

**Universidad de las Ciencias Informáticas
Facultad 4**



*Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

*Sistema de Apoyo a la Toma de Decisiones del aeropuerto
Juan Gualberto Gómez*



Autores: Anadelys Rojas Jimenez

Dairon Renier de la Concepción González

Tutor: MSc. Arcadio Abad Márquez

Co-Tutores: MSc. Yuniet del Carmen Toll Palma

Ing. Manuel de Jesús Vázquez Garriga

La Habana, Cuba, 2015
"Año 57 de la Revolución"



Declaración de la auditoría

Declaración de la auditoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estime pertinente con el mismo.

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

Autor(es):

Anadelys Rojas Jimenez

Dairon R. de la Concepción González

Tutor:

MSc. Arcadio Abad Márquez



Dedicatoria

De Anadelys:

A mi primera sobrina Aidelys por ser mi fuente de inspiración.

A mi única hermana que la quiero muchísimo.

A mi abuela Oraida por estar presente en cada paso que he dado en la vida.

A mi mamá y mi papá por ser mi guía y ejemplo a seguir en todo momento, porque sin ellos no estuviera hoy donde estoy.

De Dairon:

A mi abuela por educarme desde niño y le debo todo lo que soy en la vida.

A mis padres por servirme de guías en el transcurso de la vida.

A mi único hermano por apoyarme en todo.

A mi padrino por la educación que formó en mí.

A la virgen de las mercedes por hacerme tomar las decisiones correctas y permitir a mi madre tener un hijo aunque los doctores lo creían imposible.



Agradecimientos

De Anadelys:

A mi familia que le debo todo, desde mis fracasos hasta mis triunfos. En especial a mi abuela por siempre defenderme y apoyarme en todo. A mi mamá y mi papá por el apoyo y confianza depositada en mí a lo largo de toda mi vida. A mi hermanita por crecer junto a mí, brindarme su amor y por darme una hermosa sobrina. A mi tía Ismary que me ha dado su apoyo incondicional y me ha ayudado a salir adelante ante cualquier situación. En general quiero agradecerle a toda mi familia, los que disfrutaban de la vida y los que no, porque siempre fueron fuente de inspiración para mí en cada momento, y porque sus consejos y enseñanzas siempre serán tenidos en cuenta.

A mi novio por soportar mis caprichos, por mimarme en los momentos tristes, y sacar de mí una sonrisa en las peores situaciones.

A mi familia de Matanzas Iris, Tahimy, Ilda y Nereyda. A mis suegros por ser para mí como mis padres, por el gran amor que han sabido brindarme desde que los conocí.

A Arcadio y Yuniet, nuestros tutores, por la confianza depositada, asesoramiento y ayuda.

A todos mis profesores que estuvieron presentes en mi vida de estudiante, especialmente a Mayelín, Tino, Yadiłka, Yenisleidys Piloto, Jommy y Noralvys por ser ejemplos a seguir.

A todos mis amigos, en especial Yaraís y a Dayedis por darme apoyo y no olvidarse de mí aunque estén lejos. A todos mis amigos de la universidad, donde pasamos inolvidables momentos juntos. A Héctor, Daríel Corchado, a Yiselhy, a Andy, a Noralis, Roberto, Ocleidys, Adrián Arjona y a Lizandra, por poder contar con ellos en cualquier momento. A mis compañeras del apartamento 139. Nunca los olvidaré.

A todos los que me dieron su apoyo, muchas gracias.



Agradecimientos

De Dairon:

En primer lugar a dios por permitirme estar en este mundo.

A mi familia por ayudarme a crecer en la vida del lado del bien, en especial mis abuelas, mis padres y hermano.

A mi novia por estar conmigo en los buenos y malos momentos.

A mi nueva familia pinareña.

A Yayi, Ayami, Papu y Silvita por cuidarme desde pequeño.

A Iris y Tahimí por cuidarme la familia.

A mis tíos y primos que si los pongo a todos no acabo nunca.

A mis vecinos Orlando, Popi, Yamaisi y especial a Nené Peñate que fue como el abuelo que nunca tuve.

A mis mejores amigos de la infancia Dani, Luis Orlando y el Nené Geografía.

A los profesores durante toda mi carrera por inculcarme buenos valores y conocimientos en especial a Irma y Damián que son como de la familia.

A las personas que me han hecho ver mis errores, decirme las verdades, y ayudarme a enmendarlos como Noralbis de Armas, Anisley Saez, Jommy Barban, Yenisleidy Piloto y Yadiłka Suárez-Inclán.

A los amigos de estudio de la universidad Yurian, Arturo, David, Corchado, Arjona, Héctor, Roberto y en especial el hermanado de la logia Andy.

A mis tutores todo el apoyo brindado durante el desarrollo de la tesis.

En fin a todas aquellas personas que han dejado una parte de ellos en mi corazón gracias.



Resumen

El aeropuerto Juan Gualberto Gómez necesita contar con un Sistema de Apoyo a la Toma de Decisiones capaz de obtener información relacionada con los vuelos de una manera exacta, actualizada y más rápida; para garantizar una óptima planificación de áreas, recursos y servicios en el aeropuerto. Es por ello que se decide desarrollar un sistema informático utilizando técnicas de inteligencia artificial, para contribuir al proceso de toma de decisiones respecto al estado operativo del aeropuerto. Este trabajo permite obtener una aplicación web usando tecnologías libres. Para lograr este objetivo se estudiaron las principales herramientas y tecnologías propicias para la construcción de la aplicación así como las metodologías que actualmente han tomado auge en el desarrollo de software. El desarrollo del sistema fue guiado por la metodología Programación Extrema (XP). Como lenguajes de programación del lado del cliente se utilizaron HTML, JavaScript a través de la librería JQuery y CSS apoyado en el framework de diseño Bootstrap. Como lenguaje del lado del servidor se utilizó PHP, a través del framework de desarrollo Symfony 2. Se utilizó el NetBeans IDE como Entorno de Desarrollo Integrado, Apache como servidor web y PostgreSQL como gestor de bases de datos. Obteniendo como resultado una aplicación web que mediante el análisis de la información, contribuye al apoyo a la toma de decisiones respecto al estado operativo del aeropuerto Juan Gualberto Gómez cumpliendo con las expectativas de los directivos del aeropuerto.

Palabras claves: aeropuerto, inteligencia artificial, Sistema de Apoyo a la Toma de Decisiones



Índice general

Introducción	1
Capítulo I: Fundamentación Teórica	6
1.1 Introducción	6
1.2 Toma de Decisiones	6
1.3 Sistema de Apoyo a la Toma de Decisiones	7
1.4 Inteligencia Artificial	8
1.4.1 Minería de Datos	9
1.5 Soluciones similares	12
1.5.1 Soluciones similares en el ámbito internacional	12
1.5.2 Soluciones similares en el ámbito nacional	13
1.5.3 Valoración de los sistemas analizados	14
1.5.4 Conclusiones del estudio de las soluciones similares	15
1.6 Metodología de Desarrollo	15
1.6.1 Scrum	16
1.6.2 Proceso Unificado Ágil	17
1.6.3 Programación Extrema	17
1.6.4 Fundamentación de la metodología a utilizar	18
1.7 Lenguajes de programación	18
1.7.1 Preprocesador de hipertexto (PHP 5.3)	18
1.7.2 Lenguaje de marcas de hipertexto (HTML 5)	20
1.7.3 Hojas de estilo en cascada (CSS 3)	20
1.7.4 JavaScript 1.8	20
1.8 Librería para exportar a pdf	21
1.9 Framework PHP	21
1.9.1 Zend framework	22
1.9.2 Yii framework	22
1.9.3 Symfony2	22
1.9.4 Fundamentación del marco de trabajo a utilizar	23



1.10	Frameworks de CSS. Bootstrap.....	23
1.11	Framework JavaScript. JQuery	23
1.12	Sistemas gestores de bases de datos	24
1.12.1	MySQL	24
1.12.2	PostgreSQL.....	25
1.12.3	Fundamentación del Sistema Gestor de Bases de Datos (SGBD) a utilizar.....	25
1.13	Servidores web	25
1.13.1	Apache	26
1.13.2	Nginx	26
1.13.3	Internet Information Server (Microsoft IIS)	26
1.13.4	Selección del servidor web a utilizar	27
1.14	Entorno de Desarrollo Integrado (IDE).....	27
1.14.1	PHPStorm	27
1.14.2	Eclipse.....	28
1.14.3	NetBeans.....	28
1.14.4	Selección del IDE a utilizar	28
1.15	Herramientas CASE	29
1.15.1	ERwin.....	29
1.15.2	Visual Paradigm for UML.....	29
1.15.3	Fundamentación de la herramienta CASE a utilizar	30
1.16	Herramientas utilizadas en el proceso de Minería de Datos	30
1.16.1	KNIME	30
1.16.2	RapidMiner	30
1.16.3	Weka	31
1.16.4	Fundamentación de la herramienta de Minería de Datos y algoritmo a utilizar	31
1.17	Conclusiones del Capítulo	32
Capítulo II: Exploración y Planificación		34
2.1	Introducción.....	34
2.2	Propuesta de solución.....	34
2.3	Modelo Conceptual	35



2.3.1	Descripción de los elementos del modelo conceptual	36
2.4	Personal relacionado con el sistema	37
2.5	Exploración	37
2.5.1	Funcionalidades a implementar	37
2.5.2	Historias de Usuarios (HU)	38
2.5.3	Requerimientos generales del sistema	39
2.6	Planificación	40
2.6.1	Estimación de esfuerzos por HU	40
2.6.2	Plan de Iteraciones	41
2.6.3	Plan de entregas	41
2.7	Conclusiones del capítulo	42
Capítulo III: Implementación y Prueba		43
3.1	Introducción	43
3.2	Patrón de Arquitectura	43
3.3	Patrones de diseños	44
3.3.1	Patrones GRASP	45
3.3.2	Patrones GOF	46
3.4	Tarjetas Clase-Responsabilidades-Colaboradores (CRC)	47
3.5	Modelo de datos	48
3.6	Estilos y estándares de codificación	50
3.7	Tratamiento de errores	51
3.8	Tareas de ingeniería	51
3.9	Integración SATD con Weka	52
3.10	Pruebas	53
3.10.1	Pruebas unitarias	53
3.10.2	Pruebas de aceptación	54
3.10.3	Resultados de las pruebas de aceptación	56
3.10.4	Evaluación del proceso de clasificación	58
3.11	Conclusiones del capítulo	59
Conclusiones generales		60



Índice General

Recomendaciones.....	61
Referencias Bibliográficas.....	62
Anexos	69



Índice de Figuras

Figura 1: Clasificación de las técnicas de minería de datos	11
Figura 2: Propuesta de solución	35
Figura 3: Diagrama conceptual	36
Figura 4: Representación del patrón arquitectónico MVC en la propuesta de solución	44
Figura 6: Modelo físico de la base de datos	49
Figura 7: Pruebas unitarias con PHPUnit (error)	53
Figura 8: Pruebas unitarias con PHPUnit (correcto).....	54
Figura 9: Resultado de las pruebas de aceptación por iteración	57
Figura 10: Resultados de las pruebas dirigidas al sistema en su totalidad	58
Figura 11: Evaluación del proceso de Clasificación y del Pronóstico realizado por el algoritmo “J48”	59
Figura 12: Estadísticas del uso de servidores web según W3Techs en el año 2015	70
Figura 13: Algoritmo DecisionStump.....	71
Figura 14: Algoritmo HoeffdingTree	71
Figura 15: Algoritmo J48.....	72
Figura 16: Algoritmo REPTree	72
Figura 17: Tabla sabana	78
Figura 18: Tabla usuarioCorreo.....	78
Figura 19: Tabla clasificacion	78
Figura 20: Pantalla principal	94
Figura 21: Reportes diarios, su clasificación, guardar pdf	95
Figura 22: Autenticar administrador	96
Figura 23: Sección en donde se gestionan los destinatarios de correo.....	96



Índice de tablas

Tabla 1: Comparación de los sistemas analizados	14
Tabla 2: Algoritmos de clasificación en Weka	32
Tabla 3: Funcionalidades a implementar	37
Tabla 4: HU Reportes diarios	39
Tabla 5: Puntos de estimación por historias de usuario	40
Tabla 6: Planificación de las iteraciones	41
Tabla 7: Plan de entregas	42
Tabla 8: CRC 1	48
Tabla 9: TI Vuelos por horas	51
Tabla 10: TI Pasajeros por horas	52
Tabla 11: CP Vuelos por hora	54
Tabla 12: CP Pasajeros por hora	55
Tabla 13: Resultado de las pruebas de aceptación por iteración	57
Tabla 14: HU Clasificación del día	73
Tabla 15: HU Reportes semanales	73
Tabla 16: HU Reportes mensuales	74
Tabla 17: HU Reportes anuales	74
Tabla 18: HU Autenticar administrador	74
Tabla 19: HU Gestionar correos	75
Tabla 20: HU Guardar como pdf	75
Tabla 21: HU Servicios web	76
Tabla 22: HU Diseño adaptativo	76
Tabla 23: CRC 2	76
Tabla 24: CRC 3	77
Tabla 25: CRC 4	77
Tabla 26: CRC 5	77
Tabla 27: TI Clasificación del día	79
Tabla 28: TI Vuelos por días	79



Índice de tablas

Tabla 29: TI Pasajeros por día	79
Tabla 30: TI Vuelos por mes	80
Tabla 31: TI Pasajeros por mes	80
Tabla 32: TI Vuelos por año	80
Tabla 33: TI Pasajeros por año	81
Tabla 34: TI Autenticar administrador	81
Tabla 35: TI Insertar destinatario	81
Tabla 36: TI Listar destinatarios	82
Tabla 37: TI Eliminar destinatario	82
Tabla 38: TI Editar destinatario	82
Tabla 39: TI Guardar como pdf	83
Tabla 40: TI Servicios web	83
Tabla 41: TI Diseño adaptativo	83
Tabla 42: CP Vuelos por días	84
Tabla 43: CP Pasajeros por día	85
Tabla 44: CP Vuelos por mes	85
Tabla 45: CP Pasajeros por mes	86
Tabla 46: CP Vuelos por año	87
Tabla 47: CP Pasajeros por año	88
Tabla 48: CP Autenticar administrador	89
Tabla 49: CP Insertar destinatario	89
Tabla 50: CP Listar destinatario	90
Tabla 51: CP Ver destinatario	90
Tabla 52: CP Editar destinatario	91
Tabla 53: CP Eliminar destinatario	92
Tabla 54: CP Verificar envío de correo	92
Tabla 55: CP Guardar como pdf	93
Tabla 56: CP Servicios web	93
Tabla 57: CP Diseño adaptativo	94



Introducción

La información ha sido un elemento fundamental para la toma de decisiones de cualquier naturaleza. En la actualidad este término ha adquirido relevancia social debido a que las tecnologías de la información han evolucionado e impactado de tal manera que esta época es conocida como la era digital. Actualmente las empresas ven la necesidad de utilizar las Tecnologías de la Información y las Comunicaciones (TIC) para informatizar los procesos operativos, implantar ventajas competitivas, analizar grandes volúmenes de datos, extraer conocimiento y suministrar una plataforma de información necesaria para la toma de decisiones, entre otras ventajas.

Uno de los factores que comúnmente ha sido clave para toda empresa, es la toma de decisiones gerenciales, basado en una adecuada información administrativa. A diario las organizaciones buscan implementar diversas herramientas o estrategias que les permitan apoyarse para tomar decisiones y poder lograr el cumplimiento de sus objetivos. Franklin y Krieger plantean al respecto: *“No existe un método único para tomar decisiones que todas las empresas puedan emplear en todas las situaciones. El individuo que toma una decisión debe definir con exactitud el problema en cuestión, para después generar soluciones alternativas y evaluarlas y, por último, tomar la decisión.”* (1)

Por el gran cúmulo de datos a analizar y la gran cantidad de personal asociado; entidades como los aeropuertos son zonas críticas en cuanto a la toma de decisiones y necesitan de una estrategia para optimizar la atención al cliente en los mostradores, los servicios de migración y aduanas, combustible de aeronaves, y el manejo rápido y seguro de equipajes, entre otros servicios.

Cuba es una isla situada en el mar Caribe, zona mundial que atrae mucho turismo. Está enlazada directamente por vía aérea con 33 países de América, Europa y Asia a través de 51 compañías aéreas. Más de 50 ciudades del mundo se conectan directamente con destinos turísticos cubanos, aterrizando aviones de carga y de pasajeros por los 12 aeropuertos internacionales que existen en la isla. (2)

Estos aeropuertos tienen diseñada su propia estrategia para tomar decisiones, con el objetivo de acoger y acomodar de la mejor manera posible, la cantidad creciente de pasajeros y aviones. El aeropuerto Juan Gualberto Gómez, también conocido como aeropuerto de Varadero, es un aeropuerto internacional,



situado en la provincia de Matanzas. Es el segundo aeropuerto más importante del país, en él se efectúan vuelos de arribos y despegues de muchos países, tanto de pasajeros como de carga. Una vez que los pasajeros aterrizan en el aeropuerto, se dirigen al área equipaje y posteriormente al área de Aduana¹. En esta entidad se brindan servicios gastronómicos y se venden diferentes productos para el consumo de los clientes. El aeropuerto es encargado de darle mantenimiento a los aviones y reabastecer su combustible para su próximo despegue.

Los directivos para dar seguimiento a las operaciones de planeación, organización, dirección y control, se auxilian de una aplicación informática llamada Sistema de Operaciones. Este cuenta con una base de datos general de la cual también se nutre el aeropuerto José Martí de La Habana. Permite a los trabajadores de diferentes áreas en el aeropuerto acceder a la información de los vuelos diarios, del mes y por temporadas. Además insertar en el sistema los datos reales de los vuelos para su posterior comparación con los datos planificados.

Los directivos a partir de la información obtenida por el sistema, realizan el proceso de análisis de datos del comportamiento del día, para tomar decisiones garantizando una óptima planificación de áreas, recursos y servicios en el aeropuerto. El análisis de datos se realiza al final de cada semana, para obtener un pronóstico del comportamiento del aeropuerto de los próximos siete días. Todos estos datos son usados para generar gráficos mediante la herramienta Microsoft Excel². Este proceso se hace engorroso puesto que los directivos tienen que introducir los datos de forma manual, lo cual constituye una pérdida de tiempo y puede existir algún error a la hora de introducir los datos. Además no se pueden generar reportes sobre períodos similares, para comparar y apoyar con esto posibles pronósticos. La información al ser generada en Excel se guarda en distintos archivos y carpetas en formato digital, generando gran volumen de documentación, haciendo compleja para los directivos cualquier búsqueda o consulta de datos para su estudio. Estos archivos se pueden perder por errores humanos, defecto mecánico o la información puede ser atacada por virus informáticos. Por lo que es necesario convertir la información

¹ Son oficinas públicas cuya finalidad primordial es la de controlar el paso de mercancías nacionales o extranjeras que proceden del exterior, van hacia otros territorios aduaneros o circulan entre diversos puntos de un mismo ámbito geográfico, también se encarga de la liquidación de los impuestos establecidos por las importaciones y exportaciones de mercancías. (3)

² Software que permite crear gráficas, tablas, además calcular y analizar datos.



existente en experiencia, conocimiento y sabiduría, a través de un análisis e interpretación para posibilitar una buena toma de decisiones.

La presente investigación surge ante la necesidad de dar solución a la situación problemática antes expuesta; por lo que el **problema a resolver** es el siguiente: ¿Cómo presentar la información a los directivos del aeropuerto, para la ayuda en la toma de decisiones con respecto al estado operativo del aeropuerto Juan Gualberto Gómez?

Por lo que el **objeto de estudio** del trabajo son los Sistemas de Apoyo a la Toma de Decisiones, enmarcado en el **campo de acción**: Sistema de Apoyo a la Toma de Decisiones aplicado al aeropuerto Juan Gualberto Gómez.

En correspondencia con el problema planteado se propone como **objetivo general**: Desarrollar un sistema informático utilizando inteligencia artificial, que permita la presentación de la información a los directivos del aeropuerto, para la ayuda en la toma de decisiones con respecto al estado operativo del aeropuerto Juan Gualberto Gómez.

Para alcanzar el objetivo previamente señalado, se identificó una serie de **objetivos específicos**:

- ✓ Elaborar el marco teórico partiendo del estudio de los Sistemas de Apoyo a la Toma de Decisiones a nivel mundial, y de las herramientas, tecnologías, y metodología de desarrollo para la elaboración de la aplicación.
- ✓ Diseñar una solución que contribuya al apoyo de la toma de decisiones en el aeropuerto Juan Gualberto Gómez.
- ✓ Implementar la propuesta de solución haciendo uso de las tecnologías y herramientas seleccionadas.
- ✓ Realizar las pruebas correspondientes al sistema para comprobar que los resultados obtenidos estén acorde a las exigencias del cliente.

Se plantea como **hipótesis**: el desarrollo de un sistema informático utilizando inteligencia artificial permitirá la presentación de la información a los directivos del aeropuerto, para la ayuda en la toma de decisiones con respecto al estado operativo del aeropuerto Juan Gualberto Gómez.



Como **tareas investigativas** a desarrollar se encuentran:

- ✓ Entrevistas a los directivos sobre el proceso de toma de decisiones respecto al estado operativo del aeropuerto.
- ✓ Búsqueda bibliográfica de los Sistemas de Apoyo a la Toma de Decisiones que existen en el mundo.
- ✓ Selección de las tecnologías, metodología y herramientas a utilizar para el desarrollo del sistema informático.
- ✓ Realización del modelo conceptual para un mejor entendimiento de la propuesta de solución a desarrollar.
- ✓ Definición de las Historias de Usuario (HU) en conjunto con el cliente.
- ✓ Descripción de los patrones de diseño y arquitectura utilizados en la propuesta de solución.
- ✓ Elaboración de las tarjetas Clase-Responsabilidades-Colaboradores (CRC) para identificar las clases que se verán inmersas en la implementación con sus respectivas funcionalidades y sus relaciones con las demás.
- ✓ Especificación de tareas de ingeniería.
- ✓ Implementación de las tareas de ingeniería especificadas para dar solución a las HU planteadas.
- ✓ Descripción de los casos de pruebas que permitan realizar las pruebas de aceptación a la solución obtenida.
- ✓ Verificación del correcto funcionamiento del Sistema de Apoyo a la Toma de Decisiones implementado a través de las pruebas de aceptación.
- ✓ Realización de pruebas unitarias al Sistema de Apoyo a la Toma de Decisiones implementado para garantizar el correcto funcionamiento del código.

Para dar cumplimiento a las tareas se utilizaron los siguientes **métodos científicos de investigación**.

Dentro de los **métodos empíricos** se utiliza la **entrevista** aplicada al coordinador general del aeropuerto Juan Gualberto Gómez (Ver anexo 1), para adquirir conocimiento sobre el proceso de toma de decisiones respecto al estado operativo que se lleva a cabo en el aeropuerto, además para determinar los requerimientos con que el sistema debe contar para la solución de dicha investigación. Se utiliza la **observación** (Ver anexo 2 y anexo 3) planificadas y dirigidas con el fin de realizar la fundamentación



teórica del problema para conocer la esencia de la problemática planteada, posibilitando obtener una mejor perspectiva acerca del funcionamiento y organización del aeropuerto Juan Gualberto Gómez.

Dentro de los **métodos teóricos** se tuvo en cuenta el método **analítico sintético** en el estudio de diferentes fuentes bibliográficas, para extraer los elementos más importantes y los conceptos relevantes relacionados con los Sistemas de Apoyo a la Toma de Decisiones, con el objetivo de arribar a conclusiones que sustenten la necesidad de la investigación. Además de definir las tecnologías y herramientas más adecuadas para el desarrollo de la propuesta de solución.

Otro método utilizado fue el **histórico lógico** para realizar un estudio de la evolución de Sistemas de Apoyo a la Toma de Decisiones.

La investigación está desarrollada en **tres capítulos**, cuya estructura se describe a continuación:

Capítulo I. Fundamentación teórica: En este capítulo se presenta el marco teórico en el que se abordan los conceptos relacionados al proceso de toma de decisiones, los Sistemas de Apoyo a la Toma de Decisiones y a la inteligencia artificial. Se hace un estudio valorativo realizado sobre sistemas internacionales y nacionales que tienen relación con el problema planteado, lo que permite detectar funcionalidades a tener en cuenta en la propuesta de solución. Se hace referencia a la metodología seleccionada y a las herramientas, tecnologías y lenguajes de programación para implementar el sistema.

Capítulo II. Exploración y Planificación: En este capítulo se generan diferentes artefactos que propone la metodología seleccionada como las historias de usuarios, la estimación del esfuerzo para implementar cada historia de usuario, el plan de iteraciones y el plan de entregas del producto.

