

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA

UNAN – LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA



“SISTEMA DE CONTROL AUTOMATIZADO DEL PROCESAMIENTO DEL CAMARON PARA LA EMPRESA SAHLMAN SEAFOODS, S.A”

Monografía para optar al Título de Ingeniero en Sistemas de Información

Autores:

- **Br. Manuel Eduardo González Rodríguez**
- **Br. Mario Augusto Hernández Sandoval**
- **Br. Byron Xavier Cruz Aguilera**

Tutor:

Msc. Álvaro Altamirano

León, Febrero del 2011



INDICE

INTRODUCCION	3
ANTECEDENTES	4
JUSTIFICACIÓN	5
OBJETIVOS	6
MARCO TEORICO	7
BASES DE DATOS	7
ARQUITECTURA DE BASE DE DATOS	8
MODELO ENTIDAD-RELACIÓN	8
MODELOS DE BASES DE DATOS	10
MD5 (MESSAGE-DIGEST ALGORITHM 5, ALGORITMO DE RESUMEN DEL MENSAJE 5)	11
MICROSOFT SQL SERVER	12
MICROSOFT SQL SERVER 2008 EXPRESS	12
PLATAFORMA DE .NET.....	13
FRAMEWORK	13
FRAMEWORK .NET V3.5.....	13
COMMON LANGUAGE RUNTIME (CLR)	14
BIBLIOTECA DE CLASES DE .NET	15
ESPACIOS DE NOMBRES.....	16
ARCHIVOS DLL ADMINISTRADOS (ENSAMBLADOS)	16
COMMON LANGUAGE SPECIFICATION (CLS)	17
MICROSOFT VISUAL STUDIO	18
MICROSOFT VISUAL STUDIO 2008 PROFESSIONAL	18
LENGUAJE C# (C SHARP)	18
CONTROLES RIBBON	19
DISEÑO METODOLOGICO	21
RECURSOS Y NECESIDADES	24
CRONOGRAMA	25
ESPECIFICACION DE REQUISITOS DE SOFTWARE	26
DIAGRAMA DE CASOS DE USO	75
DIAGRAMA DE CLASES	78
DIAGRAMA DE FLUJO DE DATOS	79
DIAGRAMA ENTIDAD-RELACION	83
DISEÑO DE DATOS	84
DISEÑO PROCEDIMENTAL	93
DISEÑO ARQUITECTONICO	94
INTERFACES	95
CODIFICACION	104
CONCLUSIONES	106
RECOMENDACIONES	107
BIBLIOGRAFIA	108
ANEXOS	109



INTRODUCCION

La Empresa Sahlman Seafoods, Nicaragua, se encuentra ubicada en El Viejo, municipio del departamento de Chinandega, a 138 Km de Managua y se dedica a la producción, procesamiento y comercialización de camarón proveniente de sus granjas, así como el servicio de maquila a productores de la zona.

Sahlman Seafoods Nicaragua está comprometida al uso óptimo de sus recursos a través de inversión en mejoras de equipo, charlas al personal, estricta supervisión y seguimiento de sistemas establecidos para asegurar resultados satisfactorios y reducción de desperdicios.

A medida que la Empresa ha ido creciendo y generando año con año calidad y seguridad en sus procesos de trabajo y productos, le surge la necesidad de automatizar las áreas de proceso y clasificación del camarón ya que esto ocasiona gastos en concepto de papelería y complejidad de trabajo.

Por lo tanto, hemos decidido trabajar en un lenguaje de programación orientado a objetos CSharp (C#), desarrollado en la plataforma punto NET de Visual Studio 2008, para obtener una mejor seguridad y distribución de los datos, al mismo tiempo se utilizó como gestor de bases de datos: SQL Server 2008, para obtener de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para la empresa.

Este trabajo de investigación estará organizado de manera secuencial, es decir, a medida que se vaya obteniendo información detallada por la empresa, se irá avanzando continuamente hasta lograr completar nuestra Tesis de Investigación y cumplir con todos los objetivos y requerimientos propuestos para la elaboración del sistema y la automatización de dicha empresa.



ANTECEDENTES

La Empresa Sahlman Seafoods Nicaragua en todos sus años de experiencia, se ha ido automatizando por medio de sus aplicaciones, que abarcan desde la Contabilidad hasta la Nómina de Empleados de la misma recientemente. Sin Embargo no posee ninguna aplicación que lleve un control sobre el proceso del producto de camarón, que se lleva a cabo desde la recepción hasta su debido empaque y el respectivo envío a sus clientes.

Por el Crecimiento y Desarrollo que la empresa ha tenido en los últimos años, se ha visto en la necesidad de crear un sistema que le permita agilizar y facilitar su trabajo, proporcionándoles a los que intervienen en el proceso del producto, llevar un control eficaz y seguro, de manera que la información manejada se encuentre accesible y ordenada. De esta manera el trabajo realizado por los usuarios, se reduzca considerablemente y al mismo tiempo obtener mejores resultados tanto al cliente como la toma de decisiones por parte de la empresa.

En el año 1999, en la Universidad Nacional Autónoma de Nicaragua – León, dentro del departamento de computación se realizó una aplicación de escritorio similar para brindar un Control Automatizado del Procesamiento del Camarón para la empresa ECUANICA, S.A.



JUSTIFICACIÓN

El propósito por el cual se ha realizado esta aplicación a la empresa Sahlman Seafoods, es para llevar un control automatizado y ágil en todos los procesos internos de la planta, y de esa manera obtener una mejor fiabilidad, precisión, seguridad y rapidez al capturar la información.



OBJETIVOS

GENERAL

Desarrollar un sistema de escritorio que permita a Sahlman Seafoods Nicaragua facilitar los procesos de Recepción, Peso en Planta, Control de Calidad, Liquidación y Masterización del producto terminado a sus clientes.

ESPECIFICOS

- ❖ Diseñar Interfaces que permitan el rápido y fácil ingreso de los datos.
- ❖ Desarrollar Niveles de seguridad de acceso a los usuarios, que permitan al administrador decidir quiénes pueden ingresar a los procesos presentes en el sistema.
- ❖ Utilizar Procedimientos Almacenados y Funciones por medio de SQL Server y Transact-SQL para obtener un rápido acceso a la base de datos.
- ❖ Crear Filtros de Información para generar reportes, que permitan el fácil acceso al historial de los datos.
- ❖ Respalidar copias de seguridad de la base de datos.
- ❖ Diseñar una Interfaz que permita configurar la cadena de conexión de acceso a la base de datos.



MARCO TEORICO

En la realización del sistema para la empresa SAHLMAN SEAFOODS, S.A se empleará una serie de términos y herramientas las cuales hemos decido conceptualizar para un mayor entendimiento de nuestro trabajo investigativo.

Bases de Datos

Uno de los objetivos fundamentales de un sistema de información es contar no sólo con recursos de información, sino también con los mecanismos necesarios para poder encontrar y recuperar estos recursos. De esta forma, las bases de datos se han convertido en un elemento indispensable no sólo para el funcionamiento de los grandes motores de búsqueda y la recuperación de información a lo largo y ancho de la Web, sino también para la creación de sedes web, Intranets y otros sistemas de información en los que se precisa manejar grandes o pequeños volúmenes de información. La creación de una base de datos a la que puedan acudir los usuarios para hacer consultas y acceder a la información que les interese es, pues, una herramienta imprescindible de cualquier sistema informativo sea en red o fuera de ella.

Una base de datos es una colección de datos organizados y estructurados según un determinado modelo de información que refleja no sólo los datos en sí mismos, sino también las relaciones que existen entre ellos. Una base de datos se diseña con un propósito específico y debe ser organizada con una lógica coherente. Los datos podrán ser compartidos por distintos usuarios y aplicaciones, pero deben conservar su integridad y seguridad al margen de las interacciones de ambos. La definición y descripción de los datos han de ser únicas para minimizar la redundancia y maximizar la independencia en su utilización.

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Elementos relacionados a una Base de Datos: Al trabajar con los datos de una BD, se deben mantener elementos que apoyan el acceso adecuado a las estructuras de datos. Se distinguen:

Planillas de Datos: Tablas que contienen filas, cada una representa un objeto de un tipo común para las filas de la tabla, que a su vez se compone de varios “atributos” o propiedades que se ordenan en las columnas de la planilla.

Registro: Una fila de una tabla. Representa un objeto de un tipo definido y se compone de atributos.

Campos: Son precisamente la definición de estos “atributos” de la tabla o planilla.

Llave primaria: Es una parte de los campos de un registro que permite distinguirlo en forma única del resto de registros adicionales.

Índices: Al igual que el índice de un libro, un índice de una BD, que se asocia a una sola tabla, referencia la posición de un registro dentro de una tabla, de acuerdo a un orden predefinido.



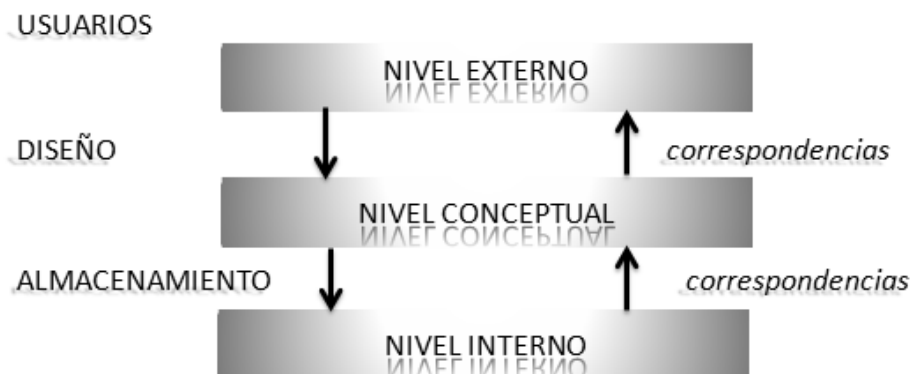
Vistas: Son “ángulos” de la BD. Al igual que una persona puede ver un auto desde el exterior, lo cual constituye la vista “externa” del vehículo, existe otra vista interior, y por su parte el objeto “vehículo” posee muchas propiedades o atributos que no son visibles en algunas vistas.

Arquitectura de Base de Datos

Nivel Interno: Es el nivel más cercano al almacenamiento físico de los datos. Permite escribirlos tal y como están almacenados en el ordenador. En este nivel se diseñan los archivos que contienen la información, la ubicación de los mismos y su organización, es decir se crean los archivos de configuración.

Nivel Conceptual: En este nivel se representan los datos que se van a utilizar sin tener en cuenta aspectos como lo que representamos en el nivel interno.

Nivel Externo: Es el más cercano al usuario. En este nivel se describen los datos o parte de los datos que más interesan a los usuarios.



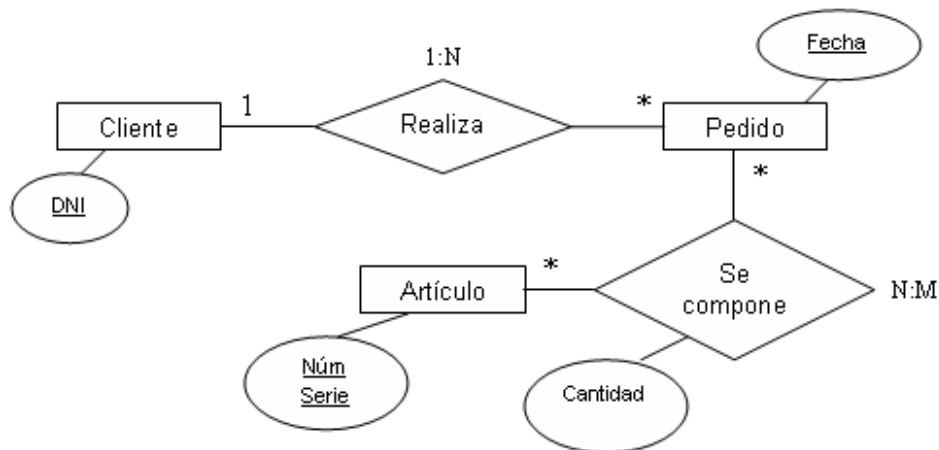
Este modelo de arquitectura permite establecer el principio de independencia de los datos, ya se trate de una independencia lógica o física. La independencia lógica significa que los cambios en el esquema lógico no deben afectar a los esquemas externos que no utilicen los datos modificados; la independencia física significa que el esquema lógico no se va a ver afectado por los cambios realizados en el esquema interno, correspondientes a modos de acceso, etc.

A la hora de diseñar una base de datos hay que distinguir por un lado el modelo de datos (instrumento) y por otro lado el esquema de datos (el resultado de aplicar ese modelo). Un modelo de datos es un conjunto de conceptos, reglas y convenciones que nos permiten describir los datos del universo del discurso. Un esquema es la estructura de datos obtenida tras aplicar dicho modelo.

Modelo Entidad-Relación

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

Ejemplo de un diagrama entidad-relación:



El **Modelo Entidad-Relación**, también conocido como DER (diagramas entidad-relación) es una herramienta de modelado para bases de datos, propuesto por Peter Chen en 1976, mediante el cual se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como entidades (se corresponde al concepto de clase, cada tupla representaría un objeto, de la Programación Orientada a Objetos) las cuales tienen unos atributos y se vinculan mediante *relaciones*.

Es una representación conceptual de la información. Mediante una serie de procedimientos se puede pasar del modelo E-R a otros, como por ejemplo el modelo relacional.

El modelado entidad-relación es una *técnica* para el modelado de datos utilizando diagramas entidad relación. No es la única técnica pero sí la más utilizada.

Brevemente consiste en los siguientes pasos:

1. Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
2. Se hace una lista de los sustantivos y verbos que aparecen.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
6. Se elabora el diagrama (o diagramas) entidad-relación.
7. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

Dado lo rudimentario de esta técnica se necesita cierto entrenamiento y experiencia para lograr buenos modelos de datos.

El modelado de datos no acaba con el uso de esta técnica. Son necesarias otras técnicas para lograr un modelo directamente implementable en una base de datos.

Brevemente:

- Transformación de relaciones múltiples en binarias.
- Normalización de una base de datos de relaciones (algunas relaciones pueden transformarse en atributos y viceversa).
- Conversión en tablas (en caso de utilizar una base de datos relacional).



En nuestro proyecto hemos llegado a la necesidad de usar el modelo de base de datos relacional ya que es el que más se acopla a nuestras necesidades, no obstante, decidimos mencionar los diferentes modelos de base de datos para dar una mayor conocimiento en el concepto de las BBDD.

Modelos de bases de datos

Un **modelo de base de datos** o esquema de base de datos es la estructura o el formato de una base de datos, descrita en un lenguaje formal soportada por el sistema de gestión de bases de datos. En otras palabras, un "modelo de base de datos" es la aplicación de un modelo de datos usado en conjunción con un sistema de gestión de bases de datos.

Los esquemas generalmente son almacenados en un diccionario de datos. Aunque un esquema se defina en un lenguaje de base de datos de texto, el término a menudo es usado para referirse a una representación gráfica de la estructura de la base de datos.

Base de Datos Jerárquica: Puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos. Este modelo tiene forma de árbol invertido en el que una rama puede tener varios hijos, pero cada hijo sólo puede tener un padre.

Base de Datos De red: Este modelo permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la base de datos puede tener varias ocurrencias superiores a él. El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector. En el modelo en red se representa el mundo real mediante registros lógicos que representan a una entidad y que se relacionan entre sí por medio de flechas

Base de Datos Transaccionales: Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

Bases de datos relacionales: Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.



Sistema de Control Automatizado del Procesamiento del Camarón para la Empresa Sahlman Seafoods, S.A.

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos, el cual es entendido como el proceso necesario para que una base de datos sea utilizada de manera óptima.

Entre las ventajas de este modelo están:

- Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.
- Garantiza la integridad referencial: Así al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

Para llevar a cabo nuestro proceso de creación y manipulación de una base de datos hemos decidido utilizar el gestor “MICROSOFT SQL SERVER 2008”, ya que nos proporciona las herramientas necesarias para cumplir con las expectativas que nos hemos planteado para con nuestro trabajo de investigación.

Debido a que el Sistema cuenta con un sistema de usuario, es de gran importancia proteger las contraseñas, con el fin de prevenir posibles vulnerabilidades en el servidor por parte de terceros, de esta manera si alguien no autorizado accede a dicho servidor no podrá ver las contraseñas sino su encriptación.

Para esto hay un sin número de algoritmos de encriptación, como lo es MD5, a continuación se da una breve explicación.

MD5 (Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5)

Es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

Historia

Es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (*Massachusetts Institute of Technology*, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad.

A pesar de su amplia difusión actual, la sucesión de problemas de seguridad detectados desde que, en 1996, Hans Dobbertin anunciase una colisión de *hash*, plantea una serie de dudas acerca de su uso futuro.

Codificación

La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal. El siguiente código de 28 bytes ASCII será tratado con MD5 y veremos su correspondiente *hash* de salida:

```
MD5("Esto sí es una prueba de MD5") = e99008846853ff3b725c27315e469fbc
```



Un simple cambio en el mensaje nos da un cambio total en la codificación *hash*, en este caso cambiamos dos letras, el «sí» por un «no».

```
MD5("Esto no es una prueba de MD5") = dd21d99a468f3bb52a136ef5beef5034
```

Aplicaciones

Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software.

Al mismo tiempo el algoritmo MD5 se utiliza para calcular el *hash* de las claves de los usuarios. En el disco se guarda el resultado del MD5 de la clave que se introduce al dar de alta un usuario, y cuando éste quiere entrar en el sistema se compara el *hash* MD5 de la clave introducida con el *hash* que hay guardado en el disco duro. Si coinciden, es la misma clave y el usuario será autenticado.

El MD5 también se puede usar para comprobar que los correos electrónicos no han sido alterados usando claves públicas y privadas.

Microsoft Sql Server

Microsoft SQL es un modelo relacional de base de datos producido por Microsoft. Sus lenguajes de consultas primarios son T-SQL y ANSI SQL.

Microsoft Sql Server 2008 Express

La versión actual de SQL Server, SQL Server 2008, tiene como objetivo hacer que los datos de ajuste de autogestión, la auto organización y el mantenimiento de la libre con el desarrollo de SQL Server siempre sobre tecnologías, para ofrecer casi cero tiempo de inactividad.

Microsoft SQL Server 2008 también incluyen soporte para datos estructurados y semi-estructurados, incluyendo formatos de medios digitales para fotos, audio, vídeo y otros datos multimedia. En las versiones actuales, tales datos multimedia se pueden almacenar como BLOB (grandes objetos binarios), pero son bitstreams genéricos.

Microsoft, SQL Server 2008 puede ser un almacenamiento de datos backend para las diferentes variedades de datos: XML, correo electrónico, la hora, calendario, archivo, documento, espacial, etc. así como realizar de búsqueda, consulta, análisis, intercambio y sincronización de todos los tipos de datos.

Microsoft SQL Server 2008 Express es un sistema de administración de datos eficaz y confiable que ofrece un variado conjunto de características, protección de datos y rendimiento para clientes de aplicaciones incrustadas, aplicaciones web ligeras y



Sistema de Control Automatizado del Procesamiento del Camarón para la Empresa Sahlman Seafoods, S.A.

almacén de datos locales. SQL Server 2008 Express, que está diseñado para una implementación sencilla y una creación de prototipos rápida, está disponible de forma gratuita y su redistribución con aplicaciones también es gratuita. Está diseñado para integrarse a la perfección con otras inversiones de infraestructura de servidor.

Plataforma de .NET

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

Su principal Objetivo es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

La plataforma .Net pretende reemplazar lo que es la API de Win32 y de Sistemas Operativos Windows.

Framework

Un framework, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. El Framework organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquico de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Framework .NET V3.5

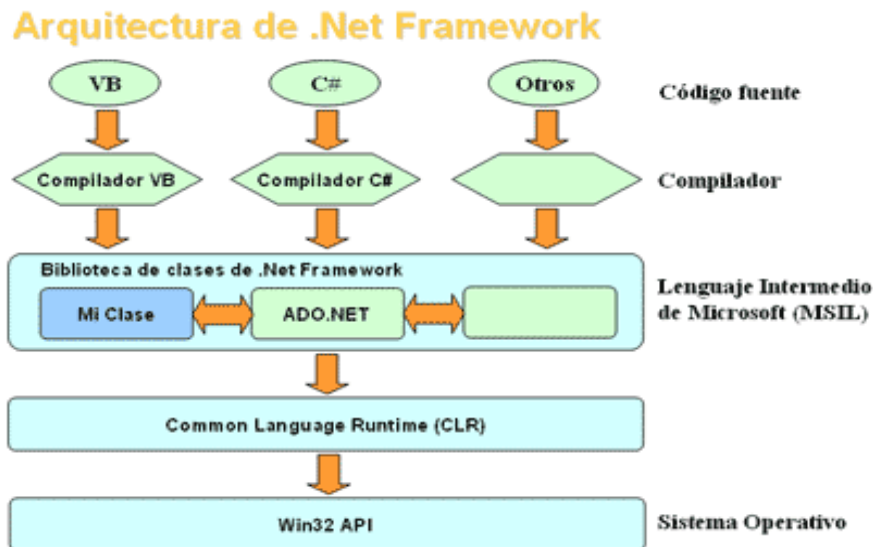
El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.

Actualmente, el Framework de .Net es una plataforma no incluida en los diferentes sistemas operativos distribuidos por Microsoft, por lo que es necesaria su instalación previa a la ejecución de programas creados mediante .Net. El Framework se puede descargar gratuitamente desde la web oficial de Microsoft.

.Net Framework soporta múltiples lenguajes de programación y aunque cada lenguaje



tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C# (C Sharp), Visual Basic o C++ hasta otros lenguajes menos conocidos como Perl o Cobol.



El Framework .Net V3.5 está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista".

El Framework consiste en el Common Language Runtime (CLR) y en las bibliotecas de clases de Framework .NET(BCL)

Common Language Runtime (CLR)

El CLR es el verdadero núcleo del Framework de .Net, ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios que ofrece el sistema operativo estándar Win32.

Es un entorno de ejecución en las que aplicaciones escritas en diferentes lenguajes de programación pueden ejecutarse al mismo tiempo sin conflicto alguno, conocido también como *interoperabilidad multilinguaje*.

El Common Language Runtime o CLR (Lenguaje común en tiempo de ejecución) es el componente de máquina virtual de la plataforma .Net de Microsoft. Es la implementación del estándar Common Language Infrastructure (CLI) que define un ambiente de ejecución para los códigos de los programas. El CLR ejecuta una forma de código intermedio (bytecode) llamada Common Intermediate Language (CIL, anteriormente conocido como MSIL -- Microsoft Intermediate Language), la implementación de Microsoft del CLI.

