

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA EN SISTEMAS**

**ANÁLISIS ONTOLÓGICO DE INTERESES DEL USUARIO,  
REPRESENTADO EN UNA BASE DE DATOS APLICADO A UN CASO  
DE ESTUDIO**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**RIVERA ORMAZA CARLOS ANDRÉS**

**c.riveraormaza@hotmail.com**

**Directora: Ing. Regina Maritzol Tenemaza Vera MSc.**

**maritzol.tenemaza@epn.edu.ec**

**Quito, marzo del 2018**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Rivera Ormaza Carlos Andrés bajo mi supervisión.

A handwritten signature in blue ink, reading "Maritzol Tenemaza", is written over a horizontal line. The signature is enclosed in a light blue oval.

**Ing. Maritzol Tenemaza MSc.**

**DIRECTORA**

## DECLARACIÓN

Yo, Rivera Ormaza Carlos Andrés, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



**Rivera Ormaza Carlos Andrés**

## **AGRADECIMIENTOS**

Agradezco a las personas que me brindaron su apoyo incondicional.

## **DEDICATORIA**

A mi familia.

# ÍNDICE

<b>ÍNDICE.....</b>	<b>VI</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>X</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>XII</b>
<b>RESUMEN.....</b>	<b>XIII</b>
<b>1. CAPÍTULO 1.- INTRODUCCIÓN.....</b>	<b>14</b>
1.1. Situación actual.....	14
1.2. Descripción del problema.....	15
<b>2. CAPÍTULO 2.- METODOLOGÍA.....</b>	<b>16</b>
2.1. Aplicaciones móviles.....	16
2.2. Metodologías.....	16
2.2.1 Metodología de creación de una ontología.....	16
2.2.1.1 Ontología.....	17
2.2.1.2 Ontologías en ingeniería de sistemas.....	18
2.2.1.3 Metodología “Desarrollo de Ontologías 101”.....	19
2.2.2 Metodología de desarrollo.....	20
2.2.2.1 Metodología Scrum.....	20
2.3. Herramientas de desarrollo.....	23
2.3.1 Android Studio.....	23
2.3.2 XAMPP.....	24
2.3.3 MySql.....	25
2.3.4 PHP.....	26
2.3.5 Servidor Apache.....	27
2.3.6 Protégé.....	28

2.3.7	Google Maps .....	28
<b>3.</b>	<b>CAPÍTULO 3.- RESULTADOS Y DISCUSIÓN .....</b>	<b>30</b>
<b>3.1.</b>	<b>Solución planteada .....</b>	<b>30</b>
<b>3.2.</b>	<b>Alcance .....</b>	<b>30</b>
<b>3.3.</b>	<b>Visión del producto .....</b>	<b>31</b>
<b>3.4.</b>	<b>Esfuerzo en el desarrollo del prototipo.....</b>	<b>31</b>
<b>3.5.</b>	<b>Personal Y Roles Del Proyecto.....</b>	<b>31</b>
<b>3.6.</b>	<b>Creación de la ontología con la metodología “Desarrollo de ontologías 101” .....</b>	<b>32</b>
3.6.1.	Determinar el dominio y la cobertura de la ontología.....	32
3.6.2.	Considerar la reutilización de ontologías existentes. ....	33
3.6.2.1.	Ontología Ubisworld - Basic User Dimensions .....	33
3.6.2.2.	Ontología Ubisworld-location .....	34
3.6.3.	Enumerar los términos importantes de la ontología. ....	34
3.6.4.	Definir las clases y su jerarquía.....	35
3.6.5.	Definir las propiedades de las clases. ....	35
3.6.5.1.	Definición de propiedades de Objetos (“Object Properties”).....	36
3.6.6.	Definir las características de las propiedades. ....	37
3.6.6.1.	Definición de dominio y rango de las propiedades de Tipo de Datos. ....	38
<b>3.7.</b>	<b>Aplicación de la metodología de desarrollo del prototipo .....</b>	<b>40</b>
<b>3.7.1.</b>	<b>Requerimientos .....</b>	<b>40</b>
3.7.1.1.	Definición del Product Backlog .....	41
3.7.1.2.	Supuestos y dependencias.....	41
3.7.1.3.	Restricciones .....	42
<b>3.7.2.</b>	<b>Diseño de interfaces.....</b>	<b>43</b>
<b>3.8.</b>	<b>Arquitectura del prototipo .....</b>	<b>45</b>
<b>3.9.</b>	<b>Release Planning .....</b>	<b>47</b>

<b>3.10. Desarrollo de los Sprints .....</b>	<b>47</b>
3.10.1. Sprint 1 .....	47
3.10.1.1. Sprint Planning .....	47
3.10.1.2. Desarrollo del Sprint .....	48
3.10.1.3. Burn Down Chart.....	59
3.10.1.4. Sprint Review.....	59
3.10.1.5. Sprint Retrospective .....	60
3.10.2. Sprint 2 .....	60
3.10.2.1. Sprint Planning .....	60
3.10.2.2. Desarrollo del Sprint .....	61
3.10.2.3. Burn Down Chart.....	64
3.10.2.4. Sprint Review.....	65
3.10.2.5. Sprint Retrospective .....	66
3.10.3. Sprint 3 .....	66
3.10.3.1. Sprint Planning .....	66
3.10.3.2. Desarrollo del Sprint .....	68
3.10.3.3. Burn Down Chart.....	71
3.10.3.4. Sprint Review.....	72
3.10.3.5. Sprint Retrospective .....	74
3.10.4. Sprint 4 .....	74
3.10.4.1. Sprint Planning .....	74
3.10.4.2. Desarrollo del Sprint .....	75
3.10.4.3. Burn Down Chart.....	79
3.10.4.4. Sprint Review.....	80
3.10.4.5. Sprint Retrospective .....	81
<b>3.11. Ejemplo del uso de la aplicación .....</b>	<b>81</b>
<b>3.12. Análisis de resultados .....</b>	<b>87</b>
3.12.1. ¿El prototipo de aplicación móvil mostró restaurantes para ser visitados? .....	88
3.12.2. De acuerdo con su experiencia con la aplicación Polirestaurantes, ¿Qué tan necesario le parece el uso de estas aplicaciones? .....	88
3.12.3. ¿Qué tan fácil de usar le pareció el prototipo de aplicación móvil? .....	89
3.12.4. ¿La información obtenida en la aplicación fue clara al momento de encontrar el restaurante? .....	90

3.12.5. ¿La ruta que sugirió la aplicación fue correcta? .....	90
3.12.6. ¿Los restaurantes presentados en la aplicación estuvieron de acuerdo con sus intereses de búsqueda? .....	91
<b>CONCLUSIONES.....</b>	<b>92</b>
<b>RECOMENDACIONES.....</b>	<b>94</b>
<b>BIBLIOGRAFÍA.....</b>	<b>95</b>
<b>ANEXOS .....</b>	<b>99</b>
<b>Anexo I. Historias de usuario.....</b>	<b>99</b>
<b>Anexo II. Script creación de base de datos.....</b>	<b>102</b>
<b>Anexo III. Encuesta.....</b>	<b>108</b>

## ÍNDICE DE FIGURAS

FIGURA 1. UBISWORLD.ORG. [11]	19
FIGURA 2. SCRUM [10]	21
FIGURA 3. ANDROID STUDIO [16].	24
FIGURA 4. XAMPP [17].	25
FIGURA 5. MYSQL [18]	25
FIGURA 6. PHP [20]	27
FIGURA 7. APACHE [21]	27
FIGURA 8. PROTÉGÉ [22].	28
FIGURA 9. ONTOLOGÍA UBISWORLD-BASIC USER DIMENSIONS [11]	33
FIGURA 10. ONTOLOGÍA UBISWORLD-LOCATION [11]	34
FIGURA 11. DEFINICIÓN DE CLASES	35
FIGURA 12. DEFINICIÓN DE PROPIEDADES DE OBJETOS	36
FIGURA 13. PROPIEDADES INVERSAS.	36
FIGURA 14. DEFINICIÓN PROPIEDADES INVERSAS.	37
FIGURA 15. PROPIEDADES DE TIPO DE DATOS.	38
FIGURA 16. DOMINIO Y RANGO DE PROPIEDAD DE TIPO DE DATO.	39
FIGURA 17. DISEÑO DE ONTOLOGÍA	40
FIGURA 18. INTERFAZ POSICIÓN	43
FIGURA 19. INTERFAZ DETALLE RESTAURANTE	44
FIGURA 20. INTERFAZ MENÚ	44
FIGURA 21. INTERFAZ HISTORIAL	45
FIGURA 22. ARQUITECTURA DEL PROTOTIPO DE APLICACIÓN MÓVIL	46
FIGURA 23. TRANSFORMACIÓN ONTOLOGÍA [27]	49
FIGURA 24. TRANSFORMACIÓN DE LAS CLASES A TABLAS DE LA BASE DE DATOS [27]	51
FIGURA 25. TRANSFORMACIÓN DE LAS PROPIEDADES DE TIPO DE DATOS [27]	52
FIGURA 26. TRANSFORMACIÓN DE LAS PROPIEDADES DE OBJETOS EN RELACIONES DE LA BASE DE DATOS [27]	54
FIGURA 27. MODELO DE DATOS	56
FIGURA 28. CONEXIÓN.PHP	57
FIGURA 29. RESTAURANTES.PHP	58
FIGURA 30. BURN DOWN CHART SPRINT1	59
FIGURA 31. CREACIÓN PROYECTO EN ANDROID STUDIO	62

FIGURA 32. CREACIÓN DE LA ACTIVIDAD MAIN ACTIVITY	63
FIGURA 33. CREACIÓN LAYOUT ACTIVITY_MAIN.XML	63
FIGURA 34. CREACIÓN ACTIVIDAD INICIO	64
FIGURA 35. BURN DOWN CHART SPRINT 2	65
FIGURA 36. CREACIÓN DE LAYOUT POSICIÓN.XML	68
FIGURA 37. CREACIÓN ACTIVIDAD POSICIÓN.	69
FIGURA 38. ACTIVIDAD DETALLE RESTAURANTE.	70
FIGURA 39. LAYOUT DETALLE RESTAURANTE.	71
FIGURA 40. BURN DOWN CHART SPRINT 3	72
FIGURA 41. LOCATIONLISTENER EN LA ACTIVIDAD DETALLE RESTAURANTE.	75
FIGURA 42. ACTIVIDAD BÚSQUEDA	76
FIGURA 43. LAYOUT BÚSQUEDA.	77
FIGURA 44. ACTIVIDAD HISTORIAL.	78
FIGURA 45. LAYOUT HISTORIAL	79
FIGURA 46. BURN DOWN CHART SPRING 4.	80
FIGURA 47. MAPA INICIAL POLIRESTAURANTES.	82
FIGURA 48. DETALLE RESTAURANTE.	83
FIGURA 49. MENÚ DEL PROTOTIPO MÓVIL.	84
FIGURA 50. BÚSQUEDA POR NOMBRE.	85
FIGURA 51. BÚSQUEDA POR CALIFICACIÓN	86
FIGURA 52. BÚSQUEDA POR TIPO DE COMIDA.	86
FIGURA 53. HISTORIAL.	87
FIGURA 54. RESULTADO PREGUNTA 1	88
FIGURA 55. RESULTADO PREGUNTA 2	89
FIGURA 56. RESULTADO PREGUNTA 3	89
FIGURA 57. RESULTADO PREGUNTA4	90
FIGURA 58. RESULTADO PREGUNTA 5	91
FIGURA 59. RESULTADO PREGUNTA 6	92

## ÍNDICE DE TABLAS

TABLA 1. ROLES DE SCRUM [14]	22
TABLA 2. CARACTERÍSTICAS PRINCIPALES [19]	26
TABLA 3. PERSONAL Y ROLES DEL PROYECTO	31
TABLA 4. PRODUCT BACKLOG INICIAL	41
TABLA 5. RELEASE PLANING	47
TABLA 6. SPRINT 1	47
TABLA 7. TABLAS RESULTANTES	50
TABLA 8. COLUMNAS EN LAS TABLAS DE LA BASE DE DATOS RELACIONAL.	53
TABLA 9. CRITERIOS DE ACEPTACIÓN SPRINT 1	60
TABLA 10. SPRINT 2	61
TABLA 11. CRITERIOS DE ACEPTACIÓN SPRINT 2	65
TABLA 12. SPRINT 3	67
TABLA 13. CRITERIOS DE ACEPTACIÓN SPRINT 3	73
TABLA 16. SPRINT 4	74
TABLA 15. CRITERIOS DE ACEPTACIÓN SPRINT 4	81

## RESUMEN

En el presente documento se exponen los pasos para el desarrollo de un prototipo móvil en sistemas nativos Android, para la geolocalización outdoor de restaurantes cercanos a la Escuela Politécnica Nacional.

En el primer capítulo, se detalla la situación actual del uso de dispositivos móviles en los últimos años, y la manera en cómo se ha incrementado el manejo de estos. En cuanto a la problemática situada en el presente trabajo de titulación, se sugiere un prototipo de aplicación que presente resultados en base a los intereses del usuario.

En el segundo capítulo se plantea las metodologías a aplicar, tanto para el desarrollo del prototipo, como para la construcción de la ontología. Además, que se describe a las herramientas con las que se construirá el prototipo y la ontología. También en dicho capítulo, se aborda el tema del uso de ontologías en las aplicaciones móviles para tales dispositivos.

En el tercer capítulo se aplica las metodologías para el desarrollo, se define la ontología y con esta se crea la base de datos que almacena la información referente a los restaurantes y usuarios. Se planifica el desarrollo del prototipo bajo la metodología elegida y se ejecutan las pruebas correspondientes.

En el cuarto capítulo se expone un ejemplo del uso de la aplicación, se diseña una encuesta la cual es realizada a un número de clientes. También se tiene el análisis de las respuestas, como la explicación de los resultados obtenidos.

# **1. CAPÍTULO 1.- INTRODUCCIÓN**

## **1.1. Situación actual**

El avance actual de la tecnología, permite desarrollar nuevas alternativas enfocadas a las necesidades y requerimientos del usuario, a través, de la implementación de aplicaciones desarrolladas para dispositivos móviles, pues, cada vez son más los usuarios quienes realizan actividades cotidianas, a través de dispositivos inteligentes denominados Smartphone. En este contexto se debe afrontar el reto de generar aplicaciones capaces de desplegarse en sistemas móviles, que respondan a los intereses del usuario.

A partir de la década de los 90, el estudio de la adaptabilidad ha sido un objetivo prioritario en el desarrollo de sistemas [1]. El tema de mayor importancia para lograr la adaptabilidad ha sido el estudio del usuario [1]. Hoy en día los sistemas web han logrado un avance significativo en lo que respecta a “adaptabilidad” al receptor final. Empresas como Google, Amazon o las redes sociales Facebook, Twitter entre otros, han desarrollado tecnologías que se adaptan a la medida y requerimientos del consumidor. Actualmente el desarrollo de aplicaciones para dispositivos móviles constituye un reto debido a la adopción de estas tecnologías. Los dispositivos inteligentes que los usuarios tienen a su disposición permiten una mayor movilidad, obtención información en tema de localización, y también acceso a sensores, con el fin de generar datos desde el usuario.

En Ecuador, el uso de los dispositivos móviles ha crecido considerablemente en los últimos años. Según datos del INEC [2], el 16,9% (1'261.944) de personas con edades de entre cinco años o más, de la población ecuatoriana que tienen celular, poseen un móvil inteligente. El principal propósito de uno de estos dispositivos es una conexión a internet, más cómoda y sin la necesidad de la utilización de un computador, pues, no se requiere de toda su capacidad de procesamiento.

## **1.2. Descripción del problema**

La generación del software móvil no adaptado a intereses del usuario genera sistemas muy básicos. No considerar la adaptabilidad provoca pérdidas de oportunidades en negocios específicos.

Otro punto que considerar es que mientras los sistemas web tienen grandes avances como bases globales de conocimiento, buscadores semánticos, entre otros. Los sistemas móviles han quedado rezagados, dejando a un lado todas las ventajas, que pueden ser aprovechadas al usar ontologías en el desarrollo de software.

## **2. CAPÍTULO 2.- METODOLOGÍA**

### **2.1. Aplicaciones móviles**

Una aplicación móvil es un conjunto de funcionalidades agrupadas con un fin en específico, y de acuerdo con los requerimientos de un grupo de usuarios. Dichas aplicaciones se generan mediante una compilación de instrucciones lógicas, que permiten interrelacionar información obtenida de diferentes fuentes como: servidores de terceros, redes sociales, bases de datos internas del dispositivo, sensores, usuarios entre otros.

A partir del avance de la tecnología, se han desarrollado múltiples aplicaciones móviles, que facilitan las necesidades que el ser humano mantiene en la actualidad, tanto en ámbitos, educativos, económicos, de salud, turismo, información, localización entre otros.

La inmediatez de acceso a la información permite a dichas aplicaciones tener una ventaja en la portabilidad sobre las aplicaciones tradicionales. La presentación de información en tiempo real generadas por dichas aplicaciones facilita al usuario la satisfacción de sus requerimientos.

Por tal motivo el presente trabajo de titulación tiene como objetivo generar un prototipo de aplicación móvil, que permita a los usuarios su vinculación con restaurantes cercanos a la Escuela Politécnica Nacional.

### **2.2. Metodologías**

#### **2.2.1 Metodología de creación de una ontología**

La creación de una ontología es un proceso muy parecido al desarrollo de software y está compuesto por una serie ordenada de actividades [3]. No existe una metodología “correcta” para el desarrollo o creación de una ontología [4], por lo que se trata de abordar los puntos que se consideran más importantes y se plantean procedimientos para la creación.

### 2.2.1.1 Ontología

La semántica de una aplicación web utiliza ontologías para ordenar, consensuar y representar el conocimiento, haciendo de esta manera que pueda ser interpretado por los navegadores. La palabra ontología hace referencia al estudio del ser, ya que proviene de las raíces griegas, οντος, ontos, que significa ser, ente, y λόγος, logos, que significa estudio, discurso, ciencia, teoría. Las ontologías se aplican en inteligencia artificial como también en la web semántica, para asimilar, clasificar y codificar el conocimiento, estableciendo las relaciones existentes en un dominio de conocimiento [5]. Las ontologías en ingeniería de software también forman un papel importante, ya que pueden ser aplicadas en todos los ciclos de vida del producto software. Es evidente que, al disponer de una ontología, o los procesos para su diseño y construcción, ayuda sobre manera, a disminuir el porcentaje de errores en el desarrollo durante todas las etapas del ciclo de vida de los productos software [6].

La importancia de las ontologías está distribuida en dos partes [7].

- El análisis ontológico clarifica la estructura del conocimiento. Dado un dominio, su ontología constituye el corazón de cualquier representación del conocimiento. Sin ontologías o sin las respectivas conceptualizaciones no puede existir un vocabulario que soporte el conocimiento. Así el primer paso para obtener un efectivo sistema de representación del conocimiento y su vocabulario es llevar a cabo un efectivo análisis ontológico del campo o dominio de estudio, un análisis pobre conlleva a bases de conocimiento incoherentes.
- Las ontologías permiten compartir conocimiento. Es posible compartir un lenguaje de representación de conocimiento con otros que tengan similares necesidades para el mismo dominio, de tal modo que se elimine la necesidad de replicar el proceso de análisis del conocimiento.

Las principales ventajas de las ontologías son [8]:

- Compartir el entendimiento común de la estructura de la información con distintas personas o agentes de software.
- Permitir la reutilización del conocimiento de dominio.

- Hacer que los supuestos del dominio sean explícitos.
- Separar el conocimiento del dominio, del conocimiento operacional.

### 2.2.1.2 Ontologías en ingeniería de sistemas

El uso de ontologías en ingeniería de sistemas plantea los siguientes propósitos [9]:

- Especificación.** - El rol que las ontologías juegan en la especificación depende del grado de formalidad y automatización dentro de la metodología de diseño del sistema. Desde un enfoque informal, las ontologías facilitan el proceso de identificación de requerimientos y la comprensión de las relaciones entre componentes. Esto es particularmente importante, cuando existen conjuntos de diseñadores distribuidos trabajando sobre diferentes dominios. Desde un enfoque formal, una ontología provee una especificación declarativa de un sistema, la cual permite a los diseñadores razonar sobre “para qué” se está diseñando el sistema en lugar de “cómo” soportar la funcionalidad.
- Confiabilidad.** - Las ontologías informales pueden mejorar la confiabilidad del sistema sirviendo como base para el chequeo manual del diseño contra la especificación, mientras que las ontologías formales permiten el chequeo de consistencia (semi) automatizado del sistema de software con respecto a la especificación declarativa
- Reusabilidad.** - Para ser eficiente, una ontología debe soportar que se puedan importar y exportar módulos entre diferentes sistemas de software. Por medio de la caracterización de las clases de dominio, y, de las tareas dentro de esos dominios, las ontologías pueden proveer un marco de trabajo, para determinar qué aspectos de una ontología pueden ser reutilizados entre dominios y tareas diferentes. La meta es conseguir librerías de ontologías que puedan ser reutilizadas y adaptadas para distintas clases de problemas y entornos.

En Ingeniería de software, se usa ontologías en distintos niveles de generalidad. Como ejemplo las ontologías a nivel de dominio, las cuales son especialmente útiles para el



Hace referencia a la búsqueda de ontologías ya desarrolladas, existentes en diferentes repositorios, reutilizar dominios o parte de ellos para la creación de una ontología a la medida.

3. Enumerar los términos importantes de la ontología.

Busca conceptos relevantes en el ámbito de conocimiento sobre el cual se aplica la ontología, y los convierte en términos importantes dentro de ella.

4. Definir las clases y su jerarquía.

Con los términos importantes se definen las clases y el orden en el que están las mismas.

5. Definir las propiedades de las clases.

Con las clases ya encontradas y en la jerarquía necesaria, se debe definir en cada clase las propiedades que esta necesita.

6. Definir las características de las propiedades.

Con las propiedades definidas se debe definir el dominio y rango de estas.

