

**ABCHILD PATERNITY GUIDE**

**CAROLINA GALEANO MONSALVE**

**SONIA PATRICIA GAMBOA HINESTROZA**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**

**FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS**

**DE LA COMPUTACIÓN**

**PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**PEREIRA**

**2016**

**ABCHILD PATERNITY GUIDE**

**CAROLINA GALEANO MONSALVE**

**SONIA PATRICIA GAMBOA HINESTROSA**

**PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE  
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

**PhD. GUILLERMO ROBERTO SOLARTE**

**ASESOR DE PROYECTO DE GRADO**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**

**FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS  
DE LA COMPUTACIÓN**

**PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**PEREIRA**

**2016**

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Dedico este proyecto de grado a mi hijo Gabriel que más que el motor de mi vida fue, es y será la parte más importante para culminar esta etapa y espero muchos más logros, a mi familia gracias por entender cada momento sacrificado para ser invertido en el desarrollo de este proyecto.

Carolina Galeano Monsalve

Dedico este logro especialmente a mi madre Mercedes Hinstroza, mi papa José Tomas Gamboa que me acompaña siempre desde el cielo y mis hermanos que siempre están en los momentos que más los necesite.

Sonia Patricia Gamboa

## AGRADECIMIENTOS

Agradezco a Dios que fue el que me permitió culminar con éxito esta hermosa etapa de mi vida, a mis padres Adriana Ma Monsalve y Oscar Galeano por haber forjado la persona que soy hoy en día, porque siempre conté con su apoyo incondicional, a mi hermana Juliana Galeano que estuvo a mi lado inclusive en los momentos y situaciones más tormentosas, siempre ayudándome. No fue sencillo culminar con éxito este proyecto, sin embargo, siempre fuiste muy motivadora y esperanzadora, me decías que lo lograría perfectamente. Mis logros son gracias a ustedes, incluyendo este logro que a pesar de tantos obstáculos al fin lo alcance.

Durante los últimos semestres se presentaron varias situaciones las cuales pudieron llegar a ser en muchos momentos, motivos convincentes para rendirse o tirar la toalla. Pero en esos mismos instantes resaltó dentro de mi vida esa persona que durante muchos días y meses fue mi compañero, mi amigo, mi consejero, mi novio. Gracias por todo Mario Eraso

Gracias a la institución que me dio la bienvenida a esta etapa de mi vida, por las oportunidades de crecer como persona y profesional, por los apoyos brindados, también agradezco mucho por la ayuda de mis maestros, de mis compañeros por lidiar con mis reiniciadas, a mis mejores amigas Luisa Zuluaga, Katherine Ceballos y Alexandra Loaiza por estar siempre conmigo en los momentos más importantes de mi vida, a Julián Piedrahita y Gerson Rodríguez les agradezco de la manera más sincera, e infinitamente por toda la ayuda y el apoyo.

Agradezco a mi compañera de proyecto de grado Sonia, un gusto culminar esta etapa juntas y espero que la vida le traiga muchas bendiciones y al PhD. Guillermo Solarte que nos asesoró y acompañó en la construcción del proyecto.

Carolina Galeano Monsalve

Quiero agradecer a mi familia por el apoyo durante todo este largo y arduo camino, en especial a mi madre Mercedes Hinestroza, que fue mi motor para culminar esta etapa de mi vida y a mis hermanos, resalto Marcelino Gamboa, Emérito Gamboa, Eduardo Gamboa, José Gamboa por su apoyo moral y económicamente durante este largo camino.

Agradecer amigos y docentes que me ayudaron el crecimiento profesional, La verdad tengo muchas personas que agradecer por tantos momentos de alegrías y tristezas, y aunque es muy grande la lista agradezco a todos aquellos futuros ingenieros con los que compartí mi etapa universitaria. Agradecimiento a carolina mi compañera de proyecto de grado, un gusto culminar esta etapa juntas y espero que tenga una buena vida ingeniera y al ingeniero Guillermo Solarte que nos asesoró y acompañó en la construcción del proyecto.

Sonia Patricia Gamboa Hinestroza

## Índice de Contenido

<b>1. Introducción.....</b>	<b>9</b>
<b>2. Capítulo I - Generalidades .....</b>	<b>10</b>
<b>2.1. Planteamiento del Problema .....</b>	<b>10</b>
<b>3. Justificación .....</b>	<b>13</b>
<b>4. Objetivos .....</b>	<b>14</b>
<b>4.1. Objetivo General.....</b>	<b>14</b>
<b>4.2. Objetivos Específicos.....</b>	<b>14</b>
<b>5. Alcance del Prototipo a Desarrollar .....</b>	<b>15</b>
<b>6. Capítulo II Marco Referencial.....</b>	<b>17</b>
<b>6.1 Marco Teórico .....</b>	<b>17</b>
<b>6.1.1 Ingeniería del Software.....</b>	<b>17</b>
.....	Error! Bookmark not defined.
<b>6.1.2. Proceso de Software.....</b>	<b>19</b>
<b>6.1.3. Modelo de Procesos Desarrollo de Software.....</b>	<b>21</b>
<b>6.1.3.1. Modelo por Prototipos.....</b>	<b>21</b>
<b>6.1.3.1.1 Ciclo de Vida de un Sistema Basado en Prototipo.....</b>	<b>23</b>
<b>6.1.4. Metodologías Ágiles .....</b>	<b>24</b>
<b>6.1.4.1. Programación Extrema.....</b>	<b>26</b>
<b>6.1.5. Levantamiento y Análisis de Requerimientos .....</b>	<b>28</b>
<b>6.1.5.1. Historias de Usuario .....</b>	<b>28</b>
<b>6.1.6. Diseño de Software.....</b>	<b>29</b>
<b>6.1.6.1. UML (Lenguaje Unificado de Modelado).....</b>	<b>29</b>
<b>6.1.6.1.1. Casos de Uso.....</b>	<b>33</b>
<b>6.1.6.1.2. Diagrama de Clases .....</b>	<b>33</b>
<b>6.1.6.1.3. Diagrama de Componentes.....</b>	<b>34</b>
<b>6.1.6.1.4. Diagrama de Actividades .....</b>	<b>34</b>
<b>6.1.6.1.5. Diagrama de Despliegue.....</b>	<b>35</b>

6.1.6.2. Modelo Relacional.....	35
6.1.6.3. Arquitectura de Software.....	36
6.1.6.3.1 Modelo vista Controlador .....	36
6.1.7 Codificación.....	38
6.1.8 Pruebas .....	38
6.1.9 Entrega.....	39
6.1.10 Estándares de Web .....	39
6.1.10.1 HTML5.....	39
6.1.10.2 CSS.....	40
6.1.10.3. APP Inventor.....	40
6.1.10.4. Visual Studio .....	41
6.2.3 Vacunas.....	42
6.2.4 Test.....	43
6.2.5 Ciudad Ubicación (Geolocalización).....	43
7. Capítulo III Otras Consideraciones.....	45
7.1. Marco Legal y Normativo .....	45
7.2. Marco Geográfico .....	46
8. Capítulo IV Definiciones y Requerimiento del Sistema.....	47
8.1. Levantamiento de Requerimiento .....	47
8.2. Diseño Del Sistema.....	58
8.2.1. Vista de Escenarios (CASOS DE USOS).....	58
11. Bibliografía .....	70



## **1. Introducción**

Se vive en una época en la cual la tecnología y el uso de las Tic's se ha convertido en una herramienta primordial por esto se diseñará la aplicación ABCChild Paternity Guide para proporcionar información de desarrollo del bebé de acuerdo a su edad, tips de cuidado básicos de la primera infancia, sistema de lectura RSS, opciones de llamadas a centro médicos o familiares en caso de emergencia, tendrá sistema de localización y acceso a mapas de la ciudad donde se encuentra el usuario, para así saber la ubicación de los sitios de interés, como por ejemplo los centros de salud más cercanos, de la ciudad en que se encuentre. Y se concentrará en la elaboración de un proyecto totalmente operativo y eficiente y el manual de usuario. El propósito fundamental de este proyecto es completar la funcionalidad del sistema, clarificando los requerimientos necesarios.

Este documento muestra el análisis y uso de una arquitectura empleada en una aplicación del campo de la salud que está destinada al cuidado de los niños en sus primeros meses de vida, logrando hacer más fácil la vida de los padres primerizos, familiares o cualquier persona que tenga un bebé a su cuidado. Ya en casa, siempre se tiene incertidumbre e inseguridad como padres primerizos y se dan cuenta de que no se tiene ni idea de lo que hay que hacer, enfrentándose a una situación de desconocimientos en el proceso para el cuidado de una forma apropiada. El utilizar una guía de recomendaciones puede ayudar a los padres primerizos a sentirse seguros sobre el tipo de cuidados que requiere un recién nacido en un abrir y cerrar de ojos.

## **2. Capítulo I – Generalidades**

### **2.1. Planteamiento del Problema**

Ser padres es el rol más importante en la existencia de cualquier persona. Significa adoptar nuevas responsabilidades indelegables, y cambiar la forma de vida para siempre. Un problema al que se enfrentan al momento que serán padres, es el desconocimiento de correctos hábitos y cuidados, para que sus hijos alcancen adecuado crecimiento y desarrollo.

Surgió la idea de crear un sistema de información llamado ABCChild Paternity Guide, por la influencia de los celulares y las redes sociales en la vida actual; debido a que la gran mayoría de las personas poseen un dispositivo móvil inteligente y lo utilizan casi todos los días desde que se levantan hasta que se acuestan a dormir. Esta tecnología a permitido mejorar la vida desde el momento que se pueden comunicar con cualquier persona como amigo, familia desde cualquier lugar siempre y cuando tengan señal de teléfono o de internet. Un software tiene varias categorías dentro de las cuales están las aplicaciones conocidas como app, estas aplicaciones son creadas para utilizarlas en los móviles.

Las aplicaciones también llamadas apps están presentes en los teléfonos desde hace tiempo; de hecho, ya estaban incluidas en los sistemas operativos de Nokia o Blackberry años atrás. Los móviles de esa época, contaban con pantallas reducidas y muchas veces no táctiles, y son los que ahora llamamos feature phones, en contraposición a los smartphones, más actuales. [1]

Los padres con tecnología aprovecharán la guía en el cuidado de sus hijos, herramienta para usar en tiempo real que proporcionará nombres, centros médicos cercanos, llamadas de emergencia, consejos, tests de desarrollo cognitivo y motriz, y posiblemente se observen irregularidades que luego serán preguntadas al médico o profesional de la salud.

La aplicación llevará una guía contigo, así se monitorea el crecimiento de tu bebé, el control de vacunas, dosis y su respectivo cronograma, con facilidad afrontarán el proceso de ser padres primerizos y sentir la seguridad como padres experimentados.

Andrés Montes, biólogo dedicado a la producción agroecológica, dice “Los alimentos orgánicos tienen un mejor contenido nutricional, menos porcentaje de agua y son secos; ese contenido de materia seca se relaciona con una mayor cantidad de nutrientes”. [2]

Las futuras madres antes y en la gestación que consumen alimentos saludables, cultivos orgánicos, procrean hijos sanos y dinámicos que enfrentarán los quebrantos de salud, invirtiendo solo lo necesario.

El pañal es uno de los elementos indispensables en el día a día del nuevo bebé. Y modifica los gastos de la canasta familiar en una proporción muy valiosa.

Mensualmente, un 32 por ciento de los hogares gasta entre 50 y 60 mil pesos, siendo este el consumo promedio más alto. Al año, la cifra asciende a un promedio entre 234.000 y 960.000 pesos.

En general, diariamente, los bebés usan 4 pañales, pero esta cifra varía entre 3 y 5, según el estrato. El 1, por ejemplo, es el que menos usa con un promedio de 3.52 unidades; le sigue el

estrato 2 con 3.78 pañales, el 3 con 3.87 y, obviamente, los mayores consumidores pertenecen a los estratos 4, 5 y 6 con un promedio de 4.36 pañales por día. [3]

Por medio de los tips de la app ABChild Paternity Guide se reducirá la cantidad de pañales que son desechados en los botaderos certificados, se pretende dar uso racional a los pañales optimizando el uso y compra de estos.

### **3.Justificación**

La implementación de la aplicación está basada en el campo de la salud que está destinada al cuidado de los niños en sus primeros meses de vida, logrando hacer más fácil la vida de los padres primerizos, familiares o cualquier persona que tenga un bebé a su cuidado. La aplicación ABChild Paternity Guide proporcionará información de desarrollo, tips de cuidado, sistema de lectura RSS, opciones de llamadas a centro médicos o familiares en caso de emergencia, tendrá sistema de localización y acceso a mapas de la ciudad donde se encuentra el usuario, para así saber la ubicación de los sitios de interés, como por ejemplo los centros de salud más cercanos, de la ciudad en que se encuentre.

## **4. Objetivos**

### **4.1. Objetivo General**

Crear un sistema de información para padres primerizos (ABChild PATERNITY GUIDE)

### **4.2. Objetivos Específicos**

- Recolectar información acerca de los tipos de enfermedades que se presenta en etapas tempranas en los niños recién nacidos de 1 a 2 años
  - Definir los requerimientos de software y hardware para la implementación del sistema de gestión “ABChild Paternity Guide”,
  - Investigar sobre las herramientas actuales tanto públicas como privadas.
  - Realizar un análisis comparativo con la herramienta creada.
  - Aplicar las etapas de la arquitectura de sistema para el desarrollo de la aplicación
- Desarrollar un sistema de información para padres primerizos (ABChild Paternity Guide).
- Realizar pruebas y validación del sobre el software desarrollado.

