



**Trabajo Final de Graduación para optar por el título  
Bachiller en Ingeniería en Computación**

**“Sistema de Gestión de Ventas e Inventario”**

**Elaborado por José Eduardo Méndez Araya**

**Carrera Ingeniería en Computación**

**Instituto Tecnológico de Costa Rica**

**Prof. Asesor: Oscar Víquez Acuña**

**Sede San Carlos**

**Junio, 2010**

# Resumen Ejecutivo

Este documento presenta un informe sobre el proyecto de práctica de especialidad llamado Sistema de Gestión de Ventas e Inventario (SGVI), desarrollado en la empresa Ticofrut S.A., específicamente en el departamento de Tecnologías de Información (TI).

El problema que se soluciona, es la implementación de un sistema informático, que permita realizar la gestión de ventas de productos, el control de inventario de la empresa y la generación de facturas por ventas.

Aquí se detalla una descripción del quehacer de la empresa y del mercado al que se dirige, un análisis total de problema que se abarca especificando las necesidades que se desean cubrir, los beneficiados con el nuevo sistema y los objetivos y alcances que se lograron.

Se muestra un diseño de la estructura planteada para la solución del mismo, y las clases y datos que se manejan en el desarrollo de la aplicación.

# Tabla de Contenidos

<b>Resumen Ejecutivo</b> .....	<b>2</b>
<b>Tabla de Contenidos</b> .....	<b>3</b>
<b>Descripción del problema</b> .....	<b>4</b>
<b>Contexto del Proyecto</b> .....	<b>4</b>
<b>Detalle del Problema</b> .....	<b>5</b>
Enunciado del Problema .....	5
Enunciado de la Solución.....	6
<i>Descripción de los Stakeholders</i> .....	8
<i>Necesidades y expectativas</i> .....	10
<i>Perspectivas, supuestos y dependencias del producto.</i> .....	11
<i>Requerimientos no funcionales</i> .....	11
<b>Análisis de riesgos</b> .....	<b>12</b>
<b>Objetivos y alcances del sistema</b> .....	<b>13</b>
<b>Diagrama de flujo del sistema</b> .....	<b>20</b>
<b>Solución Implementada</b> .....	<b>22</b>
<b>Modelo de diseño</b> .....	<b>22</b>
<i>Descripción de la arquitectura</i> .....	22
<i>Modelo de subsistemas</i> .....	25
<i>Diagrama de clases</i> .....	27
<i>Especificación de componentes del sistema</i> .....	54
<i>Modelo de la base de datos</i> .....	55
<b>Conclusiones y comentarios</b> .....	<b>68</b>
<b>Glosario de términos</b> .....	<b>69</b>
<b>Bibliografía</b> .....	<b>71</b>

# Descripción del problema

## *Contexto del Proyecto*

Ticofrut S.A. es una empresa costarricense procesadora de naranja y piña, fundada en 1988, ubicada en Aguas Zarcas, San Carlos, Alajuela. Sus principales productos son los jugos de naranja concentrados y pasteurizados de naranja y piña. También se producen los siguientes subproductos:

- Aceite esencial
- Aceite de cáscara
- Aromas (naranja y piña)
- D-Limoneno
- Citropulpa pelletizada (pellets)
- Pulpa de naranja congelada estabilizada

Ticofrut es una sociedad anónima, en donde su mayor accionista es el señor Carlos Eduardo Odio Soto.

En la empresa trabajan aproximadamente 600 empleados fijos, más el personal de la finca y el personal que trabaja por contrato.

Dentro de las actividades que se desarrollan en la planta se encuentra el proceso de la naranja, el cual se realiza en los meses de Octubre a Julio; además se encuentra el proceso de la piña que se desarrolla a lo largo de todo el año.

Por otra parte es importante mencionar que el mercado al cual se dirige Ticofrut S.A. es al mercado estadounidense y al mercado europeo. Algunos de los productos que se exportan son los siguientes:

- Jugo pasteurizado de Naranja
- Jugo concentrado de Naranja
- D-Limoneno
- Aceite de cáscara de naranja
- Aceite esencial de naranja
- Aroma de naranja
- Citropulpa (base de concentrado para ganado)
- Jugo de piña concentrado congelado
- Jugo pasteurizado de piña
- Aroma natural (de piña y naranja).

Hoy día la empresa posee la certificación ISO 9001, la cual se refiere a la calidad, esto se desarrolla mediante un conjunto de actividades coordinadas para planificar, implementar, analizar y mejorar lo relacionado con calidad. También logró certificarse ISO 14001 a través de un conjunto de actividades coordinadas para analizar, valorar, evaluar y controlar los aspectos ambientales. [1]

El nuevo proyecto de Sistema de Gestión de Ventas e Inventario (SGVI), beneficia directamente a 3 grandes departamentos de la empresa los cuales son: Departamento de Ventas, Departamento de Producción y Departamento de Calidad.

El Departamento de Ventas realiza en SGVI los procesos de gestión de navieras, gestión de clientes y creación de solicitudes de despacho. Además también controla el proceso de facturación.

El Departamento de Producción lleva la gestión de las solicitudes de despacho, la asignación de producto a las solicitudes, el manejo de fechas de entrega de producto, el manejo del inventario existente, la creación de órdenes de producción y la entrega del producto.

El Departamento de Calidad se encarga de la aprobación de producto acorde a las especificaciones de los clientes y del manejo de documentos solicitados por los mismos.

Actualmente en Ticofrut el sistema utilizado para estos procesos se denomina CPT (Edna), un sistema desarrollado en Visual Basic, el cual ha presentado problemas actualmente por muchas litaciones entre las cuales están:

1. No da la posibilidad de asignar varios productos a las solicitudes de despacho.
2. No lleva un control para la creación de solicitudes de despacho.
3. No lleva un manejo de facturación.
4. Da problemas en el proceso de asignación de producto y el cálculo de de libras solido.
5. No genera reportes sobre ventas.

Esto aparte de que se considera un sistema ya viejo, el cual se quiere sea sustituido por algo tecnológicamente más actual y más sencillo de utilizar.

## *Detalle del Problema*

### Enunciado del Problema

Actualmente Ticofrut S.A. utiliza un sistema llamado CPT (Control de Producto Terminado), el cual cuenta con algunas carencias que no permiten realizar un proceso de control de venta y producción lo más eficiente y ágil posible para la empresa, lo cual trae como consecuencia pérdidas económicas y en tiempo.

Entre las carencias más importantes que presenta el actual sistema están:

- No se cuenta con algún sistema eficiente que permita llevar un control sobre las ventas de producto, ya que actualmente la creación de solicitudes de despacho se maneja de una manera poco eficiente mediante envío por fax de la misma, en donde se debe buscar dentro del inventario producto que coincida con las especificaciones del pedido.
- CPT solo permite asignar un tipo de producto a una solicitud de despacho, lo que limita la venta a una sola clase de producto por orden.
- Se presentan problemas en el sistema actual a la hora de realizar el cálculo de libras sólido en los Packing List de las solicitudes de despacho.

- No se cuenta con la información exacta de los productos existentes en inventario, lo cual trae como consecuencia que mucho producto quede almacenado por mucho tiempo, hasta vencerse su vida útil.
- El proceso de facturación no está ligado con el de cuentas por cobrar, de modo que se invierte mucho tiempo en el papeleo y procesamiento de la misma.
- Se presenta la necesidad de generar reportes sobre las ventas que se realizan esto con el fin de conocer los movimientos de producto, los clientes que más compran producto, y llevar estadísticas en general.
- Se desea contar con la calendarización de las solicitudes de despacho de manera que se pueda visualizar las solicitudes que están más próximas a despachar, y también se pueda consultar el estado de todas.
- Se requiere también llevar un control en la gestión de medios de transporte, navieras, clientes y etiquetas que son elementos utilizados en la elaboración de solicitudes de despacho.
- Se carece de un modulo que permita llevar la gestión de elementos que los clientes piden al realizar el pedido de producto como lo son: Certificados de Calidad, Termógrafos en los contenedores de envío, temperatura de los contenedores, entre otros.

Por otra parte también se siente la necesidad de migrar a tecnología más actual, ya que CPT fue desarrollado en Visual Basic, y muchas de sus utilidades ya no son rentables para la empresa.

## Enunciado de la Solución

Ante el problema enunciado anteriormente es que se busca la creación de este nuevo sistema que se oriente a controlar de manera ágil y eficiente el proceso de venta de producto, desde el ingreso de una nueva solicitud de despacho, hasta su facturación.

El proyecto se divide en 3 grandes fases las cuales son:

- Fase 1: Gestión de datos para la Solicitud de Despacho (SD) y creación de SD.
- Fase 2: Asignación de batchs, Gestión de inventario y escogencia de producto.
- Fase 3: Facturación y cuentas por cobrar.

Para la presentación del sistema como proyecto en la práctica de especialidad, se determino la implementación únicamente de las 2 primeras fases, ya que se considera que la especificación, diseño y programación de las mismas pueden abarcar el tiempo requerido de práctica (Aproximadamente 6 meses). Con la incorporación de la tercera fase se calcula un tiempo

promedio de 9 meses aproximadamente para todo el proyecto. Para la descripción del proyecto como practica de especialidad, se describirán las tres fases.

Para la primera fase pretende realizar procesos que permitan la fácil elaboración de una solicitud de despacho, siguiendo los estándares utilizados actualmente. También se toma en cuenta la gestión de los clientes compradores de producto y de las unidades de transporte que se utilizan para el envío del producto, además de la gestión de documentación que los clientes requieren con el envío de producto (Documentos de análisis, documentos de producción, entre otros). Además se incluye una sección que permite administrar distintos diseños de etiquetas que el cliente quiera utilizar en sus pedidos.

En la segunda fase se incorpora ya la gestión de las solicitudes de despacho, el proceso de asignación de producto a las solicitudes de despacho, el proceso de aprobación de producto por parte de Calidad en caso de existir producto que pueda servir para un cliente aunque no cumpla con las especificaciones claves, y la creación de ordenes de producción en caso de que no exista producto pedido por el cliente.

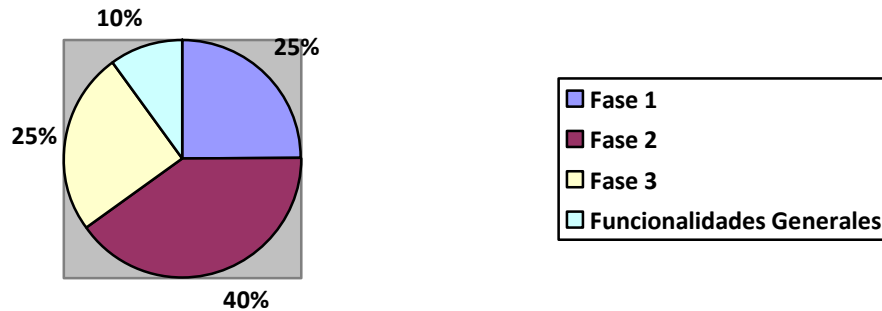
En la tercera fase se involucra la generación de facturas por cuanto a las ventas, y se implementa la generación de reportes en cuanto a ventas, como uno de los puntos principales, y la implementación de algunos otros como requeridos en el sistema.

Como parte del desarrollo de la aplicación se implementan módulos que serán utilizados y modificados en cada una de las fases de la creación del sistema.

Entre las funcionalidades generales que se formaran en cada una de las fases están las siguientes.

