



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

TEMA:

“ANÁLISIS DE LA PLATAFORMA GWT PARA DESARROLLO DE
APLICACIONES DE GEO-REFERENCIACIÓN PARA PERSONAS CON
DISCAPACIDAD VISUAL”

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo
la obtención del título de Ingeniera en Sistemas Computacionales e Informáticos.

SUBLINEA DE INVESTIGACIÓN: Aplicaciones WEB

AUTOR: Sinthia Elizabeth Rivadeneira Arregui

TUTOR: Ing. Mg. Carlos Israel Núñez Miranda

Ambato – Ecuador

Septiembre/2017

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: “**ANÁLISIS DE LA PLATAFORMA GWT PARA DESARROLLO DE APLICACIONES DE GEO-REFERENCIACIÓN PARA PERSONAS CON DISCAPACIDAD VISUAL**”, de la señorita Sinthia Elizabeth Rivadeneira Arregui, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato septiembre, 2017

EL TUTOR



Ing. Mg. Carlos Israel Núñez Miranda

AUTORÍA

El presente Proyecto de Investigación titulado: **“ANÁLISIS DE LA PLATAFORMA GWT PARA DESARROLLO DE APLICACIONES DE GEO-REFERENCIACIÓN PARA PERSONAS CON DISCAPACIDAD VISUAL”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato septiembre, 2017



Sinthia Elizabeth Rivadeneira Arregui

CC: 020198107-3

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación, como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato septiembre, 2017



Sinthia Elizabeth Rivadeneira Arregui


CC: 020198107-3

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes: Mg. Hernando Buenaño, Mg. Oswaldo Paredes, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “ANÁLISIS DE LA PLATAFORMA GWT PARA DESARROLLO DE APLICACIONES DE GEO-REFERENCIACIÓN PARA PERSONAS CON DISCAPACIDAD VISUAL”, presentado por la señorita Sinthia Elizabeth Rivadeneira Arregui de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



PRESIDENTE DEL TRIBUNAL



DOCENTE CALIFICADOR



DOCENTE CALIFICADOR

DEDICATORIA

Esta tesis dedico a Dios por haberme regalado la vida, por ser mi fortaleza en mis momentos de debilidad y por hacer de mi viaje en este mundo lleno de experiencia, felicidad y permitirme el haber llegado hasta este momento tan importante de mi formación profesional.

A mis padres, porque creyeron en mí y me apoyaron para emprender y hacer realidad cada sueño dandome ese ejemplo digno de admiración y gracias a ello hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles y porque el orgullo que sienten por mi fue lo que hizo que culminara con éxito. Va por ustedes, por lo que valen, porque admiro su fortaleza, trabajo y por todo lo que han hecho y son capaces de hacer por mi.

Sinthia Elizabeth Rivadeneira Arregui

AGRADECIMIENTO

Agradezco a todo el personal tanto docente como administrativo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por brindarme sus conocimientos y darme la oportunidad de haberme formado profesionalmente.

Por último, un sincero agradecimiento al Ing. Carlos Núñez por su guía en el desarrollo del proyecto y compartir sus conocimientos.

Sinthia Elizabeth Rivadeneira Arregui

Índice de Contenidos

CAPÍTULO 1	1
EL PROBLEMA	1
1.1 Tema de Investigación.....	1
1.2 Planteamiento del Problema.....	1
1.3 Delimitación	2
1.4 Justificación.....	2
1.5 Objetivos	3
1.5.1 Objetivo General.....	3
1.5.2 Objetivos Específicos	3
CAPÍTULO 2	5
MARCO TEÓRICO	5
2.1 Antecedentes Investigativos	5
2.2 Fundamentación Teórica	6
2.2.1 Plataforma GWT.....	6
2.2.2 Introducción	6
2.2.3 Definición de GWT	7
2.2.4 Características de GWT	8
2.2.5 Arquitectura GWT	8
2.2.6 Funciones y Funcionamiento de GWT	12
2.2.5.1 Funciones	12
2.2.5.2 Funcionamiento	13
2.2.6 Ciclo de desarrollo	14
2.2.7 Modelo de programación en GWT	14
2.2.8 Estructura de un proyecto GWT	15
2.2.9 Ventajas y Desventajas de GWT	16
2.2.9.1 Ventajas de GWT.....	16
2.2.9.2 Desventajas de GWT	17
2.2.10 Base de Datos.....	17
2.2.11 Sistema de Gestión de Bases de Datos	17

2.2.12	Metodología Ágil	18
2.2.13	Metodología de desarrollo XP	19
2.2.14	Análisis Comparativo de IDE's	20
CAPÍTULO 3.....		24
METODOLOGÍA		24
3.1	Modalidad Básica de Investigación.....	24
3.2	Población y Muestra	24
3.3	Recolección de Información.....	25
3.4	Procesamiento y Análisis de Datos	25
3.5	Desarrollo del Proyecto	25
CAPITULO 4.....		26
DESARROLLO DE LA PROPUESTA		26
4.1	Planificación del proyecto	26
-	Recolección de la información.....	26
4.2	Diseño.....	27
4.3	Codificación	28
-	Desarrollo de la aplicación.....	28
4.4	Pruebas	109
4.4.1	Pruebas de caja negra.....	109
4.4.2	Pruebas de caja blanca	109
CAPITULO 5.....		111
CONCLUSIONES Y RECOMEDACIONES.....		111
5.1	Conclusiones	111
5.2	Recomendaciones	112
Bibliografía		113
Anexos		115

Índice de Gráficos

Gráfico 1. Arquitectura GWT	8
Gráfico 2. Clases fundamentales.....	11
Gráfico 3. Herramientas en el desarrollo de aplicaciones Java.....	11
Gráfico 4. Funcionamiento de GWT.....	13
Gráfico 5. Instalación de repositorios.	28
Gráfico 6. Instalación de repositorios.	29
Gráfico 7. Add Repository.	29
Gráfico 8. Opciones del repositorio.	29
Gráfico 9. Opciones de instalación de plugin de GWT.	30
Gráfico 10. Gráfico 10. Opciones de GWT.	30
Gráfico 11. Gráfico 11. Creación de un proyecto de GWT.	31
Gráfico 12. Datos para la creación de la aplicación.....	31
Gráfico 13. Estructura del proyecto	32
Gráfico 14. Compilación de un proyecto GWT.....	33
Gráfico 15. Formas de compilación de un proyecto GWT.....	33
Gráfico 16. Instalación del plugin GWT en Firefox.	34
Gráfico 17. Vistazo de una aplicación web en el navegador.	34
Gráfico 18. Configuración de GXT	35
Gráfico 19. Gráfico 19. Página de descarga GWT.....	36
Gráfico 20. Localización de la carpeta gxt-2.2.5.	36
Gráfico 21. Gráfico 21. Verificación de la versión de Java.....	37
Gráfico 22. Creación del Servicio Remoto	38
Gráfico 23. Especificación del nombre del servicio remoto	38
Gráfico 24. Creación de la interfaz de la clase asincrónica	39
Gráfico 25. Creación de una clase.....	40
Gráfico 26. Creación de una nueva clase de java.....	41
Gráfico 27. Componentes GXT: Window	41
Gráfico 28. Parte visual de la ventana.....	42
Gráfico 29. Componentes GXT	42
Gráfico 30. GXT panels	43
Gráfico 31. GXT layouts.....	44
Gráfico 32. GXT widgets.....	46
Gráfico 33. GXT forms	48
Gráfico 34. Gráfico 30. GXT menu	50
Gráfico 35. GXT toolbar	51
Gráfico 36. GXT layouts.....	52
Gráfico 37. Añade el diseño “Border Layout”.....	52
Gráfico 38. Componentes GXT “Border Layout”.	53
Gráfico 39. Gráfico 32. Vista General del Sistema.....	53

Gráfico 40. Creación del servicio remoto.	54
Gráfico 41. Asigna el nombre del servicio remoto.	55
Gráfico 42. Creación de la clase asincrónica RPCApplicationAsync	55
Gráfico 43. Clase asincrónica RPCApplicationAsync.....	56
Gráfico 44. Añade el jar al Path, para crear una conexión con la BD.	56
Gráfico 45. Código de conexión de la Base de Datos.....	57
Gráfico 46. Creación del método Login.....	57
Gráfico 47. Genera el método en la clase asincrónica	58
Gráfico 48. Creación del método Login en la clase asincrónica.....	58
Gráfico 49. Llamado del método Login de la clase asincrónica.	58
Gráfico 50. Llamado del método Login en la clase Inicio.	59
Gráfico 51. Llamado del método Login en el botón Ingresar.	59
Gráfico 52. Vista General del Sistema.....	60
Gráfico 53. Form Panel GPS.....	61
Gráfico 54. Combo box Personas	61
Gráfico 55. Data picker, fecha fin.	61
Gráfico 56. Time field, hora inicio y fin.	62
Gráfico 57. Propiedades de time field.....	62
Gráfico 58. Visualización de la búsqueda.....	63
Gráfico 59. Biblioteca Open Layers.....	63
Gráfico 60. Librería jar – manipulación de mapas.....	64
Gráfico 61. Código javascript – OpenLayers de GWT.....	64
Gráfico 62. Código OpenLayers - mapas.....	65
Gráfico 63. Archivo de configuración openlayers.	66
Gráfico 64. Llamado en el tab panel de los mapas.	66
Gráfico 65. Marcadores de ubicación en el mapa.	67
Gráfico 66. Código de mapa por defecto.	67
Gráfico 67. Vista del mapa por defecto.	68
Gráfico 68. Código del mapa por defecto.	68
Gráfico 69. Creación de la clase para el paso de parámetros.....	69
Gráfico 70. Ingreso de datos de la clase.....	70
Gráfico 71. Código de ID serializable.	70
Gráfico 72. Solución de Advertencia.	71
Gráfico 73. Declaración de las variables de la BD.	71
Gráfico 74. Métodos y funciones para manipular las variables.	72
Gráfico 75. Generate Getters and Setters.....	72
Gráfico 76. Método Getters and Setters.....	73
Gráfico 77. Consulta de coordenadas.....	74
Gráfico 78. Código while.....	74
Gráfico 79. Método de la clase RPCApplication Server.....	75
Gráfico 80. Advertencia: creación de métodos en la clase asincrónica	75
Gráfico 81. Creación de métodos en la clase asincrónica.....	75

Gráfico 82. Método para graficar los marcadores en el mapa.	76
Gráfico 83. Visualización del mapa.	76
Gráfico 84. Código para graficar el mapa.	76
Gráfico 85. Código del botón buscar.	77
Gráfico 86. Código para crear una lista.	77
Gráfico 87. Uso de la lista creada anteriormente.	78
Gráfico 88. Error de método – clase asíncrona.	78
Gráfico 89. Método llenar combo box personas.	78
Gráfico 90. Método llenar personas.	79
Gráfico 91. Añade la lista al combo box.	79
Gráfico 92. Interacción con el combo box.	79
Gráfico 93. Ingreso de un nuevo usuario.	80
Gráfico 94. Código para controlar el ingreso de los datos de usuarios.	80
Gráfico 95. Perfiles de ingreso al sistema.	81
Gráfico 96. Creación de una clase nueva, para el componente.	81
Gráfico 97. Creación de una clase con herencia.	82
Gráfico 98. Creación de una clase con herencia.	82
Gráfico 99. Mensaje de error del correo electrónico.	83
Gráfico 100. Elementos de la paleta.	83
Gráfico 101. Datos del componente.	84
Gráfico 102. Carpeta Custom – componente propio.	84
Gráfico 103. Código para insertar usuarios.	85
Gráfico 104. Solución de error, creación del método.	85
Gráfico 105. Generación del método en la clase asíncrona.	86
Gráfico 106.. Trigger para la inserción de datos en la tabla personas.	86
Gráfico 107. Vista del sistema – Perfil Institucional.	87
Gráfico 108. Ingreso de personas.	87
Gráfico 109. Código para el ingreso de personas.	88
Gráfico 110. Código para la inserción de datos de personas.	88
Gráfico 111. Error en la clase RPCApplicationAsync.	89
Gráfico 112. Solución del error en la clase RPCApplicationAsync.	89
Gráfico 113. Código del método insertarDatos.	89
Gráfico 114. Datos Personales.	90
Gráfico 115. Actualizar datos del usuario.	91
Gráfico 116. Error en la clase RPCApplicationAsync.	91
Gráfico 117. Solución del error en la clase RPCApplicationAsync.	92
Gráfico 118. Código del método actualizarDatosUsuario.	92
Gráfico 119. Trigger para actualizar datos del perfil personal.	93
Gráfico 120. Vista General del Sistema - Principal.	93
Gráfico 121. Gráfico 112. Registro de personas.	94
Gráfico 122. Mapa de búsqueda.	94
Gráfico 123. Marcadores de búsqueda.	95

Gráfico 124. Nuevo proyecto Android.....	95
Gráfico 125. Versión de Android.....	96
Gráfico 126. Plantilla Login.....	96
Gráfico 127. Programación activity_login.....	97
Gráfico 128. Vista principal - Login.....	98
Gráfico 129. Vista principal - Login.....	98
Gráfico 130. Vista principal - Login.....	99
Gráfico 131. Clase conexión.....	99
Gráfico 132. Código de la clase conexión.....	100
Gráfico 133. Activity_lista_personas.....	100
Gráfico 134. Acceso del Sistema.....	101
Gráfico 135. Actividad Principal.....	101
Gráfico 136. Posición del dispositivo.....	102
Gráfico 137. Detener Proceso.....	102
Gráfico 138. Habilitar permisos del GPS.....	103
Gráfico 139. Activity_buscar.....	103
Gráfico 140. Iniciar la búsqueda.....	104
Gráfico 141. Actividad Buscar.....	104
Gráfico 142. Ubicación mediante marcadores.....	104
Gráfico 143. Activity_mapa.....	105
Gráfico 144. Acceso a mapas de Google.....	106
Gráfico 145. Uso de los mapas de Google.....	106
Gráfico 146. Mapas de Google.....	107
Gráfico 147. Mapas de Google.....	107
Gráfico 148. Mapas de Google.....	108
Gráfico 149. Ubicación con marcadores en mapas de Google.....	108

Índice de Tablas

Tabla 1. Estructura del Proyecto	15
Tabla 2. Análisis de comparación de IDE's.....	22
Tabla 3. Formulario de Ingreso al Sistema	26
Tabla 4. Formulario de Búsqueda	27

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERIA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
INGENIERIA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

“ANÁLISIS DE LA PLATAFORMA GWT PARA DESARROLLO DE
APLICACIONES DE GEO-REFERENCIACIÓN PARA PERSONAS CON
DISCAPACIDAD VISUAL”

Autor: Sinthia Elizabeth Rivadeneira Arregui.

Tutor: Ing. Mg. Carlos Israel Núñez Miranda.

Fecha: 07 de septiembre del 2017.

RESUMEN EJECUTIVO

Se realizó en análisis de la plataforma GWT para el desarrollo de una aplicación web que permite la localización de personas con discapacidad visual. Para el desarrollo de la investigación se realizó el estudio teórico de conceptos para entender el funcionamiento del framework GWT, se estudió todo sobre el lenguaje de desarrollo JAVA como: definiciones y características, la máquina virtual de java (JVM), el JRE, entre otros aspectos. Además se utilizó el IDE Eclipse y la plataforma GWT, en donde se desarrolló la aplicación web con el objetivo de facilitar a entrenadores y familiares a localizar a las personas con discapacidad visual que forman parte de su entorno.

Para la realización de la investigación se realizó encuestas a los entrenadores y familiares de las personas con discapacidad, así como también entrevistas a las personas que poseen la discapacidad, dando lugar al análisis de los procesos, para concluir con el desarrollo de la aplicación web, cumpliendo los requerimientos.

Para el desarrollo se eligió Eclipse como el IDE más adecuado porque permite el desarrollo de la aplicación web usando la plataforma GWT, esta plataforma es de gran ayuda para los desarrolladores de JAVA, porque permite programar solo en este lenguaje sin preocuparse del código HTML y JavaScript reduciendo el tiempo de desarrollo.

Palabras Clave: Plataforma, Aplicación WEB, Discapacidad, IDE, JAVA.

INTRODUCCIÓN

El desarrollo del informe final del presente trabajo de investigación se encuentra dividido en capítulos, los mismos que facilitan la comprensión del contenido:

Capítulo I.- Denominado “EL PROBLEMA”, se encuentra el problema a investigar, la justificación de la investigación y el planteamiento de los objetivos a obtener tras la culminación del trabajo de investigación.

Capítulo II.- Contiene el “MARCO TEÓRICO”, en donde se establece el conjunto de conocimientos que sustentan la investigación, también contiene investigaciones previas que sirven de soporte a la investigación y se establece la propuesta de solución del problema.

Capítulo III.- Comprende la “METODOLOGÍA”, en la cual se especifica la metodología de investigación a utilizar, el proceso de recolección de la información, el procesamiento y análisis de la información y define las etapas para el desarrollo del proyecto.

Capítulo IV.- Consta del “DESARROLLO DE LA PROPUESTA”, que detalla la el análisis de la plataforma para el desarrollo de aplicaciones.

Capítulo V.- Contiene las conclusiones y recomendaciones a las que llega el investigador mediante el desarrollo del proyecto.

CAPÍTULO 1

EL PROBLEMA

1.1 Tema de Investigación

“Análisis de la plataforma GWT para desarrollo de aplicaciones de geo-referenciación para personas con discapacidad visual”

1.2 Planteamiento del Problema

En la actualidad muchos desarrolladores desconocen este framework y su utilización, GWT es una herramienta de fácil manejo para crear sitios web, con la ventaja de que no necesita de conocimientos avanzados acerca de AJAX y HTML.

Google Web Toolkit (GWT) es un framework de desarrollo en Java de código abierto, que permite escapar de las tecnologías usadas actualmente para desarrollar aplicaciones AJAX las mismas que son propensas a errores así como también son difíciles de utilizar.[5]

El framework GWT permite desarrollar y depurar aplicaciones AJAX usando el lenguaje de programación Java en el entorno de desarrollo de nuestra preferencia, es decir el sistema operativo y los IDE's. [5]

Una vez terminada la aplicación desarrollada en Java, GWT permite compilar y traducir dicho programa a JavaScript y HTML el mismo que será compatible con cualquier navegador web. [5]

En la ciudad de Ambato existe un club llamado “FÉNIX” que está conformado por personas con discapacidad visual, es un club deportivo el mismo que se encarga de realizar caminatas y participar en competencias deportivas, pero la dificultad que tiene al realizar este tipo de actividades es que no saben dónde se

encuentran sus deportistas al momento de participar en las actividades en las que participa el club, para ello es necesario que cada participante asista con un familiar por lo que es necesario y factible que el club cuente con un sistema que permita conocer la ubicación de cada uno de sus integrantes.

Con el desarrollo de este proyecto los integrantes del club “Fénix” se beneficiarán, porque conocerán la ubicación en tiempo real de los deportistas que forman parte del club.

El proyecto también podrá ser utilizado como una guía para los estudiantes de la carrera de ingeniería en sistemas, porque les guiara de forma clara y sencilla en el desarrollo de aplicaciones optimizando el tiempo de desarrollo, así como también conocer un poco más acerca de la plataforma GWT.

1.3 Delimitación

Delimitación del Contenido

- **Área Académica:** Software
- **Línea De Investigación:** Desarrollo de Software
- **Sub-Línea De Investigación:** Aplicaciones WEB

Delimitación Espacial

- La presente investigación se la realizó en la ciudad de Ambato.

Delimitación Temporal

- El presente proyecto de investigación tuvo una duración de seis meses, a partir del 15 de marzo, al 06 de septiembre del 2017.

1.4 Justificación

El desarrollo de este proyecto permitió aprovechar los conocimientos adquiridos durante la carrera, que facilito el estudio de la plataforma, al estar en una época donde la mayoría de las cosas se realizan mediante la web y, el análisis de esta plataforma es de gran utilidad porque permite la

creación de páginas web dinámicas y bien diseñadas sin la necesidad de conocer mucho sobre JavaScript y HTML.

El framework de Google Web Toolkit es desarrollado por google de última tecnología, que permite al usuario desarrollar aplicaciones basadas en java de código abierto y usando a su vez AJAX, lo cual facilita cargar una página web.

El presente trabajo es factible porque GWT se presenta como una solución simple y segura para desarrolladores que deseen aprovechar el potencial que tiene AJAX, debido a que no es un proceso simple y no existen muchas herramientas que faciliten el trabajo.

La plataforma GWT permite desarrollar aplicaciones de gran rendimiento porque los desarrolladores pueden crear y mantener rápidamente aplicaciones.

Las aplicaciones realizadas mediante esta plataforma son de alta disponibilidad, porque admite un conjunto indefinido de protocolos de transferencia de información y una forma sencilla de comunicación con el servidor web.

Los beneficiarios inmediatos del presente trabajo serán las personas con discapacidad visual.

1.5 Objetivos

1.5.1 Objetivo General

Realizar un estudio de la metodología GWT, para contribuir con los desarrolladores.

1.5.2 Objetivos Específicos

- ✓ Obtener los requerimientos necesarios para el desarrollo del sistema.
- ✓ Analizar las características, ventajas y desventajas de la plataforma Google Web Toolkit.
- ✓ Elaborar el prototipo para el estudio correcto de la plataforma así como también para determinar las ventajas y desventajas que se tiene al usar la plataforma de desarrollo GWT.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Antecedentes Investigativos

Rivera Arbeláez David en el año (2013), desarrolló el bastón “Ojo guía” para personas con discapacidad visual, se apoya en un software que permite emitir una onda que determina la distancia a la que se encuentran los objetos, fracasaron en su primer lanzamiento al mercado, presentaron fallas en la georreferenciación de ciudades, red de amigos y reportes sobre lugares poco accesibles, este trabajo reposa en la revista SENA

En su trabajo de tesis, (Murillo Vélez & Riera Idrovo , 2010), realizaron el análisis de la plataforma GWT mediante el desarrollo de una aplicación para la refinería de Esmeraldas, para el desarrollo de la investigación utiliza el método científico, analítico y descriptivo, después del análisis y desarrollo del proyecto, el autor recomienda realizar las conexiones para extraer la información de la base de datos, además de agregar los plugins necesarios para el correcto funcionamiento.

En la actualidad existe una amplia gama de aplicaciones y smartphones diseñadas para personas con discapacidad visual, las mismas que tiene un objetivo e común que es permitir a las personas con discapacidad visual mejorar su orientación espacial a través de la realidad aumentada.

En cuanto al desarrollo de aplicaciones para personas con discapacidad visual Google ha desarrollado una aplicación que permite manipular en móvil sin necesidad de mirar la pantalla, además dispone de un servicio de localización que combina con un sistema de GPS mediante la utilización del servicio de Google Maps. (Ju & Illescas, 2012)

2.2 Fundamentación Teórica

2.2.1 Plataforma GWT

2.2.2 Introducción

El entorno GWT fue creado a finales de 2006 por Google, y anunciado en la conferencia JavaOne de 2006, existen muchas versiones de GWT que han sido publicadas en un mismo año, esto se debe a que Google ha implementado nuevas funcionalidades en este entorno de desarrollo. [1]

Además, los navegadores disponibles en el mercado van evolucionando y es necesario actualizar el motor de GWT. La buena noticia es que el código de una versión anterior se puede volver a recompilar con la nueva versión sin necesidad de modificar el código anterior. [5]

En agosto de 2010 Google adquirió la empresa Instantiations, responsable del entorno de desarrollo rápido de Java "Eclipse", disponible para plataformas Windows, Linux y Mac. Google ha introducido recientemente las librerías necesarias en este entorno para desarrollar aplicaciones Web con GWT. [5]

GWT es un conjunto de módulos de código abierto, por lo que no requiere la adquisición de ninguna licencia. Además, Google permite que los usuarios contribuyan al mismo con mejoras. [5]

Existe un equipo de GWT, el mismo que ha creado un repositorio donde está publicado todas las aplicaciones que usan GWT, entre las aplicaciones que más utilizan todo el potencial de GWT está la de Lombardi Blueprint. [2]

En la actualidad existe una amplia gama de aplicaciones y smartphones diseñadas para personas con discapacidad visual, las mismas que tiene un objetivo e común

que es permitir a las personas con discapacidad visual mejorar su orientación espacial a través de la realidad aumentada.

En cuanto al desarrollo de aplicaciones para personas con discapacidad visual Google ha desarrollado una aplicación que permite manipular en móvil son necesidad de mirar la pantalla, además dispone de un servicio de localización que combina con un sistema de GPS mediante la utilización del servicio de Google Maps. [3]

Google Web Toolkit (GWT), es un framework desarrollado por google de última tecnología, que permite desarrollar aplicaciones basadas en java de código abierto, traduciendo y compilando el programa a JavaScript y HTML con cualquier navegador. [4]

GWT no es solo una interface de programación, proporciona un conjunto de herramientas que permiten desarrollar funcionalidades JavaScript de alto rendimiento en el navegador del cliente. [4]

El compilador de GWT soporta la mayoría de las características del lenguaje de programación, mientras que las librerías GWT runtime emulan una gran parte de las librerías de Java. [4]

El uso de GWT es de importancia, porque en la actualidad, por disposición del gobierno, todas las instituciones públicas deben orientarse a la utilización de sistemas operativos, programas, bases de datos y herramientas de código libre, por lo tanto el uso de esta plataforma facilitará el trabajo de los desarrolladores y permitirá cumplir con las disposiciones del gobierno.

2.2.3 Definición de GWT

GWT es un framework Open Source, desarrollado por Google, con el objetivo de facilitar el desarrollo de aplicaciones de Internet; permitiendo que el desarrollador

escriba todo el FrontEnd de un sistema web, en Java, olvidándose de programar JSPs, JavaScripts, HTML.

El manejo de varios lenguajes al mismo tiempo, lleva a código poco confiable y difícil de mantener e imposible de depurar.

2.2.4 Características de GWT

- Es compatible con la mayoría de navegadores.
- Es de código abierto.
- Permite el desarrollo de aplicaciones orientado a objetos.
- Posee componentes de interface de usuario dinámicos y reutilizables.
- Se puede encontrar un amplio conjunto de bibliotecas desarrolladas por Google que mejoran las funcionalidades de GWT.
- Crea aplicaciones y librerías de internacionalización de forma rápida y fácil.
- Integración con JUnit, mediante la cual se pueden probar las aplicaciones y depurarlas en un navegador mientras se construyen.
- Depuración en tiempo real, el código se traduce en JavaScript por medio de una Máquina Virtual de Java, permitiendo que en la fase de desarrollo se depure la aplicación con sistemas debugging y manipulación de excepciones incluidos en IDE's como Eclipse.
- Los errores comunes en JavaScript son controlados en tiempo de compilación, como es el caso de la incompatibilidad de los tipos de datos, simplificando el desarrollo de funciones de JavaScript complejas.

2.2.5 Arquitectura GWT



Gráfico 1. Arquitectura GWT
Elaborado por: El Investigador

GWT está formado por cuatro componentes principales:

- **Compilador GWT de Java a JavaScript**

El compilador Java a JavaScript es el componente más importante de GWT, es el encargado de transformar el código Java escrito en Eclipse y producir distintas versiones en JavaScript similares, que se pueden todos los navegadores soportados por GWT. [5]

Este compilador se encarga de traducir el código Java al lenguaje JavaScript, esta transformación se realiza al momento que se ejecute en modo web. [1]

El código JavaScript es generado en archivos para cada tipo de navegador e idioma.

Las optimizaciones aplicadas más importantes son:

- Eliminación de código que no es usado por la aplicación y no se incluye en el fichero JavaScript creado por GWT, cuando existe herencia el código de salida solo se genera para los métodos que realmente lo necesitan. [5]
- El cálculo automático de constantes permite que el código JavaScript se ejecute más rápido debido a que la concatenación de la cadena ya está hecha. [5]
- La eliminación de cadenas repetidas evita la creación de cadenas repetidas por el programador. [5]
- La inclusión de código de un método que sea corto y sencillo, GWT sustituye el código de este método en el sitio donde aparece la llamada original. [5]

Todas las optimizaciones de código permiten que el código JavaScript final será muy eficiente, la desventaja es que GWT no permite hacer compilaciones parciales.

GWT estudia el código fuente en su totalidad y realiza una compilación monolítica para maximizar el número de optimizaciones posibles.

Esta característica de GWT impide la reutilización de los módulos previamente compilados, se trata de una característica de diseño de desarrollo de google, donde se busca un mayor rendimiento en las aplicaciones creadas con GWT.

- **Navegador Web de GWT**

Es un navegador de modo hospedado en la máquina virtual de java, que permite correr la aplicación sin necesidad de pasar el código al lenguaje JavaScript, facilitando al desarrollador el manejo de excepciones y corrigiendo errores. [4]

- **Librerías de emulación JRE**

En una aplicación normal de Java se pueden utilizar todas las clases y métodos disponibles, por la forma de funcionar del compilador GWT, este requiere tener acceso al código fuente real de cualquier clase que se desee usar. [5]

JRE (Java Runtime Environment), es un conjunto de utilidades que permite la ejecución de programas Java, por lo tanto GWT proporciona una implementación especial y parcial denominada biblioteca de emulación JRE (JRE Emulation Library). [5]

GWT incluye la mayoría de las clases del paquete java.lang y un subconjunto de clases del paquete java.util, el resto de las librerías no son soportadas por GWT. [4]

- **java.lang:** Proporciona clases que son fundamentales en el diseño de aplicaciones en el lenguaje de programación Java, incluye excepciones, clases y métodos de utilidad general e interfaces. [5]

EXCEPCIONES	IndexOutOfBoundsException NegativeArraySizeException NullPointerException NumberFormatException RuntimeException StringIndexOutOfBoundsException Throwable UnsupportedOperationException ArithmeticException	ArrayIndexOutOfBoundsException ArrayStoreException AssertionError ClassCastException Error Exception IllegalArgumentException IllegalStateException
CLASES	Character Boolean Byte Double Float Class Object Short	String StringBuffer StringBuilder Integer Long Number
UTILIDADES	Maths	System
INTERFACES	Appendable CharSequence Cloneable	Comparable Iterable Runnable

Gráfico 2. Clases fundamentales
Elaborado por: El Investigador

- **java.util:** Proporciona varias colecciones de herramientas útiles en el desarrollo de aplicaciones Java. [5]

EXCEPCIONES	ConcurrentModificationException EmptyStackException	MissingResourceException NoSuchElementException TooManyListenersException
CLASES	AbstractCollection AbstractHashMap AbstractList AbstractMapEntry AbstractMap AbstractQueue EventObject HashMap HashSet IdentityHashMap LinkedHashMap LinkedHashSet AbstractSequentialList AbstractSet	ArrayList Arrays Collections Date EnumMap EnumSet LinkedList MapEntryImpl PriorityQueue Stack TreeMap TreeSet Vector
INTERFACES	Collection Comparator Enumeration EventListener Iterator ListIterator List	Map Queue RandomAccess Set SortedMap SortedSet

Gráfico 3. Herramientas en el desarrollo de aplicaciones Java
Elaborado por: El Investigador

- **Librerías de clases de GWT**

Es considerado el núcleo de las librerías de interfaz para crear aplicaciones a través de widgets. Las librerías de clases de interfaz de usuario de GWT son un conjunto de interfaces y clases personalizadas que permiten crear “widgets” para el navegador, como botones, cajas de texto, imágenes, entre otras. [4]

2.2.6 Funciones y Funcionamiento de GWT

2.2.5.1 Funciones

- **Comunicarse con el servidor mediante una llamada RCP.-** GWT admite un conjunto indefinido de protocolos de transferencia, como JSON y XML, pero el mecanismo de llamada a procedimiento remoto RCP de GWT permite el establecimiento de comunicaciones Java de forma sencilla y eficaz. [1]
- **Utiliza componentes de la interfaz de usuario en varios proyectos.-** se puede crear artilugios reutilizables, mediante la composición de otros artilugios y colocarlos en paneles, que permitirán reutilizar en otro proyecto, sólo se tiene que prepararlo para que se pueda usar en un archivo JAR. [1]
- **Utiliza otras bibliotecas JavaScript y Código JavaScript original.-** si la biblioteca de clases de GWT no satisface las necesidades, se puede mezclar manualmente JavaScript en el código fuente Java mediante la interfaz JSNI.

La ventaja de poder crear modelos de objetos JavaScript como tipos de Java sin tener que contar con una mayor cantidad de memoria ni sufrir una disminución de velocidad, posibilitando así un uso óptimo de las estructura de JSON. [1]

- **Localiza las aplicaciones de una forma eficaz.-** Puede crear fácilmente bibliotecas y aplicaciones internacionalizadas con las potentes técnicas de vinculación aplazada de GWT, a partir de la versión 1.5 GWT admite la bidireccionalidad. [1]

- **Elige las herramientas de desarrollo de forma productiva.-** GWT utiliza Java, por lo que se puede utilizar todas las herramientas favoritas de desarrollo con Java (Eclipse, IntelliJ, JProfiler, JUnit, NetBeans, etc) al crear las aplicaciones.

De esta forma, el desarrollador web, beneficiará el aumento de productividad derivado de la creación de código y de la reestructuración de Java automatizada. [1]

- **Extensión o contribución con el software libre Google Web Toolkit.-** todo el código de GWT está disponible bajo la licencia de Apache 2.0.

2.2.5.2 Funcionamiento

Durante el desarrollo de una aplicación, puede repetir rápidamente el mismo ciclo “editar, actualizar, ver” y aprovechar la ventaja añadida para depurar y recorrer una a una todas las líneas de código Java. Cuando ya está listo para la implementación, GWT compilará el código fuente Java en archivos JavaScript optimizados e independientes. GWT permite crear fácilmente una página web como una aplicación completa. [1]

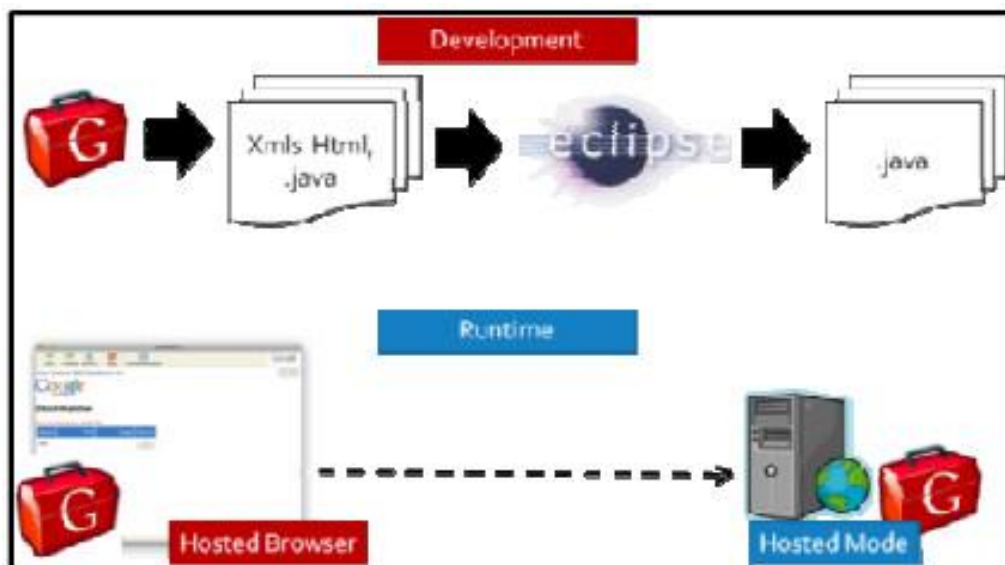


Gráfico 4. Funcionamiento de GWT.
Elaborado por: El Investigador

2.2.6 Ciclo de desarrollo

El ciclo de Desarrollo de GWT sería el siguiente:

- Usa el entorno de desarrollo integrado IDE que permita al desarrollador escribir y depurar una aplicación JAVA, usando las librerías de GWT necesarias.
- Usa el compilador de Java a JavaScript de GWT para transformar la aplicación en un conjunto de archivos JavaScript y HTML, que se pueda colgar en cualquier servidor y ejecutar desde un navegador web.
- Verifica que las aplicaciones trabajen sobre todos y cada uno de los navegadores que pueden ser utilizados por los clientes.

En GWT se pueden usar componentes de interfaz de usuario llamados widgets, para construir aplicaciones con interfaz de usuario atractivas. Al igual que en la mayoría de los lenguajes de programación, los componentes de la interfaz de usuario son agrupados en paneles que determinan la ubicación de los mismos. [6]

2.2.7 Modelo de programación en GWT

En el modelo de programación que propone GWT se elimina la programación en JavaScript, este modelo cambia la forma en que hasta ahora se han desarrollado las aplicaciones web y propone que toda la aplicación se programe solo en lenguaje Java. [5]

Este framework se encarga de compilar automáticamente la parte cliente de la aplicación a JavaScript, dando así lugar al uso de las siguientes herramientas de desarrollo: Eclipse, NetBeans, IntelliJ, entre otras. [4]

El compilador GWT compila un subconjunto del API disponible en Java, para las clases cliente, se usa los propios paquetes de GWT **com.google.gwt.user.client.*** [5]

GWT también ofrece dos modos de ejecución, el modo web y el hosted mode, el modo web compila el código con el compilador GWT y lo ejecuta en un navegador habitual, el hosted mode consiste en ejecutar la aplicación dentro de una máquina virtual Java. [5]

2.2.8 Estructura de un proyecto GWT

Un proyecto GWT, tiene una estructura más similar a un proyecto Java que a un proyecto J2EE. GWT necesita ciertos ficheros adicionales y el seguimiento de una cierta nomenclatura en los paquetes Java.

La siguiente tabla muestra la estructura que debería tener este proyecto.

Tabla 1. Estructura del Proyecto

src/
src/com/ree/ree/Principal.gwt.xml
src/com/ree/ree/client/
src/com/ree/ree/client/Principal.java
src/com/ree/ree/server/ (opcional)
src/com/ree/ree/public/
src/com/ree/ree/public/Principal.html
bin/
www/

La carpeta “src” contiene:

- Las fuentes de Java para el lado cliente que serán compilados a JavaScript, carpeta “src/com/ree/ree/client”
- Las fuentes de los servicios RCP del servidor se compilaran a .class, carpeta “src/com/ree/ree/server”
- Los recursos estáticos que necesite la aplicación como por ejemplo imágenes, carpeta “src/com/ree/public/”
- El descriptor de módulo (src/com/ree/reePrincipal.gwt.xml). GWT basa sus aplicaciones en módulos, y cada módulo debe tener su fichero de configuración, en el fichero se declaran los servicios

RPC que van a ser invocados desde el cliente así como los puntos de entrada a la aplicación.

- La página contenedora (src/com/ree/ree/public/Principal.html), aunque con GWT no es necesario escribir código HTML, existe una pequeña excepción a esta regla. Debe existir al menos una página HTML que contenga la aplicación. Por ejemplo se puede crear el diseño HTML en el que se muestren banners, esta será la página que se invocará desde un navegador para proyectar la aplicación GWT.