7. Crear instancias (de ser necesario).

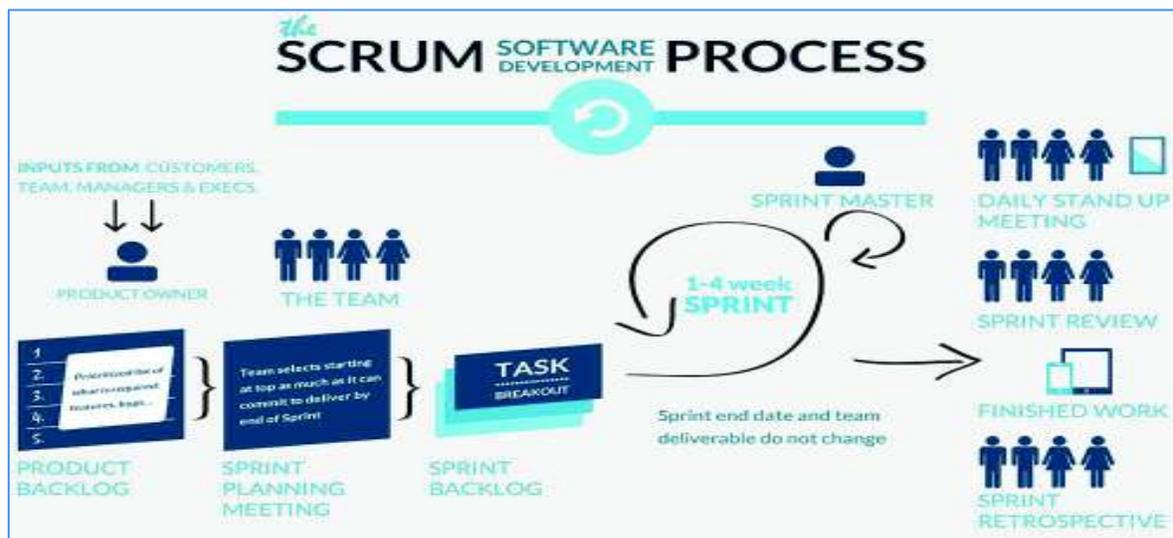
De ser necesario crear instancias de cada clase.

## **2.2.2 Metodología de desarrollo**

Para el desarrollo del prototipo se ha decidido usar el marco SCRUM. Para llevar a cabo esta tarea de ingeniería, se han desarrollado métodos, técnicas y procedimientos que se han convertido en las denominadas metodologías de desarrollo de software [13].

### **2.2.2.1 Metodología Scrum**

La metodología de desarrollo de software Scrum, de la cual se observa una descripción en la Figura 2, es un marco de referencia que nace a partir de un modelo de desarrollo de productos tecnológicos que fue consolidándose en una metodología ágil de desarrollo de proyectos de software que se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; además se caracteriza por ser una metodología de desarrollo simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan sino en la adaptación continua a las circunstancias de la evolución del proyecto [14].



**Figura 2. Scrum [10]**

**Artefactos de Scrum:** Los artefactos de Scrum, están diseñados para un mejor manejo de la información vital del proyecto, entre los diferentes roles que existen dentro de esto puede evidenciar:

- **Product Backlog:** O lista de producto, es una lista en la que se especifican todos los requisitos del producto.
- **Sprint Backlog:** O lista de pendientes de un Sprint. Es un listado de algunos requisitos de la lista de producto, que han sido seleccionados por el Scrum Team, los cuales van a ser desarrollados en el Sprint. [15]

**Roles de Scrum:** En esta metodología existen varios roles, los cuales desempeñan distintas funciones dentro del desarrollo, mencionados en la Tabla 1:

**Tabla 1. Roles de Scrum [14]**

<b>Rol</b>	<b>Descripción</b>
Scrum Master	Es el responsable de mantener los procesos y trabaja de forma similar al director del proyecto. No es el líder, sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga.
Product Owner	Representa la voz del cliente; es el encargado de presentar a los stake holders (grupos de interés externos o internos).
Team	Es el rol que incluye a los desarrolladores. Los que tiene la responsabilidad de entregar el producto.

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software cuyo principal objetivo es maximizar el retorno de la inversión para la empresa. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación. Permite desarrollar una solución tecnológica de calidad y conforme a los requerimientos y necesidades de los clientes con la finalidad de tener un producto de software acorde a las necesidades de la organización y cliente final [14].

La razón fundamental de la utilización de la metodología ágil Scrum es el ciclo interactivo e incremental para el desarrollo del prototipo de aplicación móvil, que se describe a continuación.

- Sistema modular. El prototipo de aplicación móvil Polirestaurantes permite desarrollar una base funcional mínima y sobre esta incrementar las funcionalidades del proyecto.
- Entrega frecuente de módulos terminados.

- Previsible cambio en los requisitos. Cambios en la especificación y definición de los requisitos iniciales.
- En la ejecución del proyecto puede haber prioridades y cambios en la entrega de los módulos.

**Valores de trabajo:** Los valores en el desarrollo de la aplicación móvil, que se debe practicar entre el grupo de trabajo bajo la metodología Scrum, y, para que esta tenga éxito son:

- Autonomía del equipo.
- Respeto con el equipo de trabajo.
- Responsabilidad.
- Foco en una tarea.
- Transparencia.

## **2.3. Herramientas de desarrollo**

Para el desarrollo del prototipo de aplicación móvil, que permita la gestión de la información que requiere una representación ontológica de los intereses del usuario, se necesitan las herramientas de desarrollo que se describen a continuación.

### **2.3.1 Android Studio**

Es una herramienta de desarrollo para las aplicaciones bajo la plataforma Android, desarrollada por Google Inc., como un entorno de desarrollo integrado que ha madurado desde la primera versión estable, lanzada en diciembre del 2014. Esta herramienta reemplaza a Eclipse como IDE oficial de desarrollo para las aplicaciones Android y se puede observar su ícono en la Figura 3.



**Figura 3. Android Studio [16].**

Entre las características de Android Studio tenemos las siguientes [16]:

- Renderización en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.

Esta herramienta se encuentra disponible tanto para sistemas Windows, como Linux y Mac OS X, los requerimientos mínimos para dichos sistemas se los puede revisar en el sitio oficial, el cual se encuentra en la siguiente dirección electrónica: <https://developer.android.com/sdk/index.html> [16].

### **2.3.2 XAMPP**

Esta herramienta es un gestor de aplicaciones que está compuesta de un motor de base de datos, servidor web y dos intérpretes de lenguajes PHP y Pearl. El motor de base de datos es MariaDB, el cual es administrado desde la consola Web PhpMyAdmin. Como servidor web utiliza Apache las versiones de los mismos dependen de la fecha en la que se consiga dicha herramienta [17], y podemos observar su ícono en la Figura 4.



**Figura 4. Xampp [17].**

XAMPP es un servidor independiente de plataforma de código libre, que, facilita la instalación sencilla de un servicio de aplicaciones Apache, dentro del ordenador indistintamente de la plataforma que se disponga. Además, incluye servidores de datos como MySQL y SQLite administrado mediante phpMyAdmin y phpSQLiteAdmin. Incorpora también un lenguaje intérprete de código PHP, así como un intérprete de Perl, servidores de FTP como ProFTPD o FileZilla FTP Server, entre otras aplicaciones que facilitan la construcción de aplicaciones.

### **2.3.3 MySQL**

Gestor de base de datos basado en servicios y con una estructura relacional entre entidades (tablas), que facilita el almacenamiento de información relacionada de forma dinámica. Es la base de datos de código abierto ampliamente extendida en todo el mundo, y permite la oferta económica de aplicaciones de bases de datos fiables, de alto rendimiento, y fácilmente ampliables, basadas en la web e integradas. Esta aplicación es muy utilizada por su velocidad en la lectura, pero, puede tener errores en sistemas de alta concurrencia, por esto es muy utilizado en sistemas web, los cuales son de baja concurrencia, como Joomla, Wordpress, entre otros [18]. Observamos su ícono en la Figura 5.



**Figura 5. MySQL [18]**

Este será el motor de base de datos a utilizarse para el almacenamiento de la información tanto de los usuarios, como de los restaurantes. Las características principales se observan en la Tabla 2.

**Tabla 2. Características principales [19]**

<b>Característica</b>	<b>Descripción</b>
Multiplataforma	Se puede utilizar en diferentes tipos de sistemas operativos.
Multitarea	Puede atender múltiples peticiones
Sensible a la cantidad de memoria RAM	Una cantidad limitada de memoria RAM puede afectar el rendimiento.
SQL	Las peticiones se realizan mediante SQL.
Capacidad límite de tablas	El tamaño límite por tabla es de más de 8 millones de terabytes
Entidad – Relación	Permite establecer relaciones entre las diferentes tablas a fin de relacionar la información

### **2.3.4 PHP**

Es un software de codificación de aplicaciones web dinámicas, cuyo ícono se observa en la Figura 6, usa una mezcla entre interpretación y compilación, que ofrece a los programadores la mejor herramienta de rendimiento y flexibilidad para la construcción de aplicaciones web.

*“Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga”* [20]



**Figura 6. Php [20]**

PHP compila el código mediante una serie de instrucciones (llamadas opcodes), siempre que estas sean accedidas mediante instrucciones ejecutadas una a una, hasta que el script termina permitiendo el desarrollo limpio de aplicaciones web.

### **2.3.5 Servidor Apache**

Es un programa que implementa el protocolo HTTP (hypertext transfer protocol), está diseñado para transferir hipertextos a páginas web o HTML (hypertext markup language), además de textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos a aplicación web [21]. Se puede observar su ícono en la Figura 7.



**Figura 7. Apache [21]**

Esta herramienta es un servidor web HTTP el cual ha sido desarrollado de forma que funcione sobre distintas plataformas, como Windows, Macintosh y Linux. Apache es una popular y

eficiente alternativa que ofrece servicios web. Este web server es uno de los logros más grandes del software libre y la punta de lanza del mundo de las páginas web. La funcionalidad de este servidor es modular, es decir, consta de un *Core* o núcleo, y la mayor parte de su funcionalidad se implementa a través de módulos. [21]. Sobre este servidor web se expondrán a la red los webs services.

### 2.3.6 Protégé

Plataforma libre, de código abierto que ofrece una comunidad de usuarios cada vez mayor con un conjunto de herramientas para la construcción de modelos de dominio y las aplicaciones basadas en el conocimiento con ontologías [22], se observa su ícono en la Figura 8.



**Figura 8. Protégé [22].**

Esta herramienta se utilizará para la edición y construcción de la ontología a usarse para el diseño y construcción de la base de datos que se encargará del almacenamiento de la información que requiere el prototipo móvil.

### 2.3.7 Google Maps

Es una API que provee interfaces de localización, que ayudan a graficar mapas personalizados, y amigables tanto para los desarrolladores como para los usuarios. La empresa que brinda este servicio de mapas es Google Inc.

Los beneficios que la empresa Google Inc. presta a los usuarios que requieren su producto son variados pues, son efectivos y amigables en ámbitos de implementación, visualidad y de uso [16].

Los mapas de Google permiten geolocalizar los puntos de interés para una rápida referencia de su ubicación y la ubicación del usuario [23].

## **3. CAPÍTULO 3.- RESULTADOS Y DISCUSIÓN**

### **3.1. Solución planteada**

Este proyecto busca generar un prototipo de aplicación móvil, el cual permita analizar los intereses del usuario en base a la aplicación de una ontología, y estos intereses se vean reflejados en los datos mostrados por el prototipo; ofertando restaurantes de acuerdo con el tipo de alimentación preferida por el consumidor.

Las características del prototipo de aplicación móvil son:

- El prototipo se desarrollará bajo la aplicación de la metodología Scrum.
- La ontología Ubisworld será la base para la adaptación de la ontología a usarse en este proyecto.
- Los datos de los restaurantes serán recogidos en un radio de 500 metros alrededor de la Escuela Politécnica Nacional.

### **3.2. Alcance**

El prototipo de aplicación móvil analizará intereses del usuario y propondrá restaurantes alrededor de la EPN en un radio de búsqueda de 500 m.

Los requisitos funcionales comprenden el servicio que el software proveerá a los usuarios, de tal forma que permita la gestión de información de restaurantes. El proceso que infieren en la automatización comprende los siguientes aspectos y actividades:

- Geolocalización de los restaurantes en el mapa.
- Búsquedas de restaurantes por categoría (Calificación restaurante, tipo de comida).
- Historial de búsqueda de los usuarios.

El levantamiento de los requisitos funcionales, y no funcionales necesarios para el desarrollo del prototipo de aplicación móvil, se ha definido entre el desarrollador del producto de software y el cliente, que se han traducido a historias de usuario en base a las necesidades e interacciones de los mismos.

### 3.3. Visión del producto

Ser un prototipo de aplicación móvil que facilite la localización de restaurantes acorde a los gustos del usuario, en un radio de 500 metros alrededor de la Escuela Politécnica Nacional. Brindar una ruta hasta el restaurante.

### 3.4. Esfuerzo en el desarrollo del prototipo

Para el desarrollo del prototipo se tienen las siguientes consideraciones:

- Se realizarán jornadas de 4 horas al día
- Se trabajarán 5 días a la semana
- Existe solo un desarrollador

### 3.5. Personal Y Roles Del Proyecto

Las personas que trabajarán en este proyecto se listan en la Tabla 3 con los roles a desempeñar.

**Tabla 3. Personal Y Roles Del Proyecto**

NOMBRE	DESCRIPCIÓN	RESPONSABILIDAD
Director Proyecto	<b>Ing. Maritzol Tenemaza MSc.</b> <b>maritzol.tenemaza@epn.edu.ec</b>	Scrum Manager
Responsable Desarrollo	<b>Rivera Ormaza Carlos Andrés</b> <b>c.riveraormaza@hotmail.com</b>	Scrum Team

### **3.6. Creación de la ontología con la metodología “*Desarrollo de ontologías 101*”**

Para la creación de la ontología se utiliza la metodología descrita en el punto 2.2.1.3. Esta propone un proceso sistemático desarrollado a continuación:

#### **3.6.1. Determinar el dominio y la cobertura de la ontología.**

Para determinar el dominio y la cobertura de la ontología a construir se debe responder las siguientes preguntas [12]:

¿Qué dominio cubrirá la ontología?

Se quiere desarrollar una aplicación que presente restaurantes de acuerdo con los gustos e interés del usuario.

¿Para qué usaremos la ontología?

La ontología deberá mostrar los distintos restaurantes, visualizando primero aquellos que responden al interés del usuario.

¿A qué tipos de preguntas ha de responder la ontología?

La ontología debe conocer los detalles de dichos restaurantes como, por ejemplo: nombre, dirección, posición geográfica, horarios, entre otros.

- ¿Qué restaurantes se encuentran cercanos a la Escuela Politécnica Nacional?
- ¿Dónde se encuentran ubicados dichos restaurantes?
- ¿Qué información de dichos restaurantes puedo conocer?
- ¿Cuáles de estos restaurantes son los preferidos por el usuario?

- ¿Cuáles restaurantes son los favoritos para el usuario?
- ¿Qué tipo de comida ofrece el restaurante?

De estas preguntas se deduce que, la referencia geográfica principal es la Escuela Politécnica Nacional. Así como también es necesario conocer la localización de dichos restaurantes. Adicionalmente se requiere la información sobre el tipo de comida que ofrece cada restaurante.

### 3.6.2. Considerar la reutilización de ontologías existentes.

Como bien lo menciona [12], las ontologías se construyen con el fin de compartir conocimiento, no es necesario construir algo que ya está hecho. Por lo que se procede con la búsqueda de ontologías que se adapten a las necesidades, y ayuden a responder las preguntas planteadas en el punto 3.6.1. Se ha realizado la búsqueda con ayuda de las herramientas Swoogle y Ubisworld.

#### 3.6.2.1. Ontología Ubisworld - Basic User Dimensions

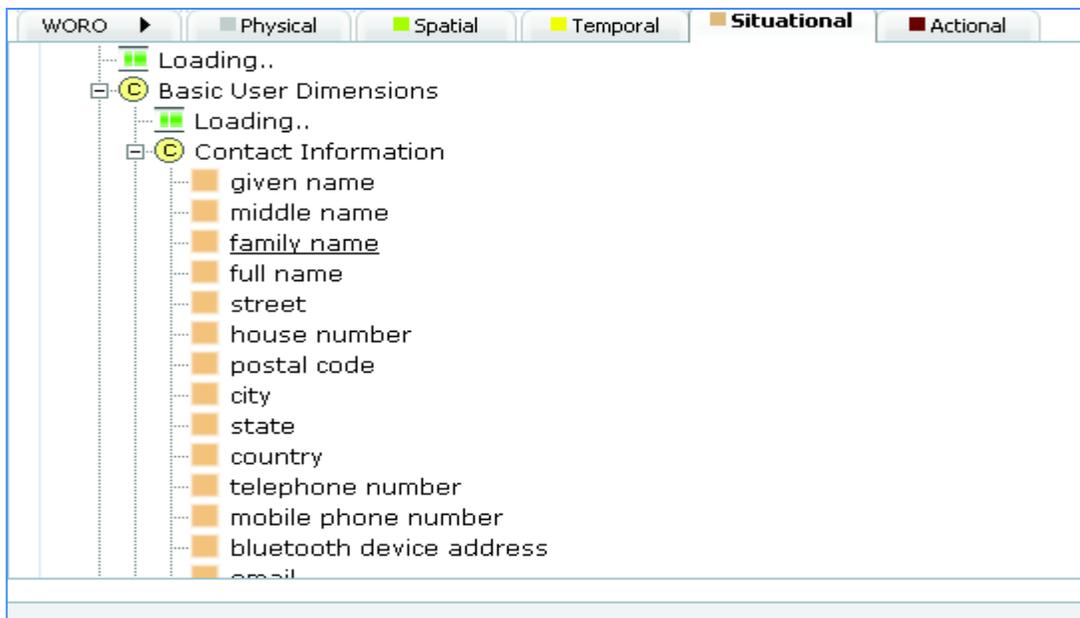


Figura 9. Ontología Ubisworld-Basic User Dimensions [11]

La Figura 9 muestra la ontología Ubisworld-Basic User Dimensions la cual presenta todos los datos necesarios para un usuario en un contexto general. Se escogerán las propiedades necesarias para resolver el problema, respetando la descripción de la ontología.

### 3.6.2.2. Ontología Ubisworld-location

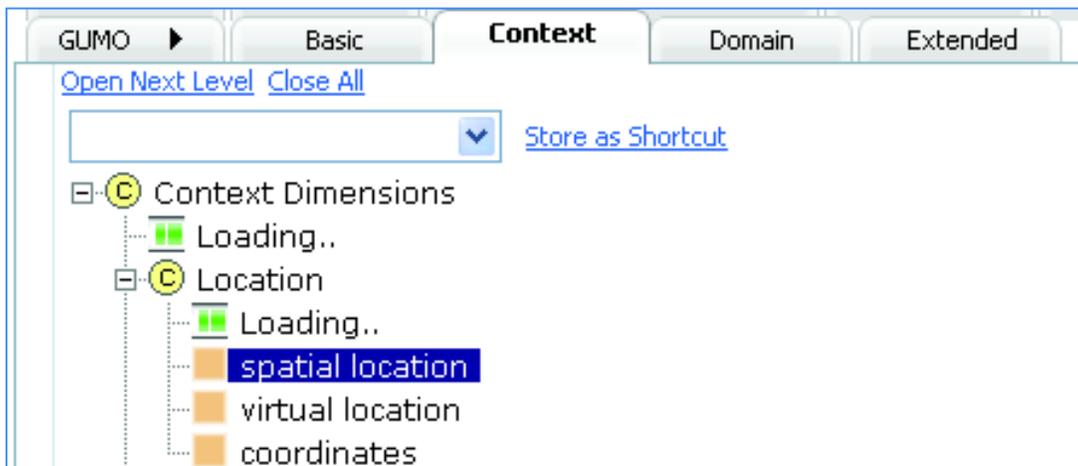


Figura 10. Ontología Ubisworld-location [11]

La ontología que se observa en la Figura 10 posee el campo “*coordinates*”, usado en este trabajo para la geolocalización de los restaurantes [11].

### 3.6.3. Enumerar los términos importantes de la ontología.

Se listan todos los términos del dominio de la ontología.

En el caso del proyecto se tienen los siguientes términos:

- ✓ Restaurantes
- ✓ Comida
- ✓ Tipo de comida
- ✓ Usuario

- ✓ Horario
- ✓ Localización
- ✓ Visitas
- ✓ Platos
- ✓ Favoritos

#### 3.6.4. Definir las clases y su jerarquía.

En ontologías, una clase es la representación de un concepto, son grupos abstractos, conjuntos o colecciones de objetos. Los conceptos en la ontología deben estar cerca de objetos (físicos o lógicos) y relaciones en el dominio de interés o de estudio. [24]. Según las preguntas planteadas en puntos anteriores, se han identificado las clases que se observan a continuación. Como se observa en la Figura 11, se creó la clase Things y dentro de ella las clases definidas.



Figura 11. Definición de Clases

#### 3.6.5. Definir las propiedades de las clases.

### 3.6.5.1. Definición de propiedades de Objetos (“Object Properties”)

Las propiedades de objetos que se han identificado se observan en la Figura 12.

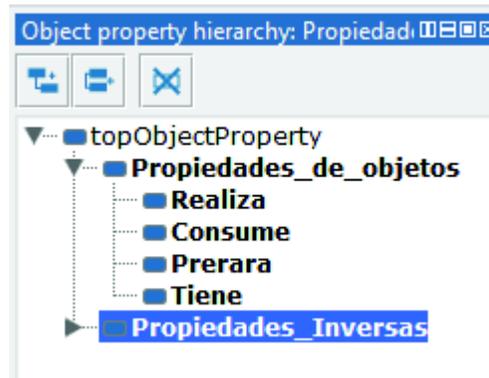


Figura 12. Definición de propiedades de Objetos

Uno de los conceptos principales de las ontologías son las propiedades inversas. Si una propiedad une o enlaza dos objetos, como A y B, la propiedad inversa debe unir B con A. Las propiedades inversas que se definen se observan en la Figura 13.

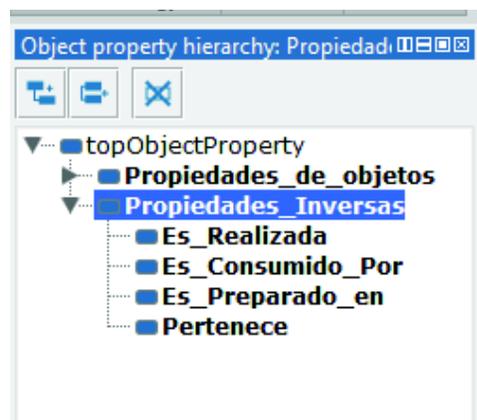


Figura 13. Propiedades Inversas.

Se define qué propiedad es inversa de otra propiedad. Un ejemplo de esto se realiza con la propiedad inversa “Es Realizado”, esto se observa en la Figura 14.



Figura 14. Definición propiedades inversas.

### 3.6.6. Definir las características de las propiedades.

Se ha definido las propiedades de datos que se observan en la Figura 15.



Figura 15. Propiedades de Tipo de Datos.