## 5. Alcance del Prototipo a Desarrollar

Este documento es un desarrollo de análisis y procesos para lograr con éxito la construcción del software dedicado a la gestión de ayudar los padres primerizos, familiares o cualquier persona que tenga un bebé a su cuidado, para llevar a cabo con facilidad las primeras meses de vida, utilizando metodologías para el levantamientos de requerimientos para conocer la elaboración de la aplicación ABChild Paternity Guide, el seguimiento de del desarrollo del bebé por medio de test y tips.

El control de manejo sobre las consultas y búsqueda de test y tips todo desarrollado por técnica de la ingeniería de software utilizando como implementación la metodología programación extrema (HTML), vista 4+1, el diseño de un plan de pruebas y los manuales necesarios para mejorar el rendimiento de los usuarios. Este aplicativo se compone de los siguientes módulos de funcionalidad y sus respectivas funciones

- **Módulo de gestión de usuarios.** Se encarga de la autenticación de los usuarios y de la administración de sus respectivos perfiles.
- **Módulo de gestión de correos.** Se utiliza para enviar un correo con la información de la encuesta realizada
- **Módulo de geolocalización.** Se encarga de geolocalizarse en una ciudad o en un lugar geográfico exacto (normalmente representado por un mapa en la web) determinado por unas coordenadas

- **Módulo de llamadas.** Lo utiliza para realizar llamadas a su médico de cabecera, farmacias, centro asistencial
- **Visualización de información de vacunas.** Les permitirá a los usuarios visualizar la información sobre las distintas vacunas que se deben aplicar en el momento correspondiente
- **Conexión con redes sociales.** La utilizan para tener una conexión con las diferentes redes sociales



## **6. Capítulo II Marco Referencial**

### **6.1 Marco Teórico**

En este capítulo se mostrarán cada uno de los conceptos, procedimientos y herramientas que se utilizaron en la construcción del aplicativo del sistema (lenguajes de programación, técnicas de desarrollo, metodologías entre otros).

#### **6.1.1 Ingeniería del Software**

Que es el software?

El software de una computadora es una utilidad que construye los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Contienen programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, incluido contenido que se presenta a medida de que se ejecutan los programas de cómputo e información característica tanto en una copia dura como en formatos virtuales que incluyen virtualmente a cualesquiera medios electrónicos. La ingeniería de software está formada por un proceso, un agregado de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad.[4]

En la actualidad, el software tiene un papel dual. Es un producto y al mismo tiempo es el vehículo para entregar un producto. En su forma de producto, brinda el potencial de cómputo incorporado en el hardware del cómputo o, con más amplitud, en una red de computadoras a las que se accede por medio de un hardware local. Ya sea que resida en un teléfono móvil u opere en el interior de una computadora central, el software es un transformador de

información -produce, administra, adquiere, modifica, despliegue o transmite información que puede ser tan simple como un solo bit o tan compleja como una presentación con multimedios generada a partir de datos obtenidos de decenas de fuentes independientes-. Como vehículo utilizado para distribuir el producto, el software actúa como la base para el control de la computadora (sistemas operativos), para la comunicación de información (redes) y para la creación y control de otros programas (herramientas y ambientes de software).

El software distribuye el producto más importante de nuestro tiempo: información. Transforma los datos personales (por ejemplo, las transacciones financieras del individuo) de modo que puedan ser más útiles en un contexto local, administra la información de negocios para mejorar la competitividad, provee una vía para las redes mundiales de información (el internet) y brinda los medios para obtener la información en todas sus formas [5]

¿Que es la ingeniería del software?

La ingeniería de software es una disciplina del ares de la ingeniería que comprende todos los aspectos de la producción del software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases claves:

I. Método de ingeniería. Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando existan teorías y

métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta restricciones

II. Todos los exteriores de producción del software. La ingeniería del software no sólo comprende los procesos técnicos de desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y desarrollo de herramientas, métodos y teorías de apoyo de producción de software[6]

### **6.1.2. Proceso de Software**

Un proceso es un conjunto de actividades, operaciones y labores que se producen cuando va a establecer cierto producto del trabajo. Una actividad busca lograr un objetivo capaz a desarrollar sin importar el dominio de la aplicación, dimensión del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería del software. Una acción es un acumulado de tareas que producen un producto importante de trabajo. Una tarea se ajusta a un objetivo pequeño pero bien definido (por ejemplo, ejecutar una prueba unitaria) que produce un resultado notorio. .

La estructura del proceso establece el fundamento el proceso completo de la ingeniería de software por medio de la caracterización de un número pequeño de actividades organizadas que sean adaptables a todos los proyectos de software, sin importar su tamaño o complejidad. Una estructura del proceso general para la ingeniería de software consta de cinco actividades:

**Comunicación.** Al comenzar cualquier trabajo técnico, tiene importancia comunicarse y colaborar con el cliente (usuarios y otros involucrados). Se busca entender los objetivos de los involucrados respecto al proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.

**Planeación.** Cualquier desplazamiento que se emprende se reduce si existe un mapa. Un proyecto de software es un desplazamiento difícil, y la planeación crea un “mapa” que guía al equipo mientras se desplaza. El mapa -llamado plan del proyecto de software- concreta el trabajo de ingeniería de software al describir las labores sistemáticas por ejecutar, los riesgos probables, los recursos que se pretenden, los productos del trabajo que se lograrán y una programación de las actividades.

**Modelado.** Crea un bosquejo del objeto para entender la perspectiva general. Si se pretende, refine el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y cómo resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender con facilidad los requerimientos del software y el diseño.

**Construcción.** Esta actividad acoge la generación de código (ya sea manual o automatizada) y las pruebas que se demandan para descubrir errores en este.

**Despliegue.** El software como objeto completar o como un agregado parcialmente ejecutado se entrega al diseñador que lo evalúa, le da retroalimentación, que se basa en dicha evaluación. [7]

### **6.1.3. Modelo de Procesos Desarrollo de Software**

#### **6.1.3.1. Modelo por Prototipos**

El prototipo está basado en el software a realizar, no un producto de ingeniería en sí. Aunque el prototipo puede crear falsas expectativas al cliente en sentido de que es un producto consolidado.

Un prototipo es un programa, que es usado para realizar una definición más rápida de los requisitos de software por construir surgieron los prototipos como un método. El punto de partida es hacer un modelo de la aplicación y presentarla al cliente, sobre todo su nivel de interfaces y otras salidas. El cliente realizará sus observaciones acerca de lo que ve en ese modelo, y el programador de acuerdo con dichas observaciones llevará a cabo las modificaciones de ese modelo. Se repite este proceso hasta alcanzar cubrir todos los requerimientos del producto de un software, Para luego pasar a construir la aplicación.

Es importante determinar los alcances del producto, en cada revisión para evitar superar los límites establecidos, pues de no tener claros los requerimientos del producto llevará a un proyecto sin fin.

El esquema de prototipos admite que todo el sistema, o algunos de sus fragmentos, se edifiquen ágilmente para entender con facilidad y esclarecer ciertos aspectos en los que se aseguren que el desarrollador, el usuario, el cliente estén de acuerdo en lo que se necesita así como la solución que se plantea para dicha necesidad y de esta forma minimizar el peligro y la inseguridad en el desarrollo, este modelo se encarga del desarrollo de diseños para que estos sean estudiados y excluidos a medida que se consoliden nuevas descripciones, es ideal para calcular el alcance del producto, pero no se certifica su uso real.

Este esquema principalmente se emplea cuando un cliente especifica un conjunto de objetivos generales para el software a desarrollarse sin determinar detalladamente los requisitos de entrada proceso y salida, es decir cuando el diseñador no está seguro de la eficacia de un algoritmo, de la versatilidad del sistema. Este esquema se encarga especialmente de ayudar al ingeniero de sistemas y al cliente a razonar de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos. [8]

### **6.1.3.1.1 Ciclo de Vida de un Sistema Basado en Prototipo**

Una modelo o prototipo de pantallas muestra la interfaz de la aplicación, su cara externa, pero dicha interfaz está fija, estática, no procesa datos. El prototipo no tiene desarrollada una lógica interna, sólo muestra las pantallas por las que irá pasando la futura aplicación. Por su parte, el prototipo funcional progresivo desarrolla un procedimiento que cumple los requisitos y necesidades que se han entendido claramente.

Ejecuta, por tanto, un proceso real de datos, para comprobar con el usuario. Se va modificando y desarrollando sobre la marcha, según las apreciaciones del cliente. Esto frena el proceso de desarrollo y disminuye la desconfianza, puesto que el software está continuamente variando, pero, a la larga, genera un producto más seguro, en cuanto a la complacencia de las necesidades del cliente.

Cuando un modelo o prototipo se desarrolla con el sólo propósito especificar mejor las necesidades del cliente y después no se va a emplear ni total ni parcialmente en la implementación del sistema final se habla de un prototipo innecesario. Para que la construcción de prototipos sea posible se debe contar con la participación activa del cliente [9]

#### 6.1.4. Metodologías Ágiles

Los principios de desarrollo destacan la entrega sobre el análisis y el diseño no obstante estas actividades no se intimidan y continúa entre desarrolladores y clientes.

Un conjunto de principios de desarrollo se compagina como una ideología a la ingeniería de software ágil. Esta ideología hace énfasis en la satisfacción del cliente y en la entrega rápida del software incremental, los grupos pequeños y muy determinados para desarrollar el proyecto, los procesos informales, los productos del trabajo con mínima ingeniería de software y la simpleza general en el desarrollo.

¿Por qué es importante?

En la situación actual de negocios crea sistemas basados, en computadora y productos de software evolucionada rápido y constantemente. La ingeniería de software ágil representa una alternativa razonable a la ingeniería de software convencional para ciertas clases de software y en algunos tipos de proyectos. Al mismo tiempo se ha confirmado que remata con celeridad sistemas exitosos.

Perduran las actividades estructurales fundamentales: comunicación, planeación, modelado, construcción y despliegue. Pero se convierten en un conjunto mínimo de labores que lleva al equipo del proyecto hacia la construcción y entrega. A subes el cliente, el ingeniero de software tiene la misma perspectiva: el único producto del trabajo realmente



importante es un incremento de software operativo se entrega al cliente debidamente en la fecha establecida.[10]

¿Qué es un proceso ágil?

Cualquier proceso del software ágil se caracteriza por la forma en la que aproxima cierto número de suposiciones clave acerca de la mayoría de proyectos de software.

a. Es difícil prever qué requerimientos de software permanecerán y cuales cambiarán. También es difícil predecir cómo cambiarán las prioridades del cliente a medida que progresa el proyecto.

b. Para muchos tipos de software, el diseño y la construcción están alternos. Es decir, ambas actividades deben desarrollarse en forma paralela, de modo que los modelos de diseño se certifiquen a medida que se crean. Es difícil prever cuánto boceto se necesita antes de que se use la construcción para probar el diseño.

c. El análisis, el diseño, la construcción y las pruebas no son tan predecibles como nos gustaría (desde un punto de vista de planeación)[11]

#### 6.1.4.1. Programación Extrema

La programación extrema es una de las metodologías ágiles que tiene como objetivo principal lograr aumentar la productividad a la hora de desarrollar un proyecto de software.

La programación extrema se basa en la simplicidad, una de las grandes diferencias de la programación extrema a las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad, la cual abarca un conjunto de reglas y prácticas que suceden en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas.

**Planeación.** La actividad de planeación comienza escuchando -actividad para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionalidad que se requiere-. Escuchar lleva a la creación de algunas “historias” que describen la salida necesaria, características y funcionalidad del software que se va a elaborar.

**Diseño.** El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se

desalienta el diseño de funcionalidad adicional porque el desarrollador supone que requerirá después.

**Codificación.** Después que las historias han sido desarrolladas y de que se ha hecho el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir en la entrega en curso (incremento de software). Una vez creada la prueba unitaria, el desarrollador está mejor capacitado para centrarse en lo que debe implementarse para pasar la prueba. No se agrega nada extraño (MS), Una vez que el código es terminado, se le aplica de inmediato una prueba unitaria, con lo que se obtiene retroalimentación instantánea para los desarrolladores.

**Pruebas.** Ya se dijo que la creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas (de modo que puedan ejecutarse en repetidas veces y con facilidad). Esto estimula una estrategia de pruebas de regresión siempre que se modifique el código (lo que ocurre con frecuencia, dada la filosofía en XP). [12]

## 6.1.5. Levantamiento y Análisis de Requerimientos

### 6.1.5.1. Historias de Usuario

Es una técnica utilizada en el desarrollo de software, consiste en la redacción de un requerimiento en unas pocas frases, adicional a esto se utiliza la expresión común del usuario. Las historias de usuario son ideales para desarrollos con metodologías ágiles, definen las descripciones del proyecto, posteriormente y debido a la prioridad asignada a cada una, se acondicionan dentro de las diferentes iteraciones y se plasman los estimados en tiempos para la realización de cada historia, este proceso lo ejecutan los desarrolladores. Características de una historia de usuario:

**Independiente.** Pueden existir historias que se vinculen con otras, es obligatorio separar de manera que resulten lo más independiente posible, ya que esta independencia es clave a la hora de la etapa del diseño.

**Negociable.** Las pruebas de validación son la manera en que los clientes pueden comprobar el alcance de la historia de usuario.

