



Instituto Superior Minero Metalúrgico

“Dr. Antonio Núñez Jiménez”.

Facultad de Geología - Minas

*Trabajo de Diploma para optar por el Título de Ingeniero*

*Informático*

Título:

Sistema Informático de Registro D

Operaciones a Buques.

Autor:

Alex Rosabal Sánchez

Tutores:

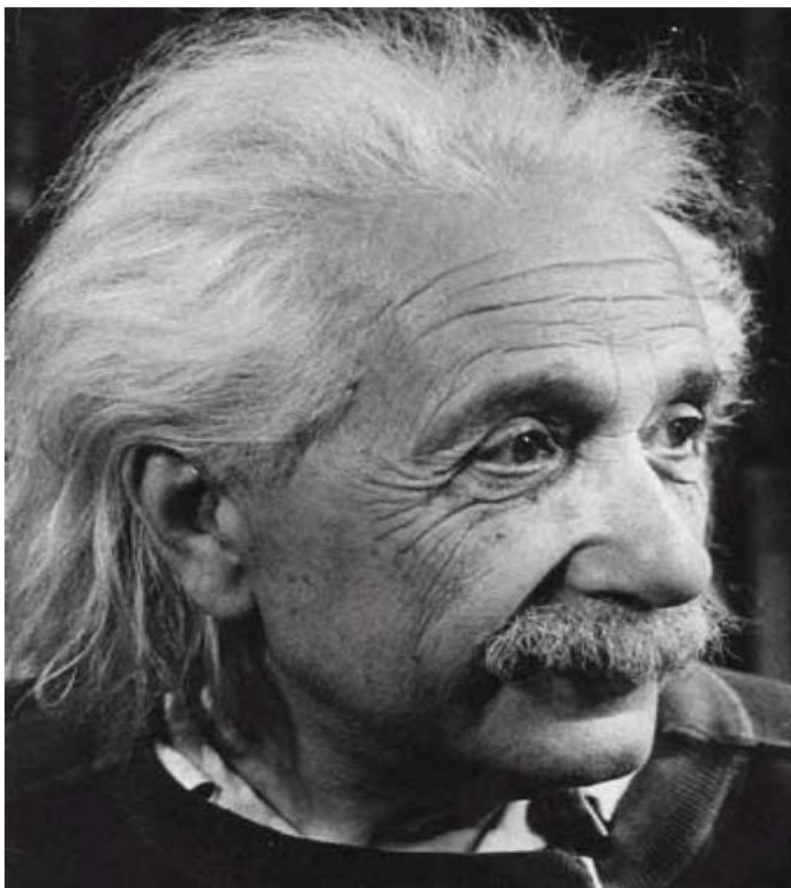
Ing. Daira Argüelles Blanco

Ing. Edgar Ruñez Torres

Moa, Holguín

2012

## *Pensamiento*



*La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos.*

*Albert Einstein*



## Agradecimientos

- *A mi madre ante todo por siempre haber estado hay en todo momento y por encima de todo por cada momento vivido junto a ella y por los que quedan.*
- *A mi familia porque siempre me apoyo en todo momento en las buenas y las malas*
- *A mis hijos que son unas de las fuentes de inspiración de este trabajo.*
- *A mis amigos que de una forma u otra han estado ahí siempre por su ayuda incondicional en todo.*
- *A mis tutores Ing. Yadira Arguelles Blanco y el Ing. Edgar Nuñez Torres por su inestimable ayuda y guía en todo momento*

*A todos muchas gracias por estar ahí...*

*Alex Rosabal Sánchez*



## Dedicatoria

*A mis padres a mi familia y a mis amigos para todos ellos es esta tesis..*

*Alex Rosabal Sánchez*



**Declaración de autoría**

**OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA**



**OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA**

## **Resumen**

El ascendente desarrollo que ha acontecido en los últimos años, ha situado las tecnologías de la información en un lugar relevante en el ámbito del intercambio y gestión de la información e informatización de los diferentes sistemas.

La Empresa Consignataria Mambisa del municipio de Moa en específico el área de operaciones se desenvuelve en un entorno de niveles considerable de información proveniente de las operaciones que se le realizan a los barcos en el puerto estas operaciones se manejan de forma manual. La gestión de la información se ha convertido en un tema de amplia significación, lo que propicia la necesidad de crear un sistema informático que permita llevar el control de las operaciones que realiza la empresa.

Con el desarrollo de esta investigación, se propone la informatización de la gestión de la información correspondiente a las operaciones de la entidad, a partir de la creación de una aplicación de escritorio que gestione la información de manera ágil y segura.

Para la elaboración de la misma se realizó un estudio bibliográfico sobre las aplicaciones y las herramientas Builder C++, MySQL para la construcción de las mismas. En este documento se recoge un resumen del estudio realizado, y se presenta la metodología XP de Ingeniería de Software que se siguió para el desarrollo del software que se propone como solución de la problemática encontrada.



## **Abstract**

The upward development that has occurred in recent years has placed the information technologies in a prominent place in the area in the area of exchange and information management and computerization of different systems.

The Empresa Consignataria Mambisa of municipality of Moa specific in the area of operations, is developed in an environment of significant levels of information from operations to be carried to the ships the port, this operations are handled manually. The information management has become a topic of broad significance, which promotes the need for a computerized system to keep control of operations performed by the company.

With the development of this research, we propose the computerization of information management for the operations of the entity, from the creation of a desktop application that manages information swiftly and safely.

For the preparation of this work conducted a literature review on the applications and tools Builder C++, MySQL for building them. This document provides a summary of the study, and presents the software the engineering methodology XP that was followed for software development and that is proposed as a solution to the problems found.

# Índice

## Contenido

Pensamiento	I
Agradecimientos	II
Dedicatoria	III
Declaración de autoría	IV
OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA	V
OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA	VI
Resumen	VII
Abstract	VIII
Índice	IX
Introducción	1
Capítulo I Fundamentación Teórica	5
1.1    Introducción	5
1.2    Estado del Arte	5
1.2.1    Antecedentes de la navegación.	5
1.2.2    Sistemas automatizados existentes vinculados al campo de acción	7
1.3    Herramientas para el desarrollo de Aplicaciones.	7
1.3.1    Lenguajes de Programación.	7
1.3.2    C++	8
1.3.3    Java	8
1.3.4    Delphi	9
1.4    Sistemas de Gestión de Bases de Datos.	10
1.4.1    ¿Qué es un Sistema Gestor de Bases de Datos?	10
1.4.2    Oracle.	10
1.4.3    MySQL.	11
1.4.4    PostgreSQL:	12
1.4.5    Comparación entre PostgreSQL y MySQL.	13
1.5    Fundamentación de la selección de las herramientas.	13
1.5.1    ¿Por qué seleccionar C++?	13

1.5.2	¿Por qué seleccionar PostgreSQL?	14
1.6	Metodología de desarrollo de sistemas informáticos	14
1.6.1	Metodología propuesta para el desarrollo de la aplicación	14
1.6.2	Rational Unified Process (RUP)	15
1.6.3	SXP	16
1.6.4	Programación extrema (XP).	17
1.7	Patrones arquitectónicos	21
1.7.1	Arquitectura por Capas	21
1.7.2	Patrón de arquitectura MVC (Modelo vista controlador)	23
1.8	Lenguajes de Modelado.	25
1.8.1	UML	25
1.9	Conclusiones del capítulo.	26
Capítulo 2 Planificación y Diseño		27
2.1	Introducción	27
2.2	Funcionalidades generales	27
2.2.1	Personal relacionado con el sistema	28
2.2.2	Lista de reserva	28
2.2.3	Historias de usuario	29
2.3	Planificación de entregas	31
2.3.1	Estimación de esfuerzo por historias de usuario	32
2.3.2	Planificación de iteraciones	32
2.3.3	Plan de duración de las iteraciones	34
2.4	Clases, responsabilidades y colaboradores	35
2.5	Conclusiones	36
Capítulo 3 Desarrollo y pruebas.		37
3.1	Introducción	37
3.2	Modelo de Datos	37
3.3	Desarrollo de las iteraciones.	38
3.3.1	Tareas por historias de usuario	38
3.4	Pruebas	40
3.4.1	Desarrollo dirigido por pruebas	41



3.4.2 Pruebas de aceptación	41
3.5 Conclusiones.	43
Capítulo 4 Estudio de Factibilidad	44
4.1 Introducción	44
4.2 Efectos Económicos	44
4.2.1 Efectos directos	44
4.2.2 Efectos indirectos	45
4.2.3 Externalidades	45
4.2.4 Intangibles	45
4.3 Ficha de Costo	45
4.4 Conclusiones del Capítulo	48
Conclusiones Generales	49
Recomendaciones	50
Referencia Bibliográfica	51
GLOSARIO DE TERMINOS MARITIMO - PORTUARIOS	53
Anexo 1: Interfaz de Usuario	54
Anexo 2: Historias de Usuarios	58
Anexo 3: Tarjetas CRC	62
Anexo 4: Modelo de datos	65
Anexo 5: Tarjetas de Tareas	66
Anexo 6: Prueba de Aceptación	76



## Introducción

Desde la antigüedad los hombres han realizado el traslado de las mercancías y personal utilizando como medio de transporte los barcos, muchos de ellos fueron usados para diversos fines. Estos se ha expandido por todo el mundo y por generaciones, en la actualidad el hombre la mayoría de sus mercancías las transporta por los barcos de puerto en puerto. Para esto es necesario tener un control de las operaciones que se le realizan a los buques en los puertos.

La esfera de influencia de la informática en el mundo contemporáneo es muy amplia y multidisciplinaria, relacionando disciplinas establecidas, como son la ciencia de la computación y la ingeniería; que abarca las relaciones informativas en sistemas hombre-hombre y hombre-máquina.

La informatización solo es viable, si al evaluar los beneficios económicos y sociales de las mejoras que se puedan obtener al informatizar, son mayores a los costos de operación y mantenimiento del sistema, sin embargo, no hay dudas de que la informatización de un proceso frente a la ejecución manual del mismo, ofrece ventajas de orden económico, social, y tecnológico; esto se debe a que se asegura una mejora en la calidad del trabajo por parte del que lo realiza y en el desarrollo del proceso, la que dependerá de la eficiencia del sistema implementado, que se traduce en la reducción de los costos, debido a la racionalidad del trabajo, logra reducir los tiempos de procesamiento de información y un conocimiento más detallado, mediante la recopilación de información y datos estadísticos del proceso.

En Cuba el desarrollo de la informática y las nuevas tecnologías ha tenido un gran auge, son varias las instituciones que se suman al grupo que utilizan estas para su desarrollo, una de ellas es la Empresa Consignataria Mambisa del municipio Moa, en específico el área de operaciones de buques en la se desea hacer uso de las nuevas tecnologías e informatizar las actividades que allí se realizan. Estas operaciones que se le realizan a los buques pasan por 6 etapas próximos arribos, permisos de entrada y descarga, reporte de arribo, permiso



de remoción y atraque, SOF (Estado de Hecho), Permiso de Salida. En los próximos arribos se realiza un control con la entrada en las próximas 24 horas, 72 horas y 7 días, en los permisos de entrada y descarga van recogidos todos los datos de los buques que atracaran al puerto, dentro se encuentra los reportes de solicitud de remolcadores si es necesario y el de prácticos para la entrada al canal del puerto, y el permiso de descarga se realiza en el momento que el barco está entrando en el puerto o está atracado en la bahía, en el reporte de arribo se recogen todos los datos del buque entre ellos la hora y día que atraco y donde atraco, en el permiso de remoción y atraque este es necesario para mover el buque de lugar, en el SOF (Estado de Hecho) se registran todo las maniobras que realiza el barco en el puerto desde que entra hasta que sale, el Permiso de Salida se refiere a como parte el buque del puerto y hacia dónde se dirige. Todo este proceso expuesto anteriormente es llevado a cabo mediante el trabajo manuscrito, el que se hace cada vez más engorroso, realizándose sin la rapidez necesaria.

Luego de realizar un análisis de lo anteriormente expuesto, se plantea, como **problema científico**, la no existencia de un sistema informático para el registro y operaciones a buques de la Empresa Consignataria Mambisa, que permita rapidez y eficiencia en el desarrollo de este proceso.