#### 3.6.6.1. Definición de dominio y rango de las propiedades de Tipo de Datos.

El dominio en los tipos de datos es el que utiliza el tipo de dato, mientras que el rango muestra la clase de tipo de dato. Por ejemplo, el tipo de dato “Nombre” es de tipo “Name” o “String” y está asociado a las clases Persona, Usuario, Tipo\_de\_Comida y Restaurante. Esto se puede apreciar en la Figura 16.

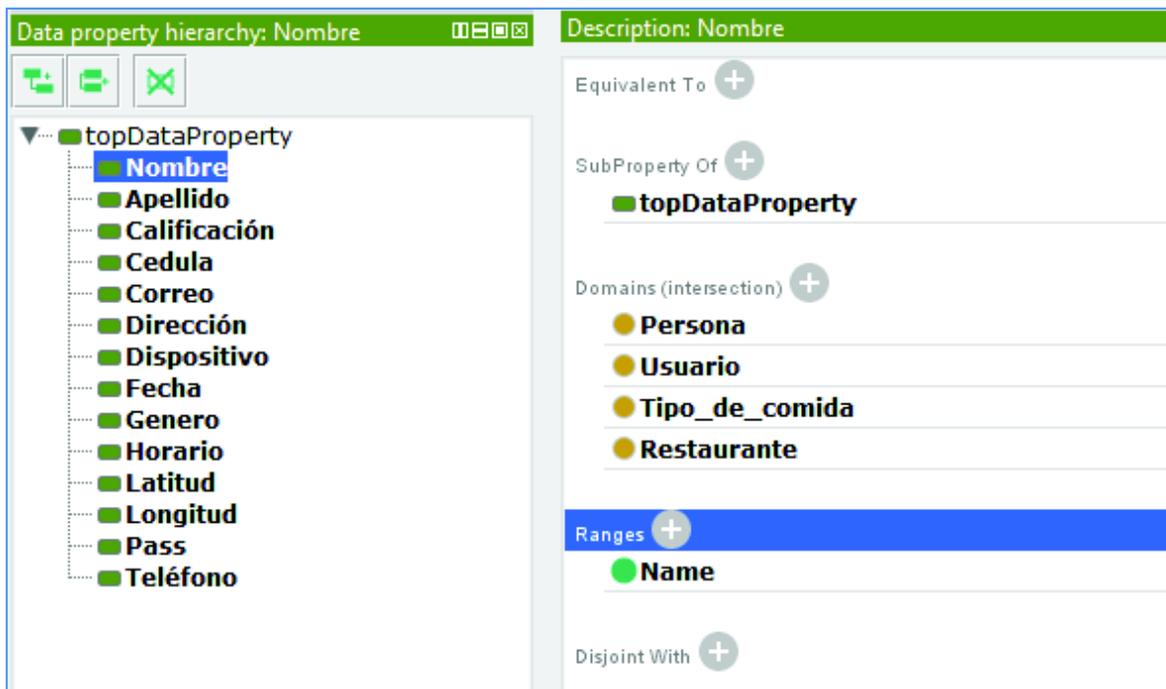
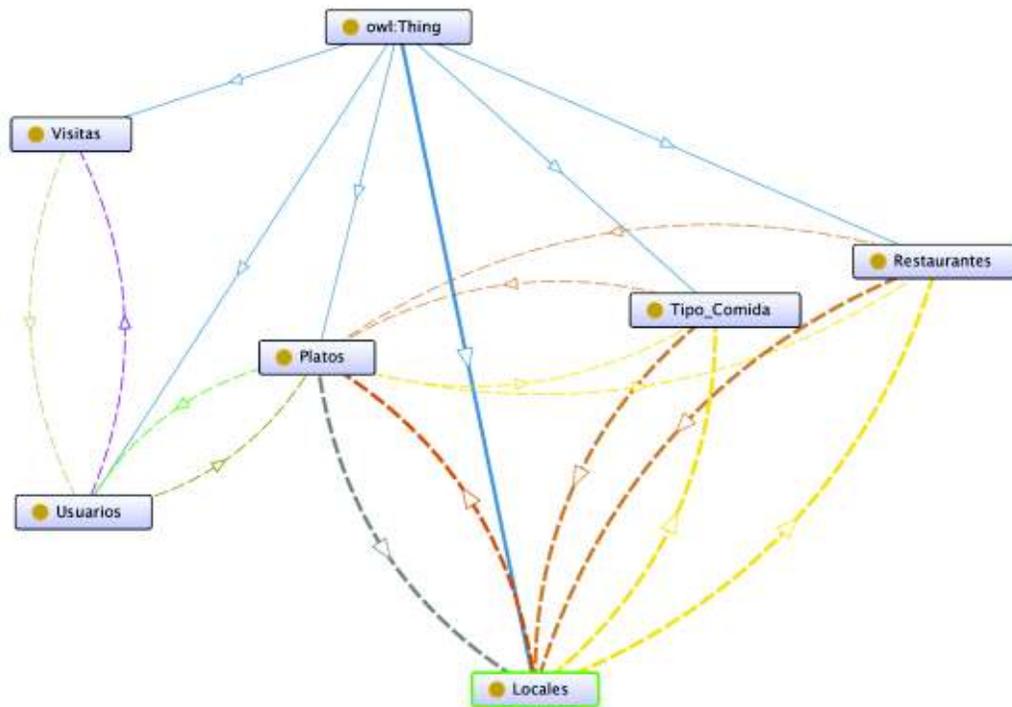


Figura 16. Dominio y rango de propiedad de tipo de dato.

Para este prototipo de aplicación móvil, centrada en los intereses de los usuarios que la usarán, se ha definido la ontología que se presenta en la Figura 17.



**Figura 17. Diseño de ontología**

### **3.7. Aplicación de la metodología de desarrollo del prototipo**

#### **3.7.1. Requerimientos**

El prototipo de aplicación móvil producto del desarrollo de software, requiere presentar sobre una interfaz de mapas, restaurantes basados en los intereses del usuario. Estos intereses se obtendrán por las visitas realizadas anteriormente por el usuario, y por los restaurantes que dicho usuario marque como favoritos. En el momento que el usuario seleccione un restaurante a ser visitado, podrá observar información importante del local como es dirección, teléfonos, horarios, distancia, y sobre una interfaz de mapa, la posición del usuario, la posición del restaurante y la ruta más cercana para llegar al mismo.

### 3.7.1.1. Definición del Product Backlog

Para la definición del product backlog, se toma en cuenta las historias de usuario que se encuentran en la sección Anexos, véase Anexo I.

Una vez revisadas y analizadas dichas historias se tiene la pila de producto que se la puede observar en la Tabla 4

**Tabla 4. Product Backlog inicial**

<b>Nro.</b>	<b>Título</b>	<b>Puntos de priorización</b>
HU1	Creación de Backend	95
HU2	Búsqueda de restaurante por ubicación	90
HU3	Detalles del restaurante	80
HU4	Navegar hacia el restaurante	80
HU5	Búsqueda de restaurante	70
HU6	Historial de restaurantes visitados	50

### 3.7.1.2. Supuestos y dependencias

Se desarrollará una aplicación nativa para sistemas Android con la utilización de los sensores disponibles en estos dispositivos. Los dispositivos Android deben contar con los siguientes requisitos:

- ✓ Android 4.2 + (API Level 17+)
- ✓ Brújula

- ✓ GPS
- ✓ Acelerómetro
- ✓ Dispositivos de alta resolución (hdpi)
- ✓ Soporte para OpenGL 2.0

De igual manera, el dispositivo deberá contar con una conexión a internet, ya sea a través de un plan de datos, o de una conexión WIFI, debido que los restaurantes son obtenidos en tiempo real, y la ruta se la obtiene de igual manera desde los servidores de Google.

#### **3.7.1.3. Restricciones**

- El prototipo de aplicación móvil considerará restaurantes localizados en un radio de 500 metros alrededor de la Escuela Politécnica Nacional.
- Se deberá contar con una conexión a internet para el correcto funcionamiento de la aplicación.

### 3.7.2. Diseño de interfaces

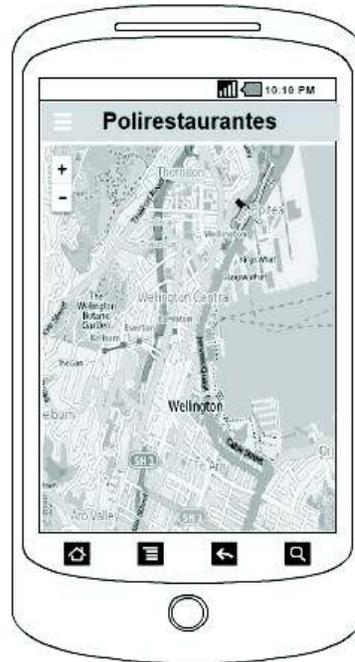


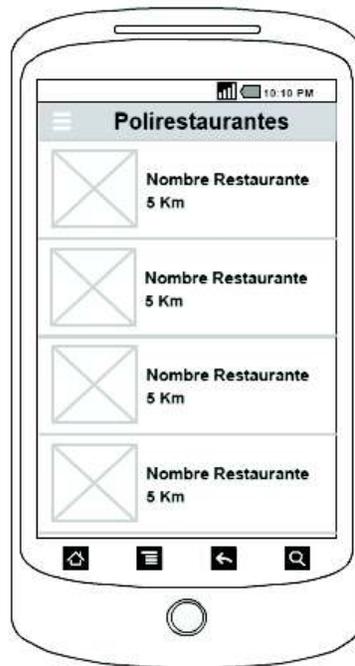
Figura 18. Interfaz Posición



Figura 19. Interfaz Detalle Restaurante



Figura 20. Interfaz Menú



**Figura 21. Interfaz Historial**

### **3.8. Arquitectura del prototipo**

Para el desarrollo del prototipo de aplicación móvil, se ha decidido utilizar uno de los marcos más conocidos para el diseño de interfaces gráficas de usuario. El marco de trabajo Modelo – Vista – Controlador soporta la presentación de los datos de diferentes formas e interacciones independientes con cada una de estas formas. Cuando los datos se modifican a través de una de las presentaciones, el resto de las presentaciones son actualizadas [25]. A continuación, se detalla cada una de sus partes [26]:

- i. Modelo. - Es la aplicación del dominio, no tiene conocimiento específico de la vista, que es la interfaz de usuario, ni de los controladores, que son los manejadores de eventos.
- ii. Vista (Interfaz de usuario). -Tiene conocimiento de los controladores (los manejadores de eventos) porque los crea y especifica qué tipo de evento, emitido por cada componente debe de atender cada manejador. Sin embargo, la vista no tiene conocimiento del Modelo.

- iii. Controlador. - Tienen conocimiento del Modelo, porque crean objetos del dominio al que mandan mensajes, derivados fundamentalmente de los eventos producidos en la interfaz de usuario; y tiene conocimiento de la Vista (Interfaz de usuario) porque mandan mensajes a sus componentes para recibir o enviarles información.

La arquitectura del prototipo de aplicación móvil se la puede apreciar en el la Figura 22.

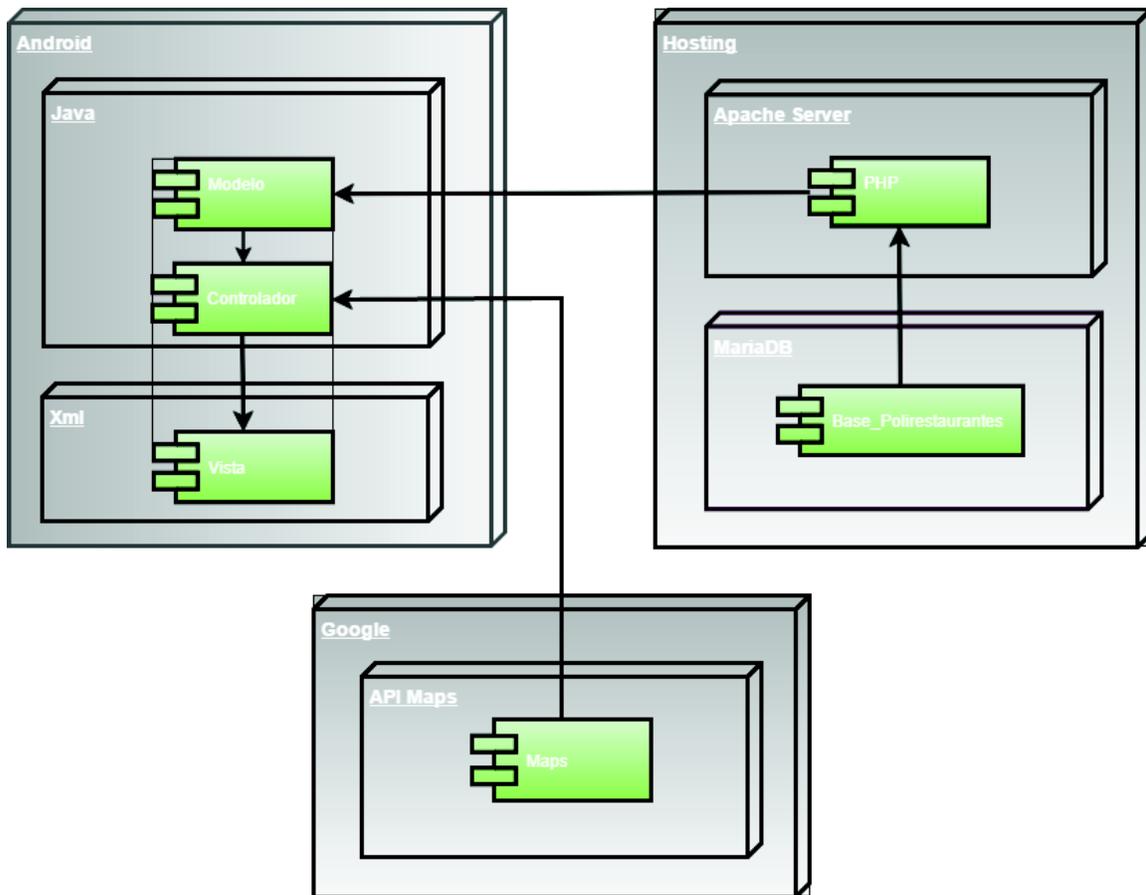


Figura 22. Arquitectura del Prototipo de aplicación móvil

### 3.9. Release Planning

Con ayuda del Product Backlog definido, el equipo de desarrollo ha acordado una duración de cada sprint de 2 semanas. Se define también el Release Planning que se observa en la Tabla 5.

**Tabla 5. Release Planing**

<b>Sprint 1</b>	<b>Sprint 2</b>	<b>Sprint 3</b>	<b>Sprint 4</b>
HU6	HU3	HU3	HU1
		HU2	HU5
		HU4	

### 3.10. Desarrollo de los Sprints

#### 3.10.1. Sprint 1

##### 3.10.1.1. Sprint Planning

###### 3.10.1.1.1. *Objetivo.*

Transformar la ontología desarrollada en el punto 3.6 a una base de datos relacional.

###### 3.10.1.1.2. *Tareas del Sprint*

Las tareas por desarrollar en este sprint son las que se observan en la Tabla 6.

**Tabla 6. Sprint 1**

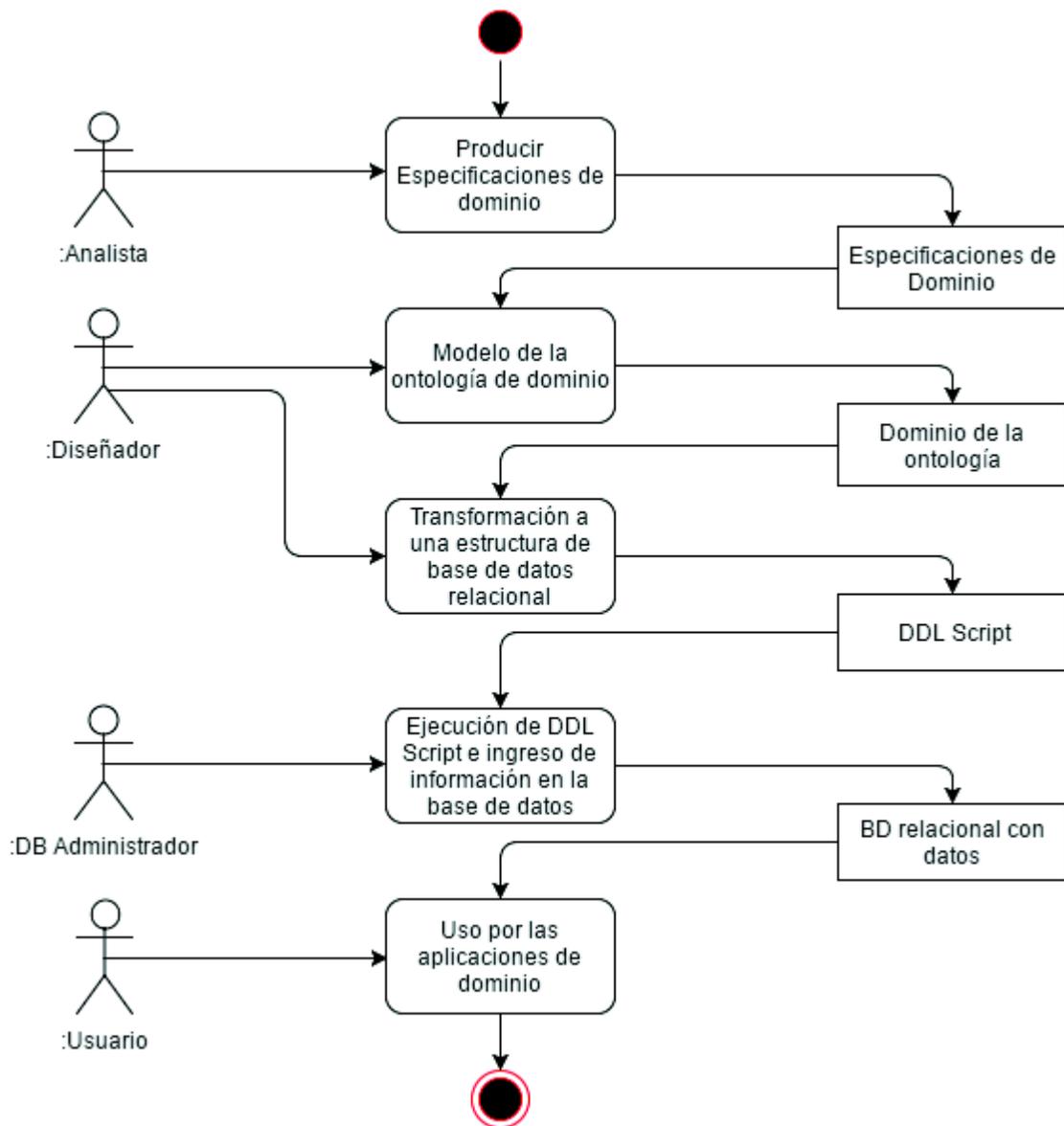
<b>Código</b>	<b>Historia de Usuario</b>	<b>Criterios de aceptación</b>	<b>Tarea</b>	<b>Estimación</b>
HU6	Yo como desarrollador quiero que se cree el	Observar la base de datos creada	Transformación de la ontología a una base de datos relacional	25

	Backend de la aplicación	Realizar consultas al web service y observar las respuestas con los datos que se requieren.	Creación del web services	15
--	--------------------------	---	---------------------------	----

### 3.10.1.2. Desarrollo del Sprint

#### 3.10.1.2.1. Transformación de la ontología en una base de datos relacional

En [27] proponen el proceso que se observa en la Figura 23 para desarrollar y transformar una ontología de dominio en una base de datos relacional.



**Figura 23. Transformación Ontología [27]**

Para lograr este cometido se procede, después del análisis correspondiente de la ontología a usarse, ejecutar la siguiente secuencia de pasos:

### **Transformación de las clases a tablas de la base de datos**

Para conseguir una transformación apropiada de las clases de la ontología en tablas en la base de datos relacional, se sigue el flujo que se observa en la Figura 24.

Luego de realizar el proceso, se obtiene como resultado las tablas requeridas en la base de datos relacional, dichas tablas son las que se observa en la Tabla 7.

**Tabla 7. Tablas resultantes**

<b>Tablas base de datos relacional.</b>
Usuarios
Visitas
Restaurantes
Tipo_Comida
Platos
Locales

**Transformación de las propiedades de tipo de datos en columnas en las tablas de la base de datos.**

A continuación, se procede a la transformación de las propiedades de tipo de datos en columnas en las tablas de la base de datos, como se muestra en la siguiente Figura 25.

