



CARSHARING

**Aplicación para compartir el carro entre miembros de una comunidad
Caso de estudio: Universidad de los Llanos**

**JONATHAN ACOSTA MORALES
SERGIO ANDRÉS MURILLO JIMÉNEZ**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
ESCUELA DE INGENIERÍA
PROGRAMA DE INGENIERIA DE SISTEMAS
VILLAVICENCIO
2017**



CARSHARING

**Aplicación para compartir el carro entre miembros de una comunidad
Caso de estudio: Universidad de los Llanos**

**SERGIO ANDRÉS MURILLO JIMÉNEZ
JONATHAN ACOSTA MORALES**

**Trabajo presentado como requisito parcial para optar título de
Ingeniero de Sistemas**

**Directora:
OLGA LUCERO VEGA MARQUEZ
MsC en Ingeniería de Sistemas y Computación**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
ESCUELA DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
VILLAVICENCIO
2017**

NOTA ACEPTACION

OLGA LUCERO VEGA MARQUEZ
Directora

CESAR AUGUSTO DÍAZ CELIS
Jurado

JESUS REYES CARVAJAL
Jurado

Fecha

AGRADECIMIENTOS

A Dios porque es quién hace que todo sea posible, a mi padre, que desde su partida prometí mirar al cielo y dedicarle mi futuro grado como profesional. Viejo querido, lo logré.

De igual manera a mi mamá que es el motor de mi vida y sin su apoyo difícilmente esto sería posible. Todo lo que soy y lo que soñaba ser, fue gracias a ti.

A mi novia, que siempre creyó en mí cuando yo no creía en mí mismo. Gracias por ser mi compañera, mi amiga, mi consejera y mi complemento hacia la felicidad. "Amor" siempre fue una palabra, hasta que llegaste tú y le diste significado.

Al grupo de profesores del programa de Ingeniería de Sistemas, quienes aportaron sus conocimientos a mi formación académica.

Una mención especial a la Ingeniera Mónica Silva por su total amabilidad, por su permanente disposición, pero sobre todo por su don de persona.

Jonathan Acosta

RESUMEN

El presente trabajo consiste en el desarrollo de una aplicación web para compartir el carro entre miembros de la comunidad universitaria Unillanos, utilizando las metodologías Design Science Research (DSR) en el proceso investigativo y Personal Software Process (PSP) en lo que tiene que ver con el desarrollo de software, además de la aplicación de la tecnología Java Enterprise Edition (JavaEE).

El trabajo tiene su origen en una problemática en el transporte desde y hacia la Universidad de los Llanos, a pesar de que se cuenta con servicios de ruta ofrecidos por la institución, éstos presentan varios inconvenientes, entre los que se encuentran: No cubrimiento total de la ciudad, horarios limitados, algunos horarios presentan sobrecupo en los buses, algunas rutas incumplen horarios establecidos, entre otros.

Con este trabajo se investigó, siguiendo el enfoque “Design Science Research” (la ciencia del diseño), una solución informática que permita a los propietarios de vehículos poder compartir sus asientos disponibles en su recorrido habitual, con personas de la misma comunidad, que así lo requieran, siendo esta solución un beneficio y una ventaja para todos los miembros de la Universidad de los Llanos.

La TIC seleccionada fue la plataforma Java Enterprise Edition (JavaEE) ya que permite el desarrollo y ejecución de aplicaciones empresariales a la medida, enfocadas en la solución de problemas críticos con un diseño de arquitectura de software en múltiples capas, que separen los aspectos de interface, modelo de negocio y modelo de datos. [1] Esto facilita las posibilidades de mantenimiento debido a que cada capa es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo, un diseño de arquitectura escalable que puede ampliarse con facilidad en caso que las necesidades aumenten.

El proceso de desarrollo utilizado muestra procedimientos estrictamente diseñados, tareas medibles, puntos de control identificados, tiempos de desarrollo determinados y documentación para facilitar la detección de fallas y la oportunidad de mejora, tal y como lo plantea la metodología de desarrollo de software PSP. De esta forma, se construyen productos de software de alta calidad que garantizan funcionalidad, eficiencia, escalabilidad, confiabilidad, portabilidad y fácil mantenimiento.

ABSTRACT

This work consists in the development of a web application to share the car with members of the university community Unillanos, using the Design Science Research (DSR) for the research process, the Personal Software Process for Software Development and applying the Java Enterprise Edition technology to construct the product.

This project was born in an attempt to solve the existing transport problem from and to the Universidad de los Llanos, although there is a bus service offered by the institution, this

service is very limited as its scope does not cover all the city, the service schedule is insufficient, there are not enough buses during certain times of the day and some routes are simply not done.

Using DSR (Design Science Research) approach, an informatic solution was developed, with the aim of allowing the owners of vehicles to share the available seats in their habitual journeys, with people from the community, giving benefit to all university members.

The ICT selected was de Java Enterprise Edition platform which allows the personalized enterprise applications development and deploy, focused on the critical problems solutions with a layers software architecture which separates the interface, business model and data model issues. [1] It make easier to do maintenance activities due each layer is independent of each other and it is possible to make changes or updates without changing the application at overall. In the same way, the design is a scalable solution that can be extended in a easy way in case of be necessary.

The development process shows strictly designed processes, measurable tasks, contro points identified, development time determined and documentation to facilitate fail prevention and improvement opportunity detection such as in the way proposed by PSP development methodology. Thus, the software product are quality valued and guarantee functionality, efficiency, scalability, confidence, portability and easier maintenance.

TABLA DE CONTENIDO

RESUMEN	5
ABSTRACT	5
1. INTRODUCCION	14
1.1 Tema.....	14
1.2 Descripción del problema.....	14
1.3 Justificación	15
2. OBJETIVOS	17
2.1 Objetivo general.....	17
2.2 Objetivos específicos	17
3. MARCO TEÓRICO.....	18
3.1 Sistemas de transporte e iniciativas en el mundo	18
3.1.1 Transporte masivo en Latinoamérica	18
3.1.2 Contexto del sistema de transporte masivo en Colombia	19
3.1.3 El transporte en la Universidad de los Llanos.....	20
3.1.4 Otras iniciativas de compartir el carro ("Car Sharing").....	20
3.2 Contexto de Ingeniería de Software y tecnologías disponibles	20
3.2.1 Ingeniería de software	20
3.2.2 Aplicaciones Web	21
3.2.3 Java	21
3.2.4 Java EE	21
3.2.4.1 Arquitectura Java EE	21
3.2.5 Html5.....	22
3.2.6 Css	22
3.2.7 JavaScript	23
3.2.8 XML	23
3.2.9 Bases de datos	23
3.2.10 Desarrollo móvil	23
3.2.10.1 Phonegap Build	24
4. APLICACIÓN DE LA METODOLOGIA.....	25
4.1 Definición del problema	25
4.2 Definición de los objetivos.....	26
4.2.1 Modelo de dominio	27
4.2.1.1 Glosario	27
4.2.1.2 Diagrama conceptual del modelo de dominio.....	28
4.2.1.3 Modelo BPMN	29
4.3 Diseño y desarrollo.....	31
4.3.1 Criterios de entrada.....	32
4.3.1.1 Descripción del problema	32
4.3.1.2 Formulario de registro de tiempo("PSP0 Time Recording Logs").....	33

4.3.1.3 Formulario de registro de defectos("PSP0 Defect Recording Logs")	33
4.3.2 Actividades	33
4.3.2.1 Planeación	33
4.3.2.1.1 Criterios de entrada.....	34
4.3.2.1.2 Especificación de requerimientos (Program Requirements)	34
4.3.2.1.2.1 Requerimientos Funcionales	34
4.3.2.1.2.1.1 Actores	34
4.3.2.1.2.1.2 Casos de Uso.....	35
4.3.2.1.2.1.2.1 Caso de uso general.....	35
4.3.2.1.2.1.2.2 Caso de uso Publicar Recorrido	36
4.3.2.1.2.1.2.3 Caso de uso Cancelar Recorrido.....	36
4.3.2.1.2.1.2.4 Caso de uso Cancelar Reserva de Persona en su Recorrido	37
4.3.2.1.2.1.2.5 Caso de uso Cancelar Recorrido.....	37
4.3.2.1.2.1.2.6 Caso de uso Cancelar Cupo Reservado.....	38
4.3.2.1.2.1.3 Requerimientos No Funcionales	38
4.3.2.1.3 Estimación de recursos	39
4.3.2.1.4 Criterios de salida	40
4.3.2.2 Desarrollo	40
4.3.2.2.1 Criterios de entrada.....	41
4.3.2.2.2 Diseño	41
4.3.2.2.2.1 Arquitectura de la solución	42
4.3.2.2.2.2 Diseño de la interfaz	43
4.3.2.2.2.2.1 Interfaz Inicio de Sesión	43
4.3.2.2.2.2.2 Interfaz formulario Cambiar Contraseña	43
4.3.2.2.2.2.3 Interfaz para Publicar Recorrido	45
4.3.2.2.2.2.4 Interfaz Buscar Recorrido	45
4.3.2.2.2.2.5 Interfaz Reservar Cupo	46
4.3.2.2.2.2.6 Interfaz Cancelar Cupo Reservado	46
4.3.2.2.2.2.7 Interfaz Cancelar Recorrido	47
4.3.2.2.2.2.8 Interfaz Cancelar Reserva de Persona en su Recorrido	47
4.3.2.2.2.3 Modelo Estático.....	48
4.3.2.2.2.4 Modelo de datos (Modelo Entidad Relación)	49
4.3.2.2.2.5 Modelo de Procesos	51
4.3.2.2.3 Codificación	51
4.3.2.2.3.1 Paquete Entidades.....	53
4.3.2.2.3.2 Paquete DAO	54
4.3.2.2.3.3 Paquete Negocio	55
4.3.2.2.3.4 Paquete Presentación.....	56
4.3.2.2.3.4.1 Webapp	57
4.3.2.2.4 Compilación	58
4.3.2.2.5 Pruebas.....	59
4.3.2.2.6 Pruebas de funcionabilidad del sistema	59
4.3.2.2.6.1 CP001. Caso de prueba Publicar Recorrido	59
4.3.2.2.6.2 CP002. Caso de prueba Cancelar Recorrido	66
4.3.2.2.6.3 CP003. Caso de prueba Buscar Recorrido	68
4.3.2.2.6.4 CP004. Caso de prueba Reservar Cupo	70
4.3.2.2.6.5 CP005. Caso de prueba Cancelar Cupo Reservado	72
4.3.2.2.6.6 CP006. Caso de prueba Cancelar Reserva de Persona en su Recorrido	73
4.4. Demostración.....	75

4.5 Evaluación.....	76
4.6 Comunicación.....	77
5. RESULTADOS.....	78
5.1 Relacionados con la generación de conocimiento y/o nuevos desarrollos tecnológicos	78
5.2 Conducentes al fortalecimiento de la capacidad científica institucional o nacional	78
5.3 Dirigidos a la apropiación social del conocimiento	78
6. CONCLUSIONES Y RECOMENDACIONES	79
6.1 Conclusiones desde el dominio del negocio.....	79
6.2 Conclusiones desde la ingeniería de software.....	79
6.3 Recomendaciones	80
7. REFERENCIAS BIBLIOGRÁFICAS	81
8. ANEXOS.....	86

LISTADO DE TABLAS

Tabla 1. Cronograma de actividades.....	39
Tabla 2. Caso de prueba Registro y publicación de Recorrido.....	60
Tabla 3. Caso de prueba anulación de Recorrido.....	66
Tabla 4 Caso de prueba búsqueda de Recorrido.....	68
Tabla 5 Caso de prueba reserva de cupo	69
Tabla 6 Caso de prueba cancelación de cupo reservado.....	72
Tabla 7 Caso de prueba cancelación de reserva de persona en su recorrido	73
Tabla 8 Generación de nuevo conocimiento.....	78
Tabla 9 Fortalecimiento de la comunidad.....	78
Tabla 10 Apropiación social del conocimiento.....	78

LISTADO DE FIGURAS

Figura 1. Encuesta Comunidad Unillanos	14
Figura 2. Sistema Férreo en Santiago de Chile. Tomada de [4].....	19
Figura 3. Arquitectura de JavaEE basada en tres capas. Imagen tomada de presentación de [19].....	22
Figura 4 . Tecnologías Web, PhoneGap Build y Sistemas Operativos. Tomada de [27].	24
Figura 5. Proceso de aplicación de la metodología DSRM. Tomada de [31].	25
Figura 6. Fase de definición del problema. Adaptada de [31].	26
Figura 7. Fase de definición de los objetivos. Adaptada de [31].	26
Figura 8. Imagen tomada del documento glosario anexo 8.1.....	28
Figura 9. Diagrama Conceptual General	29
Figura 10. Modelo BPMN del proceso A Villavo con vehículo.....	30
Figura 11. Modelo BPMN del proceso A Villavo sin vehículo	30
Figura 12. Fase de diseño y desarrollo. Adaptada de [31]......	31
Figura 13. Creación del proyecto en Process DashBoard.....	31
Figura 14. PSP0 Process Script.....	32
Figura 15. PSP0 Time Recording Logs.Tomada del Dashboard.....	33
Figura 16. PSP0 Defect Recording Logs.Tomada del Dashboard	33
Figura 17. PSP0 Planning Script.....	34
Figura 18. Actividad de especificación de los requerimientos	34
Figura 19. Actores del Sistema	35
Figura 20. Caso de uso general	36
Figura 21. Caso de uso Publicar Recorrido.....	36
Figura 22. Caso de uso Cancelar Recorrido	37
Figura 23. Caso de uso Cancelar Reserva de Persona en su Recorrido	37
Figura 24. Caso de uso Buscar Recorrido	38
Figura 25. Caso de uso Cancelar Cupo Reservado	38
Figura 26. PSP0 Planning Script.....	39
Figura 27. Criterios de salida PSP0 Planning Script	40
Figura 28. PSP0 Development Script.....	40
Figura 29.Actividades del diseño.....	41
Figura 30. Estructura especificación de diseño PSP Tomada de [37].....	42
Figura 31. Estructura diseño de la aplicación.....	42
Figura 32. Arquitectura de la aplicación	43
Figura 33. Mockup Inicio de Sesión	44
Figura 34. Mockup formulario Cambiar Contraseña.....	44
Figura 35. Mockup Publicar Recorrido	45
Figura 36. Mockup Buscar Recorrido	46
Figura 37. Mockup configurara la convocatoria.....	46
Figura 38. Mockup Cancelar Cupo Reservado.....	47
Figura 39. Mockup Cancelar Recorrido.....	47
Figura 40. Mockup Cancelar Reserva de Persona en su Recorrido.....	48
Figura 41. Diagrama de clases.....	49

Figura 42. Modelo Entidad Relación	50
Figura 43. Modelo BPMN del proceso Ir a Villavo con Automóvil TOBE	51
Figura 44. Modelo BPMN del proceso Ir a Villavo con Automóvil TOBE	51
Figura 45. Actividades de la codificación	52
Figura 46. Estructura de carpetas del proyecto	53
Figura 47. Clases Entidad	53
Figura 48. Código fuente clase Recorrido.java	54
Figura 49. Clases DAO	54
Figura 50. Código fuente clase RecorridoDAO.java	55
Figura 51. Clases de negocio	55
Figura 52. Código fuente clase iGestionBean.java	56
Figura 53. Código fuente clase GestionBean.java	56
Figura 54. Clases de presentación	57
Figura 55. Código fuente clase LoginModel.java	57
Figura 56. Recursos web	58
Figura 57. Código fuente página login.xhtml	58
Figura 58. Actividades de la compilación	59
Figura 59. Actividades para la fase de pruebas	59
Figura 60. Página de inicio	¡Error! Marcador no definido.
Figura 61. Datos de prueba	61
Figura 62. Preferencias	62
Figura 63. Recorrido listo para publicar	62
Figura 64. Recorrido creado	63
Figura 65. Punto origen coincide con destino	63
Figura 66. Error por puntos de origen y destino iguales	64
Figura 67. Hora de ida mayor hora de vuelta	64
Figura 68. Error por hora de ida mayor que hora vuelta	65
Figura 69. No ingreso de un paradero	65
Figura 70. Error por no ingreso de un paradero	65
Figura 71. Perfil Persona con Carro	66
Figura 72. Recorridos activos	67
Figura 73. Ingreso datos de búsqueda	67
Figura 74. Recorrido cancelado	67
Figura 75. Datos de prueba ingresados	69
Figura 76. Recorrido encontrado	69
Figura 77. Recorrido reservado	70
Figura 78. Error al no seleccionar un paradero	71
Figura 79. Error cuando hay una reserva en un recorrido y se intenta volver a reservar	71
Figura 80. Recorrido activo	73
Figura 81. Cupo de reserva cancelado	73
Figura 82. Recorrido activo	74
Figura 83. Cupo de reserva cancelado	75
Figura 84. Fase de demostración. Adaptada de [38]	75
Figura 85. Instrumento de seguimiento	76

Figura 86. Fase de demostración. Tomada de [38].....	76
Figura 87. Fase de comunicación. Tomada de [38]	77

1. INTRODUCCION

1.1. Tema

Desarrollo de una aplicación web que contribuya a mitigar la problemática en el transporte desde y hacia la Universidad, utilizando las metodologías Design Science Research (DSR) en el proceso investigativo y Personal Software Process (PSP) en lo que tiene que ver con el desarrollo de software y la tecnología Java Enterprise Edition (JavaEE) en la construcción de la aplicación empresarial en capas.

1.2. Descripción del problema

Estudiantes, profesores y personal administrativo de la Universidad de Los Llanos afrontan dificultades en el transporte entre Villavicencio y la universidad, por diferentes razones. Por un lado, aunque la universidad ofrece rutas de buses, éstas presentan varios inconvenientes, entre los que se cuentan:

- No cubren toda la ciudad
- Los horarios son limitados.
- Algunos horarios presentan sobrecupo en los buses.
- Algunas rutas incumplen los horarios establecidos.

De otra parte, las rutas de servicio público son escasas, por lo que tardan mínimo media hora entre una y otra y finalmente el servicio de taxis tiene un recargo por ser rural, de \$2,000, lo que hace que el recorrido mínimo para llegar o salir de la universidad cueste \$9,000.

En una encuesta recientemente realizada a 300 personas distribuidas en estudiantes, profesores y administrativos de la universidad (Ver Figura 1), se obtuvo que el 91% están de acuerdo con el desarrollo del aplicativo para compartir el carro entre miembros de la universidad. Anexo B: **Encuesta**



Figura 1. Encuesta Comunidad Unillanos

Finalmente, las motocicletas y automóviles particulares se convierten en la forma más efectiva de movilizarse a y desde la universidad; sin embargo, un número significativo de personas de la comunidad universitaria no cuenta con estos vehículos y las personas que sí los tienen cuentan con 3 o 4 espacios disponibles que eventualmente podrían ser compartidos con aquellas personas que requieren el servicio, siendo esta solución un beneficio y una ventaja para toda la comunidad universitaria. Esta alternativa requiere de una solución de software que permita que los propietarios de vehículos puedan compartir sus asientos disponibles en su recorrido habitual, con personas de la misma comunidad, que así lo requieran.

Con este trabajo se investigó, siguiendo el enfoque “Design Science Research” (la ciencia del diseño), una solución informática que permita a los propietarios de vehículos poder compartir sus asientos disponibles en su recorrido habitual, con personas de la misma comunidad, que así lo requieran, siendo esta solución un beneficio y una ventaja para todos los miembros de la Universidad de los Llanos.

1.3. Justificación

Cada estudiante, profesor o administrativo de la Universidad de los Llanos, al momento de dirigirse a su lugar de destino (ya sea Unillanos o Villavicencio), se enfrenta a una serie de inconvenientes: Rutas que no cubren toda la ciudad, horarios reducidos, sobrecupo en los buses en algunos horarios, demoras en la salida de algunas rutas, etc. De otra parte, las rutas de servicio público son escasas, por lo que tardan mínimo media hora, entre una y otra; finalmente, el servicio de taxis tiene un recargo por ser rural, de \$2,000, lo que hace que el recorrido mínimo para llegar o salir de la universidad cueste \$9,000. Esta situación, que ocurre a diario en la Universidad de los Llanos, muchas veces genera a los actores que son los estudiantes, profesores y personal administrativo de la Universidad, llegar tarde al punto de destino, viéndose comprometido el horario de llegada y las labores de cada uno. Esto se convierte en una problemática con la cual estas personas conviven a diario.

Hay personas de la comunidad universitaria, propietarios de vehículos, que están dispuestas a compartir sus cupos disponibles con las personas que así lo requieren, sin embargo, para que esto suceda, se requiere un mecanismo formal y controlado, a través del cual se pueda publicar oferta de cupos, especificando la ruta, horario y sitios de parada, entre otros, y con el cual los interesados puedan reservar puesto. Gracias a las tecnologías de la información actuales, es posible ofrecer una solución a través de un aplicativo web en el que el estudiante, profesor o administrativo asegura un cupo en la ruta o trayecto que desea, de una oferta publicada por un propietario de vehículo. Es oportuno precisar que la solución debe garantizar condiciones de seguridad para que el acceso esté limitado a los miembros de la comunidad universitaria, ya sea estudiantes, profesores o administrativos.

Un ejemplo de una solución de este estilo puede evidenciarse en la Universidad de los Andes, ubicada en el centro de Bogotá, ciudad que por ser capital sufre a diario

problemas de movilidad. Profesores y estudiantes de esta universidad crearon la aplicación para compartir el carro, llamada Portal Viaje, dirigida únicamente a integrantes de la comunidad Uniandina [2], que promueve compartir el viaje, desde y hacia la Universidad a través de la plataforma, la cual le permite publicar un viaje, compartir puestos, ponerse en contacto con los pasajeros e incluso recibir notificaciones [3]. Adicionalmente, la aplicación busca impactar con beneficios ambientales, reduciendo las emisiones de CO₂.

Dado que en Unillanos se aborda el desarrollo de software y se cuenta con una línea de investigación en Ingeniería de Software, es posible construir un aplicativo orientado a la web, que utilice las tecnologías emergentes y las metodologías propuestas en el programa, que además de permitir avanzar en conocimiento de dichas tecnologías, ofrezca una solución al problema de movilidad desde y hacia la Universidad de los Llanos.

2. OBJETIVOS

2.1. Objetivo general

Desarrollar un aplicativo que permita a la comunidad de Unillanos compartir los asientos disponibles en los vehículos particulares, para transportarse desde y hacia la universidad.

2.2. Objetivos específicos

- Ofrecer a la comunidad de la Universidad de los Llanos, un medio seguro, ágil y sencillo para compartir un recurso crítico como lo es el transporte desde y hacia la universidad.
- Ampliar la oferta de transporte para la comunidad de la Universidad, desde y hacia la Universidad, a través del uso de un aplicativo en la web o en dispositivos móviles.
- Desarrollar habilidades para la construcción de aplicaciones cuya capa de presentación sea web y móvil sobre Android.
- Conocer y aplicar las estrategias requeridas para cumplir con los requerimientos no funcionales de seguridad y concurrencia que requiere una aplicación como la que se construirá para compartir el carro.

3. MARCO TEÓRICO

Este trabajo contempla dos saberes disciplinares, por una parte los sistemas de transporte y por otra la ingeniería de software en lo que tiene que ver con el desarrollo de software orientado a la web y las tecnologías disponibles. A continuación se esbozan cada uno de estos aspectos.

3.1. Sistemas de transporte e iniciativas en el mundo

A continuación se contextualiza el tema del transporte: Se describe el sistema masivo en Latinoamérica, el resto del mundo y en Colombia, para luego citar otras iniciativas de “compartir el carro”.