**Valoradas por clientes o usuarios.** Las historias de usuarios son importantes para clientes y usuarios, deben mostrar lo se solicita, no lo que el desarrollador quiera hacer.

**Estimables.** Cuando se evalúa los tiempos para cada historia de usuario es más fácil precisar cuánto tiempo se tardará un proyecto.

**Pequeñas.** Deben ser pequeñas por la naturaleza reiterada de los proyectos donde se utilizan.

**Verificables.** Debe ser posible la verificación de los logros obtenidos en las historias de usuario en cada etapa de entrega del proyecto de los proyectos [13]

### **6.1.6. Diseño de Software**

El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo del sistema o un producto de alta calidad. Los principios de diseño establecen una metodología general que guía el trabajo de diseño que debe ejecutarse. Se deben comprender los conceptos de diseño antes de emplear el funcionamiento de este, y la práctica de modelo en sí lleva a la creación de distintas característica del software que sirve como guía para la agilidad de desarrollo que siga.[14]

#### **6.1.1.6.1. UML (Lenguaje Unificado de Modelado)**

El UML (Unified Modeling Language) o lenguaje relacionado de modelado es el sucesor de concentración de métodos de análisis y diseño orientado a objetos (OOA&D) que surgió a finales de la década de 1980 y principios de la siguiente. El UML relaciona, sobre todo, los métodos de Booch, Rumbaugh (OMT) y Jacobson, pero su alcance llegar a ser mucho más extenso.

Se puede afirmar, que el UML es un lenguaje de modelado, y no un método. La mayor parte de los métodos consisten, al menos en principio, en un lenguaje y en un proceso para modelar. El lenguaje de modelado es la notación gráfica de que se corresponden los métodos para expresar los diseños. El proceso es la orientación que nos dan sobre los pasos a seguir para hacer el diseño.

Cuando se utiliza UML para el diseño de software, se trata de llevar la idea original o la conceptualización del problema a una parte del software y su ejecución. UML proporciona un medio para apropiarse y examinar requisitos (diagramas de casos de uso), un concepto importante pero, a veces, difícil de entender para un programador joven. Existen diagramas para apropiarse cuáles son las partes del software que cumplen con ciertos requisitos (diagramas de colaboración). Otros diagramas sirven para apropiarse exactamente cómo realizan sus requisitos esas partes del sistema (diagramas de secuencia y diagramas de estado). Por último, hay diagramas para mostrar todo lo que se acopla y ejecuta de modo conjunto (diagramas de componentes y diagramas de despliegue) [15]

Vistas 4+1 herramientas para representar modelos de arquitectura anteriores al UML es la denominada vistas 4+1 propuesta por Kruchten. Surge esta necesidad cuando un desarrollador inicia el modelado de un problema; lógicamente en este proceso se hace énfasis

en proporcionar la mayor información del mismo a través de los diagramas que se utilicen para su descripción, el cual puedan suceder conflictos en la representación de la arquitectura del sistema. Esta arquitectura de software propuesta por Kruchten es una forma de resolver esta problemática.

El modelo describe la arquitectura de software del sistema a través de 5 vistas concurrentes. Kruchten las agrupa estas 5 vistas por su naturaleza en 3 apartados; el conceptual donde sitúa a la vista lógica y la de procesos, el físico que lo compone las vistas de componentes y la funcional la que se refiere a la vista de casos de uso.

**Vista lógica.** Se trata del modelado de clases y subsistemas, tiene las siguientes características: soporta los requerimientos eficaces (los cuales son estipulados por el usuario final), identifica elementos y diseña manuales comunes a través del sistema; utiliza los diagramas de clases y la notación de Booch. Además de utilizar el estilo arquitectónico orientado a objetos.

**Vista concurrente o de procesos.** Representa el modelado de concurrencia y aspectos de sincronización. Detalla las líneas de mando que se ejecutan en cada operación de cada una de las clases notables en la vista lógica. Los diseñadores realizan esta vista en varios niveles de abstracción, además de dividir el software en conjuntos independientes de tareas.

**Vista de componentes o de desarrollo.** Representa la organización estática de software en las situaciones de desarrollo. Es una importante característica de lógica en casos de vistas, independientes, persistentes y de distribución, representa la intervención en diferentes instrucciones, comprueba si existe persistencia entre un objeto y otro, además comprueba el estado de los objetos y operaciones de accesibilidad por muchos nodos.

**Vista distribuida o física.** Representa la ejecución en módulos y segmentación en muchas capas. Recopila las categorías de clases y grupos. Representa el mapeo del software en el hardware y toma en cuenta los requerimientos funcionales del sistema, como seguridad, respuesta y escalabilidad.

**Vista de casos de uso o escenarios.** Administra los requerimientos que son obligatorios para el usuario final, y construye manuales comunes a través del sistema. Esta vista es coherente con el conjunto que integran las anteriores de ahí que se le denomina +1, pero su inclusión es vital ya que desempeña dos relaciones importantes, actúa como guía que ayuda al diseñador a descubrir los manuales de la arquitectura durante su diseño y valida e ilustra el diseño de la misma. Es equivalente a la que se utiliza en la vista lógica, a excepción de que usa los conectores de la vista de procesos para indicar la interacción entre objetos[16]



#### **6.1.6.1.1. Casos de Uso**

Un caso diagrama de caso de uso representa la forma que un usuario interactúa con el sistema, especificando los movimientos requeridos para lograr una meta específica. Las transiciones en la secuencia de caminos describen varios escenarios. Un diagrama UML de caso de caso es una perspectiva de todos los casos de uso y sus relaciones. El mismo suministra un gran cuadro de la funcionalidad del sistema [17]

Una vez establecidos los requisitos específicos de la aplicación, para iniciar se definirán los casos de uso con claridad.

Con los casos de uso se especificará el comportamiento deseado del sistema. con el conjunto de acciones se especificara qué resultado debe producir el sistema, esto también capturará el comportamiento deseado del sistema.

#### **6.1.6.1.2. Diagrama de Clases**

El diagrama de clases representa los tipos de objetos que hay en el sistema y las diversas clases de relaciones estáticas que existen entre ellos. Hay dos tipos principales de relaciones estáticas: agrupaciones y subtipos. Los diagramas de clases también muestran los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos. [18]

#### **6.1.6.1.3. Diagrama de Componentes**

El diagrama de componentes representa la estructura física del sistema de software y, casualmente, de su entorno organizativo en componentes, a efectos de construcción y funcionamiento. La descomposición del diagrama de componentes se ejecuta en términos de componentes y de relaciones entre los mismos. [19]

#### **6.1.6.1.4. Diagrama de Actividades**

Un diagrama de actividad UML define el comportamiento eficiente de un sistema o parte del sistema a través del flujo de control entre operaciones que realiza el sistema. Es similar a un diagrama de flujo, excepto porque el diagrama de actividad puede mostrar flujos concurrentes. El componente principal de un diagrama de actividad es un nodo acción, representado mediante un rectángulo redondeado, que corresponde a una tarea realizada por el sistema de software. Las flechas desde un nodo acción hasta otro indican el flujo de control; es decir, una flecha entre dos nodos acción significa que, después de completar la primera operación, comienza la segunda operación. Un punto negro sólido forma el nodo inicial que indica el punto de inicio de la actividad. Un punto negro rodeado por un círculo negro es el nodo final que indica el fin de la actividad. [20]

#### **6.1.6.1.5. Diagrama de Despliegue**

El diagrama de despliegue reconoce (en inglés, deployment) exponer la arquitectura en tiempo de elaboración del sistema relación a hardware y software. El diagrama de despliegue se utiliza en el diseño y la implementación. Se pueden diferenciar componentes como los del diagrama de componentes y nodos, así como las relaciones entre todos estos. Es más definido que el diagrama de componentes, en el sentido de que representa la organización del sistema solo en período de elaboración, pero no en período de desarrollo o compilación. Sin embargo, resulta más amplio en el sentido de que puede contener más clases de elementos. [21]

#### **6.1.6.2. Modelo Relacional**

El modelo relacional sin dudas el soporte de la tecnología moderna de base de datos; este soporte hace de este campo una ciencia. Entonces, todo libro sobre fundamentos de la tecnología de bases de datos que no envuelva detalladamente una cobertura de modelo relacional es por definición superficial. De igual forma, arduamente se puede justificar cualquier afirmación de someter el campo de las bases de datos si quien lo afirma no comprende a fondo el modelo relacional. El modelo relacional se ocupa de tres aspectos principales de la información: la estructura de datos, la manipulación de datos y la integridad de datos. [22]

### **6.1.6.3. Arquitectura de Software**

Cuando se piensa en la arquitectura de una construcción, llegan a la mente muchos atributos distintos. En el nivel más sencillo, se considera la forma general de la estructura física. Pero, en realidad, la arquitectura es mucho más que eso. Es la manera en la que los distintos componentes del edificio se integran para formar un todo cohesivo. Es la forma en la que la construcción se adapta a su ambiente y se integra a los demás edificios en la vecindad. Es el grado en el que el edificio cumple con su propósito y en el que satisface las necesidades del propietario. Es la sensación estética de la estructura —el efecto visual de la edificación— y el modo en el que se combinan texturas, colores y materiales para crear la fachada en el exterior y el “ambiente de vida” en el interior. Es los pequeños detalles: diseño de las lámparas, tipo de piso, color de las cortinas... la lista es casi interminable. Y, finalmente, es arte.

La arquitectura no es el software operativo. Es una representación que permite 1) analizar la efectividad del diseño para cumplir los requerimientos establecidos, 2) considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil 3) reducir los riesgos asociados con la construcción del software[23].

#### **6.1.6.3.1 Modelo vista Controlador**

El esquema de arquitectura MVC (Modelo Vista Controlador) es un esquema que especifica la estructura independiente del Piloto (Objetos de Negocio), la Vista (interfaz con el

usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación). De esta forma, se divide el método en tres capas en el cual, como se especificará más adelante, se asume la encapsulación de los datos, la interfaz o vista y por último la lógica interna o controlador. El esquema de arquitectura "modelo vista controlador", es una filosofía de bosquejo de aplicaciones, compuesta por:

**Modelo.** Contiene el núcleo de la funcionalidad de la aplicación. Encapsula el estado de la aplicación. No sabe nada / independiente del Controlador y la Vista.

**Vista.** Es la introducción del Modelo. Puede acceder al Modelo pero nunca cambiar su estado. Puede ser notificada cuando hay un cambio de estado en el Modelo.

**Controlador.** Reacciona a la petición del Cliente, elaborando la acción adecuada y creando el modelo pertinente.

Para deducir cómo funciona el patrón Modelo vista controlador, se debe deducir la división a través del conjunto de estos tres manuales y cómo estos módulos se comunican unos con los otros y con otras vistas y controladores externos al modelo principal. Para ello, es importante saber que el controlador descifra las entradas del usuario (tanto teclado como el ratón), enviado el mensaje de acción al modelo y a la vista para que se proceda con los cambios que se consideren adecuados. [24]

### **6.1.7 Codificación**

En esta etapa se tienen que traducir dichos algoritmos a un lenguaje de programación específico; es decir, las acciones definidas en los algoritmos hay que convertirlas a instrucciones. Para codificar un algoritmo hay que conocer la sintaxis del lenguaje al que se va a traducir. Sin embargo, independientemente del lenguaje de programación en que esté escrito un programa, será su algoritmo el que determine su lógica. La lógica de un programa establece cuáles son sus acciones y en qué orden se deben ejecutar. Por tanto, es conveniente que todo programador aprenda a diseñar algoritmos antes de pasar a la fase de codificación. [25]

### **6.1.8 Pruebas**

Una entrega es el resultado del proyecto que se entrega al cliente. De forma general, se entrega al final de una fase principal del proyecto como la especificación, el diseño. [26]

Una vez finalizado el desarrollo de la página web y de la aplicación y su posterior instalación en dispositivos móviles se procede a realizar las pruebas de funcionamiento.

Se verifica que las opciones de la app sean funcionales y muestren el contenido gráfico y escrito correctamente, luego continuamos evaluando los botones de tips, desarrollo del bebé, vacunas y finalmente pruebo la opción llamadas de emergencias teniendo como resultado en todas las pruebas un correcto funcionamiento.

## **6.1.9 Entrega**

Una entrega es el resultado del proyecto que se entrega al cliente. De forma general, se entrega al final de una fase principal del proyecto como la especificación, el diseño. [27]

## **6.1.10 Estándares de Web**

Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí. El W3C, organización independiente y neutral, desarrolla estándares relacionados con la Web también conocidos como Recomendaciones, que sirven como referencia para construir una Web accesible, interoperable y eficiente, en la que se puedan desarrollar aplicaciones cada vez más robustas. [28]

### **6.1.10.1 HTML5**

HTML5 es un lenguaje markup (de hecho, las siglas de HTML significa Hyper Text Markup Language) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue creado en 1990. A fines del año pasado, la W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Por así decirlo, qué es HTML5 está relacionado también con la entrada en

decadencia del viejo estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación. [29]

#### **6.1.10.2 CSS**

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. [30]

#### **6.1.10.3. APP Inventor**

App Inventor parte de una idea conjunta del Instituto Tecnológico de Massachusetts y de un equipo de Google Education. Se trata de una herramienta web de desarrollo para iniciarse en el mundo de la programación. Con él pueden hacerse aplicaciones muy simples, y también muy elaboradas, que se ejecutarán en los dispositivos móviles con sistema operativo Android. App Inventor es un lenguaje de programación basado en bloques (como piezas de un juego de construcción), y orientado a eventos. Sirve para indicarle al “cerebro” del dispositivo



móvil que queremos que haga, y cómo. Es por supuesto muy conveniente disponer de un dispositivo Android donde probar los programas según los vamos escribiendo. [31]

#### **6.1.10.4. Visual Studio**

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. [32]

## **6.2 Marco Conceptual**

### **6.2.1 Tips de Cuidado**

El cuidado de primera infancia desempeña un papel fundamental y esencial en el desarrollo del niño, y es cada vez más habitual hoy en día, los tips de cuidados infantiles buscan brindar un valioso apoyo a las familias con niños pequeños. Por lo tanto, es importante vislumbrar el impacto de estos consejos en cuidadores primerizos para asegurar un cuidado de calidad para los bebés.