En busca de una solución al problema antes planteado nos proponemos como **objeto de estudio** el proceso de registro y operaciones de los buques en la Empresa Consignataria Mambisa del municipio Moa.

El **campo de acción** se enmarca en la informatización del proceso de registro y operaciones de buques en la Empresa Consignataria Mambisa del municipio Moa.

Se plantea como **Idea a defender** que el desarrollo de una herramienta informática para el registro y operaciones de buques de la Empresa



Consignataria Mambisa facilitará el trabajo de los especialistas permitiéndole mayor rapidez y eficiencia en el proceso de operaciones de buques.

Para dar solución al problema se planteó el siguiente **objetivo general**:

Implementar un software para informatizar el proceso de registro y operaciones de buques.

De acuerdo a esta propuesta se derivan los siguientes **objetivos específicos**:

1. Establecer el estado del arte sobre la información disponible tanto nacional e internacional relacionada con los sistemas informáticos en el campo del registro y operaciones de buques.
2. Realizar un estudio de las herramientas a utilizar.
3. Análisis, diseño e implementación de la aplicación.
4. Realizar el estudio de factibilidad e implantar el software..

Para dar cumplimiento a los siguientes objetivos específicos se plantearon las siguientes **tareas**:

1. Análisis bibliográfico relacionado con el proceso de registro y operaciones de buques.
2. Selección del lenguaje de programación para el desarrollo de la aplicación.
3. Diseño e implementación de la herramienta informática para el registro y operaciones de buques.
4. Validar el sistema informático creado.

Entre los métodos y técnicas de investigación aplicados principalmente están  
Métodos Empíricos:

- Entrevistas para determinar los requerimientos funcionales del sistema informático que se quiere construir. Se llevó a cabo un diálogo con personas expertas en la materia.
- Análisis de documentos: para elaborar los fundamentos teóricos que se relacionan con el campo de acción.



- Revisión de documentos: se utiliza para conocer los detalles del funcionamiento del proceso de registro y operaciones del buque.

El presente trabajo consta de cuatro capítulos:

**Capítulo 1 “Fundamentación Teórica”:** se ofrece una breve descripción del concepto de operaciones a buques. Se realiza un estudio acerca de los diferentes sistemas existentes vinculados al campo de acción, además se presenta la metodología y las herramientas a utilizar en el desarrollo del sistema propuesto.

**Capítulo 2 “Planificación y Diseño”:** se hace uso de la metodología expuesta en el capítulo inicial para el desarrollo del proyecto, abordando sus dos primeras fases.

**Capítulo 3 “Desarrollo y Pruebas”:** se presentan la implementación de las tarjetas de ingeniería así como las pruebas realizadas con sus resultados.

**Capítulo 4 “Estudio de Factibilidad”:** se realiza un estudio para ver la factibilidad del producto. Además se ilustra el gráfico con el tiempo que demora el proceso de evaluación de forma manual y automatizada.





## Capítulo I Fundamentación Teórica

### 1.1 Introducción

En este capítulo se abordarán los conceptos fundamentales para el desarrollo del proceso de elaboración y ejecución del producto, sus principales características, así como las herramientas que se pueden utilizar para obtener un buen producto. Se realizará un estudio de los diferentes Gestores de Base de Datos, lenguajes de programación y metodologías existentes para determinar cuáles son más convenientes.

### 1.2 Estado del Arte

#### 1.2.1 Antecedentes de la navegación.

El ser humano ha navegado desde su pasado más remoto, aunque las pruebas directas más antiguas que nos han llegado de ello correspondan al Mesolítico. A Australia se llegó hace más de 40.000 años desde la costa del Sudoeste cruzando cortos tramos de los estrechos, y muchos homínido tuvieron que atravesar ríos caudalosos y mares muchos miles de años antes, con embarcaciones muy rudimentarias o simples troncos, pero llegaban al otro lado. El tipo de navegación en aquellos tiempos se realizaba o bien por aguas interiores o bordeando la costa, intentando evitar el adentrarse en alta mar siempre que se podía. Lógicamente, también la época del año era importante, la mejor era el verano y, por las condiciones climáticas existentes, se desarrolló el comercio, la navegación y difusión cultural de mejor forma en las zonas templadas, donde la mayor parte del año se podía navegar, es decir, a lo largo de todo el Mediterráneo.

Los primeros vestigios son dados por Lloyd's Register es una Sociedad de clasificación y una organización de análisis de riesgos. Históricamente, Lloyd's



Register of Shipping, era una organización exclusivamente de ámbito marítimo. A finales del siglo XX la compañía se diversificó en otros sectores.

Al igual que la famosa compañía de seguros, Lloyd's of London, Lloyd's Register toma su nombre y origen de la cafetería londinense del siglo XVII frecuentada por mercaderes, agentes de seguros y armadores, unidos todos ellos por sus negocios dentro del sector marítimo. El propietario Edward Lloyd, inventó un sistema de intercambiar información circulando una hoja informativa con las noticias que recibía. En 1760, la «sociedad de registro» (actualmente sociedad de clasificación) se formó con los clientes de la cafetería.

Aparte de esta conexión histórica, Lloyd's Register no tiene ninguna otra relación con la agencia de seguros Lloyd's of London.

La sociedad editó el primer *Registro de buques* en 1764 para dar tanto a los agentes de seguros como a los armadores una idea del estado de los buques que aseguraban y fletaban. La clasificación "A1", de la que se deriva la actual nomenclatura de las «sociedades de clasificación», apareció por primera vez en la edición 1775-1776 del registro.

El registro, con información de buques mercantes de más de 100 TPM, se ha venido publicando anualmente desde entonces.

En Cuba la historia naval tiene su origen y se encuentra indisolublemente ligada a la española; desde el comienzo de la colonización de la Isla caribeña, dicha plaza se convirtió en un enclave, para las reparaciones y avituallamiento de los buques de la corona, hasta que se fueron erigiendo los primeros astilleros; en fechas tan tempranas como 1517 se reciben las primeras autorizaciones para que se pudiesen construir navíos, lo que fue confirmado en 1518 por el monarca Carlos V, aunque la primera noticia que se tiene de una nao construida en Cuba, está recogida en el Libro Primero del Registro de la Casa de Contratación de Sevilla.



### **1.2.2 Sistemas automatizados existentes vinculados al campo de acción**

#### Ventanilla única de comercio exterior

El componente de servicios portuarios, usualmente conocido como Ventanilla Única Portuaria – VUP es un sistema integrado de procesos optimizados que permite, a través de medios electrónicos, asegurar la facilitación, el cumplimiento y el control eficiente de los procesos relacionados con la obtención de licencias, permisos y autorizaciones de servicios portuarios; y con los procesos vinculados a los servicios prestados a las naves y a su carga, que se desarrollan previo a la llegada, durante su estadía y previo a la salida.

El objetivo general de este componente es la facilitación de los procesos transaccionales a realizar por los actores que conforman las Comunidades Portuarias del Perú (Comunidades Portuarias de los puertos marítimos con tráfico internacional) frente a las entidades del Estado relativas a la entrada y salida de naves en los puertos, así como a la tramitación de las licencias de operación que requieren estos actores. (Citado en <https://www.vuce.gob.pe>).

## **1.3 Herramientas para el desarrollo de Aplicaciones.**

### **1.3.1 Lenguajes de Programación.**

¿Qué es un Lenguaje de programación?

Un lenguaje de programación es un lenguaje con una sintaxis y un léxico particular. Se utiliza para escribir programas que se ejecutan por medio de un computador. Cada lenguaje tiene su propia sintaxis, es decir, forma de escribirse que es diferente en cada uno por la forma que fue creado y por la forma que trabaja su compilador para revisar, acomodar y reservar el mismo programa en memoria. (MARTEENS, 2005) Los lenguajes de programación de alto nivel, facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas. Son herramientas que nos permiten crear software. Estos proporcionan un marco conceptual necesario para el desarrollo, análisis, optimización y comprensión de los



programas y, en general, de las tareas de programación. Entre ellos tenemos Delphi, Visual Basic, Pascal, Java, C#, C++, C, etc.

### 1.3.2 C++

C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C y le agrega la capacidad de abstraer tipos como en Smalltalk. Proporciona soporte a clases, objetos, herencia, es decir, todo lo característico de un lenguaje orientado a objetos. Estas cualidades facilitan la reutilización de código, consiguiendo con ello un ahorro de trabajo y tiempo. Una de sus principales características está dada en el soporte de plantillas o programación genérica (templates).

Posee un entorno de desarrollo simple, flexible y potente al mismo tiempo, especializado en la creación de proyectos Windows. Tiene un compilador rápido y brinda la posibilidad de obtener un ejecutable que puede ser utilizado aún cuando el programa no se encuentre instalado en la computadora.

El lenguaje tiene como conceptos clave, entre otros, la clase (class), que facilita la creación de tipos de datos definidos por el usuario juntamente con funciones o métodos para tratar con ellos, el encapsulamiento de datos, la asignación dinámica de memoria y la sobrecarga de operadores. Se diseñó explícitamente para lograr una mayor flexibilidad en la programación avanzada, aunque manteniendo un grado de simplicidad apreciable, basándose para todos los efectos en el lenguaje C, de ya muy amplia difusión, y en la metodología básica del lenguaje simula, en lo referente a la programación orientada a objetos.

### 1.3.3 Java

La sintaxis del lenguaje heredó características de C y C++, adoptando una muy similar a la del C++. Actualmente, dentro de los lenguajes populares es uno de los mejores en cuanto a definición, debido a que goza de total independencia del implementador del lenguaje y de sus clases auxiliares.



Proporciona los tipos de datos primitivos similares a los de C++, sólo que carece de punteros y mediante su biblioteca de clases estándar proporciona todas las estructuras contenedoras "clásicas" antes mencionadas. Tiene cuatro niveles de empaquetamiento: variables y funciones, al igual que los lenguajes anteriores y otros dos propios de él, denominados: clases y paquetes.

Este lenguaje cuenta con una interfaz orientada a objetos para acceder de un modo portable a cualquier base de datos, promoviendo la portabilidad. Es poseedor de una extensa biblioteca utilitaria y la portabilidad alcanzada es cualitativamente superior a la que se puede obtener con los lenguajes C/C++.

Java fue creado desde el inicio para ser ejecutado en cualquier clase de dispositivo o CPU, y uno de sus aspectos más interesantes es que no se compila directamente en el lenguaje máquina del CPU en uso, sino en un "pseudo lenguaje máquina" denominado "byte code", que puede ser transportado a cualquier computador en el cual se dispone de un programa especial encargado de la traducción del "byte code" al verdadero "lenguaje máquina" del CPU en uso.

#### 1.3.4 Delphi

La plataforma de implementación (IDE) Delphi en su versión 7 está diseñada para la programación de alto nivel y de propósito general. Tiene como basamento el lenguaje Pascal. A pesar de contar con un editor de HTML, XML y Servicios y Aplicaciones Web, su principal uso está dado por la realización de aplicaciones de escritorio. Es un Software propietario. Sirve de interfaz para generar entornos gráficos, aplicaciones multimedia, Internet, gestión de datos, de sistemas, de redes, etcétera. Permite intercambiar, transformar y manipular documentos y datos XML. En la construcción de aplicaciones nativas Windows,

Delphi permite una fácil creación y reutilización de DLLs, controles COM, así como el desarrollo de licencias comerciales para ventas de software profesional. Construye y utiliza Windows COM, COM+, ActiveX y automatización de objetos. En sus diferentes variantes, permite producir



archivos ejecutables para Windows, Linux y la plataforma .NET. Existe una versión de Delphi para sistemas Unix y Linux, denominada Kylix.

El programador de Delphi cuenta con una gran cantidad de componentes para realizar la conexión, manipulación, presentación y captura de los datos, algunos de ellos liberados bajo licencias de códigos abiertos o gratuitos. Estos componentes de acceso a datos pueden enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos, gracias al polimorfismo.

## **1.4 Sistemas de Gestión de Bases de Datos.**

### **1.4.1 ¿Qué es un Sistema Gestor de Bases de Datos?**

Según (MATO, 2006) “El software que permite la utilización y la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina Sistema Gestor de Bases de Datos (SGBD). Su principal objetivo es suministrar al usuario herramientas que le permitan manipular, en términos abstractos, los datos, o sea de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.” Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Entre los más conocidos y utilizados se encuentran Oracle, PostgreSQL, MySQL, SQL Server.

