



Facultad 1

Módulo Flota del Sistema de Gestión para la Agencia de Renta de Vehículos REX

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor

Hector Mario García Hernández

Tutores

MSc. Walber Mengana Cuesta

Ing. Evelyn Yanez Clark

DECLARACION DE AUTORIA

Declaro por este medio que yo Hector Mario García Hernández con CI: 93100805701 soy el autor principal del trabajo titulado: “Módulo Flota del Sistema de Gestión para la Agencia de Renta de Vehículos REX” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Firma del autor

Hector Mario García Hernández

Firma del tutor

Evelyn Yanez Clark

Firma del tutor

Walber Mengana Cuesta

AGRADECIMIENTOS

DEDICATORIA

RESUMEN

El presente trabajo consistió en el desarrollo de un módulo que perfeccione el proceso de gestión de la flota de autos, utilizando tecnologías web, en el Sistema de Gestión para la Agencia de Renta de Vehículos REX. Se analizan varios sistemas de gestión para agencias de renta atendiendo a los servicios que brindan y las tecnologías utilizadas. El módulo de la flota de autos gestiona las características relacionadas con la flota de autos perteneciente a la agencia REX, además de incorporar nuevas características como: la gestión de los tipos de mantenimientos y los accesorios de los autos y la implementación de una base de datos centralizada evitará retrasos en la gestión de la información. El desarrollo de la propuesta de solución estuvo guiado por la metodología AUP en su variación para la UCI; se utilizó como herramienta para la implementación el *framework web* Django en su versión 1.10.1; lenguaje de programación Python 2.7.12, implementado en Pycharm 2016.1; como Sistema Gestor de Bases de Datos PostgreSQL 9.4, Visual Paradigm como herramienta de modelado utilizando el lenguaje UML, para la maquetación se utilizó HTML 5, CSS 3 y Bootstrap. Al *software* resultante se le aplicaron pruebas funcionales, de rendimiento, seguridad y aceptación, obteniendo resultados satisfactorios.

Palabras clave: autos, Django, flota, gestión, módulo.

Índice de contenido

| | |
|---|----|
| Introducción | 4 |
| Capítulo 1: Estudio de sistemas de gestión de información para agencias de rentas, herramientas, tecnologías y metodología para su desarrollo | 9 |
| 1.1 Gestión de información..... | 9 |
| 1.2 Sistemas de gestión para agencias de rentas..... | 9 |
| 1.3 Metodología de desarrollo, herramientas y tecnologías | 13 |
| 1.4 Conclusiones parciales del capítulo | 23 |
| Capítulo 2: Caracterización del módulo Flota del Sistema de Gestión para la Agencia de Renta de Vehículos REX | 24 |
| 2.1 Descripción de la propuesta de solución | 24 |
| 2.2 Modelo de dominio | 24 |
| 2.3 Requisitos de la propuesta de solución..... | 26 |
| 2.4 Historias de usuario..... | 28 |
| 2.5 Descripción de la arquitectura de software y los patrones de diseño..... | 31 |
| 2.6 Diagrama de clases de diseño | 35 |
| 2.7 Diagrama de componentes | 36 |
| 2.8 Modelo de datos | 37 |
| 2.9 Modelo de despliegue..... | 38 |
| 2.10 Conclusiones parciales del capítulo | 39 |
| Capítulo 3: Implementación y pruebas del módulo Flota del Sistema de gestión para la Agencia de Renta de Vehículos REX..... | 40 |
| 3.1 Estándar de codificación. | 40 |
| 3.2 Estrategia de pruebas de software | 42 |
| 3.3 Conclusiones parciales del capítulo | 50 |
| Conclusiones | 52 |
| Referencias bibliográficas | 54 |
| Bibliografía | 57 |
| Anexos..... | 59 |

Índice de tablas

| | |
|---|----|
| Tabla 1: Requisitos funcionales | 26 |
| Tabla 2: HU Gestionar auto | 28 |
| Tabla 3: HU2 Listar ficha técnica | 29 |
| Tabla 4: HU3 Gestionar tipo de mantenimiento..... | 30 |

| | |
|--|----|
| Tabla 5: Estándar de codificación | 40 |
| Tabla 6: Variables empleadas en el caso de prueba basado en la HU "Insertar tipo de mantenimiento" | 43 |
| Tabla 7: Caso de prueba correspondiente a la HU "Insertar tipo de mantenimiento" .. | 44 |
| Tabla 8: Resultado de las pruebas funcionales..... | 47 |
| Tabla 9: Pruebas de seguridad | 48 |
| Tabla 10: Informe agregado de jmeter | 50 |
| Tabla 11: Pruebas de integración | 50 |
| Tabla 12: HU Gestionar marca de auto..... | 59 |
| Tabla 13: HU Gestionar categoría..... | 59 |
| Tabla 14: HU Gestionar modelo de auto..... | 60 |
| Tabla 15: HU Gestionar transmisión | 61 |
| Tabla 16: HU Gestionar estado de motor..... | 62 |
| Tabla 17: HU Gestionar combustible | 62 |
| Tabla 18: HU Gestionar cilindrada | 63 |
| Tabla 19: HU Gestionar motor | 64 |
| Tabla 20: HU Gestionar batería | 64 |
| Tabla 21: HU Gestionar marca de neumático | 65 |
| Tabla 22: HU Gestionar posición de neumático | 65 |
| Tabla 23: HU Gestionar neumático | 66 |
| Tabla 24: HU Gestionar contrato..... | 67 |
| Tabla 25: HU Gestionar chapa..... | 68 |
| Tabla 26: HU Gestionar estado..... | 68 |
| Tabla 27: HU Gestionar causa..... | 69 |
| Tabla 28: HU Gestionar color..... | 69 |
| Tabla 29: HU Gestionar tipo de componente | 70 |
| Tabla 30: HU Gestionar estado de oferta..... | 70 |
| Tabla 31: HU Gestionar oferta | 71 |
| Tabla 32: HU Gestionar ubicación | 72 |

Índice de figuras

| | |
|--|----|
| Figura 1: Fases e iteraciones de AUP para la UCI..... | 18 |
| Figura 2: Modelo de dominio..... | 25 |
| Figura 3: Funcionamiento Django..... | 32 |
| Figura 4: Clase Entidad Auto..... | 33 |
| Figura 5: Clase Controladora InsertarAuto..... | 34 |
| Figura 6: Función insertarAuto()..... | 34 |
| Figura 7: Función insertarAuto()..... | 35 |
| Figura 8: Función para insertar un neumático..... | 35 |
| Figura 9: Diagrama de clases de diseño. HU1: Gestionar auto..... | 36 |
| Figura 10: Diagrama de componentes..... | 36 |
| Figura 11: Modelo de datos..... | 38 |
| Figura 12: Diagrama de despliegue..... | 38 |
| Figura 13: Gráfico de inconformidades..... | 47 |

Introducción

A nivel mundial se ha reconocido el papel protagónico que desempeñan las Tecnologías de la Informática y las Comunicaciones (TIC) en los aspectos de la vida cotidiana. El acelerado avance de estas tecnologías ofrece oportunidades únicas para alcanzar altos grados de desarrollo. Además, han contribuido a eliminar las barreras temporales y fronteras geográficas, garantizando la interconexión entre las personas e instituciones a nivel mundial. En los últimos años, casi todos los países del mundo han establecido e implementado proyectos, políticas y estrategias para promover el uso de las TIC y aprovechar los beneficios y los aportes que estas ofrecen, entre estos países, se encuentra Cuba.

Nuestro país, no ha estado exento de este desarrollo vertiginoso de las tecnologías de la información y las comunicaciones a pesar de las dificultades económicas que enfrenta y es por ello que, siguiendo con los Lineamientos de la Política Económica y Social del Partido y la Revolución, se forman en la actualidad, muy buenos profesionales altamente calificados para trabajar con las nuevas y diversas tecnologías que deberán afrontar en los centros laborales así como, propiciar un mejor manejo y obtención de la información, trayendo como consecuencia un aumento en la calidad de vida de la sociedad cubana. Esferas como la salud, con nuevos equipamientos tecnológicos que facilitan y agregan calidad al trabajo; la educación, con un vertiginoso aumento de los ordenadores a nivel nacional que ha beneficiado ampliamente a las nuevas generaciones desde el punto de vista educacional e incorporando nuevos paradigmas para la enseñanza. También destacan una amplia gama de *software* de producción nacional que ayudan al desempeño, desde los más pequeños hasta los altos profesionales.

El desarrollo de las TIC también se pone de manifiesto en las telecomunicaciones, donde el número de usuarios con terminales telefónicas en sus hogares y terminales móviles se ha visto incrementado en los últimos años y, en la actualidad, se cuenta con acceso a la red de redes donde las personas pueden interactuar para obtener la mejor información y también, establecer diálogos con otras personas que pueden ayudarnos en la solución de determinado problema o, simplemente desde el punto de vista humano. *“Lograr que las Tecnologías de la Información y las Comunicaciones (TIC) se conviertan en un sector de desarrollo estratégico, con una economía basada en el conocimiento, es una premisa para Cuba”*, aseguró el viceministro de Comunicaciones Wilfredo González, al intervenir durante la apertura de la Primera Conferencia Internacional Nuevos escenarios de la Comunicación Política en el Ámbito Digital 2015. La Universidad de las Ciencias Informáticas (UCI) es un ejemplo y fruto del desarrollo de las TIC contando con excelentes tecnologías, centro referenciado nacionalmente en

cuanto a la calidad de *software* siendo, además, un pilar fundamental de la estrategia del Comandante en Jefe Fidel Castro Ruz la cual es: “*graduar profesionales altamente capacitados y cada vez más comprometidos con la Revolución*”, capaces de afrontar las emergentes tecnologías que se van incorporando a Cuba. La Universidad tiene, entre sus principales objetivos, la producción de *software* y servicios informáticos a través de su modelo de formación que incluye la vinculación estudio-trabajo [1]. Dentro de la estructura de la universidad, segmentada por facultades y centros de producción, se encuentra la Facultad 1, siendo, uno de sus centros de producción el Centro de Identificación y Seguridad Digital (CISED) cuyo principal objetivo es el desarrollo de *software* de soluciones integrales en sectores como la identificación y la seguridad digital, contando entre sus proyectos, con el nuevo Sistema de Gestión para la Agencia de Renta de Vehículos REX (SIGREX).

El creciente desarrollo del turismo consolida a Cuba como destino turístico a partir de un crecimiento acelerado del número de arribos e ingresos. “*De acuerdo con el asesor del Centro de Investigaciones de la Economía Mundial, la llamada industria del ocio se destaca como el área de mejor desempeño y de mayor impacto positivo durante el año anterior*”. Ha provocado el aumento de la flota de autos para la renta incluyendo la mejora constante de la relación costo-beneficio, precios de los coches, sitios web con interfaces amigables para el cliente, facilidades de pago en diferentes monedas además de la experiencia de la agencia en el negocio. La Agencia de Renta de Autos Exclusivos REX es la principal dependencia perteneciente al Grupo Empresarial Transtur S.A destinada al arrendamiento de limosinas y autos de lujos. Actualmente esta agencia gestiona un gran cúmulo de información referente a los autos y los datos técnicos que estos poseen, así como, toda la información referente a su alquiler.

La agencia dispone de la versión actual del SIGREX implementado en ocho módulos para manipular toda la información de la renta de autos REX, este sistema acumula doce años de explotación continua y está diseñado como una aplicación escritorio con una arquitectura cliente-servidor. Como servidor de base de datos se utiliza el Sistema Gestor de Base de Datos Microsoft SQL Server 2000 y se utiliza el tipo de base de datos descentralizada. La aplicación cliente está desarrollada sobre la herramienta Visual Estudio 2003 y lenguaje de programación C#.

La agencia presenta varias sucursales a lo largo de la Isla y el SIGREX está implementado sobre una base de datos distribuida, basada en el principio de funcionamiento de replicar la información de forma tal que, todas las bases de datos de cada una de sus sucursales actualicen sus informaciones de acuerdo a los nuevos datos proporcionados por la sucursal en la que fue gestionada la nueva información. El

SIGREX fue la solución correcta a la necesidad que presentaba REX garantizando, para aquel entonces, los servicios necesarios asociados a la renta de autos pero, factores como el continuo avance de las TIC han dejado al SIGREX obsoleto y, comienza a presentar un mal funcionamiento relacionado con la gestión de los datos debido a que, la base de datos no devuelve la información con la veracidad ni la actualización requerida, ocasionado por la demora en la actualización de las bases de datos de las distintas sucursales de todo el país, la cual se efectúa cada diez minutos aproximadamente, pues puede suceder que dos sucursales renten un mismo auto, trayendo como consecuencia que los servicios no se puedan brindar con la calidad esperada. Otra desventaja que presenta el sistema es que no permite la modificación de los datos ya insertados en la base de datos, obligando al especialista a realizar las acciones directamente en la base de datos aumentando las posibilidades de ocurrencia de errores humanos. La versión actual del SIGREX presenta incompatibilidad con versiones posteriores a la versión siete del Sistema Operativo Windows, debido a que las versiones de las herramientas que utiliza para su funcionamiento se encuentran en desuso. Su condición de aplicación de escritorio constituye una desventaja adicional debido a que impide su despliegue en distintas plataformas.

El módulo Flota perteneciente a la versión actual del SIGREX posibilita la gestión de la flota de los autos. Para mantener dicha flota actualizada, se precisa previamente la actualización de clasificadores generales como son marcas, modelos y tipos de combustibles. Se gestionan los autos en explotación, de baja y, en general el estado de operación. Por otra parte, también se tiene un control de los motores de los autos, los mantenimientos, los Contratos *Leasing* (arrendamiento financiero) y las licencias otorgadas por la Empresa de Administración Vial y Diagnóstico Automotor (FICAV¹) perteneciente al Ministerio del Transporte. Este módulo actualmente presenta inconvenientes como la ausencia de un registro para llevar el control de las especificaciones de los accesorios que poseen los autos, así como, una interfaz poco amigable para los usuarios que posean bajos conocimientos relacionados con las computadoras y la inexistencia de varios tipos de mantenimientos para los vehículos provocando la pérdida sin reclamación de accesorios y un incorrecto registro de los mantenimientos de los autos.

Acciones como las mencionadas anteriormente se efectúan diariamente en la agencia, siendo estos inconvenientes un problema al cual los especialistas han tenido que encontrarles soluciones alternativas, como acceder directamente a la base de datos

¹ Logran descubrir, incluso al detalle, cualquier anomalía técnica que presente un medio de transporte con peligro para transitar por calles y carreteras cubanas.

local y actualizar los campos necesarios, implicando un aumento del tiempo y la complejidad para desarrollar la tarea.

Debido a los elementos expuestos en cuanto al desempeño del sistema SIGREX, surge el siguiente **problema científico**: ¿Cómo mejorar el proceso de gestión de la renta para la Agencia de Renta de Vehículos REX?

Con vista a la solución del problema antes mencionado se enmarca el **objeto de estudio**: el proceso de gestión de rentas de vehículos.

Para dar solución al problema descrito se establece como **objetivo general**: Desarrollar un módulo que mejore el proceso de gestión de la flota de autos, utilizando tecnologías web, en el Sistema de Gestión para la Agencia de Renta de Vehículos REX.

Delimitando el **campo de acción** a: la gestión de información de la flota de autos del Sistema de Gestión para la Agencia de Renta de Vehículos REX.

Para darle cumplimiento al objetivo planteado se responderán las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos de la investigación que sustentan el desarrollo del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX?
2. ¿Qué fundamentos presenta el análisis y diseño del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX?
3. ¿Qué características debe tener el módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX?
4. ¿Cómo validar la contribución del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX?

Para darle cumplimiento a los objetivos específicos se proponen las siguientes **tareas de investigación**:

1. Caracterización los fundamentos teóricos que sustentan la investigación relacionados con el desarrollo del módulo de gestión de la flota de autos.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar el módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX.
3. Selección de la metodología de desarrollo del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX.
4. Diseño del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX.
 1. Implementación del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX.

2. Validación de las funcionalidades del módulo de gestión de la flota de autos en el Sistema de Gestión para la Agencia de Renta de Vehículos REX.

Métodos teóricos

- **Analítico-Sintético:** para la selección de los elementos más importantes de las fuentes bibliográficas relacionados con la renta.
- **Histórico-lógico:** se analizó la existencia de aplicaciones web para gestionar la información vinculada a la renta cronológicamente.
- **Modelación:** se aplicó en la realización de diagramas, características del sistema, relación entre objetos y las actividades que intervienen en los procesos.