Capítulo III. Implementación y Pruebas: En este capítulo se realiza la implementación del sistema haciendo uso de la metodología, los lenguajes y herramientas seleccionadas. Se define la arquitectura del sistema y los patrones de diseño que se utilizan, se realiza el modelo físico de la base de datos, se presentan las tarjetas Clase-Responsabilidades-Colaboradores (CRC). También se muestran las tareas de la ingeniería asociadas a las diferentes historias de usuario. Por último se prueba la herramienta para garantizar la calidad que el cliente espera.



Capítulo I: Fundamentación Teórica

Capítulo I: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se pretende abordar los aspectos más importantes relacionados con los sistemas que se emplean en el apoyo a la toma de decisiones en empresas. Se tratan los principales conceptos de toma de decisiones, Sistema de Apoyo a la Toma de Decisiones e inteligencia artificial. Además un estado del arte del tema tratado a nivel nacional e internacional. Se describe la metodología a seguir para el desarrollo del Sistema de Apoyo a la Toma de Decisiones y las herramientas y tecnologías a usar para su implementación.

1.2 Toma de Decisiones

En la vida diaria ante cualquier situación se toman decisiones, muchas veces sin la necesidad de analizar a fondo para obtener una respuesta, sin darse cuenta que hay momentos en los que depende de una simple decisión el que un escenario se transforme. Muchas veces se toma una decisión porque se cree que es la mejor, guiándose por la intuición, por la experiencia, o porque resultó siendo buena en otro caso.

El diccionario de la Real Academia de la Lengua Española define decisión como: *“determinación, resolución que se toma o se da en una cosa dudosa”*. (4)

Carlo Vercellis define decisión como *“una opción desde múltiples alternativas, normalmente hecha con un grado justo de racionalidad”*. (5)

Herbert Simon, el teórico más reconocido en el tema, define en el año 1960 la toma de decisiones como *“(…) proceso de selección entre cursos alternativos de acción, basado en un conjunto de criterios, para alcanzar uno o más objetivos”* y el proceso de toma de decisiones empresariales como un proceso grupal: *“un esfuerzo planeado y cooperativo, en el cual cada participante tiene un papel reconocido que cumplir y unas obligaciones o tareas por ejecutar. Se le asignan a estas obligaciones para lograr el objetivo de la organización más que para satisfacer preferencias individuales, aunque frecuentemente coinciden ambos efectos”*. (6)



Capítulo I: Fundamentación Teórica

Según Schein *“La toma de decisiones es el proceso de identificación de un problema u oportunidad y la selección de una alternativa de acción entre varias existentes, es una actividad diligente clave en todo tipo de organización”*. (7)

A partir de los conceptos antes mencionados, se puede decir que cuando el estado real de las cosas difiere del estado deseado surge un problema. Resolver un problema conduce a un proceso de identificación y selección de la acción o acciones adecuadas para su solución. A este proceso se le denomina toma de decisiones, donde se selecciona la alternativa de mayor relevancia que permita el logro de los resultados deseados.

Hoy en día las empresas manejan una extensa cantidad de información, muchas de ellas cuentan con altas tecnologías y sin embargo no son capaces de explotarlas al máximo, siendo un experto humano quien tome la alternativa que estime correcta ante una situación presentada. Desde hace mucho tiempo, se vienen desarrollando sistemas capaces de almacenar, transformar, analizar y visualizar dicha información con el fin de apoyar las actividades de la empresa o negocio para así poder llegar a una oportuna y correcta toma de decisiones, estos se conocen como Sistemas de Apoyo a la Toma de Decisiones.

1.3 Sistema de Apoyo a la Toma de Decisiones

Los Sistemas de Apoyo a la Toma de Decisiones (DSS³) son sistemas informáticos utilizados por diferentes empresas para la ayuda en la toma de decisiones. Permiten que personas de diferentes niveles de una empresa puedan recolectar datos necesarios y procesarlos de formas diferentes simulando varias situaciones reales y así poder elegir la mejor opción tomando la decisión más correcta. (8)

Un Sistema de Apoyo a las Decisiones está diseñado específicamente para apoyar a los usuarios en la resolución de problemas y la toma de decisiones mediante la compilación de información de diferentes fuentes que contienen gran cantidad de datos.(8)

Según Daniel Power un “Sistema de Soporte a Decisiones es un sistema basado en un software interactivo destinado a ayudar a los tomadores de decisiones, donde compila, analiza y manipula la

³ Por sus siglas en inglés Decision Support System.



Capítulo I: *Fundamentación Teórica*

información a partir de una gran cantidad de datos, documentos, marcos de conocimiento, para identificar, resolver problemas y tomar decisiones”.(9)

Raymond McLeod afirma que *“un DSS es un sistema productor de información enfocado hacia un problema específico que un gerente debe resolver y hacia las decisiones que el gerente debe tomar”*.(10)

Existen otras definiciones de DSS, pero de manera general los DSS son sistemas de información basados en computadoras los cuales combinan modelos y datos para intentar resolver problemas utilizando una interfaz amigable para dar soporte a la toma de decisiones.

1.4 Inteligencia Artificial

El mundo actual se desarrolla en aras de hacer cada vez más fácil, eficiente y sofisticado el trabajo que realiza el hombre. Por tanto existen áreas capaces de hacerlo realidad como es la de la Inteligencia Artificial.

El término Inteligencia Artificial (IA) es actualmente muy conocido, aunque queda mucho terreno por recorrer en este sentido y mucho más por investigar. Se puede garantizar que se han dado pasos sólidos en el desarrollo de esta rama de la ciencia de la computación.

En la actualidad, la IA es una rama de la ciencia que está teniendo un crecimiento acelerado en cuanto a su aplicación en el desarrollo científico-técnico. Se está aplicando a numerosas actividades realizadas por los seres humanos con la finalidad de lograr una mayor similitud a estos (11).

El diccionario de la Real Academia de la Lengua Española define inteligencia como *“capacidad de entender o comprender, de resolver problemas”*, y artificial como *“desarrollo y utilización de ordenadores con los que se intenta reproducir los procesos de la inteligencia humana”*.(12)

Se define IA a *“la ciencia que enfoca su estudio a lograr la comprensión de entidades inteligentes (13).”* Es el estudio de la forma en que se transforma el pensamiento humano en un ente artificial; dándole “vida” a un programa de cómputo mediante el desarrollo de software y hardware⁴ que imitan métodos de razonamiento análogos a los seres humanos.

⁴ Conjunto de los componentes que integran la parte material de una computadora. (14)



Capítulo I: Fundamentación Teórica

Según John McCarthy, quien acuñó el término de IA, *“es la ciencia e ingeniería de hacer máquinas inteligentes, especialmente programas inteligentes de cómputo (15).”*

Alberto García Serrano afirma *“(...) no puede verse la IA como la receta mágica capaz de resolver cualquier problema, sino como un conjunto de técnicas que por sí solas o en combinación con otras nos ayudarán a encontrar una solución (no necesariamente la mejor) a un problema cuya resolución es compleja e incluso inabordable por una persona humana”. (16)*

Muchas personas que estudian el tema definen a su manera lo que es la IA, pero muchos coinciden que es capaz de convertir a las computadoras en “entes inteligentes” con la creación de software permitiendo imitar algunas de las funciones principales del cerebro humano con el objetivo de resolver determinados problemas.

En la actualidad la IA se aplica en numerosas actividades realizadas por el ser humano, para hacer más cómodo y efectivo el trabajo, entre las que se destacan las siguientes líneas de investigación científica: la robótica, la visión artificial, técnicas de aprendizaje y la gestión del conocimiento. Para hacer posible la realización de estas actividades es necesaria una técnica computacional, la Minería de Datos.

1.4.1 Minería de Datos

Tener información y conocimiento es muy valioso hoy en día. A veces se dispone de la información suficiente y no se sabe cómo analizar tal cantidad de información, e identificar aquella que permita tomar decisiones acertadas en el menor tiempo posible, haciendo necesario aplicar algún tipo de sistema de análisis. Para esto se han desarrollado diferentes herramientas, una especial es la Minería de Datos (MD) o por su nombre en inglés Data Mining.

“La Minería de Datos o Data Mining puede definirse como una extracción de información desconocida no trivial y potencialmente útil de una gran cantidad de información.” Siendo el objetivo general del proceso de minería de datos extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior. (17)

“La minería de datos puede definirse inicialmente como un proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias al examinar grandes cantidades de datos.” (18)



Capítulo I: Fundamentación Teórica

De forma general se puede resumir, teniendo en cuenta conceptos definidos por diferentes autores, que la tarea fundamental de la minería de datos es descubrir conocimiento a partir de grandes volúmenes de datos, de tal manera que su uso ayude a tomar decisiones más seguras que reporten algún tipo de beneficio a las organizaciones.

Hoy en día la base de datos es una herramienta comúnmente utilizada para el almacenamiento de datos. La mayoría de las veces tienden a almacenar una gran cantidad de datos, que al ser humano se le hace engorroso el poder reducirlos y analizarlos sin el uso de técnicas de análisis automatizadas. Existe un proceso para la extracción de información, llamado KDD (Knowledge Discovery in Databases por sus siglas en inglés o Descubrimiento de Conocimiento en Bases de Datos en español).

KDD puede definirse como: *“la integración de un conjunto de áreas que tienen como propósito la identificación de un conocimiento obtenido a partir de las bases de datos que aporten información hacia la toma de decisiones”* (17), además es el proceso completo que incluye pre-procesamiento, minería y post-procesamiento de los datos.

La minería de datos constituye una fase dentro del proceso de extracción del conocimiento (KDD) utilizando técnicas predictivas y técnicas descriptivas (Ver Figura 1), incluyendo algoritmos capaces de obtener relaciones entre distintos atributos o conceptos para ayudar a la toma de decisiones. (18) (19)



Capítulo I: Fundamentación Teórica

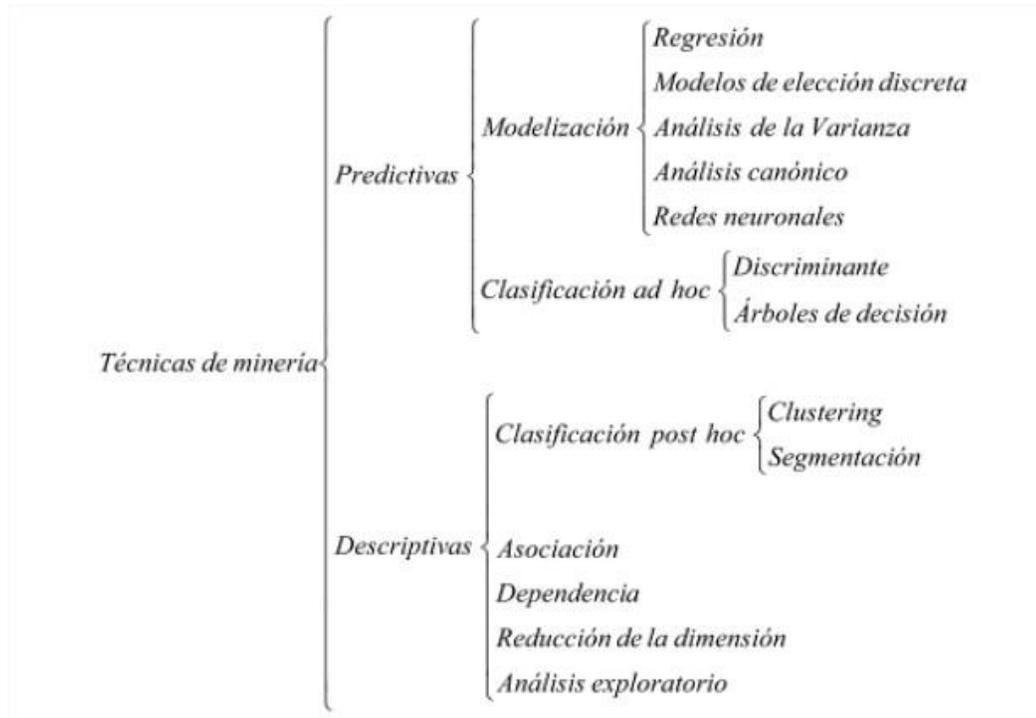


Figura 1: Clasificación de las técnicas de minería de datos (18)

Muchas de estas técnicas de minería de datos han sido utilizadas desde hace varios años para el procesamiento de datos, convirtiéndolos en conocimiento útil para el apoyo en la toma de decisiones en diferentes empresas. Se decidió utilizar los árboles de decisión en la presente investigación para apoyar la toma de decisiones en el aeropuerto Juan Gualberto Gómez.

Árboles de Decisión

Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas, ayudando a tomar la decisión más acertada. Estos se caracterizan por la sencillez de su representación y de su forma de actuar, además de la fácil interpretación dado a que pueden ser expresados en forma de reglas de decisión. (20)

Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar. Está



Capítulo I: Fundamentación Teórica

compuesto por nodos que contienen los atributos; arcos o ramas, los valores posibles del nodo padre, es decir, los distintos caminos que se pueden emprender cuando se toma una decisión o bien ocurre algún evento aleatorio; y las hojas valores del atributo clase. (20)

Los árboles de decisión, constituyen una de las principales técnicas de minería de datos. Son fáciles de usar, tolerantes al ruido, a atributos no significativos y a valores faltantes. Permiten además reducir la cantidad de variables para realizar el análisis. Representan una gran ventaja respecto a las demás técnicas de clasificación, ya que permiten representar el conocimiento extraído en un conjunto de reglas. (21)

1.5 Soluciones similares

El estudio de soluciones similares a la que se pretende desarrollar, es un paso fundamental durante el proceso de investigación. A continuación se realiza un estudio de diferentes Sistemas de Apoyo a la Toma de Decisiones que utilizan diferentes empresas hoy en día.

1.5.1 Soluciones similares en el ámbito internacional

En la actualidad el uso de los DSS se ha extendido debido a su capacidad de analizar grandes volúmenes de datos y a la forma que tienen de presentar en resumen esta información. A continuación se presentan algunos de los programas y software empleados para ayudar a directivos en empresas e instituciones a realizar el proceso de toma de decisiones.

AAIMS⁵: Sistema de Gestión de la Información Analítica es un exitoso sistema estadounidense de apoyo a las decisiones desarrollado por American Airlines. Fue tan buena la aplicación que otras aerolíneas, fabricantes de aeronaves, analistas financieros de aerolíneas, consultores y asociaciones lo aplican. El AAIMS respalda las decisiones sobre aerolíneas a través del análisis de los datos que se recopilan sobre estadísticas de uso de aeronaves en las aerolíneas, la capacidad y uso de los asientos y sobre el tráfico aéreo. Además genera pronósticos sobre participación de mercado, ingresos y rentabilidad de las aerolíneas, a su vez ayuda a la gerencia a tomar decisiones sobre asignación de aeronaves, solicitudes de ruta, clasificación de pasajes y fijación de precios. El ambiente de trabajo es de escritorio y es una herramienta privativa. (22)

⁵ Por sus siglas en inglés American Analytical Information Management System.



Capítulo I: Fundamentación Teórica

Cognos: software canadiense para Inteligencia de Negocio y Administración del Desempeño Financiero, es utilizado por empresas y organizaciones líderes a nivel mundial. Convierte los datos en vistas del pasado, el presente y el futuro de las operaciones y el rendimiento de su organización de modo que los responsables de tomar decisiones, puedan aprovechar las oportunidades y minimizar los riesgos. Proporciona opciones de consulta, creación de informes en formas de gráficos, análisis, paneles de control y cuadros de mando con planificación, modelado de escenarios y analíticas predictivas, entre otras funcionalidades. Da soporte a la manera de pensar y trabajar de las personas al intentar comprender el rendimiento empresarial y tomar decisiones con herramientas para interactuar, buscar y ensamblar todas las perspectivas del negocio. Además ofrece gran seguridad y confiabilidad. El ambiente de trabajo web y escritorio, es una herramienta privativa. (23)

MicroStrategy: esta plataforma norteamericana ofrece al usuario mayor velocidad, rendimiento, sencillez y eficiencia, todo esto enfocado a la toma de decisiones más efectiva. Permite crear informes en forma de gráficos y ofrece una plataforma unificada para el análisis, reporte y monitorización siendo este el núcleo de cualquier sistema de soporte a decisiones. El ambiente de trabajo es web, escritorio y móvil, es una herramienta privativa. (24)

1.5.2 Soluciones similares en el ámbito nacional

En Cuba se han desarrollado disímiles Sistemas de Apoyo a la Toma de Decisiones destinados a diferentes áreas como lo militar, la salud, la educación, etc. En la Universidad de las Ciencias Informáticas (UCI) se han desarrollado muchos de ellos, estos son utilizados por empresas, proporcionando que el trabajo sea más factible y los objetivos y metas trazadas se cumplan con éxito. Ejemplos de algunos sistemas son los siguientes:

Sistema Informático Inteligente de Apoyo a la Toma de Decisiones en la FAR(SIIFAR): es una aplicación web desarrollada en software libre, brinda la posibilidad a los jefes militares, de cualquier nivel de mando, tomar una decisión correcta en un tiempo mínimo cuando las situaciones son complejas, teniendo en cuenta experiencias pasadas almacenadas en la base de casos. Utiliza técnicas de inteligencia artificial, específicamente el razonamiento basado en casos. (25)

Componente para la Toma de Decisiones Clínicas del Sistema Integral para la Atención Primaria de Salud (CTDS): este módulo incluye en su concepción un mercado de datos clínicos, con los indicadores



Capítulo I: Fundamentación Teórica

manipulados en la Atención Primaria de Salud en Cuba y esenciales para la toma de decisiones en este nivel de atención, un subsistema de consultas y finalmente un aplicativo especializado en la Hipertensión Arterial que usa técnicas de inteligencia artificial para su funcionamiento. Facilita las sugerencias diagnósticas de la Hipertensión Arterial mediante experiencias acumuladas inteligentemente y el cruzamiento de criterios de la información contenida en las Historias Clínicas Electrónicas del paciente que disminuya errores clínicos. Utiliza como técnica de inteligencia artificial el razonamiento basado en casos. (26)

1.5.3 Valoración de los sistemas analizados

Tabla 1: Comparación de los sistemas analizados (elaboración propia)

	AAIMS	Cognos	MicroStrategy	SIIFAR	CTDS
Soporte	Aplicación de escritorio	Aplicación web, escritorio	Aplicación web, escritorio y móvil	Aplicación web	Aplicación web
Información que gestionan	Pronósticos sobre participación de mercado, ingresos y rentabilidad de las aerolíneas.	Creación de informes en forma de gráficos, análisis, paneles de control y cuadros de mando.	Permite crear informes en forma de gráficos y ofrece una plataforma unificada para el análisis, reporte y monitorización.	Posibilita a los militares tomar decisiones a partir de experiencias pasadas.	Facilita las sugerencias diagnósticas de la Hipertensión Arterial mediante experiencias acumuladas.
Posibilidad de adaptar al tema	No	No	No	No	No
Disponibilidad del código fuente	No	No	No	No	No
Licencia	Privativa	Privativa	Privativa	Libre	Libre
Implementación de IA	Si	Si	Si	Si	Si



1.5.4 Conclusiones del estudio de las soluciones similares

Después de haberse realizado un análisis sobre algunos DSS que utilizan diferentes empresas en la actualidad, no se encontró ninguno que se pueda utilizar para el apoyo a la toma de decisiones en el aeropuerto, ya que:

- ✓ Son sistemas hechos a la medida según las características de cada empresa.
- ✓ No contemplan la totalidad de los procesos a informatizar.
- ✓ No pueden ser analizados a profundidad ya que no está publicada amplia documentación sobre los mismos.
- ✓ Muchos son privativos y se hace necesario el pago de licencias y/o cuentas para acceder a todas las funcionalidades que ofrecen.

Del análisis de estos sistemas se identificó que la mayoría tienen un ambiente de desarrollo web. Por lo que aportan una serie de elementos significativos para la futura propuesta de solución tales como:

- ✓ Interfaz gráfica amigable.
- ✓ Reportes y graficado.
- ✓ Visualización de la información.
- ✓ Implementación de técnicas de inteligencia artificial.

El estudio de los diferentes sistemas similares de apoyo a la toma de decisiones, permitirá la reutilización de ideas que llevadas a la propuesta de solución, harán de esta una aplicación web eficiente, que satisfaga los requerimientos de los directivos del aeropuerto.

1.6 Metodología de Desarrollo

El desarrollo de software es un proceso riesgoso y difícil de controlar, pero si no se tiene en cuenta la aplicación de una metodología lo que se tiene como resultado es un producto con falta de calidad y por ende clientes insatisfechos. No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.). Existen dos grandes distinciones, las metodologías tradicionales y metodologías ágiles. (27)



Capítulo I: Fundamentación Teórica

Las denominadas metodologías tradicionales o pesadas, se caracterizan por un fuerte énfasis en la planificación y control del proyecto, y en especificación precisa de requisitos y modelado, además requiere una extensa documentación, ya que pretende prever todo de antemano. Han sido utilizadas teniendo un efecto positivo en proyectos de gran tamaño, a lo que respecta a tiempo y recursos. (28)

Las metodologías ágiles dan mayor valor a la colaboración estrecha y directa con el cliente y al desarrollo incremental del producto con iteraciones muy cortas. Son muy útiles en el desarrollo de proyectos que cuentan con cortos períodos de tiempo para su ejecución, sin dejar a un lado la calidad. (29)

A partir de lo analizado con anterioridad, teniendo en cuenta que el sistema a desarrollar no posee un gran número de funcionalidades a implementar y el equipo de desarrollo es pequeño, se decide enfocarse en el estudio de las metodologías ágiles por ser las idóneas para este tipo de proyecto.

Existen muchas metodologías ágiles, según un estudio comparativo realizado en el año 2008, destacan las seis que se relacionan a continuación (30):

- ✓ Proceso Unificado Ágil (AUP)
- ✓ Crystal Methods
- ✓ Dynamic Systems Development Method
- ✓ Scrum
- ✓ Test Driven Development
- ✓ Programación Extrema (XP)

Las metodologías ágiles que se analizaron fueron AUP, Scrum y XP por ser las más utilizadas (29) en la Universidad de las Ciencias Informáticas (UCI⁶).

1.6.1 Scrum

La metodología Scrum es un modelo de referencia que define un conjunto de prácticas y roles que pueden tomarse como punto de partida para el proceso de desarrollo de software. Esta metodología está indicada para proyectos con un alto grado de cambios en los requisitos. Su principal característica es que el desarrollo de software se realiza mediante iteraciones denominadas sprint, donde cada sprint representa

⁶ Centro de estudios universitarios, es una universidad productiva, cuya misión es producir software y servicios informáticos.



Capítulo I: *Fundamentación Teórica*

un incremento del producto y las reuniones diarias que se llevan a cabo a lo largo del proyecto para la coordinación e integración de las actividades a desarrollar. (31)

Las principales características de la metodología Scrum son las siguientes (32):

- ✓ Permite la creación de equipos auto-organizados impulsando la comunicación verbal entre todos los miembros involucrados en el proyecto.
- ✓ Posibilita que los clientes puedan cambiar de idea sobre lo que quieren y necesitan durante el desarrollo del proyecto.
- ✓ Maximiza la capacidad del equipo de realizar las entregas rápidamente y de respuesta a los requisitos emergentes.

1.6.2 Proceso Unificado Ágil

Proceso Unificado Ágil (AUP), es una versión simplificada del Proceso Unificado Racional (RUP), la cual se describe en una forma simple y fácil de entender, y brinda un enfoque de desarrollo de software utilizando técnicas ágiles y conceptos de RUP. (33)

Las principales características de la metodología AUP son (33):

- ✓ Versión simplificada de la metodología RUP.
- ✓ Abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente.
- ✓ El modelado agrupa los tres primeros flujos de RUP (Modelamiento del negocio, Requerimientos y Análisis y Diseño).
- ✓ Dispone de cuatro fases igual que RUP: Incepción o Creación, Elaboración, Construcción y Transición.

1.6.3 Programación Extrema

Programación Extrema (XP) es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo y corto de equipo. Se enfoca en fomentar las relaciones entre los miembros del equipo de desarrollo como la clave del éxito, promoviendo el trabajo en equipo. XP se adecua a proyectos con requisitos imprecisos y propensos al cambio ya que se caracteriza por la sencillez de las soluciones desarrolladas y en la audacia para enfrentar los cambios que surjan. (27)



Capítulo I: *Fundamentación Teórica*

En XP para especificar los requerimientos del software se utiliza como técnica las historias de usuario. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer. Estas historias de usuario pueden reemplazarse, romperse, añadirse nuevas o modificarlas. La programación siempre se define en pareja con el objetivo de lograr resultados con más calidad, mientras uno codifica una funcionalidad el otro se encarga de analizar si dicha funcionalidad es adecuada y está bien diseñada.(27)

1.6.4 Fundamentación de la metodología a utilizar

Después de hacer un estudio de las metodologías expuestas anteriormente, se determinó que sería utilizada la metodología XP dado que permite la creación de un plan de entregas de las funcionalidades al cliente, define la programación del software en pareja, se cuenta con poco tiempo para la entrega de la herramienta y la participación activa del cliente en el proceso de desarrollo. Además es capaz de adaptarse a los cambios que puedan surgir en cualquier punto del ciclo de vida del proyecto.