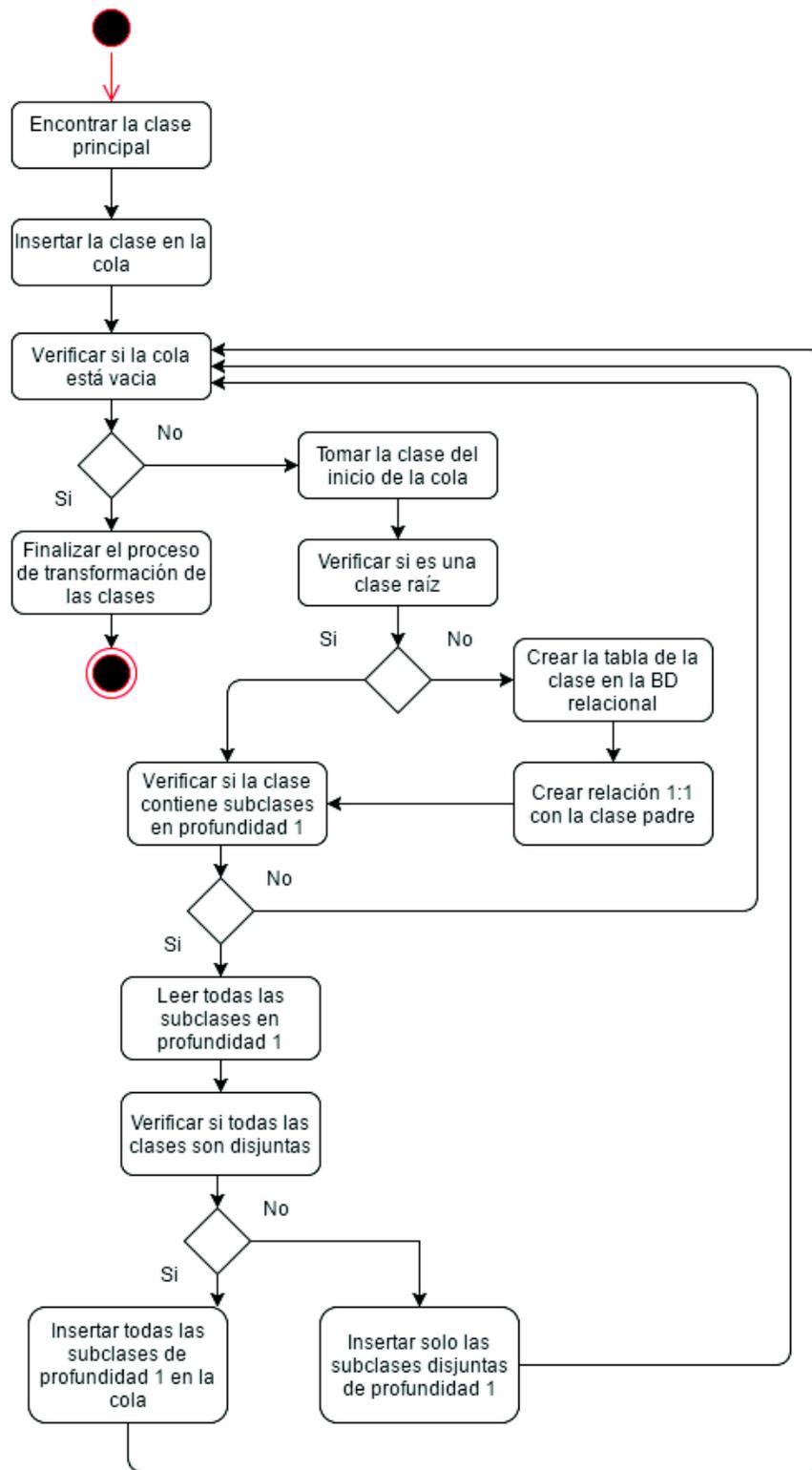
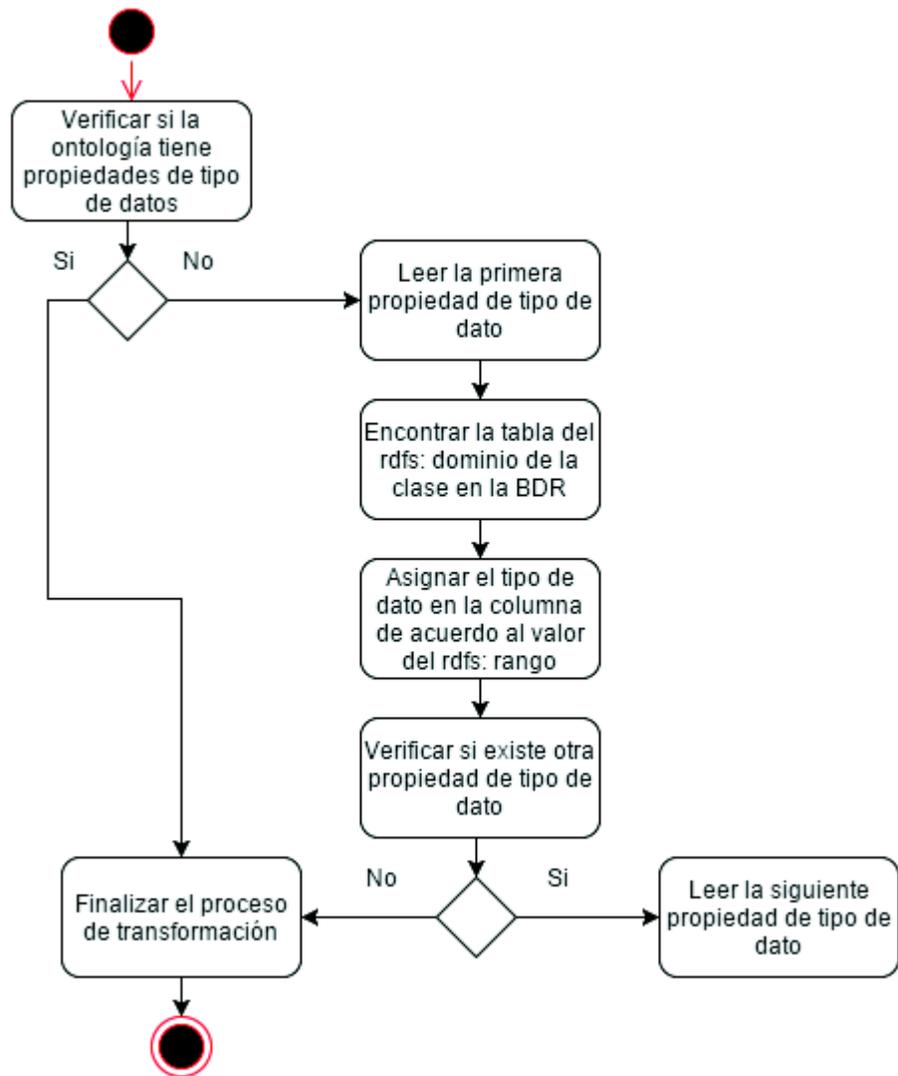


Figura 24. Transformación de las clases a tablas de la base de datos [27]



**Figura 25. Transformación de las propiedades de tipo de datos [27]**

Al realizar los procesos de transformación de los pasos anteriores, se obtiene una base de datos relacional, compuesta por las tablas y columnas que pueden observar en la Tabla 8.

**Tabla 8. Columnas en las tablas de la base de datos relacional.**

<b>Tabla</b>	<b>Columnas</b>
Usuarios	id
	nombre
	telefono
	correo
	dispositivo
Visitas	id
	fecha
	calificacion
Restaurantes	id
	representante_legal
	ruc
	razon_social
Tipo_Comida	id
	nombre
Platos	id
	nombre
	descripcion
Ubicación_Locales	id
	nombre
	direccion
	telefono
	latitud
	longitud
	hora_apertura
	hora_cierre

**Transformación de las propiedades de objetos en relaciones en la base de datos.**

Una vez que se cuenta con las tablas y con las columnas pertenecientes a cada tabla, se procede a la transformación de las propiedades de objetos en las relaciones que unan a las mismas. Para lograr este proceso se vincula con los pasos descritos en el diagrama que se observa en la Figura 26.

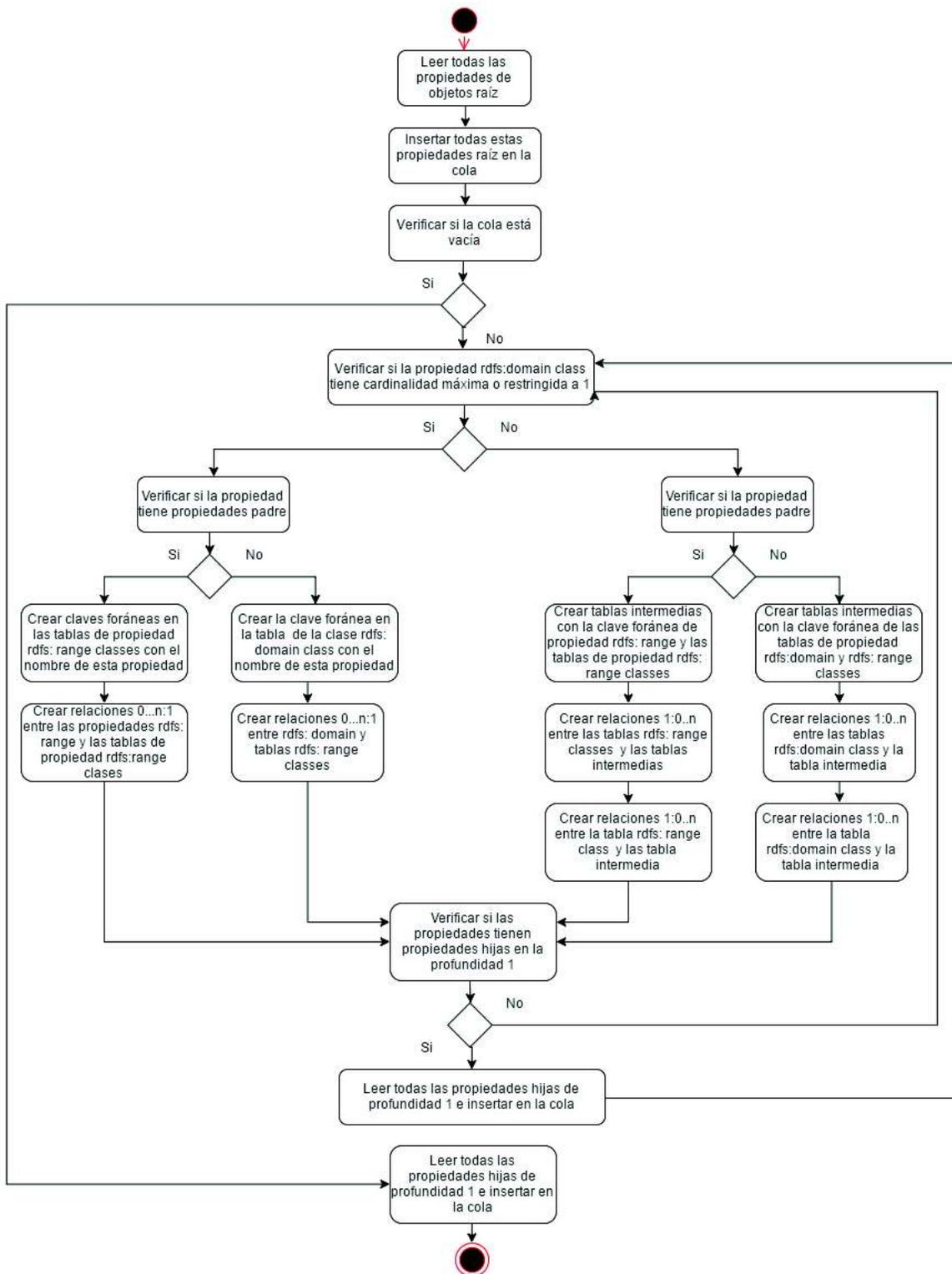


Figura 26. Transformación de las propiedades de objetos en relaciones de la base de datos [27]

## **Creación de la base de datos**

El modelo de base de datos está articulado con la información de la ontología definida para el prototipo de aplicación móvil, mediante la definición de tablas, campos y atributos a través de relaciones y entidades necesarias para el desarrollo del proyecto. El resultado final luego de ejecutar todos los pasos anteriores se los puede observar con claridad en la Figura 27.

Una vez que se ha creado el diagrama de la base de datos relacional, se procede a realizar la generación del Script para la creación de la base de datos, el cual se encuentra en la sección Anexos, véase Anexo II.

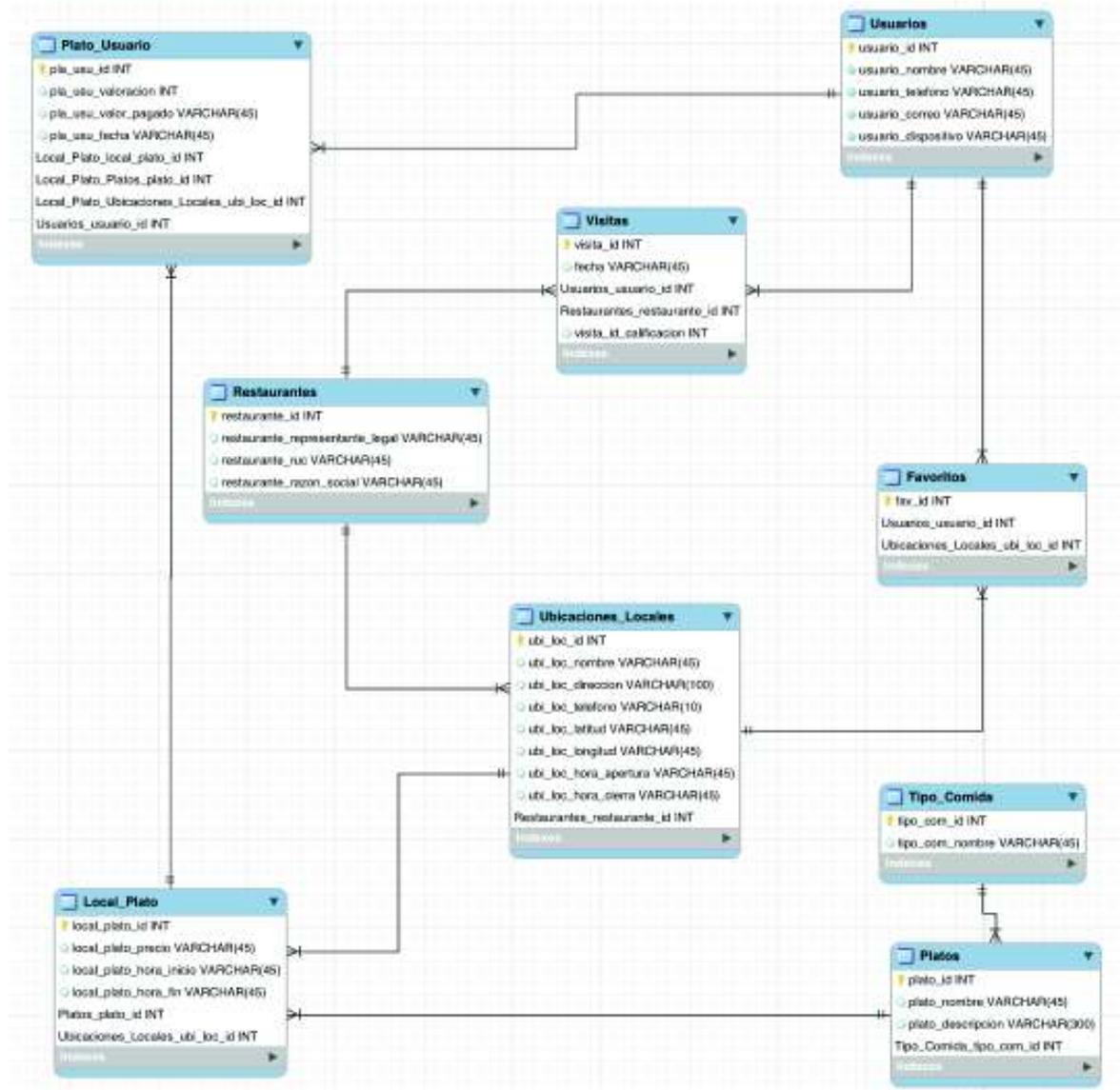


Figura 27. Modelo de datos

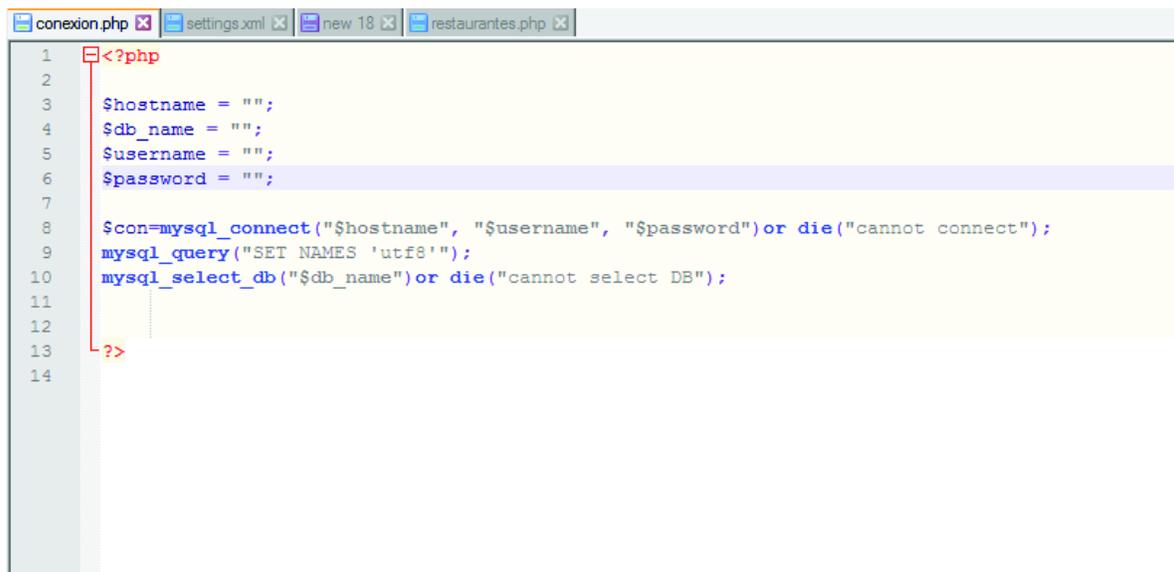
### 3.10.1.2.2. Creación de los Web Services

Para este proyecto se considera la creación de dos archivos con extensión php de los cuales, el primero contiene la conexión hacia la base de datos, y, el segundo los métodos necesarios para la presentación de los datos requeridos por el prototipo de aplicación. La creación de los

servicios web se lo realiza en la herramienta “Notepad ++”, son escritos en lenguaje Php y contienen la siguiente estructura:

Archivo conexión.php

Contiene información para la conexión con la base de datos. Este se lo puede observar en la Figura 28



```
1 <?php
2
3 $hostname = "";
4 $db_name = "";
5 $username = "";
6 $password = "";
7
8 $con=mysql_connect("$hostname", "$username", "$password")or die("cannot connect");
9 mysql_query("SET NAMES 'utf8'");
10 mysql_select_db("$db_name")or die("cannot select DB");
11
12
13 ?>
14
```

**Figura 28. Conexión.php**

Archivo restaurantes.php

Este archivo consta de los métodos que se pueden apreciar en la Figura 29:

```
conexion.php x restaurantes.php x settings.xml x new 18 x
1 <?php
2 include_once 'credentials.php';
3
4 $vall = $_GET["param1"];
5 switch ($vall)
6 {
7
8     case 'consultarFavoritos':
9     {
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40     case 'setFavorito':
41     {
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64     case 'registrarVisita':
65     {
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105     case 'tipoComida':
106     {
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126     case 'consultarPersona':
127     {
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150     case 'registroUsuario':
151     {
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188     case 'actualizarPersona':
189     {
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241     case 'eliminarPersona':
242     {
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268     case 'nombreCalificacionTipo':
269     {
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298     case 'nombreCalificacion':
299     {
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330     case 'nombreTipo':
331     {
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359     case 'nombre':
360     {
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391     case 'restaurantesPosicion':
392     {
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416     case 'restaurantesPosicionCalificacion':
417     {
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444 }
445 header('Content-type: text/html;charset=utf-8');
446 echo json_encode($json);
447
448
449 ?>
```

Figura 29. Restaurantes.php

### 3.10.1.3. Burn Down Chart

En la Figura 30 se muestra la velocidad de desarrollo del primer Sprint, se tiene que las horas restantes de trabajo están por debajo de las horas ideales, lo que denota que la velocidad del equipo es buena y va acorde a lo planificado. Al final del sprint se tiene como resultado que las tareas asignadas se desarrollan en el tiempo estimado.

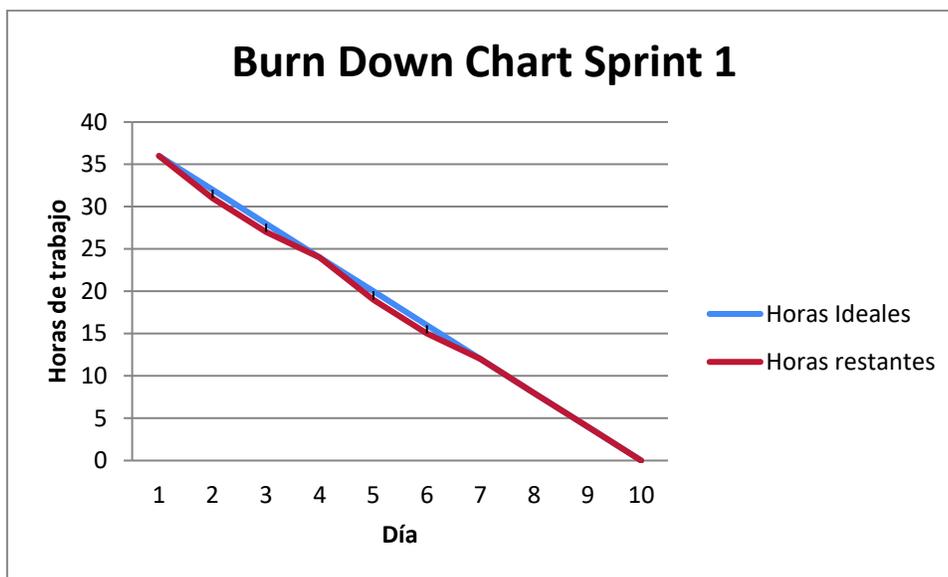


Figura 30. Burn down chart sprint1

### 3.10.1.4. Sprint Review

Se transformó la ontología en una base de datos relacional para que sea usada por el prototipo de aplicación móvil. Se construyeron los servicios web, iniciando por el archivo de conexión a la base de datos, y los métodos del servicio web que envían la información al prototipo móvil.

En la Tabla 9 se aprecia el criterio de aceptación de la historia de usuario desarrollada.

**Tabla 9. Criterios de aceptación Sprint 1**

<b>Código</b>	<b>Historia de Usuario</b>	<b>Criterios de aceptación</b>	<b>Tarea</b>	<b>Aceptado/Rechazado</b>	<b>Comentarios</b>
HU6	Yo como desarrollador quiero que se cree el Backend de la aplicación	Observar la base de datos creada	Transformación de la ontología a una base de datos relacional	Aceptado	
		Realizar consultas a los servicios web y observar las respuestas con los datos que se requieren.	Creación de los servicios web.	Aceptado	

### **3.10.1.5. Sprint Retrospective**

Al finalizar el primer sprint se evidencian problemas con la adopción de la metodología, por lo que se propone realizar una capacitación rápida sobre esta, ya que sería de gran ayuda al desarrollo del proyecto.

## **3.10.2. Sprint 2**

### **3.10.2.1. Sprint Planning**

#### *3.10.2.1.1. Objetivo.*

Mostrar restaurantes en una interfaz de mapas, cercanos a la posición del dispositivo.

### 3.10.2.1.2. Tareas del Sprint

En este sprint se completarán las actividades que se observan en la Tabla 10.