Métodos empíricos

- **Observación:** cuando se están realizando las pruebas de validación a la propuesta solución para observar el comportamiento del mismo bajo ciertas pruebas de estrés.
- **Análisis documental:** para extraer información necesaria de bibliografía existentes y consultar información en sitios de interés nacional e internacional para apoyar en las tareas definidas en la investigación.

El presente documento se encuentra conformado por tres capítulos como se expresa a continuación:

Capítulo 1: Estudio de sistemas de gestión de información para agencias de rentas, herramientas, tecnologías y metodología para su desarrollo.

En este capítulo se definen las herramientas y tecnologías, así como metodología de desarrollo de *software* que se utilizará, constituyendo la base teórica de la investigación.

Capítulo 2: Caracterización del módulo Flota del Sistema de Gestión para la Agencia de Renta de Vehículos REX.

Este capítulo abarca el desarrollo del flujo actual de los procesos, y se describe la propuesta de solución para resolver el problema planteado. Además de, especificar los requisitos funcionales, no funcionales y, los elementos fundamentales del diseño y de la arquitectura.

Capítulo 3: Implementación y pruebas del módulo Flota del Sistema de Gestión para la Agencia de Renta de Vehículos REX.

Centrado en la programación a través de los requisitos, además de, las pruebas desarrolladas para su validación, verificando los requisitos de calidad de la solución.

Capítulo 1: Estudio de sistemas de gestión de información para agencias de rentas, herramientas, tecnologías y metodología para su desarrollo

En este capítulo se abarca una serie de conceptos como: breve descripción, características y ventajas asociadas al uso de las tecnologías y herramientas (lenguaje de programación, Entorno de Desarrollo Integrado (IDE) a emplear y gestor de bases de datos) ampliamente difundidas a escala global para el desarrollo de aplicaciones *web*. Además de definir la metodología que guiará el proceso de desarrollo de *software*.

1.1 Gestión de información

La gestión de información es coordinar, dirigir y controlar los flujos de información que tiene lugar en cada sistema, de manera sistemática. Tomando en cuenta segmentos típicos de los sistemas, como pueden ser el ambiente, los procesos que se generan, las personas y las tecnologías implicadas entre otros; que de conjunto interrelacionan. Así como del conocimiento actualizado y preciso del ciclo de vida de los documentos que genera el sistema, producto de la Gestión Documental (GD). Según Burk y Horton, los principales objetivos de la Gestión de Información (GI) son los siguientes:

1. Maximizar el valor y los beneficios derivados del uso de la información.
2. Minimizar el costo de adquisición, procesamiento y uso de la información.
3. Determinar responsabilidades para el uso efectivo, eficiente y económico de la información.
4. Asegurar un suministro continuo de la información [2].

1.2 Sistemas de gestión para agencias de rentas

A nivel internacional la renta de autos ha alcanzado un desarrollo vertiginoso, enfrascado en la mejora constante de la calidad de los servicios brindados para satisfacer una demanda cada vez mayor de clientes. Se trabaja constantemente en agregar valor al negocio debido a la gran competencia en el sector.

Hertz

Hertz Corporation opera en el negocio de alquiler de vehículos a través de sus marcas *Hertz*, *Dollar*, *Thrifty* y *Firefly* en aproximadamente 10.400 oficinas corporativas y franquicias Norte de América, Europa, América Latina, Asia, Australia, África, Oriente Medio y Nueva Zelanda. Fundada en el año 1926. Hertz es la mayor compañía mundial de alquiler de vehículos que opera desde aproximadamente 8.800 oficinas corporativas y franquicias en aproximadamente 150 países. Es además, la marca de alquiler de coches número uno en aeropuertos de Estados Unidos y de 111 aeropuertos principales de Europa. Por su parte, *Dollar* y *Thrifty* cuentan con aproximadamente 1.580 oficinas corporativas y franquicias en alrededor de 80 países.

Lleva 95 años ofreciendo soluciones de calidad de alquiler de vehículos a clientes corporativos y de ocio.

La aplicación de la agencia ofrece productos y servicios innovadores como *Hertz Gold Plus Rewards*, el sistema de satélite de navegación especialmente diseñado *NeverLost®*, y coches únicos en las categorías *Prestige & Fun*, sitúan a *Hertz* muy lejos de sus competidores.

Algunos de los servicios que ofrecen son:

- Abastecimiento de combustible para autos *Hertz*.
- Rápida recogida y devolución más rápido: El tiempo de recogida / devolución electrónicamente por lo que es más rápido y menos trámites.
- Limpio y seguro: los clientes pueden verificar si el vehículo se encuentra limpio y seguro antes de rentarlo guiándose por una lista que contiene treinta y cinco puntos para la inspección del auto.
- Venta de autos *Hertz*: se ofrecen excelentes alternativas de autos *Hertz* usados en comparación con las tradicionales tiendas de autos usados [3].

Además, fue desarrollada por *Hertz* en el año 1997, actualmente presenta una interfaz amigable, ofrece una amplia gama de autos que van desde furgonetas hasta autos de lujo y servicios para toda clase de clientes pero, brinda otros servicios como la venta de autos usados, recogida y devolución electrónicamente y el uso de radio por satélite *SIRIUS*; servicios como los mencionados anteriormente Cuba no puede permitirse debido al bloqueo económico impuesto por los Estados Unidos (EE.UU) hacia la Isla que obstaculizan la adquisición de algunas tecnologías así como, sus costos de utilización.

Fue desarrollada con la herramienta *DoubleClick for Publisher* y *Google AdSense* para la red de publicidad, *Modernizr*, *Backbone.js*, *RequireJS*, *jQuery* y *Underscore.js* como *frameworks* JavaScript y, *Google Tag Manager* como administrador de etiquetas.

Interhome

Interhome fue fundada en 1965 en Suiza siendo una compañía que ha adquirido una gran experiencia, a lo largo de varias décadas, en el alquiler de alojamientos vacacionales. Hoy en día, *Interhome* es el mayor proveedor europeo de casas y apartamentos turísticos en todo el mundo. Disponen de quince sucursales. *Interhome* se encuentra en *Glattbrugg*, Suiza, y es propiedad de *Hotelplan Holding AG*. Esta empresa es, a su vez, propiedad de *Migros*, la compañía de comercio minorista más grande de Suiza, cuyas oficinas centrales tienen sede en *Zurich*. Su calidad suiza proporciona y asegura a los clientes una fantástica experiencia en sus vacaciones. Además, *Interhome* es uno de los sitios web más populares en lo que a alquiler de

alojamiento de vacaciones se refiere. Ofrece más de 33.000 casas y apartamentos de vacaciones y una gestión profesional de alto nivel. Interhome prácticamente funciona como una agencia. Puedes ver los anuncios, realizar la reserva y el pago online. La calidad de cada vivienda está certificada por una declaración expresa a través de un sistema de puntuación mediante estrellas. Ofrece un servicio de información al cliente muy completo. Disponible las 24 horas del día de lunes a domingo. De esta manera podrás contactar con alguien del sitio antes, durante y después de tu estancia.

Ofrece servicios como:

- Reservación de una gran variedad de casas que van desde alojamientos en las montañas hasta castillos en disímiles ubicaciones como montañas, playas y grandes ciudades
- Servicio de información al cliente muy completo. Disponible las 24 horas del día de lunes a domingo. De esta manera podrás contactar con alguien del sitio antes, durante y después de tu estancia. [4].

La nueva aplicación web fue creada en el año 2010 y a partir de entonces en el año 2013 ganó el primer premio de la categoría usabilidad, pero, tecnologías privativas usadas para su desarrollo como: ASP.NET como *framework web*, constituyen una desventaja para la economía nacional.

Airbnb

Fundado en 2008 y con sede en San Francisco (California) surge de la original idea de tres jóvenes y, en la actualidad, es un mercado comunitario basado en la confianza en el que las personas publican, descubren y reservan alojamientos únicos en todo el mundo, ya sea desde su ordenador, tableta o teléfono móvil. La aplicación web de Airbnb ofrece servicios que van desde alquilar un departamento para pasar una noche como un castillo durante una semana o una villa durante un mes.

Otros elementos a tener en cuenta son:

- Buenas fotografías del alojamiento turístico. Es el mayor reclamo para los huéspedes potenciales.
- Actualización de los precios y las tarifas (enlace al servicio). Cambian continuamente en base a la época (temporada alta, media, baja... conciertos, eventos importantes...)
- Buena atención al huésped, que repercute en buenas valoraciones en Airbnb [5].

La aplicación web Airbnb fue lanzada el mismo año que surge la compañía, muestra una interfaz sencilla, con la esencia del negocio, sin embargo, fue desarrollada sobre

tecnologías como: Ruby on Rails como *framework web* que, a pesar de ser de código abierto el poco tiempo de explotación hace que sea un lenguaje poco conocido.

Havanautos

Fundada en 1982 como la primera empresa de alquiler de coches de Cuba. Havanautos le ofrece acceso a la mayor flota de autos y más de 70 oficinas de alquiler en toda la Isla. Con una flota moderna manejada y distribuida por un montón de agencias y más de 70 oficinas de alquiler en toda la Isla, Havanautos es, sin duda, la empresa de alquiler de coches con la mejor estructura y posibilidades para ofrecer un servicio de emergencia en carretera las veinticuatro horas.

La aplicación de la agencia Havanautos ofrece servicios al cliente como:

- Renta al mostrador.
- Servicio de renta de autos vendido directamente al buró.
- Renta lineal.
- Renta de un auto por un período mínimo de tres meses.
- Renta con chofer.
- Renta del auto con chofer profesional.
- Renta de autos para safaris.
- Renta de autos 4x4 para su uso en zonas rurales [6].

La aplicación web presenta desventajas como:

- Uso de herramientas privativas utilizadas para la implementación de la aplicación web como son: *Microsoft ASP.NET* como *framework web* y como sistema operativo Windows Server limitan su uso en la Isla.

Grupo gaviota

Es una empresa transportista del Grupo de Turismo Gaviota S.A. concebida para satisfacer las necesidades de transportación y renta de vehículos al turismo en el año 1988. Cuentan con más de cien Unidades de Venta, distribuidas convenientemente por todo el país, según los requerimientos del mercado y los polos turísticos además de la garantía en los servicios de auxilio en la vía durante las 24 horas y una experiencia avalada de veintinueve años.

En el sistema de gestión Gaviota se brindan servicios como:

- Arrendamiento de autos en los principales aeropuertos y polos turísticos del país, con modelos y marcas de vehículos competitivos que brindan un alto confort y seguridad.
- Servicio de alquiler de ómnibus con chofer, con una flota de vehículos de diferentes capacidades, todos con un alto confort y equipados con audio, aire

acondicionado, nevera, asientos reclinables, maleteros y un equipo de profesionales que harán agradable y segura la estancia turística en el país [7]. La aplicación fue implementada con las tecnologías: Drupal como gestor de contenido, Modernizr y JQuery como *frameworks* JavaScript y, php como lenguaje de programación.

Valoración crítica de los sistemas estudiados

Luego del análisis de los diferentes sistemas de gestión de información para agencias de renta, el autor, considera que responden a las necesidades del negocio pero, poseen desventajas como: el poco conocimiento acerca de la tecnología *Ruby on Rails* con la que fue implementado el sistema de gestión *Airbnb*, el carácter privativo de *Microsoft ASP.NET* con la que fue desarrollado el sistema de gestión de la agencia *Interhome*, además de los altos costos que suponen el uso de estos sistemas internacionales por el pago de licencias y servicios de soporte a la aplicación, en el sistema de gestión del grupo Gaviota, la alta complejidad que presenta Drupal para el aprendizaje y el sistema de la agencia Havanautos también fue creado con la tecnología *ASP.Net* de carácter privativo. A modo de conclusión los sistemas analizados, aunque no constituyeron una solución a la situación problemática, sirvieron de apoyo al proceso de obtención de requisitos de la propuesta de solución.

1.3 Metodología de desarrollo, herramientas y tecnologías

Durante el desarrollo de módulo de gestión de la flota de autos se aplican diversas tecnologías como lenguajes de programación, el uso de herramientas para el modelado, gestores de bases de datos, así como, la metodología seleccionada para definir el proceso de desarrollo del *software*.

Metodologías de desarrollo de software

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos *software* [8].

El desarrollo de *software* no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Existen aquellas propuestas ágiles y se encuentran las más tradicionales, centradas especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán.

Metodologías ágiles

En febrero de 2001, tras una reunión celebrada en Utah, EE.UU, nace el término “ágil” aplicado al desarrollo de *software*. En esta reunión participan un grupo de diecisiete expertos de la industria del *software*, incluyendo algunos de los creadores o impulsores

de metodologías de *software*. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar *software* rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de *software* tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas [9]. Como consecuencia surgen las metodologías ágiles, enfocadas al dinamismo y la variabilidad de la industria del *software*. Se aplican para el desarrollo rápido de aplicaciones, sus objetivos son aumentar la productividad a corto plazo, así como, satisfacer las necesidades cambiantes de los clientes en el menor tiempo posible para proporcionar un mayor valor al negocio.

SXP

Es una metodología ágil para el desarrollo de *software* y está compuesta por SCRUM² y XP, la misma se recomienda para trabajar en proyectos pequeños, así como con equipos de tamaño similar. Esta metodología cuenta con roles específicos que se encargan de realizar las actividades que le corresponden a cada uno de ellos, generando los artefactos que se esperan al cumplir con cada una de las tareas desarrolladas. SCRUM-*Xtreme Programming* (SXP) al igual que otras metodologías ágiles es flexible ante los cambios y posee una rápida entrega de los resultados. Cuenta con cuatro fases: la de planificación-definición, desarrollo y mantenimiento.

SXP es una metodología compuesta por las metodologías SCRUM y *Xtreme Programming* (XP) que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de *software* para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad [10].

Ventajas:

- La metodología establece el uso de sistemas automatizados para la generación de algunos artefactos. Y además los recomienda de manera explícita.
- La descripción de una historia de usuario suele ser más sencilla que la de un caso de uso.

² La palabra se usa en rugby y prácticamente significa melé.

- SXP ha sido excelente para equipos pequeños, siempre ha sido recomendado para menos de 20 personas [10].

Metodología de desarrollo eXtreme Programming (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en una realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Esta metodología se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. El desarrollo es la parte más importante en el proceso de la programación extrema. Todos los proyectos tienen como objetivo que se programen lo más rápido posible, sin interrupciones y en dirección correcta. También es muy importante el diseño, y se establecen los mecanismos para que éste sea revisado y mejorado de manera continua a lo largo del desarrollo del proyecto según se vayan añadiendo funcionalidades al mismo. Esta metodología de desarrollo ligero se basa en una serie de valores y buenas prácticas que persiguen el objetivo de aumentar la productividad a la hora de desarrollar un *software*.

Ventajas:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

Desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar [10].

AUP

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP por sus siglas en inglés) es una versión simplificada del Proceso Unificado de *Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas.
- Modelado ágil.
- Gestión de Cambios ágil.
- Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

Fases AUP:

- **Inicio:** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- **Transición:** El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción [11].

Variación de AUP para la UCI.

Al no existir una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

“Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello esta metodología adaptada a la universidad se apoya en el Modelo CMMI-DEV³ v1.3 el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora” [11]. Estas prácticas se centran en el desarrollo de productos y servicios de calidad.

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, nombrada Ejecución y se agrega la fase de Cierre [11].

Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a uno la cantidad

³ Es el modelo de referencia para la mejora de las diferentes áreas de proceso en los proyectos de desarrollo y de mantenimiento de software. Conjunto de modelos de buenas prácticas, para la mejora de los procesos.

de metodologías que se usaban y de más de veinte roles en total que se definían, se redujeron a once [11].

Esta metodología define cuatro escenarios para el trabajo con requisitos teniendo en cuenta las características del equipo de proyecto y del negocio. Estos escenarios se muestran a continuación:

Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los Casos de uso del negocio (CUN) muestran como los procesos son llevados a cabo por personas y los activos de la organización.

Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una historia de usuario (HU) no debe poseer demasiada información. Las disciplinas definidas en la variación de AUP para la UCI son: modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación y pruebas de aceptación. Estas disciplinas (desde Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales.