Los tips o consejos pueden ser transmitidos de generación en generación creadas a partir de la experiencia. Lista de tips son una serie de instrucciones para realizar un actividad en particular, como por ejemplo el cambio de un pañal.

### **6.2.2 Desarrollo del Bebé**

Es muy importante comprender la necesidad de invertir en el desarrollo de los niños pequeños, ya que así se maximiza su bienestar en el futuro.

Es de vital importancia ser conscientes del impacto directo en el desarrollo emocional, social y físico que un niño pequeño tiene, pues esto afecta en el adulto en el que se convertirán.

Los primeros vínculos que forman con sus padres y sus primeras experiencias, afectan profundamente su desarrollo físico, cognitivo, emocional y social en el futuro.

### **6.2.3 Vacunas**

El ser conscientes de la importancia y la responsabilidad individual o familiar, dado que una sola persona no vacunada expone a todos sus contactos y a la sociedad a adquirir enfermedades.

La vacunación de las poblaciones ha permitido que muchas enfermedades que causaban epidemias en la antigüedad, hoy en día prácticamente hayan desaparecido.

Las vacunas o inmunizaciones han defendido a los niños de enfermedades graves por más de 50 años. Las vacunas estimulan la producción de anticuerpos, que son las defensas

contra una enfermedad determinada. De esta manera, el niño será inmune a esa enfermedad si en algún momento se pone en contacto con el germen que la causa.

#### **6.2.4 Test**

El crecimiento y desarrollo de la primera infancia es el proceso de evolución dinámica y rápida en los niños durante su ciclo de vida. Y por esto importante el cuidado y acompañamiento del mismo, así como la detección temprana y atención oportuna de sus alteraciones, por este motivo se ha creado diferentes maneras, como lo son test de desarrollo psicomotriz y de lenguaje para así monitorear estos aspectos.

Los Tests de Desarrollo son un instrumento de valoración del desarrollo infantil, es una evaluación que permite conocer el nivel de rendimiento del niño mediante la observación de la conducta del niño frente a situaciones cotidianas.

#### **6.2.5 Ciudad Ubicación (Geolocalización)**

La geolocalización es la capacidad para conseguir la ubicación geográfica real de un objeto, como un radar, un teléfono móvil o un computador conectado a Internet.

El usuario que utilice Abchild Parenty Guide podrá geolocalizarse en una ciudad o en un lugar geográfico exacto (normalmente representado por un mapa en la web) determinado por unas coordenadas.

El sistema de geolocalización ayudará a optimizar las rutas de los sitios de interés que se deseen, mostrará los detalles de los contactos y sitios de interés, verá que el mapa que muestra en donde se encuentra la dirección.

## **7. Capítulo III Otras Consideraciones**

### **7.1. Marco Legal y Normativo**

Los Derecho de los niños y niñas son prevalentes frente a los demás derechos, esto significa que los niños y niñas tienen una especial protección constitucional, lo cual implica que frente a un conflicto de Derechos prevalecerán los Derechos de los menores quienes según la constitución política de 1991 gozan de los derechos fundamentales a la vida, la integridad física, la salud y la seguridad social, la alimentación equilibrada, su nombre y nacionalidad, tener una familia y no ser separados de ella, el cuidado y amor, la educación y la cultura, la recreación y la libre expresión de su opinión.

Posteriormente en el año 2006 se expide el código de infancia y adolescencia como legislación prevalente para la protección, cuidado y garantías legales de los derechos de los niños y niñas, la finalidad del código de infancia y adolescencia es la de garantizar que los niños y niñas se desarrollaran de manera armoniosa en ambientes sanos y con los debidos cuidados requeridos, recordando la dignidad humana, la igualdad y la no discriminación como principios generales para el cuidado de los menores. Los menores gozan de derechos irrenunciables, prevalentes e interdependientes los cuales serán aplicados de manera imperativa por todas las personas garantizando la satisfacción de los derechos humanos de los menores.

## **7.2. Marco Geográfico**

La aplicación Abchild Parenty Guide se desarrolló en el Departamento de Risaralda, específicamente en el municipio de Pereira, este proyecto busca ayudar y potenciar el desarrollo y el cuidado infantil durante los primeros años de vida, que permanecen bajo el cuidado de padres primerizos.

Los módulos de esta aplicación están orientados al cuidado de los niños y niñas de primera infancia y sus familias y/o cuidadores, de manera prioritaria a los niños y niñas entre los primeros meses y hasta menores de 1 año.

## **8. Capítulo IV Definiciones y Requerimiento del Sistema**

### **8.1. Levantamiento de Requerimiento**

En esta sección del proyecto se identificarán los distintos requerimientos del sistema, además se pasará a representarlos haciendo uso de la técnica de prototipos no funcionales, esto con el fin de visualizar cómo será la interfaz gráfica del sistema junto con todas sus funcionalidades. Por lo tanto, a partir de lo anterior se levantaron los requerimientos para la creación del software para el cuidado en el desarrollo de la primera infancia, test de desarrollo psicomotriz y de lenguaje y llamadas de emergencia.

<b>ID</b>	<b>Nombre del Requerimiento</b>	<b>Descripción</b>
<b>F1</b>	<b>Registro y edición de usuarios</b>	<b>La aplicación permitirá crear un perfil para los usuarios y permitirá editar dicho perfil.</b>
<b>F2</b>	<b>Sistema Geolocalización</b>	<b>La aplicación permitirá geolocalizarse en una ciudad o en un lugar geográfico exacto (normalmente representado por un mapa en la web) determinado por unas coordenadas</b>
<b>F3</b>	<b>Envíos de correos</b>	<b>La aplicación enviara un correo con la información de la encuesta realizada</b>
<b>F4</b>	<b>Visualización de información de vacunas</b>	<b>La aplicación les permitirá a los usuarios visualizar la información sobre las distintas vacunas que se deben aplicar en el momento correspondiente.</b>
<b>F5</b>	<b>Comunicación con redes sociales</b>	<b>La aplicación les permitirá a los usuarios publicar información sobre el desarrollo del niño en redes sociales o alguna noticia que sea de su interés en sus diferentes redes sociales.</b>
<b>F6</b>	<b>Servicio de llamadas</b>	<b>La aplicación permitirá realizar llamadas a su médico de cabecera , farmacias, centro asistencial del usuario, contacto en caso de emergencia o al 911</b>
<b>F7</b>	<b>Consulta de sitios de interés, conectado al sistema Bing Maps y google maps</b>	<b>El sistema Bing Maps y google maps ayudará a optimizar las rutas de los sitios de interés que se deseen, mostrará los detalles de los contactos y sitios de interés, verá que el mapa que muestra en donde se encuentra la dirección.</b>



Tipo de Driver AC / Rest	Descripción del Driver	Prioridad
AC	<b>Usabilidad:</b> -Intuitivo y de fácil uso para los usuarios	4
AC	<b>Accesibilidad:</b> -Acceso multiplataforma -Acceso multidispositivo	4
AC	<b>Escalabilidad:</b> -Por medio de AzureAnalytics realizamos estudios de la cantidad de solicitudes de usuarios, como también sus momentos picos, donde se podrá ajustar por medio de Azure, en aumento de las características de los servidores donde se aloja los servicios de la app	4
AC	<b>Rendimiento :</b> -Servicio alojado en Microsoft azure	5

### 8.1.1. Historias de Usuario

Los requerimientos del sistema se construyeron a partir de las práctica de historias de usuario (HU), cada una de las HU conciernen a un proceso detallado en el levantamiento de requerimientos del sistema. Dentro de la metodología de ingeniería utilizada para el desarrollo del proyecto las HU son un fragmento importante para entender de los procesos y en la arquitectura del diseño del sistema. (Ver ANEXO 1 Historias de Usuario)

caso de uso1	
<b>Caso de uso</b>	<b>Registro</b>
<b>Actores</b>	<b>Usuario</b>
<b>Propósito</b>	Este caso de uso permite a los usuarios acceder a la plataforma, diligenciando los campos nombre de usuario y contraseña de usuario.

<b>Resumen</b>	Este caso de uso comienza cuando el usuario decide iniciar sesión para ingresar al sistema y acceder las funcionalidades de la aplicación.
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1. El usuario ingresa nombre y contraseña.	2. El sistema comprueba que los datos ingresados por el usuario coincidan con los de la bases de datos.
	3. El sistema deja al usuario acceder al perfil.
<b>Curso Alterno Iniciar Sesión</b>	
<b>Acción 2:</b> Si el sistema detecta que los datos ingresados por el usuario son incorrectos debe notificarle al usuario que los datos son incorrectos y debe regresar al paso 1.	
<b>Sección verificar registro</b>	
1. El usuario ingresa datos y verifica su registro	2. El sistema envía mensaje que se encuadra registrado
3. El usuario debe ingresar el correo que registro en la base de datos.	4. El sistema verifica el correo del usuario sea el que se encuentra en la base de datos.
	5. El sistema envía un correo electrónico al correo que corresponde el nombre de usuario con datos de registro
<b>Curso Alterno Registro exitoso</b>	
<b>Acción 4.</b> Si el nombre de usuario ingresado no corresponde con el asociado en la base de datos del sistema, regresa al paso 3.	

<b>Caso de uso # 2</b>	
<b>Caso de uso</b>	<b>Consultar</b>
<b>Actores</b>	<b>Usuario</b>
<b>Propósito</b>	Este caso de uso les permite a los usuarios consultar las funciones de la app
<b>Resumen</b>	Este caso de uso comienza cuando el usuario decide consultar la información de su propia cuenta del usuario.
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
<b>1.</b> El usuario decide Consultar la información desde su móvil pc	<b>2.</b> El sistema comprueba que el usuario que está tratando de acceder tenga el permiso.
	<b>3.</b> El sistema muestra lista de opciones que el usuario puede consultar.
<b>4.</b> El usuario selecciona los tips que desea consultar	<b>5.</b> El sistema muestra información de los tips que el usuario seleccionó
<b>6.</b> El usuario termina sección si no tiene más opciones a consultar.	
<b>Curso Alternativo Consultar Usuario</b>	
<b>Acción 2:</b> Si el sistema identifica que el usuario no tiene los permisos para acceder a la funcionalidad, cancela la acción y vuelve al paso 1.	
<b>Acción 6:</b> Si el usuario desea consultar otros tips del sistema, regresará al paso 3.	

<b>Caso de uso # 3</b>	
<b>Caso de uso</b>	<b>Reporte</b>

<b>Actores</b>	<b>Sistema</b>
<b>Propósito</b>	Este caso de uso les permite al sistema mostrar todas las funciones que tiene la aplicación.
<b>Resumen</b>	Este caso de uso comienza cuando el sistema muestras todas las opciones que cuentan los usuarios al descargar la aplicación.
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1. El sistema cuenta con información el cual será útil para los usuarios	2. El sistema comprueba que todos las tis de la app funcione excelentemente
	3. El sistema muestra una lista de tips que el usuario puede consultar
4. El sistema se muestra lo accesible que es ingresar y descargar app	
<b>Curso Alterno a búsqueda del sistema</b>	
<b>Acción 2:</b> El sistema accede que el usuario haga modificaciones y consulta sobre las opciones que muestran en la app si no cancela la acción y vuelve al paso 1.	