1.7 Lenguajes de programación

Un lenguaje de programación *“es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes”*. (34)

Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo.

A continuación se presentan los lenguajes de programación a utilizar para el desarrollo de la solución.

1.7.1 Preprocesador de hipertexto (PHP 5.3)

PHP⁷ es un lenguaje de programación utilizado para la creación de sitios web. Es un lenguaje de script utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML⁸ y ejecutadas en el servidor. No necesita ser compilado para ejecutarse, pero para su funcionamiento necesita tener instalado un servidor web, ejemplo de servidores web que soportan PHP se encuentran Apache, Nginx, Internet Information Services con las librerías de PHP. Los archivos cuentan con la extensión (php). Es un lenguaje de una sintaxis muy simple y fácil de aprender que contiene documentación amplia, segura y

⁷ Del inglés Hypertext Preprocessor.

⁸ Del inglés HyperText Markup Language.



Capítulo I: Fundamentación Teórica

factible. Permite aplicar técnicas de programación orientada a objetos y está diseñado para establecer conexión a numerosas bases de datos tales como MySQL, PostgreSQL, Oracle, entre otras. Es multiplataforma y se presenta como una alternativa de fácil acceso para todos, pues se encuentra publicado bajo la PHP License, que está considerada como software libre. Posee una amplia documentación en su página oficial. (35)

A través de PHP se pueden brindar y consumir servicios web desde otras aplicaciones. Un **servicio web** (en inglés, web service o web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores. La interoperabilidad se consigue mediante la adopción de estándares abiertos. (36)

Para brindar el servicio web de la aplicación se estudiaron dos protocolos REST y SOAP.

REST casi siempre utiliza protocolo HTTP⁹ como método de comunicación y XML¹⁰ o JSON¹¹ para intercambiar datos. Cada URL¹² representa un objeto sobre el que se pueden utilizar los métodos de la web como POST, GET, PUT y DELETE. (37)

Los servicios webs mediante **SOAP** representan toda una infraestructura basada en XML, cada objeto puede tener métodos definidos por el programador con los parámetros que sean necesarios. SOAP también utiliza un WSDL¹³ que es un XML con todas las funciones de forma jerárquica a las que se les puede hacer llamadas remotas. SOAP es más seguro que REST por lo que su uso es más adecuado en proyectos grandes donde la seguridad es un aspecto fundamental. (36)

⁹ Protocolo de transmisión del hipertexto.

¹⁰ Del inglés Extensible Markup Language.

¹¹ Del inglés JavaScript Object Notation o Notación de Objetos de JavaScript en español, es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo.

¹² Localizador Uniforme de Recursos. Es el conjunto de caracteres que posibilita la asignación de una dirección exclusiva a un recurso que se encuentra disponible en el espacio virtual.

¹³ Del inglés Web Services Description Language, un formato XML que se utiliza para describir servicios web.



Capítulo I: Fundamentación Teórica

Luego del análisis de los protocolos que se utilizan para brindar servicios web se escoge REST, pues es ligero y con poca configuración ahorrando la carga del servidor. Además se lee fácilmente porque son URLs y siendo la solución una aplicación web es más aconsejable; además no hace falta nada especial para implementarlo. En la aplicación no hay necesidad de seguridad en el servicio web. El SOAP es mucho más ambicioso, para proyectos grandes.

1.7.2 Lenguaje de marcas de hipertexto (HTML 5)

HTML 5 es la quinta revisión del lenguaje básico HTML. Está diseñado no solo para estructurar textos sino también para presentarlos en forma de hipertextos que es el formato estándar de las páginas web. Es un lenguaje interpretado, por lo que necesita de un programa que interprete las etiquetas que están dentro del cuerpo del documento en formato HTML y los muestre en el navegador. Se ha convertido en el formato más usado para la transmisión de documentos electrónicos a través de Internet debido a que la gran variedad de navegadores y exploradores lo soportan, además es compatible con navegadores antiguos. HTML5 responde a las necesidades de los últimos años, puede llegar a dispositivos móviles. (38) (39)

1.7.3 Hojas de estilo en cascada (CSS 3)

CSS¹⁴ es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML, XML2 o XHTML¹⁵. El W3C¹⁶ es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. Detrás del desarrollo de CSS se persigue la idea de separar la estructura de un documento de su presentación. La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. Puede definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo style. (39) CSS3 está dividido en varios documentos separados, llamados "módulos". Debido a la modularización del CSS3, diferentes módulos pueden encontrarse en diferentes estados de su desarrollo. (40)

1.7.4 JavaScript 1.8

JavaScript es un lenguaje de script del lado del cliente, se apoya de la librería JQuery. Este lenguaje de scripts fue desarrollado por Netscape con el objetivo de incrementar las funcionalidades del lenguaje

¹⁴ Del inglés cascading style sheets.

¹⁵ Del inglés de eXtensible Hyper Text Markup Language.

¹⁶ Del inglés World Wide Web Consortium.



Capítulo I: *Fundamentación Teórica*

HTML. No requiere de compilación debido a que es un lenguaje interpretado donde el navegador del usuario se encarga de interpretar y ejecutar las sentencias JavaScript contenidas en una página HTML. Gran parte de la programación en JavaScript se centra en describir objetos y escribir funciones que respondan a movimientos del mouse, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos, ya que es un lenguaje basado en objetos y orientado a eventos. Las páginas web elaboradas pueden ser vistas desde varios navegadores y exploradores web, además los desarrolladores de interfaces de usuario pueden acceder a las diferentes partes y elementos de una página web (botones, textos, imágenes, videos, etiquetas) teniendo así un mejor control de estas. (41)

1.8 Librería para exportar a pdf

Se hace necesario el uso de una librería para convertir los reportes generados en HTML por la aplicación, en formato pdf¹⁷. En el aeropuerto existe instalada una librería con esta funcionalidad llamada Wkhtmltopdf y es usada por otras aplicaciones de la entidad.

Wkhtmltopdf es una librería de código abierto bajo la licencia de LGPLv3¹⁸, su ejecución es mediante la línea de comando para convertir un HTML en un pdf usando el motor Render QT Webkit. Los HTML se convierten en su totalidad "sin cabecera" y no requieren de un servicio de visualización. (43)

Para el desarrollo de la aplicación se selecciona la librería Wkhtmltopdf porque se encuentra instalada en el aeropuerto, es libre y su funcionamiento, mediante la línea de comandos, facilita el trabajo de los desarrolladores.

1.9 Framework PHP

Un framework (marco de trabajo) en el desarrollo de software, es una aplicación con una estructura definida, conceptos, prácticas y criterios del desarrollo de un software, normalmente con artefactos o módulos concretos, que permite de una forma más fácil y organizada la creación de otro proyecto de software. Los frameworks permiten acelerar el proceso, reutilizar código ya existente y promover las buenas prácticas con el uso de patrones. (44)

¹⁷ Formato de archivo de texto.

¹⁸ Del inglés Lesser General Public License o Licencia Pública General Reducida en español. (42)



Capítulo I: Fundamentación Teórica

Algunos de los marcos de trabajo que se analizaron para el desarrollo de aplicaciones web fueron: Zend Framework, Yii Framework y Symfony2 los cuales se detallan a continuación.

1.9.1 Zend framework

Zend framework (ZF) es un marco de trabajo de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. Ofrece un gran rendimiento y una robusta implementación del Modelo Vista Controlador (MVC), una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando, de una manera sencilla, la interfaz orientada a objetos. (45)

1.9.2 Yii framework

Yii es un marco de trabajo utilizado para desarrollar todo tipo de aplicaciones web. Entre sus principales características se destacan que es orientado a objetos, de software libre, posee un alto rendimiento basado en componentes PHP y marco de trabajo de aplicaciones web. Posee generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD¹⁹, entre otros. (46)

1.9.3 Symfony2

Symfony2 es un framework flexible y fácil de aprender que le permite a los desarrolladores construir aplicaciones webs más mantenibles. Está basado en la arquitectura MVC y se ha convertido en uno de los framework más populares de PHP que existe en la actualidad. Se construye a base de bundles²⁰, permitiendo que los desarrolladores configuren y personalicen el sistema de una forma limpia. Es un framework de código abierto, por lo que, los desarrolladores son libres de hacer lo que deseen con él, incluso en un entorno comercial. Todos los detalles de implementación están ocultos detrás de un buen sistema de configuración que permite personalizar todo a través de archivos XML o YAML²¹, o incluso a través de código PHP. Es uno de los marcos de trabajo que mayor documentación presenta. (47)

¹⁹ Es el acrónimo de crear, obtener, actualizar y borrar. Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software.

²⁰ Conjunto estructurado de archivos que implementan una característica única y que puede ser fácilmente compartido con otros desarrolladores. (47)

²¹ Del inglés YAML Ain't Markup Language.



Capítulo I: *Fundamentación Teórica*

1.9.4 Fundamentación del marco de trabajo a utilizar

Para la realización del sistema se decide utilizar Symfony en su versión 2.5 por haber sido ideado para exprimir al límite todas las nuevas características de PHP en su versión 5.3 y por eso es uno de los marcos de trabajo PHP con mejor rendimiento. De todas las novedades introducidas por PHP 5.3 las más relevantes para los programadores de Symfony2 son las funciones anónimas y los namespaces brindando una mejor organización del código de la aplicación (48). Una de ventajas de Symfony es que su arquitectura interna está completamente desacoplada, permitiendo que se reemplacen o eliminen fácilmente aquellas partes que no encajen en el proyecto, lo cual le permite adaptarse con facilidad a los nuevos cambios que puedan surgir durante el proceso de desarrollo del software. También existen módulos que se han desarrollado para ser integrados a Symfony 2 y además serán utilizados para el desarrollo del proyecto para realizar operaciones tales como: generación de reportes en formato pdf, manipulación de seguridad y usuarios, gestión de envíos de correos electrónicos, entre otros. Además Symfony2 hace uso de los lenguajes HTML 5 y de CCS 3 que serán utilizados para ofrecerle al usuario un entorno vistoso y una mejor interacción con la aplicación desarrollada. (47) Los desarrolladores tienen experiencia en su uso y funcionalidad.

1.10 Frameworks de CSS. Bootstrap

En la actualidad el uso de marcos de trabajo se ha hecho muy común entre desarrolladores para hacer más rápido y fácil la creación de diseños web. Se decide utilizar el marco de trabajo Bootstrap 3 debido a que es fácil de integrar con cualquier proyecto web y trae consigo varios componentes que facilitan un estilo elegante e intuitivo al sitio web a desarrollar. Combina CCS3, HTML 5 y JavaScript para lograr una interfaz agradable y vistosa al usuario. Puede integrarse con diferentes librerías de JavaScript como jQuery. Además puede crear interfaces que se adapten a distintos navegadores gracias a su potente marco de trabajo constituido por numerosos componentes web facilitando el ahorro de esfuerzo y tiempo. (49)

1.11 Framework JavaScript. JQuery

JQuery es un framework JavaScript (50), pero en la presente investigación será denominado como librería o biblioteca (51). Es un software libre y de código abierto bajo la Licencia Pública General de GNU GPL²².

²² En español Licencia Pública General, su propósito es declarar que el software cubierto por esta licencia es software libre.



Capítulo I: *Fundamentación Teórica*

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM²³ y peticiones AJAX²⁴. Es un producto serio, estable, bien documentado y cuenta con un gran equipo de desarrolladores a cargo de la mejora y actualización del marco de trabajo; además facilita la compatibilidad con diferentes navegadores. Tiene una gran comunidad de creadores de componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc. (50)

Por las ventajas y características que brinda JQuery, ajustándose al desarrollo de la presente investigación, se decide utilizar JQuery en su versión 1.11 para desarrollar una interfaz amigable e intuitiva para el usuario, además es requerido por Bootstrap.

1.12 Sistemas gestores de bases de datos

En la actualidad son muchas las formas que se utilizan para guardar la información, una de las más utilizadas son las bases de datos. Los Sistemas de Gestión de Bases de Datos (SGBD) permiten la utilización y/o la actualización de los datos almacenados en una o varias Base de Datos (BD²⁵) por uno o varios usuarios simultáneamente desde distintos puntos de vistas. (52)

Dentro de los gestores de base de datos más conocidos mundialmente se encuentran MySQL y PostgreSQL.

1.12.1 MySQL

MySQL surge como idea de la empresa MySQL AB²⁶ con el objetivo de que cumpla el estándar SQL²⁷. Es un sistema de gestión de base de datos relacional, esto significa que archiva los datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Es multihilo por lo que puede realizar varias tareas a la vez y es también multiusuario, lo que le permite satisfacer de forma simultánea, las necesidades de varios usuarios que comparten los mismos recursos. Funciona además sobre múltiples

²³ Del inglés Document Object Model.

²⁴ JavaScript y XML asíncronos, es una técnica de desarrollo web para crear aplicaciones interactivas.

²⁵ Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora, o sea, que puede considerarse una colección de datos variables en el tiempo. (52)

²⁶ Es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL, o sea es la compañía propietaria de MySQL. (53)

²⁷ Lenguaje de Consulta Estructurado.



Capítulo I: *Fundamentación Teórica*

plataformas. (54) El software MySQL tiene licencia dual²⁸, se puede usar de forma gratuita bajo licencia GNU GPL o bien adquiriendo licencias comerciales de MySQL AB. (56)

1.12.2 PostgreSQL

PostgreSQL es un potente sistema de BD, el código fuente de PostgreSQL está disponible bajo una licencia de código abierto liberal PostgreSQL, es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Cuenta con una arquitectura probada que a través de un conjunto de programas permite crear y mantener una BD, asegurando su integridad, confidencialidad y seguridad. Incluye diferentes tipos de datos, también es compatible con el almacenamiento de grandes objetos binarios, como imágenes, sonidos, o de vídeo. Es multiplataforma y extensible, ampliamente popular, ideal para tecnologías web. (57)

1.12.3 Fundamentación del Sistema Gestor de Bases de Datos (SGBD) a utilizar

Como Sistema Gestor de Base de datos se seleccionó PostgreSQL en su versión 9.3 ya que es un servidor de BD relacional y libre. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Cuenta con una gran comunidad de desarrollo en Internet y es una herramienta multiplataforma. Además la BD de la cual se nutrirá el Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez estará en PostgreSQL. Para conectar las bases de datos de PostgreSQL se utilizará PgAdmin III que también es libre.

1.13 Servidores web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada de HTTP). (58)

Existen numerosos servidores web dentro de los que se encuentran Microsoft IIS, Apache, Nginx, LiteSpeed, Google Servers, Tomcat, Lighttpd, IBM Servers, Yahoo Traffic Server, Jetty, AOLserver, Roxen

²⁸ Algunos proyectos intentan financiarse mediante el uso de concesiones de licencia duales, en los cuáles trabajos propietarios derivados deben pagar al propietario de los derechos para utilizar el código, mientras sigue siendo libre para el uso en otros proyectos libres. (55)



Capítulo I: Fundamentación Teórica

y Caudium. Los tres servidores web más usados actualmente son Apache, Nginx y Microsoft IIS como se puede apreciar en el anexo 4. (59)

1.13.1 Apache

El servidor HTTP Apache es un servidor web de código abierto y uso gratuito, multiplataforma, muy robusto y que destaca por su seguridad y rendimiento. Su objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan. Apache es altamente configurable, es flexible, extensible, modular y muy popular por lo que es fácil adquirir ayuda y soporte. (60) En la actualidad Apache es el servidor HTTP más usado, dando servicios al 70% de los sitios web en el mundo y creciendo aún su cuota de mercado. (61)

1.13.2 Nginx

Es un servidor web HTTP de código abierto bajo la Licencia BSD²⁹ simplificada. Utiliza una arquitectura basada en eventos que hace que su consumo de recursos crezca de forma predecible. Inicialmente se creó para funcionar en sistemas operativos Unix, pero más tarde también apareció una versión compatible con Windows, por tanto se utiliza en sistemas como GNU/Linux, FreeBSD, Solaris, Mac OS X y Windows. Los módulos no pueden ser cargados de forma dinámica, deben ser incluidos en la compilación. Además, no se pueden desactivar en tiempo de ejecución debido a que están completamente compilados e integrados en el binario principal. No posee aún la cantidad de documentación y usuarios de otros servidores web. (63) Su instalación y configuración no es tan sencilla como otros servidores web como Apache; porque administra gran cantidad de módulos y soporta otros lenguajes distintos de PHP como Python. (64)

1.13.3 Internet Information Server (Microsoft IIS)

Es un servidor web privativo que contiene una serie de servicios para los ordenadores que funcionan con Windows. Ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones web. El servidor web se basa en varios módulos que le dan capacidad para procesar distintos

²⁹ Es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). (62)



Capítulo I: Fundamentación Teórica

tipos de páginas, por ejemplo Microsoft incluye los ASP³⁰ y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl. (65)

1.13.4 Selección del servidor web a utilizar

Después de haber analizado los servidores web expuestos anteriormente, se concluye que Nginx es de código abierto, multiplataforma, pero no existe publicada una amplia documentación del mismo y resulta un poco difícil su instalación y configuración. Microsoft IIS no es multiplataforma o sea solo es compatible con Windows, debe ser usado bajo licencia y está dirigido más a páginas ASP que las PHP. Mientras que Apache es un proyecto libre y por tanto gratuito, además, cuenta con una gran comunidad de usuarios que facilita encontrar solución tanto a errores del propio servidor como a errores en la configuración del mismo. Por tanto se decide seleccionar Apache en su versión 2.4 como servidor web.

1.14 Entorno de Desarrollo Integrado (IDE)

IDE es una aplicación que facilita el desarrollo de aplicaciones. Consiste en un editor de código, un compilador, un depurador y una interfaz gráfica, que ayudan a un desarrollador de software a crear aplicaciones, de forma independiente o que sean parte de aplicaciones ya existentes. (66)

En la actualidad existen varios Entorno de Desarrollo Integrado, ejemplo PHPStorm, Eclipse y Netbeans. A continuación se hará un estudio de los mismos.

1.14.1 PHPStorm

PHPStorm es un IDE de programación desarrollado por JetBrains. Es un producto comercial para los desarrolladores comerciales, y libre para los desarrolladores de código abierto. (67) (68) Permite editar código no sólo del lenguaje de programación PHP como lo indica su nombre sino de JavaScript, CSS, HTML, Sass, Menos, CoffeeScript, y muchos otros idiomas. Para PHP incluye autocompletado de consultas SQL, y gestión integrada de bases de datos tipo MySQL, Oracle, SQL Server, PostgreSQL, entre otras. Es multiplataforma pues es compatible con Sistemas Operativos Windows, Linux y Mac OS X. (69) Ofrece una integración casi perfecta con Symfony gracias a su plugin³¹ para Symfony2 (71).

³⁰ Del inglés Active Server Pages.

³¹ Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). No se trata de un parche ni de una actualización, es un módulo aparte que se incluye opcionalmente en una aplicación. (70)



1.14.2 Eclipse

Eclipse es un IDE de código abierto y multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación. Admite distintos tipos de lenguaje como PHP, C++ y Java. Integra herramientas y funciones necesarias para el trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. Eclipse puede considerarse actualmente como el IDE referencia en el mundo del software libre, con el inconveniente de funcionar mejor sobre una máquina virtual Java que no es libre.⁽⁷²⁾ Se integra con Symfony 2 a través de plugin. ⁽⁷³⁾

1.14.3 NetBeans

NetBeans es un entorno de desarrollo que permite crear aplicaciones de escritorio, aplicaciones web, entre otras. Es una aplicación gratuita y sin restricciones de uso. Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas y ha tenido un gran éxito y aceptación por la comunidad de desarrolladores del software libre. Se integra con Symfony 2 sin tener que utilizar plugins solamente especificando un fichero de Symfony comprimido en zip³². ⁽⁷⁵⁾

Las principales características del NetBeans son ⁽⁷⁵⁾:

- ✓ Entorno de desarrollo multiplataforma, multilenguaje y amigable tanto para usuarios novatos como para profesionales.
- ✓ Disponible en muchos idiomas.
- ✓ Es de código abierto.
- ✓ Desarrollado por módulos. Brinda la posibilidad de agregar nuevos módulos para aumentar su funcionalidad.
- ✓ Cuenta con una amplia documentación y una gran comunidad de usuarios.
- ✓ Admite distintos tipos de lenguaje como PHP, C++, Java, entre otros.

1.14.4 Selección del IDE a utilizar

Se utiliza como entorno de desarrollo NetBeans 8.0, debido a que es un proyecto de código abierto, soporta lenguajes dinámicos como PHP y Java Script siendo lenguajes de programación utilizados en la solución, es multiplataforma, tiene una interfaz muy amigable e intuitiva. Esta herramienta tiene

³² Se utiliza para la comprensión de datos (documentos de texto, imágenes, programas, etc.) sin pérdida de calidad. ⁽⁷⁴⁾



Capítulo I: Fundamentación Teórica

integración con el marco de trabajo Symfony 2 que es el marco de desarrollo a utilizar para la aplicación. Además ofrece soporte para frameworks como jQuery, que es uno de los utilizados, agilizando el desarrollo de la solución. Los desarrolladores tienen experiencia en su uso y funcionalidad, además cuenta con soporte para PHPUnit³³, herramienta seleccionada para realizar las pruebas.

1.15 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering por sus siglas en inglés o Ingeniería de Software Asistida por Computadora), es un conjunto de programas destinados a aumentar la productividad en el desarrollo del software. Esta herramienta brinda ayuda a los analistas, ingenieros de software y desarrolladores durante todos los pasos de ciclo de vida de desarrollo de un software. (77)

Como herramientas CASE se analizaron Erwin y Visual Paradigm for UML.

1.15.1 ERwin

La herramienta ERwin Data Modeling ofrece un entorno de modelado de datos de colaboración para administrar datos empresariales con una interfaz intuitiva y gráfica. Las partes técnicas y de negocio pueden compartir una vista de la información en contexto. Con una vista centralizada de definiciones de datos claves, puede comprender mejor los datos corporativos, que se administran de manera más eficiente y rentable, mediante la colaboración por diseño. Brinda productividad en diseño, generación, y mantenimiento de aplicaciones desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada. (78)

1.15.2 Visual Paradigm for UML

Visual Paradigm (Paradigma Visual en español) fue concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Ofrece un completo conjunto de herramientas de equipos de desarrollo de software necesario para la captura de requisitos, software de planificación, la planificación de controles, el modelado de clases, modelado de datos. Permite la rápida construcción de aplicaciones de calidad a un menor coste, soporta aplicaciones

³³ Framework para realizar pruebas unitarias en PHP. Se trata de un miembro de la familia xUnit y proporciona un marco que realiza de manera fácil la escritura y la ejecución de las pruebas, así como el análisis de los resultados. Las pruebas son fáciles de aprender, de escribir y de ejecutar. (76)



Capítulo I: Fundamentación Teórica

del tipo web, además dibujar todos los tipos de diagramas de clases, genera código desde diagramas, es fácil de instalar y posee licencia gratuita. (79)

1.15.3 Fundamentación de la herramienta CASE a utilizar

Se decide utilizar durante la etapa de modelación y para diseñar el sistema como herramientas CASE el Visual Paradigm en su versión 8.0 debido que posee una alta capacidad de integración con UML³⁴ que será utilizado para el desarrollo del sistema. Además por su alta capacidad de integración con lenguajes de programación. Es una herramienta multiplataforma, de código abierto, disponible en varios idiomas además de ser fácil de instalar y actualizar. Visual Paradigm permitirá generar el modelo de dominio y el modelo de físico de la base de datos.

1.16 Herramientas utilizadas en el proceso de Minería de Datos

A continuación se exponen las herramientas necesarias para llevar a cabo el proceso de descubrimiento de conocimientos. Teniendo en cuenta las ventajas que proveen las tecnologías de software libre sobre las privativas en cuanto a precio, accesibilidad, calidad y diseño; se decide desarrollar este proceso sobre tecnologías libres atendiendo a la flexibilidad y características que poseen estas herramientas. Dentro de las más utilizadas y disponibles en el mercado para enfrentar un proceso KDD se pueden encontrar KNIME, RapidMiner y Weka.