- Se maneja una gestión de correos, la cual permitirá informar a los usuarios sobre los procesos que se llevan con los diferentes pedidos entrantes
- Se gestionan los procesos en la interacción del sistema, de modo que se pueden asignar tareas para que el sistema las efectúe. Se contara con una opción para registrar cada proceso en una bitácora si así lo determina el sistema o el usuario.
- Se realizara una gestión de tipos de usuarios, con la cual se podrá asignar usuarios a distintos grupos y a la vez se podrá asignar permisos a los procesos a dichos grupos.

En el siguiente diagrama se muestra el porcentaje de duración estimado para cada una de las fases, tomando en cuenta las funcionalidades generales a implementar.



Se estima un 10% en lo que respecta a programación de las funciones generales utilizadas en las 3 fases del desarrollo. Un 25% para las fases 1 y 3, y un 40% para la fase 2 que contempla la mayor parte de trabajo y cuidado. [2]

### *Descripción de los Stakeholders*

Las personas que hacen uso de esta herramienta, están relacionadas con el Departamento de Producción, el Departamento de Ventas y el Departamento de Calidad de la empresa. Son personas ya familiarizadas con los conceptos y el entorno de las tecnologías de información, que día a día hacen uso de ella.

Por un lado está el Departamento de Ventas, donde el personal se encarga de realizar ordenes acorde a pedidos de clientes o consignatarios, y Luego factura las ventas realizadas.

El Departamento de Producción se encarga de la búsqueda del producto que cumpla con los requerimientos del pedido, de no ser el caso notifica a calidad para su aprobación de otro producto similar al pedido y de no ser positiva la respuesta ordena la producción de más producto.

En el Departamento de Calidad se da visto bueno al producto que se le entregara al cliente, maneja todo lo relacionado con los pedidos de muestras por parte de los clientes y negocia con el cliente la existencia de producto que cumpla con algunas de las especificaciones que este pida en caso de no existir el específico por el cliente.



## Matriz de involucrados.

<b>Rol</b>	<b>Encargado de Ventas</b>
<b>Descripción</b>	<b>Se encarga de realizar ordenes de acuerdo a los pedidos de los clientes o consignatarios y su facturación</b>

<b>Rol</b>	<b>Asistente de Ventas</b>
<b>Descripción</b>	<b>Da asistencia en cuanto a creación de solicitudes de despacho y facturación.</b>

<b>Rol</b>	<b>Gerente de Producción</b>
<b>Descripción</b>	<b>Se encarga de los procesos de producción en la planta</b>

<b>Rol</b>	<b>Asistente de Producción</b>
<b>Descripción</b>	<b>Se encarga de la gestión de las órdenes una vez ya ingresadas al sistema</b>

<b>Rol</b>	<b>Gerente de Laboratorio</b>
<b>Descripción</b>	<b>Administra y controla los procesos que se realizan en el laboratorio de calidad</b>

<b>Rol</b>	<b>Asistente de Laboratorio</b>
<b>Descripción</b>	<b>Realiza trabajos pertinentes al análisis de laboratorio de calidad.</b>

## *Necesidades y expectativas*

Las necesidades que se cubren con la implementación de este sistema son las siguientes:

- Realizar el proceso de creación de solicitudes de despacho en un programa especializado, ya que actualmente se crean por medio de Microsoft Word.
- Poder realizar la gestión de las solicitudes de despacho en un modulo del sistema con todas las opciones pertinentes.
- Poder realizar una gestión más eficaz de los clientes y consignatarios que adquieren producto.
- Poder realizar una gestión más eficiente de las unidades de transporte utilizadas para el traslado de producto.
- Transforma el proceso de asignación de producto a solicitudes de despacho en una acción más ágil, sin errores y con todas las funciones que se requieran.
- Permitir la incorporación de varios productos en las solicitudes de despacho.
- Generación de reportes detallados en lo que respecta a ventas.
- Generación de reportes en cuanto a movimiento de producto en las distintas bodegas con las que cuenta la empresa (Cámaras frías, Tank Farm, Tanques de Moin, etc).
- Manejo de la facturación de venta de producto de manera digital.

Las expectativas en cuanto a este sistema son la de la incorporación de las siguientes funcionalidades:

- Agilizar el proceso de creación de solicitudes de despacho.
- Notificaciones por medio de correo electrónico a la hora de ingreso de una nueva solicitud de despacho u orden de producción.
- Lograr la asignación de más de un producto a las solicitudes de despacho.
- Obtener fechas y estados en cuanto a solicitudes de despacho y órdenes de producción.
- Obtener datos sobre movimientos de producto en bodega.
- Obtener información exacta de las existencias en el inventario.
- Obtener información detallada en cuanto a las ventas de producto.
- Agilizar el proceso de generación de facturas.

### *Perspectivas, supuestos y dependencias del producto.*

La aplicación viene a remplazar algunas áreas del trabajo que realiza el actual sistema CPT, de manera que estos procesos remplazados sean más eficientes, y permitan mejorar la calidad del trabajo y economizar tiempo y dinero en los departamentos de Ventas, Producción y Calidad.

Los factores que pueden afectar los requerimientos planteados son:

- Poco tiempo para el desarrollo del sistema.
- Aparición de errores en el sistema durante su evaluación.
- Requisitos muy complejos que afectarían el tiempo y esfuerzo para otros.
- Cambio constante en los requisitos.
- Problemas de los usuarios para acoplarse al nuevo sistema.
- Mala comunicación con los stakeholders.

### *Requerimientos no funcionales*

- La interfaz de usuario para cada una de las pantallas del sistema debe seguir el estándar que exige la empresa, con el logo de la empresa, los componentes de acceso (botones) en un lugar determinado, los componentes de entrada de datos en otra posición, y los de selección en otra.
- Para el ingreso a la aplicación se deberá ingresar un usuario y una contraseña, la cual determinara las funciones que el usuario podrá efectuar en la aplicación. En el caso de ventas se presenta las pantallas de creación de órdenes, y la de creación de facturas para las ventas. En producción se visualiza el inventario existente, y el control de asignación de productos a despacho. En despacho se trabaja con la asignación a contenedores. En cuentas por cobrar se trabaja con las pantallas del actual sistema ERP Exactus.
- La fuente en los textos en todas las ventanas será de tipo Arial como estándar.
- La aplicación trabajara sobre una resolución de 800 x 600 como la base, aunque el usuario podrá definir la que más le convenga.
- El sistema debe permitir que en cada una de las pantallas donde aparezca un checkbox, sea posible seleccionarlos uno por uno, en cualquier región donde sea seleccionado, o poder ser seleccionados por regiones de los mismos.

- El sistema debe tener distribuidos los botones en la pantallas con un buen grado de separación, y estos deben ser de un tamaño adecuado que impida su pulsación errónea.
- Se tiene el atributo orden de la tabla ctp\_orden como llave primaria, y este no es un consecutivo sino que es de tipo texto que provoca que deba ser modificado, lo que acarrea que tenga que modificarse en muchas ocasiones las tablas de las bases de datos.
- El sistema debe manejar un atributo adicional en la tabla cpt\_orden con el nombre cod\_orden donde esta sea la llave primaria y que además sea generada automáticamente por el sistema con un numero consecutivo.
- Sistema Operativo Windows 2000, XP, Vista o 7.

## *Análisis de riesgos*

### Riesgos en tiempo.

- Incremento o cambio de requerimientos durante el desarrollo del sistema.
- Problemas en el uso de la tecnología o desconocimiento de herramientas de desarrollo.
- Fallas en el equipo de hardware utilizado para la producción de la aplicación.

### Riesgos en costo.

- Posible adquisición de nuevas tecnologías o herramientas para el buen desarrollo del mismo.
- Atrasos en el tiempo de entrega que supondrán más costo de mano de obra y de otros recursos.
- Posibles errores que puedan aparecer durante las pruebas del sistema, que supondrán la inversión de más tiempo y por ende más costo económico.

### Riesgos en calidad.

- Al presentarse poco tiempo para la implementación de la base principal del proyecto (proceso de órdenes y facturación), puede quedar por fuera algunos aspectos que garanticen la calidad del sistema.
- Un sistema de gran complejidad como el presente, debe ser tratado con un fuerte análisis paso a paso de cada una de sus funciones y procesos, lo que demanda tiempo y aprendizaje del mismo, y en parte puede dejar por paso el aseguramiento de la calidad.

## *Objetivos y alcances del sistema*

### **Objetivo Principal**

Mejorar y agilizar el proceso de solicitudes de despacho, desde su pedido, hasta su facturación y puesta en cuentas por cobrar.

### **Objetivos Secundarios**

- Agilizar el proceso de producción en el ámbito del manejo de solicitudes de despacho de producto.
- Mejorar el proceso de asignación de producto a las solicitudes de despacho en estado de tránsito.
- Mejorar cada una de las etapas que comprenden el recibo de una nueva solicitud de despacho, como lo son su creación, la verificación sus datos, la selección de los batchs que cumplen con sus especificaciones, la producción de producto, de no existir con respecto a la especificación; el envío a despacho para su empaque y la facturación.
- Realizar la tarea de facturación de una manera ordenada y precisa que permita economizar tiempo en su elaboración.
- Ligar los clientes que se encuentran en el sistema de producción con los ya existentes en el sistema de cuentas por cobrar.
- Obtener reportes detallados en cuanto a ventas e inventario se refiere.

### **Alcances**

Alcances del sistema [3]

- Estructura ORM (Mapeo Objeto-Relacional), para el manejo de las tablas de la base de datos que se gestionarán en la aplicación.
- Proceso de envío de correos electrónicos a los usuarios del sistema cuando se presenten los eventos de creación, modificación o cancelación de una solicitud de despacho y la creación, modificación o cancelación de una orden de producción.
- Módulo para la gestión de los clientes compradores de producto, que permite asignarles tareas comúnmente solicitadas en sus pedidos de productos a los clientes, y los productos que el cliente pide. Además asigna tanto los consignatarios para un cliente determinado, como

asignación un mismo cliente como consignatario de otros clientes. La información pertinente a los clientes es:

- Cliente: Corresponde a un identificador único del cliente, es de tipo alfanumérico, en mayúscula, soporta un máximo de 50 caracteres, es un dato requerido.
  - Nombre: Describe al cliente, sea por su nombre, siglas o valor que lo caracterice, es de tipo alfanumérico y soporta un máximo de 100 caracteres, es un dato requerido.
  - País: Especifica la nacionalidad del cliente, es de tipo alfabético, seleccionable por el usuario dentro de un conjunto de países previamente definidos dentro del sistema, es un dato requerido.
  - Dirección: Especifica la dirección exacta del domicilio del cliente, es de tipo alfanumérico, soporta un máximo de 200 caracteres y es un dato opcional.
  - Números de Teléfono: Especifica una lista de números telefónicos del cliente, sean estos comerciales, de fax o celulares. Cada número de teléfono es de tipo alfanumérico, soporta un máximo de 60 caracteres, es un dato opcional.
  - Página Web: Especifica la dirección en Internet del sitio del cliente, es de tipo alfanumérico, soporta un máximo de 200 caracteres y no es un dato requerido.
  - Correos Electrónicos: Especifica una lista de correos electrónicos del cliente. Cada correo electrónico es de tipo alfanumérico, soporta un máximo de 500 caracteres. No es un dato requerido.
  - Productos Solicitados Comúnmente: Especifica una lista de productos solicitados comúnmente por el cliente. No es un dato requerido.
  - Documentos solicitados por el cliente: especifica una lista de documentos que el cliente solicita comúnmente a la hora de realizar compra de producto. Es un valor seleccionable de una lista de documentos ya existentes y no es requerido.
  - Notas: Especifica cualquier detalle que sea necesario almacenar con respecto al cliente, es de tipo alfanumérico, soporta un máximo de 500 caracteres y es un dato opcional.
  - Estado: Permite activar o desactivar el cliente para que sea visible o no en otras operaciones donde sea requerido utilizar. Es de tipo selección única, es un dato requerido. Para clientes nuevos se define el estado como activo.
  - Datos de Auditoria: Describe información de auditoría para la creación o modificación en la información de un cliente.
- Módulo para la gestión de navieras, que controla la información de las mismas para su asignación en las programaciones de entregas. La información pertinente a las navieras es:
- Naviera: Corresponde a un identificador único de la naviera, es de tipo alfanumérico, soporta un máximo de 50 caracteres, es un dato requerido.
  - Nombre: Describe la naviera, sea por su nombre, siglas o valor que lo caracterice, es de tipo alfanumérico y soporta un máximo de 100 caracteres, es un dato requerido.
  - País: Especifica la nacionalidad de la naviera, es de tipo alfanumérico, seleccionable por el usuario dentro de un conjunto de países previamente definidos dentro del sistema, es un dato requerido (ver **¡Error! No se encuentra el origen de la referencia.**).
  - Dirección: Especifica la dirección exacta de las oficinas o planta de la naviera, es de tipo alfanumérico, soporta un máximo de 200 caracteres y es un dato opcional.

