales. [8]

Las ciudades inteligentes son la tendencia a futuro del desarrollo urbano, la construcción de las mismas generalmente se puede dividir en 3 niveles, la construcción de una infraestructura pública, construcción de una plataforma pública para la ciudad inteligente y la construcción de aplicaciones, esta última ha tomado mayor importancia, dichos sistemas pueden ser aplicados a diferentes aspectos como: construcción de una ciudad en la nube, construcción de una casa inteligente, servicios públicos inteligentes y construcción de administración social, construcción de transporte inteligente, construcción de tratamientos médicos inteligentes, construcción de una administración urbana inteligente, construcción de una ciudad verde y la construcción de un turismo inteligente. [8]

Muchos enfoques de ciudades inteligentes han sido propuestos, muchas de estas tecnologías o soluciones son la re-aplicación de sistemas y arquitecturas tradicionales implementadas para corporaciones y empresas, en la figura 2.1 se muestra una imagen de la arquitectura de una ciudad inteligente, así como en la tabla 2.1 se muestran las características de los modelos de una ciudad inteligente.

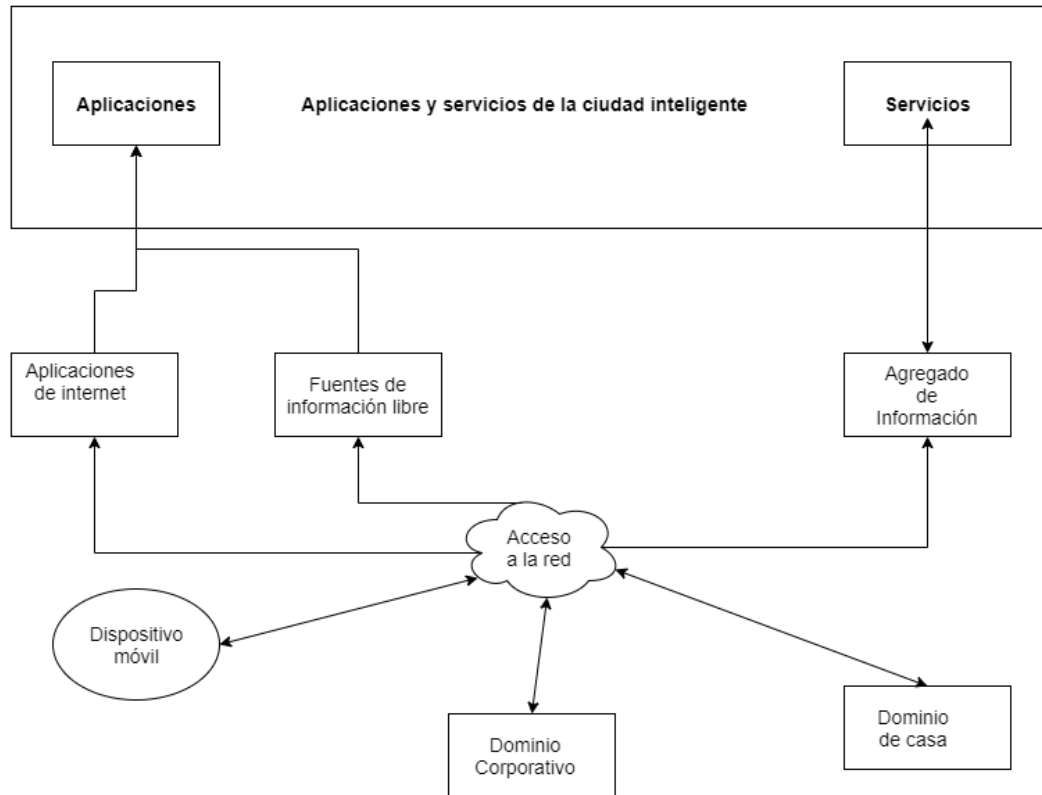


Figura 2.1 Arquitectura de una ciudad inteligente

Tabla 2.1 Categorías de los modelos de negocio en las ciudades inteligentes

Categoría	Ejemplo	Impacto técnico
Mejora ambiental	Medidores inteligentes, red inteligente y monitoreo de calidad del aire	Nuevos dispositivos conectados a la red
Crecimiento económico	Incubadoras, educación inteligente e iniciativas de crecimiento verde	Información libre y agregación de información
Costo eficiencia	Remover información irrelevante	Cómputo en la nube e información libre
Bienestar	Alarmas de incendio, evadir el transporte alrededor del incendio	Nuevos dispositivos y fuentes de información, información libre y agregación de información
Calidad de vida	Ciclos de retroalimentación en la planeación urbana a partir de datos de toda la ciudad	Agregación de información y administración de información
Ciudadanos conectados	Aplicaciones de transporte para un viaje conectado	Privacidad, agregación de información e información libre
Nuevos modelos de negocio	Usando información de dispositivos móviles en la ciudad para crear nueva publicidad y fuentes de ingresos para las empresas locales	Privacidad, agregación de información, información libre y procedencia de información

### **2.1.2. Turismo o turismo inteligente**

El creciente desarrollo de tecnología de información y comunicación está cambiando la esencia del turismo, los métodos, herramientas de administración y el marketing, esto brinda la oportunidad de renovar conceptos y metodologías utilizados en turismo [4]. El turismo es la industria más grande del mundo y hace una mejor contribución al desarrollo de la economía y el desarrollo de los países [5].

El turismo inteligente es la única forma de trasladar la información, debe basarse en la información e infraestructura turística existente, tomando ventaja de la información digital y el internet de las cosas para alcanzar a establecer un conjunto de soluciones, las cuales pueden ser consideradas para satisfacer la gestión y las tareas relacionadas con el turismo, tales como servicios turísticos en línea, gestión de la relación con los clientes, desarrollo de un mercado turístico nacional e internacional, sistema de gestión inteligente de monitoreo, recopilación de información turística y pronóstico de desarrollo turístico. [8]

### **2.1.3. Dispositivos Móviles**

Con el crecimiento del uso de los dispositivos móviles en la actualidad, ha cambiado la preferencia del uso de los celulares tradicionales y computadoras portátiles a uso de los celulares inteligentes y tabletas [8]

El internet puede ser visto como un sistema distribuido a gran escala con una gran cantidad de información y usuarios, eso es influenciado según la ingeniería de los sistemas de información por las siguientes tendencias: [9]

La evolución de un contexto estático a los servidores web. La evolución de los sistemas cliente servidor a sistemas punto a punto y penetrantes, y finalmente en conjunto con desarrollos en tecnologías de comunicación inalámbrica. La tendencia de cambiar los sistemas estacionarios a sistemas de administración de información

distribuidos móviles, para la administración de estos sistemas se hace uso de los servidores web.

Los servidores web son componentes de software que pueden ser acusados por internet mediante mecanismos y protocolos web los cuales definiremos más a detalle en capítulos posteriores, con la instrucción de la tercera y posteriores generaciones de las tecnologías en comunicación móvil como UMTS, GPRS/EDGE, la velocidad de comunicación se incrementó significativamente, así como el poder y las capacidades de los dispositivos móviles nos permiten construir mejores aplicaciones y el uso de dichas aplicaciones en diferentes dominios [9].

Dicho esto, permite la comunicación de dispositivos móviles con las interfaces de los servidores web haciendo clientes a los dispositivos móviles de los servidores web, con esto es posible decir que en el futuro la mayoría de los servidores web serán consumidos por aplicaciones móviles y en la actualidad ya están apareciendo soluciones en el mercado que utilizan estas tecnologías. La demanda del acceso realizado por dispositivos móviles es cerca de un 60% del tráfico total en el 2016. [8], [9]

La naturaleza de los recursos que tienen los dispositivos móviles, especialmente lo limitado de la batería es una problema para la implementación de aplicaciones y servicios móviles, de acuerdo al estudio U.S. Wireless Smartphone Customer Satisfaction realizado en el 2012 la vida de la batería es el aspecto más deficiente de los dispositivos móviles, mientras que los nuevos teléfonos celulares tienen CPU más rápidos, mayor capacidad de almacenamiento, pantallas más grandes y el ancho de banda paso de kilobytes por segundo a megabytes por segundo en unos pocos años y requieren de más uso de la batería. [8]

El paradigma del cómputo móvil en la nube o MCC (Mobile Cloud Computing) salió a la luz para resolver el conflicto mencionado anteriormente, donde los servidores en la

nube tienen mayor cantidad de recursos que los dispositivos móviles, “en vez de consumir recursos de manera local en los dispositivos móviles, un dispositivo móvil puede utilizar la nube para recolectar, almacenar y procesar la información del dispositivo móvil” [8], en la figura 2.2 se muestra cómo se pueden utilizar los recursos de la nube desde cualquier dispositivo móvil.

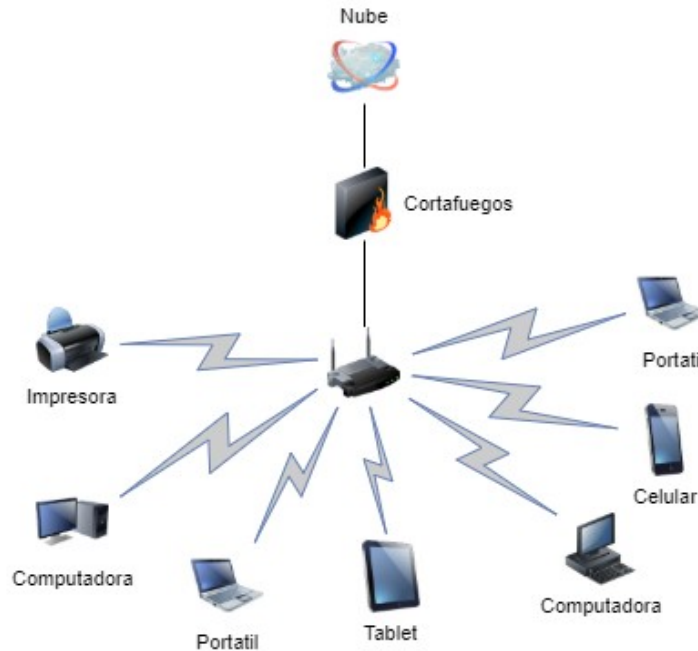


Figura 2.2 Dispositivos móviles en la nube

#### 2.1.4. Seguimiento de Trayectorias

El método de la observación directa puede ser descrita con las siguientes palabras: identifica, sigue, observa y mapea; este método involucra al investigador que acompaña al individuo bajo el escrutinio en persona, alternativamente, el observador puede seguir al observador a distancia, almacenando el patrón de sus actividades a través del tiempo, este método es llamado observador no participativa. [6]

En la actualidad el método más común de recolección de trayectorias en las ciencias sociales en general o en el turismo en particular, es con el uso de las tecnologías, es un método no observador. [6]

Global Positioning System o GPS es básicamente una serie de satélites que orbitan las señales terrestres de radio difusión capturadas por un sistema receptor, triangulando la información recibida de al menos 4 satélites, es posible determinar la ubicación del receptor, la tecnología GPS tomada como una herramienta para rastrear la actividad de los peatones donde su principal ventaja es ser global, en principio abarcando todo el mundo. [6]

Sistemas de rastreo terrestre o land-based tracking systems, esta tecnología ha existido desde mediados del siglo XX pero durante los últimos 10 años se ha hecho ampliamente disponible y mejoró su precisión, estos sistemas trabajan bajo el principio de que las señales electromagnéticas trabajan a una velocidad conocida, por lo general son redes locales ya que se distribuyen los detectores de radio frecuencia en una zona determinada. [6]

#### *2.1.4.1. Existen tres tipos principales de rastreo terrestre: [6]*

*Time Difference of Arrival (TDOA):* es la diferencia de tiempo que se tarda en llegar la transmisión entre las estaciones terrestres, esta información se transfiere a la estación central y requiere de al menos tres estaciones y de esta forma calcular la distancia.

*Angle of Arrival:* es el ángulo de llegada de la transmisión, en este caso las estaciones individuales cuentan con tres antenas, cada una en diferente dirección, esta información se transfiere a la estación central y se requiere de al menos dos estaciones para calcular la distancia.

*Cell Sector Identification:* este tipo de rastreo terrestre surgió debido a la gran difusión de las redes celulares, basándose en los principios de ángulo de llegada, identifica la ubicación del dispositivo triangulando dentro de la misma red celular.

Un estudio realizado en Hong Kong, el cual recolectó información de los log de archivos GPS utilizando un software GIS, la información que fue recolectada es desde noviembre del 2008 hasta septiembre 2009 [10]. La población objetivo fueron turistas que se hospedaban en un hotel ubicado en Kowloon, fueron turistas que no planeaban tomar algún tour y se les preguntó si deseaban participar, les dieron un GPS logger el cual estaba programado para registrar la ubicación en intervalos de 10s. [10]

Se tuvieron un total de 489 turistas, los cuales aceptaron participar, se obtuvieron 363 secuencias GPS viables para su análisis, de las cuales 233 era su primera visita y 130 con su segunda o posterior visita, de las 126 secuencias se eliminaron por problemas técnicos en el dispositivo. Las secuencias viables se analizaron usando el software comercial ESRI's ArcInc 9.3 Gis. [10]

## **2.2. Técnicas de Programación**

La programación estructurada en 1980 era la técnica de programación dominante, en la mayoría de los desarrollos de software y sus metodologías, fue formulada principalmente por Dijkstra, Wirth y Parnas, la cual cuenta de dos fases: en la primera se realiza un diseño de arriba-abajo, o refinamiento escalonado, el problema se descompone en problemas más pequeños, esta descomposición se continúa hasta que los problemas son muy simples para hacerles frente directamente y la descomposición produciría un algoritmo completamente detallado. En la segunda fase implica trabajar hacia arriba, de los objetos y funciones concretas hasta los objetos y funciones abstractas utilizados a lo largo de los módulos producidos en la primera fase. [11]

El paradigma de programación estructurada sirve para extender el uso del diseño, permitiendo construir programas que son altamente complejos para ser diseñados sin el uso de una metodología. [11]

### **2.2.1. La programación orientada a objetos**

Un objeto es un tipo de dato abstracto, el cual encapsula la información y sus acciones, los lenguajes de programación orientada a objetos proveen modularidad y una forma de ocultar la información, para esto usan dos características principales, las cuales son polimorfismo y herencia. [12]

#### *2.2.1.1. Poliformismo*

El polimorfismo se refiere a que un objeto o acción de un objeto puede tomar diferentes formas o comportamientos, y la herencia dice que todas las clases tienen una clase padre o superclase, clases con la misma superclase por lo general están relacionadas cercanamente. [12]

También se puede definir a los objetos como una colección de operaciones que comparten su estado, estas operaciones determinan a qué llamadas el objeto puede responder mientras que su estado permanece oculto desde afuera de la clase y solo es accesible por las operaciones del objeto, las variables o atributos muestran el estado interno del objeto y son llamadas variables de instancia, mientras que sus operaciones son llamadas métodos y así el conjunto de sus métodos determinan su interfaz o comportamiento. [13]

#### *2.2.1.2. Clases*

Las clases son la plantilla de la cual se crean los objetos, se puede decir que un objeto es la instancia de una clase [13]. Un ejemplo de eso puede ser la clase estudiante, con sus atributos y métodos y una instancia de la clase estudiante puede ser el estudiante “Juan” en el cual se define el valor de sus atributos.

#### *2.2.1.3. Herencia*



La herencia también permite el reuso del comportamiento de una clase en otras clases, una sub clase o clase que heredó el comportamiento de una clase padre se le puede agregar comportamiento nuevo o nuevas variables de instancia. [14]

Las características de los sistemas desarrollados en el paradigma orientado a objetos deben tener las siguientes características: [15]

*Adaptabilidad:* se refiere a la facilidad de transporte de un sistema a otro, en java se ejecutan sobre una máquina virtual (Java Virtual Machine o JVM), por lo tanto se puede transponer o adaptar a cualquier sistema que utilice la misma máquina virtual.

*Reusabilidad:* total o parcial, puede reducir costos en cuanto a tiempo de desarrollo reutilizando componentes de versiones anteriores cuya fiabilidad esté comprobada.

*Mantenibilidad:* esta característica se basa en que los programas sean fáciles de comprender y mantener ya que desde el desarrollo inicial del software es común que las personas que lo desarrollan cambien y el software siga requiriendo mantenimiento en base a mejoras o correcciones.

### **2.2.2. Programación orientada a aspectos**

La programación orientada a aspectos se basa en la idea de que los sistemas computacionales se programan de mejor forma especificando por separado las propiedades o áreas de interés del sistema y la descripción de sus relaciones, y confiar en los mecanismos de la programación orientada a aspectos para enlazarlos de forma coherente. Las propiedades o áreas de interés pueden ir desde un nivel alto como la seguridad, hasta un nivel bajo como el almacenamiento de cache y búfer. [16]

Mientras que la tendencia de la programación orientada a objetos es encontrar la uniformidad entre clases y empujar hacia arriba en el árbol de herencia, la programación orientada a aspectos intenta realizar las propiedades o áreas de interés como elementos de primera clase y expulsarlos de la estructura de objetos. [16]

La técnica de programas orientada a aspectos es como otras técnicas de programación, debe abordar tanto lo que el programador pueda decir y cómo el sistema realizará el programa en un sistema de trabajo, por lo tanto un objetivo de los sistemas orientados a aspectos no es solo proveer la forma de expresar las propiedades o áreas de interés en el sistema, sino también que estas propiedades sean sencillas y con implementaciones eficientes. [16]

#### *2.2.2.1. Consideraciones al utilizar la técnica de programación orientada a aspectos*

Los enfoques de la programación orientada a aspectos pueden examinar el programa y aspectos internos, produciendo una mezcla, por otro lado el enfoque de caja negra tiene componentes de cubierta con envolturas de aspecto.

Un sistema orientado a aspectos especifica los aspectos definiendo los puntos de unión, los lugares donde el aspecto interactúa con el sistema, parametrización del aspecto, la medida en que los aspectos pueden ser personalizados, encapsulación fuente, los requerimientos del código fuente para especificar los puntos donde se unen, y la dependencia del paradigma orientado a objetos, la medida en que el mecanismo orientado a aspectos puede utilizarse en sistemas no orientados a objetos. [16]

#### **2.2.3. Tipos de datos en java**

Los tipos de datos primitivos se utilizan para obtener los tipos de datos objeto por lo cual del correcto uso de los datos primitivos dependerá de que el código sea fiable y las variables estén bien definidas. [15]

Un identificador es un nombre que se asigna a los distintos elementos de un programa como pueden ser variables, funciones, etc. Los identificadores pueden incluir en java dígitos del 0 al 9, letras de la 'a' a la 'z' tanto mayúsculas como minúsculas, el signo "\_" y "\$". Lo único que no permite java es que el identificador no empiece con número. [15]

Las palabras reservadas son identificadores que el lenguaje de programación toma con algún significado para el compilador, en la siguiente tabla se muestran las palabras reservadas en java[15]:

Tabla 1.2. Palabras reservadas en Java

<b>Abstract</b>	<b>Const</b>	<b>float</b>	<b>Int</b>	<b>public</b>	<b>throw</b>
<b>Boolean</b>	Continue	for	Interface	return	throws
<b>Break</b>	Default	goto	Long	short	transient
<b>byte</b>	Do	if	Native	static	try
<b>case</b>	Double	else	New	super	void
<b>catch</b>	Extends	implements	Package	switch	volatile
<b>char</b>	Final	import	Private	synchronized	while
<b>class</b>	Finally	instanceof	Protected	this	

Una variable es la representación alfanumérica de una posición en memoria, tiene tres características: posición de memoria que almacena el valor, tipo de datos almacenados y nombre que se refiere a esa posición en memoria. [15]

En java hay ocho tipos de datos primitivos, de los cuales se construyen los demás[15]:

Tabla 2.3. Datos primitivos en Java

Tipo	Bits	Contenido
Byte	8	Entero de octeto
Short	16	Entero corto
Int	32	Entero
Long	64	Entero largo
Float	32	Real en formato IEEE 754
Double	64	Real en formato IEEE 754 extendido
Char	16	Carácter Unicode
Boolean	8	Lógico (true o false)

## 2.3. Bases de datos

Con el procesamiento de datos se impulsó el crecimiento de las computadoras, la automatización de tareas de procesamiento de datos precede a las computadoras. Las tarjetas perforadas fueron inventadas por Hollerith, a principios del siglo XX, para registrar los datos del censo en los Estados Unidos, desde entonces, las técnicas de almacenamiento de datos han evolucionado con el tiempo. [17]

En la década de 1950 y comienzos de 1960, se desarrollaron las cintas magnéticas, éstas ayudaron con el almacenamiento de datos en cintas de tareas como la automatización de nóminas, el procesamiento de datos consistía en leer datos de las cintas y escribir en nuevas cintas, las cintas solo se podían leer de forma secuencial y el tamaño de datos que se guardaba era mínimo.

A finales de la década de 1960 y la década de 1970, el uso de discos físicos a finales de la década de 1960 cambió en gran parte la forma en que se realizaba el procesamiento de datos ya que los discos físicos permitían el acceso directo a los datos, con el uso de estos discos se pudieron desarrollar las bases de datos en red y jerárquicas. En 1970 en un artículo histórico de Codd se definió el modelo relacional y nacieron las bases de datos relacionales. [17]

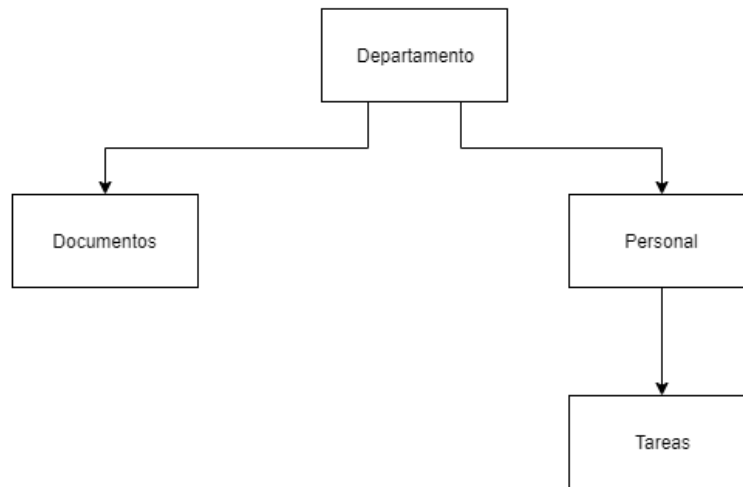
En la década de 1980, las bases de datos relacionales no pudieron competir con las jerárquicas y en red, hasta la salida de un proyecto de IBM “System R”, el cual desarrolló técnicas para la construcción de un sistema de base de datos relacional eficiente. En esta década el dominio del modelo relacional consiguió el reinado entre todos los modelos de datos, también se realizó una gran investigación en las bases de datos paralelas y distribuidas así como el comienzo de las bases de datos orientadas a objetos. [17]

A principios de 1990, el diseño del lenguaje SQL se realizó para las aplicaciones de toma de decisiones, el objetivo de las bases de datos en esta época fueron las aplicaciones de procesamiento de transacciones, muchas vendedoras de bases de datos introdujeron productos de bases de datos paralelas y comenzaron a ofrecer bases de datos orientadas a objetos. [17]

A finales de la década de 1990, con el uso de la *World Wide Web* las bases de datos se implementaron más eficientemente, ahora soportan tasas de transacciones más altas, alta fiabilidad y disponibilidad 24x7. [17]

En la implementación de bases de datos se utiliza el concepto de modelo de datos el cual es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. [17]

Modelo Jerárquico: en este modelo se organiza la información de forma jerárquica padre – hijo, se relacionan de forma que los nodos hijo puedan tener un solo padre y los nodos padre varios hijos, las entidades de este modelo se llaman segmentos y los atributos campos, la forma visual de este modelo es de árbol invertido como se muestra en la imagen siguiente. [18]



*Figura 2.3 Modelo Jerárquico*

### **2.3.1. Modelo en red**

Este modelo organiza la información entre registros y enlaces, los registros son las entidades del modelo entidad relación, los registros almacenan datos utilizando atributos y los enlaces permiten relacionar los registros de la base de datos. [18]

### **2.3.2. Modelo entidad relación**

Este modelo se basa en una percepción del mundo real que consta de una colección de objetos básicos llamados entidades y relaciones, una entidad es una cosa u objeto que es diferente de otros objetos, se describen en una base de datos mediante un conjunto de atributos, en tanto que una relación es una asociación entre varias entidades [17].

Un esquema de desarrollo entidad-relación puede definir ciertas restricciones a las que los contenedores de la base de datos deben adaptarse, hay dos tipos de restricciones [17]:

#### **2.3.2.1. Correspondencia de cardinalidades**

Representa el número de entidades a las que otra entidad puede estar asociada.

- Uno a uno: una entidad A se asocia a lo mucho con una entidad B y una entidad B se asocia a lo mucho con una entidad A.
- Uno a varios: una entidad A se asocia a cualquier número de entidades B, sin embargo una entidad B se asocia a lo mucho con una entidad A.
- Varios a uno: una entidad A se asocia a lo mucho con una entidad B, sin embargo una entidad B se asocia con cualquier número de entidades A.
- Varios a varios: cualquier número de entidades A se asocian con cualquier número de entidades B y cualquier número de entidades B se asocian con cualquier número de entidades A.

#### 2.3.2.2. *Restricciones de participación*

La participación de un conjunto de entidades E en un conjunto de relaciones R se dice que es el total si cada entidad en E participa al menos en una relación en R, de lo contrario se llama parcial.

Modelo Relacional: en el modelo relacional se utilizan un grupo de tablas para representar los datos y las relaciones entre ellas, cada tabla está compuesta por columnas y cada columna tiene un nombre único, este modelo está basado en registros, cada tabla contiene registros de un tipo en particular, cada tipo de registro define un número fijo de campos o atributos. Este modelo de datos es el más ampliamente usado, y una gran mayoría de sistemas de base de datos se basan en este modelo. [17]

Para pasar el modelo entidad relación al modelo relaciones se realizaron los siguientes cambios: las entidades pasan a ser tablas, los atributos pasan a ser columnas de las tablas, los identificadores primarios pasan a ser claves primarias y los identificadores candidatos pasan a ser claves candidatas. [18]

Algunos posibles problemas que pueden ocasionar fallas en el diseño de la base de datos con este modelo, son la redundancia, que se refiere a que datos que se repiten en las tablas de la base de datos, la ambigüedad que se refiere a que los datos no clarifican suficientemente el registro al que representan, la pérdida de restricciones de integridad y anomalías en operaciones de modificación de datos, que hace alusión al hecho de que al insertar un elemento haya que repetir tuplas en una tabla para variar unos pocos datos, o al eliminar que se tengan que eliminar otras tuplas. [18]

Las teorías de formas normales es una teoría matemática que en la actualidad se representa de forma intuitiva, fue iniciada por Codd en 1970 y continuada por otros autores. [18]

La primera forma normal impide que un atributo pueda tomar más de un valor, la segunda forma normal ocurre cuando la tabla se encuentra en primera forma normal y cada atributo que no sea clave depende de forma funcional completa respecto de cualquiera de las claves, la tercera forma normal ocurre cuando una tabla está en segunda forma normal y ningún atributo depende de la clave de forma transitivamente. [18]

La forma normal de Boyce-Codd ocurre cuando una tabla esta en tercera forma normal y además todo determinante es una clave candidata, la cuarta forma normal ocurre cuando una tabla está en la forma normal de Boyce-Codd y toda dependencia multivaluada es una dependencia funcional y la quinta forma normal es la forma más compleja y polémica de todas, fue definida por Fagin, se debe a restricciones muy



concretas si ocurre una restricción especial: la dependencia que produce esta restricción es lejana y se llama de reunión, una tabla se encuentra en quinta forma normal si hay una descomposición de esa tabla que muestra la misma información que la original.

Modelo de datos orientado a objetos: este modelo de datos está recibiendo una atención creciente, se puede observar como una extensión del modelo entidad relación con las nociones de encapsulamiento, métodos e identidad de objetos. [17]

Modelo de datos relacional orientado a objetos: este modelo combina las características del modelo de datos orientado a objetos y el modelo de datos relacional. [17]:

## **2.4. Arquitectura de software**

### **2.4.1. Patrones de diseño**

Desde su creación a principios de los 90's (llamados patrones en ese tiempo), fue adoptado por la comunidad de software con gran aceptación, por científicos y practicantes, la aceptación ha sido ayudada por las ideas intuitivamente atractivas detrás de dichos patrones, hay muchos casos de éxito de compañías que los utilizaron. [19].

Aspectos importantes para la selección de un patrón de diseño según:

Basic Elements: los elementos esenciales de un patrón son: nombre del patrón, problema que denota en qué situación se puede aplicar dicho concepto, solución el cual provee una descripción abstracta de un sistema genérico de elementos y sus relaciones para resolver el problema, y las consecuencias que implica el uso de dicho patrón, por ejemplo sus costos y beneficios.

**Critical Success Factors:** existen muchos factores de criterio de éxito para la aplicación de patrones, las investigaciones empíricas muestran que el entendimiento del estado del arte es uno de estos factores, la decisión del patrón a utilizar se basa mucho en juicio personal para aplicar el patrón en una situación particular.

**Design Options:** un patrón no es necesariamente una buena práctica, es más bien un enfoque que proveen los expertos en el área, por lo tanto un patrón es un conjunto de opciones de diseño durante el desarrollo de software. La decisión de un patrón de diseño se basa en los objetivos de calidad del proyecto de software.

**Flexibility:** la flexibilidad es un concepto clave en los patrones.

**Cost:** los patrones conducen generalmente a un número mayor de artefactos de software, que normalmente aumentan la complejidad del software, además cuando las asociaciones se instancian en tiempo de ejecución y esto aumenta la complejidad dinámica del software.

**Software Quality:** el concepto calidad de software en si denota un concepto multidimensional, frecuente mente se mencionan los siguientes atributos de calidad: flexibilidad, mantenibilidad, comprensibilidad, etc.

**Software Lifecycle:** para lidiar con la incertidumbre en un entorno dinámico, es crear flexibilidad en nuestro producto, para que pueda modificarse fácilmente, los patrones proporcionan una forma de crear flexibilidad en un producto de software.

**Removal:** en el caso de eliminar un patrón del código se tienen que analizar tres cuestiones: el posible beneficio del patrón, el posible costo extra del patrón y el costo de su eliminación.

Empirical Research: el uso de patrones no estaba planeado para la investigación en un inicio y aún no está listo, se puede ver más bien como un practicante electrico, esto se refleja más bien en la falta de investigación sobre patrones, contrariamente a la creencia común los efectos benéficos de los patrones no son universalmente obvios. Por lo tanto, hay concluyeron que en la situación actual existe una falta grave de evidencia científica en el uso de patrones.

Los patrones de diseño se categorizan en tres tipos[20] patrones Creacionales: esta categoría se de patrones se identifica con creación de nuevas instancias (ej Factory), patrones Estructurales: la implementación de este tipo de patrones se basa en el diseño de las relaciones estructurales entre clases (ej Bridge) y patrones de Comportamiento: este tipo de patrones se encargan de realizar las soluciones comunas (Ej Command).

The Proxy Pattern es un patrón simple, con una lógica fácil de entender, es un típico patrón para principiantes, la idea básica detrás de este patrón es utilizar un marcador de posición llamado proxy para otro objeto llamado sujeto real para controlar su acceso. El proxy constituye una capa adicional de indirección y gestiona las peticiones realizadas por los clientes al sujeto real [19].

Problemas con el uso de este patrón[19]: el uso de este patrón en muchos casos se justifica por la necesidad de flexibilidad a futuro, cuya necesidad no se materializa y la razón detrás de su uso desaparece, este patrón conduce a un aumento en el número de clases, por lo general su implementación nos lleva al uso de una interfaz y 2 clases concretas, esto nos lleva a que el tamaño y la complejidad del software aumente considerablemente, en el caso de que se proponga realizar un pre procesamiento y post procesamiento con este patrón de diseño, puede hacer que la interacción entre objetos sea bastante complicada y este patrón introduce un nivel adicional de indirección que impide la comprensión del flujo dinámico del control en tiempo de ejecución debido a que hincha la pila de ejecución, también puede hacer la depuración más lenta.

The Observer Pattern es muy conocido y se originó en el entorno SmallTalk donde se había desarrollado para facilitar el diseño de la interfaz gráfica y donde se conoce como el patrón modelo vista controlador (MVC). El enfoque original MVC divide una aplicación grafica interactiva en dos partes fundamentales: una aplicación abstracta (modelo) y una interfaz de usuario (vista y controlador), MVC define un protocolo genérico para la comunicación de sus tres capas, hay dos aspectos importantes en este enfoque, el primero es las dependencias de uno a muchos entre un modelo y varias vistas y la segunda es facilitar la reutilización del software y la división del trabajo entre los objetos participantes[19].

La programación MVC es la aplicación de la factorización de tres capas [21] el Modelo donde los objetos de diferentes clases asumen las operaciones relacionadas con el dominio de la aplicación, esto puede ser tan simple como un entero o una cadena, o puede ser un objeto complejo que es la instancia de una subclase, la vista la cual es la visualización del estado de la aplicación, se tratan de todo lo gráfico, solicitan datos de un modelo y muestran estos datos y el controlador el cual se encarga de la interacción del usuarios con el modelo y la vista, es la interfaz entre los modelos, vistas asociadas y dispositivos de entrada, se ocupan también de programar la interacción entre vistas.

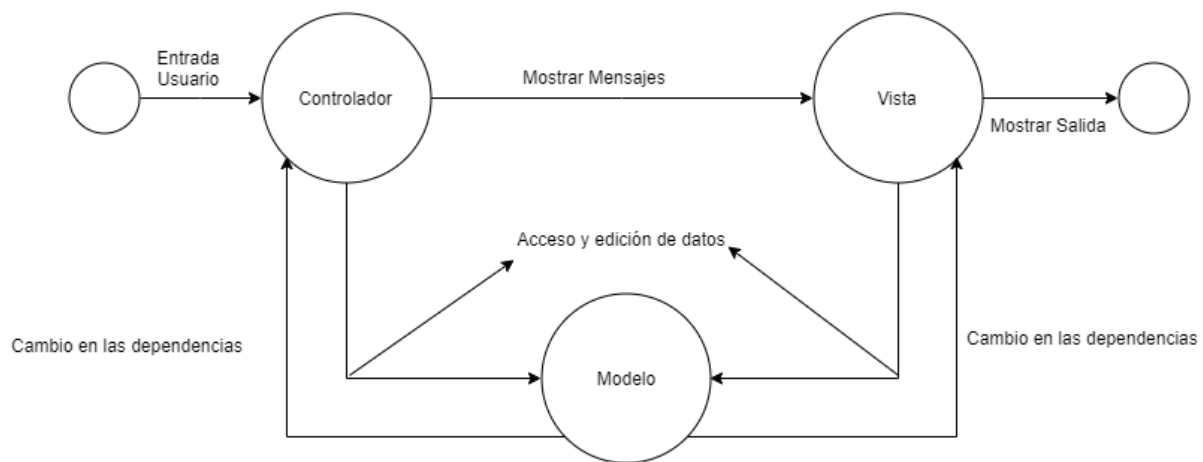


Figura 2.4 Ejemplo envío de mensajes MVC

The Bridge Patter: este patrón proporciona una forma de desacoplar una abstracción y su implementación, la abstracción no es un valor en sí mismo, sino que ayuda a la comprensión y comunicación humana, por lo tanto un diseño de software debe contener un grado útil de abstracción, muchos lenguajes de programación proporcionan características que permiten un cierto grado de independencia entre la abstracción y la implementación. [19]

Este patrón se caracteriza por dos jerarquías paralelas, una es una versión abstracta del modelo y la otra es la implementación, como se muestra en la siguiente imagen[20]:

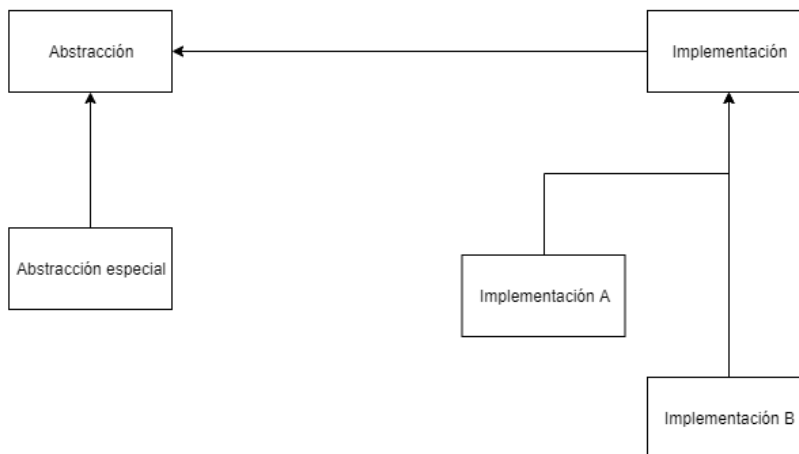


Figura 2.5 Patrón de puente

The Command Pattern: este patrón se basa en la representación de comandos por objetos, por lo tanto es ideal para representar las estructuras de comandos complejas mediante un modelo de objetos flexible, este enfoque puede utilizarse para diseñar aplicaciones que sean fácilmente extensibles. [19]

The Strategy Pattern: este patrón se basa en la delegación de un algoritmo a una interfaz o una clase donde la elección de una subclase es posible, a continuación se muestra una imagen de este patrón [20]:

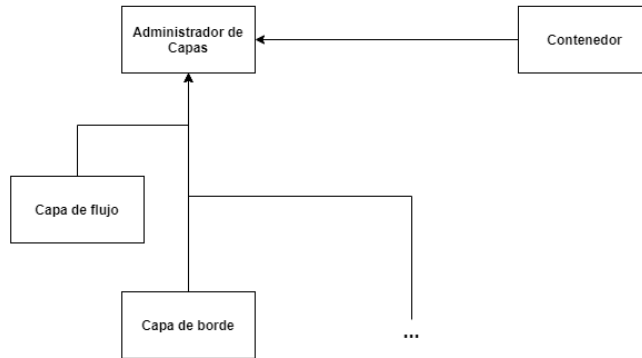


Figura 2.6 Patrón de Estrategia

## 2.5. Frameworks

### 2.5.1. Hibérnate

Cuando trabajas con una base de datos SQL en una aplicación java utilizamos la conexión vía JDBC Api (Java DataBase Connectivity) [22], en este proceso entramos en el conflicto de que la base de datos se encuentra en un modelo relacional y nuestra aplicación java trabaja en un modelo orientado a objetos, para usamos el framework Hibernate.