1.16.1 KNIME

Es un entorno totalmente gratuito para el desarrollo y ejecución de técnicas de minería de datos. Fue desarrollado originalmente en el departamento de bioinformática y minería de datos de la Universidad de Constanza, en la actualidad continúa su desarrollo, además de prestar servicios de formación y consultoría. Soporta varias tareas de minería de datos clustering, clasificación, regresión, entre otras. KNIME usa componentes de Weka y está desarrollado sobre la plataforma Eclipse. (80)

1.16.2 RapidMiner

Es una herramienta de código abierto implementada en Java, basada en el aprendizaje automático para el descubrimiento de conocimiento y la minería de datos. RapidMiner tiene implementado un amplio rango

³⁴ Del inglés Unified Modeling Language o Lenguaje Unificado de Modelado en español.



Capítulo I: *Fundamentación Teórica*

de algoritmos y técnicas de minería de datos. Trabaja bajo las plataformas Windows y Linux. Soporta técnicas de minería de datos como agrupamiento, clasificación, visualización, entre otras. (81)

1.16.3 Weka

Entre las herramientas libres más utilizadas para la minería de datos se encuentra **Weka** (Waikato Environment for Knowledge Analysis), es una herramienta visual de distribución libre para el análisis y extracción de conocimiento a partir de datos. (82)

Principales ventajas de la herramienta (82):

- ✓ Es multiplataforma.
- ✓ Contiene una extensa colección de técnicas para pre-procesamiento y modelado de datos.
- ✓ Soporta varias técnicas de minería de datos especialmente pre-procesamiento, agrupamiento, clasificación, regresión, visualización y selección.
- ✓ Permite la combinación de varios algoritmos basados en técnicas de minería de datos, para obtener mejores resultados en el descubrimiento de conocimiento.
- ✓ Es capaz de mostrar los datos en varios tipos de gráficos con el objetivo de una mejor comprensión y análisis.

Weka es una herramienta desarrollada en Java y utiliza la **máquina virtual de Java** para su funcionamiento.

La máquina virtual de Java es el entorno en el que se ejecutan los programas Java, su misión principal es la de garantizar la portabilidad de las aplicaciones Java. Define esencialmente un ordenador abstracto y especifica las instrucciones que este ordenador puede ejecutar. El intérprete Java específico ejecuta las instrucciones que se guardan en los archivos cuya extensión es .class. Las aplicaciones desarrolladas en Java no necesitan un sistema operativo específico (Windows, Linux) para su ejecución ya que la máquina virtual es la que interpreta las instrucciones posibilitando que sea multiplataforma. (83)

1.16.4 Fundamentación de la herramienta de Minería de Datos y algoritmo a utilizar

Las herramientas investigadas comparten características acordes a las necesidades de esta investigación puesto que son herramientas libres, requieren conocimiento de minería de datos, son multiplataforma, presentan una interfaz amigable e implementan los árboles de decisión que es la técnica de inteligencia



Capítulo I: Fundamentación Teórica

artificial a implementar en la solución del problema a resolver. Se decide seleccionar Weka en su versión 3.7.10, por las características que presenta, además se tiene un conocimiento previo de la herramienta por parte de los desarrolladores, permitiendo un ahorro considerable de tiempo. Adicionalmente de ser una herramienta de uso libre, su código fuente es abierto, lo que significa que no solo se puede hacer uso de los algoritmos implementados, sino también puede analizarse la implementación realizada de cada uno de ellos.

Existen varios algoritmos de clasificación basados en árboles de decisión en la herramienta Weka, a continuación se presentan los resultados de la ejecución de algunos de ellos para fundamentar su elección. Para este experimento se contó con una muestra de 1846 instancias de planes de vuelos diarios del aeropuerto Juan Gualberto Gómez ya clasificadas por los directivos del aeropuerto. (84) A continuación se muestra una tabla con los algoritmos y los resultados arrojados. (Ver anexo 5)

Tabla 2: Algoritmos de clasificación en Weka (elaboración propia)

Algoritmos	Instancias clasificadas correctamente	Porcentaje de instancias clasificadas correctamente	Instancias clasificadas incorrectamente	Porcentaje de instancias clasificadas incorrectamente
DecisionStump	1453	78.71 %	393	21.28%
HoeffdingTree	1758	95.23%	88	4.76%
J48	1845	99.94%	1	0.05%
REPTree	1843	99.83%	3	0.16%

Después de un análisis de los resultados arrojados por los algoritmos en Weka se decide utilizar el algoritmo “J48” para la clasificación de los reportes diarios de los vuelos, porque de los algoritmos anteriores es el que mejor clasifica este conjunto de datos.

1.17 Conclusiones del Capítulo

Los métodos científicos de investigación empleados en el estudio permitieron desarrollar la teoría que sustenta esta investigación. El análisis de las soluciones existentes en el ámbito nacional e internacional



Capítulo I: *Fundamentación Teórica*

permitió estudiar las características de cada una de ellas, llegando a la conclusión que estos sistemas no cumplen en su totalidad con las características necesarias para el desarrollo del Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez. Se seleccionó la metodología XP para guiar el proceso de desarrollo, como lenguaje de modelado fue elegido UML y como herramienta CASE para realizar el modelado del software se optó por Visual Paradigm for UML 8.0. Tras determinarse que la propuesta de solución será una aplicación web, las tecnologías para su desarrollo se distinguen por su uso en el servidor y en el cliente. En el lado del servidor el lenguaje a utilizar será PHP 5.3, este lenguaje será utilizado aprovechando las potencialidades que provee el framework de desarrollo Symfony 2.5. En el lado del cliente será utilizado HTML 5 como lenguaje de maquetado, como lenguaje para proveer estilos visuales se empleará CSS 3, este lenguaje será usado a través del framework de diseño Bootstrap 3, como lenguaje de scripting en el cliente será empleado JavaScript 1.8 a través de la librería JQuery 1.11. Para el almacenamiento de la información se seleccionó PostgreSQL 9.3 como gestor de bases de datos. El servidor web a utilizar será Apache 2.4 y por sus disímiles características que lo hacen un potente IDE fue seleccionado NetBeans en su versión 8.0 para desarrollar la propuesta de solución.



Capítulo II: Exploración y Planificación

Capítulo II: Exploración y Planificación

2.1 Introducción

Después de haberse seleccionado la metodología, las herramientas y tecnologías para el desarrollo de la propuesta de solución se abordarán temas relacionados con el funcionamiento del sistema, siendo guiados por las dos primeras fases en las que se divide la metodología XP: Exploración y Planeación. En estas fases se generarán artefactos como las historias de usuario, la estimación del esfuerzo por cada una de las historias de usuario, el plan de iteraciones y el plan de entregas.

2.2 Propuesta de solución

Para darle respuesta al problema analizado, se decide implementar una aplicación web de apoyo a la toma de decisiones sobre el estado operativo del aeropuerto, que contribuya a informatizar el proceso de análisis de datos. Se decide desarrollar la aplicación en un ambiente web porque en el aeropuerto trabajan varios directivos, los cuales tienen experiencia en interactuar con aplicaciones web, y así todos podrán tener acceso a esta de una forma fácil a través de un navegador. Además los directivos cuentan con smartphones y el acceso mediante la web será fundamental.

El sistema contará con dos escenarios fundamentales:

El escenario de configuración permitirá al administrador gestionar los usuarios para enviarles los pronósticos de reportes diarios con un día de anticipación.

El escenario de reportes contiene las funciones elementales del Sistema de Apoyo a la Toma de Decisiones, permitirá a cualquier usuario ver los reportes, anuales, mensuales, semanales y diarios, estos últimos con su clasificación según el comportamiento operativo del día, además generar todos los reportes en formato pdf.



Capítulo II: Exploración y Planificación

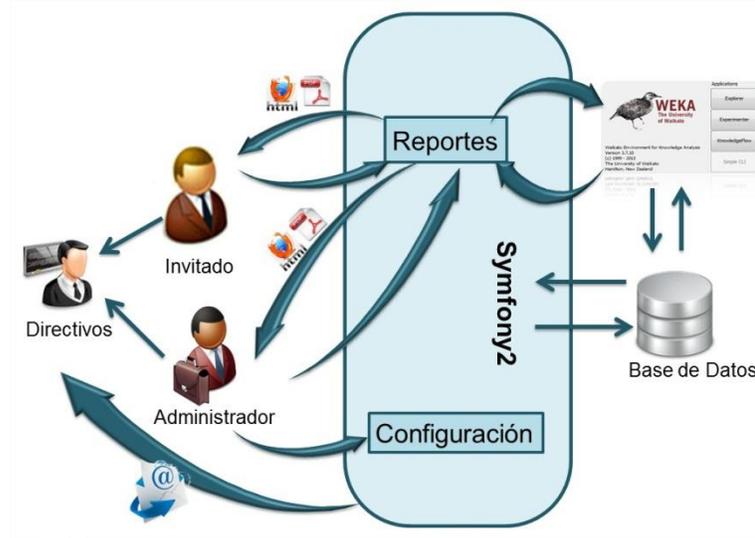


Figura 2: Propuesta de solución (elaboración propia)

2.3 Modelo Conceptual

Un modelo del dominio o modelo conceptual es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. 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 y sus atributos. (85)

Se propone la realización de un modelo conceptual para un mejor entendimiento de la propuesta de solución a desarrollar, donde se muestran los principales conceptos a utilizar manejándolos con un vocabulario común para entender el contexto en que se ubica el sistema. La Figura 4 muestra los principales conceptos y sus relaciones presentes en el modelo conceptual.



Capítulo II: Exploración y Planificación

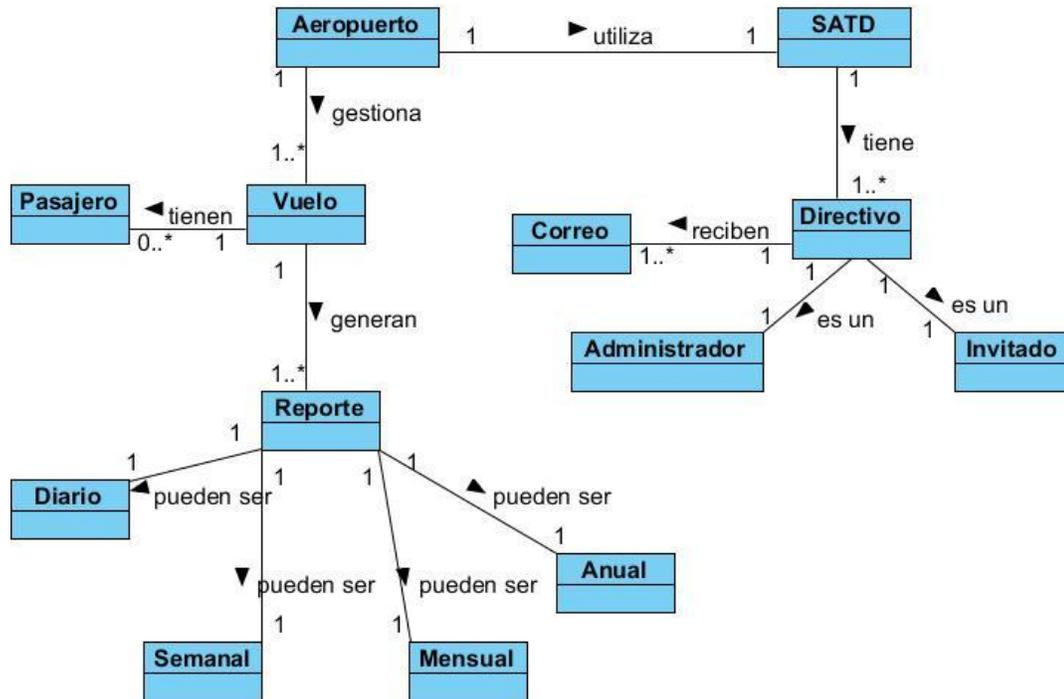


Figura 3: Diagrama conceptual (elaboración propia)

2.3.1 Descripción de los elementos del modelo conceptual

Aeropuerto: empresa que gestiona vuelos.

Vuelo: viaje de un avión de un lugar a otro.

Pasajero: persona que viaja en un vuelo.

Reporte: información analítica de vuelos.

Diario: información analítica de vuelos por días.

Semanal: información analítica de vuelos por semana.

Mensual: información analítica de vuelos por mes.

Anual: información analítica de vuelos por año.

Directivo: persona que puede ser administrador o invitado, analiza los reportes, envía y recibe correos y trabaja en el aeropuerto.

Correo: mensaje enviado de una persona a otra.

SATD: Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez.

Administrador: directivo que tiene permiso para la gestión de correos.

Invitado: persona que puede ver todos los reportes.



Capítulo II: Exploración y Planificación

2.4 Personal relacionado con el sistema

Una de las premisas fundamentales a tener en cuenta cuando se comienza el desarrollo de cualquier sistema informático, la constituye el delimitar el personal al cual va dirigido el mismo. Se define como personas relacionadas con el sistema a aquellas personas que interactúan con la aplicación. Para estos existen algunas restricciones específicas.

Usuario administrador: es el usuario encargado principalmente a la gestión de destinatarios a los que se les envía correos.

Usuario invitado: es el usuario que utiliza la aplicación para obtener todo tipo de información relacionada con los reportes.

2.5 Exploración

La metodología XP comienza con la fase de exploración, en esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. (27)

2.5.1 Funcionalidades a implementar

Luego de varias entrevistas con el cliente se definieron las funcionalidades a informatizar en la propuesta de solución con sus prioridades (Ver tabla 2).

Tabla 3: Funcionalidades a implementar (elaboración propia)

Funcionalidades	Prioridad
1. Reportes Diarios	Alta
1.1. Mostrar vuelos por hora	Alta
1.2. Mostrar pasajeros por hora	Alta
2. Mostrar clasificación del día	Alta
3. Reportes Semanales	Alta
3.1. Mostrar vuelos por día	Alta
3.2. Mostrar pasajeros por día	Alta
4. Reportes Mensuales	Alta



Capítulo II: Exploración y Planificación

4.1. Mostrar vuelos por mes	Alta
4.2. Mostrar pasajeros por mes	Alta
5. Reportes Anuales	Alta
5.1. Mostrar vuelos por año	Alta
5.2. Mostrar pasajeros por año	Alta
6. Autenticar administrador	Alta
7. Gestionar correos	Media
7.1. Insertar destinatario	Media
7.2. Listar destinatario	Media
7.3. Eliminar destinatario	Media
7.4. Editar destinatario	Media
8. Guardar como pdf cada uno de los reportes	Media
8.1. Guardar como pdf los vuelos por hora	Media
8.2. Guardar como pdf los pasajeros por hora	Media
8.3. Guardar como pdf vuelos por día	Media
8.4. Guardar como pdf los pasajeros por día	Media
8.5. Guardar como pdf vuelos por mes	Media
8.6. Guardar como pdf los pasajeros por mes	Media
8.7. Guardar como pdf vuelos por año	Media
8.8. Guardar como pdf los pasajeros por año	Media
9. Servicios web	Media
10. Diseño adaptativo	Media

2.5.2 Historias de Usuarios (HU)

La metodología XP genera diferentes artefactos, uno de los más importantes y utilizados para especificar las funcionalidades que brindará el sistema son las denominadas HU. No lograr determinar todas las HU desde un inicio no es un problema, puesto que al inicio de cada iteración se registran los cambios en estas y a partir de ahí se planifica la próxima iteración. Las HU emplea terminología del cliente sin lenguaje técnico, son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración. (27)



Capítulo II: Exploración y Planificación

La metodología XP no define una plantilla específica para representar las HU, por lo que se definió una plantilla que contendrá, el nombre, el número, el usuario que realiza la acción dentro del sistema, la estimación del tiempo de desarrollo, el nivel de prioridad en el negocio, el riesgo en caso de no realizarse, la iteración en la que será implementada, una breve descripción y un apartado para agregar alguna observación si es necesario.

El equipo de desarrollo a partir de una reunión con el cliente, identificaron un total de 10 HU, a continuación se ejemplifica con la HU Reportes diarios y las restantes HU se describen en el anexo 6.

Tabla 4: HU Reportes diarios (elaboración propia)

Historias de Usuario	
Número: 1	Nombre: Reportes diarios.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Medio.
Iteraciones Asignadas: 1	Puntos de Estimación: 1 semana.
Descripción: El usuario podrá consultar los reportes diarios de los pasajeros y los vuelos por horas del día, los vuelos visualizado en arribos y despegues. La información se reflejará en diferentes tipos de gráficos.	
Observaciones: Para visualizar un reporte dado un día, este debe estar en la base de datos, de lo contrario el sistema mostrará un gráfico sin información.	

2.5.3 Requerimientos generales del sistema

Cada sistema presenta distintas características no funcionales que hacen que el producto sea atractivo. Aunque estos requerimientos no definen el éxito del producto, influyen considerablemente en la evaluación del mismo. Para lograr la satisfacción del cliente y una buena calidad en el sistema se listaron los siguientes:

Usabilidad: para utilizar el sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente web en sentido general.

Disponibilidad: el sistema deberá tener un 100% de disponibilidad por lo que podrá ser usado las 24 horas del día por todos los usuarios.



Capítulo II: Exploración y Planificación

Hardware: el sistema debe poder ejecutarse en una computadora con un microprocesador de 2ghz o superior, 1gb de memoria RAM o superior y 1gb o más de espacio libre en disco.

Software: en las computadoras clientes se requiere un sistema operativo con navegadores Mozilla Firefox en su versión 30 o superior o Chrome en su versión 35 o superior. En el servidor se debe contar con el sistema operativo Ubuntu 14.04, la máquina virtual de Java en su versión 1.6.0.20, el servidor web Apache en su versión 2.4, PHP en su versión 5.3 y PostgreSQL en su versión 9.3.

Seguridad: solo podrá ser editada la información únicamente por las personas que tienen permisos para realizar esta actividad. Solo el administrador podrá gestionar los usuarios para el envío de correos.

Interfaz de usuario: el diseño de la interfaz visual debe ser minimalista sin muchas animaciones ni imágenes pesadas, que perjudiquen la rapidez de la aplicación. La interfaz debe ser diseñada para hacer la navegación sencilla permitiendo el fácil entendimiento de las funcionalidades que brinda el sistema, además de poseer colores refrescantes para una mejor interacción entre el usuario y la aplicación.

2.6 Planificación

En esta fase el cliente establece la prioridad de cada HU, posteriormente los programadores realizan una estimación del esfuerzo que cuesta implementar cada una de ellas. Además se generan otros artefactos de suma importancia para el desarrollo del proyecto como el plan de iteraciones y el plan de entregas. (86)

2.6.1 Estimación de esfuerzos por HU

Para el desarrollo satisfactorio de la solución propuesta, se realizó una estimación de esfuerzo para cada una de las HU, arrojando los siguientes resultados:

Tabla 5: Puntos de estimación por historias de usuario (elaboración propia)

No	Historias de Usuario	Puntos de estimación (semanas)
1	Reportes Diarios	1
2	Clasificación del día	2
3	Reportes Semanales	1
4	Reportes Mensuales	1
5	Reportes Anuales	1
6	Autenticar administrador	1
7	Gestionar correos	2



Capítulo II: Exploración y Planificación

8	Guardar como pdf	1
9	Servicios web	2
10	Diseño adaptativo	2

2.6.2 Plan de Iteraciones

En el plan de iteraciones se especifican cuáles son las HU definidas por el cliente que serán implementadas en cada iteración del sistema. La duración ideal de una iteración es de 1 a 3 semanas. Al terminar cada iteración la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos. (86)

A modo de resumen se presenta la siguiente tabla que muestra las cinco iteraciones analizadas previamente con las HU que incluyen y su duración:

Tabla 6: Planificación de las iteraciones (elaboración propia)

Iteraciones	Historias de Usuario	Duración total
1	Reportes Diarios	3 semanas
	Clasificación del día	
2	Reportes Semanales	3 semanas
	Reportes Mensuales	
	Reportes Anuales	
3	Autenticar administrador	3 semanas
	Gestionar correos	
4	Guardar como pdf	3 semanas
	Servicios web	
5	Diseño adaptativo	2 semanas

2.6.3 Plan de entregas

El cronograma de entregas establece por agrupaciones las HU para cada entrega y el orden de las mismas, además este cronograma se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. (86)



Capítulo II: Exploración y Planificación

El plan de entrega lo definió el equipo de desarrollo de la propuesta de solución junto al cliente a partir de una reunión. En el siguiente plan de entrega se reflejan las fechas de culminación de las iteraciones y sus correspondientes HU.

Tabla 7: Plan de entregas (elaboración propia)

Entregable	Final Iteración # 1	Final Iteración # 2	Final Iteración # 3	Final Iteración # 4	Final Iteración # 5
Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez versión 1.0.	2 de febrero de 2015	13 de marzo de 2015	3 de abril de 2015	24 de abril de 2015	8 de mayo de 2015

2.7 Conclusiones del capítulo

En este capítulo se realizó el modelo conceptual para el acercamiento al dominio del problema, identificándose 13 conceptos. Las fases Exploración y Planificación de la metodología XP, permitieron generar artefactos como las HU, el plan de iteraciones y el plan de entregas, los cuales proporcionaron establecer claridad en aspectos importantes relacionados con el sistema. La confección de 10 HU logró definir las necesidades funcionales del sistema; además se especificaron los requerimientos generales del sistema que aseguran que el producto sea agradable para el cliente.



Capítulo III: Implementación y Prueba

3.1 Introducción

En este capítulo se presenta una descripción de la arquitectura de la propuesta de solución, en este caso Modelo-Vista-Controlador, además de los patrones de diseño determinados por el uso del marco de trabajo Symfony; se presentan las tarjetas Clase-Responsabilidades-Colaboradores y el modelo de datos. Se muestran las tareas de ingeniería necesarias para llevar a cabo el proceso de desarrollo. Por último se define la estrategia de pruebas a seguir, verificando de forma unitaria las funcionalidades de la herramienta y generando los casos de pruebas para realizar las pruebas de aceptación.

3.2 Patrón de Arquitectura

Un patrón arquitectónico o patrón de arquitectura de software es un esquema genérico probado que ofrece soluciones a un problema específico y recurrente dentro de un cierto contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para su propia evolución. (87)

Como base para el desarrollo de la aplicación propuesta se utilizó el marco de trabajo Symfony 2.5 que implementa, como lo hacen muchos otros, el patrón arquitectónico MVC (Modelo Vista Controlador). Es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario; facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla. El patrón arquitectónico MVC permite no mezclar lenguajes de programación en el mismo código, por lo que lo separa en 3 capas: (87)

El Modelo: es la representación de la información con la cual la aplicación opera, gestionando todos los accesos a dicha información, definiendo la lógica del negocio. Responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador). (45) Este está representado en las clases “Entity” que posee Symfony.

La Vista: convierte el modelo en una página web que facilita al usuario interactuar con ella. En el caso de una aplicación web, la “Vista” sería una página HTML con contenido dinámico sobre la que el usuario puede realizar sus operaciones. (45) Estas se representan dentro del directorio “views” que trae Symfony, las cuales poseen la extensión “html.twig”.



Capítulo III: Implementación y Prueba

El Controlador: es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que esta, lo presente al usuario, de forma “humanamente legible”. (45) Este está constituido por las clases “Controller” de la aplicación.

En la Figura 5 se puede apreciar la representación en concreto del patrón arquitectónico MVC en la propuesta de solución.



Figura 4: Representación del patrón arquitectónico MVC en la propuesta de solución (elaboración propia)

3.3 Patrones de diseños

Craig Larman define en la segunda edición de su libro “UML y Patrones” a un patrón como “(...) un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos” (85). Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software. Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. (88)

El marco de trabajo Symfony utiliza en su implementación un conjunto de patrones de diseño que garantizan una arquitectura robusta e íntegra, además proporcionan buenas prácticas durante la implementación del sistema. Para el desarrollo de la propuesta de solución que se describe en la



Capítulo III: Implementación y Prueba

investigación solo fueron usados los implementados por el framework, por tal motivo serán los descritos a continuación.

3.3.1 Patrones GRASP

Los patrones GRASP, acrónimo de General Responsibility Assignment Software Patterns en español Patrones Generales de Software para Asignación de Responsabilidades, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones; son una serie de buenas prácticas enfocadas a la calidad del software (85). Dentro de los utilizados en el desarrollo del sistema se encuentran los siguientes:

Experto: es uno de los patrones que más se utiliza cuando se trabaja con Symfony; es el principio básico de asignación de responsabilidades. Este patrón asigna la responsabilidad de la creación de un objeto o la implementación de un método, a la clase que tiene toda la información necesaria para crearlo o ejecutarlo, obteniendo un diseño con mayor cohesión y manteniendo la información encapsulada disminuyendo así el acoplamiento. (85) (89) Este patrón se aprecia en el sistema desarrollado por ejemplo en la clase “usuarioCorreo.php”, esta posee la información necesaria para cumplir con cada una de las responsabilidades que le corresponden.