<b>caso de uso #4</b>	
<b>Caso de uso</b>	<b>Eliminar</b>

<b>Actores</b>	<b>Usuario</b>
<b>Propósito</b>	<b>Este caso de uso le permite al usuario Chef eliminar</b>
<b>Resumen</b>	<b>Este caso de uso comienza cuando el usuario dice eliminar información del sistema</b>
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
<b>1. El usuario decide eliminar información.</b>	<b>2. El sistema comprueba que el usuario que está intentando realizar esta acción tenga los permisos necesarios.</b>
	<b>3. El sistema muestra el listado de información a eliminar.</b>
<b>4. El usuario elige el lista de información eliminar y realizar la acción.</b>	<b>5. El sistema acepta la solicitud y guarda los.</b>
<b>Curso Alterno Eliminar Menú</b>	
<b>Acción 2: Si el sistema detecta que el usuario no tiene permisos para acceder a esta aplicación, el sistema devuelve al usuario al paso 1 y muestra un mensaje de error en la autorización del usuario.</b>	
<b>Acción 4: Si el usuario decide cancelar la acción puede volver a la ventana principal.</b>	

<b>Caso de uso # 5</b>	
<b>Caso de uso</b>	<b>Actualizar</b>

<b>Actores</b>	<b>Usuario</b>
<b>Propósito</b>	<b>Este caso de uso le permite al usuario actualizar información.</b>
<b>Resumen</b>	<b>Este caso de uso comienza cuando el usuario decide actualizar información que se encuentra en el sistema.</b>
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
<b>1. El usuario decide eliminar un menú.</b>	<b>2. El sistema comprueba que el usuario que está intentando realizar esta acción tenga los permisos necesarios.</b>
	<b>3. El sistema muestra el listado de la tips a actualizar</b>
<b>4. El usuario elige los tips a actualizar y realizar la acción.</b>	<b>5. El sistema acepta la solicitud y guarda los.</b>
<b>Curso Alterno Actualizar información</b>	
<b>Acción 2: Si el sistema detecta que el usuario no tiene permisos para acceder a esta aplicación, el sistema devuelve al usuario al paso 1 y muestra un mensaje de error en la autorización del usuario.</b>	
<b>Acción 4: Si el usuario decide cancelar la acción puede volver a la ventana principal.</b>	

<b>Caso de uso # 6</b>	
<b>Caso de uso</b>	<b>Bing mapas</b>

<b>Actores</b>	<b>Sistema y Usuario</b>
<b>Propósito</b>	<b>Este caso de uso permite al usuario acceder al sistema para visualizar las rutas donde pueden acceder.</b>
<b>Resumen</b>	<b>Este caso de uso comienza cuando se tienen rutas específicas para llegar al lugar escogido.</b>
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
<b>1. El usuario ingresa a Lista de rutas.</b>	<b>2. El sistema comprueba que el usuario que está tratando escoger l ruta mas cerca.</b>
	<b>3. El sistema muestra una lista de rutas cercanas.</b>
<b>4. El usuario escoge ruta más rápida.</b>	<b>5. El sistema muestra la información de la ruta (Sección escoger ruta).</b>
	<b>6. El sistema actualiza la lista de rutas para repetir el proceso.</b>
<b>Curso Alterno Bing mapas</b>	
<b>Acción 2: Si el sistema identifica que el usuario no tiene los permisos para acceder a la funcionalidad, cancela la acción y vuelve al paso 1.</b>	
<b>Acción 6: Si el sistema tiene más órdenes en lista vuelve al paso 3.</b>	

**Caso de uso #7**

<b>Caso de uso</b>	<b>Geo localización</b>
<b>Actores</b>	<b>Sistema y Usuario</b>
<b>Propósito</b>	<b>Este caso de uso permite al usuario acceder al sistema para geo localizar los lugares más cercanos pueden acceder.</b>
<b>Resumen</b>	<b>Este caso de uso comienza cuando se tienen los lugares específicos para buscar.</b>
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los Eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
<b>1. El usuario ingresa a lista de lugares.</b>	<b>2. El sistema comprueba que el usuario que está tratando buscar lugares más cerca.</b>
	<b>3. El sistema muestra una lista de los lugares cercanos.</b>
<b>4. El usuario escoge el lugar más rápido.</b>	<b>5. El sistema muestra la información del lugar (Sección escoger lugar).</b>
	<b>6. El sistema actualiza la lista de lugares para repetir el proceso.</b>
<b>Curso Alternativo Bing mapas</b>	
<b>Acción 2: Si el sistema identifica que el usuario no tiene los permisos para acceder a la funcionalidad, cancela la acción y vuelve al paso 1.</b>	
<b>Acción 6: Si el sistema tiene más órdenes en lista vuelve al paso 3.</b>	



<b>Caso de uso # 8</b>	
<b>Caso de uso</b>	<b>Revisión correos</b>
<b>Actores</b>	<b>Usuario</b>
<b>Propósito</b>	<b>Este caso de uso les permite a los usuarios revisar su cuenta de usuario.</b>
<b>Resumen</b>	<b>Este caso de uso comienza cuando el usuario decide editar o modificar una su cuenta de usuario.</b>
<b>Tipo</b>	<b>Esencial</b>
<b>Referencias</b>	
<b>Curso Normal de los eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
<b>1. El usuario decide realizar cambios a su cuenta de usuario, a través de la funcionalidad revisión correos.</b>	<b>2. El sistema comprueba que el usuario tenga los permisos para acceder a esta funcionalidad.</b>
	<b>3. El sistema muestra los datos para modificar.</b>
<b>4. El usuario realiza los cambios.</b>	<b>5. El sistema comprueba que todos los campos del formulario coincidan.</b>
	<b>6. El sistema acepta los cambios y los guarda.</b>
<b>Curso Alternativo Revisar correos</b>	
<b>Acción 2: Si el sistema identifica que el usuario no tiene los permisos para acceder a la funcionalidad, cancela la acción y vuelve al paso 1.</b>	

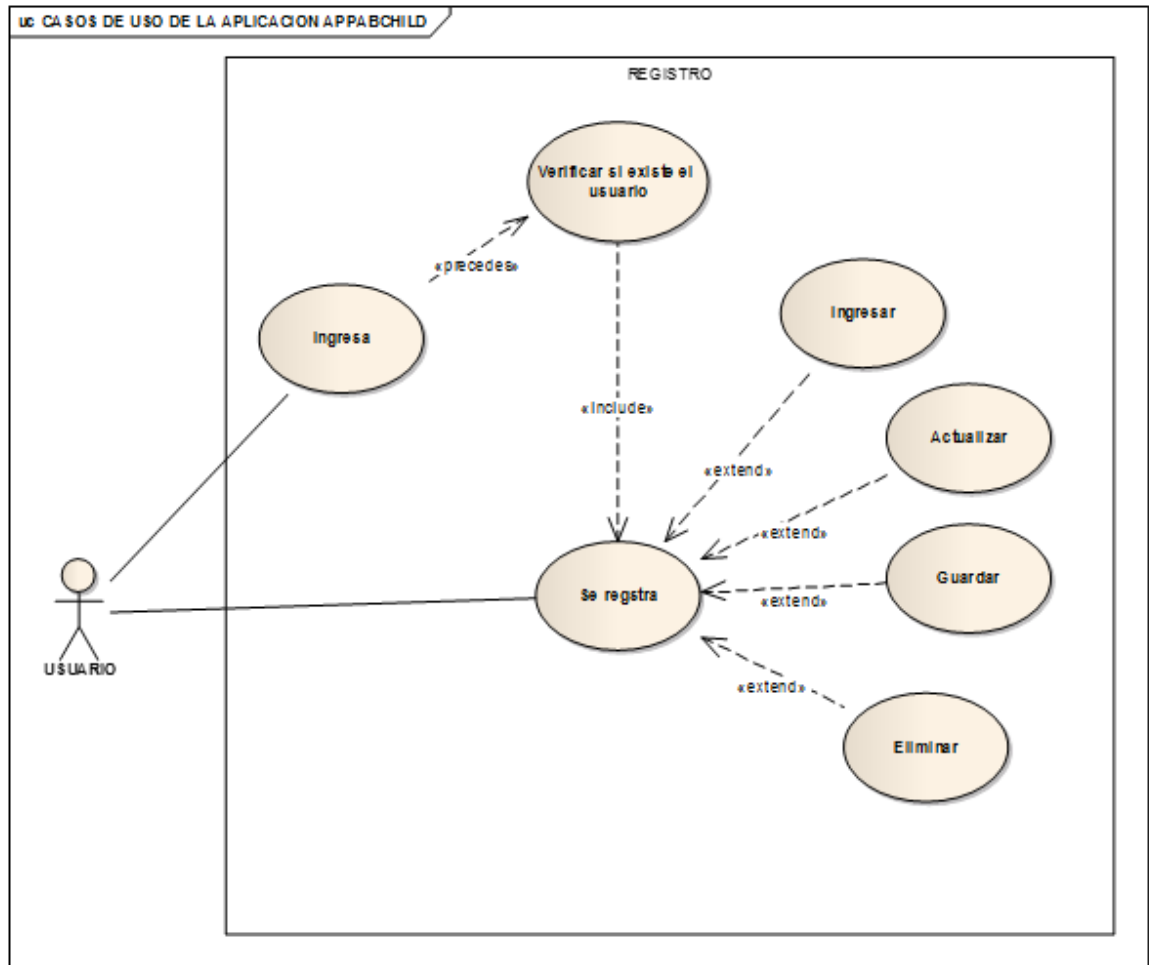
**Acción 5: Si uno o varios de los datos no coinciden, el sistema cancela la acción y vuelve al paso 4 para mostrar al usuario los datos incorrectos.**

## **8.2. Diseño Del Sistema**

La siguiente etapa de la metodología de ingeniería del software es el diseño, a continuación, se describen las vistas 4+1 del proyecto.

### **8.2.1. Vista de Escenarios (CASOS DE USOS)**

Cada escenario modelado por un diagrama de caso de uso viene descrito en una tabla en el cual se enfrenta el usuario y el sistema para cada proceso una búsqueda detallada sobre lo que sucede habitualmente por cada actividad del sistema. (Ver ANEXO 2 Especificaciones de Casos de Uso)