- Números de Teléfono: Especifica una lista de números telefónicos de la naviera, sean estos comerciales, de fax o celulares. Cada número de teléfono es de tipo alfanumérico, soporta un máximo de 60 caracteres, es un dato opcional.
  - Página Web: Especifica la dirección en Internet del sitio Web de la naviera (si esta cuenta con ello), es de tipo alfanumérico, soporta un máximo de 200 caracteres y es un dato opcional.
  - Correos Electrónicos: Especifica una lista de correos electrónicos de la naviera. Cada correo electrónico es de tipo alfanumérico, soporta un máximo de 100 caracteres. Es opcional.
  - Notas: Especifica cualquier detalle que sea necesario almacenar con respecto a la naviera, es de tipo alfanumérico, soporta un máximo de 500 caracteres y es opcional.
  - Estado: Permite activar o desactivar la naviera para que sea visible o no en otras operaciones donde sea requerida utilizar. Es de tipo selección única, es un dato requerido. Para navieras nuevas se define el estado como activa.
  - Datos de Auditoria: Describe información de auditoría para la creación o modificación en la información de una naviera (ver **¡Error! No se encuentra el origen de la referencia.**).
- Módulo de asignación de producto (Tambores, Bins o Granel) a los Packing List, según las especificaciones de la solicitud de despacho requeridas por el cliente.
- Módulo de gestión de tipos de medios de transporte utilizados para asignar una categoría a los distintos transportes de la empresa como ajenos a ella. La información pertinente a los tipos de medios de transporte es:
- Código: Corresponde a un identificador único del tipo de unidad de transporte, es de tipo alfanumérico, soporta un máximo de 50 caracteres, es un dato requerido.
  - Descripción: Es el tipo de unidad de transporte descrito por su nombre, siglas o valor que lo caracterice, es de tipo alfanumérico y soporta un máximo de 100 caracteres, es un dato requerido.
  - Capacidad: es el tipo de capacidad que transporta, sea esta en toneladas métricas para las entregas a granel o cantidad de tambores en el caso de contenedores.
  - Notas: Especifica cualquier detalle que sea necesario almacenar con respecto al tipo de unidad de transporte, es de tipo alfanumérico, soporta un máximo de 500 caracteres y es opcional.
  - Estado: Permite activar o desactivar el tipo de unidad de transporte para que sea visible o no en otras operaciones donde sea requerido utilizar. Es de tipo selección única, es un dato requerido. Para tipos de unidades de transporte nuevos se define el estado como activo.
  - Datos de Auditoria: Describe información de auditoría para la creación o modificación en la información de un tipo de medio de transporte.
- Módulo de mantenimiento de medios de transporte donde se especifica la unidad de transporte para las entregas de producto. La información pertinente a los medios de transporte es:

- Placa: Corresponde a el número de matrícula del transporte, el cual funge como identificador único de la unidad de transporte, es de tipo alfanumérico, soporta un máximo de 50 caracteres, es un dato requerido.
  - Descripción: Es la marca estilo o nombre característico de la unidad de transporte, es de tipo alfanumérico y soporta un máximo de 100 caracteres, es un dato requerido.
  - Notas: Especifica cualquier detalle que sea necesario almacenar con respecto a la unidad de transporte, es de tipo alfanumérico, soporta un máximo de 500 caracteres y es opcional.
  - Imagen de Muestra: Especifica un archivo de imagen que permite visualizar la unidad de transporte. Es seleccionable por el usuario desde los archivos de su computadora, es un dato opcional.
  - Tipo de unidad de transporte: Especifica el tipo de transporte al que pertenece una determinada unidad de transporte de una naviera, soporta un máximo de 50 caracteres y es requerido.
  - Naviera: Especifica la naviera a la cual pertenece la unidad de transporte, soporta un máximo de 50 caracteres y es requerido.
  - Estado: Permite activar o desactivar la unidad de transporte para que sea visible o no en otras operaciones donde sea requerido utilizar. Es de tipo selección única, es un dato requerido. Para unidades de transporte nuevas se define el estado como activa.
  - Datos de Autoría: Describe información de auditoría para la creación o modificación en la información de una unidad de transporte.
- Módulo de mantenimiento de países que gestiona las naciones asignadas tanto a los clientes como a las navieras en el sistema. La información pertinente a los países es:
- Código de país: Especifica el código de país, es de tipo texto alfanumérico, suporta un máximo de 3 caracteres, es un dato requerido. Se sugiere que se utilice el estándar ISO 3166 Alpha 3.
  - Nombre: Especifica el nombre del país, es de tipo texto, alfabético, soporta un máximo de 100 caracteres, es un dato requerido.
  - Notas: Especifica cualquier detalle que sea necesario almacenar con respecto al país, es de tipo alfanumérico, soporta un máximo de 500 caracteres y es un dato opcional.
  - Estado: Permite activar o desactivar el país para que sea visible o no en otras operaciones donde sea requerido utilizar. Es de tipo selección única, es un dato requerido. Para países nuevos se define el estado como activo.
  - Datos de Auditoria: Datos de Auditoria: Describe información de auditoría para la creación o modificación en la información de un cliente. Ver Glosario de Términos.
- Módulo de mantenimiento de etiquetas, para el manejo de un catalogo de etiquetas asignadas en una solicitud de despacho para un producto específico. La información pertinente al manejo de etiquetas es:
- Código de etiqueta: Corresponde a un identificador único de la etiqueta, es de tipo alfanumérico, soporta un máximo de 50 caracteres, es un dato requerido.
  - Descripción: Especifica un detalle para la etiqueta, sea su nombres, siglas, o valor que la caracterice, es de tipo alfanumérico, soporta un máximo de 500 caracteres. Es un dato opcional.



- Imagen de Muestra: Especifica un archivo de imagen que permite visualizar una muestra de la etiqueta personalizada. Es seleccionable por el usuario desde los archivos de su computadora, es un dato opcional.
  - Estado: Permite activar o desactivar una etiqueta para que sea visible o no en otras operaciones donde sea requerido utilizar. Es de tipo selección única, es un dato requerido. Para etiquetas nuevas se define el estado como activa.
  - Datos de Auditoría: Describe información de auditoría para la creación o modificación en la información de una etiqueta.
- Módulo de mantenimiento de tareas para solicitudes de despacho, donde se gestionan las tareas requeridas por los clientes a la hora de solicitar pedido de producto. La información pertinente a las tareas para solicitudes de despacho es:
- Código: Corresponde al código de la tarea, es un dato alfanumérico que soporta un máximo de 10 caracteres, es un dato requerido y debe de ser autogenerado por el sistema.
  - Descripción: Corresponde a la directiva o instrucción que debe de ser acatada por un usuario con la finalidad de asegurar la completitud de una SD, es un dato alfanumérico que soporta un máximo de 500 caracteres, es un dato requerido.
  - Notas: Especifica cualquier detalle descriptivo que sea necesario almacenar, es de tipo alfanumérico, soporta un máximo de 500 caracteres y no es un dato requerido.
  - Tipo de Tarea: Corresponde a la clasificación que se le asignará a la tarea, esta información se obtendrá del catálogo de tipos de tareas, es un dato requerido.
  - Área Responsable: Corresponde al área que será responsable de darle seguimiento a la tarea, es de tipo selección y se obtiene del catálogo de áreas registradas en el sistema.
  - Estado: Permite activar o desactivar una tarea, es de tipo marca, es un dato requerido que por defecto tiene su valor como activo.
  - Datos de Autoría: Permite ver que usuario creo o modifiko una tarea por última vez.
- Módulo de mantenimiento de machotes, donde se gestionan distintos formatos de machotes, para facturación, memos, solicitudes de despacho, ordenes de producción, entre otros. La información pertinente para el manejo de machotes es:
- Machote: Corresponde a un identificador único para el machote, es de tipo alfanumérico, soporta un máximo de 50 caracteres, es un dato requerido.
  - Descripción: Especifica cualquier detalle que sea necesario almacenar con respecto al machote, es de tipo alfanumérico, soporta un máximo de 500 caracteres y es opcional.
  - Cliente: Especifica el cliente asignado a un determinado machote, es un dato seleccionable por el usuario dentro de un conjunto de clientes previamente definidos dentro del sistema, es un dato requerido.
  - Imagen de muestra: Especifica un archivo de imagen que permite visualizar una muestra del machote. Es seleccionable por el usuario desde los archivos de su computadora, es un dato opcional.
  - Estado: Permite activar o desactivar el machote para que sea visible o no en otras operaciones donde sea requerido utilizar. Es de tipo marca, es un dato requerido que por defecto esta marcado.
  - Datos de Autoría: Permite ver que usuario creo o modifiko un machote por última vez.

- Módulo de mantenimiento de solicitudes de despacho, donde se controla los datos expresados en la creación de solicitudes de despacho, de manera que si se requiere el cambio o cancelación en alguno de los aspectos se realizan desde este módulo.
  
- Módulo de mantenimiento de tipos de tareas, que permite asignar categorías a las tareas que se asignan a las solicitudes de despacho. La información pertinente a los tipos de tareas es:
  - Tipo de Tarea: Corresponde al código de identificación, es de tipo alfanumérico soporta un máximo de 10 caracteres, con valores en mayúscula, es un dato requerido.
  - Descripción: Corresponde a la identificación más apropiada que se le dé al tipo de tarea, es de tipo alfanumérico, soporta un máximo de 100 caracteres, es un dato requerido.
  - Estado: Permite activar o desactivar un tipo de tarea, es de tipo marca, es un dato requerido que por defecto tiene su valor como activo.
  - Datos de Autoría: Permite ver que usuario creo o modifíco un tipo de tarea por última vez.
  
- Módulo tipo Wizard que captura paso a paso los datos necesarios para la creación de solicitudes de despacho.
  
- Módulo tipo Wizard que captura paso a paso los datos necesarios para la creación de ordenes de producción.
  
- Módulo de despacho de contenedores, donde se controla la salida del producto de bodegas a los contenedores con la información específica en cuanto al cliente, la solicitud de despacho, los pesos del producto, el número de contenedor y la naviera responsable.
  
- Módulo de programación de entregas donde se especifican las entregas por semana que se deben realizar con la información de orden, cliente, producto, cantidad, medio de transporte, naviera y fecha de despacho de planta.
  