Controlador: es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. La arquitectura MVC brinda una capa específicamente para los controladores, que son el núcleo de este, y especifica la presencia de este patrón. (89) Este patrón se puede evidenciar dentro del sistema desarrollado en las clases controladoras que posee Symfony2. Un ejemplo es la clase “usuarioCorreoController.php”.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación o instanciación de nuevos objetos o clases. Su utilización permite visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. (89) Por ejemplo, la clase “usuarioCorreoController.php” es la responsable de crear una nueva instancia de la entidad “usuarioCorreo.php”.



Capítulo III: Implementación y Prueba

Alta cohesión³⁵: soluciona el problema de “asignar una responsabilidad de manera que la cohesión permanezca alta”. (85) Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. A todas las clases les son asignadas responsabilidades con el objetivo de que trabajen en una misma área de aplicación y no tengan mucha complejidad.

Bajo acoplamiento³⁶: resuelve el problema de “asignar una responsabilidad de manera que el acoplamiento permanezca bajo”. Este patrón es un principio a tener en cuenta en todas las decisiones de diseño. (85) Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja. Un ejemplo es la clase “usuarioCorreoController.php”, le son asignadas responsabilidades de forma tal que solo se comunique con la entidad “usuarioCorreo.php”, garantizando un mínimo de dependencia, además del resto de las clases que no crean instancias de otras clases.

3.3.2 Patrones GOF

Los patrones GOF es la abreviación del grupo Gang of Four, compuesto por Erich Gamma, Richard Helm, Ralph Jhonson y John Vlisodes, quienes en su publicación “Design Patterns”, describen 23 patrones de diseño comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelamiento UML. Se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento. (85) El marco de trabajo Symfony2 implementa varios patrones, por lo cual se explicará el más significativo de cada una de las categorías antes mencionadas.

Abstract Factory³⁷: es un patrón de tipo creación³⁸, su objetivo es facilitar una interfaz para crear “familias” de objetos relacionados o dependientes sin que sea necesario especificar su clase. (90) Cuando el marco de trabajo necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea. Como la definición por defecto de

³⁵ El grado de cohesión mide la coherencia de una clase, lo coherente que es la información que almacena una clase con las responsabilidades y relaciones que ésta tiene con otras clases. (89)

³⁶ El grado de acoplamiento indica lo vinculadas que están unas clases con otras, lo que afecta un cambio en una clase a las demás y por tanto lo dependientes que son unas clases con otras. (89)

³⁷ Conocida en español como Factoría Abstracta.

³⁸ Este patrón resuelve problemas relativos a la creación de objetos.



Capítulo III: Implementación y Prueba

la factoría para las peticiones es “getRequest ()”, Symfony2 crea un objeto de esta clase para tratar con las peticiones.

Decorator³⁹: es un patrón de tipo estructura, ya que permite que clases y objetos sean utilizados para componer estructuras de mayor tamaño. Este patrón añade dinámicamente nuevas responsabilidades a un objeto. (91) Symfony 2 contienen un decorador que permite agregar funcionalidades dinámicamente a las aplicaciones desarrolladas bajo sus principios. Cada una de las vistas generadas hereda su diseño de la plantilla “plantilla.html.twig”, siendo esta la plantilla contenedora de la estructura y el diseño básico de las vistas.

Observer⁴⁰: conocido como el patrón Publish-Subscribe, es un patrón de diseño de comportamiento⁴¹, que define una relación de uno a muchos de tal manera que cuando un objeto cambia su estado, todos los objetos dependientes son notificados de la actualización automáticamente. (90) El componente “Event Dispatcher” de Symfony2 implementa este patrón evidenciándose, por ejemplo, cuando se llama al componente “HttpKernel” y este, una vez que el objeto “Response” ha sido creado, puede ser utilizado en otros componentes del sistema para modificarlo, por ejemplo, agregarle una cabecera de caché antes de que sea utilizado realmente. Para hacer esto posible el núcleo de Symfony2 lanza un evento “kernel.response”.

3.4 Tarjetas Clase-Responsabilidades-Colaboradores (CRC)

La metodología XP en lugar de utilizar diagramas para desarrollar modelos representa las clases mediante tarjetas. Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos. Cada tarjeta contiene el nombre de la clase colocado como título, en la parte izquierda se colocan las responsabilidades u objetivos y en la parte derecha las clases que colaboran con cada responsabilidad. (92)

Se obtuvieron un total de cinco tarjetas CRC, las que permitieron definir y simular los escenarios que garantizan el buen funcionamiento del diseño. A continuación se muestra el ejemplo de una de las tarjetas generadas en la investigación, el resto las puede consultar en el anexo 7.

³⁹ Conocida en español como Decorador.

⁴⁰ Conocida en español como Observador.

⁴¹ Este patrón resuelve problemas relativos a la interacción entre objetos.



Capítulo III: Implementación y Prueba

Tabla 8: CRC 1 (elaboración propia)

Clase: diaController	
Responsabilidades: <ul style="list-style-type: none">- Buscar vuelos por horas.- Generar pdf de los vuelos por horas.- Buscar pasajeros por horas.- Generar pdf de los pasajeros por horas.- Clasificar los días en ligero, normal o trabajoso mediante la inteligencia artificial.- Publicar servicio web.	Colaboradores: <ul style="list-style-type: none">sabanaclasificacion

3.5 Modelo de datos

Las bases de datos son un gran pilar de la programación actual, ya que permiten almacenar, organizar y manipular cantidades ingentes de datos con cierta facilidad. Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad.

En la Figura 7 se representa el modelo físico correspondiente a la base de datos de la propuesta de solución, el mismo fue generado por la herramienta CASE Visual Paradigm. Para observar las tablas con más detalles ver anexo 8.



Capítulo III: Implementación y Prueba

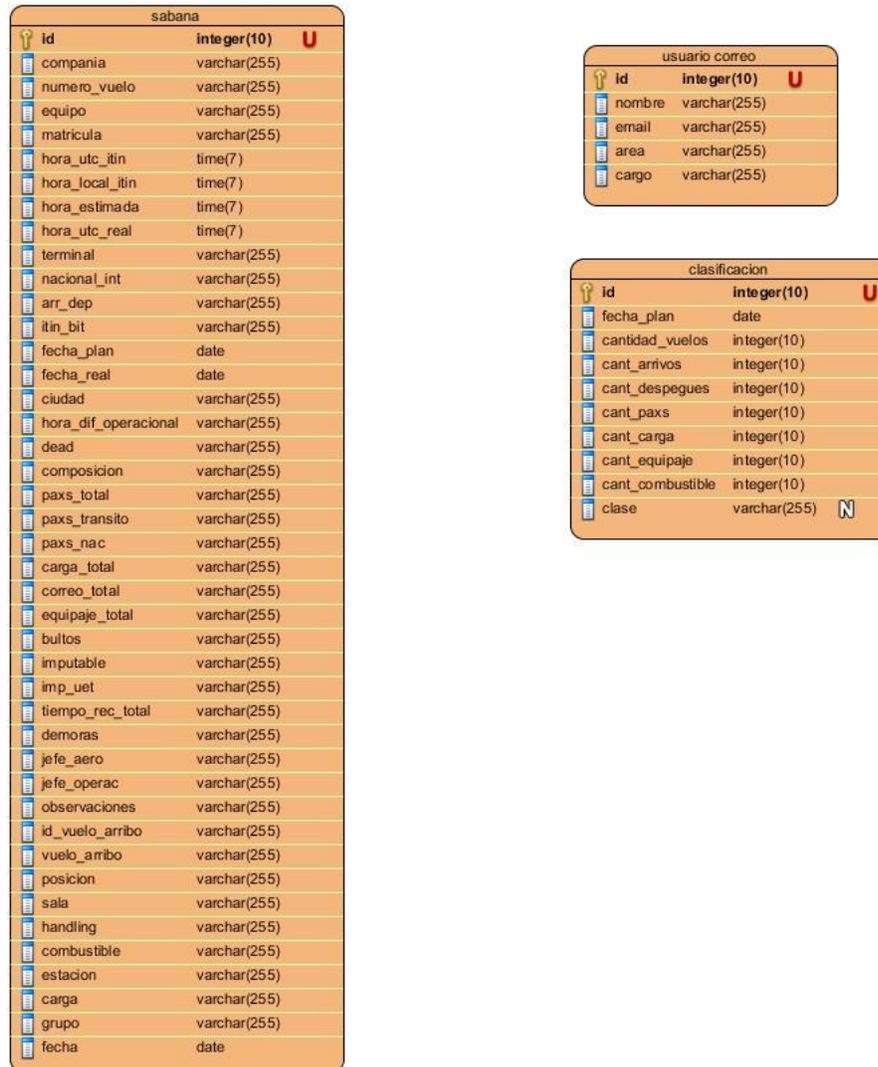


Figura 5: Modelo físico de la base de datos (elaboración propia)

A continuación se muestra una descripción de las tablas que conforma la base de datos:

sabana: almacena la información referente a todos los vuelos planificados y reales.

usuarioCorreo: se utiliza para almacenar los datos de los usuarios a los cuales se les envían los correos con los reportes.

clasificación: almacena la clasificación de cada uno de los días.



Capítulo III: Implementación y Prueba

3.6 Estilos y estándares de codificación

Es el conjunto de reglas o normas usadas para escribir código fuente y que incluye varios aspectos dentro del proceso de codificación (93). La metodología XP propone el uso de estándares de codificación, de modo que el código no sea conocido por una sola persona del equipo de desarrollo. Esto permite que ambos miembros del equipo puedan realizar cambios en el código, y así poder integrar de forma continua los nuevos cambios. (27)

A continuación se muestran los principales estándares de codificación utilizados en la implementación de la propuesta de solución.

- ✓ El código se encuentra tabulado a través del formato que aplica la combinación de teclas ALT+SHIFT+F del NetBeans IDE.
- ✓ Los comentarios de implementación son del tipo “//...” y los comentarios de documentación son del tipo “/** ... */”.
- ✓ Las líneas de código que debido a su tamaño ocupan más del espacio visible que ofrece el NetBeans IDE para escribir código (se refiere al ancho de la ventana maximizada), deben ser divididas en varias, preferiblemente después de una coma o un operador.
- ✓ Se utiliza la notación Camel⁴² con el estándar lowerCamelCase.
- ✓ Los nombres de las tablas son en singular y en minúscula.
- ✓ Todos los atributos de las clases del modelo son privados y el acceso a ellos se hace por medio de los métodos getAtributo⁴³ y/o setAtributo⁴⁴.
- ✓ Las variables deben ser explícitas, aunque se pueden usar abreviaturas siempre y cuando no violen este principio.

La utilización de un buen estilo de código influye positivamente en las labores de mantenimiento de un software como lo son la corrección de errores y el desarrollo de una nueva funcionalidad. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

⁴² Consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula: EndOfFile. Existen dos variantes:

-UpperCamelCase, CamelCase o PascalCase: en esta variante la primera letra también es mayúscula.

-lowerCamelCase, camelCase o dromedaryCase: la primera letra es minúscula.

⁴³ Método de acceso que permite obtener el valor de un atributo determinado.

⁴⁴ Método de acceso que permite cambiar el valor de un atributo determinado.



Capítulo III: Implementación y Prueba

3.7 Tratamiento de errores

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Son una forma clara para controlar los errores sin confundir el código con muchas instrucciones de control del error. Cuando se verifica un error se pone en marcha una excepción que, si se recibe enseguida permite gestionar un error. Para depurar los errores se utilizará JavaScript validando los formularios y evitando consultas a la base de datos sin sentido o que tenga malas intenciones como son las inyecciones SQL, garantizando además que los datos introducidos por los usuarios sean válidos o les sea posible corregirlos en caso contrario.

3.8 Tareas de ingeniería

Otro elemento que define la metodología XP es las tareas de ingeniería, son actividades que se elaboran para simplificar la programación de una HU, una HU puede tener uno o más tareas de ingeniería. Estas tareas son asignadas a los programadores para ser desarrolladas en la iteración que le corresponda.

Las HU se dividieron en 17 tareas de ingeniería, a continuación se muestran las tareas correspondientes a la HU Reportes diarios, el resto las puede consultar en el anexo 9.

Tabla 9: TI Vuelos por horas (elaboración propia)

Tarea de Ingeniería	
Número: 1	Número de HU: 1
Nombre: Vuelos por horas.	
Tipo de tarea: Desarrollo.	Estimación: 3 días.
Fecha de inicio: 2/2/2015	Fecha fin: 4/2/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se muestran la cantidad de vuelos por las horas del día en diferentes tipos de gráficos.	



Capítulo III: Implementación y Prueba

Tabla 10: TI Pasajeros por horas (elaboración propia)

Tarea de Ingeniería	
Número: 2	Número de HU: 1
Nombre: Pasajeros por horas.	
Tipo de tarea: Desarrollo.	Estimación: 2 días.
Fecha de inicio: 5/2/2015	Fecha fin: 6/2/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se muestran la cantidad de pasajeros por las horas del día en diferentes tipos de gráficos.	

3.9 Integración SATD con Weka

Para la integración se requiere conectar la base de datos del Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez (SATD) con Weka. Para ello se necesita:

- ✓ El conector java o driver para la conexión entre Weka y el SGBD PostgreSQL.
- ✓ Configurar los parámetros de conexión entre Weka y el SGBD PostgreSQL en el fichero DatabaseUtils.props en la carpeta del usuario del sistema.

Weka posee una Interfaz de Línea de Comando (CLI) que permite hacer llamadas a las clases Java definidas en la herramienta Weka.

PHP dispone de la función `exec()` que se encarga de ejecutar un comando en el `shell` del sistema operativo, dicha función recibe como parámetros el comando que será ejecutado, un arreglo con el valor de la ejecución del comando y una variable que almacenará el estado del retorno del comando. Los dos últimos parámetros son opcionales.

Haciendo una llamada a través de la función `exec()` al `shell` del sistema y ejecutando comandos de la CLI de Weka se integran el lenguaje PHP que utiliza SATD y el lenguaje Java de Weka.

Para clasificar el nuevo plan diario de vuelos la herramienta Weka genera un modelo de conocimiento analizando los datos almacenados en la base de datos. Esta clasificación se almacena para ser tenida en cuenta para futuras clasificaciones con Weka.



Capítulo III: Implementación y Prueba

3.10 Pruebas

William Howden expresó "(...) *probar es la parte inevitable de cualquier esfuerzo responsable por desarrollar un sistema de software.*" Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática, ocupando el mayor porcentaje del esfuerzo técnico en el proceso del software. (94)

Uno de los pilares de la metodología XP es el uso de pruebas, encaminadas a medir la calidad del software y el cumplimiento de las funcionalidades establecidas por el cliente, reduciendo de esta manera el número de errores no detectados. En esta metodología las pruebas se dividen en dos grupos: pruebas unitarias encargadas de verificar el código y pruebas de aceptación que están orientadas a probar las funcionalidades del sistema.

3.10.1 Pruebas unitarias

Pressman define como pruebas de unidad aquellas pruebas que se concentran en cada componente individual, asegurando que funciona de manera apropiada como unidad, o sea, no es posible probar una sola operación de manera aislada sino como parte de una clase. (79)

Al concluir cada iteración se realizaron las pruebas unitarias para garantizar el correcto funcionamiento del código. Para el desarrollo de las mismas se utilizó la herramienta PHPUnit desarrollado sobre el lenguaje PHP y que se encuentra perfectamente integrado con Symfony 2.5. En este marco de trabajo las pruebas unitarias se ubican en el directorio "Test" nombrándose "NombreClaseTest".

En la siguiente imagen se muestra un ejemplo de la ejecución de 4 pruebas en la primera iteración, las cuales arrojaron los siguientes resultados:

```
FAILURES!  
Tests: 4, Assertions: 1, Errors: 3.
```

Figura 6: Pruebas unitarias con PHPUnit (error)

Al ejecutar las pruebas unitarias se encontraron varios errores como valores nulos y posiciones fuera de índice. Luego de corregir los principales problemas encontrados se ejecutaron nuevamente las pruebas y se obtuvo un resultado satisfactorio.



Capítulo III: Implementación y Prueba

```
Configuration read from /var/www/html/aero/app/phpunit.xml.dist
....
Time: 45 ms, Memory: 2.50Mb
OK (4 tests, 4 assertions)
```

Figura 7: Pruebas unitarias con PHPUnit (correcto)

3.10.2 Pruebas de aceptación

Las pruebas de aceptación o pruebas funcionales, las especifica el cliente y se enfocan en las características generales y la funcionalidad del sistema, elementos visibles y revisables por el cliente. Las pruebas de aceptación se derivan de las HU que se han implementado como parte de un lanzamiento de software. (94)

El objetivo específico de esta prueba es garantizar que los requerimientos han sido cumplidos y que el sistema ha sido aceptado. Se realizaron un total de 18 casos de pruebas de aceptación definidas para el Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez. (Ver anexo 10).

Casos de Prueba de la primera iteración

Tabla 11: CP Vuelos por hora (elaboración propia)

Caso de prueba de aceptación	
Código: HU1_CP1	Número de HU: 1,2
Nombre: Mostrar vuelos por horas.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de vuelos por horas del día.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
Entrada/ Pasos de ejecución: <ol style="list-style-type: none">1. El administrador o invitado selecciona la opción: Días.2. El administrador o invitado escoger la opción que desea: Vuelos por horas. Pasajeros por horas. (Ver prueba HU1_CP2).3. El administrador o invitado selecciona la opción: Vuelos por horas.	



Capítulo III: Implementación y Prueba

<ol style="list-style-type: none"> 4. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. 5. Se muestra la información de la cantidad de vuelos por horas del día en diferentes tipos de gráficos, la cantidad total de vuelos en el día y la clasificación del día. 6. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9).
<p>Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de vuelos por horas del día.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 12: CP Pasajeros por hora (elaboración propia)

Caso de prueba de aceptación	
Código: HU1_CP2	Número de HU: 1
Nombre: Mostrar pasajeros por horas.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de pasajeros por horas del día.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
Entrada/ Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El administrador o invitado selecciona la opción: Días. 2. El administrador o invitado escoger la opción que desea: Vuelos por horas (Ver prueba HU1_CP1). Pasajeros por horas. 3. El administrador o invitado selecciona la opción: Pasajeros por horas. 4. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. 5. Se muestra la información de la cantidad de pasajeros por horas del día en diferentes tipos de gráficos, mostrándose los arribos y despegues y la clasificación del día. 6. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9). 	



Capítulo III: Implementación y Prueba

Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de pasajeros por horas del día.
--

Evaluación de la prueba: Prueba satisfactoria.

3.10.3 Resultados de las pruebas de aceptación

Todas las pruebas de aceptación se realizaron de conjunto con el cliente. Cada iteración ejecutada concluyó con la realización de un conjunto de pruebas de aceptación para verificar el cumplimiento de los requisitos. Las no conformidades detectadas en cada iteración fueron mitigadas antes de dar paso a una nueva iteración.

Para la **primera iteración** se realizaron dos casos de pruebas de aceptación, identificando tres no conformidades, dos no significativas y una significativa. Para la **segunda iteración** se realizaron seis casos de pruebas de aceptación, identificando cuatro no conformidades, dos no significativas y dos significativas. Para la **tercera iteración** se realizaron siete casos de pruebas de aceptación, identificando seis no conformidades, cuatro no significativas y dos significativas. Para la **cuarta iteración** se realizaron dos casos de pruebas de aceptación, identificando dos no conformidades, una no significativa y una significativa. Debido a la complejidad mínima las no conformidades, fueron todas resueltas y no quedaron casos de pruebas pendientes para las siguientes iteraciones. Para la **quinta iteración** se realizó un solo caso de prueba de aceptación, identificando dos no conformidades, de ellas una significativa y una no significativa. Estas dos no conformidades se solucionaron dando fin al proceso de pruebas de aceptación y obteniendo resultados satisfactorios. (Ver tabla 12 y figura 9)

Las no conformidades, no significativas, se centraron en errores ortográficos como omisiones de tildes, paréntesis, cambio de mayúscula por minúscula, y las significativas, en errores de validación, en campos de selección y cambios en el diseño.



Capítulo III: Implementación y Prueba

Tabla 13: Resultado de las pruebas de aceptación por iteración (elaboración propia)

	Iteración # 1	Iteración # 2	Iteración # 3	Iteración # 4	Iteración # 5
CP de Aceptación	2	6	7	2	1
No conformidades	3	4	6	2	2
Significativas	1	2	2	1	1
No significativas	2	2	4	1	1
Resueltas	3	4	6	2	2
Pendientes	0	0	0	0	0

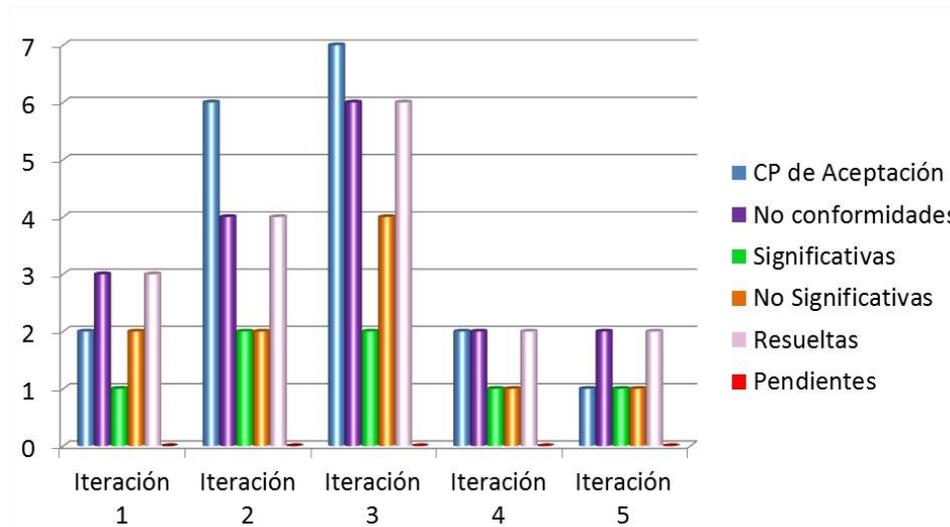


Figura 8: Resultado de las pruebas de aceptación por iteración (elaboración propia)

Después de terminada la aplicación se ejecutaron pruebas dirigidas al sistema en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Durante las tres iteraciones realizadas, se detectaron un total de cinco no conformidades, quedando todas resueltas, como se muestra en la figura 10.



Capítulo III: Implementación y Prueba

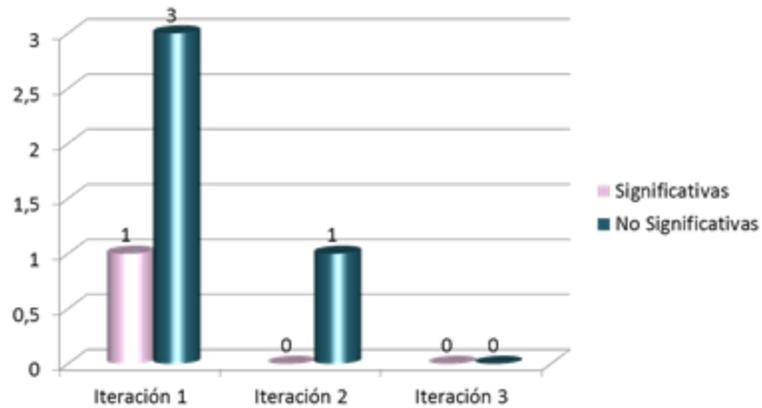


Figura 9: Resultados de las pruebas dirigidas al sistema en su totalidad (elaboración propia)

3.10.4 Evaluación del proceso de clasificación

Al aplicar el algoritmo de clasificación “J48” a una muestra de 1846 días clasificados cualitativamente como ligeros (l), normal (n) o trabajosos (t), se obtuvo un árbol de decisión que permite pronosticar el comportamiento del aeropuerto para los próximos días, teniendo en cuenta datos históricos con la información de los vuelos diarios.