### **1.4.2 Oracle.**

Oracle es una de las mejores bases de datos que existen en el mercado, es un SGBD robusto, tiene muchas características que garantizan la seguridad e integridad de los datos, que las transacciones se ejecuten de forma correcta sin



causar inconsistencias, ayuda a administrar y almacenar grandes volúmenes de datos, estabilidad, escalabilidad y es multiplataforma. Es un producto vendible a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo sea utilizado por empresas muy grandes y multinacionales, por norma general.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de gestores de bases de datos comerciales y de la oferta de otros con licencia software libre como PostgreSQL, MySQL o FireBird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux. El principal inconveniente de utilizar Oracle como SGBD es el alto precio de su obtención.

### 1.4.3 MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. El objetivo que se persigue con MySQL es que cumpla con el estándar de Lenguaje de Consulta Estructurado (SQL), pero sin sacrificar velocidad, fiabilidad o usabilidad. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. En las aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a este SGBD ideal para las aplicaciones de este tipo. Es un sistema de administración relacional de bases de datos, ya que archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. MySQL es software de código abierto, usa Licencia Pública General (GNU/GPL) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Usar MySQL tiene ventajas adicionales:



- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- Conectividad: Permite conexiones entre diferentes máquinas con distintos sistemas operativos.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.
- Seguridad: Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

#### 1.4.4 PostgreSQL:

PostgreSQL está considerado como el gestor de base de datos de software libre más potente, se encuentra a la misma altura que otros SGBD como SQL server y Oracle. Es al igual que MySQL, un software de código abierto. Funciona sobre distintas plataformas Unix y Windows, así como en sistemas de tiempo real para microprocesadores del tipo x86. Tiene la posibilidad de manejar grandes bases de datos, (Se han llegado a mantener bases de datos de hasta de 60 GB). PostgreSQL es un sistema objeto - relacional, pues incluye aspectos del paradigma orientada a objetos, tales como la herencia, tipos de datos, funciones, restricciones, triggers, reglas e integridad transaccional, aunque no llega a ser un gestor con orientación a objetos pura. Da la posibilidad de realizar una subconsulta en una consulta SQL. Presenta llaves primarias y la posibilidad de que campos de una tabla sean únicos. En el caso de las llaves extranjeras, permite las operaciones de borrar o modificar en cascada a partir de ellas. Implementa los triggers como reglas, las cuales se aplican en los casos de ocurrencia de las operaciones de SELECT, INSERT, DELETE y UPDATE.

Entre sus principales ventajas están:





- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones.
- Tiene mejor soporte para triggers y procedimientos en el servidor que MySQL.
- Soporta un subconjunto de SQL92 mayor que el MySQL.

#### **1.4.5 Comparación entre PostgreSQL y MySQL.**

MySQL y PostgreSQL son, sin duda alguna, los SGBD más empleados en el mundo. Algunas diferencias entre ellos quedan expuestas a continuación:

- MySQL es más rápida que PostgreSQL a la hora de resolver consultas.
- MySQL tiene mejor documentación y se ha orientado más a facilitarle la vida al desarrollador proporcionando mejores herramientas de administración.
- PostgreSQL ofrece una garantía de integridad en los datos mucho más fuerte que MySQL.
- Aunque sea más lenta respondiendo a una única consulta, PostgreSQL presenta una mejor escalabilidad y rendimiento bajo grandes cargas de trabajo.
- PostgreSQL consume más recursos que MySQL.

### **1.5 Fundamentación de la selección de las herramientas.**

#### **1.5.1 ¿Por qué seleccionar C++?**

Al estudiar todos los demás lenguajes de programación y realizar una comparación entre ellos de manera general se decidió usar C++, por las ventajas que este proporciona. Posibilita redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales. C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto en la programación a alto como a bajo nivel. Es un lenguaje multiparadigma. Es multiplataforma, es decir no



existirá limitaciones por el sistema operativo que utilice la PC en que se encuentre.

### **1.5.2 ¿Por qué seleccionar PostgreSQL?**

Al estudiar todos los demás SGBD y realizar una comparación entre ellos de manera general se decidió usar PostgreSQL, por las ventajas que este proporciona. Referente a la seguridad este SGBD es recomendable para sistemas que no requieran la pérdida ni siquiera de un solo registro. Es tan potente que se compara con SGBD de alto nivel como Oracle y SQL Server y como ventaja sobre ellos se tiene que es un software libre. Es multiplataforma, es decir no existirá limitaciones por el sistema operativo que utilice la PC en que se encuentre la base de datos.

## **1.6 Metodología de desarrollo de sistemas informáticos**

### **1.6.1 Metodología propuesta para el desarrollo de la aplicación**

Hoy en día con el auge de la tecnología, y con el objetivo de agilizar y automatizar los procesos en el desarrollo de software, nos vemos en la necesidad de implantar Metodologías de Desarrollo de Software que nos ayuden a entregar un producto de calidad en tiempo y costo estimados, las metodologías ágiles de desarrollo de software han despertado interés gracias a que proponen simplicidad y velocidad para crear sistemas.

Las metodologías tradicionales no se adaptan a las nuevas necesidades o expectativas que tienen los usuarios hoy en día, en parte que los métodos usados no son flexibles ante la posibilidad de la exigencia de nuevos requerimientos. Estos cambios generalmente implican altos costos, demanda de tiempo y la reestructuración total del proyecto que se esté llevando; en contraparte, los métodos ágiles permiten un desarrollo iterativo y adaptable que



permite la integración de nuevas funcionalidades a lo largo del desarrollo del proyecto; para que tanto el cliente como el desarrollador queden satisfechos porque el producto final tiene una calidad adecuada.

Este enfoque ha mostrado su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, no obstante, mantiene una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

### **1.6.2 Rational Unified Process (RUP)**

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental. Según [Kruchten, 2000], los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo. Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo. Basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso.



La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [Kruchten, 2000].

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

### **1.6.3 SXP**

SXP es un híbrido cubano de metodologías ágiles que tiene como base las metodologías SCRUM y XP que permiten actualizar los procesos de desarrollo de software para el mejoramiento de su producción. Consta de 4 fases: Planificación-Definición, Desarrollo, Entrega y Mantenimiento, cada una desglosada en flujos de trabajo y actividades que generan artefactos.

Esta metodología ayuda a fortalecer el trabajo en equipo, enfocados en una misma dirección, permitiendo además seguir de forma clara el avance de las tareas a realizar, a partir de la inserción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la producción, aumentando el nivel de interés del equipo.

La metodología SXP durante sus dos años de despliegue y ejecución ha sido estudiada, refinada, adaptada y aún le queda la profunda convicción al grupo de



investigadores de la misma que es perfectible aún más, sin perder de vista su carácter ágil y revolucionador.

SXP, es un híbrido cubano de metodologías ágiles, que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad.

#### **1.6.4 Programación extrema (XP).**

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.



La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. Los Valores originales de la programación extrema son: simplicidad, comunicación, retroalimentación (feedback) y coraje. Un quinto valor, respeto, fue añadido en la segunda edición de Extreme Programming Explained. Los cinco valores se detallan a continuación:

Las características fundamentales del método son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.



- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Hay diversas prácticas inherentes a la Programación Extrema, en cada uno de las fases de desarrollo del proyecto.

- Planificación del proyecto: Historias de usuario, planificación de entregas, iteraciones, velocidad del proyecto, reuniones diarias.
- Diseño: Diseños simples, Riesgos, Tarjetas C.R.C.
- Codificación: refactorizar, programación en pareja.
- Pruebas: Test de aceptación.

#### **1.6.4.1 Fases de la metodología XP**

##### **Fase I: Planificación**

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para historias de usuarios (user stories) y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (release planning), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se



desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.

- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

### **Fase II: Diseño**

- Se eligen los diseños funcionales más simples.
- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

### **Fase III: Codificación**

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.
- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje





con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.

- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

#### **Fase IV: Pruebas**

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las user stories elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que el correspondiente user stories se ha completado.

### **1.7 Patrones arquitectónicos**

#### **1.7.1 Arquitectura por Capas**

Una de las técnicas más comunes que para descomponer un problema informático complejo. La capa superior utilizan los servicios definidos por la inferior, pero la capa inferior no es consciente de la capa superior. Además cada capa normalmente esconde los niveles más bajos de las capas por encima.

Ventajas:



- Sólo es necesario conocer la capa superior y es posible despreocuparse de las otras capas por debajo.
- Se pueden reemplazar capas por implementaciones alternativas de los mismos servicios básicos.
- Se minimizan las dependencias entre capas.
- Una capa puede ser reutilizada por múltiples servicios de niveles superiores.

Desventajas:

- Posibilidad de cambios en cascada (p.e. nuevo campo en la interfaz de usuario supone cambio en la base de datos y en todas las capas intermedias).
- Las capas extra pueden reducir el rendimiento.

La parte más complicada es decidir que capas y que funcionalidad poner en cada capa.

### **1.7.1.1 La Arquitectura de 3 Capas**

Esta arquitectura, la más común en la actualidad, está compuesta por las siguientes capas:

- Presentación: comprende la lógica para manejar la interacción entre el usuario y la aplicación. Posibles opciones son la línea de comandos, sistemas de menús basados en texto, el cliente rico o la interfaz de usuario HTML.
- Fuentes de Datos: tiene que ver con la comunicación con otros sistemas que llevan a cabo tareas en nombre de la aplicación. Monitores de transacciones, otras aplicaciones, sistemas de mensajería, pero generalmente base de datos.
- Lógica de Dominio: la lógica de dominio o empresarial es la funcionalidad específica que la aplicación debe hacer para el dominio de trabajo. Los cálculos sobre la base de las entradas y los datos almacenados, la validación de los datos de entrada o determinar la fuente de datos implicada son ejemplos de funcionalidades en la capa de dominio.



Un aspecto difuso de esta arquitectura es lo que sucede cuando no hay una persona utilizando el software, por ejemplo un servicio web o un proceso por lotes (batch). En este caso se difumina la diferencia entre presentación y datos, ya que las dos se refieren a conexiones con el exterior de la aplicación. La distinción a hacer es entre una interfaz que la aplicación proporciona como un servicio a otros, y el uso que se haga de los servicios proporcionados por otros como fuentes de datos.

### **1.7.2 Patrón de arquitectura MVC (Modelo vista controlador)**

Para el diseño de aplicaciones con sofisticadas interfaces se emplea el patrón de diseño MVC. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces, la consecuencia será que, cuando se necesite cambiar la interfaz, tendrá que modificarse trabajosamente los componentes de negocio, por lo que propiciará mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con el fin de perfeccionar la reusabilidad. De este modo las modificaciones son las que impactan en menor medida en la lógica de negocio o de datos. (Jiménez, 2009)

Elementos del patrón:

- ✓ Modelo: datos y reglas de negocio.
- ✓ Vista: muestra la información del modelo al usuario.
- ✓ Controlador: gestiona las entradas del usuario.