Hibérnate provee la opción de hacer el mapeo de tablas en clases, pasa la información de la base de datos a las clases y provee la opción de poder manipular la base de datos, lo cual llamaremos persistencia[22].

Hibernate ORM o mapeo relacional de objetos es la persistencia automática y transparente de los objetos en una aplicación java a las tablas en una base de datos relacional usando metadatos para describir el mapeo entre los objetos y la base de datos, ORM se implementa como un middleware y tiene muchas oportunidades de mejora que una implementación a mano de la capa de persistencia y su costo es menor, una solución ORM consiste de [22] una librería de las operaciones con objetos básicas de la persistencia de clases, un lenguaje o librería para la especificación de consultas de clases y sus propiedades, una forma de especificar el mapeo de los metadatos y una técnica para la implementación ORM para interactuar con objetos transaccionales para realizar dirty checking, lazy association fetching y otras funciones de optimización.

Problemas con el uso de ORM genéricos [22] ¿Cómo deben lucir las clases persistentes?, ¿Cómo se define el mapeo de los metadatos?, ¿Cómo mapear la herencia de las clases?, ¿Cómo se relaciona la llave primaria de los objetos y de la base de datos?, ¿Cómo interactúa la lógica de persistencia en tiempo de ejecución con los objetos del dominio de negocio?, ¿Cuál es el ciclo de vida de un objeto persistente?, ¿Qué facilidades ofrece para clasificar, buscar y agregar? Y ¿Cómo obtenemos la información con asociaciones efectivamente?

Razones por las que es importante el uso de un ORM [22] productividad debido a que el código persistente puede ser el código más tedioso en java, hibernate elimina la mayor parte y te deja concentrarte en la regla de negocio y reduce significativamente el tiempo de desarrollo, mantenibilidad nos ayuda la reducción en las líneas de código porque hace el sistema más entendible y cambiar más fácilmente y con la automatización de la persistencia se reduce sustancialmente la cantidad de líneas de código y claro que tomar el conteo de líneas de código de una aplicación como medida de complejidad de la aplicación es una métrica en discusión. Sin embargo no es la única razón de porque usar hibernate, en sistemas donde la persistencia se hace a mano existe una tensión inevitable entre la representación relacional y el modelo de objetos que implementa el dominio, el cambio en un modelo

implica el cambio en el otro, con el uso de ORM se aíslan los modelos e implican menos cambios, rendimiento porque la persistencia codificada a mano siempre es al menos tan rápida como la persistencia automatizada, así como el código en ensamblador siempre es al menos igual de rápido que el código desarrollado en java, aquí lo que hay que considerar es la cantidad de esfuerzo utilizada y considerar el tiempo y restricciones presupuestales e independencia del vendedor por el uso de un ORM, si la herramienta soporta varias bases de datos diferentes (la mayoría lo hacen), entonces esto le da un nivel de portabilidad a la aplicación.

### **2.5.2. Spring**

El objetivo de spring es desarrollar el software de una aplicación web de forma precisa, económica y eficiente, también es importante que sea simple, testeable y portable. Lo simple que sea desarrollar y mantener el código facilita la eficiencia del código a través del tiempo. [23]

Spring es una solución ligera y potente para construir tus aplicaciones empresariales, es modular permitiéndote usar solo los módulos que necesitas, puede ser usado con Struts, Hibernate o JDBC entre otros, también permite la gestión de transacciones declarativas, el acceso remoto a tu lógica mediante RMI o servicios Web y varias opciones para la persistencia de datos. Spring ofrece un maco Modelo Vista Controlador (MVC) con todas sus funciones. [24]

El uso del Spring permite construir aplicaciones desde “Plain Old Java Objects” o POJOs y aplicar servicios empresariales de forma no invasiva a los POJOs, utilizando java SE y EE. Las aplicaciones Java consisten en objetos que colaboran para formar la aplicación, la plataforma java carece de una forma para organizar los bloques de construcción, dejando esta tarea a los arquitectos y/o desarrolladores. Es cierto que se pueden utilizar los patrones de diseño para solucionar este problema, pero estos solo son las mejores prácticas. [24]



El componente de inversión de control en Spring aborda esta preocupación proporcionando un método formalizado de unir componentes dispersos en una aplicación funcional y lista para su uso, Spring codifica los patrones de diseño como objetos de primera clase que pueden integrarse en nuestra aplicación. [24]

Spring se componen de varios módulos agrupados de la siguiente forma[24]: Core Container, Data Access/Integration, Web, AOP (Programación orientada a aspectos) e Instrumentation.

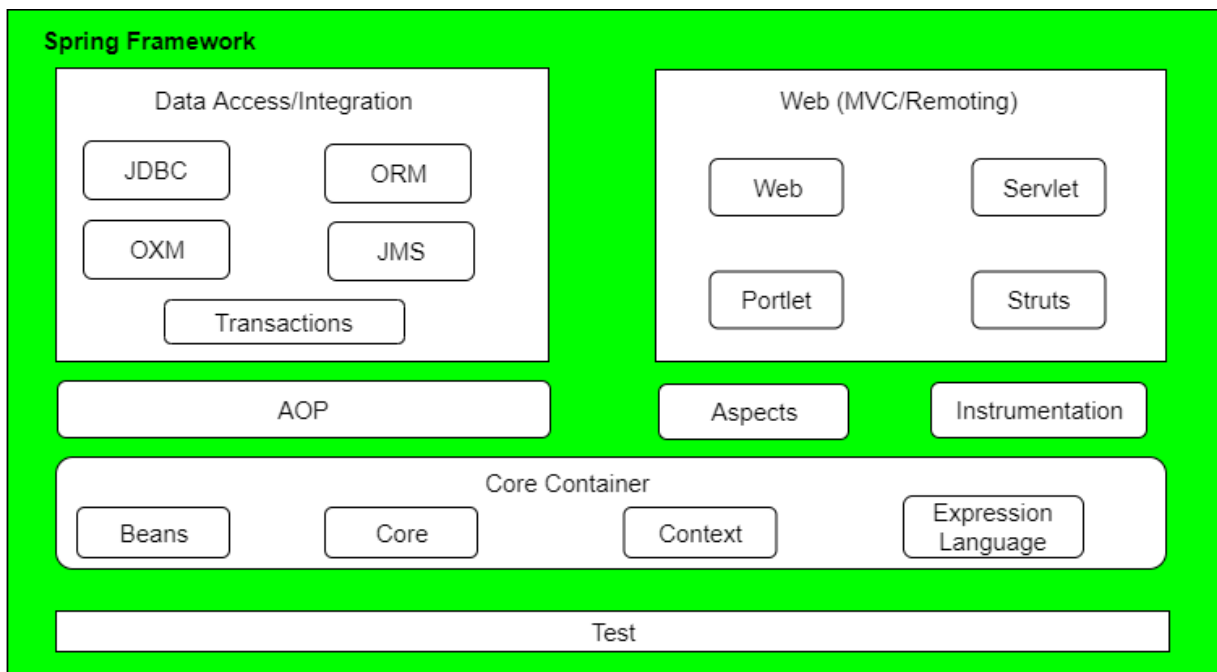


Figura 2.7 Spring Framework Runtime

El acceso a datos en este framework se forma en los siguientes módulos: administrador de transacciones, soporte de “Data Access Objects”, acceso a los datos mediante JDBC, acceso a los datos mediante ORM y Marshalling XML usando O/X Mappers[24].

Tradicionalmente los desarrolladores Java EE tenían dos formas de administrar las transacciones de manera global o local. Las transacciones globales nos permiten trabajar con múltiples recursos transaccionales, típicamente bases de datos relacionales y colas de mensajes, el uso de este tipo de transacciones se limita en el reusó del código de la aplicación solo en el ambiente del servidor de aplicación. Las transacciones locales son recursos específicos, se asocian a la conexión mediante JDBC, son más fáciles de usar, pero no pueden trabajar con recursos múltiples y son invasivas al modelo de programación. [24]

Resolviendo las limitaciones de las transacciones locales y globales, spring permite a los desarrolladores de aplicaciones un modelo de programación consistente en cualquier ambiente, este framework ofrece administración de transacciones programáticas y declarativas, aunque las declarativas son las más recomendadas en la mayoría de los casos[24]. La clave de la abstracción de transacciones en spring es la noción de una estrategia transaccional, esta estrategia está definida por la interfaz `org.springframework.transaction.PlatformTransactionManager`. [24]

El soporte de DAO o (Data Access Objects) se realiza para facilitar el uso de tecnologías como JDBC, Hibernate, JPA o JDO de una manera consistente. [24]

El soporte para el acceso a los datos que ofrece el framework spring para las conexiones JDBC se muestra en la siguiente tabla[24]:

Tabla 2.2 Spring JDBC

Acción	Spring	Tu
Definir los parámetros de conexión		X
Abrir la conexión	X	
Ejecutar la sentencia SQL		X
Declarar parámetros y proveer valores a los parámetros		X
Preparar y ejecutar las sentencias	X	
Configurar los ciclos para iterar en el conjunto de resultados	X	

Hacer el trabajo en cada iteración		X
Manejar las excepciones	X	
Manejar transacciones	X	
Cerrar la conexión, sentencias y conjunto de resultados	X	

### 2.5.3. Spring Security

Spring Security es una de las librerías en el universo de java, estándares como la autenticación en java (JAAS) o la seguridad de java EE son una forma de realizar la autenticación o autorización, spring security contiene todo lo que se necesita para implementar una solución de seguridad en tu aplicación porque se implementa de principio a fin y de forma consistente y sensible. [25]

Algunos de los conceptos clave relacionados con la seguridad en el desarrollo de aplicaciones [25] es autenticación la cual es el proceso de verificar que los usuarios de la aplicación sean realmente quien dicen que son, se puede mostrar en el día a día en la autenticación basada en credenciales con el uso de usuario o correo electrónico y contraseña, autenticación de dos factores este tipo de autenticación es similar a la basada en credenciales solo que el usuario esta encriptado en la tira magnética de la tarjeta. Y autenticación por hardware un claro ejemplo de este tipo de autenticación es cuando arrancas un automóvil utilizando la llave y encendiendo el carro, otro concepto clave es la autorización típicamente incluye dos factores que combinados describen la accesibilidad del sistema de seguridad, el primero es el mapeo de la autenticación inicial con 1 o más roles y el segundo es el uso de los recursos de seguridad.

### 2.5.4. Servidores Web

La arquitectura orientada a servicios (SOA) es una arquitectura que representa la funcionalidad del software como servicios en la red, los servicios web se definen como un conjunto de estándares como: SOAP (Simple Object Access Protocol) protocolo para la comunicación entre aplicaciones, UDDI (Universal Description

Discovery Integration) define un registro para que los proveedores de servicios publiquen sus servicios, WSDL (Web Services Description Language) para describir las capacidades de un servicio web y la interfaz para invocarlo, que permiten a las aplicaciones interactuar entre sí en la red[26]. A continuaciones veremos dos tipos de servidores web: servidores web “Big” y servidores web REST.

Los servidores web “BIG” fueron construidos para trabajar la comunicación entre clientes y servidores web usando XML, permite a los desarrolladores escribir orientado a mensaje, así como servidores web orientados a llamada de procedimientos remotos (RPC-oriented). [27]

El API para la programación de este tipo de servidores es JAX-WS, la invocación de una operación de este tipo de servidores web involucra un protocolo basado en XML como SOAP, las especificaciones SOAP definen la estructura, las reglas de codificación, y las convenciones para la representación de las invocaciones y respuestas del servidor web, las cuales se representan como mensajes SOAP en HTML. [27]

Los servidores web REST fueron construidos para trabajar mejor en la web, el significado de este tipo de servidor web REST es transferencia de estado representacional o Representational State Transfer es un estilo de arquitectura que maneja restricciones tales como una interfaz uniforme la cual indica propiedades deseables tales como rendimiento escalabilidad y modificabilidad. [27]

En el estilo arquitectónico REST los datos y la funcionalidad se consideran recursos y se adquieren mediante URI (Uniform Resource Identifiers o Identificadores Uniformes de Recursos), normalmente enlazados en la web y los recursos se manejan utilizando un conjunto de operaciones simples y bien definidas.

El estilo arquitectónico REST es una arquitectura cliente/servidor y está diseñada para utilizar un protocolo de comunicación sin estado como HTML, los clientes y servidores intercambian recursos mediante el uso de una interfaz y un protocolo normalizados.

Los siguientes principios alientan las aplicaciones REST a ser simples, ligeras y rápidas [27]:

- Identificación de recursos mediante URI: un servicio web REST expone un conjunto de recursos que identifican los objetivos de la interacción con sus clientes, los recursos se identifican mediante URI, que proporcionan un espacio de direccionamiento global para el descubrimiento de recursos y servicios.
- Interfaz uniforme: los recursos se manipulan utilizando un conjunto fijo de cuatro operaciones de creación, lectura, actualización y eliminación: PUT, GET, POST y DELETE.
- PUT crea un nuevo recurso, que puede eliminarse mediante DELETE.
- GET recupera el estado actual de un recurso en alguna representación.
- POST transfiere un nuevo estado a un recurso.
- Mensajes auto descriptivos: los recursos se desacoplan de su representación para que su contenido pueda ser accedido en una variedad de formatos, como HTML, XML, texto sin formato, PDF, JPEG, JSON y otros. Los metadatos sobre el recurso están disponibles y se utilizan, por ejemplo, para controlar el almacenamiento en caché, detectar errores de transmisión, negociar el

formato de representación adecuado y realizar la autenticación o el control de acceso.

- Interacciones de estado a través de hipervínculos: toda interacción con un recurso es apátrida, es decir, los mensajes de solicitud son independientes. Las interacciones de estado se basan en el concepto de transferencia estatal explícita. Existen varias técnicas de intercambio de estado, como URI reescritura, cookies y campos de formulario ocultos. Estado se puede incrustar en mensajes de respuesta para señalar a estados futuros válidos de la interacción.

El API para la programación de servicios web RESTful en java es JAX-RS, se basa en el uso de anotaciones para simplificar el desarrollo de servidores web RESTful, dichas anotaciones se utilizan en tiempo de ejecución, a continuación se muestra una lista de las anotaciones utilizadas por JAX-RS [27]:

**@Path** – Esta anotación se utiliza para indicar el valor relativo de la ruta del URI que indica donde está almacenado el código java, **@GET** – Esta anotación es diseñada para implementar el método HTTP get como petición al recurso java, **@POST** – Esta anotación es diseñada para implementar el método HTTP post como petición al recurso java, **@PUT** - Esta anotación es diseñada para implementar el método HTTP put como petición al recurso java, **@DELETE** - Esta anotación es diseñada para implementar el método HTTP delete como petición al recurso java, **@HEAD** - Esta anotación es diseñada para implementar el método HTTP head como petición al recurso java, **@PathParam** – Esta anotación es un tipo de parámetro que puede ser extraído para su uso en tu recurso java, los parámetros de la ruta de tu URI y sus nombres corresponden al templete de nombres especificado en las anotaciones del **@Path**, **@QueryParam** - Esta anotación es un tipo de parámetro que puede ser extraído para su uso en tu recurso java, estos parámetros son extraídos de la

petición URI de sus parámetros de consulta, @Consumes – Esta anotación se utiliza para especificar las representaciones de los tipos de medios MIME que un recurso puede consumir y fueron enviados al cliente, @Produces - Esta anotación se utiliza para especificar las representaciones de los tipos de medios MIME que un recurso puede consumir y fueron enviados al cliente y @Provider – Esta anotación se utiliza para cualquier cosa que sea de interés en tiempo de ejecución para JAX-RS, como MessageBodyReader y MessageBodyWriter.

SOAP y REST son métodos de comunicación entre dos aplicaciones que permiten el acceso de información o datos, SOAP es una especificación del protocolo del intercambio de información estructurada en redes computacionales y REST es un estilo arquitectónico del 'World Wild Web' e intenta simplificar el método de solicitar información[28].

Diferencias entre algunos aspectos clave [28]:

Formato: SOAP se basa en XML exclusivamente, y utiliza protocolos de comunicación como HTML y SMPT para intercambiar información entre aplicaciones, es más fácil para los modelos SOAP

Uso: SOAP es más usado en las aplicaciones empresariales para su integración con una aplica gama de sistemas y REST se basa en las URI's y HTTP integradas en 'World Wild Web'.

Peticiones: en REST las peticiones se basan en la petición de objetos usando GET y en la modificación de estos objetos usando POST, PUT o DELETE. Por su parte SOAP requiere de un XML para petición o respuesta de información, requiere del conocimiento de la especificación del archivo XML y una herramienta para el parseo de los resultados. REST regresa la información mediante JSON mientras que SOAP en una estructura de datos XML.

Seguridad: la información debe de manejarse de manera segura y no puede ser enviada como parámetro en una URI, como las peticiones REST se basan en el protocolo HTTP, el administrador, firewall o cualquier intermediario puede analizarla, las peticiones GET se pueden considerar seguras debido a que solo consultan información. SOAP usa POST para transmitir la información, no hay forma de obtener la información que se está consultando o afectar la base de datos. SOAP provee más herramientas de seguridad tales como WS-Security.

Transferencia de Archivos Adjuntos: SOAP hace la transferencia de archivos de manera fácil, transportando datos y binarios sin algún problema, REST por otro lado puede manejar grandes cantidades de información, pero puede ser complicado y exceder los límites del URI.

Cache: las peticiones REST son simples peticiones GET, proxys intermedios o reversos pueden obtener las peticiones fácilmente, mientras que las peticiones SOAP son peticiones POST que requieren una petición XML compleja lo que hace un poco más difícil recibirla.

## 2.6. Big Data

Big Data es un concepto abstracto, aparte de tener un gran volumen de datos tiene otras características lo cual lo diferencia de grandes cantidades de información o Big Data. En 2010 Apache Hadoop definió big data como “Conjuntos de datos que no pudieron ser capturados, administrados y procesados por ordenadores dentro de determinado ámbito”[29]. También el concepto de Big Data puede ser usado para grandes conjuntos de datos que sobrepasan las capacidades de software convencional para su captura, administración y procesamiento en un corto tiempo[30]. El valor del uso de Big Data se basa en la ayuda a tomar decisiones en tiempo real, Big Data no es algo nuevo, la información ya existe, solo que ahora se utiliza de otra forma.



Como fuentes de información tenemos a las maquinas, gente y organizaciones, por parte de las maquinas se puede tomar información de los sensores en tiempo real, maquinaria industrial, vehículos que guardan sus rutas, sensores ambientales, registros personales de salud, entre otras fuentes, por parte de la gente se utilizan datos de redes sociales y de las organizaciones se toman de las bases de datos ya sean estructuradas o no estructuradas. [31]

Un dispositivo es inteligente porque [31] registra información, se conecta con otros dispositivos o redes, proporciona un contexto ambiental y recolecta y analiza datos autónomamente.

Las ventajas de la información generada por maquinas son [31]: con las notificaciones en tiempo real se pueden generar acciones en tiempo real y todo gracias a la información recolectada, así como los cambios culturales para realizar acciones en tiempo real incrementar la relación con los clientes, ayuda en la detección de fraudes y realiza el control y monitoreo del sistema.

La gente genera grandes volúmenes de información cada día con su actividad en redes sociales, compartiendo imagen y video, esta información es “text-heavy” y “Unstructured”. [31]

¿Cómo utilizar la información generada por la gente?[31] Para ayudar con el procesamiento de grandes data-sets en un ambiente computacional distribuido se diseñaron varios Frameworks (Hadoop, Apache Storm, etc).

Se considera tradicional comparada con los otros tipos de información de Big Data, pero es tan importante como los otros tipos. Las organizaciones almacenan su información para usarla en el presente o futuro, así como para analizar su pasado[31].

Se considera como Big Data a cualquier conjunto de datos grande y complejo que exceda el procesamiento y capacidad de sistemas y técnicas de administración de datos convencionales, también debe de contar con las siguientes características:

Volumen: cantidad de información generada cada segundo, minuto, hora y día de nuestro mundo digital, es la dimensión de Big Data relacionada al tamaño y su crecimiento exponencial[32]. Nos proporciona los retos con el acceso, procesamiento y almacenamiento debido al costo, escalabilidad y rendimiento.

A partir del 2012 cerca de 2,5 exabytes de datos se crean cada día y esta cifra se duplica cada 40 meses aproximadamente, más datos cruzan en internet cada segundo, esto da a las empresas la oportunidad de con muchos petabytes de datos, por ejemplo se estima que Walmart trabaja con más de 2,5 petabytes de datos cada hora de sus transacciones con clientes. [33]

Variedad: se refiere a las diferentes formas o formatos por los cuales llega la información como texto, imágenes, voz y datos geoespaciales[32], los tipos de variedad son variedad estructural en formatos y modelos, variedad del medio por el cual se obtiene la información, variedad semántica de cómo interpretar y operar los datos y por último variedad de disponibilidad en tiempo real o intermitente.

Big Data se encuentra en diferentes formatos: mensajes, actualizaciones, imágenes publicadas en redes sociales, lecturas de sensores, señales GPS de celulares, y muchos más, muchas de estas fuentes son nuevas, por lo tanto las bases de datos estructuradas que almacenan esta información no son adecuadas para almacenar y procesar Big Data. A medida que se digitaliza más y más actividad empresarial, nuevas fuentes de información y equipos cada vez más baratos nos introducen en una nueva era. Los datos disponibles a menudo son no estructurados, no organizados en una base de datos, pesados y con ruido, simplemente esperando a ser eliminados. La llegada de las analíticas trajo rigurosas técnicas para la toma de

decisiones, Big Data es más simple y más potente, como dice el director de investigación de Google, Peter Norvig “No tenemos mejores algoritmos, solo tenemos más datos”.

Velocidad: se refiere a la velocidad a la cual la información es generada y a la velocidad en la que la información es almacenada y analizada. [32]

Para muchas aplicaciones la velocidad de creación de los datos es más importante que el volumen, la información en tiempo real, o casi en tiempo real hace posible que una empresa sea más ágil que sus competidores, por ejemplo Alex Pentland y su grupo de MIT Media Lab usaron datos de localización de teléfonos móviles para inferir cuantas personas estaban en el estacionamiento de Macy en el Black Friday, esto hizo posible estimar las ventas a minoristas incluso antes de que la tienda registrara esas ventas, las ideas rápidas como esta pueden proporcionar ventaja competitiva obvia a los analistas de Wall Street y a los gerentes de Main Street. [33]

Veracidad: se refiere al sesgo, ruido y anomalías en los datos, tomando en cuenta la exactitud de los datos, confiabilidad de la fuente y contexto sin análisis.

Valencia: se refiere a la conectividad de Big Data en forma de grafos.

Valor: obtener valor de Big Data[32] la predicción con información histórica más información en tiempo real.

Pasos en el proceso iterativo de la ciencia de los datos [31]: **paso 1** adquirir información, identificar que data está disponible y es relevante para tu problema, para esto existen técnicas para la obtención de datos como las bases de datos tradicionales (PostgreSQL, Oracle, etc), archivos de texto con scripting languages (Php, Ruby, js, R, etc), datos remotos con el uso de web services (HTML5, Json, xml, etc) y almacenamiento no sql (MongoDB, cassandra, Apache Hbase, etc); **paso 2**

preparar información, explorar la información (entender la información, usando correlaciones, tendencias e identificando outliers) y pre procesar la información (remover información con valores perdidos, eliminando registros duplicados, removiendo outliers y generar estimados para datos inválidos): **paso 3** analizar la información, utilizar técnicas de análisis como: clasificación, regresión, clusterización y análisis asociativo y de grafos; **paso 4**: comunicar los resultados; **paso 5**: aplicar los resultados.

El uso de Big Data requiere de grandes tecnologías para procesar grandes cantidades de información en un corto periodo de tiempo, las tecnologías que se aplican a big data incluyen: bases de datos de procesamiento de datos masivo paralelo, redes de minería de datos, sistemas de archivos distribuidos, bases de datos distribuidas, computo en la nube, internet y sistemas de almacenamiento escalables. [30]

Una característica fundamental en Big Data es la respuesta en tiempo real o casi en tiempo real, por lo cual hay que evitar la latencia siempre que sea posible, se tienen una gran cantidad de técnicas y tecnologías que se basan en el uso de la estadística, informática, matemática aplicada y economía para agrupar, manipular, analizar y visualizar grandes cantidades de datos. [30]

El sistema distribuido de archivos de big data tiene como objetivo principal el conseguir una “High Concurrency & Low Consistency”, esto se logra utilizando un particionado y replicación de la información y de esta forma tenemos un sistema escalable, tolerante a fallos y con una alta concurrencia. [31]

El proyecto Apache Hadoop desarrolla software de código abierto, para un cómputo distribuido, confiable y escalable, la librería de software Apache Hadoop es un framework que permite el procesamiento distribuido de grandes conjuntos de datos en un cluster de computadoras, permite que las aplicaciones trabajen con miles de

computadoras y petabytes de datos, Hadoop se derivó de Google Map Reduce y Google File System. [30]

Hadoop está formado principalmente por un paradigma de programación “Map Reduce” y su sistema de archivos “HDFS”[34]. Hadoop Map Reduce es el framework más popular de código libre propuesto por google. [35]

Una de las ventajas del uso de Hadoop Map Reduce es que permite correr tareas de analítica por usuarios que no son expertos sobre big data, este framework le da control completo de como los data sets son procesados, los usuarios realizan las consultas en java en vez de SQL, hace que este framework sea usado por los desarrolladores de manera sencilla sin la necesidad de tener conocimientos en bases de datos y solo un conocimiento básico de la tecnología Java. [35]

El Sistema Distribuido de Archivos Hadoop HDFS es un sistema que provee tolerancia a fallas y está diseñado para ejecutarse en hardware común, nos proporciona un alto rendimiento de acceso a los datos de la aplicación y es adecuado para aplicaciones con grandes cantidades de datos. [30]

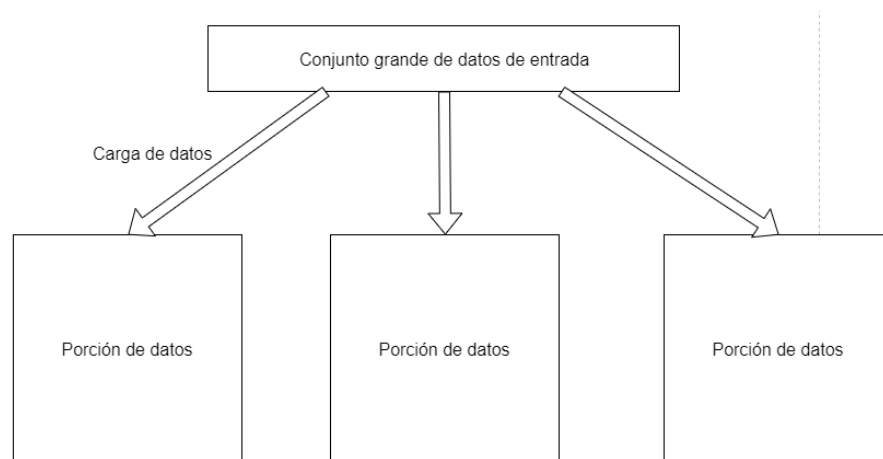


Figura 2.8 HDFS

Map Reduce es un paradigma de programación que permite escalabilidad masiva, básicamente realiza 2 tareas, map y reduce[34]: las tareas de Map reciben información de HDFS, producen una secuencia de pares clave-valor de la entrada y se realiza de acuerdo a las reglas definidas en el código de la función Map, estos valores son recogidos por el controlador maestro y se ordenan por clave y se dividen entre las tareas e Reduce. Las tareas de Reduce combinan todos los valores asociados a una clave, este proceso depende de las reglas definidas en el código de la función Reduce.

La idea de cloud computing o computo en la nube no es nueva, John McCarthy en la década de los 60's ya preveía que las instalaciones de computo se proporcionarían al público en general como utilidad, el término "nube" también se ha utilizado antes, pero fue hasta que el CEO de Google Eric Schmidt presento el modelo de negocios de prestar servicios a través de internet en el 2006, que se popularizo el término. [36]

Cloud Computing es un modelo que permite acceso a recursos computacionales configurados a través de red bajo demanda tales como almacenamiento, aplicaciones, servicios, redes y servidores, que pueden ser rápidamente asignados y liberados con un mínimo esfuerzo. [32]

El Instituto Nacional de Estándares y Tecnología (NIST), define el cómputo en la nube como un modelo de acceso conveniente bajo demanda a un conjunto compartido de recursos computacionales configurables que pueden ser asignados y liberados con un esfuerzo mínimo de gestión o un proveedor de servicios. [36]

El computo en la nube es comparado frecuentemente con las siguientes tecnologías, las cuales comparten ciertas características con el computo en la nube [36]:

Grid Computing: es un paradigma de computación distribuida que coordina los recursos en red, fue originalmente impulsado para aplicaciones científicas de

computo intensivo, se asemeja al cómputo en la nube en la utilización de recursos distribuidos, sin embargo el computo en la nube además aprovecha el uso de tecnologías de virtualización para realizar el uso compartido de recursos y la asignación de recursos dinámicamente.

Utility Computing: este modelo proporciona recursos bajo demanda y el cobro de los recursos utilizados en vez de una tarifa fija, el computo en la nube adopta su forma de cobro debido a que la asignación de recursos bajo demanda y los precios basados en su utilización, pueden maximizar la utilización de recursos y minimizar sus costos operativos.

Virtualization: es una tecnología que obtiene los detalles del hardware físico y proporciona recursos virtualizados, un servidor virtualizado se le llama comúnmente máquina virtual, la virtualización forma la base del cómputo en la nube ya que nos da la capacidad de agrupar recursos de clusters de servidores y asignar estos recursos bajo demanda.

Autonomic Computing: acuñado por IBM en el 2001, tiene como objetivo el construir sistemas computacionales capaces de auto gestionar, es decir, reaccionar a observaciones internas y externas sin intervención humana, su objetivo es superar la complejidad de gestión de los sistemas de información actuales, aunque el computo en la nube presenta ciertas características autónomas, como la asignación de recursos su objetivo es reducir los costos en vez de la complejidad.

Generalmente la arquitectura del cómputo en la nube se divide en 4 capas [36]:

Capa de Hardware: esta capa se encarga de administrar los recursos físicos de la nube, como servidores físicos, enrutadores, switches, sistemas de energía y enfriamiento, los problemas típicos de esta capa incluyen la configuración del

hardware, la tolerancia de fallos, la gestión del tráfico, la administración de recursos de poder y enfriamiento.

Capa de Infraestructura: también conocida como la capa de virtualización, crea un conjunto recursos de cómputo y almacenamiento mediante la partición de recursos físicos utilizando tecnologías de virtualización. Esta capa es un componente esencial del cómputo en la nube, ya que muchas características clave como la asignación de recursos dinámica, solo están disponibles a través de tecnologías de virtualización.

Capa de Plataforma: esta capa consiste de sistemas operativos y frameworks de aplicación, su propósito es minimizar la tarea de desplegar aplicaciones directamente en contenedores de máquina virtual.

Capa de Aplicación: es la capa más alta, consta de las aplicaciones reales en la nube, a diferencia de las aplicaciones tradicionales, las aplicaciones en la nube aprovechan la característica de escala automática para lograr un mejor rendimiento, disponibilidad y menor costo operativo.

Los modelos de servicio en la nube típicamente son [32]: PaaS, el cual se refiere a diferentes recursos en la nube para proporcionar una plataforma computacional para usuarios finales, SaaS, se refiere a aplicaciones en una infraestructura de nube remota ofrecidas por el proveedor de la nube como servicio que puede ser accedido a través de internet y IaaS, se refiere a equipamiento de hardware ofrecido como servicio mediante la nube.

Características deseadas del cómputo en la nube [37]: Self-Service: los consumidores de los servicios de cómputo en la nube esperan los recursos bajo demanda y de acceso casi instantáneo, para satisfacer esta necesidad la nube permite auto servicio para los clientes de peticiones, especialización, pago y uso sin la intervención de operadores humanos. Pay-Usage Metering and Billing: el pago de



los servicios se realiza solo por el tiempo que se ocupa el servicio. Elasticity: el cómputo en la nube da la ilusión de que se cuenta con recursos ilimitados, se espera que los recursos puedan ser proporcionados automáticamente cuando la aplicación lo requiera. Customization: los recursos proporcionados por la nube deben poder ser personalizados a cierto nivel.

En la siguiente imagen se muestran los elementos que contribuyen en la creación del cómputo en la nube [37]:

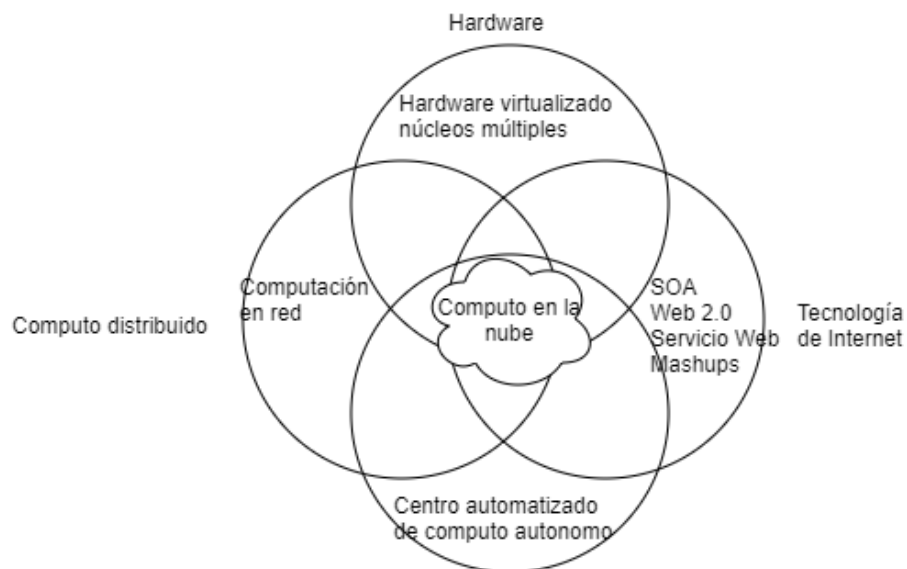


Figura 2.9 Cómputo en la nube

En la actualidad la minería de datos se utiliza para resolver varios problemas como por ejemplo la clasificación de correos en spam y no spam, las sugerencias de amigos en facebook, las búsquedas en google, etc. La minería de datos se encarga de hacer que las máquinas aprendan sin ser programadas, mediante el uso de redes neuronales imitando cómo funciona el cerebro humano [38].