La carpeta “bin” contiene el resultado de compilar las clases Java mediante el compilador java. Hay que notar que aquí se compilarán todas las clases Java, incluso las del lado cliente. Cuando se ejecute la aplicación en el “hosted Mode” se ejecutarán estas clases y no el resultado de la compilación a JavaScript.

La carpeta “www” contiene la aplicación para desplegar en un servidor web. Es decir, aquí se encuentran las clases cliente, compiladas a JavaScript mediante el compilador GWT.

2.2.9 Ventajas y Desventajas de GWT

2.2.9.1 Ventajas de GWT

Dentro de las ventajas que se obtiene al trabajar con GWT son:

- Usa el entorno de desarrollo favorito para escribir y depurar una aplicación java usando las librerías de GWT.
- Emplea el compilador de Java a JavaScript para transformar la aplicación en archivos JavaScript y HTML que puede ejecutarse desde un navegador web.
- Las aplicaciones desarrolladas mediante GWT ejecutan código Java del lado del servidor utilizando RPC, el mismo que se utiliza para la comunicación entre el cliente y el servidor.

- Existen desarrolladores que más conocen Java debido a que Java tiene abundancia de herramientas para el desarrollador.

2.2.9.2 Desventajas de GWT

Entre las desventajas que se encuentran al trabajar con la plataforma GWT son las siguientes:

- Solo desarrolladores JAVA
- Consumo de memoria en el navegador
- El tiempo de compilación desde el navegador propio de GWT a cualquier navegador que se desee presentar la aplicación.

2.2.10 Base de Datos

Una Base de Datos (BD) es una colección o depósito de datos, almacenados en soporte secundario y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos y su definición (estructura de la BD), única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual debe permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos. [7]

Una base de datos es un lugar de almacenamiento de información, y la ventaja de usarla es que controla la integridad y redundancia de los datos. También provee medidas de seguridad a los datos como lo es la recuperación evitando así pérdidas de la información.

2.2.11 Sistema de Gestión de Bases de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto coordinado de programas, procedimientos, lenguajes, herramientas, etc., que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o administradores de una BD, los medios necesarios para describir y manipular los

datos integrados en la BD, manteniendo su integridad, confidencialidad y disponibilidad. [7]

Entre los diferentes SGBD, se puede encontrar los siguientes:

- MySql
- PostgreSql
- Oracle
- Access
- Microsoft SQL Server

Los sistemas gestores de bases de datos son herramientas que proporcionan un fácil acceso y manipulación de las bases de datos, por medio de componentes gráficos. Estas permiten una fácil administración y mantenimiento, para personas con alto y bajo conocimiento en bases de datos. [7]

2.2.12 Metodología Ágil

Las metodologías ágiles han ganado bastante popularidad desde hace unos años, son una muy buena solución para proyectos a corto plazo, especialmente para proyectos en donde los requerimientos están cambiando constantemente.

Tener una buena arquitectura implica que el sistema tiene atributos de calidad que dará un valor muy importante en el software.

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas

metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero falta una mayor difusión y reconocimiento. [8]

2.2.13 Metodología de desarrollo XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes donde existe un alto riesgo técnico. [9]

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente especifica el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué desarrollar, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador desarrolla ese valor de negocio.
5. Se regresa al paso 1. [9]

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.

De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. [9]

Las fases de la metodología XP son las siguientes:

1. Planificación

- Historias de Usuario
- Iteraciones
- Velocidad del Proyecto
- Programación en pareja
- Reuniones diarias

2. Diseño

- Diseños simples
- Glosario de términos
- Riesgos
- Funcionalidad extra
- Tarjetas C.R.C

3. Codificación

4. Pruebas

- Test de Aceptación

2.2.14 Análisis Comparativo de IDE's

Los IDE's (Integrated Development Environment), son un conjunto de herramientas para el programador, que suelen incluir, un buen editor de código, administrador de proyectos y archivos, enlace a compiladores y debuggers e integración con sistemas controladores de versiones o repositorios, permitiendo versatilidad en la depuración de programas.[17]

GWT no tiene IDE específico, por lo tanto se realizó un análisis comparativo de los IDE's, que usa GWT para el desarrollo de aplicaciones web, se puede utilizar cualquier entorno de desarrollo integrado que use el lenguaje de programación JAVA. [17]

En la actualidad existen mucho IDE's para desarrollar aplicaciones web usando el lenguaje de programación JAVA, se tomó como referencia una serie de páginas web y criterios de diversos autores, para la elección del IDE más adecuado.

(Maldonado Daniel , 2008), analista en sistemas de la Universidad Católica de Santiago del Estero, habla de los IDE's más populares para programar en Java, a continuación se detallan: [17]

- BlueJ
- Eclipse
- IntelliJ IDEA
- Jbuilder
- JCreator
- JDeveloper
- NetBeans
- Sun Java Studio Enterprise

En base a las conclusiones de un debate, (Snyflex, 2010) nos presenta los IDE's más adecuados para el desarrollo de aplicaciones web en Java son: [17]

- Eclipse
- NetBeans

(Teknoda, 2012), nos da a conocer una serie de IDE's populares de código abierto, que se listan a continuación: [17]

- Sun ONE Studio 4
- JEdit
- JCreator
- Eclipse
- Netbeans

Los IDE's que nos recomienda utilizar son, JEdit, Eclipse y Netbeans, por no tener costo y ser más conocidas por los programadores de Java.

Según (Burke , 2009), los IDE's más adecuados que nos recomienda usar para GWT, por la facilidad de instalación, uso, soporte técnico, son: [17]

- Eclipse
- NetBeans

- IntelliJIDEA

Los IDE's Eclipse, Netbeans, IntelliJIDEA, gozan de mayor popularidad y los desarrolladores de Java los recomiendan porque existe información para su respectiva utilización.

Los IDE's Eclipse y Netbeans son open source, por lo que no tiene costo, además tienen bastante soporte técnico como documentación, que es uno de los puntos más importantes que se debe considerar al momento de elegir un IDE para trabajar con el framework de Google GWT que también es de código abierto.[17]

Por otro lado el IDE IntelliJ IDEA, a pesar de ser licenciado, es decir pagado, cuenta con soporte técnico necesario para poder realizar cualquier aplicación web, lo que es de mucha importancia para los desarrolladores.[17]

Tabla 2. Análisis de comparación de IDE's

Parámetro	Variable	Eclipse	Netbeans	IntelliJIDEA
Acceso a la Base de Datos	Utilización de RPC.	3	3	2
	Número de Plug-in para el acceso a la base.	2	3	3
	Grado de dificultad para el acceso a la base de datos.	3	3	3
	Número de pasos que se necesitan para el acceso	2	1	1
Compilación	Número de líneas de código.	2	3	2
	Tiempo mínimo de depuración.	3	2	2
	Capacidad para detectar errores.	3	2	2
	Capacidad para depurar errores	3	3	2
Interfaz gráfica de	Diseñador visual para GUI	1	1	1
	Generación de métodos	3	3	1

usuario (GUI)	set y get			
	Menor coste	3	3	1
	Reutilización de componentes gráficos.	3	2	1
Seguridad	Encapsulamiento del código Java	3	3	3
	Encapsulamiento de la página HTML	3	3	3
	Número de protocolos	3	3	3
	Consistencia de las claves para el acceso al sistema	3	3	3
SopORTE técnico de GWT	Ayuda en línea del sitio oficial de GWT	3	2	1
	Páginas de ayuda de GWT	3	2	1
	Licencias	3	3	1
TOTAL:		52	48	36

Fuente: <https://cioperu.pe/articulo/18327/comparacion-de-4-ides-de-java/>

- Los IDE's Eclipse y Netbeans han demostrado ser los más idóneos en lo que se refiere al acceso de la base de datos usando la plataforma GWT.
- En cada compilación de una aplicación web usando la plataforma GWT, Eclipse resulta más conveniente por tener menos tiempo de respuesta, dando al usuario una experiencia más cómoda y amigable para trabajar.
- Eclipse permite detectar y depurar los errores de una manera fácil, porque tiene la opción de depuración que analiza paso a paso la aplicación.

Después del análisis comparativo y según los resultados obtenidos por desarrolladores, teniendo en cuenta cada uno de los parámetros con sus respectivas variables, concluyen que para usar la plataforma GWT con todas las ventajas y desventajas el IDE más adecuado es Eclipse. [17]

CAPÍTULO 3

METODOLOGÍA

3.1 Modalidad Básica de Investigación

Para la realización de la presente investigación se ha considerado utilizar las siguientes modalidades:

Modalidad documental – bibliográfica

El trabajo de investigación es bibliográfico, debido a que se basa en artículos, libros, y revistas confiables, que permitieron revisar, analizar, sintetizar, ampliar, profundizar y comparar diferentes puntos de vista de varios autores, criterios y temas relacionados con las variables.

Modalidad de Campo

El presente trabajo de investigación es de campo porque, se realizó en el club “Fénix” de la provincia de Tungurahua, donde mediante encuestas se trabajó con los entrenadores y familiares de los deportistas con discapacidad visual.

Se utiliza la metodología de programación XP debido a que es un método ágil y permite al sistema adaptarse a cambios no previstos durante la etapa del diseño.

3.2 Población y Muestra

La población para esta investigación está compuesta por 25 personas con discapacidad visual y un grupo de 20 personas que está conformado por entrenadores y familiares de las personas con discapacidad visual.

La población es finita porque conocemos con exactitud la cantidad de personas que posee la misma.

3.3 Recolección de Información

Para la recolección de información se hace uso de encuestas, las mismas que darán a conocer la factibilidad y usabilidad del proyecto.

3.4 Procesamiento y Análisis de Datos

Se analiza y procesa la información obtenida en cuadros resumen, gráficos, cuadros comparativos.

3.5 Desarrollo del Proyecto

Para el cumplimiento del desarrollo de este proyecto de investigación, se procede en forma secuencial los siguientes métodos, procesos y/o actividades en base a la metodología de desarrollo ágil XP.

Planificación del proyecto

- Recolección de la información
- Historias de Usuario

Diseño

- Base de Datos

Codificación

- Desarrollo de la aplicación

Pruebas

- Test de Aceptación

CAPITULO 4

DESARROLLO DE LA PROPUESTA

4.1 Planificación del proyecto

- Recolección de la información

La recolección de la información necesaria para el desarrollo de este proyecto se realizó mediante una entrevista y una encuesta, para conocer las necesidades que se presentan.

La entrevista fue dirigida a las personas con discapacidad visual, quienes son los beneficiarios directos, la encuesta se realizó a los entrenadores y familiares, que comparten la mayor parte del tiempo con las personas que tiene discapacidad visual.

En el Anexo N° 1, 2 y 3 se encuentran los instrumentos utilizados para la recolección y análisis de la información.

- Historias de Usuario

Ingreso al Sistema

Tabla 3. Formulario de Ingreso al Sistema

Formulario de ingreso al Sistema
Descripción: YO como Usuario del Sistema DESEO controlar la autenticación de usuarios y contraseñas PARA acceder a las distintas funcionalidades del sistema
Criterio de aceptación: Dado: Que los campos usuario y contraseña se encuentren llenos Cuando: el usuario accione el botón 'ingresar' Entonces: permitir acceder al sistema si la autenticación se realizó correctamente, caso contrario mostrar un mensaje 'usuario o contraseña incorrectos'.

Elaborado por: El Investigador

Buscar ubicación

Tabla 4. Formulario de Búsqueda

Formulario Buscar
Descripción: YO como Usuario del Sistema DESEO encontrar la ubicación de una persona en el mapa PARA saber los lugares en los que se estaba dicha persona
Criterio de aceptación: Dado: Que los campos persona, fecha fin, hora inicio, hora fin estén llenos Cuando: el usuario accione el botón 'buscar' Entonces: permitir visualizar la ubicación de la persona mediante marcadores dentro del mapa.

Elaborado por: El Investigador

4.2 Diseño

- Base de Datos

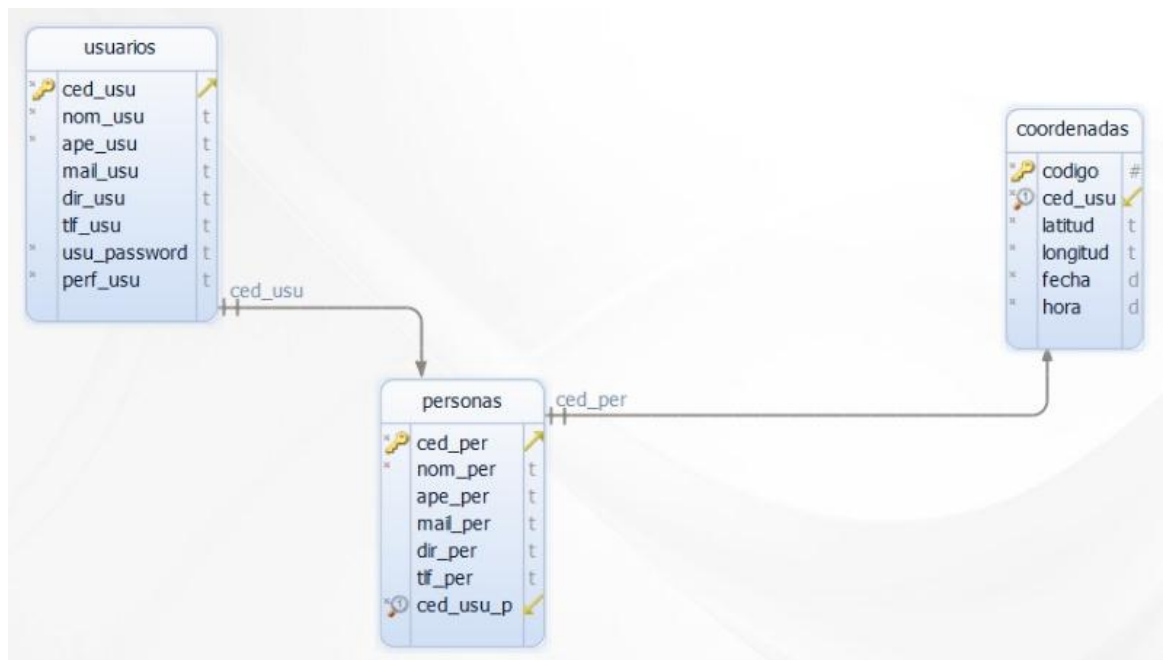


Gráfico 5. Diseño de la Base de Datos

Elaborado por: El Investigador

4.3 Codificación

- Desarrollo de la aplicación

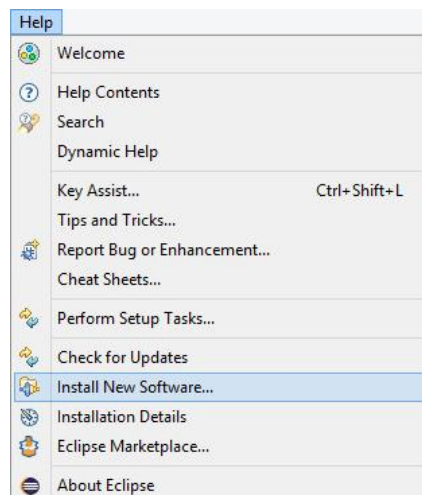
Software requerido

Lo primero que se debe instalar es:

- Eclipse Java EE IDE for Web Developers, version: Mars.2 Release (4.5.2)
- gwt-2.5.1
- gxt-2.2.5
- jdk 1.7
- Firefox 24.0 version para desarrolladores.

Instalación del plugin GWT para Eclipse

1. Primero descargar gwt-2.5.1 y descomprimir el zip para instalarlo.
2. En el menú seleccionar **Help** y hacer clic en **Install New Software**.



*Gráfico 6. Instalación de repositorios.
Elaborado por: El Investigador*

3. Para añadir el repositorio del plugin GWT, hacer clic en **Add...**

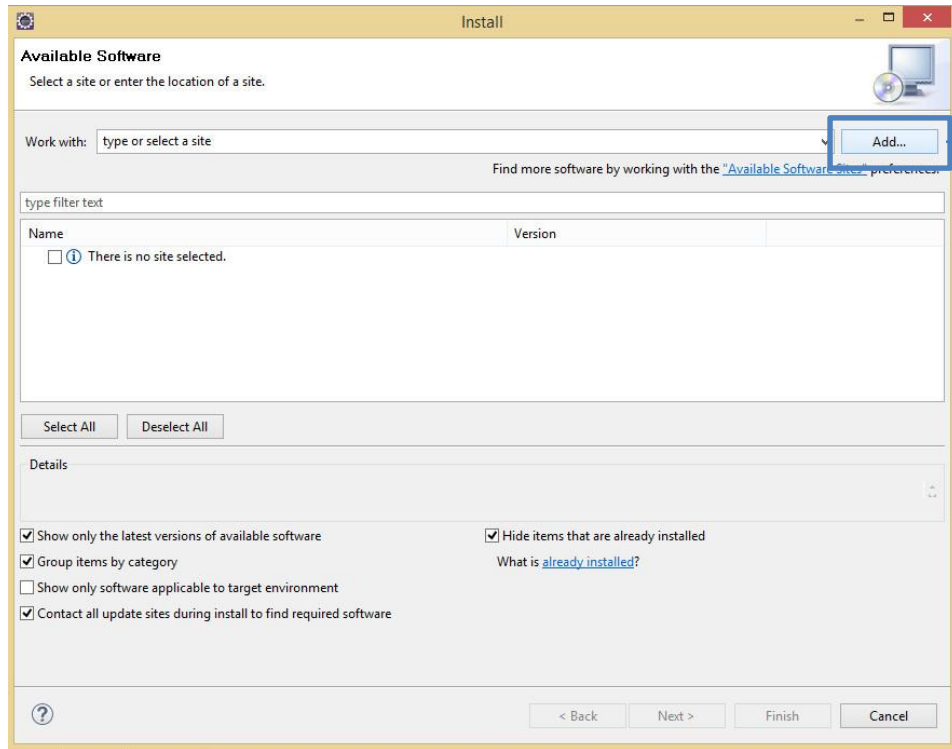


Gráfico 7. Instalación de repositorios.
Elaborado por: El Investigador

4. Aparecerá la siguiente ventana que permite añadir el repositorio, y llenar los datos **Name**, **Location** y hacer clic en **OK**.

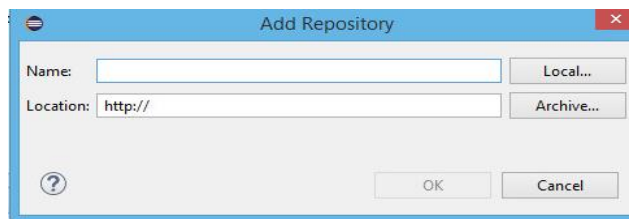


Gráfico 8. Add Repository.
Elaborado por: El Investigador

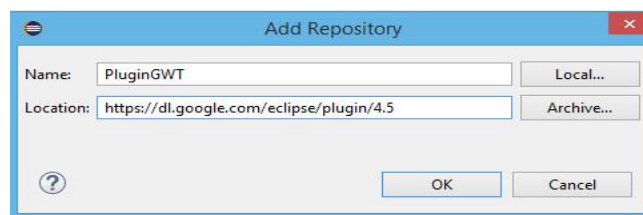


Gráfico 9. Opciones del repositorio.
Elaborado por: El Investigador

5. A continuación Eclipse muestra las opciones a instalar, solo se debe seleccionar las opciones del plugin de google y dar clic en **Next**, aceptar la licencia y dar clic en **Finish**.

Esperar a que eclipse termine de instalar el plugin, luego aparece un mensaje para reiniciar el IDE y dar clic en **Restart now**.

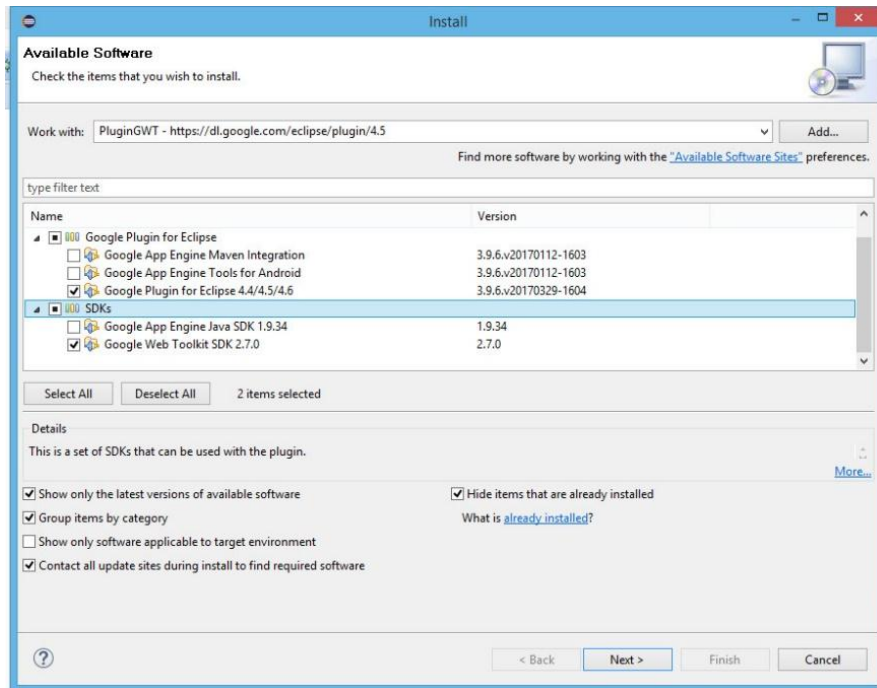


Gráfico 10. Opciones de instalación de plugin de GWT.
Elaborado por: El Investigador

6. Después que se reinicie Eclipse, aparece un icono con el logo de Google, que permite crear nuevos proyectos GWT.

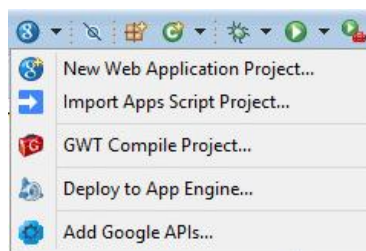



Gráfico 11. Gráfico 10. Opciones de GWT.
Elaborado por: El Investigador

Creación de un nuevo proyecto

7. Para crear un nuevo proyecto, dar clic en el logo de Google , y seleccionar la opción **New Web Application Project...**

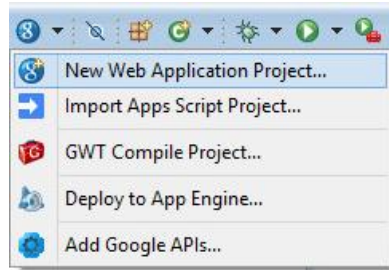
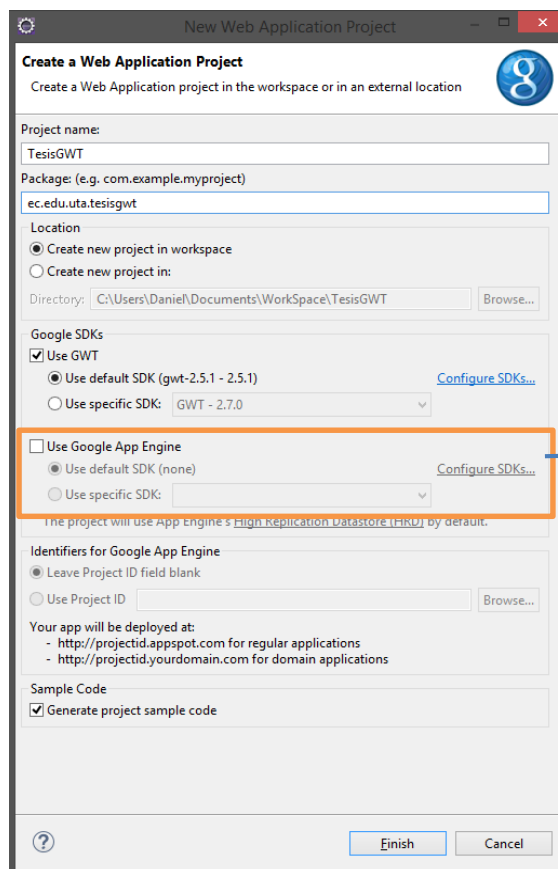


Gráfico 12. Gráfico 11. Creación de un proyecto de GWT.
Elaborado por: El Investigador

8. Es necesario especificar el nombre del proyecto **“Project name”** y el paquete **“Package”** del proyecto.

Quitar el check de la casilla Google App Engine porque se va a utilizar el compilador propio de eclipse y no uno externo.

Para ello utilizar la versión 2.5.1 de GWT por su estabilidad y compatibilidad con Java 7.



Desactivar la opción:
Use Google App Engine

Gráfico 13. Datos para la creación de la aplicación.
Elaborado por: El Investigador

Estructura del proyecto GWT

9. Al crear el proyecto la estructura que se presenta es la siguiente:

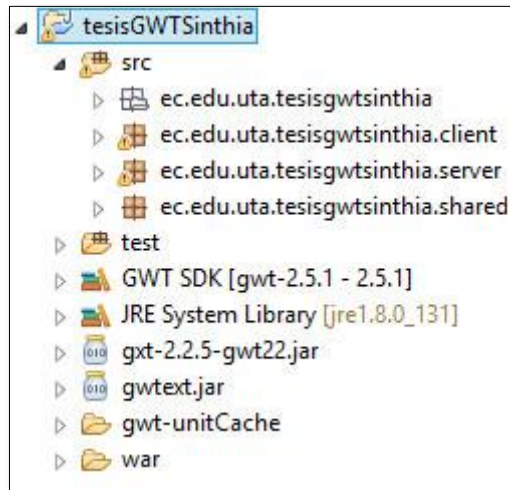


Gráfico 14. Estructura del proyecto
Elaborado por: El Investigador

La estructura del proyecto es la siguiente:

tesisGWTSinthia: Carpeta principal del proyecto, contiene todas las clases.

src: esta carpeta contiene las clases del servidor, cliente y clases compartidas.

ec.edu.uta.tesisgwtsinthia.client: Esta clase contiene la clase Inicio, TabItemMapa y la interfaz asincrónica.

ec.edu.uta.tesisgwtsinthia.server: Se encuentra la clase GreetingServiceImpl y RPCApplicationImpl.

ec.edu.uta.tesisgwtsinthia.shared.modelo: Dentro de esta carpeta está la clase de la base de datos, donde se encuentran declaradas todas las variables a utilizar.

GWT SDK [gwt-2.5.1 – 2.5.1]: Se encuentran los jar de gwt que van a ser utilizados.

JRE System Library [jre 1.8.0_131]: Están todas las librerías JRE.

war: En esta carpeta se encuentran los OpenLayers y extensiones de GWT.

10. Para ejecutar un proyecto de GWT, hacer clic derecho en la carpeta del proyecto, seleccionar **Run As y Web Application**.

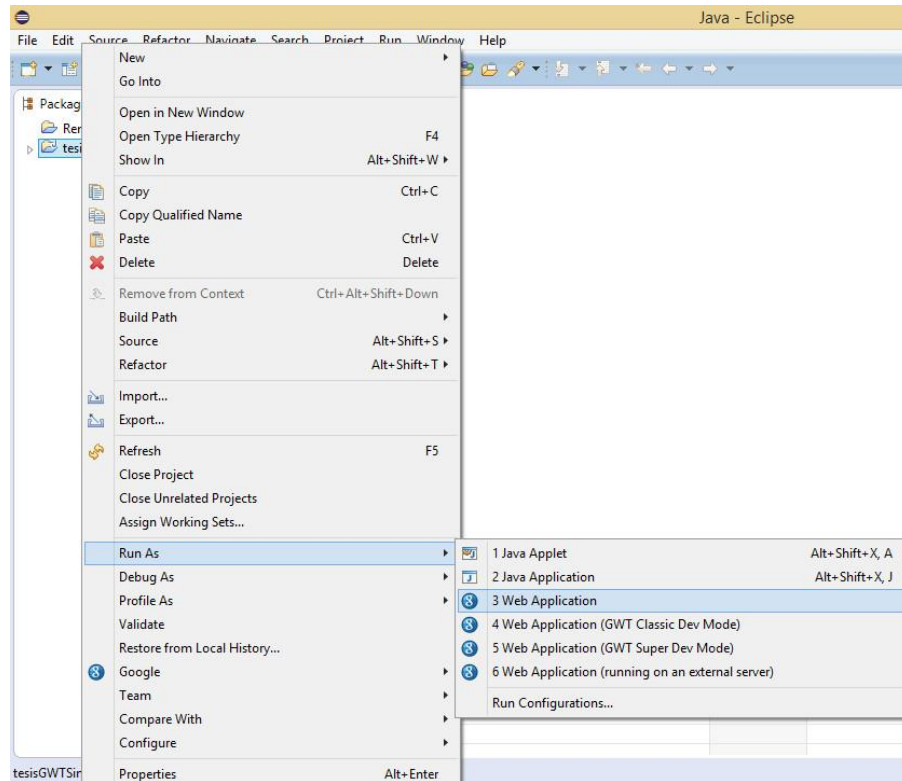


Gráfico 15. Compilación de un proyecto GWT.
Elaborado por: El Investigador

Para ejecutar proyectos GWT existen varias formas:

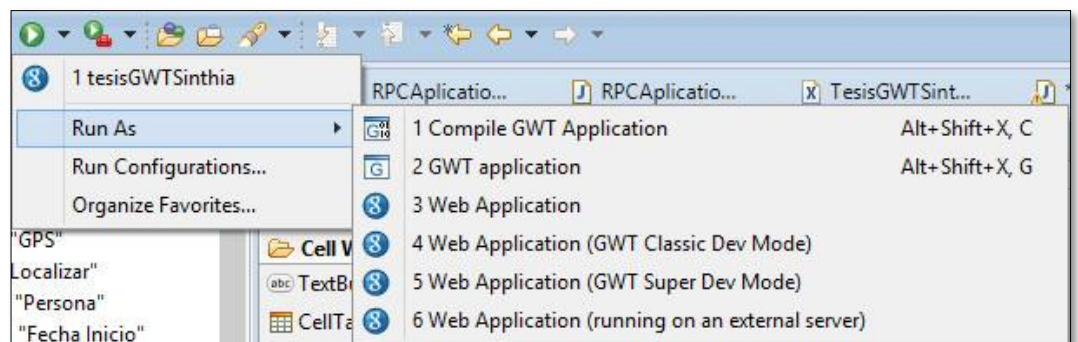


Gráfico 16. Formas de compilación de un proyecto GWT.
Elaborado por: El Investigador

Compile GWT Application: Compila la aplicación GWT.

GWT application: Aplicación GWT.

Web Application: Aplicación Web.

Web Application (GWT Classic Dev Mode): Aplicación web, modo clásico.

Web Application (GWT Super Dev Mode): Aplicación web, modo super.

Web Application (running on an external server): Aplicación web, ejecución en un servidor externo.

Firefox Versión 24.0: Para ejecutar los proyectos es necesario contar con la versión 24.0 de Firefox, porque es una edición de desarrolladores, la misma que ofrece una serie de características útiles para la construcción de proyectos web donde se incluyen herramientas para desarrollar aplicaciones en todos los navegadores, un depurador de JavaScript y editores para Web Audio y CSS.

11. Al ejecutar la aplicación web por primera vez, pedirá instalar el plugin de GWT en Firefox para poder compilar los proyectos.

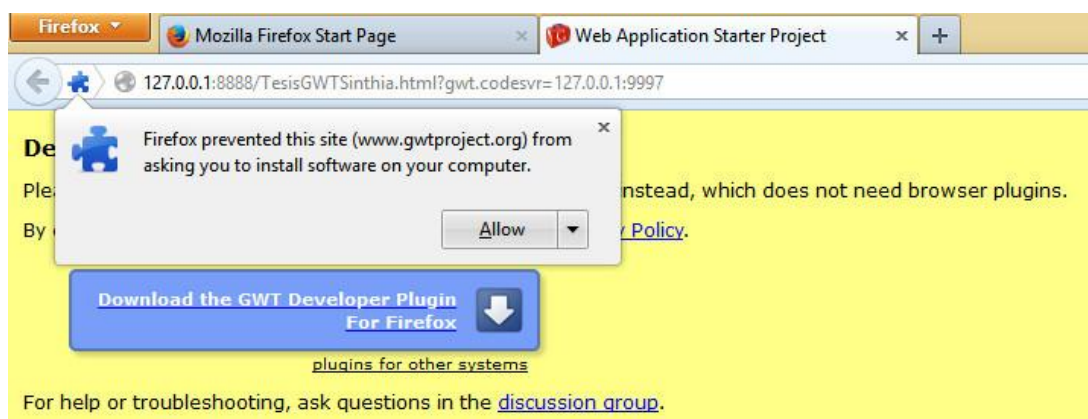


Gráfico 17. Instalación del plugin GWT en Firefox.
Elaborado por: El Investigador

12. Si la instalación esta correcta, el primer vistazo será el siguiente:

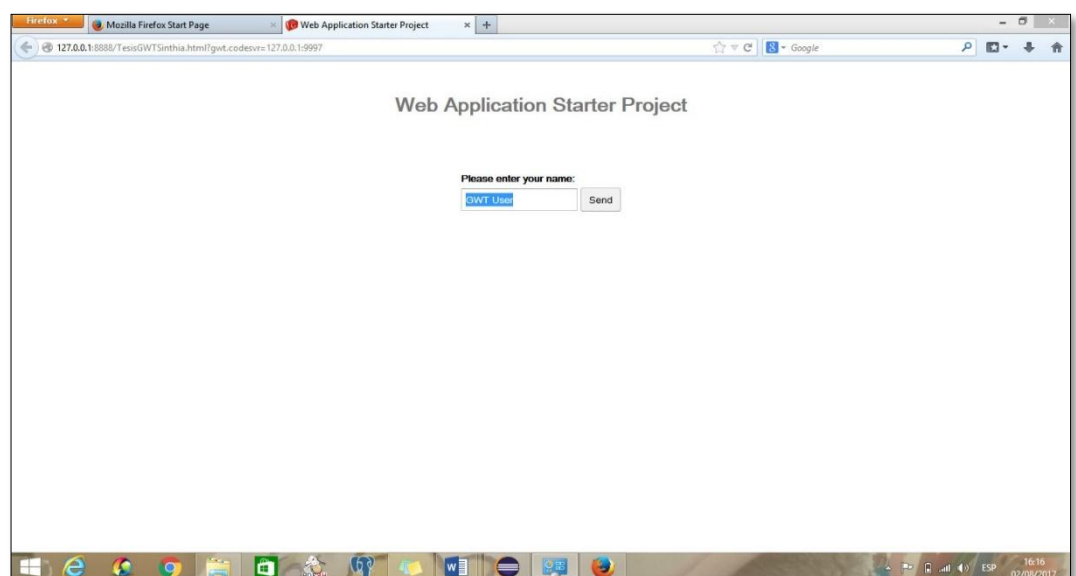


Gráfico 18. Vistazo de una aplicación web en el navegador.
Elaborado por: El Investigador

Anexo de componentes GXT en el proyecto

Para crear aplicaciones en GWT, es necesario tener anexados los componentes GXT en el proyecto, estos componentes poseen propiedades especiales que son propias del desarrollo web con GWT.

Se deben tener descargados y descomprimidos los componentes GXT, evidentemente deben ser de la misma versión del GWT.

13. Hacer clic derecho sobre el paquete Cliente, para visualizar las opciones que contiene. Seleccionar **Google Web Toolkit** y escoger la opción **Configure for using EXT GWT (GXT)**, luego aparecerá un cuadro de dialogo donde se especifica la carpeta GXT que contiene componentes especiales de GWT.

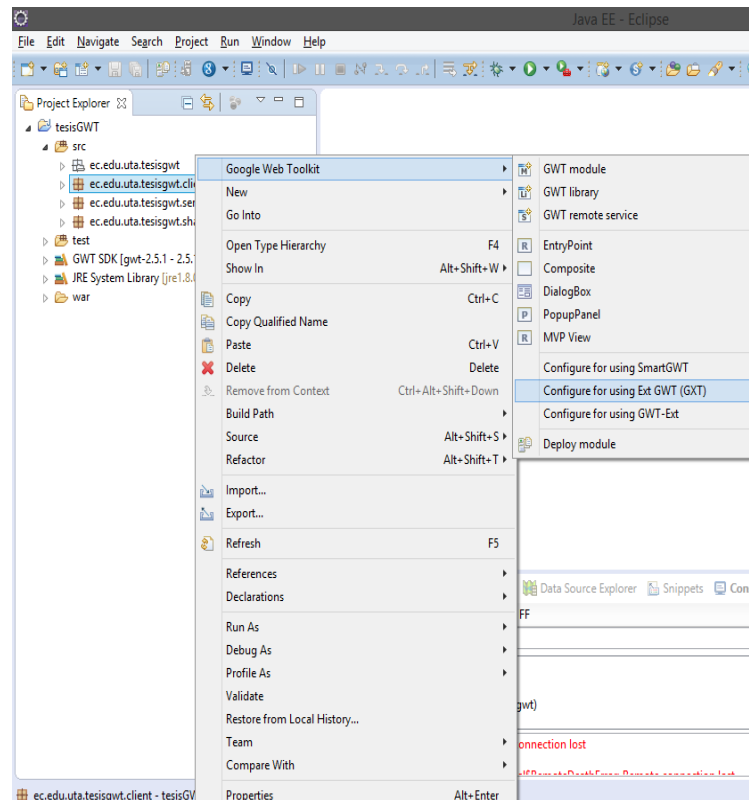
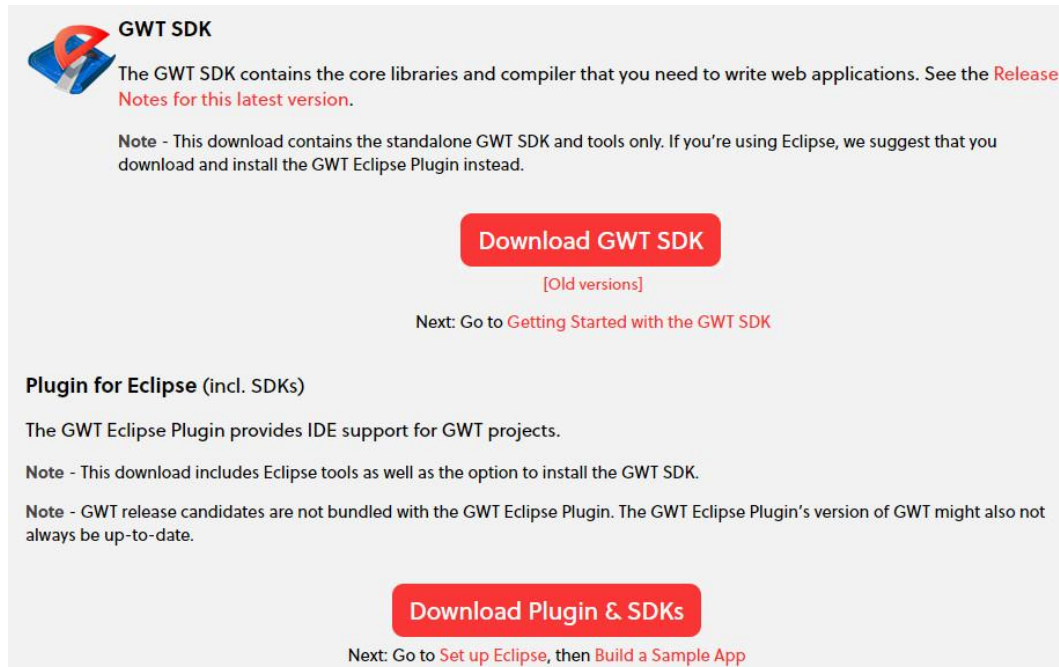


Gráfico 19. Configuración de GXT
Elaborado por: El Investigador

Para continuar con el anexo de los componentes de GXT del proyecto se debe descargar la carpeta **gwt-2.2.5** de la página oficial de GWT que se encuentra en el siguiente link: <http://www.gwtproject.org/download.html>



GWT SDK

The GWT SDK contains the core libraries and compiler that you need to write web applications. See the [Release Notes for this latest version](#).