Los desarrolladores que usan CLR escriben el código en un lenguaje como C# o VB.Net. En tiempo de compilación, un compilador.NET convierte el código a MSIL (Microsoft Intermediate Language). En tiempo de ejecución, el compilador en tiempo de ejecución (Just-in-time compiler) del CLR convierte el código MSIL en código nativo para el sistema operativo. Alternativamente, el código MSIL es compilado a código nativo en un proceso separado anterior a la ejecución. Esto acelera las



posteriores ejecuciones del software debido a que la compilación de MSIL a nativo ya no es necesaria.

Common Language Runtime facilita el diseño de los componentes y de las aplicaciones cuyos objetos interactúan entre lenguajes distintos. Los objetos escritos en lenguajes diferentes pueden comunicarse entre sí, lo que permite integrar sus comportamientos de forma precisa.

A pesar de que algunas implementaciones del Common Language Infrastructure se ejecutan en sistemas operativos no Windows, el CLR se ejecuta solo en sistemas operativos de Microsoft Windows.

La manera en que la máquina virtual se relaciona con el CLR permite a los programadores ignorar muchos detalles específicos del CPU que estará ejecutando el programa. El CLR también permite otros servicios importantes, incluyendo los siguientes:

- Administración de la memoria
- Administración de hilos
- Manejo de excepciones
- Recolección de basura
- Seguridad

Biblioteca de Clases de .NET

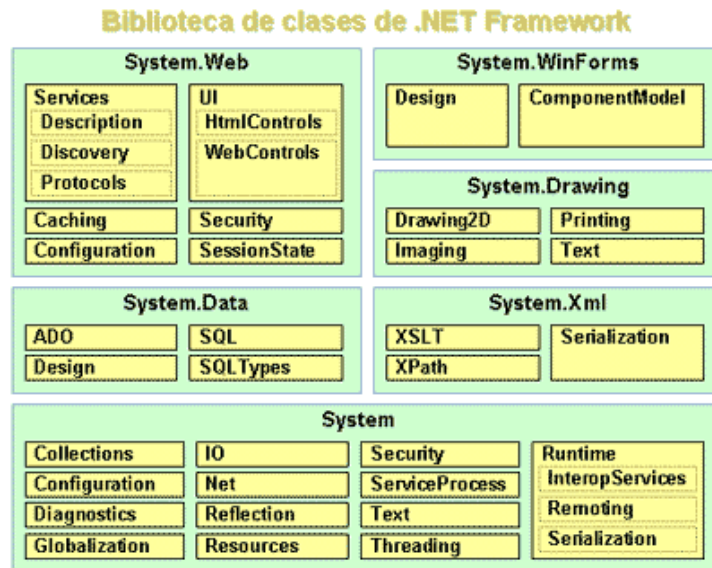
Las bibliotecas de clase de framework .NET son muy importantes para proporcionar interoperabilidad al lenguaje, porque permiten a los desarrolladores usar una sola interfaz de programación para toda la funcionalidad expuesta por el CLR.

Cuando se está programando una aplicación muchas veces se necesitan realizar acciones como manipulación de archivos, acceso a datos, conocer el estado del sistema, implementar seguridad, etc. El Framework organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquico de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita.

Para ello, el Framework posee un sistema de tipos universal, denominado Common Type System (CTS). Este sistema permite que el programador pueda interactuar los tipos que se incluyen en el propio Framework (biblioteca de clases de .Net) con los creados por él mismo (clases). De esta forma se aprovechan las ventajas propias de la programación orientada a objetos, como la herencia de clases predefinidas para crear nuevas clases, o el polimorfismo de clases para modificar o ampliar funcionalidades de clases ya existentes.

La biblioteca de clases de .NET Framework está constituida por espacios de nombres. Cada espacio de nombres contiene tipos que se pueden utilizar en el programa: clases, estructuras, enumeraciones, delegados e interfaces.

Cuando se crea un proyecto de Visual Basic o Visual C# en Visual Studio, se sigue haciendo referencia a las DLL más comunes de la clase base (ensamblados). No obstante, si necesita utilizar un tipo incluido en una DLL a la que aún no se hace referencia, deberá agregar la referencia de esa DLL.



Espacios de Nombres

La biblioteca de clases de .NET Framework está constituida por espacios de nombres. Cada espacio de nombres contiene tipos que se pueden utilizar en el programa: clases, estructuras, enumeraciones, delegados e interfaces.

Los espacios de nombres proporcionan ámbito: dos clases con el mismo nombre pueden utilizarse simultáneamente en el programa siempre que se encuentren en diferentes espacios de nombres y se califique cada nombre con el espacio de nombres correspondiente. El nombre del espacio de nombres forma parte del nombre completo del tipo (namespace.typename).

Todos los espacios de nombres suministrados por Microsoft empiezan por System o por Microsoft.

Archivos DLL Administrados (Ensamblados)

La funcionalidad de la biblioteca de clases de .NET Framework no reside en una única DLL. Al distribuir la funcionalidad de las clases base entre varias DLL, un programa administrado no necesita cargar, en el inicio, una gran DLL, sino una o varias DLL más reducidas. De esta forma, se reduce el tiempo de inicio de un programa.

Para poder utilizar espacios de nombres, antes hay que definirlos en el proyecto. Sin embargo, normalmente se utilizan tipos procedentes de espacios de nombres que residen en una DLL administrada. Una DLL administrada se conoce también como *ensamblado*.

Cuando se crea un proyecto de Visual Basic o Visual C# en Visual Studio, se sigue haciendo referencia a las DLL más comunes de la clase base (ensamblados). No obstante, si necesita utilizar un tipo incluido en una DLL a la que aún no se hace referencia, deberá agregar la referencia de esa DLL.



Common Language Specification (CLS)

Para poder interactuar completamente con otros objetos, sea cual sea el lenguaje en que se hayan implementado, los objetos deben exponer a los llamadores sólo aquellas características que sean comunes para todos los lenguajes con los que deben interoperar.

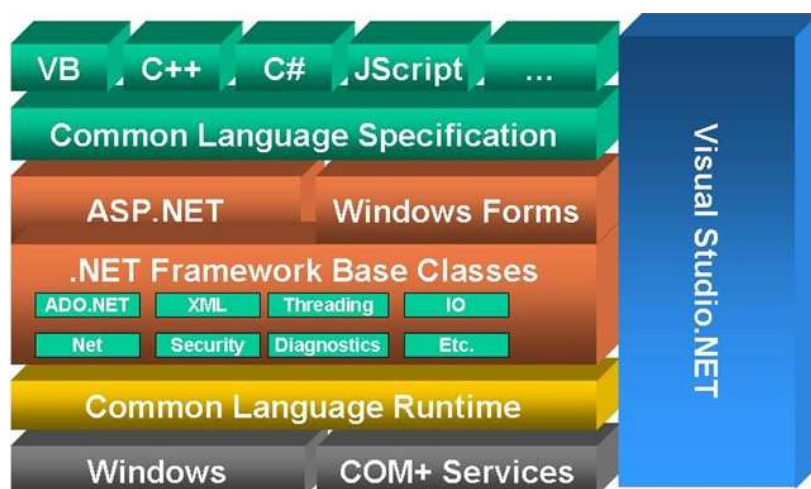
Por este motivo, se ha definido Common Language Specification (CLS), que es un conjunto de características de lenguaje básicas requeridas por la mayoría de las aplicaciones. Las reglas de CLS definen un subconjunto del Sistema de tipos comunes, es decir, todas las reglas que se aplican al sistema de tipos común se aplican también a CLS, salvo que se definan reglas más estrictas en CLS.

CLS ayuda a mejorar y garantizar la interoperabilidad entre lenguajes mediante la definición de un conjunto de características en las que se pueden basar los programadores y que están disponibles en una gran variedad de lenguajes. CLS también establece los requisitos de compatibilidad con CLS; estos requisitos permiten determinar si el código administrado cumple la especificación CLS y hasta qué punto una herramienta dada admite la programación de código administrado que utilice las características de CLS.

Si un componente sólo utiliza las características de CLS en la API que expone a otro código (incluidas las clases derivadas), se garantiza que se puede obtener acceso al componente desde cualquier lenguaje de programación que admita CLS. Los componentes que cumplen las reglas de CLS y usan sólo las características incluidas en CLS se conocen como componentes compatibles con CLS.

CLS se diseñó de manera que fuese lo suficientemente amplio como para incluir las construcciones de lenguaje que normalmente necesitan los programadores y lo suficientemente pequeño como para que todos los lenguajes pudieran admitirlo. Además, se ha excluido de CLS cualquier construcción de lenguaje de la que no se puede verificar rápidamente la seguridad de tipos del código, de manera que todos los lenguajes compatibles con CLS pueden generar código que es posible comprobar.

En la siguiente imagen se muestra como están organizados los diferentes conjuntos y características de Visual Studio .Net que permiten al programador ofrecer un desarrollo óptimo y eficaz.



Microsoft Visual Studio

Microsoft Visual Studio es el entorno de desarrollo hemos decidido utilizar para diseñar y desarrollar nuestra aplicación, a continuación se muestra un breve concepto de su definición:

Visual Studio es un completo conjunto de herramientas para la creación tanto de aplicaciones de escritorio como de aplicaciones web empresariales para trabajo en equipo. Aparte de generar aplicaciones de escritorio de alto rendimiento, se pueden utilizar las eficaces herramientas de desarrollo basado en componentes y otras tecnologías de Visual Studio para simplificar el diseño, desarrollo e implementación en equipo de soluciones empresariales.

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET.

Permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Microsoft Visual Studio 2008 Professional

Visual Studio 2008 Professional es un conjunto global de herramientas que hace realidad la visión del desarrollador de una forma muy rápida. Visual Studio 2008 Professional se diseñó para admitir proyectos de desarrollo pensados para Web (ASP.NET AJAX incluido), Windows Vista, Windows Server 2008, 2007 Microsoft Office System, SQL Server 2008 y dispositivos de Windows Mobile.

Lenguaje C# (C Sharp)

El lenguaje es muy sencillo, sigue el mismo patrón de los lenguajes de programación modernos. Incluye un amplio soporte de estructuras, componentes, programación orientada a objetos, manipulación de errores, recolección de basura, etc., que es construido sobre los principios de C++ y Java.



Las clases son la base de los lenguajes de programación orientados a objetos, lo cual permite extender el lenguaje a un mejor modelo para solucionar problemas. C# contiene las herramientas para definir nuevas clases, sus métodos y propiedades, al igual que la sencilla habilidad para implementar encapsulación, herencia y polimorfismo, que son los tres pilares de la programación orientada a objetos.

C# tiene un nuevo estilo de documentación XML que se incorpora a lo largo de la aplicación, lo que simplifica la documentación en línea de clases y métodos. C# soporta también interfaces, una forma de estipular los servicios requeridos de una clase. Las clases en C# pueden heredar de un padre pero puede implementar varias interfaces. C# también provee soporte para estructuras, un concepto el cual ha cambiado significativamente desde C++.

C# provee características de componentes orientados, como propiedades, eventos y construcciones declaradas (también llamados atributos). La programación orientada a componentes es soportada por el CLR. C# provee soporte para acceder directamente a la memoria usando el estilo de punteros de C++ y mucho más.

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes.

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones (API); mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco de DotGNU - Mono que genera programas para distintas plataformas como Win32, UNIX y Linux.

Dentro de la Interfaz de Usuario que hemos decidido plantear, hablamos de los controles ribbon ya que comprende todos los puntos de contacto entre el usuario y el equipo, y normalmente suelen ser fáciles de entender y fáciles de accionar.

Controles Ribbon

Es una interfaz donde un conjunto de barras de herramientas se colocan en las pestañas en una barra de pestañas.

Los controles ribbon ofrecen la interfaz de usuario de una aplicación con una barra de herramientas grande lleno de representaciones gráficas de los elementos de control que están agrupados por funcionalidad diferente. La cinta también puede contener etiquetas para exponer diferentes conjuntos de elementos de control que elimina la necesidad de una gran cantidad de diferentes barras de herramientas basadas en iconos.



Algunas de estas fichas son contextuales y solo aparecen cuando un cierto tipo de objeto seleccionado, ofreciendo herramientas específicas para los artículos tales como tablas e imágenes.

Las cintas tienen la forma moderna para ayudar a los usuarios a encontrar, entender y utilizar los comandos de manera eficiente y directa, con un número mínimo de clics, con menos necesidad de recurrir a prueba y error y sin tener que referirse a la Ayuda.

Cada aplicación tiene un conjunto diferente de pestañas que albergan las opciones para la aplicación específica. Dentro de cada ficha, diversas opciones relacionadas pueden ser agrupadas juntas. La cinta de opciones se puede minimizar haciendo clic en la pestaña activa



DISEÑO METODOLOGICO

La ingeniería del software está centrada en obtener aplicaciones económicas y fiables que funcionen de manera eficiente en máquinas reales. Nuestro Sistema consta de diferentes herramientas que unidas entre sí permiten una mejor calidad en el procesamiento de los datos.

Para el desarrollo y finalización de nuestro trabajo investigativo utilizaremos el MODELO DE CICLO DE VIDA en CASCADA, ya que este descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas la entrada requerida por la siguiente. En este modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

Actividades del Ciclo de Vida en Cascada

❖ Ingeniería y Análisis del Sistema

Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

❖ Análisis de los requisitos del Software

En esta actividad se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

❖ Diseño

El diseño del software se enfoca en cuatro atributos distintos del programa: La estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

❖ Codificación

El diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

❖ Prueba

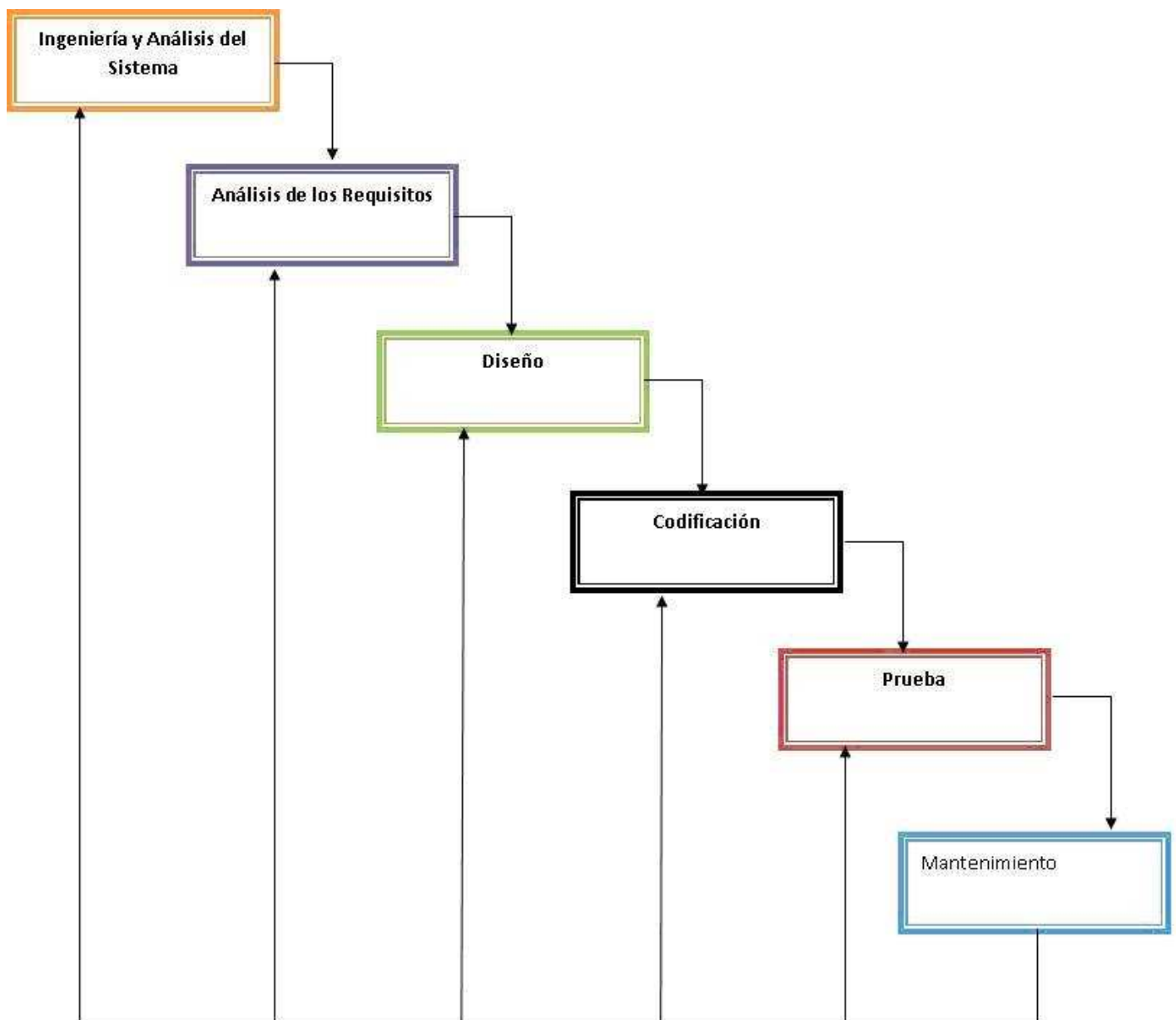
Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.



❖ *Mantenimiento*

El software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

El modelo de ciclo de vida en cascada se representa gráficamente de la siguiente manera:





En nuestro Trabajo de Investigación se recopiló la información, por medio de entrevistas y por correo electrónico, realizadas a los encargados de las respectivas áreas para el desarrollo del sistema, ofreciéndonos todos los requerimientos y funciones que necesitamos para realizar los procesos necesarios, con el objetivo de guiarnos y proporcionarnos los datos necesarios para la finalización del proyecto.

Y para el buen diseño e implementación del sistema, se hizo uso de los Libros y la Internet para la recopilación de los datos necesarios que utilizamos en el transcurso del proyecto.



RECURSOS Y NECESIDADES

❖ **Recurso Humano:**

Se constará con las personas involucradas en la realización y finalización de este proyecto, así como del encargado de facilitarnos toda la información disponible.

❖ **Herramientas Hardware y Software:**

Las herramientas Hardware necesarias para el diseño del sistema, serán las computadoras personales del recurso humano involucrado en el desarrollo del proyecto.

Las herramientas software necesarias para el diseño del sistema, será implementado bajo el sistema operativo Windows 7 Ultimate, Entorno de programación Microsoft Visual Studio 2008, Gestor de Bases de Datos SQL Server 2008 Express, Lenguaje de Programación C#.

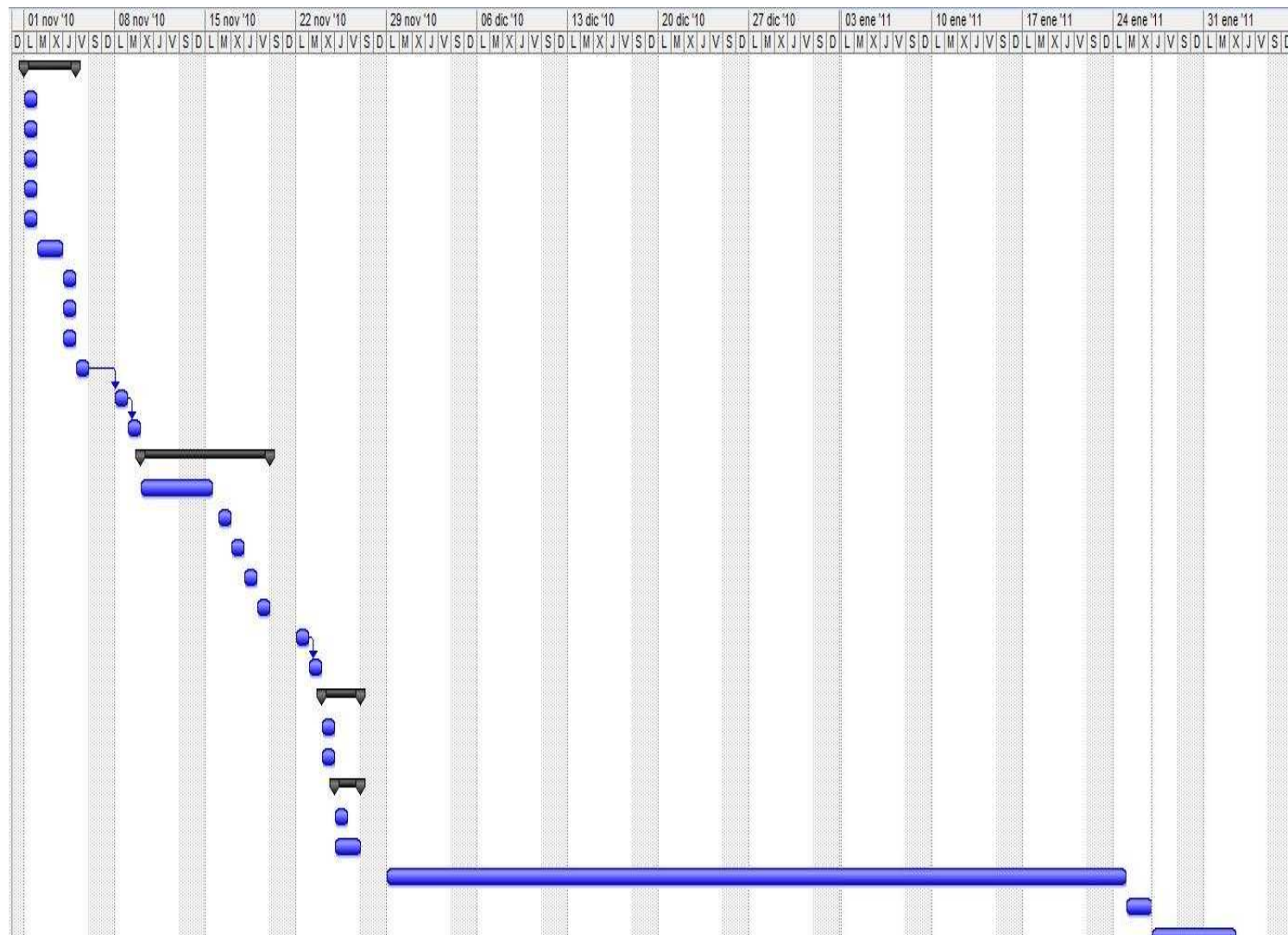
❖ **Recursos Económicos:**

No se contará con ningún recurso económico para la finalización de este proyecto.