En una iteración se repite el flujo de trabajo de las disciplinas, Requisitos, Análisis y diseño, Implementación y Pruebas internas. De esta forma se brinda un resultado más

completo para un producto final de manera creciente. Para llegar a lograr esto, cada requisito debe tener un completo desarrollo en una única iteración [11].

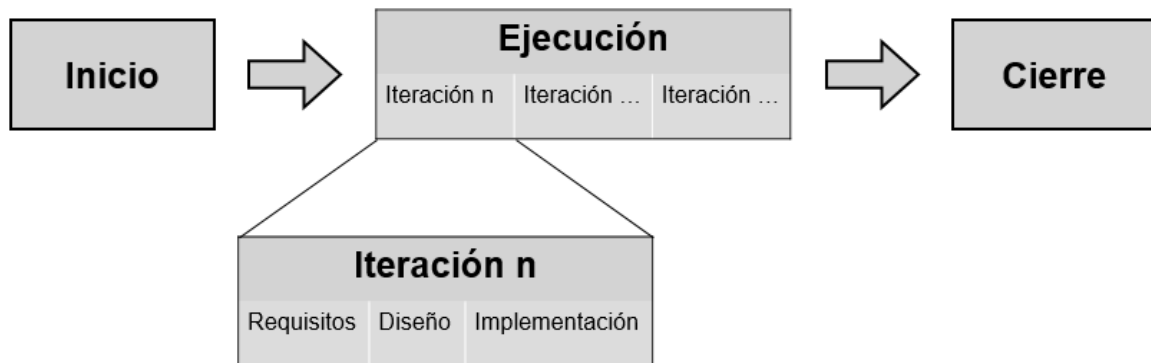


Figura 1: Fases e iteraciones de AUP para la UCI.

Metodología escogida

Para la realización de la documentación del proyecto se definió la metodología AUP-UCI debido al enfoque ágil que posee, dirigida a aumentar la eficiencia de las personas involucradas en los proyectos productivos de la universidad. AUP-UCI garantiza que el trabajo sea exitoso, satisface las necesidades cambiantes del cliente, así como, la relación estrecha que establece con el mismo por lo que se adapta fácilmente a las circunstancias y es ideal para equipos de trabajo pequeños. Además, se apoya en las buenas prácticas que propone CMMI-DEV v1.3.

Herramientas y tecnologías

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. A continuación, se describen las herramientas y tecnologías utilizadas para el desarrollo de la propuesta de solución.

Framework

Es una estructura *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta [12].

Framework Django

La evolución de Python es cada vez mayor, debido a que es un lenguaje poderoso y fácil de aprender. En los últimos años han surgido herramientas que hacen el trabajo más simple y eficiente con este lenguaje de programación. Una de esas herramientas es Django, el *framework* hecho en Python para perfeccionistas.

Django es un *framework* de alto nivel en Python Web que fomenta un desarrollo rápido y un diseño limpio y pragmático. Construido por desarrolladores experimentados, se

encarga de gran parte de la molestia de desarrollo web, por lo que puede centrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es de código abierto y gratuito [14].

Ventajas

- Aparte de las ventajas que tiene por ser *framework*, Django promueve el desarrollo rápido, se construyen aplicaciones en cuestión de días y con el conocimiento suficiente esos días se pueden reducir a horas.
- Django impulsa el desarrollo de código limpio al promover buenas prácticas de desarrollo *web*, sigue el principio DRY⁴ (conocido también como una vez y sólo una) [12].
- Usa una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (*Model - Template - View*), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático [13].

Lenguaje de Marcado de Hipertexto (HTML)

HTML5 es un lenguaje de marcas de hipertexto (*Hyper Text Markup Language*) usado para estructurar y presentar el contenido para la web. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. Es un nuevo lenguaje de marcado de hipertexto para presentar y estructurar el contenido en internet. Es la quinta y nueva versión del estándar HTML. Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance, presenta nuevas características que proporcionan un rico soporte multimedia (video y audio), también mejoran el apoyo para la creación de aplicaciones web [15].

Características de HTML5:

- Semántica: Permite describir con mayor precisión cuál es su contenido.
- Conectividad: Permite comunicarse con el servidor de formas nuevas e innovadoras.
- Sin conexión y almacenamiento: Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.
- Multimedia: Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.

⁴ Establece que, en un entorno informático, la información no debe repetirse. Es decir, el conocimiento almacenado en un programa informático debe mantenerse en un, y sólo en un, lugar

- Gráficos y efectos 2D/3D: Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo [15].

Hojas de Estilo en Cascada (CSS)

Las hojas de estilo en cascada (CSS3) proponen una navegación más rápida, menores tiempos de respuesta producidos por una reducción de imágenes, las cuales ya no serán requeridas para diseñar botones o efectos de texto. También deja atrás una excesiva dependencia de *Java Script* para fines de representación visual, como lo son las animaciones, dando como resultado menos código y mejor rendimiento. Representa una futura menor dependencia de *software* para gráficos que resultan bastante caros, como son *Photoshop*, *Illustrator* o *Corel*, que si bien seguirán siendo útiles su función se enfocará a efectos o ediciones más avanzadas. Con CSS3 se logra tener un apoyo, que nos permite llegar a una buena usabilidad y accesibilidad, nos permite mejorar el rendimiento y acelerar el desarrollo de los sitios web. Las características de CSS 3 incluyen bordes redondeados, a través del atributo *border-radius*, que define la curvatura que debe tener el borde del elemento dándole un mejor acabado, se emplea para romper las palabras que son demasiado largas y no caben enteras por la anchura de una caja, la propiedad *box-shadow* que permite que un elemento tenga un resplandor exterior. Las novedades de CSS3 nos permiten ahorrarnos tiempo y trabajo al poder seguir varias técnicas (bordes redondeados, sombra en el texto, sombra en las cajas, etc.) sin necesidad de usar editores gráficos [16].

Bootstrap

Bootstrap es un marco de trabajo (*Framework*), creado por *Twitter*, permite crear interfaces web con CSS y *JavaScript*. Emplea una técnica de diseño conocida como “*responsive design*” o diseño adaptativo. Esta permite adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Este marco de trabajo trae varios elementos con estilos predefinidos fácilmente configurables: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración con *jQuery* para ofrecer ventanas y *tooltips* dinámicos [17].

Sistemas Gestores de Bases de Datos

Un sistema gestor de bases de datos (SGBD) es el *software* que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos, además, en estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.

El éxito del SGBD reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios [18].

PostgreSQL 9.4

PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) objeto-relacional, distribuido bajo licencia BSD⁵ y con su código fuente disponible libremente. Es el SGBD de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [19].

Esta versión agrega nuevas características que mejoran la flexibilidad, escalabilidad y rendimiento de PostgreSQL para diferentes tipos de usuarios de bases de datos, incluyendo mejoras al soporte para JSON⁶, replicación y rendimiento de los índices [20].

Herramienta CASE

Se puede definir a las herramientas de Ingeniería de Software Asistida por Ordenador (CASE por sus siglas en inglés) como cualquier herramienta que se utiliza para automatizar alguna actividad [21].

Visual Paradigm

Visual Paradigm presenta todos los diagramas de tipo Lenguaje Unificado de Modelado (UML) y herramientas de tipo Diagrama Entidad Relación (ERD) esencialmente en el diseño de sistemas y bases de datos. Herramientas de modelado innovadoras como *Resource Catalog*, *Transitor* y *Nicknamer* hacen el modelado de sistemas fácil y rentable. Doc. Composer le permite producir especificaciones de diseño detalladas listas para usar en la discusión con sólo unos pocos clics [22].

⁵ Es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*)

⁶ *JavaScript Object Notation*, es un formato de texto ligero para el intercambio de datos.

IDE

Un Entorno de Desarrollo Integrado (IDE) es un *software* compuesto por un conjunto de herramientas de programación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica [23].

Pycharm 2016.1

PyCharm es el IDE más popular para programación en Python [24]. PyCharm viene en tres versiones: edición profesional, de la comunidad y educativa. Tiene funciones avanzadas como el código automático, terminación, resaltado de código, refactorización, capacidades de desarrollo remoto y soporte para varios *frameworks web*. Está disponible para varias plataformas como Windows, Linux y OS X [25].

Esta es la primera actualización de la serie de lanzamientos prevista para 2016. Sus novedosas funciones para Python profesional, *web* y desarrollo científico trabajan juntas sin problemas para ofrecerle una experiencia de codificación única [26].

Lenguaje de programación

Diseñado para expresar instrucciones que puedan entender los ordenadores. Pueden controlar el desempeño lógico y físico de un sistema informático mediante la creación de programas que contienen algoritmos.

Python

El *software* libre se ha convertido en uno de los movimientos tecnológicos de mayor auge en el siglo XXI. Para su desarrollo ha sido necesario contar con un grupo de herramientas que hagan óptima su utilización y sean fáciles de aprender. Python es un lenguaje de programación que cumple con lo planteado y se viene perfilando como una opción recomendada para el desarrollo de *software* libre [27].

Es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible [27].

Cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas. Fue basado en el lenguaje ABC y se dice que fue influenciado por otros como C, Algol 60, Modula-3 e Icon según su propio autor [27].

Una de las fortalezas de Python, y quizás la mayor, es la librería estándar con que cuenta. Con decenas de módulos cubre la mayoría de las necesidades básicas de un programador y mucho más. En esta se le da cobertura de forma muy intuitiva a tópicos como:

- Cadenas.
- Estructura de datos.

- Funciones numéricas y matemáticas.
- Compresión de datos.
- Formatos de archivo.
- Criptografía.
- Servicios de los Sistemas Operativos.
- Comunicación entre Procesos.
- Manejo de datos de Internet.
- Servicios multimedia.
- Manejo de excepciones [27].

Lenguaje Unificado de Modelado (UML)

El Lenguaje de Modelado Unificado (UML) es un lenguaje estándar para escribir diseños de *software*. El UML puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de *software* intensivo [28].

Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de *software*. Además, entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de *software* reutilizable [29].

1.4 Conclusiones parciales del capítulo

A partir de los elementos presentados se arriba a las siguientes conclusiones:

- El estudio de las diferentes herramientas, tecnologías y tendencias actuales para la renta permitió evidenciar las características que constituyen las premisas para el planteamiento de las funcionalidades que se definirán en la propuesta de solución.
- Se efectuó el análisis de los sistemas de renta, las metodologías, herramientas y tecnologías que apoyaron la elección de AUP-UCI como metodología de desarrollo, UML como lenguaje de modelado, Visual Paradigm como herramienta CASE, Python como lenguaje de programación, Django como framework web, IDE Pycharm, PostgreSQL como Sistema Gestor de Bases de Datos y para la maquetación web HTML, CSS y Bootstrap.

Capítulo 2: Caracterización del módulo Flota del Sistema de Gestión para la Agencia de Renta de Vehículos REX

En el presente capítulo se realiza una descripción de las características de la propuesta de solución. Se definen los requisitos funcionales (RF) y requisitos no funcionales (RNF), las Historias de Usuarios (HU) definidas por el equipo de desarrollo, la arquitectura, patrones de diseño y los artefactos generados que guiarán la implementación de la propuesta de solución.

2.1 Descripción de la propuesta de solución

Para darle solución a la problemática planeada se implementará un módulo Flota que mejorará los procesos de registro, mantenimiento y seguimiento de la flota de autos. Incorporará la gestión de nuevas características como: las especificaciones de los accesorios y los tipos de mantenimiento que requieren los autos. El diseño de una base de datos centralizada permitirá gestionar información actualizada desde cualquier sucursal, una interfaz amigable que permitirá la interacción del especialista con el sistema y, su condición de aplicación *web responsive* permitirá que se pueda ejecutar y adaptar a cualquier plataforma.

2.2 Modelo de dominio

El modelo del dominio muestra (a los modeladores) clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objeto. Es una representación de las clases conceptuales del mundo real, no de componentes software. Es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales (término utilizado en la primera edición del libro de Larman), modelo de objetos del dominio y modelos de objetos de análisis. Utilizando la notación UML, un modelo del dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación.

Pueden mostrar:

- Objetos del dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales [30].

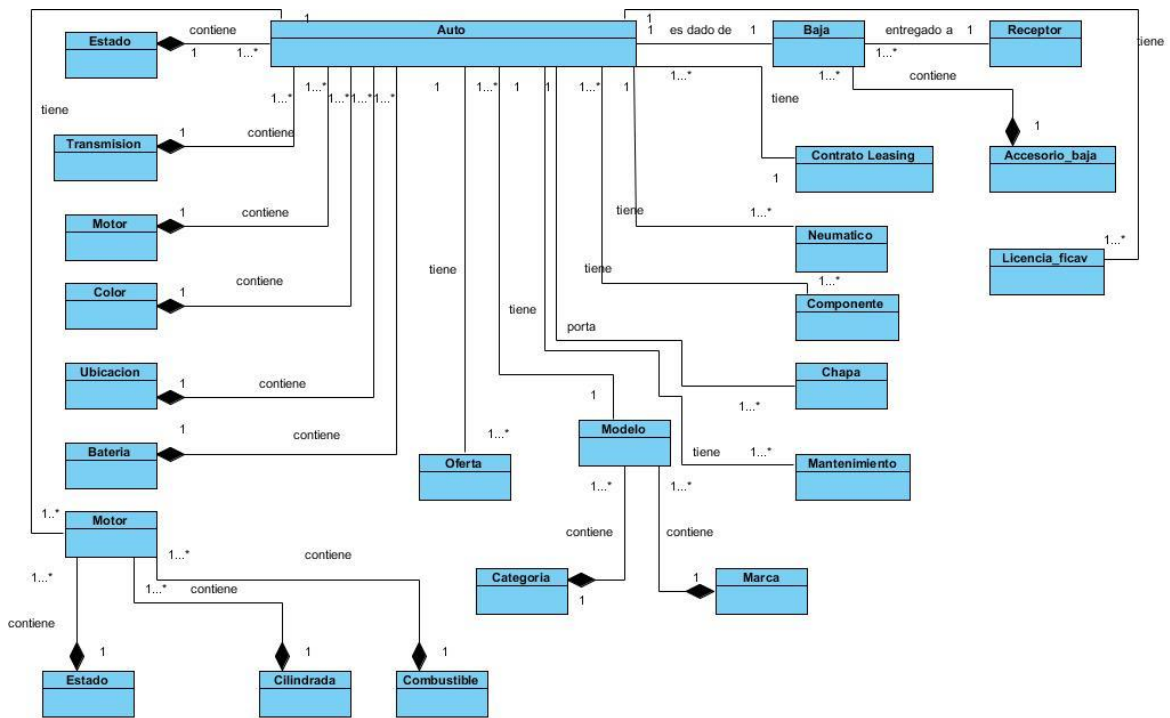


Figura 2: Modelo de dominio

Descripción de los elementos del modelo de dominio

- **Estado:** Estados en los que se encuentra un auto.
- **Transmisión:** Tipo de transmisión de un auto.
- **Motor:** Descripción de las características de un motor.
- **Estado_motor:** Define el estado en que se encuentra un motor.
- **Cilindrada:** Número de cilindrada de un motor.
- **Color:** Define el color de un auto.
- **Ubicación:** Nombre de la sucursal donde se encuentra el auto.
- **Batería:** Describe las características de una batería.
- **Oferta:** Define la oferta de un auto.
- **Auto:** Autos que se le aplicará una o varias operaciones.
- **Baja:** Autos que son dados de baja de la flota de REX.
- **Receptor:** Entidades que recepcionarán los autos después que estos sean dados de baja.
- **Chapa:** Define características de una chapa.
- **Mantenimiento:** Define características del mantenimiento de un auto.
- **Combustible:** Tipos de combustibles utilizados por los autos.
- **Modelo:** Modelos de los autos que operan en REX.
- **Categoría:** Define la categoría de un auto.

- **Neumático:** Características de un neumático.
- **Componente:** Define características de un componente.
- **Marca:** Marcas de los autos que operan en REX.
- **Contrato Leasing:** Contratos leasing para los autos que operan en REX.
- **Accesorio:** Accesorios para los autos de la flota de REX.

2.3 Requisitos de la propuesta de solución

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información [31].

Los requisitos son capacidades y condiciones con las cuales debe ser conforme el sistema -y más ampliamente, el proyecto. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (que normalmente significa registrar) lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo [32].

Requisitos funcionales

Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de *software* que se esté desarrollando, de los usuarios esperados del *software* y del enfoque general que adopta la organización cuando se escriben los requerimientos [31].