Para esto existen diferentes tipos de aprendizaje [38]:

- Supervisado al cual le damos un conjunto de datos “Respuestas correctas” y en base a esos se obtiene otra respuesta correcta, estos a su vez se clasifican:
  - Problemas de regresión, se tiene de salida N datos.
  - Problemas de clasificación, se tiene una salida binaria: bueno y malo, si y no, etc.
- No supervisado, este tipo de aprendizaje se basa en que no se sabe porque característica se van a agrupar los datos.

Una red neuronal multicapa consiste de un gran número de neuronas unidas por sus conexiones, los objetos en una red neuronal generalmente son de tres tipos[39]:

- Neuronas de entrada: reciben información para procesarla.
- Neuronas de salida: donde se encuentran los resultados del procesamiento.
- Neuronas intermedias u ocultas.

Redes neuronales Feed-forward, este tipo de redes trasladan la señal de entrada a salida con un solo camino, primero la red es entrenada con un conjunto de pares de datos para determinar el mapeo de entrada y salida, los pesos de las conexiones entre las neuronas son corregidos y la red es utilizada para determinar la clasificación de un nuevo conjunto de datos [39].

Los sistemas de lógica difusa, son únicos debido a que pueden manejar datos numéricos y el conocimiento lingüístico, es un mapeo no lineal de un vector de entrada de datos a una salida escalar, la teoría y lógica difusa establece la

especificación del mapeo no lineal. La riqueza de este tipo de sistemas es que hay un gran número de posibilidades que nos llevan a varios mapeos, el uso de estos sistemas requiere de un amplio entendimiento de la lógica difusa y de los elementos del sistema. [40]

Los algoritmos genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización, se basan en los procesos genéticos de los organismos biológicos, a través de varias generaciones de acuerdo con los principios de selección natural y supervivencia de Charles Darwin. Al imitar este proceso los algoritmos genéticos son capaces de evolucionar soluciones de problemas del mundo real. [41]

# Materiales y Métodos

---

Para la creación de una ciudad inteligente y un turismo inteligente nos enfocamos en la creación de aplicaciones para la obtención y análisis de la información del turismo, estas aplicaciones conforman un sistema de cómputo en la nube.

## 3.1. Materiales

Para poder llevar a cabo la creación del sistema de cómputo en la nube necesitamos una serie de paquetes de software y los elementos de hardware necesarios con cierto nivel de compatibilidad para que el desarrollo de los proyectos de software concluya satisfactoriamente.

A continuación se describen los elementos de software y hardware necesarios.

### 3.1.1. Software

Los elementos de software que se utilizaron para el desarrollo e implantación del proyecto son herramientas de software libre porque son más económicos y para el proyecto desarrollado no tuvo costo, también le dio al proyecto la libertad de distribución necesaria sin necesidad de pagar licencias de uso para su implementación, a continuación se muestra un listado de los paquetes de software utilizados:

- Android Studio es un entorno de desarrollo integrado para el desarrollo de aplicaciones Android basado en IntelliJ IDEA, (se puede descargar en <https://developer.android.com/studio>).
- Java es un lenguaje de programación orientado a objetos, la descargar el JDK o Java Development Kit se obtienen las herramientas para construir o desarrollar aplicaciones con el lenguaje de programación Java, (se puede descargar en <https://www.java.com/es/download/>).
- Apache Maven es una herramienta que permite construir y administrar cualquier proyecto basado en Java, (Se puede descargar en <https://maven.apache.org/download.cgi>).
- MySQLServer es una herramienta que nos permite gestionar y administrar bases de datos relacional de código abierto basado en un lenguaje de consultas SQL, (Se puede descargar en <https://dev.mysql.com/downloads/mysql/>).
- MySQLWorkbench es una herramienta administrativa visual para el modelado, desarrollo y comprensión del servidor MySQLServer, (se puede descargar en <https://dev.mysql.com/downloads/workbench/>).
- Spring Tools 4 es un entorno de desarrollo integrado de software libre para el desarrollo de aplicaciones, (se puede descargar en <https://spring.io/tools>).
- Apache Tomcat es una herramienta de software libre que permite implementar Java Servlet, Java Server Pages, Java Expression Language y Java WebSocket technologies, (se puede descargar en <https://tomcat.apache.org/download-90.cgi>).

### 3.1.2. Hardware

Los elementos de hardware que se utilizaron para el desarrollo e implementación del proyecto son una computadora portátil marca HP con 8gb de memoria RAM, un procesador Intel(R) Core( tm) I7-7500u cpo 2.70 GHz 2.90GHz, con un sistema operativo Windows 10 a 16 bits y un dispositivo móvil Android.

## 3.2. Metodología

Cuando tenemos seleccionados los materiales necesarios para la creación del sistema de cómputo en la nube se procede a la instalación y configuración de los paquetes de software.

La siguiente figura muestra el proceso de creación del sistema de cómputo en la nube:



Figura 3.1 Metodología

### 3.2.1. Instalación y configuración de paquetes de software

En esta sección se muestra el proceso de instalación y configuración de los paquetes de software necesarios para el desarrollo del proyecto.

### 3.2.1.1. MySQLServer

Primeramente se procede a la descarga del paquete desde la página de la comunidad de MySQL y seleccionamos el paquete de MySQL Community Server en la sección de descargas como se muestra en la siguiente figura.

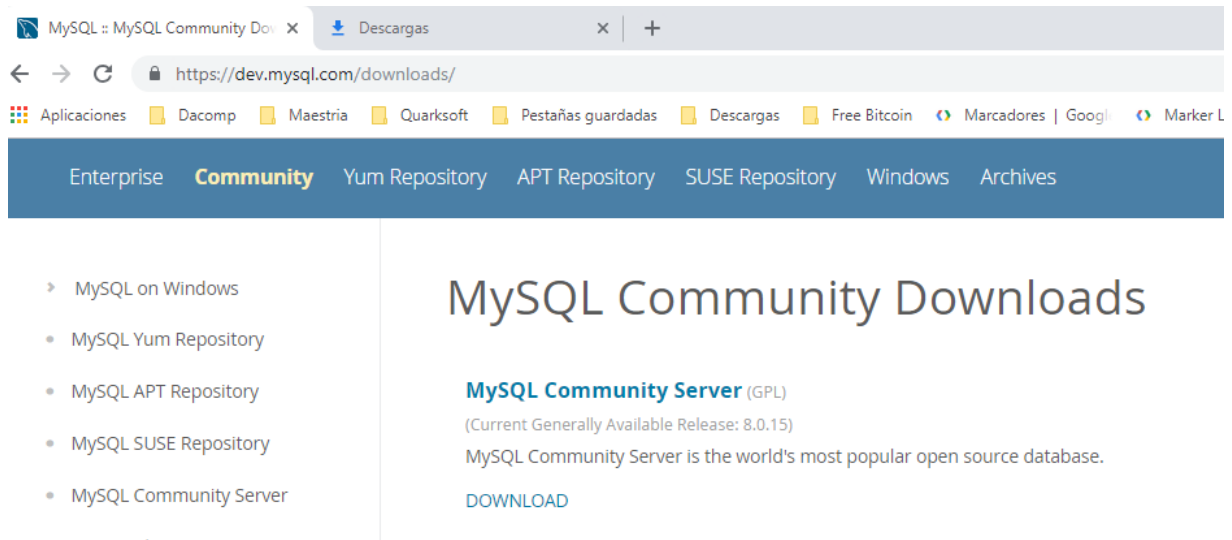


Figura 3.2 MySQL Community Download

Al seleccionar el paquete MySQL Community Downloads da la opción de seleccionar el sistema operativo y en caso de ser Windows permite seleccionar el instalador de 32 o 64 bit como se muestra en la siguiente figura.

**MySQL Community Server 8.0.15**

Select Operating System:  
Microsoft Windows

Looking for previous GA versions?

**Recommended Download:**

**MySQL Installer for Windows**  
All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

**Windows (x86, 32 & 64-bit), MySQL Installer MSI** [Go to Download Page >](#)

**Other Downloads:**

Windows (x86, 64-bit), ZIP Archive (mysql-8.0.15-winx64.zip)	8.0.15	184.1M	<a href="#">Download</a>
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite	8.0.15	270.3M	<a href="#">Download</a>

Figura 3.3 MySQL Versión

Cuando se selecciona el paquete a instalar pide una confirmación de la descarga como se muestra en la siguiente figura.

Este tipo de archivo puede dañar tu ordenador. ¿Quieres descargar mysql-installer-co...msi de todos modos?

[Descargar](#) [Rechazar](#)

Figura 3.4 Confirmación descarga MySQL

Al terminar la descarga damos click en el archivo descargado y procedemos a la instalación de software.

**mysql-installer-community-8.0.15.0.msi**

<https://cdn.mysql.com//Downloads/MySQLInstaller/mysql-installer-community-8.0.15.0...>

[Mostrar en carpeta](#)

Figura 3.5 Archivo Descargado MySQL



El instalador muestra una pantalla como se muestra en la siguiente figura, en la cual seleccionamos la opción de Add.

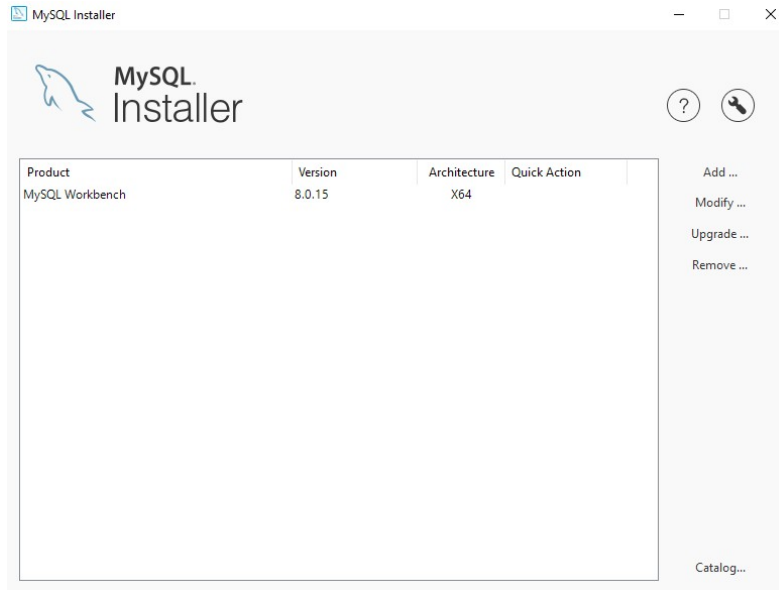


Figura 3.6 Instalador MySQL

Al seleccionar la opción de Add nos pide aceptar la licencia de uso del paquete de software como se muestra en la siguiente figura.

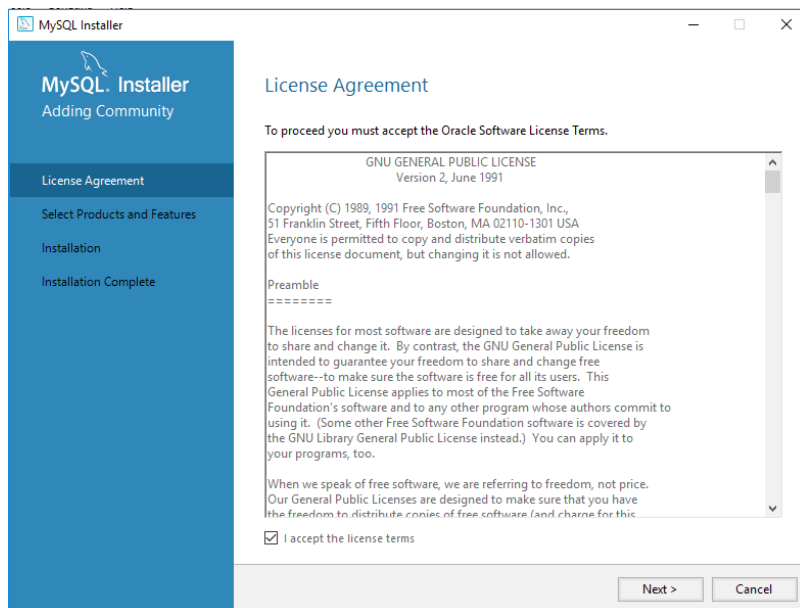


Figura 3.7 Licencia MySQL

Después de aceptar la licencia se selecciona el paquete de software a instalar, en el caso del proyecto es MySQL Server 8.0.15 – X64 como se muestra en la siguiente figura.

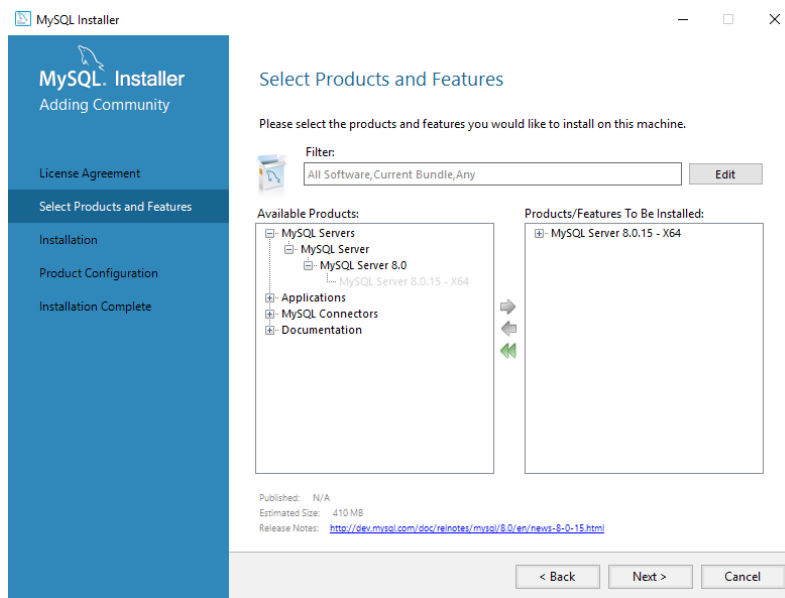


Figura 3.8 Productos MySQL

Ahora se selecciona el directorio de instalación del programa como se muestra en la siguiente figura.

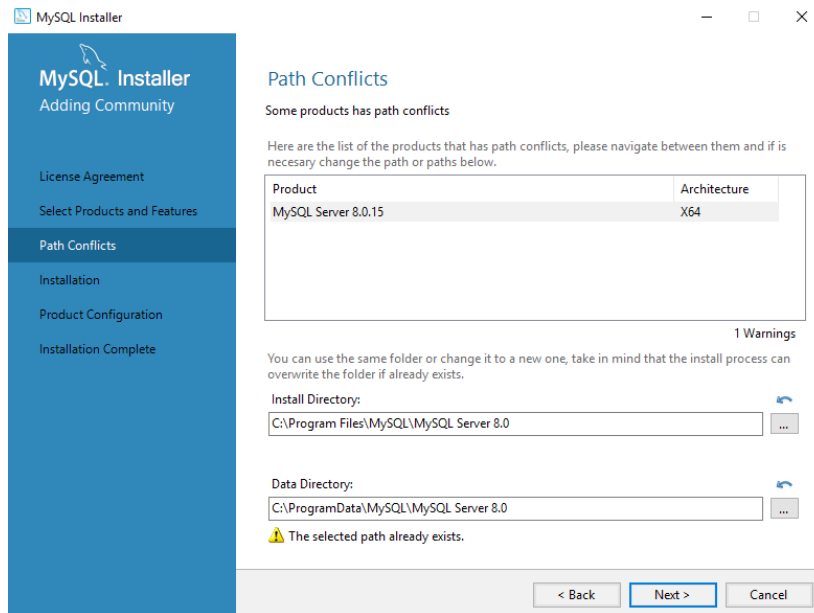


Figura 3.9 Directorio de instalación MySQL

Ya que se seleccionó el paquete a instalar así como el lugar en el que se instalara se procede a ejecutar la instalación como se muestra en la siguiente figura.

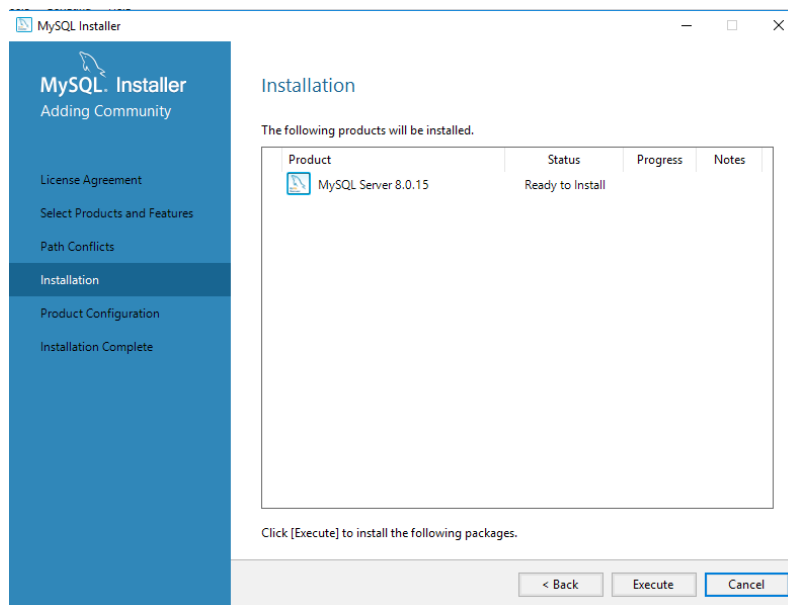


Figura 3.10 Confirmación de instalación MySQL

Al terminar la instalación el producto debe estar en estatus completo en el listado como se muestra en la siguiente figura.

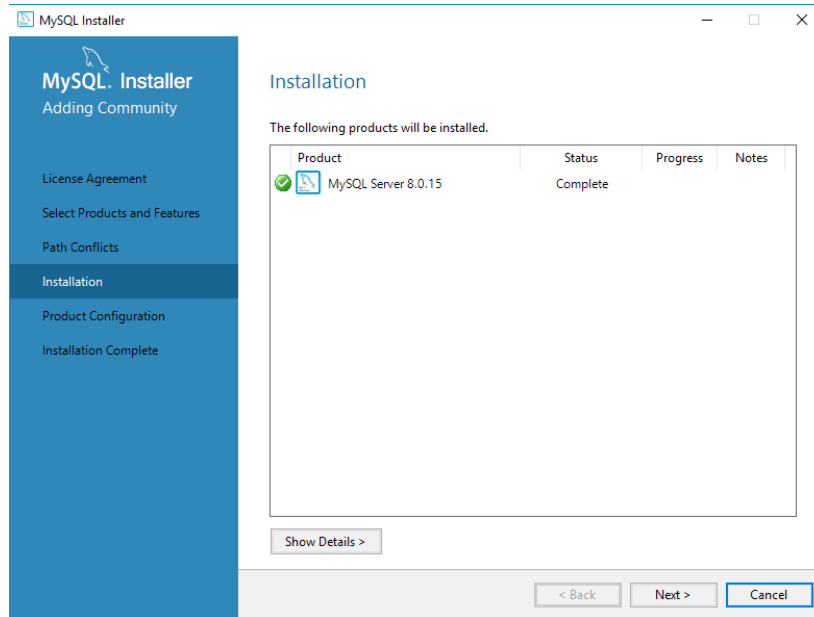


Figura 3.11 Confirmación de instalación MySQL

Al finalizar la instalación del producto inicia el proceso de configuración del programa y el estatus debe de ser listo para configurar como se muestra en la siguiente figura.

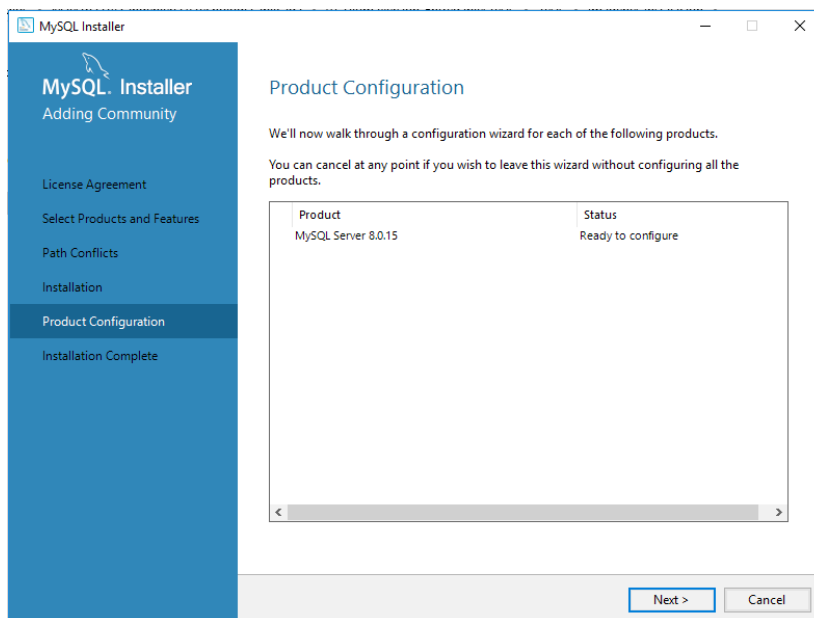


Figura 3.12 Finalización de instalación MySQL

Para el grupo de replicación se selecciona “Standalone MySQL Server” como se muestra en la siguiente figura debido a que no solo lo utilizaremos para hacer pruebas sino que también en un ambiente de producción.

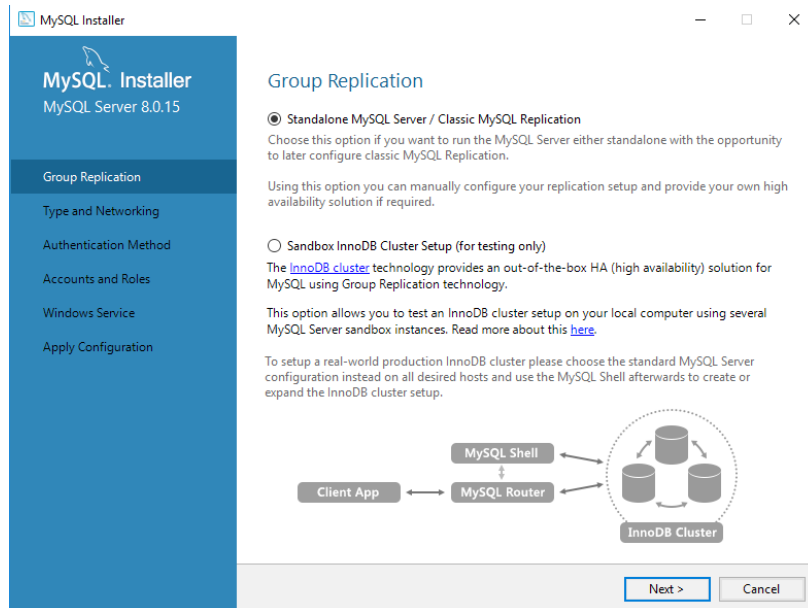


Figura 3.13 Replicación MySQL

Se selecciona el protocolo de comunicación así como el puerto por el cual tendremos acceso a las bases de datos como se muestra en la siguiente figura.

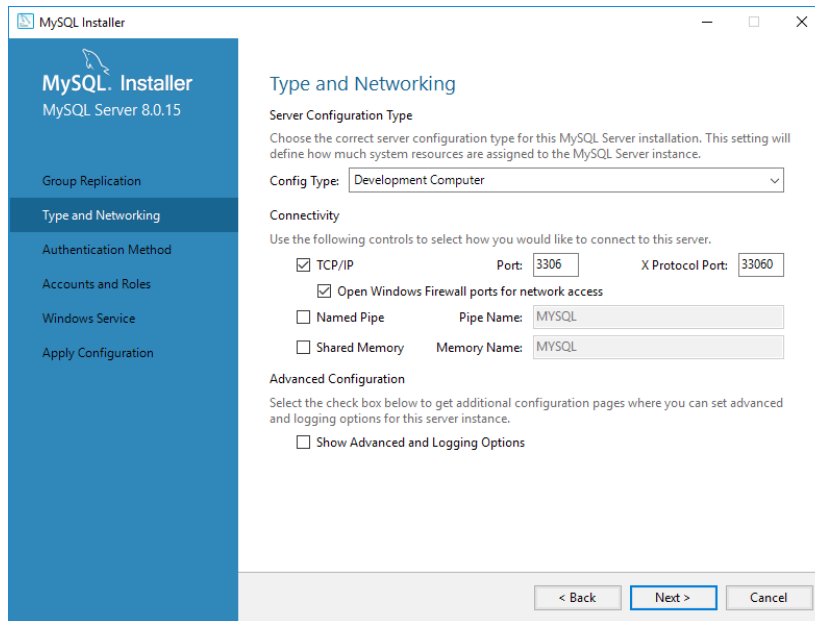


Figura 3.14 Red MySQL

Se utiliza el método de autenticación recomendado como se muestra en la siguiente figura.

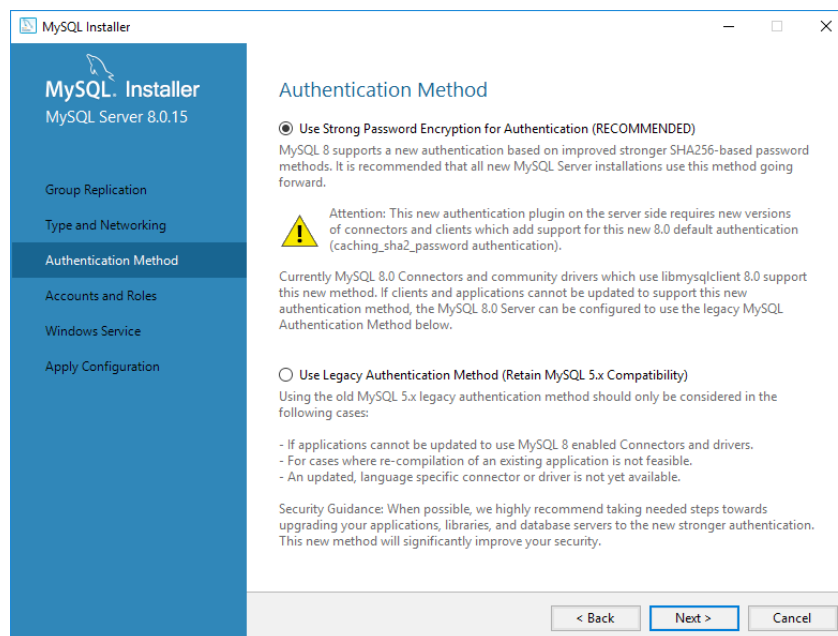


Figura 3.15 Autenticación MySQL

Se coloca la contraseña para el usuario Root y se revisa que este correcto con el botón de “Check” como se muestra en la siguiente figura.

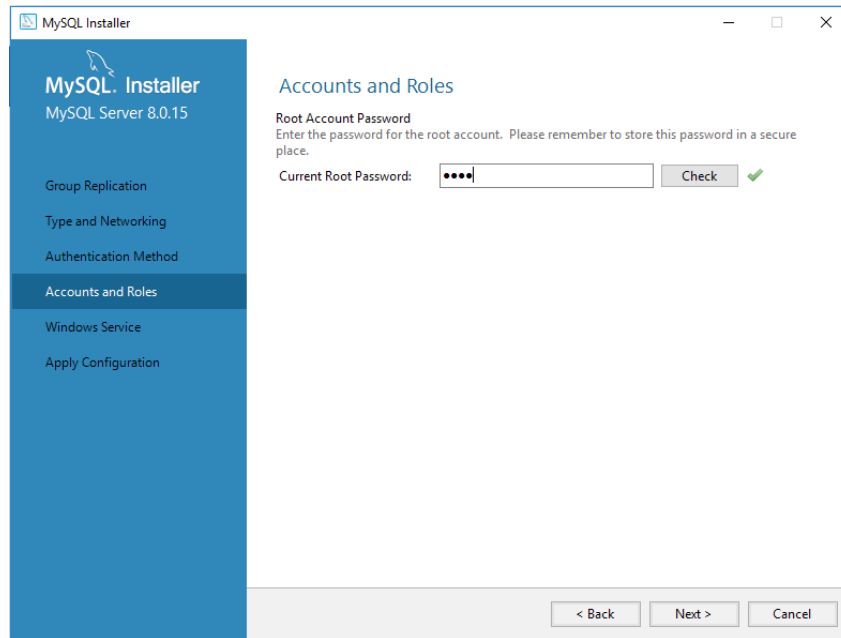


Figura 3.16 Cuentas y Roles MySQL

Se provee un nombre para el servicio de Windows y se selecciona la opción de iniciar MySQL server cuando inicie el sistema como se muestra en la siguiente figura.

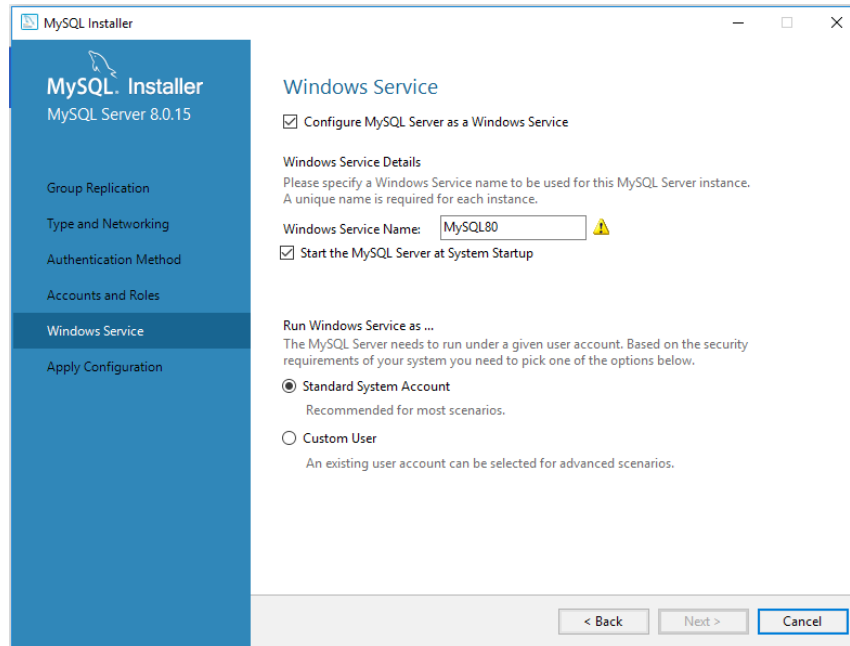


Figura 3.17 Servicio MySQL

Por último se procede a aplicar las configuraciones previamente seleccionadas como se muestra en la siguiente imagen.

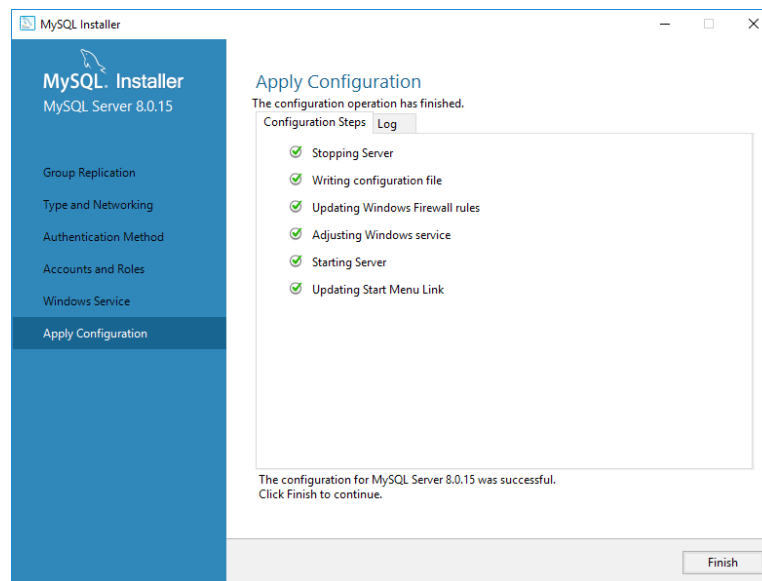


Figura 3.18 Aplicar configuración MySQL

Al final manda un mensaje de instalación completa como se muestra en la siguiente figura con esto ya se tiene instalado y configurado MySQL correctamente.



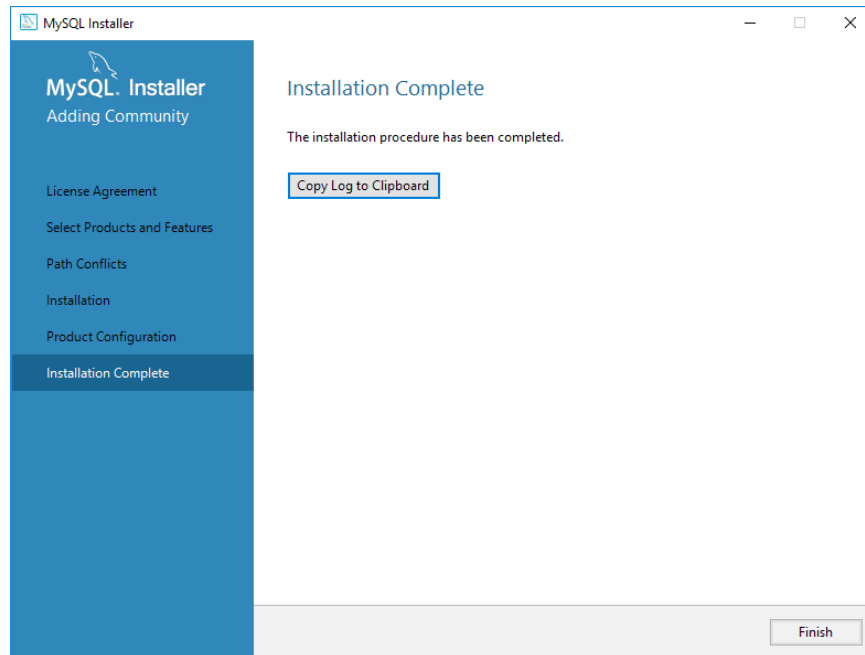


Figura 3.19 Configuración completa MySQL

- Java

Primeramente se procede a la descarga del paquete desde la página de Oracle Java en la sección de descargas y seleccionamos la opción de descarga gratuita de Java como se muestra en la siguiente figura.



Figura 3.20 Descarga Java

Al seleccionar la opción de descarga gratuita nos pide aceptar la licencia de uso del paquete de software para iniciar la descarga como se muestra en la siguiente figura.

## Descargar Java para Windows

**Recomendado Version 8 Update 201 (Tamaño de archivo: 1.87 MB)**

Fecha de lanzamiento: 15 de enero de 2019

**Aceptar e iniciar descarga gratuita**

Al descargar Java, confirma que ha leído y aceptado los términos del acuerdo de licencia de usuario final

Figura 3.21 Términos y condiciones Java

Cuando se acepta la licencia de uso del paquete a instalar pide una confirmación de la descarga como se muestra en la siguiente figura.

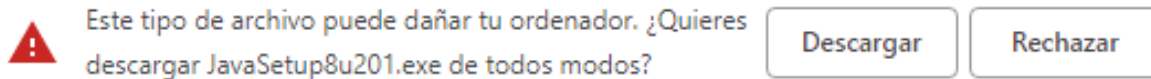


Figura 3.22 Confirmar descarga Java

Al terminar la descarga damos click en el archivo descargado y procedemos a la instalación de software.

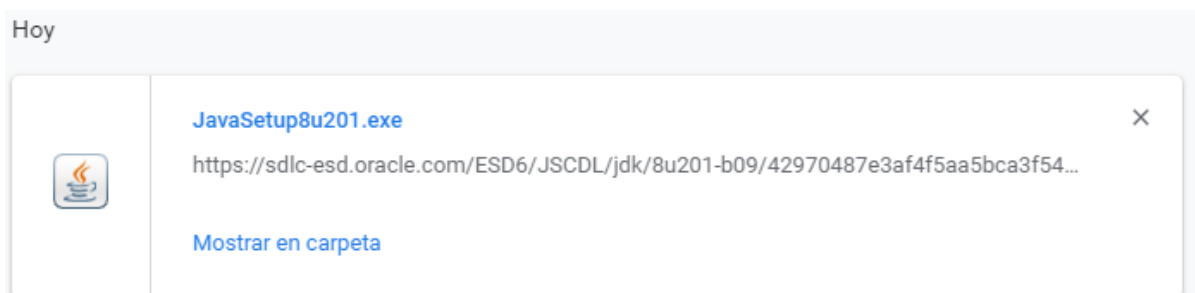


Figura 3.23 Directorio de descarga Java

El instalador muestra una pantalla donde nos permite cambiar la carpeta destino del paquete y la opción de instalar como se muestra en la siguiente figura.