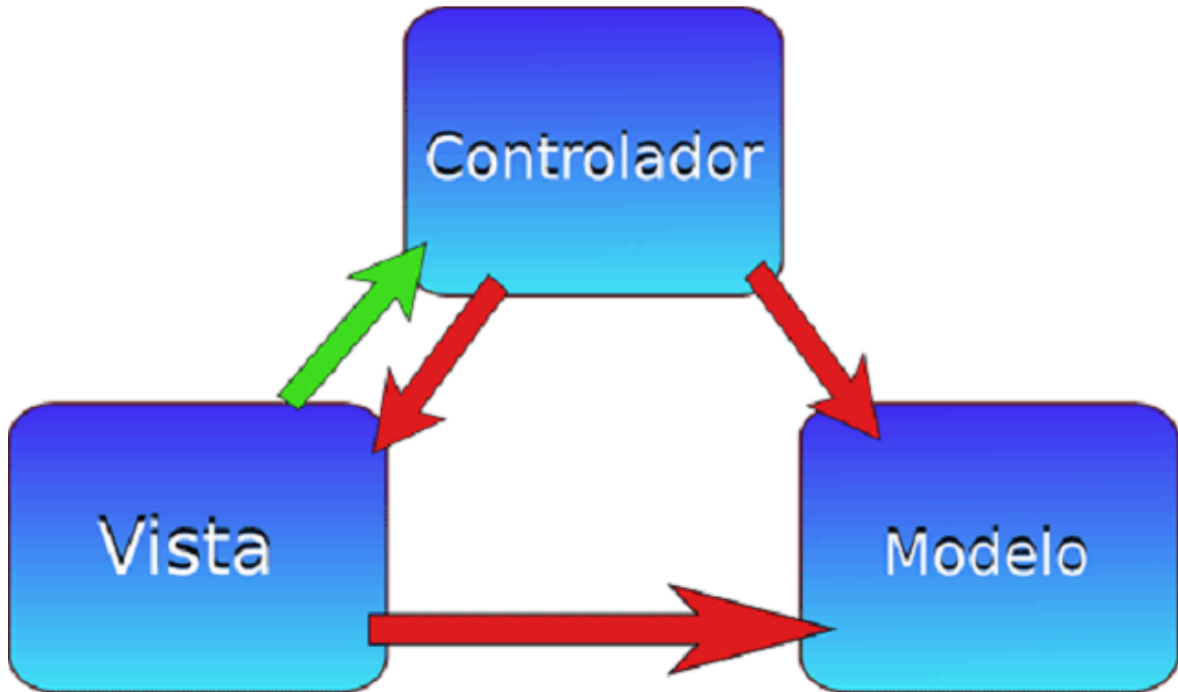


Figura 1.1 Modelo vista controlador

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información en una base de datos, que puede presentarse de diversas formas: diagrama de pastel, de barras, tabular, etc. Analizan cada componente:

El modelo es responsable de:

- ❖ Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- ❖ Definir las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “si la mercancía solicitada no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.
- ❖ Llevar un registro de las vistas y controladores del sistema.
- ❖ Si se está en presencia un modelo activo, el mismo notificará a las vistas los cambios que en los datos pueda producir un agente externo (ejemplo: un fichero bat que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador es responsable de:



❖ Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.). Contiene reglas de gestión de eventos, del tipo “si el evento z, entonces acción w”. estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar.

Las vistas son responsables de:

- ❖ Recibir datos del modelo y mostrarlo al usuario.
- ❖ Tienen un registro de su controlador asociado, (normalmente porque además lo demanda).
- ❖ Pueden suministrar el servicio de actualizar, para que sea solicitado por el controlador o por el modelo, (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

## **1.8 Lenguajes de Modelado.**

### **1.8.1 UML**

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componente reutilizables. Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como



el Proceso Unificado Racional o RUP), pero no especifica en sí mismo que metodología o proceso usar.

### **1.9 Conclusiones del capítulo.**

En este capítulo se hace un estudio sobre el funcionamiento de las operaciones a buques particularizando nuestro problema en especial y se exponen los conceptos necesarios para el entendimiento del software, se toman decisiones importantes, luego de un estudio de las distintas herramientas para el desarrollo de software tales como la elección de los lenguajes de programación y metodologías a utilizar.



## Capítulo 2 Planificación y Diseño

### 2.1 Introducción

En este capítulo, se introduce la fase de planificación y diseño, donde se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de informatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema, para su terminación.

### 2.2 Funcionalidades generales

La gestión del registro y operaciones a buques de la Empresa Consignataria Mambisa se realiza de forma manual, lo que hace el proceso más lento, con el consiguiente aumento de gastos de papel y otros materiales de oficina e incrementa la posibilidad de que se produzcan errores en el momento de manipular la información de los mismos, lo que trae como consecuencia afectaciones en la gestión del mismo debido a que pueden ejecutarse servicios cuyos registros no estén vigentes.

De la descripción antes expuesta, se identifican como requisitos generales del sistema, el análisis del estado del registro y operaciones a buques y el control de su ejecución.

Análisis del estado de los registros y operaciones a buques y control de su ejecución.

Consiste en darle seguimiento a los registro de operaciones que se encuentran en ejecución a través de la generación de reportes, los



mismos incluyen, conocer día que se emitió y para que buque fue emitido, la vigencia de los mismo, la caducidad, otras operaciones etc.

### 2.2.1 Personal relacionado con el sistema

Personas relacionadas con el sistema	Justificación
Especialista	Es la persona que tiene conocimiento en la materia de registro y operaciones a buques y está encargada de la gestión de información de los mismos.
Administrador	Se encarga de asesorar y dar seguimiento al estado del proceso de desarrollo.
Desarrollador	Es la persona responsable de llevar a cabo la implementación del sistema.

Tabla 2.1 Personas relacionadas con el sistema

### 2.2.2 Lista de reserva

Después de conocer el personal relacionado e identificar los requisitos generales, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para dar respuesta a los mismos. Para ello se enumerarán mediante una lista de reserva, las funcionalidades que el sistema debe ser capaz de cumplir.

De acuerdo a lo antes expuesto, el sistema debe ser capaz de:

- ❖ Insertar datos en el registro de buques.
- ❖ Agregar nuevos buques.
- ❖ Modificar el registro de buques.
- ❖ Mostrar los próximos arribos de los buques.
- ❖ Actualizar la información de los buques.
- ❖ Registrar buques.
- ❖ Actualizar los permisos de entrada.





- ❖ Actualizar los permisos de salida.
- ❖ Actualizar los permisos de descarga.
- ❖ Actualizar los permisos de remoción.
- ❖ Actualizar los reportes de arribos.
- ❖ Actualizar los estados de hechos de los buques.
- ❖ Actualizar las solicitudes de prácticos.
- ❖ Actualizar las solicitudes de remolcadores.
- ❖ Modificar estado de hecho de los buques.
- ❖ Listar información de los buques.
- ❖ Eliminar buques.
- ❖ Listar los próximos arribos de los buques.
- ❖ Listar los permisos de entrada.
- ❖ Listar los permisos de descarga.
- ❖ Listar los permisos de salida.
- ❖ Listar los permisos de remoción.

### 2.2.3 Historias de usuario

Las HU, son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas tormenta de ideas (brain storm) arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario.

Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. Para definir las se emplea la siguiente plantilla.



Modelo de planilla de historia de usuario:

Historia de usuario	
<b>Número:</b> No. Historia de usuario	<b>Usuario:</b> Usuario entrevistado para obtener la función requerida a automatizar.
<b>Nombre:</b> nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente.	
<b>Prioridad en el negocio:</b> Importancia: Alta / Media / Baja	<b>Riesgo en desarrollo:</b> Dificultad: Alta / Media / Baja
<b>Puntos estimados:</b> Estimación: de 1 a 3 puntos	<b>Iteración asignada:</b> Iteración a la que corresponde
<b>Programador responsable:</b> Nombre de encargado de programación.	
<b>Descripción:</b> Se especifican las operaciones por parte del usuario y las respuestas del sistema.	
<b>Observaciones:</b> Algunas observaciones de interés, como glosario, información sobre usuario etc.	

Tabla 2.2 Planilla de historia de usuario

HU N° 1: Gestión de los próximos arribos

Historia de Usuario	
Número: 1	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión de los próximos arribos	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 1
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos de los buques que arribaran a puerto en las próximas 24 horas, 3 días y 7 días, además debe ser almacenada la información de los responsables de los buques y la mercancía a descargar. Una vez añadidos los buques, se podrá mostrar actualizar la información de los mismos.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.2 HU No 1 Gestión de los próximos arribos

HU N° 2: Registro de Buques



Historia de Usuario	
Número: 2	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión del registro de buques	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 1
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos de los buques, además debe ser almacenada la información de los trabajos hechos por el buque. Una vez añadidos los buques, se podrá mostrar, actualizar y eliminar la información de los mismos.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.3 HU No 2 Gestión del registro de buques

Ver **Anexo 2: Historias de Usuarios**

### 2.3 Planificación de entregas

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de



estimación correspondientes a las HU, que fueron terminadas en la última iteración.

### 2.3.1 Estimación de esfuerzo por historias de usuario

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las HU identificadas, y se obtienen los resultados que se muestran a continuación:

Estimación de esfuerzo por historia de usuario

Historias de usuario	Puntos de estimación
Gestión de los próximos arribos	3 Semanas
Gestión del registro de buques	2 Semanas
Gestión del S.O.F.(Statements of Fact)	3 Semanas
Gestión de los reportes de arribo	2 Semanas
Gestión de los reportes de entrada	1 Semana
Gestión de los reportes de salida	1 Semana
Gestión de los reportes de descarga	1 Semana
Gestión de los reportes de remoción	1 Semana
Gestión de las solicitudes de remolcadores y prácticos	1 Semana

Tabla 2.4 Estimación de esfuerzo por historia de usuario

### 2.3.2 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este



plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un reléase, lo cual debe ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en 3 iteraciones, las cuales se explican de forma detalla a continuación:

### **Primera iteración:**

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir con dicha iteración se constara con todas las funcionalidades descritas en las HU 1 y 2, las cuales hacen alusión a la inserción y visualización de la información de los buques, la agregación de nuevos buques para los próximos arribos que estos tendrán a puerto. Además se tendrá la primera versión de prueba, que contará con dos modelos de desarrollo que incorporan todas las funcionalidades antes vistas, estos modelos se presentarán al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

### **Segunda iteración:**

Esta iteración tiene como finalidad el desarrollo de las HU 3 y 4. Las mismas son las que brindan las funcionalidades de mostrar y obtener toda la información de los buques en el puerto cuando arribo y cómo fue su estadía en dicho puerto y visualizar la información de estos. La versión que se obtenga de esta iteración en unión con la entregada en la iteración anterior se le facilitará al cliente para comprobar si cumple con las necesidades antes acordadas con él.

### **Tercera iteración:**

Esta última iteración del módulo tiene como propósito llevar a cabo el desarrollo de la HU 5, 6, 7, 8 y 9. Las cuales proporcionan los trámites que se le



realizan a los buques en el puerto, además esos se utilizarán para cada buque que toque dicho puerto. Estas HU serán integradas con el resultado de las iteraciones anteriores, y como fruto de esta integración se obtendrá la versión 1.0 del producto final. A partir de este momento el software será puesto a un proceso de prueba para evaluar el desempeño del mismo.

### 2.3.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto guiado por la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas.

Plan de duración de las iteraciones

Iteración	Historias de usuario	Duración total
Iteración 1	Gestión de los próximos arribos	5 semanas La entrega se realizará en la primera semana de marzo
	Gestión del registro de buques	
Iteración 2	Gestión del S.O.F.	5 semanas La entrega se realizará en la segunda semana de abril
	Gestión de los reportes de arribo	
Iteración 3	Gestión de los reportes de entrada	5 semanas La entrega se realizará en la tercera semana de mayo
	Gestión de los reportes de salida	
	Gestión de los reportes de descarga	
	Gestión de los reportes de remoción	
	Gestión de las solicitudes de remolcadores	



y prácticos

Tabla 2.5 Plan de duración de las iteraciones

**2.4 Clases, responsabilidades y colaboradores**

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como tarjetas CRC, las cuáles se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño. Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.

**Tarjetas CRC**

Tarjeta CRC N° 1

Nombre de la clase: Próximos Arribos	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar datos de los próximos arribos	Nuevos_Proximos_Arribos
Modificar datos de los próximos arribos	Mod_Prox_Arribos
Listar los próximos arribos	Mostrar_arribos
Eliminar los próximos arribos	Elim_Arribos

Tabla 2.13: Tarjetas CRC N° 1

**Ver Anexo 3: Tarjetas CRC**



## 2.5 Conclusiones

En este capítulo se abordó la fase de planeación y diseño donde se delinearón las HU con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas y además, las principales clases mediante el empleo de las tarjetas CRC, culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.





## Capítulo 3 Desarrollo y pruebas.

### 3.1 Introducción

En este capítulo se inicia la fase de desarrollo y pruebas conforme a la metodología XP. Se presenta el modelo de datos empleado para la aplicación concluyente, se presentan además las tarjetas de tareas, clases, responsabilidades y colaboradores, que permitirán trabajar con una metodología basada en objetos, se realiza el desarrollo de las iteraciones a partir del desglose de las HU en tareas. Se muestran las interfaces gráficas de usuario diseñadas para la aplicación final. Se describen igualmente las pruebas realizadas y se indican las respuestas de la aplicación en el empleo de las diferentes funcionalidades, así como los posibles mensajes de error, información o aceptación que emite la misma cuando se utiliza una de estas funcionalidades.