Note - This download contains the standalone GWT SDK and tools only. If you're using Eclipse, we suggest that you download and install the GWT Eclipse Plugin instead.

Download GWT SDK

[\[Old versions\]](#)

Next: Go to [Getting Started with the GWT SDK](#)

Plugin for Eclipse (incl. SDKs)

The GWT Eclipse Plugin provides IDE support for GWT projects.

Note - This download includes Eclipse tools as well as the option to install the GWT SDK.

Note - GWT release candidates are not bundled with the GWT Eclipse Plugin. The GWT Eclipse Plugin's version of GWT might also not always be up-to-date.

Download Plugin & SDKs

Next: Go to [Set up Eclipse](#), then [Build a Sample App](#)

Gráfico 20. Gráfico 19. Página de descarga GWT
Elaborado por: El Investigador

14. Seleccionar la carpeta de **gwt-2.2.5** de la ubicación donde esta guarda, para continuar con la configuración.



Gráfico 21. Localización de la carpeta gxt-2.2.5.
Elaborado por: El Investigador

Project Facets.- Las facetas definen características y requisitos para proyectos Java EE y se utilizan como parte de la configuración de tiempo de ejecución.

Cuando se agrega una faceta a un proyecto, está configurado para realizar una determinada tarea, cumplir con ciertos requisitos o tener ciertas características.

15. Al hacer clic en la carpeta del proyecto, seleccionar **Properties**, donde se encuentran las configuraciones del proyecto, y en la pestaña **Project Facets** se puede verificar que el proyecto esté haciendo referencia a la versión 7 de java.

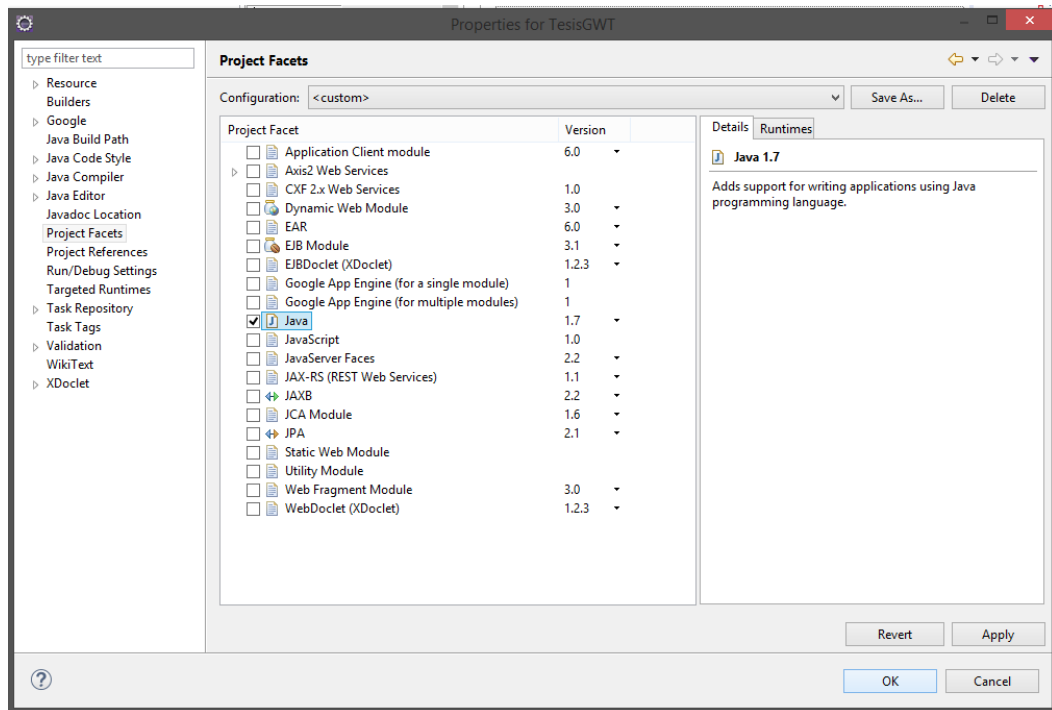


Gráfico 22. Gráfico 21. Verificación de la versión de Java
Elaborado por: El Investigador

Cuando se trabaja con GWT es necesario tener un servicio remoto que, permita tener conexión entre las capas dentro del proyecto.

16. En la opción **Google Web Toolkit**, seleccionamos **GWT remote service**, allí se podrá crear el servicio remoto.

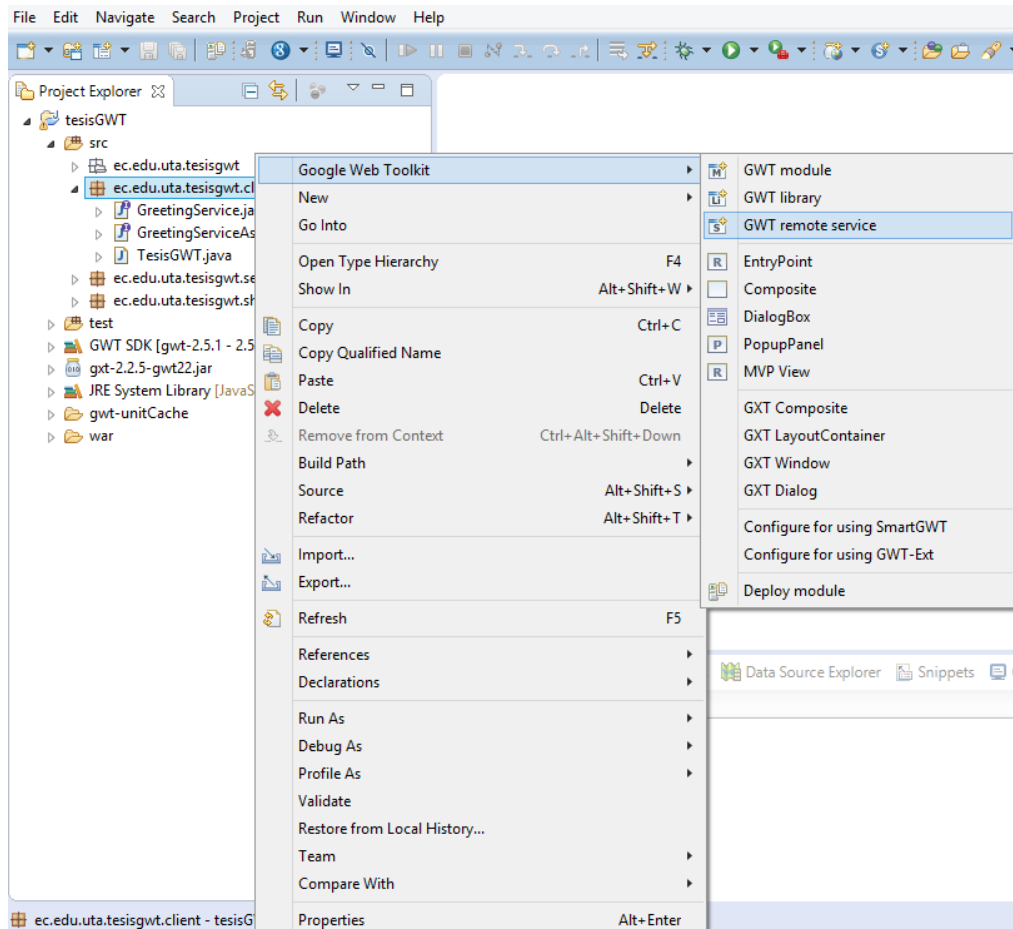


Gráfico 23. Creación del Servicio Remoto
Elaborado por: El Investigador

17. En la siguiente ventana se especifica el nombre del servicio remoto que se va a crear “**Service name**” y dar clic en **Finish**.

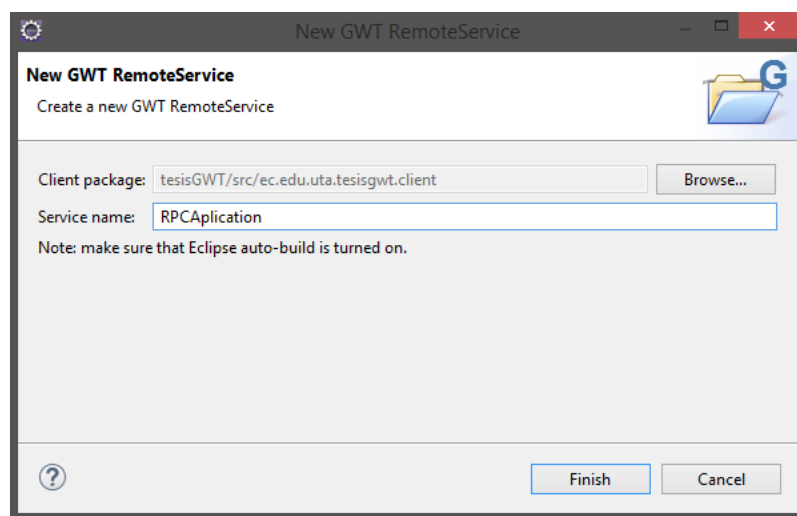


Gráfico 24. Especificación del nombre del servicio remoto
Elaborado por: El Investigador

Clase Sincrónica: es el intercambio de información por Internet en tiempo real.

Clase Asincrónica: El modelo de programación asincrónica, también denominado patrón AsyncResult.

Cuando se termina de crear la clase remota, se crea con errores, esto se debe a que no está creada la clase asincrónica del remote service.

La **clase asincrónica** funciona como puente o conexión entre la capa del servidor y la capa del cliente.

18. Para solucionar este error, se hace clic sobre el primer icono de error a la izquierda y en las opciones de solución, seleccionar la opción **Create interface "RPCApplicationAsync"**, es necesario seleccionar la opción de interfaz porque mediante las propiedades de la interfaz se mantiene la comunicación.

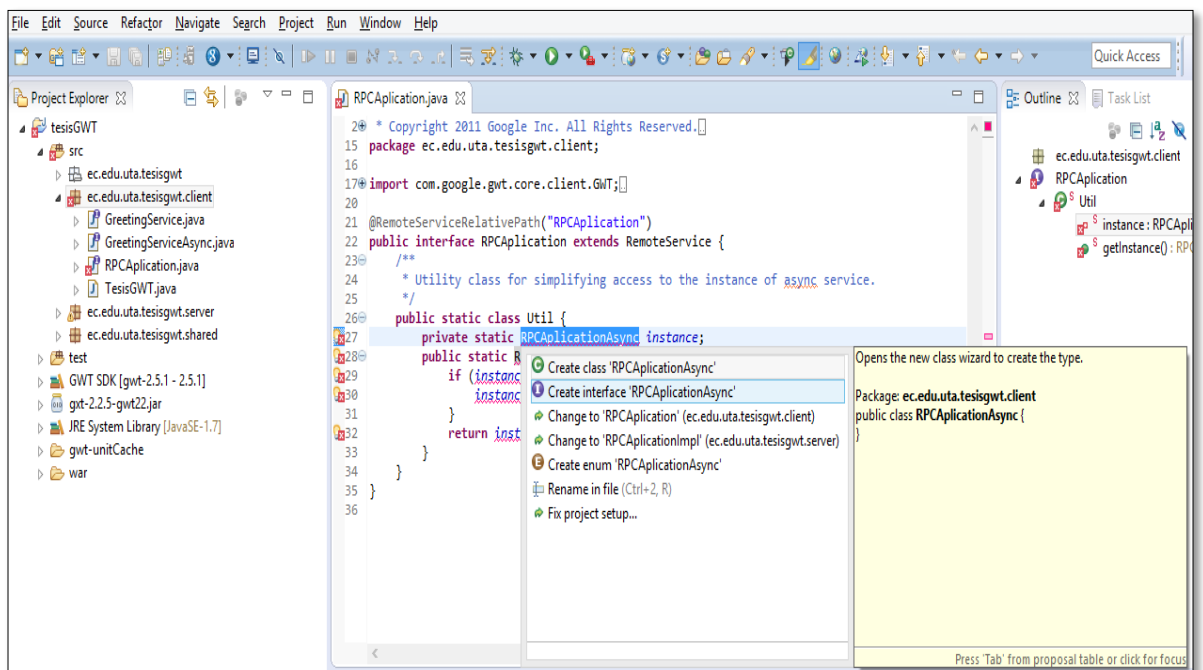


Gráfico 25. Creación de la interfaz de la clase asincrónica
Elaborado por: El Investigador

Creación de una clase

19. Para crear una clase nueva en el proyecto, dar clic derecho en la carpeta donde se va a agregar la clase, seleccionar **new**, y a continuación **Class**, como se visualiza en la imagen.

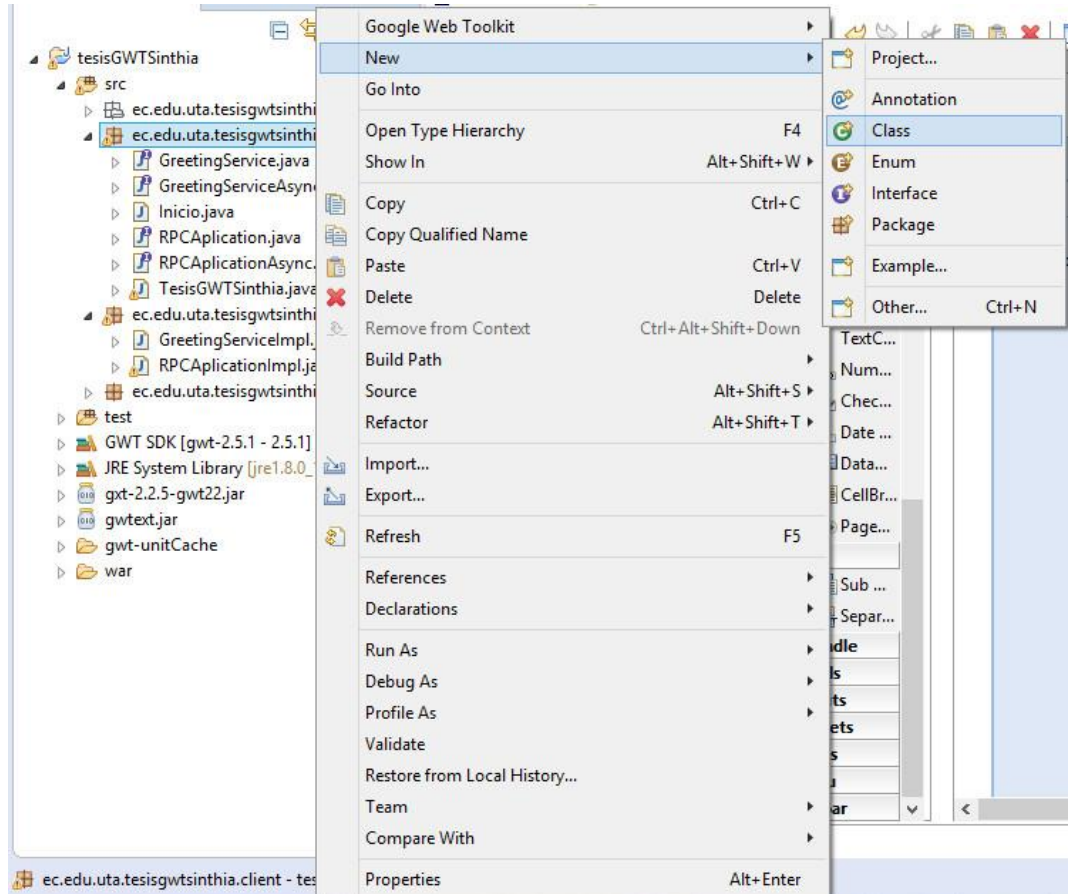


Gráfico 26. Creación de una clase.
Elaborado por: El Investigador

20. A continuación se visualizará la siguiente ventana, en la que se debe ingresar el nombre de la clase **Name** y **Finish**.

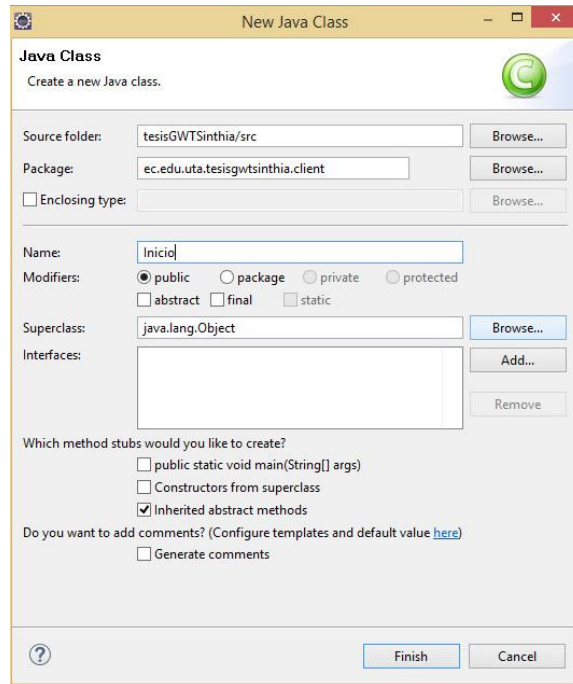


Gráfico 27. Creación de una nueva clase de java.
Elaborado por: El Investigador

Configuración de una clase

- Para que la clase muestre una ventana, se da clic en el botón **Browse** en la parte de propiedades. Permite acceder a los componentes de GXT, en el que se busca **Window** y se visualizan las posibles extensiones con las que se puede trabajar.

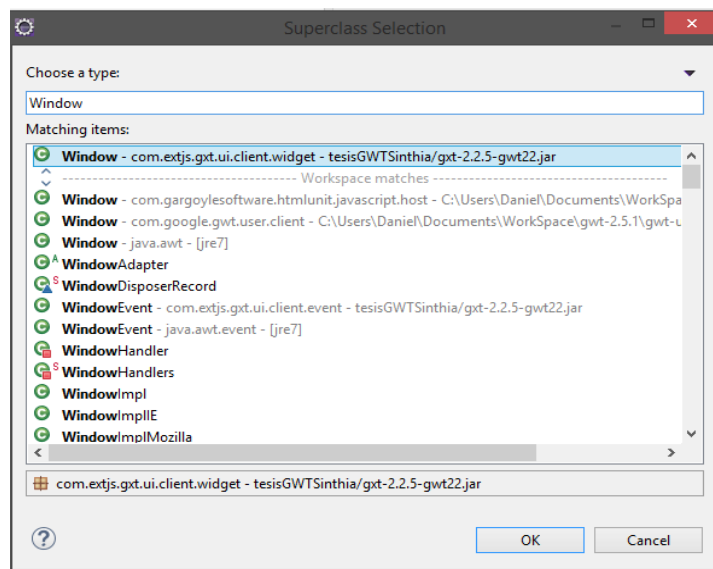


Gráfico 28. Componentes GXT: Window
Elaborado por: El Investigador

22. Para editar la el diseño del proyecto, hacer clic derecho en la clase principal, seleccionar **“Open With”** y seleccionar **“WindowBuilder Editor”**

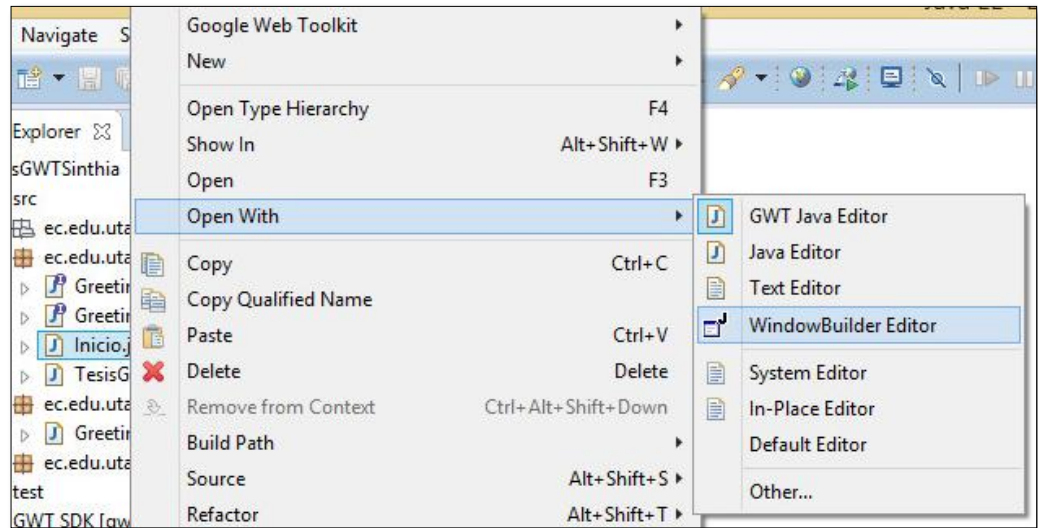


Gráfico 29. Parte visual de la ventana
Elaborado por: El Investigador

Componentes GXT

GWT y GXT son librerías de Java. Por lo tanto lo único que se tiene que hacer para tenerlas disponibles es descomprimir los archivos que bajamos de los sitios correspondientes en distintos directorios. Ej: C:\GWT-1.5M2 para GWT y C:\Gxt-1.0beta3 para GXT.

Los archivos importantes son los JARs de cada una de las librerías que se tienen que referenciar luego en los proyectos GXT.

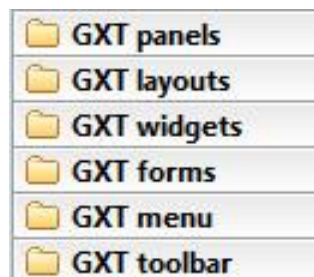


Gráfico 30. Componentes GXT
Elaborado por: El Investigador

GXT panels

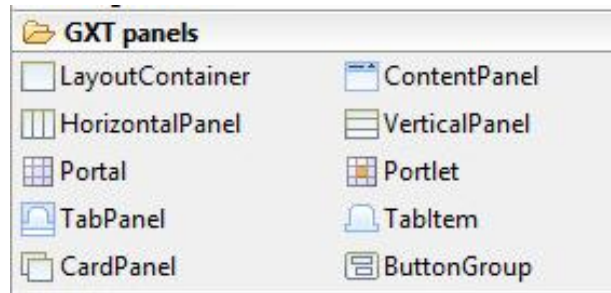


Gráfico 31. GXT panels
Elaborado por: El Investigador

- 1. LayoutContainer.-** Un contenedor que organiza los elementos usando un Layout. Los diseños son responsables de conectar los componentes secundarios al contenedor. Los diseños son muy flexibles ya que pueden crear cualquier estructura interna de elementos, insertando componentes secundarios en cualquier ubicación. [10]
- 2. ContentPanel.-** Es un contenedor de componentes que tiene funcionalidad específica y componentes estructurales que lo convierten en el bloque de construcción perfecto para interfaces de usuario orientadas a aplicaciones. El Panel contiene barras de herramientas inferior y superior, junto con secciones separadas de encabezado, pie de página y cuerpo. También proporciona un comportamiento expandible y plegable incorporado, junto con una variedad de botones de herramientas pre-construidos que se pueden conectar para proporcionar otro comportamiento personalizado. [10]
- 3. HorizontalPanel.-** Establece a sus elementos en una sola fila usando un TableRowLayout. Cada widget puede controlar las propiedades de su celda utilizando una instancia TableData asociada. [10]
- 4. VerticalPanel.-** Establece a sus elementos en una sola columna usando un TableRowLayout. Cada widget puede controlar las propiedades de su celda utilizando una instancia TableData asociada. [10]
- 5. Portal.-** Es contenedor de Portlets. [10]
- 6. Portlet.-** Un panel de contenido especializado contenido en un Portal. [10]
- 7. TabPanel.-** Contenedor para TabItem-s. [10]

8. **TabItem.-** se agregan a un TabPanel. TabItems puede ser cerrado, deshabilitado y los iconos de apoyo. [10]
9. **CardPanel.-** Este contenedor contiene varios widgets, cada uno se ajusta al contenedor, donde sólo un solo widget puede ser visible en un momento dado. Este estilo es más comúnmente usado para asistentes, implementaciones de pestañas, etc. [10]
10. **ButtonGroup.-** Un panel de contenido especializado para mostrar grupos de botones. [10]

GXT layouts

Los gxt layouts son propiedades que se añade a un contenedor, permitiendo de esta forma organizar los componentes.

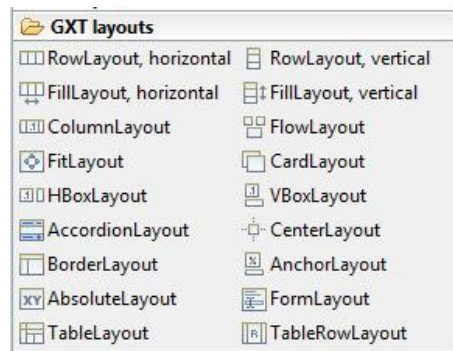


Gráfico 32. GXT layouts
Elaborado por: El Investigador

1. **RowLayout, horizontal.-** Diseño que posiciona a los elementos del contenedor en una sola fila horizontal. Cada componente puede especificar su altura y ancho en píxeles o como porcentaje. [10]
2. **RowLayout vertical.-** Diseño que posiciona a los elementos del contenedor en una sola fila vertical. Cada componente puede especificar su altura y ancho en píxeles o como porcentaje. [10]
3. **FillLayout horizontal.-** Diseño que coloca a los elementos del contenedor en una sola fila horizontal, obligándolos a tener el mismo tamaño. [10]
4. **FillLayout vertical.-** Diseño que coloca a los elementos del contenedor en una sola fila vertical, obligándolos a tener el mismo tamaño. [5]
5. **ColumnLayout.-** Esta diseño posiciona y dimensiona los elementos del contenedor en columnas horizontalmente. Cada componente puede

especificar su ancho en píxeles o como porcentaje de la anchura restante de los contenedores. [5]

- 6. FlowLayout.-** Este diseño simplemente hace que cada componente hijo este en su contenedor. El tamaño y la posición no se alteran por este diseño. [10]
- 7. FitLayout.-** Diseño que contiene un único elemento que se expande automáticamente para rellenar el contenedor del diseño. [10]
- 8. CardLayout.-** Este diseño contiene varios widgets, cada uno apropiado para el contenedor, donde sólo un solo widget puede ser visible en un momento dado. [10]
- 9. HBoxLayout.-** Controla que los elementos estén en una fila horizontal. Debido a este diseño de posiciones el contenedor no tendrá altura. Una altura se debe especificar directamente o por una disposición principal. [10]
- 10. VBoxLayout.-** Controla que los elementos estén en una fila vertical. Debido a este diseño de posiciones el contenedor no tendrá altura. Una altura se debe especificar directamente o por una disposición principal. [10]
- 11. AccordionLayout.-** Este es un diseño que contiene varios ContentPanels en un estilo de acordeón expandible tal que sólo un panel puede estar abierto en un momento dado. [5]
- 12. CenterLayout.-** Diseño que centra un solo widget dentro de su contenedor. [5]
- 13. BorderLayout.-** Diseño que simplemente hace que cada componente hijo este en su contenedor. El tamaño y la posición de los contenedores no se alteran por este diseño. [5]
- 14. AnchorLayout.-** Se trata de un diseño que permite el anclaje de widgets contenidos en relación con las dimensiones del contenedor. Si se cambia el tamaño del contenedor, todos los elementos anclados se vuelven a representar automáticamente de acuerdo con sus reglas de anclaje. [10]
- 15. AbsoluteLayout.-** Este diseño tiene capacidad para posicionamiento izquierdo / superior y permite también el anclaje de widgets contenidos en

relación con las dimensiones del contenedor. Si se cambia el tamaño del contenedor, todos los elementos anclados se vuelven a representar automáticamente de acuerdo con sus reglas de anclaje. [10]

16. FormLayout.- Diseño de campos de formulario y sus etiquetas. FormLayout sólo renderizará subclases de campo. Todos los demás componentes serán ignorados. [10]

17. TableLayout.- Permite renderizar contenido fácilmente, en una tabla HTML. Se puede especificar el número total de columnas. [10]

18. TableRowLayout.- Una sola fila de TableLayout. [10]

GXT widgets

Son pequeños artilugios que permiten acceder a funciones y proveer de información visual.

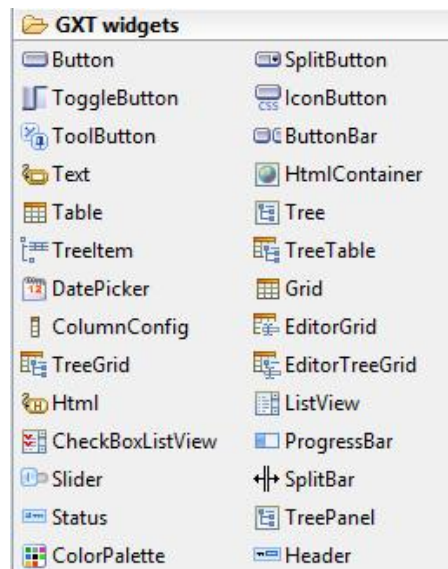


Gráfico 33. GXT widgets
Elaborado por: El Investigador

- 1. Button.-** un componente de botón.
- 2. SplitButton.-** Un botón de división que proporciona una flecha desplegable incorporada que puede activar un evento por separado del evento de clic predeterminado del botón. [5]
- 3. ToggleButton.-** es un botón de conmutación de 2 estados.

4. **IconButton.-** Un simple botón de estilo css con 3 estados: normal, más y desactivado. [10]
5. **ToolButton.-** Un IconButton que admite un conjunto de estilos predefinidos. [5]
6. **ButtonBar.-** Una fila horizontal de botones.
7. **Text.-** Un componente que contiene texto. El valor de texto no se trata como HTML. [10]
8. **HtmlContainer.-** Un contenedor especializado cuyo contenido se puede especificar como un elemento existente, un fragmento html o una URL remota. Al agregar elementos, se utiliza un selector css para identificar el elemento en el que se insertará el elemento. [10]
9. **Table.-** La tabla se utiliza para mostrar la tabla bidimensional de celdas.
10. **Tree.-** Un widget de árbol jerárquico estándar. El árbol contiene una jerarquía de TreeItem-s, que el usuario puede abrir, cerrar y seleccionar. [5]
11. **TreeItem.-** Un elemento en un árbol. Todos los eventos se burbujan al árbol padre del elemento. [10]
12. **TreeTable.-** Un widget de árbol jerárquico con soporte para columnas adicionales. El árbol contiene una jerarquía de TreeTableItems que el usuario puede abrir, cerrar y seleccionar. [10]
13. **DatePicker.-** Recogedor simple de la fecha.
14. **Grid.-** Esta clase representa la interfaz primaria de un control de cuadrícula basado en componentes. La cuadrícula requiere un ListStore y un ColumnModel cuando se construye. Cada modelo en el almacén se representará como una fila en la cuadrícula. [10]
15. **ColumnConfig.-** Una columna config para una columna en un modelo de columna. [5]
16. **EditorGrid.-** Agrega capacidades de edición a la cuadrícula. [10]
17. **TreeGrid.-** Una rejilla de árbol jerárquica vinculada a un TreeStore. [5]
18. **EditorTreeGrid.-** Agrega funciones de edición a TreeGrid. [5]

19. **Html.-** Un componente que contiene texto HTML arbitrario. La etiqueta HTML de envoltente se puede configurar mediante la propiedad "tagName". [10]
20. **ListView.-** Un mecanismo para mostrar datos usando plantillas de diseño personalizadas. ListView utiliza un XTemplate como su mecanismo de plantilla interna. [10]
21. **CheckBoxListView.-** Extensión de ListView para comprobar algunos elementos. [10]
22. **ProgressBar.-** Objeto de interfaz de usuario seleccionable y no seleccionable que se utiliza para mostrar el progreso.
23. **Slider.-** Objeto de interfaz de usuario seleccionable que representa un rango de valores numéricos positivos. [10]
24. **SplitBar.-** Un splitter arrastrable en el lado de un widget. [10]
25. **Status.-** Componente para mostrar el icono de estado y el texto, normalmente dentro de la barra de herramientas. [10]
26. **TreePanel.-** Un widget de árbol jerárquico enlazado directamente a un TreeStore. TreePanel no contiene widgets secundarios, sino que el árbol se procesa en función de los modelos contenidos en el almacén de árboles. Una vez enlazado, el árbol permanecerá en sincronía con la tienda de árbol vinculada. [5]
27. **ColorPalette.-** Componente simple para elegir el color.
28. **Header.-** Un componente personalizado que admite un icono, un texto y un área de herramientas. [10]

GXT forms

Permite dar formato a las ventanas.

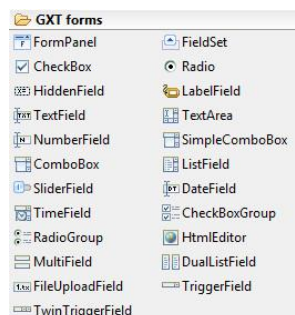


Gráfico 34. GXT forms
Elaborado por: El Investigador

1. **FormPanel.-** Un panel para mostrar widgets de formularios. De forma predeterminada, FormPanel utiliza FormLayout, pero esto puede anularse cuando se utilizan diseños anidados. [10]
2. **FieldSet.-** Un contenedor que envuelve su contenido en un campo HTML. [10]
3. **CheckBox.-** Campo de casilla de verificación individual. A diferencia de otros campos, las activaciones de casilla de verificación cambian los eventos cuando se cambia el estado de los radios, no en el desenfoco. [10]
4. **Radio.-** Campo de radio único. Igual que CheckBox, pero proporcionado como una conveniencia para configurar automáticamente el tipo de entrada. El agrupamiento de radio es manejado automáticamente por el navegador si se da a cada radio en un grupo con el mismo nombre. [5]
5. **HiddenField.-** Un campo oculto básico para almacenar valores ocultos en formularios que necesitan ser pasados en el formulario. [5]
6. **LabelField.-** Muestra texto estático.
7. **TextField.-** Campo de texto básico de una sola línea.
8. **TextArea.-** Campo de texto Multilínea. Se puede utilizar como un reemplazo directo para los campos de área de texto tradicionales. [10]
9. **NumberField.-** Campo de texto numérico que proporciona filtrado automático de pulsaciones de teclas y validación numérica.
10. **SimpleComboBox.-** Una subclase ComboBox que admite cualquier tipo de datos simple. [10]
11. **ComboBox.-** Un componente de cuadro combinado. [10]
12. **ListField.-** Un campo de lista de selección múltiple. [5]
13. **SliderField.-** Campo que envuelve Slider. [10]
14. **DateField.-** Proporciona un campo de entrada de fecha con una lista desplegable de DatePicker y validación automática de fecha. [10]
15. **TimeField.-** Proporciona un campo de entrada de tiempo con una lista desplegable de tiempo y una validación automática de hora. [10]
16. **CheckBoxGroup.-** Un grupo de CheckBox.
17. **RadioGroup.-** Un grupo de Radio.

18. **HtmlEditor.**- Proporciona un ligero componente HTML Editor. [10]
19. **MultiField.**- Muestra varios campos en una sola fila o columna. [10]
20. **DualListField.**- Combina dos campos de lista y permite que las selecciones se muevan entre campos utilizando botones o arrastrando y soltando selecciones. [10]
21. **FileUploadField.**- Se ha quedado un archivo. Cuando se utiliza este campo, la codificación del panel de formulario que contiene debe establecerse en MULTIPART, y el método debe establecerse en POST. [10]
22. **TriggerField.**- Proporciona una envoltura conveniente para TextFields que agrega un botón de activación que se puede hacer clic (se ve como una combobox por defecto). [10]
23. **TwinTriggerField.**- Proporciona un envoltorio conveniente para TextFields que agrega dos botones de activación que se pueden hacer clic (parece un cuadro combinado de forma predeterminada). [10]

GXT menú

Están los tipos de menú que se va a utilizar en el sistema.

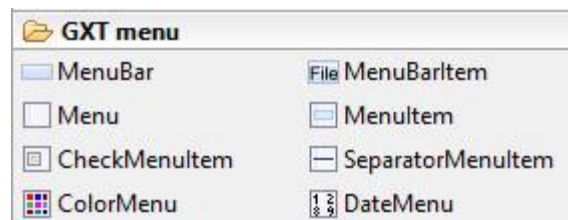


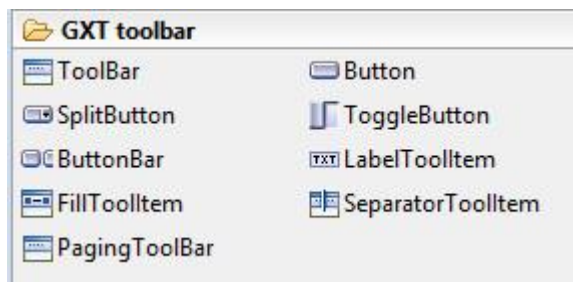
Gráfico 35. Gráfico 30. GXT menu
Elaborado por: El Investigador

1. **MenuBar.**- Una barra de menú horizontal.
2. **MenuItem.**- Elemento en MenuBar, contiene elementos Menu.
3. **Menu.**- Un componente de menú - contenedor para MenuItem-s.
4. **MenuItem.**- Las instancias de esta clase representan un objeto de interfaz de usuario seleccionable que representa un elemento en un menú. [5]
5. **CheckMenuItem.**- Añade un elemento de menú que contiene una casilla de verificación por defecto, pero también puede formar parte de un grupo de radio. [5]

6. **SeparatorMenuItem.**- Añade una barra separadora a un menú, utilizada para dividir grupos lógicos de elementos de menú. [5]
7. **ColorMenu.**- Menú especializado que muestra ColorPalette.
8. **DateMenu.**- Menú especializado que muestra DatePicker.

GXT toolbar

Es una barra de herramientas, es un componente que contiene iconos o botones.