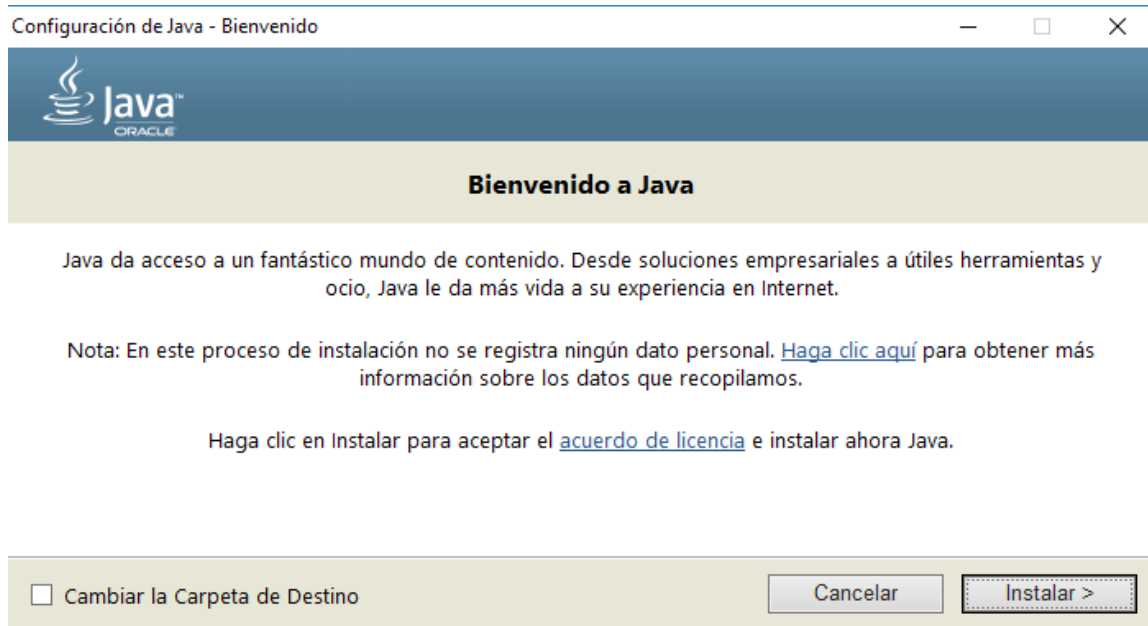


Figura 3.24 Carpeta de instalación Java

Se muestra una barra de progreso de instalación del paquete como se muestra en la siguiente figura.



Figura 3.25 Instalación Java

Al final manda un mensaje de instalación completa como se muestra en la siguiente figura, con esto ya se tiene instalado Java correctamente.



Figura 3.26 Instalación completa Java

- Apache tomcat

Primeramente se procede a la descarga del paquete desde la página de Apache Tomcat en el menú de lado izquierdo en la sección de descarga como se muestra en la siguiente figura.

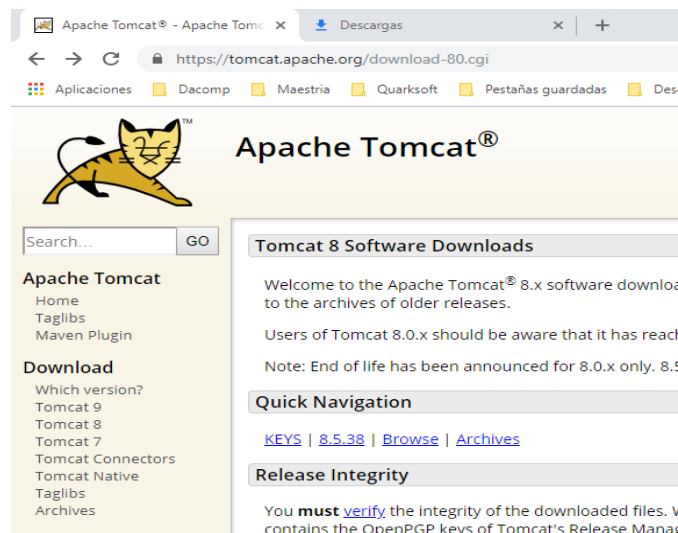


Figura 3.27 Descarga Tomcat

Se selecciona el paquete del listado que se muestra en la siguiente figura para descargar que para el proyecto es la versión de Windows de 64 bit.

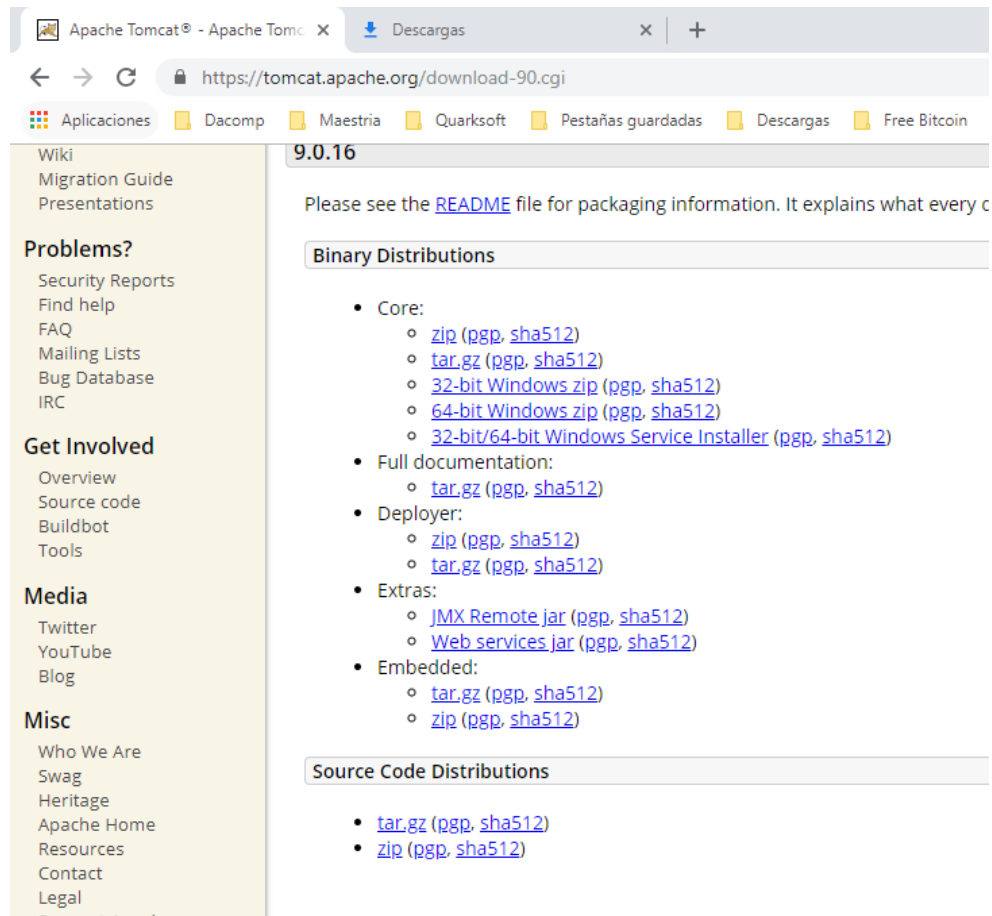


Figura 3.28 Distribución Tomcat

Cuando se selecciona el paquete a instalar pide una confirmación de la descarga como se muestra en la siguiente figura.

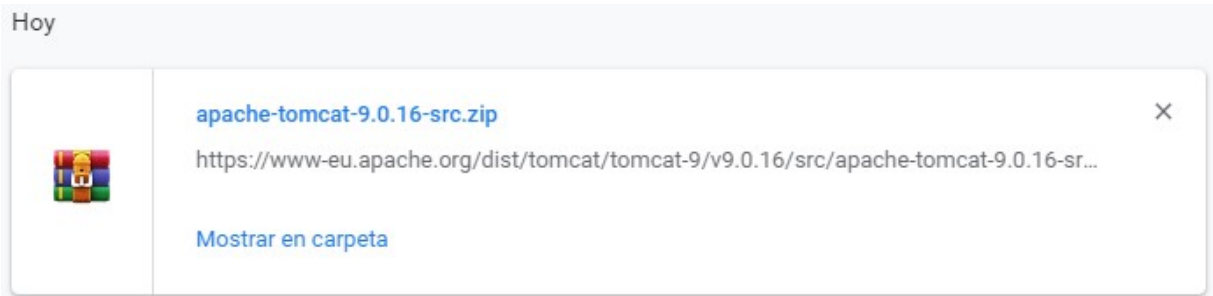


Figura 3.29 Directorio descarga Tomcat

Al terminar la descarga damos click en el archivo descargado y se descomprime en la carpeta destino como se muestra en la siguiente imagen.

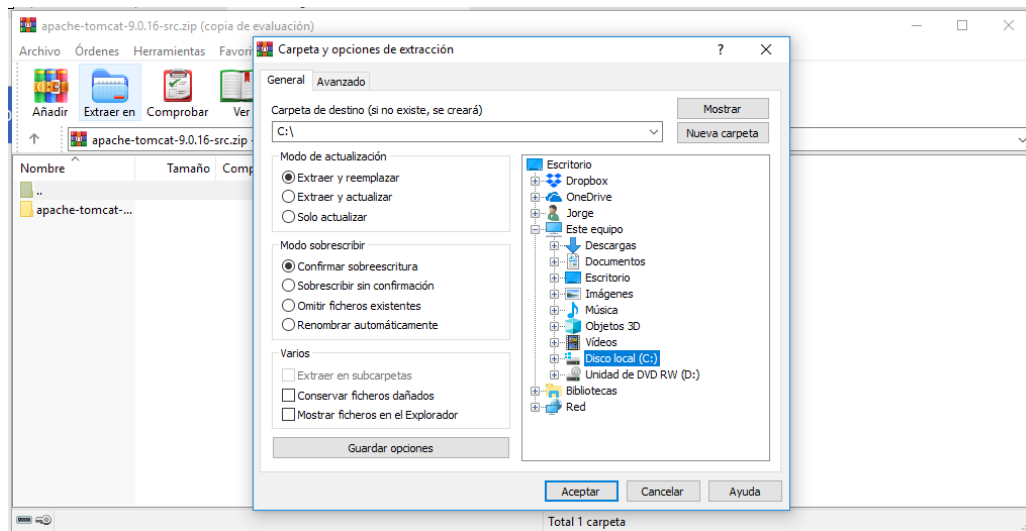


Figura 3.30 Descomprimir Tomcat

Al terminar la descompresión del archivo se mostrara la carpeta de Apache Tomcat como se muestra en la siguiente figura.

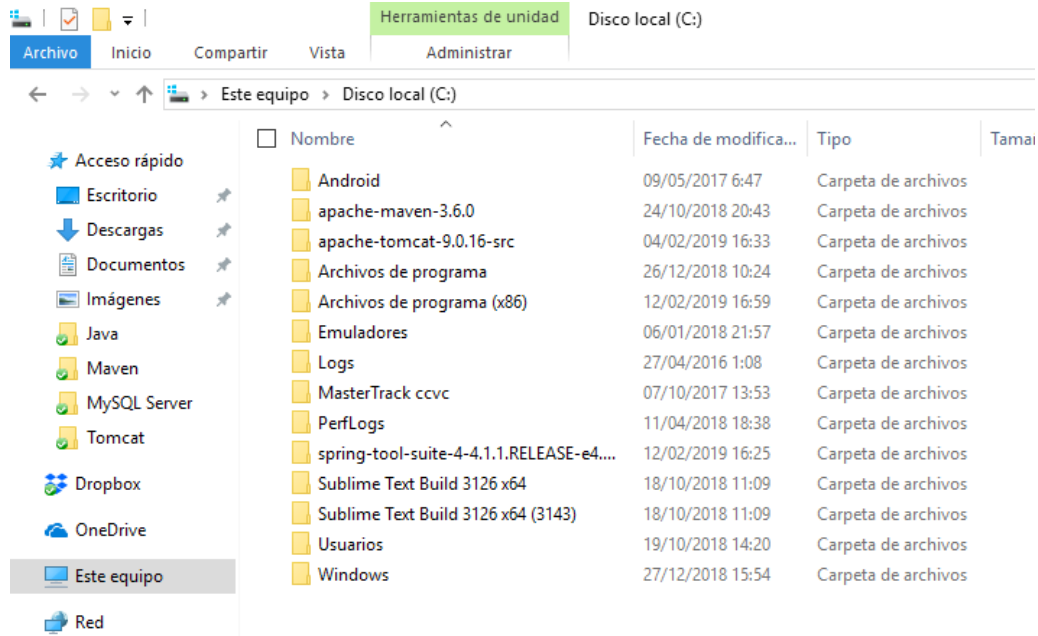


Figura 3.31 Directorio de archivos Tomcat

- Maven

Primeramente se procede a la descarga del paquete desde la página de Apache Maven en el menú de lado izquierdo en la sección de descargas como se muestra en la siguiente figura.

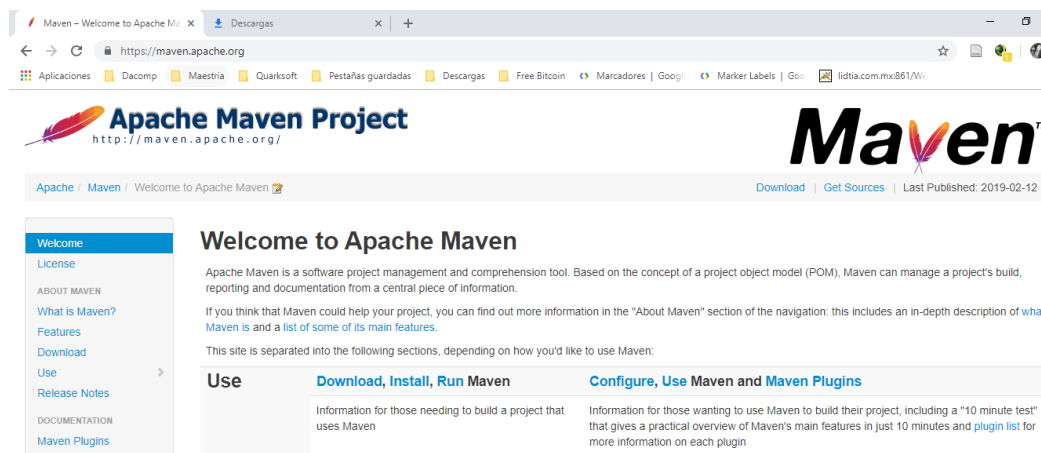


Figura 3.32 Descarga Maven

Se selecciona el paquete del listado que se muestra en la siguiente figura para descargar la versión requerida.

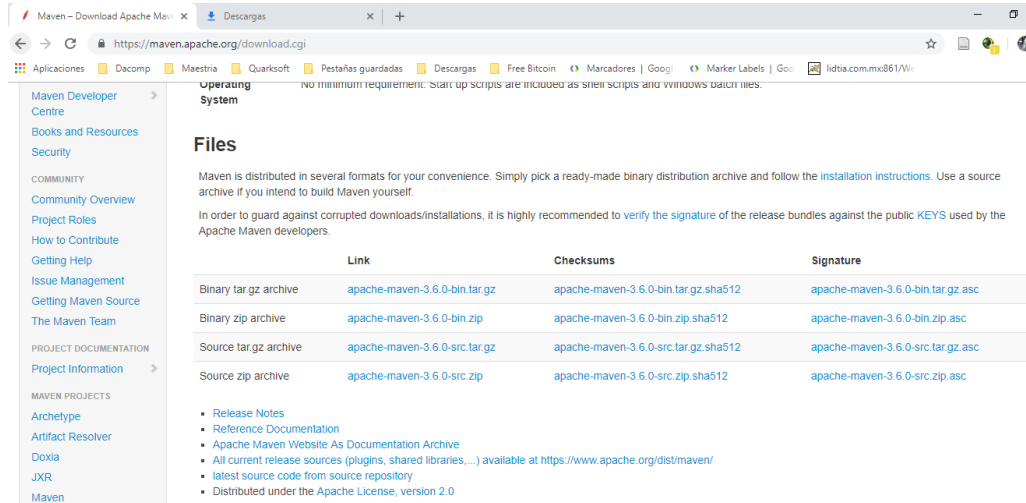


Figura 3.33 Distribución Maven

Cuando se selecciona el paquete a instalar pide una confirmación de la descarga como se muestra en la siguiente figura.

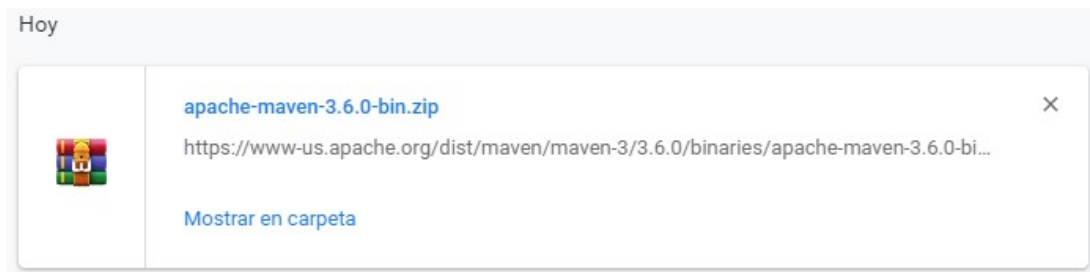


Figura 3.34 Directorio descarga Maven

Al terminar la descarga damos click en el archivo descargado y se descomprime en la carpeta destino como se muestra en la siguiente imagen.



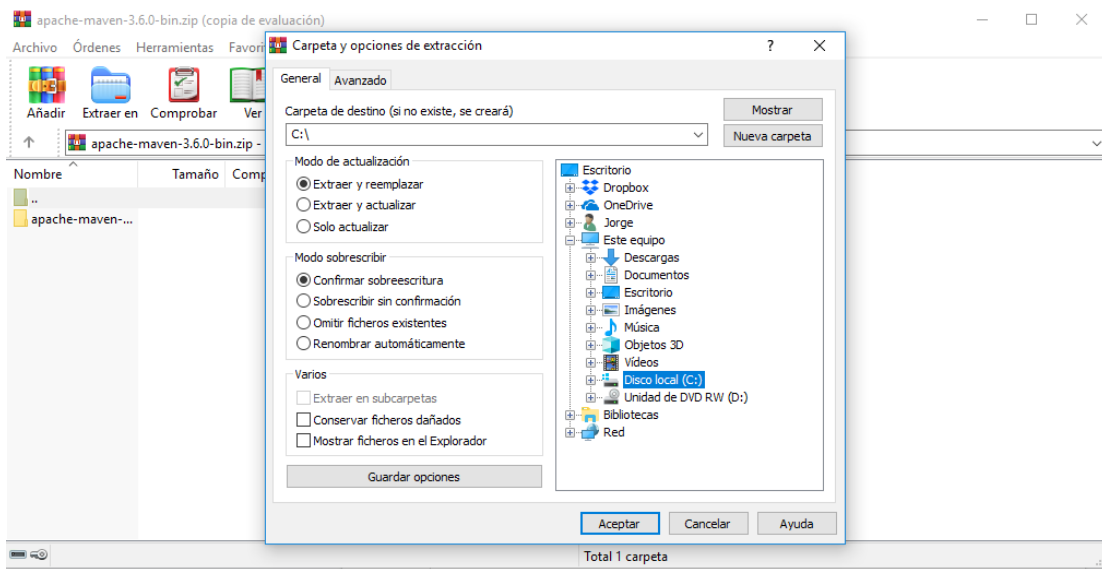


Figura 3.35 Descomprimir Maven

Al terminar la descompresión del archivo se mostrara la carpeta de Apache Maven como se muestra en la siguiente figura.

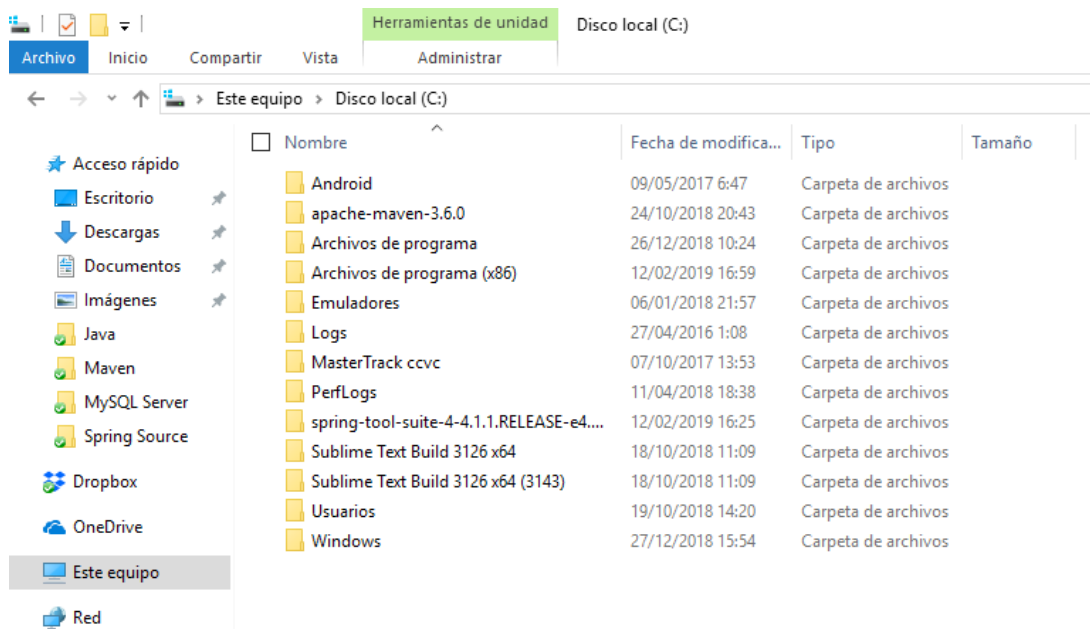
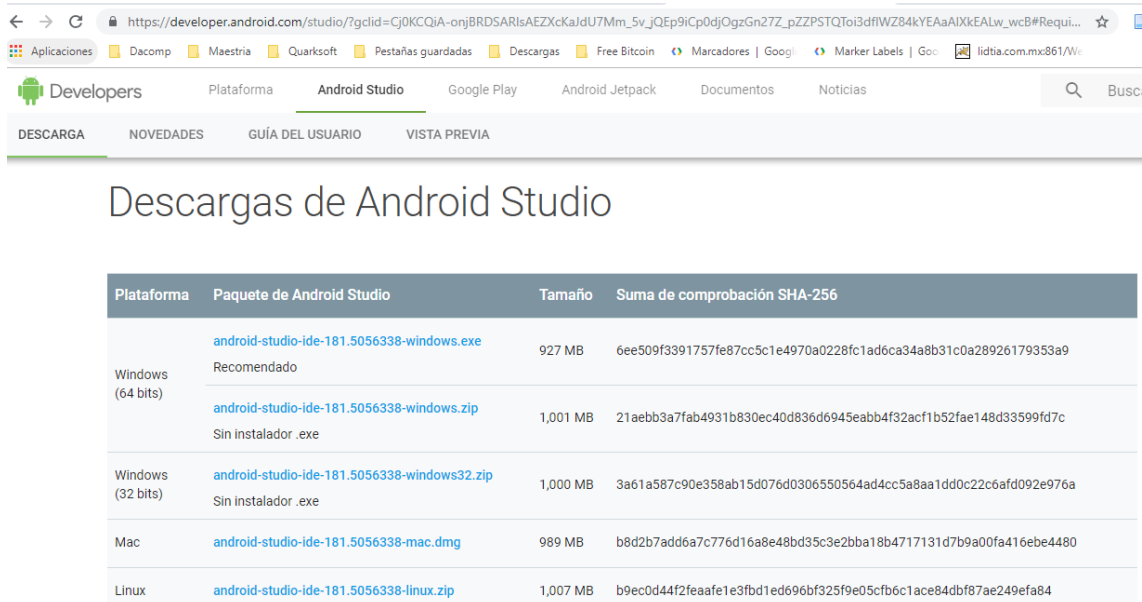


Figura 3.36 Directorio de archivos Maven

- Android Studio

Primeramente se procede a la descarga del paquete desde la página de Android Studio en la sección de descarga y seleccionamos el paquete que en nuestro caso es para Windows de 64 bit como se muestra en la siguiente figura.



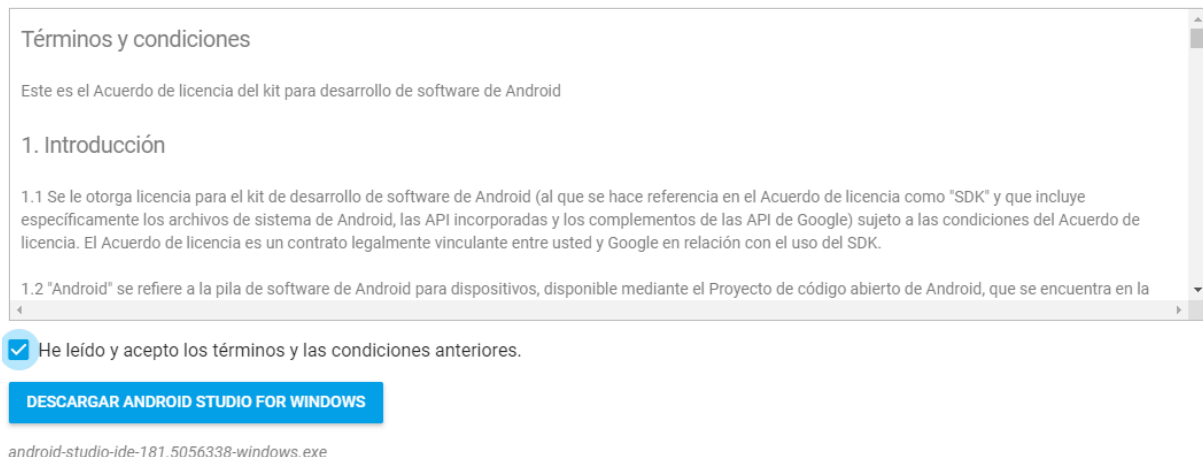
Plataforma	Paquete de Android Studio	Tamaño	Suma de comprobación SHA-256
Windows (64 bits)	<a href="#">android-studio-ide-181.5056338-windows.exe</a> Recomendado	927 MB	6ee509f3391757fe87cc5c1e4970a0228fc1ad6ca34a8b31c0a28926179353a9
	<a href="#">android-studio-ide-181.5056338-windows.zip</a> Sin instalador .exe	1,001 MB	21aebb3a7fab4931b830ec40d836d6945eabb4f32ac1b52fae148d33599fd7c
Windows (32 bits)	<a href="#">android-studio-ide-181.5056338-windows32.zip</a> Sin instalador .exe	1,000 MB	3a61a587c90e358ab15d076d0306550564ad4cc5a8aa1dd0c22c6afd092e976a
Mac	<a href="#">android-studio-ide-181.5056338-mac.dmg</a>	989 MB	b8d2b7add6a7c776d16a8e48bd35c3e2bba18b4717131d7b9a00fa416ebe4480
Linux	<a href="#">android-studio-ide-181.5056338-linux.zip</a>	1,007 MB	b9ec0d44f2feaafe1e3fbd1ed696bf325f9e05cfb6c1ace84dbf87ae249efa84

Figura 3.37 Descarga Android Studio

Al seleccionar el paquete nos pide aceptar los términos y condiciones de uso del software como se muestra en la siguiente imagen.

## Android Studio

Antes de la descarga, debes aceptar los términos y las condiciones.



Términos y condiciones

Este es el Acuerdo de licencia del kit para desarrollo de software de Android

### 1. Introducción

1.1 Se le otorga licencia para el kit de desarrollo de software de Android (al que se hace referencia en el Acuerdo de licencia como "SDK" y que incluye específicamente los archivos de sistema de Android, las API incorporadas y los complementos de las API de Google) sujeto a las condiciones del Acuerdo de licencia. El Acuerdo de licencia es un contrato legalmente vinculante entre usted y Google en relación con el uso del SDK.

1.2 "Android" se refiere a la pila de software de Android para dispositivos, disponible mediante el Proyecto de código abierto de Android, que se encuentra en la

He leído y acepto los términos y las condiciones anteriores.

**DESCARGAR ANDROID STUDIO FOR WINDOWS**

`android-studio-ide-181.5056338-windows.exe`

Figura 3.38 Términos y condiciones Android Studio

Cuando se aceptan los términos pide una confirmación de la descarga como se muestra en la siguiente figura.

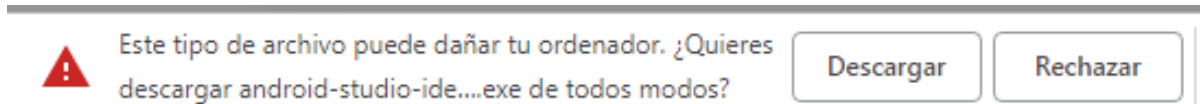


Figura 3.39 Confirmar descarga Android Studio

Al terminar la descarga damos click en el archivo descargado y procedemos a la instalación de software.

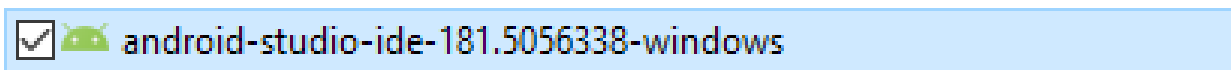


Figura 3.40 Instalación Android Studio

El instalador muestra una pantalla como se muestra en la siguiente figura, en la cual inicia el proceso de instalación de la aplicación con la opción de siguiente.



Figura 3.41 Recomendaciones Android Studio

Ahora se selecciona el directorio de instalación del programa como se muestra en la siguiente figura.

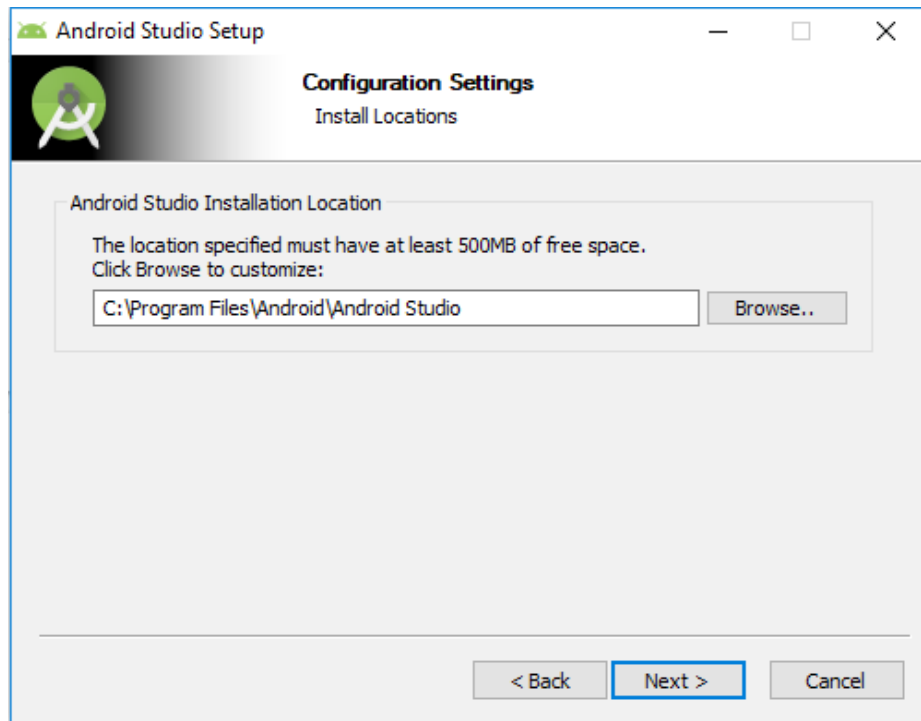


Figura 3.42 Directorio de instalación Android Studio

Al terminar la instalación la barra de estatus debe de estar en vender sin marcar ningún error como se muestra en la siguiente figura.

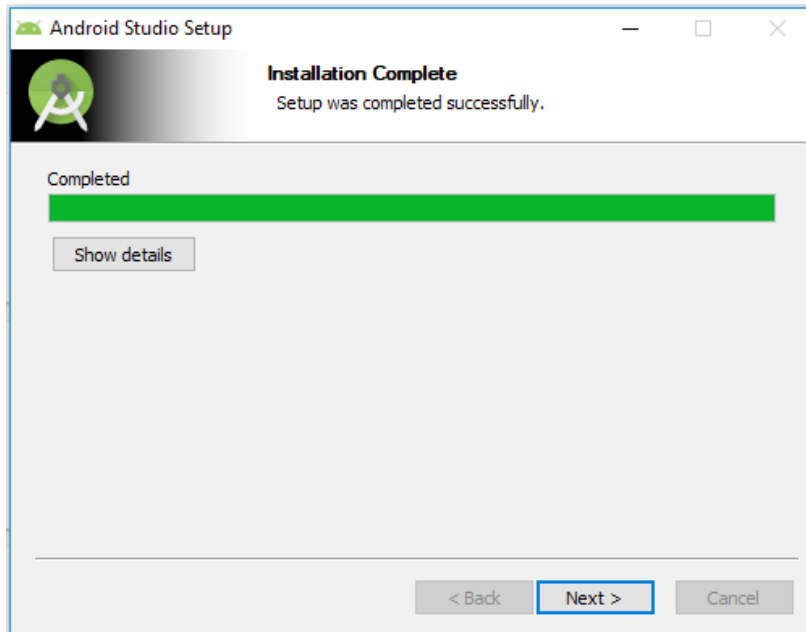


Figura 3.43 Instalación completa Android Studio

Por ultimo nos muestra la siguiente figura con la opción de iniciar el ID Android Studio para el desarrollo de aplicaciones móviles.

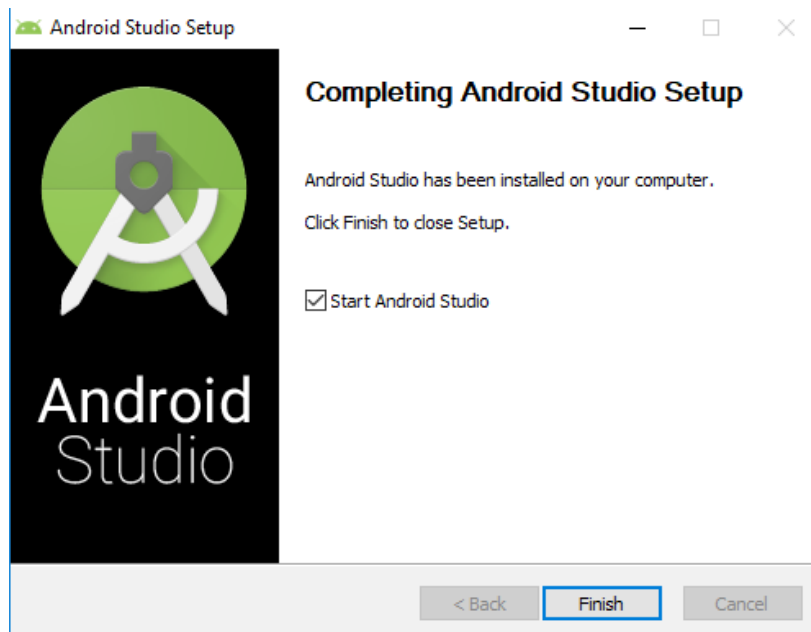


Figura 3.44 Instalación completa Android Studio

- Virtualbox

Primeramente se procede a la descarga del paquete desde la página de Virtual Box en el menú de lado izquierdo en la sección de descarga como se muestra en la siguiente figura.



Figura 3.45 Descarga VirtualBox

Se procede a la selección del archivo a descargar que en nuestro caso es para sistema operativo Windows, como se muestra en la siguiente figura.



Figura 3.46 Distribución VirtualBox

Se acepta la descarga del archivo, como se muestra en la siguiente figura.