Tabla 1: Requisitos funcionales. Fuente: Elaboración propia.

| | |
|---------------------------------------|---------------------------------------|
| RF1: Gestionar marca de auto | RF17: Gestionar color |
| RF2: Gestionar categoría | RF18: Gestionar tipo de componente |
| RF3: Gestionar modelo de auto | RF19: Gestionar estado de oferta |
| RF4: Gestionar transmisión | RF20: Gestionar oferta |
| RF5: Gestionar estado de motor | RF21: Gestionar ubicación |
| RF6: Gestionar combustible | RF22: Gestionar auto |
| RF7: Gestionar cilindrada | RF23: Gestionar receptor |
| RF8: Gestionar motor | RF24: Gestionar tipo de baja |
| RF9: Gestionar batería | RF25: Gestionar baja |
| RF10: Gestionar marca de neumático | RF26: Gestionar accesorio |
| RF11: Gestionar posición de neumático | RF27: Gestionar accesorio de baja |
| RF12: Gestionar neumático | RF28: Gestionar licencia FICAV |
| RF13: Gestionar contrato | RF29: Gestionar tipo de mantenimiento |

| | |
|-------------------------------|--------------------------------------|
| RF14: Gestionar chapa | RF30: Gestionar mantenimiento |
| RF15: Gestionar estado | RF31: Gestionar DTI |
| RF16: Gestionar causa | RF32: Gestionar trasiego |
| | RF33: Listar paralización |
| | RF34: Listar ficha técnica |

Requisitos no funcionales

Los requerimientos no funcionales (RNF), como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O o las representaciones de datos usados en las interfaces con otros sistemas [31].

Usabilidad

RNF1: El módulo debe tener una interfaz amigable para los usuarios con conocimientos básicos en el manejo de una computadora y de un ambiente web.

RNF2: El módulo debe mostrar los mensajes, títulos y demás textos que aparezcan en la interfaz en idioma español.

Rendimiento

RNF3: El módulo debe dar respuesta en un tiempo aceptable de diez segundos.

Operaciones

RNF4: Para la distribución del producto y su instalación se necesitará un entorno de trabajo compuesto por:

- Base de datos: Postgres 9.4
- Framework web: Django versión 1.10.1
- Lenguaje de programación: Python versión 2.7.12

Mantenimiento y soporte

RNF5: La escalabilidad debe ser una característica del módulo, permitiendo incorporarle nuevas características en caso de ser necesarias.

Apariencia o interfaz externa

RNF6: El módulo debe ser compatible con los navegadores *web* Chrome, Firefox, Safari, Opera e Internet Explorer.

Seguridad

RNF7: El sistema será utilizado solo por los usuarios registrados y usuario administrador.

RNF8: Protección de la integridad de los datos ante operaciones no autorizadas.

RNF9: En caso de errores, mostrar la menor cantidad de detalles protegiendo la integridad del sistema.

RNF10: El sistema cerrará la sesión del usuario autenticado en caso de cerrar el navegador.

Hardware

RNF10: Los requisitos mínimos de los servidores de aplicación y base de datos deben ser: CPU Core i3 Cuarta generación a 2.0 GHz con 16 Gb de RAM.







RNF11: Para el servidor de base de datos una capacidad de almacenamiento mínima de 5 Gb y 10 Gb para el servidor de aplicaciones *web*.

2.4 Historias de usuario

Las historias de usuario se utilizan para definir los requerimientos de un sistema de software y también para crear estimaciones para la planeación de las interacciones. Estas son escritas por los clientes determinando las cosas que quieren que el sistema haga por ellos. Deben describir comportamientos externos del sistema, los cuales puedan ser entendidos por los involucrados [33].

Tabla 2: HU Gestionar auto. Fuente: Elaboración propia.

| HU Gestionar auto | |
|--|--------------------------------------|
| Número: 0 | Nombre del requisito: Gestionar auto |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un auto se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro. <ul style="list-style-type: none">• Modelo (Obligatorio. Campo selector. Permite la selección de un modelo).• Transmisión (Obligatorio. Campo selector. Permite la selección de una transmisión).• Batería (Obligatorio. Campo selector. Permite la selección de una batería).• Estado (Obligatorio. Campo selector. Permite la selección de un estado).• Color (Obligatorio. Campo selector. Permite la selección de un color).• Ubicación (Obligatorio. Campo selector. Permite la selección de una ubicación).• Accesorio (Obligatorio. Campo selector. Permite la selección de un accesorio).• Contrato Leasing (Obligatorio. Campo selector. Permite la selección de un contrato). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de autos. | |
| Prototipo de interfaz: | |

| | |
|-------|---|
| auto1 |   |
| auto2 |   |
| auto3 |   |

Insertar auto

▼

▼

▼

▼

▼

▼

▼

Tabla 3: HU2 Listar ficha técnica. Fuente: Elaboración propia.

| | |
|--|--|
| Requisito | |
| Número: 1 | Nombre del requisito: Listar ficha técnica |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| Descripción: En la interfaz se mostrará un listado con los valores asociados a los autos almacenados en la base de datos y adjunto a cada uno de ellos se mostrará la opción de visualizar la ficha técnica. | |
| Observaciones: La interfaz presenta un botón llamada "Ficha técnica" que le permitirá al usuario observar la ficha técnica del auto a partir de esta vista. | |

Prototipo de interfaz:

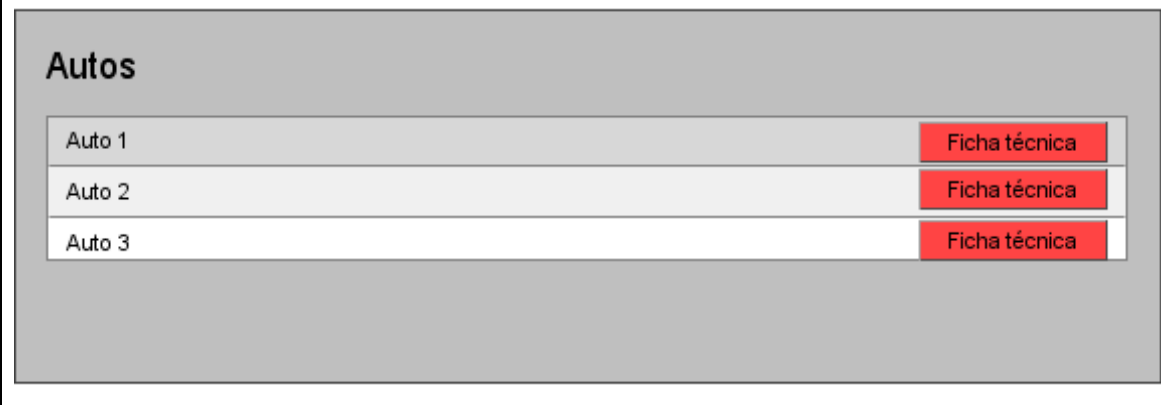








Tabla 4: HU3 Gestionar tipo de mantenimiento. Fuente: Elaboración propia.

| | |
|---|---|
| Número: 1 | |
| Número: 1 | Nombre del requisito: Gestionar tipo de mantenimiento |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un tipo de mantenimiento se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). • Variación de ejecución (Obligatorio. Campo de tipo texto. La entrada de caracteres alfabéticos y espacios). • Fecha de inicio (Obligatorio. Campo de tipo fecha. Permite la selección de una fecha en un calendario). • Fecha de fin (Obligatorio. Campo de tipo fecha. Permite la selección de una fecha en un calendario). • Descripción (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfanuméricos y espacios). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de tipos de mantenimientos. | |
| Prototipo de interfaz: | |
| tipo_mantenimiento 1 |   |
| tipo_mantenimiento 2 |   |
| tipo_mantenimiento 3 |   |

Insertar tipo de mantenimiento

Fecha de inicio
▼

| febrero | de | 2017 |
|---------|----|------|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Fecha de fin
▼

| febrero | de | 2017 |
|---------|----|------|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

2.5 Descripción de la arquitectura de software y los patrones de diseño

Arquitectura

Django se suele llamar un *framework* Modelo Vista Plantilla (MVT por sus siglas en inglés) debido a que está fuertemente influenciado por Modelo Vista Controlador (MVC) y es incluso posible argumentar que la terminología MVC es el único patrón de cambios en Django. Las tres capas básicas son el modelo, la vista, y la plantilla [14].

Se podría clasificar a Django como parte de la tercera generación del desarrollo *web* [14].

Sin embargo, más allá de las clasificaciones que podrían existir, está el entender cómo funciona realmente, al entenderlo, se puede llegar a dominarlo. Esto se debe a que los desarrolladores, no tuvieron la intención de seguir algún patrón de desarrollo, sino hacer el *framework* lo más funcional posible [14].

El modelo en Django sigue siendo modelo, la vista en Django se llama Plantilla (*Template*) y el controlador en Django se llama Vista.

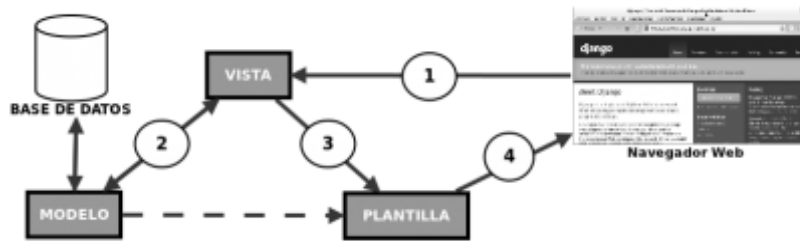


Figura 3: Funcionamiento Django

A continuación, se explica el funcionamiento del MVT de Django representado en la figura anterior:

1. El Navegador envía una solicitud.
2. La vista interactúa con el modelo para obtener datos.
3. La vista llama a la plantilla.
4. La plantilla renderiza la respuesta a la solicitud del navegador.

El modelo

El modelo define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros y posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos [14].

La vista

La vista se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados, entre otras cosas más. El ORM (*Object Relational Mapping*) de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista [14]. También se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y, la validación de datos a través de formularios [14].

La plantilla

Es la encargada de recibir los datos de la vista y luego los organiza para la presentación al navegador web. La plantilla es básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solamente crea contenido en HTML (también XML, CSS, JavaScript, CSV, etc.) [14].

Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación [34].

Los patrones de diseño de *software* son los que permiten describir fragmentos de diseño y reutilizar ideas de diseño, ayudando a beneficiarse de la experiencia de otros. Los patrones dan nombre y forma a heurísticas abstractas, reglas y buenas prácticas de técnicas orientadas a objetos. Ningún ingeniero razonable quiere partir de una pizarra

en blanco, y este libro ofrece una paleta de patrones de diseño que pueden utilizarse fácilmente.

Patrones Generales de Asignación de Responsabilidad (GRASP): constituyen un apoyo para la enseñanza que ayuda a uno a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades [30].

Experto

El Experto en Información se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. El Experto no pretende ser una idea oscura o extravagante; expresa la "intuición" común de que los objetos hacen las cosas relacionadas con la información que tienen. Indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo [30]. Este patrón se evidencia en la **CE_Auto** ya que es la que contiene toda la información acerca de un auto y va a tener la responsabilidad de la implementación de los métodos `crearAuto()`, `editarAuto()`, `eliminarAuto()` y `listaAuto()`.

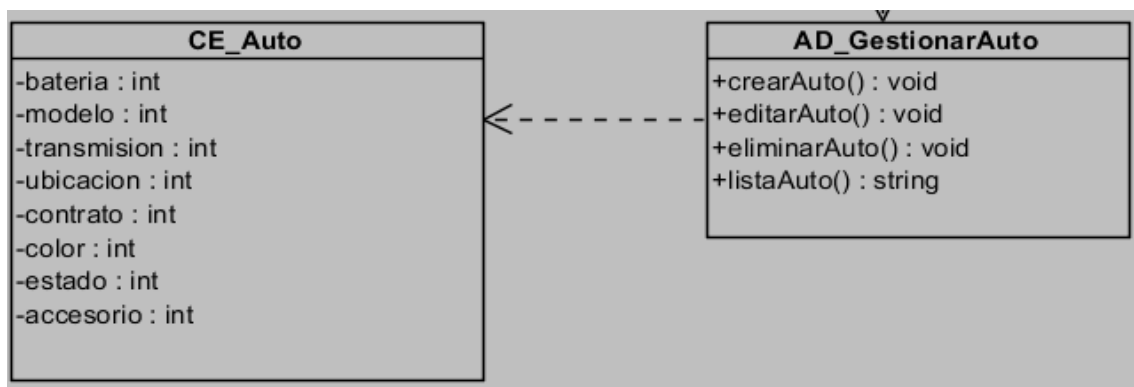


Figura 4: Clase Entidad Auto

Creador

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Eligiéndolo como el creador se favorece el bajo acoplamiento [30]. Este patrón se evidencia en la **CE_Auto** debido a que para realizar la creación de un auto utiliza una instancia de la clase `Color`.

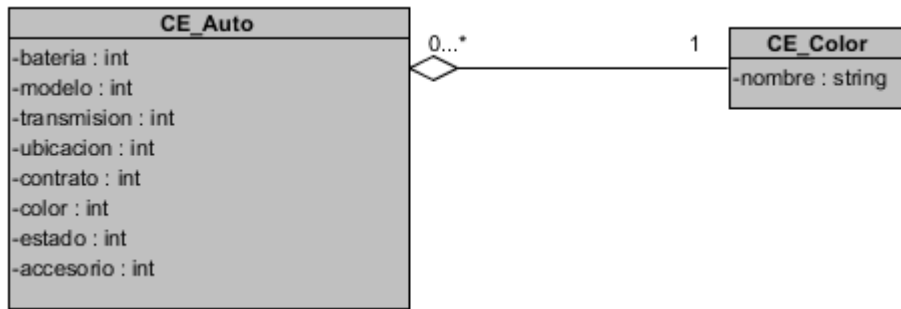


Figura 5: Clase Controladora InsertarAuto

Bajo acoplamiento

Describe que debe existir una alta reutilización entre las funcionalidades de las clases con una mínima dependencia, contribuyendo así al mantenimiento de las mismas [35] ver [Figura 9](#). Django implementa este patrón por defecto evitando altas dependencias entre los componentes, es decir, no importan los cambios que se realicen en alguno de estos componentes influirán de manera mínima en el resto.

Alta cohesión

Una clase con alta cohesión es ventajosa porque es relativamente fácil de mantener, entender y reutilizar. El alto grado de funcionalidad relacionada, combinada con un número pequeño de operaciones, también simplifica el mantenimiento y las mejoras. El grano fino de la funcionalidad altamente relacionada también aumenta el potencial de reutilización [30]. Este patrón se puede en la **CC_GestionarAuto** que contiene una serie de métodos estrechamente relacionados con la clase Auto.

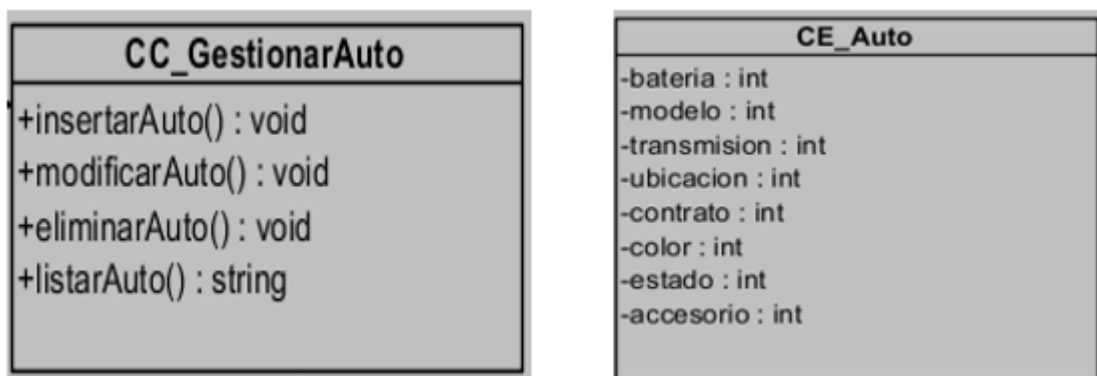


Figura 6: Función insertarAuto()

Controlador

El patrón Controlador proporciona guías acerca de las opciones generalmente aceptadas y adecuadas. Es una especie de fachada en la capa del dominio para la capa de la interfaz [30]. Este patrón se evidencia en la **CC_GestionarAuto** ya que es la encargada de recibir los datos del usuario y la que los envía a las distintas clases según el método llamado.