*Gráfico 36. GXT toolbar
Elaborado por: El Investigador*

1. **ToolBar.**- Una barra de herramientas estándar.
2. **Button.**- Un componente de botón.
3. **SplitButton.**- Un botón de división que proporciona una flecha desplegable incorporada que puede activar un evento por separado del evento de clic predeterminado del botón. [5]
4. **ToggleButton.**- Un botón de conmutación de 2 estados.
5. **ButtonBar.**- Una fila horizontal de botones.
6. **LabelToolItem.**- Elemento para mostrar texto estático.
7. **FillToolItem.**- Rellena el ancho de la barra de herramientas, empujando los elementos recién agregados a la derecha. [10]
8. **SeparatorToolItem.**- Un separador de barra de herramientas.
9. **PagingToolBar.**- Una barra de herramientas especializada que está enlazada a un ListLoader y proporciona controles de paginación automáticos. [10]

Creación de Ventanas

En la creación de ventanas lo primero que se debe hacer es añadir un diseño al contenedor, para ello se hace uso de los layouts “GXT layouts”.

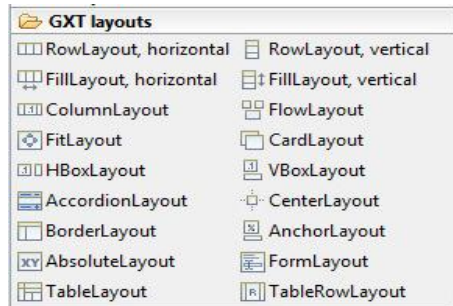


Gráfico 37. GXT layouts
Elaborado por: El Investigador

19. El diseño más utilizado en GWT para las ventanas es el **Border Layout**, que permite dividir a la ventana en regiones de trabajo. Para asignar un diseño a la ventana, se hace clic derecho sobre la ventana y dentro de la pestaña **Set Layout**, y a continuación se escogerá el tipo de diseño que desea utilizar.

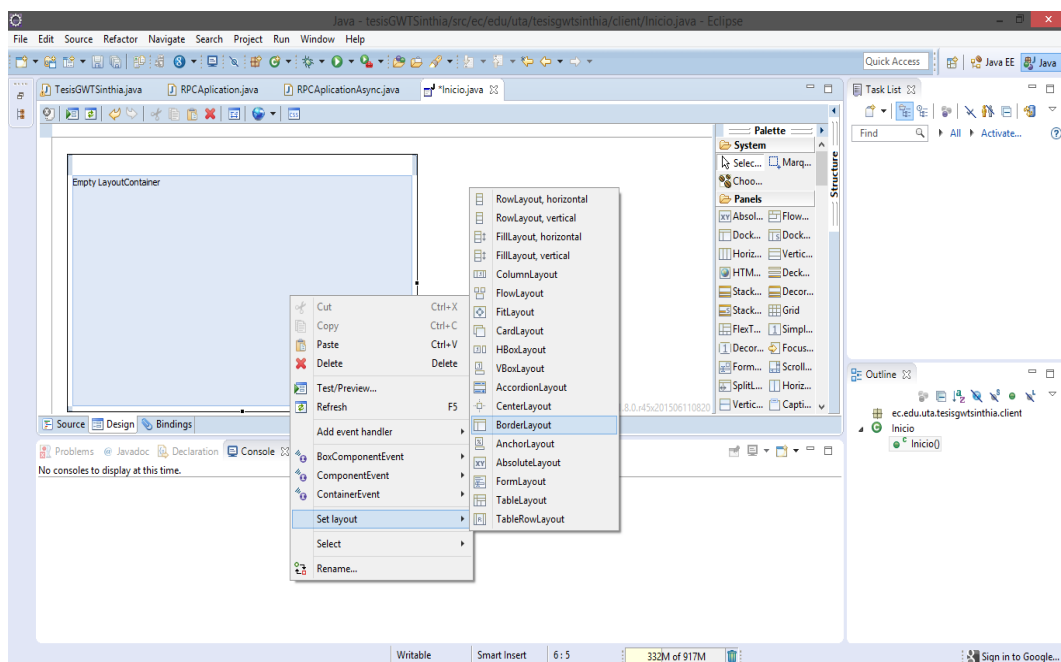


Gráfico 38. Añade el diseño “Border Layout”.
Elaborado por: El Investigador

20. Para hacer uso de los componentes de GXT, es necesario crear un **Layout**, el mismo que permite organizar de mejor manera los componentes dentro de la ventana.

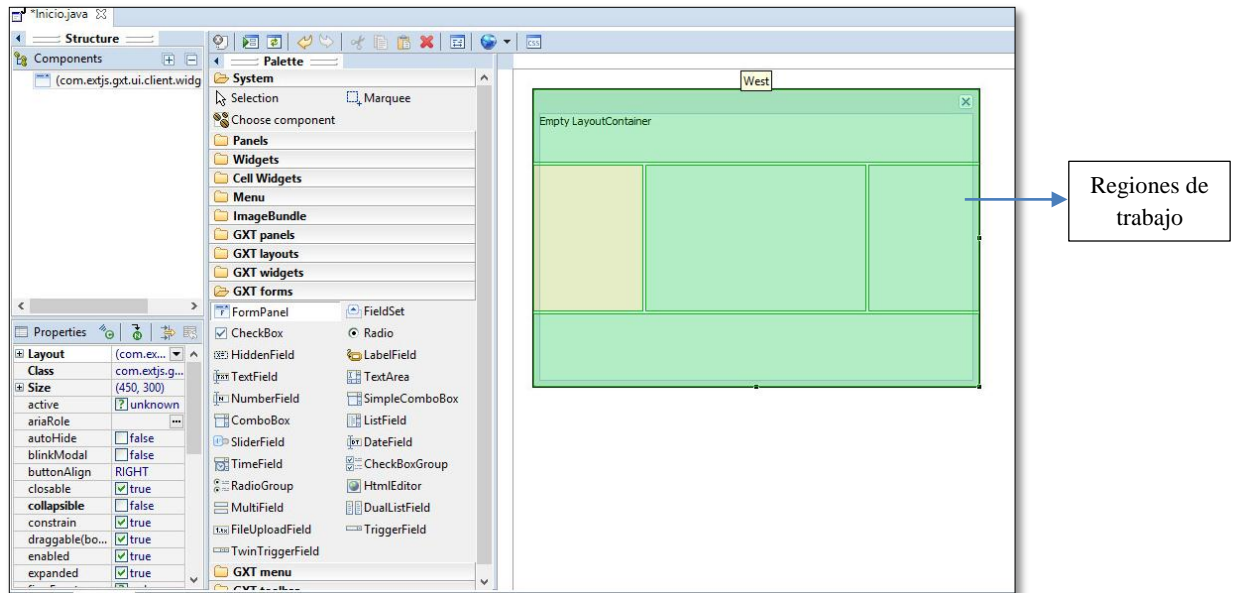


Gráfico 39. Componentes GXT “Border Layout”.
Elaborado por: El Investigador

Al compilar la aplicación la página principal se visualizará de la siguiente forma.

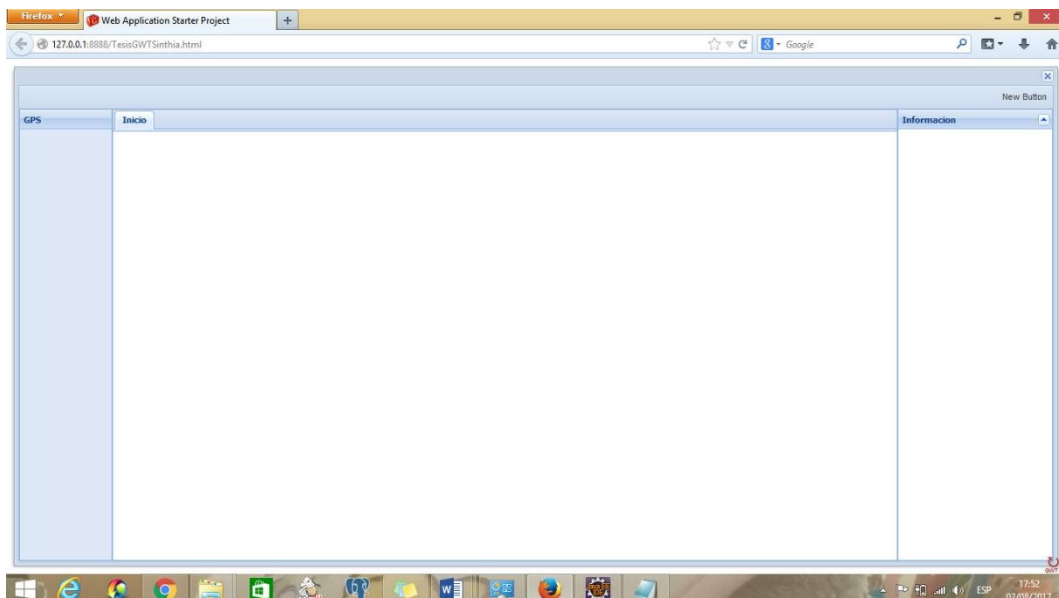


Gráfico 40. Gráfico 32. Vista General del Sistema
Elaborado por: El Investigador

Una propiedad de la programación orientada a objetos, es encapsular las variables y trabajar con métodos y funciones.

Para que la clase cumpla dicho objetivo es necesario que esta implemente la interfaz **Serializable** y **BeanModelTag**.

Serialización de un objeto: Implementar Serializable, para que un programa java pueda convertir un objeto en un montón de bytes y pueda luego recuperarlo, el objeto necesita ser Serializable. [6]

BeanModelTag: Interfaz de etiqueta utilizada para identificar Java Beans que se pueden utilizar para generar instancias de BeanModel. [6]

Creación del Servicio Remoto

21. Para crear un servicio remoto, seleccionamos el proyecto y realizamos clic derecho, a continuación selecciona la opción **Google Web Toolkit** y **GWT remote service**.

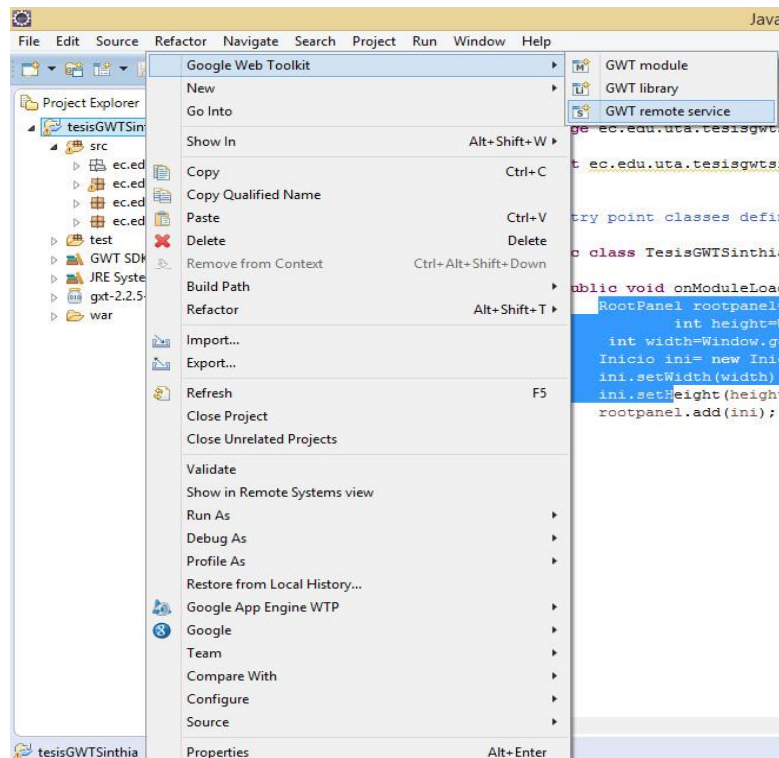


Gráfico 41. Creación del servicio remoto.
Elaborado por: El Investigador

22. Asignar el nombre del servicio remoto y hacer clic en **Finish**.

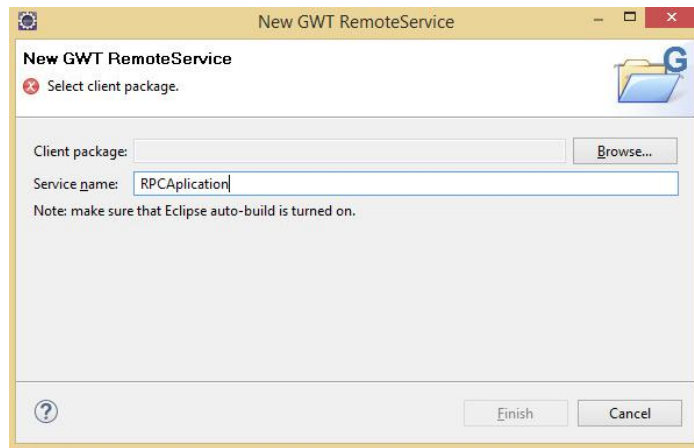


Gráfico 42. Asigna el nombre del servicio remoto.
Elaborado por: El Investigador

23. Al crear el servicio remoto con el nombre **RPCApplication**, se creará con un error, el mismo que indica que para crear un servicio remoto es necesario tener la clase asincrónica de tipo interfaz.

Para solucionar el error, dar clic derecho sobre la línea de código que indica el error y seleccionar la opción **Create interface 'RPCApplicationAsync'**.

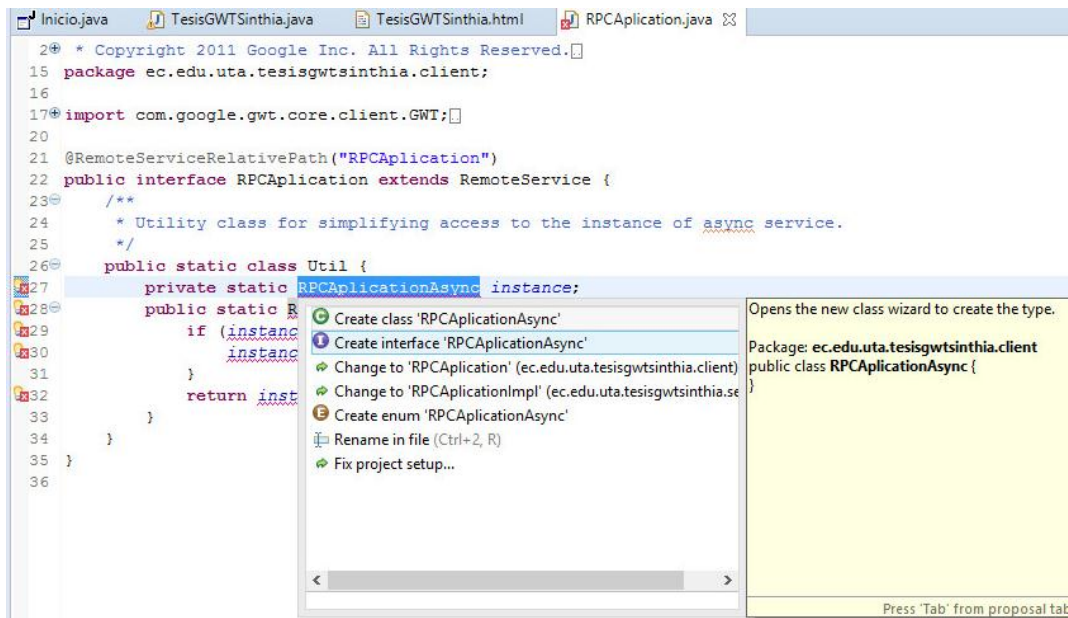
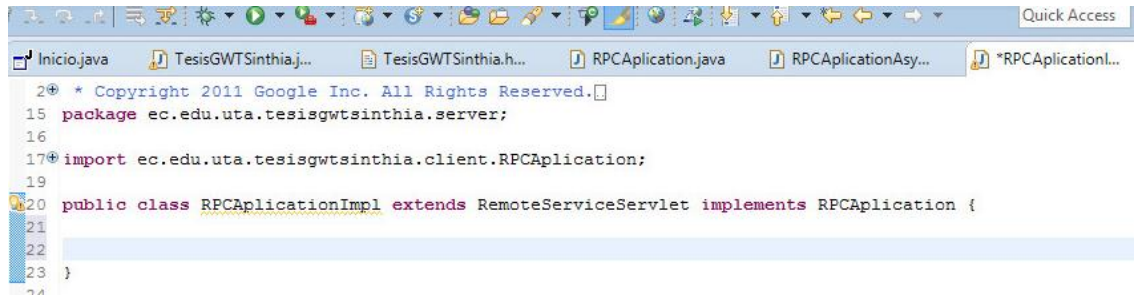


Gráfico 43. Creación de la clase asincrónica **RPCApplicationAsync**
Elaborado por: El Investigador

24. La clase asincrónica se crea de forma automática, después de haber seleccionado la solución recomendada por eclipse.

Creación de una conexión

25. En la clase asincrónica creada anteriormente se programarán los métodos y funciones que van a interactuar directamente con la Base de Datos.



```
20 * Copyright 2011 Google Inc. All Rights Reserved.
15 package ec.edu.uta.tesisgwtsinthia.server;
16
17 import ec.edu.uta.tesisgwtsinthia.client.RPCApplication;
18
19
20 public class RPCApplicationImpl extends RemoteServiceServlet implements RPCApplication {
21
22
23
24 }
```

Gráfico 44. Clase asincrónica RPCApplicationAsync.
Elaborado por: El Investigador

26. Para crear la conexión es necesario añadir el **jar** al **Path**. Para ello hay que tener el **jar postgres**, en la carpeta **war – WEB-INF – lib**.

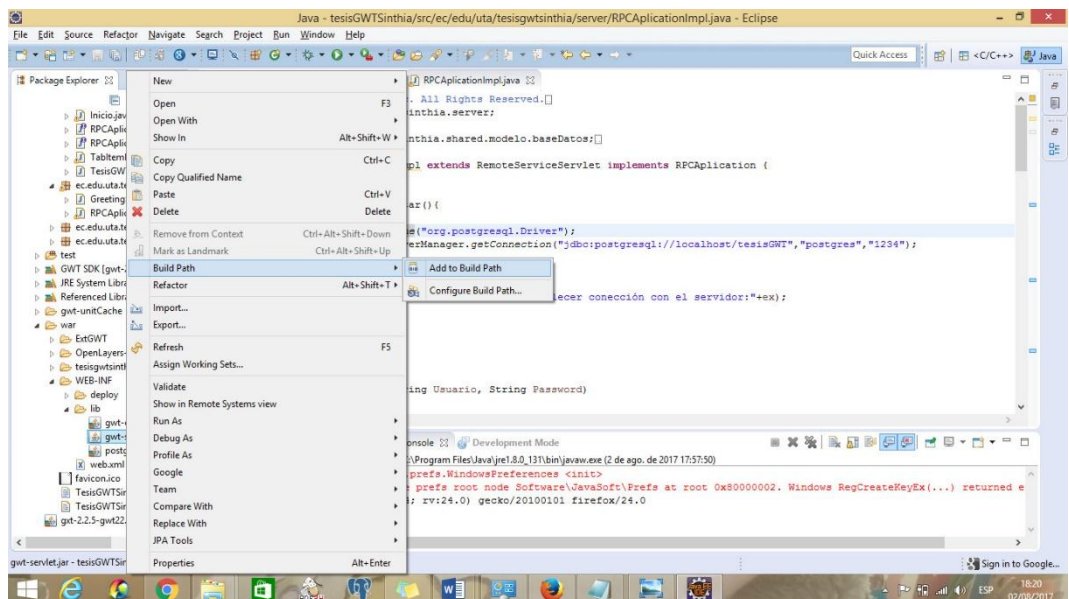


Gráfico 45. Añade el jar al Path, para crear una conexión con la BD.
Elaborado por: El Investigador

27. La cadena de conexión se la programa en el lado del servidor en la clase **RPCApplicationImpl**.


```

package ec.edu.uta.tesisgwtsinthia.server;

import ec.edu.uta.tesisgwtsinthia.client.RPCApplication;

import java.sql.Connection;
import java.sql.DriverManager;

import com.google.gwt.user.client.Window;
import com.google.gwt.user.server.rpc.RemoteServiceServlet;

public class RPCApplicationImpl extends RemoteServiceServlet implements RPCApplication {

    Connection connect=null;
    public Connection conectar(){
        try {
            Class.forName("org.postgresql.Driver");
            connect=DriverManager.getConnection("jdbc:postgresql://localhost/tesisGWT","postgres","1234");
        } catch (Exception ex) {
            System.out.println(ex);
            Window.alert("No se ha podido establecer conexión con el servidor:"+ex.getMessage());
        }

        return connect;
    }
}

```

Gráfico 46. Código de conexión de la Base de Datos.
Elaborado por: El Investigador

Creación del método Login

28. La creación del método **Login** se la realiza en el lado del servidor en la clase **RPCApplicationImpl.java**.

Se realiza una consulta a la tabla **user** para comprobar la existencia del usuario.

```

public boolean Login(String Usuario, String Password)
{
    boolean result =false;
    String sql="select * from usuarios where ced_usu= '" + Usuario + "' and usu_password='" + Password +"'";
    Connection cn=conectar();
    try {
        Statement st=cn.createStatement();
        ResultSet rs=st.executeQuery(sql);
        while (rs.next())
        {
            return true;
        }
    } catch (SQLException e) {
        System.out.println(e);
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return result;
}

```

Gráfico 47. Creación del método Login.
Elaborado por: El Investigador

29. En la parte del cliente en la clase **RPCApplication.java** copia el método **Login** creado anteriormente, el mismo que se creará con un error, que se solucionará mediante la herramienta.

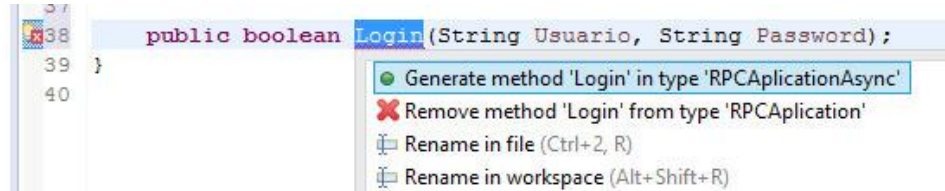


Gráfico 48. Genera el método en la clase asincrónica
Elaborado por: El Investigador

30. En la clase asincrónica **RPCApplicationAsync.java**, crea el método el mismo que va a ser llamado en las vistas del sistema.

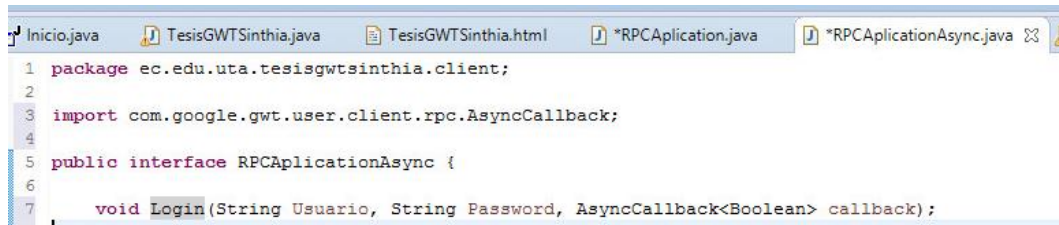


Gráfico 49. Creación del método Login en la clase asincrónica.
Elaborado por: El Investigador

31. En la vista del cliente se realiza el llamado del método que está en la clase asincrónica creado en el paso anterior.

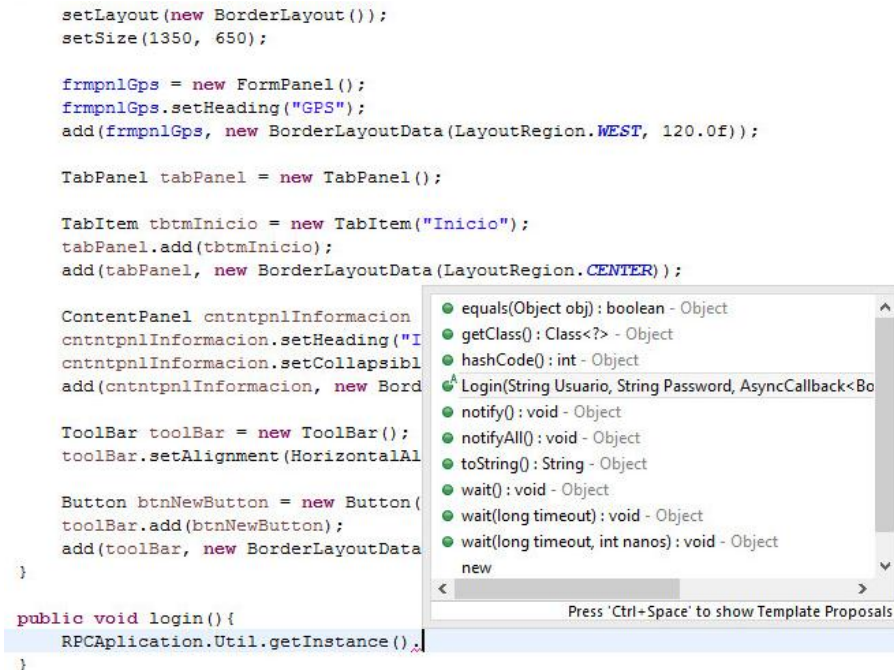


Gráfico 50. Llamado del método Login de la clase asincrónica.
Elaborado por: El Investigador

32. En el lado del cliente, en la clase Inicio se realiza el llamado del método login a la vista de datos.

```

public void Login(){
    RPCApplication.Util.getInstance().Login(txtfldUsuario.getValue().toString(), txtfldContrasea.getValue().toString(), ne

    @Override
    public void onFailure(Throwable caught) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onSuccess(Boolean result) {
        // TODO Auto-generated method stub
        if(result==false){
            com.google.gwt.user.client.Window.alert("Usuario o Clave de acceso Incorrectos");
            txtfldUsuario.setValue("");
            txtfldContrasea.setValue("");
        }else{
            cntntpnlInicioSesion.setVisible(false);
            lblfldUsuario.setText("Usuario: "+txtfldUsuario.getValue().toString());
            txtfldUsuario.setValue("");
            txtfldContrasea.setValue("");
            btnCerrarSesion.setVisible(true);
            frmmpnlDatosPersonales.setVisible(true);

            int height=tabPanel.getHeight()-25;
            int width=tabPanel.getWidth();
            habilitarMapaGps(height,width);

```

Gráfico 51. Llamado del método Login en la clase Inicio.
Elaborado por: El Investigador

33. A continuación se realiza el llamado del método en el botón **Ingresar**.

```

'''
    btnIngresar.addSelectionListener(new SelectionListener<ButtonEvent>() {
        public void componentSelected(ButtonEvent ce) {
            if(txtfldUsuario.isValid() && txtfldContrasea.isValid()){
                Login();
            }else{
                com.google.gwt.user.client.Window.alert("Debe ingresar Datos");
            }
        }
    });
    cntntpnlInicioSesion.add(btnIngresar, new FormData("100%"));
    cntntpnlInformacin.add(cntntpnlInicioSesion);
    inicioSistema();
    add(cntntpnlInformacin, new BorderLayoutData(LayoutRegion.EAST, 215.0f));
}

```

Gráfico 52. Llamado del método Login en el botón Ingresar.
Elaborado por: El Investigador

34. La vista del sistema es la siguiente.

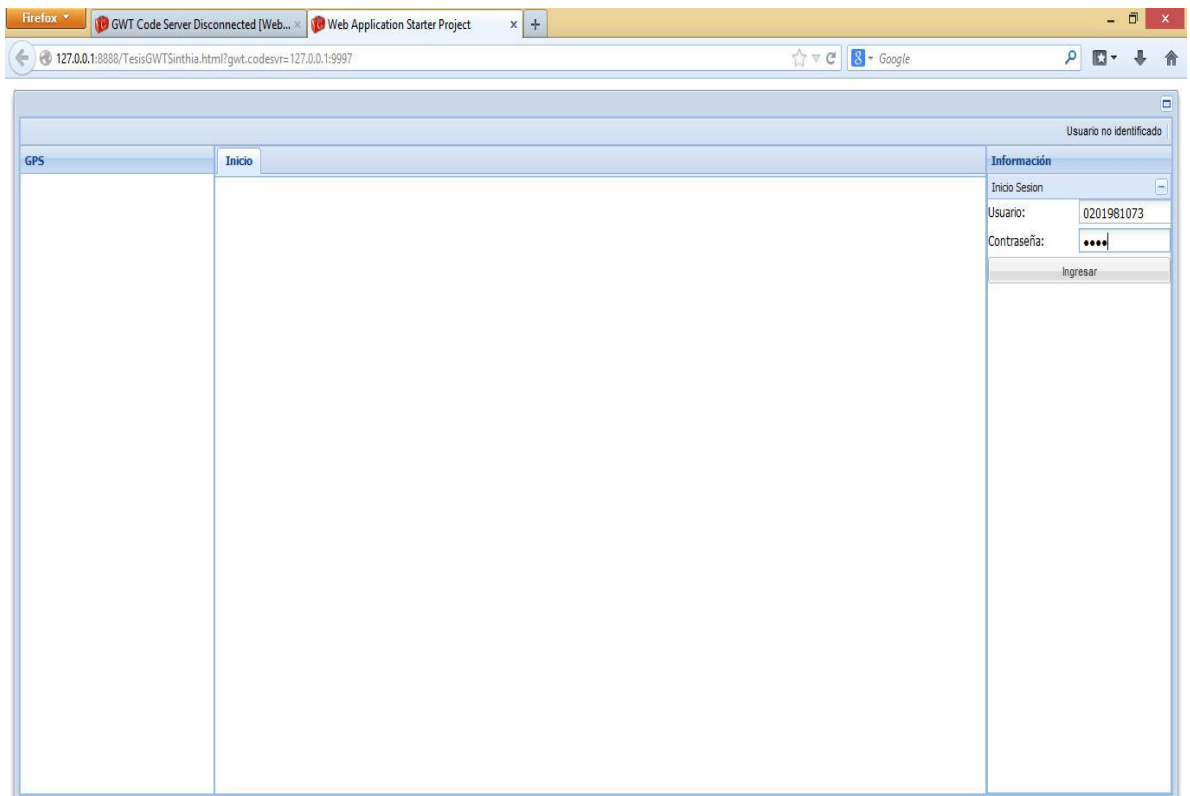


Gráfico 53. Vista General del Sistema..
Elaborado por: El Investigador

Ingreso al Sistema

35. Después de ingresar el usuario y la contraseña el sistema se visualiza de la siguiente forma:

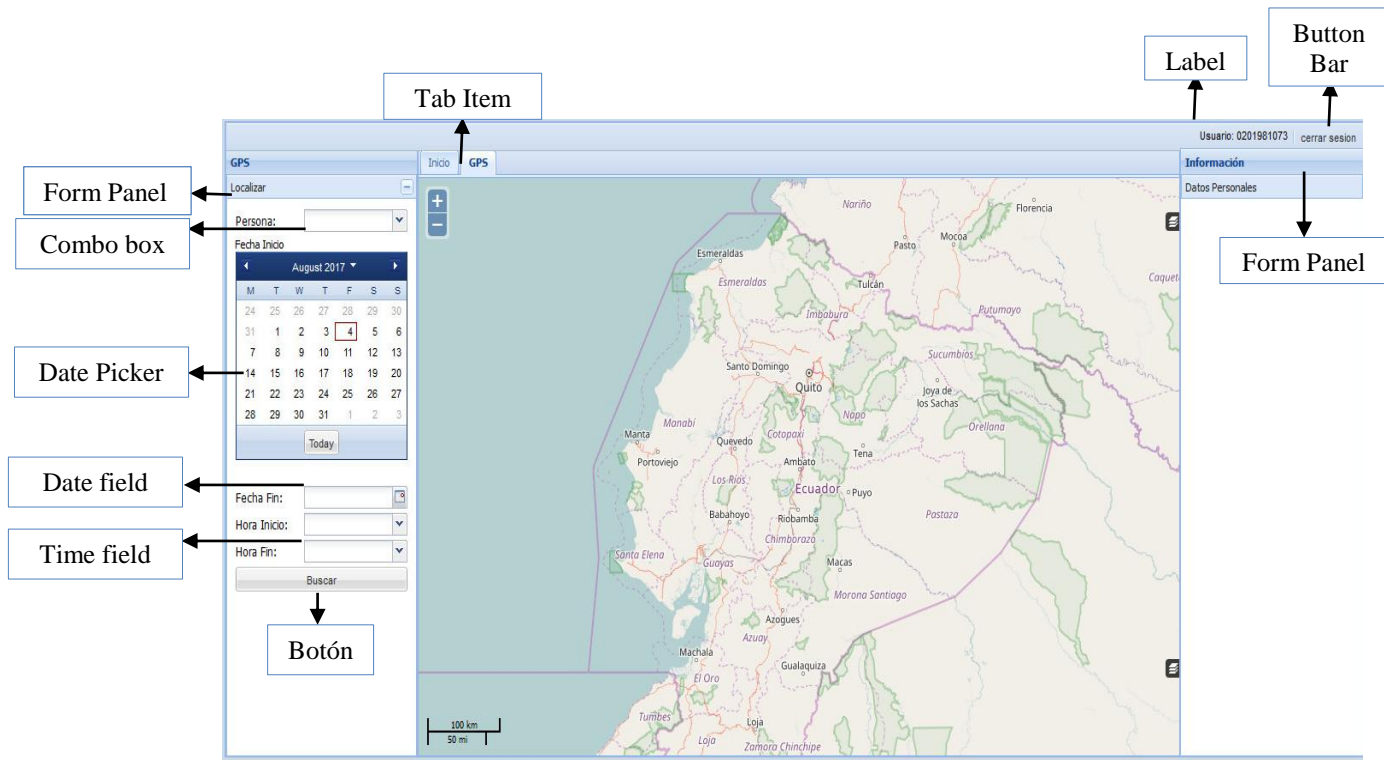


Gráfico 54. Form Panel GPS.
Elaborado por: El Investigador

36. Para la selección de las personas se utiliza un **combo box**, que permite cargar las personas designadas para el usuario en caso que sea un usuario institucional.



Gráfico 55. Combo box Personas.
Elaborado por: El Investigador

37. Permite seleccionar una fecha a través de un **data picker** adjunto en el componente.

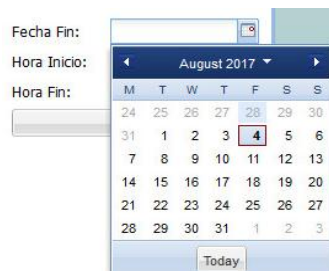


Gráfico 56. Data picker, fecha fin.
Elaborado por: El Investigador

38. Para la hora de inicio y fin se utilizó un **time field**, que permite seleccionar la hora con un cierto lapso de tiempo.

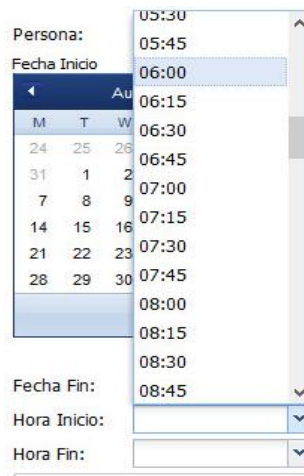


Gráfico 57. Time field, hora inicio y fin.
Elaborado por: El Investigador

Para el incremento de tiempo se configura las propiedades que presenta el **time field**.

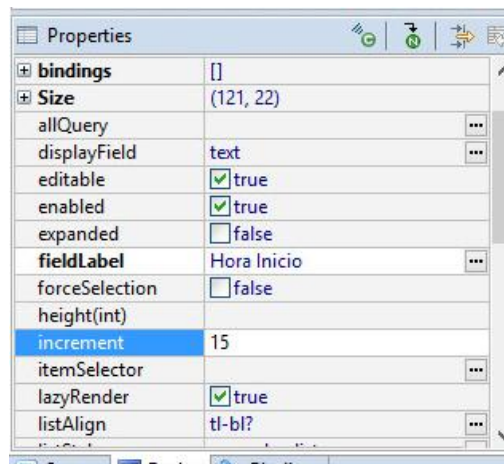


Gráfico 58. Propiedades de time field.
Elaborado por: El Investigador

39. Después de llenar los datos y seleccionar **Buscar**, la búsqueda se visualizará de la siguiente forma:

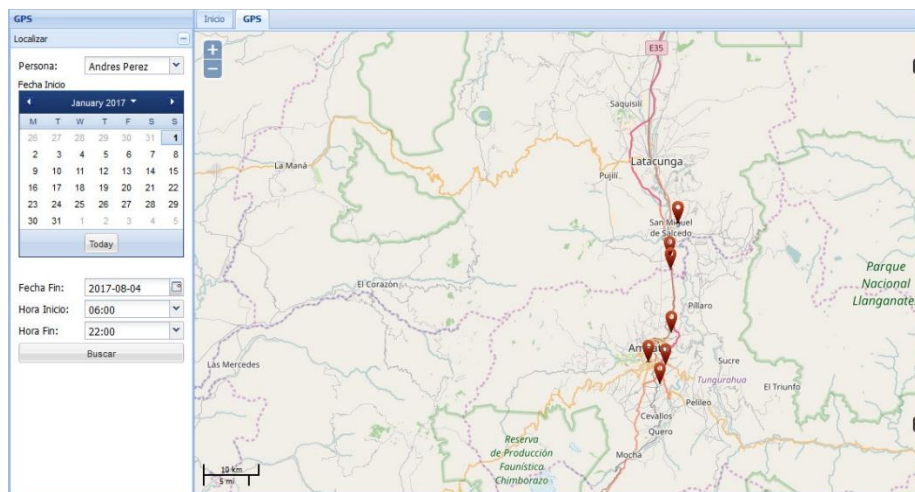


Gráfico 59. Visualización de la búsqueda.
Elaborado por: El Investigador

OpenLayers

OpenLayers hace que sea fácil poner un mapa dinámico en cualquier página web. Puede mostrar mosaicos de mapa, datos vectoriales y marcadores cargados desde cualquier fuente. OpenLayers ha sido desarrollado para fomentar el uso de información geográfica de todo tipo. Es completamente gratuito, JavaScript de código abierto.

Configuración del uso de Open Layers

40. En el presente proyecto se hace uso de la biblioteca **OpenLayers**, por lo tanto, es necesario descargar y descomprimir la carpeta OpenLayers que va a estar ubicada dentro de la **carpeta war** del proyecto, donde se encuentran las librerías y componentes de esta biblioteca.