- Calendario de solicitudes de despacho, donde se visualiza por medio de un calendario tipo Gantt las solicitudes de despacho existentes en estado de transito, y se pueden consultar sus fechas de entregas, así como modificar sus entregas de ser necesario.
  
- Calendario de ordenes de producción, donde se visualiza por medio de un calendario tipo Gantt las ordenes de producción existentes en estado de transito, y se pueden consultar las fechas de finalización de la producción y a la vez modificar dichas fechas de ser necesario.

- Módulo de administración de correos de usuarios, donde se gestionan los correos para el envío de notificaciones en cuanto a creación o cancelación de ordenes de producción y solicitudes de despacho.
- Módulo de administración de departamentos, donde se gestiona los datos referentes a los departamentos implicados en el sistema SGVI.
- Módulo de administración de áreas, donde se gestionan las áreas de cada departamento implicadas en el sistema SGVI.
- Módulo para la asignación de permisos a usuarios, donde se selecciona el usuario requerido, y se le asigna mediante checks box en un árbol los permisos a los que tendrá acceso.
- Módulo para el cambio de contraseña para los usuarios, donde los mismos ingresan digitan su nombre de usuario, su clave actual y luego la nueva clave.
- Manejo de auditoria para las acciones de los usuarios, donde se registran las modificaciones que los usuarios realizan en los distintos módulos del sistema.
- Manejo de una bitácora para el registro historial de los trabajos o procesos realizados por los usuarios.
- Manejo de sesiones de usuarios, de manera que el usuario ingresa al sistema y puede cambiar de sesión sin necesidad de cerrar la aplicación.
- Proceso de exportación e impresión de datos a archivos de tipo Excel o PDF.
- Proceso de cálculo de libras de sólido en el módulo de despacho de contenedores.

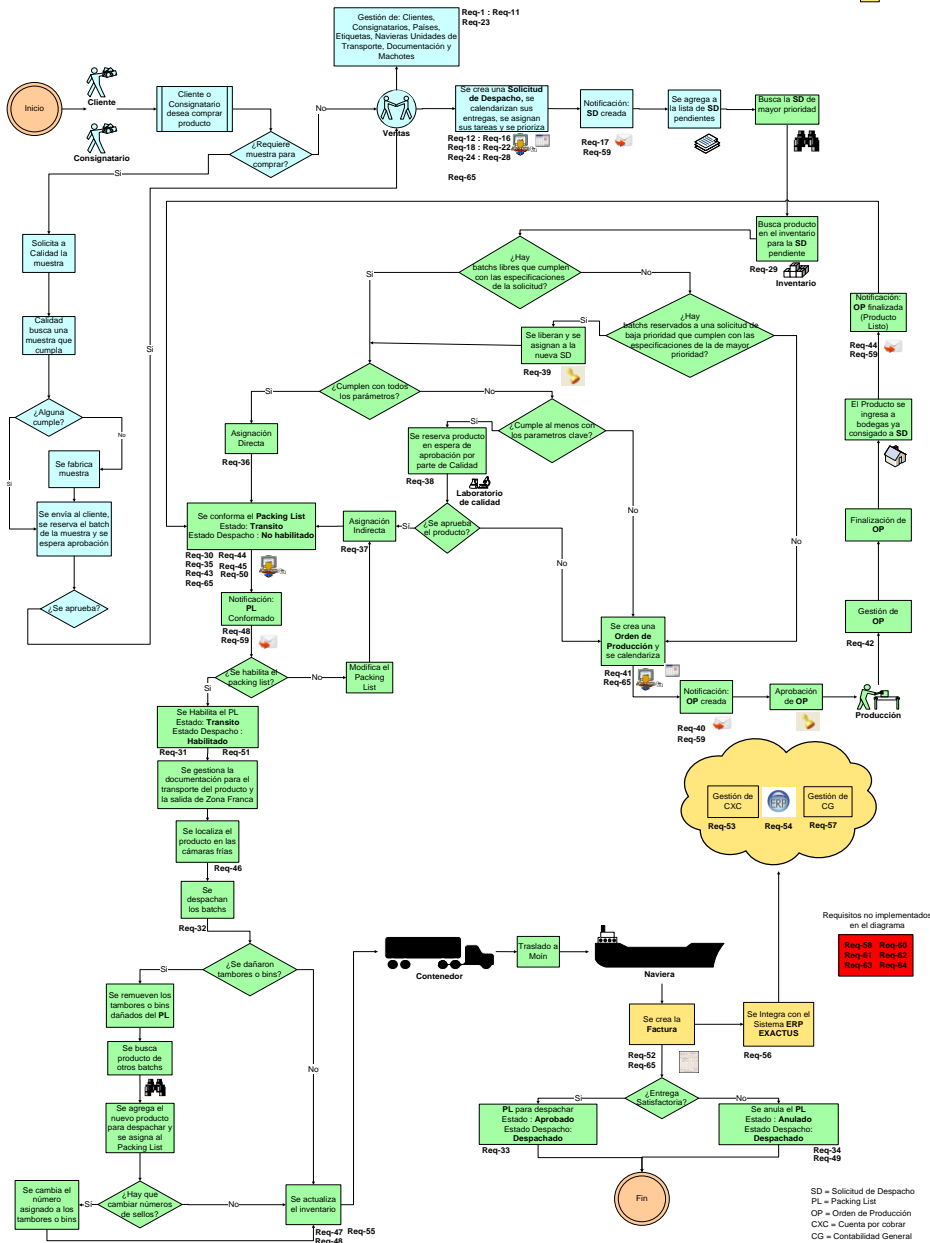
# Diagrama de flujo del sistema

Este diagrama representa las tres fases (fase 1: color celeste, fase 2: color verde, fase 3: color naranja) que abarca el sistema en donde se contempla paso a paso los procesos desarrollados que dan vida a SGVI.



## Sistema de Gestión de Ventas e Inventario

- Fase 1 ■
- Fase 2 ■
- Fase 3 ■



En él se muestra desde el inicio con el pedido de una nueva solicitud de despacho por parte de un cliente o consignatario, en donde el Departamento de Ventas crea y envía una nueva solicitud de despacho, se notifica al Departamento de Producción para su procesamiento, en donde este verifica su prioridad e inicia la selección de producto para la misma, de existir producto sin ser asignado y que cumpla con todas las especificaciones solicitadas por el cliente, se inicia con la asignación del producto a los Packing List. Si existe producto ya asignado a otras solicitudes de menor prioridad, estas son reasignadas a la nueva solicitud de despacho. De existir producto que cumpla con las especificaciones claves del pedido y no así con las específicas, se le comunica al Departamento de Calidad, para que negocie con el cliente y determinar si tal producto puede servir para suplir sus necesidades. De existir acuerdo, dicho producto es asignado a los Packing List de la solicitud de despacho. De no existir producto que cumpla con todos los requerimientos solicitados por el cliente, se crea y envía una orden de producción para la fabricación del producto solicitado.

Con el producto listo y asignado a los Packing List, se notifica al área de empaque para que busque el producto lo empaque en los tambores u otro tipo de empaque solicitado por el cliente, lo almacene en las bodegas, y lo mantenga listo para su fecha de despacho.

En la de fecha de despacho se carga el producto en los contenedores u otro tipo de transporte solicitado, se actualiza el inventario, y se prepara y envía la facturación del producto. Luego se registra la venta en el sistema de Exactus para el registro de la cuenta por cobrar.

# Solución Implementada

## Modelo de diseño

### Descripción de la arquitectura

La implementación del SGVI se desarrollo en el lenguaje de programación Delphi. Este sistema se encuentra sobre una estructura (framework), ya creada en el mismo lenguaje, la cual se denomina "APLICACIÓN\_TICOFRUT". Esta estructura está dividida de una manera muy similar a una arquitectura por capas, en donde se cuenta con una interfaz llamada TFPANTALLAS que controla las clases referentes a la capa de presentación, se cuenta con otra interfaz llamada TFLOGICA que se en carga de controlar las funciones y métodos referentes a la capa de negocio y finalmente esta la interfaz llamada TFDATOS que cuenta con los métodos para el manejo de la información referente a la base de datos.

Esta estructura provee a este y a otros sistemas que se desarrollan, múltiples funcionalidades, entre las cuales se mencionan:

- La estructuración de pantallas: las aplicaciones que se desarrollan en Ticofrut S.A. manejan básicamente 4 módulos de funcionalidades, las cuales son: Mantenimientos, Procesos, Consultas y reportes.
  - Mantenimientos: son las pantallas que permiten la gestión de datos (Ingresar, modificar y borrar) que luego serán utilizados para su procesamiento y obtención de resultados.
  - Procesos: son las pantallas que con la información que se almacena, realizan cálculos y procesos para obtener resultados queridos.
  - Consultas: permiten obtener información general y precisa sobre algún resultado de algún proceso, ejemplo: calendario con fechas de inicio y fin de una solicitud de despacho.
  - Reportes: despliegue detallado de información requerida por un cliente de los resultados de un proceso en el sistema.
- Estructuración de la capa de datos: implementa los métodos necesarios para la conexión a la base de datos, y el manejo de consultas y otros procesos referentes a los datos en el sistema. Esto evita la implementación de los métodos de conexión y consulta a la base de datos, en cada una de los módulos o pantallas que se crean.

- Manejo de la seguridad en el sistema: esta estructura cuenta con los métodos necesarios para la verificación de permisos de los usuarios tanto en SGVI, como en otros sistemas existentes. Su funcionalidad se basa en mostrar o alertar al usuario, dependiendo del caso, sobre sus privilegios en algunas de las funcionalidades del sistema.
- Manejo de validaciones: implementa el proceso de validaciones con el cual por cada modulo o pantalla se realiza la comprobación de los datos ingresados o faltantes, y se alerta al usuario sobre la situación. Esto como en la estructuración de la capa de datos, permite ser implementado en solo esta estructura, y no en cada modulo ya sea en el de SGVI o de otro que lo utilice.
- Manejo de auditoría: contiene funciones para el almacenamiento y control en el sistema de las auditorias de cada usuario, lo cual permite tener un manejo de las modificaciones o alteraciones que se realicen dentro del mismo, y conocer el o los responsables en caso de presentarse problemas (registros borrados o alterados que sean de suma importancia).

Esta estructura se encuentra supervisada por una clase controlador, que se encarga de verificar en cada uno de los sistemas, las acciones que ejecute el usuario, de modo, como por ejemplo si el Sistema de Gestión de Ventas e Inventario es accedido por un usuario para la creación de una nueva solicitud de despacho, este controlador realiza la verificación del usuario según los permisos que tenga en este sistema, a la hora de iniciar la sesión le da las opciones a la cuales tiene permiso y con ello si cuenta con privilegios sobre la creación de una nueva solicitud de despacho el controlador le habilitara la opción y responderá a las acciones que el usuario realice en el proceso de creación de la misma.

Con la incorporación del Sistema de Gestión de Ventas e Inventario se modificaron y perfeccionaron, muchas de las funciones que provee esta estructura, como la incorporación de un procedimiento para el envío de correos electrónicos, el manejo de sesiones para los usuarios y el despliegue de funcionalidades al usuario, acorde a sus privilegios.

En resumen esta estructura viene a ser como el esqueleto o la base de apoyo para el SGVI, permitiendo a la vez incorporar otras funcionalidades que se utilizaran en este proyecto y en otros futuros.