CRONOGRAMA

	Nombre de tarea	Duración	Comienzo	Fin
1	☒ Elaboración de Protocolo	4.21 días	lun 01/11/10	jue 04/11/10
2	Tema	1 día	lun 01/11/10	lun 01/11/10
3	Introducción	1 día	lun 01/11/10	lun 01/11/10
4	Antecedentes	1 día	lun 01/11/10	lun 01/11/10
5	Justificación	1 día	lun 01/11/10	lun 01/11/10
6	Objetivos	1 día	lun 01/11/10	lun 01/11/10
7	Marco Teórico	2 días	mar 02/11/10	mié 03/11/10
8	Diseño Metodológico	1 día	jue 04/11/10	jue 04/11/10
9	Recursos y Necesidades	1 día	jue 04/11/10	jue 04/11/10
10	Bibliografía	1 día	jue 04/11/10	jue 04/11/10
11	Entrega de Protocolo	1 día	vie 05/11/10	vie 05/11/10
12	Corrección de Protocolo	1 día	lun 08/11/10	lun 08/11/10
13	Entrevista con el Tutor	1 día	mar 09/11/10	mar 09/11/10
14	☒ Análisis	8.64 días	mié 10/11/10	vie 19/11/10
15	ERS	4 días	mié 10/11/10	lun 15/11/10
16	Diag. Casos de Uso	1 día	mar 16/11/10	mar 16/11/10
17	Diag. de Clases	1 día	mié 17/11/10	mié 17/11/10
18	Diag. de Flujo de Datos	1 día	jue 18/11/10	jue 18/11/10
19	Modelo ER	1 día	vie 19/11/10	vie 19/11/10
20	Entrega de Análisis	1 día	lun 22/11/10	lun 22/11/10
21	Corrección de Análisis	1 día	mar 23/11/10	mar 23/11/10
22	☒ Diseño	3.07 días	mié 24/11/10	vie 26/11/10
23	Tablas	1 día	mié 24/11/10	mié 24/11/10
24	Diseño Arquitectónico	1 día	mié 24/11/10	mié 24/11/10
25	☒ Diseño Detallado	2 días	jue 25/11/10	vie 26/11/10
26	Procedimiento	1 día	jue 25/11/10	jue 25/11/10
27	Interfaz	2 días	jue 25/11/10	vie 26/11/10
28	Codificación	45 días	lun 29/11/10	lun 24/01/11
29	Implementación	2 días	mar 25/01/11	mié 26/01/11
30	Prueba y Mantenimiento	5 días	jue 27/01/11	mié 02/02/11





ESPECIFICACION DE REQUISITOS DE SOFTWARE

1. Introducción.

1.1. Propósito.

Definición del conjunto de especificaciones de requisitos software que debe cumplir la aplicación de **“SISTEMA DE CONTROL AUTOMATIZADO DEL PROCESAMIENTO DEL CAMARON PARA LA EMPRESA SAHLMAN SEAFOODS, S.A”**, que consiste en la automatización de los procesos de Recepción, Liquidación y Embarque necesarios para el procesamiento de producto entre otros.

Este documento se dirige al departamento de contabilidad y producción de la empresa, así como también para los usuarios finales que deberán manipular dicha aplicación.

1.2. Alcance.

El nombre con el que se conocerá esta aplicación será: **“SISTEMA DE CONTROL AUTOMATIZADO DEL PROCESAMIENTO DEL CAMARON PARA LA EMPRESA SAHLMAN SEAFOODS, S.A”**.

La aplicación realizará las siguientes funciones:

- Registrar y Actualizar datos generales Proveedores.
- Registrar y Actualizar datos generales Clientes.
- Registrar y Actualizar datos Granjas.
- Registrar y Actualizar datos correspondientes Magfor y Piscinas.
- Registrar y Actualizar Tallas del Producto.
- Registrar y Actualizar datos correspondientes de los Bines.
- Registrar y Actualizar información de Control Bines y Orden de Salida Hielo.
- Registrar y Actualizar datos Devolución de Hielo.
- Registrar y Actualizar datos correspondientes Tipos Master.
- Registrar y Actualizar datos Marca del Producto.
- Registrar y Actualizar datos correspondientes Tipos Cajas.
- Registrar y Actualizar datos correspondientes Tipos Procesos.
- Registrar y Actualizar datos Recepción General de Producto.
- Registrar y Actualizar datos Recepción de Producto Conicsa.
- Registrar y Actualizar Datos Control de Calidad General.
- Registrar y Actualizar Datos Control de Calidad Conicsa.
- Registrar y Actualizar Datos Peso Bruto en Planta.
- Registrar y Actualizar Datos Liquidación General para Camarón Entero.
- Registrar y Actualizar Datos Liquidación General para Camarón Cola.
- Registrar y Actualizar Datos Liquidación Conicsa para Camarón Entero.
- Registrar y Actualizar Datos Liquidación Conicsa para Camarón Cola.
- Registrar Master en bodega.
- Actualizar Master en Bodega.
- Registrar y Actualizar Control de Acceso de los Usuarios.



- Crear Copia de Seguridad.
- Configurar Cadena de Conexión.
- Ingresar al Sistema.
- Generación de los siguientes informes por periodos y fecha específica:
 - ❖ Informe de Recepción de Producto.
 - ❖ Informe de Control de Calidad de Producto.
 - ❖ Informe de Peso Bruto en Planta.
 - ❖ Informe de Liquidación de Producto.
 - ❖ Informe de Control de los Bines.
 - ❖ Informe de Libras Empacadas por Área.
 - ❖ Informe de Rendimiento de Camarón Entero.

1.3. Definiciones, acrónimos y abreviaturas.

- Proveedor: Empresa que suministra el producto para ser procesado por la empresa SAHLMAN; el resto del ERS lo llamara PROVEEDOR o PROVEEDORES.
- Clientes: Empresa que realiza compra del producto final, generado por el proveedor.
- Camarón Cola: Referente al camarón descabezado
- Camarón Entero: Referente al camarón, que contiene su cuerpo completo sin ninguna extracción.
- Marca: Se refiere al tipo de empaque que contiene al producto.
- Master: Se refiere al empaque o caja del producto final, listo para ser exportado.
- Proceso: Referente al tipo de producto elaborado en planta.
- HOSO: Tipo de Camarón o Proceso.
- HLSO: Tipo de Camarón o Proceso.
- PUD: Tipo de Camarón o Proceso.
- P&D: Tipo de Camarón o Proceso.
- Código: Conjunto de caracteres, números y fecha para identificar el proceso.
- Torre: Instrumento utilizada para almacenar temporalmente el producto.
- Bin o Bines: Contenedor donde se almacenan los camarones fresco, enviados por las granjas de los proveedores.
- Merma por Ecurrido: Humedad que traen los camarones fresco, al ser enviados de las granjas.
- Tara: peso de una cajilla.
- Materia Extraña: Animales y materiales que se extraen de las piscinas cosechadas.
- Piscinas: Lugar de cultivo de camarón situadas en las granjas.
- Unidad de Medidas: Libras, Kilogramos, Quintales, Caja.
- Talla: Cantidad de camarones por kilogramos.
- Color: Combinación de un número y letra determinado por el departamento de control de calidad.
- Clase: Nivel de calidad del Camarón determinado por el departamento de control de calidad.



1.4. Referencias.

- ✚ Tesis “Sitio Web de la Vicerrectoría, de Investigación y Postgrados de la UNAN-LEON”. Arauz Escobar, Eunices Elizabeth García. Suministrado por la Biblioteca Central UNAN – LEON.
- ✚ “Control Automatizado del Procesamiento del Camarón para la Empresa Ecuánica, S.A”. Luisa Elena Espinoza Valladares, Emma Margarita Vallecillo Hernández, Ana Walkiria Vega.

1.5. Visión General.

A continuación se hará una descripción general de la aplicación que se desea desarrollar y luego se estudiarán cada uno de los requisitos específicos.

2. Descripción General.

2.1. Relaciones de la Aplicación.

La aplicación será desarrollada en un equipo servidor con las siguientes características:

- o Memoria RAM de 12 GB
- o Disco Duro 160 GB(x2)
- o Procesador Xeon de 3.2 GHZ
- o Windows Server 2008 R2 MSDN 64 Bit

La instalación de dicha aplicación será realizada en el servidor de la empresa SAHLMAN SEAFOOD, para luego ser ejecutada por los usuarios que harán uso de ella por medio de “Cliente/Servidor”.

2.2. Funciones de la Aplicación.

La aplicación debe contener los campos necesarios para realizar las siguientes funciones:

1. Cuando se recibe el camarón se debe realizar la captura de datos de forma interactiva en el terminal.
2. Cuando se halla llevado a cabo el proceso de Recepción, se realizara la captura de datos referente al Control de Calidad.
3. Una vez evaluado el camarón, se procederá a ingresar la información correspondiente al Peso Bruto en Planta.
4. Cuando el camarón este procesado y listo para entrar en refrigeración, el usuario deberá registrar la entrada del mismo con sus respectivos datos.
5. Una vez refrigerado el producto, se capturaran los datos concernientes al proceso de Masterización.
6. Cuando el producto está terminado y listo para ser exportado se realizara un informe de rastreabilidad de producto terminado.
7. El usuario podrá disponer de los siguientes informes de frecuencia diaria o por periodo.
 - ❖ Informe de Recepción de Producto.
 - ❖ Informe de Control de Calidad de Producto.



- ❖ Informe de Peso Bruto en Planta.
- ❖ Informe de Liquidación de Producto.
- ❖ Informe de Control de los Bines.
- ❖ Informe de Libras Empacadas por Área.
- ❖ Informe de Rendimiento de Camarón Entero.

2.3. Características del Usuario.

Los usuarios finales de la aplicación deberán tener conocimientos previos en informática, sin embargo se incluirá un documento de ayuda y se capacitaran a dichos usuarios para la correcta utilización.

2.4. Restricciones Generales.

El lenguaje de programación para las bases de datos es SQL SERVER 2008 con Transact-SQL y para el diseño de la aplicación se utilizara Visual Studio 2008 con C# como lenguaje bajo la plataforma .NET (Windows) como sistema operativo.

2.5. Suposiciones y dependencias.

3. Requisitos Específicos.

3.1. Requisitos Específicos Funcionales.

3.1.1. Ingresar al Sistema.

3.1.1.1. Especificación.

3.1.1.1.1. Introducción.

Los usuarios registrados podrán ingresar al sistema para hacer uso de sus funciones.

3.1.1.1.2. Entradas.

Los usuarios deberán proporcionar al sistema la siguiente información: Nombre y Contraseña de usuario.

3.1.1.1.3. Proceso.

Se validaran los datos introducidos y si son correctos se procederá a ingresar al sistema.

3.1.1.1.4. Salidas.

Si la información ingresada es válida se habilitaran los controles correspondientes al nivel del usuario, de lo contrario se mostrara un mensaje indicando que los datos no son válidos.

3.1.2. Registrar y Actualizar Datos Generales Proveedores.

3.1.2.1. Especificación.

3.1.2.1.1. Introducción.

Este proceso deberá realizar la captura por terminal de todos los datos pertenecientes a los proveedores.

3.1.2.1.2. Entradas.

Por pantalla:

- ✓ **Ruc:** es un dato obligatorio. Almacena el número correspondiente que identifica a un proveedor debe existir en el fichero maestro(o tabla) de proveedores o se enviara un mensaje de error.



- ✓ **Nombre:** es un dato obligatorio. Almacena el nombre del proveedor.
- ✓ **Nombre Comercial:** no es un dato obligatorio. Almacena el nombre comercial que posee un proveedor.
- ✓ **Codigo_Planta:** es un dato obligatorio. Almacena el código planta que hace referencia a un proveedor.
- ✓ **Departamento:** es un dato obligatorio. Almacena el departamento donde se encuentran ubicadas las instalaciones del proveedor.
- ✓ **Dirección:** es un dato obligatorio. Almacena la dirección física donde se encuentran ubicadas las instalaciones del proveedor.
- ✓ **Contacto:** es un dato obligatorio. Almacena el nombre de la persona encargada de contactar a la empresa.
- ✓ **email:** es un dato obligatorio. Almacena la dirección de correo electrónico que posee el proveedor y que se usará como uno de los medios de comunicación. Este deberá llevar el formato correspondiente a los correos electrónicos.
- ✓ **Estado:** es un dato obligatorio. Almacena el estado del proveedor si es activo o inactivo.
- ✓ **Sobrepeso:** es un dato obligatorio. Almacena el sobrepeso del empaque del producto propio de cada proveedor.
- ✓ **Numero:** es un dato obligatorio. Almacena los números telefónicos.
- ✓ **Descripción:** es un dato obligatorio. Almacena una descripción del número telefónico; celular, casa, trabajo, fax.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.2.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Rut**
- ✓ **Nombre**
- ✓ **Nombre_Comercial**
- ✓ **Codigo_Planta**
- ✓ **Departamento**
- ✓ **Dirección**
- ✓ **Contacto**
- ✓ **Email**



- ✓ **Estado**
- ✓ **Sobrepeso**
- ✓ **Numero**
- ✓ **Descripción**

Los campos Estado y Departamento podrán ser seleccionados de forma selectiva de una lista desplegable.

Para guarda los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.2.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros de los datos generales de los proveedores.

3.1.3. Registrar y Actualizar Datos Generales Clientes.

3.1.3.1.Especificación.

3.1.3.1.1. Introducción.

Este proceso deberá realizar la captura por terminal de todos los datos pertenecientes a los Clientes.

3.1.3.1.2. Entradas.

Por pantalla:

- ✓ **Ruc:** es un dato obligatorio. Almacena el número correspondiente que identifica a un proveedor debe existir en el fichero maestro(o tabla) de proveedores o se enviara un mensaje de error.
- ✓ **Nombre:** es un dato obligatorio. Almacena el nombre del Cliente.
- ✓ **Dirección:** es un dato obligatorio. Almacena la dirección física donde se encuentran ubicadas las instalaciones del Cliente.
- ✓ **Contacto:** es un dato obligatorio. Almacena el nombre de la persona encargara de realizar el contrato con el proveedor.
- ✓ **Numero:** es un dato obligatorio. Almacena los números telefónicos.
- ✓ **Descripción:** es un dato obligatorio. Almacena una descripción del número telefónico; celular, casa, trabajo, fax.



Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.3.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Rut**
- ✓ **Nombre**
- ✓ **Dirección**
- ✓ **Contacto**
- ✓ **Numero**
- ✓ **Descripción**

Para guarda los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.3.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros de los datos generales de los Clientes.

3.1.4. Registrar y Actualizar Datos Granjas.

3.1.4.1. Especificación.

3.1.4.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema los datos de cada una de las granjas que pertenecen a un proveedor.

3.1.4.1.2. Entradas.

Por pantalla:

- ✓ **CodGranja:** es un dato obligatorio. Almacena el código granja con el que se identifica a cada granja que posee el proveedor
- ✓ **Ruc:** es un dato obligatorio. Almacena el número correspondiente que identifica a un proveedor debe



existir en el fichero maestro(o tabla) de proveedores o se enviara un mensaje de error.

- ✓ **Nombre:** es un dato obligatorio. Almacena el nombre de cada una de las granjas propiedad del proveedor.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.4.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **CodGranja**
- ✓ **Ruc**
- ✓ **Nombre**

El campo Ruc podrá ser seleccionado de forma selectiva de una lista desplegable.

Para guarda los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.4.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros de las granjas.

3.1.5. Registrar y Actualizar Datos Correspondientes Magfor y Piscinas.

3.1.5.1.Especificación.

3.1.5.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema los datos correspondientes al número Magfor y las piscinas.

3.1.5.1.2. Entradas.

Por pantalla:

- ✓ **Codigo_Magfor:** es un dato obligatorio. Almacena el código asignado por el Magfor.
- ✓ **CodGranja:** es un dato obligatorio. Almacena el código granja con el que se identifica a cada granja que posee el proveedor.
- ✓ **Numero:** es un dato obligatorio. Almacena el número que identifica a cada piscina.



- ✓ **Descripción:** es un dato opcional. Almacena un breve descripción de las piscinas.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.5.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Codigo_Magfor**
- ✓ **CodGranja**
- ✓ **Numero**
- ✓ **Descripción**

El campo CodGranja podrá ser seleccionado de forma selectiva de una lista desplegable.

Previamente antes de introducir una piscina se debe registrar el número magfor a la que corresponde.

Para guarda los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro a través de las direccionales pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

3.1.5.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros del Magfor y de las Piscinas.

3.1.6. Registrar y Actualizar Tallas del Producto.

3.1.6.1. Especificación.

3.1.6.1.1. Introducción.

Este proceso deberá realizar la captura de toda la información acerca de las tallas del Producto.

3.1.6.1.2. Entradas.

Por pantalla:

- ✓ **Nombre:** es un dato obligatorio. Almacena el conjunto de caracteres que conforman una talla.



- ✓ **Camarón:** es un dato obligatorio. Almacena el tipo de producto al que pertenece una talla.
- ✓ **IdProceso:** es un dato obligatorio. Almacena el id del proceso al que pertenece una talla.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.6.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Nombre**
- ✓ **Camarón**
- ✓ **IdProceso**

El campo IdProceso podrá ser seleccionado de forma selectiva de una lista desplegable.

Para guarda los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.6.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros referentes a las tallas del producto.

3.1.7. Registrar y Actualizar Datos correspondiente de los Bines.

3.1.7.1.Especificación.

3.1.7.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información correspondiente al control y manejo de los bines.



3.1.7.1.2. Entradas.

Por pantalla:

- ✓ **FechaAdq:** es un dato obligatorio. Este campo almacena la fecha en que fue adquirido dicho Bin.
- ✓ **NumeroBin:** es un dato obligatorio. Almacena el número único correspondiente a cada bin, debe existir en el fichero maestro(o tabla) de Bines o se indicara un mensaje de error.
- ✓ **Descripción:** es un dato opcional. Almacena una información adicional respecto al bin.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.7.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **FechaAdq**
- ✓ **NumeroBin**
- ✓ **Descripción**

Para guarda los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.7.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros correspondiente a los bines.



3.1.8. Registrar y Actualizar Información de Control Bines y Orden de Salida Hielo.

3.1.8.1. Especificación.

3.1.8.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información correspondiente al control de los bins y ordenes de salida de hielo.

3.1.8.1.2. Entradas.

Por pantalla:

- ✓ **Fecha:** es un dato obligatorio. Este campo almacena la fecha en que fue realizado el préstamo del bin.
- ✓ **Ruc:** es un dato obligatorio. Almacena el número correspondiente que identifica a un proveedor.
- ✓ **Vehículo:** es un dato obligatorio. Almacena una información detallada del vehículo.
- ✓ **E_S:** es un dato obligatorio. Almacena el tipo de operación que se está realizando.
- ✓ **NumeroBin:** es un dato obligatorio. Almacena el número único correspondiente a cada bin.
- ✓ **Propiedad:** es un dato obligatorio. Almacena el propietario del bin.
- ✓ **CantHielo:** es un dato obligatorio. Almacena la cantidad de hielo por bin.
- ✓ **IdUM:** es un dato obligatorio. Almacena el id correspondiente a una unidad de medida.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.8.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**
- ✓ **Ruc**
- ✓ **Vehículo**
- ✓ **E_S**
- ✓ **NumeroBin**
- ✓ **Propiedad**
- ✓ **CantHielo**

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.



Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.8.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros acerca del control de bins y la orden de salida de hielo.

3.1.9. Registrar y Actualizar Datos Devolución de Hielo.

3.1.9.1. Especificación.

3.1.9.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a una devolución de hielo.

3.1.9.1.2. Entradas.

Por pantalla:

- ✓ **Fecha:** es un dato obligatorio. Este campo almacena la fecha en que fue realizado la devolución de hielo.
- ✓ **Nombre:** es un dato obligatorio. Almacena el nombre de la persona encargada de llevar a cabo la devolución de hielo.
- ✓ **Empresa:** es un dato obligatorio. Almacena el nombre de la empresa que realiza la devolución.
- ✓ **CantDev:** es un dato obligatorio. Almacena la cantidad de hielo devuelta.
- ✓ **IdUM:** es un dato obligatorio. Almacena el id correspondiente a una unidad de medida.
- ✓ **Placa:** es un dato obligatorio. Almacena la placa del vehículo que realiza la devolución.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.9.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**



- ✓ **Nombre**
- ✓ **CantDev**
- ✓ **IdUM**
- ✓ **Placa**

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.9.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros respecto a la devolución de hielo realizada por determinada empresa.

3.1.10. Registrar y Actualizar Datos Correspondientes Tipos Master.

3.1.10.1. Especificación.

3.1.10.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a los tipos de master existentes en la empresa.

3.1.10.1.2. Entradas.

Por pantalla:

- ✓ **Marca:** es un dato obligatorio. Este campo almacena información del tipo de empaque que contiene al producto.
- ✓ **IdUM:** es un dato obligatorio. Almacena el id correspondiente a una unidad de medida.
- ✓ **Peso:** es un dato obligatorio. Almacena el peso del producto en la unidad de medida seleccionada.
- ✓ **Capacidad:** es un dato obligatorio. Almacena la cantidad de cajas que puede contener un master según la marca seleccionada.
- ✓ **Descrip:** es un dato obligatorio. Almacena una breve descripción del master.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada



usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.10.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Marca**
- ✓ **IdUM**
- ✓ **Peso**
- ✓ **Capacidad**
- ✓ **Descrip**

Los campos Marca e IdUM se seleccionaran de manera selectiva de una lista desplegable.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.10.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros respecto a los tipos de master.

3.1.11. Registrar y Actualizar Datos Marca del Producto.

3.1.11.1. Especificación.

3.1.11.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la marca del producto.

3.1.11.1.2. Entradas.

Por pantalla:

- ✓ **Marca:** es un dato obligatorio. Este campo almacena información del tipo de empaque que contiene al producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.



- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.11.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Marca**

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.11.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros respecto a la Marca del producto.

3.1.12. Registrar y Actualizar Datos Correspondientes Tipos Cajas.

3.1.12.1. Especificación.

3.1.12.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a los tipos de cajas utilizadas durante el procesamiento del producto.

3.1.12.1.2. Entradas.

Por pantalla:

- ✓ **IdMarca:** es un dato obligatorio. Este campo almacena id correspondiente a una marca.
- ✓ **Peso:** es un dato obligatorio. Almacena el peso del producto en la unidad de medida seleccionada.
- ✓ **IdUM:** es un dato obligatorio. Almacena el id correspondiente a una unidad de medida.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.



- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.12.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **IdMarca**
- ✓ **Peso**
- ✓ **IdUM**

Los campos Marca y IdUM se seleccionaran de manera selectiva de una lista desplegable.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.12.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros respecto al Tipo de Caja del producto.

3.1.13. Registrar y Actualizar Datos Correspondientes Tipos Procesos.

3.1.13.1. Especificación.

3.1.13.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a los Tipos de Procesos utilizadas durante el procesamiento del producto.

3.1.13.1.2. Entradas.

Por pantalla:

- ✓ **Proceso:** es un dato obligatorio. Este campo almacena información referente al tipo de proceso utilizado durante el procesamiento del producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema



guarda de tal forma que se solicita una sola vez cuando se inicia sesión.

- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.13.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Proceso**

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.13.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros respecto al Tipo de Proceso.

3.1.14. Registrar y Actualizar Datos Recepción General de Producto.

3.1.14.1. Especificación.

3.1.14.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la Recepción General de Producto.

3.1.14.1.2. Entradas.

Por pantalla:

- ✓ **Fecha:** es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **Ruc:** es un dato obligatorio. Almacena el número correspondiente que identifica a un proveedor.
- ✓ **Granja:** es un dato obligatorio. Almacena el nombre de la granja propiedad del proveedor.
- ✓ **NumPiscina:** es un dato obligatorio. Almacena el número de piscina del que ha sido extraído el producto.
- ✓ **CicloP:** es un dato obligatorio. Almacena el ciclo producción con en que fue procesado el producto.
- ✓ **Conductor:** es un dato obligatorio. Almacena el nombre del conductor encargado de transportar el producto.



- ✓ **VehPlaca:** es un dato obligatorio. Almacena la placa que posee un vehículo.
- ✓ **VehMarca:** es un dato obligatorio. Almacena la marca que posee un vehículo.
- ✓ **SupRecepción:** es un dato obligatorio. Almacena el nombre de la persona encargada de supervisar la Recepción General del Producto.
- ✓ **SupGranja:** es un dato obligatorio. Almacena el nombre de la persona designada por el proveedor para supervisar el procesamiento de su producto.
- ✓ **Observaciones:** no es un dato obligatorio. Almacena información sobre el proceso.
- ✓ **NumeroBin:** es un dato obligatorio. Almacena el número único que identifica a cada Bin.
- ✓ **Peso:** es un dato obligatorio. Almacena el peso en libras correspondientes a cada Bin.
- ✓ **Num_Entrega:** Almacena el número correspondiente a la entrega de Producto por vehículo.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.14.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**
- ✓ **Ruc**
- ✓ **Granja**
- ✓ **NumPiscina**
- ✓ **CicloP**
- ✓ **Conductor**
- ✓ **VehPlaca**
- ✓ **VehMarca**
- ✓ **SupRecepción**
- ✓ **SupGranja**
- ✓ **Observaciones**
- ✓ **NumeroBin**
- ✓ **Peso**
- ✓ **Num_Entrega**

Los campos Fecha, Ruc, Granja y Piscina podrán ser seleccionados de forma selectiva de una lista desplegable.



Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.14.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros a la Recepción General de Producto.

3.1.15. Registrar y Actualizar Datos Recepción de Producto Conicsa.

3.1.15.1. Especificación.

3.1.15.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la Recepción de Producto Conicsa.

3.1.15.1.2. Entradas.

Por pantalla:

- ✓ **Fecha:** es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **Ruc:** es un dato obligatorio. Almacena el número correspondiente que identifica a un proveedor.
- ✓ **Granja:** es un dato obligatorio. Almacena el nombre de la granja propiedad del proveedor.
- ✓ **CicloP:** es un dato obligatorio. Almacena el ciclo producción con en que fue procesado el producto.
- ✓ **Naviera:** es un dato obligatorio. Almacena el nombre de la compañía encargada de transportar el producto.
- ✓ **NumContenedor:** es un dato obligatorio. Guarda el nombre del contenedor donde es almacenado el producto.
- ✓ **Conductor:** es un dato obligatorio. Almacena el nombre del conductor encargado de transportar el producto.
- ✓ **VehPlaca:** es un dato obligatorio. Almacena la placa que posee un vehículo.
- ✓ **VehMarca:** es un dato obligatorio. Almacena la marca que posee un vehículo.
- ✓ **SupRecepción:** es un dato obligatorio. Almacena el nombre de la persona encargada de supervisar la Recepción General del Producto.
- ✓ **SupGranja:** es un dato obligatorio. Almacena el nombre de la persona designada por el proveedor para supervisar el procesamiento de su producto.



- ✓ **Observaciones:** no es un dato obligatorio. Almacena información sobre el proceso.
- ✓ **Barco:** es un dato obligatorio. Almacena el nombre del barco encargado de transportar el producto.
- ✓ **IdProceso:** es un dato obligatorio. Almacena el id perteneciente a un proceso.
- ✓ **Color:** es un dato obligatorio. Almacena el color de los productos estos pueden ser PINK O WHITE.
- ✓ **Peso:** es un dato obligatorio. Almacena el peso en libras correspondientes al proceso seleccionado.
- ✓ **NCajillas:** es un dato obligatorio. Almacena la cantidad de cajillas que contiene al producto de acuerdo al proceso seleccionado.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.15.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**
- ✓ **Ruc**
- ✓ **Granja**
- ✓ **CicloP**
- ✓ **Naviera**
- ✓ **NumContenedor**
- ✓ **Conductor**
- ✓ **VehPlaca**
- ✓ **VehMarca**
- ✓ **SupRecepción**
- ✓ **SupGranja**
- ✓ **Observaciones**
- ✓ **Barco**
- ✓ **IdProceso**
- ✓ **Color**
- ✓ **Peso**
- ✓ **NCajillas**

Los campos Fecha, Ruc, Granja y IdProceso podrán ser seleccionados de forma selectiva de una lista desplegable.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.



Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.15.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros a la Recepción de Producto Conicsa.

3.1.16. Registrar y Actualizar Datos Control de Calidad General.

3.1.16.1. Especificación.

3.1.16.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente al Control de Calidad General.

3.1.16.1.2. Entradas.

Por pantalla:

- ✓ **Fecha:** es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **Código:** es un dato obligatorio. Almacena un conjunto de caracteres que identifican el proceso.
- ✓ **TotalLbsReport:** es un dato obligatorio. Almacena el total en libras reportadas por entrega.
- ✓ **SupCalidad:** es un dato opcional. Almacena el nombre de la persona encargada de supervisar el proceso.
- ✓ **Producción:** es un dato opcional. Almacena el nombre de la persona encargada de la producción.
- ✓ **ControlCalidad:** es un dato opcional. Almacena el nombre de la persona encargada de realizar el proceso de control de calidad.
- ✓ **SupProducto:** es un dato opcional. Almacena el nombre de la persona encargada de supervisar el producto.
- ✓ **Producción:** es un dato opcional. Almacena el nombre de la persona encargada de la producción.
- ✓ **Observaciones:** no es un dato obligatorio. Almacena información sobre el resultado del proceso.
- ✓ **NumeroBin:** es un dato obligatorio. Almacena el número único que identifica a cada Bin.
- ✓ **Temperatura:** es un dato obligatorio. Almacena información referente a la temperatura del producto.
- ✓ **NumCamKG:** es un dato obligatorio. Almacena la cantidad de camarones por kilogramos.
- ✓ **Conteo2KG:** es un dato obligatorio. Almacena la cantidad de camarones por kilogramos duplicada.



- ✓ **MateriaExtrana:** es un dato opcional. Almacena información sobre materia extraña que puede presentar el producto.
- ✓ **VANN:** es un dato opcional. Almacena la cantidad de camarones que presentaron este defecto en lo sucesivo VANN.
- ✓ **LARVA:** es un dato opcional. Almacena la cantidad de camarones que presentaron este defecto en lo sucesivo LARVA.
- ✓ **STYL:** es un dato opcional. Almacena la cantidad de camarones que presentaron este defecto en lo sucesivo STYL.
- ✓ **Defecto:** es un dato obligatorio. Almacena el tipo de defecto que podría presentar cierta cantidad de producto.
- ✓ **Cantidad:** es un dato obligatorio. Almacena la cantidad de camarones en relación a un defecto seleccionado.
- ✓ **Porcentaje:** es un dato obligatorio. Almacena el porcentaje correspondientes a una cantidad en relación a un tipo de defecto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.16.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**
- ✓ **Código**
- ✓ **TotalLbsReport**
- ✓ **SupCalidad**
- ✓ **Producción**
- ✓ **ControlCalidad**
- ✓ **SupProducto**
- ✓ **Producción**
- ✓ **Observaciones**
- ✓ **NumeroBin**
- ✓ **Temperatura**
- ✓ **NumCamKG**
- ✓ **Conteo2KG**
- ✓ **MateriaExtrana**
- ✓ **VANN**
- ✓ **LARVA**
- ✓ **STYL**



- ✓ Defecto
- ✓ Cantidad
- ✓ Porcentaje

Los campos NumeroBin y Defecto podrán ser seleccionados de forma selectiva de una lista desplegable.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.16.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros correspondientes al Control de Calidad General.

3.1.17. Registrar y Actualizar Datos Control de Calidad Conicsa.

3.1.17.1. Especificación.

3.1.17.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente al Control de Calidad Conicsa.

3.1.17.1.2. Entradas.

Por pantalla:

- ✓ **Fecha:** es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **Código:** es un dato obligatorio. Almacena un conjunto de caracteres que identifican el proceso.
- ✓ **IdProceso:** es un dato obligatorio. Almacena el id perteneciente a un proceso.
- ✓ **SupCalidad:** es un dato opcional. Almacena el nombre de la persona encargada de supervisar el proceso.
- ✓ **Producción:** es un dato opcional. Almacena el nombre de la persona encargada de la producción.
- ✓ **ControlCalidad:** es un dato opcional. Almacena el nombre de la persona encargada de realizar el proceso de control de calidad.
- ✓ **SupProducto:** es un dato opcional. Almacena el nombre de la persona encargada de supervisar el producto.
- ✓ **Observaciones:** no es un dato obligatorio. Almacena información sobre el resultado del proceso.



- ✓ **Color:** es un dato obligatorio. Almacena el color de los productos estos pueden ser PINK O WHITE.
- ✓ **Defecto:** es un dato obligatorio. Almacena el tipo de defecto que podría presentar cierta cantidad de producto.
- ✓ **PesoMuestra:** es un dato obligatorio. Almacena el peso de una muestra que serán sometidas a control de calidad.
- ✓ **Peso:** es un dato obligatorio. Almacena el peso correspondiente a cierto defecto sobre el peso de muestra.
- ✓ **Porcentaje:** es un dato obligatorio. Almacena el porcentaje correspondientes a una cantidad en relación a un tipo de defecto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.17.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**
- ✓ **Código**
- ✓ **IdProceso**
- ✓ **SupCalidad**
- ✓ **Producción**
- ✓ **ControlCalidad**
- ✓ **SupProducto**
- ✓ **Observaciones**
- ✓ **Color**
- ✓ **Defecto**
- ✓ **PesoMuestra**
- ✓ **Peso**
- ✓ **Porcentaje**

Los campos IdProceso, Defecto y Color podrán ser seleccionados de forma selectiva de una lista desplegable.

Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.



Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.17.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros correspondientes al Control de Calidad Conicsa.

3.1.18. Registrar y Actualizar Datos Peso Bruto en Planta.

3.1.18.1. Especificación.

3.1.18.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente al Control de Calidad Conicsa.

3.1.18.1.2. Entradas.

Este proceso deberá realizar la captura de toda la información referente al peso bruto reportado en la planta; dicho proceso incluirá la extracción de la materia extraña presente en los bines, que posteriormente será restado del peso bruto y obtener el peso neto total en planta.

Por pantalla:

- ✓ **Fecha:** Es un dato obligatorio. Almacena la fecha de dicho proceso que se está realizando.
- ✓ **Código:** Es un dato obligatorio. Almacena información que está compuesta por caracteres y números específicos asignados a los proveedores por la empresa.
- ✓ **Origen:** Es un dato obligatorio. Almacena la especie del producto (Farmed o Wild.)
- ✓ **TipoPeso:** Es un dato obligatorio. Almacena el tipo de peso que se está realizando.
- ✓ **Peso:** Es un dato obligatorio. Almacena el peso de las gavetas con producto, realizadas por dicho proceso.
- ✓ **UndMedida:** Es un dato obligatorio. Almacena la unidad de Medida con que se está trabajando en dicho proceso.
- ✓ **PNConicsa:** Es un dato obligatorio. Almacena el peso neto, que la empresa procesara y congelará, para CONICSA.
- ✓ **CajProc:** Es un dato obligatorio. Almacena la cantidad total de cajillas con respecto al peso neto que procesará la empresa, para CONICSA.
- ✓ **Proceso:** Es un dato obligatorio. Almacena el nombre del tipo de proceso que está llevando a cabo dicho proceso, para CONICSA.
- ✓ **Barco:** Es un dato obligatorio. Almacena el número del barco que obtuvo el producto y llevándose a cabo en dicho proceso, para CONICSA.



- ✓ **LbsEntero:** Es un dato obligatorio. Almacena la Cantidad Total de Libras de Camarón Entero que se procesaran en dicho proceso.
- ✓ **Observaciones:** Es un dato opcional. Almacena alguna información adicional de dicho proceso, ocurrido en ese instante.
- ✓ **SupRecep:** Es un dato opcional. Almacena el nombre de la persona encargada de supervisar la recepción del producto.
- ✓ **SupGranja:** Es un dato opcional. Almacena el nombre de la persona designada por el proveedor para supervisar el procesamiento de su producto.
- ✓ **GteProduccion:** Es un dato opcional. Almacena el nombre del gerente de producción de la empresa.
- ✓ **Gaveta:** Es un dato obligatorio. Genera Consecutivamente el número de gaveta en la que se está depositando el producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.18.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha**
- ✓ **Código**
- ✓ **Origen**
- ✓ **TipoPeso**
- ✓ **Peso**
- ✓ **UndMedida**
- ✓ **PNConicsa**
- ✓ **CajProc**
- ✓ **Proceso**
- ✓ **Barco**
- ✓ **LbsEntero**
- ✓ **Observaciones**
- ✓ **SupRecep**
- ✓ **SupGranja**
- ✓ **GteProduccion**
- ✓ **Gaveta**

Los campos Fecha, Origen, Barco y Proceso podrán ser seleccionados de forma selectiva de una lista desplegable.



Para guardar los datos introducidos en la base de datos se deberá pulsar el botón guardar.

Para actualizar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos seleccionamos el registro de un datagrid pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.18.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardaran los nuevos registros correspondientes al Peso Bruto en Planta.

3.1.19. Registrar y Actualizar Datos Liquidación General para Camarón Entero.

3.1.19.1. Especificación.

3.1.19.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la liquidación de producto, Camarón Entero, para un proveedor General.

3.1.19.1.2. Entradas.

Por Pantalla:

- ✓ **Fecha:** Es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **IdProceso:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de proceso.
- ✓ **Para:** Es un dato obligatorio. Almacena el nombre del cliente a quien va dirigido el producto terminado.
- ✓ **IdTCaja:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de caja.
- ✓ **Código:** Es un dato obligatorio. Almacena un conjunto de caracteres que identifican el proceso.
- ✓ **Color:** Es un dato obligatorio. Almacena el color de los productos, estos pueden ser: Pink o White.
- ✓ **Liquidador:** Es un dato obligatorio. Almacena el nombre de la persona encargada de la liquidación del producto.
- ✓ **Observaciones:** No es un dato obligatorio. Almacena información sobre el resultado del proceso.
- ✓ **IdTalla:** Es un dato Obligatorio. Almacena el id correspondiente referente a una talla específica del producto.
- ✓ **Torre:** Es un dato obligatorio. Almacena el número o identificador correspondiente de una torre.
- ✓ **CantCajas:** Es un dato obligatorio. Almacena la cantidad de cajas obtenidas por talla y por torre.



- ✓ **Blass:** Es un dato obligatorio. Almacena el nombre del cuarto de congelación del producto.
- ✓ **Clasificación:** Es un dato obligatorio. Almacena el nombre que identifica la clasificación del producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.19.1.3. Proceso.

Se mostrará al usuario una interfaz amigable en la cual se deberán de ingresar los siguientes campos:

- ✓ **Fecha.**
- ✓ **IdProceso.**
- ✓ **Para.**
- ✓ **IdTCaja.**
- ✓ **Código.**
- ✓ **Color.**
- ✓ **Liquidador.**
- ✓ **Observaciones.**
- ✓ **IdTalla.**
- ✓ **Torre.**
- ✓ **CantCajas.**
- ✓ **Blass.**
- ✓ **Clasificación.**

El campo Para podrá ser seleccionado de forma selectiva de una lista desplegable.

El campo IdTCaja podrá ser seleccionado por medio de direccionales, donde se mostrará al usuario una información referente a la caja con la que se desea liquidar.

Para guardar los datos ingresados en el formulario, es necesario presionar el botón guardar para ser almacenado en la base de datos.

Para actualizar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón editar, se habilitarán las cajas de textos correspondientes a dicho registro; una vez realizada la modificación se pulsará el botón guardar.

Para eliminar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón eliminar; luego aparecerá un mensaje de confirmación donde



pulsaremos la opción “sí” y automáticamente se eliminará el registro.

3.1.19.1.4. Salidas.

Con los datos antes mencionados se guardarán los nuevos registros con la información relativa acerca del proceso liquidación General del producto camarón entero.

3.1.20. Registrar y Actualizar Datos Liquidación General para Camarón Cola.

3.1.20.1. Especificación.

3.1.20.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la liquidación de producto, Camarón Cola, para un proveedor General.

3.1.20.1.2. Entradas.

Por Pantalla:

- ✓ **Fecha:** Es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **IdProceso:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de proceso.
- ✓ **Para:** Es un dato obligatorio. Almacena el nombre del cliente a quien va dirigido el producto terminado.
- ✓ **IdTCaja:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de caja.
- ✓ **Código:** Es un dato obligatorio. Almacena un conjunto de caracteres que identifican el proceso.
- ✓ **Color:** Es un dato obligatorio. Almacena el color de los productos, estos pueden ser: Pink o White.
- ✓ **Liquidador:** Es un dato obligatorio. Almacena el nombre de la persona encargada de la liquidación del producto.
- ✓ **Observaciones:** No es un dato obligatorio. Almacena información sobre el resultado del proceso.
- ✓ **IdTalla:** Es un dato Obligatorio. Almacena el id correspondiente referente a una talla específica del producto.
- ✓ **Torre:** Es un dato obligatorio. Almacena el número o identificador correspondiente de una torre.
- ✓ **CantCajas:** Es un dato obligatorio. Almacena la cantidad de cajas obtenidas por talla y por torre.
- ✓ **Blass:** Es un dato obligatorio. Almacena el nombre del cuarto de congelación del producto.
- ✓ **Clasificación:** Es un dato obligatorio. Almacena el nombre que identifica la clasificación del producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema



guarda de tal forma que se solicita una sola vez cuando se inicia sesión.

- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.20.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha.**
- ✓ **IdProceso.**
- ✓ **Para.**
- ✓ **IdTCaja.**
- ✓ **Código.**
- ✓ **Color.**
- ✓ **Liquidador.**
- ✓ **Observaciones.**
- ✓ **IdTalla.**
- ✓ **Torre.**
- ✓ **CantCajas.**
- ✓ **Blass.**
- ✓ **Clasificación.**

El campo Para podrá ser seleccionado de forma selectiva de una lista desplegable.

El campo IdTCaja podrá ser seleccionado por medio de direccionales, donde se mostrará al usuario una información referente a la caja con la que se desea liquidar.

Para guardar los datos ingresados en el formulario, es necesario presionar el botón guardar para ser almacenados en la base de datos.

Para actualizar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón editar, se habilitarán las cajas de textos correspondientes a dicho registro; una vez realizada la modificación se pulsará el botón guardar.

Para eliminar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón eliminar; luego aparecerá un mensaje de confirmación donde pulsaremos la opción “si” y automáticamente se eliminará el registro.

3.1.20.1.4. Salidas.

Con los datos antes mencionados se guardarán los nuevos registros con la información relativa acerca del proceso liquidación General del producto camarón cola.



3.1.21. Registrar y Actualizar Datos Liquidación Conicsa para Camarón Entero.

3.1.21.1. Especificación.

3.1.21.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la liquidación de producto, Camarón Entero, para Conicsa.

3.1.21.1.2. Entradas.

Por Pantalla:

- ✓ **Fecha:** Es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **IdProceso:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de proceso.
- ✓ **Para:** Es un dato obligatorio. Almacena el nombre del cliente a quien va dirigido el producto terminado.
- ✓ **IdTCaja:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de caja.
- ✓ **Código:** Es un dato obligatorio. Almacena un conjunto de caracteres que identifican el proceso.
- ✓ **Color:** Es un dato obligatorio. Almacena el color de los productos, estos pueden ser: Pink o White.
- ✓ **Liquidador:** Es un dato obligatorio. Almacena el nombre de la persona encargada de la liquidación del producto.
- ✓ **Observaciones:** No es un dato obligatorio. Almacena información sobre el resultado del proceso.
- ✓ **IdTalla:** Es un dato Obligatorio. Almacena el id correspondiente referente a una talla específica del producto.
- ✓ **Torre:** Es un dato obligatorio. Almacena el número o identificador correspondiente de una torre.
- ✓ **CantCajas:** Es un dato obligatorio. Almacena la cantidad de cajas obtenidas por talla y por torre.
- ✓ **Blass:** Es un dato obligatorio. Almacena el nombre del cuarto de congelación del producto.
- ✓ **Clasificación:** Es un dato obligatorio. Almacena el nombre que identifica la clasificación del producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.



3.1.21.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ Fecha.
- ✓ IdProceso.
- ✓ Para.
- ✓ IdTCaja.
- ✓ Código.
- ✓ Color.
- ✓ Liquidador.
- ✓ Observaciones.
- ✓ IdTalla.
- ✓ Torre.
- ✓ CantCajas.
- ✓ Blass.
- ✓ Clasificación.

El campo Para podrá ser seleccionado de forma selectiva de una lista desplegable.

El campo IdTCaja podrá ser seleccionado por medio de direccionales, donde se mostrará al usuario una información referente a la caja con la que se desea liquidar.

Para guardar los datos ingresados en el formulario, es necesario presionar el botón guardar para ser almacenados en la base de datos.

Para actualizar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón editar, se habilitarán las cajas de textos correspondientes a dicho registro; una vez realizada la modificación se pulsará el botón guardar.

Para eliminar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón eliminar; luego aparecerá un mensaje de confirmación donde pulsaremos la opción “si” y automáticamente se eliminará el registro.

3.1.21.1.4. Salidas.

Con los datos antes mencionados se guardarán los nuevos registros con la información relativa acerca del proceso liquidación Conicsa del producto camarón entero.



3.1.22. Registrar y Actualizar Datos Liquidación Conicsa para Camarón Cola.

3.1.22.1. Especificación.

3.1.22.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a la liquidación de producto, Camarón Cola, para Conicsa.