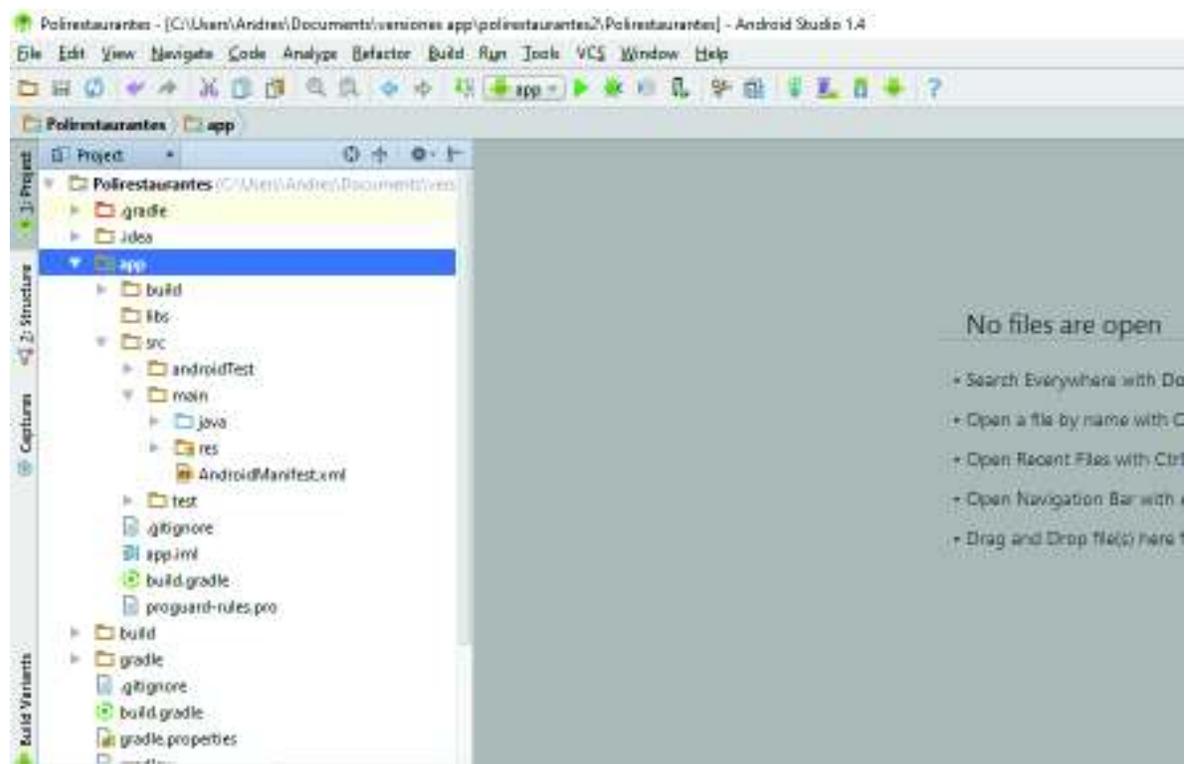
**Tabla 10. Sprint 2**

<b>Código</b>	<b>Historia de Usuario</b>	<b>Criterios de aceptación</b>	<b>Tarea</b>	<b>Estimación (horas)</b>
HU3	Yo como usuario quiero observar en el mapa mi posición, y a través de esta los restaurantes cercanos.	Observar en un mapa las marcas de la posición del usuario y el o los restaurantes más cercanos al usuario. Luego de analizar el interés del usuario.	Creación del Proyecto Android	3
			Creación de la actividad Main Activity	7
			Creación de la actividad Inicio	30

### 3.10.2.2. Desarrollo del Sprint

#### 3.10.2.2.1. Creación del Proyecto Android

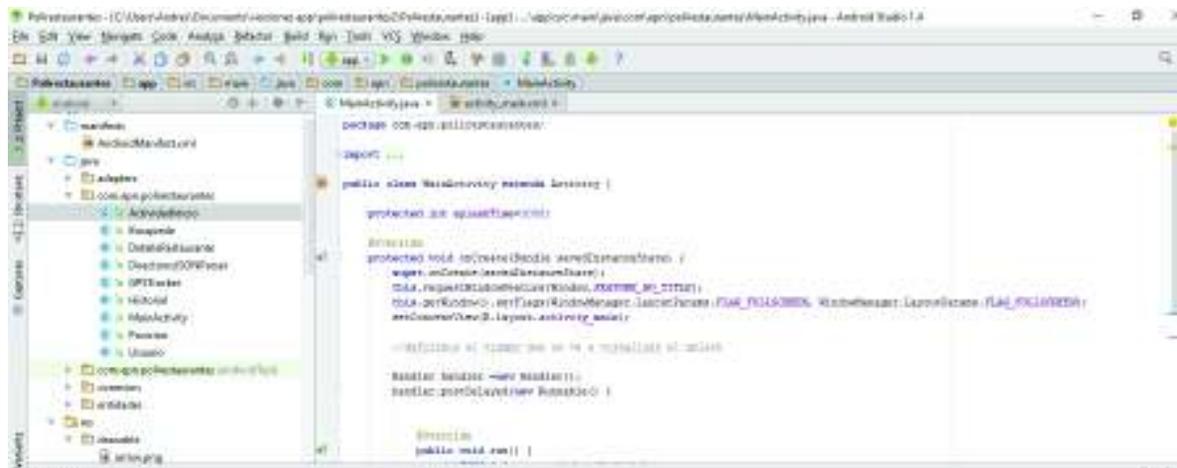
Para construir el prototipo de aplicación móvil, se utiliza la herramienta Android Studio descrita en la sección 2.3, y se puede apreciar en la Figura 31.



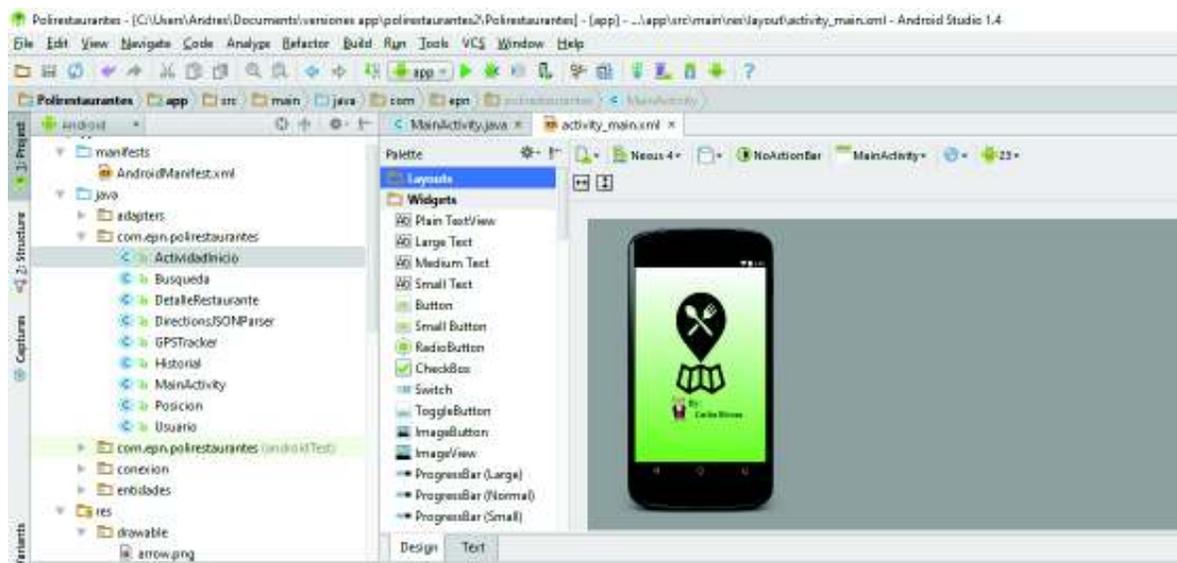
**Figura 31. Creación Proyecto en Android Studio**

### **3.10.2.2.2. Creación de la actividad Main Activity**

Esta actividad se encarga de presentar el splash del prototipo de aplicación móvil por un segundo, antes de dar paso a la actividad Inicio, la actividad se la puede observar en la Figura 32, mientras que el layout que contiene dicho splash se observa en la Figura 33.



**Figura 32. Creación de la actividad Main Activity**



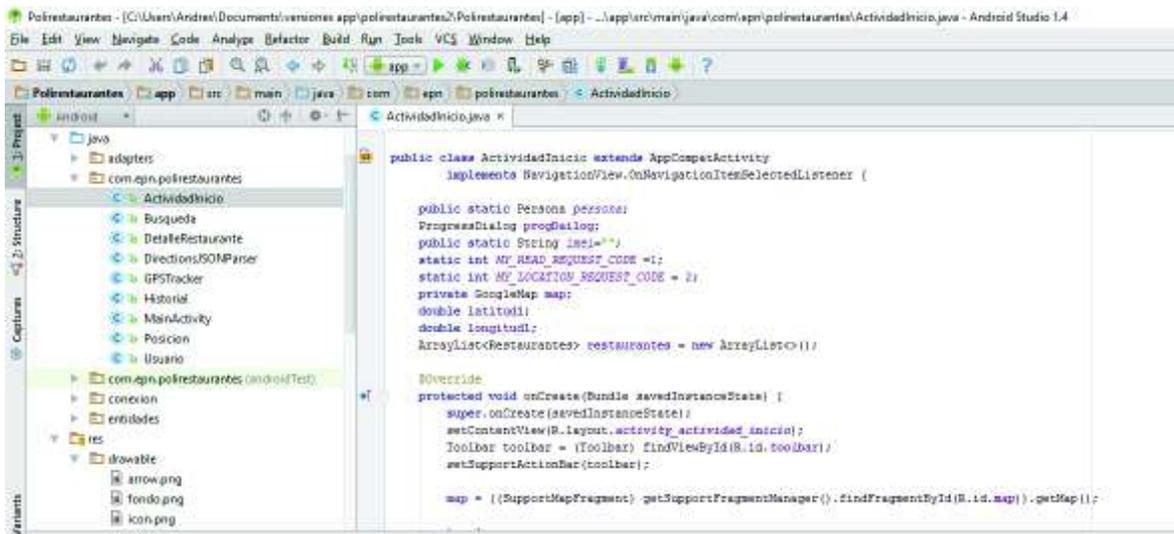
**Figura 33. Creación Layout Activity\_main.xml**

### 3.10.2.2.3. Creación de la actividad Inicio

Esta actividad se encarga de alojar las siguientes tareas, y se puede apreciar en la Figura 34:

1. Asigna el fragmento al mapa

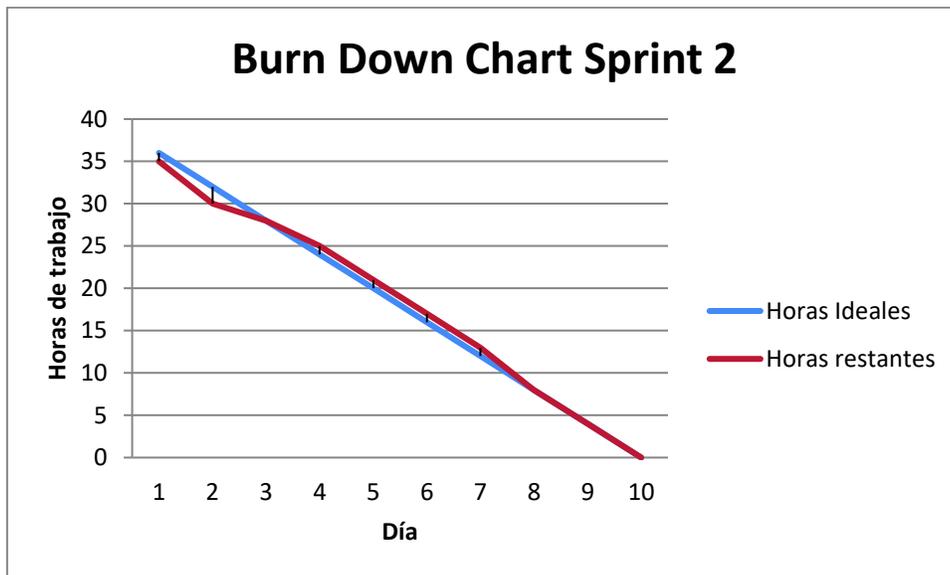
2. Solicita los permisos necesarios para la lectura de la posición.
3. Obtiene la posición actual del dispositivo.
4. Obtiene el "IMEI" del dispositivo.
5. Busca los datos del usuario asociado al dispositivo.
6. Está a la escucha de si es seleccionado algún fragmento desde el menú.
7. Carga el fragmento posición.
8. Dibuja las marcas en el mapa con cada restaurante.



**Figura 34. Creación Actividad Inicio**

### 3.10.2.3. Burn Down Chart

A continuación, en la Figura 36 se muestra la velocidad de desarrollo del equipo en el sprint 2. Al inicio de este sprint se tiene una mejora en el tiempo de desarrollo. Y conforme avanza el tiempo se va teniendo un ligero retraso, el cual es solventado y el sprint concluye dentro del tiempo establecido.



**Figura 35. Burn down chart sprint 2**

### 3.10.2.4. Sprint Review

Se inició el proyecto Android añadiendo la primera actividad con su pantalla, la cual muestra por unos segundos una imagen de presentación de la aplicación. Se añade además la segunda actividad que se encarga de presentar el primer fragmento el cual muestra un mapa con las marcas de los restaurantes.

Los criterios de aceptación de este sprint se observan en la Tabla 11.

**Tabla 11. Criterios de Aceptación Sprint 2**

Código	Historia de Usuario	Criterios de aceptación	Tarea	Aceptado/Rechazado	Comentarios
HU3	Yo como usuario quiero observar en el	Observar en un mapa las marcas de la posición del	Creación del Proyecto Android	Aceptado	

mapa mi posición, y a través de esta los restaurantes cercanos.	usuario y el o los restaurantes más cercanos al usuario. Luego de analizar el interés del usuario.	Creación de la actividad Main Activity	Aceptado	
		Creación de la actividad Inicio	Aceptado	

### 3.10.2.5. Sprint Retrospective

La metodología es aceptada de mejor manera. Se realiza una capacitación sobre la metodología, con la finalidad de lograr una mejor implementación de la misma.

### 3.10.3. Sprint 3

#### 3.10.3.1. Sprint Planning

##### 3.10.3.1.1. *Objetivo.*

Implementar funcionalidad para la búsqueda de restaurantes por ubicación, presentando detalles del restaurante y navegación hacia el mismo.

##### 3.10.3.1.2. *Tareas del Sprint*

En este sprint se completarán las actividades que se observan en la Tabla 12.

Tabla 12. Sprint 3

Código	Historia de Usuario	Criterios de aceptación	Tarea	Estimación (horas)
HU3	Yo como usuario quiero observar en el mapa mi posición, y a través de esta los restaurantes cercanos.	Observar en un mapa las marcas de la posición del usuario y el o los restaurantes más cercanos al usuario. Luego de analizar el interés del usuario.	Creación del Layout posición.xml	10
HU2	Yo como usuario quiero observar información relevante del restaurante que he seleccionado, además del mapa mostrando la zona donde se ubica dicho restaurante	Observar detalles del restaurante, información, teléfono, horarios de atención, entre otros. Además, observar un mapa con la ubicación de dicho restaurante	Creación de la Actividad Detalle restaurante	7
			Creación del Layout detalleRestaurante.xml	7
HU4	Yo como usuario, una vez seleccionado el restaurante al que deseo dirigirme, quiero navegar a través del GPS hacia dicho restaurante.	Observar en un mapa, tanto la posición en la que se encuentra el usuario como las marcas de los lugares en donde se encuentran los restaurantes.	Implementar funcionalidad de posición en el mapa	16

### 3.10.3.2. Desarrollo del Sprint

#### 3.10.3.2.1. Creación del Layout posición.xml

El layout contiene un fragmento en el cual se dibuja el mapa con las marcas de cada restaurante. Este layout se lo puede observar en la Figura 36.

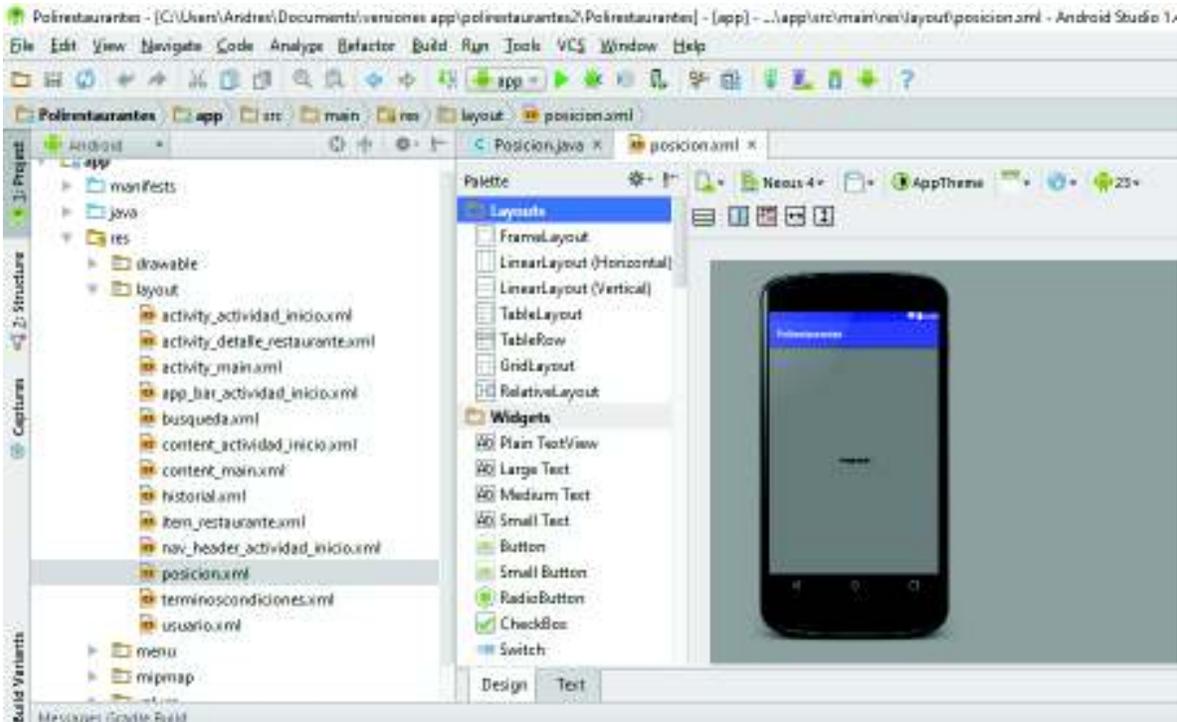
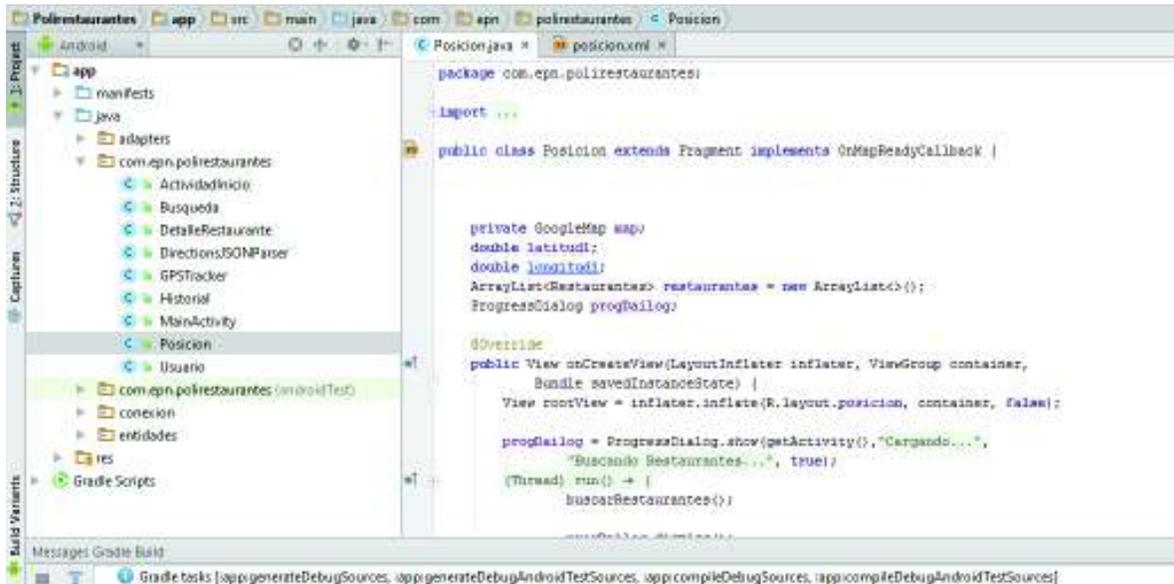


Figura 36. Creación de layout posición.xml

#### 3.10.3.2.2. Implementar funcionalidad de posición en el mapa

Para implementar esta funcionalidad se crea la Actividad *Posición*, la cual cumple las siguientes funciones, se la observa en la Figura 37:

1. Recupera el layout posición.xml.
2. Recupera todos los restaurantes cercanos a la posición del usuario.
3. Asigna el mapa al fragmento en el layout posición.xml.
4. Se configuran las opciones necesarias para la visualización del mapa.
5. Se dibujan los marcadores en el mapa, de acuerdo con la posición de cada restaurante.



**Figura 37. Creación actividad Posición.**

### 3.10.3.2.3. Creación de la Actividad Detalle restaurante

La actividad Detalle restaurante se encarga de las siguientes tareas, se las observa en la Figura 38:

1. Cargar el layout detalle restaurante y sus componentes.
2. Recuperar del bundle los extras enviados desde la actividad anterior.
3. Cargar la información del restaurante en los componentes del layout detalle restaurante.
4. Registrar la visita del restaurante.
5. Obtener la posición actual del dispositivo.
6. Dibujar la posición del restaurante en el mapa.
7. Dibujar la ruta más cercana al restaurante en el fragmento del layout.
8. Poner un listener en el mapa, el cual escucha los cambios de posición del dispositivo, para dibujar nuevamente la posición del dispositivo móvil en el mapa.

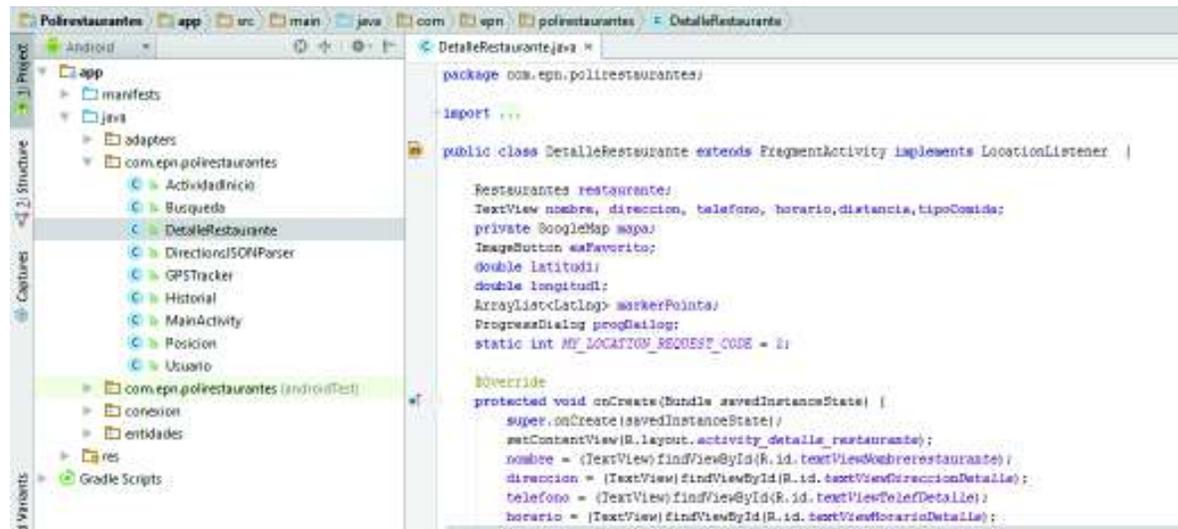


Figura 38. Actividad Detalle Restaurante.