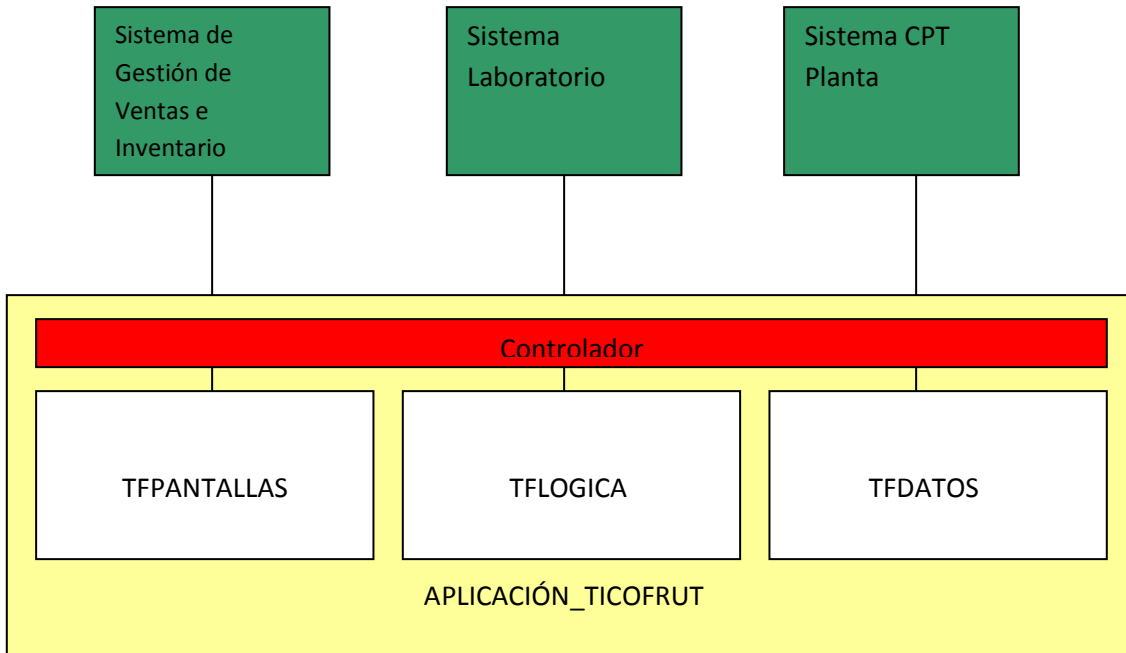


Diagrama de la arquitectura del sistema.



## *Modelo de subsistemas*

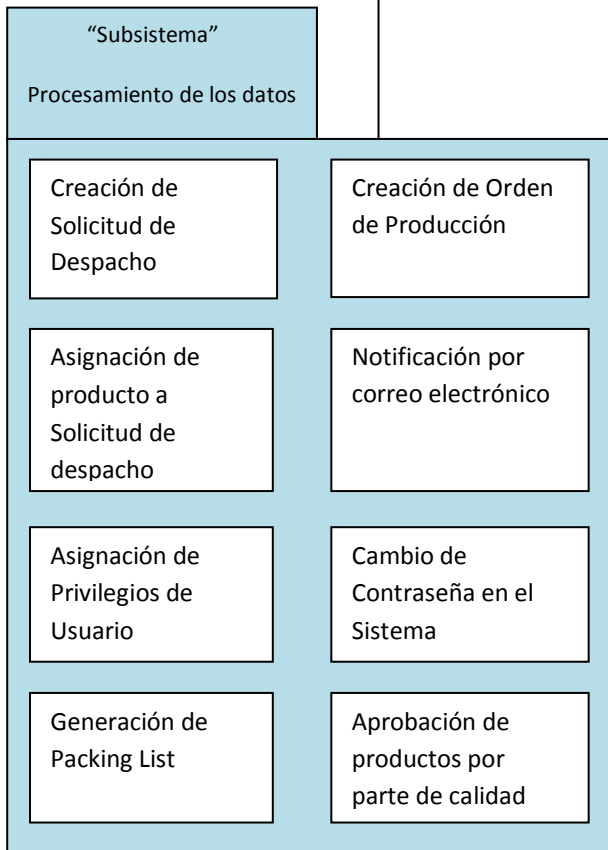
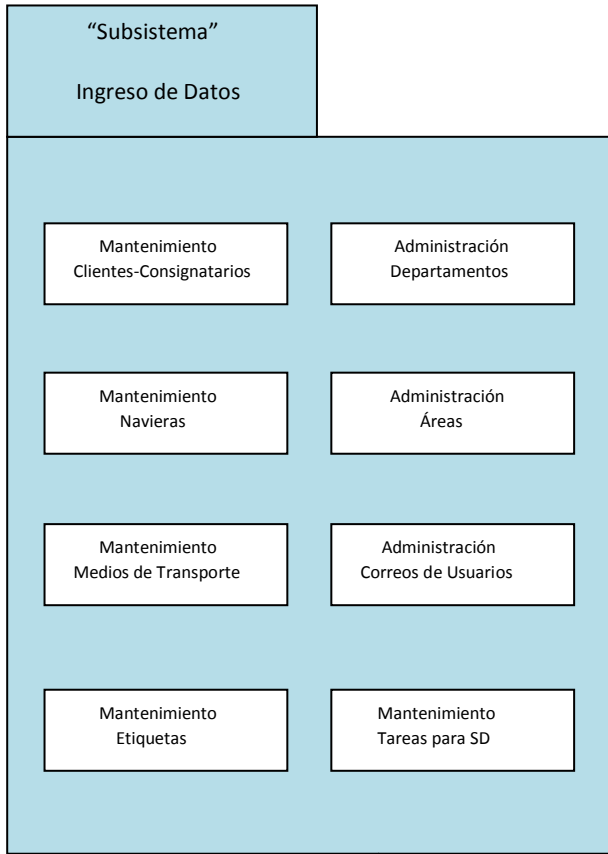
La aplicación esta visualizada en 4 subsistemas en lo que respecta en su totalidad de funcionamiento, donde cada subsistema realiza métodos y funcionalidades de las que dependerán los demás subsistemas, teniendo así un punto de inicio y un punto final.

Primero se inicia con el subsistema de ingreso de datos, en el cual se concentra la recolección de datos de origen remoto, estos datos de entrada son los que serán procesados en el subsistema de procesamiento de datos. La información necesaria para el funcionamiento de SGVI será administrada por el modulo de mantenimientos (Mantenimiento de Clientes, Mantenimiento de Navieras, Mantenimiento de Etiquetas de producto, entre otros), el cual provee a la aplicación de los medios para la recolección de los datos.

Luego se trabaja con el subsistema de procesamiento de los datos, en donde con la información recopilada en el subsistema Ingreso de Datos se realizan los cálculos y se ejecutan los procesos que darán vida a los resultados o información requerida por el usuario. Mediante la ejecución de este subsistema, se logra obtener información vital para la generación de consultas y reportes, y es en si la parte principal de Sistema de Gestión de Ventas e Inventario.

El subsistema de almacenamiento de los datos se enfoca básicamente en la base de datos, de manera que la información que se maneja tanto en el ingreso, procesamiento como consulta en el sistema esta supervisada por este subsistema. Tanto los mantenimientos que permiten la gestión de información de entrada, como Los resultados obtenidos en el procesamiento de datos son almacenados en este subsistema, para su posterior requerimiento.

El subsistema de despliegue de información está orientado a manejar lo que respecta a reportes y consultas que se quieran generar por parte del usuario. La información es tomada del subsistema de almacenamiento de datos que previamente almaceno los datos requeridos. En SGVI se generaran reportes consiguientes a ventas de producto, Movimientos de producto en el inventario (movimiento entre bodegas) y reportes de existencia y cantidad de producto; y se manejan consultas en cuanto a la las fechas de inicio y fin en las solicitudes de despacho y las ordenes de producción y en detalles de las mismas.

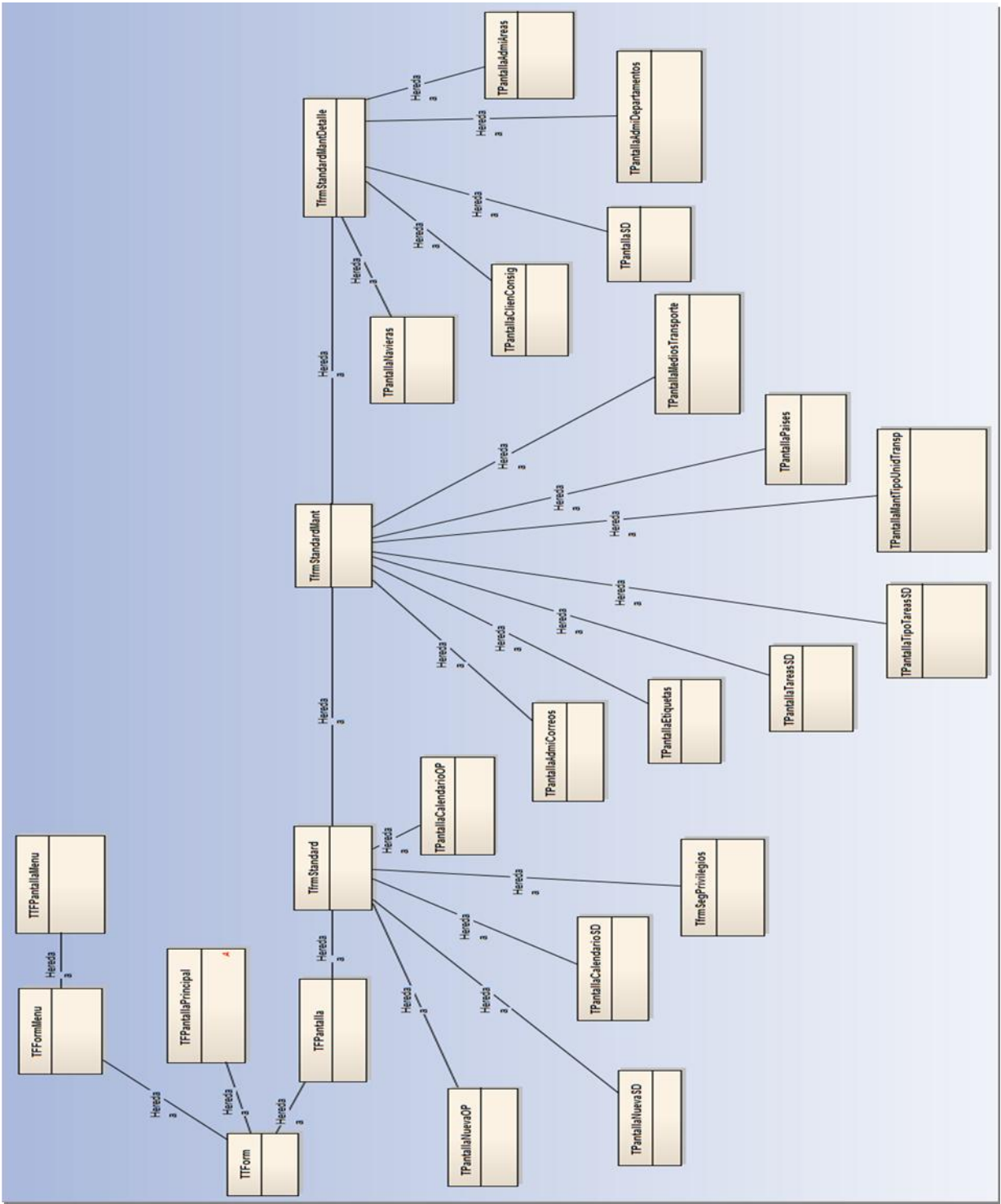


### *Diagrama de clases*

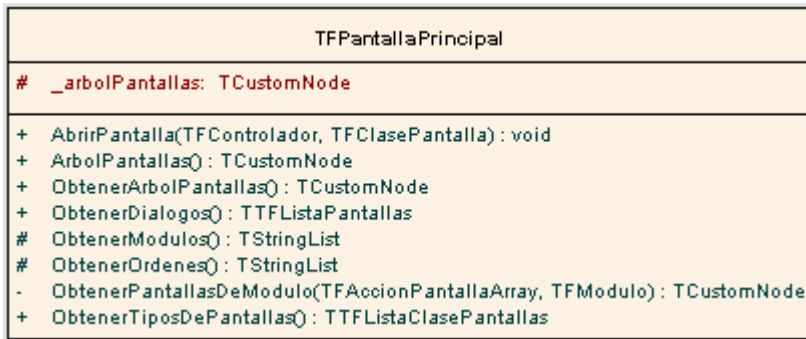
El diagrama de clases se presenta mediante 3 subdiagramas que conforman la totalidad del proyecto de práctica, donde se exponen cada uno de sus módulos, y se especifica las propiedades y operaciones de cada una de las clases.