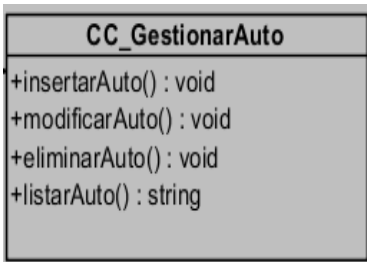


Figura 7: Función insertarAuto()

Banda de los cuatro (GoF): describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos [30].

Decorador

Es un patrón estructural que extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase [30]. Este patrón Django lo incluye por defecto ejemplo de ello lo constituye la función: **@login_required**.

```
@login_required
def insertarPosicionNeumatico(request):
    if request.method == 'POST':
        nombre = request.POST['nombre']
        siglas = request.POST['siglas']
```

Figura 8: Función para insertar un neumático

2.6 Diagrama de clases de diseño

Un Diagrama de Clases de Diseño (DCD) representa las especificaciones de las clases e interfaces *software* (por ejemplo, las interfaces de Java) en una aplicación [31].

A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño de los DCD muestran las definiciones de las clases *software* en lugar de los conceptos del mundo real [31].

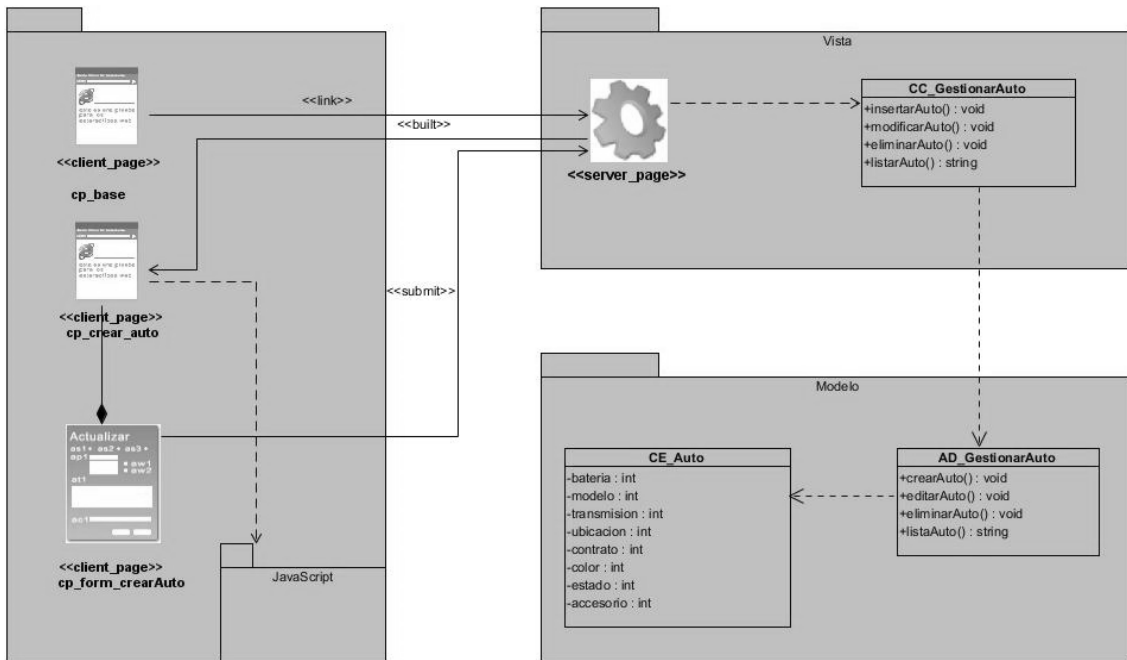


Figura 9: Diagrama de clases de diseño. HU1: Gestionar auto.

2.7 Diagrama de componentes

El diagrama de componentes ayuda a modelar el aspecto físico de un sistema de *software* orientado a objetos. Ilustra las arquitecturas de los componentes de *software* y las dependencias entre ellos. Aquellos componentes de *software* que incluyen componentes en tiempo de ejecución, componentes ejecutables y, los componentes de código fuente [36].

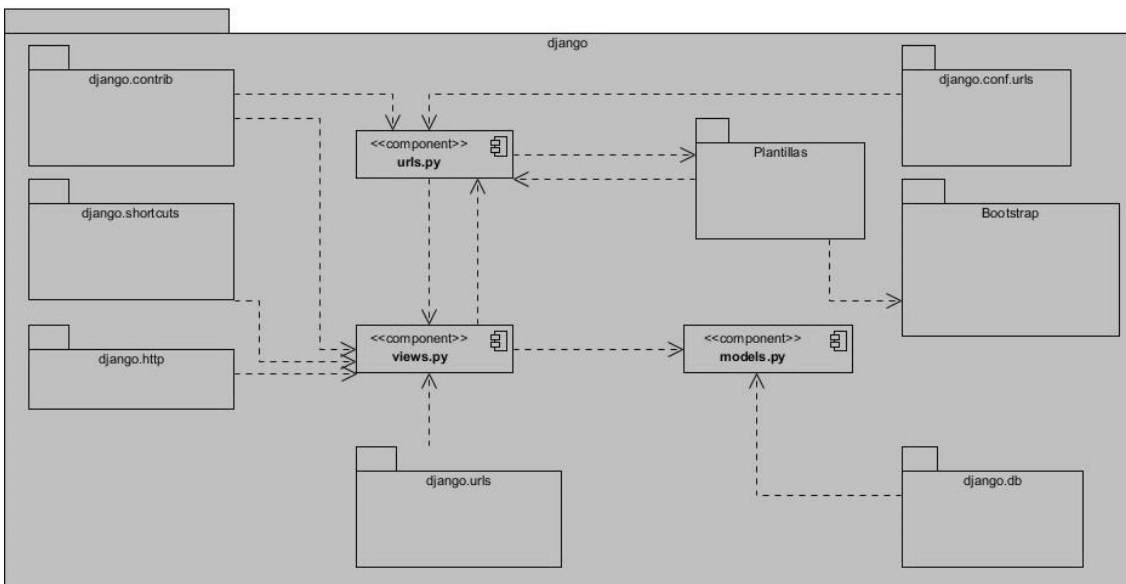


Figura 10: Diagrama de componentes

Django.conf.urls: es un paquete que gestiona las configuraciones relacionadas con las urls en Django.

Django.contrib: Se suministra con una variedad de herramientas adicionales opcionales que resuelven problemas comunes de desarrollo *web*.

Django.shorcuts: Este paquete recopila funciones de ayuda y clases que "abarcán" múltiples niveles de MVC.

Django.http: Este paquete recopila una serie de parámetros Http.

Django.contrib.auth: Este sistema maneja tanto la autenticación, que verifica si un usuario tiene permiso para acceder a un sitio determinado, como la autorización, que define lo que un usuario puede hacer o no en el sitio.

Django.db: va a manejar los datos referentes al trabajo con la base de datos.

Django.contrib.auth.models: Implementa una serie de módulos asociados a la autenticación.

urls.py: archivo que contiene todas las direcciones que va a contener la aplicación permitiendo encontrar la vista correcta.

models.py: archivo que contiene todas las clases modelo que posibilitan la interacción con la base de datos.

views.py: archivo que contiene todas las funciones de nuestra aplicación y que mediante las direcciones interactúa con la plantilla.

Plantillas: paquete que contiene todas las plantillas relacionadas con la aplicación.

Bootstrap: paquete que contiene todos los archivos JavaScript y CSS utilizados en la aplicación.

2.8 Modelo de datos

El modelo de datos es una colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad. Entre sus clasificaciones está el modelo de datos lógico que se usa para describir datos a nivel conceptual y externo caracterizándose por tener una amplia capacidad expresiva, son muy flexibles y permiten especificar ciertas restricciones de integridad [37].

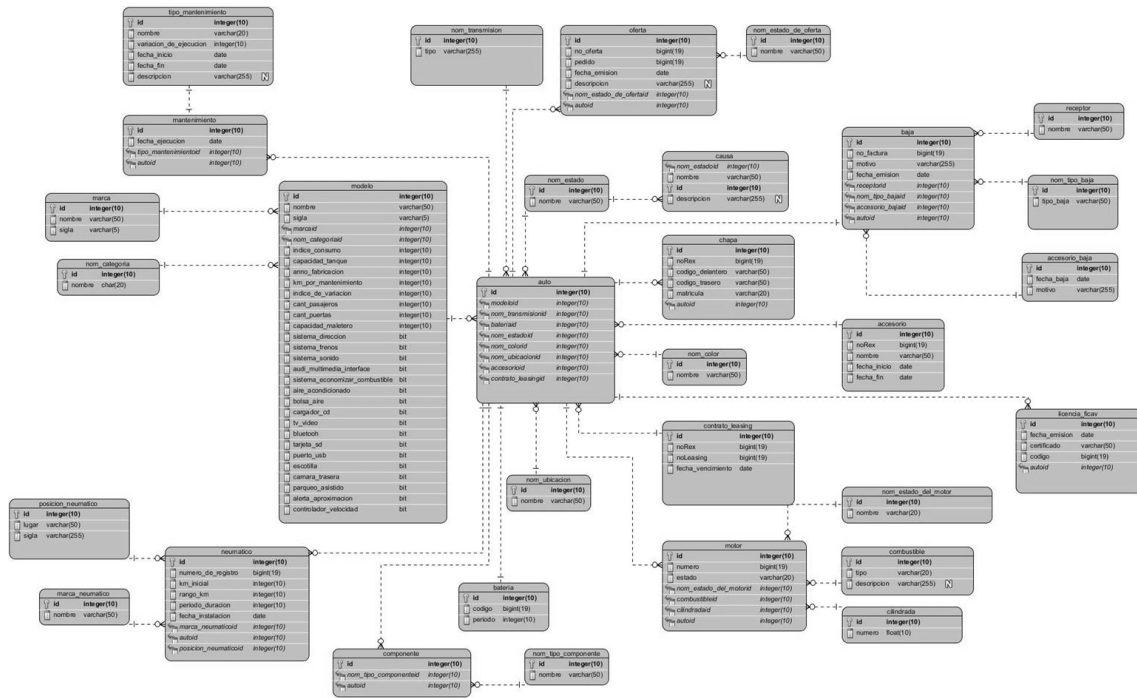


Figura 11: Modelo de datos

2.9 Modelo de despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de *hardware* y el *software* que ejecuta cada uno de ellos. Representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP. Son útiles para facilitar la comunicación entre los ingenieros de *hardware* y los de *software* [38].

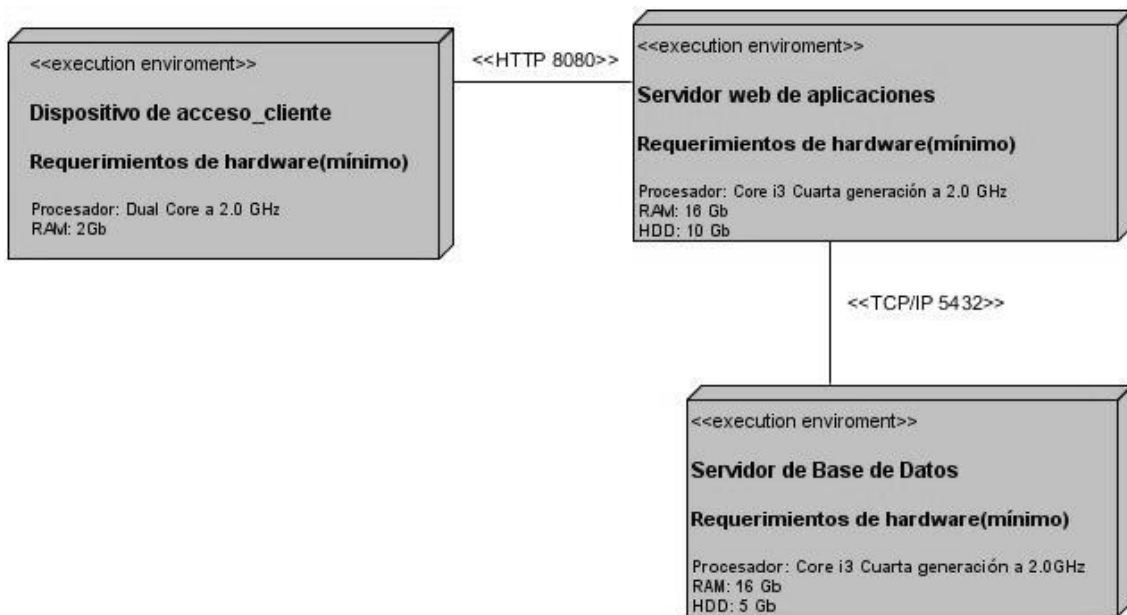


Figura 12: Diagrama de despliegue

El **Dispositivo de acceso_cliente** representa el dispositivo desde el cual el usuario, mediante un navegador web, podrá ejecutar alguna gestión. El **Servidor web de aplicaciones** es el encargado de atender las solicitudes del cliente el **Servidor de Base de Datos** provee servicios de base de datos. La comunicación entre el **Dispositivo de acceso_cliente** y el **Servidor web de aplicaciones** se realiza mediante el Protocolo Seguro de Transferencia de Hipertexto (HTTPS) y con el **Servidor de Base de Datos** mediante TCP/IP⁷.

2.10 Conclusiones parciales del capítulo

En desarrollo de este capítulo se trataron aspectos correspondientes al análisis y diseño del módulo Flota llegándose a las siguientes conclusiones:

- La elaboración de los requisitos funcionales y no funcionales, historias de usuario y demás artefactos generados permitió conocer las funcionalidades del sistema y sus comportamientos específicos y, la aplicación de patrones GRASP y GoF para un adecuado diseño de la propuesta de solución.
- La definición de la arquitectura del sistema y los patrones de diseño a utilizar permitieron establecer buenas prácticas de programación así como, facilitar los cambios futuros en los componentes del sistema o en el código fuente.

⁷ Provee conectividad de extremo a extremo especificando cómo los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario.

Capítulo 3: Implementación y pruebas del módulo Flota del Sistema de gestión para la Agencia de Renta de Vehículos REX

En el presente capítulo se definen los estándares de codificación que debe utilizar el programador en la fase de implementación, logrando un mejor entendimiento del código por otros o la hora de su reutilización. También se generan las pruebas seleccionadas para la validación de la implementación, verificando en cada una de ellas que los resultados sean los esperados.

3.1 Estándar de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Una faceta del desarrollo de *software* en la que todos los programadores influyen especialmente es en la técnica de codificación.

“El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación” [39].

Tabla 5: Estándar de codificación

| Tipo de estándar | Descripción |
|---------------------------|---|
| Sangría | <ul style="list-style-type: none">Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes).Utilizar una indentación de una tabulación para cada línea con excepción de la primera.La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios. |
| Máxima longitud de líneas | <ul style="list-style-type: none">Todas las líneas deben estar limitadas a un máximo de setenta y nueve (79) caracteres.Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas. |
| Líneas en blanco | <ul style="list-style-type: none">Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco.Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco.Se puede utilizar líneas en blanco escasamente para separar secciones lógicas. |
| Codificación | <ul style="list-style-type: none">Utilizar la codificación UTF-8. |

| | |
|---|---|
| | <ul style="list-style-type: none"> Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”. |
| Importaciones | <ul style="list-style-type: none"> Las importaciones deben estar en líneas separadas. Siempre deben colocarse al comienzo del archivo. Deben quedar agrupadas de la siguiente forma: <ol style="list-style-type: none"> Importaciones de la librería estándar. Importaciones terceras relacionadas. Importaciones locales de la aplicación / librerías. Cada grupo de importaciones debe estar separado por una línea en blanco. |
| Espacios en blanco en expresiones y declaraciones | <ul style="list-style-type: none"> Evitar utilizar espacios en blanco en las siguientes situaciones: <ul style="list-style-type: none"> Inmediatamente dentro de paréntesis, corchetes y llaves. Inmediatamente antes de una coma, un punto y coma o dos puntos. Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función. Inmediatamente antes de un corchete que empieza una indexación. Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro. Deben rodearse con exactamente un espacio los siguientes operadores binarios: <ol style="list-style-type: none"> Asignación (=). Asignación de aumentación (+=, -=, etc.). Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not). Expresiones lógicas (and, or, not). Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad. No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto. |
| Comentarios | <ul style="list-style-type: none"> Los comentarios deben ser oraciones completas. Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula. Nunca cambiar las minúsculas y mayúsculas en los identificadores de clases, objetos, funciones, etc. Si un comentario es corto el punto final puede omitirse. |
| Comentarios en bloque | <ul style="list-style-type: none"> Deben estar indentados al mismo nivel que el código a comentar. Cada línea de un comentario en bloque comienza con un numeral (#) y un espacio en blanco. |
| Comentarios en la línea | <ul style="list-style-type: none"> Se recomienda utilizarlos moderadamente. |

| | |
|------------------------------|--|
| | <ul style="list-style-type: none"> • Se debe definir comenzando por un numeral (#) seguido de un espacio en blanco. • Deben ubicarse en la misma línea que se desea comentar. |
| Cadenas de documentación | <ul style="list-style-type: none"> • Deben quedar documentados todos los módulos, funciones, clases y métodos públicos. • Para definir una cadena de documentación debe quedar encerrada dentro de (""). • Los (") que finalizan una cadena de documentación deben quedar en una línea a no ser que la cadena sea de una sola línea. |
| Convenciones de nomenclatura | <ul style="list-style-type: none"> • Nunca se deben utilizar como simple caracteres para nombres de variables los caracteres ele minúscula "l", o mayúscula "O", ele mayúscula "L" ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0). • Los módulos deben tener un nombre corto y en minúscula. • Los nombres de clases deben utilizar la convención "CapWords" (palabras que comienzan con mayúsculas). • Los nombres de las excepciones deben estar escrito también en la convención "CapWords" utilizando el sufijo "Error". • Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guión bajo "_". • Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guión bajo (_). |

3.2 Estrategia de pruebas de software

Una estrategia de prueba de *software* proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados, además, debe ser suficientemente flexible para promover un uso personalizado de la prueba. Al mismo tiempo, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto [28].