#### 3.10.3.2.4. Creación del Layout detalle restaurante

El layout detalle restaurante tiene los elementos que se pueden apreciar en la Figura 39:

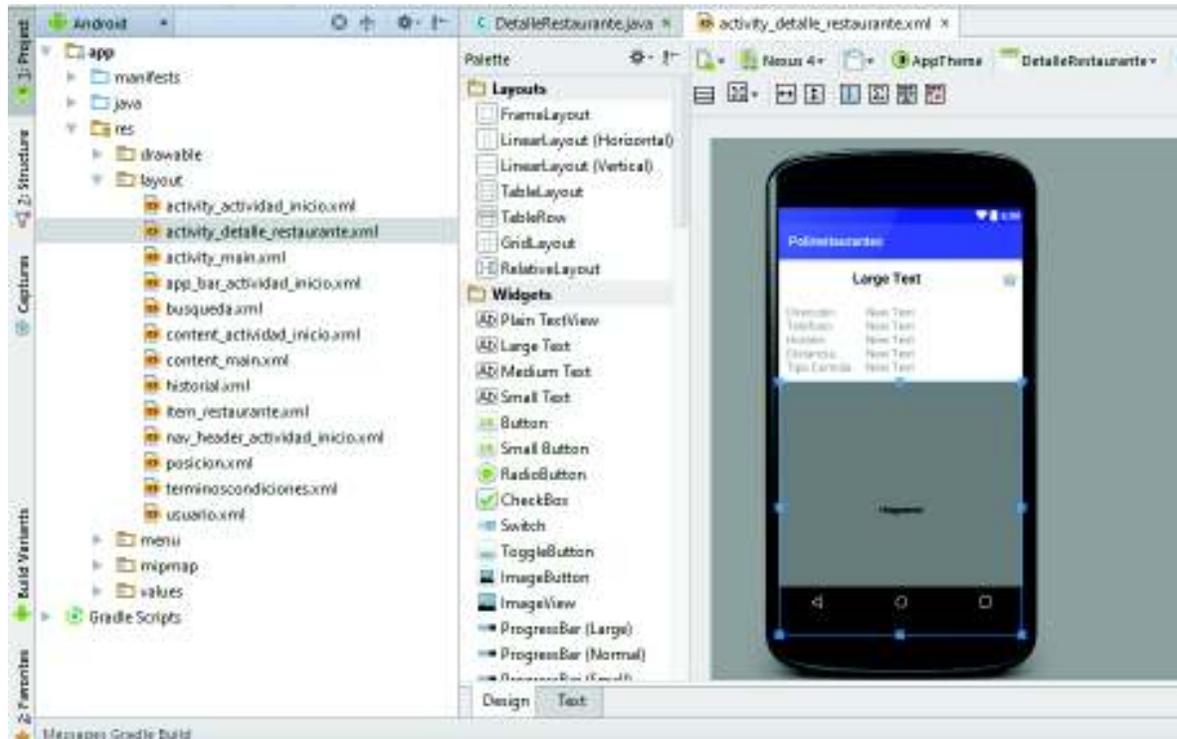


Figura 39. Layout Detalle Restaurante.

### 3.10.3.3. Burn Down Chart

En la Figura 40 se observa la velocidad del equipo en el desarrollo del sprint 3. En este sprint se tuvo que solventar la falta a un día de desarrollo por lo que el trabajo se retrasó. Se pudo concluir el sprint a tiempo por el compromiso del equipo de desarrollo.

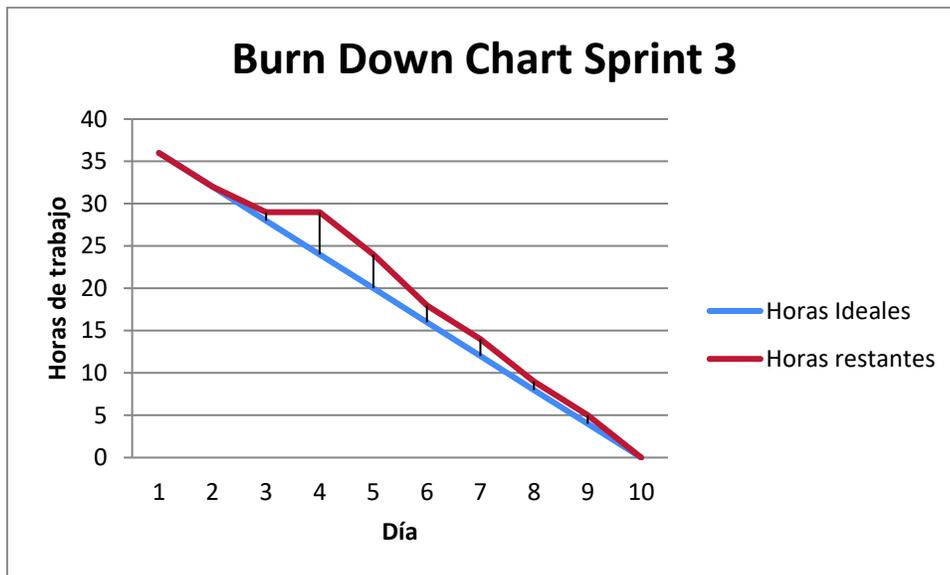


Figura 40. Burn down chart sprint 3

#### 3.10.3.4. Sprint Review

En este sprint se logró la creación del layout que presenta la posición del dispositivo móvil en el mapa, como también se creó la actividad y layout para la presentación de los detalles del restaurante.

Los criterios de aceptación de este sprint se los observa en la Tabla 13.

**Tabla 13. Criterios de aceptación Sprint 3**

<b>Código</b>	<b>Historia de Usuario</b>	<b>Criterios de aceptación</b>	<b>Tarea</b>	<b>Aceptado / Rechazado</b>	<b>Comentarios</b>
HU3	Yo como usuario quiero observar en el mapa mi posición, y a través de esta los restaurantes cercanos.	Observar en un mapa las marcas de la posición del usuario y el o los restaurantes más cercanos al usuario. Luego de analizar el interés del usuario.	Creación del Layout posición.xml	Aceptado	
HU2	Yo como usuario quiero observar información relevante del restaurante que he seleccionado, además del mapa mostrando la zona donde se ubica dicho restaurante	Observar detalles del restaurante, información, teléfono, horarios de atención, entre otros	Creación de la Actividad Detalle restaurante	Aceptado	
		Mapa con la ubicación de dicho restaurante	Creación de la Actividad Detalle restaurante	Aceptado	
			Creación del Layout detalle restaurante	Aceptado	
HU4	Yo como usuario, una vez seleccionado el restaurante al que deseo dirigirme, quiero navegar a través del GPS hacia dicho restaurante.	Observar en un mapa, tanto la posición en la que se encuentra el usuario como las marcas de los lugares en donde se encuentran los restaurantes.	Implementar funcionalidad de posición en el mapa	Aceptado	

### 3.10.3.5. Sprint Retrospective

El conocimiento sobre la metodología es más profundo, por lo que el desarrollo es más eficiente. El equipo de desarrollo no asiste al desarrollo por motivos de calamidad domestica por lo que las tareas se retrasan. Se logra recuperar las tareas faltantes y el sprint concluye en el tiempo planificado.

## 3.10.4. Sprint 4

### 3.10.4.1. Sprint Planning

#### 3.10.4.1.1. *Objetivo.*

Presentar restaurantes después de realizar una búsqueda, o los visitados anteriormente por el usuario, implementar la actualización de la posición en el dispositivo

#### 3.10.4.1.2. *Tareas del Sprint*

En este sprint se completarán las actividades que se observan en la Tabla 16.

Tabla 14. Sprint 4

Código	Historia de Usuario	Criterios de aceptación	Tarea	Estimación (horas)
HU1	Yo como usuario quiero buscar un restaurante a través del nombre	Observar una lista con los nombres de los restaurantes que coincidan con el nombre de la búsqueda, además de la distancia hasta el mismo	Creación actividad búsqueda	11
			Creación layout búsqueda	9

HU5	Yo como usuario quiero observar una lista con los restaurantes visitados.	Observar en una lista los restaurantes visitados anteriormente	Creación actividad historial	11
			Creación layout historial	9

### 3.10.4.2. Desarrollo del Sprint

#### 3.10.4.2.1. Implementar actualización de posición en el mapa

Para la actualización de la posición del dispositivo en el mapa se debe implementar la interfaz “LocationListener”, la cual está a la escucha de los cambios de posición del dispositivo para realizar una acción determinada, se observa en la Figura 41.

```

@Override
public void onLocationChanged(Location location) {

    double latitud;
    double longitud;

    SharedPreferences sp = new SharedPreferences(DetalleRestaurante.this);
    latitud = sp.getFloat("latitud");
    longitud = sp.getFloat("longitud");

    final LatLng nuevaLocalizacion = new LatLng(latitud, longitud);
    final LatLng restaurantePosicion = new LatLng(Double.parseDouble(sp.getString("latitud", "")), Double.parseDouble(sp.getString("longitud", "")));
    sp.edit().putFloat("latitud", location.getLatitude()).putFloat("longitud", location.getLongitude()).commit();

    mMap.setLocationSource(new LocationSource() {
        @Override public Location getLocation() {
            return nuevaLocalizacion;
        }
    });
}

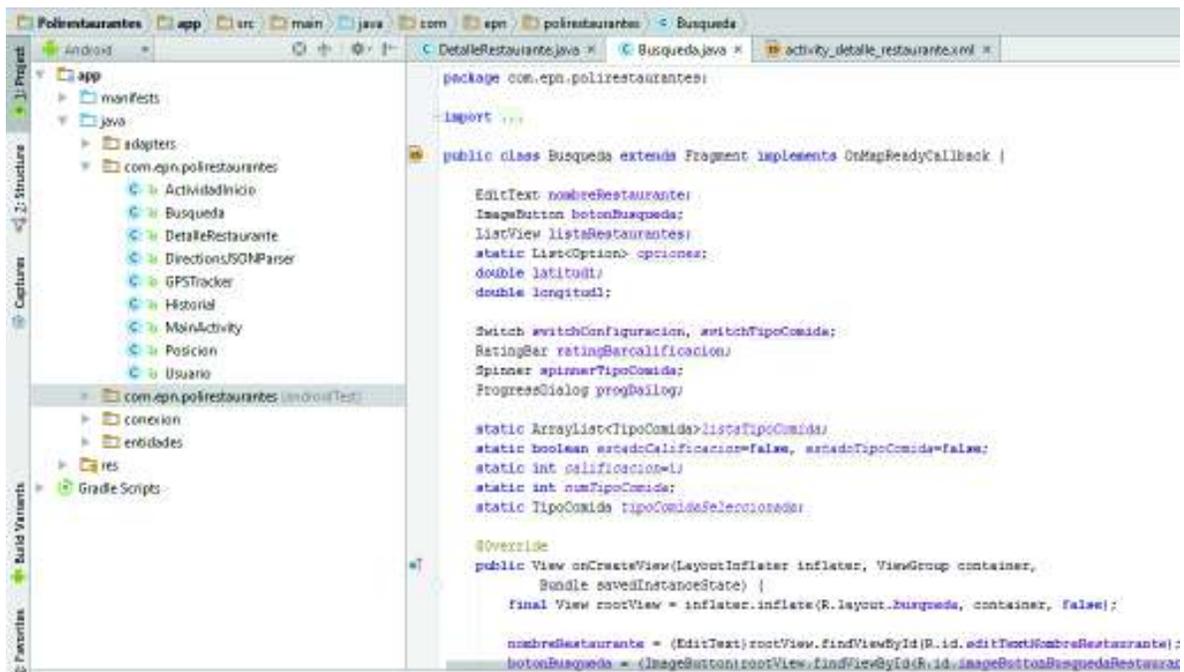
```

Figura 41. LocationListener en la actividad Detalle Restaurante.

### 3.10.4.2. Creación actividad búsqueda

La actividad, búsqueda, se encarga de las siguientes tareas y se observa en la Figura 42:

1. Cargar el layout de búsqueda
2. Obtener los datos de la posición del dispositivo
3. Definir el método de búsqueda y asignarlo al botón
4. Obtener los restaurantes
5. Dibujar una lista con los restaurantes
6. Estar a la espera de que el usuario seleccione un restaurante



```
package com.epn.polirestaurantes;

import android.os.Bundle;
import android.support.design.widget.Snackbar;
import android.support.design.widget.TextInputLayout;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.RatingBar;
import android.widget.Spinner;
import android.widget.Switch;
import android.widget.Toast;
import android.widget.ProgressBar;
import android.widget.ProgressDialog;

import com.epn.polirestaurantes.adapters.ListaRestaurantes;
import com.epn.polirestaurantes.adapters.ListaTiposComida;
import com.epn.polirestaurantes.modelo.Restaurante;
import com.epn.polirestaurantes.modelo.TipoComida;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.Task;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.Query;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class Busqueda extends Fragment implements OnMapReadyCallback {

    EditText nombreRestaurante;
    ImageButton botonBusqueda;
    ListView listaRestaurantes;
    static List<Option> opciones;
    double latitud;
    double longitud;

    Switch switchConfiguracion, switchTipoComida;
    RatingBar ratingBarCalificacion;
    Spinner spinnerTipoComida;
    ProgressDialog progressDialog;

    static ArrayList<TipoComida> listaTipoComida;
    static boolean estadoCalificacion=false, estadoTipoComida=false;
    static int calificacion=1;
    static int numTipoComida;
    static TipoComida tipoComidaSeleccionada;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        final View rootView = inflater.inflate(R.layout.busqueda, container, false);

        nombreRestaurante = (EditText)rootView.findViewById(R.id.editNombreRestaurante);
        botonBusqueda = (ImageButton)rootView.findViewById(R.id.imageBotonBusquedaRestaurante);
        listaRestaurantes = (ListView)rootView.findViewById(R.id.listaRestaurantes);
        switchConfiguracion = (Switch)rootView.findViewById(R.id.switchConfiguracion);
        switchTipoComida = (Switch)rootView.findViewById(R.id.switchTipoComida);
        ratingBarCalificacion = (RatingBar)rootView.findViewById(R.id.ratingBarCalificacion);
        spinnerTipoComida = (Spinner)rootView.findViewById(R.id.spinnerTipoComida);
        progressDialog = (ProgressDialog)rootView.findViewById(R.id.progressBar);

        return rootView;
    }
}
```

Figura 42. Actividad Búsqueda

### 3.10.4.2.3. Creación layout búsqueda

El layout búsqueda contiene los elementos para que el usuario localice restaurantes utilizando ciertos filtros de búsqueda, dichos elementos se pueden observar en la Figura 43.

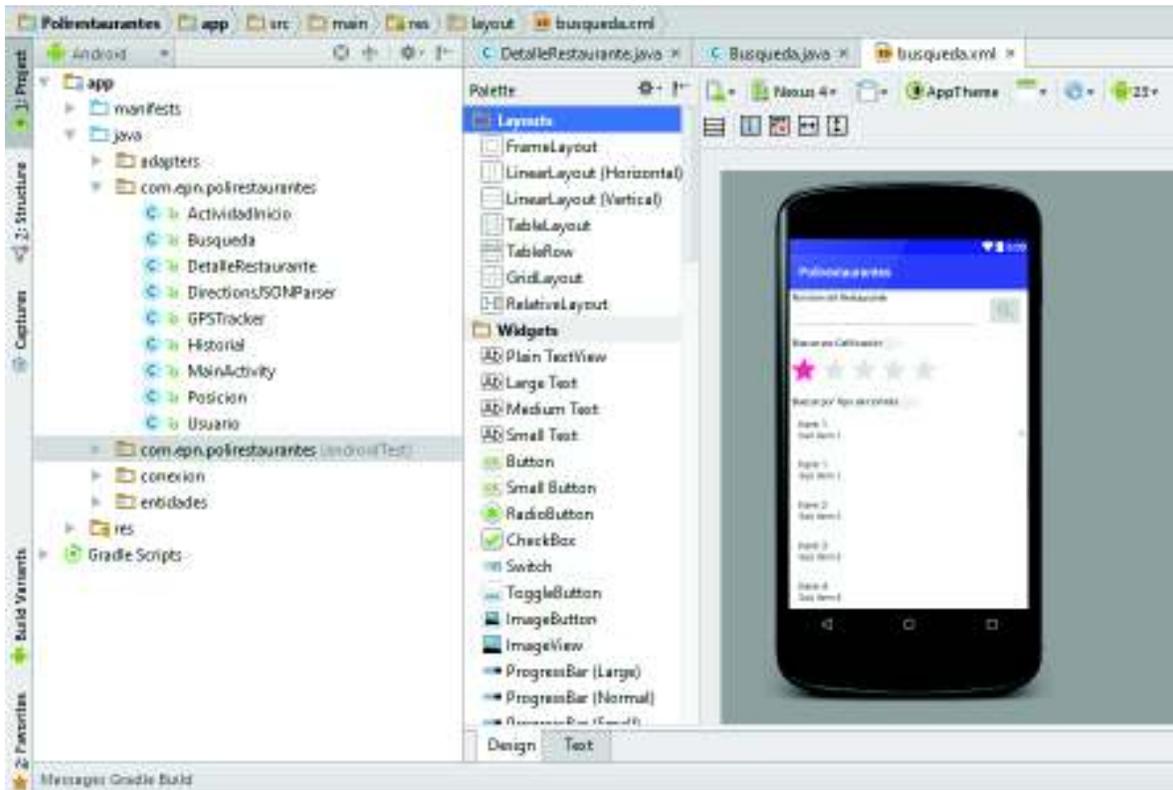


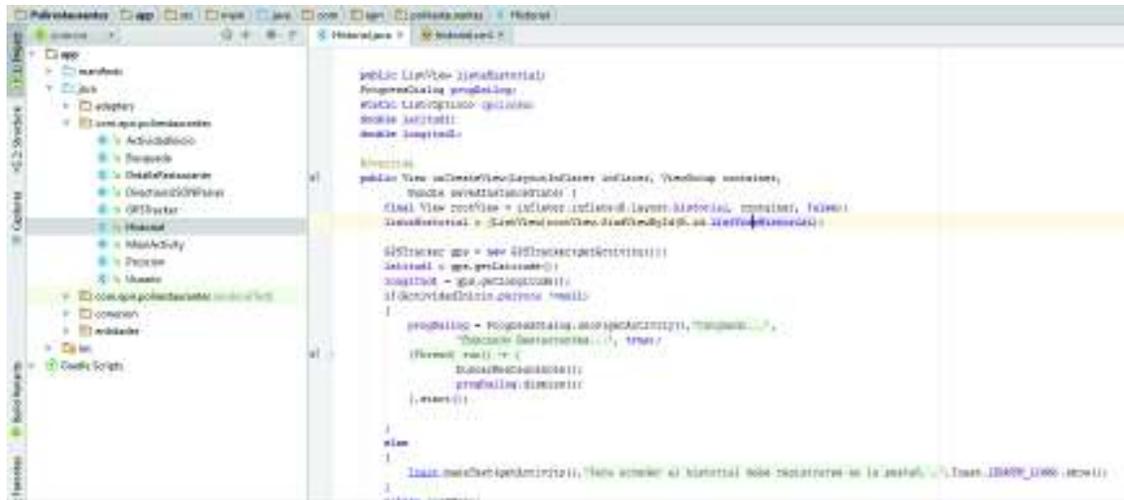
Figura 43. Layout Búsqueda.

### 3.10.4.2.4. Creación actividad historial

La actividad historial, se encarga de presentar los restaurantes visitados por el usuario, para esto realiza las siguientes actividades, se observan en la Figura 44:

1. Cargar el layout Historial
2. Obtener datos de localización del dispositivo
3. Obtener datos del usuario

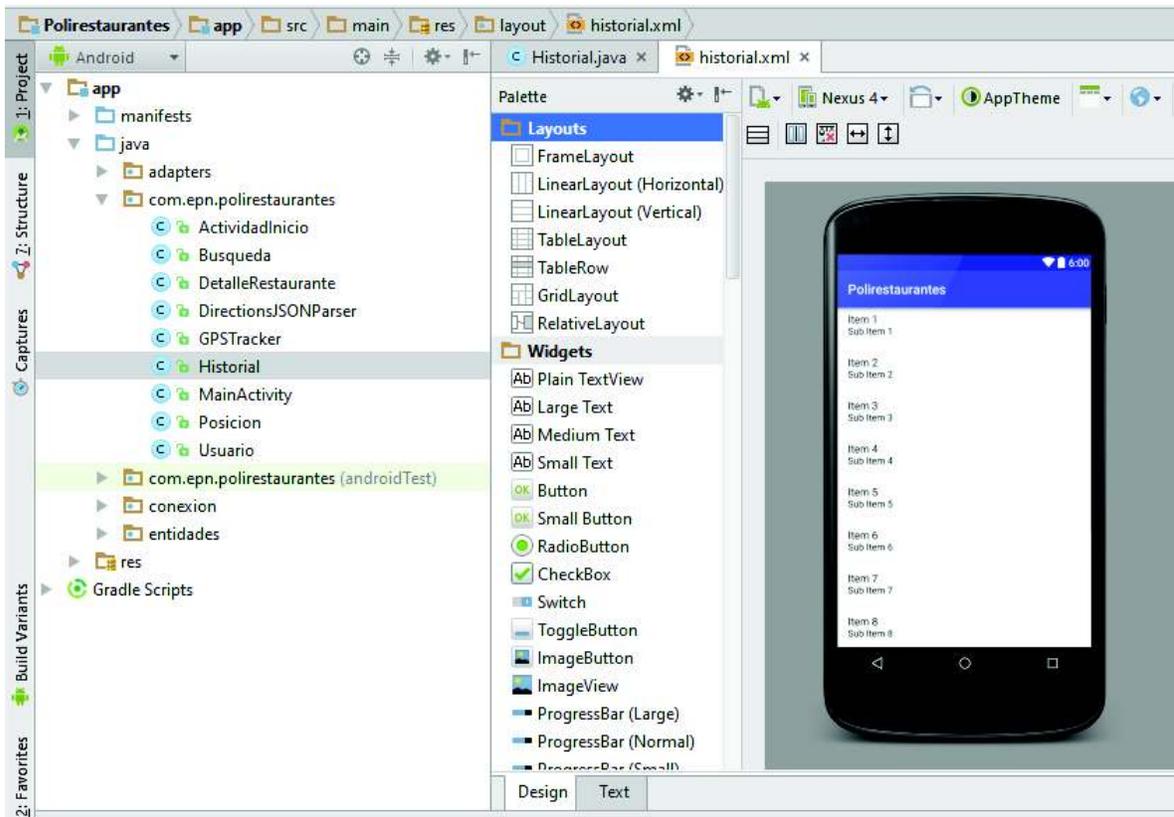
4. Consultar los restaurantes visitados
5. Dibujar una lista con los restaurantes
6. Estar a la espera de que el usuario seleccione un restaurante



**Figura 44. Actividad Historial.**

### 3.10.4.2.5. Creación layout historial

El layout historial es el encargado de presentar una lista de restaurantes que el usuario ha visitado, tal layout se puede apreciar en la Figura 45.