#### Subdiagrama 1:

Este Subdiagrama representa las clases en cuanto a pantallas en el sistema se refiere, se contemplan las pantallas principales y las herencias de cada una.



Clases:



Esta clase es la que implementa la pantalla principal del sistema, cuenta con las siguientes propiedades y operaciones:

- **\_arbolPantallas**: Implementa nodos para cada una de las pantallas hijas.
- **AbrirPantalla**: realiza el proceso de abrir ventanas al cliente por medio del controlador.
- **ObtenerArbolPantalla**: obtiene el nodo del árbol de pantalla.
- **ObtenerDialogos**: obtiene los diálogos del sistema.
- **ObtenerModulos**: recibe la lista de módulos del sistema.
- **ObtenerOrdenes**: recibe la lista de órdenes en el caso de una creación de solicitud de despacho.
- **ObtenerPantallasDeModulo**: recibe las acciones (pantallas) del sistema.
- **ObtenerTiposDePantallas**: obtiene los distintos tipos de pantallas que se manejan en el sistema.



Esta clase permite la implementación de las funcionalidades de guardar, nuevo y demás en todas las pantallas de la aplicación. Sus propiedades y operaciones:

- `_tipoPantalla`: determina el tipo de pantalla.
- `Anterior`: implementa la funcionalidad de anterior en el sistema.
- `Catalogo`: maneja un catalogo de pantallas a las que le atribuye sus valores.
- `Cerrar`: implementa la función de cerrar la pantalla en el sistema.
- `CodigoPantalla`: determina el código de la pantalla.
- `DoClose`: realiza operaciones requeridas en el cierre de una pantalla.
- `Eliminar`: implementa la funcionalidad de eliminar en el sistema.
- `Excel`: implementa la funcionalidad de exportar a Excel.
- `Imprimir`: implementa la funcionalidad de imprimir en el sistema.
- `Nuevo`: implementa la funcionalidad de nuevo registro en el sistema.
- `ObtenerCatalogo`: recibe el catalogo de pantallas para la implementación de las funcionalidades a las mismas.
- `ObtenerCodigoPantalla`: devuelve el código de pantalla según la recibida.
- `PantallaTipo`: devuelve el tipo de pantalla.
- `Primero`: implementa la funcionalidad de primero en el sistema.
- `Refrescar`: implementa la funcionalidad de refrescar pantalla en el sistema.
- `Salvar`: implementa la funcionalidad de salvar un registro en el sistema.
- `Siguiente`: implementa la funcionalidad de siguiente en el sistema.
- `Ultimo`: implementa la funcionalidad de último en el sistema.

TFormMenu	
#	_aplicacionPadre: TApplication
-	_eCambioPantallaActiva: TFEventoPantalla
#	_formPrincipal: TFPantallaPrincipal
<hr/>	
+	AbrirPantalla(Boolean, TFControlador, TFClasePantalla): void
#	Anterior(): void
-	application_AplicacionIniciada(): void
#	CargarParametrosConexion(): void
#	Cerrar(): void
+	CrearPantalla(TFControlador, TFClasePantalla): TFPantalla
+	ECambioPantallaActiva(): TFEventoPantalla
#	Eliminar(): void
#	EnCambioPantallaActiva(TForm): void
+	EnviarMensaje(TTFControllerMessage): void
#	Excel(): void
#	Imprimir(): void
+	Inicializar(TFControlador, TFSesion, TADODconnection, TADODdataset): void
+	Inicializar(TADODdataset, TADODconnection, TApplication): void
#	InicializarPantallas(): void
#	Nuevo(): void
#	ObtenerTipoFormPrincipal(): TFClaseFormPrincipal
#	Primero(): void
#	RealizarLogin(): TFSesion
#	RealizarLogout(): void
#	Refrescar(): void
+	RegistrarMensaje(TTFControllerMessage): void
#	Salvar(): void
#	Siguiente(): void
#	Ultimo(): void
#	VerificarPantallaAbierta(TFClasePantalla): TFPantalla

Esta clase permite la implementación del menú principal del sistema donde se detallan todas las funcionalidades del mismo. Sus propiedades y operaciones:

- \_aplicacionPadre: hace referencia a la clase de la que hereda.
- \_eCambioPantallaActiva: permite cambiar las funcionalidades en el momento que se cambie una pantalla en el sistema.
- \_formPrincipal: permite la comunicación con la pantalla principal.
- AbrirPantalla: realiza el proceso de abrir ventanas al cliente por medio del controlador.
- aplicación\_AplicacionIniciada: establece un estado para cuando la aplicación esta iniciada.
- CargarParametros: carga los parámetros del menú principal.
- CrearPantalla: crea la pantalla que se selecciona en el menú.
- ECambioPantallaActiva: establece los cambios requeridos en un tipo de pantalla.
- EnCambioPantallaActiva: establece los cambios requeridos en otro tipo de pantalla.
- EnviarMensaje: envía mensajes según lo seleccionado por el usuario.
- Inicializar: esta operación es imprescindible, ya que ejecuta las premisas de cada pantalla al ser inicializadas.
- InicializarPantalla: igual a la anterior pero con pantallas específicas.
- ObtenerTipoFormPrincipal: devuelve el tipo de “form” principal que se maneja (Pantalla o Menu)
- RealizarLogin: implementa la función de “login”.

- RealizarLogout: implementa la función de “logout”.
- RegistrarMensaje: Registra los mensajes que envía el sistema al usuario.
- VerificarPantallaAbierta: Realiza la verificación de alguna pantalla abierta (En Acción)
- Anterior: implementa la funcionalidad de anterior en el sistema.
- Cerrar: implementa la función de cerrar la pantalla en el sistema.
- Eliminar: implementa la funcionalidad de eliminar en el sistema.
- Excel: implementa la funcionalidad de exportar a Excel.
- Imprimir: implementa la funcionalidad de imprimir en el sistema.
- Nuevo: implementa la funcionalidad de nuevo registro en el sistema.
- Primero: implementa la funcionalidad de primero en el sistema.
- Refrescar: implementa la funcionalidad de refrescar pantalla en el sistema.
- Salvar: implementa la funcionalidad de salvar un registro en el sistema.
- Siguiente: implementa la funcionalidad de siguiente en el sistema.
- Ultimo: implementa la funcionalidad de último en el sistema.

```

TForm
# _controlador: TFControlador
# _ePantallaActivada: TFEventoPantalla
# _inicializado: Boolean
# _mainConnection: TADODConnection
# _mainDataset: TADODDataSet
# _pantallaMenu: TFormMenu
# _sesion: TFSesion

# Activate()
# Controlador(): TFClaseControlador
# Controlador(): Boolean
# controllerEControlerMessage(TFFControllerMessage, TObject): void
# EnPantallaActivada()
^ EPantallaActivada(): TFEventoPantalla
+ Inicializar(TADODConnection, TFControlador, TFSesion, TADODDataSet): void
+ PantallaMenu(): TFormMenu
+ ProcesarMensajeXml(TFFControllerMessage): void

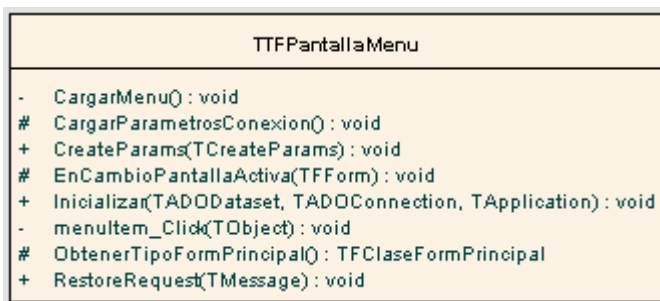
```

Es la clase padre de todas las pantallas y base del sistema. Sus propiedades y operaciones:

- \_controlador: contiene el controlador de “framework” y del sistema en sí.
- \_ePantallaActiva: valor de estado para una pantalla activa.
- \_inicializado: para determinar si el estado está en inicializado en el sistema.
- \_mainConnection: controla la conexión con el subsistema de datos.
- \_mainDataset: contiene el “Dataset” por default para cada pantalla que lo utilice.
- \_pantallaMenu: Ligue con la pantalla y el menú principal.
- \_sesion: controla los argumentos de sesión.

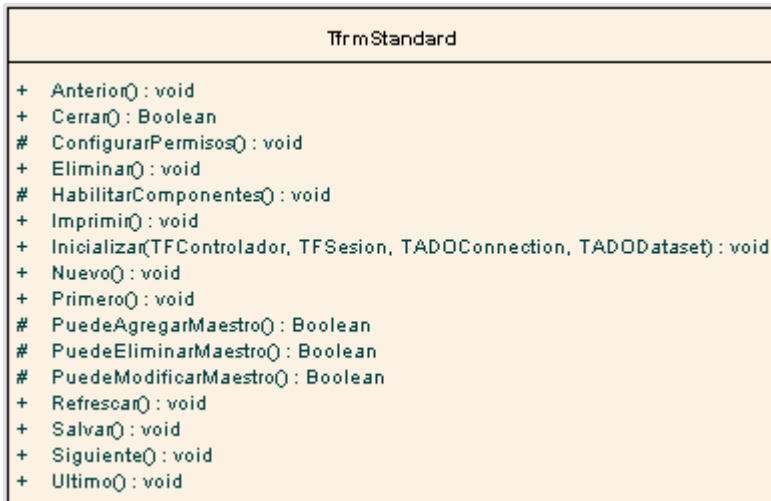


- **Activate:** realiza la función de activo en el sistema.
- **Controlador:** maneja el controlador del sistema.
- **controllerEControlerMessage:** maneja los mensajes del controlador.
- **EnPantallaActiva:** serie de funcionalidades en algunas pantallas como activas.
- **EPantallaActiva:** serie de funcionalidades en otras pantallas como activas.
- **Inicializar:** ejecuta las premisas necesarias para el arranque de la aplicación.
- **PantallaMenu:** funciones en la relación pantalla principal, menú principal.
- **ProcesarMensajeXML:** utilizado más que todo para el procesamiento del archivo de parámetros de inicialización (ini).



Esta clase comunica las acciones del menú con las de la pantalla principal. Sus propiedades y operaciones:

- **CargarMenu:** se encarga de cargar el menú en la pantalla principal.
- **CargarParametrosConexion:** establece los parámetros de conexión entre los dos componentes.
- **CreateParams:** crea los parámetros para la conexión.
- **EnCambioPantallaActiva:** establece los cambios requeridos en otro tipo de pantalla.
- **Inicializar:** inicializa las premisas necesarias dentro de la pantalla y el menú para su comunicación.
- **MenuItem\_Click:** maneja un evento para determinar la pantalla a desplegar.
- **ObtenerTipoFormPrincipal:** devuelve el tipo de “form” principal que se maneja (Pantalla o Menú)
- **RestoreRequest:** devuelve los valores por default entre los dos componentes (pantalla y menú).