**FIGURA 1. DIAGRAMA CASO DE USO: REGISTRO**

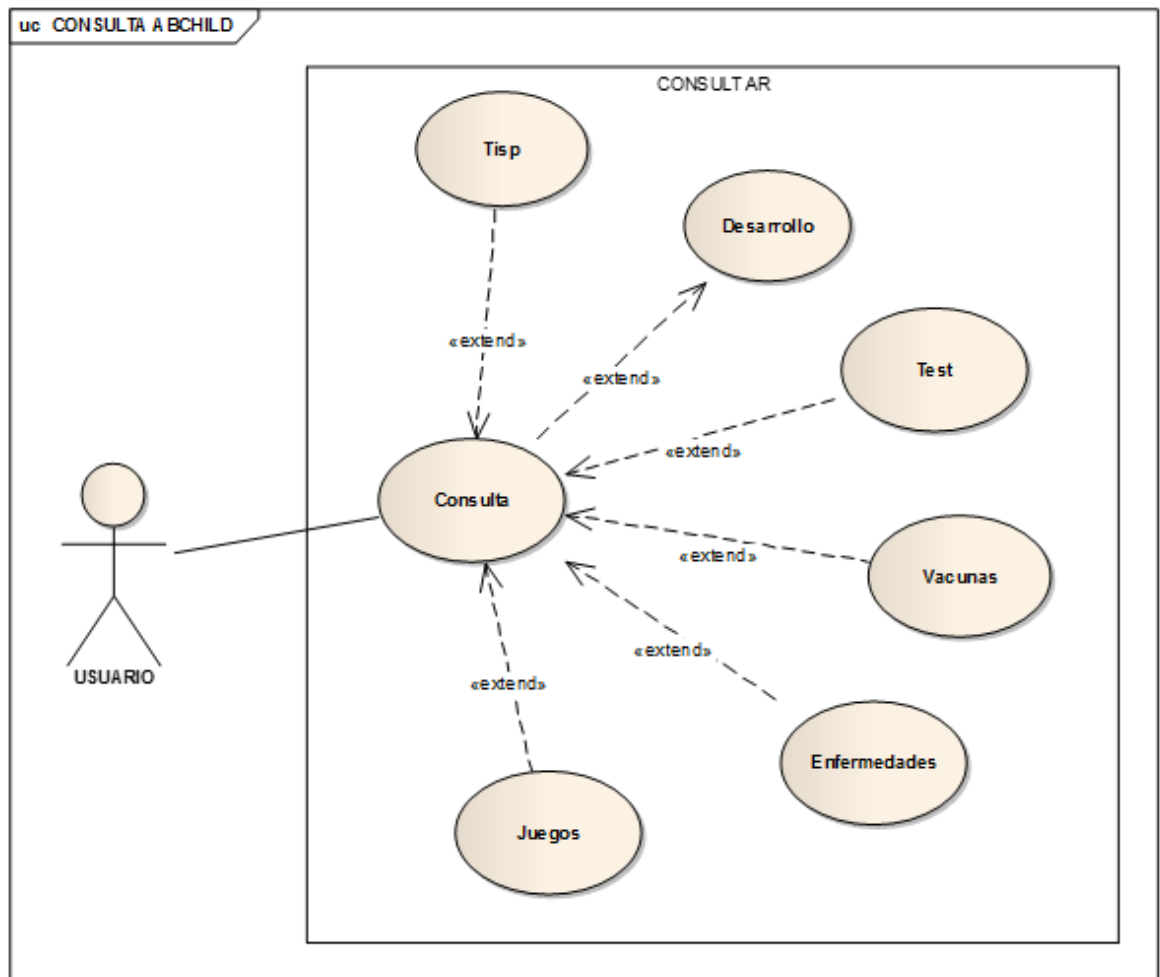
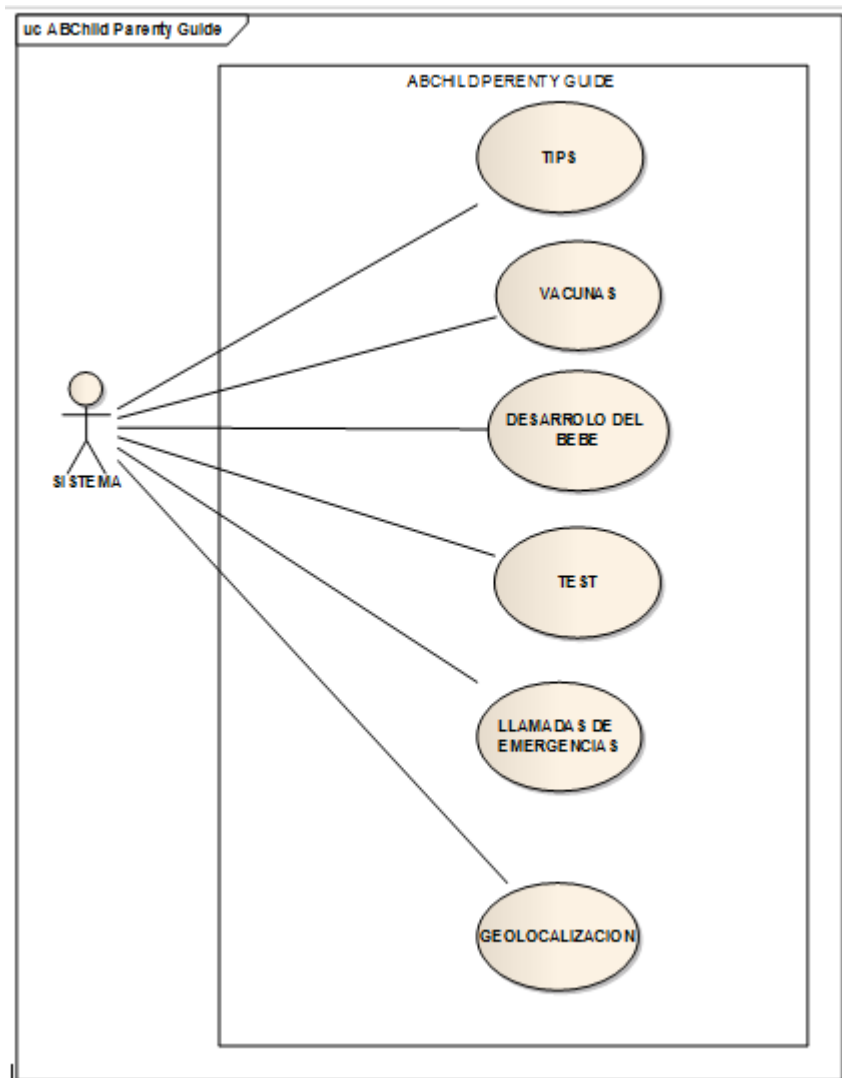
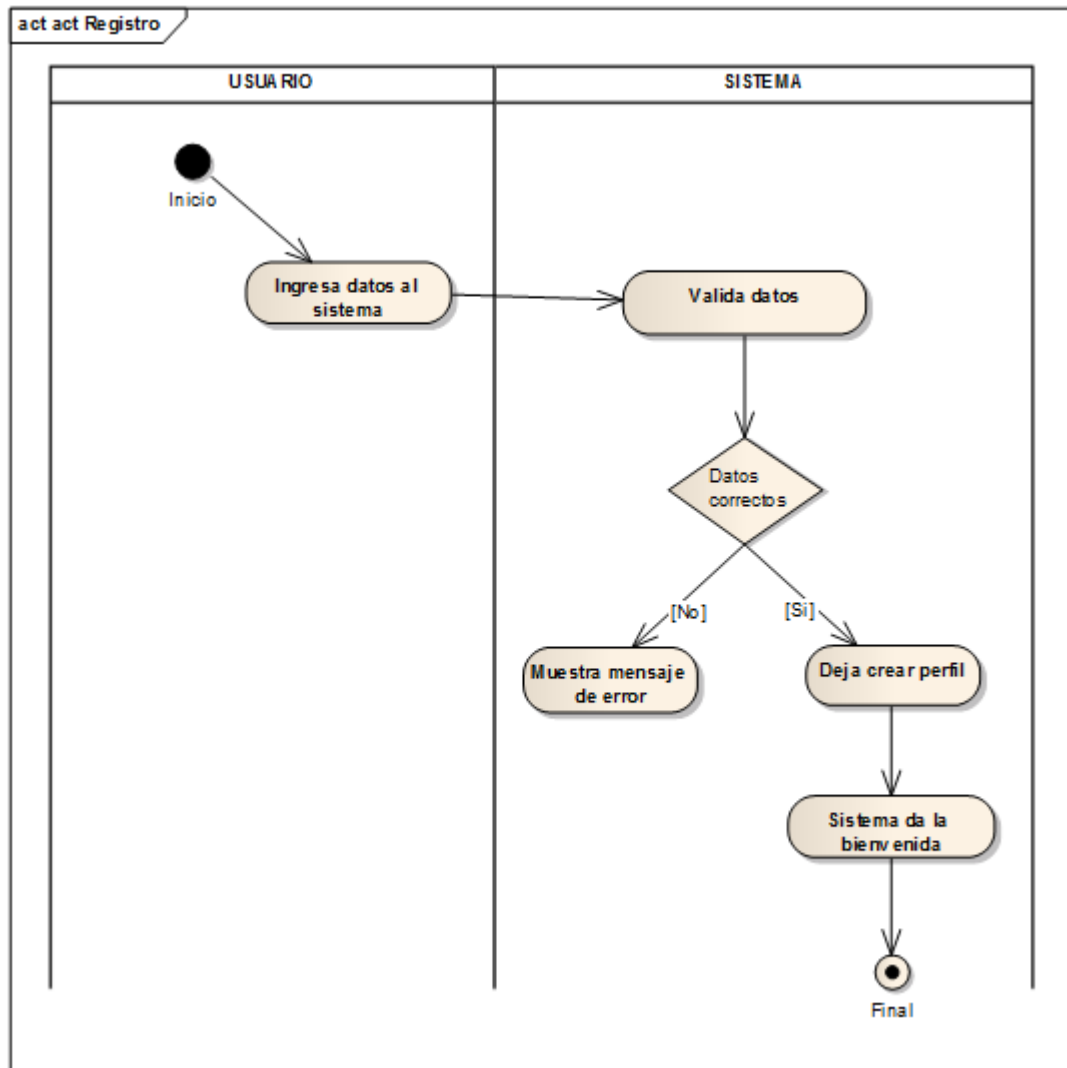


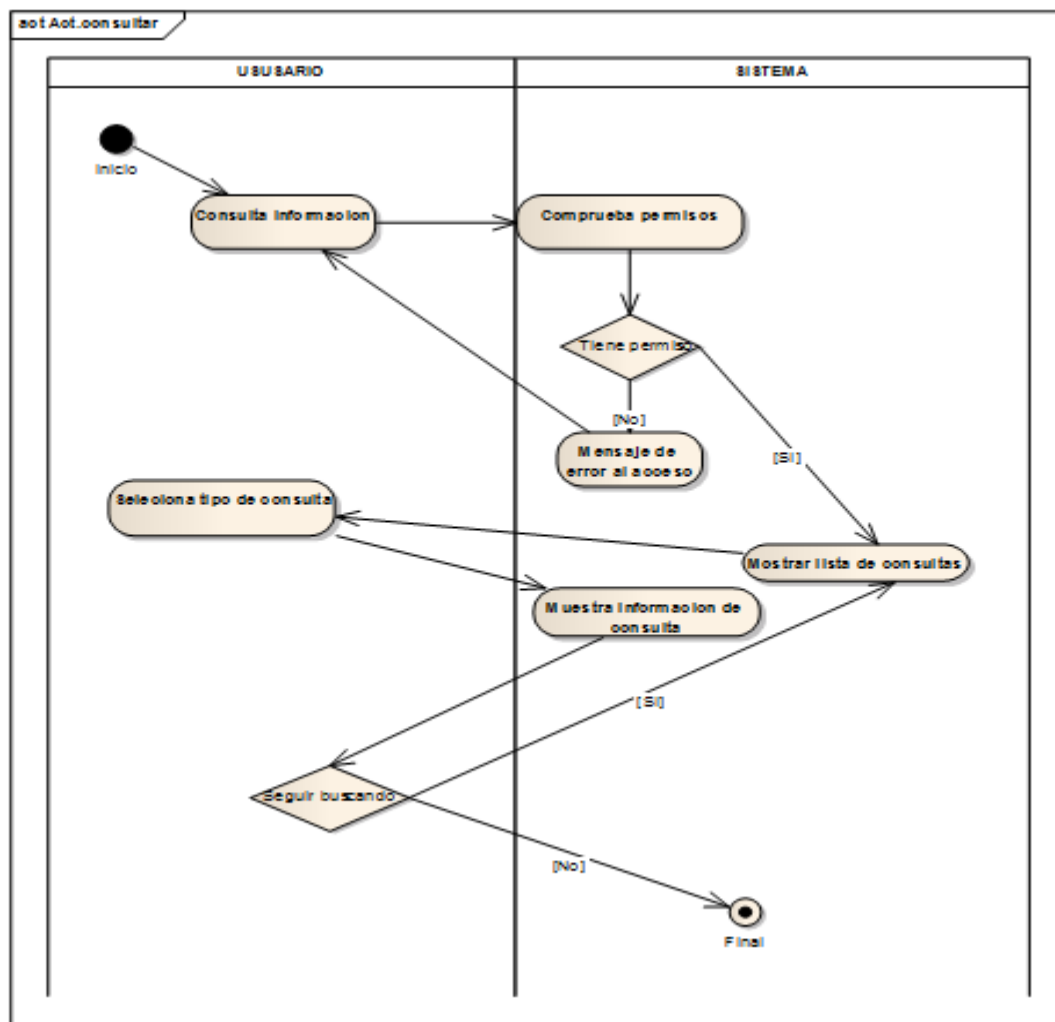
FIGURA 2. DIAGRAMA CASO DE USO: CONSULTAR



**FIGURA 3. DIAGRAMA CASOS DE USO: PRINCIPAL**



**FIGURA 4. DIAGRAMAS DE CLASES REGISTRO**



**FIGURA 5. DIAGRAMA DE CLASES: CONSULTAR**

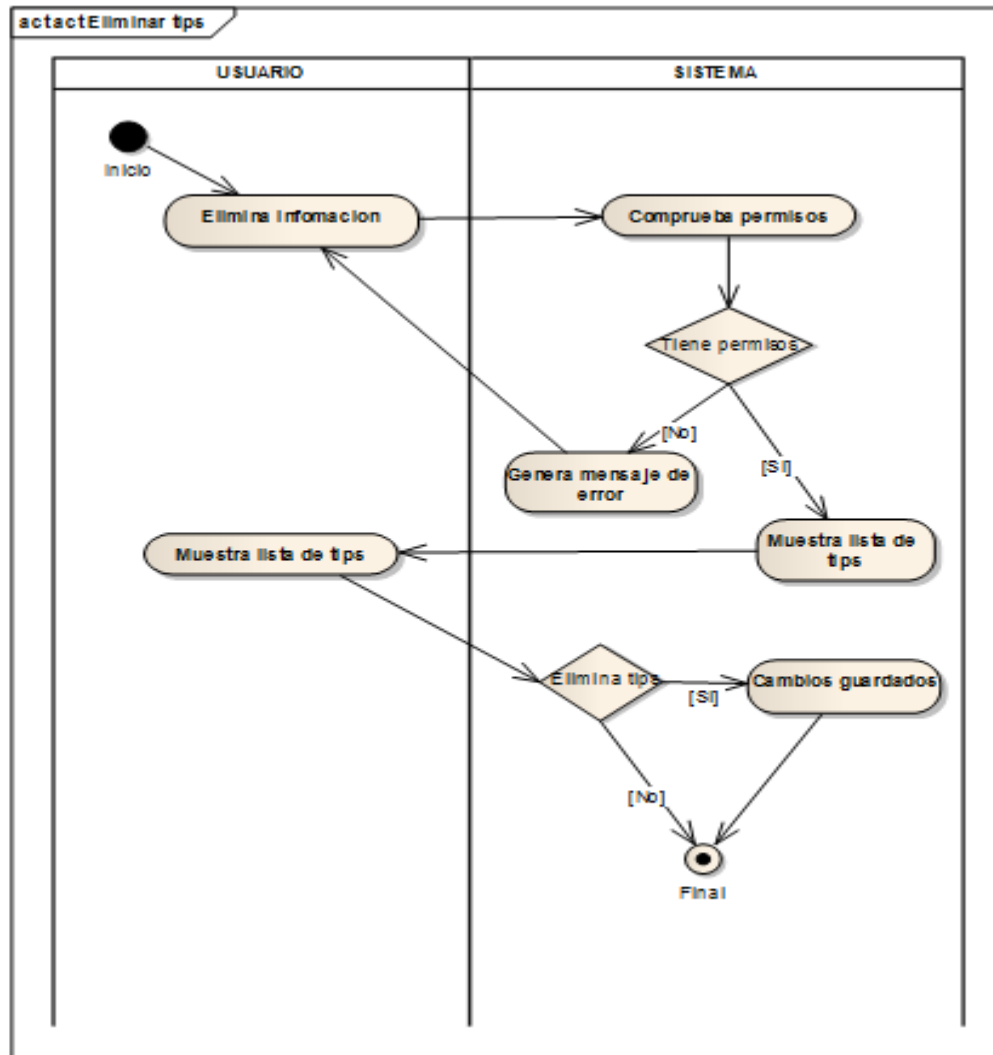
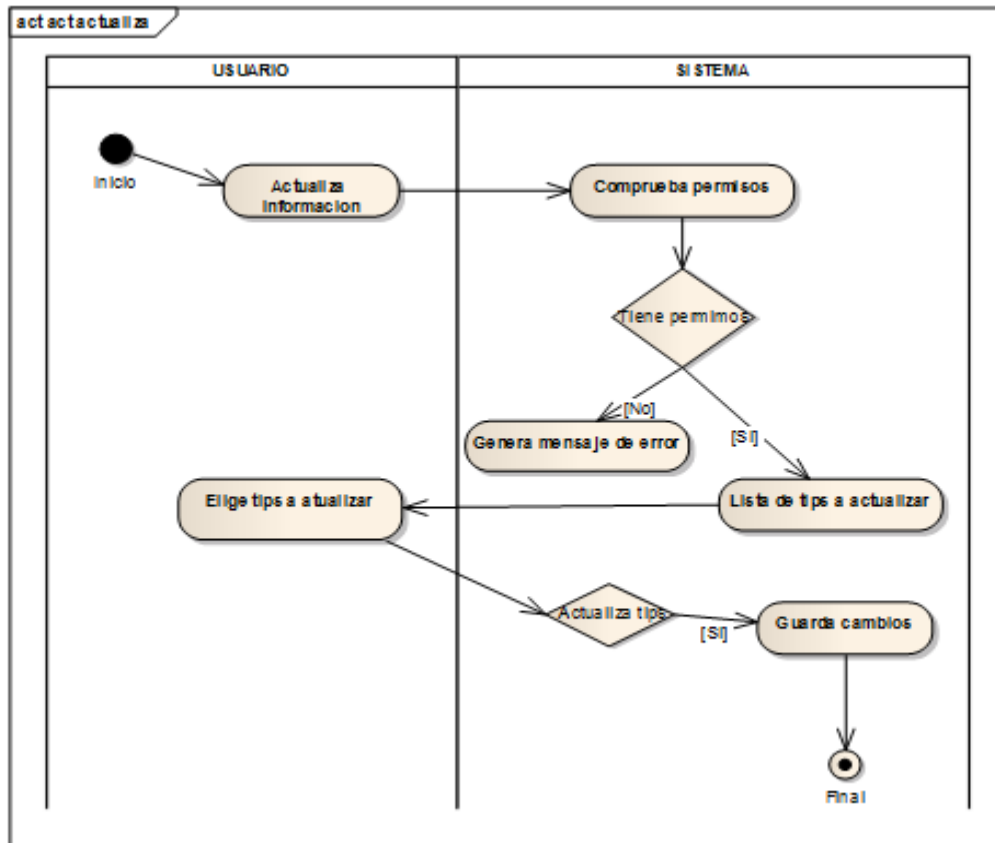