Capítulo III: Implementación y Prueba

=== Summary ===

```
Correctly Classified Instances      1845
Incorrectly Classified Instances      1
Kappa statistic                     0.9991
Mean absolute error                 0.0004
Root mean squared error             0.019
Relative absolute error              0.0861 %
Root relative squared error         4.1508 %
Coverage of cases (0.95 level)     99.9458 %
Mean rel. region size (0.95 level)  33.3333 %
Total Number of Instances           1846
```

99.9458 %
0.0542 %

Por ciento de instancias
clasificadas correctamente

Por ciento de instancias
clasificadas incorrectamente

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,998	0,000	1,000	0,998	0,999	0,999	0,999	0,999	n
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	t
	1,000	0,001	0,997	1,000	0,999	0,998	1,000	0,997	l
Weighted Avg.	0,999	0,000	0,999	0,999	0,999	0,999	1,000	0,999	

=== Confusion Matrix ===

```
 a  b  c  <-- classified as
560  0  1  | a = n
  0 892  0  | b = t
  0  0 393 | c = l
```

n- normal
t- trabajoso
l- ligero

Figura 10: Evaluación del proceso de Clasificación y del Pronóstico realizado por el algoritmo “J48”

En la figura 10 se muestra los resultados de Weka, donde se puede apreciar que el algoritmo utilizado J48 clasificó incorrectamente 1845 instancias de las 1846 evaluadas, por tanto se refiere que clasificó correctamente el 99.9458 % de las mismas, clasificando una sola instancia incorrecta lo que equivale a un 0.0542 %.

3.11 Conclusiones del capítulo

En el presente capítulo se crearon cinco tarjetas CRC, las cuales brindaron claridad en aspectos importantes como las principales funcionalidades que presentan las clases y la relación existente entre ellas. Se hace alusión a las etapas de implementación y prueba de la herramienta en desarrollo, donde se realizaron 17 tareas de ingeniería para dar solución a las HU planteadas. Se describe el proceso de prueba, uno de los más importantes para garantizar el éxito de la aplicación, éste se llevó a cabo con el objetivo de brindarle al cliente un producto funcional que cumpla con sus necesidades especificadas.



Conclusiones generales

Con la realización del presente trabajo de diploma se desarrolló la versión 1.0 del Sistema de Apoyo a la Toma de Decisiones del aeropuerto Juan Gualberto Gómez. Por lo tanto los objetivos que se plantearon fueron cumplidos de manera satisfactoria obteniendo una serie de resultados:

- ✓ Los métodos científicos empleados confirmaron su validez, al permitir el desarrollo de la teoría que sustenta la investigación y la elaboración de las definiciones que facilitaron comprender el problema de la investigación. El estudio realizado permitió definir las herramientas que posibilitaron una base sólida para el desarrollo del sistema y seleccionar como metodología para la guía del proceso a XP, la que posibilitó generar los artefactos fundamentales para el desarrollo de la aplicación.
- ✓ El diseño desarrollado para la aplicación, permitió la implementación de funcionalidades que dieron solución al objetivo general de la investigación.
- ✓ Las herramientas y lenguajes de programación seleccionados permitieron el desarrollo de la solución propuesta.
- ✓ Las pruebas realizadas a la solución permitieron asegurar la calidad y el correcto funcionamiento de la aplicación, cumpliendo con las expectativas del cliente.



Recomendaciones

Tomando como base la investigación realizada y el análisis de los resultados obtenidos se recomienda:

- ✓ Implementar un módulo en la aplicación para trabajar con diferentes bases de datos logrando su uso en otros aeropuertos.
- ✓ Desarrollar aplicaciones nativas para dispositivos móviles que consuman el servicio web de esta aplicación para un funcionamiento más óptimo.



Referencias Bibliográficas

1. FRANKLIN, Enrique Benjamín and KRIEGER, Mario. Toma de decisiones empresariales. Reseña de “Comportamiento organizacional, enfoque para América Latina.” . 2011. Vol. 6, no. 11, p. 113–120.
2. Transporte aereo en Cuba. *Hicuba.com* [online]. [Accessed 2 May 2015]. Available from: <http://www.hicuba.com/aereo.htm>
3. Aduana. *Aduana-Cuba* [online]. [Accessed 2 May 2015]. Available from: http://www.aduana.co.cu/index.php?option=com_content&view=article&id=51&Itemid=204&lang=esAduana General de la República de Cuba
4. Diccionario de la lengua española. [online]. 2014. [Accessed 13 January 2015]. Available from: <http://lema.rae.es/drae/?val=decisi%C3%B3n>
5. VERCELLIS, Carlo. *Business Intelligence: Data Mining and Optimization for Decision Making*. John Wiley & Sons Ltd, 2009. ISBN 9780470511381.
6. VERGARA, Leonor Cabeza de, SANTIAGO, Alberto Elías Muñoz and SANTIS, Sandra Milena Vivero. Aproximación al proceso de toma de decisiones en la empresa barranquillera. . December 2004. No. 17, p. 1–38.
7. SCHEIN, Erik de Haan. *Fearless Consulting: Temptations, Risks and Limits of the Profession*. John Wiley & Sons Ltd, 2006. ISBN 9780470026953.
8. *Forum Guide to Decision Support Systems: A Resource for Educators*. ED Pubs, 2006.
9. POWER, Daniel. *Decision Support Systems: Frequently Asked Questions*. iUniverse, 2004. ISBN 0595339719.
10. MCLEOD, Raymond. *Sistemas de información gerencial*. Pearson Educación, 2000. ISBN 9701702557.
11. RAUCH-HINDIN, Wendy B. *Aplicaciones de la inteligencia artificial en la actividad empresarial, la ciencia y la industria*. Ediciones Díaz de Santos, 1989. ISBN 9788487189074.
12. Diccionario de la lengua española. [online]. 2014. [Accessed 13 January 2015]. Available from: <http://lema.rae.es/drae/?d=drae&val=inteligencia+artificial&x=0&y=0>
13. TABEADA, Carolina García. *Inteligencia artificial*. . 2006.
14. Diccionario de la lengua española. [online]. [Accessed 12 March 2015]. Available from: <http://lema.rae.es/drae/?d=drae&val=hardware&x=0&y=0>



15. SANZ, Dersu García. NewsMonitor: Desarrollo de un sistema de seguimiento de noticias. *Serie Científica* [online]. 2013. [Accessed 4 February 2015]. Available from: <http://addi.ehu.es/handle/10810/10179>
16. SERRANO, Alberto García. *Inteligencia artificial: fundamentos, práctica y aplicaciones*. RC Libros, 2012. ISBN 9788493945022.
17. TORRADO, Mercedes. parte 1: *Taller Minería de datos aplicados a la educación*. Universidad de Barcelona, 2011.
18. LÓPEZ, César Pérez and DANIEL SANTÍN GONZÁLEZ. *Minería de datos: técnicas y herramientas*. Editorial Paraninfo, 2007. ISBN 9788497324922.
19. TOMÁS ALUJA BANET. La minería de datos, entre la estadística y la inteligencia artificial. *Questiò: Quaderns d'Estadística, Sistemes, Informàtica i Investigació Operativa*. 2001. Vol. 25, no. 3, p. 479–498.
20. SILVENTE, Vanesa Berlanga, HURTADO, María José Rubio and BAÑOS, Ruth Vilà. Cómo aplicar árboles de decisión en SPSS. [online]. Vol. 6, no. 1. [Accessed 5 January 2015]. DOI 10.1344/reire2013.6.1615. Available from: <http://revistes.ub.edu/ice/reire.html>
21. MARTÍNEZ, Guillermo Roberto Solarte and MEJÍA, José A. Soto. Árboles de decisiones en el diagnóstico de enfermedades cardiovasculares. *Scientia et Technica*. 2011. Vol. 3, no. 49, p. 104–109.
22. POWER, Daniel J. Ask Dan! about DSS - What are some classic business DSS? *Decision Support Systems* [online]. 9 April 2011. [Accessed 3 February 2015]. Available from: <http://dssresources.com/faq/index.php?action=artikel&id=205>
23. IBM Cognos software. [online]. 19 January 2015. [Accessed 3 February 2015]. Available from: <http://www.cognos.com>
24. Analytics, Movilidad y Seguridad corporativas | MicroStrategy. [online]. [Accessed 3 February 2015]. Available from: <http://www.microstrategy.com/es/>
25. LLANES, Kathrin Rodriguez. Sistema Informático Inteligente de apoyo a la toma de decisiones en la FAR. *Serie Científica* [online]. 2010. Vol. 3, no. 8. [Accessed 3 January 2015]. Available from: <http://publicaciones.uci.cu/index.php/SC/article/view/366>
26. CORALES, Yovannys Sánchez. *Componente para la Toma de Decisiones Clínicas del Sistema Integral para la Atención Primaria de Salud*. Maestría. Habana : Universidad de las Ciencias Informáticas, 2013.
27. LETELIER, Patricio and PENADÉS, Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [online]. 2006. Vol. 5, no. 26. [Accessed 19 May 2015]. Available from: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm



28. FIGUEROA, Roberth G., SOLÍS, Camilo J. and CABRERA, Armando A. *Metodologías Ágiles Vs Metodologías Tradicionales*. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, [no date].
29. DUARTE, Ailin Orjuela and ROJAS, Mauricio. las metodologías de desarrollo ágil como una oportunidad para la ingeniería de software educativo. *Avances en Sistemas e Informática* [online]. 2009. Vol. 5, no. 2. [Accessed 22 May 2015]. Available from: <http://www.revistas.unal.edu.co/index.php/avances/article/view/10037>
30. RIOLA, Jose Carlos Carvajal. *Metodologías Ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial*. Barcelona, 2008.
31. RUBIN, Kenneth S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley, 2012. ISBN 9780321700377.
32. KNIBERG, Henrik. *Scrum y XP desde las trincheras*. 2007. ISBN 978-1-4303-2264-1.
33. CANTÓ, Carles Escrivà, ESQUEMBRE, Rubén Durá, MARTÍNEZ, Antonio Mudarra and CHINCHILLA, Jorge Lilao. *Análisis y Especificación de Sistemas Multimedia*. 2012.
34. Lenguaje de programación. *Definicion.org* [online]. [Accessed 30 April 2015]. Available from: <http://www.definicion.org/lenguaje-de-programacion>
35. PHP: Hypertext Preprocessor. *php* [online]. [Accessed 29 May 2015]. Available from: <http://php.net/>
36. Introducción a SOA y servicios web. *IBM developerWorks* [online]. 2007. [Accessed 31 May 2015]. Available from: <http://www.ibm.com/developerworks/ssa/webservices/newto/service.html>
37. FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures* [online]. California : Universidad de California, Irvine, 2000. [Accessed 31 May 2015]. Available from: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
38. FRANGANILLO, Jorge. *HtmI5: el nuevo estándar básico de la Web*. 2011.
39. GAUCHAT, Juan Diego. *El gran libro de HTML5, CSS3 y Javascript* [online]. Marcombo, 2012. [Accessed 16 March 2015]. ISBN 8426717829. Available from: http://books.google.com.co/books?id=szDMIRzwzuUC&dq=html5&hl=es&source=gbs_navlinks_s
40. All CSS specifications. *W3C* [online]. [Accessed 6 February 2015]. Available from: <http://www.w3.org/Style/CSS/specs>
41. GARCÍA, Carlos Egea. *DISEÑO WEB PARA TOD@S II*. Editorial Icaria, 2008. ISBN 9788474269574.
42. FREE SOFTWARE FOUNDATION. *gnu.org*. *GNU Operating System* [online]. [Accessed 31 May 2015]. Available from: <http://www.gnu.org/licenses/lgpl.html>



43. wkhtmltopdf. *WK<html>topdf* [online]. [Accessed 31 May 2015]. Available from: <http://wkhtmltopdf.org/>
44. CHAUX, Hernando Recaman and ALARCÓN, Carlos Andrés Guerrero. Marco de trabajo para aplicaciones web de código abierto en instituciones universitarias. *Memorias*. 2012. Vol. 10, no. 18, p. 62–72. DOI 10.16925/me.v10i18.54.
45. MUÑOZ, Vicente Javier Eslava. *El nuevo PHP. Conceptos avanzados*. Vicente Javier Eslava Muñoz, 2013. ISBN 9788468644349.
46. MANGER, Christian, TREJDEROWSKI, Tomasz and PADUCH, Jarosław. Advantages and disadvantages of framework programming with reference to Yii php framework, gideon .net framework and other modern frameworks. *Studia Informatica*. 2010. Vol. 31, no. 4A, p. 119–137.
47. EGUILUZ, Javier. *Desarrollo web ágil con Symfony2*. 2013.
48. PHP: PHP 5.3.0 Release Announcement. *php* [online]. [Accessed 31 May 2015]. Available from: http://php.net/releases/5_3_0.php
49. THORNTON, Jacob and OTTO, Mark. Bootstrap 3, el manual oficial. *LibrosWeb* [online]. 2011. [Accessed 13 March 2015]. Available from: https://librosweb.es/libro/bootstrap_3/
50. ÁLVAREZ, Miguel Angel. Introducción a jQuery. *DesarrolloWeb.com* [online]. 2009. [Accessed 13 March 2015]. Available from: <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>
51. jQuery. *JQuery* [online]. [Accessed 13 March 2015]. Available from: <http://jquery.com/>
52. Base de Datos. *Universidad de Belgrano* [online]. [Accessed 8 February 2015]. Available from: <http://www.ub.edu.ar/catedras/ingenieria/Datos/contenidos.htm>
53. COBO, Ángel, GÓMEZ, Patricia, PÉREZ, Daniel and ROCHA, Rocío. *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos. España, 2005. ISBN 8479787066.
54. GARCÍA, o Pérez. *Desarrollo de herramientas web de gestión docente*. [online]. Universidad Politécnica de Cartagena, 2007. [Accessed 28 April 2015]. Available from: <http://repositorio.bib.upct.es/dspace/bitstream/10317/179/1/pfc2475.pdf>
55. FOGEL, Karl. Concesión de licencias dual. In : *Producir Software de Código Abierto* [online]. [no date]. ISBN 0-596-00759-0. Available from: <http://producingoss.com/es/dual-licensing.html>
56. ORACLE CORPORATION. MySQL. *MySQL* [online]. [Accessed 29 May 2015]. Available from: <http://www.mysql.com/about/legal/>
57. PostgreSQL. *PostgreSQL* [online]. 2014. [Accessed 4 February 2015]. Available from: <http://www.postgresql.org/>



58. CIBERNETIA. Conceptos básicos del servidor web. *Cibernetia* [online]. [Accessed 28 April 2015]. Available from: http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php
59. Usage Statistics and Market Share of Web Servers for Websites. *Web Technology Surveys* [online]. 2015. [Accessed 29 May 2015]. Available from: http://w3techs.com/technologies/overview/web_server/all
60. The Apache HTTP Server Project. *Apache* [online]. [Accessed 6 February 2015]. Available from: http://httpd.apache.org/ABOUT_APACHE.html
61. Apache Now the Leader in SSL Servers. *Netcraft* [online]. [Accessed 29 May 2015]. Available from: http://news.netcraft.com/archives/2006/04/26/apache_now_the_leader_in_ssl_servers.html
62. FREE SOFTWARE FOUNDATION. El problema de la licencia BSD. *gnu* [online]. 2015. [Accessed 8 June 2015]. Available from: <http://www.gnu.org/philosophy/bsd.es.html>
63. NEDELICU, Clement. *Nginx HTTP Server*. Packt Publishing Ltd, 2010. ISBN 9781849510875.
64. nginx documentation. [online]. [Accessed 29 May 2015]. Available from: <http://docs.prod.uci.cu/online/Nginx.docset/Contents/Resources/Documents/nginx.org/en/docs/index.html>
65. Internet Information Services. *Microsoft* [online]. 2007. [Accessed 28 April 2015]. Available from: <http://www.microsoft.com/spain/windowsserver2003/technologies/Webapp/iis.mspx>
66. Qué es un entorno de desarrollo integrado, IDE. [online]. [Accessed 16 March 2015]. Available from: <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>
67. RODRÍGUEZ, Carlos. Gestionar los proyectos y archivos haciendo uso de PHPStorm. *Joomla* [online]. 2014. [Accessed 31 May 2015]. Available from: <http://magazine.joomla.org/es/ediciones-antteriores/mayo-2014/item/2091-desarrollo-practico-3-proyecto>
68. PhpStorm Licensing and Renew. *jetbrains* [online]. [Accessed 31 May 2015]. Available from: <https://www.jetbrains.com/phpstorm/buy/?PS5PR>
69. PHP IDE JetBrains PhpStorm. *jetbrains* [online]. [Accessed 30 May 2015]. Available from: <https://www.jetbrains.com/phpstorm/?fromFooter>
70. Definición de Plugin. *alegsa.com.ar Diccionario de informática y tecnología* [online]. [Accessed 31 May 2015]. Available from: <http://www.alegsa.com.ar/Dic/plugin.php>
71. Conoce el nuevo plugin de Symfony2 para PHPStorm. *symfony.es* [online]. [Accessed 30 May 2015]. Available from: <http://symfony.es/noticias/2014/08/25/conoce-el-nuevo-plugin-de-symfony2-para-phpstorm/>
72. Help - Eclipse Platform. [online]. 2004. [Accessed 29 April 2015]. Available from: <http://help.eclipse.org/juno/index.jsp>



73. Symphony Plugin. *eclipse market place* [online]. [Accessed 31 May 2015]. Available from: <https://marketplace.eclipse.org/content/symfony-plugin>
74. Definición de zip. *Definición.de* [online]. [Accessed 8 June 2015]. Available from: <http://definicion.de/zip/>
75. ORACLE CORPORATION. NetBeans. [online]. 2012. [Accessed 8 February 2015]. Available from: <http://www.netbeans.org>
76. BERGMANN, Sebastian. PHPUnit Manual. *phpunit.de* [online]. 2015. [Accessed 1 May 2015]. Available from: <https://phpunit.de/manual/current/en/>
77. Libro HERRAMIENTAS CASE. [online]. [Accessed 13 March 2015]. Available from: http://www.academia.edu/4513393/Libro_HERRAMIENTAS_CASE
78. CA ERwin Data Modeling. *CA ERwin Data Modeler* [online]. 2013. [Accessed 28 April 2015]. Available from: <http://erwin.com/worldwide/spanish-latin-america>
79. Visual Paradigm y UML. [online]. [Accessed 8 February 2015]. Available from: <http://www.visual-paradigm.com/>
80. CUBERO, Juan Carlos and BERZAL, Fernando. *Herramientas de minería de datos. Introducción a KNIME*. [online]. Universidad de Granada, 2005. [Accessed 5 April 2015]. Available from: <http://elvex.ugr.es/decsai/intelligent/workbook/D1%20KNIME.pdf>
81. RapidMiner. *RapidMiner* [online]. [Accessed 21 March 2015]. Available from: <https://rapidminer.com/>
82. RAMESH, V., PARKAVI, P. and YASODHA, P. Performance Analysis of Data Mining Techniques for Placement Chance Prediction. [online]. 2011. Vol. 2, no. 8. [Accessed 12 May 2015]. Available from: <http://www.ijser.org/paper/Performance-Analysis-of-Data-Mining-Techniques-for-Placement-Chance-Prediction.html>
83. Java Virtual Machine. *Java Virtual Machine* [online]. [Accessed 29 May 2015]. Available from: <http://java-virtual-machine.net/>
84. GARZON, paula Andrea Vizcaino. *Aplicación de técnicas de inducción de árboles de decisión a problemas de clasificación mediante el uso de Weka* [online]. Bogotá, Colombia : Fundación Universitaria Konrad Lorenz, 2008. [Accessed 31 May 2015]. Available from: http://www.konradlorenz.edu.co/images/stories/suma_digital_sistemas/2009_01/final_paula_andrea.pdf
85. LARMAN, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Pearson Educación, 2003. ISBN 9788420534381.
86. JOSKOWICZ, José. *Reglas y Prácticas en eXtreme Programming*. 2008.



Referencias Bibliográficas

87. BAHIT, Eugenia. *El paradigma de la Programación Orientada a Objeto en PHP y el Patrón de Arquitectura de Software MVC*. 2011.
88. GRACIA, Joaquin. Patrones de diseño. Análisis y Diseño. Ingeniería del Software. [online]. 2005. [Accessed 11 March 2015]. Available from: <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>
89. CARMONA, Juan García. *Solid y GRASP. Buenas prácticas hacia el éxito en el desarrollo de software*. 2012.
90. *Patrones del "Gang of Four"* [online]. Universidad Politécnica de Madrid, 2012. [Accessed 17 March 2015]. Available from: http://is.ls.fi.upm.es/docencia/proyecto/docs/patrones_gof.pdf
91. SWEAT, Jason E. *php|architect's Guide to PHP Design Patterns* [online]. Canada, 2005. [Accessed 14 March 2015]. ISBN 0-9735898-2-5. Available from: <http://www.phparch.com/books/phparchitects-guide-to-php-design-patterns/>
92. GÓMEZ, Alveiro Rosado, DUARTE, Alexander Quintero and GÜEVARA, Cesar Daniel Meneses. Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio UFPSO*. 2014. Vol. 5, no. 1, p. 24–29.
93. CALLEJA, Manuel Arias. *Estándares de codificación* [online]. 2007. [Accessed 29 April 2015]. Available from: <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>
94. PRESSMAN, Roger S. *Ingeniería del software. Un enfoque práctico*. 6. 2005, [no date]. ISBN 9701054733.

Anexos

Anexo 1: Entrevista realizada al Ing. Manuel de Jesús Vázquez Garriga coordinador general del aeropuerto Juan Gualberto Gómez.

- ¿Existe algún software que le ayude a los directivos en la toma de decisiones con respecto al estado operativo del aeropuerto?

R/ No, existe un software llamado Sistema de Operaciones que recoge todas las informaciones por las diferentes áreas del aeropuerto.

- ¿Conoce de algún aeropuerto que utilice algún software para esto?

R/ No. Incluso es uno de los temas que se debaten en nuestros encuentros nacionales.

- ¿Le interesaría tener un software que le ayude en la toma de decisiones con respecto al estado operativo del aeropuerto?

R/ Creemos que es fundamental y muy necesario, sentimos que en ese punto estamos atrasados con respecto al mundo.

- ¿Qué funcionalidades debería incluir este software?

R/ Debería generar reportes diarios, semanales, mensuales y anuales, poder exportarlos a pdf. Además que se envíen correos a los directivos con un pronóstico del día siguiente y el comportamiento del día, si será ligero, normal o trabajoso según el trabajo planeado para ese día.

- ¿Quiénes tendrían acceso a este software?

R/ Tendrían acceso al software solamente los directivos.

- ¿Qué seguridad debería tener esta aplicación?

R/ El coordinador general del aeropuerto es el único que puede decidir a quién enviar los reportes.

- ¿Con qué tecnología cuenta la entidad para soportar el software?

R/ Es una entidad que cuenta con altas tecnologías, pero no conozco exactamente con qué tipo de tecnología tratamos.

Anexo 2: Guía de observación para el proceso de investigación.

- ✓ Estudio de las soluciones similares
 - Para analizar cómo se emplean los diferentes Sistemas de Apoyo a la Toma de Decisiones a nivel mundial y cuáles sirven de base para la investigación.



- ✓ Selección de la técnica de inteligencia artificial y análisis de los resultados
 - Se selecciona la técnica de inteligencia artificial con el propósito de realizar un análisis de los datos del aeropuerto, de forma tal que facilite la toma de decisiones respecto al estado operativo del aeropuerto Juan Gualberto Gómez.
- ✓ Desarrollo de la investigación
 - Observar si las funcionalidades implementadas cumplen con los artefactos elaborados.
- ✓ Resultados de la investigación
 - Se aprecia el grado de complejidad de las no conformidades detectadas y cómo erradicarlas.

Anexo 3: Guía de observación para los procesos de apoyo a la toma de decisiones en el aeropuerto Juan Gualberto Gómez.

- ✓ Tecnología existente en la entidad para soportar la solución que se propone.
- ✓ Forma en que los directivos del aeropuerto analizan la información para la toma de decisiones.
- ✓ Estructura existente en la red de la entidad para garantizar la seguridad de la información que se procesa en ese nivel.
- ✓ Decisiones tomadas por los directivos sobre el estado operativo del aeropuerto.