3.1.1 Transporte masivo en Latinoamérica

El transporte público en América Latina, se ha caracterizado por los continuos esfuerzos por mejorar la cobertura y calidad de servicio. Partiendo del éxito en varios países del mundo, se empezó a implementar el modelo alternativo de transporte basado en sistemas férreos para contrarrestar las características e impactos negativos del transporte público tradicional, predominantemente, en ciudades capitales. Los sistemas férreos construidos en América Latina se remontan desde 1913, año en el cual Buenos Aires lo implementó, siendo el primero en hacerlo, seguido por México en 1968, Sao Paulo en 1974, Santiago de Chile en 1975, Rio de Janeiro en 1979, Caracas en 1983 y Medellín en 1995. Todos estos mejoraron sustancialmente la calidad de prestación del servicio en estas ciudades. Santiago de Chile, cuenta con uno de los sistemas férreos más exitosos de América Latina. Figura 2 [4].

Actualmente, junto a este sistema férreo, se encuentra el sistema denominado BRT o Bus Rápido (en inglés, *Bus Rapid Transit*) que ha sido de gran ayuda para mitigar el transporte atomizado y desorganizado en las principales ciudades de Latinoamérica. El BRT consiste en un carril exclusivo para los buses, copiando ciertas características de los sistemas ferros, como lo son el pago de pasajes en estaciones determinadas, programación de las rutas de los buses, infraestructura exclusiva, estaciones como elemento central del sistema, entre otras. *“Una diferencia crucial con los sistemas de tipo férreo que se construyeron anteriormente, es que el sistema de Bus Rápido ha buscado integrar los operadores existentes de transporte público, al tiempo que busca reemplazar el sistema tradicional por el de bus rápido. En gran parte de los casos de soluciones férreas los sistemas de transporte público basados en buses se mantuvieron intactos, y su prestación de servicio fue igual a la que habían tenido antes”* [4].



Figura 2. Sistema Férreo en Santiago de Chile. Tomada de [4].

3.1.1 Contexto del sistema de transporte masivo en Colombia

En la actualidad, Colombia está rezagada en el crecimiento en lo relacionado a los sistemas de transporte masivo, lo cual se evidencia en ciudades como la capital Bogotá, que sufre diariamente problemas de movilidad debido a un mal diseño en el modelo de transporte implementado después de 1948 cuando sucedió el famoso “*Bogotazo*”, momento en que se tomó la decisión por parte del estado de apostarle a un sistema de buses colectivos que por esos tiempos era el *boom* en las principales ciudades del mundo, y en ningún momento se contempló una solución mixta, es decir en la cual además de la inclusión de buses colectivos intervinieran el tren, el metro y tranvía. Este modelo lleva 68 años y en ha generado algunos efectos nocivos: polución (contaminación del medio ambiente), trancones, huecos en las calles, precios altos, piratería, conductores irresponsables, muchas paradas en la misma cuadra, buses en notable condición de deterioro, ruido, baja cobertura en los extremos de la ciudad, entre otros; todo lo anterior es la consecuencia natural de tener un modelo mal diseñado [5].

A pesar de que en Bogotá se implementó el sistema de transporte masivo Transmilenio, éste ha sido insuficiente para satisfacer el flujo de personas que permanentemente utilizan este medio, (hay dificultades dadas por la complejidad de las rutas, se evidencia congestión en las estaciones, el precio de los pasajes genera inconformidad y, en general, es manifiesta la ineficiencia de este sistema.) a lo que se suma la falta de coordinación entre el gobierno nacional y el alcalde de la ciudad, lo que ha impedido la toma de decisiones que permitan brindar las soluciones requeridas para un sistema de transporte con la cobertura para todos los lugares de la capital, en condiciones adecuadas. En contraposición a lo sucedido en la capital, la ciudad de Medellín es una de las pioneras en implementar un servicio de transporte integrado con un Metro, un Metrocable y últimamente el Tranvía, lo que convierte a la ciudad en líder a nivel nacional toda vez que han logrado llevar el servicio de transporte a la mayoría de los lugares de la ciudad, en mejores condiciones que los demás sistemas del país [6].

Por otra parte, hay ciudades que han venido buscando mecanismos para satisfacer las necesidades de los usuarios como la ciudad de Cali a través del MIO, en el eje cafetero Pereira mediante el Megabus, en el centro-oriente del país por medio del Metrolínea de Bucaramanga y finalmente en el caribe colombiano con el uso del Transmetro de Barranquilla y Transcaribe de Cartagena; sin embargo las medidas son insuficientes frente al crecimiento de las ciudades y sus habitantes que cada vez se ven más afectados por la escasez de transporte.

3.1.2 El transporte en la Universidad de los Llanos

Según la *Oficina de Admisiones Registro y Control Académico* de la Universidad de los Llanos, la cantidad de estudiantes, profesores y personal administrativo está sobre los 6255 aproximadamente [7] [8] [9], para los que se cuenta con la siguiente oferta de transporte público: a) servicio que brinda la universidad con 19 buses, con capacidad de 40 personas, distribuidos en 10 rutas. Este servicio es contratado con la empresa *American Tour S.A.S.* b) servicio de transporte público de busetas, que por la ubicación veredal (Vereda Barcelona, a 12 km. de Villavicencio), tiene una frecuencia de 30 minutos, tanto desde y hacia la universidad.

3.1.3 Otras iniciativas de compartir el carro (“Car Sharing”).

La iniciativa de compartir el carro es de amplia trayectoria a nivel mundial, con resultados de investigación que miden los beneficios ambientales y de movilidad, así como las posibilidades que esta estrategia puede representar para comunidades enteras, de ser utilizadas o promovidas a través de iniciativas gubernamentales, es decir, que hagan parte de las políticas de transporte de un país. Entre las publicaciones realizadas al respecto, se puede dar cuenta de autores que han estudiado los casos en Sidney-Australia, el prof. Robyn Dowling y la prof. Jennifer Kent [10], Lisbon-Portugal, la dr. Patrícia Baptist, la dra. Sandra Melo y la est. Catarina Rolim [11], Austria, el dr. KarlW. Steininger y el doc. Gabriel Bachner [12], Singapore, el dr. Foo Tuan Seik [13].

3.2 Contexto de Ingeniería de Software y tecnologías disponibles.

Para ofrecer una solución de software a la problemática planteada y que ésta sea una solución de calidad, se requiere que su desarrollo sea guiado por la ingeniería de software la cual se define más adelante. De igual forma, es importante reconocer cuáles son las tecnologías disponibles y que pueden aprovecharse en este proyecto: Java, Html5, Css3, JavaScript, Xml y bases de datos. También es importante recalcar la importancia del desarrollo móvil como complemento en este tipo de aplicaciones web.

3.2.1 Ingeniería de software

Este concepto ha venido evolucionando con el tiempo y para este trabajo adoptamos el concepto acuñado por Presman [14], a partir de Bohem [15]: *“Ingeniería de software es la aplicación práctica del conocimiento científico en el diseño y construcción de*

programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software”.

3.2.2 Aplicaciones Web

Las Aplicaciones Web son una categoría de software informático eminentemente de red, las gobiernan los datos y se encuentran en evolución continua. La inmediatez que dirige su desarrollo, la necesidad apremiante de seguridad en su operación y la demanda de estética, así como la entrega de contenido funcional son factores diferenciales adicionales. Al igual que otros tipos de software, las aplicaciones web pueden valorarse mediante una diversidad de criterios de calidad que incluyen facilidad de uso, funcionalidad, escalabilidad y tiempo para comercialización [16].

3.2.3 Java

“Java es un lenguaje de programación orientado a objetos que se incorporó al ámbito de la informática en los años noventa. La idea de Java es que pueda realizarse programas con la posibilidad de ejecutarse en cualquier contexto, en cualquier ambiente, siendo así su portabilidad uno de sus principales logros. Fue desarrollado por Sun Microsystems, posteriormente adquirido por Oracle” [17]. Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa; actualmente se cuentan más de 9 millones de desarrolladores en todo el mundo (traducido de [17]).

3.2.4 Java EE

“La plataforma Java Enterprise Edition (Java EE) son un conjunto de especificaciones que facilitan el desarrollo y despliegue de aplicaciones empresariales multi-capas. Java EE ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones de múltiples niveles de funcionalidad basadas en servidores. Se reduce el costo y complejidad de desarrollo, lo cual resulta en servicios que se pueden desplegar y extender fácilmente” [18].

3.2.4.1 Arquitectura Java EE

Una aplicación Java EE típica presenta una solución basada en una arquitectura de 3 capas, ilustradas en la Figura 3:

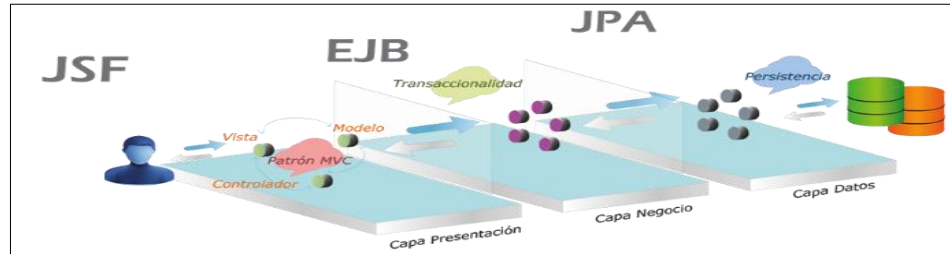


Figura 3. Arquitectura de JavaEE basada en tres capas. Imagen tomada de presentación de [19]

A continuación se resume la definición de las tres capas, tal como lo expone [20]

- Presentación (capa de usuario)

Esta capa hace referencia a lo que manipula el cliente en su computador, a través de un browser. Es la interfaz gráfica del sistema y se encarga de interactuar con el usuario. La capa de presentación se puede construir con JSF, que es una tecnología que fue diseñada para simplificar la construcción de interfaces de usuario para aplicaciones web java.

- Control (capa de negocio)

Esta capa hace referencia a las reglas que deben cumplirse en el sistema, en esta capa se reciben las peticiones del usuario y se envían respuestas tras uno o varios procesos. La capa de control se elabora con la tecnología EJB. A pesar de que está comunicada con la capa de Presentación y la capa de Persistencia, no tiene una conexión directa con éstas, es decir los cambios realizados en la capa de datos no afectan a la capa de negocio; de igual manera sucede con la capa de Usuario.

- Persistencia (capa de datos)

Esta capa hace referencia a los datos manejados en el sistema, se utiliza para este proyecto JPA. Esta capa debe ser capaz de abstraer por completo el mecanismo de persistencia de la primera capa y no debe contener ninguna lógica de negocio [21].

3.2.5 Html5

HTML5 es un conjunto de estándares para diseñar y desarrollar páginas web. Este conjunto representa la forma como se muestra la información en el browser y también la forma como interactuar con ella. Además, posibilita una mejor interacción tanto entre páginas web como de contenido media, es decir, audio, video, entre otros. HTML5 permite que la codificación del diseño básico sea más simple y sencilla. [22]

3.2.6 Css

Mientras que HTML5 permite la maquetación o estructura de la página web, CSS (Cascading Style Sheets) permite el estilo y personalización de ésta en diferentes

dispositivos, como computadores de escritorio, portátiles, tablets, celular o cualquier otro medio que pueda mostrar contenido web. Una buena administración de CSS por parte de los desarrolladores, les permite un completo control total sobre el estilo y formato de las páginas web [23].

3.2.7 JavaScript

Javascript es un lenguaje de programación utilizado por quienes desarrollan y diseñan páginas web. Javascript no necesita compilación alguna, ya que se encuentra alojado en un documento HTML, es decir del lado del cliente; por tanto al ejecutar una determinada página web, se está ejecutando inmersamente Javascript; sin la necesidad de instalar otro programa para ser visualizado. Mientras que otros lenguajes de programación si requieren una máquina virtual o un determinado programa para ser ejecutados. Existen dos tipos de Javascript: Por un lado se encuentra el Javascript ejecutado del lado del cliente, denominado también Navigator JavaScript. También existe un Javascript que es ejecutado del lado del servidor, éste recibe el nombre de LiveWire Javascript [24].

3.2.8 XML

“XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información” [25]

3.2.9 Bases de datos

“Una base de datos es una colección de archivos relacionados que permite el manejo de la información de alguna compañía. Cada uno de dichos archivos puede ser visto como una colección de registros y cada registro está compuesto de una colección de campos. Cada uno de los campos de cada registro permite llevar información de algún atributo de una entidad del mundo real” [26].

Una base de datos es administrada a través de un gestor de base de datos, que es el grupo de herramientas con las que se puede crear la base de datos, gestionar las tablas, los roles, registros y relaciones entre éstos. El gestor de bases de datos que se utilizará en este proyecto es Oracle, que es con el que se gestiona la información por el SIIF (Sistema Integrado de Información Financiera), donde se encuentra la información pertinente de la comunidad que podrá utilizar el aplicativo.

3.2.10 Desarrollo móvil (Phonegap)

Es un framework para desarrollo móvil creado por Nitobi y adquirido por Adobe. Facilita la creación de aplicaciones multiplataforma utilizando tecnologías como HTML5 y CSS3. Ofrece diferentes apis para controlar recursos como la cámara, el acelerómetro, el GPS, las notificaciones, el almacenamiento, entre otros. Soporta desarrollo para Android, iOS, BlackBerry OS, Windows Phone, Web OS, Symbian y Bada [27].

3.2.10.1 Phonegap Build

Phonegap Build es un servicio que se encuentra alojado en la nube y lo brinda adobe, al cual se puede acceder a través de una cuenta adobe, permitiendo un archivo por medio de una cuenta en GitHub o ya sea un archivo .zip. En la actualidad la mejor herramienta a la hora de desarrollar aplicaciones multiplataforma es Phonegap, ya que es un framework gratuito y cuenta con una documentación bastante amplia y una suit de herramientas para creación y desarrollo de aplicaciones [27]. La Figura 4 describe este funcionamiento.

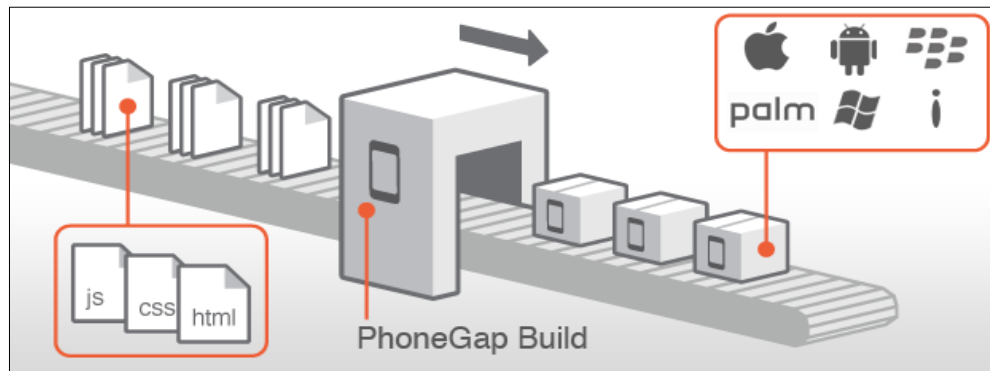


Figura 4 . Tecnologías Web, PhoneGap Build y Sistemas Operativos. Tomada de [27].

4. APLICACIÓN DE LA METODOLOGIA

Este trabajo sigue los lineamientos del marco de referencia para la investigación en sistemas de información propuesto por Hevner A y otros [28], que hace parte de la investigación en las ciencias de lo artificial, tal como lo plantea Herbert A Simon [29]. Consecuentemente, la metodología general a aplicar es la de la Investigación del Diseño descrita por Vaishnavi y Kuechler [30].

La aplicación de la Metodología de la Investigación de Diseño (DSRM por sus siglas en inglés Design Science Research Metodology) se realizará siguiendo el modelo de proceso sugerido por Peffers y otros [31], que se muestra en la Figura 5. DSRM abarca el proceso de desarrollo de software desde las etapas de identificación y definición del problema, hasta la publicación profesional de resultados, incluyendo un proceso de diseño y desarrollo de artefactos para el que se usará PSP, ya que provee los mecanismos necesarios para la construcción de los artefactos de software. El proceso de desarrollo de este proyecto se llevara a cabo usando la metodología PSP-TSP, que entrega a los ingenieros un marco de referencia de disciplina personal para mejorar su trabajo y realizarlo con alta calidad, con el propósito de ayudarlos a aprender y practicar aquellos métodos para producir software que son más efectivos para ellos. Entendiendo como principio fundamental que con un proceso de calidad los productos derivados de éste serán también de calidad [32].

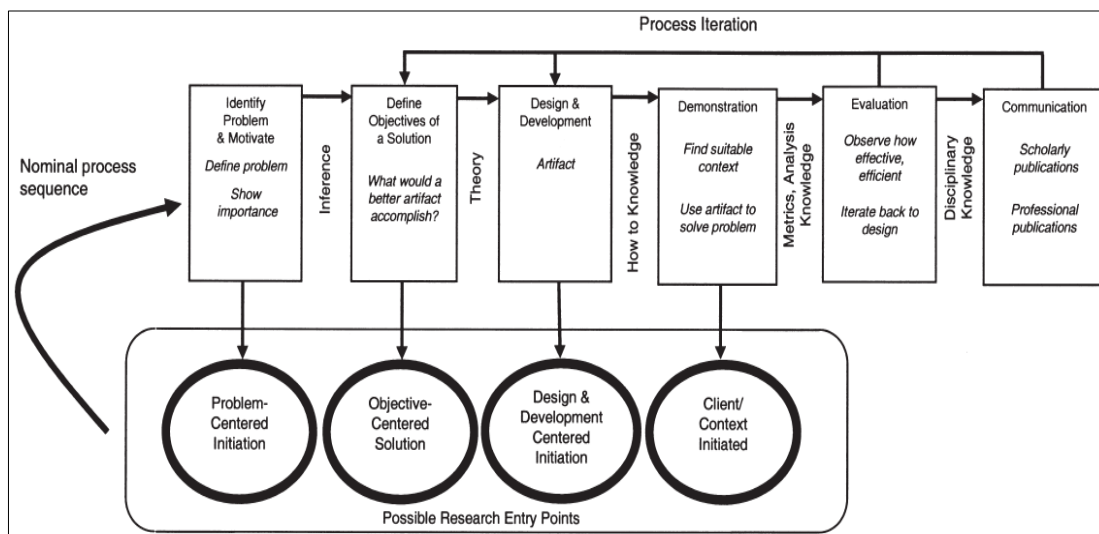


Figura 5. Proceso de aplicación de la metodología DSRM. Tomada de [31].

A continuación se desglosan los anteriores pasos:

4.1. Definición del problema

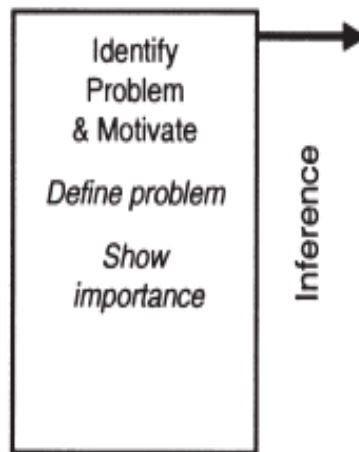


Figura 6. Fase de definición del problema. Adaptada de [31].

Debido a las evidentes dificultades (No cubrimiento total de la ciudad, horarios limitados, algunos horarios presentan sobrecupo en los buses, algunas rutas incumplen horarios establecidos, entre otros.) que afrontan los estudiantes, profesores y personal administrativo en el transporte entre Villavicencio y la universidad, se logra identificar y definir el problema.

Por esta razón se hace necesario desarrollar una aplicación web que contribuya a mitigar la problemática en el transporte desde y hacia la Universidad. Permitiendo a los propietarios de vehículos poder compartir sus asientos disponibles en su recorrido habitual, con personas de la misma comunidad, que así lo requieran, siendo esta solución un beneficio y una ventaja para todos los miembros de la Universidad de los Llanos.

4.2. Definición de los objetivos

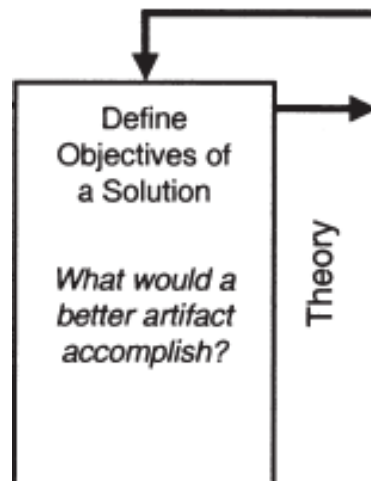


Figura 7. Fase de definición de los objetivos. Adaptada de [31].

En relación a la definición del problema se estableció como objetivo general Desarrollar un aplicativo que permita a la comunidad de Unillanos compartir los asientos disponibles

en los vehículos particulares, para transportarse desde y hacia la universidad.

Por su parte para lograr el objetivo general se deben cumplir los siguientes objetivos específicos:

- Ofrecer a la comunidad de la Universidad de los Llanos, un medio seguro, ágil y sencillo para compartir un recurso crítico como lo es el transporte desde y hacia la universidad.
- Ampliar la oferta de transporte para la comunidad de la Universidad, desde y hacia la Universidad, a través del uso de un aplicativo en la web o en dispositivos móviles.
- Desarrollar habilidades para la construcción de aplicaciones cuya capa de presentación sea web y móvil sobre Android.
- Conocer y aplicar las estrategias requeridas para cumplir con los requerimientos no funcionales de seguridad y concurrencia que requiere una aplicación como la que se construirá para compartir el carro.

La actividad desarrollada para esta fase fue el “levantamiento de requerimientos” [33] en la que se construyeron los siguientes artefactos:

- Glosario
- Mapa Conceptual
- Modelo de procesos en BPMN.

Con los cuales se logró una comprensión del modelo de dominio, para definir la estructura y dinámica de la aplicación a diseñar comprendiendo de manera general el proceso de negocio a intervenir.

En el BPMN, se identificaron las actividades a ser sistematizadas en el proyecto con miras a mejorar dicho proceso, involucrando la TI JavaEE.

A continuación se detallan los artefactos elaborados:

4.2.1. Modelo de dominio

4.2.1.1. Glosario

El documento glosario es una representación de los términos manejados a lo largo de todo el proceso de compartir el carro, que tiene como propósito definir con exactitud y sin ambigüedad la terminología manejada de tal modo que se cree un lenguaje común entre el cliente y el desarrollador, a continuación se puede ver en la Figura 8 algunos de los conceptos tratados, para ver en detalle todas las definiciones ver el documento glosario anexo 8.2.

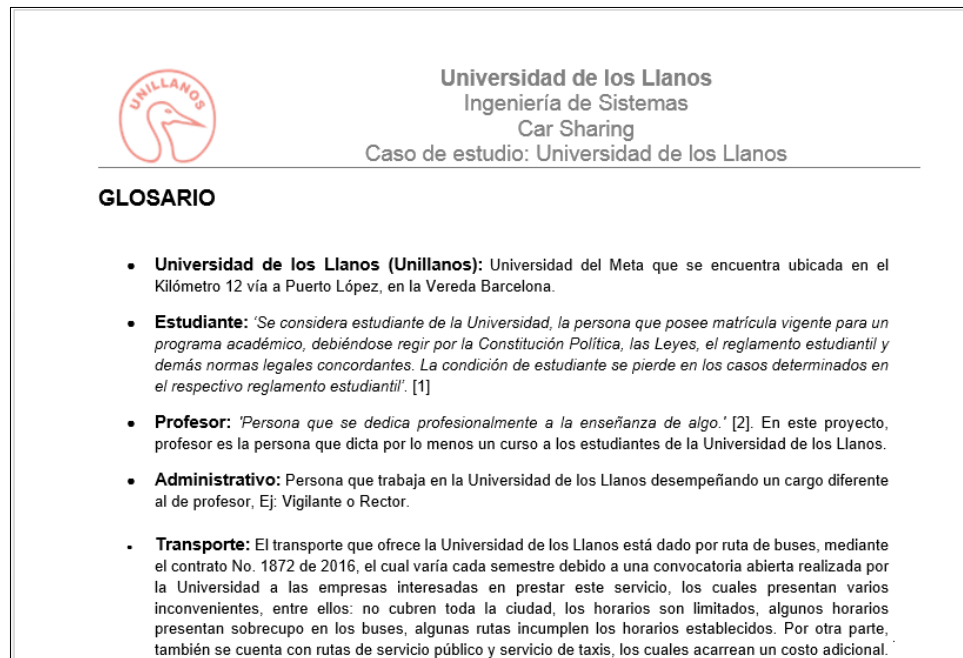


Figura 8. Imagen tomada del documento glosario anexo 8.2

4.2.1.2. Diagrama conceptual del modelo de dominio

El siguiente artefacto generado fue el diagrama conceptual en el cual se organiza y se representa gráficamente la jerarquía y relación que existe entre las definiciones encontradas sobre el tema de compartir el carro entre miembros de la comunidad universitaria; la Figura 9 muestra el diagrama en detalle.