### 3.2 Modelo de Datos

En esta parte se muestra el modelo de datos empleado para la aplicación. Donde los datos son almacenados en la tabla de los nuevos\_proximos\_arribos. Cuando se realiza un reporte de entrada se guarda en la tabla de los RP\_Entrada para mostrar el reporte correspondiente a cada barco. En el RP\_Salida se almacenan los datos del buque que parte del puerto. RP\_Descarga se almacena los datos que se solicitan a la tabla de nuevos\_proximos\_arribos. La tabla RP\_Remoción almacena los datos del buque que va a ser removido de lugar. La tabla RP\_Remolcador almacena los datos del buque y el remolcador que se le va a asignar para la maniobra. La tabla RP\_Practicos se utiliza para almacenar los datos los prácticos que participaran en la maniobra del buque. La tabla de Registro\_Buques almacena los datos del buque que parte del puerto y como este se va del puerto.

Ver **Anexo 4: Modelo de datos**



### 3.3 Desarrollo de las iteraciones.

Durante la fase planificación y diseño fueron detalladas las HU correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

#### 3.3.1 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las tarjetas de tareas.

#### Distribución de tareas por historia de usuario

Historia de usuario	Tareas
Gestión de Próximos Arribos	<ul style="list-style-type: none"><li>• Insertar Próximos Arribos</li><li>• Mostrar Próximos Arribos</li><li>• Modificar Próximos Arribos</li><li>• Eliminar Próximos Arribos</li></ul>
Gestión del Registro de Buques	<ul style="list-style-type: none"><li>• Insertar los datos del buque en el registro.</li><li>• Mostrar los datos del buque en el registro.</li><li>• Modificar los datos del buque en el registro.</li><li>• Eliminar los datos del buque en el registro.</li></ul>
Gestión del S.O.F.	<ul style="list-style-type: none"><li>• Insertar los datos al S.O.F</li><li>• Mostrar los datos del S.O.F</li></ul>



Gestión del reporte de entrada	<ul style="list-style-type: none"> <li>• Eliminar los datos del S.O.F</li> <li>• Insertar datos en el reporte de entrada</li> <li>• Mostrar el reporte de entrada</li> <li>• Eliminar reporte de entrada</li> </ul>
Gestión del reporte de salida	<ul style="list-style-type: none"> <li>• Insertar datos en el reporte de salida</li> <li>• Mostrar el reporte de salida</li> <li>• Eliminar reporte de salida</li> </ul>
Gestión del reporte de remoción	<ul style="list-style-type: none"> <li>• Insertar datos en el reporte de remoción</li> <li>• Mostrar el reporte de remoción</li> <li>• Eliminar reporte de remoción</li> </ul>
Gestión del reporte de descarga	<ul style="list-style-type: none"> <li>• Insertar datos en el reporte de descarga</li> <li>• Mostrar el reporte de descarga</li> <li>• Eliminar reporte de descarga</li> </ul>
Gestión del reporte de arribo	<ul style="list-style-type: none"> <li>• Insertar datos en el reporte de arribo</li> <li>• Mostrar el reporte de arribo</li> <li>• Eliminar reporte de arribo</li> </ul>
Gestión de la solicitud de remolcadores y prácticos	<ul style="list-style-type: none"> <li>• Insertar datos en la solicitud de prácticos y remolcadores.</li> <li>• Mostrar la solicitud de prácticos y remolcadores.</li> <li>• Eliminar solicitud de prácticos y remolcadores.</li> </ul>

Tabla 3.1 Distribución de las tareas por historia de usuarios.

Historias de usuario abordadas en la primera iteración

Historias de Usuarios	Tiempo de estimación (por semanas)	
	Estimación Inicial	Real
Gestión de los próximos arribos	3	2.5
Gestión del registro de buques	2	1.5

Tabla 3.2 Historias de usuario abordadas en la primera iteración.



Historias de usuario abordadas en la segunda iteración

Historias de Usuarios	Tiempo de estimación (por semanas)	
	Estimación Inicial	Real
Gestión del S.O.F	3	2.5
Gestión del reporte de arribo	2	1.5

Tabla 3.11 Historias de usuario abordadas en la segunda iteración

Historias de usuario abordadas en la tercera iteración

Historias de Usuarios	Tiempo de estimación (por semanas)	
	Estimación Inicial	Real
Gestión de los reportes de entrada	1	1
Gestión de los reportes de salida	1	1
Gestión de los reportes de descarga	1	1
Gestión de los reportes de remoción	1	1
Gestión de las solicitudes de remolcadores y prácticos	1	1

Tabla 3.18 Historias de usuario abordadas en la tercera iteración

**Ver Anexo 5 Tarjetas de Tareas**

### 3.4 Pruebas

En la Programación Extrema es esencial el desarrollo de las pruebas, permitiendo probar continuamente el código. Cada vez que se desea implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba, para ser pasada por el código posteriormente. El proceso constante de las pruebas permite la obtención de un producto con mayor calidad, y se ofrece a los programadores una mayor certeza en el trabajo que desempeñan. En la metodología XP hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas (TDD test driven development), desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración, se



verifica así, que se cumplió la funcionalidad requerida por el cliente. Con estas normas se obtiene un código simple y funcional de manera bastante rápida y eficiente. Por esto es importante pasar las pruebas al 100%.

### **3.4.1 Desarrollo dirigido por pruebas**

El desarrollo dirigido por pruebas, se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso para lograr un resultado, aunque no con lógica para el negocio, pero si funcional. Algunas personas confunden este término con las llamadas “pruebas de caja blanca” las cuáles se les practican a los métodos u operaciones para medir la funcionalidad del mismo, desde el punto de vista de validez del cliente. Sin embargo, el TDD se aplica antes de comenzar a implementar cada paso de la tarea en desarrollo, asumiendo que la prueba es insatisfactoria desde un inicio. Sólo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida. Luego se realiza un proceso conocido como “refactorización” de código perteneciente a una de las doce prácticas planteadas por la metodología XP, el cual consiste en mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. Es esencia el TDD y las pruebas de caja blanca se enfocan en la lógica del negocio.

### **3.4.2 Pruebas de aceptación**

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que la HU ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades



solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los paradigmas anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación.

La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación.

Planilla de prueba de aceptación

<b>Prueba de aceptación</b>
<b>HU:</b> Nombre de la historia de usuario que va a comprobar su funcionamiento.
<b>Nombre:</b> Nombre del caso de prueba.
<b>Descripción:</b> Descripción del propósito de la prueba.
<b>Condiciones de ejecución:</b> Precondiciones para que la prueba se realice.
<b>Entrada/Pasos ejecución:</b> Pasos para probar la funcionalidad.
<b>Resultado:</b> Resultado que se desea de la prueba.
<b>Evaluación de la prueba:</b> Aceptada o denegada.

Prueba de aceptación
HU: Gestión de los próximos arribos.
Nombre: Prueba para comprobar la entrada de datos.
Descripción: Validación de la entrada de datos de los próximos arribos
Condiciones de ejecución: El usuario debe introducir los datos de los próximos arribos.
Entrada/Pasos ejecución: El usuario intentara introducir los datos de los próximos arribos
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li><li>• Se intente insertar un próximo arribo que ya existe.</li></ul>



- Se intente introducir un formato no válido.

Se emite un mensaje en caso de información de éxito al insertar el próximo arribo.

Evaluación de la prueba: Aceptada.

Tabla 3.34 PA: Prueba para comprobar la entrada de datos

Ver **Anexo 6 Prueba de Aceptación**

### 3.5 Conclusiones.

En este capítulo se llevó a cabo la fase de desarrollo y prueba donde se presenta el modelo de datos de la aplicación a obtener, logrando una visión detallada de sus atributos y las relaciones entre sus clases. Se realiza el desarrollo de las iteraciones a partir de la distribución de tareas por HU, y se les practica las pruebas de aceptación a las funcionalidades de mayor importancia.



## Capítulo 4 Estudio de Factibilidad

### 4.1 Introducción

El estudio de la factibilidad para este proyecto hace uso de la Metodología costo Efectividad (Beneficio), en la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores El costo: incluye la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados. La efectividad: Es la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo). (Pérez García, A. M.)

### 4.2 Efectos Económicos

#### 4.2.1 Efectos directos

- Positivos
- Se gestiona la información necesaria a la que los usuarios finales de la aplicación podrán acceder.
  - Los especialistas en las operaciones podrán insertar los datos que toman desde diferentes puntos.
  - Los diferentes usuarios que posean acceso a los diferentes reportes lo tendrán en tiempo y forma, pudiendo tomar medidas en caso de que las requiera.
  - Se aumenta la velocidad de obtención de los reportes.
- Negativos
- Para usar la aplicación es necesario la utilización de un ordenador, paralelo a los gastos de consumo de electricidad y mantenimiento que conlleva.





#### **4.2.2 Efectos indirectos**

- Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles.

#### **4.2.3 Externalidades**

- Se contará con una herramienta que permitirá a los usuarios finales acceder a la información necesaria de una forma segura y rápida.

#### **4.2.4 Intangibles**

- En la valoración económica siempre hay elementos perceptibles por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible. A fin de medir con precisión los efectos, deberán considerarse estas situaciones:

Beneficios:

- Mayor comodidad, organización e información para los usuarios: porque este va a posibilitar al usuario mejorar la rapidez en la confección de los reportes y seguridad e integridad de los datos
- Mayor integración usuarios-artefactos: porque el usuario tendrá más integración con el sistema.
- Mejora en la calidad y visibilidad de la información: porque el usuario tendrá más organizada la información de los buques en el puerto y los reportes se emitirán más rápidos.

### **4.3 Ficha de Costo**

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una Ficha de Costo de un Producto Informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costos en Moneda Libremente Convertible

- Costos Directos.



1. Compra de equipos de cómputo: No procede.
  2. Alquiler de equipos de cómputo: No procede.
  3. Compra de licencia de Software: No procede.
  4. Depreciación de equipos: \$ 10.00.
  5. Materiales directos: No procede.
  6. Gasto por consumo de energía eléctrica: \$ 15.00.
- Total: \$ 25,00 CUC

▪ Costos Indirectos.

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Total: \$0.00.

▪ Gastos de distribución y venta.

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00.

Costos en Moneda Nacional

▪ Costos Directos.

1. Salario del personal que laborará en el proyecto: \$110.00.
2. El 5% del total de gastos por salarios se dedica a la seguridad social: No procede.

3. El 0.09% del salario total, por concepto de vacaciones a acumular: No procede.

4. Gasto por consumo de energía eléctrica: No procede.
5. Gastos en llamadas telefónicas: No procede.
6. Gastos administrativos: No procede.

▪ Costos Indirectos.

1. Know How: No procede.



Total: \$110 MN

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología, la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tómesese como costo el tiempo empleado por el administrador para mantener la aplicación actualizada en todo momento.

### Punto de Equilibrio

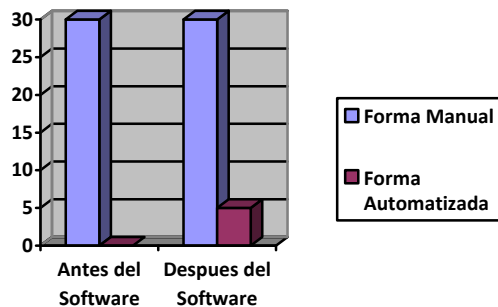


Fig. 4.1 Gráfica del punto de equilibrio

Variable sin sistema:

- Gestionar reportes de buques (30 Minutos).

Variable con el sistema:

- Gestionar reportes de buques (9 Variables (5 Minutos))

En la gráfica se muestra el tiempo que se empleaba para realizar los reportes que en la Empresa Consignataria Mambisa del municipio de Moa la barra de color morado representa el tiempo que se realiza el trabajo manuscrito y la barra de color rojo vino representa el trabajo con el software.