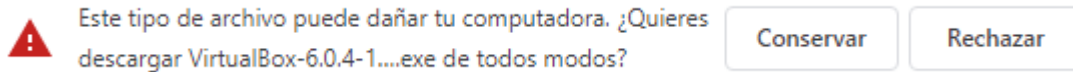


Figura 3.47 Confirmar descarga VirtualBox

Al terminar la descarga damos click en el archivo descargado y procedemos a la instalación de software.

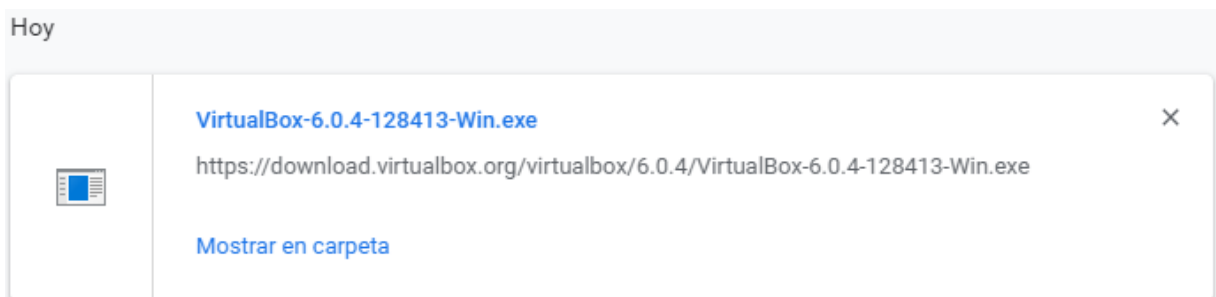


Figura 3.48 Directorio descarga VirtualBox

El instalador muestra una pantalla como se muestra en la siguiente figura, en la cual inicia el proceso de configuración de la aplicación con la opción de siguiente.



Figura 3.49 Instalación VirtualBox

Se seleccionan las funcionalidades a instalar, debido a los requerimientos de nuestro trabajo se dejan las seleccionadas por defecto.



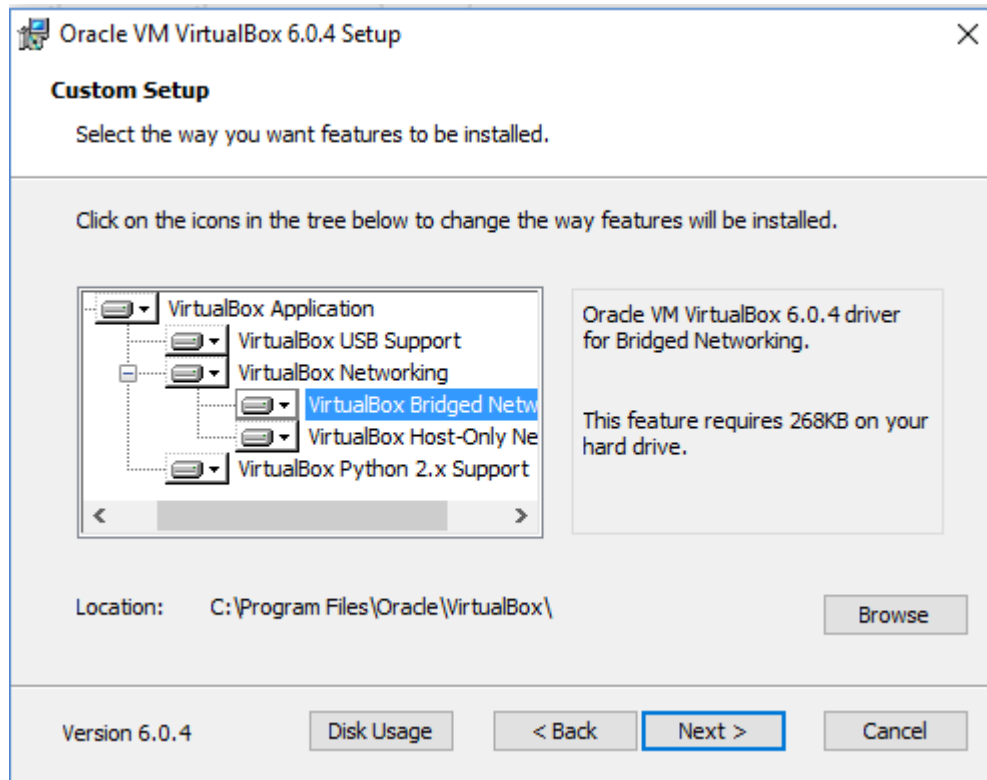


Figura 3.50 Productos VirtualBox

El instalador muestra una pantalla como se muestra en la siguiente figura, en la cual inicia el proceso de instalación de la aplicación con la opción de siguiente.



Figura 3.51 Red VirtualBox

Por ultimo nos muestra la siguiente figura con la opción de iniciar Virtual Box para la creación de maquinas virtuales.



Figura 3.52 Iniciar Configuración VirtualBox

Al iniciar la aplicación virtual box nos muestra la una pantalla donde se pueden crear, importar, exportar e iniciar maquinas virtuales como se muestra en la siguiente figura.

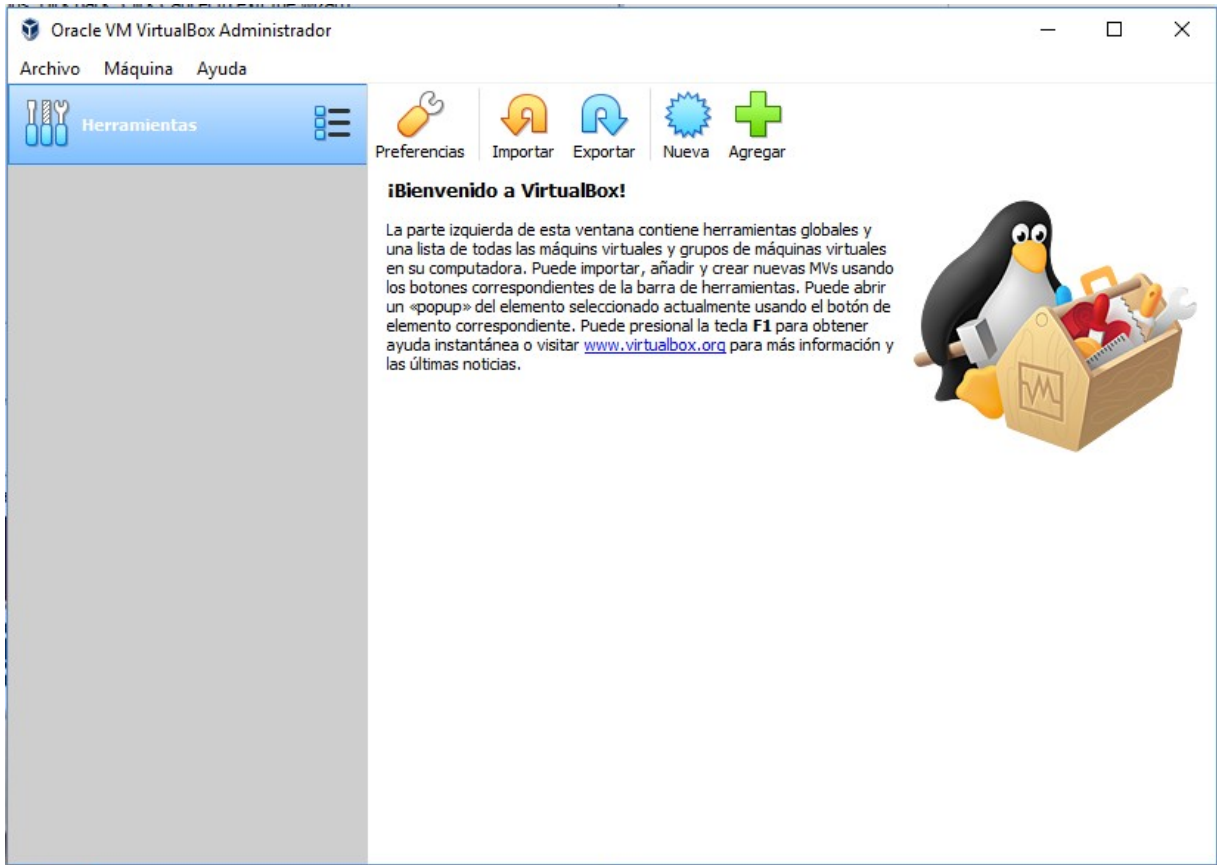


Figura 3.53 Interfaz VirtualBox

- Genymotion

Primeramente se procede a la descarga del paquete desde la página de Genymotion como se muestra en la siguiente figura.

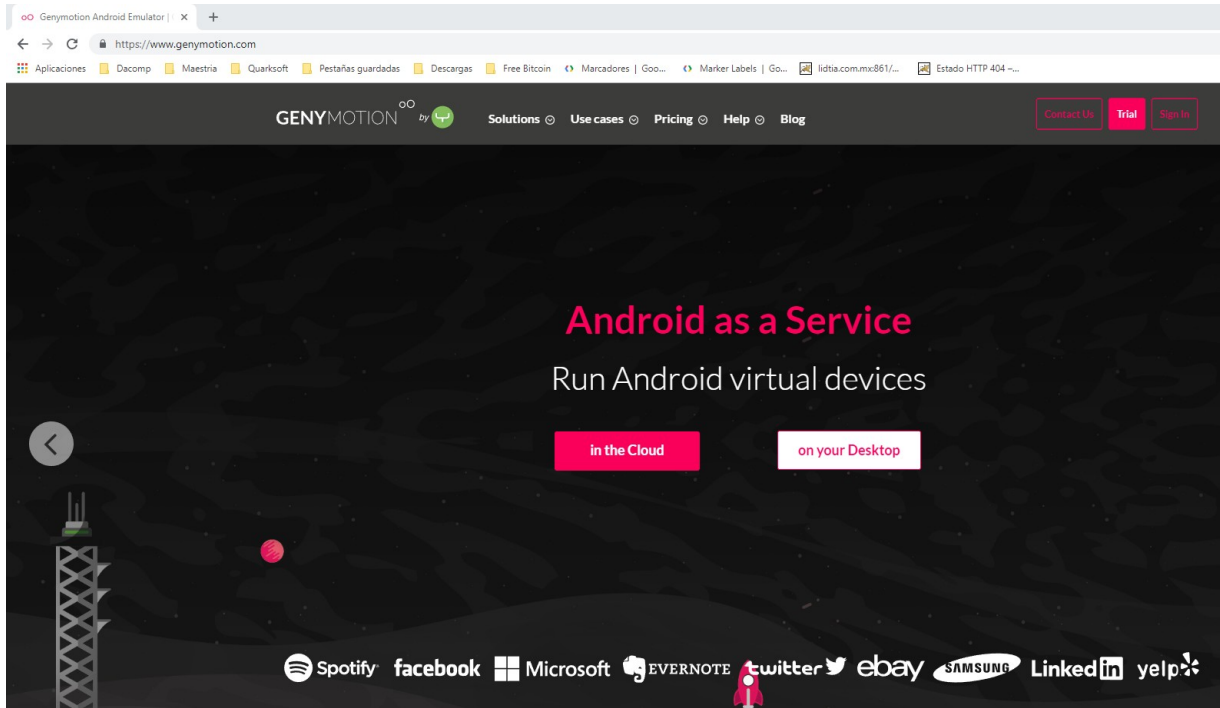


Figura 3.54 Descarga Genymotion

Se procede con la selección del sistema operativo donde se va a instalar la aplicación.

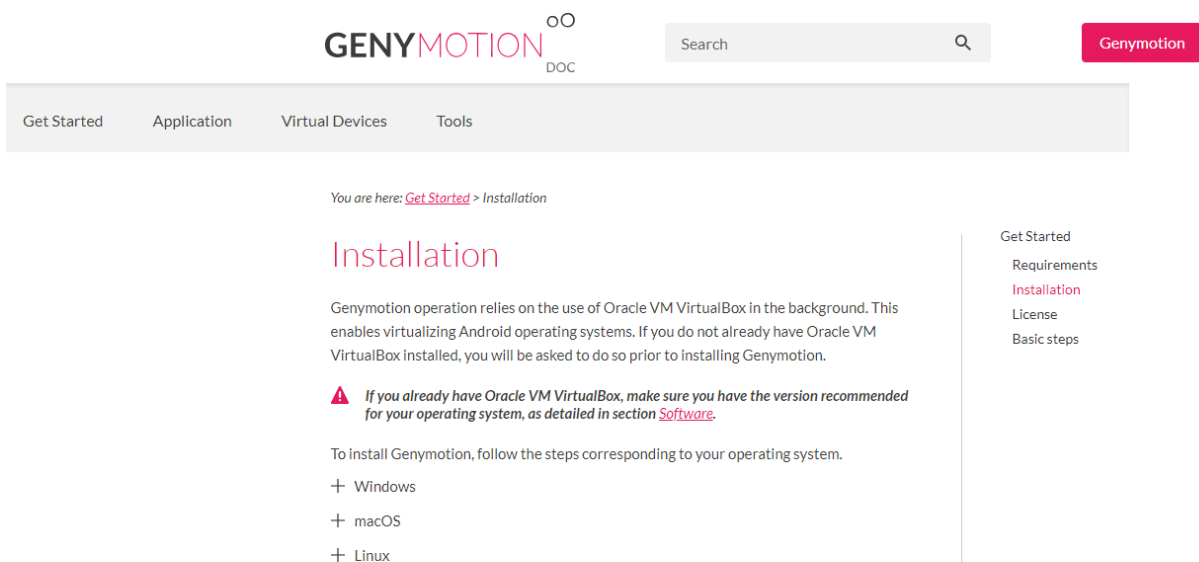
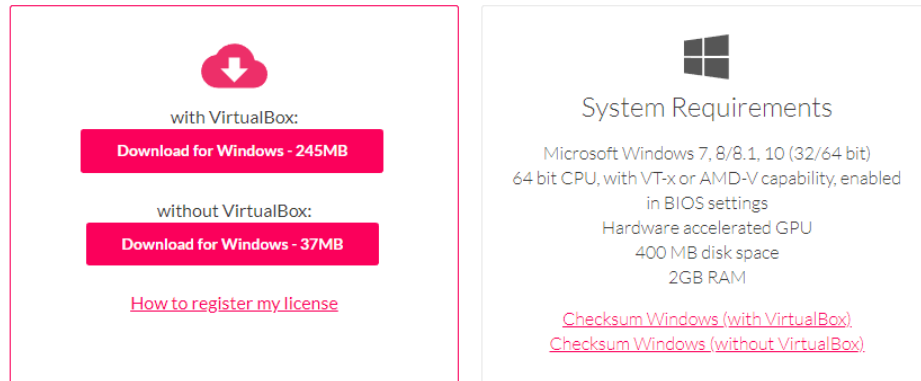


Figura 3.55 Distribución Genymotion

Se procede con la descarga de la aplicación, en la cual es recomendable descargar solo Genymotion sin la instalación de virtual box.

## Download Genymotion 3.0.1



with VirtualBox:  
**Download for Windows - 245MB**

without VirtualBox:  
**Download for Windows - 37MB**

[How to register my license](#)

### System Requirements

Microsoft Windows 7, 8/8.1, 10 (32/64 bit)  
64 bit CPU, with VT-x or AMD-V capability, enabled in BIOS settings  
Hardware accelerated GPU  
400 MB disk space  
2GB RAM

[Checksum Windows \(with VirtualBox\)](#)  
[Checksum Windows \(without VirtualBox\)](#)

Figura 3.56 Descarga sin VirtualBox Genymotion

Se confirma la descarga de la aplicación.

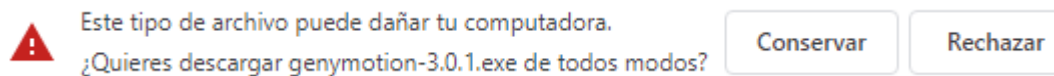


Figura 3.57 Confirmar descarga Genymotion

Al finalizar la descarga de la aplicación se procede con la instalación dando click en el archivo descargado.

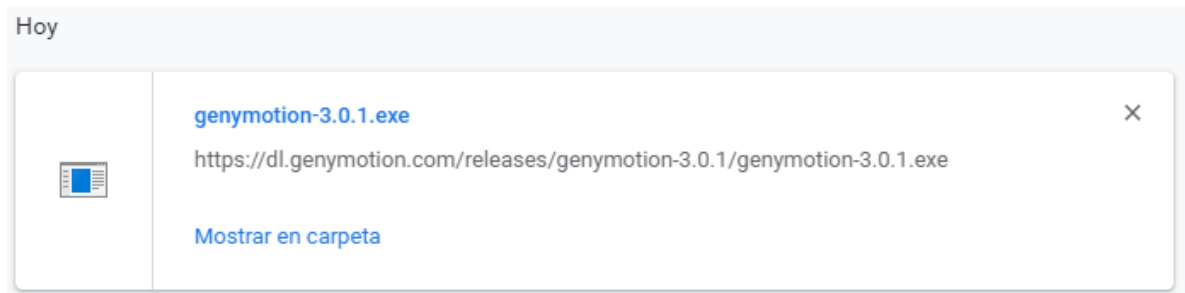


Figura 3.58 Directorio de descarga Genymotion

Se selecciona el idioma con el cual trabajamos durante la instalación de la aplicación.

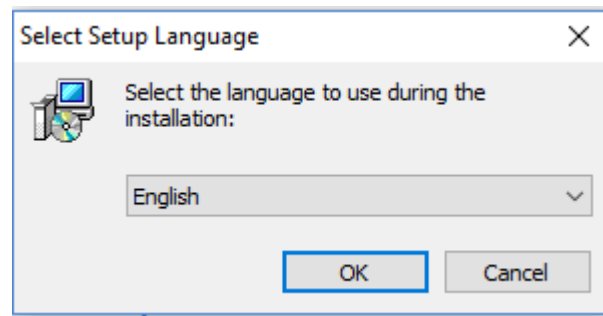


Figura 3.59 Idioma instalación Genymotion

Se selecciona la carpeta donde se guardara la aplicación como se muestra en la siguiente figura.

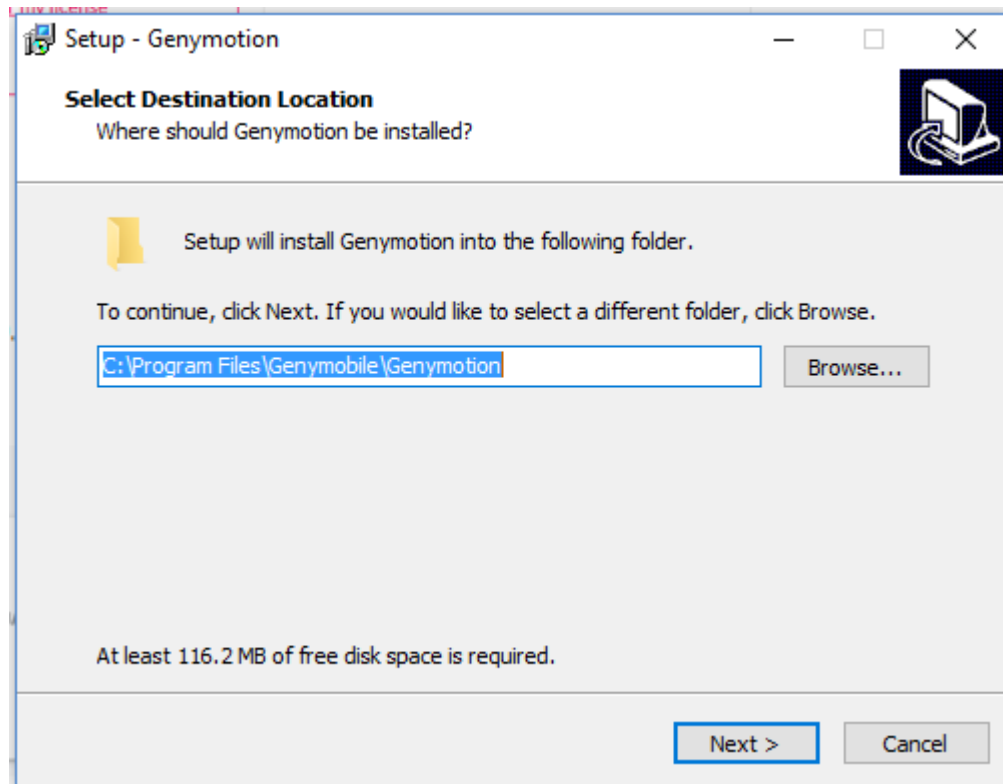


Figura 3.60 Directorio instalación Genymotion

Se introduce el nombre del atajo para nuestra aplicación, el cual podemos verlo en nuestro menú de inicio.

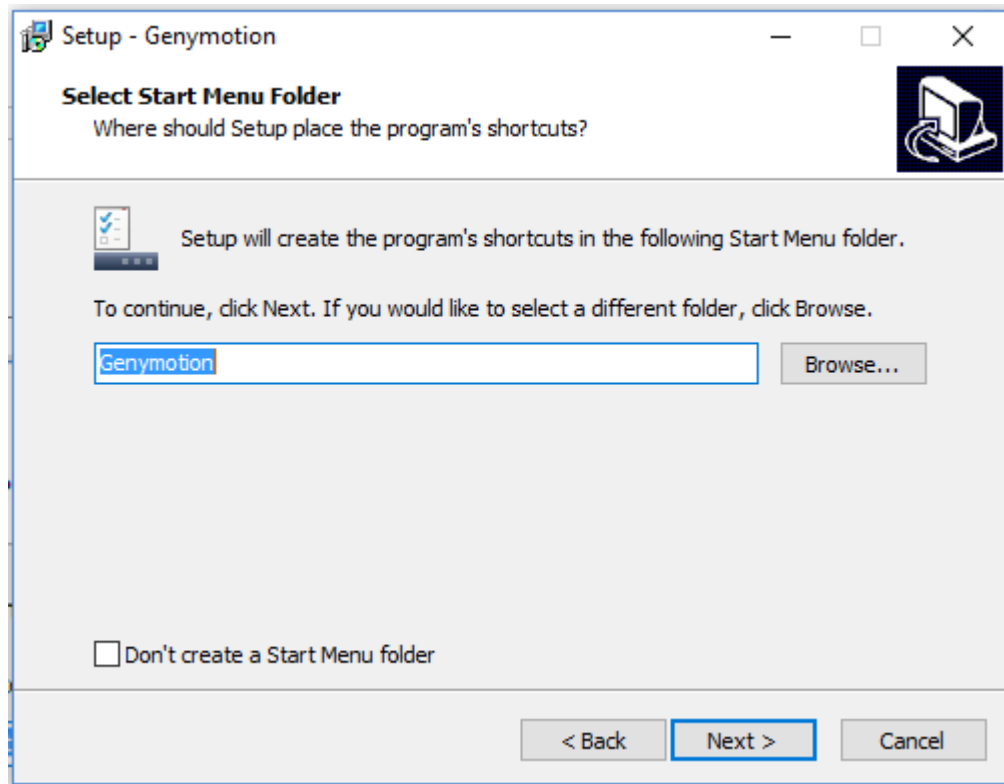


Figura 3.61 Acceso Genymotion

Nos muestra en pantalla las configuraciones seleccionadas previamente para confirmar la instalación de la aplicación como se muestra en la siguiente figura.



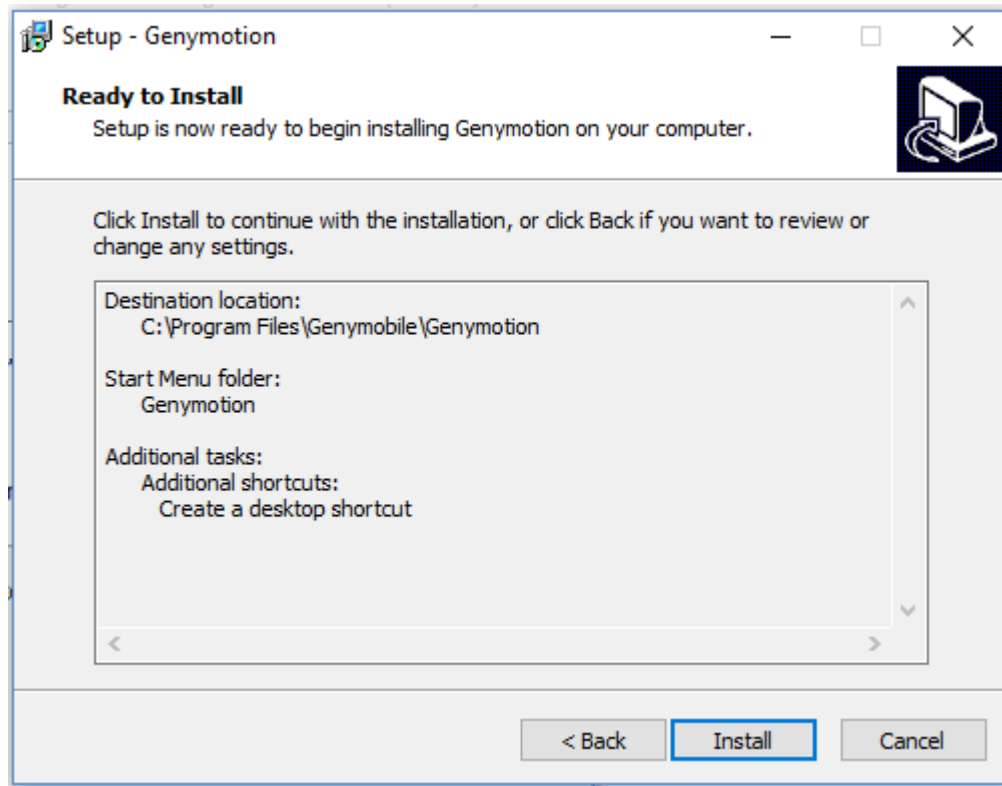


Figura 3.62 Confirmación de instalación Genymotion

AL terminar la instalación de la aplicación nos da la opción de iniciarla como se muestra en la siguiente figura.

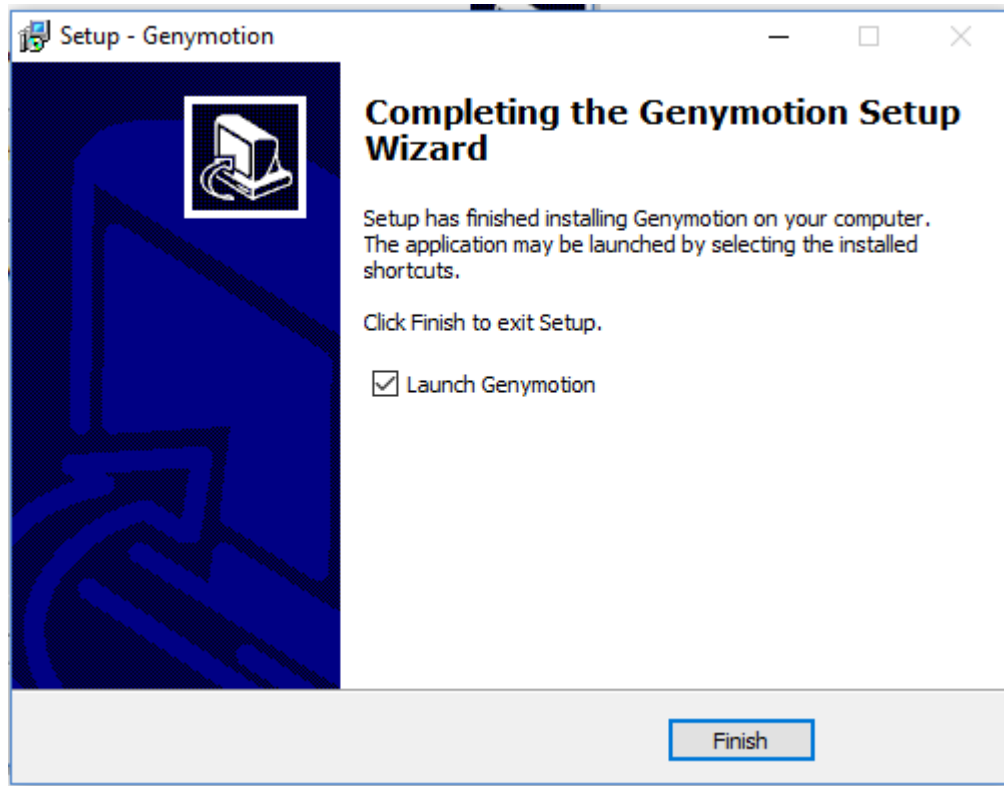


Figura 3.63 Finalizar instalación Genymotion

Al iniciar la aplicación por única vez nos pide la selección del tipo de licencia como se muestra en la siguiente imagen.

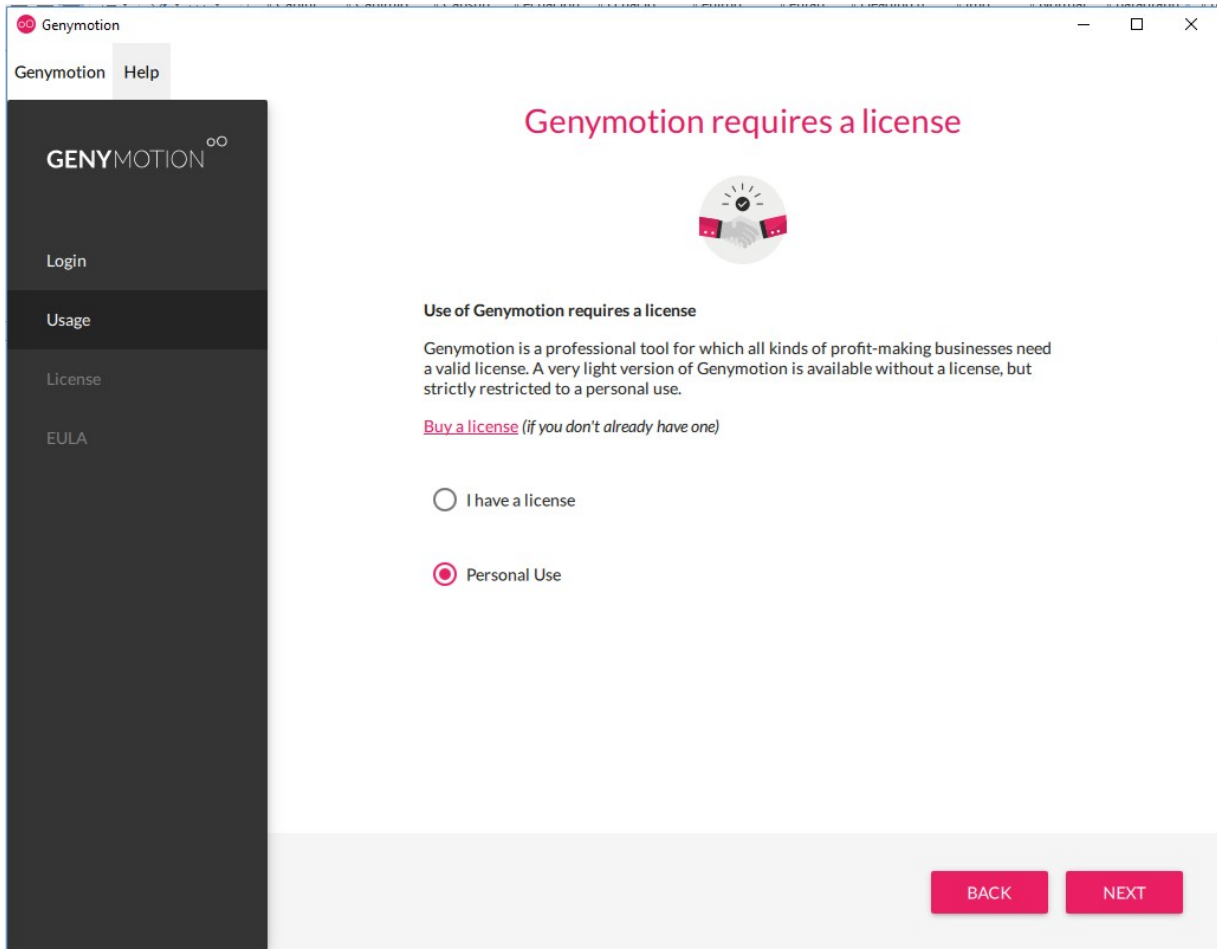


Figura 3.64 Licencia Genymotion

Se procede a aceptar los términos y condiciones de uso de la aplicación.

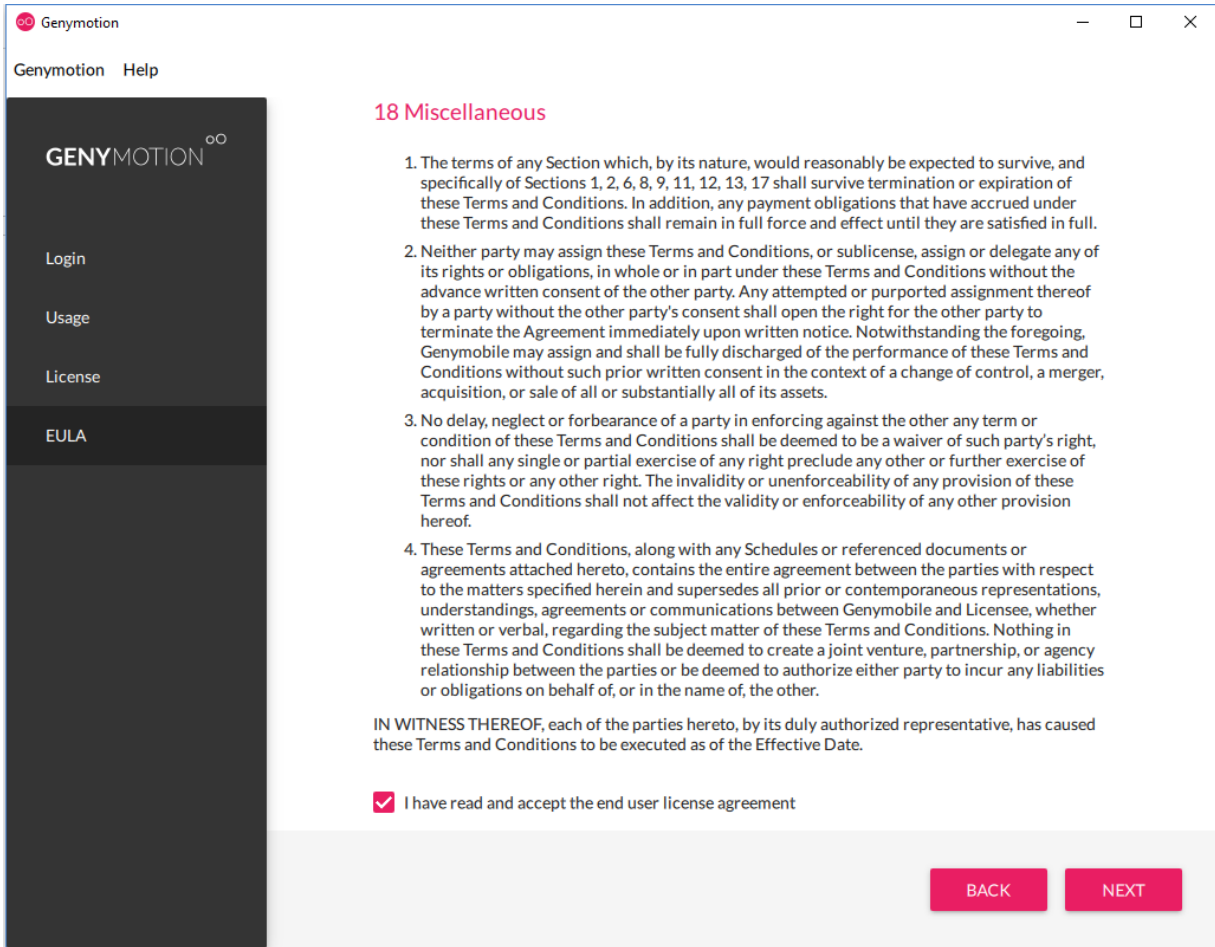


Figura 3.65 Términos y condiciones Genymotion

Ya terminando el proceso de selección de licencia y términos y condiciones de uso nos muestra la pantalla donde crearemos las máquinas virtuales de dispositivos Android para realizar las pruebas necesarias de nuestro trabajo.