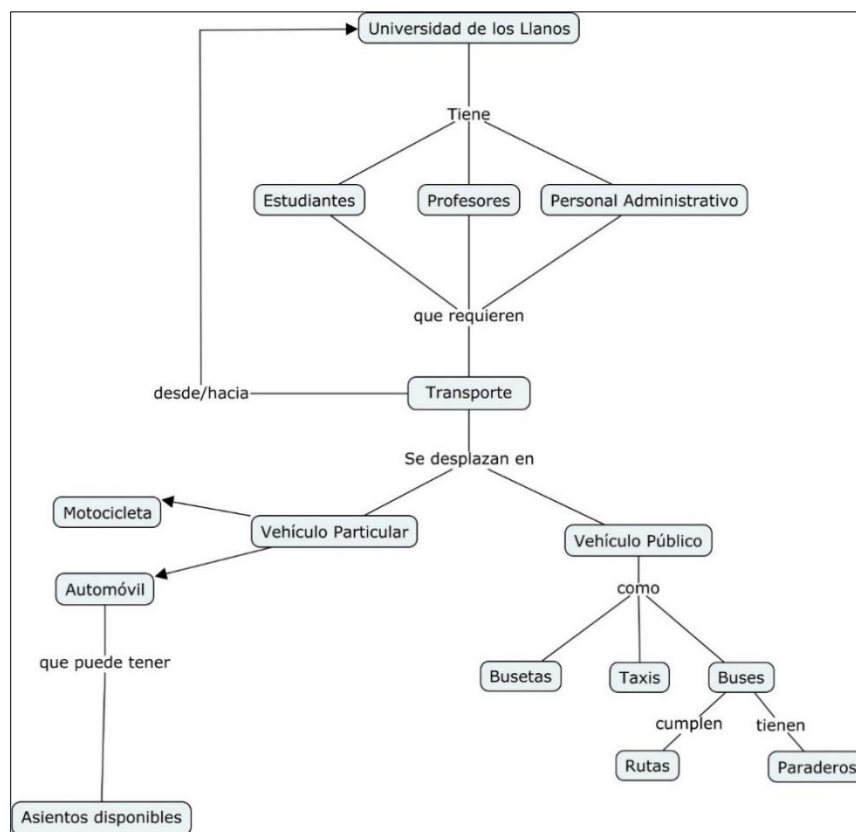


Figura 9. Diagrama Conceptual General

4.2.1.3. Modelo BPMN

Como último artefacto generado para la representación del modelo de dominio se encuentra el modelo BPMN (Notación para el Modelado de Procesos de Negocio), en él se describen los usuarios involucrados y el flujo de actividades desde el inicio hasta finalizar el proceso de compartir el carro. Se contemplan dos procesos, que se describen a continuación:

- Usuarios involucrados en el proceso A Villavo con vehículo:

En la Figura 10 se puede identificar los dos usuarios involucrados en el proceso, representados de manera horizontal:

- I. Persona Unillanos sin Vehículo
- II. Persona Unillanos con Vehículo

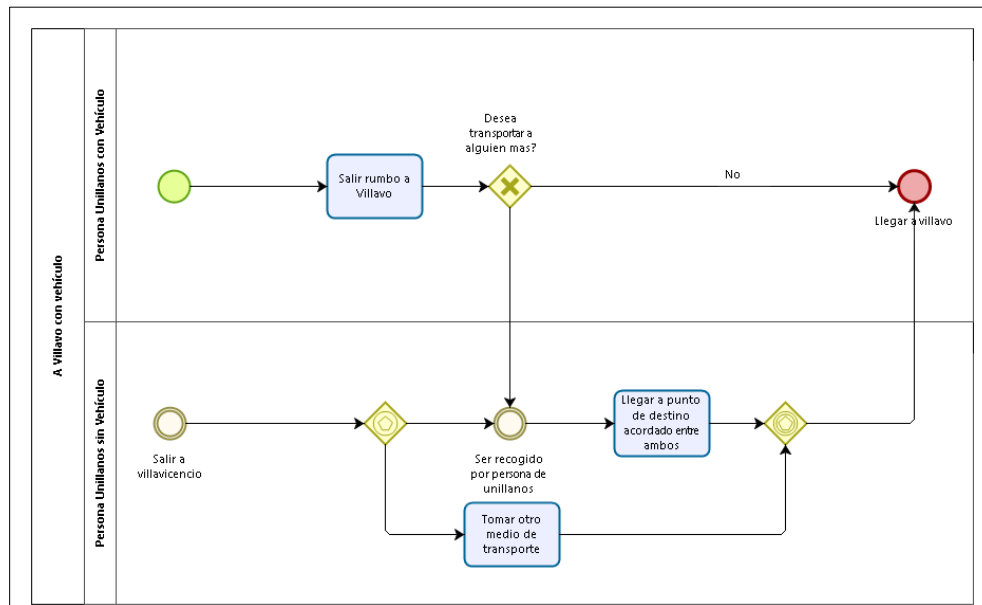


Figura 10. Modelo BPMN del proceso A Villavo con vehículo

- Usuarios involucrados en el proceso A Villavo sin vehículo:

En la Figura 11 se puede identificar los dos usuarios involucrados en el proceso, representados de manera horizontal:

- I. Persona con servicio particular
- II. Persona Unillanos sin Vehículo

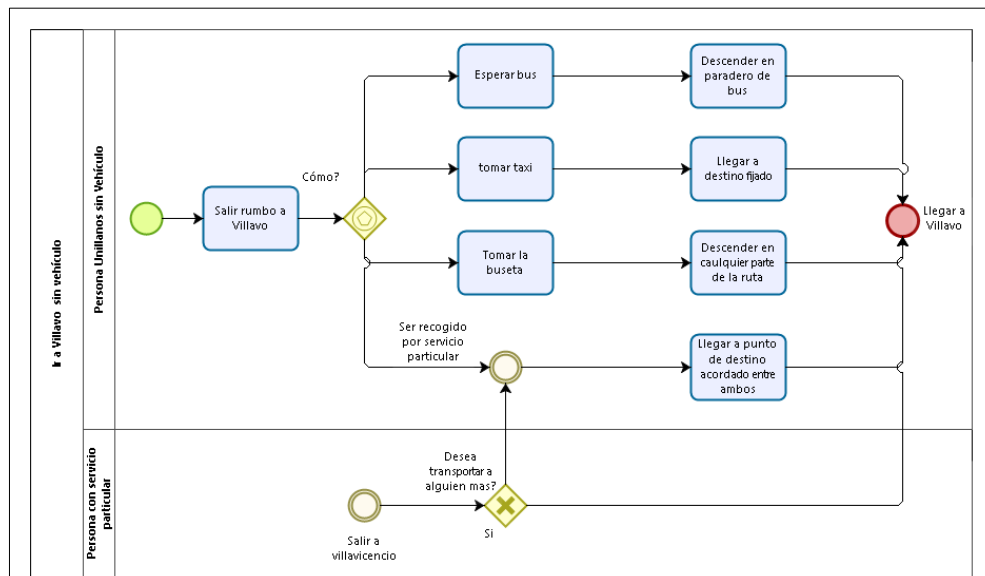


Figura 11. Modelo BPMN del proceso A Villavo sin vehículo

4.3. Diseño y desarrollo

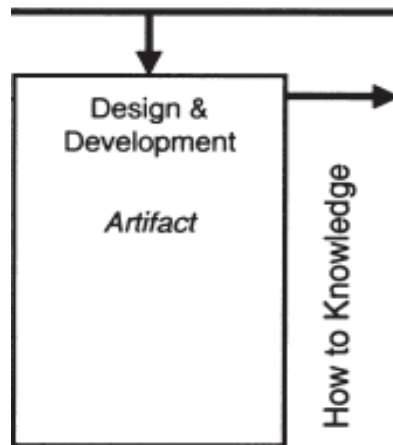


Figura 12. Fase de diseño y desarrollo. Adaptada de [31].

Para el diseño y desarrollo se aplicó la metodología PSP. Inicialmente y como línea base para futuras estimaciones y planes se utilizó el primer nivel PSP0, que tiene como objetivo inculcar practicas disciplinadas, siguiendo los diferentes scripts del proceso proporcionados por Watts S. Humphrey [34]. Este autor enseña la importancia del registro de datos, haciendo uso de los diferentes formularios y estándares, en este primer nivel nos encargamos básicamente de medir el tiempo invertido y los defectos inyectados y removidos en cada una de las fases.

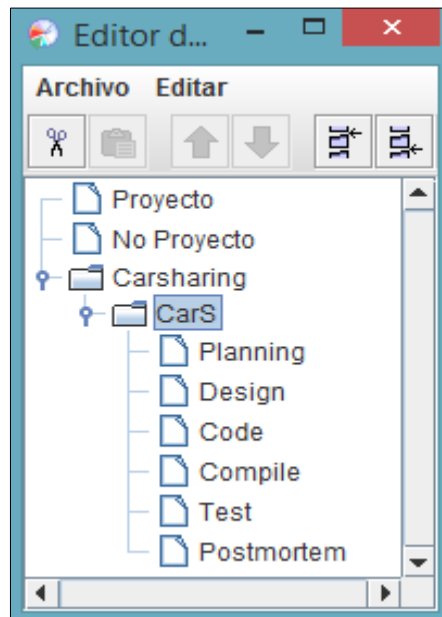


Figura 13. Creación del proyecto en Process DashBoard

Para desarrollar el proyecto y seguir el proceso planteado en PSP0 se hizo uso de la herramienta Process DashBoard [35] la cual brinda soporte para el proceso de PSP/TSP,

esta es una herramienta desarrollada en Java y con licencia Open Source, que integra todos los elementos fundamentales (Scripts, Formas, Métricas y Estándares) y de una forma muy fluida y ágil permite registrar los datos, analizar y plantear propuestas de mejora.

Para dar inicio se crea el proyecto en la herramienta como lo muestra la Figura 13 El nombre utilizado para el proyecto es “CarS” que representa las iniciales de Car Sharing, esto para no extender demasiado el campo nombre, cabe recordar que PSP tiene por definición tres fases (Planeación, Desarrollo y PostMorten), a su vez PSP0 expande la fase de desarrollo en Diseño, Codificación y Pruebas, es por esto que al momento de crear el proyecto la herramienta inmediatamente crea la estructura de carpetas con las diferentes fases.

Para aplicar PSP es necesario seguir las actividades propuestas en el Script, con Process DashBoard se puede visualizar los Scripts que se necesita a cualquier momento, para el caso de PSP 0 esta es la guía general de proceso:

PSP0 Process Script		
Purpose	To guide the development of module-level programs	
Entry Criteria	- Problem description - PSP0 Project Plan Summary form Time and Defect Recording logs Defect Type standard - Stopwatch (optional)	
Step	Activities	Description
1	Planning	- Produce or obtain a requirements statement. - Estimate the required development time. - Enter the plan data in the Project Plan Summary form. - Complete the Time Recording log.
2	Development	- Design the program. - Implement the design. - Compile the program, and fix and log all defects found. - Test the program, and fix and log all defects found. - Complete the Time Recording log.
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.
Exit Criteria		- A thoroughly tested program - Completed Project Plan Summary form with estimated and actual data - Completed Time and Defect Recording logs

Figura 14. PSP0 Process Script

Este Script es la guía que permite avanzar de manera ordenada y disciplinada en nuestro trabajo, las actividades establecidas son las siguientes:

4.3.1. Criterios de Entrada

La primera actividad es verificar que los criterios de entrada se cumplan, para este caso los criterios se cumplen así:

4.3.1.1. Descripción del problema

Se logró en la construcción del modelo del dominio de negocio que se detalló en el numeral 4.1, luego del cual se describieron las actividades susceptibles de mejora con software.

4.3.1.2. Formulario de registro de tiempo("PSP0 Time Recording Logs")

← → ↻ localhost:2468/Carsharing/CarS//reports/timelog.class ☆ ⚙ ⋮

Bitácora de TiempoPara /Carsharing/CarS

Proyecto/Tarea	Fase	Tiempo Inicio	Delta	Interrupción	Comentario
----------------	------	---------------	-------	--------------	------------

[Exportar a Excel](#)

Esta vista de la bitácora de tiempo es de solo-lectura. Para agregar registros a la bitácora, usar los botones play/pausa en el dashboard. Para editar ó borrar registros de la bitácora, usar el editor de la bitácora de tiempo (acceso en menú de Configuración -C- del dashboard).

Figura 15. PSP0 Time Recording Logs.Tomada del Dashboard

4.3.1.3. Formulario de registro de defectos("PSP0 Defect Recording Logs")

← → ↻ localhost:2468/Carsharing/CarS//reports/defectlog.class ☆ ⚙ ⋮

Bitácora de DefectosPara /Carsharing/CarS

Proyecto/Tarea	Fecha	ID	Tipo	Inyectado	Eliminado	Tiempo de Arreglo	Cuenta de correcciones	Defecto por Arreglo	Pendiente	Descripción
----------------	-------	----	------	-----------	-----------	-------------------	------------------------	---------------------	-----------	-------------

[Exportar a Excel](#)

Esta vista de la bitácora de defectos es de solo-lectura. Para agregar registros a la bitácora, usar el botón de defecto en el dashboard. Para editar ó borrar registros de la bitácora, usar el editor de la bitácora de defectos (acceso en menú de Configuración -C- del dashboard).

Figura 16. PSP0 Defect Recording Logs.Tomada del Dashboard

Efectivamente se cumplen dichos criterios puesto que la descripción del problema se define en el numeral 4.1 y los demás ítems DashBoard permite acceder directamente al Formulario de Resumen de plan de proyecto, Formulario de Registro de Tiempo y Defectos y finalmente al Estándar de Tipos de Defectos tal y como se ven los accesos directos en la Figura 14.

4.3.2. Actividades

A continuación se muestra cómo se aplicó la metodología en las actividades de planeación, desarrollo y postmorten del script PSP0.

4.3.2.1. Planeación

El objetivo de esta etapa es planificar el trabajo y estimar los recursos con el mayor detalle posible tomando como base la descripción del problema, a partir de la cual se establece la especificación de los requerimientos de la aplicación web. Para el desarrollo de esta fase se realizaron las actividades propuestas en el Script de PSP0 para la fase de planeación, el cual se puede ver en la Figura 17.

Purpose	To guide the PSP planning process	
Entry Criteria	• Problem description • Project Plan Summary form • Time Recording log	
Step	Activities	Description
1	Program Requirements	• Produce or obtain a requirements statement for the program. • Ensure that the requirements statement is clear and unambiguous. • Resolve any questions.
2	Resource Estimate	• Make your best estimate of the time required to develop this program. • Enter the plan time data in the Project Plan Summary form.
Exit Criteria	• Documented requirements statement • Completed Project Plan Summary form with estimated development time data • Completed Time Recording log	

Figura 17. PSP0 Planning Script

4.3.2.1.1. Criterios de entrada

Para el script de planeación los criterios de entrada son los mismos del script general de PSP0, los cuales, como se explicó en el numeral 4.3.1, se cumplen a cabalidad. Así pues, se puede dar inicio a las siguientes actividades propuestas:

4.3.2.1.2. Especificación de requerimientos (Program Requirements)

En este paso se obtuvo la definición y especificación de los requerimientos de una forma clara e inequívoca, agrupados en funcionales y no funcionales. En la Figura 18 puede apreciarse la descripción de esta actividad.

Step	Activities	Description
1	Program Requirements	• Produce or obtain a requirements statement for the program. • Ensure that the requirements statement is clear and unambiguous. • Resolve any questions.

Figura 18. Actividad de especificación de los requerimientos

El primer grupo se especificó aplicando las técnicas de modelado de software orientado a objetos [36], es decir, haciendo uso de los diagramas de caso de uso, descripción y paso a paso de los mismos. Los requerimientos no funcionales simplemente se listaron. A continuación se describen los resultados obtenidos en la actividad.

4.3.2.1.2.1. Requerimientos Funcionales:

Los casos de uso representan las interacciones entre los actores y la aplicación web, definiéndose así la funcionalidad del sistema. Estos requerimientos se establecen identificando los actores y las acciones, uso, los actores son las agrupaciones de los diferentes usuarios del sistema en roles.

4.3.2.1.2.1.1. Actores

Los actores del proceso fueron descubiertos en la fase de definición de objetivos, cuando se construyó el modelo de dominio.

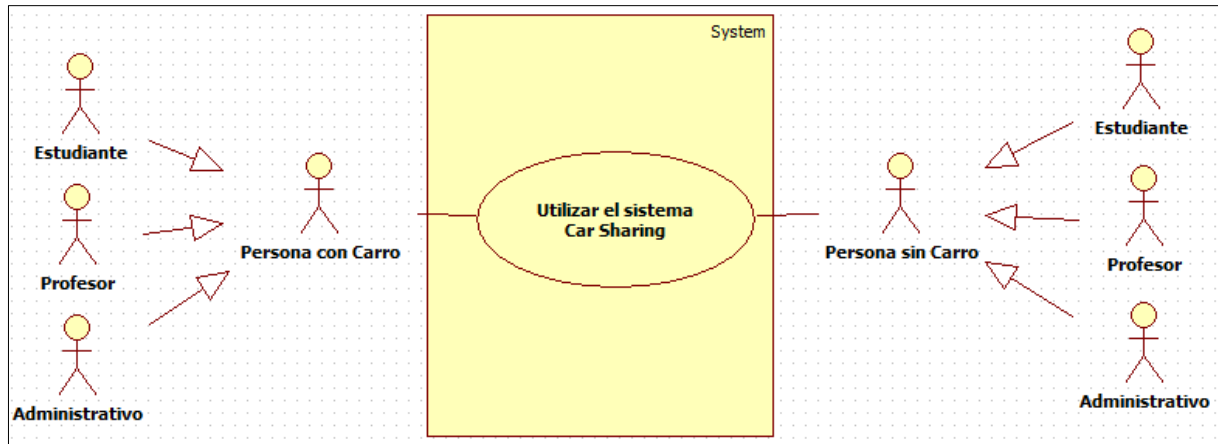


Figura 19. Actores del Sistema

A continuación se detallan estos roles,

- Persona con Carro

Es el actor que representa al personal de la Universidad de los Llanos (estudiantes, profesores y administrativos) y que además cuenta con un vehículo que eventualmente podrá compartirlo con el actor Persona sin Carro.

- Persona sin Carro

Es el actor que de igual manera representa al personal de la Universidad de los Llanos (estudiantes, profesores y administrativos) y que eventualmente podrá usar un asiento disponible del vehículo del actor Persona con Carro.

4.3.2.1.2.1.2. Casos de Uso

A continuación se presentan los casos de uso iniciando por uno general que busca describir el sistema en su plenitud, seguido de los casos de uso particulares, en forma más detallada.

4.3.2.1.2.1.2.1. Caso de uso general

El caso de uso general muestra la interacción de todos los actores con las funcionalidades establecidas para la aplicación. En este se establece que la Persona con Carro podrá publicar recorrido, cancelar recorrido y cancelar reserva de persona en su recorrido; mientras que la Persona sin Carro podrá buscar recorrido, reservar cupo y cancelar cupo reservado.

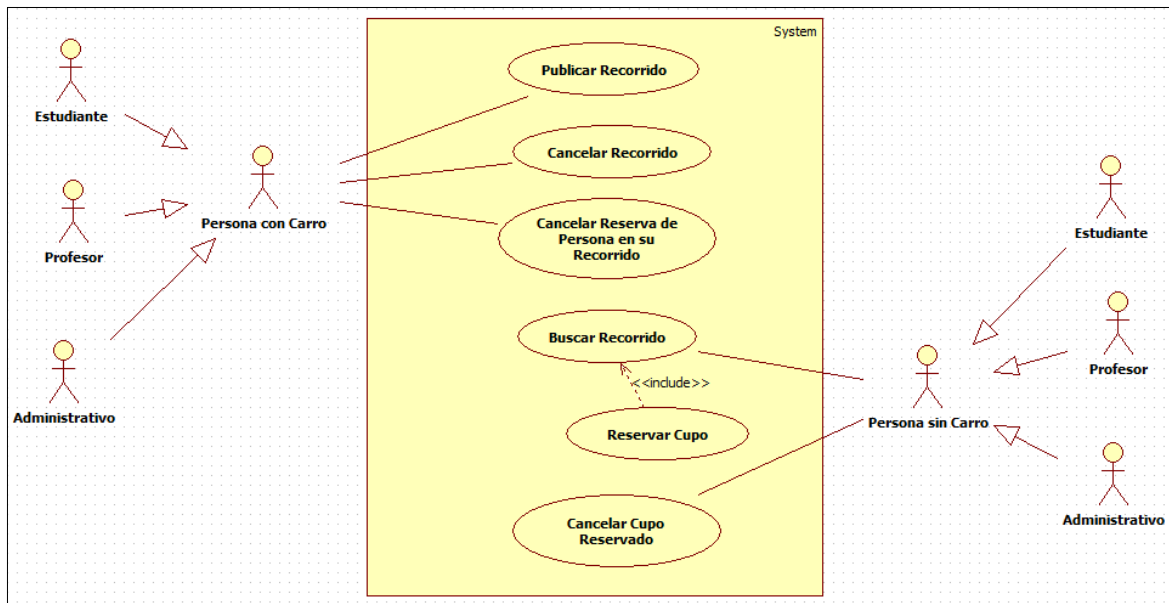


Figura 20. Caso de uso general

4.3.2.1.2.1.2.2. Caso de uso Publicar Recorrido

El caso de uso Publicar Recorrido permite a Persona con Carro publicar un recorrido en la aplicación, incluyendo el ingreso de fecha y hora de ida, ingresar lugar de salida y destino e ingresar cantidad de cupos disponibles. A su vez, podrá ingresar hora de vuelta, ingresar paraderos y seleccionar preferencias.

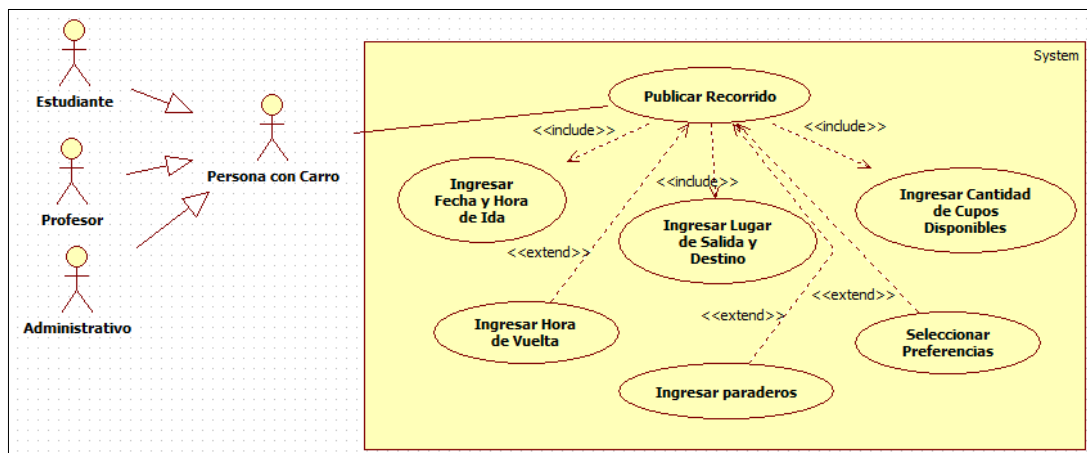


Figura 21. Caso de uso Publicar Recorrido

4.3.2.1.2.1.2.3. Caso de uso Cancelar Recorrido

El caso de uso Buscar Recorrido permite a la Persona con Carro cancelar un recorrido antes creado, para esto debe seleccionar el recorrido y posteriormente presionar la opción eliminar.