#### 4.4 Conclusiones del Capítulo

Este capítulo realizó el estudio de factibilidad mediante La Metodología Costo Efectividad (Beneficio), analizó los efectos económicos, los beneficios y costos intangibles, además se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$ 40.00 CUC y \$ 110 MN demostrándose la factibilidad del proyecto.



## Conclusiones Generales

El sistema se desarrolla siguiendo la metodología XP, y se utilizaron representaciones para la modelación de todas las fases del proyecto. El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos, para esto se siguieron los siguientes pasos:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, permitiendo el análisis del proceso de gestión de la información de los registros en el área de operaciones de la Empresa Cosignataria Mambisa para lograr una mejor comprensión del negocio.
- Se efectuó un estudio de las diferentes tecnologías y herramientas para la confección del sistema, haciendo una elección, de acuerdo a las especificaciones del cliente.
- Se efectuó un levantamiento de los requerimientos, los que propiciaron un mejor entendimiento de la problemática en cuestión, facilitando el análisis, diseño e implementación del sistema informático, el cual solucionó las dificultades antes expuestas.
- Se obtuvo un sistema informático que permite el control de la ejecución de los registros de las operaciones a buques.

Por todo lo anterior se concluye que los objetivos propuestos en el presente proyecto han sido cumplidos satisfactoriamente.



## Recomendaciones

De manera general los objetivos trazados al inicio de esta investigación han sido logrados, al mismo tiempo, en el transcurso del proceso de desarrollo, ha quedado evidenciado, que la propuesta es sólo la primera fase de un proyecto que puede ser mucho más ambicioso. Por tanto se hacen las siguientes recomendaciones.

- Continuar trabajando en el sistema con el objetivo de integrarlo a los servicios de la entidad.
- Extenderlo a las demás empresas destinadas a este objeto social.



## Referencia Bibliográfica

Basulto Aguilera, Jorge Mario. "SISTEMA DE GESTION INTEGRAL DE LA EMPRESA EMPLEADORA DEL NIQUEL - MODULO GESTION DE CONTRATOS DE COMPRAS."

Calero Solís Manuel. Una explicación de la programación extrema (XP).pdf

Gerardo Fernández Escribano Introducción a Extreme Programming 9-12-2002

Kruchten, P., the Rational Unified Process: An Introduction, 2000 Addison Wesley

LARMAN, C. UML y patrones. Introducción al análisis y diseño orientado a objeto.3 ed. La Habana: Editorial Félix Varela, 2004. 507p

MARTEENS, I. La cara oculta de Delphi. La Habana: Editorial Félix Varela, 2005. 296 p.

MATO, R.M. Sistemas de Bases de Datos. 2 ed. La Habana: Editorial Félix Varela, 2006.165p

Programación Extrema. Fases de la Programación Extrema. [En línea]. <http://programacionextrema.tripod.com/fases.htm>.

REYNOSO, C; KICCILLOF, N. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [en línea]. Disponible en: [www.willydev.net/descargas/prev/Estiloypatron.pdf](http://www.willydev.net/descargas/prev/Estiloypatron.pdf)



Wikipedia. UML. [En línea].

<http://es.wikipedia.org/wiki/UML>.

Wikipedia. Programación extrema. [En línea].

[http://es.wikipedia.org/wiki/Programación\\_extrema](http://es.wikipedia.org/wiki/Programación_extrema).

VUCE (Ventanilla única de Comercio Exterior) [en línea] Disponible en:  
<https://www.vuce.gob.pe/index.html>.





## GLOSARIO DE TERMINOS MARITIMO - PORTUARIOS

1. **Abarloamiento:** Operación de amarrar una nave a otra que se encuentra atracada a muelle o fondeada en el área de operaciones acuática del Terminal.
2. **Amarre y Desamarre:** Servicio que se presta a las naves en el amarradero para recibir y asegurar las amarras, cambiarlas de un punto de amarre a otro y largarlas.
3. **Apilar:** Colocar en forma ordenada la carga una sobre otra en las áreas de almacenamiento.
4. **Área:** Superficie delimitada por un perímetro, la cual es destinada para un uso determinado de los servicios y facilidades portuarias.
5. **Armador:** Propietario y/o quien dirige la explotación de la nave.
6. **Arqueo Bruto:** Es la expresión del volumen total de una nave, determinada de acuerdo con las disposiciones internacionales y nacionales vigentes.
7. **Arrumaje:** Colocación transitoria de la mercancía en áreas cercanas a la nave.
8. **Atraque:** Operación de conducir la nave desde el fondeadero oficial del puerto y atracarla al muelle o amarradero designado.
9. **Calado:** Es la profundidad sumergida de una nave en el agua.
10. **Consignatario:** Persona natural o jurídica a cuyo nombre viene manifestada la mercancía o que la adquiere por endoso.
11. **Desatraque:** Operación inversa al atraque.
12. **Descargar:** Es el proceso inverso al de cargar.
13. **Despacho:** Cumplimiento de las formalidades aduaneras necesarias para importar y exportar las mercancías o someterlas a otros regímenes, operaciones o destinos aduaneros.



## Anexo 1 Interfaz de Usuario



Fig. 1.1 Portada del Sistema



**Nuevos Próximos Arribos**

Buque	Bandera	Moneda	Procedencia
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Tipo de Carga	Eslora	T. Brutas	T. Netas
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
DWT	Manga	Armador	Calados Proa/Popa
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Medios Isaje Grua/Punta	Capitan	Nacionalidad	Tripulaci?n
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Desplazamiento	?ltimo Puerto	Terminal	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
ETA / POSICI?N DE EMITIDO	N? de viajes	Cantidad de Carga	
<input type="text"/>	<input type="text"/>	<input type="text"/>	

 Insertar  OK

Fig. 1.2 Interfaz de los Nuevos Próximos Arribos

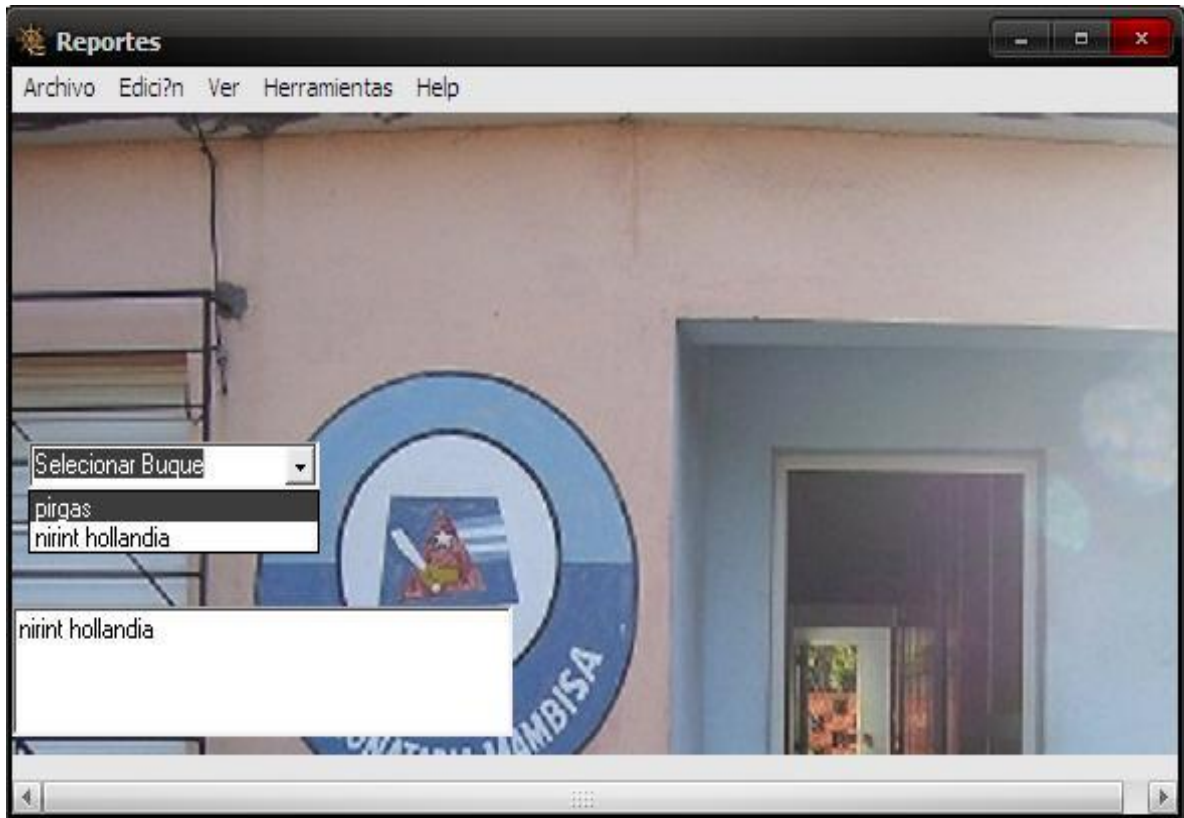


Fig. 1.3 Interfaz de los Reportes

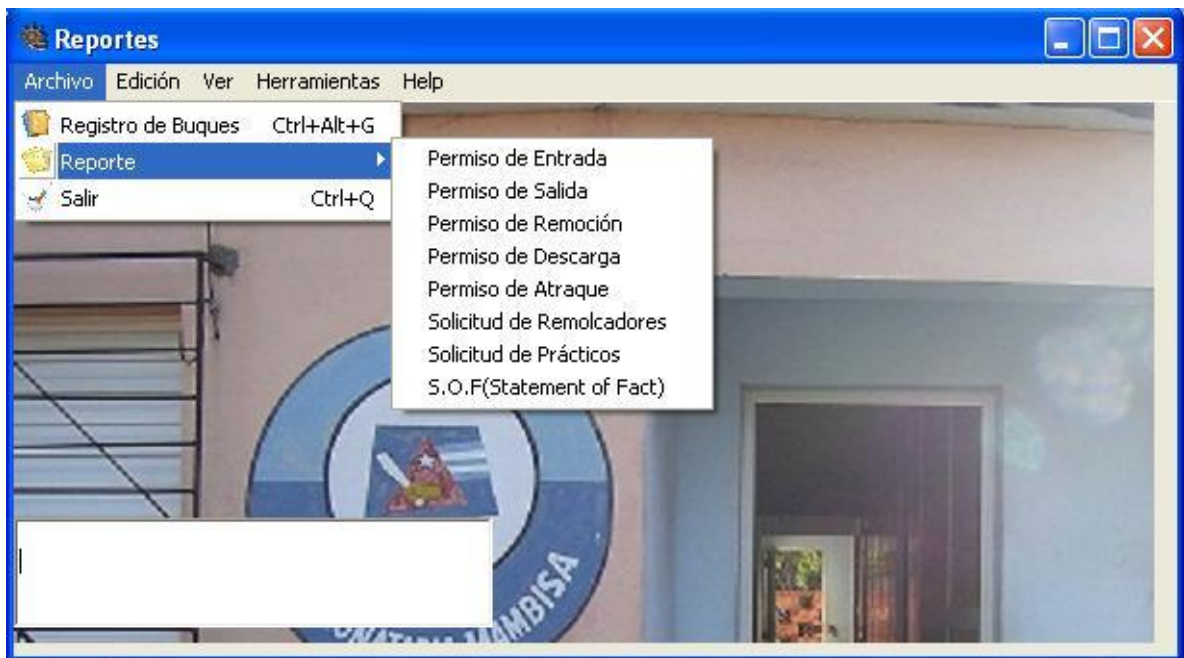


Fig. 1.4 Menús de los reportes.



N°	Nombre del Buque	Toneladas Brutas	Toneladas Netas
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
L.O.A.	Registro	Manifiesto	Descargo
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Cargo	T.Carga	Arribo	Atraco
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Comenzo	Termino	Salio	Agente
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
T/G	Bandera		
<input type="text"/>	<input type="text"/>		

Fig. 1.5 Registro de buques





## Anexo 2: Historias de Usuarios

### HU N° 3: Gestión del S.O.F.

Historia de Usuario	
Número: 3	<b>Usuario:</b> Especialista en Operaciones a buques
Nombre de historia: Gestión del S.O.F.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 2
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque así como las operaciones que se le realizan a este en su estadía en el puerto.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.6 HU No 3 Gestión del S.O.F.