3.2.1 Pruebas funcionales al sistema

Se denominan pruebas funcionales a las pruebas de *software* que tienen por objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y salida.

Método de prueba

La prueba de caja negra se refiere a las pruebas que se llevan a cabo en la interfaz del *software*. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del *software*. También llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del *software*; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa [28].

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

1. Funciones incorrectas o faltantes.
2. Errores de interfaz.
3. Errores en las estructuras de datos o en el acceso a bases de datos externas.
4. Errores de comportamiento o rendimiento.
5. Errores de inicialización y terminación [28].

Partición de equivalencia

La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos carácter) que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general [28].

El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana [28].

Tabla 6: Variables empleadas en el caso de prueba basado en la HU "Insertar tipo de mantenimiento"

| No. | Nombre del campo | Clasificación | Valor Nulo | Descripción |
|-----|------------------------|---------------|------------|---|
| 1 | Nombre | Campo texto | No | Se inserta una cadena de texto no mayor a veinte caracteres comenzando con mayúscula, puede contener valores alfanuméricos. |
| 2 | Variación de ejecución | Campo entero | No | Se introduce un número entero que representa el intervalo de kilómetros a realizar el mismo tipo de mantenimiento. |
| 3 | Fecha de inicio | Campo fecha | No | Se selecciona la fecha de inicio, es decir, la fecha de comienzo del tipo de mantenimiento. |

| | | | | |
|---|--------------|-------------|----|---|
| 4 | Fecha de fin | Campo fecha | No | Se selecciona la fecha de fin, es decir, la fecha de culminación del tipo de mantenimiento. |
| 5 | Descripción | Campo texto | Si | Se proporciona una breve descripción de las especificaciones del tipo de mantenimiento. |

Las celdas de la tabla contienen V (indica válido), I (indica inválido).

Tabla 7: Caso de prueba correspondiente a la HU "Insertar tipo de mantenimiento"

| Escenario | Descripción | V1 | V2 | V3 | V4 | V5 | R/ del sistema | Flujo central |
|---|---|------------------|------------|-----------------|-----------------|---|---|---|
| EC1.1 Introducir datos correctamente | EL usuario introduce los datos correspondientes a un tipo de mantenimiento correctamente. | V Rodamientos | V 5000 | V 2017-09-11 | V 2017-09-12 | V Se procede a realizar el engrase de los rodamientos de los cuatro rodamientos, revisión de los bujes y transmisión | El sistema crea el tipo de mantenimiento notificando al usuario de que la operación se realizó correctamente y redirrecciona a la lista de tipos de mantenimiento | El usuario rellena los campos correctamente y da clic en el botón de enviar |
| EC1.2 Campos vacíos | El usuario deja uno o varios campos vacíos | V Rodamientos | I vacío | I vacío | I vacío | V vacío | El sistema muestra un mensaje anunciando "Debe llenar los campos variación de ejecución, fecha de inicio y fecha de fin" | Completando el formulario o quedando algo |
| | | I | V | I | I | V | | |

| | | | | | | | | |
|--|--|---------------------------------|-------|------------|------------|--|--|-------------------|
| | | vacío | 5000 | vacío | vacío | vacío | El sistema muestra un mensaje anunciando "Debe llenar los campos nombre, fecha de inicio y fecha de fin" | nos campos vacíos |
| | | I | I | V | V | V | El sistema muestra un mensaje anunciando "Debe llenar los campos nombre y variación de ejecución" | |
| | | vacío | vacío | 2017-09-11 | 2017-09-12 | Se procede a realizar el engrase de los rodamientos de los cuatro rodamientos, revisión de los bujes y transmisión | | |
| | | V | I | V | I | V | El sistema muestra un mensaje anunciando "Debe llenar el campo variación de ejecución y fecha de fin" | |
| | | Cam bio de gom as de carre tera | vacío | 2017-09-10 | vacío | Se procede a realizar el desmontaje de las cuatro gom as del auto | | |

| | | | | | | | | |
|--|---|---|------|------------|------------|---|---|---|
| | | | | | | para hacer el cambio por otras nuevas | | |
| EC1.3 Introducir campos incorrectos | El usuario introduce uno a varios valores incorrectos | V | I | V | V | V | El sistema muestra un mensaje anunciando "El campo nombre debe comenzar con letra mayúscula y la variación de ejecución solo puede contener caracteres numéricos" | El usuario introduce uno o varios valores incorrectos y da clic en el botón de enviar |
| | | sustitución de caja de cambios automática | 100a | 2017-09-10 | 2017-09-13 | Se procede a realizar el desmontaje de la caja de cambios debido al desgaste del componente y se realiza luego la colocación de una nueva caja de cambios | | |
| | | V | V | V | V | V | El sistema muestra un mensaje anunciando "El campo nombre debe contener un máximo de | |
| | | Cambio y reparación del sistema | 7000 | 2017-09-10 | 2017-09-20 | vacío | | |

| | | | | | | | | |
|--|--|--------------|--|--|--|--|------------------------------|--|
| | | de dirección | | | | | 20 caracteres alfanuméricos" | |
|--|--|--------------|--|--|--|--|------------------------------|--|

Resultado de las pruebas funcionales:

Con el objetivo de validar los requisitos funcionales se realizaron tres iteraciones de pruebas al módulo Flota. En la Tabla 8 se muestran los resultados obtenidos en cada iteración de prueba, así como la corrección de cada uno de los errores.

Tabla 8: Resultado de las pruebas funcionales

| No conformidades | 1ra. Iteración | 2da. Iteración | 3ra. Iteración |
|-------------------|----------------|----------------|----------------|
| Detectadas | 32 | 13 | 0 |
| Resueltas | 32 | 13 | 0 |
| Pendientes | 0 | 0 | 0 |

En la Figura 13 se muestran los resultados obtenidos en cada una de las iteraciones de pruebas realizadas al módulo Flota.

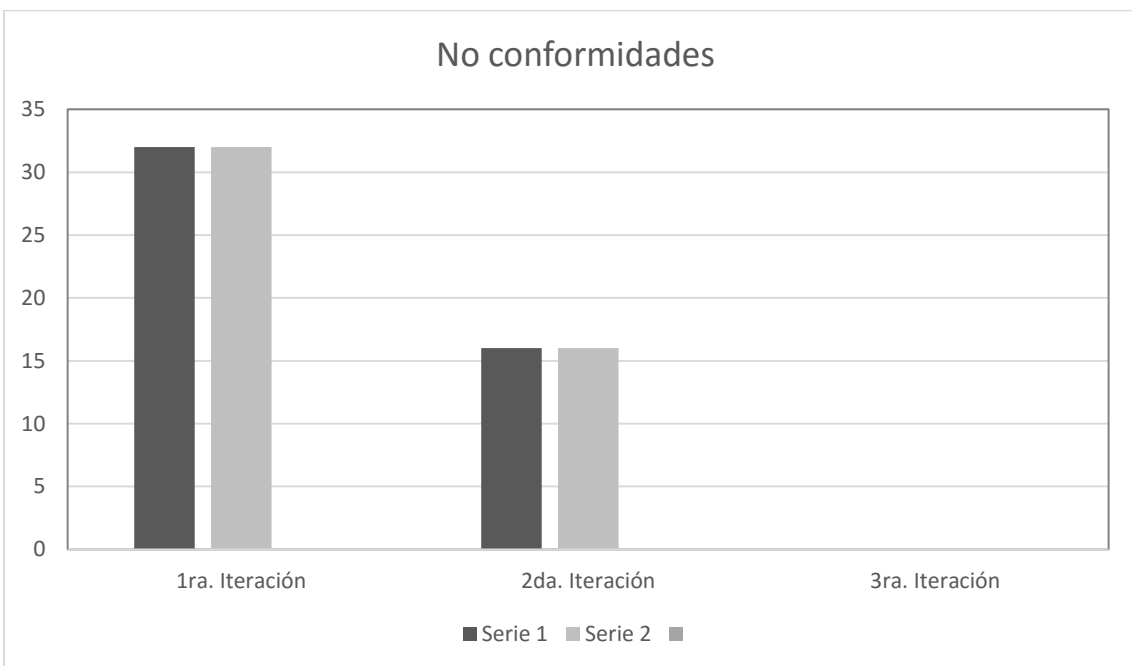


Figura 13: Gráfico de inconformidades.

Entre las no conformidades detectadas durante la realización de las pruebas funcionales se encuentran:

Tabla 9: Tipos de no conformidades. Pruebas funcionales

| No conformidades | Cantidad |
|--|----------|
| Vistas que no redireccionaban de forma correcta. | 2 |
| Plantillas que no muestran el contenido. | 3 |
| Incorrecta gestión de los datos en las vistas. | 5 |

| | |
|---|----|
| Las alertas de error no coincidían con el error. | 6 |
| Errores ortográficos en el texto de las alertas. | 20 |
| Datos que se mostraban incorrectamente. | 3 |
| Errores en las relaciones de las tablas de la base de datos | 9 |

3.2.2 Pruebas de seguridad

Las pruebas de seguridad buscan medir la Confidencialidad, Integridad y Disponibilidad de los datos, desde la perspectiva del aplicativo, es decir partiendo a identificar amenazas y riesgos desde el uso o interface de usuario final. Una vez ejecutadas las pruebas de seguridad es posible medir y cuantificar los riesgos a los cuales se ven expuestos los aplicativos tanto en la infraestructura interna como externa, valiéndose de la filosofía del Hacking ético [40].

Al módulo desarrollado se le realizaron una serie de pruebas de seguridad mediante el software Acunetix, las cuales se presentan a continuación:

- Ataques de inyección SQL.
- Cross-Site Scripting (XSS).
- Falsificación de petición (CSRF).

Vulnerabilidades detectadas

Tabla 10: Pruebas de seguridad

| Tipo | Cantidad | Descripción |
|----------------|----------|---|
| Implementación | 20 | Formularios HTML sin protección CSRF. |
| Configuración | 5 | La configuración del servidor de aplicaciones permite la visualización de información sensible del negocio. |

Observaciones:

La prueba realizada mediante la herramienta Acunetix devolvió que se detectan veinticinco no conformidades relacionadas con la configuración del servidor de aplicaciones y la implementación del *software*, solucionando las vulnerabilidades detectadas que pudieran poner en riesgo la seguridad e integridad del módulo.

3.2.3 Pruebas de rendimiento

La prueba del rendimiento ocurre a lo largo de todos los pasos del proceso de prueba. Incluso en el nivel de unidad, puede accederse al rendimiento de un módulo individual conforme se realizan las pruebas. Sin embargo, no es sino hasta que todos los

elementos del sistema están plenamente integrados cuando puede determinarse el verdadero rendimiento de un sistema [28].

Las pruebas de rendimiento por lo general requieren instrumentación de *hardware* y de *software*, es decir, con frecuencia es necesario medir la utilización de los recursos (por ejemplo, ciclos del procesador) en forma meticulosa. La instrumentación externa puede monitorear intervalos de ejecución y eventos de registro (por ejemplo, interrupciones) conforme ocurren, y los muestreos del estado de la máquina de manera regular. Con la instrumentación de un sistema, la persona que realiza la prueba puede descubrir situaciones que conduzcan a degradación y posibles fallas del sistema [28].

Pruebas de carga

Mediante la ejecución de las pruebas de Carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la misma [41].

Pruebas de estrés

Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o procesos, una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio [41].

Resultados de las pruebas de rendimiento

Para la realización de las pruebas de carga y estrés al módulo Flota se utilizó el software Apache JMeter en su versión 2.8.4.

Hardware de prueba (PC cliente):

- Tipo de procesador: Intel(R) Core(TM) 2 Duo CPU @ 2.26GHz 2.27GHz
- RAM: 3 GB DDR2.
- Tipo de Red: Ethernet 10/100Mbps.

Hardware de prueba (PC servidor):

- Tipo de procesador: Intel(R) Core(TM) i3-6100u @ 2.30GHz 2.30GHz
- RAM: 6 GB DDR3.
- Tipo de Red: Ethernet 10/100Mbps.

Software instalado en ambas PC:

- Tipo de servidor web: Apache.
- Plataforma: Windows 10 Home (PC cliente y servidor)
- Servidor de BD: PostgreSQL 9.4.

La prueba define doscientos hilos de concurrencia, los cuales simulan doscientos usuarios accediendo concurrentemente. Se simularon un total de mil peticiones a cinco direcciones del módulo en el servidor. En la Tabla 10 se puede observar los resultados obtenidos por el sistema.

Tabla 11: Informe agregado de jmeter

| Etiqueta | # Muestras | Media | Mediana | 90% Line | 95% Line | 99% Line | Mín | Máx | % Error | Rendimiento | Kb/sec | Sent KB/se |
|---------------|------------|-------|---------|----------|----------|----------|------|-------|---------|-------------|--------|------------|
| home | 200 | 2281 | 2172 | 3527 | 3909 | 4450 | 304 | 4694 | 72,50% | 42,1/sec | 109,69 | 7,5 |
| ficha tecnica | 200 | 4396 | 2005 | 10513 | 10863 | 11998 | 1742 | 12102 | 64,00% | 12,2/sec | 292,55 | 0,8 |
| insertar auto | 200 | 6431 | 8872 | 11153 | 11435 | 11744 | 2000 | 11798 | 49,00% | 8,1/sec | 218,15 | 0,8 |
| insertar m... | 200 | 4803 | 8236 | 8309 | 9032 | 9273 | 0 | 9295 | 100,00% | 7,0/sec | 13,65 | 0,0 |
| dti | 200 | 4677 | 2023 | 10057 | 10370 | 10964 | 892 | 11044 | 46,50% | 9,6/sec | 286,41 | 1,0 |
| Total | 1000 | 4518 | 2141 | 10499 | 10757 | 11585 | 0 | 12102 | 66,40% | 28,4/sec | 483,01 | 2,6 |

Descripción de los parámetros evaluados:

Muestras: cantidad de hilos utilizados para la URL.

Media: tiempo promedio en milisegundos para un conjunto de resultados.

Min: tiempo mínimo que demora un hilo en acceder a una página.

Max: tiempo máximo que demora un hilo en acceder a una página.

%Error: porciento de error de las respuestas de las peticiones.

Rendimiento: rendimiento medido en los requerimientos por segundo / minuto / hora.

Kb/s Recibidos: rendimiento medido en Kbyte por segundo.

3.2.4 Pruebas de integración

Las pruebas de integración se realizan para probar que todos los elementos (módulos) que componen el sistema funcionen correctamente las relaciones entre ellos.