3.1.22.1.2. Entradas.

Por Pantalla:

- ✓ **Fecha:** Es un dato obligatorio. Almacena la fecha del proceso que se está realizando.
- ✓ **IdProceso:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de proceso.
- ✓ **Para:** Es un dato obligatorio. Almacena el nombre del cliente a quien va dirigido el producto terminado.
- ✓ **IdTCaja:** Es un dato obligatorio. Almacena el id correspondiente a un tipo de caja.
- ✓ **Código:** Es un dato obligatorio. Almacena un conjunto de caracteres que identifican el proceso.
- ✓ **Color:** Es un dato obligatorio. Almacena el color de los productos, estos pueden ser: Pink o White.
- ✓ **Liquidador:** Es un dato obligatorio. Almacena el nombre de la persona encargada de la liquidación del producto.
- ✓ **Observaciones:** No es un dato obligatorio. Almacena información sobre el resultado del proceso.
- ✓ **IdTalla:** Es un dato Obligatorio. Almacena el id correspondiente referente a una talla específica del producto.
- ✓ **Torre:** Es un dato obligatorio. Almacena el número o identificador correspondiente de una torre.
- ✓ **CantCajas:** Es un dato obligatorio. Almacena la cantidad de cajas obtenidas por talla y por torre.
- ✓ **Blass:** Es un dato obligatorio. Almacena el nombre del cuarto de congelación del producto.
- ✓ **Clasificación:** Es un dato obligatorio. Almacena el nombre que identifica la clasificación del producto.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.



3.1.22.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **Fecha.**
- ✓ **IdProceso.**
- ✓ **Para.**
- ✓ **IdTCaja.**
- ✓ **Código.**
- ✓ **Color.**
- ✓ **Liquidador.**
- ✓ **Observaciones.**
- ✓ **IdTalla.**
- ✓ **Torre.**
- ✓ **CantCajas.**
- ✓ **Blass.**
- ✓ **Clasificación.**

El campo Para podrá ser seleccionado de forma selectiva de una lista desplegable.

El campo IdTCaja podrá ser seleccionado por medio de direccionales, donde se mostrará al usuario una información referente a la caja con la que se desea liquidar.

Para guardar los datos ingresados en el formulario, es necesario presionar el botón guardar para ser almacenados en la base de datos.

Para actualizar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón editar, se habilitarán las cajas de textos correspondientes a dicho registro; una vez realizada la modificación se pulsará el botón guardar.

Para eliminar los datos almacenados en la base de datos, seleccionamos el registro de un datagrid, pulsando el botón eliminar; luego aparecerá un mensaje de confirmación donde pulsaremos la opción “si” y automáticamente se eliminará el registro.

3.1.22.1.4. Salidas.

Con los datos antes mencionados se guardarán los nuevos registros con la información relativa acerca del proceso liquidación Conicsa del producto camarón Cola.

3.1.23. Registrar Master en Bodega.

3.1.23.1. Especificación.

3.1.23.1.1. Introducción.

Esta función permitirá al usuario registrar en el sistema toda la información referente a los master existentes en bodega.



3.1.23.1.2. Entradas.

Por Pantalla:

- ✓ **IdTMaster:** Es un dato obligatorio. Almacena el id correspondiente al tipo de master.
- ✓ **IdTalla:** Es un dato Obligatorio. Almacena el id correspondiente referente a una talla específica del producto.

Una vez ingresado los campos anteriores, el usuario deberá introducir la cantidad de cajas que desea obtener en master existentes en talla y tipo de master.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.23.1.3. Proceso.

Se mostrara al usuario una interfaz amigable en la cual se deberán ingresar los siguientes campos:

- ✓ **IdTMaster**
- ✓ **IdTalla**

El campo IdTMaster podrá ser seleccionado a través de direccionales, donde se mostrará al usuario una información referente al master.

El campo IdTalla podrá ser seleccionado a través de direccionales que nos permitirán obtener las tallas de las cajas que fueron liquidadas.

Una vez capturados los datos por pantalla, el usuario deberá presionar el botón “Masterisar”, para almacenar los master resultantes por la cantidad de cajas que fueron ingresadas.

3.1.23.1.4. Salidas.

Con los datos antes mencionados, se guardarán los nuevos registros con la información relativa acerca de los master completados en bodega.



3.1.24. Actualizar Master en Bodega.

3.1.24.1. Especificación.

3.1.24.1.1. Introducción.

Esta función permitirá al usuario Actualizar en el sistema toda la información referente a los master existentes en bodega.

3.1.24.1.2. Entradas.

El usuario deberá ingresar la cantidad de master que serán dado de bajas en bodega.

Por el Sistema:

- ✓ **NICK:** es un dato obligatorio que se solicita al iniciar sesión para tener acceso a la aplicación y que el sistema guarda de tal forma que se solicita una sola vez cuando se inicia sesión.
- ✓ **PASS:** es un dato obligatorio. Representa el conjunto de letras o símbolos que se relaciona con los usuarios, cada usuario tendrá una clave diferente y esta será conservada hasta que el usuario decida salir de la aplicación.

3.1.24.1.3. Proceso.

El usuario deberá seleccionar a través de las direccionales una liquidación, que nos permitirá conocer sus existencias, e ingresar la cantidad de master a dar de baja.

3.1.24.1.4. Salidas.

Una vez finalizado el proceso descrito anteriormente, se actualizarán los master existentes en bodega.

3.1.25. Registrar y Actualizar Control de Acceso de los Usuarios.

3.1.25.1. Especificación.

3.1.25.1.1. Introducción.

Esta función permitirá al usuario (con privilegios de Administrador) agregar, editar y quitar cuentas de usuarios con su tipo de acceso (permisos) y cambiar Nick y Pass de las cuentas Existentes.

3.1.25.1.2. Entradas.

Por Pantalla:

- ✓ **NOMBRECOMPLETO:** es un dato obligatorio. Almacena el nombre completo de un Usuario.
- ✓ **NICK:** es un dato obligatorio. Almacena al alias de un Usuario.
- ✓ **PASS:** es un dato obligatorio. Almacena un conjunto de caracteres que conforman la contraseña de un usuario.
- ✓ **TIPO_USUARIO:** es un dato obligatorio. Almacena el tipo de Usuario.



- ✓ **Catalogos:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a catálogos.
- ✓ **Recepción:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Recepción.
- ✓ **ControlCalidad:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a ControlCalidad.
- ✓ **PesoPlanta:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a PesoPlanta.
- ✓ **Liquidacion:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Liquidación.
- ✓ **Masterizacion:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Masterizacion.
- ✓ **Clasificación:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Clasificación.
- ✓ **Empaque:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Empaque.
- ✓ **Rastreabilidad:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Rastreabilidad.
- ✓ **MovInterno:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a MovInterno.
- ✓ **Reportes:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Reportes.
- ✓ **Configuración:** es un dato obligatorio. Almacena información lógica que indica si un usuario puede acceder a Configuración.

3.1.25.1.3. Proceso.

Se mostrará al usuario una interfaz amigable en la cual se deberán de ingresar los siguientes campos:

- ✓ **NOMBRECOMPLETO.**
- ✓ **NICK.**
- ✓ **PASS.**
- ✓ **TIPO_USUARIO.**
- ✓ **Catalogos.**
- ✓ **Recepción.**
- ✓ **ControlCalidad.**
- ✓ **PesoPlanta.**
- ✓ **Liquidacion.**
- ✓ **Masterizacion.**
- ✓ **Clasifiacion.**
- ✓ **Rastreabilidad.**
- ✓ **MovInterno.**



- ✓ **Reportes.**
- ✓ **Configuración.**

El campo TIPO_USUARIO podrá ser seleccionado de forma selectiva de una lista desplegable.

Para guardar los datos ingresados en el formulario, es necesario presionar el botón guardar para ser almacenado en la base de datos.

Para actualizar los datos introducidos nos ubicamos en el registro a modificar través de las direccionales, pulsamos el botón editar y se habilitaran las cajas de textos correspondientes a dicho registro, una vez realizada la modificación se pulsara el botón guardar.

Para eliminar los datos introducidos nos ubicamos en el registro a través de las direccionales, pulsamos el botón eliminar y luego aparecerá un mensaje de confirmación pulsamos si y se eliminara el registro.

3.1.25.1.4. Salidas.

Con los datos antes mencionados y de acuerdo a la acción correspondiente se guardarán los nuevos registros correspondientes al Control de Acceso de los Usuarios.

3.1.26. Crear Copia de Seguridad.

3.1.26.1. Especificación.

3.1.26.1.1. Introducción.

Esta función permitirá al usuario crear una copia de seguridad de la base de datos.

3.1.26.1.2. Entradas.

El usuario deberá pulsar la opción Generar Backup de la pestaña de configuración luego aparecerá una interfaz en la que deberá introducir el servidor, usuario y contraseña.

3.1.26.1.3. Proceso.

Una vez ingresados los datos antes mencionados se pulsara el botón Conectar. Luego de haber establecido la conexión se cargaran automáticamente en una lista desplegables las bases de datos pertenecientes a dicho servidor.

Una vez Seleccionada la base de datos pulsamos el botón Generar Backup.

3.1.26.1.4. Salidas.

Después de haber generado el Backup, se enviara un mensaje indicando que el Backup se ha generado satisfactoriamente.



3.1.27. Configurar Cadena de Conexión.

3.1.27.1. Especificación.

3.1.27.1.1. Introducción.

Esta función permitirá al usuario modificar en el sistema la cadena de conexión existente y permitir al usuario poder acceder a los datos desde cualquier lugar

3.1.27.1.2. Entradas.

El Usuario deberá de ingresar los campos requeridos en el formulario, como son: Instancia del Servidor o IP Pública, Nombre de Usuario y Contraseña del Servidor y Nombre de la base de datos para poder acceder a los datos.

3.1.27.1.3. Proceso.

Una vez ingresado en el formulario los nuevos valores para modificar nuestra cadena de conexión hacia la base de datos, el usuario pulsará el botón “Guardar” para realizar los cambios persistentes.

3.1.27.1.4. Salidas.

Una vez finalizado el proceso descrito anteriormente, se mostrará un mensaje de afirmación que pide que el usuario reinicie el sistema y se guardarán los cambios en la nueva cadena de conexión realizada anteriormente.

3.1.28. Informe de Recepción de Producto.

3.1.28.1. Especificación.

3.1.28.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes a la Recepción del Producto ya sea en forma diaria o por periodo.

3.1.28.1.2. Entradas.

Datos para generar los informes:

Informe por Periodo.

Por Pantalla.

- Fecha Origen.
- Fecha Destino.

Informe por Fecha.

Por Pantalla.

- Fecha.

Informe Proveedor, Granja y Piscina.

Por Pantalla.

- Proveedor.
- Granja.
- Piscina.

Informe por Fecha y Contenedor.

Por pantalla.

- Fecha.
- Contenedor.



Datos proporcionados por el sistema para dichos informes:

- Proveedor.
- Granja.
- Piscina.
- Ciclo Productivo
- Fecha.
- Conductor.
- Vehículo Placa.
- Vehículo Marca.
- Numero Bin.
- Numero Entrega.
- Peso
- Unidad Medida.
- Naviera.
- Contenedor.
- Barco.
- Proceso.
- Color.
- Cajillas.
- Tara.
- Total.

3.1.28.1.3. Proceso.

Se presentara la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.

3.1.28.1.4. Salidas.

Con todos los datos antes mencionados se generara el informe correspondiente a la Recepción de Producto.

3.1.29. Informe de Rendimiento de Camarón Entero.

3.1.29.1. Especificación.

3.1.29.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes al Rendimiento de Camarón Entero en forma diaria.

3.1.29.1.2. Entradas.

Datos para generar los informes:

Informe Por Fecha y Piscina.

Por Pantalla.

- Fecha.
- Piscina.

Datos proporcionados por el sistema para dichos informes:

- Fecha.
- Piscina.



- Color.
- Empaque.
- Peso Caja.
- Unidad Medida.
- Peso Granja.
- Peso Planta.
- Entero Empacado.
- Gramo Promedio.
- Clientes.
- Tallas.
- Cajas.
- Libras.
- Porcentaje.
- Sobrepeso.
- Producto Desc.

3.1.29.1.3. Proceso.

Se presentara la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.

3.1.29.1.4. Salidas.

Con todos los datos antes mencionados se generara el informe correspondiente al Rendimiento de Camarón Entero.

3.1.30. Informe de Control de Calidad de Producto.

3.1.30.1. Especificación.

3.1.30.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes al Proceso de Control de Calidad del Producto.

3.1.30.1.2. Entradas.

Datos para generar los informes:

Informe Por Periodo + Proveedor

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Proveedor

Informe Por Periodo + Proveedor + Piscina

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Proveedor
- Piscina

Informe Por Fecha + Proveedor



Por Pantalla:

- Fecha
- Proveedor

Informe Por Periodo + Contenedor

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Contenedor

Datos proporcionados por el sistema para dichos informes:

- Fecha.
- Proceso
- TotalLbsReport
- Codigo
- SupCalidad
- Observaciones
- Produccion
- ControlCalidad
- SupProducto
- Piscina
- NumeroBin
- Temperatura
- NumCamKG
- Conteo2KG
- MateriaExtrana
- VANN
- LARVA
- STYL
- Color
- PesoMuestra
- Defecto
- Peso
- Porcentaje

3.1.30.1.3. Proceso.

Se presentara la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.

3.1.30.1.4. Salidas.

Con todos los datos antes mencionados se generara el informe correspondiente Proceso de Control de Calidad.



3.1.31. Informe de Liquidación del Producto.

3.1.31.1. Especificación.

3.1.31.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes al Proceso de Liquidación del Producto.

3.1.31.1.2. Entradas.

Datos para generar los informes:

Informe Por Periodo + Proveedor

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Proveedor

Informe Por Periodo + Proveedor + Piscina

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Proveedor
- Piscina

Informe Por Fecha + Proveedor

Por Pantalla:

- Fecha
- Proveedor

Informe Por Periodo + Contenedor

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Contenedor

Datos proporcionados por el sistema para dichos informes:

- Fecha.
- Proceso
- Para
- IdTCaja
- Codigo
- Color
- Produccion
- Observaciones
- Torre
- CantCajas
- Blass
- Clasificación



3.1.31.1.3. Proceso.

Se presentara la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.

3.1.31.1.4. Salidas.

Con todos los datos antes mencionados se generara el informe correspondiente Proceso de Liquidación del producto.

3.1.32. Informe de Peso Bruto en Planta.

3.1.32.1. Especificación.

3.1.32.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes al Proceso de Peso Bruto en Planta.

3.1.32.1.2. Entradas.

Datos para generar los informes:

Informe Por Registro Actual

Por Pantalla:

- IdRecep

Informe por Entero

Por Pantalla:

- Entero

Informe por Cola

Por Pantalla:

- Cola

Informe por Fecha + Contenedor + Barco

Por Pantalla:

- Fecha
- Contenedor
- Barco

Informe por Fecha + Piscina + Proveedor

Por Pantalla:

- Fecha
- Piscina
- Proveedor

Informe por Piscina + Proveedor

Por Pantalla:

- Piscina
- Proveedor



Informe por Proveedor

Por Pantalla:

- Proveedor

Informe por Periodo

Por Pantalla:

- Fecha Inicio
- Fecha Final

Informe por Periodo + Piscina + Proveedor

Por Pantalla:

- Fecha Inicio
- Fecha Final
- Piscina
- Proveedor

Datos proporcionados por el sistema para dichos informes:

- Fecha
- Nombre_Comercial
- Codigo
- NumPiscina
- PesoGranja
- PesoBruto
- MateriaExtraña
- PesoBascula
- Merma
- PesoNeto
- NumeroBines
- Observaciones
- LbsEntero
- LbsShellOn
- Origen
- NumeroCajillas
- NumContenedor
- Barco
- Proceso
- SupGranja
- GteProducción

3.1.32.1.3. Proceso.

Se presentará la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.



3.1.32.1.4. Salidas.

Con todos los datos antes mencionados se generará el informe correspondiente al Proceso de Peso Bruto en Planta.

3.1.33. Informe de Libras Empacadas por Área.

3.1.33.1. Especificación.

3.1.33.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes al informe de Libras Empacadas por Área.

3.1.33.1.2. Entradas.

Datos para generar los informes:

Informe por Fecha

Por Pantalla:

- Fecha

Datos proporcionados por el sistema para dichos informes:

- Fecha
- Libras Entero
- Libras ShellOn
- Libras PUD
- Libras P&D
- Total de Libras

3.1.33.1.3. Proceso.

Se presentará la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.

3.1.33.1.4. Salidas.

Con todos los datos antes mencionados se generará el informe correspondiente al informe de Libras Empacadas por Área.

3.1.34. Informe de Control de los Bines.

3.1.34.1. Especificación.

3.1.34.1.1. Introducción.

Esta función permitirá al usuario visualizar de forma detallada los datos correspondientes al informe de Control de los Bines.

3.1.34.1.2. Entradas.

Datos para generar los informes:

Informe por Entradas o Salidas

Por Pantalla:



- Entrada
- Salida

Informe por Proveedor + Entrada/Salida
Por Pantalla:

- Proveedor
- Entrada o Salida

Informe por Fecha + Entrada/Salida
Por Pantalla:

- Fecha
- Entrada o Salida

Informe por Periodo + Entrada/Salida
Por Pantalla:

- Fecha Inicio
- Fecha Final
- Entrada o Salida

Datos proporcionados por el sistema para dichos informes:

- Fecha
- Nombre_Comercial
- Vehículo
- E_S
- NumeroBin
- Propiedad
- CantHielo
- UnidadMedida

3.1.34.1.3. Proceso.

Se presentará la pantalla de introducción de datos al usuario.

Los datos a introducir son de carácter obligatorio, para generar dicho informe.

3.1.34.1.4. Salidas.

Con todos los datos antes mencionados se generará el informe correspondiente al informe de Control de los Bines.

3.2. Requisitos de Funcionamiento.

Requisitos Estáticos.

No existe ninguna restricción sobre el número de terminales y de usuarios que estén trabajando simultáneamente con la aplicación.



Requisitos Dinámicos.

Es importante que el tiempo de respuesta no aumente exponencialmente con el número de usuarios.

3.3. Restricciones de Diseño.

El sistema estará diseñado bajo software libre.

El sistema se diseñará según un modelo Cliente/Servidor.

La aplicación no tiene ninguna capacidad para la toma de decisiones.

El sistema no podrá ser accedido por entes externos.

Deberá tener un diseño e implementación sencillos independientes de la plataforma o el lenguaje de programación.

3.4. Atributos.

3.4.1. Seguridad.

El sistema al inicio ejecutara una interfaz de usuario donde pedirá el Usuario y la Contraseña de acceso para poder acceder a los distintos tipos de niveles. Así como garantizara la integridad, disponibilidad y confiabilidad de los datos.

3.4.2. Mantenimiento.

El mantenimiento a la aplicación se podrá realizar dado que durante su desarrollo se utilizara la programación estructurada y orientada a objeto. Cualquier modificación que se haga en la especificación de requisitos será solamente en el módulo que corresponda a ese requisito.

3.4.3. Ayuda en Línea.

Esta aplicación no constara con ayuda en línea debido a que se capacitara a los usuarios finales, además que podrán contar con este documento.

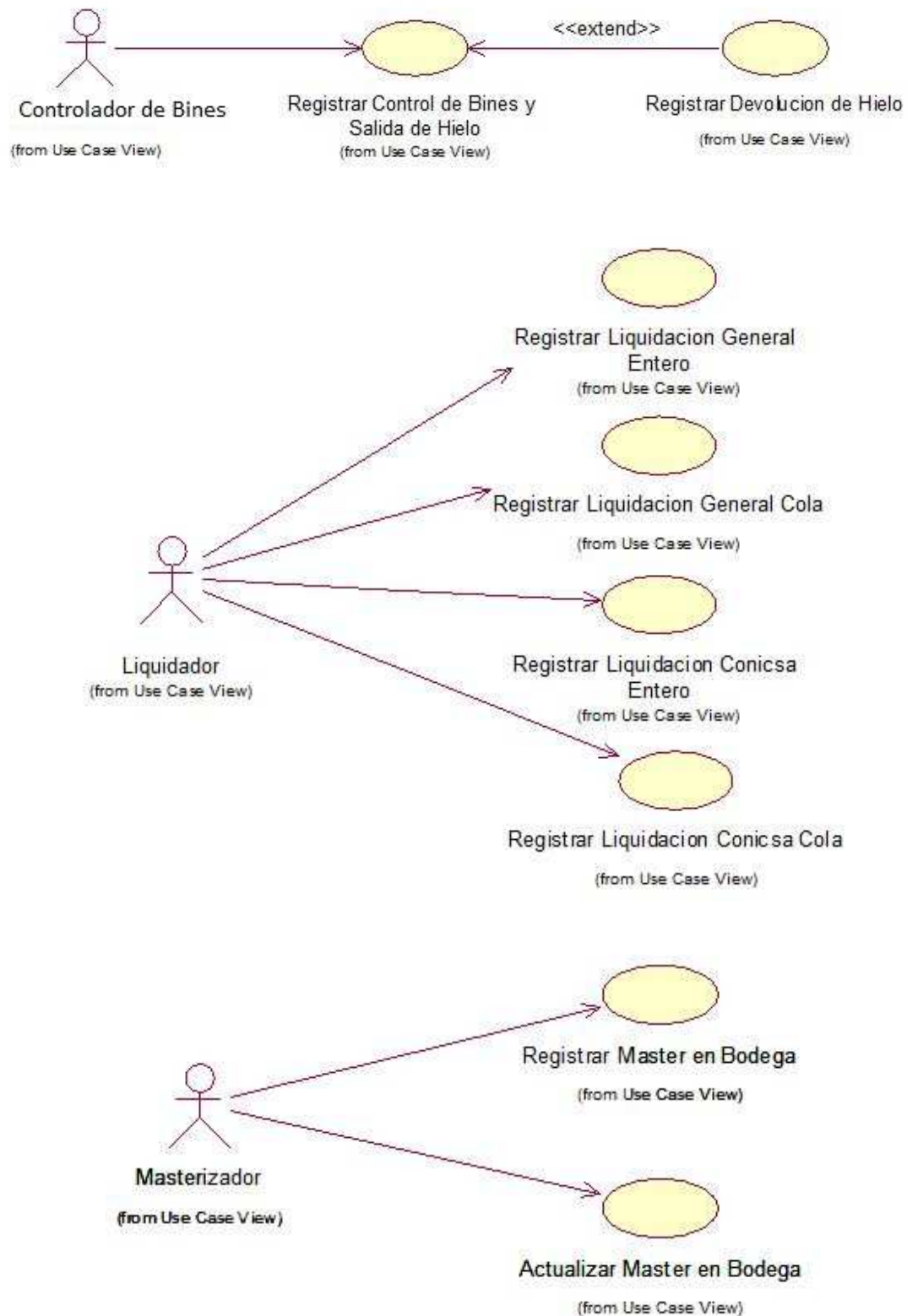
3.5. Otros Requisitos.