### HU N° 4: Gestión del Reporte de Arribo

Historia de Usuario	
Número: 4	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión del Reporte de Arribo	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque después que este arribe al puerto.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.7 HU No 4 Gestión del Reporte de Arribo



HU N° 5: Gestión de los reportes de Entrada

Historia de Usuario	
Número: 5	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión de los reportes de Entrada	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 3
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque para emitir un reporte antes de que este entre al puerto.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.8 HU No 5 Gestión de los reportes de Entrada

HU N° 6: Gestión de los reportes de Salida

Historia de Usuario	
Número: 6	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión de los reportes de Salida	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 3
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque para emitir un reporte antes de que este salga del puerto.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.9 HU No 6 Gestión de los reportes de Salida



HU N° 7: Gestión de los reportes de Descarga

Historia de Usuario	
Número: 7	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión de los reportes de Descarga	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 3
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque para emitir un reporte antes de que este comience a descargar la carga que trae.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.10 HU No 7 Gestión de los reportes de descarga

HU N° 8: Gestión de los reportes de Remoción

Historia de Usuario	
Número: 8	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión de los reportes de Remoción	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 3
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque para emitir un reporte antes de que este salga del puerto.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.11 HU No 8 Gestión de los reportes de Remoción





HU N° 9: Gestión de los reportes de Salida

Historia de Usuario	
Número: 9	<b>Usuario:</b> Especialista en Operaciones a buques
<b>Nombre de historia:</b> Gestión de las solicitudes de remolcadores y prácticos	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteración Asignada: 3
Programador responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> El usuario debe insertar los datos del buque para emitir un reporte varios reportes para solicitar del práctico que va a efectuar la maniobra de entrada al canal y el respectivo remolcador que será usado en esta maniobra.	
<b>Observaciones:</b> Confirmado con el cliente.	

Tabla 2.12 HU No 9 Gestión de las solicitudes de remolcadores y prácticos



### Anexo 3: Tarjetas CRC

#### Tarjeta CRC N° 2

<b>Nombre de la clase:</b> Registro de buques	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Insertar datos en el registro de buques	Registro_buques
Modificar datos del registro de buques	Mod_Reg_Buques
Listar el libro de los registros de buques	Mostrar_Reg_Buques
Eliminar el libro mensual del registro de buques	Elim_Reg_Buques

Tabla 2.14: Tarjetas CRC N° 2

#### Tarjeta CRC N° 3

<b>Nombre de la clase:</b> S.O.F.(Statement of Fact)	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Insertar datos en el S.O.F.	Insert_S.O.F.
Modificar datos del S.O.F.	Mod_S.O.F.
Listar el S.O.F. de los buques	Mostrar_S.O.F.
Eliminar S.O.F.	Elim_S.O.F.

Tabla 2.15: Tarjetas CRC N° 3

#### Tarjeta CRC N° 4

<b>Nombre de la clase:</b> Permiso de Entrada	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Permisos de entrada	Mostrar_Permiso_Entrada

Tabla 2.16: Tarjetas CRC N° 4

#### Tarjeta CRC N° 5



<b>Nombre de la clase:</b> Permiso de Atraque	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Permisos de atraque	Mostrar_Permiso_atraque

Tabla 2.17: Tarjetas CRC Nº 5

Tarjeta CRC Nº 6

<b>Nombre de la clase:</b> Permiso de Salida	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Permisos de Salida	Mostrar_Permiso_Salida

Tabla 2.18: Tarjetas CRC Nº 6

Tarjeta CRC Nº 7

<b>Nombre de la clase:</b> Permiso de Remoción	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Permisos de Remoción	Mostrar_Permiso_Remocion

Tabla 2.19: Tarjetas CRC Nº 7

Tarjeta CRC Nº 8

<b>Nombre de la clase:</b> Permiso de Descarga	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Permisos de Descarga	Mostrar_Permiso_Descarga

Tabla 2.20: Tarjetas CRC Nº 8

Tarjeta CRC Nº 9

<b>Nombre de la clase:</b> Solicitud de remolcadores
--



Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Solicitud de remolcadores	Mostrar_Sol_Remolcadores

Tabla 2.21: Tarjetas CRC N°9

### Tarjeta CRC N° 10

<b>Nombre de la clase:</b> Solicitud de Prácticos	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Solicitud de Prácticos	Mostrar_Sol_Practicos

Tabla 2.22: Tarjetas CRC N° 10

### Tarjeta CRC N° 11

Nombre de la clase: Reporte de Arribo	
Tipo de la clase: Utilitaria	
Responsabilidades:	Colaboradores:
Mostrar Reporte de Arribo	Mostrar_Arribo

Tabla 2.23: Tarjetas CRC N° 11



### Anexo 4: Modelo de datos

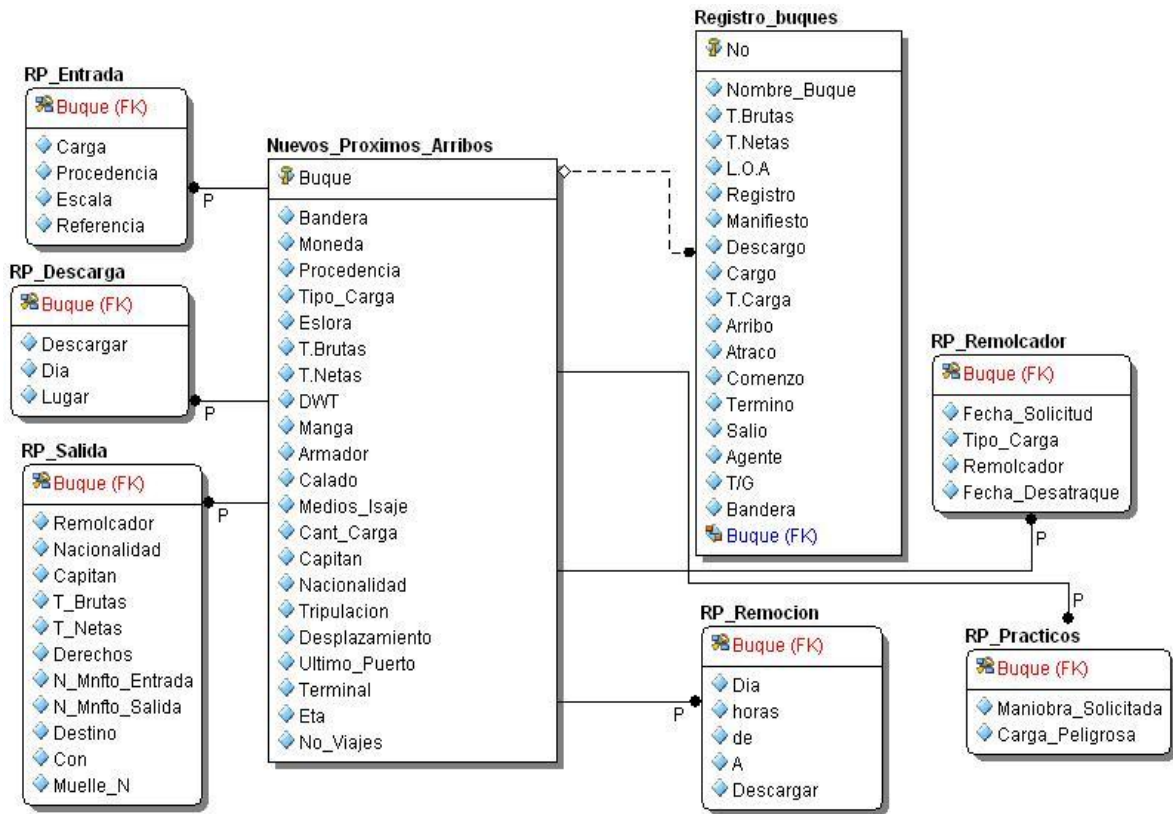


Fig. 3.1 Modelo de Datos



### Anexo 5 Tarjetas de Tareas

Tarjetas de tarea # 1: Introducir los datos de los nuevos próximos arribos

Tarea ingeniería	
Número tarea: 1	Número historia: 1
<b>Nombre de tarea:</b> Insertar los datos de los nuevos próximos arribos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 06/02/2012	Fecha de fin: 10/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita la insertar los datos de los buques	

Tabla 3.3 Tarjetas de tarea # 1

Tarjetas de tarea # 2: Modificar los datos de los nuevos próximos arribos

Tarea ingeniería	
Número tarea: 2	Número historia: 1
<b>Nombre de tarea:</b> Modificar los datos de los nuevos próximos arribos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 13/02/2012	Fecha de fin: 15/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita modificar los datos de los buques una vez que son insertados	

Tabla 3.4 Tarjetas de tarea # 2

Tarjetas de tarea # 3: Mostrar los datos de los nuevos próximos arribos

Tarea ingeniería	
Número tarea: 3	Número historia: 1
<b>Nombre de tarea:</b> Mostrar los datos de los nuevos próximos arribos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 15/02/2012	Fecha de fin: 17/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques una vez que son insertados	

Tabla 3.5 Tarjetas de tarea # 3



Tarjetas de tarea # 4: Eliminar los datos de los nuevos próximos arribos

Tarea ingeniería	
Número tarea: 4	Número historia: 1
<b>Nombre de tarea:</b> Eliminar los datos de los nuevos próximos arribos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 20/02/2012	Fecha de fin: 22/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques después que estos parten del puerto	

Tabla 3.6 Tarjetas de tarea # 4

Tarjetas de tarea # 5: Insertar los datos del registro de buques

Tarea ingeniería	
Número tarea: 5	Número historia: 2
<b>Nombre de tarea:</b> Insertar los datos del registro de buques	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 23/02/2012	Fecha de fin: 29/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques en el registro	

Tabla 3.7 Tarjetas de tarea # 5

Tarjetas de tarea # 6: Mostrar los datos del registro de buques

Tarea ingeniería	
Número tarea: 6	Número historia: 2
<b>Nombre de tarea:</b> Mostrar los datos del registro de buques	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 23/02/2012	Fecha de fin: 29/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques en el registro	

Tabla 3.8 Tarjetas de tarea # 6



Tarjetas de tarea # 7: Modificar los datos del registro de buques

Tarea ingeniería	
Número tarea: 7	Número historia: 2
<b>Nombre de tarea:</b> Modificar los datos del registro de buques	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 23/02/2012	Fecha de fin: 29/02/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita modificar los datos de los buques en el registro	

Tabla 3.9 Tarjetas de tarea # 7

Tarjetas de tarea # 8: Eliminar los datos del registro de buques

Tarea ingeniería	
Número tarea: 8	Número historia: 2
<b>Nombre de tarea:</b> Eliminar los datos del registro de buques	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29/02/2012	Fecha de fin: 02/03/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques en el registro	

Tabla 3.10 Tarjetas de tarea # 8

Tarjetas de tarea # 9: Insertar los datos del S.O.F.

Tarea ingeniería	
Número tarea: 9	Número historia: 3
<b>Nombre de tarea:</b> Insertar los datos del S.O.F.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 05/03/2012	Fecha de fin: 09/03/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de las operaciones del S.O.F.	

Tabla 3.12 Tarjetas de tarea # 9





Tarjetas de tarea # 10: Mostrar los datos del S.O.F.

Tarea ingeniería	
Número tarea: 10	Número historia: 3
<b>Nombre de tarea:</b> Mostrar los datos del S.O.F.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 12/03/2012	Fecha de fin: 14/03/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de las operaciones del S.O.F.	

Tabla 3.13 Tarjetas de tarea # 10

Tarjetas de tarea # 11: Eliminar los datos del S.O.F.

Tarea ingeniería	
Número tarea: 11	Número historia: 3
<b>Nombre de tarea:</b> Eliminar los datos del S.O.F.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 14/03/2012	Fecha de fin: 16/03/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de las operaciones del S.O.F.	

Tabla 3.14 Tarjetas de tarea # 11

Tarjetas de tarea # 12: Insertar los datos del reporte de arribo

Tarea ingeniería	
Número tarea: 12	Número historia: 4
<b>Nombre de tarea:</b> Insertar los datos del reporte de arribo	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 19/03/2012	Fecha de fin: 23/03/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques una vez	



llegado al puerto
-------------------

Tabla 3.15 Tarjetas de tarea # 12

Tarjetas de tarea # 13: Mostrar los datos del reporte de arribo.