**Figura 45. Layout Historial**

### 3.10.4.3. Burn Down Chart

En la Figura 46 se observa la velocidad del equipo en el desarrollo de las tareas del sprint 4. Este sprint tiene un ligero retraso el cual se solventa y permite terminar el sprint dentro del tiempo planificado.

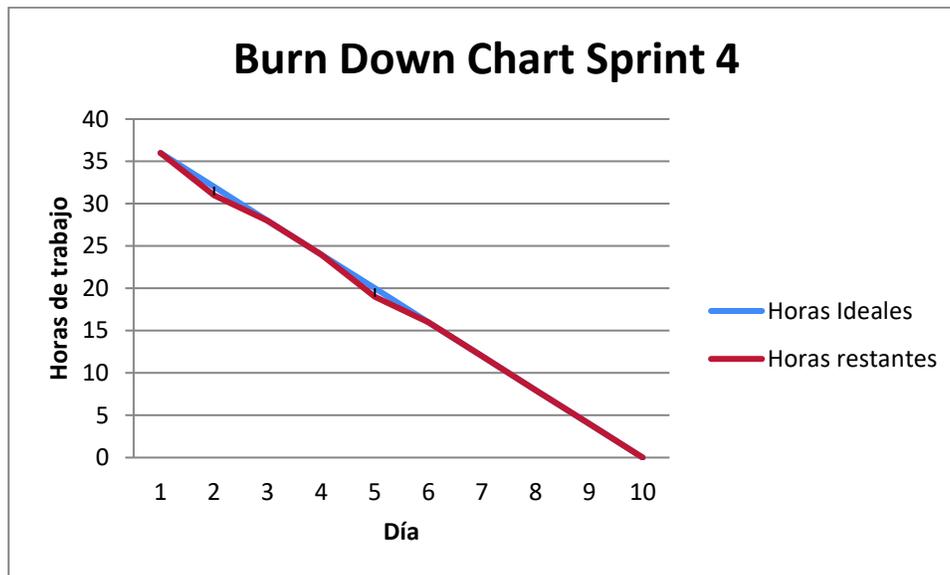


Figura 46. Burn down Chart spring 4.

#### 3.10.4.4. Sprint Review

En este sprint se concluyó con la creación del layout detalle restaurante, y se implementó funcionalidad importante como la actualización de la posición en el mapa, lo que hace que cada vez que el dispositivo cambia de lugar, la marca que se dibuja en el mapa haciendo referencia al dispositivo móvil, cambie su posición.

Los criterios de aceptación de este sprint se los observa en la Tabla 15.

**Tabla 15. Criterios de aceptación Sprint 4**

<b>Código</b>	<b>Historia de Usuario</b>	<b>Criterios de aceptación</b>	<b>Tarea</b>	<b>Aceptado /Rechazado</b>	<b>Comentarios</b>
HU4	Yo como usuario, una vez seleccionado el restaurante al que deseo dirigirme, quiero navegar a través del GPS hacia dicho restaurante.	Observar en un mapa, tanto la posición en la que se encuentra el usuario como las marcas de los lugares en donde se encuentran los restaurantes.	Implementar funcionalidad de posición en el mapa	Aceptado	
HU1	Yo como usuario quiero buscar un restaurante a través del nombre	Observar una lista con los nombres de los restaurantes que coincidan con el nombre de la búsqueda, además de la distancia hasta el mismo, en un tiempo máximo de 5s	Creación actividad búsqueda por Nombre Creación layout búsqueda	Aceptado	
HU5	Yo como usuario quiero observar una lista con los restaurantes visitados.	Observar en una lista los restaurantes visitados anteriormente.	Creación actividad historial Creación layout historial	Aceptado Aceptado	

### **3.10.4.5. Sprint Retrospective**

Este sprint finaliza el desarrollo del prototipo, con el concluye la aplicación de la metodología.

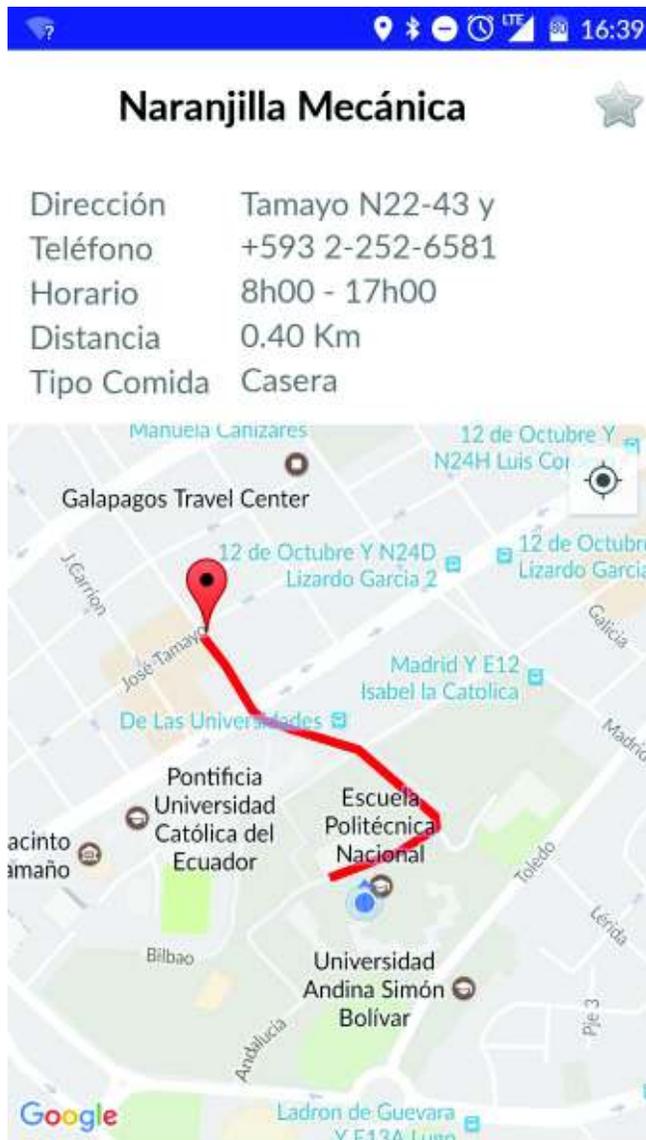
## **3.11. Ejemplo del uso de la aplicación**

Luego de instalar la aplicación, y conceder todos los permisos requeridos para su correcto funcionamiento, inicia mostrando un mapa con la ubicación actual del dispositivo y una marca de cada restaurante cercano. Esto se puede observar en la Figura 47.



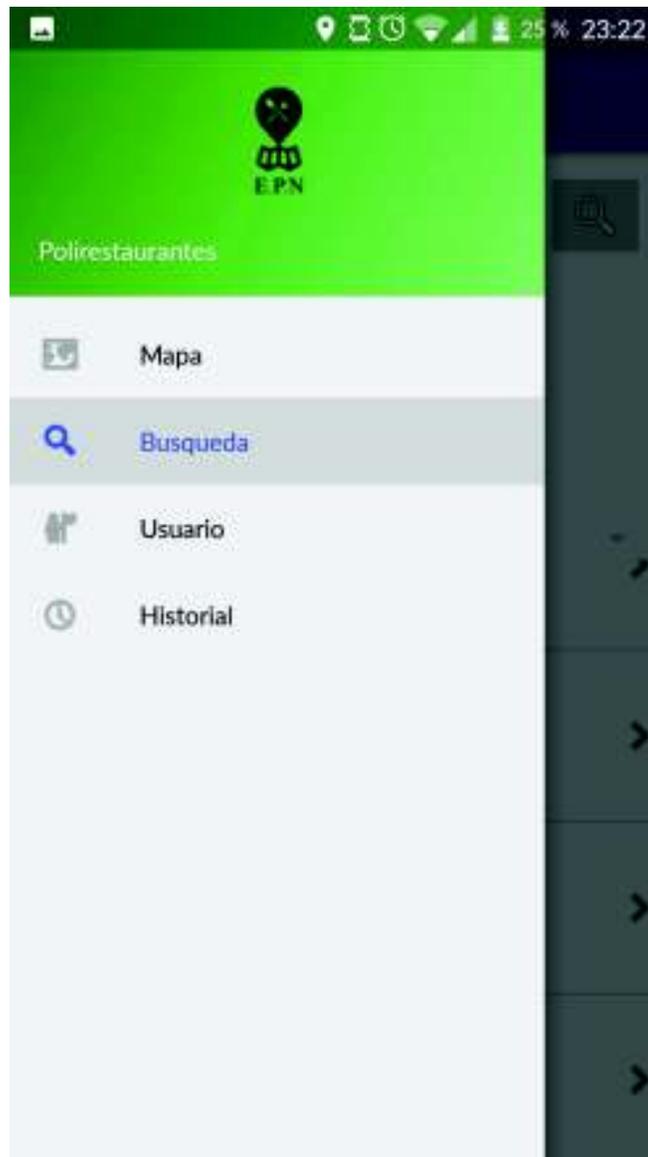
**Figura 47. Mapa inicial Polirestantes.**

Al dar clic sobre el marcador del restaurante se puede observar el nombre del mismo, y al tocar dicho nombre se procede a mostrar la información del restaurante. En la actividad Detalle restaurante se observa un mapa, en el cual se pinta una línea roja que marca la ruta más corta al restaurante. Este detalle se lo observa en la Figura 48.

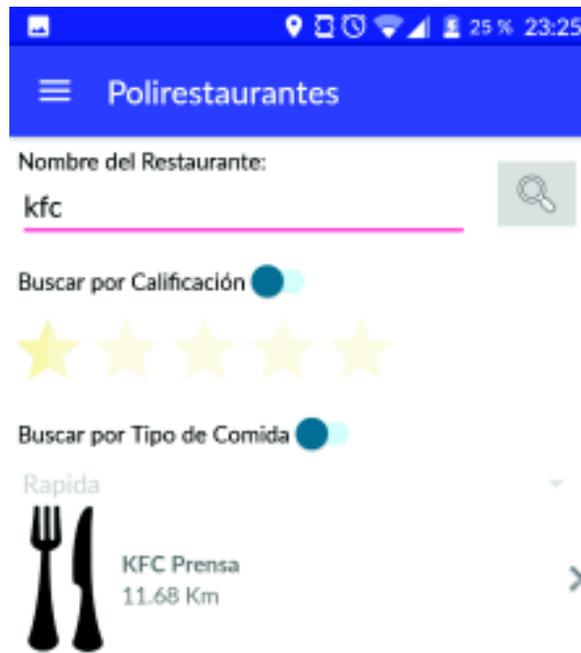


**Figura 48. Detalle restaurante.**

Al seleccionar el menú de la aplicación se presentan los diferentes módulos que contiene el prototipo, estos se observan en la Figura 49. Se selecciona el módulo de “Búsqueda” y se puede apreciar las opciones de búsqueda con las que cuenta el prototipo, se procede a ingresar un nombre y buscar, el resultado se muestra en la Figura 50.



**Figura 49. Menú del Prototipo móvil.**



**Figura 50. Búsqueda por nombre.**

Además de realizar la búsqueda de un restaurante por su nombre, se puede buscar por la calificación que se le ha concedido, o por el tipo de comida con el que cuenta, los resultados de dichas búsquedas se las presentan en la Figura 51 y Figura 52 respectivamente.

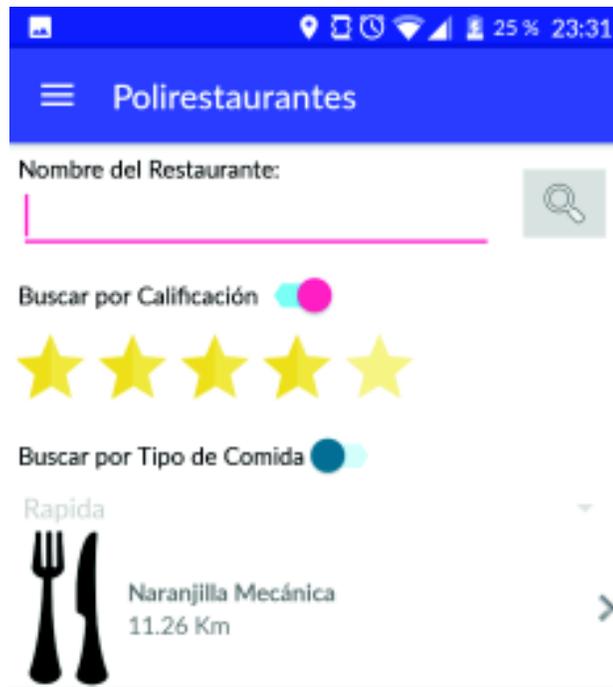


Figura 51. Búsqueda por calificación

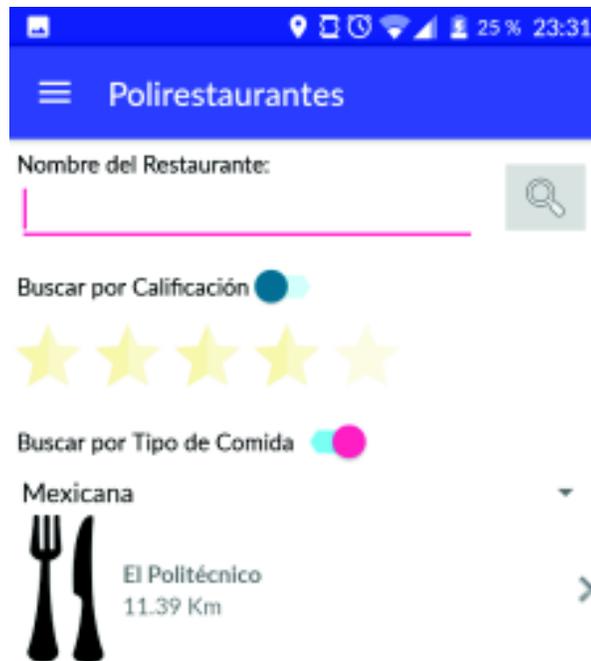


Figura 52. Búsqueda por tipo de comida.

Si sobre el menú que se observa en la Figura 46 se selecciona el módulo “Historial”, se despliega sobre la pantalla del dispositivo un listado de los restaurantes visitados ordenados de mayor a menor en el número de visitas. Un ejemplo de esto se lo puede observar en la Figura 53.

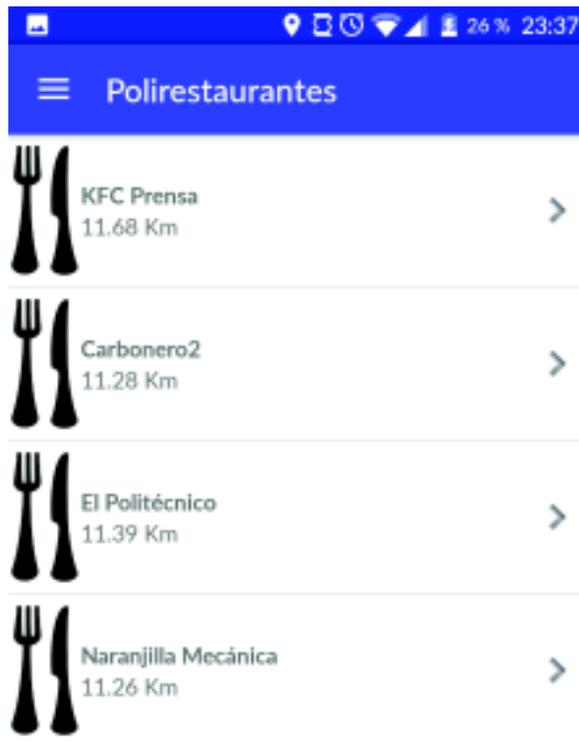


Figura 53. Historial.

### 3.12. Análisis de resultados

Se solicitó a 10 usuarios probar el prototipo de aplicación móvil y que respondan una encuesta para evaluar la satisfacción del usuario, usabilidad y que tan amigable al usuario resultó el prototipo diseñado. La encuesta fue diseñada y distribuida a través de un formulario de Google, al cual se puede tener acceso siguiendo el link: <https://goo.gl/forms/oWSBeeevcKfrXyQL2>, además podemos encontrar las preguntas de la encuesta en el Anexo III.

Luego de aplicada la encuesta se realiza una tabulación de resultados que se presentan a continuación.

### 3.12.1. ¿El prototipo de aplicación móvil mostró restaurantes para ser visitados?

Se realizó esta pregunta con la finalidad de saber si en algún momento no se presentaron datos a los usuarios.

Los resultados muestran que al total de los usuarios se les presentó información de los restaurantes. Este resultado se lo puede observar en la Figura 54.

¿El prototipo de aplicación móvil mostró restaurantes para ser visitados?  
(10 respuestas)

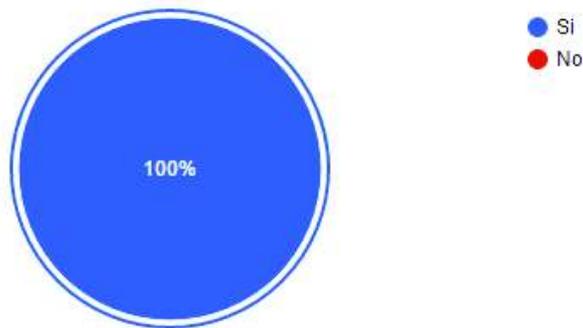


Figura 54. Resultado pregunta 1

### 3.12.2. De acuerdo con su experiencia con la aplicación Polirestaurantes, ¿Qué tan necesario le parece el uso de estas aplicaciones?

El resultado de esta pregunta fue que para el 60% de los encuestados le parece necesario el uso de aplicaciones que les ayuden en su cotidianidad, y al 20% le pareció muy necesario el uso de aplicaciones como el prototipo desarrollado. Este resultado se lo puede observar en la Figura 55.

De acuerdo a su experiencia con la aplicación Polirestantes, ¿Qué tan necesario le parece el uso de estas aplicaciones?.

(10 respuestas)

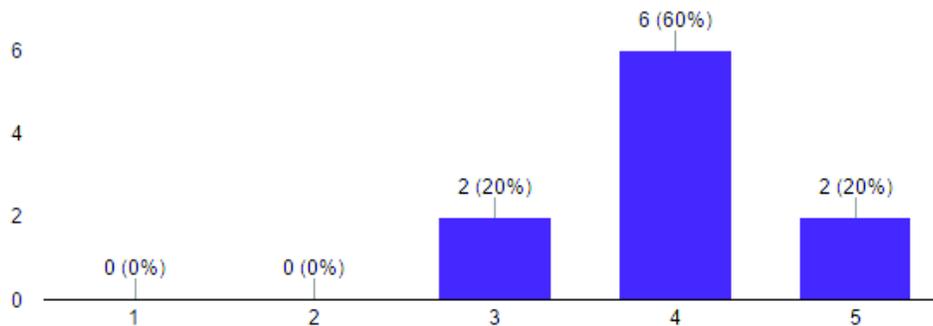


Figura 55. Resultado pregunta 2

### 3.12.3. ¿Qué tan fácil de usar le pareció el prototipo de aplicación móvil?

El resultado muestra que para el 40% de los encuestados, el prototipo de aplicación móvil le pareció de fácil uso, y al 30% de los encuestados le pareció muy fácil su uso. Este resultado se lo puede observar en la Figura 56.

¿Qué tan fácil de usar le pareció el prototipo de aplicación móvil? (10 respuestas)

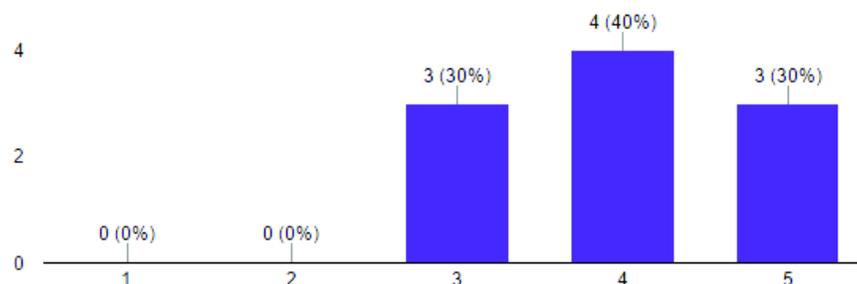


Figura 56. Resultado pregunta 3

### 3.12.4. ¿La información obtenida en la aplicación fue clara al momento de encontrar el restaurante?

El resultado de esta pregunta muestra que para el 30% de los encuestados tiene cierta inconformidad con la información prestada de los restaurantes, se explicó a dichos usuarios, que la limitante de la información presentada se debe a que es solo un prototipo de aplicación, no una aplicación móvil ya terminada. Este resultado se lo puede observar en la Figura 57.

¿La información obtenida en la aplicación fue clara al momento de encontrar el restaurante?

(10 responses)

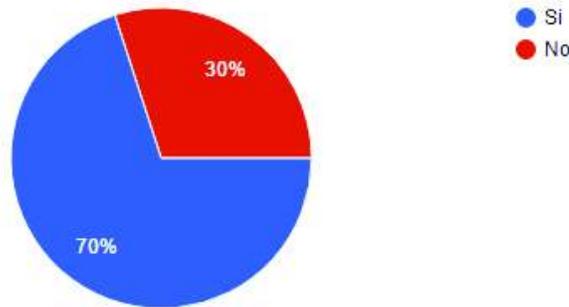
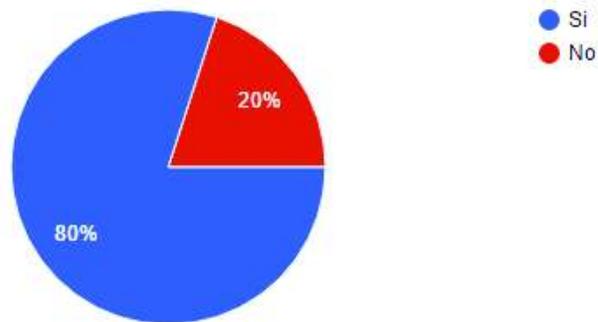


Figura 57. Resultado pregunta4

### 3.12.5. ¿La ruta que sugirió la aplicación fue correcta?

El 80% de las personas encuestadas informó a través de los resultados de esta pregunta que existieron ciertos errores en los cálculos de la ruta y la distancia desde la localización del usuario hacia el restaurante seleccionado. Este resultado se lo puede observar en la Figura 58.