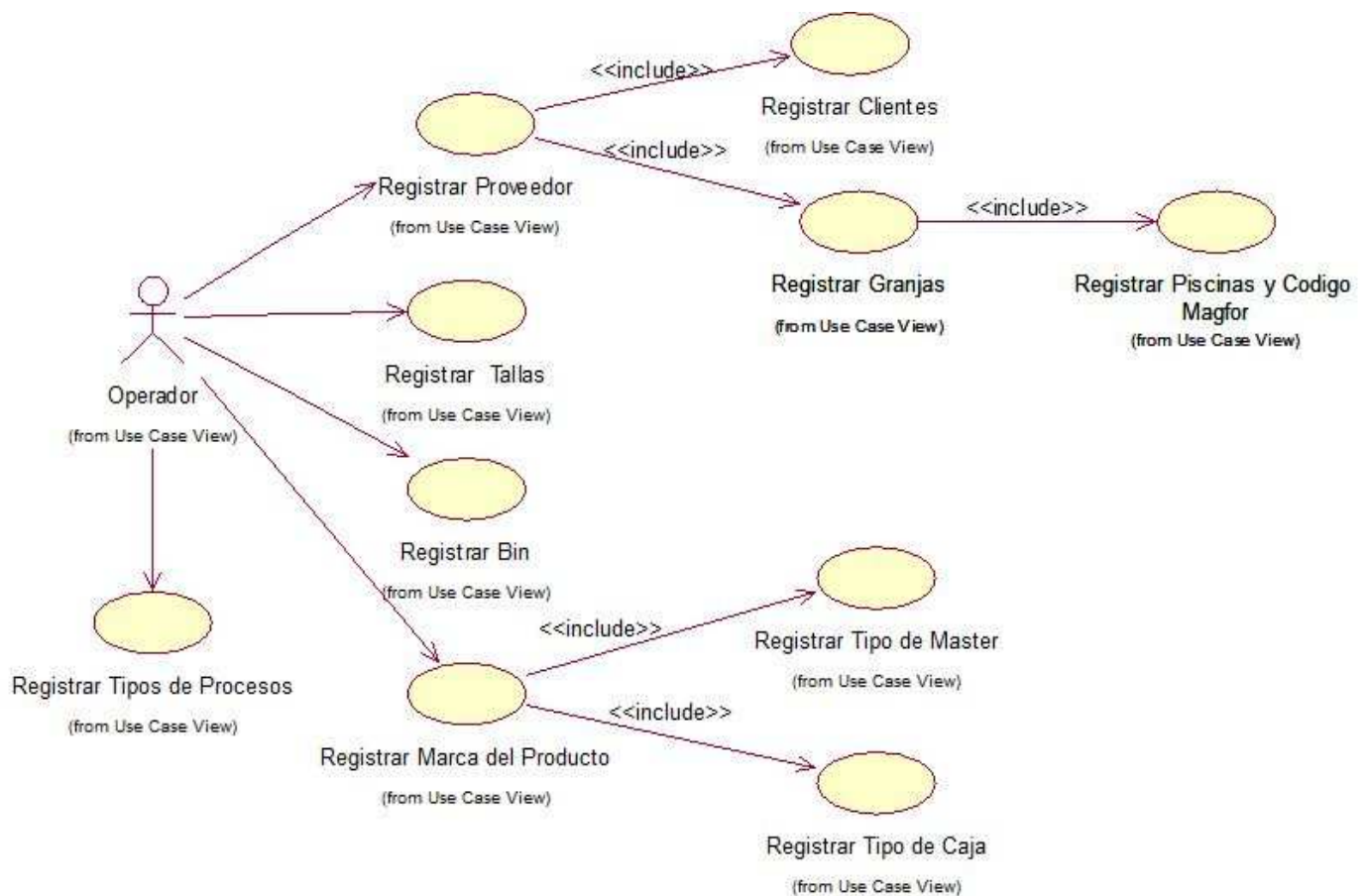
3.5.1. Base de Datos.

Se utilizara una base de datos relacional para el almacenamiento de la información.



DIAGRAMA DE CASOS DE USO





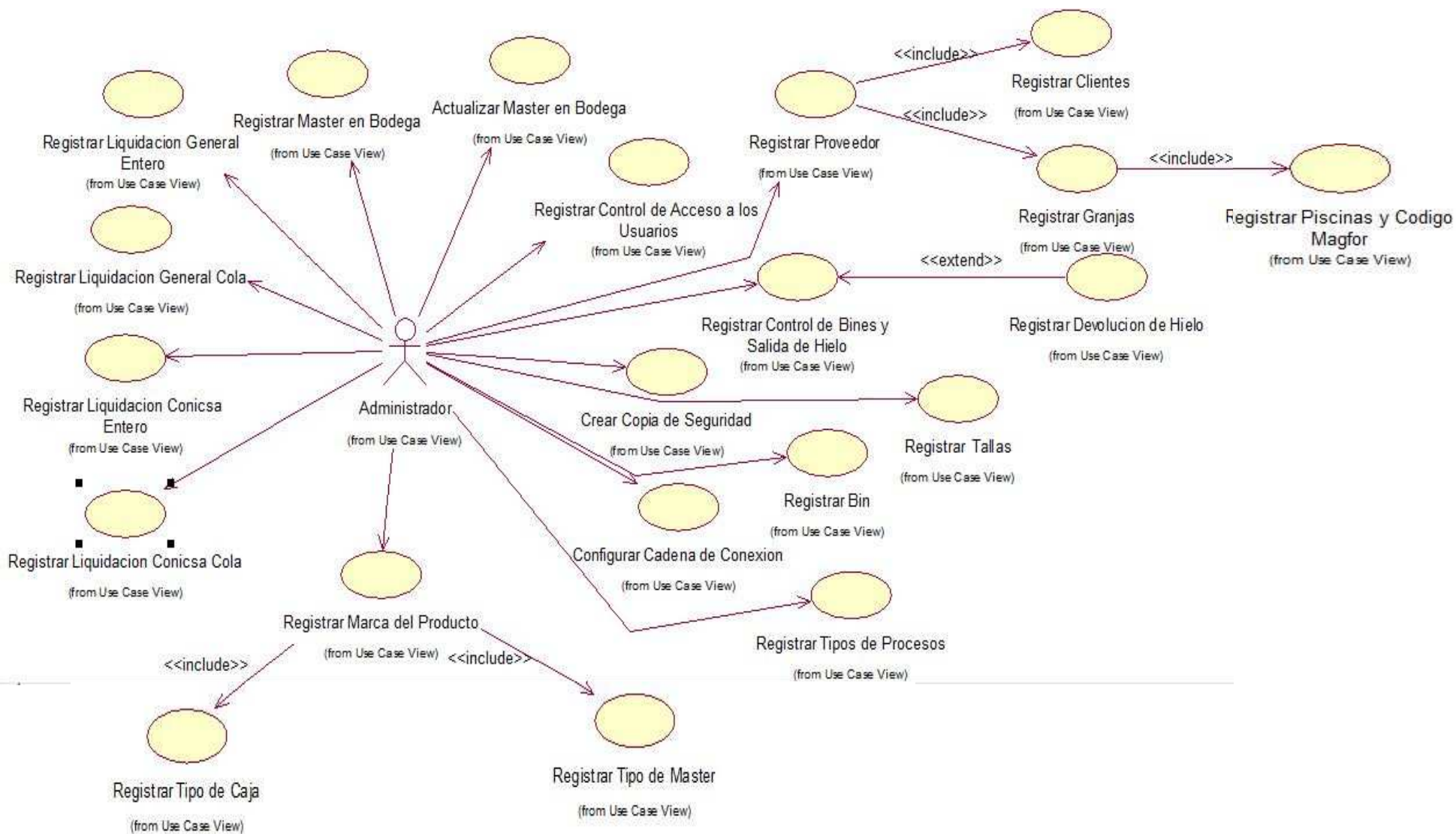




DIAGRAMA DE CLASES

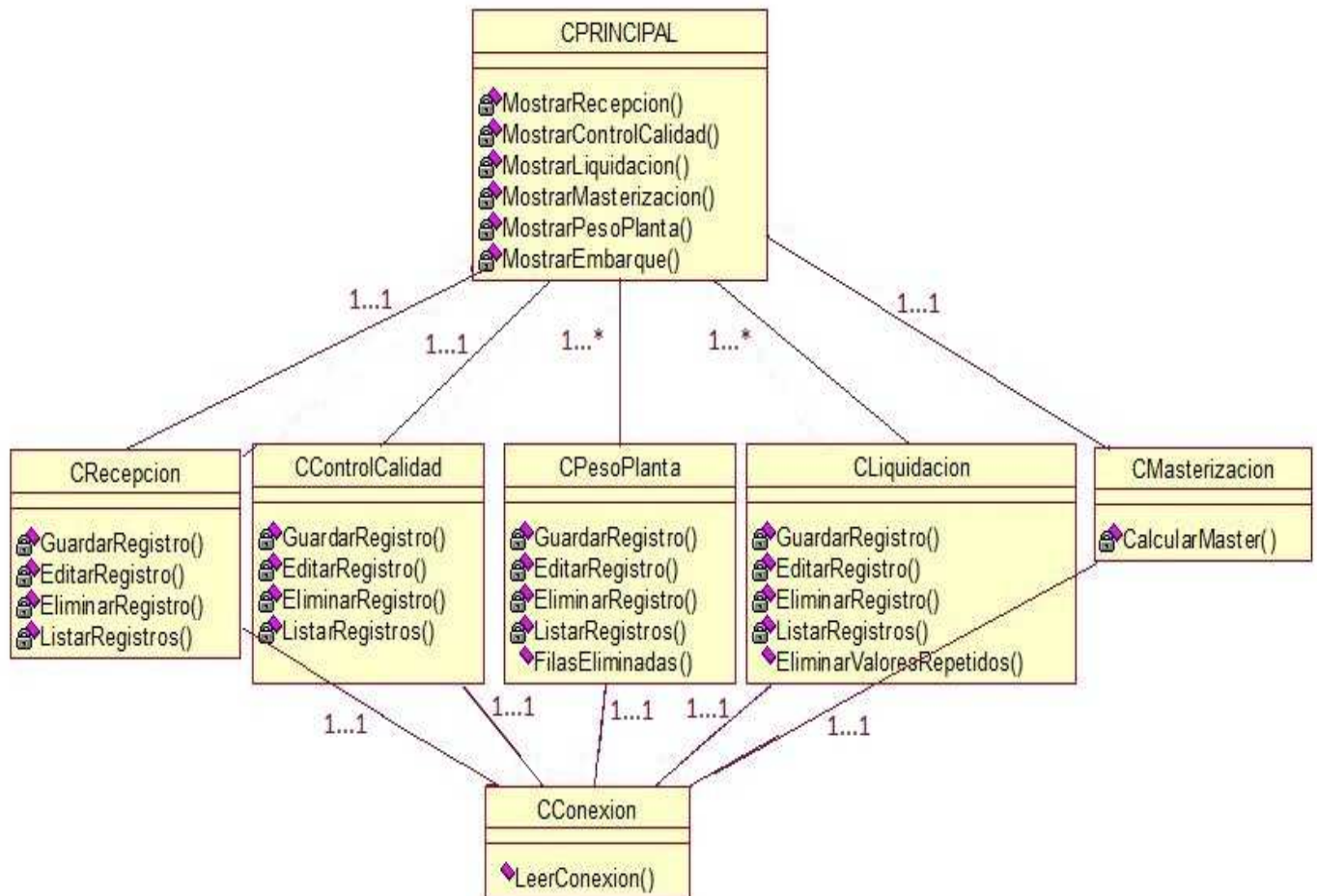


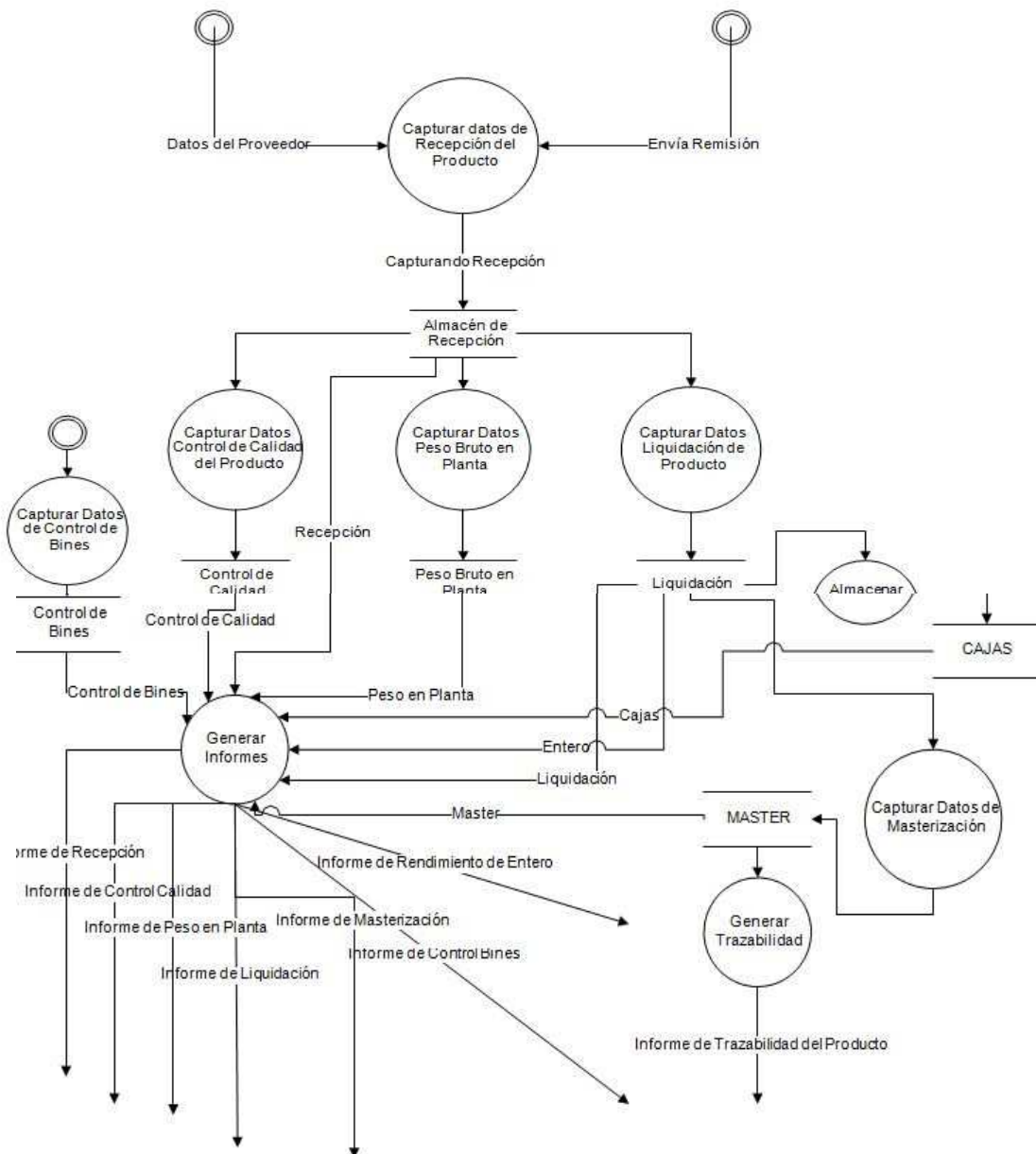


DIAGRAMA DE FLUJO DE DATOS



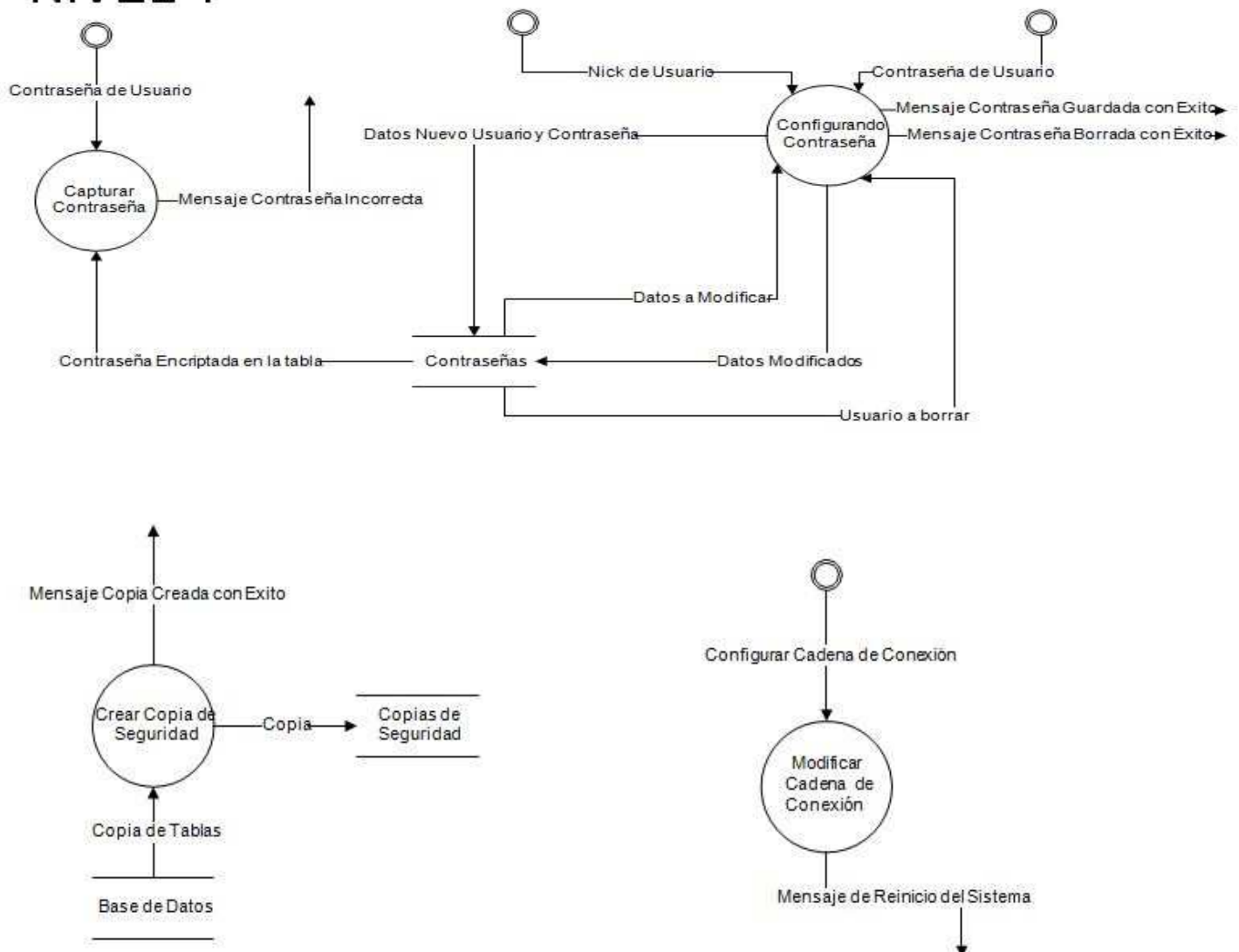


NIVEL 1





NIVEL 1





NIVEL 2

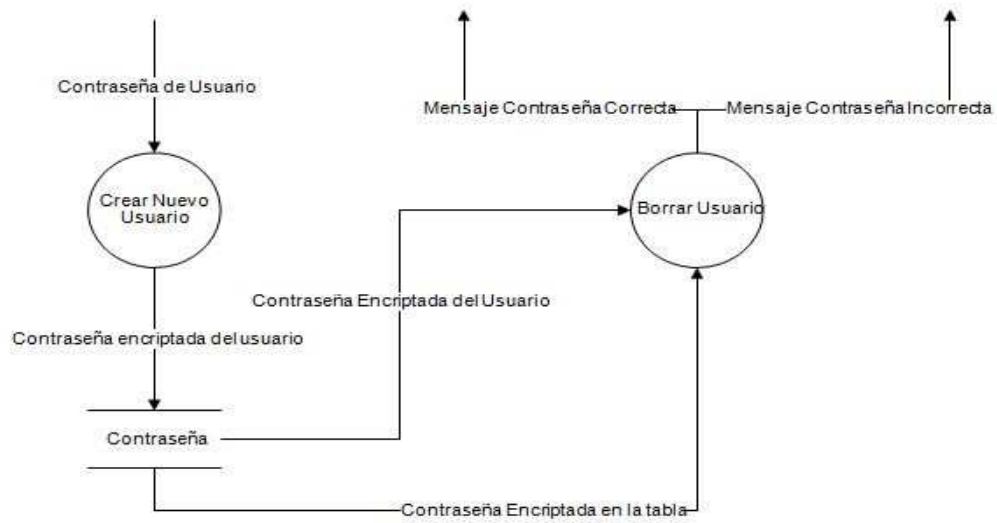
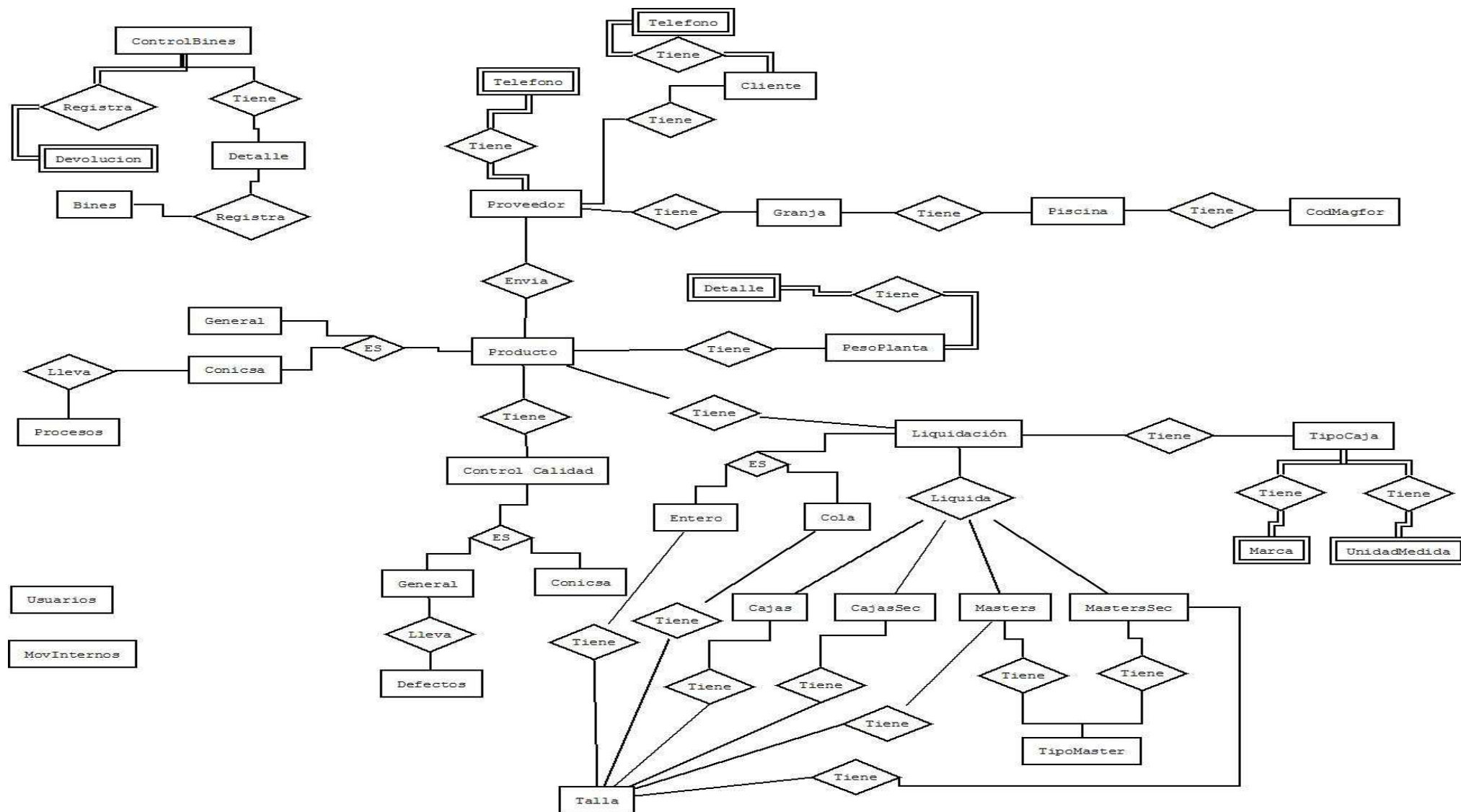