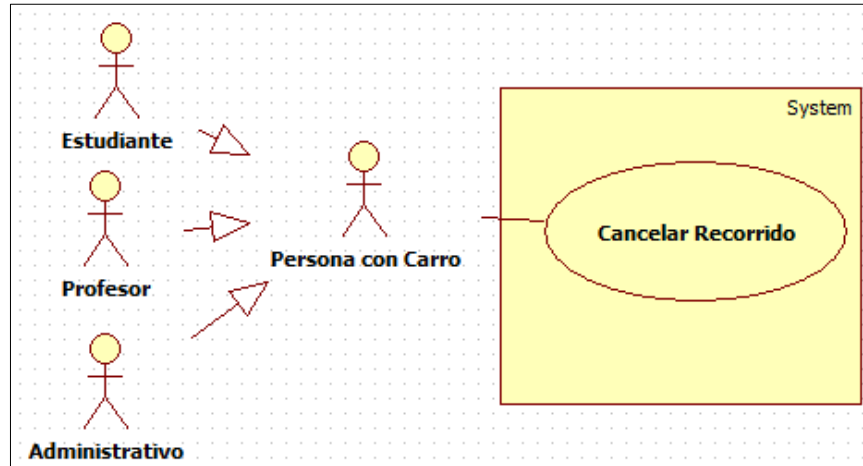


Figura 22. Caso de uso Cancelar Recorrido

4.3.2.1.2.1.2.4. Caso de uso Cancelar Reserva de Persona en su Recorrido

El caso de uso Buscar Recorrido permite a la Persona con Carro cancelar la reserva de una persona en su recorrido ya sea por preferencias u otros factores.

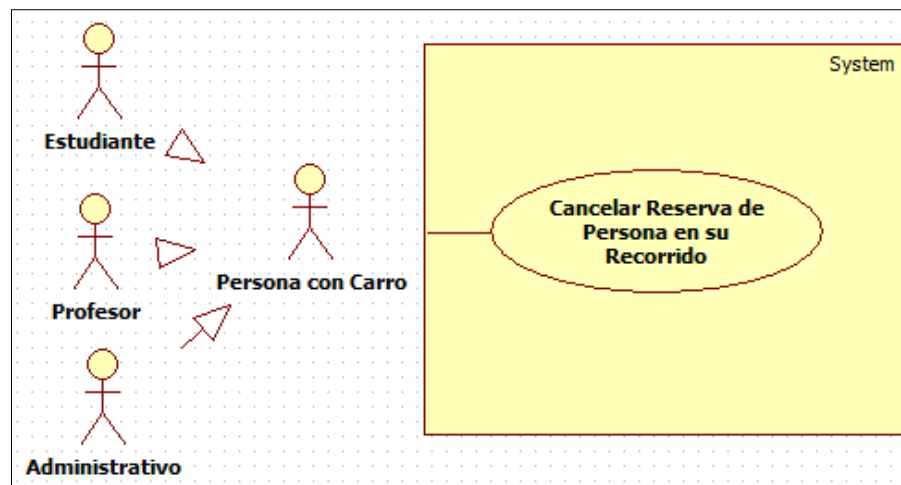


Figura 23. Caso de uso Cancelar Reserva de Persona en su Recorrido

4.3.2.1.2.1.2.5. Caso de uso Buscar Recorrido

El caso de uso Buscar Recorrido permite a la Persona sin Carro poder buscar recorridos con asientos disponibles (previamente publicados por la Persona con Carro) seleccionando si su trayecto será de Villavicencio a Unillanos o de Unillanos a Villavicencio y presionando la opción buscar.

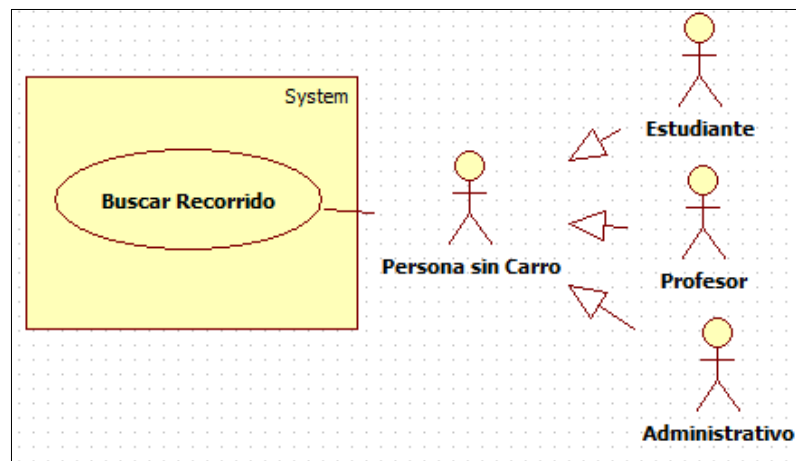


Figura 24. Caso de uso Buscar Recorrido

4.3.2.1.2.1.2.6. Caso de uso Cancelar Cupo Reservado

El caso de uso Cancelar Cupo Reservado permite a la persona sin Carro que haya reservado un asiento en un recorrido, cancelarlo seleccionando la reserva y presionando la opción cancelar.

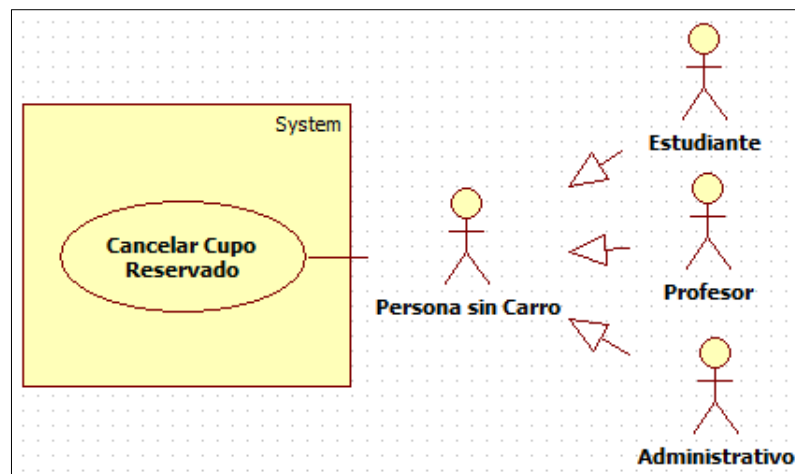


Figura 25. Caso de uso Cancelar Cupo Reservado

4.3.2.1.2.1.3. Requerimientos No Funcionales

Los requerimientos no funcionales de la aplicación describen los aspectos generales, que no incluyen una relación directa con el comportamiento funcional del sistema, estos son:

- El lenguaje de programación a usar para este proyecto de software es Java EE y XHTML

- El sistema debe permitir un nivel de concurrencia alto, puesto que en las horas pico el aplicativo de software será utilizado por muchas personas de la universidad al mismo tiempo.
- El sistema debe contar con buena seguridad para la autenticación de los usuarios, puesto que es un factor determinante para el bienestar tanto de la persona que usa el servicio como de quien lo presta.

Una vez establecidos los requerimientos, se pasa a la siguiente actividad propuesta en el script de Planeación, que es estimar los recursos, que para el caso de PSP0 solamente es el tiempo estimado para la realización de todo el proyecto.

4.3.2.1.3. Estimación de Recursos

La estimación de recursos realizada en este proyecto corresponde al valor del tiempo planeado total para el desarrollo del proyecto, para calcular este valor se debe usar un juicio personal, se apoyará comparando lo planificado respecto al tiempo utilizado en proyectos anteriores (trabajos previos, con tamaño y funcionalidades parecidas).

← → ↻ ⓘ localhost:2468/Carsharing/CarS//psp0/planning.htm 📄 ☆ 🚗

PSP0 Planning Script 📄

Purpose	To guide the PSP planning process	
Entry Criteria	• Problem description • Project Plan Summary form • Time Recording log	
Step	Activities	Description
1	Program Requirements	• Produce or obtain a requirements statement for the program. • Ensure that the requirements statement is clear and unambiguous. • Resolve any questions.
2	Resource Estimate	• Make your best estimate of the time required to develop this program. • Enter the plan time data in the Project Plan Summary form.
Exit Criteria		• Documented requirements statement • Completed Project Plan Summary form with estimated development time data • Completed Time Recording log

Figura 26. PSP0 Planning Script

Se toma el tiempo basado en el cronograma de actividades. (Tabla 1)

Tabla 1. Cronograma de actividades

	SEMANAS																	
ACTIVIDADES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Planificación																		
Diseño																		
Codificación																		
Pruebas																		
Evaluación																		

Se propuso realizar el proyecto en cuatro meses y dos semanas, incluyendo todas las actividades involucradas, estos cuatro meses y dos semanas corresponden a 18 semanas, 5 días por semana, trabajando 4 horas por día lo cual da un total de 21,600 minutos.

4.3.2.1.4. Criterios de Salida

Con la estimación de recursos se completa la fase de planeación, ahora corresponde revisar que se cumplan los criterios de salida planteados en el script de la metodología (ver Figura 27):

Exit Criteria	• Documented requirements statement • Completed Project Plan Summary form with estimated development time data • Completed Time Recording log
----------------------	---

Figura 27. Criterios de salida PSP0 Planning Script

Una vez terminada la fase de Planeación continuamos con la segunda actividad y fase de PSP0 que es el Desarrollo.

4.3.2.2. Desarrollo

Una vez realizada la planificación la siguiente fase del proceso es el desarrollo o construcción del proyecto, que contiene por si misma 4 pasos como lo muestra la Figura 28.

Tal como se aprecia, el script de Desarrollo incluye las actividades tradicionales de la ingeniería de software, sin embargo es importante hacer claridad en algunos aspectos, a la luz de la mejora en las prácticas de desarrollo de software:

PSP0 Development Script		
Purpose	To guide the development of small programs	
Entry Criteria	• Requirements statement • Project Plan Summary form with estimated program development time • Time and Defect Recording logs • Defect Type standard	
Step	Activities	Description
1	Design	• Review the requirements and produce a design to meet them. • Record in the Defect Recording log any requirements defects found. • Record time in the Time Recording log.
2	Code	• Implement the design. • Record in the Defect Recording log any requirements or design defects found. • Record time in the Time Recording log.
3	Compile	• Compile the program until there are no compile errors. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
4	Test	• Test until all tests run without error. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
Exit Criteria	• A thoroughly tested program • Completed Time and Defect Recording logs	

Figura 28. PSP0 Development Script

- El diseño debe representar la totalidad de los requerimientos.
- El código fuente se debe realizar completamente antes de probar. Es una de las recomendaciones más difíciles de aceptar para los desarrolladores, sin embargo es fundamental para evitar el re-trabajo. Si se escribe un bloque de líneas, se ejecutan, se prueban, se corrigen los errores encontrados y se repite esto varias veces.
- Si el lenguaje o entorno de desarrollo usado hace innecesario o no usa compilación esta fase puede omitirse y “pasar” a la fase de pruebas.
- En pruebas el objetivo es verificar el funcionamiento del programa, corregir y registrar los defectos encontrados.
- Cuando se pase de una fase a otra, ej. Diseño a Codificación, si hay que hacer correcciones, cambios o mejoras, esto es registrado como un defecto.

4.3.2.2.1. Criterios de entrada

Los criterios de entrada del Script de desarrollo agregan un criterio a los que se especificaron como criterios de entrada para el script general de PSP0: la declaración de los requerimientos, que ya se encuentra realizada y expuesta en el numeral 4.3.2.1.2.

Con el cumplimiento de los criterios de entrada de este script, podemos dar inicio a las siguientes actividades propuestas:

4.3.2.2.2. Diseño

El diseño es el primer paso en la etapa de desarrollo y tiene como objetivo revisar los requerimientos y producir un diseño que represente en forma abstracta, la solución al problema propuesto. La Figura 29 describe los pasos a llevar a cabo en esta actividad.

Step	Activities	Description
1	Design 	• Review the requirements and produce a design to meet them. • Record in the Defect Recording log any requirements defects found. • Record time in the Time Recording log.

Figura 29. Actividades del diseño

Para la primera actividad “Revisar los requerimientos y producir un diseño que permita satisfacerlos”, la metodología PSP no prescribe una técnica específica de diseño, a cambio, propone que éste contemple por lo menos dos categorías compuestas así:



Figura 30. Estructura especificación de diseño PSP Tomada de [37]

En este trabajo por las características particulares de la tecnología JavaEE y los requerimientos planteados (funcionales y no funcionales), se utilizan las siguientes plantillas para el diseño:

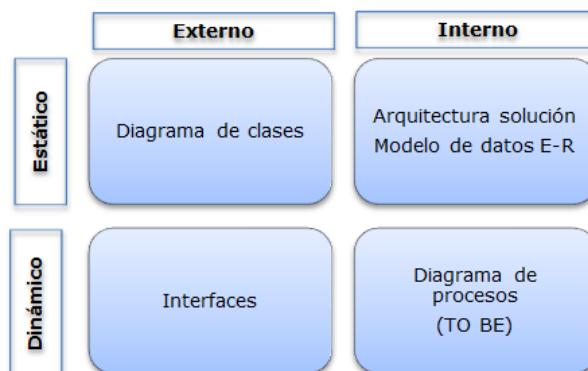


Figura 31. Estructura diseño de la aplicación

A continuación se describen los resultados de cada uno de estos componentes del diseño.

4.3.2.2.2.1. Arquitectura de la solución

La aplicación web para poder compartir el carro entre miembros de la comunidad Unillanos estuvo basada en la arquitectura cliente – servidor de tres capas. La idea de desarrollo en tres capas consiste en distribuir los diferentes componentes del sistema de acuerdo a su funcionalidad, en por lo menos tres grupos: la interfaz de usuario, las reglas del negocio y los modelos de datos. Esta separación es la razón fundamental para la utilización de la tecnología de JavaEE.

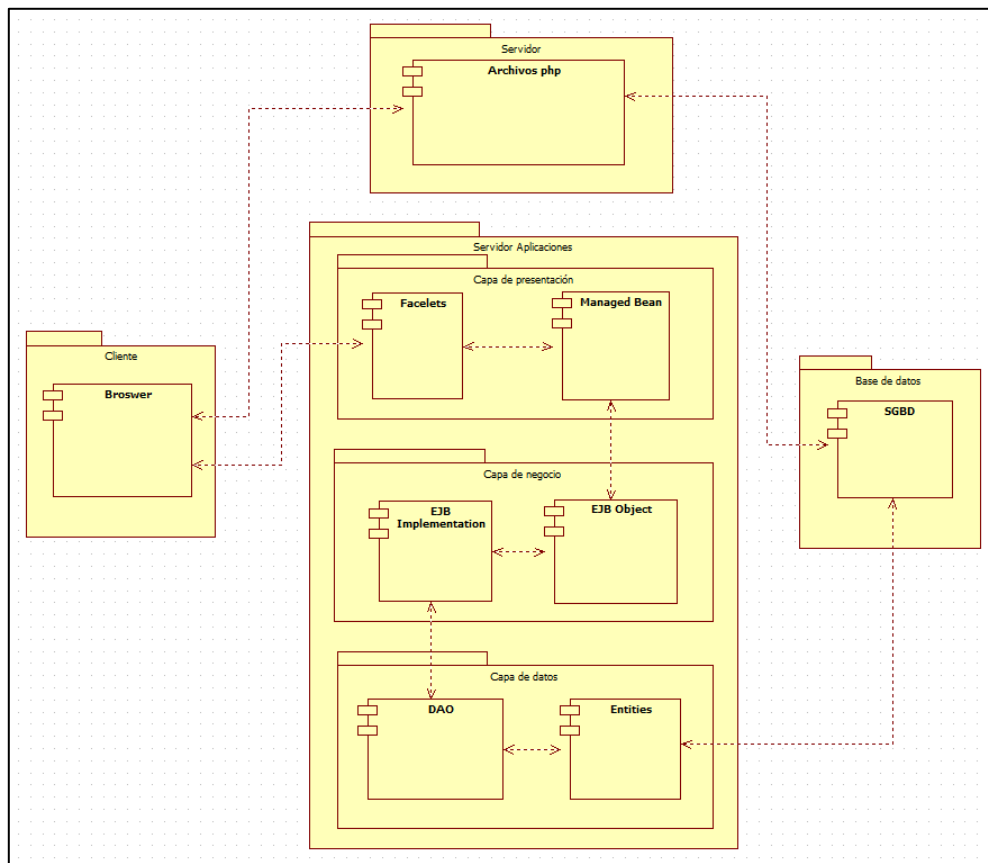


Figura 32. Arquitectura de la aplicación

La Figura 32 representa la arquitectura propuesta en la cual se pueden observar los siguientes componentes:

- **FACELETS:** son paginas XHTML usadas por el framework JSF para representar sus vistas, éstas utilizan bloques HTML como componentes del marco, apoyados por clases java.
- **MANAGED BEAN:** son las clases java usadas como modelos o controladores para los componentes de la interfaz de usuario.
- **EJB OBJECT:** son clases tipo interfaz las cuales declaran los métodos de instancia públicos, encargados de realizar las operaciones, sin su implementación.
- **EJB IMPLEMENTATION:** Son las clases donde se encuentra la lógica de negocio, implementan todos los métodos especificados por la interfaz
- **DAO:** Son las clases que abstraen los métodos de manipulación de la persistencia como todas las actividades CRUD (créate, read, update y delete).
- **ENTITIES:** Son las clases que representan una tabla del modelo de datos relacional y cada instancia u objeto corresponde a un registro en esta tabla.

4.3.2.2.2. Diseño de la interfaz

Con las funcionalidades establecidas para el sistema en los casos de uso se realizaron los siguientes prototipos de interfaz.

4.3.2.2.2.1. Interfaz Inicio de Sesión

La Figura 33 representa la interfaz para el inicio de sesión de un miembro de la comunidad Unillanos.

The mockup shows a login form titled "Unillanos CarShare". At the top is the Unillanos logo, a red circle with a stylized bird. Below the logo is the text "Por favor, ingresa tu usuario y contraseña". The form contains three input fields: "Aquí tu email", "Aquí tu contraseña", and a dropdown menu with "Conductor" and "Estudiante" options. Below these fields is a "Login" button.

Figura 33. Mockup Inicio de Sesión

4.3.2.2.2.2. Interfaz formulario Cambiar Contraseña

La Figura 34 representa el formulario de cambio de contraseña de un miembro de la comunidad Unillanos en la aplicación web.

The mockup shows a password change form titled "Unillanos CarShare". At the top is the Unillanos logo, a red circle with a stylized bird. Below the logo is the text "Cambiar contraseña". The form contains three input fields: "Contraseña actual", "Contraseña nueva", and "Confirmar contraseña". Below these fields is a "Confirmar" button. The form is part of a larger interface with a navigation bar at the top containing links: "Cambiar Contraseña", "Buscar", "Publicar", "Reservar", and a "Login" button.

Figura 34. Mockup formulario Cambiar Contraseña

4.3.2.2.2.3. Interfaz para Publicar Recorrido

La Figura 35 representa la interfaz que le permitirá a la Persona con Carro publicar un recorrido para que una persona sin carro pueda reservar un asiento disponible en su recorrido.

Para esto debe ingresar fecha y hora de ida, lugar de salida y destino, lugares de paradas y cantidad de asientos disponibles.

Adicional a esto puede: ingresar hora de vuelta y seleccionar preferencias

Unillanos CarShare Buscar Publicar Reservar Mis Recorridos Login

Publicar Recorrido:

Puntos de Encuentro

Salida: ☐ Unillanos ☒ Unicentro

Destino: ☒ Universidad de Los Llanos

Fecha y Hora ☐ Ida y Vuelta

Ida: DD/MM/AAAA HH:MM

Cupos Disponibles: 1

Paraderos

Ingrese las paradas en el recorrido:

☒ Alkosto

☒ U. Cooperativa

Preferencias

Si lo desea seleccione sus preferencias:

☐ Solo Mujeres ☐ Solo Hombres

☐ Administrativo ☐ Profesor

☐ Estudiante

Recorrido

Mapa de ruta: Universidad de Los Llanos → Unicentro

Figura 35. Mockup Publicar Recorrido

4.3.2.2.2.4. Interfaz Buscar Recorrido

La Figura 36 representa la interfaz para que la Persona sin carro puede buscar recorridos con asientos seleccionado si es de Villavicencio a Unillanos o de Unillanos a Villavicencio y presionando la opción buscar.

Unillanos CarShare Buscar Publicar Reservar Mis Recorridos Login

Buscar Recorrido:

☐ Villavicencio ☒ Unillanos

Juan Perez Estudiante	Martes, 13 de Septiembre de 2015 Ida: 08:15 AM Vuelta: 05:20 PM Paradas: ☐ Unicentro, Villavicencio ☐ Alkosto ☐ U. Cooperativa ☒ Unillanos	Cupos Disponibles: 2 Reservar Cupo
Pedro Profesor	Viernes, 16 de Septiembre de 2015 Ida: 07:30 AM Paradas: ☐ Porfia, Villavicencio ☐ Guatiquia ☒ Unillanos	Cupos Disponibles: 1 Reservar Cupo

Figura 36. Mockup Buscar Recorrido

4.3.2.2.2.5. Interfaz Reservar Cupo

La Figura 37 representa la interfaz para que la persona sin carro pueda reservar asiento de un recorrido, para esto debe haber buscado antes el recorrido en el cual reservar.

CodeName: Proyecto A

Reservar Cupo [X]

Persona: **Juan Perez**
 Fecha de Salida: **13/09/15** Salida: **Unicentro**
 Hora de Salida: **08:15 AM** Destino: **Unillanos**
 Hora de Vuelta: **05:20 PM**

Seleccione Paradero:
☐ Unicentro ☐ Alkosto ☐ U. Cooperativa

☐ Ida ☐ Vuelta ☐ Ida y Vuelta

Paradas: ☐ Guatiquia
☒ Unillanos

Figura 37. Mockup configurara la convocatoria

4.3.2.2.2.6. Interfaz Cancelar Cupo Reservado

La Figura 38 representa la interfaz en la cual la persona sin carro que haya reservado un asiento en un recorrido puede cancelarlo seleccionando la reserva y presionando la opción cancelar.

 Unillanos CarShare Buscar Publicar Reservar Login				
Gestionar Reservas:				
Persona	Fecha	Hora	Paradero	Acción
Juan Lopez	25/09/15	07:30 AM	Unicentro	Cancelar Reserva
Juan Lopez	25/09/15	06:00 PM	U. Cooperativa	Cancelar Reserva
Carolina Perez	28/09/15	07:50 AM	Unicentro	Cancelar Reserva

Figura 38. Mockup Cancelar Cupo Reservado

4.3.2.2.2.7. Interfaz Cancelar Recorrido

La Figura 39 representa la interfaz para que una persona con carro pueda cancelar un recorrido antes creado, para esto debe seleccionar el recorrido y posteriormente presionar la opción eliminar.

 Unillanos CarShare Buscar Publicar Reservar Login				
Gestionar Recorridos:				
Salida	Destino	Fecha	Hora	Acción
Unicentro	Unillanos	25/09/15	07:30 AM	Cancelar Recorrido Gestionar Reservas
Unillanos	Unicentro	25/09/15	06:00 PM	Cancelar Recorrido Gestionar Reservas
Galán	Unillanos	28/09/15	07:50 AM	Cancelar Recorrido Gestionar Reservas

Figura 39. Mockup Cancelar Recorrido

4.3.2.2.2.8. Interfaz Cancelar Reserva de Persona en su Recorrido

La Figura 40 representa la interfaz para que una persona con carro pueda cancelar la reserva de una persona en su recorrido ya sea por preferencias u otros factores.