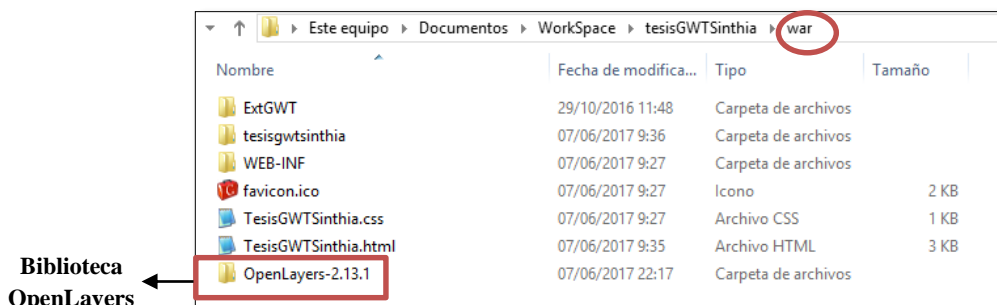


Gráfico 60. Biblioteca Open Layers.
Elaborado por: El Investigador

41. Se debe incluir la librería **.jar** para que permita manipular los mapas dentro de GWT.

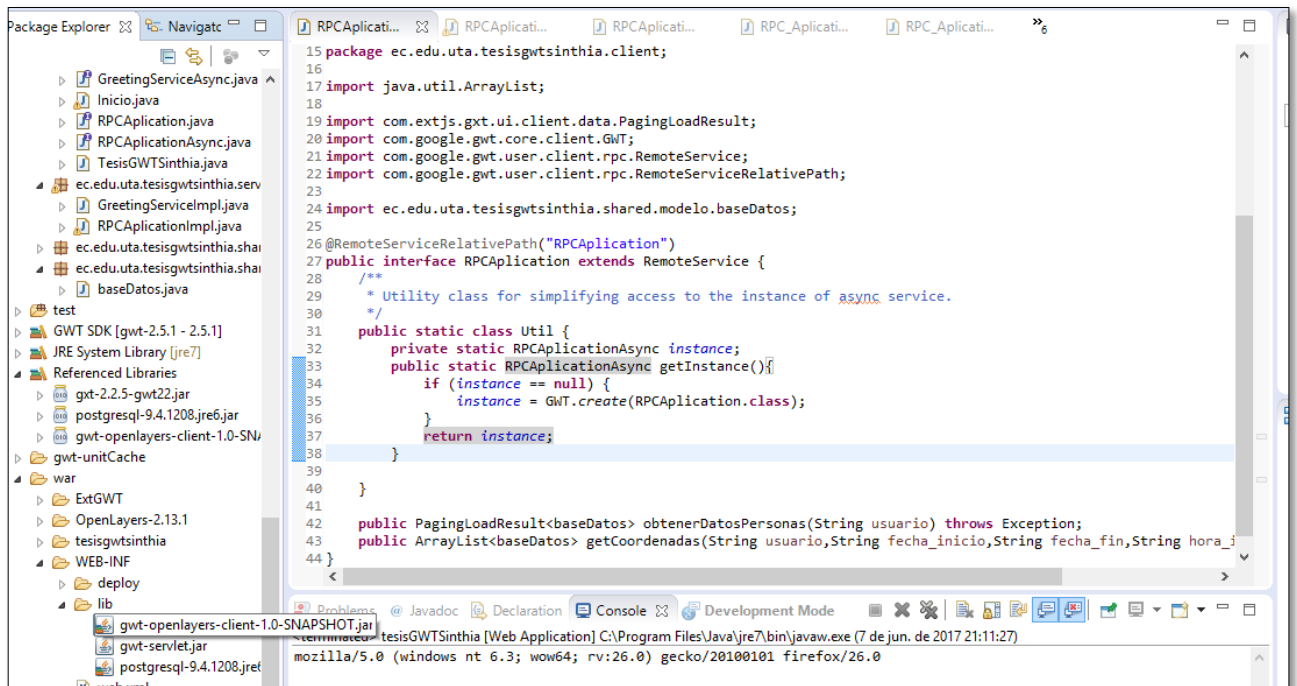


Gráfico 61. Librería jar – manipulación de mapas.
Elaborado por: El Investigador

La ventaja de GWT es que existe una página propia de OpenLayers con el siguiente link: <http://www.gwtopenlayers.org/user-reference.html>, donde se encuentran ejemplos del uso de estos mapas, y permiten copiar el código javascript para poder hacer uso de estos mapas.

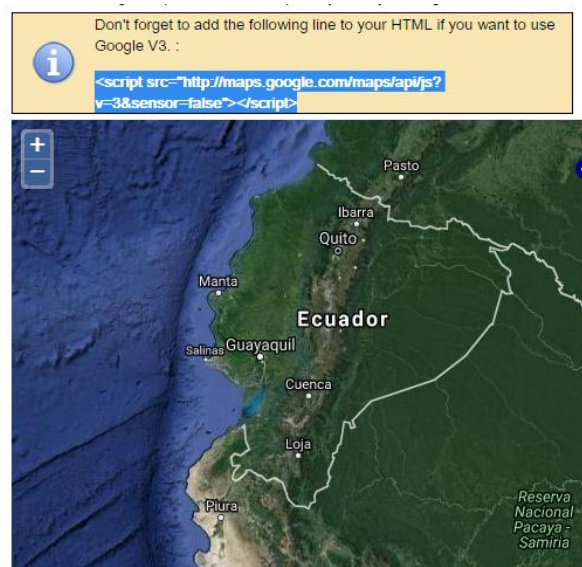
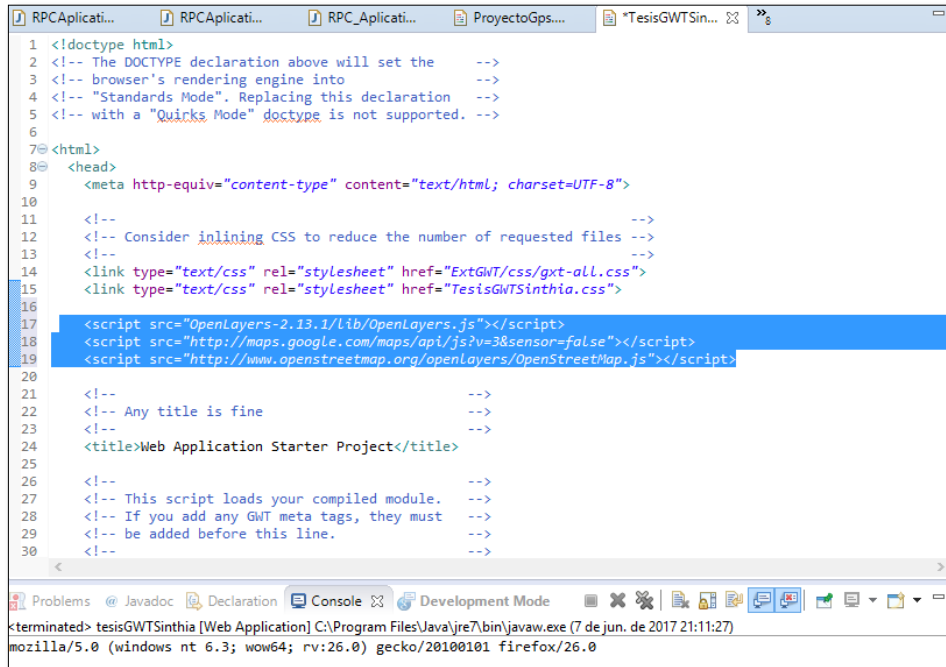


Gráfico 62. Código javascript – OpenLayers de GWT.
Elaborado por: El Investigador

El código copiado anteriormente se lo debe incluir en el archivo html del proyecto, éste se encuentra dentro de **war**.



```
1 <!doctype html>
2 <!-- The DOCTYPE declaration above will set the -->
3 <!-- browser's rendering engine into -->
4 <!-- "Standards Mode". Replacing this declaration -->
5 <!-- with a "Quirks Mode" doctype is not supported. -->
6
7 <html>
8 <head>
9 <meta http-equiv="content-type" content="text/html; charset=UTF-8">
10
11 <!-- -->
12 <!-- Consider inlining CSS to reduce the number of requested files -->
13 <!-- -->
14 <link type="text/css" rel="stylesheet" href="ExtGWT/css/gxt-all.css">
15 <link type="text/css" rel="stylesheet" href="TesisGWTSinthia.css">
16
17 <script src="OpenLayers-2.13.1/Lib/OpenLayers.js"></script>
18 <script src="http://maps.google.com/maps/api/js?v=3&sensor=false"></script>
19 <script src="http://www.openstreetmap.org/openlayers/OpenStreetMap.js"></script>
20
21 <!-- -->
22 <!-- Any title is fine -->
23 <!-- -->
24 <title>Web Application Starter Project</title>
25
26 <!-- -->
27 <!-- This script loads your compiled module. -->
28 <!-- If you add any GWT meta tags, they must -->
29 <!-- be added before this line. -->
30 <!-- -->
```

Gráfico 63. Código OpenLayers - mapas.
Elaborado por: El Investigador

42. Dentro del archivo con extensión **.gwt.xml** que se encuentra en el package por defecto del proyecto, se debe incluir la siguiente línea, para referenciar el uso de openlayers.

<inherits name='org.gwtopenmaps.openlayers.OpenLayers'!>

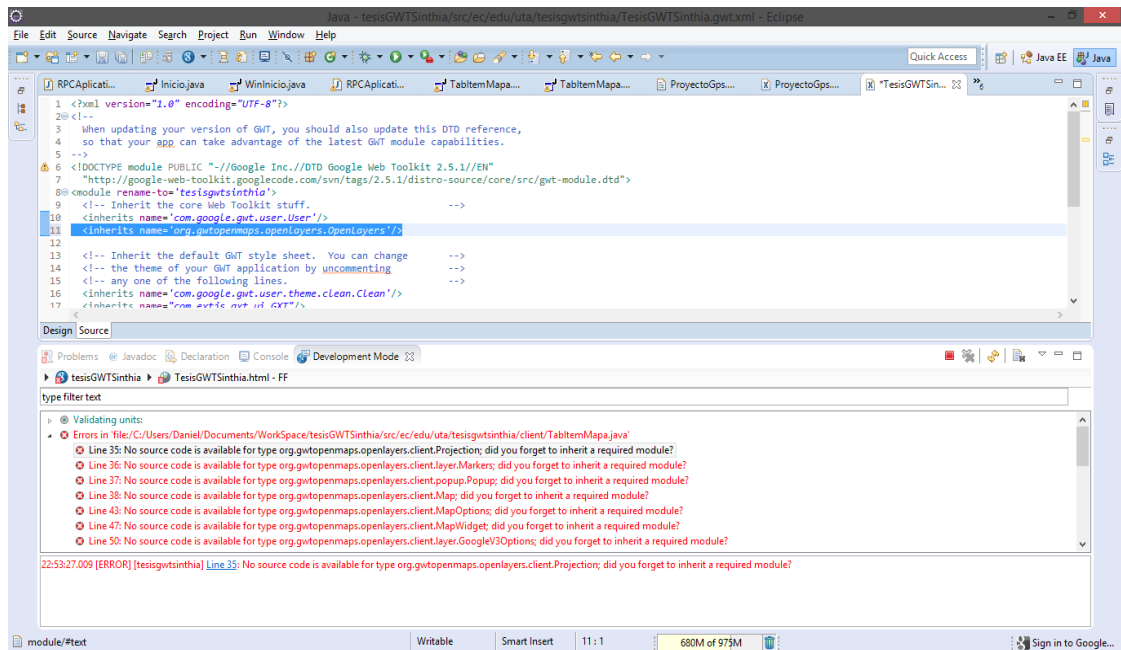


Gráfico 64. Archivo de configuración openlayers.
Elaborado por: El Investigador

43. Extiende Tab Item para hacer el llamado en el tab panel.

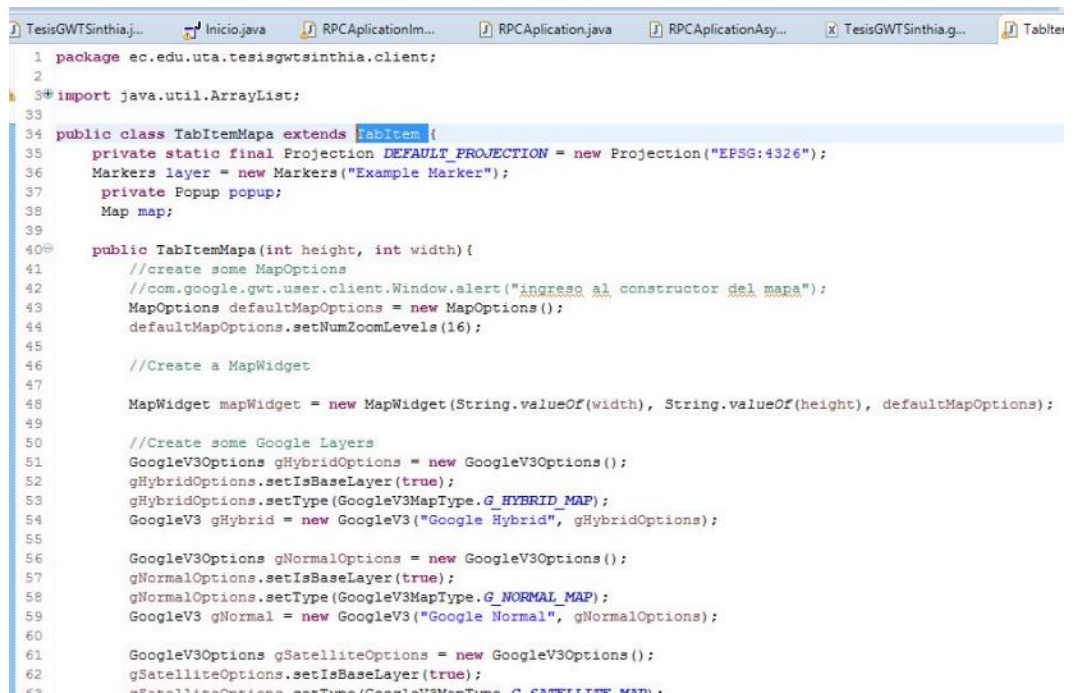


Gráfico 65. Llamado en el tab panel de los mapas.
Elaborado por: El Investigador

44. Especifica la longitud y latitud para indicar la ubicación mediante marcadores.

```

1 package ec.edu.uta.tesisgwtsinthia.client;
2
3 import java.util.ArrayList;
4
5 public class TabItemMapa extends TabItem {
6     private static final Projection DEFAULT_PROJECTION = new Projection("EPSG:4326");
7     Markers layer = new Markers("Example Marker");
8     private Popup popup;
9     Map map;
10
11 public TabItemMapa(int height, int width){
12     //create some MapOptions
13     //com.google.gwt.user.client.Window.alert("ingreso al constructor del mapa");
14     MapOptions defaultMapOptions = new MapOptions();
15     defaultMapOptions.setNumZoomLevels(16);
16
17     //Create a MapWidget
18
19     MapWidget mapWidget = new MapWidget(String.valueOf(width), String.valueOf(height), defaultMapOptions);
20
21     //Create some Google Layers
22     GoogleV3Options gHybridOptions = new GoogleV3Options();
23     gHybridOptions.setIsBaseLayer(true);
24     gHybridOptions.setType(GoogleV3MapType.G_HYBRID_MAP);
25     GoogleV3 gHybrid = new GoogleV3("Google Hybrid", gHybridOptions);
26
27     GoogleV3Options gNormalOptions = new GoogleV3Options();
28     gNormalOptions.setIsBaseLayer(true);
29     gNormalOptions.setType(GoogleV3MapType.G_NORMAL_MAP);
30     GoogleV3 gNormal = new GoogleV3("Google Normal", gNormalOptions);
31
32     GoogleV3Options gSatelliteOptions = new GoogleV3Options();
33     gSatelliteOptions.setIsBaseLayer(true);

```

Gráfico 66. Marcadores de ubicación en el mapa.
Elaborado por: El Investigador

45. El código permite visualizar el mapa por defecto que presenta el OpenLayer.

```

41 //Create some MapOptions
42 //com.google.gwt.user.client.Window.alert("ingreso al constructor del mapa");
43 MapOptions defaultMapOptions = new MapOptions();
44 defaultMapOptions.setNumZoomLevels(16);
45
46 //Create a MapWidget
47
48 MapWidget mapWidget = new MapWidget(String.valueOf(width), String.valueOf(height), defaultMapOptions);
49
50 //Create some Google Layers
51 GoogleV3Options gHybridOptions = new GoogleV3Options();
52 gHybridOptions.setIsBaseLayer(true);
53 gHybridOptions.setType(GoogleV3MapType.G_HYBRID_MAP);
54 GoogleV3 gHybrid = new GoogleV3("Google Hybrid", gHybridOptions);
55
56 GoogleV3Options gNormalOptions = new GoogleV3Options();
57 gNormalOptions.setIsBaseLayer(true);
58 gNormalOptions.setType(GoogleV3MapType.G_NORMAL_MAP);
59 GoogleV3 gNormal = new GoogleV3("Google Normal", gNormalOptions);
60
61 GoogleV3Options gSatelliteOptions = new GoogleV3Options();
62 gSatelliteOptions.setIsBaseLayer(true);
63 gSatelliteOptions.setType(GoogleV3MapType.G_SATELLITE_MAP);
64 GoogleV3 gSatellite = new GoogleV3("Google Satellite", gSatelliteOptions);
65
66 GoogleV3Options gTerrainOptions = new GoogleV3Options();
67 gTerrainOptions.setIsBaseLayer(true);
68 gTerrainOptions.setType(GoogleV3MapType.G_TERRAIN_MAP);
69 GoogleV3 gTerrain = new GoogleV3("Google Terrain", gTerrainOptions);
70
71 ///////////////////////////////////////////////////
72
73 XYZOptions xyzptions_osm = new XYZOptions();

```

Gráfico 67. Código de mapa por defecto.
Elaborado por: El Investigador

46. Después de compilar el código anterior el mapa presentará la siguiente vista.

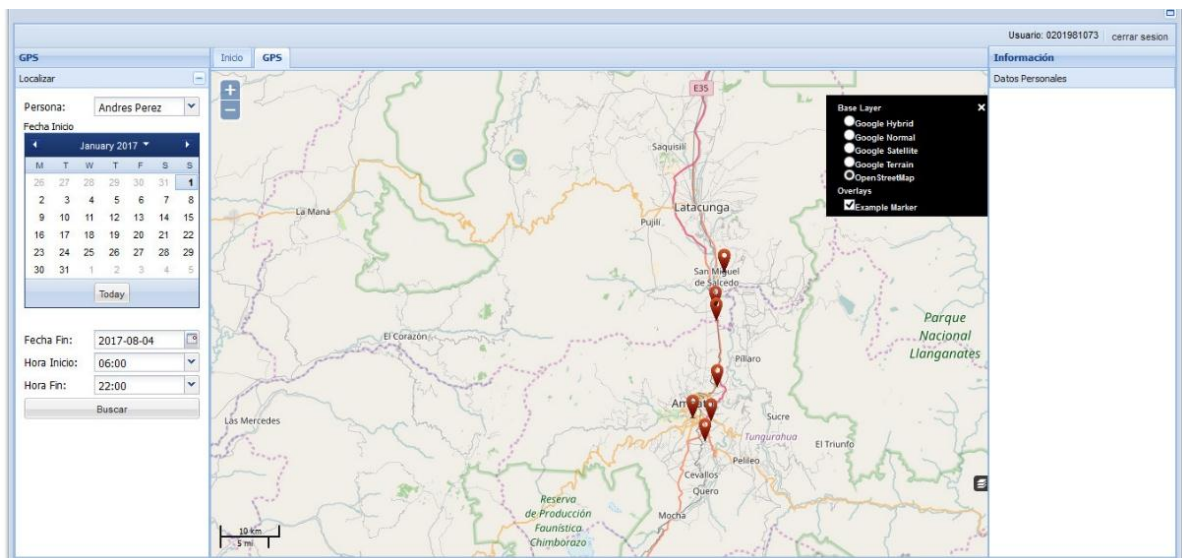


Gráfico 68. Vista del mapa por defecto.
Elaborado por: El Investigador

47. El siguiente código permite especificar el mapa por defecto que se visualizará.

```
XYZOptions xyzptions_osm = new XYZOptions();
xyzptions_osm.setIsBaseLayer(false);
xyzptions_osm.setProjection("EPSG:900913");
xyzptions_osm.setNumZoomLevels(22);
xyzptions_osm.setIsBaseLayer(true);
String url = "https://b.tile.openstreetmap.org/{z}/{y}/{x}.png"; // Parametros.get_url_osm();
XYZ osm = new XYZ("OpenStreetMap", url, xyzptions_osm);
```

Gráfico 69. Código del mapa por defecto.
Elaborado por: El Investigador

Objeto para el paso de parámetros

Dentro de la capa **shared** se puede crear una nueva clase que será compartida dentro del proyecto, esta clase servirá como contenedor general de todas las columnas de las tablas en la base de datos.

Interfaz Serializable.- Para que un programa java pueda convertir un objeto en un montón de bytes y pueda luego recuperarlo, el objeto necesita ser Serializable. Al poder convertir el objeto a bytes, ese objeto se puede enviar a través de red, guardarlo en un fichero, y después reconstruirlo al otro lado de la red.

Para que un objeto sea serializable basta con que implemente la interfaz `Serializable`, como dicha interfaz no tiene métodos, es muy sencillo implementarla, basta con un **“implements Serializable”**.

48. Creación de la clase para el paso de parámetros

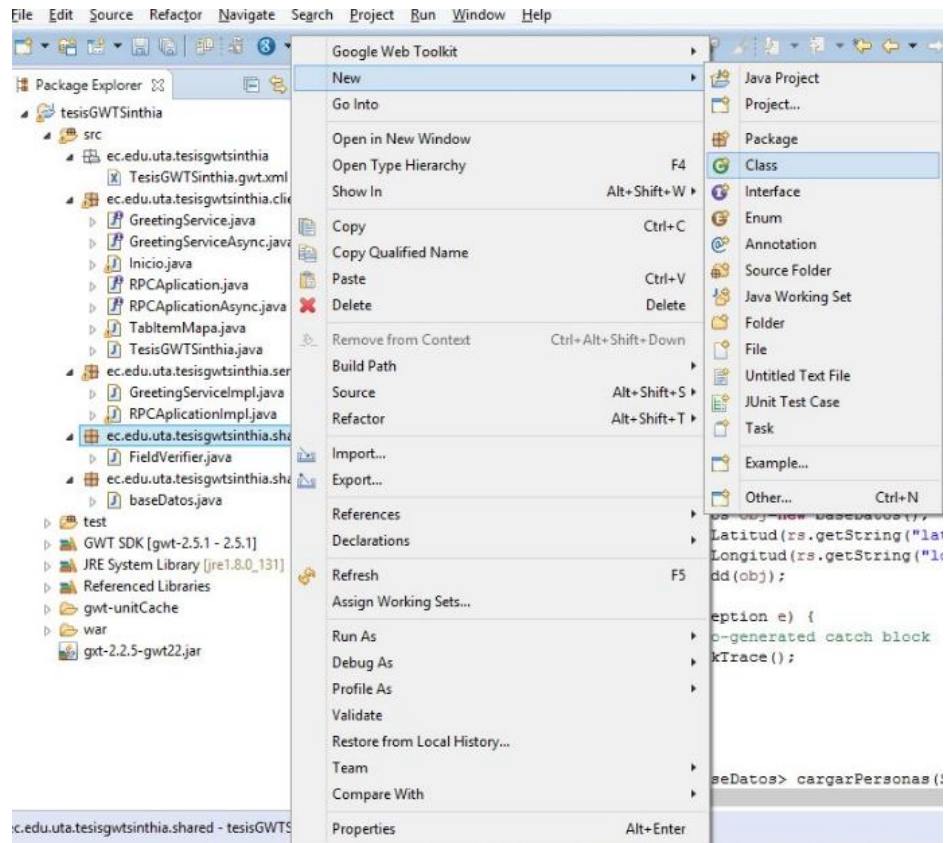


Gráfico 70. Creación de la clase para el paso de parámetros.
Elaborado por: El Investigador

49. Llena los datos de la clase mediante la cual se va a realizar el paso de parámetros.

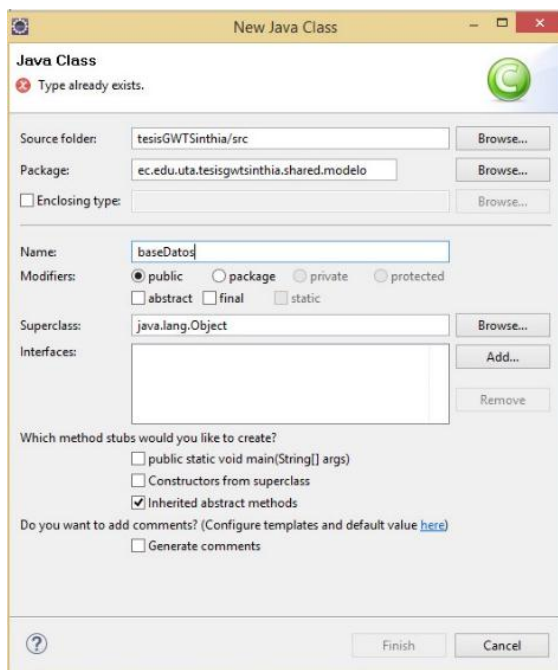


Gráfico 71. Ingreso de datos de la clase.
Elaborado por: El Investigador

50. Es importante crear un ID para el serializable, debido a que esta propiedad permite enviar los valores a través de la red, necesita un ID para poder identificarlo.

```

TesisGWTSinthia... RPCApplication.j... Inicio.java WinInicio.java *baseDatos.java
1 package ec.edu.uta.tesisgwtsinthia.shared.modelo;
2
3 import java.io.Serializable;
4
5 import com.extjs.gxt.ui.client.data.BeanModelTag;
6
7 public class baseDatos implements Serializable, BeanModelTag{
8
9     /**
10    *
11    */
12    private static final long serialVersionUID = 1L;
13
14
15
16 }

```

Gráfico 72. Código de ID serializable.
Elaborado por: El Investigador

51. Solución de advertencia, añade un ID para que el cliente y el servidor reconozca la clase que se está compartiendo.

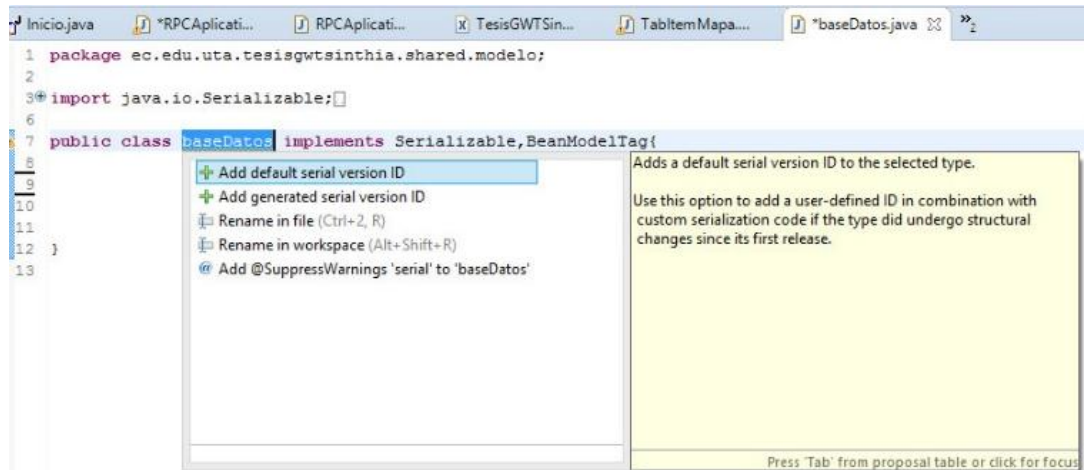


Gráfico 73. Solución de Advertencia.
Elaborado por: El Investigador

Se puede crear el código de las clases directamente, especificando las tablas de la base de datos, pero también se lo puede hacer manualmente; teniendo en cuenta los criterios y estándares de programación. Es necesario especificar las variables en privado, y el tipo de valor que dependerá de las columnas de las tablas.

```

//tabla coordenadas
private String latitud;
private String longitud;
private String fecha;
private String hora;

//tabla personas
private String ced_per;
private String nom_per;
private String ape_per;
private String dir_per;
private String nom_cmp_per;
private String dt;

```

Gráfico 74. Declaración de las variables de la BD.
Elaborado por: El Investigador

52. Es necesario que las variables tengan el mismo nombre de las columnas en las tablas. Para crear los métodos y funciones para manipular las variables dentro de la clase, se lo realiza de la siguiente manera:

- Seleccionar las variables
- Hacer clic derecho, para acceder a las opciones de las variables

- Seleccionar **Source** y **Generate Getters and Setters**.

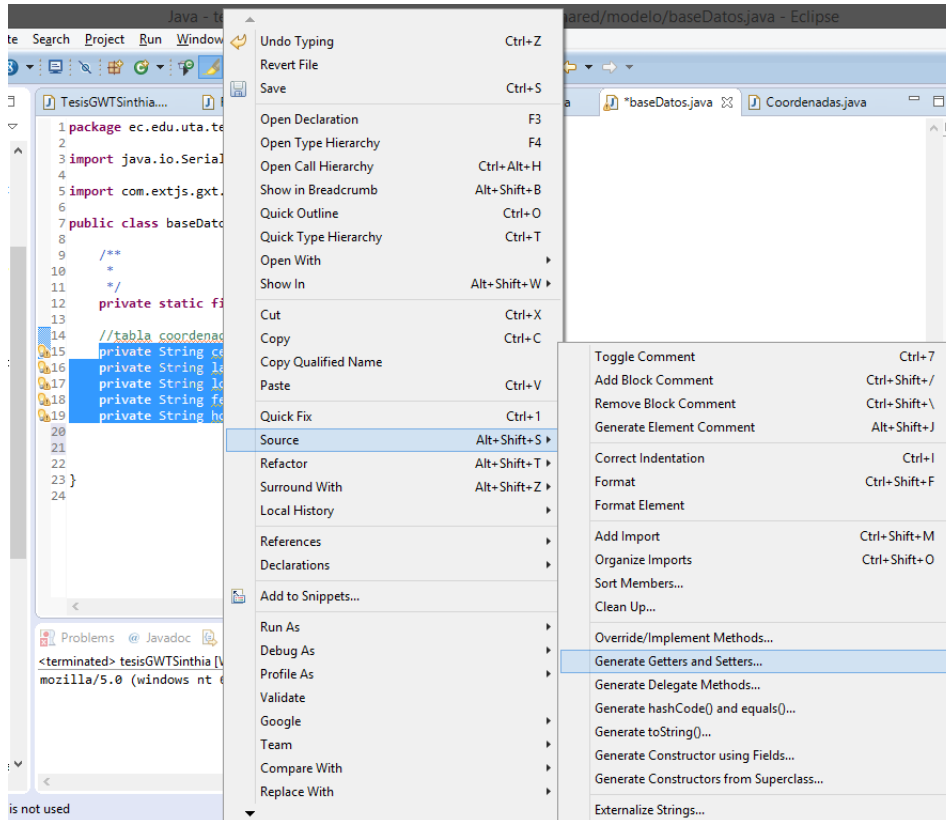


Gráfico 75. Métodos y funciones para manipular las variables.
Elaborado por: El Investigador

Declarar variables del mismo tipo que las columnas de las tablas de la Base de Datos que van a ser utilizadas.

53. Al seleccionar **Generate Getters and Setters...**, aparecerá la siguiente ventana que muestra las variables para las que se va a generar el método getters y setters.

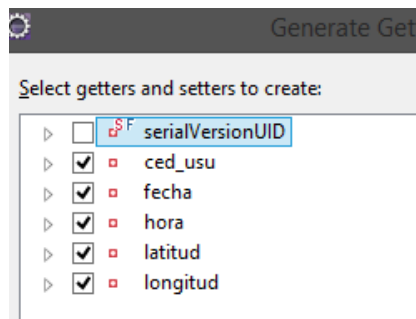
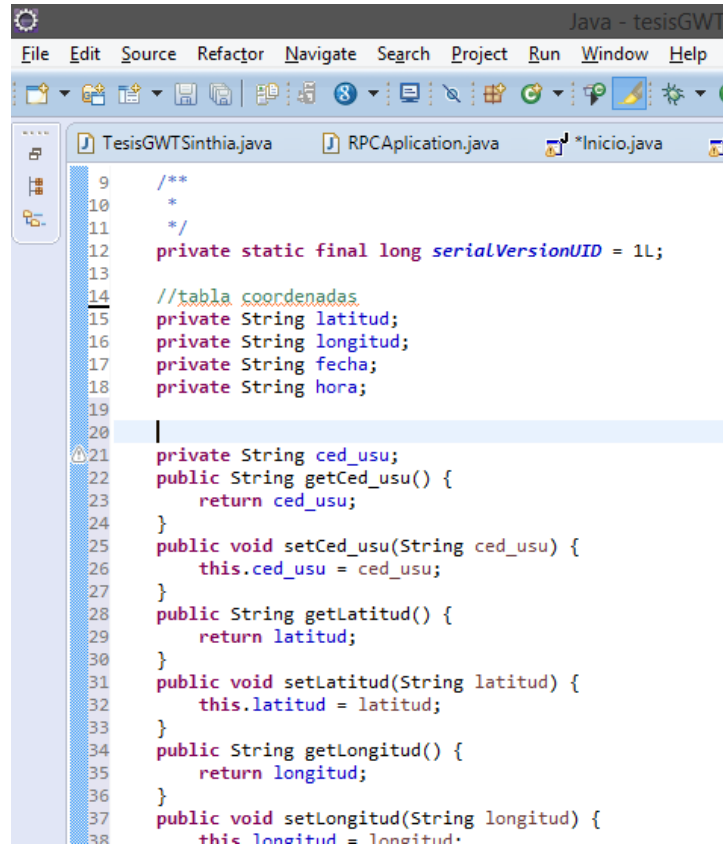


Gráfico 76. Generate Getters and Setters.
Elaborado por: El Investigador

54. Seleccionar las variables que se va a utilizar y dar clic en **OK** para generar los métodos de las variables.

55. Los métodos getters and setters se generan como se presenta a continuación:



```
9  /**
10  *
11  */
12  private static final long serialVersionUID = 1L;
13
14  //tabla coordenadas
15  private String latitud;
16  private String longitud;
17  private String fecha;
18  private String hora;
19
20  |
21  private String ced_usu;
22  public String getCed_usu() {
23      return ced_usu;
24  }
25  public void setCed_usu(String ced_usu) {
26      this.ced_usu = ced_usu;
27  }
28  public String getLatitud() {
29      return latitud;
30  }
31  public void setLatitud(String latitud) {
32      this.latitud = latitud;
33  }
34  public String getLongitud() {
35      return longitud;
36  }
37  public void setLongitud(String longitud) {
38      this.longitud = longitud;
```

Gráfico 77. Método Getters and Setters.
Elaborado por: El Investigador

Añadir Marcadores

56. La programación se la realiza en el servidor, en la clase RPCApplicationImpl

```

65     }
66     return result;
67 }
68
69 public ArrayList<baseDatos> getCoordenadas(String usuario,String fecha_inicio,String fecha_fin,String hora_inicio,String hora_fin)
70 {
71     ArrayList<baseDatos> lista=new ArrayList<baseDatos>();
72
73     Connection cn=conectar();
74     String sql="select latitud, longitud from coordenadas "
75     + "where ced_usu = '"+usuario+"' "
76     + "and fecha between '"+fecha_inicio+"' and '"+fecha_fin+"' "
77     + "and hora between '"+ hora_inicio+"' and '"+hora_fin+"' "
78     + "order by fecha, hora";
79
80     try {
81         Statement psd = cn.createStatement();
82         ResultSet rs=psd.executeQuery(sql);
83         while(rs.next())
84         {
85             baseDatos obj=new baseDatos();
86             obj.setLatitud(rs.getString("latitud"));
87             obj.setLongitud(rs.getString("longitud"));
88             lista.add(obj);
89         }
90     } catch (SQLException e) {
91         // TODO Auto-generated catch block
92         e.printStackTrace();
93     }
94     return lista;
95 }
96 }

```

Consulta de coordenadas

Gráfico 78. Consulta de coordenadas.
Elaborado por: El Investigador

Consulta de coordenadas

Se va a realizar la consulta de coordenadas filtrando: usuario, fecha inicio, fecha fin, hora inicio, hora fin.

While: Mediante el while se recorre el result set, cada recorrido extrae latitud y longitud, y guarda en un objeto de la base de datos, devolviendo una lista con los objetos.

```

try {
    Statement psd = cn.createStatement();
    ResultSet rs=psd.executeQuery(sql);
    while(rs.next())
    {
        baseDatos obj=new baseDatos();
        obj.setLatitud(rs.getString("latitud"));
        obj.setLongitud(rs.getString("longitud"));
        lista.add(obj);
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
return lista;
}

```

Gráfico 79. Código while.
Elaborado por: El Investigador

57. En el RPCApplication de la parte del cliente, se copia el método de la clase RPCApplicationImpl Server.

```

public static class Util {
    private static RPCApplicationAsync instance;
    public static RPCApplicationAsync getInstance() {
        if (instance == null) {
            instance = GWT.create(RPCApplication.class);
        }
        return instance;
    }
}

}

public PagingLoadResult<baseDatos> obtenerDatosPersonas(String usuario) throws Exception;
public ArrayList<baseDatos> getCoordenadas(String usuario,String fecha_inicio,String fecha_fin,String hora_inicio,Stri
public boolean Login(String Usuario, String Password);

```

Gráfico 80. Método de la clase RPCApplication Server.
Elaborado por: El Investigador

58. Al copiar el método se presenta la siguiente advertencia, la misma que presenta la solución mediante la herramienta, que permitirá que se cree el método automáticamente en la clase asincrónica.