DIAGRAMA ENTIDAD-RELACION





DISEÑO DE DATOS ESTRUCTURA DE TABLAS DE LA BASE DE DATOS

1) Tabla Bines

NOMBRE CAMPO	TIPO DE DATO
Fecha	Datetime
NumeroBin	Varchar(20)
Descripcion	Varchar(500)

2) Tabla Cajas

NOMBRE CAMPO	TIPO DE DATO
IdCaja	Uniqueidentifier
IdLiq	UniqueIdentifier
Fecha	Datetime
IdTalla	Int
Exist	Int

3) Tabla CajasSec

NOMBRE CAMPO	TIPO DE DATO
IdCaja	Uniqueidentifier
IdLiq	UniqueIdentifier
Fecha	Datetime
IdTalla	Int
Exist	Int

4) Tabla CC_Conicsa

NOMBRE CAMPO	TIPO DE DATO
Id	Uniqueidentifier
IdControlCalidad	Uniqueidentifier
Color	Varchar(20)
PesoMuestra	Float
Defecto	Varchar(50)
Peso	Float
Porcentaje	Float

5) Tabla CC_General

NOMBRE COMPLETO	TIPO DE DATO
Id	Uniqueidentifier
IdControlCalidad	Uniqueidentifier
NumeroBin	Varchar(20)
Temperatura	Float
NumCamKG	Int
Conteo2KG	Int



MateriaExtrana	Int
VANN	Int
LARVA	Int
STYL	Int

6) Tabla Cliente

NOMBRE COMPLETO	TIPO DE DATO
IdCliente	Uniqueidentifier
Ruc	Varchar(20)
Nombre	Varchar(100)
Dirección	Varchar(100)
Contacto	Varchar(50)

7) Tabla CodigosMagfor

NOMBRE COMPLETO	TIPO DE DATO
IdMagfor	Int
Codigo_Magfor	Varchar(50)

8) Tabla ConicsaRecep

NOMBRE COMPLETO	TIPO DE DATO
IdConicsa	Uniqueidentifier
IdRecep	Uniqueidentifier
Barco	Varchar(20)
IdProceso	Int
Peso	Float
UnidadMedida	Varchar(50)
NCajillas	Int
Color	Varchar(15)
Barco_Liquidado	Int

9) Tabla ControlBines

NOMBRE COMPLETO	TIPO DE DATO
IdControlBines	Uniqueidentifier
Fecha	Datetime
Ruc	Varchar(20)
Vehiculo	Varchar(50)
E_S	Varchar(20)

10) Tabla ControlCalidad

NOMBRE COMPLETO	TIPO DE DATO
IdControlCalidad	Uniqueidentifier
IdConicsa_Barco	Uniqueidentifier



IdRecep	Uniqueidentifier
TipoCC	Varchar(50)
Fecha	Date
IdProceso	Int
TotalLbsReport	Float
Codigo	Varchar(50)
SupCalidad	Varchar(100)
Observaciones	Varchar(500)
Produccion	Varchar(100)
ControlCalidad	Varchar(100)
SupProducto	Varchar(100)

11) Tabla Defectos

NOMBRE COMPLETO	TIPO DE DATO
Id	Int
IdCC_General	Uniqueidentifier
Defecto	Varchar(50)
Cantidad	Int
Porcentaje	Float

12) Tabla DetalleControlBines

NOMBRE COMPLETO	TIPO DE DATO
Id	Uniqueidentifier
IdControlBines	Uniqueidentifier
NumeroBin	Varchar(20)
Propiedad	Varchar(100)
CantHielo	Int
IdUM	Int

13) Tabla DetallePesoPlanta

NOMBRE COMPLETO	TIPO DE DATO
Id	Uniqueidentifier
IdPesoPlanta	Uniqueidentifier
Gaveta	Int
TipoPeso	Varchar(50)
Peso	Float
UndMedida	Varchar(50)

14) Tabla DevHielo

NOMBRE COMPLETO	TIPO DE DATO
IdDV	Uniqueidentifier
IdControlBines	Uniqueidentifier
Fecha	Datetime



Nombre	Varchar(200)
Empresa	Varchar(100)
CantDev	Decimal(10,2)
IdUM	Int
Placa	Varchar(50)

15) Tabla GeneralRecep

NOMBRE COMPLETO	TIPO DE DATOS
IdGeneral	Uniqueidentifier
IdRecep	Uniqueidentifier
NumeroBin	Varchar(20)
Peso	Decimal(18,2)
UnidadMedida	Varchar(50)
Num_Entrega	Varchar(50)

16) Tabla Granja

NOMBRE COMPLETO	TIPO DE DATOS
CodGranja	Char(10)
Ruc	Varchar(20)
Nombre	Varchar(50)

17) Tabla LiqCola

NOMBRE COMPLETO	TIPO DE DATOS
Id	Uniqueidentifier
IdLiq	Uniqueidentifier
IdTalla	Int
Torre	Varchar(10)
CantCajas	Int
Blass	Varchar(10)
Clasificacion	Char(5)

18) Tabla LiqEntero

NOMBRE COMPLETO	TIPO DE DATOS
Id	Uniqueidentifier
IdLiq	Uniqueidentifier
IdTalla	Int
Torre	Varchar(10)
CantCajas	Int
Blass	Varchar(10)
Clasificacion	Char(5)



19) Tabla Liquidación

NOMBRE COMPLETO	TIPO DE DATOS
IdLiq	Uniqueidentifier
IdRecep	Uniqueidentifier
IdConicsa_Barco	Uniqueidentifier
TipoL	Varchar(7)
IdProceso	Int
Fecha	Datetime
Para	Varchar(100)
IdTCaja	Int
Codigo	Varchar(100)
Color	Varchar(100)
Produccion	Varchar(100)
Observaciones	Varchar(250)

20) Tabla Marca

NOMBRE COMPLETO	TIPO DE DATOS
IdMarca	Int
Marca	Varchar(50)

21) Tabla Masters

NOMBRE COMPLETO	TIPO DE DATOS
IdMaster	Uniqueidentifier
IdLiq	Uniqueidentifier
IdTalla	Int
IdTMaster	Int
Exist	Int

22) Tabla MastersSec

NOMBRE COMPLETO	TIPO DE DATOS
IdMaster	Uniqueidentifier
IdLiq	Uniqueidentifier
IdTalla	Int
IdTMaster	Int
Exist	Int

23) Tabla MovInternos

NOMBRE COMPLETO	TIPO DE DATOS
Id	Uniqueidentifier
IdBodega	Uniqueidentifier
Fecha	Datetime
Origen	Varchar(50)



Destino	Varchar(50)
TipoMov	Varchar(10)
Cantidad	Int
Descripcion	Varchar(200)
RazonSoc	Varchar(200)
E_S	Int
EntregadoPor	Varchar(100)
RecibidoPor	Varchar(100)

24) Tabla PesoPlanta

NOMBRE COMPLETO	TIPO DE DATOS
IdPesoPlanta	Uniqueidentifier
IdRecep	Uniqueidentifier
TipoPP	Varchar(100)
Fecha	Datetime
Codigo	Varchar(20)
Origen	Varchar(50)
SupRecep	Varchar(100)
SupGranja	Varchar(100)
GteProduccion	Varchar(100)
Observaciones	Varchar(500)
PNConicsa	Float
CajProc	Int
Proceso	Varchar(20)
Barco	Int
LbsEntero	Decimal(18,2)

25) Tabla Piscina

NOMBRE COMPLETO	TIPO DE DATOS
CodGranja	Char(10)
Numero	Varchar(50)
Descripcion	Varchar(500)
IdMagfor	Int

26) Producto

NOMBRE COMPLETO	TIPO DE DATOS
IdRecep	Uniqueidentifier
Ruc	Varchar(20)
TipoR	Varchar(100)
Fecha	Datetime
NumPiscina	Varchar(50)
CicloP	Varchar(50)
Granja	Varchar(50)
SupRecepcion	Varchar(100)



SupGranja	Varchar(100)
Conductor	Varchar(100)
VehPlaca	Varchar(20)
VehMarca	Varchar(20)
Observaciones	Varchar(500)
NumContenedor	Varchar(50)
Naviera	Varchar(100)
Producto_Liquidado	Int
Producto_Pesado	Int

27) Tabla Proveedor

NOMBRE COMPLETO	TIPO DE DATOS
Ruc	Varchar(20)
Nombre	Varchar(100)
Nombre_Comercial	Varchar(100)
Codigo_Planta	Char(10)
Departamento	Varchar(50)
Dirección	Varchar(100)
Contacto	Varchar(50)
Email	Varchar(100)
Estado	Varchar(10)
SobrePeso	Decimal(18,2)

28) Tabla Tallas

NOMBRE COMPLETO	TIPO DE DATOS
IdTalla	Int
Nombre	Varchar(20)
Camaron	Varchar(50)
IdProceso	Int

29) Tabla TipoCaja

NOMBRE COMPLETO	TIPO DE DATOS
IdTCaja	Int
IdMarca	Int
Peso	Decimal(10,2)
IdUM	Int

30) Tabla TipoMasters

NOMBRE COMPLETO	TIPO DE DATOS
IdTMaster	Int
Descrip	Varchar(500)
Peso	Float
IdUM	Int



Marca	Varchar(100)
Capacidad	Int

31) Tabla TiposProceso

NOMBRE COMPLETO	TIPO DE DATOS
IdProceso	Int
Proceso	Varchar(50)

32) Tabla TlfnoClientes

NOMBRE COMPLETO	TIPO DE DATOS
IdTelfC	Int
IdCliente	Uniqueidentifier
Numero	Varchar(10)
Descripcion	Varchar(100)

33) Tabla TlfnoProveedores

NOMBRE COMPLETO	TIPO DE DATOS
IdTelfP	Int
Ruc	Varchar(20)
Numero	Varchar(10)
Descripcion	Varchar(100)

34) Tabla UnidaMedida

NOMBRE COMPLETO	TIPO DE DATOS
IdUM	Int
Nombre	Varchar(50)

35) Tabla Usuarios

NOMBRE COMPLETO	TIPO DE DATOS
ID	Int
NOMBRECOMPLETO	Varchar(100)
NICK	Varchar(15)
PASS	Varchar(15)
TIPO_USUARIO	Varchar(50)
LOG_ON	Int
Catalogos	Int
Recepción	Int
ControlCalidad	Int
PesoPlanta	Int
Liquidacion	Int
Masterizacion	Int
Clasificacion	Int

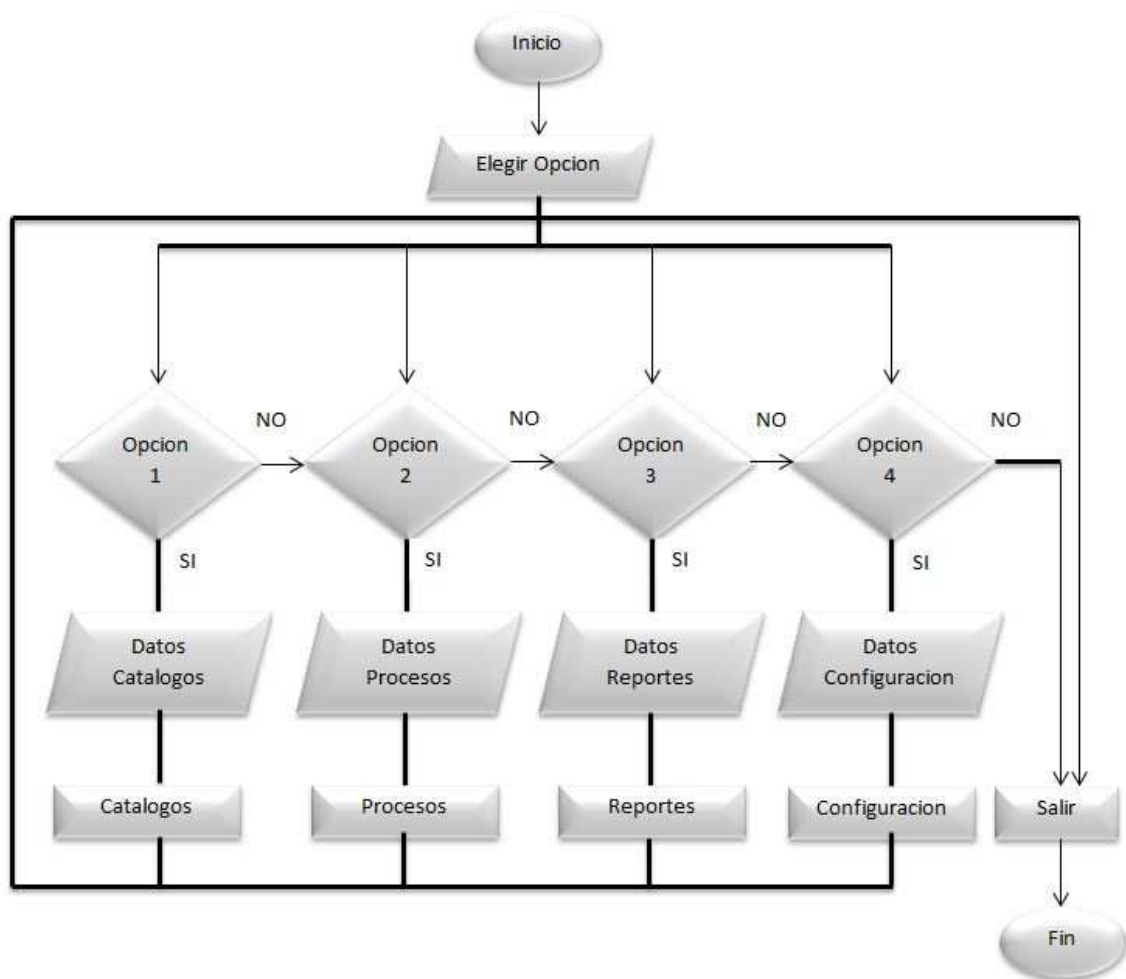


Sistema de Control Automatizado del Procesamiento del Camarón para la Empresa
Sahlman Seafoods, S.A.

Empaque	Int
Rastreabilidad	Int
MovInterno	Int
Reportes	Int



DISEÑO PROCEDIMENTAL

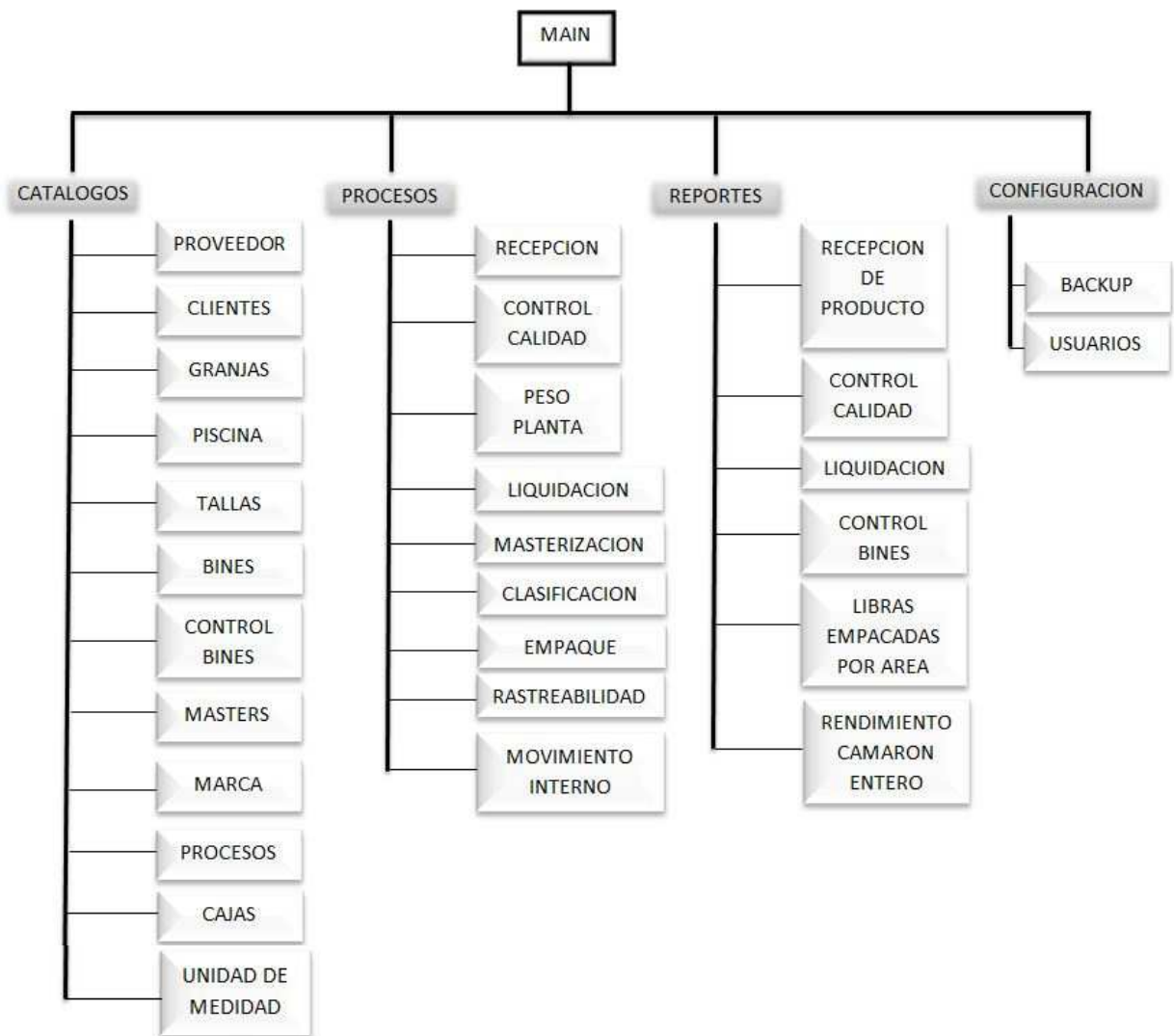




DISEÑO ARQUITECTONICO

El diseño arquitectónico, se podría definir como un conjunto de todas las vistas, donde cada vista es una perspectiva diferente del sistema.

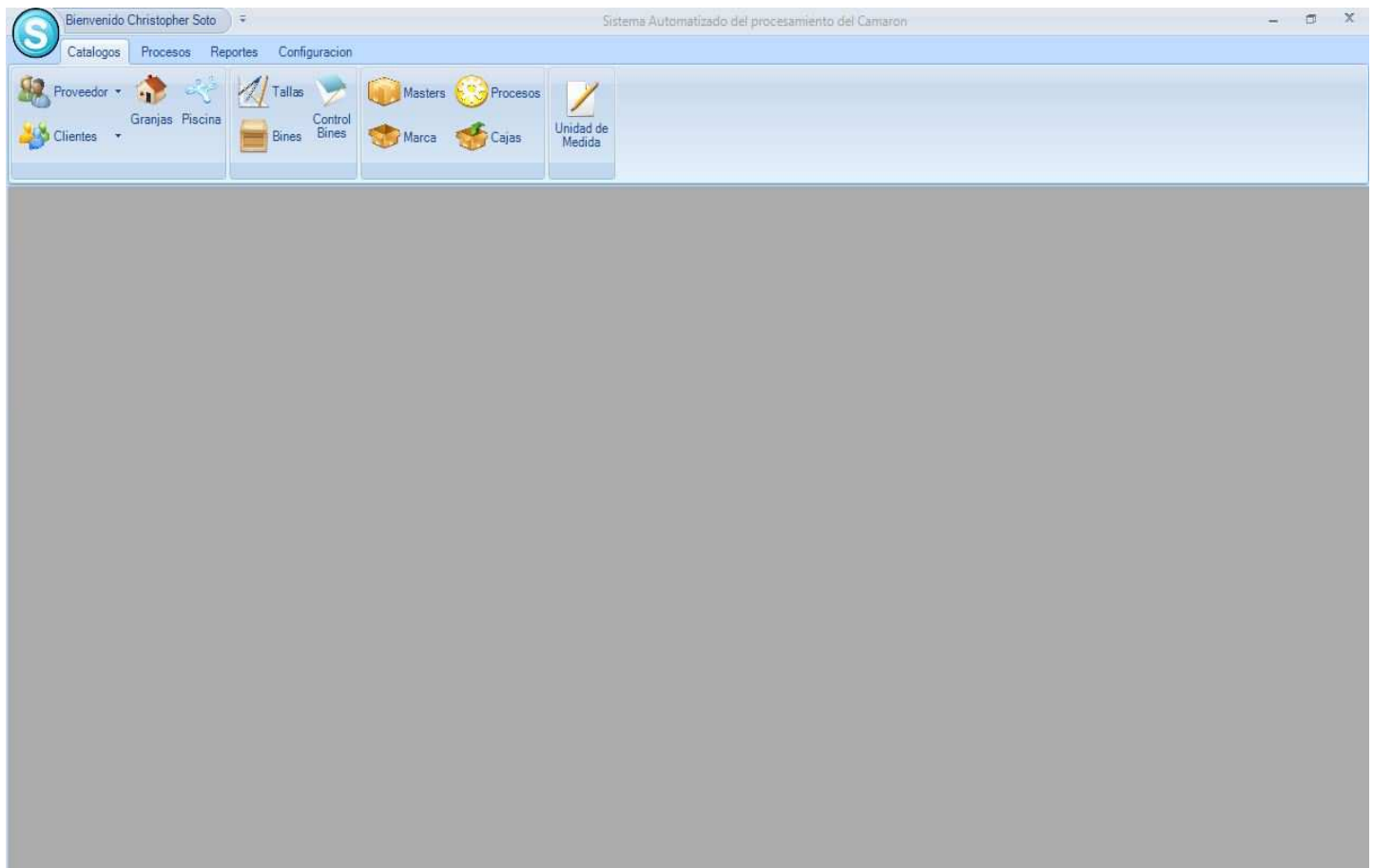
Este diseño ayuda a los desarrolladores a tener una visión común del sistema, al mismo tiempo de dividir el proyecto en clases y facilitar su reutilización.