Unillanos CarShare
Buscar Publicar Reservar Login

Gestionar Reservas en Recorrido:
 Unicentro - Unillanos: 25/09/15 (07:30 AM)

Nombre	Cargo	Género	Paradero	Acción
Edison Arango	Estudiante	Masculino	Alkosto	Cancelar Reserva
Maria Lopez	Profesora	Femenino	Unicentro	Cancelar Reserva

Figura 40. Mockup Cancelar Reserva de Persona en su Recorrido

4.3.2.2.3. Modelo Estático

En la Figura 41 se muestra el diagrama de clases que se utilizó para la implementación de la aplicación

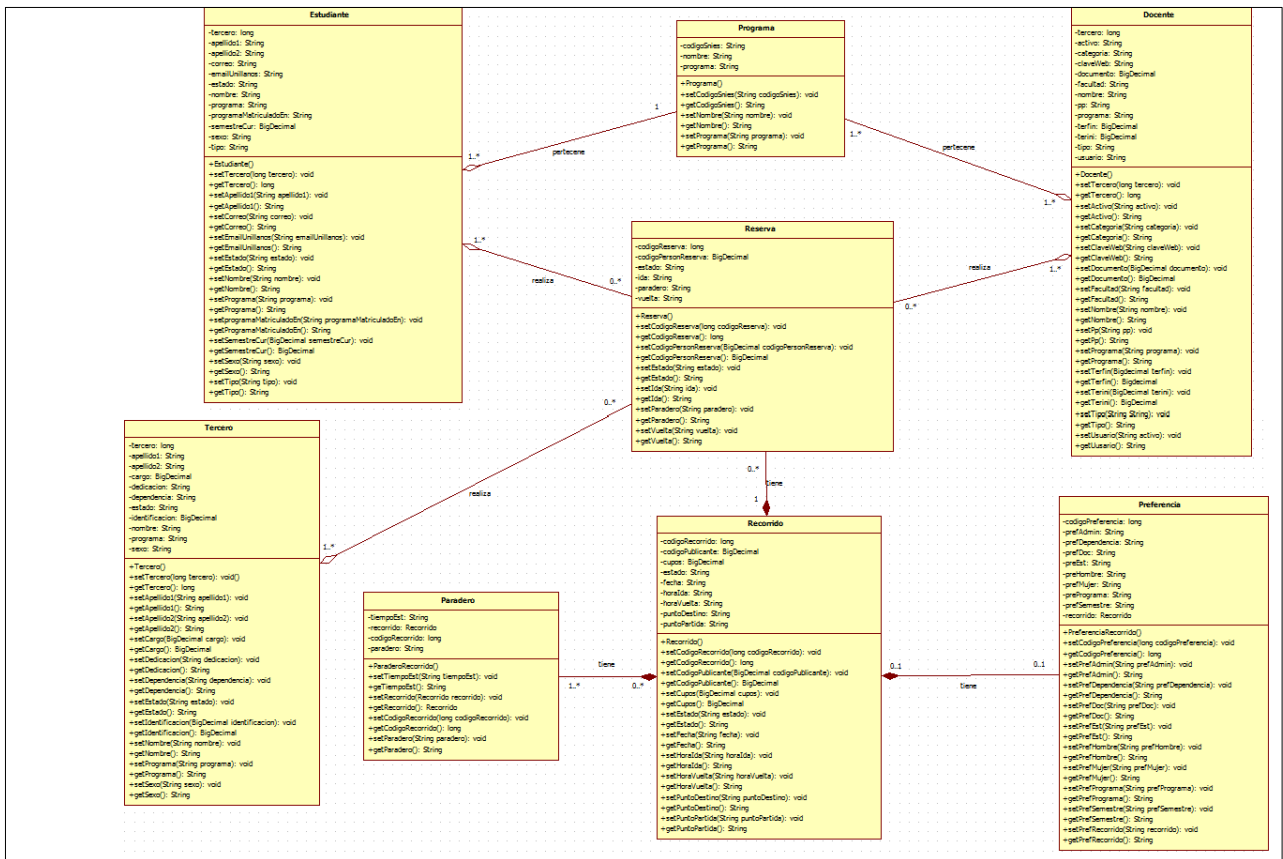


Figura 41. Diagrama de clases.

4.3.2.2.2.4. Modelo de datos (Modelo Entidad Relación)

Una vez definida la arquitectura de la aplicación se procedió con el diseño del modelo de datos el cual consistió en buscar las entidades que describen los objetos que intervienen en el problema y las relaciones entre estas entidades. Las tablas o entidades identificadas para el funcionamiento del sistema y sus relaciones se pueden observar en el modelo Entidad – Relación representado en la Figura 42.

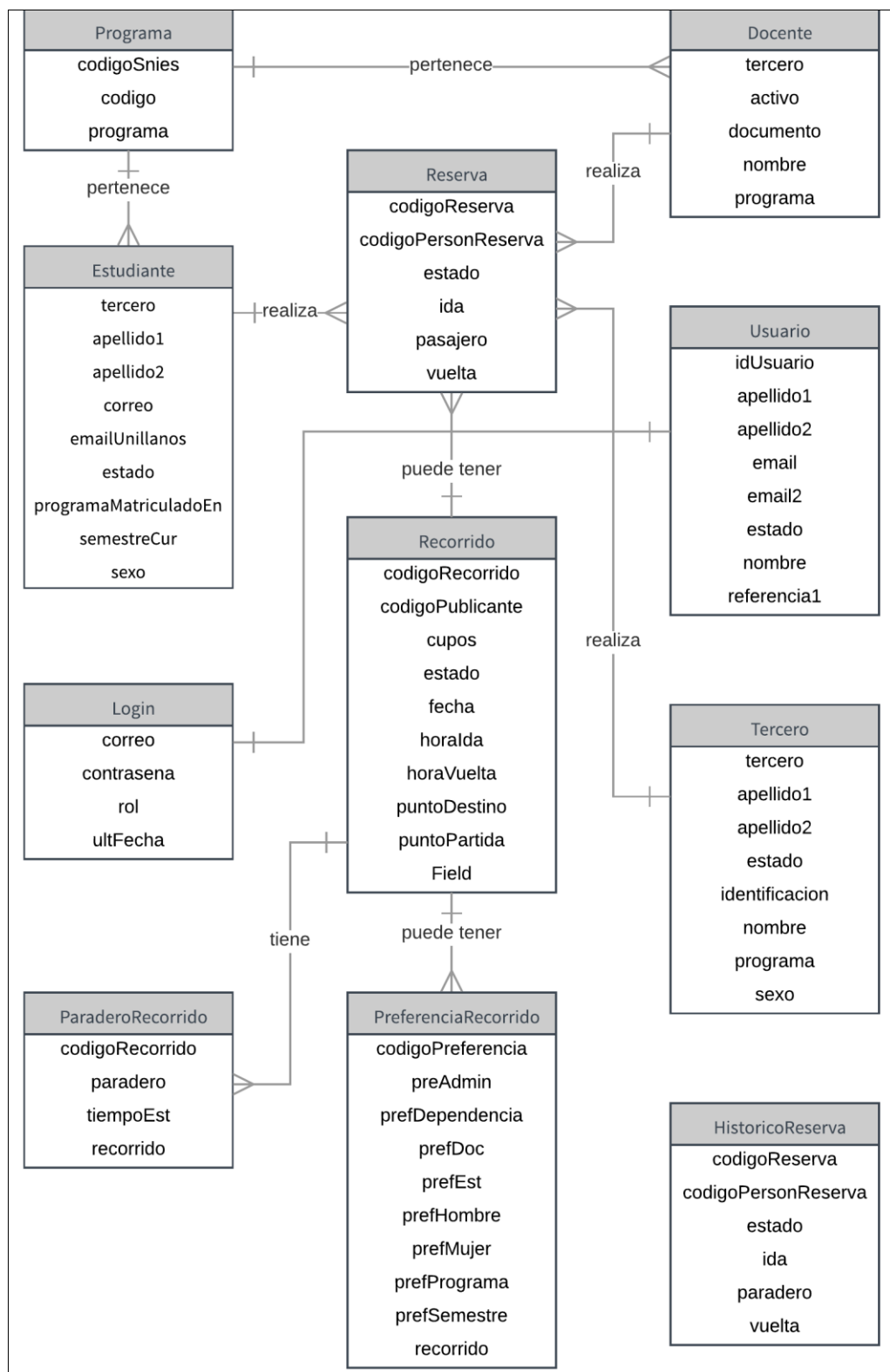


Figura 42. Modelo Entidad Relación

4.3.2.2.5. Modelo de Procesos

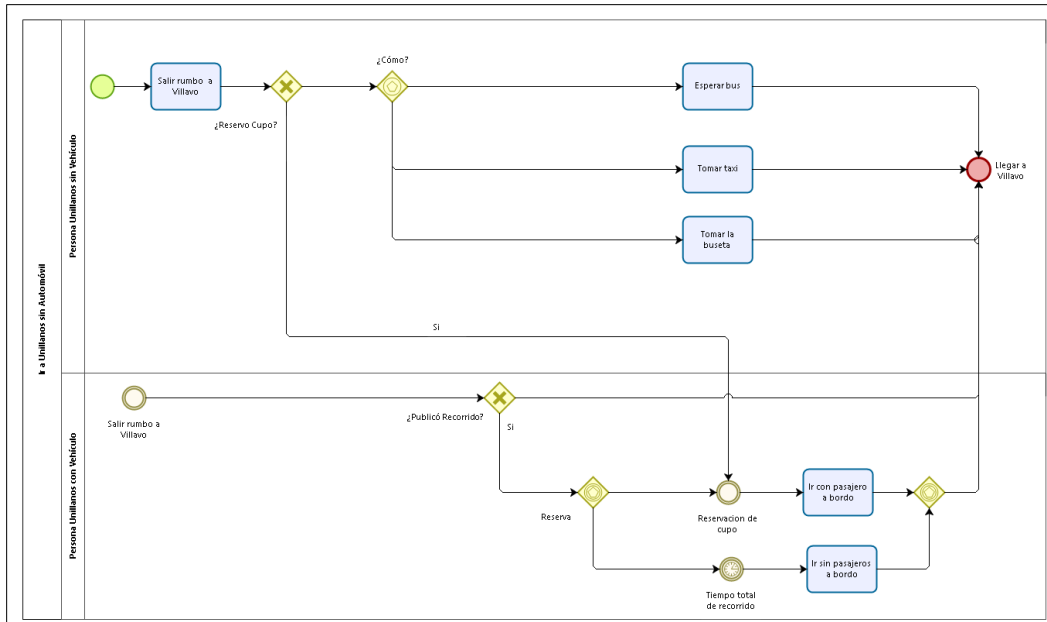


Figura 43. Modelo BPMN del proceso Ir a Villavieja con Automóvil TOBE

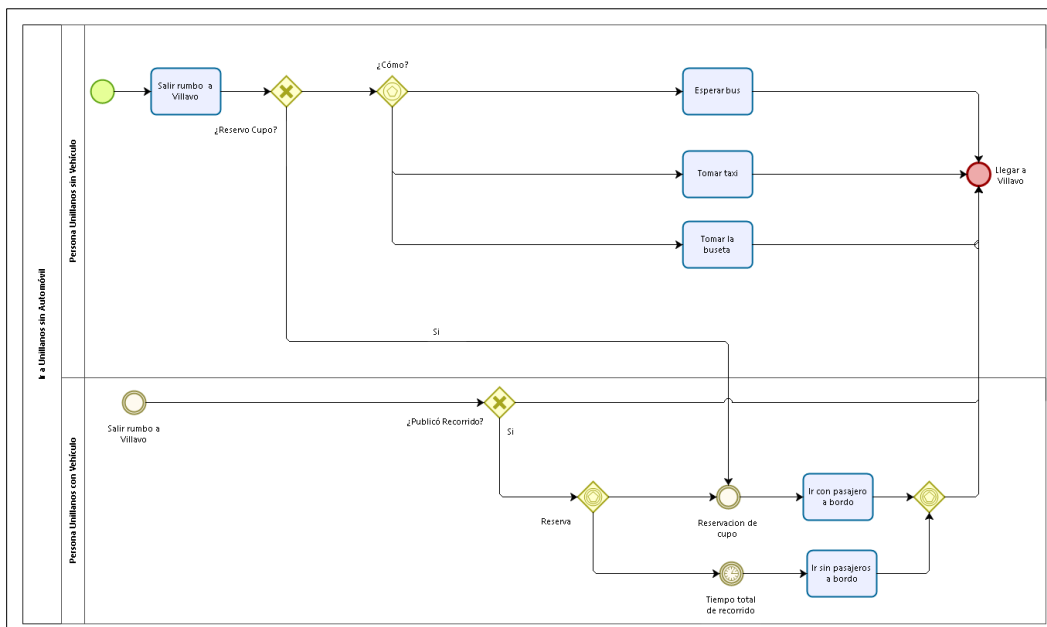


Figura 44. Modelo BPMN del proceso Ir a Villavieja con Automóvil TOBE

4.3.2.2.3. Codificación

Una vez finalizados los componentes del diseño, con sus registros de tiempo de las diferentes actividades de la fase en el Time Recording Log, se pasa la siguiente actividad del script que es la codificación. Ésta es la segunda parte del desarrollo y tiene como objetivo convertir el diseño presentado, en código fuente, por medio de

un lenguaje de programación; para este caso, usando la tecnología Java EE. La Figura 29 describe los pasos a llevar a cabo en esta actividad.



PSP0 Development Script		
Purpose		To guide the development of small programs
Entry Criteria		• Requirements statement • Project Plan Summary form with estimated program development time • Time and Defect Recording logs • Defect Type standard
Step	Activities	Description
1	Design	• Review the requirements and produce a design to meet them. • Record in the Defect Recording log any requirements defects found. • Record time in the Time Recording log.
2	Code	• Implement the design. • Record in the Defect Recording log any requirements or design defects found. • Record time in the Time Recording log.
3	Compile	• Compile the program until there are no compile errors. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
4	Test	• Test until all tests run without error. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
Exit Criteria		• A thoroughly tested program • Completed Time and Defect Recording logs

Figura 45. Actividades de la codificación

Se determinó realizar el desarrollo de la aplicación con la tecnología Java EE con los framework JSF para la capa de presentación, EJB para la lógica del negocio y JPA para el modelo de datos, debido a todos los beneficios descritos anteriormente en el numeral 3.2.4.1 respectivamente.

La implementación del diseño propuesto, utilizando la tecnología seleccionada se realizó con los siguientes entornos de desarrollo (codificación):

- Eclipse Java EE: Herramienta para crear aplicaciones JavaEE
- Maven: Sistema para manejo de proyectos de software.

La Figura 46 muestra la implementación del proyecto con toda la estructura de carpetas.

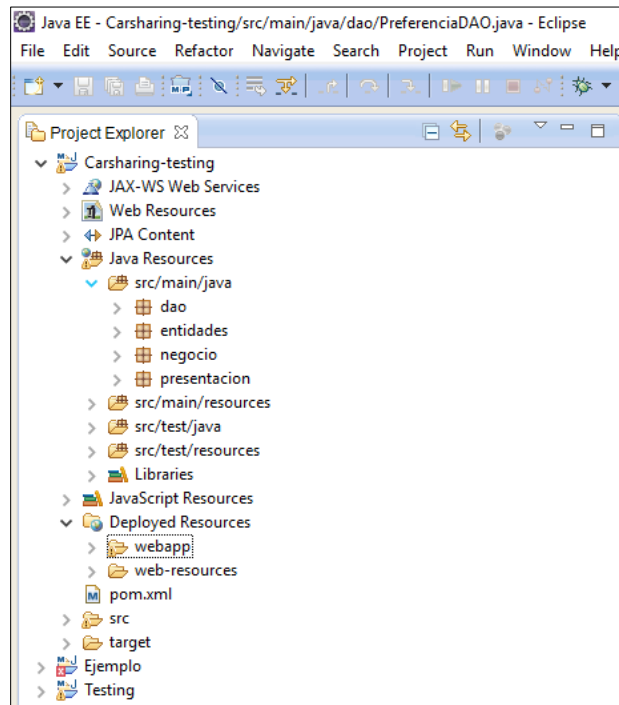


Figura 46. Estructura de carpetas del proyecto

Se puede observar los componentes del proyecto agrupados en paquetes según su funcionalidad tal como se representó en el diseño de la arquitectura de la aplicación en el numeral 4.3.2.2.1. A continuación se detallan cada uno de ellos.

4.3.2.2.3.1. Paquete Entidades

En este paquete se encuentra todas las clases entidad que representan las tablas del modelo de datos diseñado en el numeral 4.3.2.2.4.

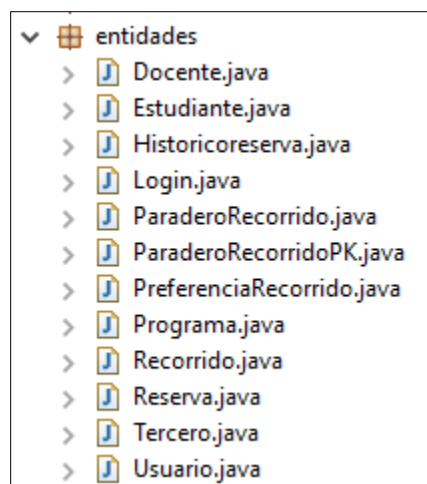


Figura 47. Clases Entidad

Para dar un ejemplo de cómo es el código fuente de una clase entidad usando la especificación JPA, a continuación se muestra la clase Recorrido.java

```

import java.io.Serializable;

/**
 * The persistent class for the RECORRIDO database table.
 *
 */
@Entity
@NamedQuery(name="Recorrido.findAll", query="SELECT r FROM Recorrido r")
public class Recorrido implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    private long codigorecorrido;

    private BigDecimal codigopublicante;

    private BigDecimal cupos;

    private String estado;

    private String fecha;

    private String horaaida;

    private String horavuelta;

    private String puntodestino;

    private String puntopartida;

    //bi-directional many-to-one association to ParaderoRecorrido
    @OneToMany(mappedBy="recorrido")
    private List<ParaderoRecorrido> paraderoRecorridos;

    //bi-directional many-to-one association to PreferenciaRecorrido

```

Figura 48. Código fuente clase Recorrido.java

4.3.2.2.3.2. Paquete DAO

En este paquete se encuentran todas las clases DAO creadas para la manipulación de las entidades del modelo datos como se observa en la Figura 49. Clases DAO.

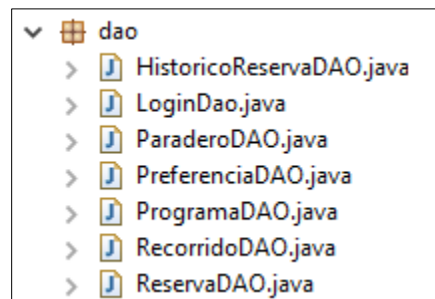


Figura 49. Clases DAO

Para dar un ejemplo de cómo es el código fuente de una clase DAO usando la especificación JPA, a continuación se muestra la clase RecorridoDAO.java

```

package dao;

import java.math.BigDecimal;

public class RecorridoDAO {

    @PersistenceContext
    EntityManager manager;

    @Transactional(TransactionalAttributeType.REQUIRED)
    public void crearRecorrido(Recorrido recorrido) {
        manager.persist(recorrido);
    }

    @Transactional(TransactionalAttributeType.REQUIRED)
    public void eliminarRecorrido(Recorrido recorrido) {
        manager.remove(recorrido);
    }

    public Recorrido consultarRecorrido(long idRecorrido) {
        return manager.find(Recorrido.class, idRecorrido);
    }

    public void actualizarRecorrido(Recorrido recorrido) {
        Recorrido recorridoenbd=manager.find(Recorrido.class, recorrido.getCodigorecorrido());
        recorridoenbd.setCupos(recorrido.getCupos());
        recorridoenbd.setFecha(recorrido.getFecha());
        recorridoenbd.setEstado(recorrido.getEstado());
        recorridoenbd.setHoraida(recorrido.getHoraida());
        recorridoenbd.setHoravuelta(recorrido.getHoravuelta());
        recorridoenbd.setPuntodestino(recorrido.getPuntodestino());
        recorridoenbd.setPuntopartida(recorrido.getPuntopartida());
    }

    public void agregarParaderoaRecorrido(Recorrido recorrido,ParaderoRecorrido paradero) {
        Recorrido recorridoenbd=manager.find(Recorrido.class,recorrido.getCodigorecorrido());
        recorridoenbd.addParaderoRecorrido(paradero);
    }
}

```

Figura 50. Código fuente clase RecorridoDAO.java

4.3.2.2.3.3. Paquete Negocio

En este paquete se encuentran las clases que representan la lógica del negocio, exactamente para este proyecto dos clases, una que representa la interfaz donde se declaran los métodos de instancia públicos y otra donde se implementan todos los métodos especificados por la interfaz. Tal y como se observa en la Figura 51. Clases de negocioFigura 51.

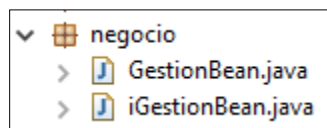


Figura 51. Clases de negocio

Para dar un ejemplo de cómo es el código fuente de las clases de negocio usando el framework EJB, a continuación se muestra las clases de interfaz e implementación.

```

package negocio;

import java.math.BigDecimal;

@Remote
public interface iGestionBean {
    public List<Programa> consultarProgramaPorColumna();
    public Programa consultarPrograma(String idPrograma);
    public void crearRecorrido(long codigopubli, String puntoPar, String puntoDes, String horaIda, String horaVuelta, String estado, String fecha, ArrayList<ParaderoRecorrido> paraderos, PreferenciaRecorrido preferencia, String fecha);
    public List<Recorrido> consultarRecorridoPorColumna();
    public void eliminarRecorrido(long recorrido);
    public Recorrido consultarRecorrido(long idRecorrido);
    public List<ParaderoRecorrido> consultarParadero(long valor);
    public boolean eliminarParadero(long recorrido, String paradero);
    public void actualizarPreferencia(PreferenciaRecorrido preferencia);
    public PreferenciaRecorrido consultarPreferencia(long idRecorrido);
    public void actualizarRecorrido(long codigorecorrido, long codigopubli, String puntoPar, String puntoDes, String horaIda, String horaVuelta, String estado, String fecha, ArrayList<ParaderoRecorrido> paraderos, PreferenciaRecorrido preferencia, String fecha);
    public List<Reserva> consultarReservaPorRecorrido(long valor);
    public void cambiarEstadoReserva(Reserva reserva);
    public List<Reserva> consultarReservaPorColumna(String parametro, String valor, String tipo, long recorrido);
    public List<Recorrido> consultarRecorridoConCupos(String fechaInicio, String fechaFin, String estado, long recorrido);
    public void crearReserva(Reserva reserva, long coderecorrido);
    public int validarLogin(long codigo, String correo, String rol);
    public int cantidadRecorridosporruta(String puntopartida, String puntodestino, long conductor);
    public int cantidadRecorridosporestado(String estado, long conductor);
    public int cantidadReservasPorPublicante(long valor);
    public int cantidadReservasporEstado(String estado, long reser);
    public int cantidadRecorridosporruta(String puntopartida, String puntodestino);
    public boolean estaRepetidaReserva(long reservante, long recorrido);
    public int editarReserva(long codigoreserva, String paradero, String estado);
    public boolean validarPreferenciaPorRol(long codigoreco, String rol, String correo);
    public String obtenerNombre(BigDecimal nombre);
}

```

Figura 52. Código fuente clase iGestionBean.java

```

package negocio;

import java.math.BigDecimal;

@Stateless
public class GestionBean implements iGestionBean {

    @Inject
    ProgramaDAO programadao;
    @Inject
    RecorridoDAO recorridodao;
    @Inject
    ParaderoDAO paraderodao;
    @Inject
    PreferenciaDAO preferenciadao;
    @Inject
    ReservaDAO reservadao;
    @Inject
    LoginDao logindao;
    @Inject
    HistoricoReservaDAO historicoDao;

    public List<Programa> consultarProgramaPorColumna() {
        return programadao.consultarProgramaPorColumna();
    }

    public Programa consultarPrograma(String idPrograma) {
        return programadao.consultarPrograma(idPrograma);
    }

    public void crearRecorrido(long codigopubli, String puntoPar, String puntoDes, String horaIda, String horaVuelta, BigDecimal cupos, String estado, ArrayList<ParaderoRecorrido> paraderos, PreferenciaRecorrido preferencia, String fecha) {
        BigDecimal codpub = new BigDecimal(codigopubli);
        Recorrido recorrido = new Recorrido();
        recorrido.setCodigopublicante(codpub);
    }
}

```

Figura 53. Código fuente clase GestionBean.java

4.3.2.2.3.4. Paquete Presentación

En el paquete presentación se encuentra las clases que gestionan los componentes de la interfaz, más exactamente las paginas XHTML que se encuentra en la carpeta webapp que se detallará más adelante.