FIGURA 6. DIAGRAMA DE CLASES: ELIMINAR TIPS





**FIGURA 7. DIAGRAMA DE CLASES: ACTUALIZAR**

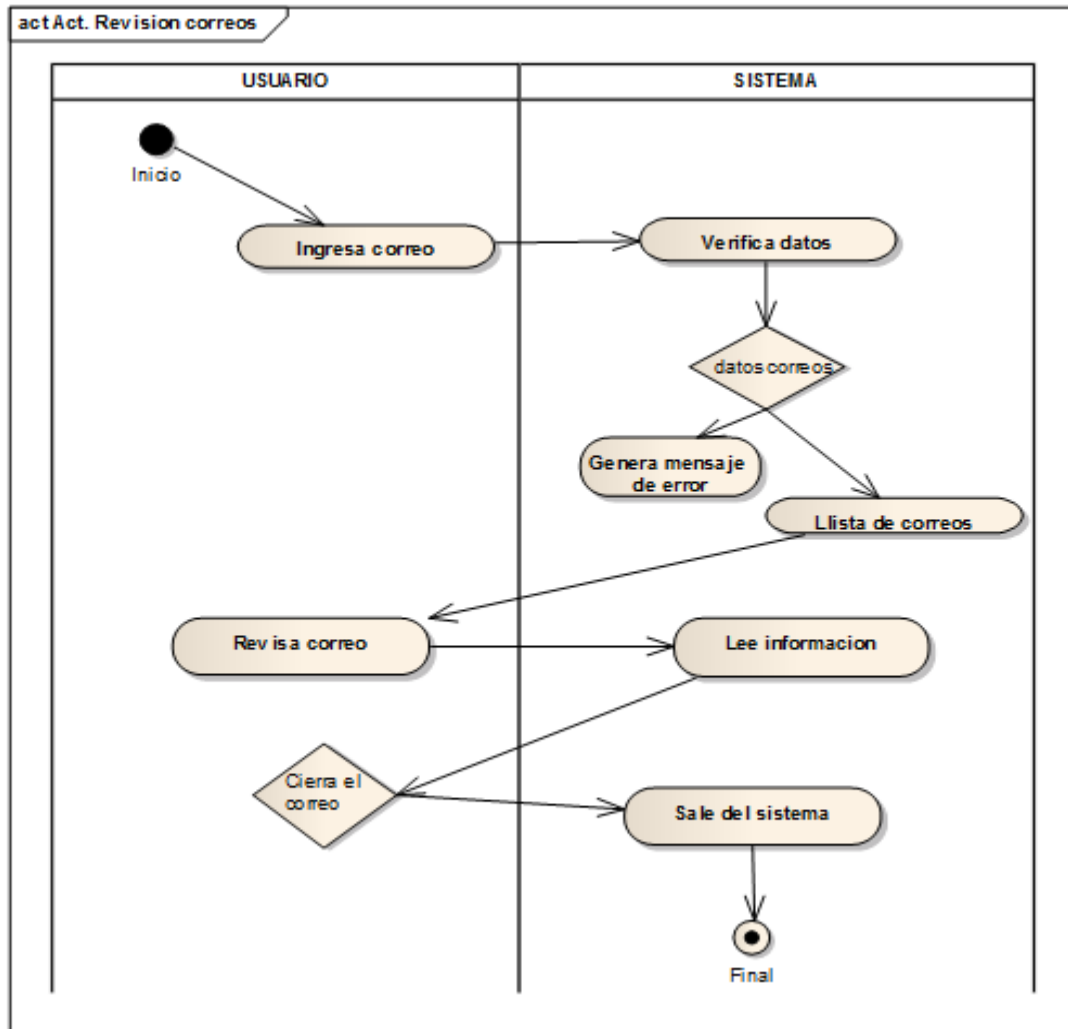
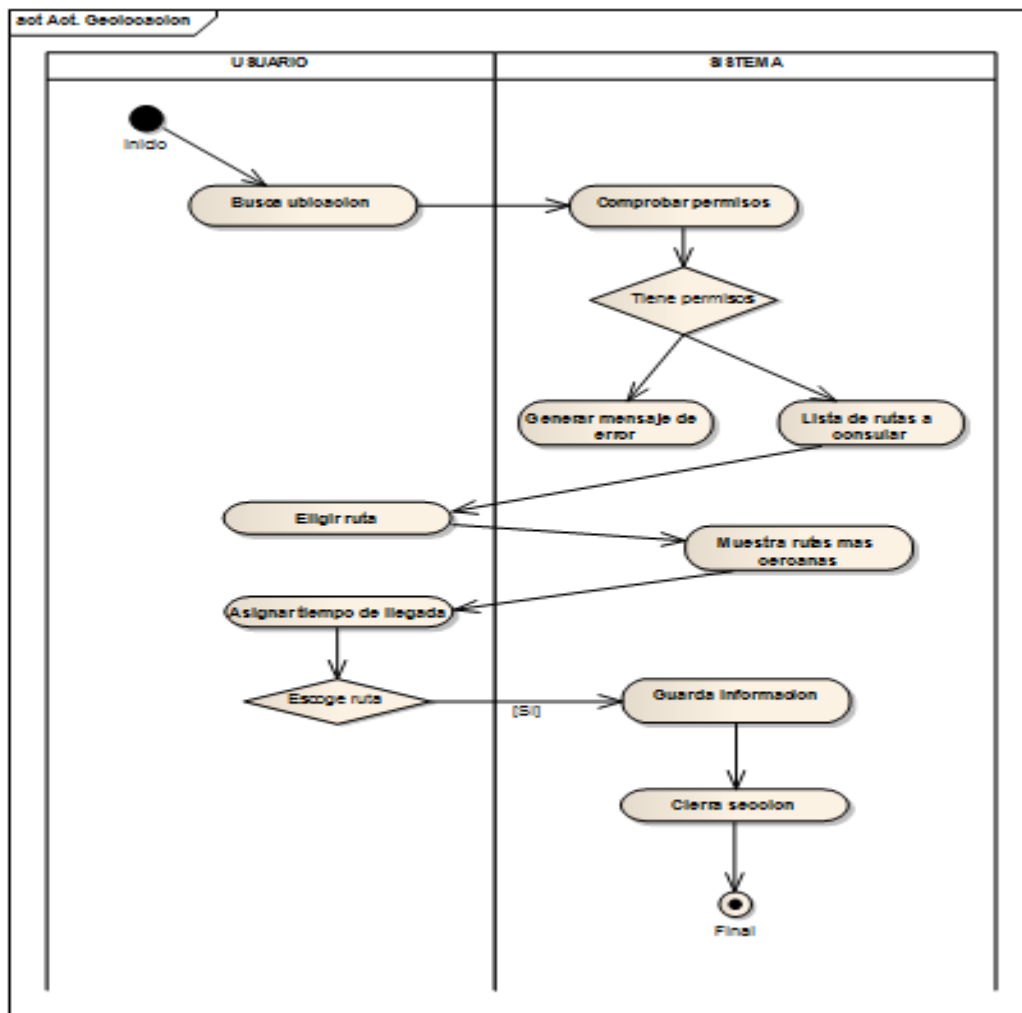


FIGURA 8. DIAGRAMA DE CLASES: REVISION DE CORREOS



**FIGURA 9. DIAGRAMA DE CLASES: GEOLOCALIZACION**

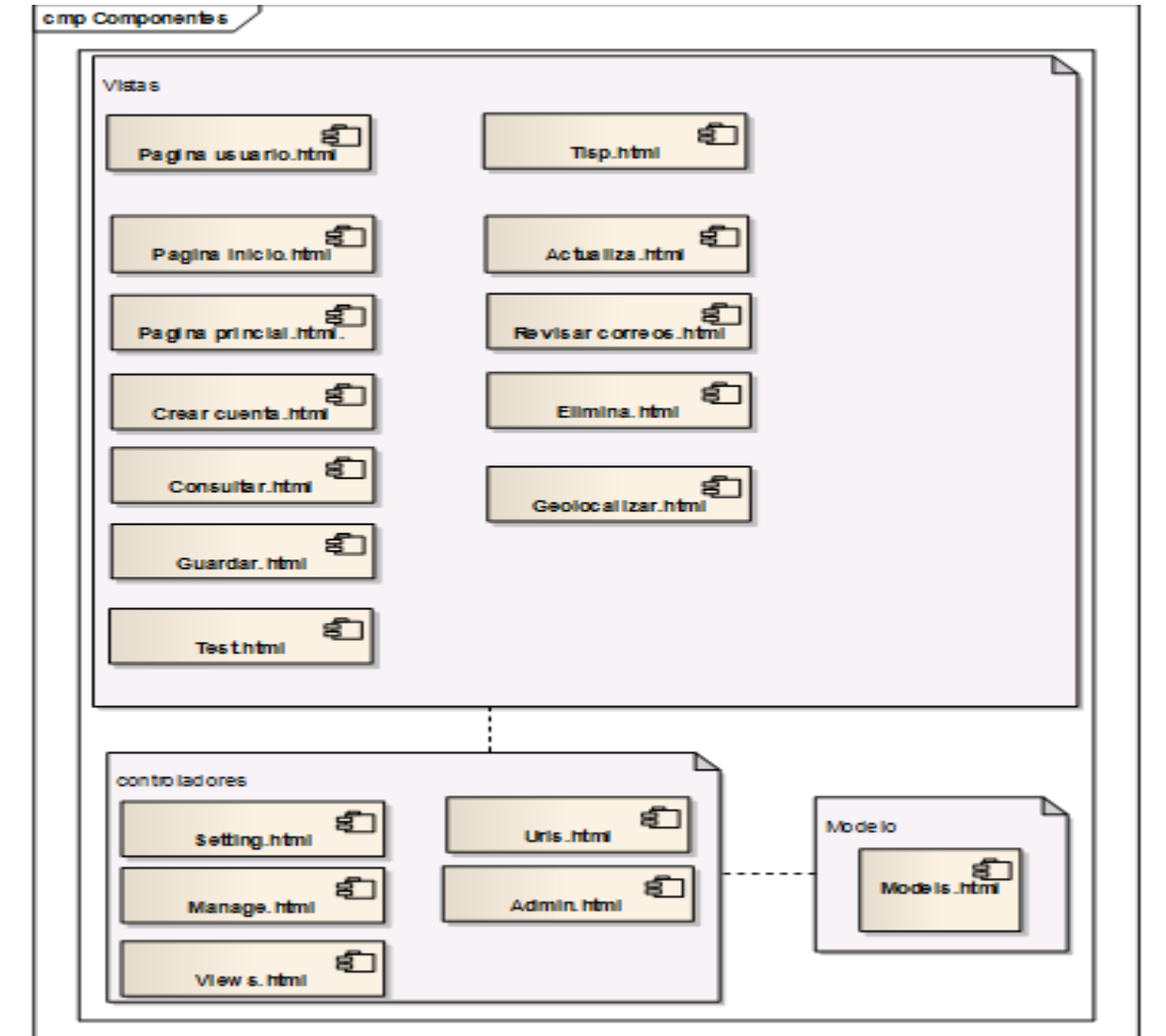
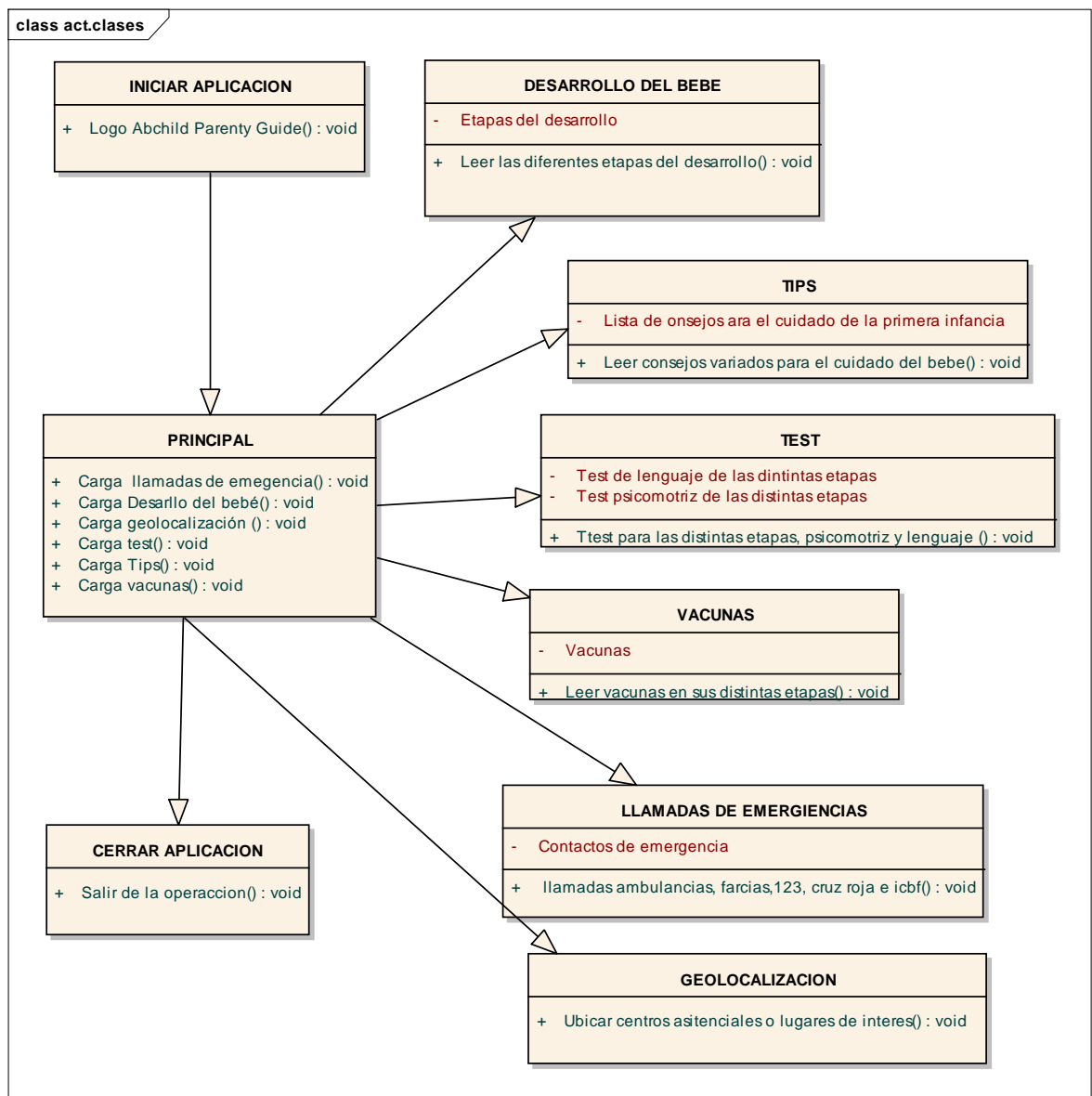