Anexo 4: Estadísticas del uso de servidores web según W3Techs

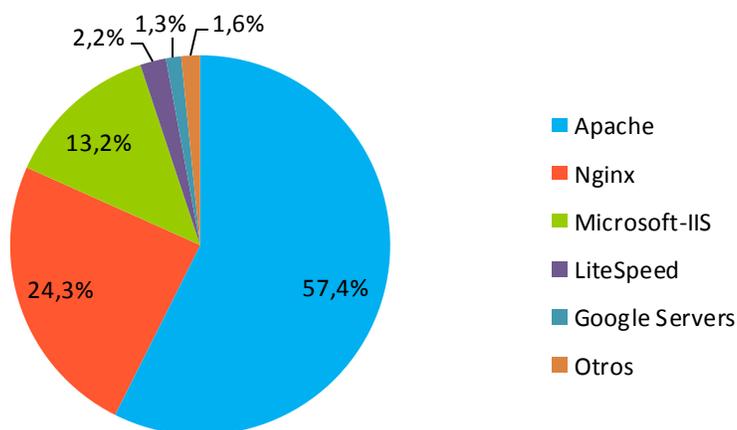


Figura 11: Estadísticas del uso de servidores web según W3Techs en el año 2015

Anexo 5: Algoritmos de clasificación en Weka

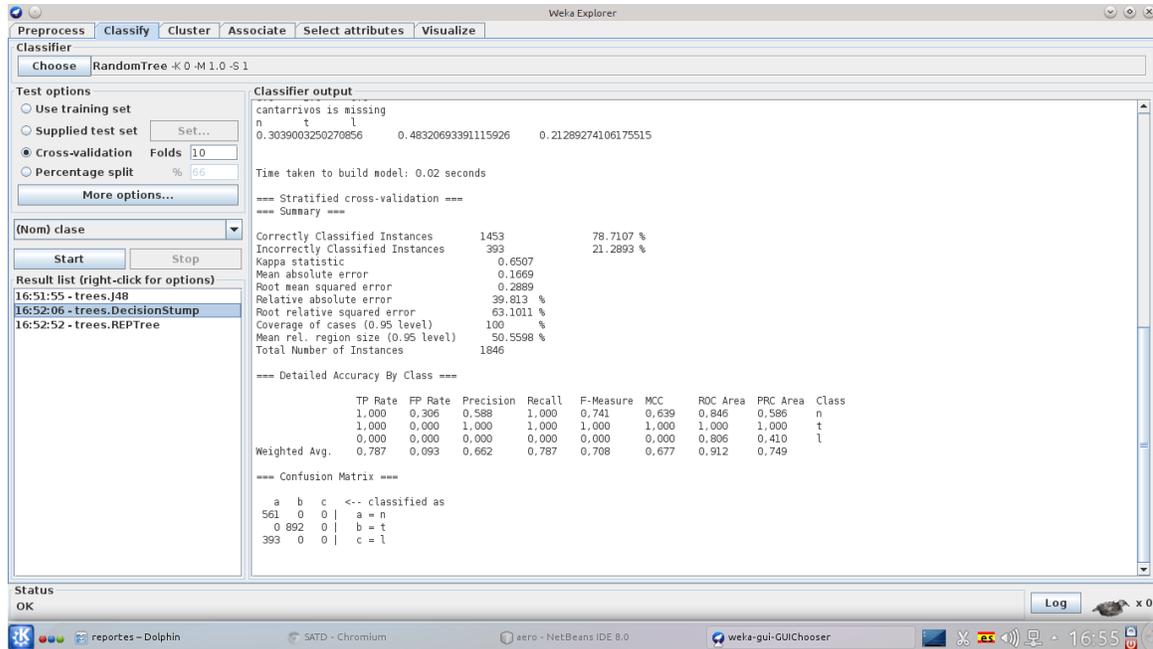


Figura 12: Algoritmo DecisionStump

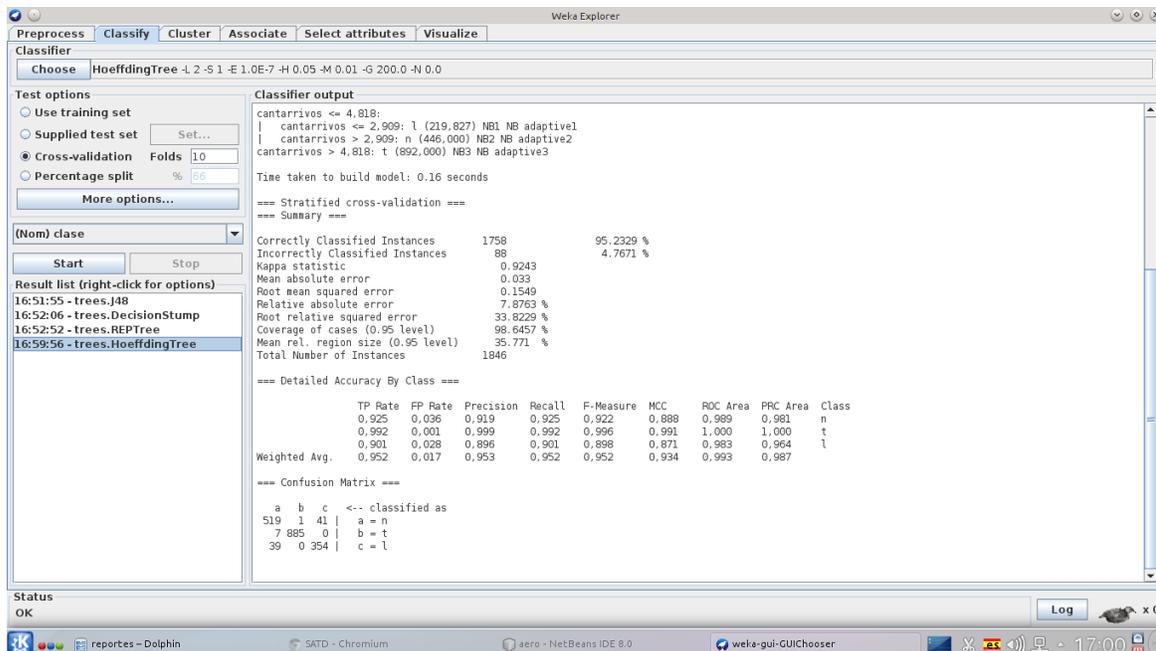


Figura 13: Algoritmo HoeffdingTree

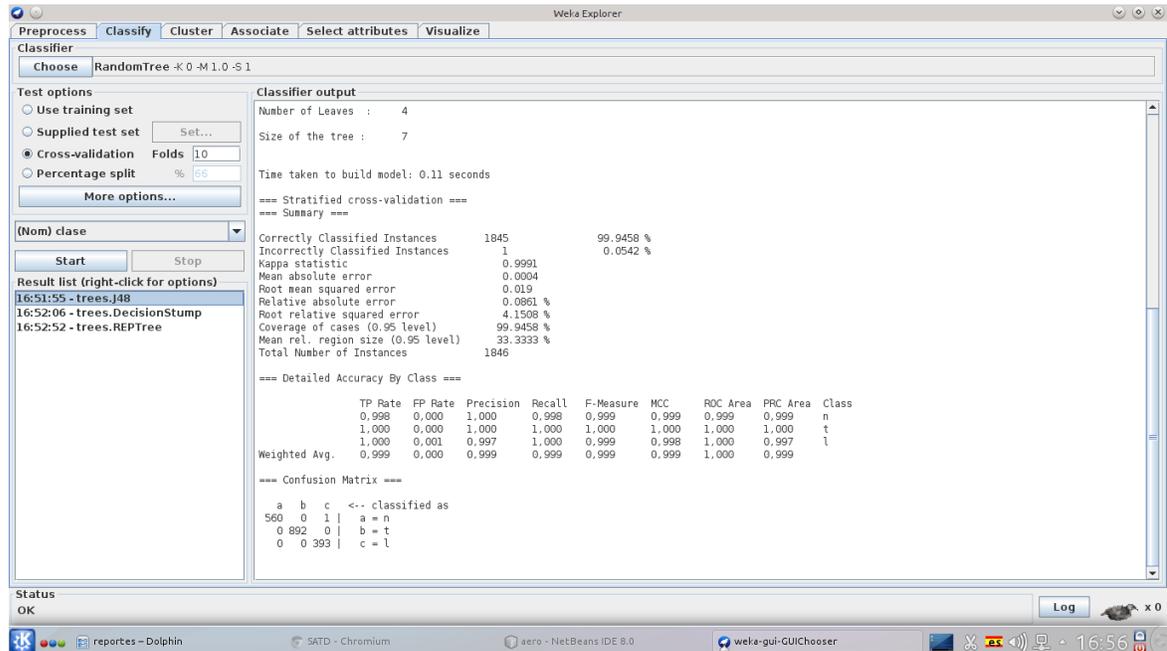


Figura 14: Algoritmo J48

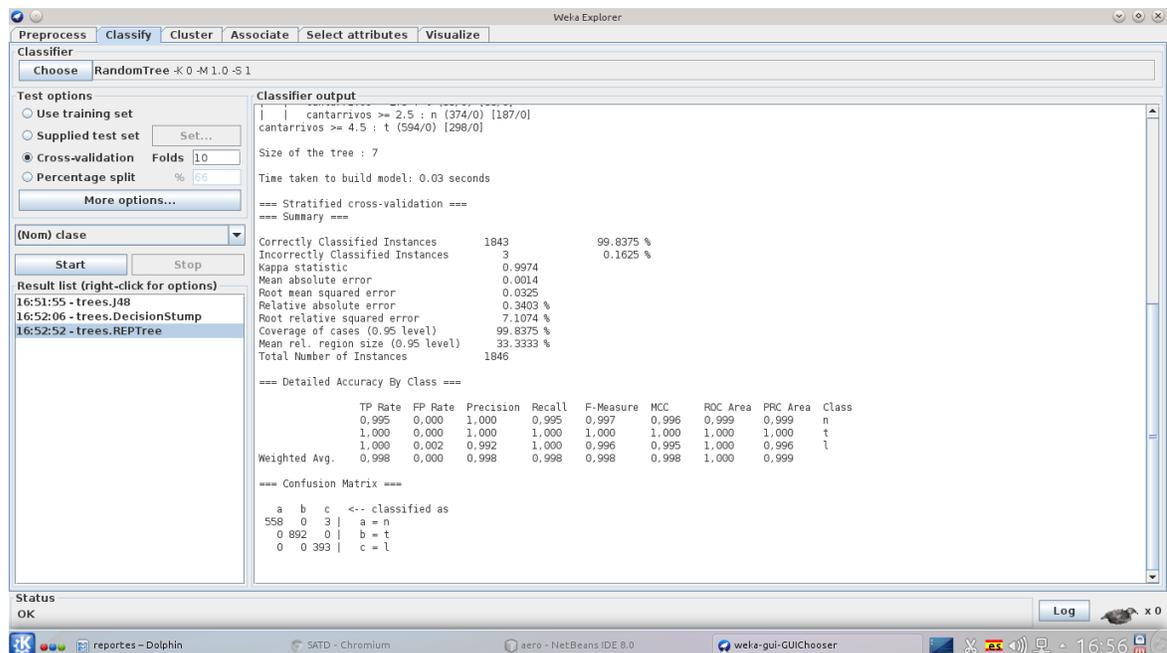


Figura 15: Algoritmo REPTree



Anexo 6: Historias de usuario

Tabla 14: HU Clasificación del día (elaboración propia)

Historias de Usuario	
Número: 2	Nombre: Clasificación del día.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Alto.
Iteraciones Asignadas: 1	Puntos de Estimación: 2 semanas.
Descripción: El día se clasificará en ligero, normal o trabajoso según el trabajo planeado para ese día.	
Observaciones: Se tomarán en cuenta los criterios de cantidad de vuelos, cantidad de arribos, cantidad de despegues, cantidad de pasajeros, cantidad de carga, cantidad de equipajes y cantidad de combustible.	

Tabla 15: HU Reportes semanales (elaboración propia)

Historias de Usuario	
Numero: 3	Nombre: Reportes semanales.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Medio.
Iteraciones Asignadas: 2	Puntos de Estimación: 1 semana.
Descripción: El usuario podrá consultar los reportes semanales de los pasajeros y los vuelos por días de la semana, los vuelos visualizado en arribos y despegues. La información se reflejará en diferentes tipos de gráficos.	
Observaciones: Para visualizar un reporte dado una semana, este debe estar en la base de datos, de lo contrario el sistema mostrará un gráfico sin información.	



Tabla 16: HU Reportes mensuales (elaboración propia)

Historias de Usuario	
Numero: 4	Nombre: Reportes mensuales.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Medio.
Iteraciones Asignadas: 2	Puntos de Estimación: 1 semana.
Descripción: El usuario podrá consultar los reportes mensuales de los pasajeros y los vuelos por días del mes, los vuelos visualizado en arribos y despegues. La información se reflejará en diferentes tipos de gráficos.	
Observaciones: Para visualizar un reporte dado un mes, este debe estar en la base de datos, de lo contrario el sistema mostrará un gráfico sin información.	

Tabla 17: HU Reportes anuales (elaboración propia)

Historias de Usuario	
Numero: 5	Nombre: Reportes anuales.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Medio.
Iteraciones Asignadas: 2	Puntos de Estimación: 1 semana.
Descripción: El usuario podrá consultar los reportes anuales de los pasajeros y los vuelos por meses del año, los vuelos visualizado en arribos y despegues. La información se reflejará en diferentes tipos de gráficos.	
Observaciones: Para visualizar un reporte dado un año, este debe estar en la base de datos, de lo contrario el sistema mostrará un gráfico sin información.	

Tabla 18: HU Autenticar administrador (elaboración propia)

Historias de Usuario	
Numero: 6	Nombre: Autenticar administrador.
Usuario: Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Medio.
Iteraciones Asignadas: 3	Puntos de Estimación: 1 semana.



Descripción: El usuario introduce sus datos (usuario y contraseña) y se autentica para de esta forma acceder a las opciones del sistema según su privilegio.

Observaciones: El usuario solo puede ser administrador.

Tabla 19: HU Gestionar correos (elaboración propia)

Historias de Usuario	
Numero: 7	Nombre: Gestionar correos.
Usuario: Administrador.	
Prioridad en Negocio: Media.	Riesgo en desarrollo: Alto.
Iteraciones Asignadas: 3	Puntos de Estimación: 2 semanas.
Descripción: Se podrá gestionar los destinatarios a recibir los reportes diarios. El sistema permitirá adicionar destinatario, modificar destinatario y eliminar destinatario, esto solo lo puede hacer administrador.	
Observaciones: Para acceder el administrador debe estar autenticado. Para modificar, eliminar, ver o buscar un destinatario, éste debe haberse creado con anterioridad.	

Tabla 20: HU Guardar como pdf (elaboración propia)

Historias de Usuario	
Numero: 8	Nombre: Guardar como pdf.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Media.	Riesgo en desarrollo: Medio.
Iteraciones Asignadas: 4	Puntos de Estimación: 1 semana.
Descripción: Se exportarán los reportes en formato pdf.	
Observaciones:	



Tabla 21: HU Servicios web (elaboración propia)

Historias de Usuario	
Numero: 9	Nombre: Servicios web.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Alta.	Riesgo en desarrollo: Alto.
Iteraciones Asignadas: 4	Puntos de Estimación: 2 semanas.
Descripción: Un petición externa podrá acceder a los reportes diarios mediante un servicio web. El resultado puede ser .json, .xml o .html.	
Observaciones:	

Tabla 22: HU Diseño adaptativo (elaboración propia)

Historias de Usuario	
Numero: 10	Nombre: Diseño adaptativo.
Usuario: Invitado, Administrador.	
Prioridad en Negocio: Media.	Riesgo en desarrollo: Alto.
Iteraciones Asignadas: 5	Puntos de Estimación: 2 semanas.
Descripción: El <u>software</u> deberá presentarse de forma correcta en cualquier dispositivo móvil en tres rangos (360x640, 768x1024 y 1280x720) de resoluciones de pantalla.	
Observaciones:	

Anexo 7: Tarjetas CRC

Tabla 23: CRC 2 (elaboración propia)

Clase: semanaController	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Buscar vuelos por día. - Generar pdf de los pasajeros por día. - Buscar pasajeros por día. - Generar pdf de los pasajeros por día. 	sabana



Tabla 24: CRC 3 (elaboración propia)

Clase: mesController	
Responsabilidades: <ul style="list-style-type: none"> - Buscar vuelos por mes. - Generar pdf de los vuelos por mes. - Buscar pasajeros por mes. - Generar pdf de los pasajeros por mes. 	Colaboradores: sabana

Tabla 25: CRC 4 (elaboración propia)

Clase: annoController	
Responsabilidades: <ul style="list-style-type: none"> - Buscar vuelos por año. - Generar pdf de los vuelos por año. - Buscar pasajeros por año. - Generar pdf de los pasajeros por año. 	Colaboradores: sabana

Tabla 26: CRC 5 (elaboración propia)

Clase: usuarioCorreoController	
Responsabilidades: <ul style="list-style-type: none"> - Crear un destinatario. - Editar un destinatario. - Eliminar un destinatario. - Listar destinatarios. - Autenticar administrador. 	Colaboradores: usuarioCorreo

Anexo 8: Tablas de la base de datos

sabana		
	id	integer(10) U
	compania	varchar(255)
	numero_vuelo	varchar(255)
	equipo	varchar(255)
	matricula	varchar(255)
	hora_utc_itin	time(7)
	hora_local_itin	time(7)
	hora_estimada	time(7)
	hora_utc_real	time(7)
	terminal	varchar(255)
	nacional_int	varchar(255)
	arr_dep	varchar(255)
	itin_bit	varchar(255)
	fecha_plan	date
	fecha_real	date
	ciudad	varchar(255)
	hora_dif_operacional	varchar(255)
	dead	varchar(255)
	composicion	varchar(255)
	paxs_total	varchar(255)
	paxs_transito	varchar(255)
	paxs_nac	varchar(255)
	carga_total	varchar(255)
	correo_total	varchar(255)
	equipaje_total	varchar(255)
	bultos	varchar(255)
	imputable	varchar(255)
	imp_uet	varchar(255)
	tiempo_rec_total	varchar(255)
	demoras	varchar(255)
	je fe_aero	varchar(255)
	je fe_operac	varchar(255)
	observaciones	varchar(255)
	id_vuelo_arribo	varchar(255)
	vuelo_arribo	varchar(255)
	posicion	varchar(255)
	sala	varchar(255)
	handling	varchar(255)
	combustible	varchar(255)
	estacion	varchar(255)
	carga	varchar(255)
	grupo	varchar(255)
	fecha	date

Figura 16: Tabla sabana

usuarioCorreo		
	id	integer(10) U
	nombre	varchar(255)
	email	varchar(255)
	area	varchar(255)
	cargo	varchar(255)

Figura 17: Tabla usuarioCorreo

clasificacion		
	id	integer(10) U
	fecha_plan	date
	cantidad_vuelos	integer(10)
	cant_arribo	integer(10)
	cant_despegues	integer(10)
	cant_paxs	integer(10)
	cant_carga	integer(10)
	cant_equipaje	integer(10)
	cant_combustible	integer(10)
	clase	varchar(255) N

Figura 18: Tabla clasificacion

Anexo 9: Tareas de Ingeniería

Tareas de Ingeniería de la primera iteración

Tabla 27: TI Clasificación del día (elaboración propia)

Tarea de Ingeniería	
Número: 3	Número de HU: 2
Nombre: Clasificación del día.	
Tipo de tarea: Desarrollo.	Estimación: 2 semanas.
Fecha de inicio: 9/4/2015	Fecha fin: 20/4/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se clasifican los días en ligeros, normal o trabajos.	

Tareas de Ingeniería de la segunda iteración

Tabla 28: TI Vuelos por días (elaboración propia)

Tarea de Ingeniería	
Número: 4	Número de HU: 3
Nombre: Vuelos por días.	
Tipo de tarea: Desarrollo.	Estimación: 3 días.
Fecha de inicio: 23/2/2015	Fecha fin: 25/2/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se muestran la cantidad de vuelos por días de la semana en diferentes tipos de gráficos.	

Tabla 29: TI Pasajeros por día (elaboración propia)

Tarea de Ingeniería	
Número: 5	Número de HU: 3
Nombre: Pasajeros por días.	
Tipo de tarea: Desarrollo.	Estimación: 2 días.
Fecha de inicio: 26/2/2015	Fecha fin: 27/2/2015



Programador responsable: Dairon y Anadelys.
Descripción: Se muestran la cantidad de pasajeros por días de la semana en diferentes tipos de gráficos.

Tabla 30: TI Vuelos por mes (elaboración propia)

Tarea de Ingeniería	
Número: 6	Número de HU: 4
Nombre: Vuelos por mes.	
Tipo de tarea: Desarrollo.	Estimación: 3 días.
Fecha de inicio: 2/3/2015	Fecha fin: 4/3/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se muestran la cantidad de vuelos por días del mes en diferentes tipos de gráficos.	

Tabla 31: TI Pasajeros por mes (elaboración propia)

Tarea de Ingeniería	
Número: 7	Número de HU: 4
Nombre: Pasajeros por mes.	
Tipo de tarea: Desarrollo.	Estimación: 2 días.
Fecha de inicio: 5/3/2015	Fecha fin: 6/3/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se muestran la cantidad de pasajeros por días del mes en diferentes tipos de gráficos.	

Tabla 32: TI Vuelos por año (elaboración propia)

Tarea de Ingeniería	
Número: 8	Número de HU: 5
Nombre: Vuelos por año.	
Tipo de tarea: Desarrollo.	Estimación: 3 días.
Fecha de inicio: 9/3/2015	Fecha fin: 11/3/2015



Programador responsable: Dairon y Anadelys.
Descripción: Se muestran la cantidad de vuelos por meses del año en diferentes tipos de gráficos.

Tabla 33: TI Pasajeros por año (elaboración propia)

Tarea de Ingeniería	
Número: 9	Número de HU: 5
Nombre: Pasajeros por año.	
Tipo de tarea: Desarrollo.	Estimación: 2 días.
Fecha de inicio: 12/3/2015	Fecha fin: 13/3/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se muestran la cantidad de pasajeros por meses del año en diferentes tipos de gráficos.	

Tareas de Ingeniería de la tercera iteración

Tabla 34: TI Autenticar administrador (elaboración propia)

Tarea de Ingeniería	
Número: 10	Número de HU: 6
Nombre: Autenticar administrador.	
Tipo de tarea: Desarrollo.	Estimación: 1 semana.
Fecha de inicio: 16/3/2015	Fecha fin: 20/3/2015
Programador responsable: Dairon y Anadelys.	
Descripción: El sistema permite al usuario introducir usuario (admin) y contraseña. Solo se autenticará el administrador.	

Tabla 35: TI Insertar destinatario (elaboración propia)

Tarea de Ingeniería	
Número: 11	Número de HU: 7
Nombre: Insertar destinatario.	
Tipo de tarea: Desarrollo.	Estimación: 3 días.

Fecha de inicio: 23/3/2015	Fecha fin: 25/3/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se adiciona un nuevo destinatario de correo.	

Tabla 36: TI Listar destinatarios (elaboración propia)

Tarea de Ingeniería	
Número: 12	Número de HU: 7
Nombre: Listar destinatarios.	
Tipo de tarea: Desarrollo.	Estimación: 2 días.
Fecha de inicio: 26/3/2015	Fecha fin: 27/3/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se listan todos los destinatarios de correo.	

Tabla 37: TI Eliminar destinatario (elaboración propia)

Tarea de Ingeniería	
Número: 13	Número de HU: 7
Nombre: Eliminar destinatario.	
Tipo de tarea: Desarrollo.	Estimación: 3 días.
Fecha de inicio: 30/3/2015	Fecha fin: 1/4/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se elimina un destinatario de correo.	

Tabla 38: TI Editar destinatario (elaboración propia)

Tarea de Ingeniería	
Número: 14	Número de HU: 7
Nombre: Editar destinatario.	
Tipo de tarea: Desarrollo.	Estimación: 2 días.
Fecha de inicio: 2/4/2015	Fecha fin: 3/4/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se edita un destinatario de correo.	



Tareas de Ingeniería de la cuarta iteración

Tabla 39: TI Guardar como pdf (elaboración propia)

Tarea de Ingeniería	
Número: 15	Número de HU: 8
Nombre: Guardar como pdf.	
Tipo de tarea: Desarrollo.	Estimación: 1 semana.
Fecha de inicio: 6/4/2015	Fecha fin: 10/4/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Se exportarán los reportes en formato pdf.	

Tabla 40: TI Servicios web (elaboración propia)

Tarea de Ingeniería	
Número: 16	Número de HU: 9
Nombre: Servicios web.	
Tipo de tarea: Desarrollo.	Estimación: 2 semanas.
Fecha de inicio: 13/4/2015	Fecha fin: 24/4/2015
Programador responsable: Dairon y Anadelys.	
Descripción: Un petición externa podrá acceder a los reportes diarios mediante un servicio web. El resultado puede ser .json, .xml o .html.	

Tareas de Ingeniería de la quinta iteración

Tabla 41: TI Diseño adaptativo (elaboración propia)

Tarea de Ingeniería	
Número: 17	Número de HU: 10
Nombre: Diseño adaptativo.	
Tipo de tarea: Desarrollo.	Estimación: 2 semanas.
Fecha de inicio: 27/4/2015	Fecha fin: 8/5/2015
Programador responsable: Dairon y Anadelys.	



Descripción: El software deberá presentarse de forma correcta en cualquier dispositivo móvil en tres rangos (360x640, 768x1024 y 1280x720) de resoluciones de pantalla.