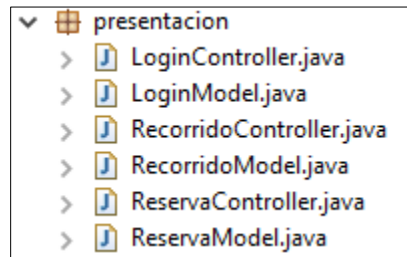


Figura 54. Clases de presentación

Para dar un ejemplo de cómo es el código fuente de las clases Managed Bean usando el framework JSF, a continuación se muestra las clases que sirven como modelo y controlador de la vista login.

```
package presentacion;

import javax.faces.bean.ManagedBean;

@ManagedBean
@SessionScoped
public class LoginModel {
    @NotEmpty(message = "El email no puede estar vacío")
    @Email(message="Formato incorrecto de Email")
    private String correo;
    private long codigo;
    private String mensaje;
    private String perfil;
    private String rol;

    public LoginModel() {}

    public String getCorreo() {
        return this.correo;
    }
    public void setCorreo(String correo) {
        this.correo=correo;
    }
    public long getCodigo() {
        return this.codigo;
    }
    public void setCodigo(long codigo) {
        this.codigo=codigo;
    }
    public String getMensaje() {
        return this.mensaje;
    }
    public void setMensaje(String mensaje) {
        this.mensaje=mensaje;
    }
    public String getPerfil() {
```

Figura 55. Código fuente clase LoginModel.java

4.3.2.2.3.4.1. Webapp

Por último en esta carpeta se encuentran los recursos web como páginas XHTML, hojas de estilo e imágenes.

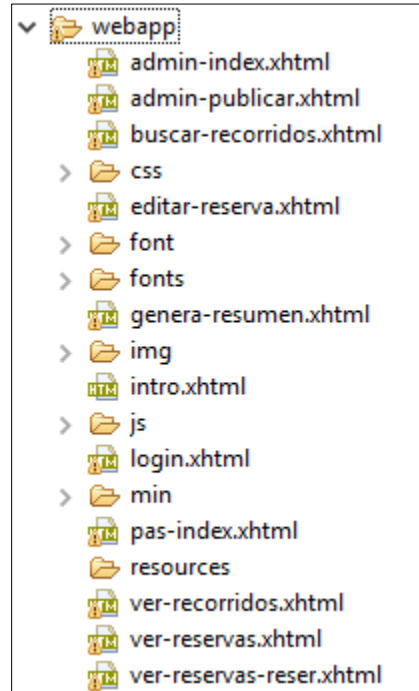


Figura 56. Recursos web

Para dar un ejemplo de cómo es el código fuente de las vistas (páginas.xhtml) usando el framework JSF, a continuación se muestra la vista de login.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/"
@<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core">
@<h:head>
  <title>login</title>
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet"/>
  <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/materi
@<style>
  body {
    display: flex;
    min-height: 100vh;
    flex-direction: column;
  }