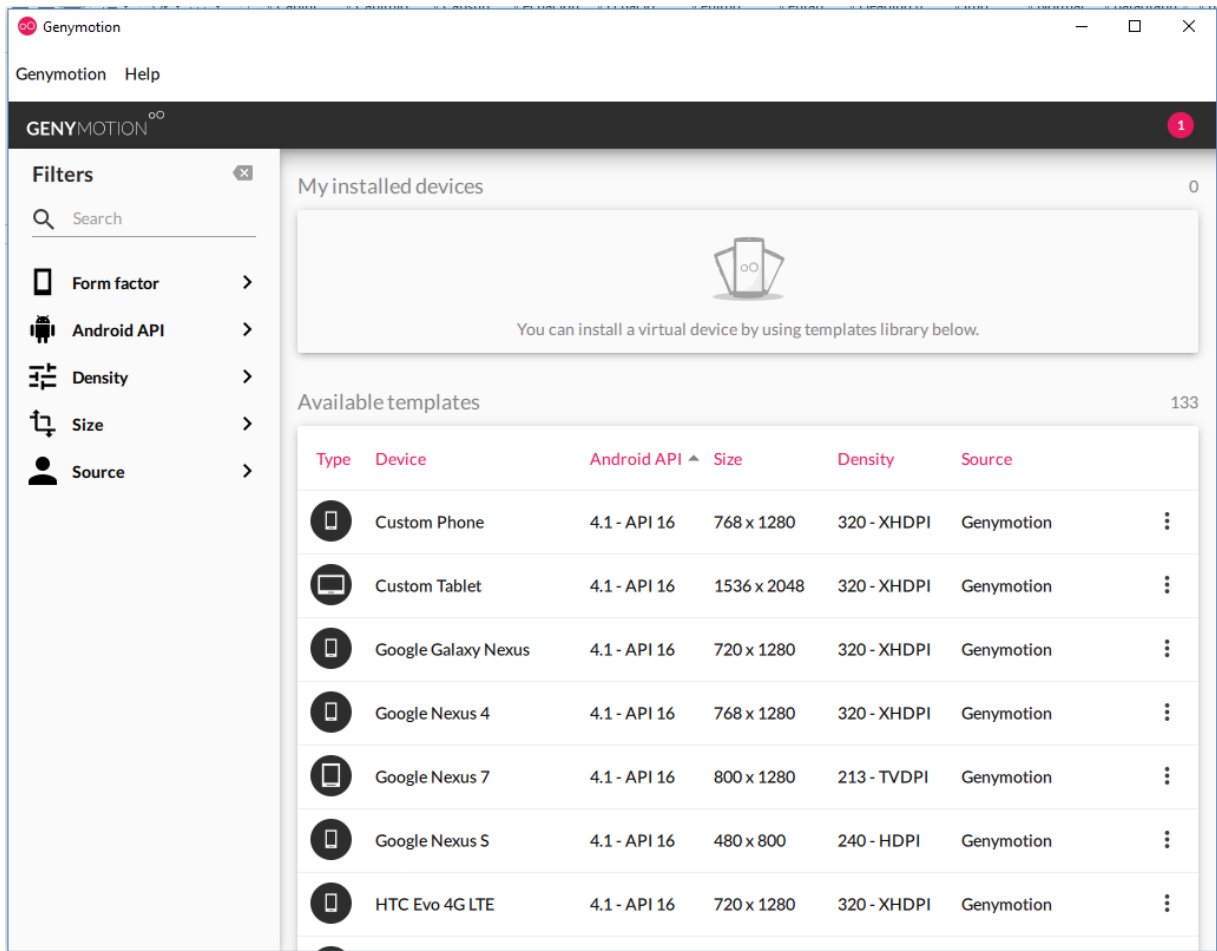


Figura 3.66 Interfaz Genymotion

- Spring source

Primeramente se procede a la descarga del paquete desde la página de Spring IO en la sección de herramientas y seleccionamos el paquete que en nuestro caso es para Windows de 64 bit como se muestra en la siguiente figura.

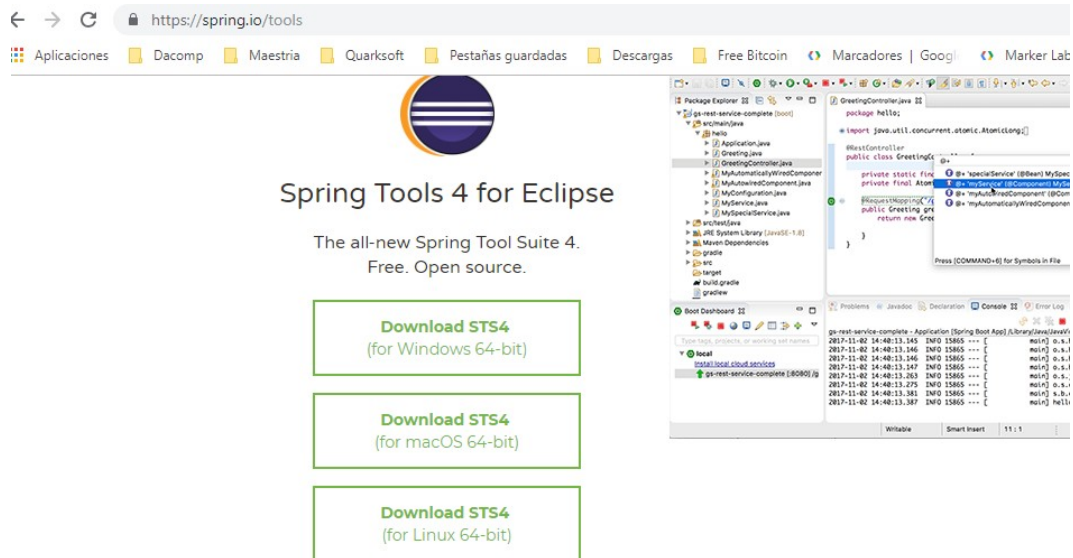


Figura 3.67 Interfaz Spring Source Tool Suit 4

Al terminar la descarga damos click en el archivo descargado y procedemos a la instalación de software.



Figura 3.68 Directorio de descarga Spring Source Tool Suit 4

Al terminar la descarga damos click en el archivo descargado y se descomprime en la carpeta destino como se muestra en la siguiente imagen.

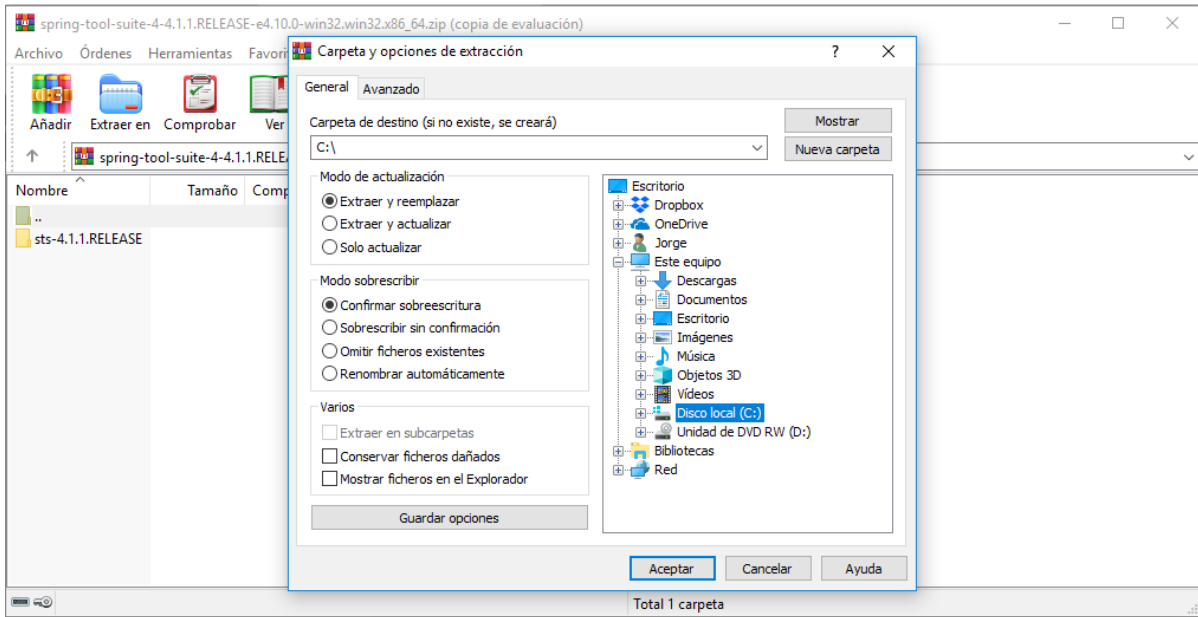


Figura 3.69 Descomprimir Spring Source Tool Suit 4

Al terminar la descompresión del archivo se mostrara la carpeta de Eclipse como se muestra en la siguiente figura.

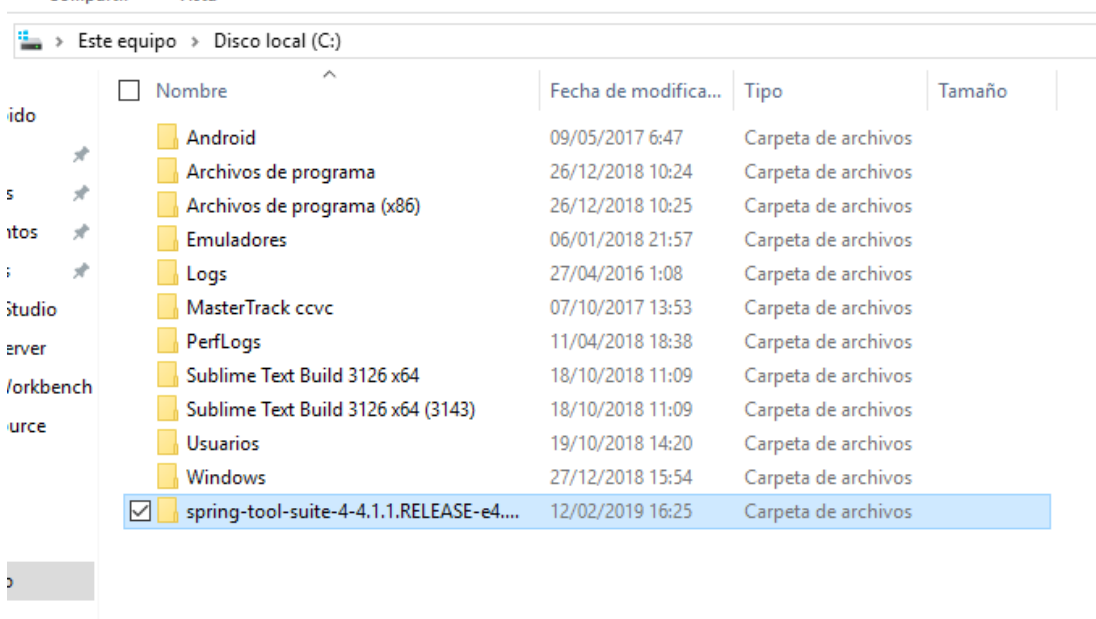


Figura 3.70 Directorio de instalación Spring Source Tool Suit 4

Dentro de la carpeta encontramos otra con la versión instala de la aplicación como se muestra en la siguiente imagen.

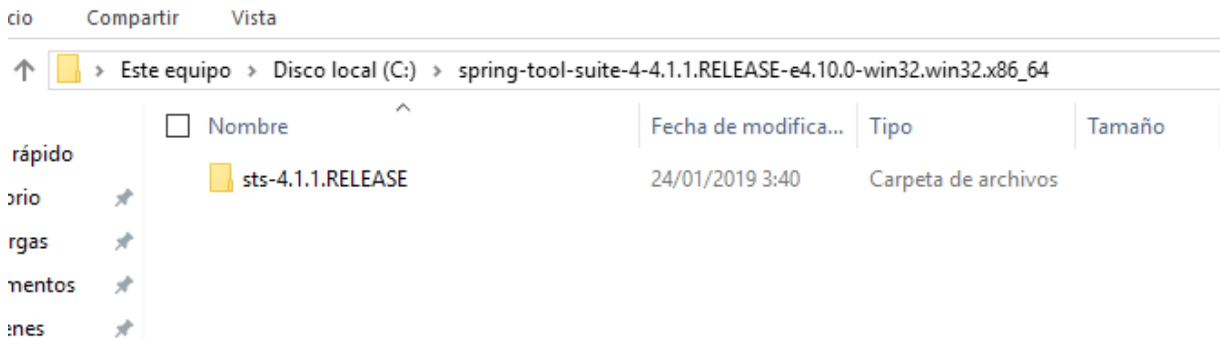


Figura 3.71 Directorio Spring Source Tool Suit 4

Dentro de la carpeta encontramos el iniciador de la aplicación SpringToolSuite4 como se muestra en la siguiente imagen, mediante el cual tenemos acceso al ID de desarrollo STS en el cual desarrollamos nuestro web service y aplicación web.

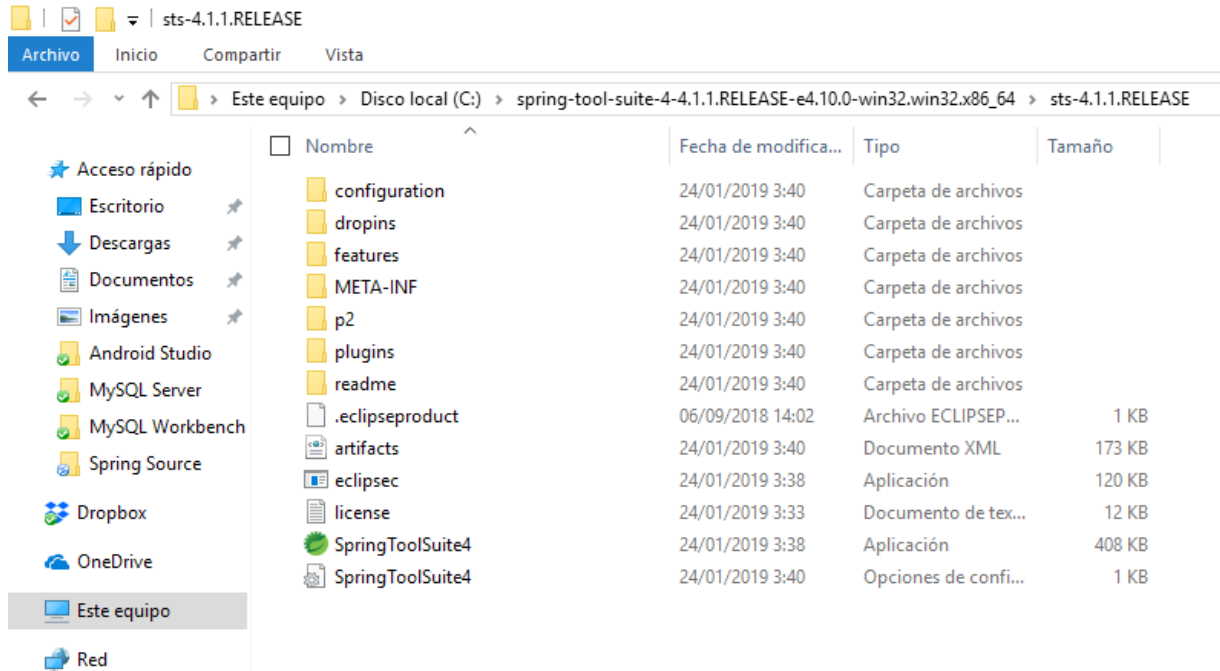


Figura 3.72 Directorio Spring Source Tool Suit 4

- Mysql workbench



Primeramente se procede a la descarga del paquete desde la página de MySQL en la sección de workbench como se muestra en la siguiente figura.

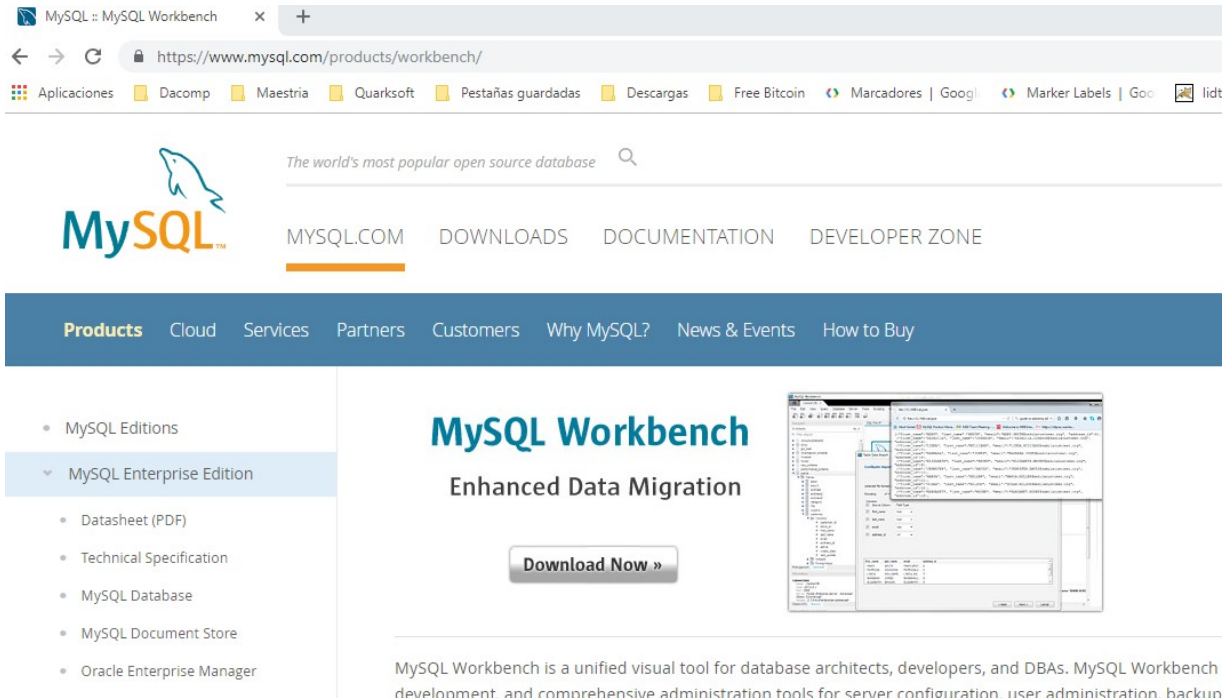


Figura 3.73 Descarga MySQL Workbench

Se selecciona el sistema operativo en el cual se instalara la aplicación.



Figura 3.74 Distribución MySQL Workbench

Se confirma la descarga de la aplicación.

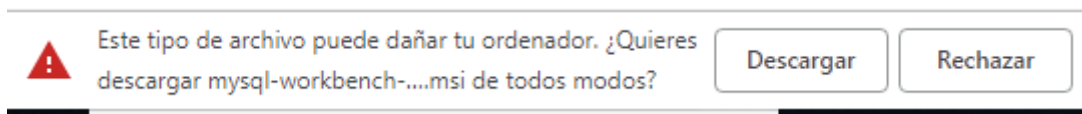


Figura 3.75 Confirmar descarga MySQL Workbench

Al terminar la descarga se da click sobre el archivo descargando para iniciar la instalación y configuración de la aplicación.

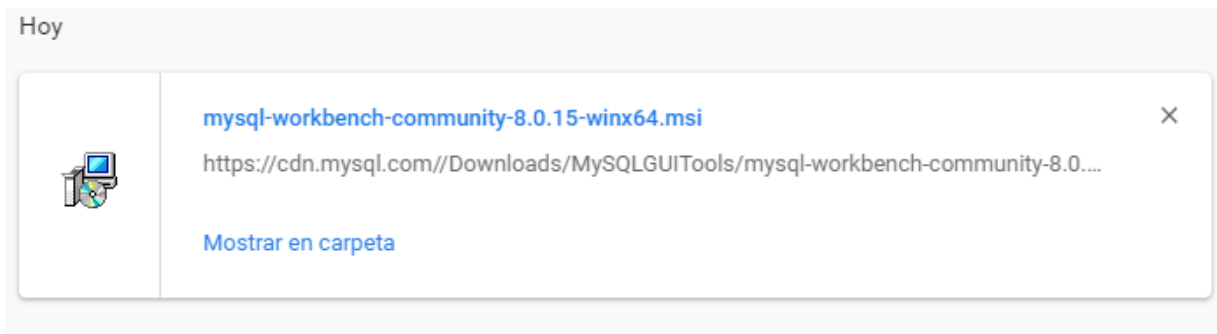


Figura 3.76 Directorio de descarga MySQL Workbench

Se inicia el proceso de instalación de la aplicación como se muestra en la siguiente imagen.

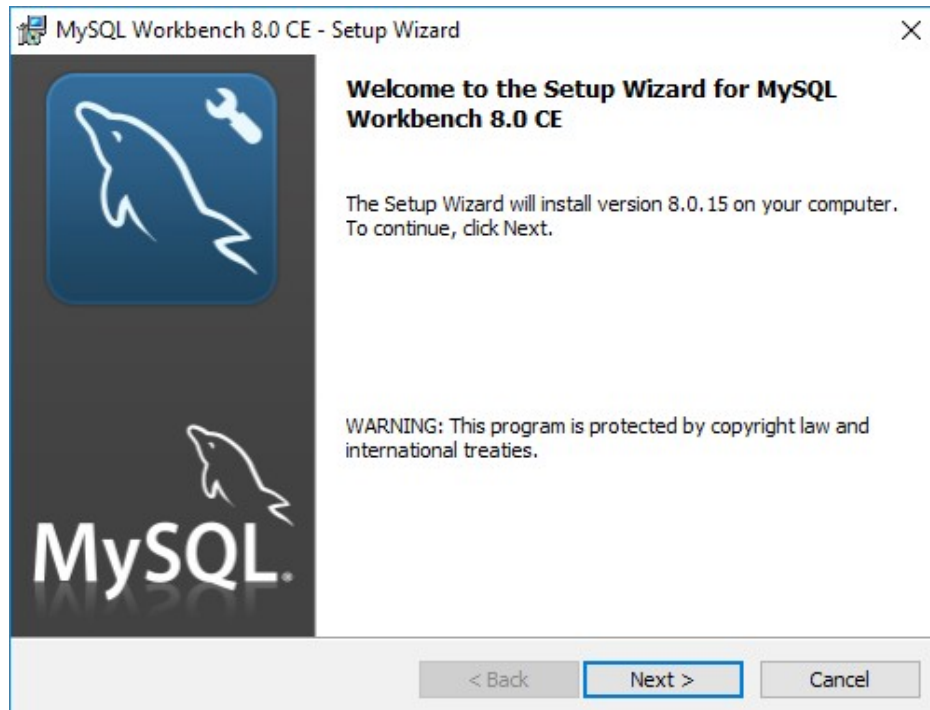


Figura 3.77 Instalación MySQL Workbench

Se selecciona el tipo de instalación que debido a las necesidades de nuestro proyecto instalamos la versión completa.

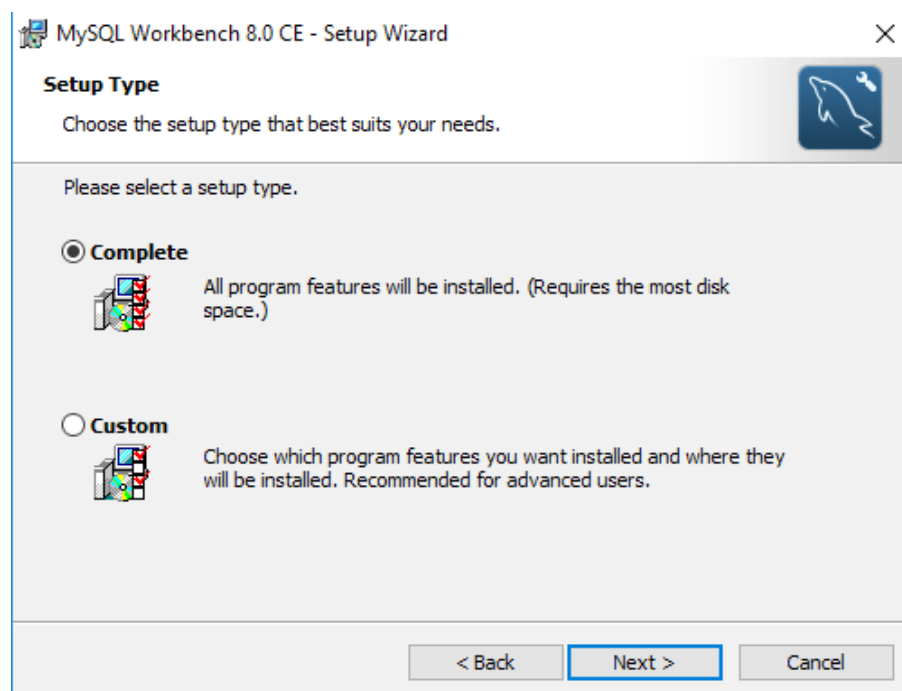


Figura 3.78 Configuración MySQL Workbench

Nos muestra un mensaje con la ruta donde se instala la aplicación como se muestra en la siguiente imagen.

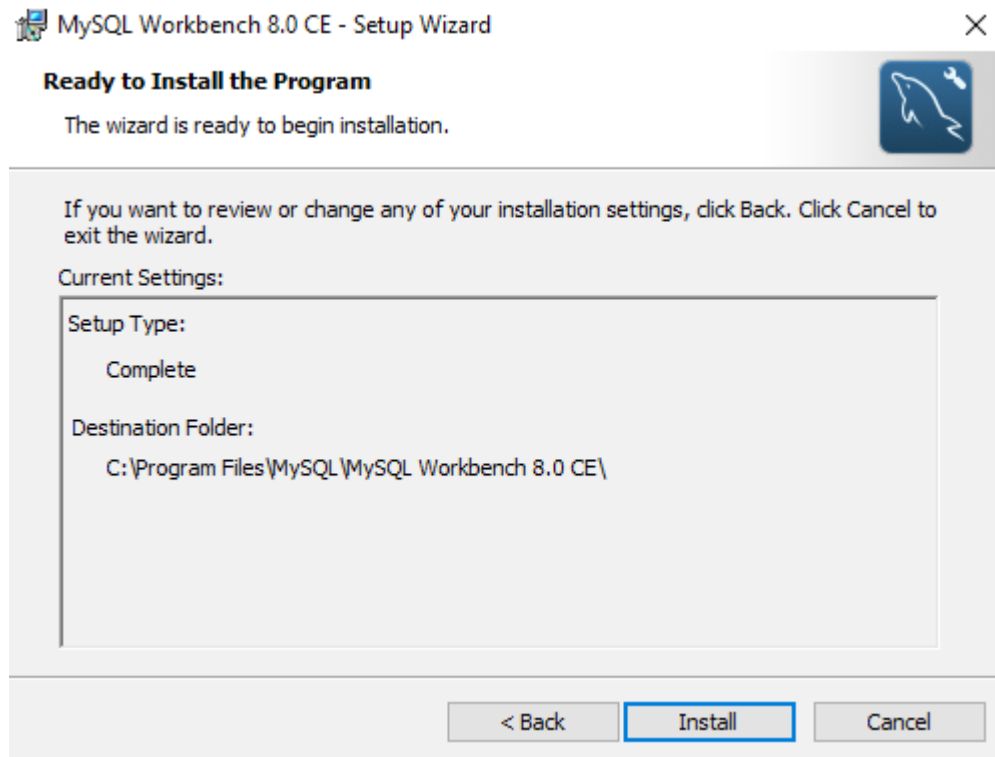


Figura 3.79 Confirmación de instalación MySQL Workbench

Al terminar la instalación de la aplicación nos permite iniciar la aplicación como se muestra en el siguiente imagen.

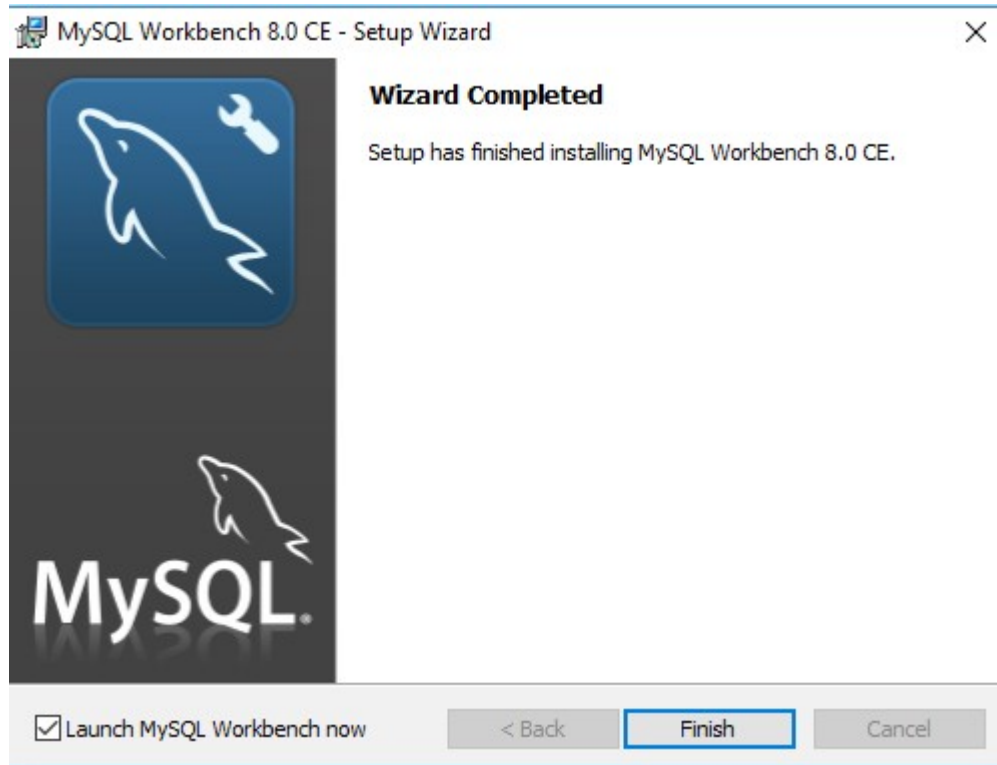


Figura 3.80 Termino de instalación MySQL Workbench

Al iniciar la aplicación nos muestra el estado de nuestro servidor MySQL con la opción de iniciarlo o detenerlo, con esto ya estamos listos para la creación y administración de nuestra base de datos.

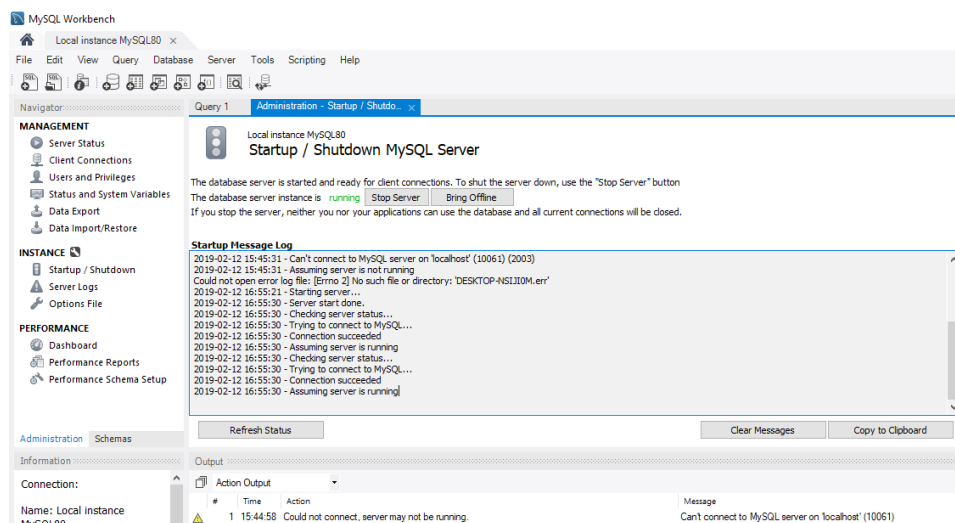


Figura 3.81 Interfaz Descarga MySQL Workbench

- Herramienta de prototipos

Primeramente se procede a la descarga del paquete desde la página de Balsamiq la versión de prueba gratuita como se muestra en la siguiente figura.

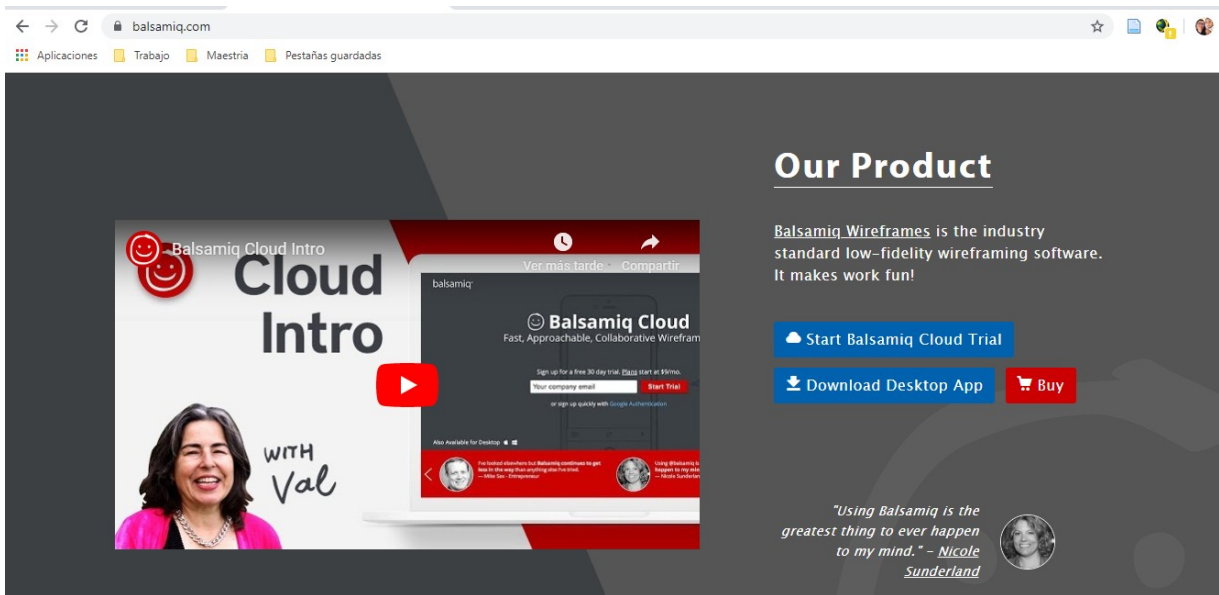


Figura 3.82 Descarga Balsamiq

Aceptamos los términos y condiciones de uso de la aplicación.

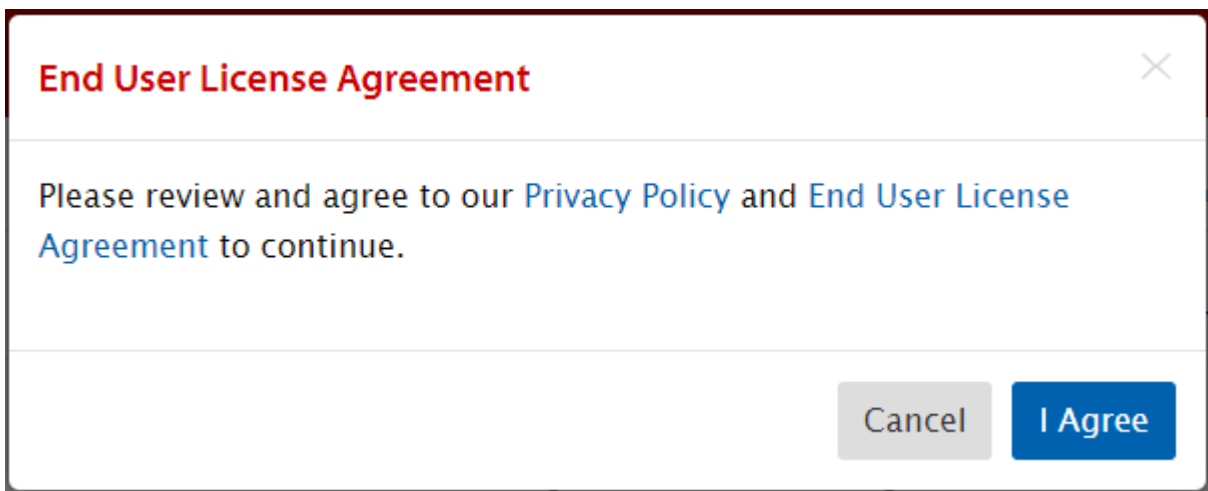


Figura 3.83 Términos y condiciones Balsamiq

Se confirma la descarga de la aplicación.