¿La ruta que sugirió la aplicación fue correcta? (10 responses)



**Figura 58. Resultado pregunta 5**

**3.12.6. ¿Los restaurantes presentados en la aplicación estuvieron de acuerdo con sus intereses de búsqueda?**

Para el 80% de las personas que usaron el prototipo de aplicación móvil los resultados muestran que los intereses del usuario fueron identificados correctamente, para luego poder presentar o sugerir restaurantes de acuerdo a estos. Este resultado se puede observar en la Figura 59.

¿Los restaurantes presentados en la aplicación estuvieron de acuerdo a sus intereses de búsqueda?

(10 responses)

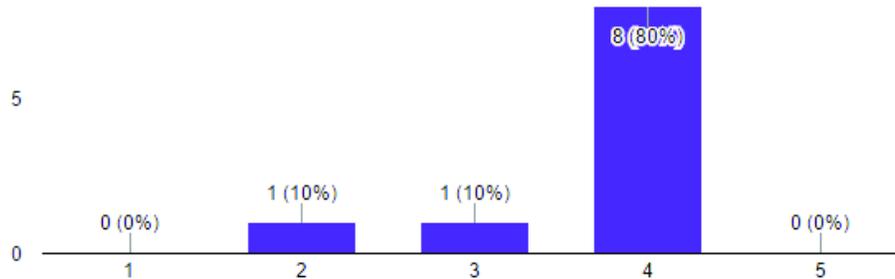


Figura 59. Resultado pregunta 6

## CONCLUSIONES

- Se construyó un prototipo de aplicación móvil, el cual presenta los restaurantes cercanos a la ubicación del dispositivo, basándose en los intereses del usuario.
- Para identificar los intereses del usuario, el prototipo de aplicación móvil almacena en la base de datos los restaurantes visitados con anterioridad y los presenta de acuerdo al número de visitas realizadas. Almacena restaurantes marcados como favoritos por el usuario, y posee un mecanismo de búsqueda por el tipo de comida, o calificación de dicho restaurante.
- Al desarrollar una aplicación en un ambiente nativo, se cuenta con ventajas como el acceso fácil a los sensores de los dispositivos, ejemplo la brújula y GPS con los cuales se logra geo posicionar el dispositivo.
- Al utilizar el marco de trabajo SCRUM se puede iniciar el desarrollo del prototipo de aplicación móvil, con la idea general de los requerimientos, e ir adecuando dichos requisitos a lo largo de dicho desarrollo.
- La entrega de información en tiempo real permite brindar mayores prestaciones a los usuarios, pero limita el funcionamiento de la aplicación sin una conexión a internet.

- Las ontologías pueden ayudar en el diseño de una base de datos relacional, pues contienen el conocimiento de un área de interés, por lo que se puede basar en ella para el diseño del modelo, entidad relación.
- Al utilizar una ontología adecuada, para la construcción del diagrama de una base de datos, se está utilizando el conocimiento reunido en ocasiones anteriores, por lo que se tiene un ahorro de tiempo en dicho diseño.
- Al utilizar una ontología en conjunto con una metodología ágil, se consigue un ahorro de tiempo considerable en el desarrollo del proyecto, pues se puede extender más allá de la reutilización de código, sino también a la reutilización del conocimiento que almacena una ontología.
- Se debe aplicar metodologías ágiles en proyectos de desarrollo de software, en los cuales los requisitos iniciales no se encuentran definidos o entendidos con claridad, o en los que los requisitos del negocio cambian con frecuencia.
- Antes del uso de una metodología ágil se capacite a todos los involucrados, con el fin de garantizar el entendimiento, aceptación y fácil aplicación de dicha metodología.

## RECOMENDACIONES

- Se recomienda el uso de inteligencia artificial, a través de algoritmos de aprendizaje, para que las respuestas sean lo más adecuadas al usuario.
- Se recomienda la vinculación de redes sociales al prototipo, para obtener mayor información de los intereses del usuario, logrando con esto un mejor entendimiento y presentar resultados más acertados.
- Se recomienda presentar mayor información sobre tipos de comida que ofrecen los establecimientos, para que el usuario pueda tener una óptima confiabilidad en la información que brinda el prototipo de aplicación móvil.
- Se recomienda que cuando se ingrese al detalle del restaurante y se dé clic sobre el nombre, se pueda visualizar la página web de dicho restaurante.
- Se recomienda aplicar un protocolo de seguridad para evitar la lectura de datos del usuario, además de protección de la información en la nube.
- Se recomienda el uso de cache en el servidor web, con el fin de mejorar el tiempo de respuesta del servidor web.

## BIBLIOGRAFÍA

- [1] G. M. Vivian y V. Jean, «uclouvain,» 02 2013. [En línea]. Available: [https://www.uclouvain.be/cps/ucl/doc/ssh-ilsim/images/Motti\\_WP\\_02-2013.pdf](https://www.uclouvain.be/cps/ucl/doc/ssh-ilsim/images/Motti_WP_02-2013.pdf). [Último acceso: 11 07 2016].
- [2] J. Larrea y J. Riofrío, «Inec,» Inec, 16 Mayo 2014. [En línea]. Available: <http://www.ecuadorencifras.gob.ec/12-millones-de-ecuatorianos-tienen-un-telefono-inteligente-smartphone/>. [Último acceso: 9 Julio 2015].
- [3] J. A. Guzmán Luna, M. López Bonilla y I. Durley Torres, «Metodologías y métodos para la construcción de ontologías,» Pereira, 2012.
- [4] N. F. Noy y D. L. McGuinness, «Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología,» 2005.
- [5] A. Tello Lozano, «Ontologías en la Web Semántica,» NA, 2010.
- [6] G. Aranda y F. Ruiz, «Clasificación y ejemplos del uso de ontologías en Ingeniería del Software,» Buenos Aires, 2005.
- [7] A. Quero y I. Besembel, Definición de una ontología para la guía de conocimiento SWEBOK, Mérida, 2007.
- [8] V. Egüez, Desarrollo de un sistema de ubicación geografico para la busqueda de información para la realizacion de estudios de cuarto nivel en las universidades nacionales y extranjeras aprobadas por la SENESCYT usando una ontologia, Quito, 2013.
- [9] G. Aranda, «Clasificación y ejmeplos de ontologias en ingeniría de software,» Buenos Aires, 2005.
- [10] Maxxor, «SCRUM Software Development Process,» Maxxor, 2016. [En línea]. Available: <https://www.maxxor.com/software-development-process>. [Último acceso: 28 06 2016].

- [11] Ubisworld, «Ubisworld,» Ubisworld, [En línea]. Available: <http://ubisworld.org/index.php>. [Último acceso: 17 Julio 2015].
- [12] Anónimo, «Desarrollo de Ontologías,» Barcelona, 2015.
- [13] J. L. Cendejas Valdéz, «Modelos y metodologías para el desarrollo de software,» 04 2014. [En línea]. Available: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>. [Último acceso: 10 10 2016].
- [14] Scrum Study, SMC Work Book, 2014.
- [15] J. Guerrero, Aplicación Móvil para la gestión de información de cines en la ciudad de Quito, Quito, 2014.
- [16] Google Developers, «Developer Android,» Google, [En línea]. Available: <https://developer.android.com/tools/studio/index.html>. [Último acceso: 10 10 2015].
- [17] Apache Friends, «XAMPP Apache + MariaDB + PHP + Perl,» [En línea]. Available: <https://www.apachefriends.org/index.html>. [Último acceso: 10 10 2015].
- [18] Oracle Corporation and/or its affiliates , «MySQL,» [En línea]. Available: <http://dev.mysql.com/doc/relnotes/mysql/5.7/en/>. [Último acceso: 24 1 2016].
- [19] C. Thibau, «Google Books,» Ediciones ENI, Mayo 2006. [En línea]. Available: <https://books.google.com.ec/books?id=wY0bHPmW-NUC&printsec=frontcover#v=onepage&q&f=false>. [Último acceso: 23 05 2016].
- [20] The PHP Group, «PHP,» [En línea]. Available: <http://php.net/manual/es/intro-whatism.php>. [Último acceso: 24 10 2015].
- [21] The Apache Software Foundation., [En línea]. Available: <https://httpd.apache.org/>. [Último acceso: 26 10 2015].

- [22] Stanford University, «Protégé,» [En línea]. Available: <http://protege.stanford.edu/products.php>. [Último acceso: 27 10 2015].
- [23] X. M. Ñauñay Puente y M. C. Toaquiza Luna, *Identificación de intereses del usuario desde la web social, aplicado a un caso de estudio*, Quito, 2016.
- [24] W. Dr. Siricharoen, «Ontology Modeling and Object Modeling in Software Engineering,» 2009.
- [25] I. Sommerville, *Ingeniería del software*, Séptima edición, Madrid: Pearson Educación SA, 2005.
- [26] F. A. Amo, L. Normand Martínez y J. Segovia Perez, *Introducción a la ingeniería de software: Modelo de desarrollo de programas*, Zaragoza: Grefol, SA, 2005.
- [27] E. Vysniauskas y N. Lina, «TRANSFORMING ONTOLOGY REPRESENTATION FROM OWL TO RELATIONAL DATABASE,» *INFORMATION TECHNOLOGY AND CONTROL*, vol. 35, nº 3.A, p. 11, 2006.
- [28] HERE Sdk, «HERE,» 2016. [En línea]. Available: <https://developer.here.com/develop/mobile-sdks>. [Último acceso: 11 07 2016].
- [29] T. Rodríguez y J. Aguilar, «Construcción de una Ontología OWL con protégé 4,» Mérida, 2012.
- [30] M. Stoica, M. Mircea y B. Ghilic-Micu, «Software Development: Agile vs. Traditional,» *Informatica Economică*, pp. 64-76, 2013.
- [31] K. Schwaber y J. Sutherland, «The Scrum Guide,» 2014. [En línea]. Available: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>.
- [32] M. Báez, Á. Borrego, J. Cordero, L. Cruz, M. González y F. Hernández, *Introducción a Android*, Madrid: Creative Commons la Universidad Complutense de Madrid.
- [33] S. Lujan, «Accesibilidad Web,» [En línea]. Available: <http://accesibilidadweb.dlsi.ua.es/?menu=fu-semantica>. [Último acceso: 18 07 2015].

- [34] J. Capdevila, «Servicio de Alerta,» 21 Mayo 2012. [En línea]. Available: [http://docdigital.typepad.com/servicio\\_de\\_alerta/2012/05/google-evolucionahacia-el-buscador-sem%C3%A1ntico.html](http://docdigital.typepad.com/servicio_de_alerta/2012/05/google-evolucionahacia-el-buscador-sem%C3%A1ntico.html). [Último acceso: 17 Julio 2015].
- [35] B. Semánticos. [En línea]. Available: <https://sites.google.com/site/buscadoressemanticos/-ques-un-buscador-semantico>. [Último acceso: 18 07 2015].
- [36] B. Kent, B. Mike, v. B. Arie, C. Alistair, C. Ward, F. Martin, G. James, H. Jim, H. Andrew, J. Ron, K. Jon, M. Brian, C. M. Robert, M. Steve, S. Ken y S. Jeff, «Manifiesto por el Desarrollo Ágil de Software,» [En línea]. Available: <http://agilemanifesto.org/iso/es/>. [Último acceso: 03 07 2015].
- [37] H. Kniberg, Scrum y XP desde las trincheras, Estados Unidos de America: C4media, 2007.
- [38] Don Ho 2015, «Notepad++,» [En línea]. Available: <https://notepad-plus-plus.org/>. [Último acceso: 27 10 2015].
- [39] Oracle Corporation and/or its affiliates , «MySQL,» [En línea]. Available: <https://www.mysql.com/products/workbench/>. [Último acceso: 26 1 2016].
- [40] M. Cohn, «Agile Succeeds Three Times More Often Than Waterfall,» [En línea]. Available: <https://www.mountangoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>. [Último acceso: 10 07 2016].
- [41] A. Tores, «sasa,» 121 wewe 1212. [En línea]. Available: [ww.aksakls.com](http://ww.aksakls.com).
- [42] E. Misuft, «Servidor web Apache,» 2012. [En línea]. Available: <http://descargas.pntic.mec.es/mentor/visitas/Apache.pdf>. [Último acceso: 21 Julio 2015].

## ANEXOS

### Anexo I. Historias de usuario.

<b>Historia de usuario</b>	
<b>Número:</b> 1	<b>Nombre:</b> búsqueda de restaurante por nombre
<b>Usuario:</b> usuario general	
<b>Modificación de historia número:</b>	<b>Iteración asignada:</b>
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 20
<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b> Yo como usuario quiero buscar un restaurante a través del nombre.	
<b>Criterios de evaluación:</b> Observar una lista con los nombres de los restaurantes que coincidan con el nombre de la búsqueda, además de la distancia hasta el mismo.	

<b>Historia de usuario</b>	
<b>Número:</b> 2	<b>Nombre:</b> Detalles del Restaurante
<b>Usuario:</b> usuario general	
<b>Modificación de historia número:</b>	<b>Iteración asignada:</b>
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 20
<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b> Yo como usuario quiero observar información relevante del restaurante que he seleccionado, además del mapa mostrando la zona donde se ubica dicho restaurante	

**Criterios de evaluación:** Observar detalles del restaurante, información, teléfono, horarios de atención, entre otros. Además, observar un mapa con la ubicación de dicho restaurante.

Historia de usuario	
<b>Número:</b> 3	<b>Nombre:</b> Búsqueda de restaurante por ubicación
<b>Usuario:</b> usuario general	
<b>Modificación de historia número:</b>	<b>Iteración asignada:</b>
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 30
<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b> o como usuario quiero observar en el mapa mi posición, y a través de esta los restaurantes cercanos.	
<b>Criterios de evaluación:</b> Observar en un mapa las marcas de la posición del usuario y el o los restaurantes más cercanos al usuario. Luego de analizar el interés del usuario.	

Historia de usuario	
<b>Número:</b> 4	<b>Nombre:</b> Navegar hacia el restaurante
<b>Usuario:</b> usuario general	
<b>Modificación de historia número:</b>	<b>Iteración asignada:</b>
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 20
<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b> Yo como usuario, una vez seleccionado el restaurante al que deseo dirigirme, quiero navegar a través del GPS hacia dicho restaurante.	
<b>Criterios de evaluación:</b> Observar en un mapa, tanto la posición en la que se encuentra el usuario como las marcas de los lugares en donde se encuentran los restaurantes	

<b>Historia de usuario</b>	
<b>Número:</b> 5	<b>Nombre:</b> Historial de restaurantes visitados
<b>Usuario:</b> usuario general	
<b>Modificación de historia número:</b>	<b>Iteración asignada:</b>
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 10
<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b> Yo como usuario quiero observar una lista con los restaurantes visitados.	
<b>Criterios de evaluación:</b> Observar en una lista los restaurantes visitados anteriormente.	

<b>Historia de usuario</b>	
<b>Número:</b> 6	<b>Nombre:</b> Creación de backend
<b>Usuario:</b> desarrollador	
<b>Modificación de historia número:</b>	<b>Iteración asignada:</b>
<b>Prioridad en Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 10
<b>Riesgo en Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b> Yo como desarrollador quiero que se cree el Backend de la aplicación.	
<b>Criterios de evaluación:</b> Observar la base de datos creada. Realizar consultas a los servicios web y observar las respuestas con los datos que se requieren.	

## Anexo II. Script creación de base de datos.

```
CREATE TABLE IF NOT EXISTS `historial` (  
  `idHistorial` int(11) NOT NULL AUTO_INCREMENT,  
  `Persona_idPersona` int(11) NOT NULL,  
  `Restaurante_idRestaurante` int(11) NOT NULL,  
  `esFavorito` tinyint(1) DEFAULT NULL,  
  `fechaHistorial` datetime NOT NULL,  
  `numeroVisitas` int(11) NOT NULL,  
  PRIMARY KEY (`idHistorial`),  
  KEY `fk_Historial_Persona1_idx` (`Persona_idPersona`),  
  KEY `fk_Historial_Restaurante1_idx` (`Restaurante_idRestaurante`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

-----

--

-- Estructura de tabla para la tabla `persona`

--

```
CREATE TABLE IF NOT EXISTS `persona` (  
  `idPersona` int(11) NOT NULL AUTO_INCREMENT,  
  `cedulaPersona` varchar(13) NOT NULL,  
  `nombrePersona` varchar(45) NOT NULL,  
  `apellidoPersona` varchar(45) NOT NULL,  
  `direccionPersona` varchar(45) DEFAULT NULL,  
  `correoPersona` varchar(45) DEFAULT NULL,
```

```
`telefonoPersona` varchar(45) DEFAULT NULL,  
`generoPersona` varchar(45) DEFAULT NULL,  
PRIMARY KEY (`idPersona`),  
UNIQUE KEY `cedulaPersona_UNIQUE` (`cedulaPersona`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=7 ;
```

```
-----
```

```
--
```

```
-- Estructura de tabla para la tabla `restaurante`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `restaurante` (  
  `idRestaurante` int(11) NOT NULL AUTO_INCREMENT,  
  `nombreRestaurante` varchar(45) NOT NULL,  
  PRIMARY KEY (`idRestaurante`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=6 ;
```

```
-----
```

```
--
```

```
-- Estructura de tabla para la tabla `restaurante_tipocomida`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `restaurante_tipocomida` (  
  `TipoComida_idTipoComida` int(11) NOT NULL,  
  `Restaurante_idRestaurante` int(11) NOT NULL,  
  KEY `fk_Restaurante_TipoComida_TipoComida1_idx` (`TipoComida_idTipoComida`),
```

```
KEY `fk_Restaurante_TipoComida_Restaurante1_idx` (`Restaurante_idRestaurante`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----
```

```
--
```

```
-- Estructura de tabla para la tabla `tipocomida`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tipocomida` (  
  `idTipoComida` int(11) NOT NULL AUTO_INCREMENT,  
  `nombreTipoComida` varchar(45) NOT NULL,  
  `descripcionTipoComida` varchar(200) DEFAULT NULL,  
  PRIMARY KEY (`idTipoComida`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

```
-----
```

```
--
```

```
-- Estructura de tabla para la tabla `ubicaciongeografica`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `ubicaciongeografica` (  
  `idUbicacionGeografica` int(11) NOT NULL AUTO_INCREMENT,  
  `latitudUbicacionGeografica` varchar(20) NOT NULL,  
  `longitudUbicacionGeografica` varchar(20) NOT NULL,  
  `calificacionRestaurante` int(11) NOT NULL DEFAULT '1',  
  `Restaurante_idRestaurante` int(11) NOT NULL,
```

```

`horarioRestaurante` varchar(45) DEFAULT NULL,
`telefonoRestaurante` varchar(45) DEFAULT NULL,
`direccionRestaurante` varchar(45) DEFAULT NULL,
PRIMARY KEY (`idUbicacionGeografica`),
KEY `fk_UbicacionGeografica_Restaurante1_idx` (`Restaurante_idRestaurante`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=7 ;

```

-----

```

--
-- Estructura de tabla para la tabla `usuario`
--

```

```

CREATE TABLE IF NOT EXISTS `usuario` (
  `idUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `passUsuario` tinyblob NOT NULL,
  `idDispositivo` varchar(50) DEFAULT NULL,
  `correoUsuario` varchar(45) NOT NULL,
  `Persona_idPersona` int(11) NOT NULL,
  PRIMARY KEY (`idUsuario`),
  KEY `fk_Usuario_Persona_idx` (`Persona_idPersona`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;

```

```

--
-- Restricciones para tablas volcadas
--
--

```

```

-- Filtros para la tabla `historial`
--
ALTER TABLE `historial`
  ADD CONSTRAINT `fk_Historial_Persona1` FOREIGN KEY (`Persona_idPersona`)
  REFERENCES `persona` (`idPersona`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_Historial_Restaurante1` FOREIGN KEY
  (`Restaurante_idRestaurante`) REFERENCES `restaurante` (`idRestaurante`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Filtros para la tabla `restaurante_tipocomida`
--
ALTER TABLE `restaurante_tipocomida`
  ADD CONSTRAINT `fk_Restaurante_TipoComida_Restaurante1` FOREIGN KEY
  (`Restaurante_idRestaurante`) REFERENCES `restaurante` (`idRestaurante`) ON DELETE
  CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_Restaurante_TipoComida_TipoComida1` FOREIGN KEY
  (`TipoComida_idTipoComida`) REFERENCES `tipocomida` (`idTipoComida`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Filtros para la tabla `ubicaciongeografica`
--
ALTER TABLE `ubicaciongeografica`
  ADD CONSTRAINT `fk_UbicacionGeografica_Restaurante1` FOREIGN KEY
  (`Restaurante_idRestaurante`) REFERENCES `restaurante` (`idRestaurante`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Filtros para la tabla `usuario`
--
ALTER TABLE `usuario`

```

```
ADD CONSTRAINT `fk_Usuario_Persona` FOREIGN KEY (`Persona_idPersona`)
REFERENCES `persona` (`idPersona`) ON DELETE CASCADE ON UPDATE CASCADE;
```

## Anexo III. Encuesta

### Polirestaurantes

Evaluación del prototipo Polirestaurantes

**¿El prototipo de aplicación móvil mostró restaurantes para ser visitados? \***

- Si
- No

**De acuerdo a su experiencia con la aplicación Polirestaurantes, ¿Qué tan necesario le parece el uso de estas aplicaciones?. \***

1 2 3 4 5

---

no necesario      muy necesario

**¿Qué tan fácil de usar le pareció el prototipo de aplicación móvil? \***

1 2 3 4 5

---

nada amigable      muy amigable

**¿La información obtenida en la aplicación fue clara al momento de encontrar el restaurante? \***

- Si
- No

**¿La ruta que sugirió la aplicación fue correcta? \***

- Si
- No

**¿Los restaurantes presentados en la aplicación estuvieron de acuerdo a sus intereses de búsqueda? \***

1 2 3 4 5

---

En desacuerdo      De acuerdo

---