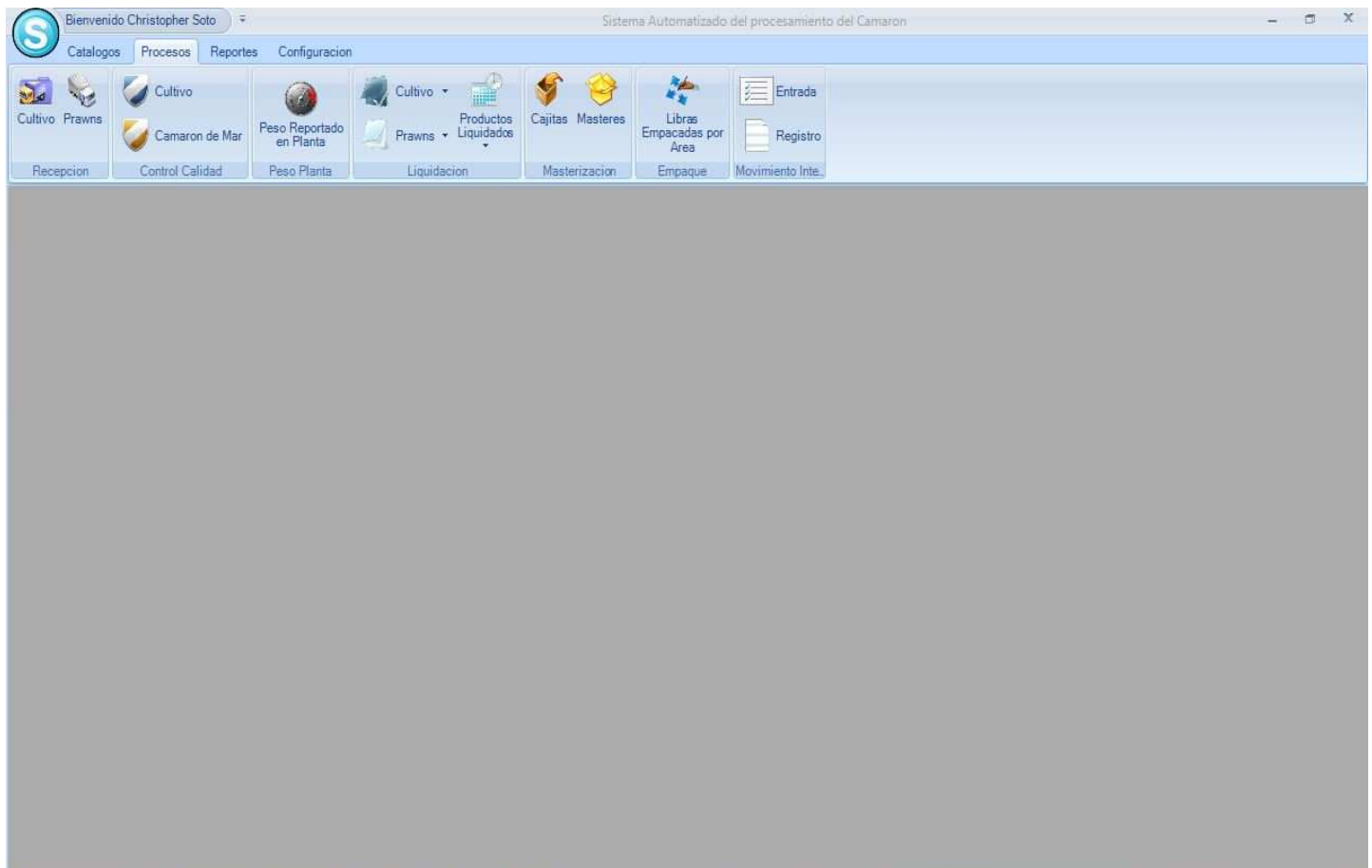
INTERFACES

Menú Principal - Catálogos



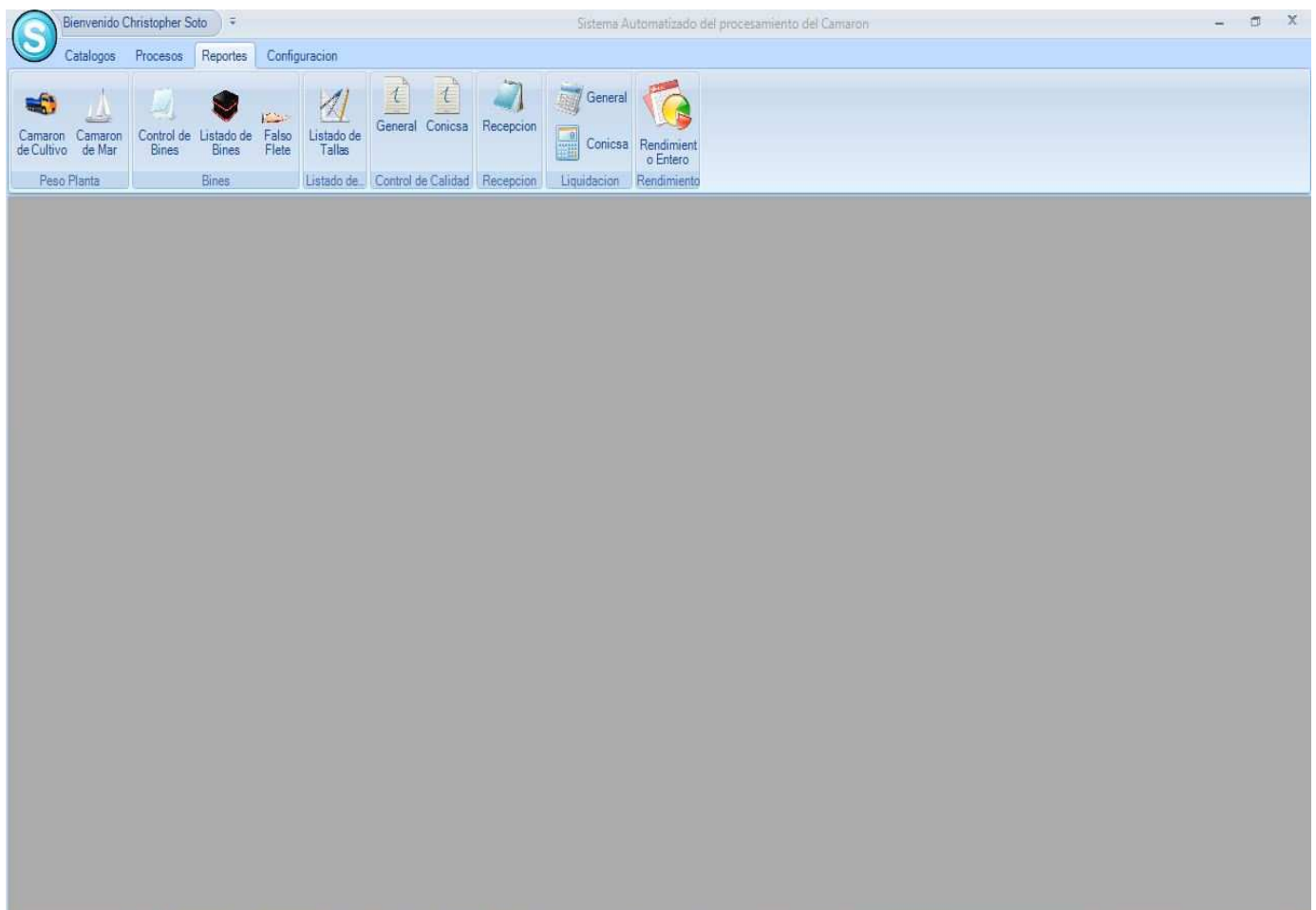


Menú Principal – Procesos





Menú Principal – Reportes





Proveedores

Mantenimiento a Proveedores

Nuevo Registro Guardar Eliminar Cancelar Salir

Ruc : 230296-9518 Contacto : Julio Juarez

Nombre : Sahlman Seafoods of Nicarag E-Mail : jjuares@sahlmanseafood.com.ni

Nombre Comercial : Sahlman Seafoods Estado : Activo

Codigo Planta : C Departamento : Chinandega

SobrePeso : 2.50 Direccion : Km 138 Carretera Chinandega

Listado de Proveedores

Ruc	Nombre	Nombre_Comercial	Codigo_Planta	Departamento
213675	seafoods internas...	seafoods	R	Chinandega
230296-9518	Sahlman Seafoo...	Sahlman Seafoods	C	Chinandega
978065	Conic SA	Conicsa	H	Managua

Granjas

Mantenimiento a Granjas

Nuevo Registro Guardar Eliminar Cancelar Salir

Codigo Granja : AR

Proveedor : seafoods internacional

Nombre : Acuicola Real

Listado de Granjas

CodGranja	Ruc	Nombre
AR	213675	Acuicola Real
AV	213675	acuicola la virge
MA	230296-9518	Mangles Altos
SH	978065	Conicsa



Sistema de Control Automatizado del Procesamiento del Camarón para la Empresa Sahlman Seafoods, S.A.

Producto

Recepcion General de Producto

Nuevo Registro Guardar Editar Eliminar Cancelar Salir visualizar Detalle

Fecha: jueves , 25 de noviembre de 2010 Piscina: 31 Vehivulo Placa: VARIOS

Proveedor: Sahlman Seafoods of Nicaragua SA Ciclo Productivo: 2 Vehiculo Marca: VARIOS

Granja: Mangles Altos Conductor: VARIOS Supervisor Recepcion: VICTOR DIAZ

Observaciones: VARIOS Supervisor Granja: MANUEL GONZALEZ

Listado de Recepcion General

IdRecep	TipoR	Fecha	Ruc	Granja	NumPiscina	CicloP	Conductor	Veh
475d71d1-ae4f-4...	GENERAL	25/11/2010 02:0...	230296-9518	Mangles Altos	31		VARIOS	VARIOS

Detalle Producto

Detalle Adjunto de Recepcion de Producto

Nuevo Registro Guardar Eliminar Cancelar Salir

Numero Bin : 0037 Ingresar Peso de Bin Ingresar Numero de Entrega

Detalle Adjunto de Recepcion de Producto

IdGeneral	IdRecep	NumeroBin	Peso	UnidadMedida
3fa19d724570-4...	475d71d1-ae4f-4...	0037	785.00	Libras
40a11905-d201-...	475d71d1-ae4f-4...	0065	785.00	Libras
b620a1be-2ff-41...	475d71d1-ae4f-4...	0132	785.00	Libras
dacb76c4-7702-...	475d71d1-ae4f-4...	0095	785.00	Libras
3cb1a1c9-662f-4...	475d71d1-ae4f-4...	0023	785.00	Libras
7a2b8d28-a4ef-4...	475d71d1-ae4f-4...	003	785.00	Libras
0b286689-093a-...	475d71d1-ae4f-4...	0116	785.00	Libras
29aa8b87-d8e7-...	475d71d1-ae4f-4...	0119	785.00	Libras
9e8f56f0-b369-4...	475d71d1-ae4f-4...	0135	785.00	Libras



Control Calidad

Sahlman Seafoods

Info. General Detalle

Control Calidad - Proveedor GENERAL

Info. General

Fecha: 10/02/2011

Codigo: C02MA31100211

Total Lbs Report: 14820

Extra

Sup Calidad: Eduardo Pantoja

Produccion: Ismael Bonilla

Control Calidad: Jaime Garcia

Sup Producto: Luis Hernandez

Observaciones: Producto empacado entero en fondo 750 marcado, 800 gr y en caja Farallon 750 kg marcado 106 kg. Color A3

Guardar Editar Cancelar

Liquidación

Sahlman Seafoods - Piscina: 31 - Destino: ancora 7

Liquidacion Camaron Entero

Info. General

Id Liq: 98-4bf0-bd7b-75a7b1689d48 Proceso: HOSO

Id Recep: 475d71d1-ae4f-428b-8e31-84 Importador: ancora 7

Fecha: 10/02/2011 Color: A3

Codigo: C02MA31100211 Liquidador: Manuel Gonzalez

Tipo Caja: 1

Observaciones:

Guardar Cancelar Editar Visualizar Detalle



Detalle Liquidación

Detalle Liquidacion - Piscina: 31 - Destino: ancora 7

Liquidacion

Liquidacion Camaron Entero

Nuevo Editar Eliminar Guardar Cancelar

Talla: 20-30 Clasificacion: M Blast: Blast 1

Cant Cajas: 60 Torre: 150

	Id	IdLiq	Talla	Torre	CantCajas		Blass
	df7cbde3-ac14-4...	ddada0c0-b098-...	20-30	08	78	M	Blast 1
	af9e90b6-0b5c-4...	ddada0c0-b098-...	30-40	08	147	M	Blast 1
	32014dce-4ef9-4...	ddada0c0-b098-...	20-30	39	66	M	Blast 1
	b26630de-a112-...	ddada0c0-b098-...	30-40	39	129	M	Blast 1
	e1fc6859-7c59-4...	ddada0c0-b098-...	10-20	19	1	M	Blast 1
	257bcfbc-8176-4...	ddada0c0-b098-...	20-30	19	68	M	Blast 1
	f3f93052-0db1-4...	ddada0c0-b098-...	30-40	19	123	M	Blast 1
	bdc0cec3-e84c-...	ddada0c0-b098-...	20-30	01	69	M	Blast 1
	57b42746-ce89-...	ddada0c0-b098-...	30-40	01	150	M	Blast 1
	ef780556-6975-4...	ddada0c0-b098-...	20-30	50	78	M	Blast 1
	e7371275-6505-...	ddada0c0-b098-...	30-40	50	123	M	Blast 1
▶	89f2a0aa-6dac-4...	ddada0c0-b098-...	20-30	150	60	M	Blast 1

Control de Bines

Control de Bines....

Fecha: 22/11/2010 11:42:50 a.m. Entrada Salida

Cliente: Sahlman Seafoods Vehiculo: Mazda CH-3455

Fecha	Cliente	Vehiculo	Control
22/11/2010 10:41 a.m.	Sahlman Seafoods	Mazda CH-3455	Salida

Nuevo Guardar Editar Eliminar Cancelar Ingresar Bin



Peso Planta

Peso Reportado en Planta

Peso Planta

Fecha: 10/02/2011

Código: C02MA3110022011

Origen: Farmed

Libras de Entero: 0.00 **Obtener**

CONIC S.A.

Barco: Peso Neto:

Cajillas: Proceso:

Supervisores

Recepción: Mario Hernandez

Granja: Byron Cruz

Producción: Manuel Gonzalez

Observaciones: Monografía para optar al Título de Ingeniero en Sistemas de Información

Fecha	Código	Origen	LbsEntero
10/02/2011 11:43 a.m.	C02MA3110022011	Farmed	0.00

Acciones:

- Nuevo
- Guardar
- Editar
- Eliminar
- Cancelar
- Generar Código
- Gavetas
- Reporte

Libras Empacadas por Área

Libras Empacadas por Area

Fecha: jueves , 10 de febrero de 2011

Elaborado por: Mario Hernandez

Observaciones: Ninguna

Reporte



Conexión

The dialog box 'Modificando Cadena de Conexión....' has two main sections. The left section, 'TIPO DE ACCESO', contains two radio buttons: 'Sahlman Local' (selected) and 'Sahlman Remoto'. Below this are three buttons: 'Nuevo', 'Editar', and 'Guardar'. The right section, 'CADENA DE CONEXION', contains four text input fields: 'Servidor:' with the value 'NTNIC2\COMPAC', 'Usuario:' with the value 'sa', 'Contraseña:' with the value 'namlhas@9518', and 'Base de Datos:' with the value 'Sisconpi'.

Generar Backup

The dialog box 'Generar Backup' has two main sections. The top section, 'Ingrese los Datos del Servidor SQL', contains three text input fields: 'Servidor :' with the value 'NTNIC2\COMPAC', 'Usuario :' with the value 'sa', and 'Contraseña :' with a masked password '*****'. The bottom section, 'Listado de Bases De Datos', contains a dropdown menu for 'Bases de Datos :' with the value 'Sisconpi'. At the bottom of the dialog are three buttons: 'Conectar' (with a globe icon), 'Generar Backup' (with a server icon), and 'Salir' (with a power button icon).



CODIFICACION

Código asociado al Detalle de Liquidación

```
public void EliminarValoresRepetidos(ref List<string> Lista)//FUNCION
ENCARGADA DE ELIMINAR TALLAS Y TORRES REPETIDAS
{
    Lista.Sort();
    for (int i = 0; i < Lista.Count; i++)
    {
        if (i == 0)
        {
        }
        else
        {
            if (Lista[i].CompareTo(Lista[i - 1]) == 0)
            {
                Lista.RemoveAt(i);
                EliminarValoresRepetidos(ref Lista);
            }
        }
    }
}

private void consolidadoToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (TypeLiquidation)//Liquidacion Camaron Entero
    {
        for (int i = 0; i < dataGridViewX1.Rows.Count; i++)
        {
            tallas.Add(dataGridViewX1[2, i].Value.ToString());
            torres.Add(dataGridViewX1[3, i].Value.ToString());
        }
        EliminarValoresRepetidos(ref tallas);
        EliminarValoresRepetidos(ref torres);
    }
    else//Liquidacion Camaron Cola
    {
        for (int i = 0; i < dataGridViewX2.Rows.Count; i++)
        {
            tallas.Add(dataGridViewX2[2, i].Value.ToString());
            torres.Add(dataGridViewX2[3, i].Value.ToString());
        }
        EliminarValoresRepetidos(ref tallas);
        EliminarValoresRepetidos(ref torres);
    }
    //CREACION DEL FORM QUE CONTIENE CONSOLIDADO
    //SE LE PASA IDLIQUIDACION, TALLAS, TORRES, ID DEL TIPO DE
    MASTER, CANTIDAD DE CAJAS DE DICHO MASTERS, TIPO DE
    LIQUIDACION(ENTERO, COLA);
    Cajas = new FormRellenarMaster_Cajas(pIdLiq.ToString(),
tallas, torres, pCantidadCajas, TypeLiquidation);
    Cajas.pIdMaster = this.pIdMaster;
    Cajas.pCliente = this.pCliente;
    Cajas.pNPiscina = this.pNPiscina;
    Cajas.ShowDialog();
}
```



Código asociado al Detalle Peso Reportado en Planta

```
private void FilasEliminadas(int NFilas)
{
    for (int i = 0; i < NFilas; i++)
    {
        if
        (int.Parse(DetalledataGridViewX.Rows[i].Cells[2].Value.ToString()) ==
        (i+1))
        {
            continue;
        }
        else
        {
            this.IncGaveta = i;
            return;
        }
    }
    this.IncGaveta = NFilas;
}
```



CONCLUSIONES

Hemos logrado cumplir con cada uno de los Objetivos propuestos, dando como respuesta el desarrollo de la aplicación: **“SISTEMA DE CONTROL AUTOMATIZADO DEL PROCESAMIENTO DEL CAMARON PARA LA EMPRESA SAHLMAN SEAFOODS, S.A”**, basado en los conocimientos adquiridos sobre Programación y Base de Datos, al mismo tiempo, aplicamos técnicas o metodologías de Ingeniería del Software que involucran una serie de Herramientas, Métodos y Procedimientos que nos han permitido la implementación y desarrollo de dicho Sistema.

La aplicación realizada cubre los procesos que actualmente la planta procesadora aplica al producto entrante, garantizando así, la disminución en el tiempo de recolección, procesamiento y entrega de la Información. Cabe destacar que la aplicación cuenta con un Sistema de Control de Usuario, para dar de alta y baja a las personas que tendrán acceso al Sistema, de esta manera, cada Usuario registrado accederá únicamente a la sección del Sistema que se le ha asignado, según su rol en la planta.

Como se mencionó anteriormente, el Software realizado cubre los procesos de la planta, de una manera en que se aprecie el flujo de procesamiento para el producto, de forma totalmente ordenada, que al final del día con los resultados obtenidos por el sistema, ayudara a una mejor toma de decisiones.



RECOMENDACIONES

1. Con el objetivo de agilizar el proceso de Liquidación y llevar un mejor control sobre el producto saliente, se recomienda agregar un módulo donde se codifique e implemente Códigos de Barra para su posterior lectura a través de handheld.
2. Agregar nuevas funcionalidades para que el sistema interactúe en perfectas condiciones con las aplicaciones que ya tiene la empresa en torno a la Contabilidad de la misma.



BIBLIOGRAFIA

Libros

- CEBALLOS, Francisco Javier. Enciclopedia de Microsoft Visual C#. 2da Edición. Editorial RA-MA. Madrid, España. 2007. p1012. ISBN: 9788478978106.
- Silberschatz, Abraham, Korth, Henry F; BELL Laboratories. Sudarshan, S; Instituto Indio de Tecnología. Fundamentos de Bases de Datos. 4ta Edición. Editora: Concepción Fernández Madrid. Editora de mesa: Susana Santos Prieto. España. Copyright @ MMI, por McGraw-Hill Inc. ISBN: 0-07-228363-7, 84-481-3654-3.

Tesis

- Tesis “Sitio Web de la Vicerrectoría de Investigación y Postgrados de la UNAN-León”. Arauz Escobar, Eunices Elizabeth García. Suministrado por la Biblioteca Central UNAN-León
- “Control Automatizado del Procesamiento del camarón para la Empresa Ecuánica, S.A”. Luisa Elena Espinoza Valladares, Emma Margarita Vallecillo Hernández, Ana Walkiria Vega.

Páginas Web

- www.sahlmanseafood.com.ni
- <http://es.wikipedia.org/wiki/MD5>
- <http://www.devcomponents.com/dotnetbar/>
- http://es.wikipedia.org/wiki/Base_de_datos
- <http://msdn.microsoft.com/es-es/library/ms187079.aspx>
- http://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- <http://es.wikipedia.org/wiki/Framework>
- http://es.wikipedia.org/wiki/Microsoft_.NET
- http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso
- http://es.wikipedia.org/wiki/Diagrama_de_clases



ANEXOS