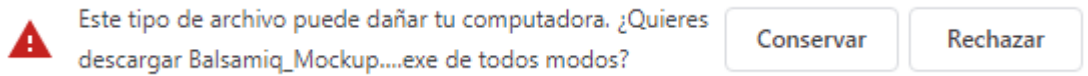


Figura 3.84 Confirmar descarga Balsamiq

Al terminar la descarga de proceder con la instalación dando click en el archivo descargado como se muestra en la siguiente imagen.

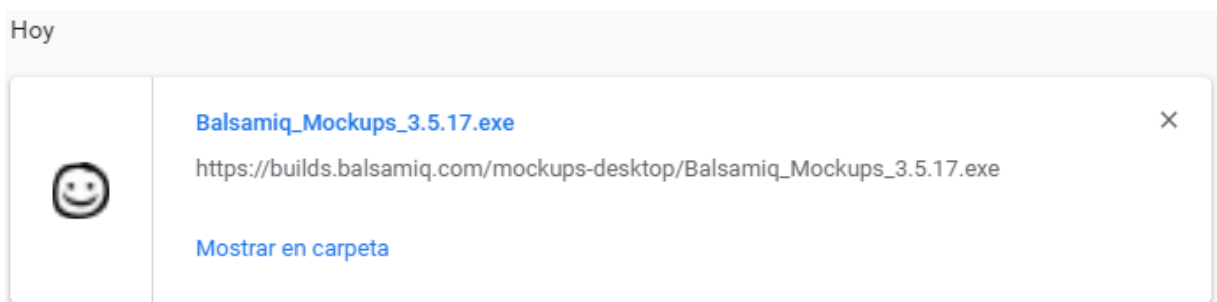


Figura 3.85 Directorio de descarga Balsamiq

Al iniciar la instalación de la aplicación nos muestra en pantalla la siguiente figura.

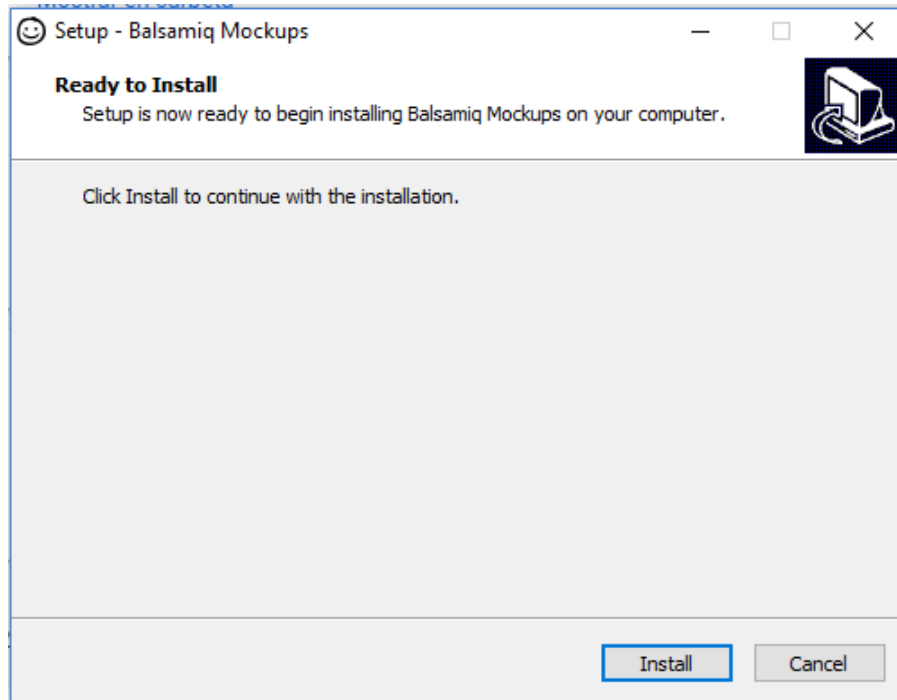


Figura 3.86 Instalación Balsamiq

Al terminar la instalación de la aplicación nos permite iniciar la aplicación como se muestra en la siguiente figura.

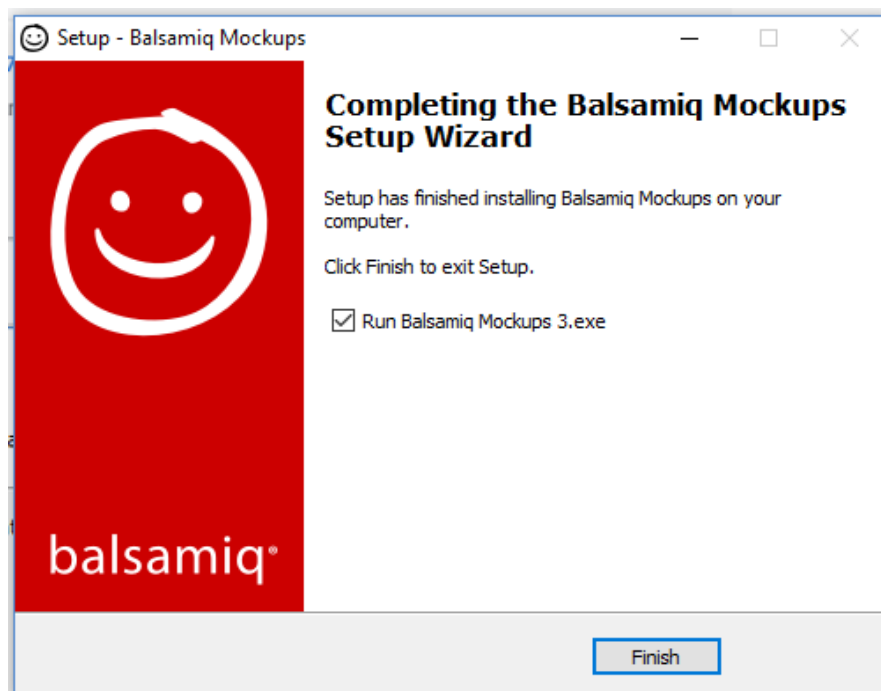


Figura 3.87 Finalización de instalación Balsamiq



Al iniciar la aplicación nos muestra la siguiente figura lo cual nos confirma que la aplicación fue instalada correctamente.

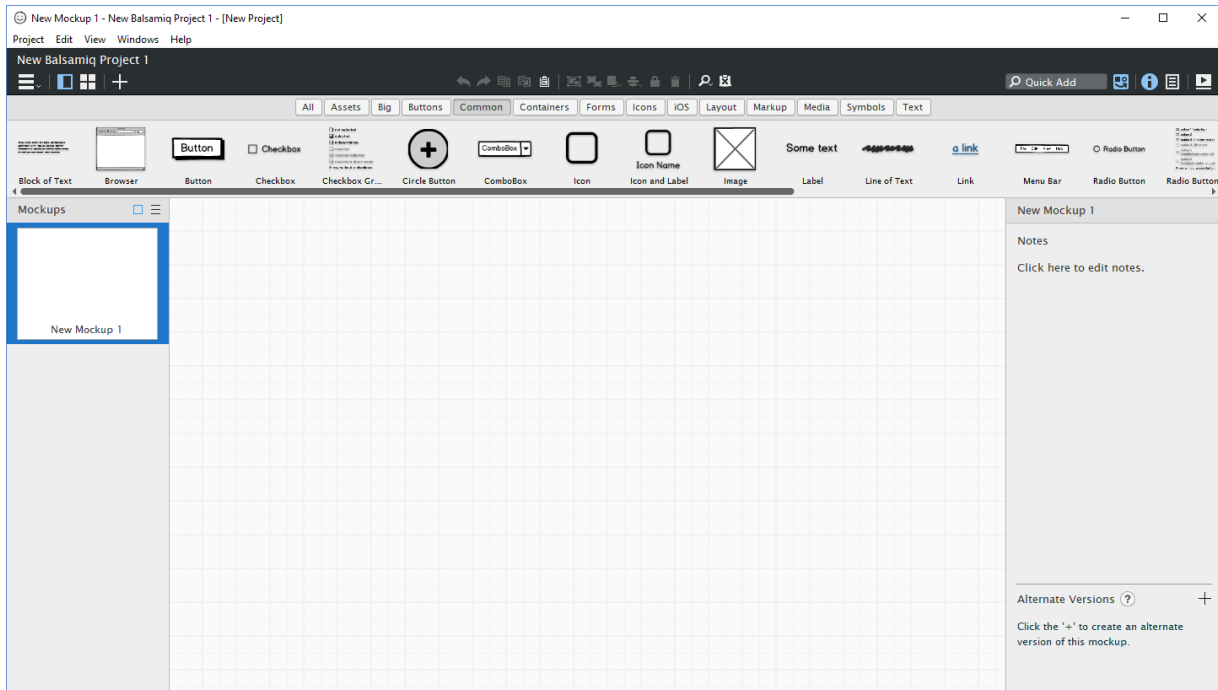


Figura 3.88 Interfaz Balsamiq

### 3.2.2. Desarrollo de prototipos para la aplicación Android y Web.

Comenzamos con el diseño de prototipos para la aplicación móvil la cual se encarga de recolectar la información de los dispositivos móviles, dicha aplicación consta de diferentes módulos: acceso al sistema, edificios históricos, preferencias por usuario y cerca de mí.

El módulo de acceso al sistema es el encargado de validar mediante un usuario y contraseña el acceso a la aplicación móvil, también permite el registro de nuevos usuarios, en la siguiente imagen se muestra el prototipo de la interfaz gráfica para el login.



Figura 3.89 Prototipo login aplicación móvil.

La aplicación cuenta con un menú principal donde se muestran los diferentes módulos de la aplicación los cuales son Edificios Históricas, Likes y Cerca de mí, el menú principal también cuenta con la opción de salir para borrar las credenciales de sesión activa.

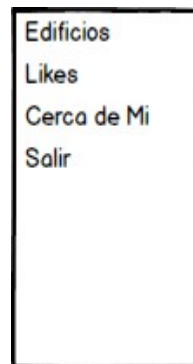


Figura 3.90 Prototipo menú principal aplicación móvil.

El módulo de edificios históricos se encarga de mostrar los edificios históricos almacenados en la base de datos, así como darle me gusta o no me gusta, posteriormente de ser evaluados no se muestra el edificio histórico.



Descripcion del Edificio Historico

Figura 3.91 Prototipo modulo edificios históricos aplicación móvil.

En el módulo de preferencias de usuario se muestra un listado de edificios históricos ordenado por nivel de preferencia del conjunto de usuarios.

Edificio Historico	Me gusta
Museo X	140
Plaza X	84
Iglesia X	13

Figura 3.92 Prototipo módulo likes aplicación móvil.

El módulo de cerca de mí se encarga de mostrar la ubicación actual así como los edificios históricos cercanos a dicha ubicación.



Figura 3.93 Prototipo módulo mapa aplicación móvil.

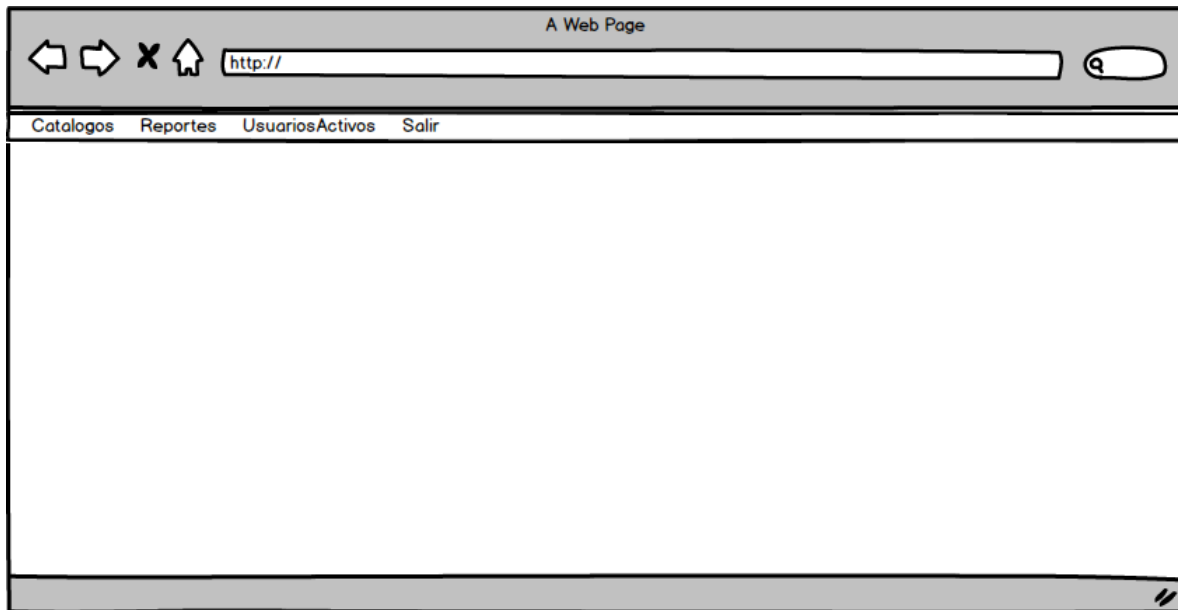
Continuamos con el diseño de prototipos para la aplicación Web la cual se encarga de administrar y monitorear la información recolectada por los dispositivos móviles, dicha aplicación consta de diferentes módulos: acceso al sistema, catálogo de usuarios, catálogo de edificios históricos, reporte de visitas a edificios históricos, reporte de trayectorias por usuario y mapa con la ubicación de los usuarios.

El módulo de acceso al sistema es el encargado de validar mediante un usuario y contraseña el acceso a la aplicación web, en la siguiente imagen se muestra el prototipo de la interfaz gráfica para el login.



Figura 3.94 Prototipo login aplicación web.

La aplicación cuenta con un menú principal donde se dividen los módulos en diferentes categorías las cuales son catálogos, reportes y usuarios activos, el menú principal también cuenta con la opción de salir para borrar las credenciales de sesión activa.



*Figura 3.95 Prototipo menú principal aplicación web.*

El catálogo de usuarios nos muestra los usuarios registrados en la aplicación móvil, en dicho catalogo podemos revisar el identificador único asignado al usuario así como el correo electrónico con el cual se dio de alta en nuestra aplicación móvil.

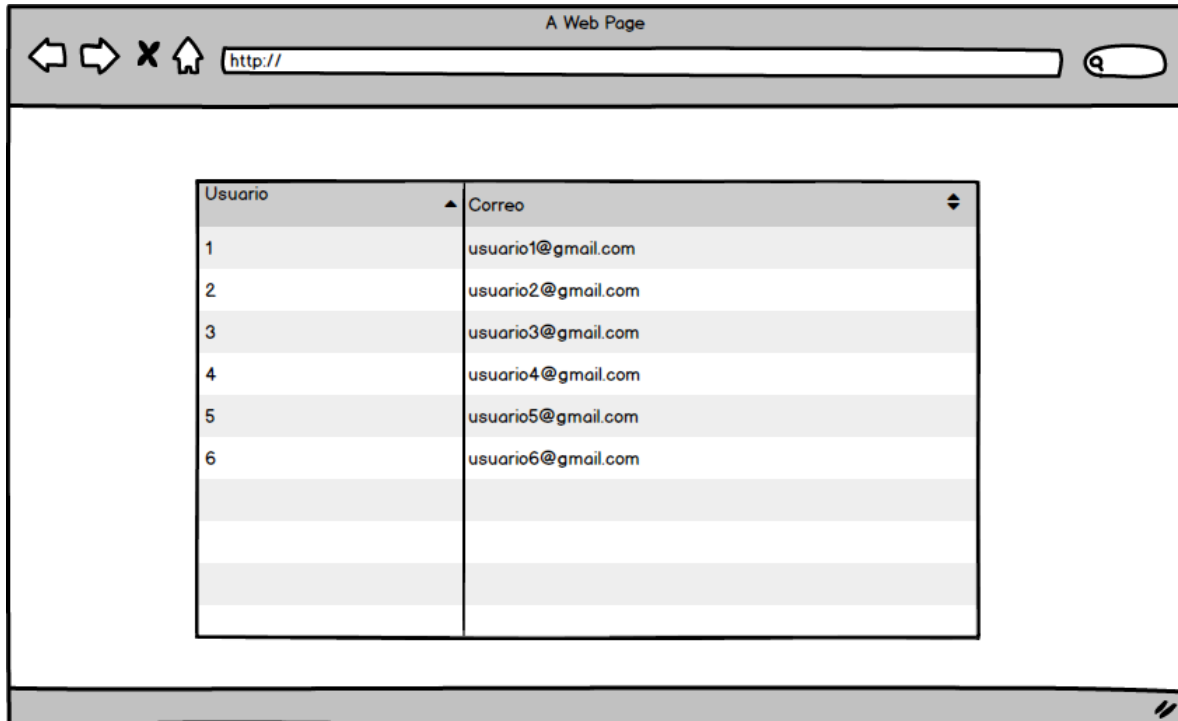
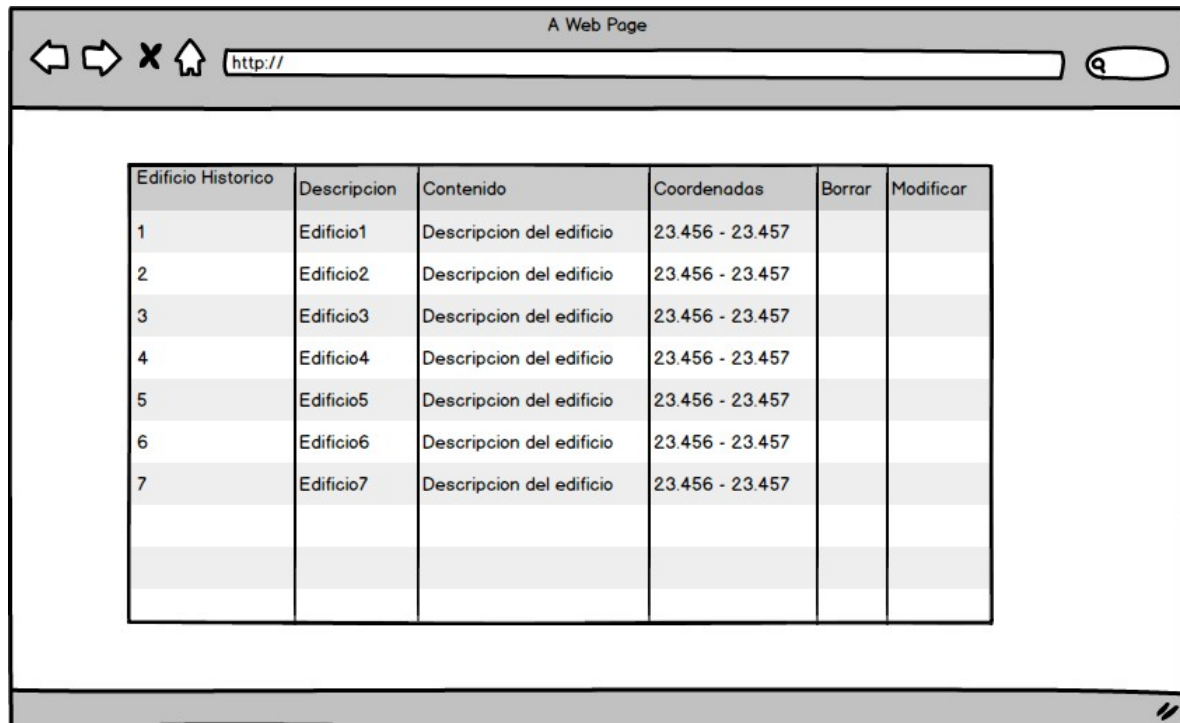


Figura 3.96 Prototipo catalogo usuarios aplicación web.

El catálogo de edificios históricos es el encargado de dar de alta, modificar, eliminar y consultar los edificios históricos mostrados en nuestra aplicación, en dicho catalogo se almacena una descripción o nombre del edificio, una descripción breve o el contenido, las coordenadas donde se encuentra ubicado así como la imagen representativa de dicho elemento.



The image shows a web browser window titled "A Web Page". The address bar contains "http://". The main content area displays a table with the following data:

Edificio Historico	Descripcion	Contenido	Coordenadas	Borrar	Modificar
1	Edificio1	Descripcion del edificio	23.456 - 23.457		
2	Edificio2	Descripcion del edificio	23.456 - 23.457		
3	Edificio3	Descripcion del edificio	23.456 - 23.457		
4	Edificio4	Descripcion del edificio	23.456 - 23.457		
5	Edificio5	Descripcion del edificio	23.456 - 23.457		
6	Edificio6	Descripcion del edificio	23.456 - 23.457		
7	Edificio7	Descripcion del edificio	23.456 - 23.457		

Figura 3.97 Prototipo catalogo edificios históricos aplicación web.

El reporte de visitas a edificios históricos se encarga de mostrar una tabla con el identificador de usuario, el edificio histórico y la fecha en la cual el usuario estuvo en el edificio histórico, este reporte debe de aceptar como parámetros de filtrado el usuario y el edificio histórico, si no se selecciona un usuario se toman todos los usuarios y si no se selecciona un edificio histórico se toman todos los edificios históricos.

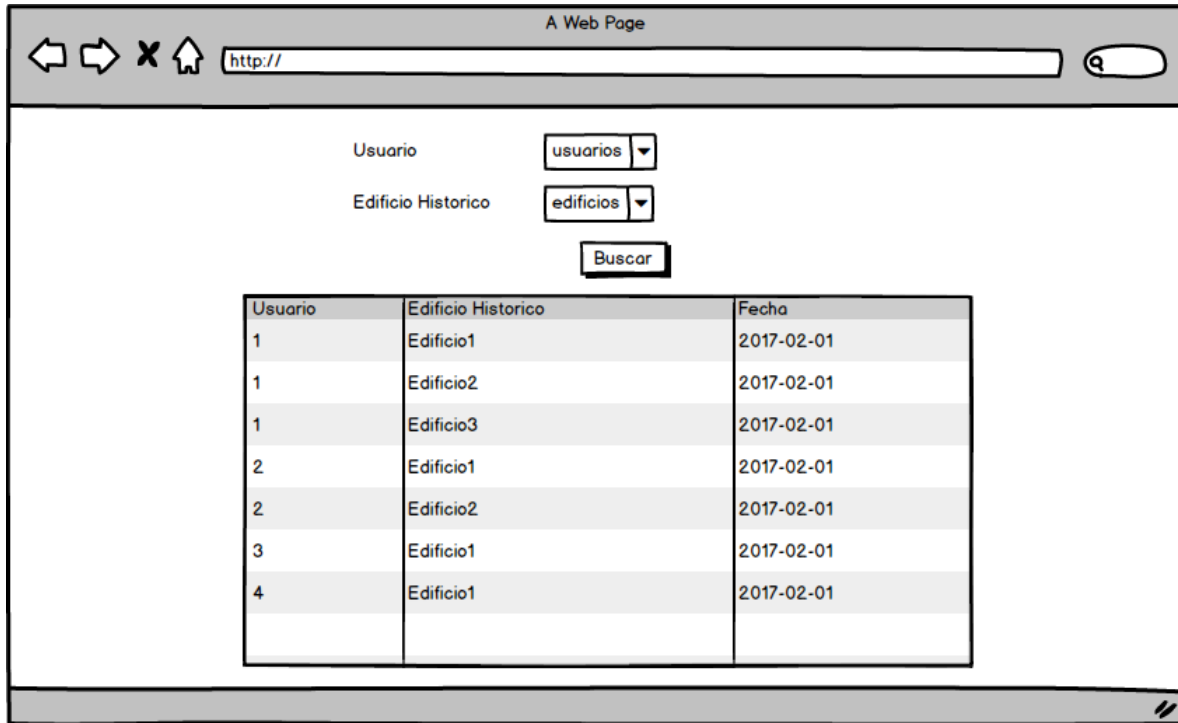


Figura 3.98 Prototipo reporte visitas a edificios historicos aplicación web.

El reporte de trayectorias por usuario se encarga de mostrar en un mapa las trayectorias de un usuario en una fecha determinada, como el reporte anterior nos permite dejar vacío el usuario y/o fecha para traer todo el conjunto de datos.



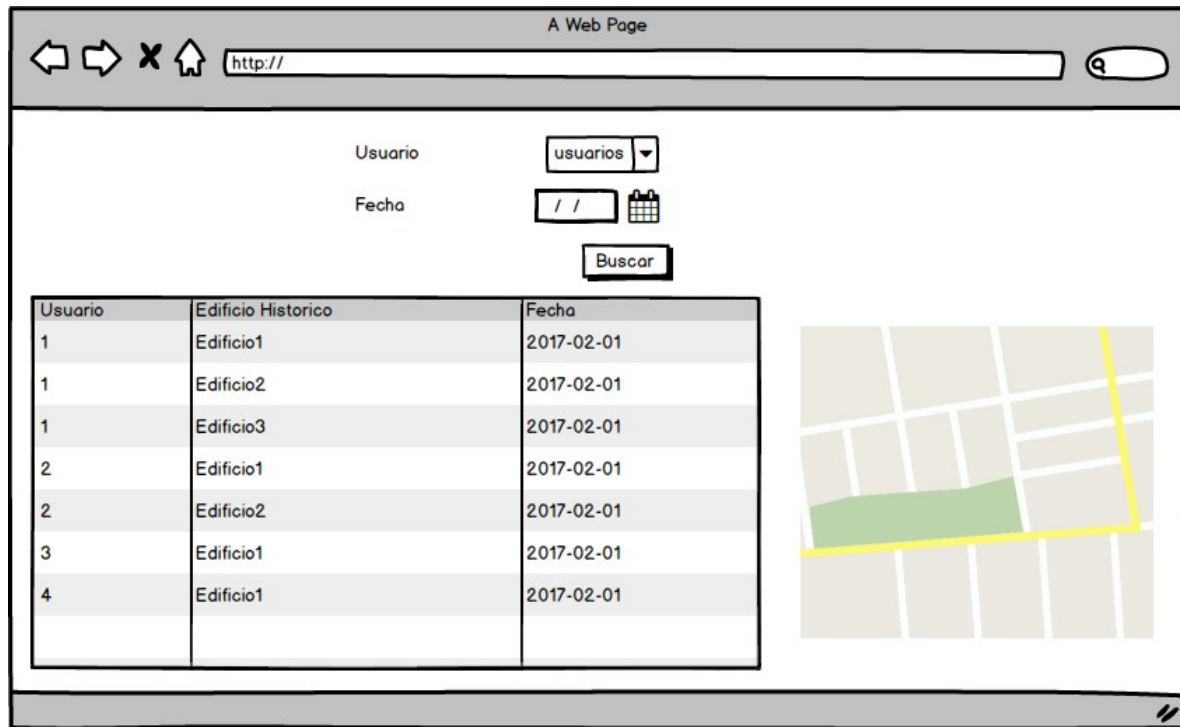


Figura 3.99 Prototipo reporte de trayectorias aplicación web.

El módulo de mapa se encarga de mostrar la ubicación de los usuarios que están en nuestra aplicación móvil.

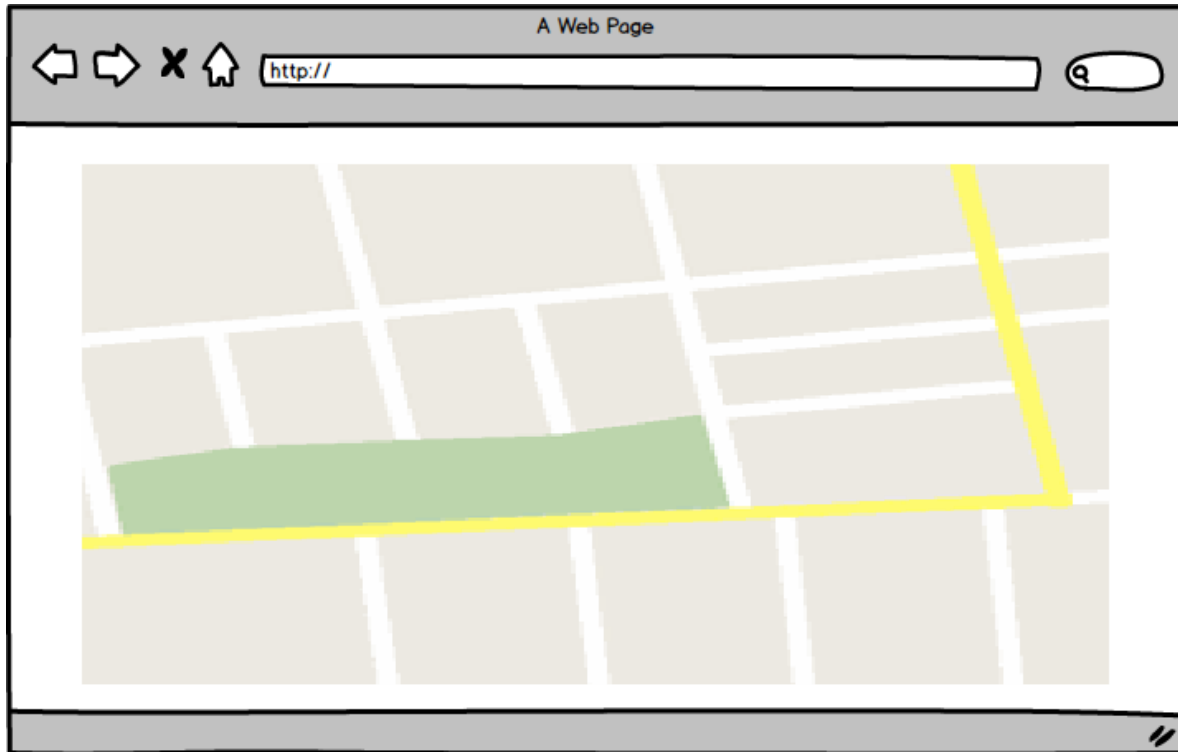


Figura 3.100 Prototipo mapa aplicación web.

### 3.3. Creación de mapas conceptuales para el servidor web y las aplicaciones Android y web:

El usuario accede a la aplicación web mediante el módulo **Login** el cual es el encargado de la autenticación al usuario, este solo permitirá el acceso en caso de que sea un usuario válido, posteriormente de autenticarse muestra el menú principal de nuestra aplicación web, la **Petición 5** se refiere a consulta del usuario si es válido o no y la **Respuesta 6** se refiere a la respuesta que se obtienen de la petición realizada en la **Petición 5**.

El usuario accede al módulo **Edificios Históricos** de nuestra aplicación web, en el cual se le muestra un listado de los edificios históricos así con las opciones de inserción, actualización, búsqueda o eliminación, el usuario selecciona cualquiera de estas opciones para llamar la **Petición 7** a la base de datos, se obtiene una **Respuesta 8**.

El usuario accede al módulo **Usuarios** de nuestra aplicación web, en el cual se le muestra un listado de los usuarios así con las opciones de inserción, actualización, búsqueda o eliminación, el usuario selecciona cualquiera de estas opciones para llamar la **Petición 9** a la base de datos, se obtiene una **Respuesta 10**.

El usuario accede al módulo **Visitas** de nuestra aplicación web, en el cual se le muestra un listado de los visitas así con las opciones de búsqueda, el usuario selecciona la opción de búsqueda para llamar la **Petición 11** a la base de datos, se obtiene una **Respuesta 12**.

El usuario accede al módulo **Trayectorias** de nuestra aplicación web, en el cual se le muestra un listado de los visitas así con las opciones de búsqueda, el usuario selecciona la opción de búsqueda para llamar la **Petición 13** a la base de datos, se obtiene una **Respuesta 14**.

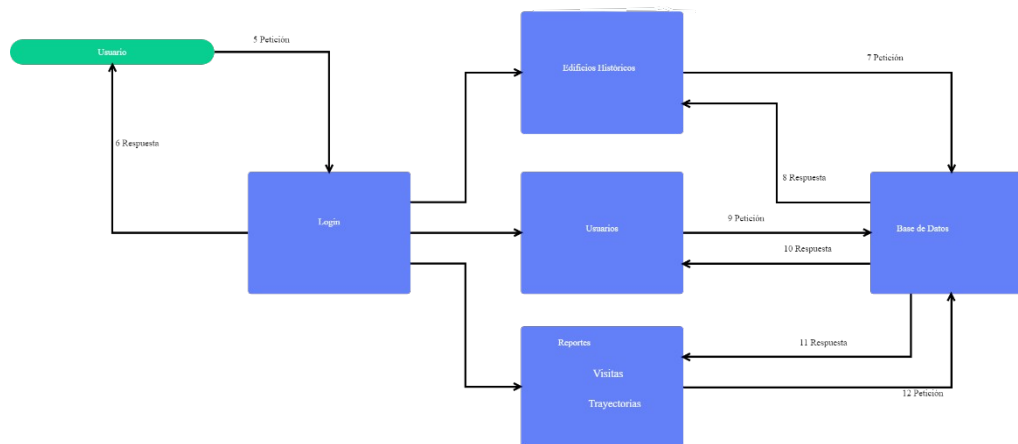


Figura 3.101 Mapa conceptual aplicación móvil.

El usuario accede a la aplicación móvil mediante el módulo **Login** el cual es el encargado de la autenticación y/o registro del usuario, este solo permitirá el acceso en caso de que sea un usuario válido, posteriormente de autenticarse muestra el menú principal de nuestra aplicación móvil, la **Petición 1** se refiere a las peticiones put, post, get o delete que realiza el dispositivo Android al servidor web sobre los

objetos **Edificios Históricos**, **Usuarios**, **Visitas**, **Registro Coordenadas** y **Preferencias Usuario** y la **Respuesta 2** se refiere a las respuestas que se obtienen de las peticiones realizadas en la **Petición 1**.

El usuario accede al módulo **Preferencias Usuario** en el cual se le muestra un listado de los edificios históricos así como una breve reseña, su imagen y una opción de like o dislike, el usuario selecciona la opción de la like o dislike y en la **Petición 1** se manda al servidor web el cual hace una **Petición 3** a la base de datos, se obtiene una **Respuesta 4** y se regresa una **Respuesta 2** al dispositivo móvil.

El usuario accede al módulo **Visitas** en el cual se le muestra un listado de los edificios históricos así como su calificación, para obtener esta información se manda una **Petición 1** al servidor web el cual hace una **Petición 3** a la base de datos, se obtiene una **Respuesta 4** y se regresa una **Respuesta 2** al dispositivo móvil.

El usuario accede al módulo **Registro Coordenadas**, el cual muestra un mapa con la ubicación actual del usuario y los marcadores de los **Edificios Históricos**, para obtener esta información se manda una **Petición 1** al servidor web el cual hace una **Petición 3** a la base de datos, se obtiene una **Respuesta 4** y se regresa una **Respuesta 2** al dispositivo móvil.

El usuario selecciona un marcador en el módulo de **Registro Coordenadas**, se muestra una reseña del Edificio Histórico y se registra la visita, para obtener esta información se manda una **Petición 1** al servidor web el cual hace una **Petición 3** a la base de datos, se obtiene una **Respuesta 4** y se regresa una **Respuesta 2** al dispositivo móvil.

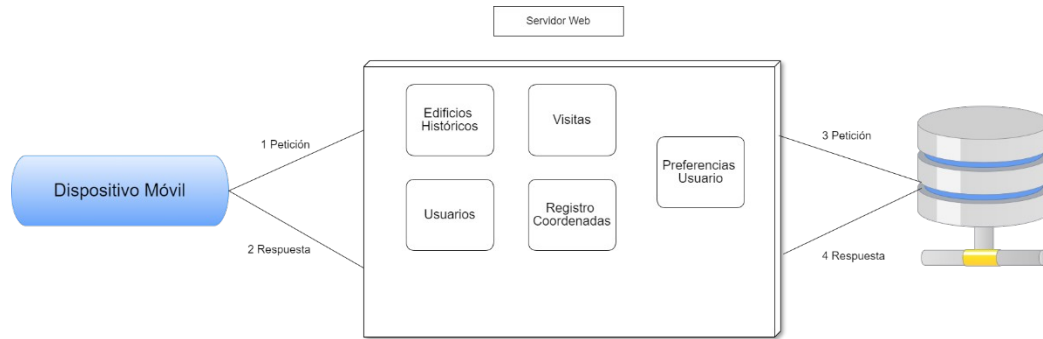


Figura 3.102 Mapa conceptual servidor web.