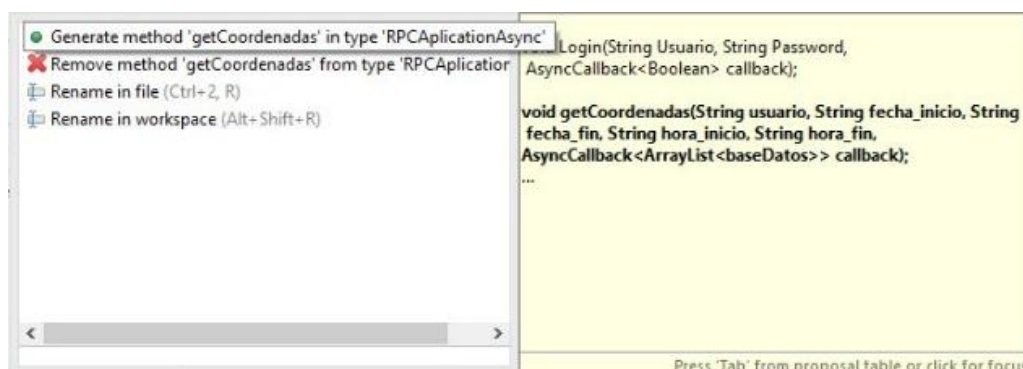


Gráfico 81. Advertencia: creación de métodos en la clase asincrónica
Elaborado por: El Investigador

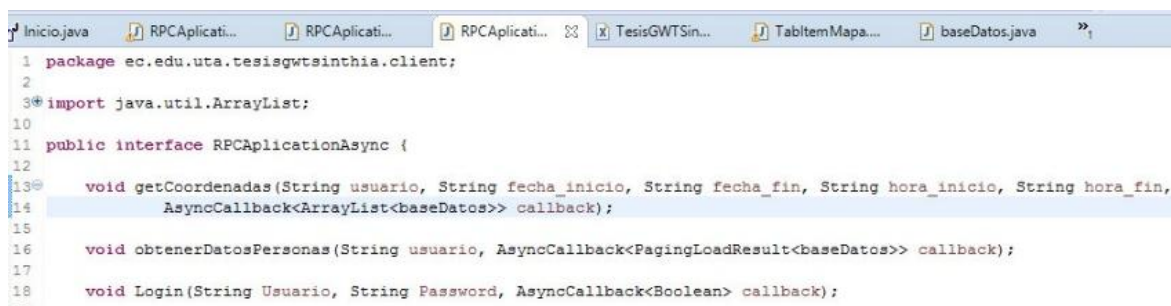


Gráfico 82. Creación de métodos en la clase asincrónica.
Elaborado por: El Investigador

59. En la clase **TabItemMapa** del cliente se programa la gráfica del marcador en el mapa, mediante el siguiente método:

```

public void onSuccess(ArrayList<baseDatos> result) {
    // TODO Auto-generated method stub
    layer.clearMarkers();
    for(baseDatos c:result)
    {
        LonLat lonLat = new LonLat(Double.valueOf(c.getLongitud()), Double.valueOf(c.getLatitud()));
        lonLat.transform("EPSG:4326", "EPSG:900913");
        // Icon icon = new Icon("http://icongal.com/gallery/image/99728/map_pin_location_push_pin_gps_pushpin.png",
        Icon icon = new Icon("https://cdn0.iconfinder.com/data/icons/locationicons/pin.png",
            new Size(32, 32));
        final Marker marker = new Marker(lonLat, icon);
        layer.addMarker(marker);

        //colocar mensale: no esta en funcionamiento
        marker.addBrowserEventListener(EventType.MAP_CLICK, new MarkerBrowserEventListener() {

```

Gráfico 83. Método para graficar los marcadores en el mapa.
Elaborado por: El Investigador

Llamado TabItem a TabPanel

60. El llamado se programa en la clase **Inicio** del cliente, para visualizar el mapa.

```

try{
    mapa=new TabItemMapa (height,width);
    //com.google.gwt.user.client.Window.alert ("mapa instanciado");
    mapa.setText ("GPS");
}catch(Exception ex){
    com.google.gwt.user.client.Window.alert(":::"+ex);
}

if(tabPanel.getItemCount ()>1){

if(!tabPanel.getItem(1).getText().equals ("GPS")){

//mapa.setClosable(true);
tabPanel.add(mapa);
tabPanel.setSelection(tabPanel.getItem(1));

```

Gráfico 84. Visualización del mapa.
Elaborado por: El Investigador

61. El siguiente código permite recorrer el tamaño del TabPanel y grafica el mapa.

```

int height=tabPanel.getHeight()-25;
int width=tabPanel.getWidth();
habilitarMapaGps (height,width);

```

Gráfico 85. Código para graficar el mapa.
Elaborado por: El Investigador

Llamado del método que grafica los marcadores en el mapa

62. Para realizar el llamado del método para graficar los marcadores el siguiente código va a permitir extraer datos y guardar en variables, para mostrar las diferentes ubicaciones dentro del mapa.

```
btnBuscar = new Button("Buscar");
btnBuscar.addSelectionListener(new SelectionListener<ButtonEvent>() {
    public void componentSelected(ButtonEvent ce) {

        fecha_inicio=DateTimeFormat.getShortDateFormat().format(dpFechaInicio.getValue());
        fecha_fin=DateTimeFormat.getShortDateFormat().format(dtfldFechaFin.getValue());
        String[] tiempoI=DateTimeFormat.getLongTimeFormat().format(tmfldHoraInicio.getDateValue()).split(" ");
        String[] tiempoF=DateTimeFormat.getLongTimeFormat().format(tmfldHoraFin.getDateValue()).split(" ");
        hora_inicio=tiempoI[0];
        hora_fin=tiempoF[0];
        String usuario=cmbxPersona.getValue().get("ced_per");

        //com.google.gwt.user.client.Window.alert(cmbxPersona.getValue().get("ced_per")+":"+fecha_inicio":"+fecha_fin+":"+hora_inicio+":"+hora_fin);
        mapa.coordenadas(usuario,fecha_inicio,fecha_fin,hora_inicio,hora_fin);
    }
});
```

Gráfico 86. Código del botón buscar.
Elaborado por: El Investigador

Llenar un combo box

63. El siguiente código permitirá llenar un combo box de personas que se va a utilizar para la búsqueda de ubicación dentro del mapa.

```
public ArrayList<baseDatos> cargarPersonas(String usuario){
    ArrayList<baseDatos> lista=new ArrayList<baseDatos>();
    Connection cn=conectar();
    String sql="select ced_per,concat(nom_per,' ',ape_per) nom_cmp_per from personas "
        + " where ced_usu_p = '"+usuario+"' ";
    try {
        Statement st=cn.createStatement();
        ResultSet rs=st.executeQuery(sql);
        while(rs.next())
        {
            baseDatos obj=new baseDatos();
            obj.setNom_cmp_per(rs.getString("nom_cmp_per"));
            obj.setCed_per(rs.getString("ced_per"));
            lista.add(obj);
        }
        cn.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        System.out.println("No se pudo cargar los datos personas");
    }

    return lista;
}
```

Gráfico 87. Código para crear una lista.
Elaborado por: El Investigador

64. Para hacer uso de la lista creada en el paso anterior, primero guarda los datos en un modelo de datos denominado **BasePagingLoadResult**, que permitirá manipularlos en la vista.

```
public PagingLoadResult<baseDatos> obtenerDatosPersonas(String usuario) throws Exception
{
    ArrayList<baseDatos> lista=cargarPersonas(usuario);
    BasePagingLoadResult<baseDatos> retorno=new BasePagingLoadResult<baseDatos>(lista,0,0);
    return retorno;
}
```

Gráfico 88. Uso de la lista creada anteriormente.
Elaborado por: El Investigador

65. El método para obtener los datos de una persona en un modelo de datos, se copia en la clase **RPCApplication** del cliente, en donde se genera un error indicando que no existe el método en la clase asincrónica.

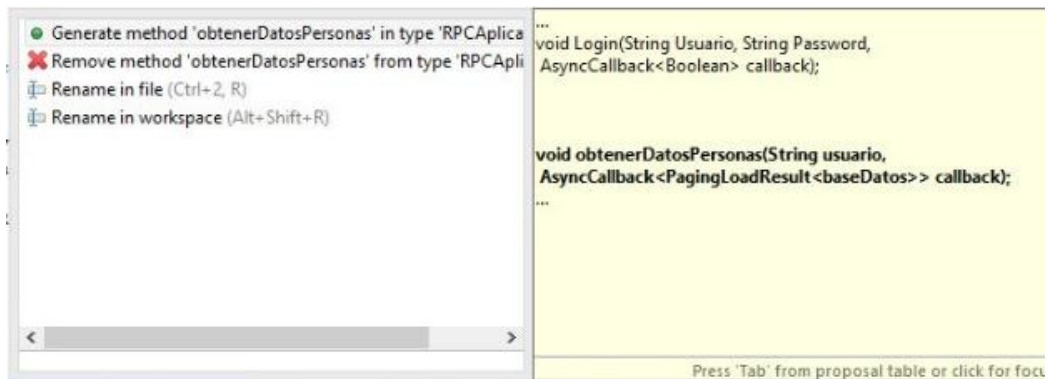


Gráfico 89. Error de método – clase asincrónica.
Elaborado por: El Investigador

66. Al seleccionar la solución presentada por la herramienta, en la clase asincrónica se crea el método que permitirá llenar el combo box de personas.

```
package ec.edu.uta.tesisgwtsinthia.client;
import java.util.ArrayList;

public interface RPCApplicationAsync {
    void getCoordenadas(String usuario, String fecha_inicio, String fecha_fin, String hora_inicio, String hora_fin, AsyncCallback<ArrayList<baseDatos>> callback);
    void obtenerDatosPersonas(String usuario, AsyncCallback<PagingLoadResult<baseDatos>> callback);
    void Login(String Usuario, String Password, AsyncCallback<Boolean> callback);
}
```

Gráfico 90. Método llenar combo box personas.
Elaborado por: El Investigador

67. Para llenar los datos de una persona, se obtiene los datos de

```
public void llenarPersonas(String usuario){
    RPCApplication.Util.getInstance().obtenerDatosPersonas(usuario, new AsyncCallback<PagingLoadResult<baseDatos>>() {

        @Override
        public void onFailure(Throwable caught) {

            // TODO Auto-generated method stub

        }

        @Override
        public void onSuccess(PagingLoadResult<baseDatos> result) {
            // TODO Auto-generated method stub
            listaPersonas.removeAll();
            for(baseDatos obj:result.getData()){
                BeanModelFactory factory=BeanModelLookup.get().getFactory(baseDatos.class);
                BeanModel model=factory.createModel(obj);
                listaPersonas.add((ModelData)model);
            }
        }
    });
}
```

PagingLoadResult, y se crea una lista de personas.

*Gráfico 91. Método llenar personas.
Elaborado por: El Investigador*

68. En el constructor de la clase **Inicio**, se añade la lista al combo, mediante el siguiente código:

```
cmbxPersona.setValueField("ced_per");
cmbxPersona.setDisplayField("nom_cmp_per");
cmbxPersona.setStore(listaPersonas);
```

*Gráfico 92. Añade la lista al combo box.
Elaborado por: El Investigador*

69. En la interacción con el combo box, se selecciona un elemento, y no se borran los demás elementos.

```
//lineas necesarias para interactuar con el comboBox
cmbxPersona.setTypeAhead(true);
cmbxPersona.setTriggerAction(TriggerAction.ALL);
```

*Gráfico 93. Interacción con el combo box.
Elaborado por: El Investigador*

Se hace el llamado al método “llenar personas” cuando se obtiene acceso al sistema.

Ingreso de un nuevo usuario

70. Para el ingreso de un nuevo usuario se ha creado una clase que extiende de **TabItem** para poder abrirla dentro del **TabPanel**.

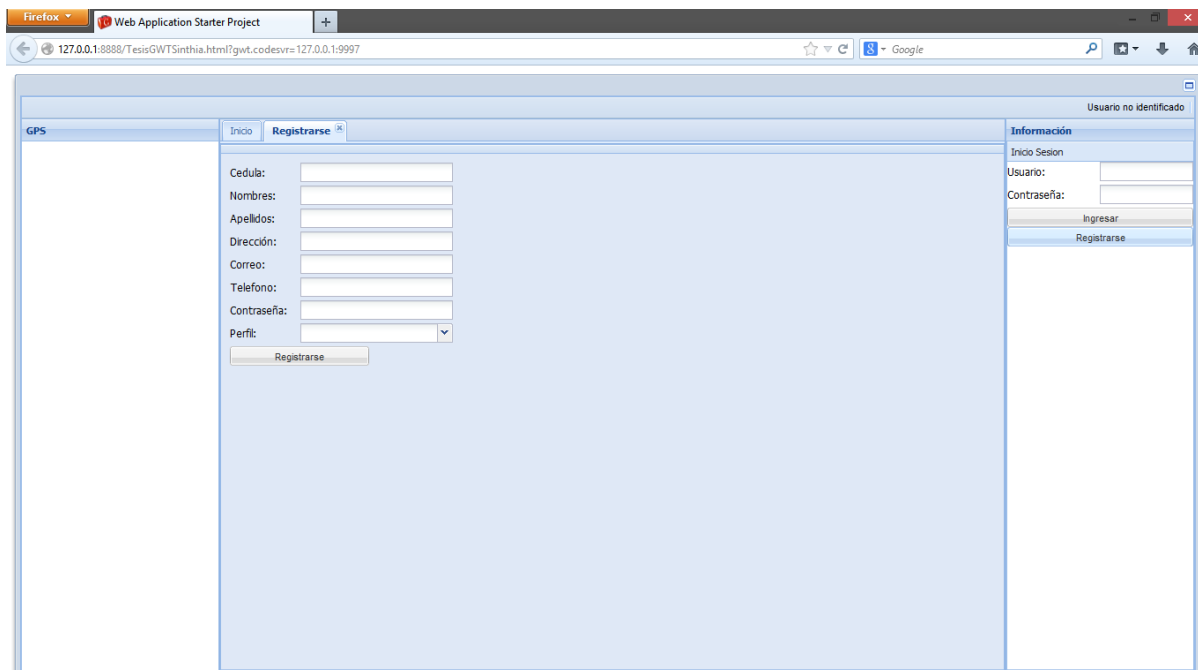


Gráfico 94. Ingreso de un nuevo usuario.
Elaborado por: El Investigador

71. En esta clase se encuentra un **FormPanel** que contiene:
- TextFields** para los campos: Nombres, Apellidos, Dirección y Contraseña.
 - NumberFields** para los campos: Cédula y Teléfono, el cual permite ingresar solo números.
 - Button:** para ejecutar el evento **insertarDatos**.
 - TextFieldCorreo:** es exclusivo para el control del correo electrónico.

```
public class IngresoUsuarios extends TabItem {
    private TextField txtfldDireccin;
    private TextField txtfldApellidos;
    private TextField txtfldNombres;
    private ComboBox cmbxPerfil;
    private TextField txtfldContrasea;
    private Button btnRegistrarse;
    private ListStore<ModelData> listaPerfil;
    private FormPanel formPanel;
    private NumberField nmbfldCedula;
    private NumberField nmbfldTelefono;
    private TextFieldCorreo txtfldcrCorreo;
}
```

Gráfico 95. Código para controlar el ingreso de los datos de usuarios.
Elaborado por: El Investigador

72. Para seleccionar el perfil de usuario mediante un combo box, esta pre-cargado los perfiles **Institucional** y **Personal**.
- El perfil **Institucional**, permite el ingreso de varias personas para el control del gps; y el **Personal**, es solo para el ingreso de la persona que lo va a utilizar.

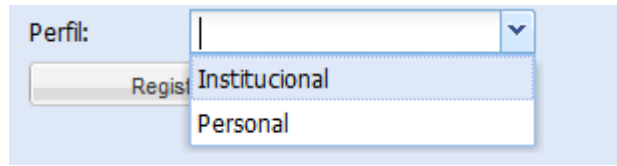


Gráfico 96. Perfiles de ingreso al sistema.
Elaborado por: El Investigador

Creación de un componente modificado

73. Para crear un nuevo componente se debe crear una nueva clase y heredar del componente que se desea modificar las propiedades.

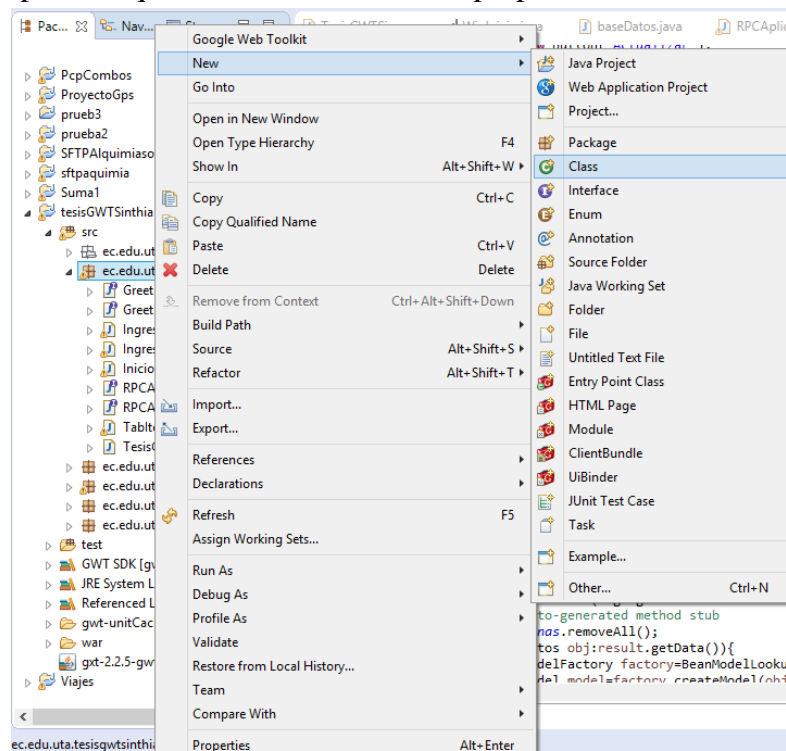


Gráfico 97. Creación de una clase nueva, para el componente.
Elaborado por: El Investigador

74. En la siguiente imagen se muestra como se hereda la clase del componente **TextField**, basta con dar clic en Browse y buscar el componente que se desea heredar.

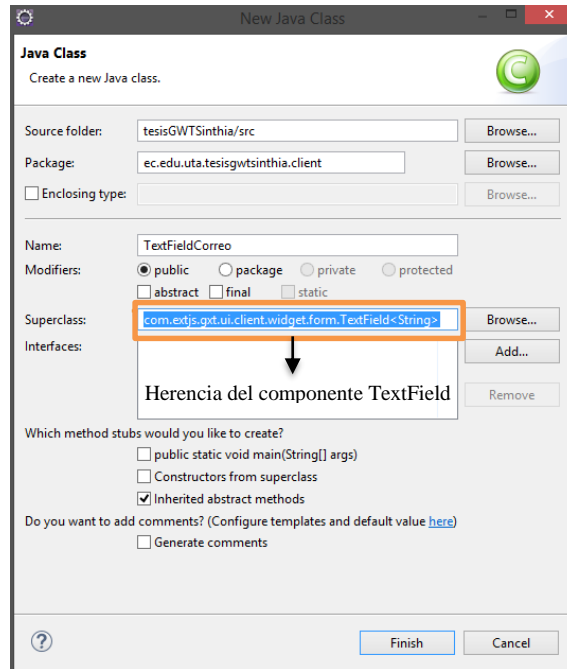


Gráfico 98. Creación de una clase con herencia.
Elaborado por: El Investigador

75. Una vez creada la clase, es necesario crear un constructor y allí se puede programar los eventos que posee el componente de acuerdo a los requerimientos.

```
public class TextFieldCorreo extends TextField<String> {

    public TextFieldCorreo(){
        addListener(Events.Blur, new Listener<FieldEvent>() {
            public void handleEvent(FieldEvent e) {
                try{
                    String EMAIL_PATTERN =
                        "^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
                        + "[A-Za-z0-9-]+(\\.+[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$";

                    if(getValue()!=null){
                        if(!getValue().toString().matches(EMAIL_PATTERN)) //el metodo matches permite validar la cadena de caracteres
                        {
                            com.google.gwt.user.client.Window.alert("Correo Invalido : estructura de ejemplo ejemplo@dominio.com");
                            focus();
                            setValue("");
                        }
                    }
                }catch(Exception ex){
                }
            }
        });
    }
}
```

Gráfico 99. Creación de una clase con herencia.
Elaborado por: El Investigador

En este caso se ha modificado el evento **Blur**, el mismo que se ejecuta cuando el componente pierde el foco.

La cadena **EMAIL_PATTERN** contiene los caracteres que debe cumplir una dirección de correo electrónico y con el método **matches** se puede hacer la

validación de otra cadena con la del email, si la cadena no tiene estructura de correo, el componente muestra el siguiente mensaje de error.

Correo Invalido : estructura de ejemplo example@dominio.com

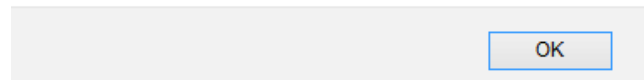


Gráfico 100. Mensaje de error del correo electrónico.
Elaborado por: El Investigador

Para poder usar un nuevo componente es necesario añadirlo a la parte de componentes.

76. Para añadir un componente se debe seleccionar la opción **chooser** que se encuentra al inicio de la paleta, el cual muestra una ventana de búsqueda donde se puede buscar el componente creado y usarlo.

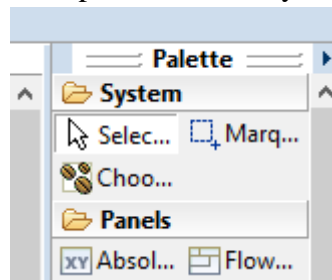


Gráfico 101. Elementos de la paleta.
Elaborado por: El Investigador

77. Para añadir el componente se debe registrar el nombre en la siguiente ventana, y dar clic en **OK**.

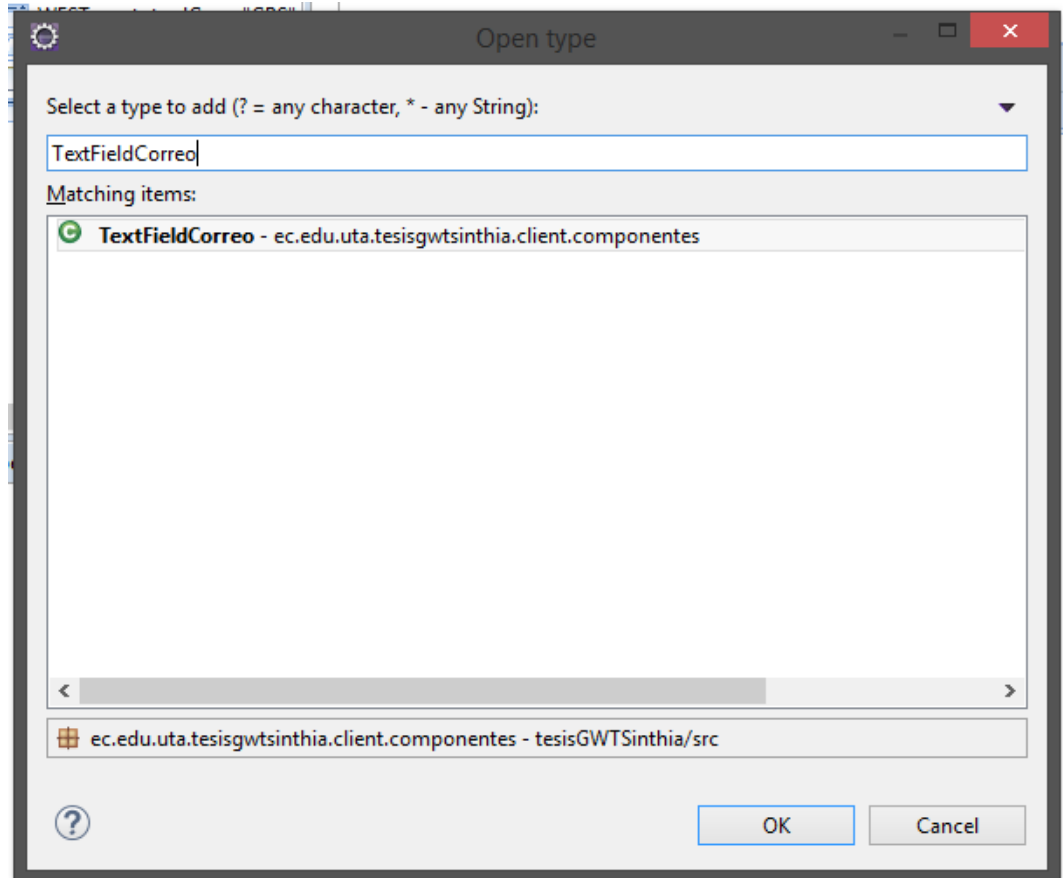


Gráfico 102. Datos del componente.
Elaborado por: El Investigador

78. Se crea automáticamente una carpeta **Custom** donde se adjuntan los componentes, una vez que el componente se encuentra en la paleta, se puede hacer uso de él tal cual como si fuera otro componente de gxt.

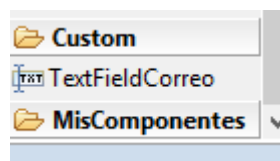


Gráfico 103. Carpeta Custom – componente propio.
Elaborado por: El Investigador

La ventaja de crear un componente propio es que se puede hacer controles, como validar el correo y con esto el componente puede ser usado en varios formularios.

Método para insertar datos de usuarios

79. El siguiente código permite insertar los datos de nuevos usuarios en el sistema.

```
public String insertarDatosUsuarios(baseDatos objUsuario)
{
    String mensaje="";
    try {
        String sql="insert into usuarios "
            + " values('"+objUsuario.getCed_usu()+"', '"+objUsuario.getNom_usu()+"', '"+objUsuario.getApe_usu()+"', "
            + "'"+objUsuario.getMail_usu()+"', '"+objUsuario.getDir_usu()+"', '"+objUsuario.getTlf_usu()+"', "
            + objUsuario.getUsu_password()+"', '"+objUsuario.getPerf_usu()+"')";

        Connection cn=conectar();
        PreparedStatement psd=cn.prepareStatement(sql);
        int n=psd.executeUpdate();
        if(n>0){
            mensaje="Datos Insertados";
        }else{
            mensaje="error... No se insertaron los datos en la tabla Usuarios";
        }
        cn.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        mensaje=e.getMessage();
    }
    return mensaje;
}
```

Gráfico 104. Código para insertar usuarios.
Elaborado por: El Investigador

80. La herramienta presenta la siguiente solución, generar el método en la clase asíncronica.

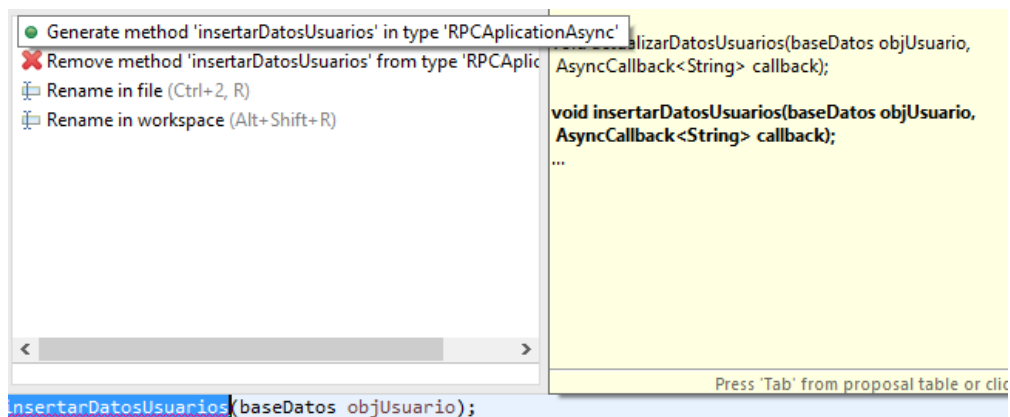


Gráfico 105. Solución de error, creación del método.
Elaborado por: El Investigador

```

package ec.edu.uta.tesisgwtsinthia.client;

import java.util.ArrayList;

import com.extjs.gxt.ui.client.data.PagingLoadResult;
import com.google.gwt.user.client.rpc.AsyncCallback;

import ec.edu.uta.tesisgwtsinthia.shared.modelo.baseDatos;

public interface RPCApplicationAsync {

    void getCoordenadas(String usuario, String fecha_inicio, String fecha_fin, String hora_inicio, String hora_fin,
        AsyncCallback<ArrayList<baseDatos>> callback);

    void obtenerDatosPersonas(String usuario, AsyncCallback<PagingLoadResult<baseDatos>> callback);

    void Login(String Usuario, String Password, AsyncCallback<Boolean> callback);

    void cargarComboPerfil(AsyncCallback<PagingLoadResult<baseDatos>> callback);

    void getCoordenadasFecha(String usuario, String fecha_inicio, String fecha_fin,
        AsyncCallback<ArrayList<baseDatos>> callback);

    void insertarDatosUsuarios(baseDatos objUsuario, AsyncCallback<String> callback);
}

```

Gráfico 106. Generación del método en la clase asincrónica.
Elaborado por: El Investigador

81. Para el manejo del perfil Personal se ha hecho uso de un trigger que se encarga de duplicar los datos que se ingresan la tabla Usuarios con este perfil en la tabla Personas, esto se lo hace para mantener la relación entre tablas.

```

CREATE OR REPLACE FUNCTION insertar_datos() RETURNS TRIGGER AS $insertar_datos$
DECLARE
BEGIN

    IF NEW.perf_usu = 'Personal' THEN
        insert into personas
            values (NEW.ced_usu,NEW.nom_usu,NEW.ape_usu,NEW.mail_usu,NEW.dir_usu,NEW.tlf_usu,NEW.ced_usu);

    END IF;
    RETURN NULL;
END;
$insertar_datos$ LANGUAGE plpgsql;

CREATE TRIGGER insertar_datos AFTER INSERT
ON usuarios FOR EACH ROW
EXECUTE PROCEDURE insertar_datos();

```

Gráfico 107.. Trigger para la inserción de datos en la tabla personas.
Elaborado por: El Investigador

Insertar datos personas

82. Al momento que se ingresa al sistema, se habilita un botón para el registro de personas, este botón se hace visible únicamente para los usuarios con perfil Institucional.

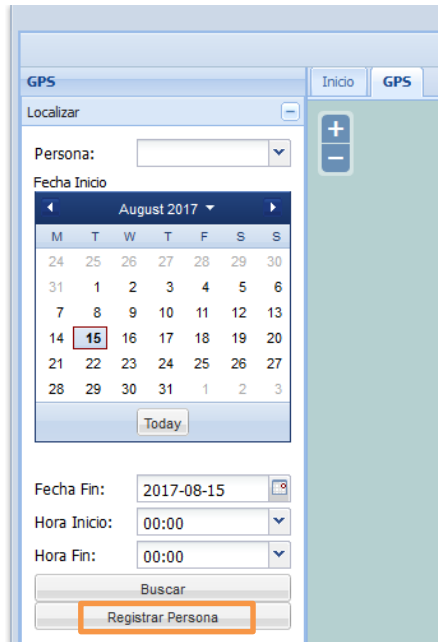


Gráfico 108. Vista del sistema – Perfil Institucional.
Elaborado por: El Investigador

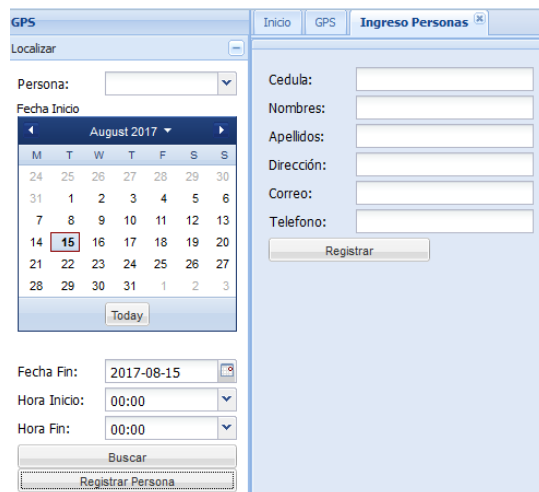


Gráfico 109. Ingreso de personas.
Elaborado por: El Investigador

83. Se crea una clase **TabItem** para el ingreso de una nueva persona, como se muestra a continuación:

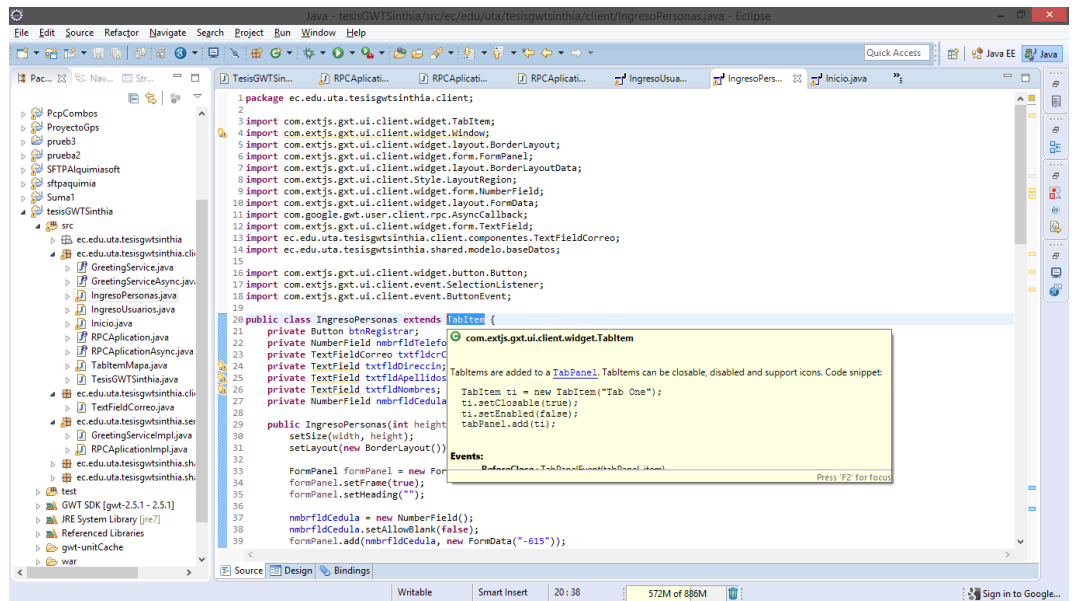


Gráfico 110. Código para el ingreso de personas.
Elaborado por: El Investigador

En esta clase se encuentra un fromPanel que contiene:

TextField: para los campos nombres, apellidos, dirección.

NumberField: para los campos cédula y teléfono.

TextFieldCorreo: componente modificado para validar correos electrónicos.

Button: el cual ejecuta el evento **insertarDatos**.

Inserción de datos de personas

84. Método para la inserción de datos en la tabla personas

```
public String insertarDatosPersonas(baseDatos objPersonas)
{
    String mensaje="";
    try {
        String sql="insert into personas "
            + " values('"+objPersonas.getCed_per()+"','"+objPersonas.getNom_per()+"','"+objPersonas.getApe_per()+"',"
            + ""+objPersonas.getMail_per()+"','"+objPersonas.getDir_per()+"','"+objPersonas.getTlf_uper()+"','"+
            +objPersonas.getCed_usu_p()+"')";

        Connection cn=conectar();
        PreparedStatement psd=cn.prepareStatement(sql);
        int n=psd.executeUpdate();
        if(n>0){
            mensaje="Datos Insertados";
        }else{
            mensaje="error... No se insertaron los datos en la tabla Personas";
        }
        cn.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        mensaje=e.getMessage();
    }
    return mensaje;
}
```

Gráfico 111. Código para la inserción de datos de personas.
Elaborado por: El Investigador

85. Al programar el método de inserción de datos, en la clase RPCApplication, se genera un error, el mismo que presenta la siguiente solución.

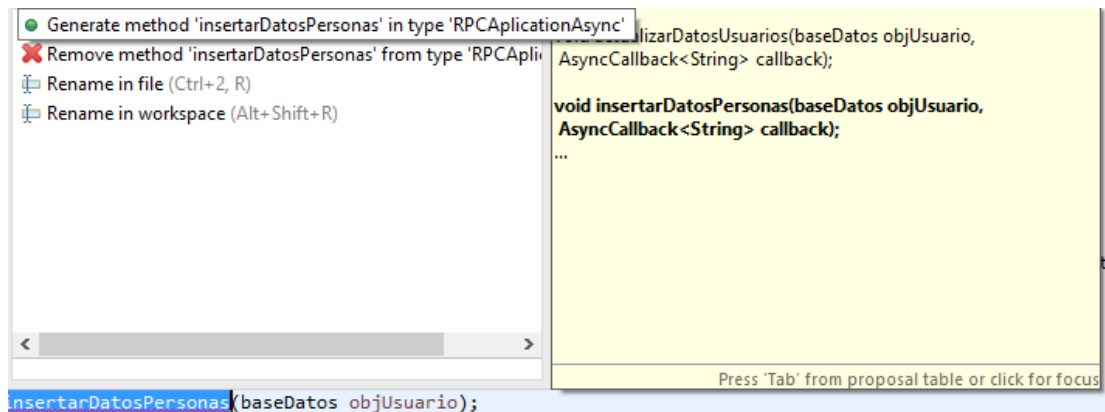


Gráfico 112. Error en la clase RPCApplicationAsync.
Elaborado por: El Investigador

```

TesisGWTSin... RPCAplicati... *RPCAplicati... *RPCAplicati... *IngresoUsu... *TextFieldC... Iniciojava
1 package ec.edu.uta.tesisgwtsinthia.client;
2
3 import java.util.ArrayList;
4
5 import com.extjs.gxt.ui.client.data.PagingLoadResult;
6 import com.google.gwt.user.client.rpc.AsyncCallback;
7
8
9 import ec.edu.uta.tesisgwtsinthia.shared.modelo.baseDatos;
10
11 public interface RPCApplicationAsync {
12
13     void getCoordenadas(String usuario, String fecha_inicio, String fecha_fin, String hora_inicio, String hora_fin,
14         AsyncCallback<ArrayList<baseDatos>> callback);
15
16     void obtenerDatosPersonas(String usuario, AsyncCallback<PagingLoadResult<baseDatos>> callback);
17
18     void Login(String Usuario, String Password, AsyncCallback<Boolean> callback);
19
20     void cargarComboPerfil(AsyncCallback<PagingLoadResult<baseDatos>> callback);
21
22     void getCoordenadasFecha(String usuario, String fecha_inicio, String fecha_fin,
23         AsyncCallback<ArrayList<baseDatos>> callback);
24
25     void insertarDatosUsuarios(baseDatos objUsuario, AsyncCallback<String> callback);
26
27
28     void consultarPerfilPersonal(String cedula, AsyncCallback<Boolean> callback);
29
30     void insertarDatosPersonas(baseDatos objUsuario, AsyncCallback<String> callback);

```

Gráfico 113. Solución del error en la clase RPCApplicationAsync.
Elaborado por: El Investigador

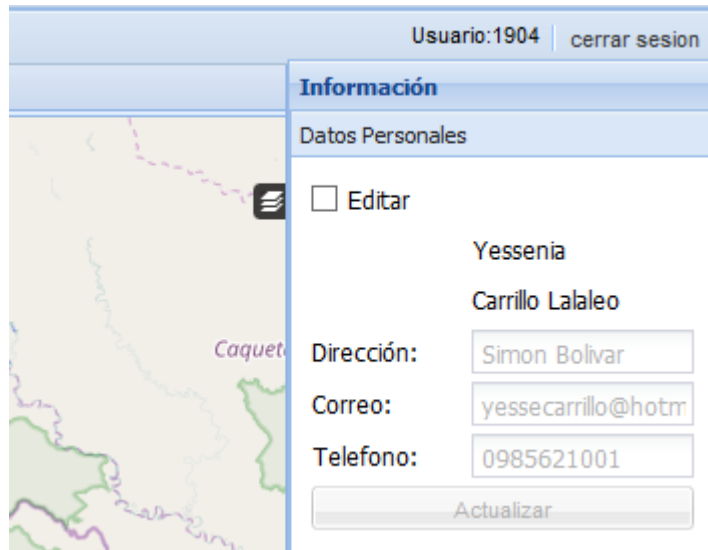
```

private void insertarDatos(baseDatos objPersonas)
{
    RPCApplication.Util.getInstance().insertarDatosPersonas(objPersonas, new AsyncCallback<String>() {
        @Override
        public void onSuccess(String result) {
            // TODO Auto-generated method stub
            com.google.gwt.user.client.Window.alert(result);
        }
        @Override
        public void onFailure(Throwable caught) {
            // TODO Auto-generated method stub
        }
    });
}
}

```

Gráfico 114. Código del método insertarDatos.
Elaborado por: El Investigador

86. Esta es la vista de los datos personales que se ingresan.



The screenshot shows a web interface with a user profile form. At the top right, it says 'Usuario:1904' and 'cerrar sesion'. The form is titled 'Información' and 'Datos Personales'. It features a 'Editar' checkbox, a map on the left, and input fields for 'Dirección' (Simon Bolivar), 'Correo' (yessecarrillo@hotmail.com), and 'Telefono' (0985621001). An 'Actualizar' button is at the bottom.