Esta clase es el estándar padre para todas las demás pantallas que utiliza el usuario. Sus propiedades y operaciones:

- Anterior: implementa la funcionalidad de anterior en el sistema.
- Cerrar: implementa la función de cerrar la pantalla en el sistema.
- ConfigurarPermisos: permite establecer la configuración de permisos para ingreso al sistema.
- Eliminar: implementa la funcionalidad de eliminar en el sistema.
- HabilitarComponentes: acorde a los permisos con que cuenta el usuario esta operación le habilita componentes.
- Imprimir: implementa la funcionalidad de imprimir en el sistema.
- Inicializar: inicializa las premisas para la pantalla estándar.
- Nuevo: implementa la funcionalidad de nuevo registro en el sistema.
- Primero: implementa la funcionalidad de primero en el sistema.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.
- Refrescar: implementa la funcionalidad de refrescar pantalla en el sistema.
- Salvar: implementa la funcionalidad de salvar un registro en el sistema.
- Siguiente: implementa la funcionalidad de siguiente en el sistema.
- Ultimo: implementa la funcionalidad de último en el sistema.

TPantallaCalendarioOP
<pre> + Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void # ObtenerCatalogo() : String # ObtenerCodigoPantalla() : String + ObtenerNumerolImagen() : Integer </pre>

Esta clase implementa el calendario de órdenes de producción. Sus propiedades y operaciones:

Inicializar: inicializa las premisas para esta pantalla.

ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.

ObtenerCodigoPantalla: obtiene el código de esta pantalla.

ObtenerNumerolImagen: obtiene la imagen que representa esta pantalla en el menú principal.

TfrmStandardMant
<pre> # ConfigurarPermisos() : void + Excel() : void + Imprimir() : void </pre>

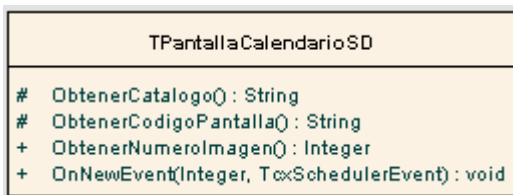
Esta clase hereda de TfrmStandard y agrega nuevas funcionalidades. Sus propiedades y operaciones:

- ConfigurarPermisos: permite establecer la configuración de permisos para ingreso al sistema.
- Excel: implementa la funcionalidad de exportar a Excel.
- Imprimir: implementa la funcionalidad de imprimir en el sistema.

TfrmStandardMantDetalle
<pre> # ConfigurarPermisos() : void + Eliminar() : void + Excel() : void + Imprimir() : void + Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void + Nuevo() : void # PuedeAgregarDetalle() : Boolean # PuedeEliminarDetalle() : Boolean # PuedeModificarDetalle() : Boolean </pre>

Esta clase hereda de TfrmStandardMant y agrega nuevas funcionalidades. Sus propiedades y operaciones:

- ConfigurarPermisos: permite establecer la configuración de permisos para ingreso al sistema.
- Eliminar: implementa la funcionalidad de eliminar en el sistema.
- Excel: implementa la funcionalidad de exportar a Excel.
- Inicializar: inicializa las premisas para la pantalla estándar con detalle.
- Nuevo: implementa la funcionalidad de nuevo registro en el sistema.
- PuedeAgregarDetalle: implementa el permiso para poder agregar en el “grid” detalle.
- PuedeModificarDetalle implementa el permiso para poder modificar en el “grid” detalle.
- PuedeEliminarDetalle: implementa el permiso para poder eliminar en el “grid” detalle.



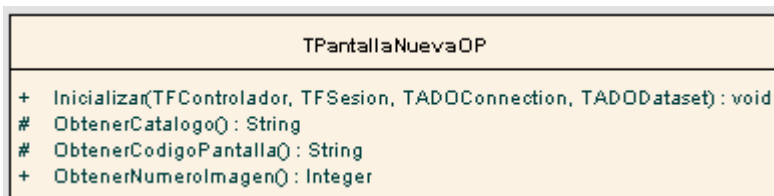
Esta clase implementa el calendario de solicitudes de despacho. Sus propiedades y operaciones:

ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.

ObtenerCodigoPantalla: obtiene el código de esta pantalla.

ObtenerNumerolImagen: obtiene la imagen que representa esta pantalla en el menú principal.

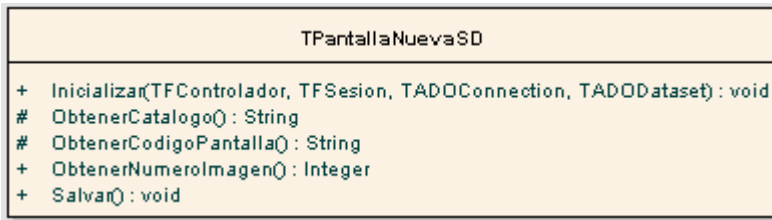
OnNewEvent: administra los eventos de cada solicitud de despacho en el calendario.



Esta clase implementa la funcionalidad para la pantalla de creación de una nueva orden de producción. Sus propiedades y operaciones:

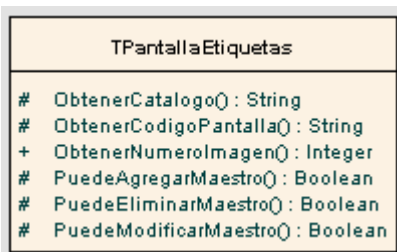
- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.

- ObtenerNumeroImagen: obtiene la imagen que representa esta pantalla en el menú principal.



Esta clase implementa la funcionalidad para la pantalla de creación de una nueva solicitud de despacho. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumeroImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- Salvar: implementa la funcionalidad de guardar registros en esta pantalla.



Esta clase es la que implementa el mantenimiento de etiquetas en el sistema. Sus propiedades y operaciones:

- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumeroImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.

TPantallaPaises	
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolImagen() : Integer
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa el mantenimiento de países en el sistema. Sus propiedades y operaciones:

- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.

TPantallaAdmi Correos	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolImagen() : Integer
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa la administración de correos de usuarios en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.

TPantallaMediosTransporte	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolImagen() : Integer
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa el mantenimiento de medios de transporte en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.

TPantallaMantTipoUnidTransp	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolImagen() : Integer
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa el mantenimiento de tipos de medios de transporte en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.

TPantallaTareasSD	
+	Inicializar(TFControlador, TFSesion, TADODconnection, TADODdataset) : void
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolmagen() : Integer
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa el mantenimiento de tareas para solicitudes de despacho en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolmagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.

TPantallaTipoTareasSD	
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolmagen() : Integer
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa el mantenimiento de tipos de medios de transporte en el sistema. Sus propiedades y operaciones:

- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolmagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.



TPantallaAdmiAreas	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolmagen() : Integer
#	PuedeAgregarDetalle() : Boolean
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarDetalle() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarDetalle() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa la administración de áreas de Ticofrut en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolmagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.
- PuedeAgregarDetalle: implementa el permiso para poder agregar en el “grid” detalle.
- PuedeModificarDetalle: implementa el permiso para poder modificar en el “grid” detalle.
- PuedeEliminarDetalle: implementa el permiso para poder eliminar en el “grid” detalle.

TPantallaAdmi Departamentos	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset) : void
#	ObtenerCatalogo() : String
#	ObtenerCodigoPantalla() : String
+	ObtenerNumerolmagen() : Integer
#	PuedeAgregarDetalle() : Boolean
#	PuedeAgregarMaestro() : Boolean
#	PuedeEliminarDetalle() : Boolean
#	PuedeEliminarMaestro() : Boolean
#	PuedeModificarDetalle() : Boolean
#	PuedeModificarMaestro() : Boolean

Esta clase es la que implementa la administración de departamentos de Ticofrut en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.

- ObtenerNumeroImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.
- PuedeAgregarDetalle: implementa el permiso para poder agregar en el “grid” detalle.
- PuedeModificarDetalle: implementa el permiso para poder modificar en el “grid” detalle.
- PuedeEliminarDetalle: implementa el permiso para poder eliminar en el “grid” detalle.

```

TPantallaClienConsig
# HabilitarComponentes(): void
+ Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset): void
+ Nuevo(): void
# ObtenerCatalogo(): String
# ObtenerCodigoPantalla(): String
+ ObtenerNumeroImagen(): Integer
# PuedeAgregarDetalle(): Boolean
# PuedeAgregarMaestro(): Boolean
# PuedeEliminarDetalle(): Boolean
# PuedeEliminarMaestro(): Boolean
# PuedeModificarDetalle(): Boolean
# PuedeModificarMaestro(): Boolean
+ Salvar(): void

```

Esta clase es la que implementa el mantenimiento de clientes y consignatarios en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- HabilitarComponentes: permite habilitar controles de edición dentro de esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumeroImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.
- PuedeAgregarDetalle: implementa el permiso para poder agregar en el “grid” detalle.
- PuedeModificarDetalle: implementa el permiso para poder modificar en el “grid” detalle.
- PuedeEliminarDetalle: implementa el permiso para poder eliminar en el “grid” detalle.
- Nuevo: implementa la funcionalidad de nuevo registro en el sistema.
- Salvar: implementa la funcionalidad de salvar un registro en el sistema.

TPantallaNavieras	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset): void
#	ObtenerCatalogo(): String
#	ObtenerCodigoPantalla(): String
+	ObtenerNumerolmagen(): Integer
#	PuedeAgregarDetalle(): Boolean
#	PuedeAgregarMaestro(): Boolean
#	PuedeEliminarDetalle(): Boolean
#	PuedeEliminarMaestro(): Boolean
#	PuedeModificarDetalle(): Boolean
#	PuedeModificarMaestro(): Boolean

Esta clase es la que implementa el mantenimiento de navieras en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.
- ObtenerNumerolmagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.
- PuedeAgregarDetalle: implementa el permiso para poder agregar en el “grid” detalle.
- PuedeModificarDetalle: implementa el permiso para poder modificar en el “grid” detalle.
- PuedeEliminarDetalle: implementa el permiso para poder eliminar en el “grid” detalle.