Tabla 12: Pruebas de integración

| Sistema | Funcionalidades | Resultado de la prueba |
|---|--|---|
| Sistema de Gestión para la Agencia de Renta de Vehículos REX. | Instalación y configuración del módulo Flota en el nuevo Sistema de Gestión para la Agencia de Renta de Vehículos REX. | El módulo se integra satisfactoriamente logrando los procesos de gestión de la flota de autos, sin afectar las funcionalidades del sistema. |

3.3 Conclusiones parciales del capítulo

En el presente capítulo se determinó:

- El uso de los estándares de codificación permitió homogeneizar la estructuración del código asegurando la calidad, evitando errores y permitiendo un fácil mantenimiento.
- La aplicación de las pruebas funcionales, aceptación, seguridad, rendimiento e integración permitieron demostrar el cumplimiento de las funcionalidades del módulo e identificar las principales deficiencias, así como, la estrategia para solucionar los errores detectados y obtener un producto con un alto valor y utilidad para una adecuada gestión de los autos.

Conclusiones

Con la investigación realizada se logró el desarrollo del módulo *web* Flota del Sistema de Gestión para la Agencia de renta de autos REX. Los objetivos propuestos fueron cumplidos con satisfacción, por lo que se puede arribar a las siguientes conclusiones:

- El análisis de las diferentes herramientas, metodología, tecnologías y tendencias actuales para la renta, permitió sentar las bases para el desarrollo de la propuesta de solución.
- La definición y especificación de la arquitectura, requisitos funcionales, no funcionales y los artefactos generados por la metodología, permitieron obtener las capacidades del sistema y estructurar la solución implementada.
- La obtención de una aplicación web para el proceso de gestión de información de la flota de autos del Sistema de Gestión para la Agencia de Renta de Vehículos REX permitirá el control de los accesorios y los tipos de mantenimientos de los autos mediante el uso de tecnologías actualizadas.
- Con la realización de las distintas pruebas aplicadas al sistema, se verificó la conformidad con los requisitos.

Recomendaciones

Al finalizar la investigación y el desarrollo de la propuesta de solución, el autor recomienda desarrollar nuevas funcionalidades que aporten al sistema características que contribuyan a mejorar la gestión y el control de la información de la flota de autos de la agencia REX como son:

- Agregar el número de serie de los neumáticos para un mejor control sobre los mismos.
- Enviar una notificación vía correo electrónico al jefe de la sucursal informando sobre las gestiones que realice el especialista para tener constancia de las gestiones que se realicen.

Referencias bibliográficas

- [1] Misión de la UCI. [En línea]. Disponible en: <http://www.uci.cu/?q=mision>. Consulta: 9 de noviembre de 2016.
- [2] Chávez Montejo, Yarelys; Sousa Pérez, Hilda. Gestión documental, Gestión de información y Gestión del conocimiento: nociones e interrelaciones. Facultad de Comunicación. Universidad de La Habana, 2013, No. 8-9.
- [3] Hertz. [En línea]. Disponible en: <https://www.hertz.com/>. Consulta: 10 de noviembre de 2016.
- [4] Interhome. [En línea]. Disponible en: <https://www.interhome.es/>. Consulta: 10 de noviembre de 2016.
- [5] Airbnb. [En línea]. Disponible en: <https://www.airbnb.es/>. Consulta: 11 de noviembre de 2016.
- [6] Havanautos. [En línea]. Disponible en: http://www.transturcarrental.com/car_rental_companies/havanautos_car_rental.asp. Consulta: 11 de noviembre de 2016.
- [7] Grupo Gaviota. [En línea]. Disponible en: <http://www.gaviota-grupo.com/autos>. Consulta: 12 de noviembre 2016.
- [8] Metodología de Desarrollo de Software. [En línea]. Disponible en: <http://www.um.es/docencia/barzana/IAGP/Iagp2.html>. Consulta: 11 de noviembre de 2016.
- [9] CANÓS, José H.; LETELIER, Patricio; PENADÉS, M^a Carmen. Metodologías ágiles en el desarrollo de software. Metodologías Ágiles en el Desarrollo de Software, 2003, p. 1.
- [10] CADAVID, Andrés Navarro. Revisión de metodologías ágiles para el desarrollo de software. Prospectiva, 2013, vol. 11, no 2, p. 30-39.
- [11] RODRÍGUEZ, Tamara. Metodología de desarrollo para la Actividad productiva de la Universidad de las Ciencias Informáticas. La Habana. 2015
- [12] GUTIÉRREZ, Javier J. ¿Qué es un framework Web? Available in: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf Accessed May, 2014, vol. 12.
- [13] Django en Español. [En línea]. Disponible en: <http://django.es/>. Consulta: 14 de noviembre de 2016.
- [14] Django. [En línea]. Disponible en: <https://www.djangoproject.com/>. Consulta: 14 de noviembre de 2016.
- [15] Mozilla Developer Network. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/HTML/HTML5>. Consulta: 12 noviembre 2016.

- [16] Webwera.es. [En línea]. Disponible en: <http://www.lawebera.es/maquetacion-web/darle-uso-apropiado-css3.php>. Consulta: 11 noviembre 2016.
- [17] OTTO, M; Thornton, J. Bootstrap 3, el manual oficial. 2013.
- [18] KUCHLING, Andrew. Interview with Guido van Rossum. Linux Journal Nov # 55[seriada en línea]. Disponible en: <http://www.linuxjournal.com/article/2959> [consulta: 3 de octubre de 2016]
- [19] Postgres. [En línea]. Disponible en: <http://www.postgresql.org/es/>. Consulta: 13 de noviembre de 2016.
- [20] CABELLO, María Victoria Nevado. Introducción a Las Bases de Datos Relacionales. Editorial Visión Libros, 2010.
- [21] BAREISA, E. Development of case tools for software process improvement. [En línea]. Disponible en: <http://www.irem.ktu.lt/index.php/ITC/article/viewFile/11997/6672>. Consulta: 13 de noviembre de 2016.
- [22] PARADIGM, Visual. Visual paradigm for UML. Visual Paradigm for UML-UML tool for software application development, 2013.
- [23] JURADO, Francisco, et al. Cole-Programming: Incorporando Soporte al Aprendizaje Colaborativo en Eclipse. IEEE-RITA, 2012, vol. 7, no 3, p. 121-130.
- [24] QUIROZ, Patricio; MUÑOZ, Roberto; NOËL, René. Desarrollo de un Lenguaje de Programación y Entorno de Desarrollo que Facilite la Programación de Robots LEGO Mindstorms. En XVII Congreso Internacional de Informática Educativa. 2012. p. 50-53.
- [25] vitm.edu.in/_notes/CSI_April_2016.pdf#page=21.
- [26] JetBrains. [En línea]. Disponible en: <https://blog.jetbrains.com/pycharm/2016/03/announcing-general-availability-of-pycharm-2016-1/> . Consulta: 1 de diciembre de 2016.
- [27] CHALLENGER-PÉREZ, Ivet; DÍAZ-RICARDO, Yanet; BECERRA-GARCÍA, Roberto Antonio. El lenguaje de programación Python. Ciencias Holguín, 2014, vol. 20, no 2, p. 1-12.
- [28] PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico, Séptima edición ed. 2010.
- [29] Lenguaje Unificado de Modelado. [En línea]. Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf> . Consulta: 9 de febrero de 2017.
- [30] LARMAN, Diapositivas. Modelo Del Dominio. de UML y Patrones, Prentice Hall, 2003, p. 23.
- [31] SOMMERVILLE, I. Ingeniería de Software (Novena edición ed.). V. C. Olgúin, Trad.) México: PEARSON EDUCACIÓN, 2011.

- [32] LARMAN, Craig. UML y Patrones. Pearson, 1999.
- [33] K. I. Tong, "Chapter 8 Managing Software Projects with User Stories," in Essential Skills for Agile Development, 2010, pp. 217 – 252.
- [34] COSMIN, Puiu Ionut. Patrones de Diseño. MoleQla: revista de Ciencias de la Universidad Pablo de Olavide, 2016, no 23, p. 36.
- [35] PERALTA, C., & DURÁN, D. Módulos de edición de plantillas y recepción de órdenes de impresión para el Sistema de Personalización de Documentos de Identidad basado en tecnologías libres. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana, 2014.
- [36] Diagrama de componentes. [En línea]. Disponible en: <https://www.visual-paradigm.com/VPGallery/diagrams/Component.html>. Consulta: 10 de febrero 2017.
- [37] CARRILLO, Silvia A, et al. Introducción a las Bases de Datos: El modelo relacional. Editorial Paraninfo, 2005.
- [38] GRAU, Xavier Ferré; SEGURA, María Isabel Sánchez. Desarrollo orientado a objetos con UML. Recuperado el, 2008, vol. 1.
- [39] DOCS, G. Estándar de Codificación. Recuperado el, vol. 3.
- [40] QUALITY, V. V. Pruebas de seguridad. V&V Quality Performance, security & Process Solutions, 2015.
- [41] QUALITY, V. V. Pruebas de rendimiento. V&V Quality Performance, security & Process Solutions, 2015.

Bibliografía

Lenguajes de programación. [En línea]. Disponible en: <http://es.ccm.net/contents/304-lenguajes-de-programacion>. Consulta: 17 de noviembre de 2016.

SEVERANCE, Charles. Python para informáticos: Explorando la información. [En línea]. Disponible en: <http://do1.dr-chuck.net/py4inf/ES-es/book.pdf> . Consulta: 7 de diciembre de 2016.

VAN ROSSUM, Guido. El tutorial de Python. [En línea]. Disponible en: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>. Consulta: 7 de diciembre de 2016. PL. Las TIC serán un sector de desarrollo estratégico en Cuba. [En línea]. Disponible en: <http://www.cubasi.cu/cubasi-noticias-cuba-mundo-ultima-hora/item/49118-las-tic-seran-un-sector-de-desarrollo-estrategico-en-cuba>. Consulta: 8 de noviembre de 2016.

Cubadebate. Las TIC tienen potencial para el desarrollo de nuestros países. [En línea]. Disponible en: <http://www.cubadebate.cu/opinion/2014/04/25/las-tics-tienen-potencial-para-el-desarrollo-de-nuestros-paises/>. Consulta: 8 de noviembre de 2016.

Cubaperiodistas. Imposible el desarrollo al margen de las TIC. [En línea]. Disponible en: <http://www.cubaperiodistas.cu/index.php/2016/03/informatica-2016-imposible-el-desarrollo-al-margen-de-las-tics/>. Consulta: 8 de noviembre de 2016.

Trabajadores. Desarrollo de las TIC por el bien ciudadano. [En línea]. Disponible en: <http://www.trabajadores.cu/20160314/desarrollo-las-tic-bien-ciudadano/>. Consulta: 9 de noviembre de 2016.

REX. [En línea]. Disponible en: <http://www.rex.cu/client/home/index.php>. Consulta: 9 de noviembre de 2016.

BERMÚDEZ SARGUERA, Dr. C. Rogelio, RODRÍGUEZ REBUSTILLO, Dra. C. Marisela. Lo empírico y lo teórico: ¿Una clasificación válida cuando se trata de los métodos de la investigación científica? [En línea]. Disponible en: http://cvi.mes.edu.cu/peduniv/index.php/peduniv/article/download/713/pdf_84.

Consulta: 9 de noviembre de 2016.

LÓPEZ VILLACIS, Dr. C. David, PILATAXI, Juan E. Métodos teóricos de la investigación y métodos empíricos de la investigación. [En línea]. Disponible en: <https://es.scribd.com/doc/316497233/metodos-teoricos-y-metodos-empiricos>. Consulta: 11 de noviembre de 2016.

VAN ROSSUM, Guido. El tutorial de Python. [En línea]. Disponible en: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>. Consulta: 15 de noviembre de 2016. SAMAMÉ SILVA, Jaime Humberto. Aplicación de una metodología ágil en el desarrollo de un sistema de información. 2013.

LÓPEZ, P y FRANCISCO, R. s/f. Ingeniería de software. [En línea]. VE. Disponible en: <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>. Consulta: 6 de mayo de 2015.

RODRÍGUEZ, E. 2012. Conceptos básicos de Ingeniería de Software. [En línea]. VE. Disponible en: <http://www.tamps.cinvestav.mx/~ertello/swe/sesion01.pdf>. Consulta: 6 de mayo de 2015.

Ecured. [En línea]. Disponible en: https://www.ecured.cu/Diagrama_de_despliegue. Consulta: 10 de febrero de 2017.

GARCÍA, Luis Ambrosio Velázquez. Gestión y tecnología para la ingeniería de requerimientos en servicios computacionales/Management and technology for engineering requirements in computer services. RECI Revista Iberoamericana de las Ciencias Computacionales e Informática, 2016, vol. 5, no 10, p. 59-78.

GUERRERO, Carlos A; SUAREZ, Johanna M. y GUTIERREZ, Luz E. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web, vol.24, n.3, pp.103-114, 2013. [18] Peralta, C., & Durán, D. Módulos de edición de plantillas y recepción de órdenes de impresión para el Sistema de Personalización de Documentos de Identidad basado en tecnologías libres. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. La Habana, 2017.

CARRILLO, SILVIA ACID, et al. Introducción a las Bases de Datos: El modelo relacional. Editorial Paraninfo, 2005.

GOÑI CAMEJO, Ivis. Algunas reflexiones sobre el concepto de información y sus implicaciones para el desarrollo de las ciencias de la información. Acimed, 2000, vol. 8, p. 201-207.

Anexos

Tabla 13: HU Gestionar marca de auto

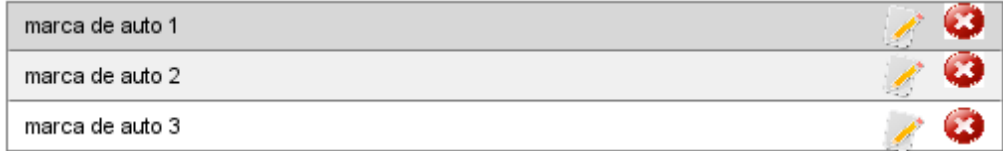
| Anexo | |
|--|---|
| Número: 0 | Nombre del requisito: Gestionar marca de auto |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una marca de auto se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Modelo (Obligatorio. Campo selector. Permite la selección de un modelo). • Transmisión (Obligatorio. Campo selector. Permite la selección de una transmisión). • Batería (Obligatorio. Campo selector. Permite la selección de una batería). • Estado (Obligatorio. Campo selector. Permite la selección de un estado). • Color (Obligatorio. Campo selector. Permite la selección de un color). • Ubicación (Obligatorio. Campo selector. Permite la selección de una ubicación). • Accesorio (Obligatorio. Campo selector. Permite la selección de un accesorio). • Contrato Leasing (Obligatorio. Campo selector. Permite la selección de un contrato). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de marcas de autos. | |
| Prototipo de interfaz: | |
|  | |

Tabla 14: HU Gestionar categoría

| Anexo | |
|--|---|
| Número: 0 | Nombre del requisito: Gestionar categoría |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una categoría se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> | |

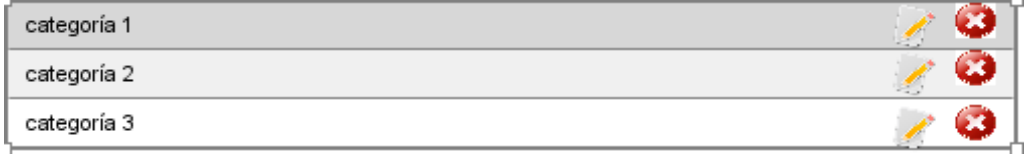
| |
|---|
| <ul style="list-style-type: none"> Modelo (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de categorías. |
| Prototipo de interfaz:  |

Tabla 15: HU Gestionar modelo de auto

| | |
|---|--|
| Número: 0 | |
| Número: 0 | Nombre del requisito: Gestionar modelo de auto |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un modelo de auto se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> Marcas (Obligatorio. Campo selector. Permite la selección de una marca). Categoría (Obligatorio. Campo selector. Permite la selección de una categoría). Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). Siglas (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). Índice de consumo (Obligatorio. Campo de tipo real. Permite la entrada de caracteres numéricos y coma). Capacidad del tanque (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Año de fabricación (Obligatorio. Campo de tipo fecha. Permite la selección de una fecha). Km por mantenimiento (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Índice de variación (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Cantidad de pasajeros (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Cantidad de puertas (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Capacidad del maletero (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Sistema dirección (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta sistema de dirección). | |

- Sistema frenos (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta sistema de frenos).
- Sistema sonido (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta sistema de sonido).
- Audi Multimedia Interface (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta MMI).
- Sistema Economizador de Combustible (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta Sistema Economizador de Combustible).
- Aire Acondicionado (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta aire acondicionado).
- Bolsa de Aire (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta bolsa de aire).
- Cargador CD (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta cargador de CD).
- TV Video (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta TV).
- Bluetooth (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta bluetooth).
- Tarjeta SD (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta tarjeta SD).
- Puerto USB (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta puerto USB).
- Escotilla (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta escotilla).
- Cámara trasera (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta cámara trasera).
- Parqueo Asistido (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta parqueo asistido).
- Alerta Aproximación (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta alerta aproximación).
- Controlador Velocidad (No obligatorio. Campo de tipo selección. Permite seleccionar si presenta controlador de velocidad).

Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de modelos.

Prototipo de interfaz:

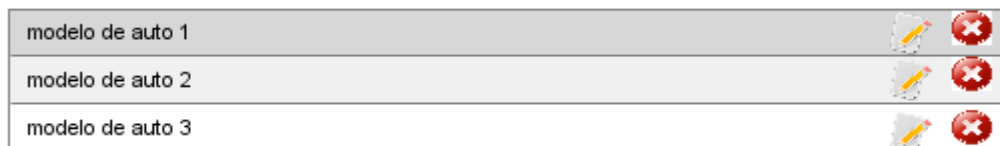


Tabla 16: HU Gestionar transmisión

| | |
|---|---|
| HU Gestionar transmisión | |
| Número: 0 | Nombre del requisito: Gestionar transmisión |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |

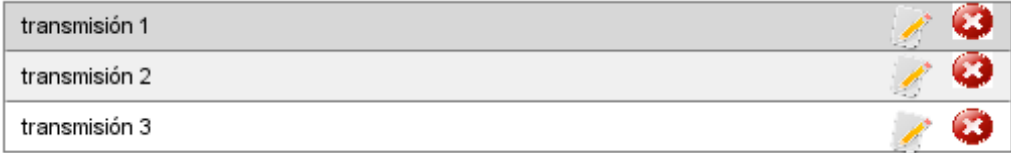
| | |
|---|-----------------------|
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una transmisión se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> Tipo (Obligatorio. Campo selector. Permite la selección de una transmisión). | |
| <p>Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de transmisiones.</p> | |
| <p>Prototipo de interfaz:</p>  | |

Tabla 17: HU Gestionar estado de motor

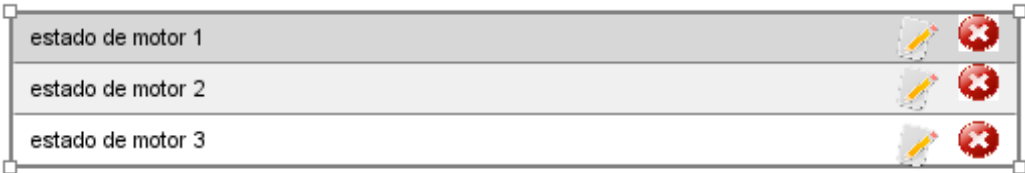
| | |
|---|---|
| HU Gestionar estado de motor | |
| Número: 0 | Nombre del requisito: Gestionar estado de motor |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un estado de motor se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). | |
| <p>Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de estados de motores.</p> | |
| <p>Prototipo de interfaz:</p>  | |

Tabla 18: HU Gestionar combustible

| |
|--------------------------|
| HU Gestionar combustible |
|--------------------------|

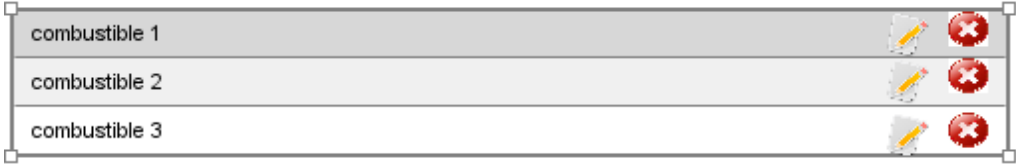
| | |
|--|---|
| Número: 0 | Nombre del requisito: Gestionar combustible |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un combustible se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). • Descripción (No obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de combustibles. | |
| Prototipo de interfaz: | |
|  | |

Tabla 19: HU Gestionar cilindrada

| | |
|--|--|
| Nombre del requisito: Gestionar cilindrada | |
| Número: 0 | Nombre del requisito: Gestionar cilindrada |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una cilindrada se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Número (Obligatorio. Campo de tipo real. Permite la entrada de caracteres numéricos y coma). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de cilindradas. | |
| Prototipo de interfaz: | |







| | | |
|--------------|---|---|
| cilindrada 1 |  |  |
| cilindrada 2 |  |  |
| cilindrada 3 |  |  |

Tabla 20: HU Gestionar motor



















| | | | | | | | | | | |
|---|---|---|---|---|---------|---|---|---------|---|---|
| | | | | | | | | | | |
| Número: 0 | Nombre del requisito: Gestionar motor | | | | | | | | | |
| Programador: Hector M. García Hernández | Iteración asignada: 1 | | | | | | | | | |
| Prioridad: Alta | Puntos estimados: 0.1 | | | | | | | | | |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 | | | | | | | | | |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un motor se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Número (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Estado (Obligatorio. Campo selector. Permite la selección de un estado de motor). • Tipo de combustible (Obligatorio. Campo selector. Permite la selección de un tipo de combustible). • Cilindrada (Obligatorio. Campo selector. Permite la selección de una cilindrada). • Auto (Obligatorio. Campo selector. Permite la selección de un auto). | | | | | | | | | | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de motores. | | | | | | | | | | |
| Prototipo de interfaz: | | | | | | | | | | |
| <table border="1"> <tr> <td>motor 1</td> <td></td> <td></td> </tr> <tr> <td>motor 2</td> <td></td> <td></td> </tr> <tr> <td>motor 3</td> <td></td> <td></td> </tr> </table> | | motor 1 |  |  | motor 2 |  |  | motor 3 |  |  |
| motor 1 |  |  | | | | | | | | |
| motor 2 |  |  | | | | | | | | |
| motor 3 |  |  | | | | | | | | |

Tabla 21: HU Gestionar batería

| | |
|---|---|
| | |
| Número: 0 | Nombre del requisito: Gestionar batería |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |

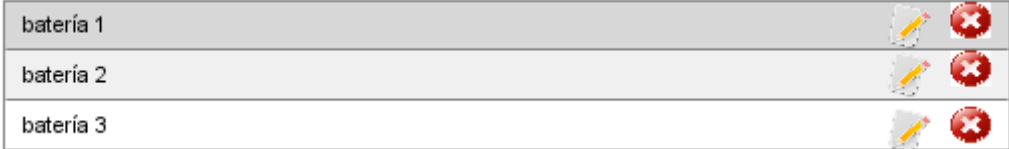
| |
|--|
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una batería se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Código (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Período (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). |
| <p>Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de baterías.</p> |
| <p>Prototipo de interfaz:</p>  |

Tabla 22: HU Gestionar marca de neumático

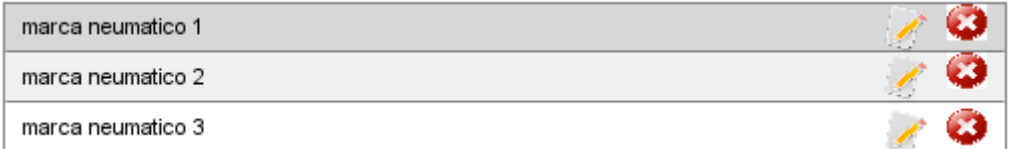
| | |
|---|--|
| HU Gestionar marca de neumático | |
| Número: 0 | Nombre del requisito: Gestionar marca de neumático |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una marca de neumático se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). | |
| <p>Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de marcas de neumáticos.</p> | |
| <p>Prototipo de interfaz:</p>  | |

Tabla 23: HU Gestionar posición de neumático

| |
|--|
| |
|--|

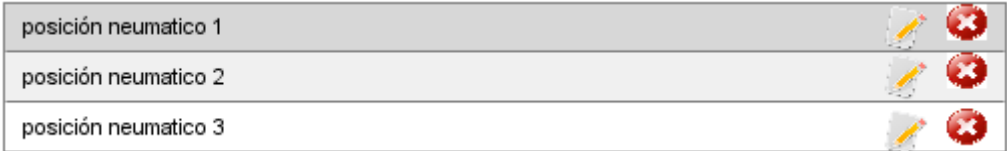
| | |
|--|---|
| Número: 0 | Nombre del requisito: Gestionar posición de neumático |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una posición de neumático se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Posición (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos y espacios). • Siglas (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos en mayúsculas). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de posición de neumáticos. | |
| Prototipo de interfaz:  | |

Tabla 24: HU Gestionar neumático

| | |
|--|---|
| HU Gestionar neumático | |
| Número: 0 | Nombre del requisito: Gestionar neumático |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un neumático se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Auto (Obligatorio. Campo selector. Permite la selección de un auto). • Marca (Obligatorio. Campo selector. Permite la selección de una marca de neumático). • Posición (Obligatorio. Campo selector. Permite la selección de una posición). • Número de registro (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Km inicial (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Rango km (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). | |

| |
|--|
| <ul style="list-style-type: none"> • Código (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Período duración (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Fecha de instalación (Obligatorio. Campo de tipo entero. Permite la selección de una fecha). |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de neumáticos. |
| Prototipo de interfaz: |
| |

Tabla 25: HU Gestionar contrato

| | |
|---|--|
| Número: 0 | |
| Número: 0 | Nombre del requisito: Gestionar contrato |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un contrato se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • No. REX (Obligatorio. Campo de tipo entero. Permite la entrada de exactamente cinco caracteres numéricos). • No. Leasing (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). • Fecha de vencimiento (Obligatorio. Campo de tipo entero. Permite la selección de una fecha). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de contratos. | |
| Prototipo de interfaz: | |
| | |

Tabla 26: HU Gestionar chapa

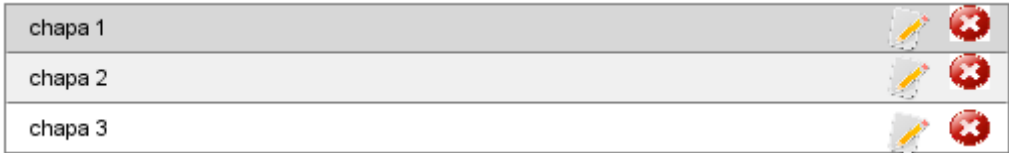
| HU Gestionar chapa | |
|---|---------------------------------------|
| Número: 0 | Nombre del requisito: Gestionar chapa |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una chapa se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Auto (Obligatorio. Campo selector. Permite la selección de un auto). • No. REX (Obligatorio. Campo de tipo entero. Permite la entrada de exactamente cinco caracteres numéricos). • Código delantero (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres alfanuméricos e igual al código trasero). • Código trasero (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres alfanuméricos e igual al código delantero). • Matrícula (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos en mayúscula). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de chapas. | |
| Prototipo de interfaz: | |
|  | |

Tabla 27: HU Gestionar estado

| HU Gestionar estado | |
|---|--|
| Número: 0 | Nombre del requisito: Gestionar estado |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un estado de un auto se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Causa (Obligatorio. Campo selector. Permite la selección de una causa). • Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). | |

Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de estados.

Prototipo de interfaz:







| | | |
|----------|---|---|
| estado 1 |  |  |
| estado 2 |  |  |
| estado 3 |  |  |

Tabla 28: HU Gestionar causa



















| | | | | | | | | | | |
|---|---|---|---|---|---------|---|---|---------|---|---|
| | | | | | | | | | | |
| Número: 0 | Nombre del requisito: Gestionar causa | | | | | | | | | |
| Programador: Hector M. García Hernández | Iteración asignada: 1 | | | | | | | | | |
| Prioridad: Alta | Puntos estimados: 0.1 | | | | | | | | | |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 | | | | | | | | | |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una causa se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> • Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). • Descripción (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). | | | | | | | | | | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de causas. | | | | | | | | | | |
| Prototipo de interfaz: | | | | | | | | | | |
| <table border="1"> <tr> <td>causa 1</td> <td></td> <td></td> </tr> <tr> <td>causa 2</td> <td></td> <td></td> </tr> <tr> <td>causa 3</td> <td></td> <td></td> </tr> </table> | | causa 1 |  |  | causa 2 |  |  | causa 3 |  |  |
| causa 1 |  |  | | | | | | | | |
| causa 2 |  |  | | | | | | | | |
| causa 3 |  |  | | | | | | | | |

Tabla 29: HU Gestionar color

| | |
|---|---------------------------------------|
| | |
| Número: 0 | Nombre del requisito: Gestionar color |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un color se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> | |

| |
|--|
| <ul style="list-style-type: none"> Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de colores. |
| Prototipo de interfaz: |
| |

Tabla 30: HU Gestionar tipo de componente

| | |
|--|--|
| HU Gestionar tipo de componente | |
| Número: 0 | Nombre del requisito: Gestionar tipo de componente |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un tipo de componente se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro. <ul style="list-style-type: none"> Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de tipos de componente. | |
| Prototipo de interfaz: | |
| | |

Tabla 31: HU Gestionar estado de oferta

| | |
|---|--|
| HU Gestionar estado de oferta | |
| Número: 0 | Nombre del requisito: Gestionar estado de oferta |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |

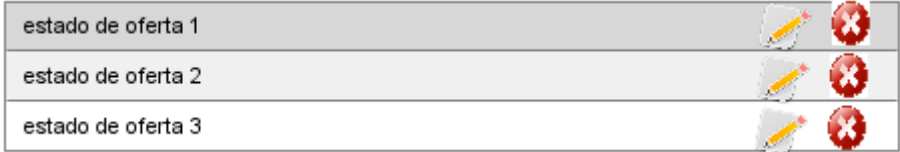
| | |
|---|--------------------|
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar un estado de oferta se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). | |
| <p>Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de estados de oferta.</p> | |
| <p>Prototipo de interfaz:</p>  | |

Tabla 32: HU Gestionar oferta

| | |
|--|--|
| | |
| Número: 0 | Nombre del requisito: Gestionar oferta |
| Programador: Hector M. García Hernández | Iteración asignada: 1 |
| Prioridad: Alta | Puntos estimados: 0.1 |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una oferta se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> Estados de oferta (Obligatorio. Campo de tipo selección. Permite la selección de un estado de oferta). Autos (Obligatorio. Campo de tipo texto. Permite la selección de un auto). No. Oferta (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Pedido (Obligatorio. Campo de tipo entero. Permite la entrada de caracteres numéricos). Fecha de emisión (Obligatorio. Campo de tipo fecha. Permite la selección de una fecha). Descripción (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfanuméricos). | |
| <p>Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de ofertas.</p> | |
| <p>Prototipo de interfaz:</p> | |

























| | | |
|----------|---|---|
| oferta 1 |  |  |
| oferta 2 |  |  |
| oferta 3 |  |  |

Tabla 33: HU Gestionar ubicación

| | | | | | | | | | | |
|---|---|---|---|---|-------------|---|---|-------------|---|---|
| | | | | | | | | | | |
| Número: 0 | Nombre del requisito: Gestionar ubicación | | | | | | | | | |
| Programador: Hector M. García Hernández | Iteración asignada: 1 | | | | | | | | | |
| Prioridad: Alta | Puntos estimados: 0.1 | | | | | | | | | |
| Riesgo en desarrollo: Alto | Puntos reales: 0.1 | | | | | | | | | |
| <p>Descripción: Permitirá insertar, modificar, eliminar y listar la flota de autos de REX. Para registrar una ubicación se muestra un formulario con una serie de datos que el usuario deberá llenar para completar el registro.</p> <ul style="list-style-type: none"> Nombre (Obligatorio. Campo de tipo texto. Permite la entrada de caracteres alfabéticos). | | | | | | | | | | |
| Observaciones: Al introducir correctamente los datos el sistema redirecciona al usuario al listado de ubicaciones. | | | | | | | | | | |
| Prototipo de interfaz: | | | | | | | | | | |
| <table border="1"> <tr> <td>ubicación 1</td> <td></td> <td></td> </tr> <tr> <td>ubicación 2</td> <td></td> <td></td> </tr> <tr> <td>ubicación 3</td> <td></td> <td></td> </tr> </table> | | ubicación 1 |  |  | ubicación 2 |  |  | ubicación 3 |  |  |
| ubicación 1 |  |  | | | | | | | | |
| ubicación 2 |  |  | | | | | | | | |
| ubicación 3 |  |  | | | | | | | | |