Anexo 10: Casos de pruebas de aceptación

Casos de Prueba de la segunda iteración

Tabla 42: CP Vuelos por días (elaboración propia)

Caso de prueba de aceptación	
Código: HU3_CP3	Número de HU: 3
Nombre: Mostrar vuelos por días.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de vuelos por días de la semana.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
Entrada/ Pasos de ejecución: <ol style="list-style-type: none"> 1. El administrador o invitado selecciona la opción: Semanas. 2. El administrador o invitado escoger la opción que desea: Vuelos por días. Pasajeros por días (Ver prueba HU3_CP4). 3. El administrador o invitado selecciona la opción: Vuelos por días. 4. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. 5. Se muestra la información de la cantidad de vuelos por días de la semana en diferentes tipos de gráficos y la cantidad total de vuelos en la semana. 6. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9). 	
Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de vuelos por días de la semana.	

Evaluación de la prueba: Prueba satisfactoria.

Tabla 43: CP Pasajeros por día (elaboración propia)

Caso de prueba de aceptación	
Código: HU3_CP4	Número de HU: 3
Nombre: Mostrar pasajeros por día.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de pasajeros por días de la semana.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
Entrada/ Pasos de ejecución: <ol style="list-style-type: none"> 1. El administrador o invitado selecciona la opción: Semanas. 2. El administrador o invitado escoger la opción que desea: Vuelos por días (Ver prueba HU3_CP3). Pasajeros por días. 3. El administrador o invitado selecciona la opción: Pasajeros por días. 4. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. 5. Se muestra la información de la cantidad de pasajeros por días de la semana en diferentes tipos de gráficos y los arribos y despegues. 6. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9). 	
Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de pasajeros por días de la semana.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 44: CP Vuelos por mes (elaboración propia)

Caso de prueba de aceptación	
Código: HU4_CP5	Número de HU: 4
Nombre: Mostrar vuelos por mes.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de vuelos por los	



días del mes.
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.
<p>Entrada/ Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El administrador o invitado selecciona la opción: Meses. 2. El administrador o invitado escoger la opción que desea: Vuelos en mes. Pasajeros en mes (Ver prueba HU4_CP6). 3. El administrador o invitado selecciona la opción: Vuelos en mes. 4. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. 5. Se muestra la información de la cantidad de vuelos por días del mes en diferentes tipos de gráficos y la cantidad total de vuelos en el mes. 6. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9).
Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de vuelos por días del mes.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 45: CP Pasajeros por mes (elaboración propia)

Caso de prueba de aceptación	
Código: HU4_CP6	Número de HU: 4
Nombre: Mostrar pasajeros por mes.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de pasajeros por los días del mes.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
<p>Entrada/ Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El administrador o invitado selecciona la opción: Meses. 2. El administrador o invitado escoger la opción que desea: Vuelos en mes (Ver prueba HU3_CP5). 	



<p>Pasajeros en mes.</p> <ol style="list-style-type: none"> El administrador o invitado selecciona la opción: Pasajeros en mes. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. Se muestra la información de la cantidad de pasajeros por días del mes en diferentes tipos de gráficos y los arribos y despegues. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9).
<p>Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de pasajeros por días del mes.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 46: CP Vuelos por año (elaboración propia)

Caso de prueba de aceptación	
Código: HU5_CP7	Número de HU: 5
Nombre: Mostrar vuelos por año.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de vuelos por meses del año.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
<p>Entrada/ Pasos de ejecución:</p> <ol style="list-style-type: none"> El administrador o invitado selecciona la opción: Años. El administrador o invitado escoger la opción que desea: Vuelos en años. Pasajeros en años (Ver prueba HU5_CP8). El administrador o invitado selecciona la opción: Vuelos en años. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. Se muestra la información de la cantidad de vuelos por meses del año en diferentes tipos de gráficos y la cantidad total de vuelos en el año. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9). 	



Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de vuelos por meses del año.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 47: CP Pasajeros por año (elaboración propia)

Caso de prueba de aceptación	
Código: HU5_CP8	Número de HU: 5
Nombre: Mostrar pasajeros por año.	
Descripción: Prueba de funcionalidad para mostrar la cantidad de pasajeros por meses del año.	
Condiciones de ejecución: Para acceder a esta información puede ser administrador o invitado.	
Entrada/ Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El administrador o invitado selecciona la opción: Años. 2. El administrador o invitado escoger la opción que desea: Vuelos en años (Ver prueba HU5_CP7). Pasajeros en años. 3. El administrador o invitado selecciona la opción: Vuelos en años. 4. Selecciona la fecha (día, mes, año) y selecciona la opción Buscar. 5. Se muestra la información de la cantidad de pasajeros por meses del año en diferentes tipos de gráficos y los arribos y despegues. 6. El administrador o invitado puede escoger la opción Guardar PDF (Ver prueba HU6_CP9). 	
Resultado esperado: Se muestra correctamente la información relacionada con la cantidad de pasajeros por meses del año.	
Evaluación de la prueba: Prueba satisfactoria.	



Casos de Prueba de la tercera iteración

Tabla 48: CP Autenticar administrador (elaboración propia)

Caso de prueba de aceptación	
Código: HU6_CP9	Número de HU: 6
Nombre: Autenticar administrador.	
Descripción: Prueba de funcionalidad para controlar el acceso a la gestión de correos.	
Condiciones de ejecución: Solo puede acceder a los destinatarios el administrador.	
Entrada/ Pasos de ejecución: <ol style="list-style-type: none">1. El administrador selecciona la opción: Destinatarios.2. El administrador introduce los datos: usuario (el campo usuario por defecto está puesto admin) y contraseña.3. El administrador selecciona Autenticar.	
Resultado esperado: Se autentica el administrador y accede a la gestión de correos.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 49: CP Insertar destinatario (elaboración propia)

Caso de prueba de aceptación	
Código: HU7_CP10	Número de HU: 7
Nombre: Insertar destinatario.	
Descripción: Prueba de funcionalidad para gestionar correos, con el fin de insertar nuevos destinatarios al sistema.	
Condiciones de ejecución: El administrador debe estar autenticado.	
Entrada/ Pasos de ejecución: Prueba satisfactoria. <ol style="list-style-type: none">1. El administrador selecciona la opción: Destinatarios.1. El administrador selecciona la opción: Añadir destinatario.2. El administrador introduce los datos para agregar un nuevo destinatario.	



<p>3. El administrador selecciona la opción: Añadir.</p> <p>4. El administrador observa los datos del destinatario añadido, correspondiente a la opción Ver (Ver prueba HU9_CP14).</p>
<p>Resultado esperado: Una vez introducidos los datos correctamente se adiciona el destinatario satisfactoriamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 50: CP Listar destinatario (elaboración propia)

Caso de prueba de aceptación	
Código: HU7_CP11	Número de HU: 7
Nombre: Listar destinatario.	
Descripción: Prueba de funcionalidad para gestionar correos, con el fin de listar todos los destinatarios.	
Condiciones de ejecución: El administrador debe estar autenticado. Para que el administrador pueda seleccionar las opciones Ver y Editar se deben haber adicionado con anterioridad al menos un destinatario.	
Entrada/ Pasos de ejecución:	
<p>1. El administrador selecciona para cada destinatario que se encuentra en la lista, la opción que desea:</p> <p style="padding-left: 40px;">Ver (Ver prueba HU9_CP14).</p> <p style="padding-left: 40px;">Editar (Ver prueba HU9_CP15).</p> <p style="padding-left: 40px;">Adicionar destinatario (Ver prueba HU9_CP12).</p>	
Resultado esperado: Se listan todos los destinatarios con todos sus datos.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 51: CP Ver destinatario (elaboración propia)

Caso de prueba de aceptación	
Código: HU7_CP12	Número de HU: 7
Nombre: Ver destinatario.	
Descripción: Prueba de funcionalidad para gestionar correos, con el fin de ver un	



destinatario.
Condiciones de ejecución: El administrador debe estar autenticado. Se deben haber adicionado con anterioridad al menos un destinatario.
Entrada/ Pasos de ejecución: <ol style="list-style-type: none"> 1. El administrador selecciona la opción: Ver. 2. El administrador observa los datos del destinatario seleccionado. 3. El administrador escoge la opción que prefiere: <ul style="list-style-type: none"> Volver (Regresa a la interfaz donde están listados todos los destinatarios). Editar (Ver prueba HU9_CP15). Eliminar (Ver prueba HU9_CP16).
Resultado esperado: Se muestra los datos correspondientes al destinatario seleccionado.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 52: CP Editar destinatario (elaboración propia)

Caso de prueba de aceptación	
Código: HU7_CP13	Número de HU: 7
Nombre: Editar destinatario.	
Descripción: Prueba de funcionalidad para gestionar correos, con el fin de editar un destinatario.	
Condiciones de ejecución: El administrador debe estar autenticado. Se deben haber adicionado con anterioridad al menos un destinatario.	
Entrada/ Pasos de ejecución: <ol style="list-style-type: none"> 1. El administrador selecciona la opción: Editar. 2. El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios). Volver (Regresa a la interfaz donde están listados todos los destinatarios). Eliminar (Ver prueba HU9_CP16). 	
Resultado esperado: Se guardan correctamente los cambios realizados del	



destinatario.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 53: CP Eliminar destinatario (elaboración propia)

Caso de prueba de aceptación	
Código: HU7_CP14	Número de HU: 7
Nombre: Eliminar destinatario.	
Descripción: Prueba de funcionalidad para gestionar correos, con el fin de borrar destinatarios.	
Condiciones de ejecución: El administrador debe estar autenticado.	
Entrada/ Pasos de ejecución: El administrador confirma el borrado, escogiendo la opción: Eliminar (Se elimina el destinatario con todos sus datos y se regresa a la interfaz donde están listados todos los destinatarios).	
Resultado esperado: Se borran los destinatarios con todos sus datos.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 54: CP Verificar envío de correo (elaboración propia)

Caso de prueba de aceptación	
Código: HU7_CP15	Número de HU: 7
Nombre: Verificar envío de correo.	
Descripción: Prueba de funcionalidad para verificar si los correos enviados fueron recibidos.	
Condiciones de ejecución: Existir como mínimo un destinatario.	
Entrada/ Pasos de ejecución: Acceder a la ruta de enviar correo <code>/enviarCorreo</code> .	
Resultado esperado: Envío correctamente del correo a los destinatarios.	
Evaluación de la prueba: Prueba satisfactoria.	



Casos de Prueba de la cuarta iteración

Tabla 55: CP Guardar como pdf (elaboración propia)

Caso de prueba de aceptación	
Código: HU8_CP16	Número de HU: 8
Nombre: Guardar como pdf.	
Descripción: Prueba de funcionalidad de mostrar la información en formato pdf.	
Condiciones de ejecución: Para poder guardar la información en formato pdf se tiene que haber accedido algún reporte (Días, Semanas, Meses o Año).	
Entrada/ Pasos de ejecución: El administrador o invitado selecciona la opción: Guardar PDF .	
Resultado esperado: La información es guardada en un documento pdf.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 56: CP Servicios web (elaboración propia)

Caso de prueba de aceptación	
Código: HU9_CP17	Número de HU: 9
Nombre: Servicios web.	
Condiciones de ejecución:	
Entrada/ Pasos de ejecución: El administrador o invitado realiza una petición a la ruta <code>/api/sget</code> del servicio web.	
Resultado esperado: El servicio web retorna una respuesta con el reporte diario de vuelos por horas y pasajeros por horas en forma de arreglo. El formato (html, json, xml) puede ser especificado en la ruta a continuación de un punto.	
Evaluación de la prueba: Prueba satisfactoria.	

Casos de Prueba de la quinta iteración

Tabla 57: CP Diseño adaptativo (elaboración propia)

Caso de prueba de aceptación	
Código: HU10_CP18	Número de HU: 10
Nombre: Diseño adaptativo.	
Descripción: Muestra la aplicación en tres rangos (360x640, 768x1024 y 1280x720) de resoluciones de pantalla.	
Condiciones de ejecución:	
Entrada/ Pasos de ejecución:	
Resultado esperado: Muestra la aplicación correctamente adaptándose a cada tamaño de pantalla.	
Evaluación de la prueba: Prueba satisfactoria.	

Anexo 11: Imágenes de la aplicación

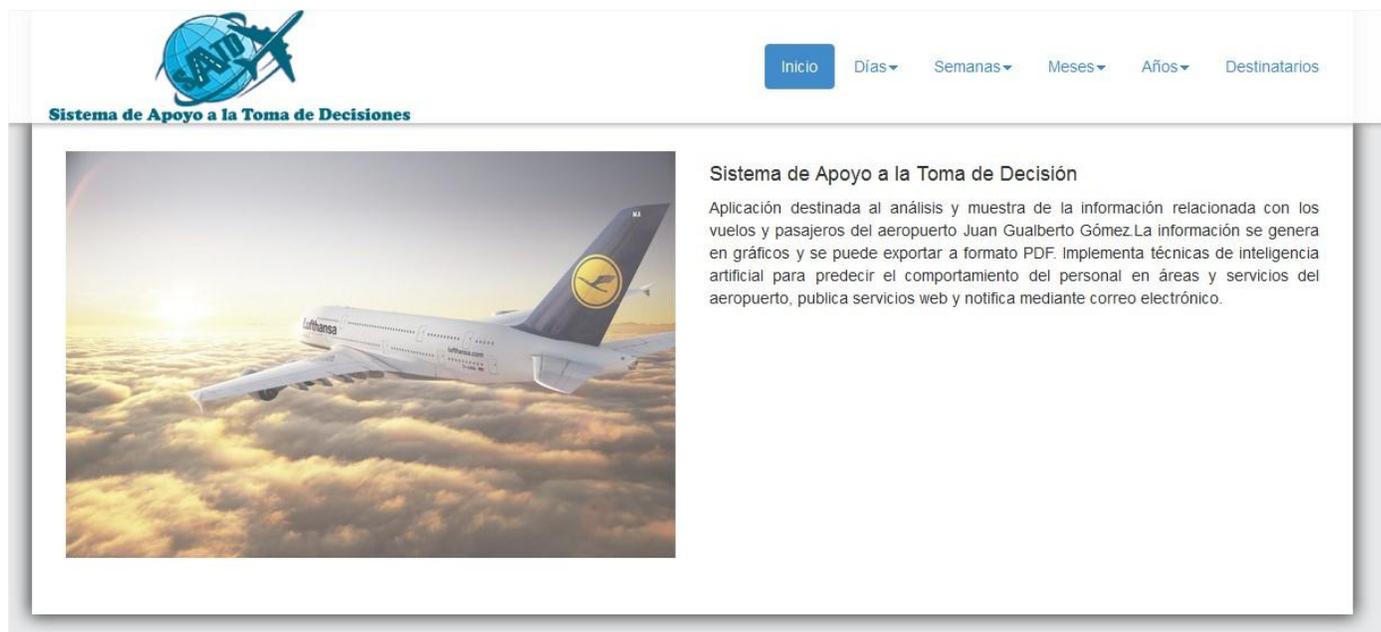


Figura 19: Pantalla principal

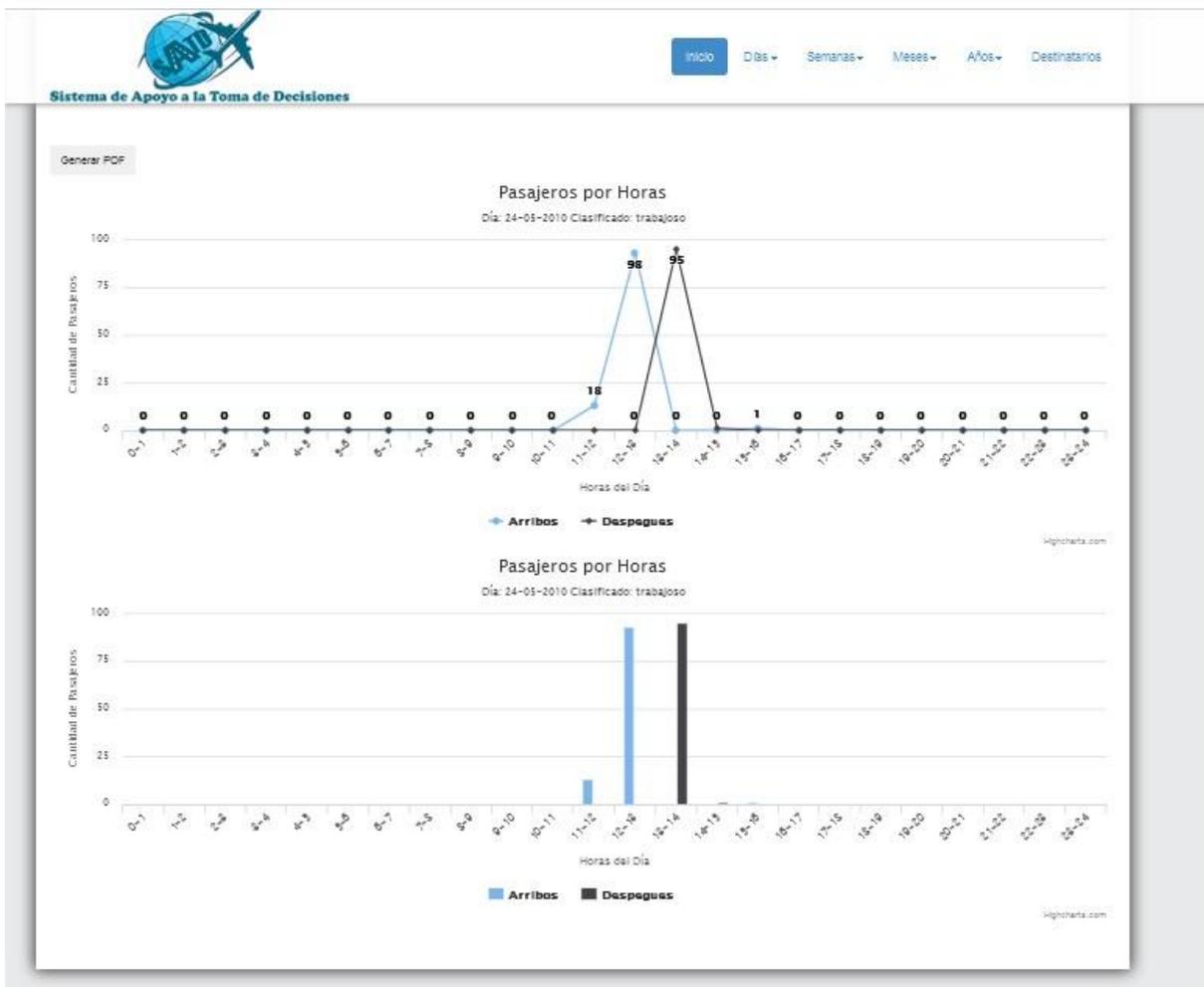


Figura 20: Reportes diarios, su clasificación, guardar pdf

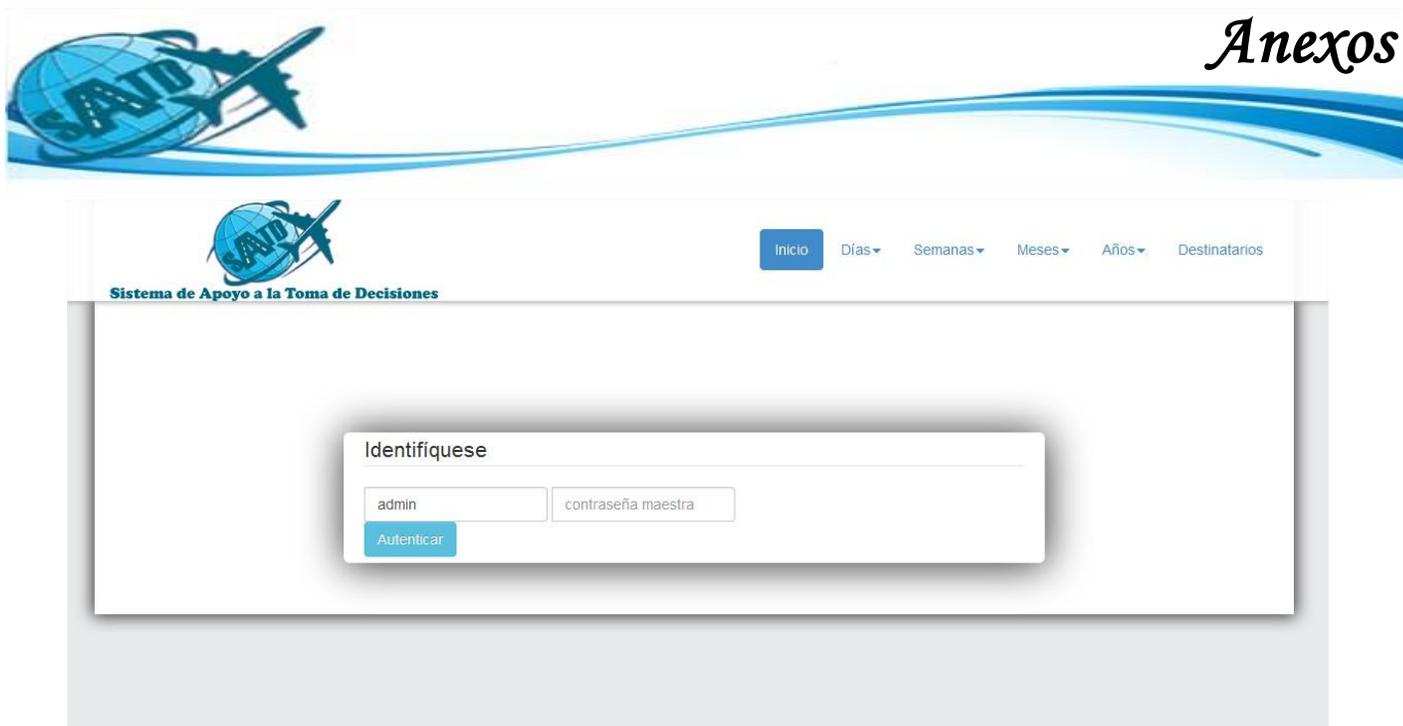


Figura 21: Autenticar administrador

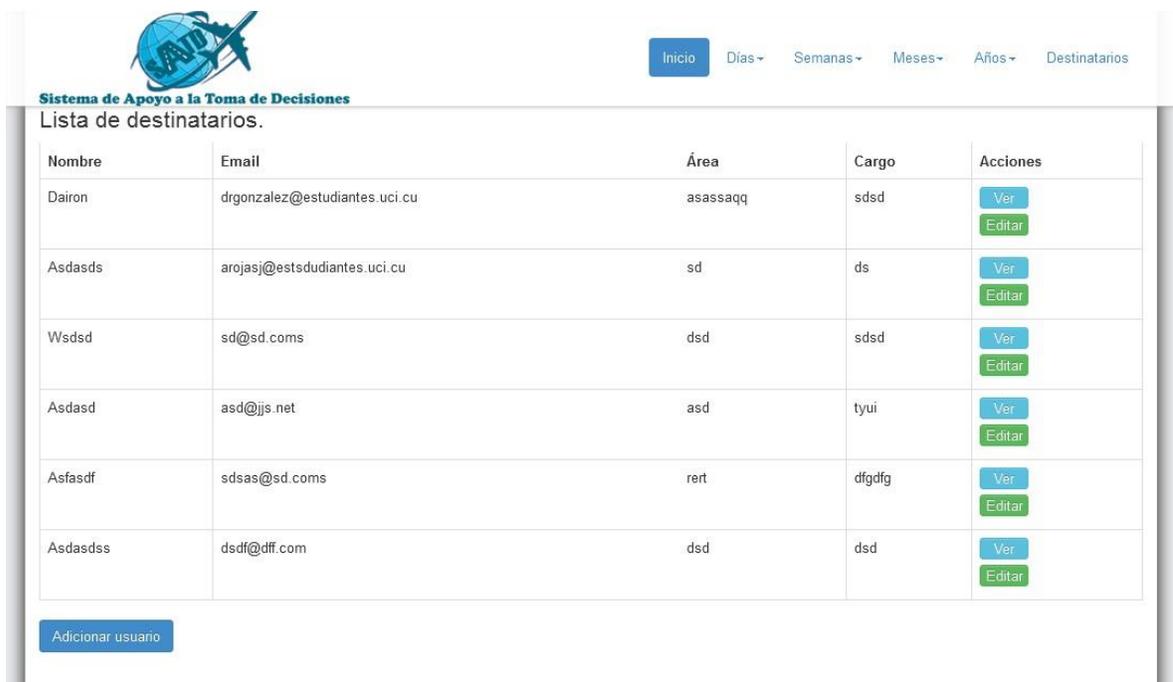


Figura 22: Sección en donde se gestionan los destinatarios de correo