  main {
    flex: 1 0 auto;
  }

  body {
    background: #E3F2FD;
  }

  .input-field input[type=date]:focus + label,
  .input-field input[type=text]:focus + label,
  .input-field input[type=email]:focus + label,
  .input-field input[type=password]:focus + label {
    color: #e91e63;
  }
```

Figura 57. Código fuente página login.xhtml

Con la implementación terminada se finaliza la fase de codificación y se avanza a la fase de compilación según el Script de desarrollo.

4.3.2.2.4. Compilación

La actividad número 3 en el Script de desarrollo (ver la Figura 58) es la fase de compilación sin embargo hay que considerar si el lenguaje o IDE de programación utilizado necesitan esta fase. Para este caso el IDE Eclipse JavaEE no requiere esta fase puesto que realiza la compilación automáticamente, por esta razón se avanza directamente a la siguiente fase de pruebas.



PSP0 Development Script		
Purpose	To guide the development of small programs	
Entry Criteria	• Requirements statement • Project Plan Summary form with estimated program development time • Time and Defect Recording logs • Defect Type standard	
Step	Activities	Description
1	Design	• Review the requirements and produce a design to meet them. • Record in the Defect Recording log any requirements defects found. • Record time in the Time Recording log.
2	Code	• Implement the design. • Record in the Defect Recording log any requirements or design defects found. • Record time in the Time Recording log.
3	Compile	• Compile the program until there are no compile errors. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
4	Test	• Test until all tests run without error. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
Exit Criteria	• A thoroughly tested program • Completed Time and Defect Recording logs	

Figura 58. Actividades de la compilación

4.3.2.2.5. Pruebas

La fase de pruebas tiene como objetivo ejecutar el proyecto desarrollado y verificar el funcionamiento correcto y en caso de presentarse errores, realizar los cambios pertinentes. La Figura 59 describe los pasos a llevar a cabo en esta actividad. El objetivo final es entregar una aplicación completamente funcional.

4	Test	• Test until all tests run without error. • Fix all defects found. • Record defects in the Defect Recording log. • Record time in the Time Recording log.
---	------	--

Figura 59. Actividades para la fase de pruebas

A continuación se ilustra la secuencia de pantallazos para cada una de las pruebas realizadas

4.3.2.2.6. Pruebas de funcionalidad del sistema

En estas pruebas se verifica lo que se espera en cada módulo, para esto se suministra datos de prueba de entrada y se analizan los resultados. De acuerdo a las funcionalidades especificadas en los casos de uso se realizaron los siguientes casos de prueba:

4.3.2.2.6.1. CP001. Caso de prueba Publicar Recorrido

- Descripción (CP001):

Tabla 2. Caso de prueba Registro y publicación de Recorrido

Identificación	CP 001 Registro y publicación de Recorrido	
Propósito	Publicación exitosa del Recorrido ofrecido por la Persona con Carro que comparta los asientos disponibles en su vehículo.	
Pre Requisitos		
Datos de prueba	Usuario y Contraseña	
	Campo	Valido
	Email	sergio.murillo@uillanos.edu.co
	Contraseña	sergio123
	Rol	Estudiante
	Tipo	Conductor
	Datos del Recorrido	
	Campo	Valido
	Identificación	1121905423
	Fecha	01/05/2017
	Hora Ida	8:00
	Hora Vuelta	18:00
	Punto Origen	Unicentro
	Punto Destino	Unillanos
	Paraderos	Semillano
	Cupos	4
	Datos de Preferencia	
	Campo	Valido
	Mujer	FALSE
	Hombre	TRUE
	Administrativo	FALSE
	Docente	FALSE
	Estudiante	TRUE
	Programa	
	Semestre	
Pasos	1. Ingresar a la página de inicio y loguearse 2. Ingresar los datos de prueba 3. Clic en Generar Resumen 4. Clic en Crear Recorrido	
Resultados esperados	Debe permitir registrar el recorrido junto con sus preferencias y sus paraderos para que posteriormente las Personas sin Carro puedan visualizar ese recorrido.	
Notas		

- Aplicación caso de prueba (CP001):

Paso1. Ingresamos a la página de inicio y loguearse (Ver **¡Error! No se encuentra el origen de la referencia.**)

Figura 60. Página de Inicio

Paso 2. Ingresar los datos de prueba (ver Figura 61 y Figura 62)

Paso 3. Clic en Generar Resumen (ver Figura 62)

Figura 61. Datos de prueba

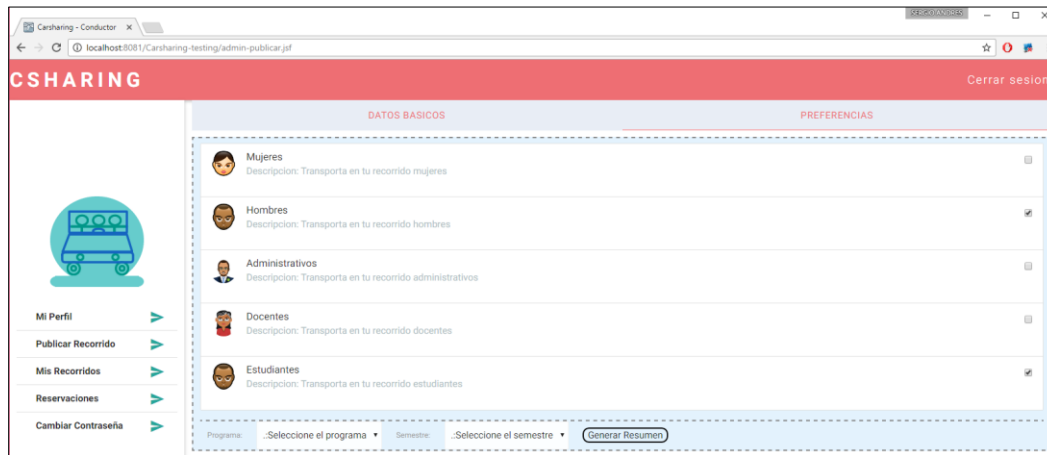


Figura 62. Preferencias

Paso 4. Clic en Crear Recorrido (ver Figura 63)

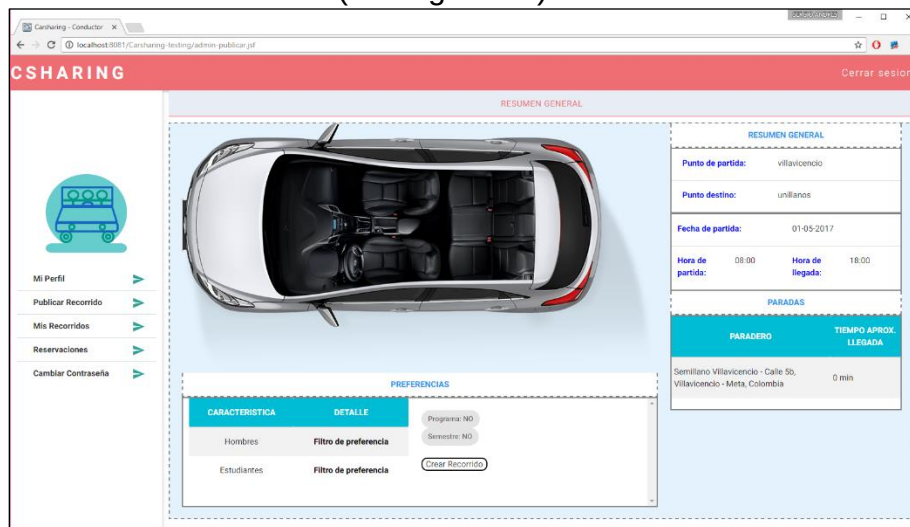


Figura 63. Recorrido listo para publicar

Resultados obtenidos (CP001): Con los datos válidos permite registrar el recorrido junto con sus preferencias y sus paraderos para que posteriormente las Personas sin Carro puedan visualizar ese recorrido.

- Con datos validos:

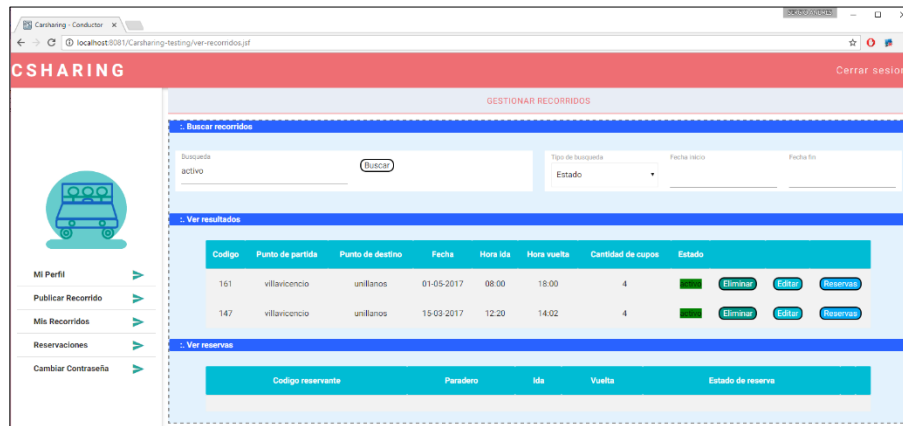


Figura 64. Recorrido creado

Página de resumen de datos una vez registrados exitosamente

- Con datos inválidos o vacíos:
 - Error cuando el punto de origen y destino coinciden

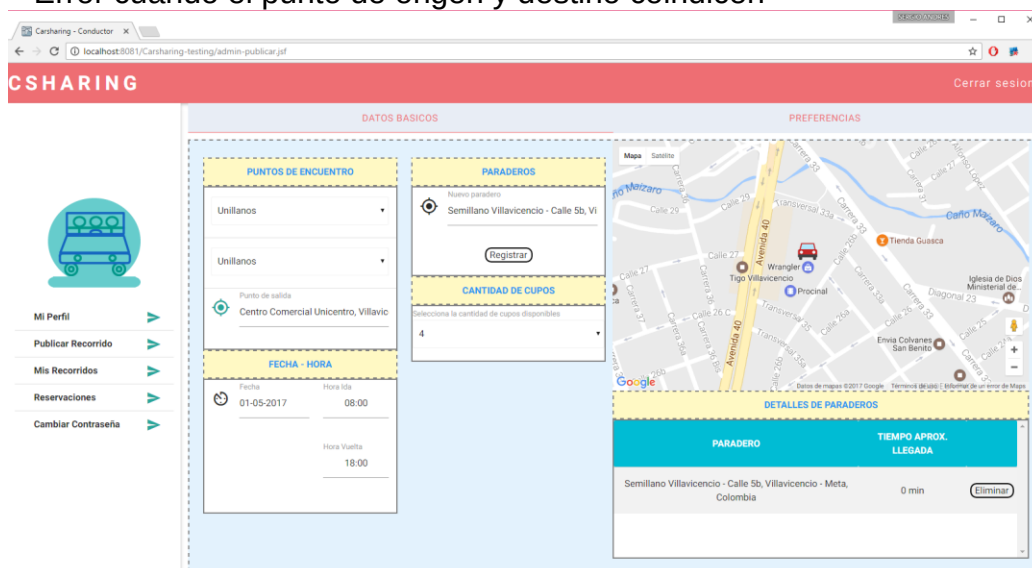


Figura 65. Punto origen coincide con destino

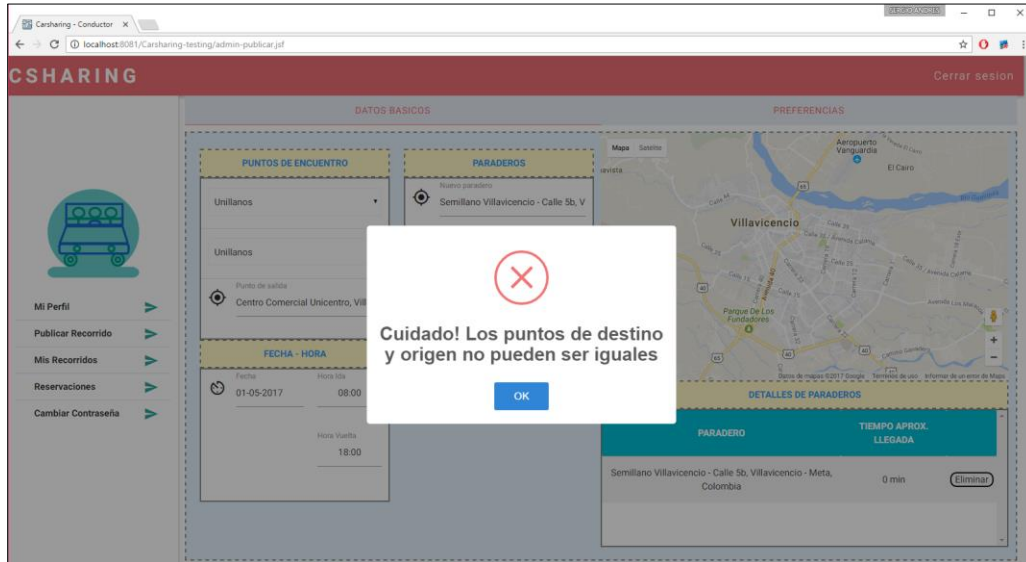


Figura 66. Error por puntos de origen y destino iguales

- Error cuando la hora de ida es mayor a la hora de vuelta

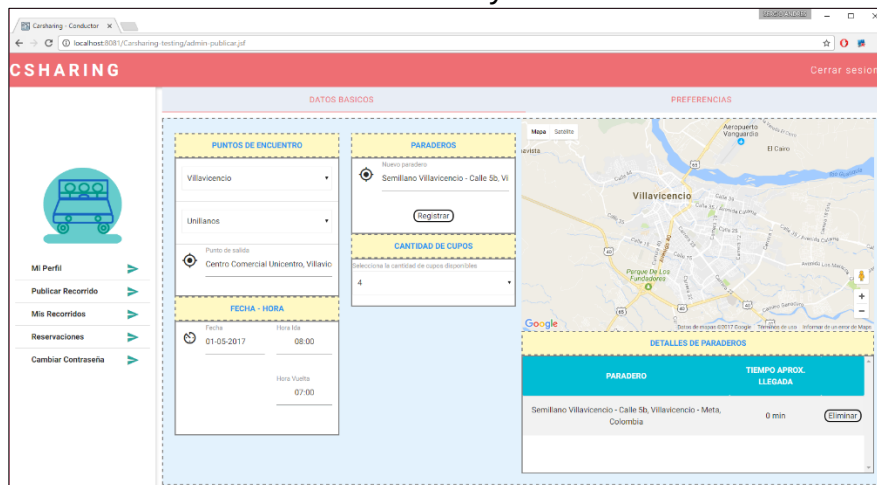


Figura 67. Hora de ida mayor hora de vuelta

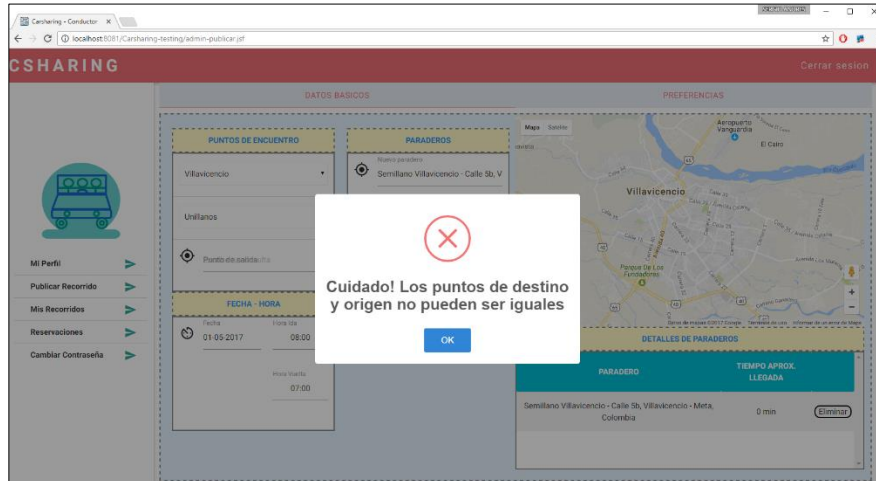


Figura 68. Error por hora de ida mayor que hora vuelta

- Error cuando no se ingresa por lo menos un paradero

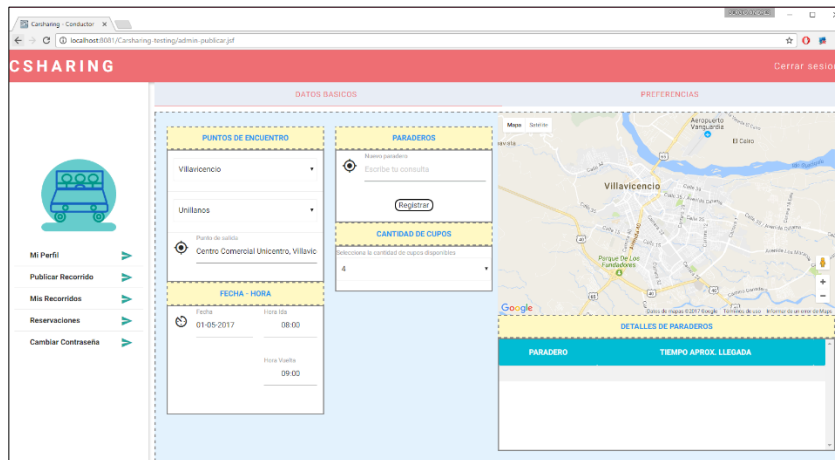


Figura 69. No ingreso de un paradero

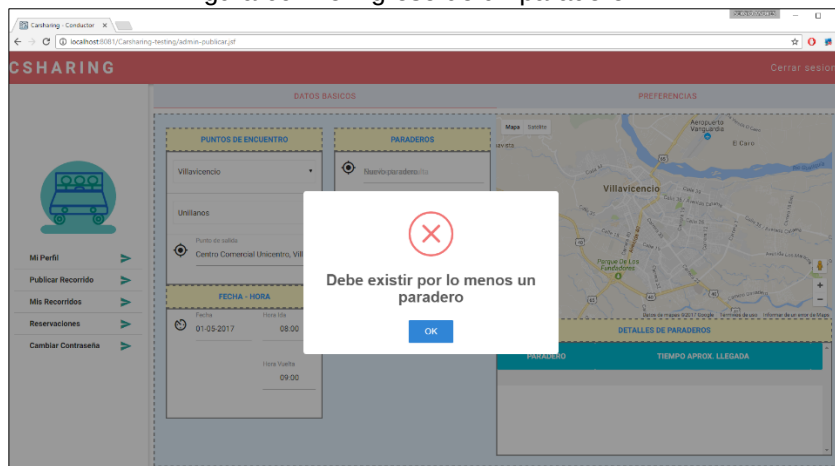


Figura 70. Error por no ingreso de un paradero

4.3.2.2.6.2 CP002. Caso de prueba Cancelar Recorrido

- Descripción (CP002):

Tabla 3. Caso de prueba anulación de Recorrido

Identificación	CP 002 anulación de Recorrido		
Propósito	Suprimir un recorrido registrado con anterioridad.		
Pre Requisitos	Recorrido creado en el sistema		
Datos de prueba	Usuario y Contraseña		
	Campo	Valido	Invalido
	Email	sergio.murillo@uillanos.edu.co	NULL
	Contraseña	sergio123	NULL
	Rol	Estudiante	NULL
	Tipo	Conductor	NULL
	Datos del Recorrido		
	Campo	Valido	Invalido
	Búsqueda	Cancelado	NULL
	Filtro	Estado	NULL
	Fecha Inicio		NULL
	Fecha Fin		NULL
Pasos	1. Ingresar a la página de inicio y loguearse 2. Ingresar datos de búsqueda 3. Ubicar el recorrido en la tabla y dar clic en el botón Cancelar		
Resultados esperados	Cambio de estado de recorrido de activo a cancelado.		
Notas			

- Aplicación del caso de prueba (CP002):

Paso1. Se ingresa a la página de inicio y loguearse (Ver **¡Error! No se encuentra el origen de la referencia.** y Figura 71)

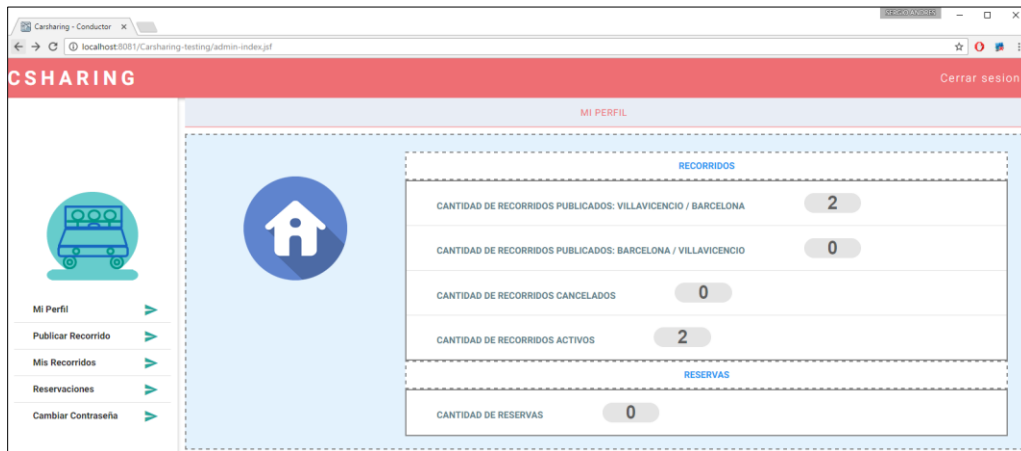


Figura 71. Perfil Persona con Carro

Paso2. Clic en Mis Recorridos

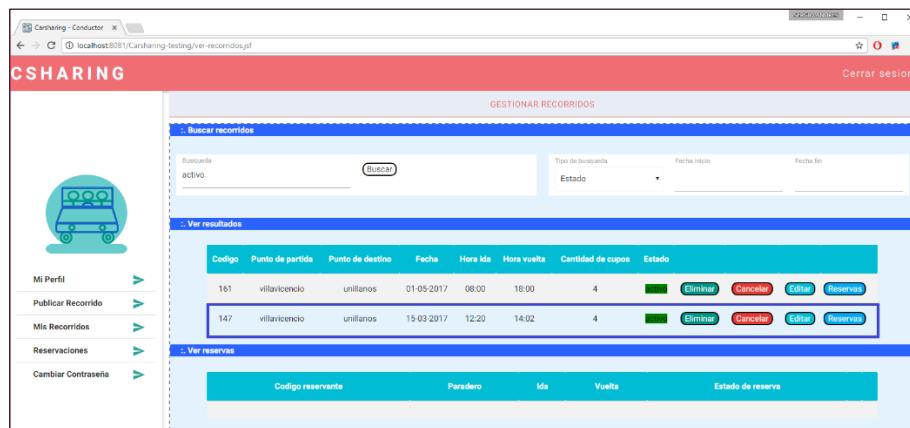


Figura 72. Recorridos activos

Paso3. Ingresar datos de búsqueda

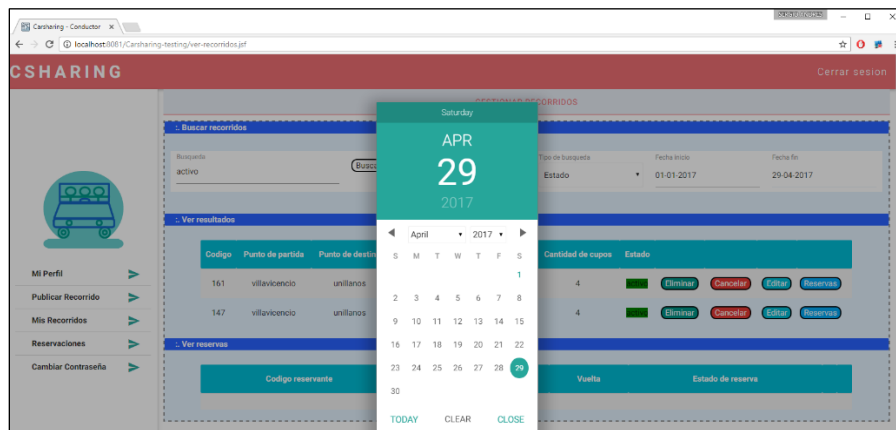


Figura 73. Ingreso datos de búsqueda

Paso4. Ubicar el recorrido en la tabla y dar clic en el botón Cancelar



Figura 74. Recorrido cancelado

- Resultados Obtenidos (CP002): Cambio de estado de recorrido de activo a cancelado.
- Con datos validos (Ver Figura 74):

4.3.2.2.6.3 CP003. Caso de prueba Buscar Recorrido

- Descripción (CP003):

Tabla 4 Caso de prueba búsqueda de Recorrido

Identificación	CP 003 Búsqueda de Recorrido	
Propósito	Resultados de búsqueda exitosos.	
Pre Requisitos	Existencia de por lo menos un recorrido e información de búsqueda correcta.	
Datos de prueba	Usuario y Contraseña	
	Campo	Valido
	Email	sergio.murillo@uillanos.edu.co
	Contraseña	sergio123
	Rol	Estudiante
	Tipo	Pasajero
	Datos del Recorrido	
	Campo	Valido
	Búsqueda	Villavicencio
	Filtro	Punto de partida
	Fecha Inicio	
	Fecha Fin	
Pasos	1. Ingresar a la página de inicio y loguearse 2. Ingresar los datos de prueba 3. Clic en Buscar	

Resultados esperados	Debe permitir desplegar los recorridos disponibles.
Notas	

- Aplicación caso de prueba (CP003):

Paso1. Ingresar a la página de inicio y loguearse (Ver **¡Error! No se encuentra el origen de la referencia.**)

Paso2. Ingresar los datos de prueba

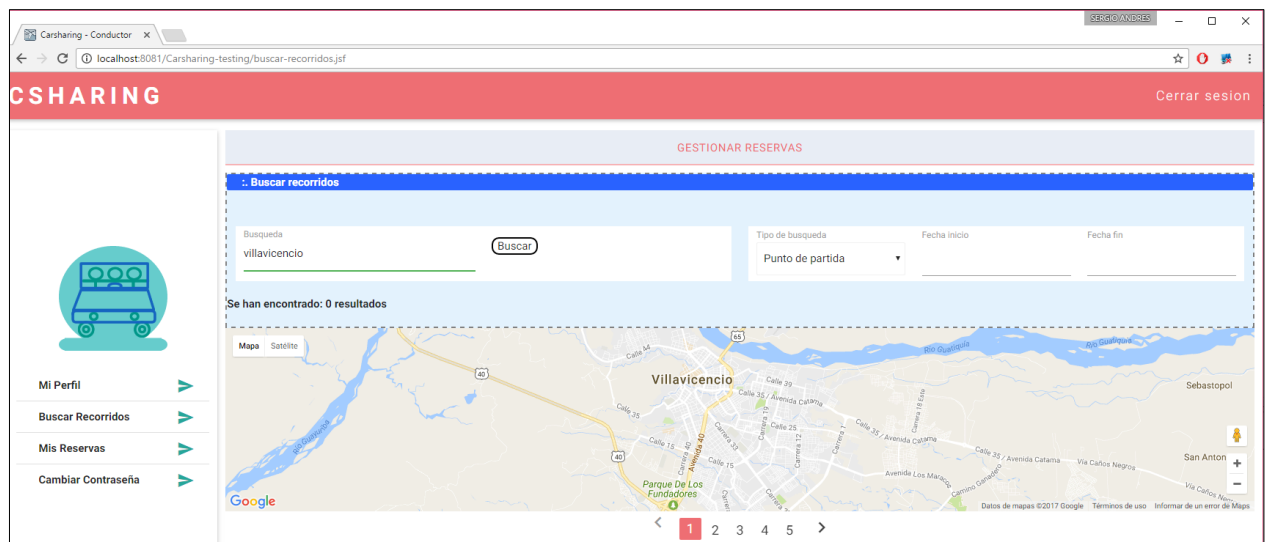


Figura 75. Datos de prueba ingresados

Paso3. Clic en Buscar

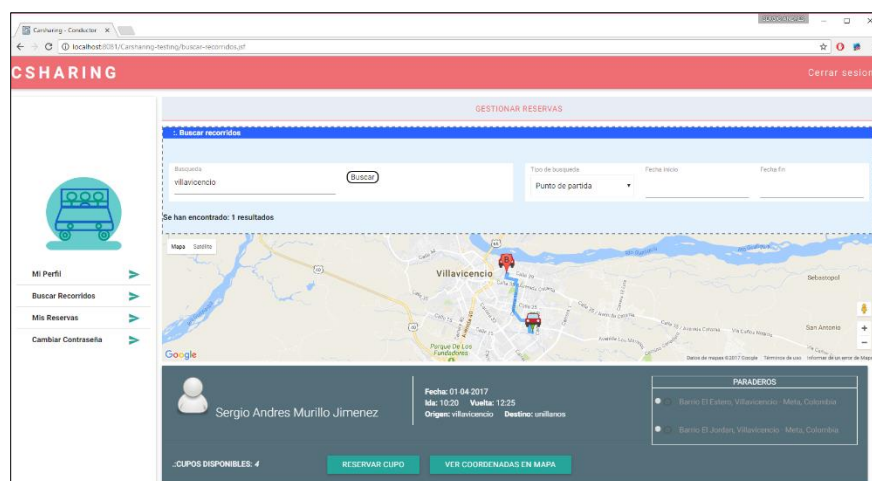


Figura 76. Recorrido encontrado

Resultados obtenidos (CP003): Se encontraron de manera satisfactoria los recorridos disponibles.

- Con datos validos (Ver Figura 76):

4.3.2.2.6.4 CP004. Caso de prueba Reservar Cupo

- Descripción (CP004):

Tabla 5 Caso de prueba reserva de cupo

Identificación	CP 004 Reserva de Cupo		
Propósito	Apartar un cupo en un determinado recorrido.		
Pre Requisitos	Existencia de por lo menos un recorrido y debe estar activo.		
Datos de prueba	Usuario y Contraseña		
	Campo	Valido	Invalido
	Email	sergio.murillo@uillanos.edu.co	NULL
	Contraseña	sergio123	NULL
	Rol	Estudiante	NULL
	Tipo	Pasajero	NULL
	Datos del Recorrido		
	Campo	Valido	Invalido
	Búsqueda	Villavicencio	NULL
	Filtro	Punto de partida	NULL
	Fecha Inicio		NULL
	Fecha Fin		NULL
Pasos	1. Ingresar a la página de inicio y loguearse 2. Ingresar los datos de prueba 3. Clic en Buscar 4. Clic en Reservar Cupo		
Resultados esperados	Reserva de cupo exitosa		
Notas			

- Aplicación caso de prueba (CP004):

Paso1. Ingresar a la página de inicio y loguearse (Ver **¡Error! No se encuentra el origen de la referencia.**)

Paso2. Ingresar los datos de prueba (Ver Figura 75)

Paso3. Clic en Buscar (Ver Figura 76)

Paso4. Clic en Reservar Cupo (Ver Figura 77)

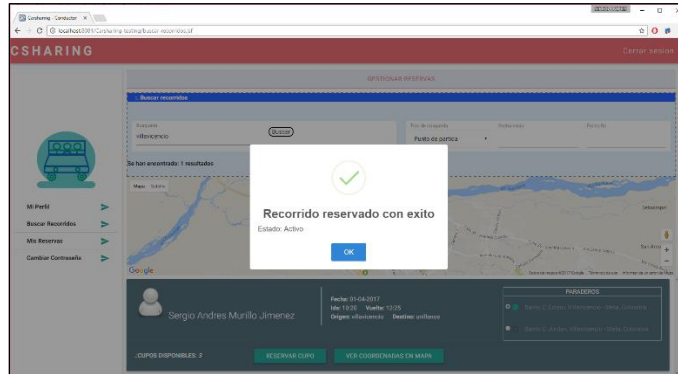


Figura 77. Recorrido reservado.

Resultados obtenidos (CP004): Se reserva de manera satisfactoria un cupo en un recorrido disponible.

- Con datos validos (Ver Figura 77):
- Con datos inválidos o vacíos:
 - Error cuando no se selecciona al menos un paradero

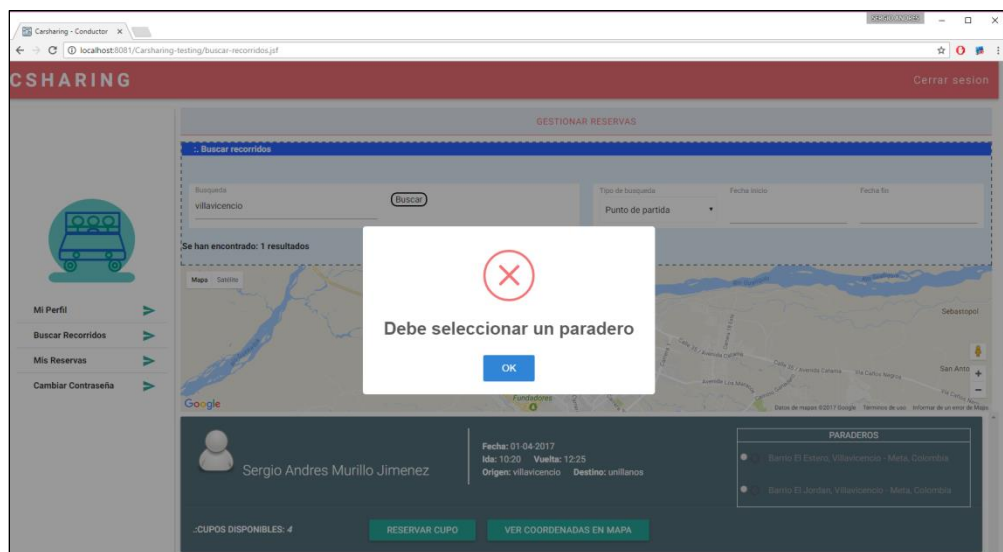


Figura 78. Error al no seleccionar un paradero

- Error cuando ya se hizo una reserva en un recorrido, y se quiere volver a reservar en el mismo recorrido

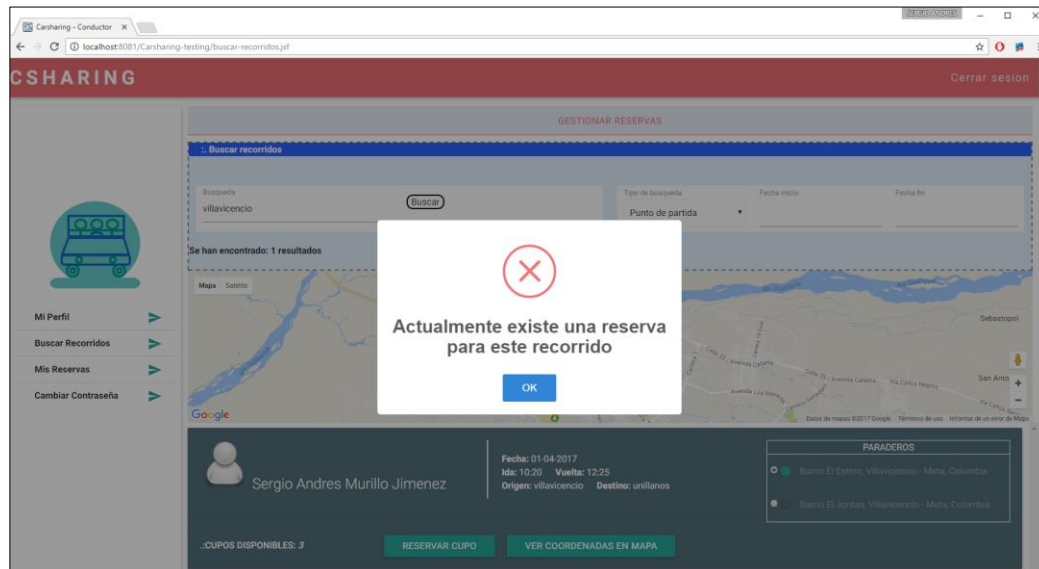


Figura 79. Error cuando hay una reserva en un recorrido y se intenta volver a reservar

4.3.2.2.6.5 CP005. Caso de prueba Cancelar Cupo Reservado

- Descripción (CP005):

Tabla 6 Caso de prueba cancelación de cupo reservado

Identificación	CP 005 Cancelación de cupo reservado	
Propósito	Cancelar un cupo que previamente se había reservado.	
Pre Requisitos	Existencia de la reserva.	
Datos de prueba	Usuario y Contraseña	
	Campo	Valido
	Email	sergio.murillo@uillanos.edu.co
	Contraseña	sergio123
	Rol	Estudiante
	Tipo	Pasajero
	Datos del Recorrido	
	Campo	Valido
	Búsqueda	Activo
	Filtro	Estado
	Fecha Inicio	
	Fecha Fin	
Pasos	1. Ingresar a la página de inicio y loguearse 2. Ingresar los datos de prueba 3. Clic en Buscar 4. Clic en Cancelar Reservar Cupo	
Resultados esperados	Cancelación de reserva de cupo de manera exitosa.	

Notas	
-------	--

- Aplicación caso de prueba (CP005):

Paso1. Ingresar a la página de inicio y loguearse (Ver **¡Error! No se encuentra el origen de la referencia.**)

Paso2. Ingresar los datos de prueba (Ver Figura 75)

Paso3. Clic en Buscar (Ver Figura 76 y Figura 80)

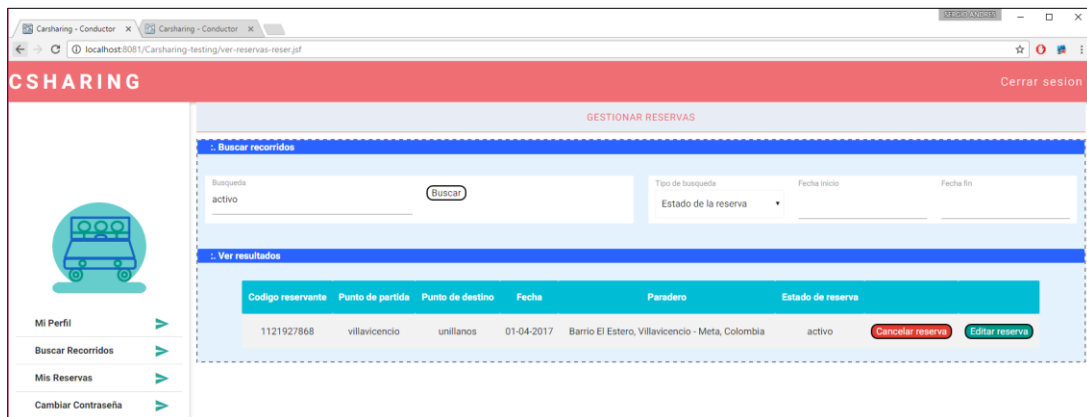


Figura 80. Recorrido activo

Paso4. Clic en Cancelar Reservar Cupo



Figura 81. Cupo de reserva cancelado

4.3.2.2.6.6 CP006. Caso de prueba Cancelar Reserva de Persona en su Recorrido

- Descripción (CP006):

Tabla 7 Caso de prueba cancelación de reserva de persona en su recorrido

Identificación	CP 006 Cancelación de reserva de persona en su recorrido		
Propósito	Cancelar un cupo que previamente se había reservado.		
Pre Requisitos	Existencia de un recorrido y por lo menos una reserva hecha por parte de la persona sin carro		
Datos de prueba	Usuario y Contraseña		
	Campo	Valido	Invalido
	Email	sergio.murillo@uillanos.edu.co	NULL
	Contraseña	sergio123	NULL
	Rol	Estudiante	NULL
	Tipo	Conductor	NULL
	Datos del Recorrido		
	Campo	Valido	Invalido
	Búsqueda	Activo	NULL
	Filtro	Estado	NULL
	Fecha Inicio		NULL
	Fecha Fin		NULL
Pasos	1. Ingresar a la página de inicio y loguearse 2. Ingresar los datos de prueba 3. Clic en Buscar 4. Clic en Cancelar Reservar Cupo		
Resultados esperados	Cancelación de reserva de cupo de manera exitosa.		
Notas			

- Aplicación caso de prueba (CP006):

Paso1. Ingresar a la página de inicio y loguearse (Ver **¡Error! No se encuentra el origen de la referencia.**)

Paso2. Ingresar los datos de prueba (Ver Figura 75)

Paso3. Clic en Buscar (Ver Figura 82)

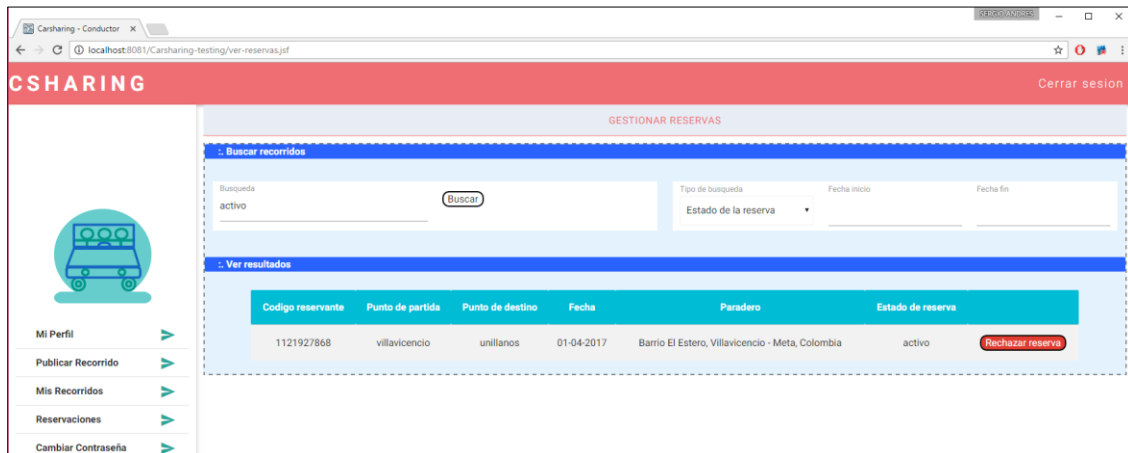


Figura 82. Recorrido activo

Paso4. Clic en Cancelar Reservar Cupo

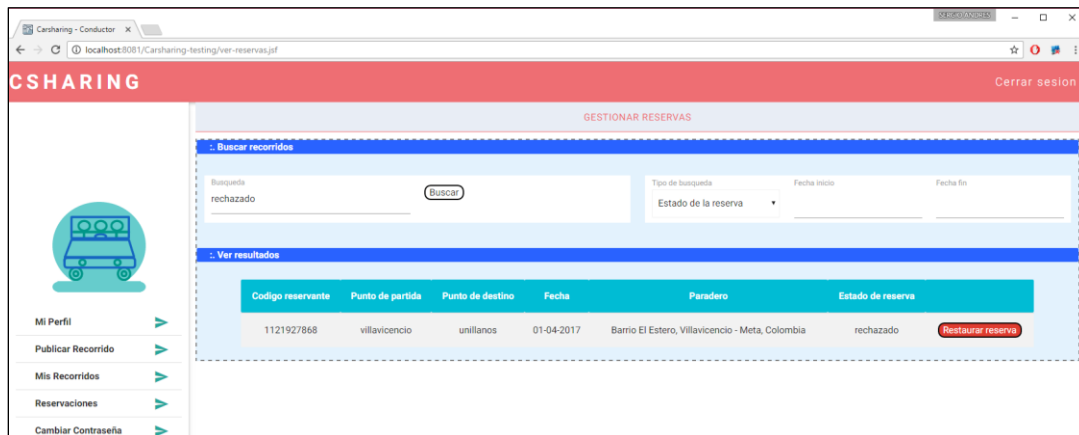


Figura 83. Cupo de reserva cancelado

Resultados obtenidos (CP006): Cancelación de reserva de cupo de manera exitosa.

- Con datos validos (Ver Figura 83):

4.4 DEMOSTRACION

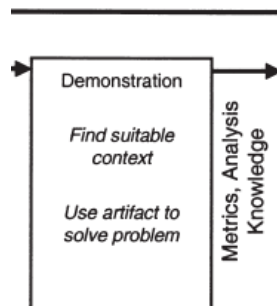


Figura 84. Fase de demostración. Adaptada de [38]

La demostración del proyecto se realizó a lo largo del desarrollo del mismo, en reuniones de seguimiento. En estas reuniones se utilizó un instrumento en el cual se registra la agenda, su desarrollo, las decisiones más importantes y un resumen de tareas a realizar para la siguiente reunión. Parte del instrumento se muestra en la Figura 85.



CAR-SHARING

 Aplicación para compartir el carro entre miembros de una comunidad

 Caso de estudio: Universidad de los Llanos

No. 1	Lugar: CiEnTiC	Fecha: 14/07/2016	Hora:	Inicio: 01:00 pm	Fin: 02:00 pm
-------	-----------------------	-------------------	-------	------------------	---------------

AGENDA

- Acordar aspectos administrativos del desarrollo del proyecto.
- Planeación del desarrollo.

DESARROLLO

- Acordar aspectos administrativos del proyecto.
 - Organización de carpetas
 - Asuntos administrativos
 - Presentación
 - Imágenes
 - Implementación, etc.
 - Revisar formato de seguimiento.
 - Establecer forma de trabajo del proyecto: Seguimiento cada 8 días, a la misma hora. Reuniones de 20 minutos (ejecutivas).
- Planeación del desarrollo.
 - Aspectos técnicos del desarrollo
 - Un gestor de versiones en la nube:
 - Administrador: Sergio
 - Cientes: Sergio, Jonathan, **olivegari**
 - Desarrollo locales en las máquinas de cada uno.
 - Tecnología.
 - Servidor de aplicaciones Wild-**fly**
 - JSP-EJB-JPA
 - PostgreSQL**
 - Conexión con ORACLE – Repositorio de usuarios autorizados
 - Conexión con Google **accounts**
 - Solución móvil – **PhoneGap** (es posible otras).
 - Artefactos
 - Modelo de negocio
 - Glosario
 - Mapa conceptual
 - Procesos de negocio BPMN AS-IS
 - Modelo de Arquitectura solución
 - Procesos de negocio BPMN TO-BE.
 - Diagramas de casos de uso.
 - Mockup**
 - Casos de prueba (escenarios con datos).
 - Diagrama de clases entidad.

Control de versión: OLVM-DV00 | No. 01 | Fecha: 20/06/2013 | Adaptación fuente: Ministerio de TIC, proyecto gobierno en línea.
 Edición de Ayudas Educativas, 2ª pila, cientic@unllanos.edu.co

Figura 85. Instrumento de seguimiento

4.5 EVALUACIÓN

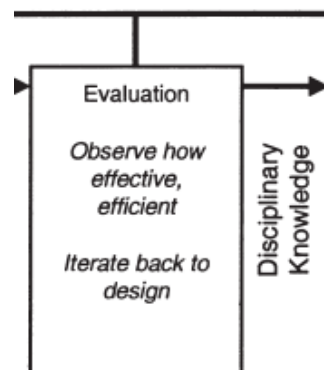


Figura 86. Fase de demostración. Tomada de [38]

La evaluación de este proyecto se realizó mediante la aplicación de varias pruebas piloto en un entorno local simulando el entorno de producción, obteniendo los resultados esperados, ejecutando paso a paso los casos de uso reflejados en los casos de prueba.

4.6 COMUNICACION

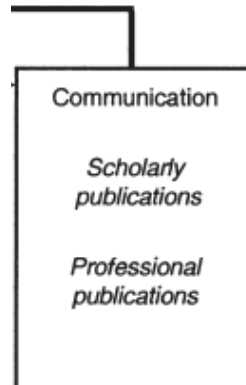


Figura 87. Fase de comunicación. Tomada de [38]

En esta última fase se comunica el problema solucionado y su importancia, la descripción del producto final, su utilidad, novedad, el diseño etc.

Para este proyecto toda esta información está plasmada en el informe final del proyecto y el artículo científico elaborado.

5. RESULTADOS

Los resultados que se obtuvieron con este proyecto se han clasificado siguiendo los lineamientos de la Dirección General de Investigaciones y de la DSR (Design Science Research), a saber:

1. Relacionados con la generación de conocimiento y/o nuevos desarrollos tecnológicos.
2. Conducentes al fortalecimiento de la capacidad científica institucional o nacional.
3. Dirigidos a la apropiación social del conocimiento.

A continuación se detallan siguiendo la anterior clasificación.

- 5.1. Relacionados con la generación de conocimiento y/o nuevos desarrollos tecnológicos:

Tabla 8 Generación de nuevo conocimiento

Resultado/Producto esperado	Indicador	Beneficiario
Un producto tecnológico de software.	Una aplicación que permite a estudiantes, profesores y administrativos ofrecer cupos en su carro, para compartirlos con quienes requieren movilizarse desde y hacia la universidad.	Comunidad académica Universidad de los Llanos. Comunidad académica de organizaciones que personalicen el software para sus integrantes.

- 5.2. Conducentes al fortalecimiento de la capacidad científica institucional o nacional:

Tabla 9 Fortalecimiento de la comunidad

Resultado/Producto esperado	Indicador	Beneficiario
2 productos de formación en grado.	2 actas de grado de ingeniería de Sistemas.	Programa de Ingeniería de Sistemas.

- 5.3. Dirigidos a la apropiación social del conocimiento:

Tabla 10 Apropiación social del conocimiento

Resultado/Producto esperado	Indicador	Beneficiario
Un artículo científico	Un documento de recepción de artículo para sometimiento a publicación.	Programa de Ingeniería de Sistemas de la Universidad de los Llanos, Comunidad académica de Ingeniería de Software.

6. CONCLUSIONES Y RECOMENDACIONES

Las conclusiones de este trabajo deben dividirse en dos tipos, por un lado los logros alcanzados para la Universidad y por el otro los logros obtenidos desde la ingeniería de software aplicando la investigación para lograr el conocimiento y aplicación de tecnologías abiertas emergentes para soluciones como la creada para este caso.