Tarea ingeniería	
Número tarea: 13	Número historia: 4
<b>Nombre de tarea:</b> Mostrar los datos del reporte de arribo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 26/03/2012	Fecha de fin: 28/03/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques una vez llegado al puerto	

Tabla 3.16 Tarjetas de tarea # 13

Tarjetas de tarea # 14: Eliminar los datos del reporte de arribo.

Tarea ingeniería	
Número tarea: 14	Número historia: 4
<b>Nombre de tarea:</b> Eliminar los datos del reporte de arribo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29/03/2012	Fecha de fin: 04/04/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques una vez llegado al puerto	

Tabla 3.17 Tarjetas de tarea # 14

Tarjetas de tarea # 15: Insertar los datos del reporte de entrada.

Tarea ingeniería	
Número tarea: 15	Número historia: 5
<b>Nombre de tarea:</b> Insertar los datos del reporte de entrada.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 05/04/2012	Fecha de fin: 12/04/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques una vez	



llegado al puerto

Tabla 3.19 Tarjetas de tarea # 15

Tarjetas de tarea # 16: Mostrar los datos del reporte de entrada.

Tarea ingeniería	
Número tarea: 16	Número historia: 5
<b>Nombre de tarea:</b> Mostrar los datos del reporte de entrada.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 13/04/2012	Fecha de fin: 17/04/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques una vez llegado al puerto	

Tabla 3.20 Tarjetas de tarea # 16

Tarjetas de tarea # 17: Eliminar los datos del reporte de entrada.

Tarea ingeniería	
Número tarea: 17	Número historia: 5
<b>Nombre de tarea:</b> Eliminar los datos del reporte de entrada.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 18/04/2012	Fecha de fin: 20/04/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques una vez que esto parten del puerto	

Tabla 3.21 Tarjetas de tarea # 17

Tarjetas de tarea # 18: Insertar los datos del reporte de Salida.

Tarea ingeniería	
Número tarea: 18	Número historia: 6
<b>Nombre de tarea:</b> Insertar los datos del reporte de Salida.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 23/04/2012	Fecha de fin: 25/04/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques en el reporte.	

Tabla 3.22 Tarjetas de tarea # 18



Tarjetas de tarea # 19: Mostrar los datos del reporte de Salida.

Tarea ingeniería	
Número tarea: 19	Número historia: 6
<b>Nombre de tarea:</b> Mostrar los datos del reporte de salida.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 25/04/2012	Fecha de fin: 04/05/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques en el reporte.	

Tabla 3.23 Tarjetas de tarea # 19

Tarjetas de tarea # 20: Eliminar los datos del reporte de Salida.

Tarea ingeniería	
Número tarea: 20	Número historia: 6
<b>Nombre de tarea:</b> Eliminar los datos del reporte de salida.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 07/05/2012	Fecha de fin: 09/05/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques en el reporte.	

Tabla 3.24 Tarjetas de tarea # 20

Tarjetas de tarea # 21: Insertar los datos del reporte de descarga.

Tarea ingeniería	
Número tarea: 21	Número historia: 7
<b>Nombre de tarea:</b> Insertar los datos del reporte de descarga.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 10/05/2012	Fecha de fin: 15/05/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques en el reporte.	

Tabla 3.25 Tarjetas de tarea # 21



Tarjetas de tarea # 22: Mostrar los datos del reporte de descarga.

Tarea ingeniería	
Número tarea: 22	Número historia: 7
<b>Nombre de tarea:</b> Mostrar los datos del reporte de descarga.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 16/05/2012	Fecha de fin: 18/05/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques en el reporte.	

Tabla 3.26 Tarjetas de tarea # 22

Tarjetas de tarea # 23: Eliminar los datos del reporte de descarga.

Tarea ingeniería	
Número tarea: 23	Número historia: 7
<b>Nombre de tarea:</b> Eliminar los datos del reporte de descarga.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 21/05/2012	Fecha de fin: 23/05/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques en el reporte.	

Tabla 3.27 Tarjetas de tarea # 23

Tarjetas de tarea # 24: Insertar los datos del reporte de remoción.

Tarea ingeniería	
Número tarea: 24	Número historia: 8
<b>Nombre de tarea:</b> Insertar los datos del reporte de remoción.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 24/05/2012	Fecha de fin: 30/05/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques en el reporte.	

Tabla 3.28 Tarjetas de tarea # 24



Tarjetas de tarea # 25: Mostrar los datos del reporte de remoción.

Tarea ingeniería	
Número tarea: 25	Número historia: 8
<b>Nombre de tarea:</b> Mostrar los datos del reporte de remoción.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 31/05/2012	Fecha de fin: 04/06/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques en el reporte.	

Tabla 3.29 Tarjetas de tarea # 25

Tarjetas de tarea # 26: Eliminar los datos del reporte de remoción.

Tarea ingeniería	
Número tarea: 26	Número historia: 8
<b>Nombre de tarea:</b> Eliminar los datos del reporte de remoción.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 05/06/2012	Fecha de fin: 07/06/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques en el reporte.	

Tabla 3.30 Tarjetas de tarea # 26

Tarjetas de tarea # 27: Insertar los datos de la solicitud de remolcadores y prácticos.

Tarea ingeniería	
Número tarea: 27	Número historia: 9
<b>Nombre de tarea:</b> Insertar los datos de la solicitud de remolcadores y prácticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 08/06/2012	Fecha de fin: 14/06/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita insertar los datos de los buques en los reportes.	



Tabla 3.31 Tarjetas de tarea # 27

Tarjetas de tarea # 28: Mostrar los datos de la solicitud de remolcadores y prácticos.

Tarea ingeniería	
Número tarea: 28	Número historia: 9
<b>Nombre de tarea:</b> Mostrar los datos de la solicitud de remolcadores y prácticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 15/06/2012	Fecha de fin: 19/06/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita mostrar los datos de los buques en los reportes.	

Tabla 3.32 Tarjetas de tarea # 28

Tarjetas de tarea # 29: Eliminar los datos de la solicitud de remolcadores y prácticos.

Tarea ingeniería	
Número tarea: 29	Número historia: 9
<b>Nombre de tarea:</b> Eliminar los datos de la solicitud de remolcadores y prácticos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 20/06/2012	Fecha de fin: 22/06/2012
Programador Responsable: Alex Rosabal Sánchez	
<b>Descripción:</b> Esta tarea facilita eliminar los datos de los buques en los reportes.	

Tabla 3.33 Tarjetas de tarea # 29



## Anexo 6 Prueba de Aceptación

Prueba de aceptación
HU: Gestión del registro de buques
Nombre: Prueba para comprobar la entrada de datos.
Descripción: Validación de la entrada de datos del registro de buque
Condiciones de ejecución: El usuario debe introducir los datos de los buques.
Entrada/Pasos ejecución: El usuario intentara introducir los datos de los buques
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li><li>• Se intente insertar un registro que ya existe.</li><li>• Se intente introducir un formato no válido.</li></ul> Se emite un mensaje en caso de información de éxito al insertar el buque.
Evaluación de la prueba: Aceptada.

Tabla 3.35 PA: Prueba para comprobar la entrada de datos del registro de buques

Prueba de aceptación
HU: Gestión de los próximos arribos.
Nombre: Prueba para comprobar la visualización de los próximos arribos.
Descripción: Selección correcta de las fechas de los próximos arribos
Condiciones de ejecución: El usuario debe seleccionar la opción listar próximo arribo
Entrada/Pasos ejecución: <ul style="list-style-type: none"><li>➤ El usuario selecciona la opción listar próximo arribo.</li><li>➤ Seleccionará la fecha de antelación.</li></ul>





<p>Resultado:</p> <ul style="list-style-type: none"> <li>➤ Se emite un mensaje de error en caso de que la fecha sea se haya pasado de los parámetros.</li> <li>➤ O son visualizados correctamente los contratos</li> </ul> <p>Se emite un mensaje en caso de información de éxito al mostrar el próximo arribo.</p>
Evaluación de la prueba: Aceptada.

Tabla 3.36 PA: Prueba para comprobar la visualización de los próximos arribos

Prueba de aceptación
HU: Gestión del registro de buques
Nombre: Prueba para comprobar la visualización de los datos.
Descripción: Validación de la entrada del nombre del buque
Condiciones de ejecución: El usuario debe introducir el nombre del buque.
Entrada/Pasos ejecución: El usuario intentara introducir los datos de los buques
<p>Resultado:</p> <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none"> <li>• No se introduzcan el dato solicitado.</li> <li>• Se intente mostrar un registro que no existe.</li> <li>• Se intente introducir un formato no válido.</li> </ul> <p>Se emite un mensaje en caso de información de éxito al mostrar el buque.</p>
Evaluación de la prueba: Aceptada.

Tabla 3.37 PA: Prueba para comprobar la visualización del registro de buques

Prueba de aceptación
HU: Gestión del reporte de entrada
Nombre: Prueba para comprobar la entrada de datos.



Descripción: Validación de la entrada de datos del reporte de entrada
Condiciones de ejecución: El usuario debe introducir los datos de los buques.
Entrada/Pasos ejecución: El usuario intentara introducir los datos del buque.
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li><li>• Se intente insertar un registro que ya existe.</li><li>• Se intente introducir un formato no válido.</li></ul> Se emite un mensaje en caso de información de éxito al insertar el buque.
Evaluación de la prueba: Aceptada.

Tabla 3.38 PA: Prueba para comprobar la entrada de datos del reporte de entrada

Prueba de aceptación
HU: Gestión del S.O.F.
Nombre: Prueba para comprobar la entrada de datos.
Descripción: Validación de la entrada de datos del S.O.F.
Condiciones de ejecución: El usuario debe introducir los datos del buque.
Entrada/Pasos ejecución: El usuario intentara introducir los datos del buque.
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li><li>• Se intente insertar un registro que ya existe.</li><li>• Se intente introducir un formato no válido.</li></ul> Se emite un mensaje en caso de información de éxito al insertar el reporte.
Evaluación de la prueba: Aceptada.

Tabla 3.39 PA: Prueba para comprobar la entrada de datos del S.O.F.



Prueba de aceptación
HU: Gestión del reporte de arribo
Nombre: Prueba para comprobar la entrada de datos.
Descripción: Validación de la entrada de datos del reporte de arribo
Condiciones de ejecución: El usuario debe introducir los datos de los buques.
Entrada/Pasos ejecución: El usuario intentara introducir los datos de los buques
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li><li>• Se intente insertar un registro que ya existe.</li><li>• Se intente introducir un formato no válido.</li></ul> Se emite un mensaje en caso de información de éxito al insertar el reporte.
Evaluación de la prueba: Aceptada.

Tabla 3.40 PA: Prueba para comprobar la entrada de datos del reporte de arribo

Prueba de aceptación
HU: Gestión del reporte de descarga
Nombre: Prueba para comprobar la entrada de datos.
Descripción: Validación de la entrada de datos del reporte de descarga
Condiciones de ejecución: El usuario debe introducir los datos del buque.
Entrada/Pasos ejecución: El usuario intentara introducir los datos de los buques
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li></ul>



<ul style="list-style-type: none"><li>• Se intente insertar un registro que ya existe.</li><li>• Se intente introducir un formato no válido.</li></ul> <p>Se emite un mensaje en caso de información de éxito al insertar el reporte.</p>
Evaluación de la prueba: Aceptada.

Tabla 3.41 PA: Prueba para comprobar la entrada de datos del reporte de descarga

Prueba de aceptación
HU: Gestión de las solicitudes de remolcadores y prácticos
Nombre: Prueba para comprobar la entrada de datos.
Descripción: Validación de la entrada de datos de las solicitudes de remolcadores y prácticos
Condiciones de ejecución: El usuario debe introducir los datos del buque.
Entrada/Pasos ejecución: El usuario intentara introducir los datos del buque
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"><li>• No se introduzcan los datos solicitados.</li><li>• Se intente insertar un registro que ya existe.</li><li>• Se intente introducir un formato no válido.</li></ul> Se emite un mensaje en caso de información de éxito al insertar la solicitud.
Evaluación de la prueba: Aceptada.

Tabla 3.42 PA: Prueba para comprobar la entrada de datos de las solicitudes de remolcadores y prácticos.