Gráfico 115. Datos Personales.
Elaborado por: El Investigador

En la parte derecha se encuentra un **FormPanel**, en donde están:

CheckBox con el nombre Editar, el cual habilitará o deshabilitará el resto de componentes.

Labels que muestran los nombres y apellidos del usuario.

TextField para el campo dirección.

TextFieldCorreo componente modificado para validar correos.

NumberField para el campo teléfono.

Button para ejecutar el evento que se encarga de actualizar los datos del usuario.

Método para actualizar los datos del usuario

87. Mediante el siguiente código se actualizan los datos del usuario.

```

public String actualizarDatosUsuarios(baseDatos objUsuario)
{
    String mensaje="";
    try {
        String sql="update usuarios "
            + "set mail_usu='"+objUsuario.getMail_usu()+"', "
            + " dir_usu='"+objUsuario.getDir_usu()+"', "
            + " tlf_usu='"+objUsuario.getTlf_usu()+"' "
            + " where ced_usu = '"+objUsuario.getCed_usu()+"'";
        Connection cn=conectar();
        PreparedStatement psd=cn.prepareStatement(sql);
        int n=psd.executeUpdate();
        if(n>0){
            mensaje="Datos Actualizados";
        }else{
            mensaje="error... No se actualizaron los datos en la tabla Usuarios";
        }
        cn.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        mensaje=e.getMessage();
    }
    return mensaje;
}

```

Gráfico 116. Actualizar datos del usuario.
Elaborado por: El Investigador

88. Al programar el método **actualizarDatosUsuarios**, se genera un error en la clase **RPCApplication**.

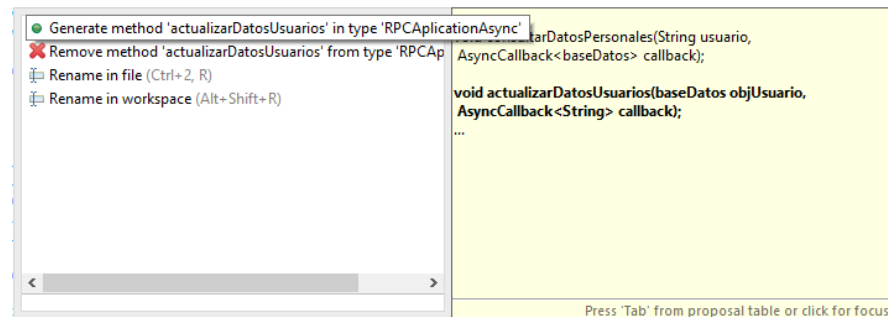


Gráfico 117. Error en la clase **RPCApplicationAsync**.
Elaborado por: El Investigador

89. Al dar clic en la solución que presenta la herramienta, se crea automáticamente el método **actualizarDatosUsuario** en la clase asincrónica **RPCApplicationSync**.

```

1 package ec.edu.uta.tesisgwtsinthia.client;
2
3 import java.util.ArrayList;
4
5 import com.extjs.gxt.ui.client.data.PagingLoadResult;
6 import com.google.gwt.user.client.rpc.AsyncCallback;
7
8
9 import ec.edu.uta.tesisgwtsinthia.shared.modelo.baseDatos;
10
11 public interface RPCApplicationAsync {
12
13     void getCoordenadas(String usuario, String fecha_inicio, String fecha_fin, String hora_inicio, String hora_fin,
14         AsyncCallback<ArrayList<baseDatos>> callback);
15
16     void obtenerDatosPersonas(String usuario, AsyncCallback<PagingLoadResult<baseDatos>> callback);
17
18     void Login(String Usuario, String Password, AsyncCallback<Boolean> callback);
19
20     void cargarComboPerfil(AsyncCallback<PagingLoadResult<baseDatos>> callback);
21
22     void getCoordenadasFecha(String usuario, String fecha_inicio, String fecha_fin,
23         AsyncCallback<ArrayList<baseDatos>> callback);
24
25     void insertarDatosUsuarios(baseDatos objUsuario, AsyncCallback<String> callback);
26
27
28     void consultarPerfilPersonal(String cedula, AsyncCallback<Boolean> callback);
29
30     void insertarDatosPersonas(baseDatos objUsuario, AsyncCallback<String> callback);
31
32
33     void consultarDatosPersonales(String usuario, AsyncCallback<baseDatos> callback);
34
35     void actualizarDatosUsuarios(baseDatos objUsuario, AsyncCallback<String> callback);
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
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

```

Gráfico 118. Solución del error en la clase RPCApplicationAsync.
Elaborado por: El Investigador

```

public void actualizarDatosUsuario(baseDatos objUsuario)
{
    RPCApplication.Util.getInstance().actualizarDatosUsuarios(objUsuario, new AsyncCallback<String>() {

        @Override
        public void onFailure(Throwable caught) {
            // TODO Auto-generated method stub

        }

        @Override
        public void onSuccess(String result) {
            // TODO Auto-generated method stub
            com.google.gwt.user.client.Window.alert(result);
            txtfldDireccin.setEnabled(false);
            txtfldcrCorreo.setEnabled(false);
            nmbrfldTelefono.setEnabled(false);
            btnActualizar.setEnabled(false);
            chckbxEditar.setValue(false);
        }
    });
}

```

Gráfico 119. Código del método actualizarDatosUsuario.
Elaborado por: El Investigador

90. Para el caso de los usuarios con perfil **Personal** se ha creado un trigger encargado de mantener la integridad de los datos entre la tabla usuarios y personas.

```

CREATE OR REPLACE FUNCTION actualizar_datos() RETURNS TRIGGER AS $act_datos$
DECLARE
BEGIN

    IF OLD.perf_usu = 'Personal' THEN
        update personas
        set     mail_per=NEW.mail_usu,
              dir_per=NEW.dir_usu,
              tlf_uper=NEW.tlf_usu
        where ced_per=OLD.ced_usu;

    END IF;
    RETURN NULL;
END;
$act_datos$ LANGUAGE plpgsql;

CREATE TRIGGER actualizar_datos AFTER UPDATE
ON usuarios FOR EACH ROW
EXECUTE PROCEDURE actualizar_datos();

```

Gráfico 120. Trigger para actualizar datos del perfil personal.
Elaborado por: El Investigador

91. Capturas del Sistema Final

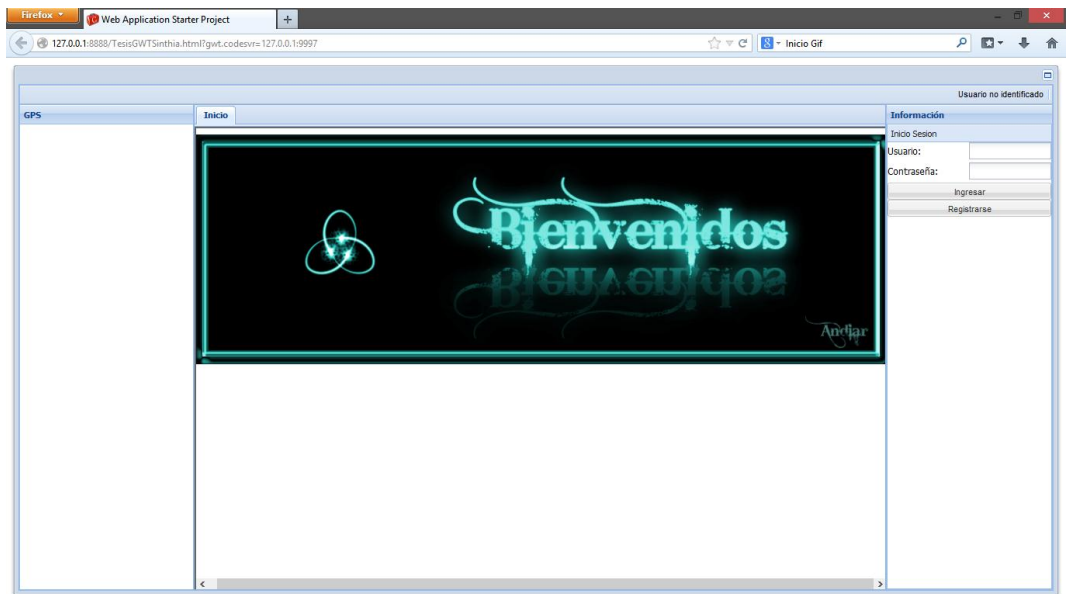


Gráfico 121. Vista General del Sistema - Principal.
Elaborado por: El Investigador

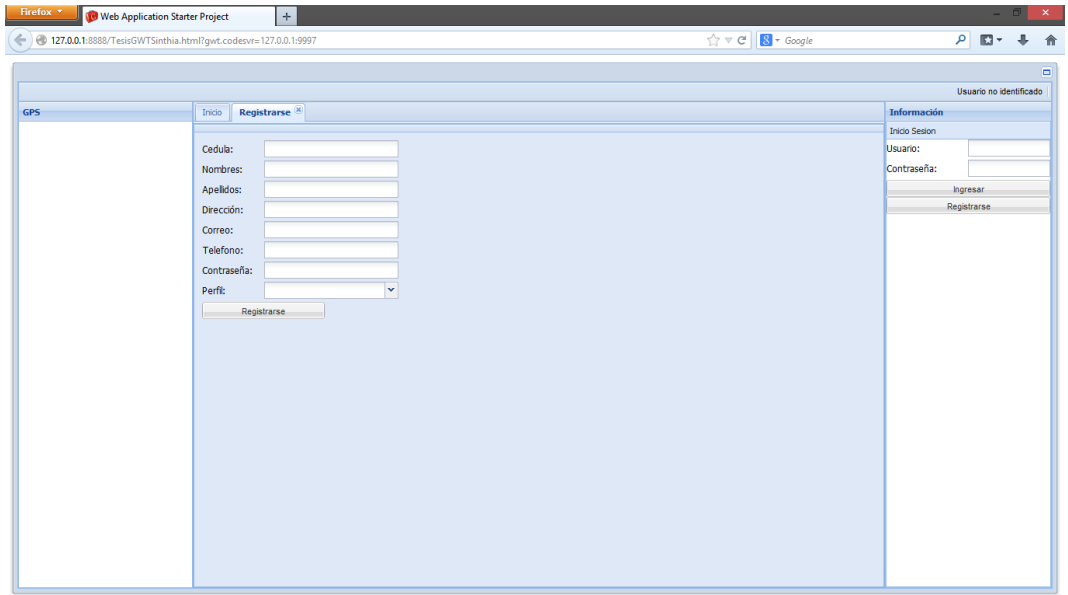


Gráfico 122. Gráfico 112. Registro de personas.
Elaborado por: El Investigador

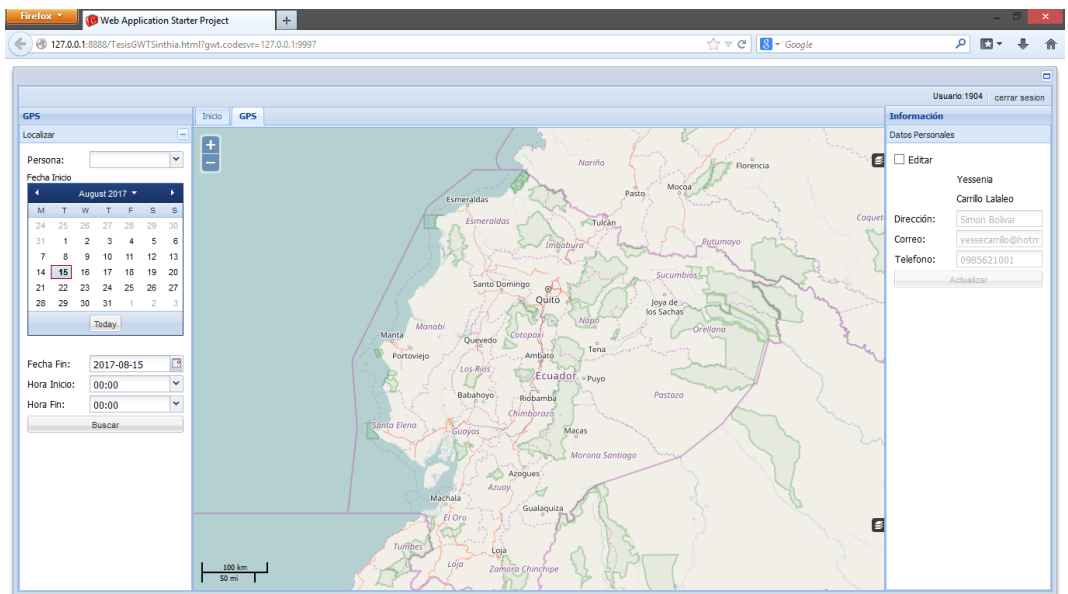


Gráfico 123. Mapa de búsqueda.
Elaborado por: El Investigador

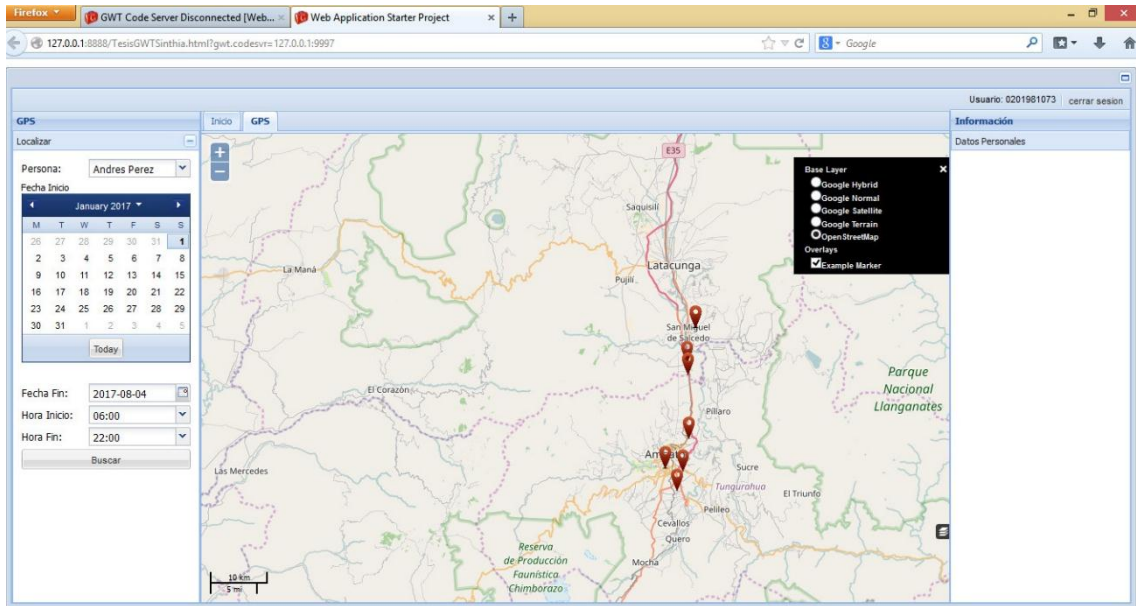


Gráfico 124. Marcadores de búsqueda.
Elaborado por: El Investigador

Para el desarrollo de la aplicación Android se ha utilizado **Android studio**.

1. Crear un nuevo proyecto y especificar el nombre de la aplicación.

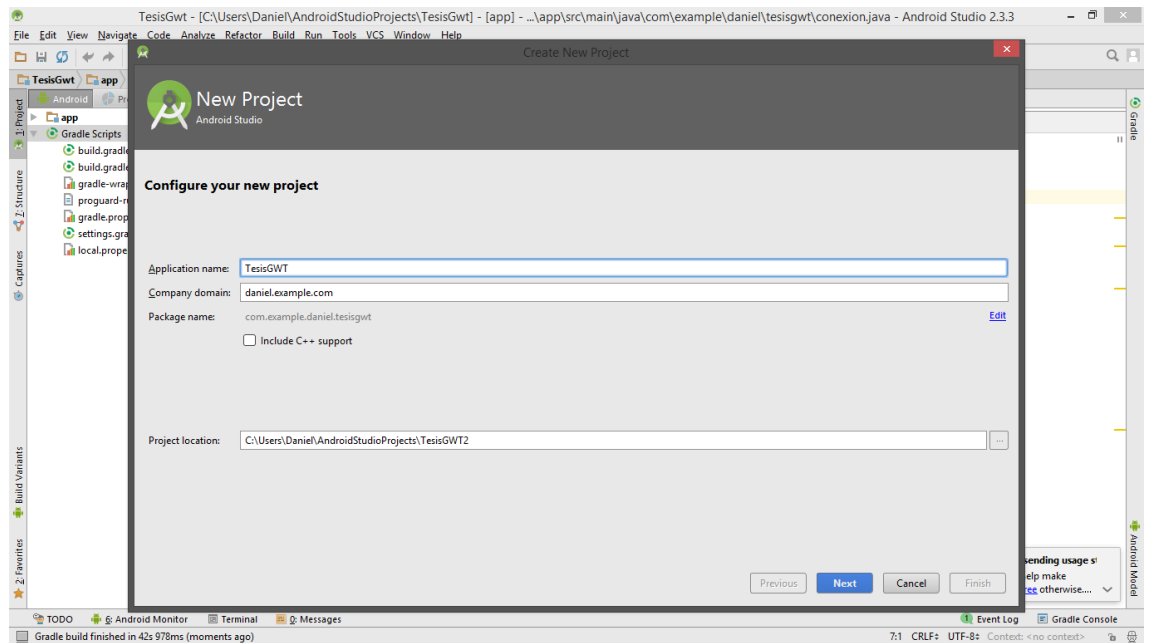


Gráfico 125. Nuevo proyecto Android.
Elaborado por: El Investigador

2. La versión mínima para la aplicación es **Android 6.0**

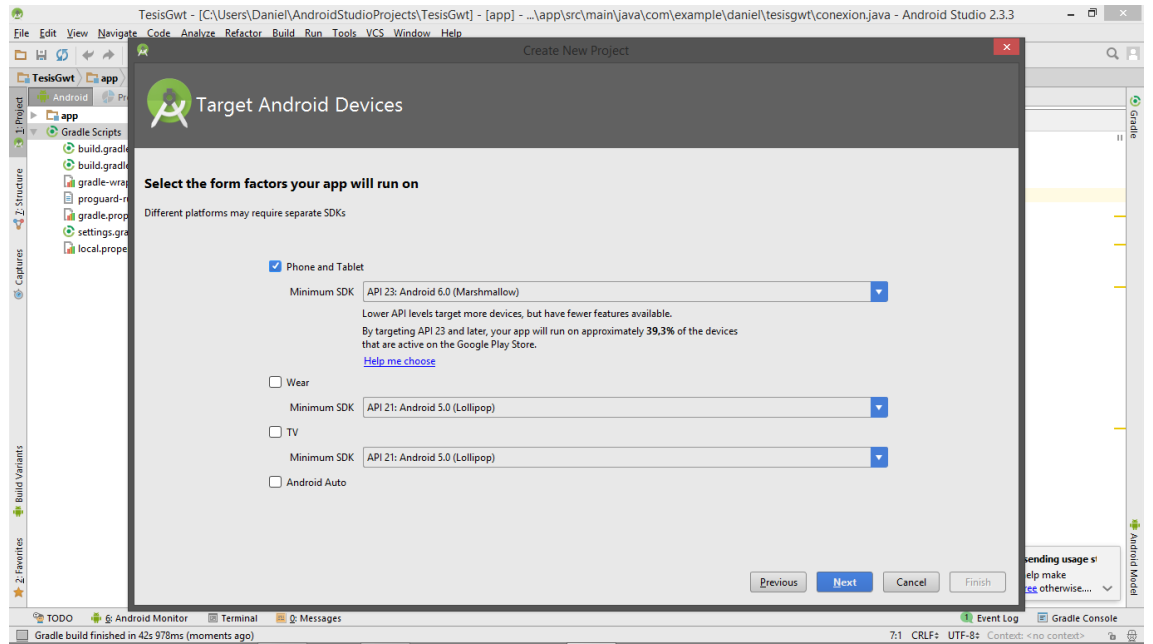


Gráfico 126. Versión de Android.
Elaborado por: El Investigador

3. Se escoge la plantilla para la creación de una actividad con Login, para utilizar los recursos que ofrece la herramienta.

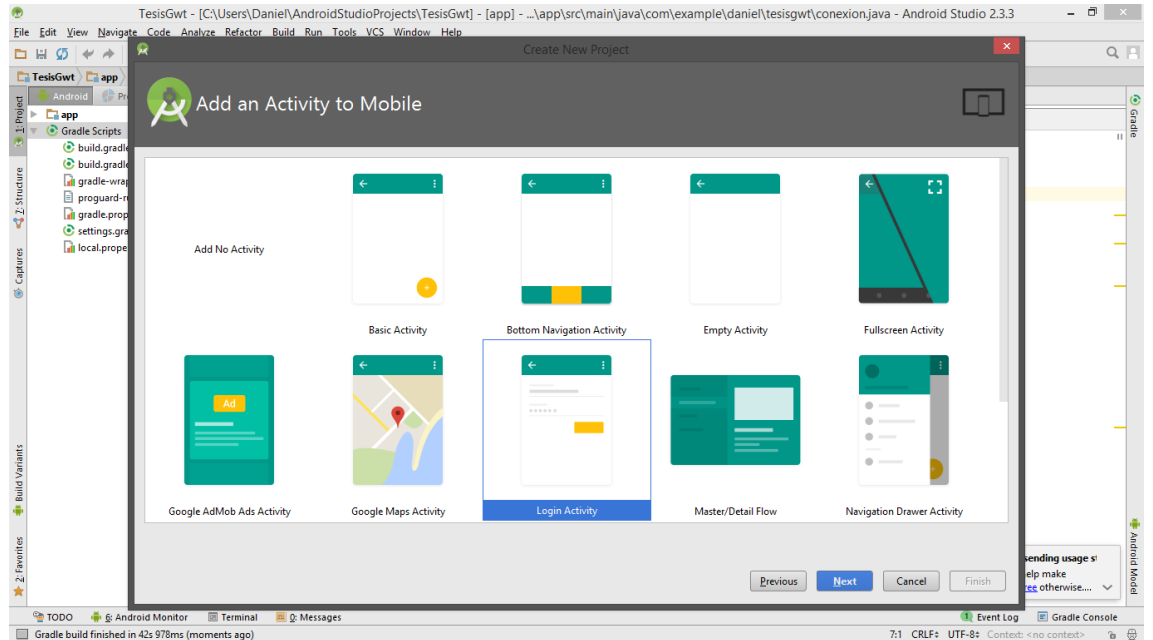


Gráfico 127. Plantilla Login.
Elaborado por: El Investigador

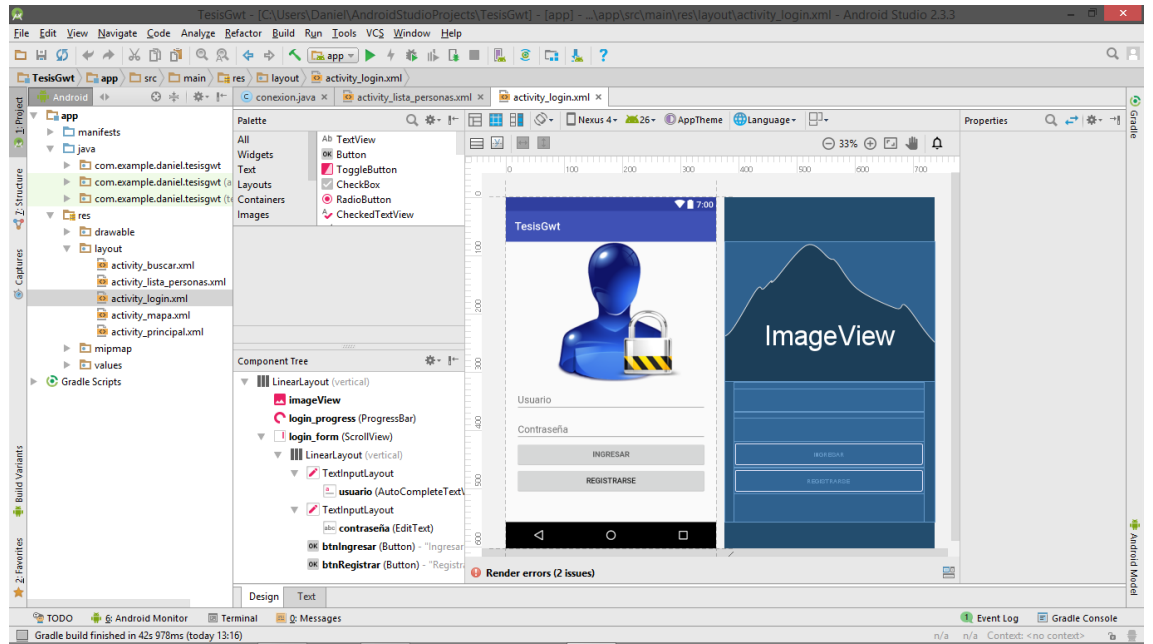


Gráfico 128. Programación activity_login.
Elaborado por: El Investigador

La actividad Login contiene:

- **ImageView:** Este componente permite mostrar una imagen en la aplicación, la imagen debe encontrarse dentro de la carpeta res\drawable.
- **TextInputLayout:** Este componente contiene una caja de texto donde se puede configurar si es un campo de tipo contraseña, numérico o cadena de caracteres.
- **Button:** Un componente que permite realizar una determinada actividad cuando se dispare un evento.

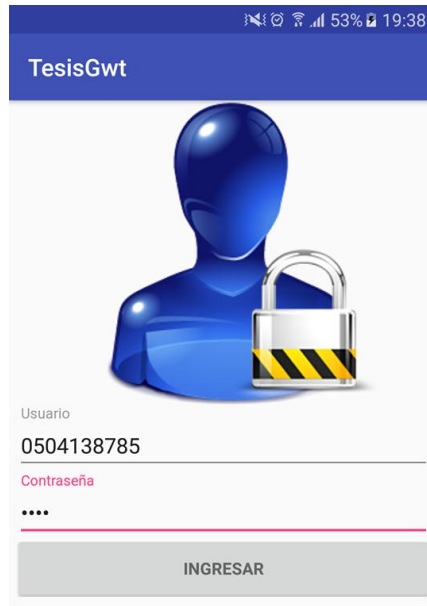


Gráfico 129. Vista principal - Login.
Elaborado por: El Investigador

4. Conexión con la base de datos

Para la parte de la conexión es necesario adjuntar la librería correspondiente para la conexión con la base de datos.

Se debe colocar la librería **.jar** en la carpeta **lib**, que se encuentra dentro del proyecto.

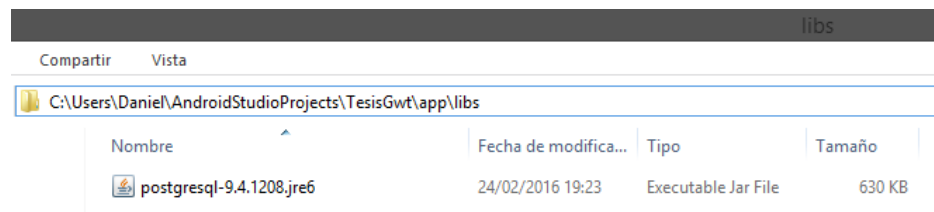


Gráfico 130. Vista principal - Login.
Elaborado por: El Investigador

También es necesario activar los permisos para tener acceso a la conexión inalámbrica y por tanto acceso a la red.

Estos permisos se colocan en el archivo **AndroidManifest.xml** que es un archivo de configuración de la aplicación.

```

manifest
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.daniel.tesisgwt">
    //permisos
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    //*****
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="TesisGwt"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Login">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ListaPersonas" />
        <activity android:name=".Principal" />
    </application>
    <!--
    The API key for Google Maps-based APIs is defined as a string resource.
    (See the file "res/values/google_maps_api.xml").
    Note that the API key is linked to the encryption key used to sign the APK.
    You need a different API key for each encryption key, including the release key that is used to
    sign the APK for publishing.
  
```

Gráfico 131. Vista principal - Login.
Elaborado por: El Investigador

5. Por último, para crear la clase conexión, se debe crear una nueva clase java.

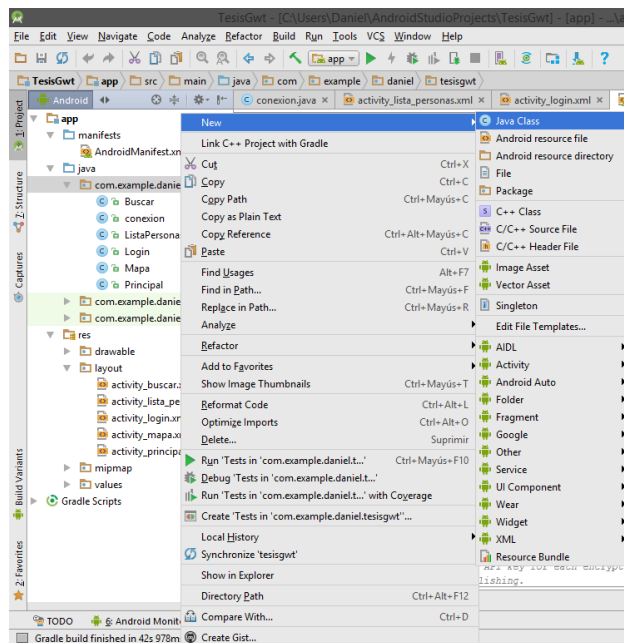


Gráfico 132. Clase conexión.
Elaborado por: El Investigador

```

conexion.java x activity_lista_personas.xml x activity_login.xml x AndroidManifest.xml x
conexion
1 package com.example.daniel.tesisgwt;
2
3 import ...
4
5
6
7
8 public class conexion {
9     private Connection conn;
10    public conexion() {
11    }
12
13    public Connection conectar() {
14
15        try {
16            Class.forName("org.postgresql.Driver");
17
18            conn = DriverManager.getConnection(
19                "jdbc:postgresql://192.168.1.6:5432/tesisGWT", "postgres", "1234");
20            // "jdbc:postgresql://192.168.137.141:5432/tesisGWT", "postgres", "1234");
21        } catch (SQLException e) {
22            e.printStackTrace();
23        } catch (ClassNotFoundException e) {
24            e.printStackTrace();
25        }
26        return conn;
27    }
28 }

```

Gráfico 133. Código de la clase conexion.
Elaborado por: El Investigador

Se crea el método conexión donde se especifica el tipo de gestor de base de datos y la cadena de conexión con la base.

Actividad Lista personas

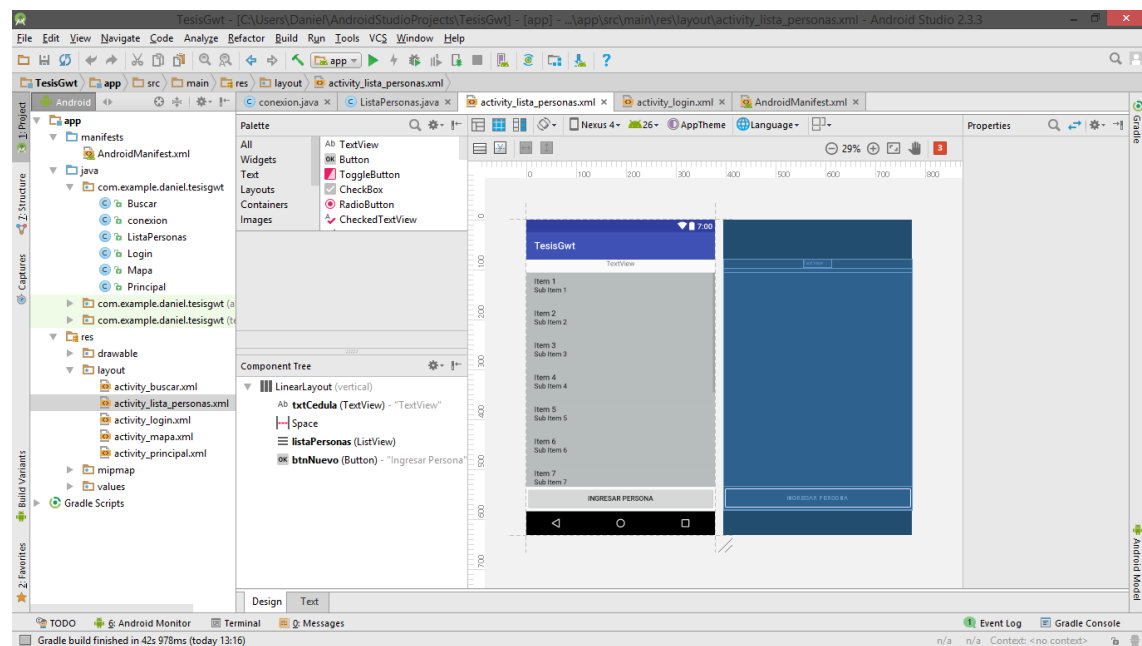


Gráfico 134. Activity lista_personas.
Elaborado por: El Investigador

- Esta actividad se muestra después de tener acceso al sistema, donde se muestra una lista con las personas registradas a la cuenta del usuario logeado.

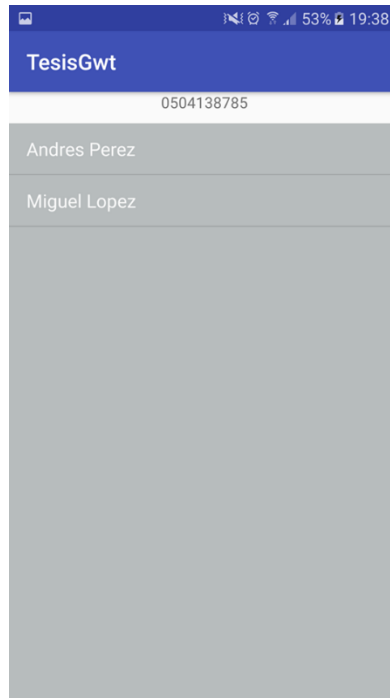


Gráfico 135. Acceso del Sistema.
Elaborado por: El Investigador

Actividad Principal

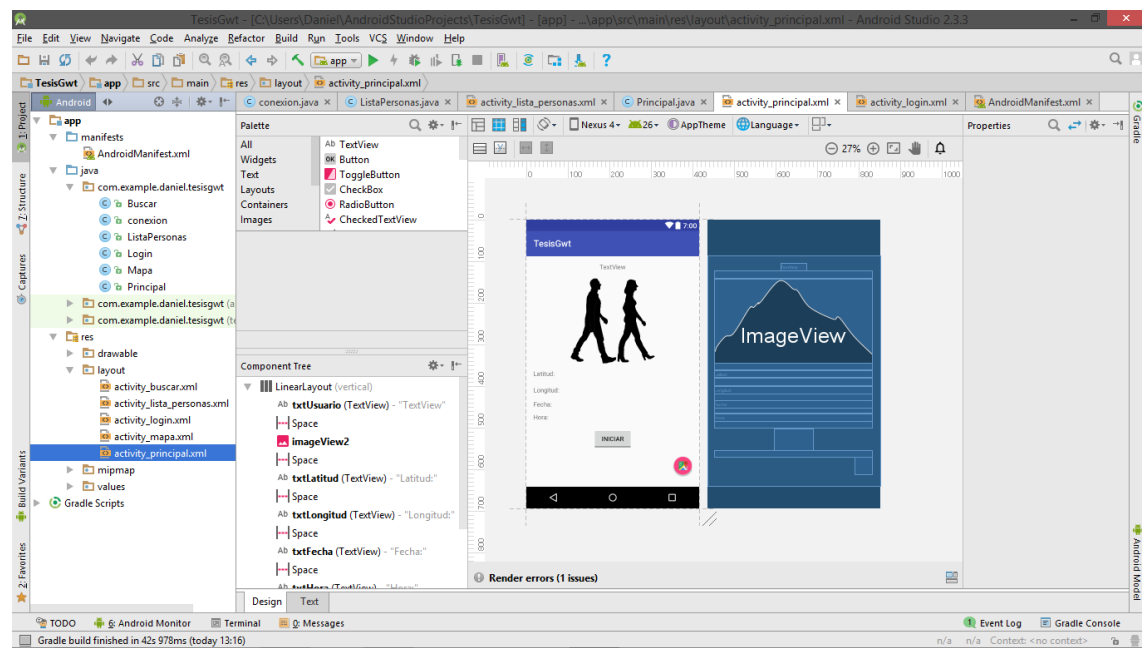


Gráfico 136. Actividad Principal.
Elaborado por: El Investigador

- Esta actividad se muestra al presionar un elemento de la lista de la actividad Lista Personas.

La actividad permite guardar en la base de datos la posición del dispositivo mediante latitud y longitud, cuando se acciona el botón iniciar.



Gráfico 137. Posición del dispositivo.
Elaborado por: El Investigador

8. También se puede detener el proceso presionando nuevamente el botón.

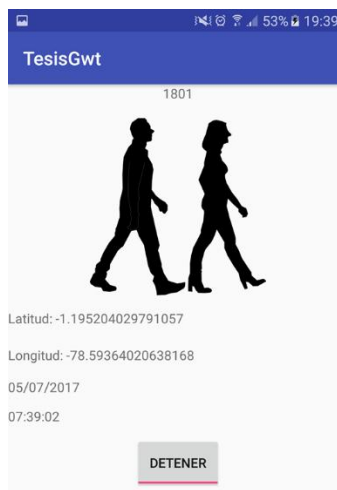


Gráfico 138. Detener Proceso.
Elaborado por: El Investigador

9. Para poder obtener la posición del dispositivo es necesario tener habilitado los permisos para acceder al gps del dispositivo.