Finalmente, el trabajo realizado permite hacer un conjunto de recomendaciones tanto para el desarrollo de software para la gestión académica de la universidad, como para la aplicación de la ingeniería de software en general.

6.1 Conclusiones desde el dominio del negocio.

El objetivo de este proyecto era desarrollar un aplicativo que permitiera a la comunidad de Unillanos compartir los asientos disponibles en los vehículos particulares, para transportarse desde y hacia la universidad. Contribuyendo a mitigar la problemática que existe actualmente, siendo esta solución un beneficio y una ventaja para todos los miembros de la Universidad de los Llanos. Este objetivo general fue cumplido a cabalidad.

Se evaluaron, analizaron e implementaron todos los posibles casos alternativos que pudiesen ocurrir entre el usuario y la aplicación para así mismo evitar posibles alteraciones e incidentes que comprometieran la funcionalidad estable del sistema.

Al evaluar las encuestas realizadas a integrantes de la comunidad Unillanista, se llegó a la conclusión de que hay una gran probabilidad de acogida (91%) y por tanto de éxito en el uso del sistema.

6.2 Conclusiones desde la ingeniería de software.

El proceso investigativo guiado por la Design Science Research (DSR) permitió que tanto la identificación de las necesidades como el diseño y desarrollo de artefactos que permitieran su realización, fuera un proceso controlado, formal y permanentemente aprobado por los involucrados en la problemática que se intentaba resolver.

Ceñir el trabajo a las etapas de Definición del problema, Definición de los objetivos, Diseño y Desarrollo, Demostración, Evaluación y Comunicación, aunque implicó un mayor esfuerzo, además de imprimirle calidad a los resultados, facilitó la apropiación de la cultura de investigación en el quehacer del desarrollo de software.

Con toda seguridad con este proceso se ha pasado de ser un desarrollador de software a ser un investigador en la ingeniería de software. Se conoce qué artefactos son fundamentales para lograr la calidad en el producto, cuáles permiten que los involucrados estén apropiados de su papel fundamental como co-creadores de la solución (desde el instrumento de seguimiento, hasta los desarrollos en la etapa de pruebas, pasando por el glosario, el modelo conceptual y los diagramas de procesos de negocios).

Por último y no menos importante, con este proceso se descubrió grandes potencialidades para el desarrollo de software en arquitectura en 3 capas con la tecnología Java EE, ya que contar con componentes de código distribuidos de acuerdo a la funcionalidad que cumplen (Sea de interfaz de usuario, negocio de la aplicación o modelo datos), facilita entre otras cosas la posibilidad de realizar modificaciones sin afectar toda la aplicación, debido a que cada capa es independiente de la otra, también es una tecnología escalable que puede ampliarse con facilidad en caso que aparezcan nuevos requerimientos.

Se afirma con total seguridad que una aplicación en Java EE, con los framework JSF para la capa de presentación, EJB para la capa de negocio y JPA para la capa de datos, permiten crear funcionalidades complejas las cuales de una forma sencilla y ágil pueden ser extendidas, mantenidas y escaladas.

6.3 Recomendaciones.

Aunque se realizó un estudio previo con estudiantes, profesores y administrativos de la Universidad de los Llanos, se recomienda llevar a cabo un segundo estudio a profundidad por parte de la universidad, en aras de poner en funcionamiento la aplicación a la comunidad universitaria, con el fin de conocer el impacto que tendrá este servicio en la problemática evidente de transporte desde y hacia la universidad. A modo de sugerencia se presentan las siguientes preguntas:

¿Cómo miembro de la comunidad Unillanos, usted usaría CARSHARING en?

- a) Computador
- b) Dispositivo Móvil
- c) Ambos
- d) No la usaría.

¿A su criterio, cual es el principal beneficio de usar una aplicación como CARSHARING?

- a) Viajar más cómodo
- b) Proteger el medio ambiente
- c) Generar interacciones entre usuarios
- d) Cooperación entre la comunidad Unillanos.

¿Cuál considera que es el mayor beneficio de usar CARSHARING en dispositivos

móviles?

- a) Accesibilidad en cualquier lugar
- b) Rapidez con la que se accede al servicio
- c) Facilidad en el uso
- d) Otro, ¿cuál?

¿Por cuál dificultad usted sería usuario de CARSHARING?

- a) La ruta se demora mucho
- b) En ocasiones voy en la ruta de pie
- c) Pocos horarios de rutas en el día
- d) Otra, ¿cuál?

¿Con qué frecuencia usaría CARSHARING?

- a) Todos los días
- b) Una a dos veces a la semana
- c) Tres a cuatro veces a la semana
- d) No la usaría.

Entre otras preguntas que considere pertinente la Universidad. Para obtener resultados confiables se recomienda que el estudio lo realicen profesionales en el área de mercadeo, ya que son los más competentes para este tipo de tareas.

Adquirir la licencia del Api de Google Maps (Geocoding) para obtener un número mayor de consultas al día, ya que el límite gratuito actual no puede sobrepasar las 2.500 solicitudes diarias. Evitando así los fallos en los módulos que competen al uso del Api en la aplicación.

Adicionar un módulo donde el pasajero pueda visualizar el trayecto que sigue el conductor en tiempo real. Con el fin de estimar los tiempos de encuentro a los paraderos que previamente han establecido tanto conductor como pasajero.

Vincular este desarrollo a las otras comunidades (empresas o instituciones educativas), principalmente a las que se radican por fuera del casco urbano de la ciudad.

Usar la metodología Design Science Research (DSR) en los proceso investigativos en el área de la Ingeniería de Software.

Usar la metodología de desarrollo de software TSP-PSP para construir productos de software funcionales y de alta calidad.

Con los datos recogidos en el desarrollo de este trabajo, se puede avanzar a la aplicación de la metodología PSP en su nivel PS1.

Utilizar la tecnología Java EE para aplicaciones de software que tengan

características similares a este proyecto como por ejemplo una lógica del Negocio particular, grandes cantidades de datos persistentes, acceso concurrente a los datos, diferentes tipos de usuario e interfaces de usuario etc.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] Oracle, [En línea]. Available:
<http://www.oracle.com/technetwork/java/javaee/overview/index.html>.
- [2] Uniandes, «PortalViaje - Uniandes,» Universidad de los Andes, 14 10 2015. [En línea]. Available: <https://viaje.uniandes.edu.co/PortalViajeReactJS/>. [Último acceso: 17 02 2016].
- [3] Uniandes, «Carpooling - Campus - Universidad de los Andes,» 14 10 2015. [En línea]. Available:
<https://campusinfo.uniandes.edu.co/es/comollegar/carpooling>. [Último acceso: 17 02 2016].
- [4] C. F. Pardo, «Los cambios en los sistemas integrados de transporte masivo en las principales ciudades de América Latina,» Naciones Unidas, 02 2009. [En línea]. Available: <http://es.slideshare.net/VirtualEsumer/cambios-en-sistemas-integrados-de-transporte>. [Último acceso: 21 02 2016].
- [5] Universidad de los Andes, «La triste historia del transporte público en Bogotá,» Bogotá, 2015.
- [6] D. J. P. M, «EL TIEMPO,» 18 08 2015. [En línea]. Available:
<http://www.eltiempo.com/colombia/medellin/transporte-masivo-en-medellin-a-2016/16252796>. [Último acceso: 17 02 2016].
- [7] Universidad de los Llanos - Oficina de Planeación - Área estadística, «Universidad de los Llanos - Planeación,» 25 05 2016. [En línea]. Available:
http://190.60.95.15:8080/pentaho/Home?userid=estadisticas&password=estadisticas&locale=es_CO. [Último acceso: 25 05 2016].
- [8] Universidad de los Llanos - Oficina de Planeación - Área estadística, «Universidad de los Llanos - Planeación,» 25 05 2016. [En línea]. Available:
http://190.60.95.15:8080/pentaho/Home?userid=estadisticas&password=estadisticas&locale=es_CO. [Último acceso: 25 05 2016].
- [9] Universidad de los Llanos - Oficina de Planeación - Área estadística, «Universidad de los Llanos - Planeación,» 25 05 2016. [En línea]. Available:
http://190.60.95.15:8080/pentaho/Home?userid=estadisticas&password=estadisticas&locale=es_CO. [Último acceso: 25 05 2016].
- [10] R. Dowling y J. Kent, «Practice and public-private partnerships in sustainable transport governance: The case of car sharing in Sydney, Australia,» *Transport Policy*, vol. 40, pp. 58-64, 17 03 2015.
- [11] P. Baptista, S. Melo y C. Rolima, «Energy, Environmental and Mobility Impacts of Car-sharing Systems. Empirical Results from Lisbon, Portugal,» *Procedia - Social and Behavioral Sciences*, vol. 111, p. 28-37, 18 02 2014.

- [12] K. Steininger y G. Bachner, «Extending car-sharing to serve commuters: An implementation in Austria,» *Ecological Economics*, vol. 101, p. 64–66, 20 03 2014.
- [13] T. S. Foo , «Vehicle ownership restraints and car sharing in Singapore,» *Habitat International*, vol. 24, p. 75–90, 19 11 1999.
- [14] R. S. Pressman, *Ingeniería de Software - Un enfoque práctico*, Quinta ed., Aravaca, Madrid, Moncloa-Aravaca: McGRAW-HILL/INTERAMERICANA DE ESPAÑA. S.A.U., 2002, p. 29.
- [15] B. W. Bohem, «Software Engineering,» *IEEE Transactions on Computers*, vol. 25, nº 12, pp. 1226-1241, 12 1976.
- [16] R. Menéndez y A. Barzanallana, «Universidad de Murcia - Actualidad Informática,» Departamento de Informática y Sistemas. Universidad de Murcia, 10 08 2012. [En línea]. Available: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>. [Último acceso: 21 02 2016].
- [17] JAVA, *The Java Language Specification*, 8 ed., California: Oracle America, Inc. , 2015, p. 21.
- [18] «Java Enterprise Edition,» 2012. [En línea]. Available: <http://www.jatun.com/web/company/training/javaee5>. [Último acceso: 22 02 2016].
- [19] Global Mentoring, «Curso Java Enterprise Edition Online (Java EE),» Global Mentoring - Experiencia y conocimiento para tu vida, 01 2016. [En línea]. Available: <http://globalmentoring.com.mx/curso-javaee/>. [Último acceso: 28 02 2016].
- [20] Java Community Process, «Java Specification Requests by Platform,» Community Development of Java Technology Specifications, [En línea]. Available: <https://www.jcp.org/en/jsr/platform?listBy=3&listByType=platform>. [Último acceso: 28 02 2016].
- [21] C. Á. Caules, *Arquitectura Java Sólida*, Lulu.com, 2012.
- [22] World Wide Web Consortium, «W3 DEV CAMPUS,» World Wide Consortium, 2013. [En línea]. Available: <http://www.w3c.es/w3devcampus/cursos/w3c-html5/>. [Último acceso: 25 02 2016].
- [23] World Wide Web Consortium, «W3 DEV CAMPUS,» World Wide Web Consortium, 2013. [En línea]. Available: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>. [Último acceso: 25 02 2016].
- [24] B. Eich, «Brendan Eich,» [En línea]. Available: <https://brendaneich.com/?s=javascript>. [Último acceso: 26 02 2016].
- [25] World Wide Web Consortium, «W3 DEV CAMPUS,» World Wide Web Consortium, 2013. [En línea]. Available: <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>. [Último acceso: 14 03 2016].

- [26] M. A. C. Chávez, «Grid Morelos,» 01 2010. [En línea]. Available: <http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/bd1.pdf>. [Último acceso: 22 02 2016].
- [27] E. Q. Anko, «SlideShare - Aplicaciones Móviles Phonegap,» 14 06 2015. [En línea]. Available: <http://es.slideshare.net/corphqelar/aplicaciones-moviles-phonegap>. [Último acceso: 22 02 2016].
- [28] A. Hevner, S. March, J. Park y S. Ram, «Desing Science Research in Information Systems,» *Management Information Systems Quarterly*, vol. 28, nº 1, pp. 75-105, 2004.
- [29] H. A. Simon, *The Sciences of the Artificial*, Cambridge, MA: MIT Press, 1996.
- [30] V. Vaishnavi y B. Kuechler, «Design Science Research in Information Systems,» 20 01 2004. [En línea]. Available: <http://desrist.org/design-research-in-information-systems/>. [Último acceso: 22 02 2016].
- [31] K. Peffers, T. Tuunanen, M. A. Rothenberger y S. Chatterjee, «A Design Science Research Methodology for Information Systems Research.,» *Journal of Management Information Systems*, pp. 45-77.
- [32] INGENIARE, «Arquitectura orientada a servicios para software de apoyo para el proceso personal de software.,» *Revista Chilena de Ingeniería*, vol. 19, nº 1, pp. 40-53, Enero-2011.
- [33] S. R. Schach, «Object - Oriented and Classical Software Engineering,» 2011.
- [34] W. S. Humphrey, *PSP A Self-Improvent Process for Software Engineeers*, Westford: Addison Wesley, 2013.
- [35] The Software Process Dashboard Initiative, 10 2014. [En línea]. Available: <http://www.processdash.com/download>.
- [36] S. R. Schach, «Object -Oriented and Classical Software Engineering,» Mc Graw Hill, 2011, p. 667.
- [37] Watts S. Humphrey, [En línea].
- [38] K. Peffers, T. Tuunanen, M. A. Rothenberger y . S. Chatterjee, «A Design Science Research Methodology for Information Systems Research.,» *Journal of Management Information Systems*, vol. 24, nº Issue 3, pp. 45-77, Winter 2007/2008.

8. ANEXOS

8.1 Anexo A: Glosario



Contenido

GLOSARIO	2
• Universidad de los Llanos (Unillanos):	2
• Estudiante:	2
• Profesor:	2
• Administrativo:	2
• Transporte:	2
• Bus:	2
• Paradero:	2
• Ruta:	2
• Buseta:	2
• Taxi:	2
• Motocicleta:	2
• Automóvil:	3
• Cupo:	3
BIBLIOGRAFIA	3



Universidad de los Llanos
Ingeniería de Sistemas
Car Sharing
Caso de estudio: Universidad de los Llanos

GLOSARIO

- **Universidad de los Llanos (Unillanos):** Universidad del Meta que se encuentra ubicada en el Kilómetro 12 vía a Puerto López, en la Vereda Barcelona.
- **Estudiante:** *'Se considera estudiante de la Universidad, la persona que posee matrícula vigente para un programa académico, debiéndose registrar por la Constitución Política, las Leyes, el reglamento estudiantil y demás normas legales concordantes. La condición de estudiante se pierde en los casos determinados en el respectivo reglamento estudiantil'.* [1]
- **Profesor:** *'Persona que se dedica profesionalmente a la enseñanza de algo.'* [2]. En este proyecto, profesor es la persona que dicta por lo menos un curso a los estudiantes de la Universidad de los Llanos.
- **Administrativo:** Persona que trabaja en la Universidad de los Llanos desempeñando un cargo diferente al de profesor, Ej: Vigilante o Rector.
- **Transporte:** El transporte que ofrece la Universidad de los Llanos está dado por ruta de buses, mediante el contrato No. 1872 de 2016, el cual varía cada semestre debido a una convocatoria abierta realizada por la Universidad a las empresas interesadas en prestar este servicio, los cuales presentan varios inconvenientes, entre ellos: no cubren toda la ciudad, los horarios son limitados, algunos horarios presentan sobrecupo en los buses, algunas rutas incumplen los horarios establecidos. Por otra parte, también se cuenta con rutas de servicio público y servicio de taxis, los cuales acarrearán un costo adicional.
- **Bus:** *'Vehículo automóvil con capacidad para gran número de viajeros, destinado al transporte de pasajeros por carretera.'* [3]. En este proyecto, los buses son contratados para recoger estudiantes en paraderos, cumpliendo con unas rutas establecidas en unos horarios predefinidos.
- **Paradero:** Para este caso es el lugar en donde paran los buses a recoger estudiantes, profesores o personal administrativo con destino a Unillanos o a Villavicencio.
- **Ruta:** Trayecto que toman los buses para ir a o volver de la Universidad.
- **Buseta:** *'Vehículo con capacidad para varios pasajeros destinado a recorrer un trayecto fijo dentro de una población'.* [4]. En este proyecto, la buseta es aquella que se dirige a o vuelve de la vereda Barcelona. Ésta pasa en intervalos aproximados de media hora.
- **Taxi:** *'Automóvil de servicio público que transporta de un lugar a otro a las personas que lo solicitan a cambio de dinero; generalmente está provisto de un taxímetro que indica el importe del viaje'.* [5]. En este proyecto, taxi es el automóvil de servicio público que transporta a los estudiantes, profesores y personal administrativo desde y hacia la Universidad.
- **Motocicleta:** Medio de transporte personal de dos ruedas mediante el cual uno o dos estudiantes, profesores o administrativos se dirige a o se devuelve de la Universidad.



Universidad de los Llanos
Ingeniería de Sistemas
Car Sharing
Caso de estudio: Universidad de los Llanos

- **Automóvil:** Medio de transporte personal de cuatro ruedas mediante el cual uno o varios estudiantes, profesores o administrativos se dirige o se devuelve de la Universidad de los Llanos.
- **Recorrido:** Trayecto que recorre un automóvil para ir a o volver de la Universidad de los Llanos.
- **Cupo:** Asiento en un automóvil, que al estar disponible puede ser ocupado en un recorrido por un estudiante, profesor o administrativo.

BIBLIOGRAFÍA

- [1] U. d. I. Llanos, «Universidad de los Llanos,» [En línea]. Available: http://www.unillanos.edu.co/index.php/en/centro-de-documentacion/doc_details/661-invitation.html?tmpl=component. [Último acceso: 24 09 2015].
- [2] DefiniciónABC, «DefiniciónABC,» [En línea]. Available: <http://www.definicionabc.com/general/profesor.php>. [Último acceso: 24 09 2015].
- [3] Wikipedia, «Wikipedia,» [En línea]. Available: <https://es.m.wikipedia.org/wiki/Autobús>. [Último acceso: 24 09 2015].
- [4] E. L. Diccionario, «El Libre Diccionario,» [En línea]. Available: <http://es.thefreedictionary.com/autobuses>. [Último acceso: 24 09 2015].
- [5] W. Reference, «Word Reference,» [En línea]. Available: <http://www.wordreference.com/definicion/taxi>. [Último acceso: 24 09 2015].

8.2 Anexo B: Encuesta

[illegible]