El servidor web en el módulo de **Edificios Históricos Controlador** donde se reciben las peticiones de los dispositivos móviles para realiza las acciones de agregar, modificar, consultar y/o eliminar manda llamar al módulo **Edificios Históricos Servicio** para validar las reglas de negocio de la acción a realizar el cual a su vez manda llamar al **Edificios Históricos DAO** para validar que la estructura de los datos sea la correcta y realizar una **Petición 13** y obtener una **Respuesta 14** de la base de datos.

El servidor web en el módulo de **Usuarios Controlador** donde se reciben las peticiones de los dispositivos móviles para realiza las acciones de agregar, modificar, consultar y/o eliminar manda llamar al módulo **Usuarios Servicio** para validar las reglas de negocio de la acción a realizar el cual a su vez manda llamar al **Usuarios DAO** para validar que la estructura de los datos sea la correcta y realizar una **Petición 15** y obtener una **Respuesta 16** de la base de datos.

El servidor web en el módulo de **Visitas Controlador** donde se reciben las peticiones de los dispositivos móviles para realiza las acciones de agregar, modificar, consultar y/o eliminar manda llamar al módulo **Visitas Servicio** para validar las reglas de negocio de la acción a realizar el cual a su vez manda llamar al **Visitas DAO** para validar que la estructura de los datos sea la correcta y realizar una **Petición 17** y obtener una **Respuesta 18** de la base de datos.

El servidor web en el módulo de **Registro Coordenadas Controlador** donde se reciben las peticiones de los dispositivos móviles para realiza las acciones de agregar, modificar, consultar y/o eliminar manda llamar al módulo **Registro Coordenadas Servicio** para validar las reglas de negocio de la acción a realizar el cual a su vez manda llamar al **Registro Coordenadas DAO** para validar que la estructura de los datos sea la correcta y realizar una **Petición 19** y obtener una **Respuesta 20** de la base de datos.

El servidor web en el módulo de **Preferencias Usuario Controlador** donde se reciben las peticiones de los dispositivos móviles para realiza las acciones de agregar, modificar, consultar y/o eliminar manda llamar al módulo **Preferencias Usuario Servicio** para validar las reglas de negocio de la acción a realizar el cual a su vez manda llamar al **Preferencias Usuario DAO** para validar que la estructura de los datos sea la correcta y realizar una **Petición 21** y obtener una **Respuesta 22** de la base de datos.

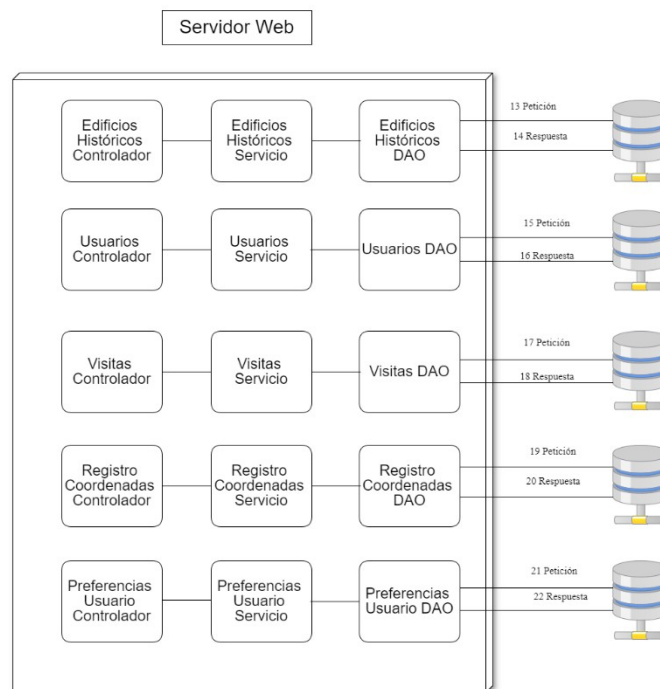


Figura 3.103 Mapa conceptual servidor web.

### 3.4. Diseño de la base de datos en el programa MySQL Workbench:

Se muestra en la siguiente figura de modo grafico la información guardada en la base de datos gracias al uso de la aplicación MySQL Workbench.

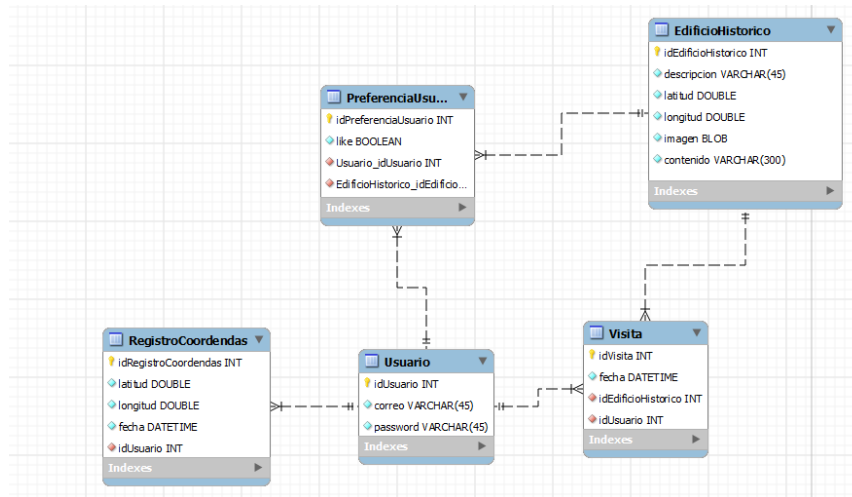


Figura 3.104 Diseño de base de datos.

### 3.5. Desarrollo de aplicaciones

Se desarrolló aplicación Android para la recolección de trayectorias utilizando el lenguaje Android, el cual se basa en la tecnología Java, el Id en el que se desarrolló la aplicación móvil es Android Studio, para la conexión con el servidor web se manejaron peticiones de tipo REST, las cuales se manipularon con el uso de JSON.

Se desarrolló de aplicación Web para el análisis y monitoreo de trayectorias utilizando la tecnología Java, el Id en el que se desarrolló la aplicación web es Spring Tool Suit versión 3.8.2, en el cual se creó un proyecto de tipo “Dynamic Web Project”.

Para la conversión del modelo relacional que nos proporciona la base de datos en mysql a un modelo orientado a objetos utilizamos el framework de java Hibernate y

con esto creamos el paquete de modelos y generamos sus archivos de configuración y su consola.

Utilizamos el framework de java Spring para la creación de singletons y su única inicialización de objetos para pasar por el modelo MVC (Model View Controller) y hacer más ligera su ejecución, los modelos son los generados previamente por el framework hibernate, los controladores son donde definimos las reglas de negocio de la aplicación y la vista en la interfaz por la cual va a interactuar el usuario con nuestra aplicación en la cual utilizamos el framework de java RichFaces.

Se desarrolló del servidor web para la conexión entre el dispositivo móvil y la base de datos, el Id en el que se desarrolló la aplicación web es Spring Tool Suit versión 3.8.2, en el cual se creó un proyecto de tipo Maven para el manejo de las librerías utilizadas.

Para la conversión del modelo relacional que nos proporciona la base de datos en mysql a un modelo orientado a objetos utilizamos el framework de java Hibernate, utilizamos el framework de java Spring para la creación de singletons y su única inicialización de objetos para pasar por el modelo MVC (Model View Controller) y hacer más ligera su ejecución.



---

## Capítulo 4

# Resultados y discusión

---

## 4.1. Aplicaciones

### 4.1.1. Aplicación Móvil

La aplicación cuenta con un módulo de logeo y/o registro y un menú principal, el cual contiene los módulos de *Edificios Históricos*, *Likes* y *Cerca de mí*.

Gracias a la creación de la aplicación móvil y el módulo logeo y/o registro permite la creación de nuevos usuarios y validación mediante el uso de un correo electrónico y una contraseña como se muestra en la siguiente figura.

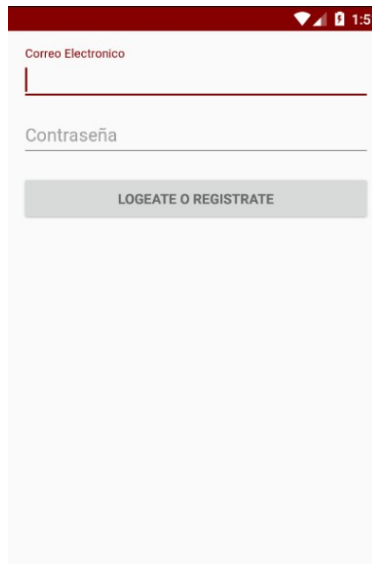


Figura 4.1 Login aplicación móvil.

Como se observa en la primera figura tenemos acceso a los diferentes módulos del sistema con el uso de un menú principal.

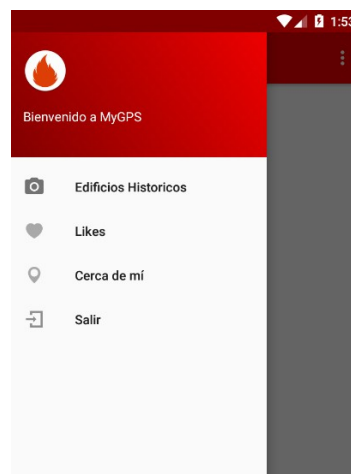


Figura 4.2 Menú principal aplicación móvil

En el módulo de Edificios históricos tenemos una colección de los edificios registrados en nuestra aplicación, así como su información para poder dar una calificación de me gusta deslizando hacia la derecha o no me gusta deslizando hacia la izquierda, de la misma forma la aplicación nos muestra los edificios históricos que no han sido calificados, tenemos una vista del módulo en la siguiente figura.

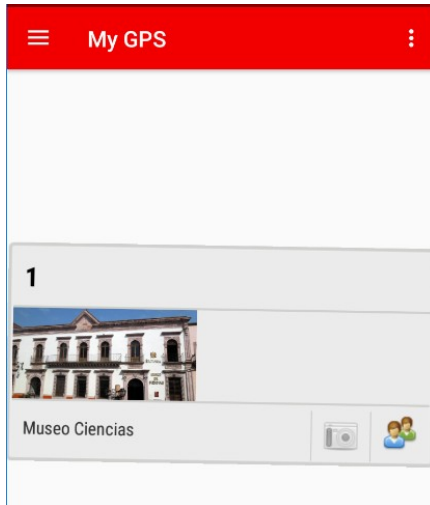


Figura 4.3 Edificios históricos aplicación móvil.

Por ultimo en la aplicación móvil nos muestra un mapa con la ubicación actual del usuario, así como los edificios históricos que se encuentran cerca de la ubicación del usuario, en la siguiente figura podemos observar el mapa de nuestra aplicación.



Figura 4.4 Mapa aplicación móvil.

#### 4.1.2. Servidor Web

El servidor web para la conexión entre el dispositivo móvil y la base de datos, contiene los servicios de Edificios Historicos, Preferencias de Usuario, Registro de Coordenadas y Usuarios.



Figura 4.7 Login aplicación web.

Para la administración de edificios históricos de manera más efectiva relacionado con las actividades de creación, actualización, eliminado y búsqueda nuestro sistema cuenta con un catálogo de edificios históricos donde podemos manipular de manera gráfica la información almacenada en nuestra base de datos como se muestra en la siguiente figura.

No. Edificio:

Descripcion:

Lista Edificios Historicos					
Edificio Historico	Descripcion	Contenido	Coordenadas	Borrar	Modificar
1	Iglesia Santo Domingo	Iglesia de Santo Domingo contenido completo	22.776833 - -102.573409		
2	Museo Ciencias	Museo de Ciencias	22.773479 - -102.575047		
5	Francisco Goytia	Francisco Goytia	22.770342 - -102.577564		
6	Museo Pedro coronel	Museo Pedro Coronel	22.776095 - -102.573664		
7	Plaza Bicentenario	plaza bicentenario	22.769537 - -102.571893		
8	Plazuela de Garcia	Plazuela de Garcia	22.782524 - -102.571287		
9	PlazuelaFransiscoGoytia	PlazuelaFransiscoGoytia	22.774909 - -102.572964		
10	UAZ Ingenieria Electrica	UAZ	22.768634 - -102.565828		
11	MiCasa	Mi casa	22.737592 - -102.485302		

Figura 4.8 Catálogo de edificios históricos aplicación web.

Para acceder de una manera más organizada y parametrizada de la información recolectada en nuestra base de datos, la aplicación cuenta con un reporte de visitas a edificios históricos donde podemos filtrar por usuario y/o edificio histórico como se muestra en la siguiente figura.

Usuario:

Edificio Historico:

Lista Edificios Historicos		
Usuario	Edificio Historico	Fecha
1	Iglesia Santo Domingo	2017-09-20 14:11:33.0
1	Museo Ciencias	2017-09-20 14:11:44.0
1	Francisco Goytia	2017-09-20 14:11:52.0
1	MiCasa	2018-01-02 22:32:34.0
1	PlazuelaFransiscoGoytia	2018-01-02 23:16:03.0

Figura 4.9 Reporte de visitas a edificios históricos aplicación web.

La aplicación también cuenta con un reporte de trayectorias de usuarios donde podemos filtrar por usuario y/o fecha de registro, el cual nos proporciona un listado de los edificios históricos visitados con un mapa con la trayectoria realizada por el usuario, así como se muestra en la siguiente figura.

Usuario: 1

Fecha: sep 20, 2017

Lista Edificios Historicos		
Usuario	Edificio Historico	Fecha
1	Iglesia Santo Domingo	2017-09-20 14:11:33.0
1	Museo Ciencias	2017-09-20 14:11:44.0
1	Francisco Goytia	2017-09-20 14:11:52.0

Buscar

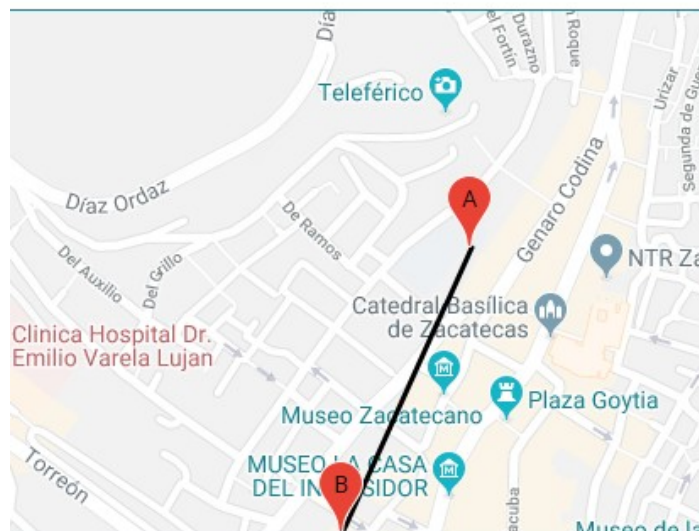


Figura 4.10 Reporte de trayectorias aplicación web.

Por ultimo nuestra aplicación cuenta con un mapa con la ubicación de los usuarios de nuestra aplicación móvil como se muestra en la siguiente imagen.

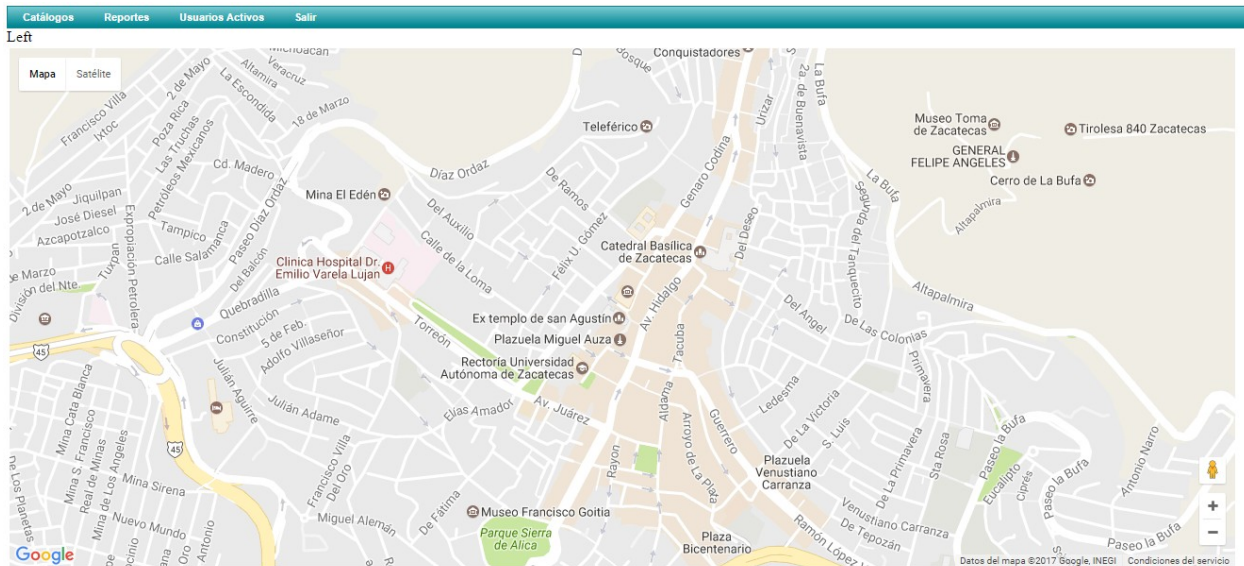


Figura 4.11 Mapa aplicación web.

## 4.2. Participación en congresos

### 4.2.1. 17vo Seminario de Investigación

Jorge Armando Ortiz Murillo, José Manuel Ortiz Rodríguez, Ma. Del Rosario Martínez Blanco, Gerardo Ornelas Vargas, Luis Octavio Solís Sánchez, Rodrigo Castañeda Miranda, Héctor Rene Vega Carrillo, José M Celaya Padilla, Jorge I Galván Tejada, Carlos E Galván Tejada, Idalia Garza Veloz, Margarita Martínez Fierro, Universidad Autónoma de Zacatecas, Av. Ramón López Velarde 801, Col. Centro 98000 Zacatecas, México, Centro de Investigación e Innovación Tecnológica Industrial (CIITI), Grupo de Investigación Regional Emergente (GIRE), Laboratorio de Innovación y Desarrollo Tecnológico en Inteligencia Artificial (LIDTIA), Unidad Académica de Ingeniería Eléctrica (UAIE), Unidad Académica de Estudios Nucleares (UAEN), Laboratorio de Medicina Molecular, Unidad Académica de Medicina



Humana y Ciencias de la Salud. Red neuronal de propagación hacia atrás para el reconocimiento de caracteres.

#### **4.2.2. XVI International Symposium on Solid State Dosimetry**

Jorge Armando Ortiz Murillo, José María Celaya Padilla, María Del Rosario Martínez Blanco, Luis Octavio Solís Sánchez, Rodrigo Castañeda Miranda, Idalia Garza Veloz, Margarita Martínez Fierro, Yamilé López Hernández, José Manuel Ortiz Rodríguez, Universidad Autónoma de Zacatecas, Av. Ramón López Velarde 801, Col. Centro 98000 Zacatecas, México, Centro de Investigación e Innovación Tecnológica Industrial (CIITI), Laboratorio de Innovación y Desarrollo Tecnológico en Inteligencia Artificial, (LIDTIA), Laboratorio de Bioinformática, Unidad Académica de Ingeniería Eléctrica (UAIE). Detección de cáncer de mama usando técnicas avanzadas de minería de datos con redes neuronales.

### **4.3. Certificaciones**

#### **4.3.1. Big Data Integration and Processing**

Certificado en integración y procesamiento Big Data autorizado por la universidad de california, san diego y ofrecido a través de coursera.

#### **4.3.2. Big Data Modeling and Management Systems**

Certificado en modelado y administración de sistemas Big Data autorizado por la universidad de california, san diego y ofrecido a través de coursera.

#### **4.3.3. Clasificación de Imágenes**

Certificado en clasificación de imágenes ¿Cómo reconocer el contenido de una imagen?, autorizado por la universidad autónoma de Barcelona y ofrecido a través de coursera.

#### **4.3.4. Fundamentals of Digital Image and Video Processing**

Certificado en fundamentos de imagen digital y procesamiento de video autorizado por la universidad de Northwestern y ofrecido a través de coursera.

#### **4.3.5. Introducción a Big Data**

Certificado en introducción a Big Data autorizado por la universidad de california, san diego y ofrecido a través de coursera.

#### **4.3.6. Machine Learning**

Certificado en machine learning autorizado por la universidad de Stanford y ofrecido a través de coursera.

### **4.4. Discusión**

El objetivo del proyecto fue diseñar e implementar un sistema de cómputo en la nube para el seguimiento y monitoreo de trayectorias con el uso de dispositivos móviles Android, el cual se pudo lograr satisfactoriamente. El uso de estas tecnologías tiene la ventaja de la recolección y monitoreo de datos en tiempo real.

Como primer punto se tiene el uso de una aplicación web para el monitoreo de los datos recolectados por los dispositivos móviles en un servidor propio al contrario de las aplicaciones de mapas ya existente donde dicha información no se puede consultar por los usuarios acorde a las necesidades del proyecto.

Como segundo punto el uso de dispositivos móviles Android para la recolección de los datos, ya que es más accesible el desarrollo de una app con el uso de las tecnologías actuales que construir un dispositivo físico para su recolección.

Como tercer punto se cuenta con un servidor web Rest para la transferencia de datos de los dispositivos móviles a la base de datos debido a que es más fácil la implementación y uso en comparación con otras tecnologías como SOAP.

Debido a los resultados obtenidos con el uso de las tecnologías antes mencionadas se tiene el beneficio de que los dispositivos móviles Android son más utilizados que los IOS y permite el monitoreo de más usuarios, también se observó que la actualización de las tecnologías no afecta el desempeño de las aplicaciones desarrolladas.

Como inconveniente del uso de la tecnología relacionada con los mapas se ve la necesidad de utilizar una versión de paga cuando la población objetivo crezca ya que se mandarían más peticiones a google maps.

# Conclusiones y prospectivas

---

## 5.1. Conclusiones

En la actualidad el desarrollo de nuevas tecnologías de información y comunicación está cambiando la esencia del turismo y esto da la oportunidad de crear nuevos conceptos y tecnologías en la industria del turismo.

A lo largo de este documento se presentó la implementación de una aplicación móvil, en la cual se utilizaron varias API's de Android basadas en la tecnología Java para la recolección de datos de geo localización GPS y preferencias de usuarios, la conexión a una base de datos remota utilizando un servidor web REST y el uso de listados y mapas en una página web desarrollada en Java con el patrón de diseño Modelo Vista Controlador.

La base de datos se desarrolló con la tecnología MySQL, la cual cuenta con 12 tablas donde se registraron todas las acciones realizadas por los usuarios de la aplicación Android. La aplicación web se utilizó para el análisis de la información recolectada por los dispositivos móviles y la administración de los catálogos del sistema tales como Usuarios y Edificios Históricos.

Se concluye que los objetivos planteados en este trabajo se alcanzaron debido al correcto funcionamiento de las aplicaciones gracias a los datos recolectado en la

base de datos y mostrados en la aplicación web de manera intuitiva o amigable al usuario.

## 5.2. Prospectivas

Debido al éxito obtenido en este trabajo, se observó la escalabilidad del proyecto al uso de nuevas tecnologías para aumentar la población objetivo del proyecto así como la eficiencia en el procesamiento de la información, a continuación se describen las tecnologías sugeridas:

- Aplicación para dispositivos móviles IOS

Existe la posibilidad de aumentar la población objetivo con la creación de una aplicación móvil para dispositivos IOS ya que son ampliamente utilizados, dicha aplicación se debe conectar con la base de datos desarrollada en este proyecto, mediante el uso del servidor web, manejando las mismas peticiones de la aplicación móvil Android para no tener la necesidad de desarrollar otro servidor web.

- Sistemas Big Data

El sistema de administración y procesamiento de datos demuestra ser funcional, pero con el uso de nuevas tecnologías como Big Data y machine learning puede optimizar el tiempo de procesamiento de la información ya que con el aumento de la población objetivo y el crecimiento exponencial de la información recolectada puede ser un problema.

- Aumento en la capacidad de los servidores

No solo el sistema de administración y procesamiento de datos se debe actualizar, sino también las características de los servidores físicos como la cantidad de memoria ram y la capacidad del procesador.

## Bibliografía

---

- [1] N. Jazdi, «Cyber physical systems in the context of Industry 4.0», en *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, 2014, pp. 1-4.
- [2] F. Shrouf, J. Ordieres, y G. Miragliotta, «Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm», en *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 2014, pp. 697-701.
- [3] C. E. A. Mulligan y M. Olsson, «Architectural implications of smart city business models: an evolutionary perspective», *IEEE Commun. Mag.*, vol. 51, n.º 6, pp. 80-85, jun. 2013.
- [4] J. Raun, R. Ahas, y M. Tiru, «Measuring tourism destinations using mobile tracking data», *Tour. Manag.*, vol. 57, pp. 202-212, dic. 2016.
- [5] M. A. Țîțu, A. S. Răulea, y Ș. Țîțu, «Measuring Service Quality in Tourism Industry», *13th Int. Symp. Manag. Manag. Econ. Crisis*, vol. 221, pp. 294-301, jun. 2016.
- [6] N. Shoval y M. Isaacson, «Tracking tourists in the digital age», *Ann. Tour. Res.*, vol. 34, n.º 1, pp. 141-159, 2007.
- [7] K. Su, J. Li, y H. Fu, «Smart city and the applications», en *2011 International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 1028-1031.
- [8] F. Liu *et al.*, «Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications», *IEEE Wirel. Commun.*, vol. 20, n.º 3, pp. 14-22, jun. 2013.
- [9] S. N. Srirama, M. Jarke, y W. Prinz, «Mobile Web Service Provisioning», en *Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06)*, 2006, pp. 120-120.
- [10] B. McKercher, N. Shoval, E. Ng, y A. Birenboim, «First and Repeat Visitor Behaviour: GPS Tracking and GIS Analysis in Hong Kong», *Tour. Geogr.*, vol. 14, n.º 1, pp. 147-161, feb. 2012.
- [11] R. W. Floyd, «The paradigms of programming», *Commun. ACM*, vol. 22, n.º 8, pp. 455-460, 1979.
- [12] R. E. Johnson y B. Foote, «Designing reusable classes», *J. Object-Oriented Program.*, vol. 1, n.º 2, pp. 22-35, 1988.
- [13] C. Gross, «Object-Oriented Programming Basics», *Begin. C 2008 Novice Prof.*, pp. 131-157, 2008.
- [14] J. Goossenaerts y J. Lewi, «Object-oriented programming: concepts», en *[Proceedings] COMPEURO 88 - System Design: Concepts, Methods and Tools*, 1988, pp. 2-8.
- [15] F. Durán, F. Gutiérrez, y E. Pimentel, *Programación orientada a objetos con Java*. Editorial Paraninfo, 2007.

- [16] T. Elrad, R. E. Filman, y A. Bader, «Aspect-oriented programming: Introduction», *Commun. ACM*, vol. 44, n.º 10, pp. 29-32, 2001.
- [17] S. B. Navathe y R. Elmasri, «Fundamentos de sistemas de bases de datos», *Addison Wesley*, 2002.
- [18] J. Sánchez, «Principios sobre bases de datos relacionales», *Creat. Commons Ira ED E. U.*, 2004.
- [19] P. Wendorff, «Assessment of design patterns during software reengineering: lessons learned from a large commercial project», en *Proceedings Fifth European Conference on Software Maintenance and Reengineering*, 2001, pp. 77-84.
- [20] J. Seemann y J. W. von Gudenberg, «Pattern-based design recovery of Java software», presentado en *ACM SIGSOFT Software Engineering Notes*, 1998, vol. 23, pp. 10-16.
- [21] G. E. Krasner y S. T. Pope, «A description of the model-view-controller user interface paradigm in the smalltalk-80 system», *J. Object Oriented Program.*, vol. 1, n.º 3, pp. 26-49, 1988.
- [22] C. Bauer y G. King, «Hibernate in action», 2005.
- [23] J. Arthur y S. Azadegan, «Spring framework for rapid open source J2EE Web application development: a case study», en *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*, 2005, pp. 90-95.
- [24] R. Johnson *et al.*, «The spring framework—reference documentation», *Interface*, vol. 21, 2004.
- [25] P. Mularien, *Spring Security 3*. Packt Publishing Ltd, 2010.
- [26] Y. Wang y J. Vassileva, «A Review on Trust and Reputation for Web Service Selection», en *Distributed Computing Systems Workshops, 2007. ICDCSW '07. 27th International Conference on*, 2007, pp. 25-25.
- [27] «What Are RESTful Web Services? - The Java EE 6 Tutorial». [En línea]. Disponible en: <http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>. [Accedido: 02-may-2017].
- [28] M. Bhatt, «SOAP vs REST».
- [29] M. Chen, S. Mao, y Y. Liu, «Big Data: A Survey», *Mob. Netw. Appl.*, vol. 19, n.º 2, pp. 171-209, 2014.
- [30] A. B. Patel, M. Birla, y U. Nair, «Addressing big data problem using Hadoop and Map Reduce», en *2012 Nirma University International Conference on Engineering (NUiCONE)*, 2012, pp. 1-5.
- [31] «Big Data», *Coursera*. [En línea]. Disponible en: <https://www.coursera.org/specializations/big-data>. [Accedido: 02-may-2017].
- [32] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, y S. Ullah Khan, «The rise of “big data” on cloud computing: Review and open research issues», *Inf. Syst.*, vol. 47, pp. 98-115, ene. 2015.
- [33] A. McAfee y E. Brynjolfsson, «Big data: the management revolution», *Harv. Bus. Rev.*, vol. 90, n.º 10, pp. 60-68, 2012.
- [34] A. Katal, M. Wazid, y R. H. Goudar, «Big data: Issues, challenges, tools and Good practices», en *2013 Sixth International Conference on Contemporary Computing (IC3)*, 2013, pp. 404-409.
- [35] J. Dittrich y J.-A. Quiané-Ruiz, «Efficient big data processing in Hadoop MapReduce», *Proc. VLDB Endow.*, vol. 5, n.º 12, pp. 2014-2015, 2012.

- [36] Q. Zhang, L. Cheng, y R. Boutaba, «Cloud computing: state-of-the-art and research challenges», *J. Internet Serv. Appl.*, vol. 1, n.º 1, pp. 7-18, 2010.
- [37] R. Buyya, J. Broberg, y A. M. Goscinski, *Cloud computing: Principles and paradigms*, vol. 87. John Wiley & Sons, 2010.
- [38] «Aprendizaje Automático», *Coursera*. [En línea]. Disponible en: <https://www.coursera.org/learn/machine-learning/home/welcome>. [Accedido: 02-may-2017].
- [39] I. G. Maglogiannis, *Emerging artificial intelligence applications in computer engineering: real word AI systems with applications in eHealth, HCI, information retrieval and pervasive technologies*, vol. 160. Ios Press, 2007.
- [40] J. M. Mendel, «Fuzzy logic systems for engineering: a tutorial», *Proc. IEEE*, vol. 83, n.º 3, pp. 345-377, mar. 1995.
- [41] D. Beasley, D. R. Bull, y R. R. Martin, «An overview of genetic algorithms: Part 1, fundamentals», *Univ. Comput.*, vol. 15, n.º 2, pp. 56-69, 1993.





# Currículum Vitae

---

**Jorge Armando Ortiz Murillo**

**Ingeniero en Computación**

Tulipanes # 80, Fracc. Gardenias Guadalupe, Zacatecas, México.

Móvil: (492) 145 07 38

E-mail: [ic.jortiz@gmail.com](mailto:ic.jortiz@gmail.com)

Skype: ic.jortiz

## Experiencia

Capacitacion de Usuarios	3 Años	Analisis y Diseño	4 Años
Documentacion	4 Años	Manejo de Bases de datos	4 Años
Soporte a Usuarios	2 Años	Desarrollo de software	6 Años

## Educación

Maestría en Ingeniería y Tecnología Aplicada	UAZ (Enero 2016- Diciembre 2017)
Ingeniería en Computación	UAZ (2005-2010)

## Experiencia profesional

Quarksoft (Agosto 2019 -Marzo 2020)

Puesto: Ingeniero de software

Desarrollo de aplicación web basada en la tecnología java para proyecto del gobierno de paraguay en el cual se realizaban las siguientes actividades:

- Uso de herramienta QSGen para la generación de código.
- Uso de maven para la compilación y deploy del proyecto, administración de librerías y ejecución de pruebas automáticas.

- Desarrollo en el lenguaje de programación java con el patrón de diseño MVC, en el cual se utilizaron los frameworks de spring, jpa, primefaces y servicios rest.
- Metodología de desarrollo agil.

DACOMP

(Febrero 2018 –Febrero 2019)

Posición: Ingeniero de software

Desarrollo de aplicación web basada en la tecnología appian para un cliente de estados unidos “fortekna”, en el cual se realizaban las siguientes actividades:

- levantamiento de requerimientos
- análisis y diseño de soluciones
- desarrollo en appian

DACOMP

(Septiembre 2014 –Mayo 2016)

Puesto: Ingeniero de software

Desarrollo de aplicación web basada en la tecnología java para un cliente de Estados Unidos “MEDCAT”, en el cual se realizaban las siguientes actividades:

- Levantamiento de requerimientos
- Análisis y diseño de solución
- Desarrollo en java con el uso de servlets, javascript, CSS y HTML
- Desarrollo de base de datos en postgres SQL y soporte a usuarios

Quarksoft

(Marzo 2013 – Abril 2014)

Puesto: Ingeniero de Procesos

Responsable del seguimiento de procesos institucionales por los equipos de desarrollo de software tales como:

- Monitoreo semanal del proyecto
- Monitoreo de lanzamientos TSP
- Auditorias de proyectos
- Revisor PSP
- Tomé los cursos CMMI, PSP y TSP

NeoTeck - INEGI

(Mayo 2012 – Junio 2012)

Puesto: Ingeniero de software

Desarrollador Java standard Edition, con el uso de una base de datos en SQL server con datos de geolocalización.

TI Consulting

(Febrero 2010 - Marzo 2012)

Puesto: Ingeniero de software

Desarrollo de aplicación web basada en la tecnología Java para la Secretaria de Finanzas, en el cual se realizaban las siguientes actividades:

- Levantamiento de requerimientos.
- Análisis y diseño de solución.
- Desarrollo en java con el uso de Hibernate, Spring y Richfaces.
- Desarrollo de base de datos en MY SQL y soporte a usuarios.

**Paquetes computacionales:**

Software	Lenguajes de Programación
<ul style="list-style-type: none"> <li>• Office</li> <li>• Eclipse</li> <li>• Spring Source</li> <li>• Sublime Text</li> <li>• Android Studio</li> <li>• NetBeans</li> <li>• IntelliJ IDEA</li> <li>• JCreator</li> <li>• Navicat</li> <li>• SQL Developer</li> <li>• MySQL Workbench</li> <li>• Tortoise SVN</li> <li>• GIT</li> <li>• ERStudio</li> <li>• Enterprise Arquitect</li> </ul>	<ul style="list-style-type: none"> <li>• Java - Intermedio                             <ul style="list-style-type: none"> <li>○ Standard Edition</li> <li>○ Enterprise Edition</li> <li>○ Micro Edition</li> </ul> </li> <li>• Java Frameworks &amp; Tools                             <ul style="list-style-type: none"> <li>○ Hibernate</li> <li>○ Spring</li> <li>○ Primefaces</li> <li>○ JSP</li> <li>○ Servlets</li> <li>○ JSF</li> </ul> </li> <li>• HTML</li> <li>• Ajax</li> <li>• Javascript</li> <li>• MySQL</li> <li>• PostgresSQL</li> <li>• XML</li> <li>• Maven</li> <li>• Jasper Reports</li> </ul>

**Certificaciones**

SCJP - Sun Certified Java Programmer - 2009

SCWCD - Sun Certified Web Component Developer - 2010

SCMAD - Certificación Sun para Aplicaciones Móviles - 2011

PSP - Certificación en PSP - 5 de Noviembre del 2012

ACP - Android Certified Professional - 3 de Diciembre del 2015

Inglés: Intermedio (TOEFL: 497).

**Cursos**

Hibérnate, Spring, JSF and RichFaces - TIconulting.

CMMI Dev and TSP- Quarksoft.

Big Data - UC San Diego Coursera