FIGURA 10. DIAGRAMA DE COMPONENTES



**FIGURA 11. DIAGRAMA DE CLASES**

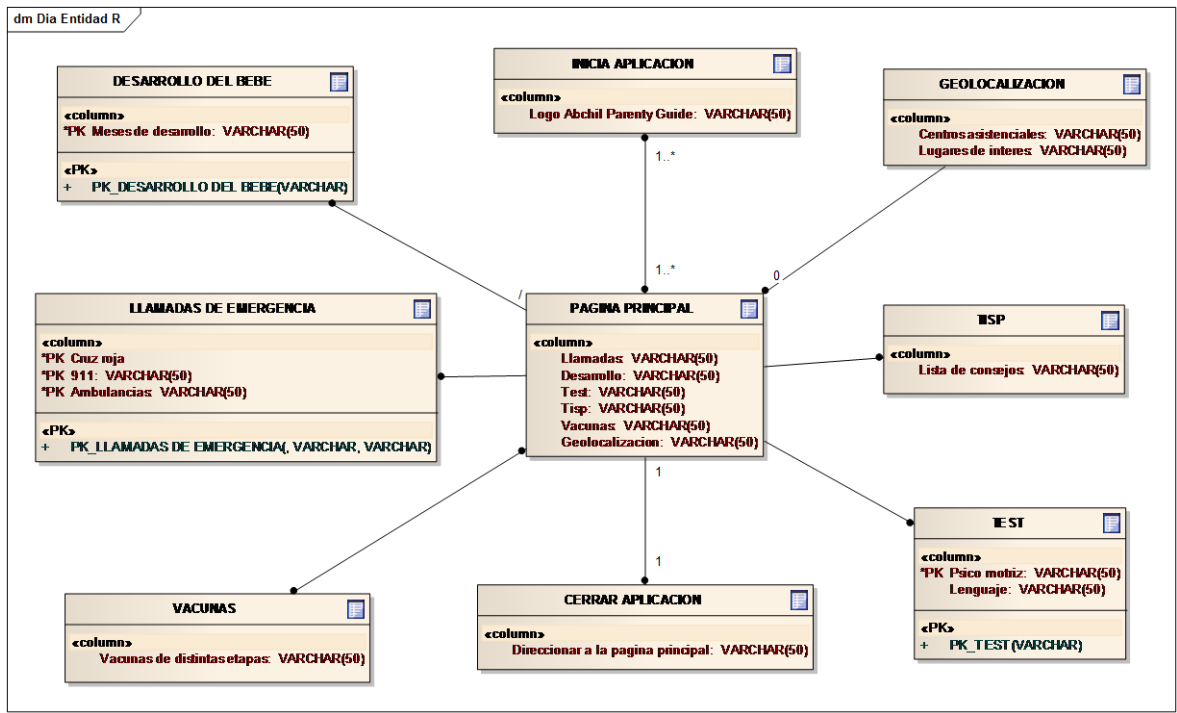
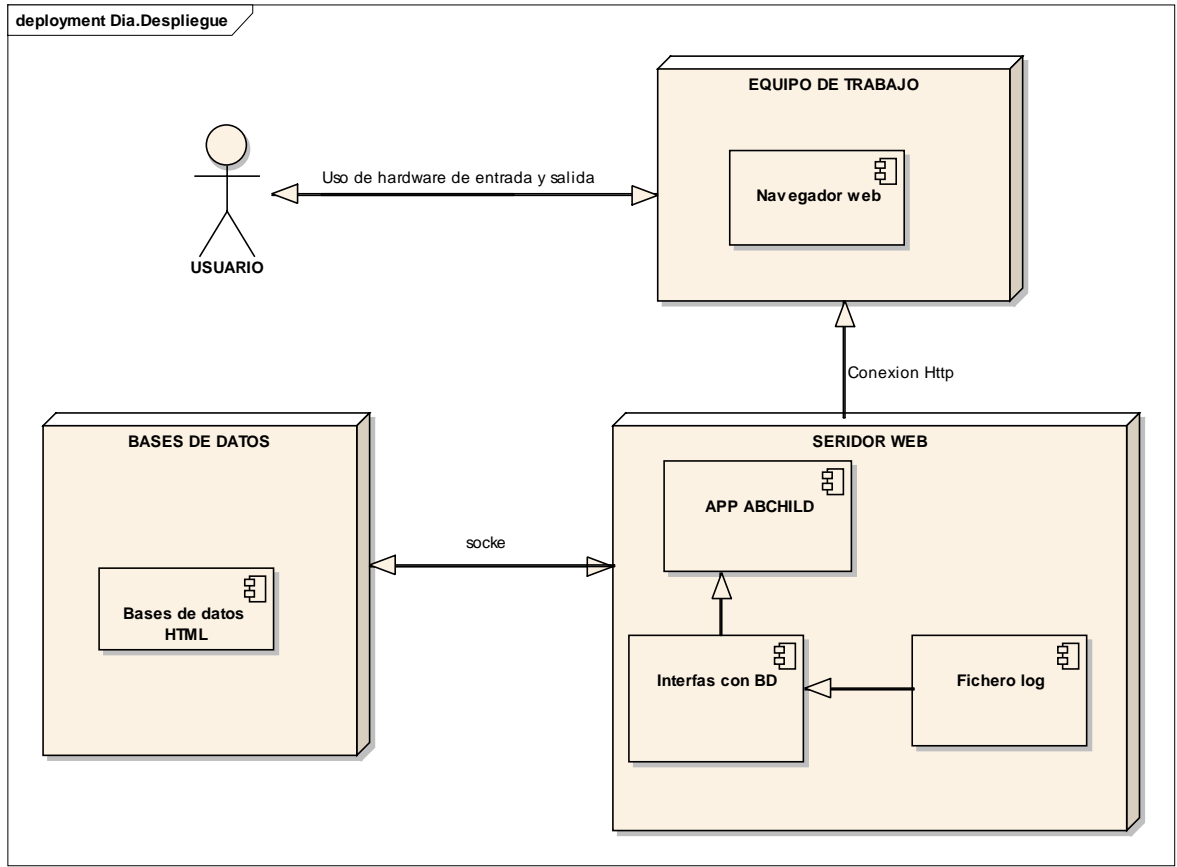


FIGURA 12. DIAGRAMA ENTIDAD RELACIÓN



**FIGURA 13. DIAGRAMA DE DESPLIEGUE**

## 11. Bibliografía

- [1 ]. Emilio Flores Villacrés, Pedro Alvarado Andino y Edgar Wilfrido Ramos Barreno (2015): “La influencia del celular y redes sociales en la vida actual”, Revista Contribuciones a las Ciencias Sociales, n. 27 (enero-marzo 2015). En línea: <http://www.eumed.net/rev/cccss/2015/01/celular-sociedad.html>
- [2]. SERGIO CAMACHO IANNINI(2013): “Comida orgánica, claves de una tendencia en auge Expertos hablan de las bondades del consumo de productos sin químicos.”, Periódico el tiempo (1 de junio de 2013). En línea: <http://www.eltiempo.com/archivo/documento/CMS-12839756>
- [3]. Redacción ABC del bebé (2013): “¿Cuántos pañales usa el bebé y cuánto valen a diario?.”, Revista ABCdelbebe (13 de Marzo de 2013). En línea: <http://www.abcdelbebe.com/etapa/bebe/0-a-6-meses/desarrollo/cuantos-panales-usa-el-bebe-y-cuanto-valen-diario>
- [4]Roger S. Pressman. Ingeniería del Software Un enfoque moderno, McGrawHill, 2010, p. 28
- [5]ibidem, p. 30.
- [6]Ian Sommerville. Ingeniería del Software, PEARSON Educación, 2005, p. 23
- [7] Roger S. Pressman. op. cit., p. 40.
- [8]"INGENIERÍA DE SOFTWARE: MODELO DE PROTOTIPO." 2011. 28 Oct. 2015. En línea: <http://gestionrrhhusm.blogspot.com/2011/05/modelo-de-prototipo.html>



[9] “Modelo Prototipo.” 2013. Abril 28. 2016. En línea:

<https://santmp.files.wordpress.com/2013/03/modelo-de-prototipo.pdf>

[10] Roger S. Pressman. op. cit., p. 55

[11] Roger S. Pressman. op. cit., p. 60

[12] Roger S. Pressman. op. cit., p. 65

[13] Mike Cohn, 2004 "User Stories Applied", , Addison Wesley

[14] Roger S. Pressman. op. cit., p. 184

[15] Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Programación en C, C++, Java y UML, 2010, p. 454

[16] "THE 4+1 VIEW MODEL OF ARCHITECTURE." . 6 May. 2016. En línea:

[http://docente.ucol.mx/almoradi/public\\_html/Respaldo/resumen3.htm](http://docente.ucol.mx/almoradi/public_html/Respaldo/resumen3.htm)

[17] Ian Sommerville. op. cit., p. 730.

[18] Martin Fowler con Kendall Scott. op. cit., p. 62

[19] Benet Campderrich Falgueras. Ingeniería de software, Editorial UOC, 2003. p.102

[20] Ian Sommerville. op. cit., p. 735.

[21] Benet Campderrich Falgueras. op. cit. p. 105.

[22] Benet Campderrich Falgueras. op. cit. p. 105..

[23] Roger S. Pressman. op. cit., p. 2

[24] “Patrón de arquitectura Modelo Vista Controlador (MVC).” , May. 15 2016. En

línea: <http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>

[25] “Codificación del software.” 2009, May. 15 2016. En línea

<https://ciclodevidasoftware.wikispaces.com/codificacion+del+software>

[26] Ian Sommerville. op. cit., p. 90.

[27] Ian Sommerville. op. cit., p. 90.

[28] “Guía breve de estándares web.” May. 12 2016. En línea

<http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>

[29] “Entendidendo html5: guia para principiantes”, 28 de mayo de 2013. En línea

<https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>

[30] “Guía breve de CSS.” May. 12 2016. En línea:

<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

[31] Guía de iniciación a APP INVENTOR (2015), op. cit., p. 4

[32] Introducción a Visual Studio, 2010. En línea:

[https://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.100).aspx)