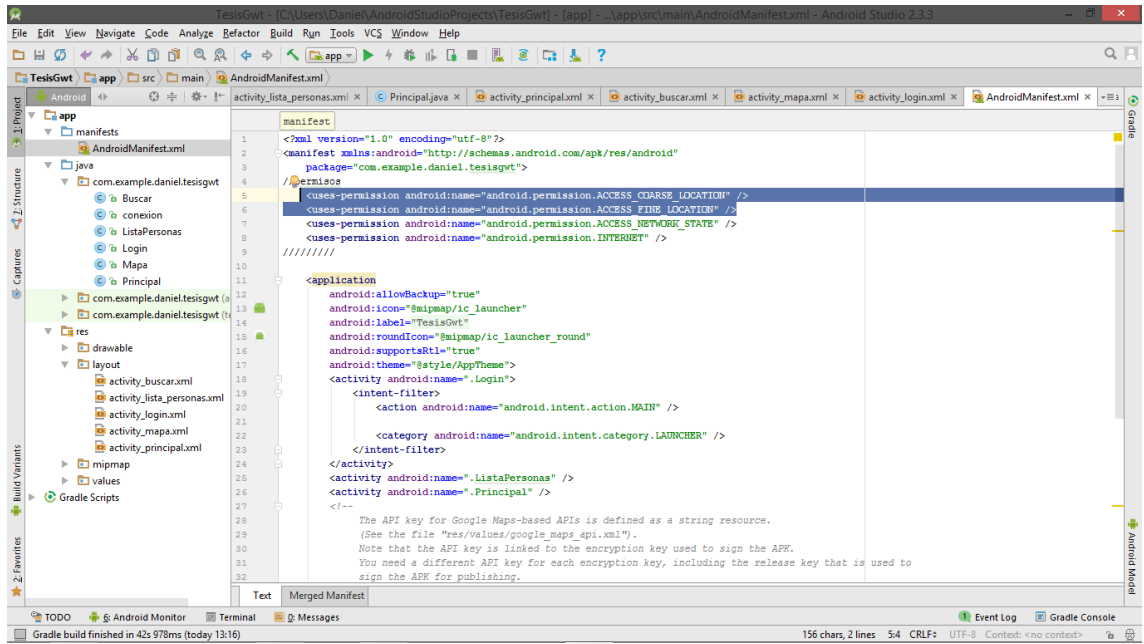


Gráfico 139. Habilitar permisos del GPS.
Elaborado por: El Investigador

Actividad Buscar

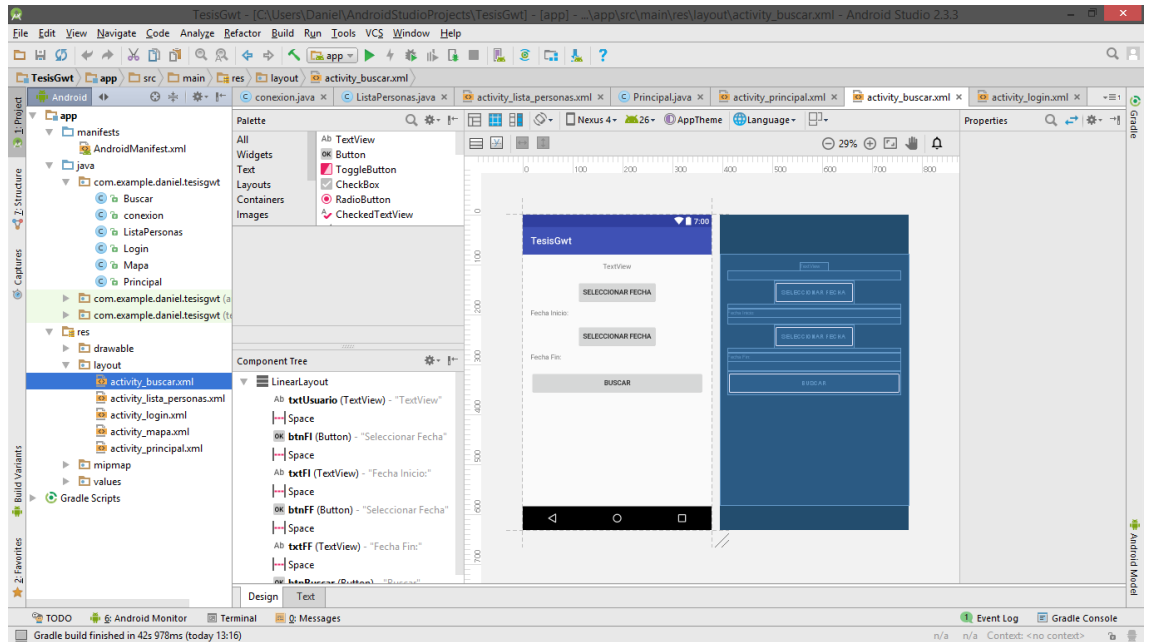


Gráfico 140. Activity_buscar.
Elaborado por: El Investigador

10. Esta actividad se muestra cuando se acciona el botón flotante con una imagen de un mapa que se encuentra en la actividad principal.



Gráfico 141. Iniciar la búsqueda.
Elaborado por: El Investigador

11. Al accionar el botón se muestra la actividad buscar donde se encuentran 2 campos de texto para especificar la fecha inicio y fin como criterio de búsqueda para mostrar los marcadores con la ubicación donde estuvo la persona en dicho tiempo.

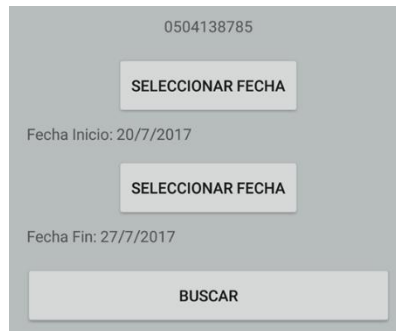
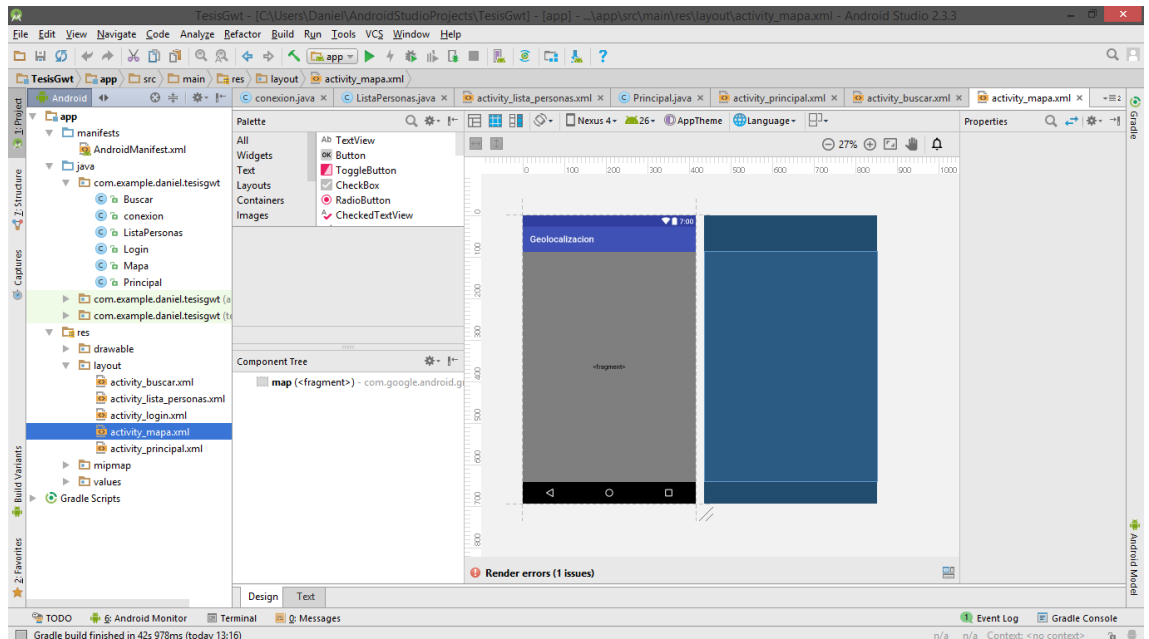


Gráfico 142. Actividad Buscar.
Elaborado por: El Investigador



Gráfico 143. Ubicación mediante marcadores.
Elaborado por: El Investigador

Actividad Mapa



*Gráfico 144. Actividad_mapa.
Elaborado por: El Investigador*

12. Esta actividad muestra un mapa donde se graficarán marcadores que representan una ubicación.

Para poder utilizar los mapas de Google es necesario tener un código de autorización para el uso de los mismos, para ello se debe acceder a la página que especifica el archivo `Google_maps_api.xml` que se crea en `res/values` al momento de crear la actividad mapa de Google.

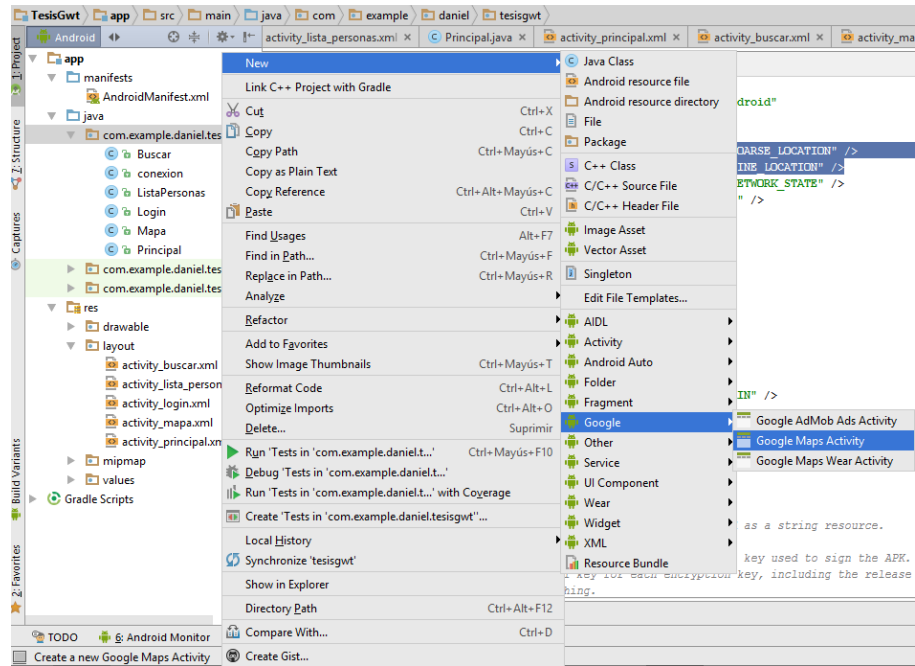


Gráfico 145. Acceso a mapas de Google.
Elaborado por: El Investigador

13. Con este código se podrá utilizar los mapas de google, para dibujar los marcadores es necesario implementar el siguiente método en la clase java perteneciente a **activity_mapa**.

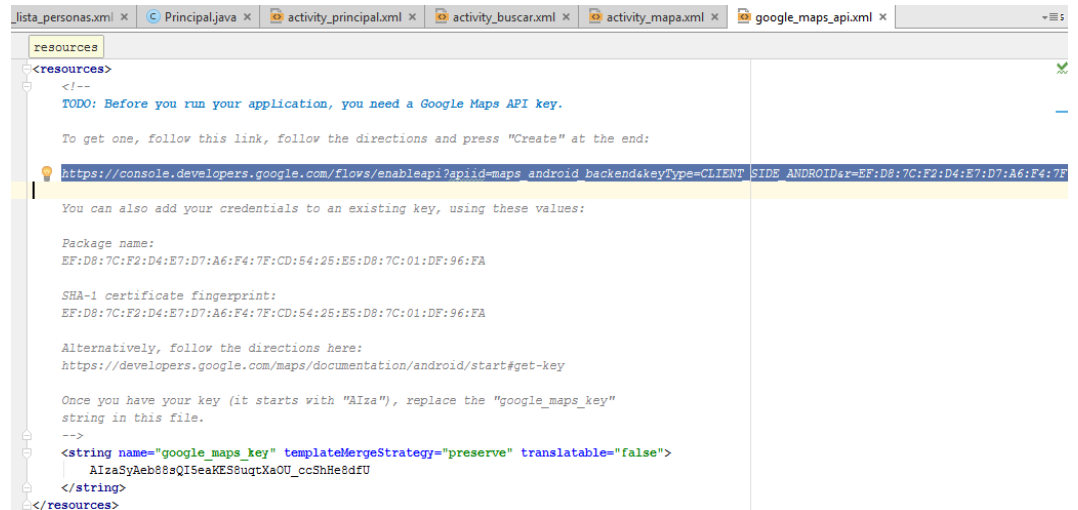


Gráfico 146. Uso de los mapas de Google.
Elaborado por: El Investigador

```
<!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow this link, follow the directions and press "Create" at the end:
  https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&rx=EF:D8:7C:F2:D4:E7:D7:A6:F4:7F:CD:54:25:E5:D8:7C:01:DF:96:FA

  You can also add your credentials to an existing key, using these values:

  Package name:
  EF:D8:7C:F2:D4:E7:D7:A6:F4:7F:CD:54:25:E5:D8:7C:01:DF:96:FA

  SHA-1 certificate fingerprint:
  EF:D8:7C:F2:D4:E7:D7:A6:F4:7F:CD:54:25:E5:D8:7C:01:DF:96:FA

  Alternatively, follow the directions here:
  https://developers.google.com/maps/documentation/android/start#get-key

  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
  string in this file.
-->
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">
  AIzaSyAeb9tsQ15eaKESuqKaOU_cc3HHeedf0
</string>
</resources>
```

Gráfico 147. Mapas de Google.
Elaborado por: El Investigador

14. En dicho método se especifica la ubicación del marcador y también se obtiene la dirección del marcador y se le adjunta como descripción al marcador.

```
onMapReady()
  * Installs Google Play services and returned to the app.
  */
  @Override
  public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    Lat=getIntent().getStringArrayExtra("lat");
    Long=getIntent().getStringArrayExtra("long");

    for(int i=0;i<Lat.length;i++){
      // Add a marker in Sydney and move the camera
      // Toast.makeText(getApplicationContext(),Double.valueOf(Lat[i])+"":*Double.valueOf(Long[i]) , Toast.LENGTH_LONG).show();

      LatLng sydney = new LatLng(Double.valueOf(Lat[i]), Double.valueOf(Long[i]));

      try {
        Geocoder geocoder = new Geocoder(this, Locale.getDefault());
        List<Address> list = geocoder.getFromLocation(Double.valueOf(Lat[i]), Double.valueOf(Long[i]), 1);
        if (!list.isEmpty()) {
          Address address = list.get(0);
          //messageTextView2.setText("Mi dirección es: \n" + address.getAddressLine(0));
          mMap.addMarker(new MarkerOptions().position(sydney).title(address.getAddressLine(0)));
        }
      } catch (IOException e) {
        e.printStackTrace();
      }

      mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }
  }
}
```

Gráfico 148. Mapas de Google.
Elaborado por: El Investigador



*Gráfico 149. Mapas de Google.
Elaborado por: El Investigador*



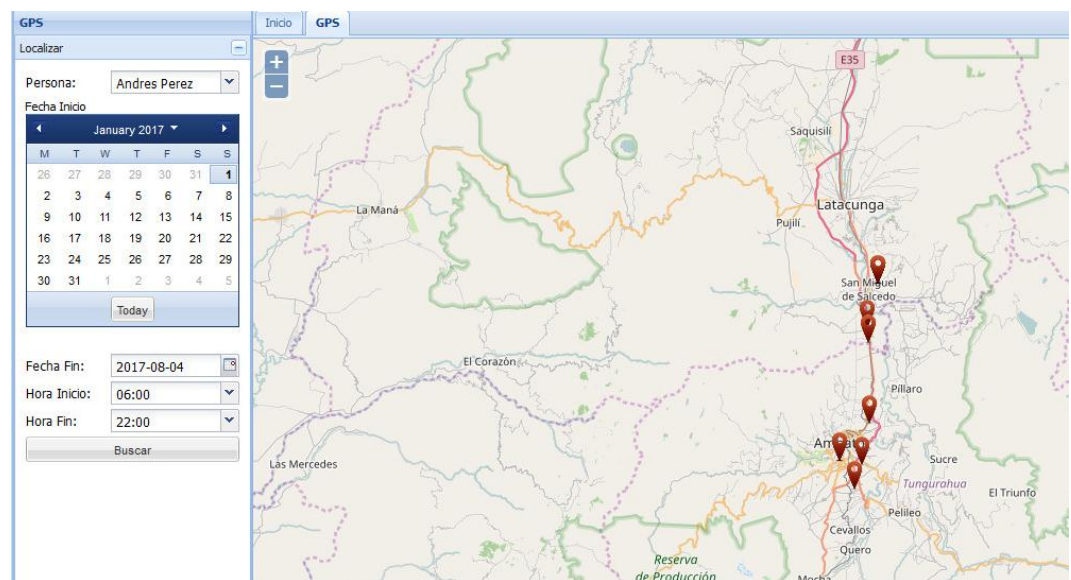
*Gráfico 150. Ubicación con marcadores en mapas de Google.
Elaborado por: El Investigador*

4.4 Pruebas

4.4.1 Pruebas de caja negra

En las pruebas de caja negra, se enfoca solamente en las entradas y salidas del sistema, sin preocuparse en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, se basa únicamente en los requerimientos de software y especificaciones funcionales.

Visualización de marcadores de búsqueda de ubicación en el mapa.



Descripción de la situación: En el mapa se debe poder visualizar la ubicación mediante marcadores de una persona en un día y lapso de tiempo específico.

Técnica de pruebas de caja negra: Requerimiento funcional

Caso 1

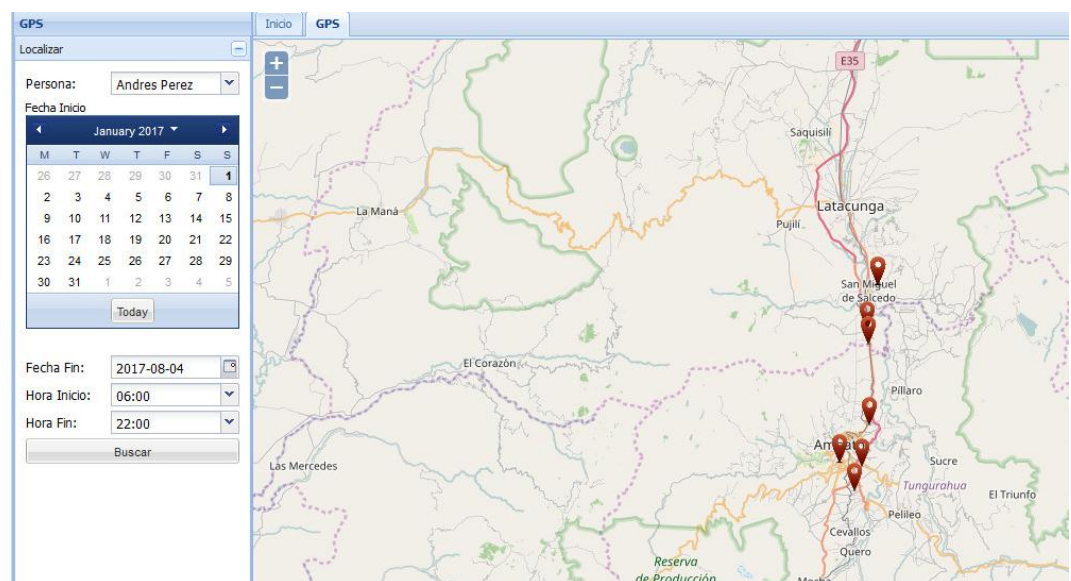
Datos de entrada: seleccionar los datos de la persona, así como también la fecha y el lapso de tiempo en el que se quiere saber su ubicación.

Resultado esperado (Salida): En la pantalla se visualiza correctamente la ubicación de la persona en el mapa mediante marcadores.

4.4.2 Pruebas de caja blanca

También suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba.

Visualización de marcadores de búsqueda de ubicación en el mapa.



Descripción de la situación: En el mapa se debe poder visualizar la ubicación de una persona mediante marcadores, en un día y lapso de tiempo específico.

Técnica de pruebas de caja blanca: Requerimiento funcional

Caso 1

Datos de entrada: Al hacer clic en el botón **Buscar**, se extraen los valores ingresados de fecha y hora.

El evento llama al método y grafica las coordenadas de la clase `tabItemMapa`.

Resultado esperado (Salida): Extrae la imagen del puntero desde un link en internet y grafica en el mapa instanciado.

CAPITULO 5

CONCLUSIONES Y RECOMEDACIONES

5.1 Conclusiones

- GWT, por ser una plataforma de código abierto, permite tener mejor control en el código facilitando el mantenimiento de las aplicaciones, además de que el ambiente de desarrollo es fácil e intuitivo, y la implementación de los sistemas hechos en esta plataforma es fácil y no requiere muchos gastos.
- Por la demanda de uso de esta plataforma existen actualizaciones y soporte, que permiten mantener y mejorar la plataforma facilitando el desarrollo de aplicaciones.
- El uso de la aplicación facilitó en la localización de los miembros del club, así como también ayudó a que los deportistas con discapacidad visual se desenvuelvan de mejor manera en sus competencias obteniendo un resultado favorable.
- Se ha elegido Eclipse como el IDE más adecuado para trabajar, porque permite agregar plugins a medida que se va desarrollando, logrando así una aplicación más liviana que es lo ideal al momento de navegar en ella.

5.2 Recomendaciones

- Es recomendable borrar el historial del navegador frecuentemente y la caché, en especial cuando se realicen cambios considerables en el sistema, de esta manera los cambios se verán reflejados en el sistema de manera inmediata.
- Crear y utilizar componentes propios, de esta manera el desarrollo será más eficiente con respecto a los controles de ingreso de datos.
- Adjuntar las librerías externas al proyecto, de esta manera no existirán problemas al momento de subir el proyecto a un servidor.

Bibliografía

- [1] Bortolotti, N. (2010). Introduction to GWT Remote Procedure Calls (RCP). Retrieved from <http://www.ajaxmatters.com/2006/05/introduction-to-gwt-remote-procedure-calls-rpc-with-example-application/>
- [2] Bravo García, E. (2013). El framework Google Web Toolkit. Retrieved from <http://www.jtech.ua.es/jornadas/charlas/presentacionGWT.pdf>
- [3] Burke , E. (2009). Selección del IDE más adecuado para GWT. Retrieved from <http://stuffthathappens.com/blog/2009/08/26/gwt-ide-seleccion/>
- [4] Carrasco Moñino, M. (2010, Octubre 15). Entendiendo GWT. Retrieved from <https://www.paradigmadigital.com/dev/entendiendo-que-es-gwt/>
- [5] Eclipse. (n.d.). Retrieved from <http://www.gwtproject.org/download.html>
- [6] Ju, Y., & Illescas, E. (2012). Ceditec. Retrieved from http://www.ceditec.etsit.upm.es/index.php?option=com_content&view=article&id=22087&Itemid=1440&lang=es
- [7] Maldonado Daniel . (2008). El Código K . Retrieved from <http://elcodigok.blogspot.com/2008/03/ides-para-programar-en-java.html>
- [8] Mora , M., & Meneses , A. (2012). GOOGLE WEB TOOLKIT . Retrieved from <http://biblioteca.unitecnologica.edu.co/notas/tesis/0062641.pdf>
- [9] Murillo Vélez , H., & Riera Idrovo , S. (2010). Análisis de la plataforma RIA GWT para el desarrollo en AJAX para el departamento de recursos humanos de la refinería de Esmeraldas . Riobamba .
- [10] Rivera Arbeláez , D. (2013). Emprendimiento en pro de la inclusión . SENA .
- [11] Robledo , C., & Robledo, D. (n.d.). Google Web Toolkit . Saavedra.
- [12] Snyflex. (2010). Informática Pura. Retrieved from <http://informaticapura.com/2010/04/14/el-ide-de-tu-preferencia/>
- [13] Teknoda. (2012). Selección de un IDE gratuito . Retrieved from <http://www.teknoda.com/tips/java/java01.pdf>
- [14] Eclipse download, Retrieved from <http://www.gwtproject.org/download.html>
- [15] Dewsbury, R. Google Web Toolki Applications. 1° ed. Massachusets. 2008. 574p. ISBN: 0-321-50196-9

[16] Agudo, C. (2012). Sistema de localización de dispositivos móviles. Retrieved from <https://e-archivo.uc3m.es/bitstream/handle/10016/5843/pfcWiLBS.pdf>

[17] Chaganti, P; Google Web Toolkit. 1° ed. Birmingham. 2007. 232p. ISBN:978-1-847191-00-7

Anexos

Anexo N°1 – Manual de Usuario

Este manual tiene el objetivo de facilitar el uso del sistema a los usuarios, y mostrar las funcionalidades del mismo.

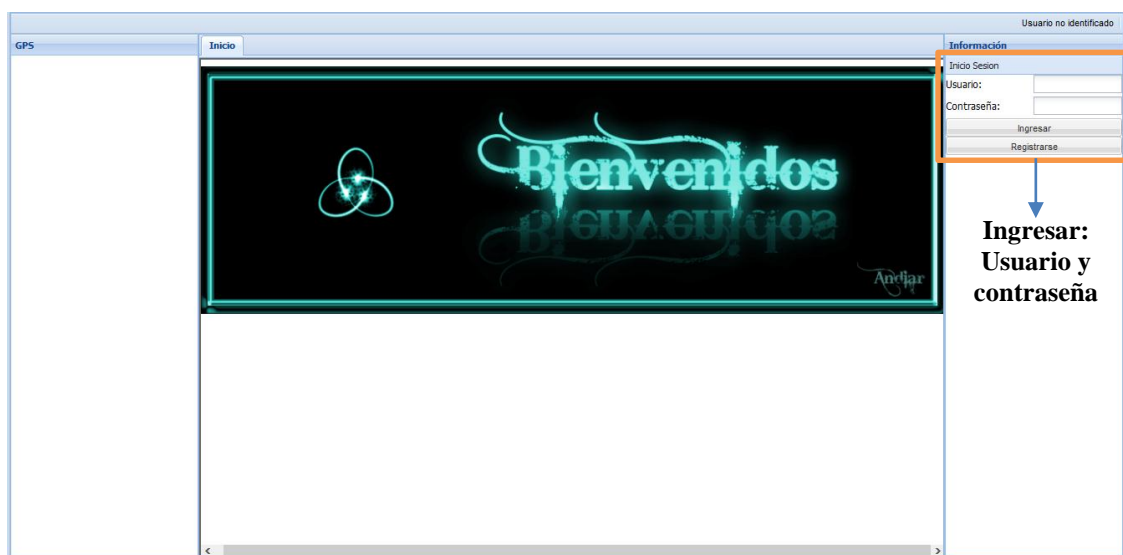
Antes de iniciar se debe tomar en cuenta:

La aplicación utilizará el navegador Mozilla Firefox versión 24.0

Ingreso al Sistema

El ingreso al sistema se realiza desde un navegador web, Mozilla Firefox, así como también en un teléfono con tecnología Android.

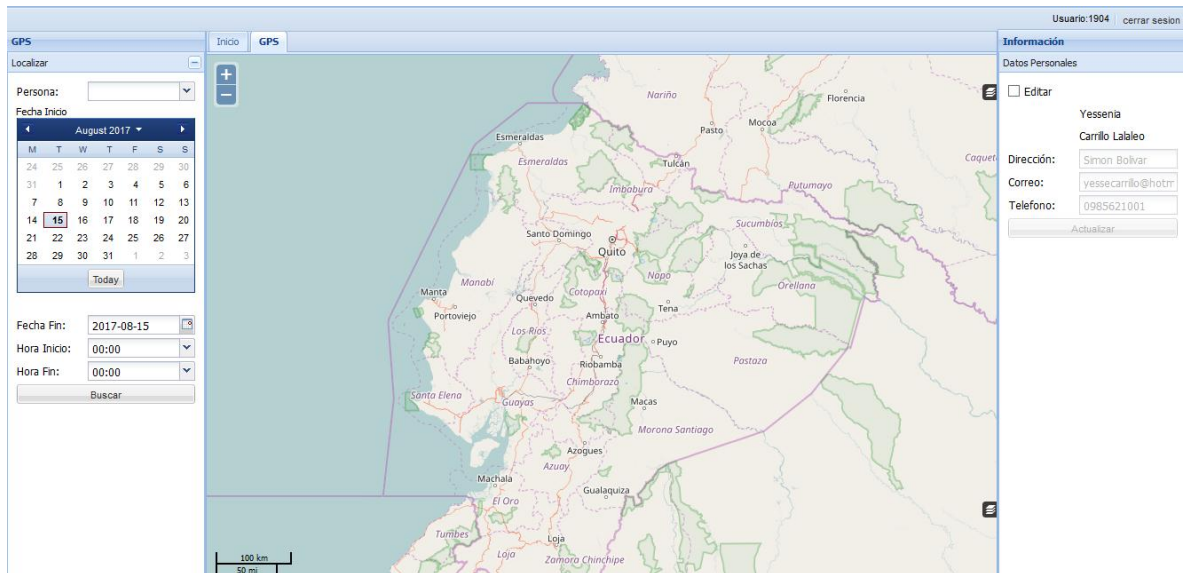
La siguiente presentación muestra la página de inicio del sistema:



Para ingresar al sistema se debe proveer las credenciales que habilitan al usuario a trabajar según el perfil asignado **Personal o Institucional**, que presenta el sistema.

Ingresar el **usuario** y la **contraseña** y, a continuación, hacer clic en el botón “Ingresar”. Estas credenciales están dadas por un código de usuario y la contraseña correspondiente. Cada usuario tiene asignado un código personal y una contraseña, personales y no transferibles.

Si el código de usuario y la contraseña son correctos, entonces el sistema desplegará la página principal, cuyas opciones de navegación dependerán de rol del usuario que ha ingresado al sistema, las mismas cumplen con el siguiente patrón de diseño:



Si los datos no son correctos, se mostrará un mensaje de error y no permitirá ingresar al sistema.

Estructuras de las vistas de trabajo

Todas las vistas de trabajo del sistema, comparten una misma estructura que consta de los siguientes elementos:

- a) **Datos:** Esta en la parte superior izquierda en donde se muestra los datos que se van a ingresar de la persona que se desea localizar.
- b) **Mapa:** Se encuentra en la parte central, ahí se visualiza el mapa que se está utilizando, dependiendo del mapa que se está utilizando.
- c) **Información:** SE encuentra en la parte superior derecha, donde se visualizan los datos del usuario, y muestra la opción para editar los datos del usuario que está utilizando.

Búsqueda de personas - localización

Para realizar la búsqueda de una persona se debe ingresar los datos, el día y el lapso de tiempo.

Localizar

Persona: → Seleccionar la persona que se desea localizar.

Fecha Inicio

January 2017						
M	T	W	T	F	S	S
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

→ Seleccionar la fecha de Inicio.

Fecha Fin: → Seleccionar la fecha de fin de la búsqueda.

Hora Inicio:

Hora Fin:

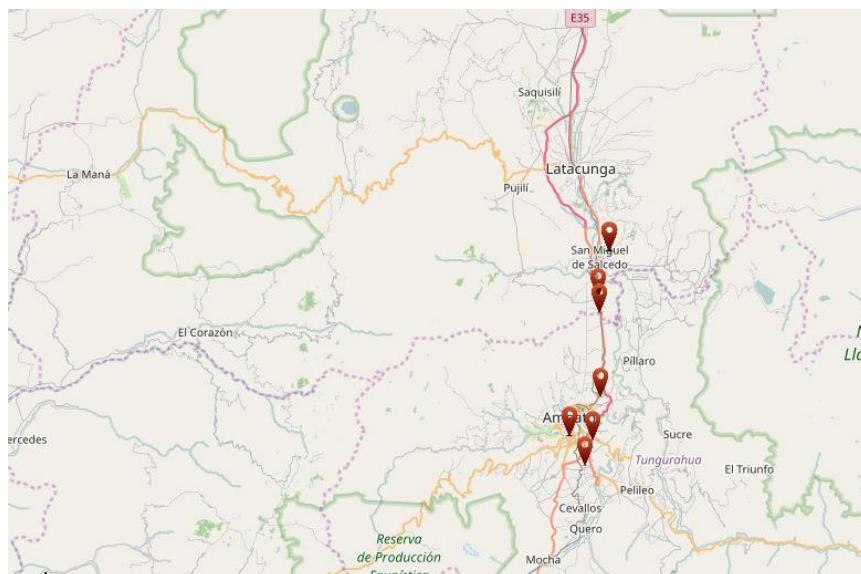
← Lapso de tiempo de la búsqueda.

Buscar

1. Se debe seleccionar el nombre de la persona que desea saber la ubicación.
2. Seleccionar la fecha de inicio, es decir desde que día necesita conocer la ubicación.
3. Seleccionar la fecha de fin, para conocer la última ubicación de la persona.
4. En la hora de inicio y fin, debe seleccionar la hora desde la que desea conocer la ubicación.

Después de haber llenado los datos mencionados anteriormente, hacer clic en Buscar.

La búsqueda se visualizará mediante marcadores, como se presenta a continuación:



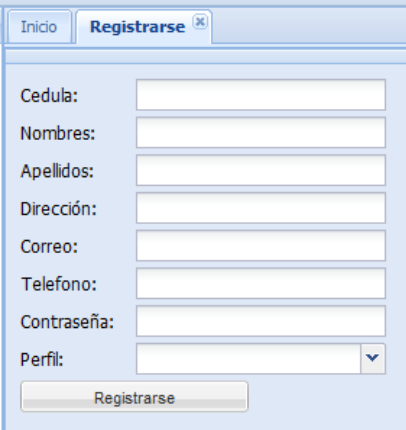
Registro de personas

1. En caso que se ingrese al sistema mediante el perfil institucional, tiene la opción de registrar personas, para ello debe hacer clic en el botón **Registrar**.



The screenshot shows a web interface titled "Información". It contains a section for "Inicio Sesión" with input fields for "Usuario:" and "Contraseña:". Below these fields are two buttons: "Ingresar" and "Registrar". An arrow points from the text "Registrar personas" to the "Registrar" button.

2. Para registrar una nueva persona, tiene que ingresar los datos que se muestran en la siguiente presentación:



The screenshot shows a registration form with the following fields: "Cedula:", "Nombres:", "Apellidos:", "Dirección:", "Correo:", "Telefono:", "Contraseña:", and "Perfil:" (a dropdown menu). A "Registrar" button is located at the bottom of the form.

3. Cuando ya estén llenos todos los campos de forma correcta, hacer clic en **Registrar**, y la persona quedará registrada en la base de datos y aparecerá en la lista de personas, para realizar la búsqueda.



The screenshot shows a search results table. The first row is highlighted and contains the following information: "Persona:" with a dropdown menu showing "Miguel Lopez", "Fecha Inicio" with a date, and "Au" (likely initials) with "Miguel Lopez". Below the table are navigation buttons: "M", "T", "W", "T", "F", "S", "S".

Editar usuario

1. Si el usuario cambio algún dato personal, para editar el usuario en el sistema, tiene que activar la opción editar, hacer clic en el check **Editar** para activar los campos de la siguiente sección:

2. Después de modificar los datos personales, hacer clic en el botón **Actualizar**, y los datos se guardarán en la base de datos.

Para el uso del sistema en Android, se lo realiza de la siguiente forma:

1. Para acceder a la aplicación, ingresar las credenciales **Usuario** y **Contraseña**, y hacer clic en el botón **Ingresar**.

2. Después de haber ingresado las credenciales de forma correcta el sistema se presentará como se muestra en la siguiente imagen:

Dependiendo de las personas que se encuentren designadas para el usuario que accede se visualizarán como se muestra en la imagen.

3. Para iniciar con la búsqueda, para visualizar la ubicación de la persona seleccionada, hacer clic en el botón **Iniciar**.



Al dar clic en **Iniciar** se guardará la longitud y latitud de la posición del dispositivo.

4. Para realizar una búsqueda específica, es decir en un día y hora determinados, hacer clic en el botón del mapa que se encuentra en la actividad principal.



5. Al dar clic en el botón del mapa, se mostrará la siguiente actividad, en la que debe llenar los datos solicitados, como se muestra en la siguiente imagen:

0504138785

SELECCIONAR FECHA

Fecha Inicio: 20/7/2017

SELECCIONAR FECHA

Fecha Fin: 27/7/2017

BUSCAR

Después de seleccionar la fecha, hacer clic en el botón **Buscar**, para que se visualice la ubicación, como está en la siguiente presentación:



Anexo N° 2 – Encuesta para el desarrollo de una aplicación de geo-referenciación para personas con discapacidad visual (Entrenadores del club “Fénix”)

**UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**



Encuesta dirigida a entrenadores del Club “Fénix” Tungurahua.

OBJETIVO:

Conocer la importancia de una aplicación de geo-referenciación en el campo deportivo para personas con discapacidad visual.

Instrucciones

- Al ser anónima la encuesta responda con toda libertad y sinceridad.
- Antes de responder las preguntas, lea atentamente, reflexione y luego de su opinión.
- Marque con una **X** dentro del paréntesis en la respuesta que considere correcta.

CUESTIONARIO

1. **¿Conoce una aplicación de geo-referenciación (ubicación) para personas con discapacidad visual?**
Si () No ()

2. **¿Le gustaría contar con una aplicación que le permita conocer la ubicación de los deportistas que forman parte del club “Fénix”?**
Si () No ()

3. **¿Qué tipo de dato le gustaría saber de su deportista que tiene discapacidad visual?**
Ubicación ()
Hora ()
Fecha ()
Todas las anteriores ()

4. **¿Con la aplicación de geo-referenciación se pueden mejorar los resultados en las diferentes competencias?**
Si () No ()

5. **¿Al contar el club con una aplicación de geo-referenciación obtendrá mayor confianza por parte de los familiares de los deportistas con discapacidad visual?**
Si () No ()

Gracias por su colaboración.

Anexo N° 3 – Encuesta para el desarrollo de una aplicación de geo-referenciación para personas con discapacidad visual (Familiares de personas con discapacidad visual del club “Fénix”)

**UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**



**Encuesta dirigida a familiares de personas con discapacidad visual del Club “Fénix”
Tungurahua.**

OBJETIVO:

Conocer la importancia de una aplicación de geo-referenciación en el campo deportivo para personas con discapacidad visual.

Instrucciones

- Al ser anónima la encuesta responda con toda libertad y sinceridad.
- Antes de responder las preguntas, lea atentamente, reflexione y luego de su opinión.
- Marque con una **X** dentro del paréntesis en la respuesta que considere correcta.

CUESTIONARIO

1. ¿Le gustaría contar con una aplicación que le permita conocer la ubicación de su familiar con discapacidad visual durante una competencia deportiva?

Si () No ()

2. ¿Qué tipo de dato le gustaría saber de su familiar que tiene discapacidad visual?

Ubicación ()
Hora ()
Fecha ()
Todas las anteriores ()

3. ¿Considera que sería indispensable contar una aplicación de geo-referenciación en el club?

Si () No ()

4. ¿Cree usted que con la aplicación de geo-referenciación se pueden mejorar los resultados en las diferentes competencias?

Si () No ()

Gracias por su colaboración.

Anexo N° 4 – Entrevista para el desarrollo de una aplicación de geo-referenciación para personas con discapacidad visual (Personas con discapacidad visual)

**UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**



**Encuesta dirigida a las personas con discapacidad visual del Club “Fénix”
Tungurahua.**

OBJETIVO:

Conocer la importancia de una aplicación de geo-referenciación en el campo deportivo para personas con discapacidad visual.

CUESTIONARIO

- 1. ¿Le gustaría que su club cuente con una aplicación que le permita conocer su ubicación durante una competencia deportiva?**

Si () No ()

- 2. ¿Utiliza algún dispositivo móvil inteligente?**

Si () No ()

- 3. ¿Qué tipo de dato le gustaría que su entrenador o familiar conozca mediante la utilización de la aplicación de geo-referenciación?**

Ubicación actual ()

Última ubicación ()

Hora ()

Fecha ()

Todas las anteriores ()

- 4. ¿Cree usted que con el uso de la aplicación de geo-referenciación se pueden mejorar los resultados en las diferentes competencias?**

Si () No ()

Gracias por su colaboración.