TPantallaSD	
+	Inicializar(TFControlador, TFSesion, TADOCconnection, TADODataset): void
#	ObtenerCatalogo(): String
#	ObtenerCodigoPantalla(): String
+	ObtenerNumerolmagen(): Integer
#	PuedeAgregarDetalle(): Boolean
#	PuedeAgregarMaestro(): Boolean
#	PuedeEliminarDetalle(): Boolean
#	PuedeEliminarMaestro(): Boolean
#	PuedeModificarDetalle(): Boolean
#	PuedeModificarMaestro(): Boolean

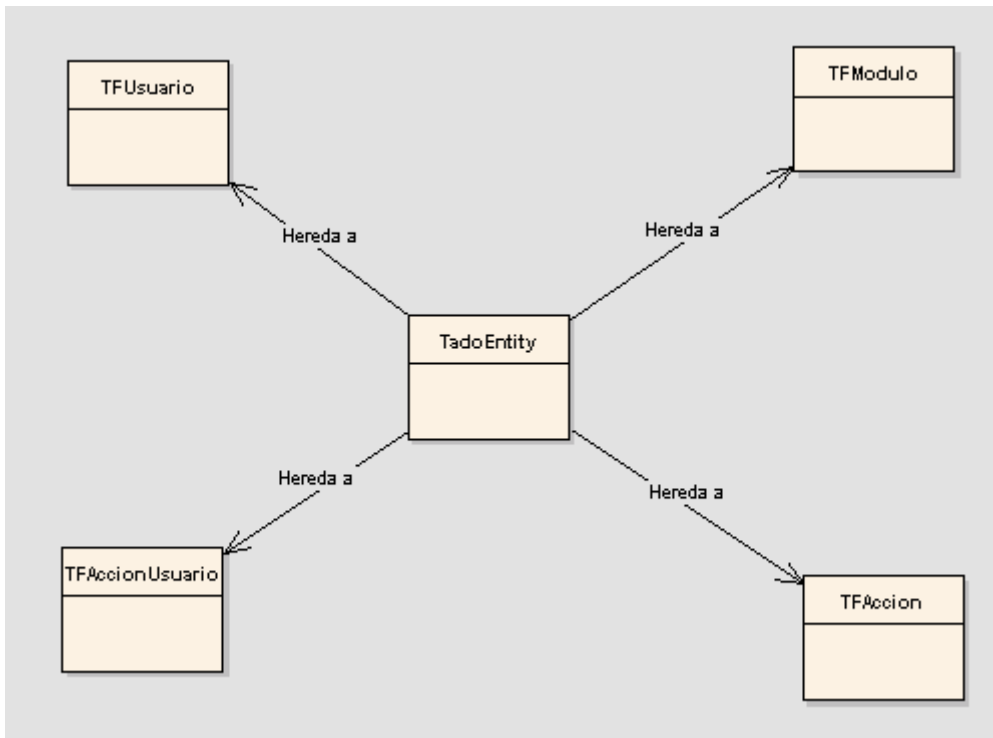
Esta clase es la que implementa el mantenimiento de solicitudes de despacho en el sistema. Sus propiedades y operaciones:

- Inicializar: inicializa las premisas para esta pantalla.
- ObtenerCatalogo: obtiene el catalogo de funcionalidades en esta pantalla.
- ObtenerCodigoPantalla: obtiene el código de esta pantalla.

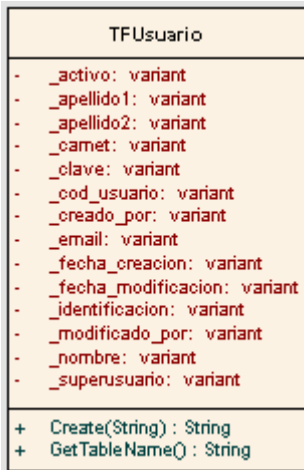
- ObtenerNumeroImagen: obtiene la imagen que representa esta pantalla en el menú principal.
- PuedeAgregarMaestro: implementa el permiso para poder agregar en el “grid” maestro.
- PuedeModificarMaestro: implementa el permiso para poder modificar en el “grid” maestro.
- PuedeEliminarMaestro: implementa el permiso para poder eliminar en el “grid” maestro.
- PuedeAgregarDetalle: implementa el permiso para poder agregar en el “grid” detalle.
- PuedeModificarDetalle: implementa el permiso para poder modificar en el “grid” detalle.
- PuedeEliminarDetalle: implementa el permiso para poder eliminar en el “grid” detalle.

#### Subdiagrama 2:

Este Subdiagrama representa las clases referentes a la seguridad del sistema, donde se manejan los usuarios, las acciones de los mismos y los módulos de acceso a las acciones.



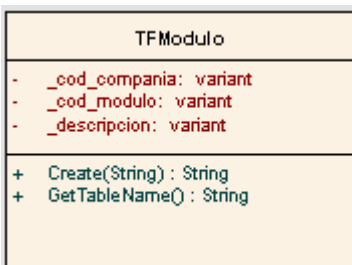
Clases:



Esta clase implementa la tabla de usuarios en el sistema y accede a sus datos.

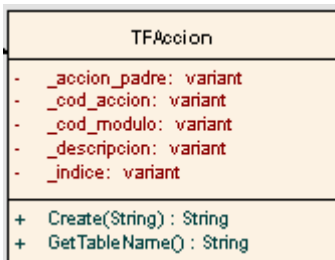
Sus propiedades y operaciones:

- \_activo: determina si el usuario es activo en el sistema.
- \_apellido1: Primer apellido del usuario.
- \_apellido2: Segundo apellido del usuario.
- \_carnet: Carnet del usuario.
- \_clave: Clave del usuario.
- \_cod\_usuario: código de usuario.
- \_creado\_por: responsable de creación de usuario.
- \_email: correo electrónico del usuario.
- \_fecha\_creacion: fecha de creación del usuario.
- \_fecha\_modificacion: fecha de modificación del usuario.
- \_identificacion: identificación del usuario (cédula).
- \_modificado\_por: responsable de modificación del usuario.
- \_nombre: nombre del usuario.
- \_superusuario: Si es super usuario.
- Create: creador de la entidad.
- GetTableName: devuelve el nombre de la tabla.



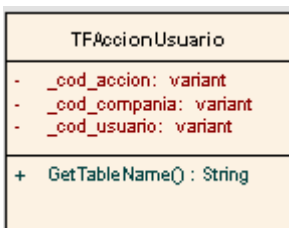
Esta clase implementa la tabla de módulos en el sistema y accede a sus datos.  
Sus propiedades y operaciones:

- `_cod_compania`: código de la compañía.
- `_cod_modulo`: código del modulo.
- `_descripcion`: descripción del modulo.
- `Create`: creador de la entidad.
- `GetTableName`: devuelve el nombre de la tabla.



Esta clase implementa la tabla de acciones en el sistema y accede a sus datos.  
Sus propiedades y operaciones:

- `_accion_padre`: acción padre de la acción.
- `_cod_accion`: código de acción.
- `_cod_modulo`: código del modulo.
- `_descripcion`: descripción de la acción.
- `_indice`: índice de la acción.
- `Create`: creador de la entidad.
- `GetTableName`: devuelve el nombre de la tabla.



Esta clase implementa la tabla de acciones por usuario en el sistema y accede a sus datos.  
Sus propiedades y operaciones:

- `_cod_accion`: código de acción.
- `_cod_compania`: código de compañía.

- `_cod_usuario`: código de usuario.
- `GetTableName`: devuelve el nombre de la tabla.

TadoEntity	
#	<code>_adoDataset: TDataSet</code>
#	<code>_canLoad: boolean</code>
#	<code>_currentOperation: TADOEntityOperation</code>
#	<code>_loadNulls: boolean</code>
#	<code>_procedureName: String</code>
#	<code>_storedProcedure: TADOStoredProc</code>
#	<code>_tableName: String</code>
#	<code>BeforeInsert(): void</code>
+	<code>GetKeyFieldsNames(): TStringList</code>
+	<code>GetOriginalValues(): variant</code>
#	<code>GetProcedureName(): String</code>
+	<code>GetTable(): String</code>
#	<code>GetTableName(): String</code>
+	<code>LoadFromDataSet(boolean, TDataSet): void</code>
+	<code>LoadFromStoredProcedure(TADOStoredProc): void</code>
#	<code>OnDeserialize(PPropInfo): void</code>
#	<code>OnSerialize(PPropInfo): void</code>
+	<code>SaveFieldsToDataSet(TDataSet): void</code>
+	<code>SaveToDataSet(TDataSet): void</code>
+	<code>SetOriginalValues(): boolean</code>

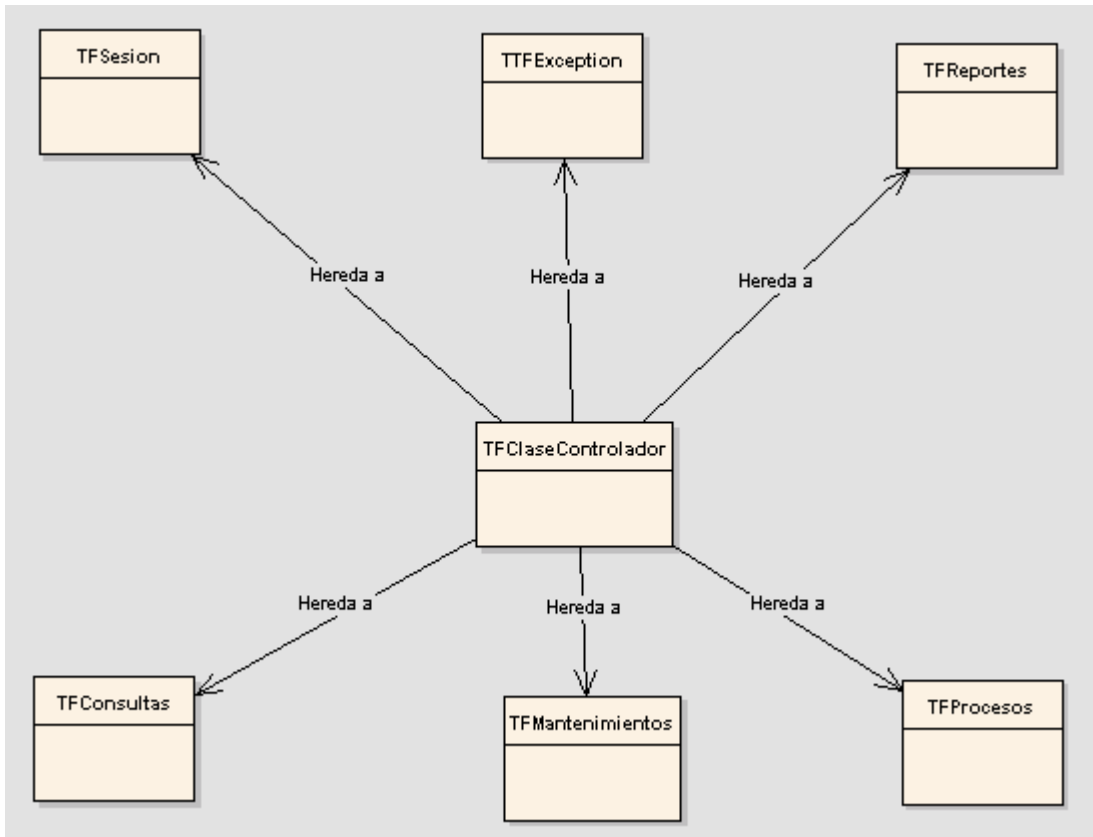
Esta clase implementa una entidad para el manejo de las tablas de seguridad en el sistema. Sus propiedades y operaciones:

- `_adoDataSet`: implementa un dataset.
- `_canLoad`: bandera para la carga de datos.
- `_currentOperation`: operación de la entidad.
- `_loadNulls`: Si obtiene valores nulos.
- `_procedureName`: almacena el nombre del procedimiento.
- `_storedProcedure`: almacena un procedimiento.
- `_tableName`: almacena el nombre de la tabla.
- `BeforeInsert`: procesos después de insertar.
- `GetKeyFieldsNames`: obtiene llaves primarias.
- `GetOriginalValues`: obtiene los valores de la entidad.
- `GetProcedureName`: obtiene el nombre del procedimiento.
- `GetTable`: obtiene la tabla.
- `GetTableName`: obtiene el nombre de la tabla.
- `LoadFromDataSet`: carga del dataset.
- `LoadFromStoredProcedure`: carga del procedimiento almacenado.
- `OnDeserialize`: continuo.
- `OnSerialize`: variante.
- `SaveFieldsToDataSet`: salvar campos en el registro.
- `SaveToDataSet`: almacenar al dataset.

- SetOriginalValues: establecer parámetros originales.

### Subdiagrama 3:

Este Subdiagrama muestra las clases administradas por el controlador, y al mismo controlador, el cual maneja lo respectivo a sesiones, excepciones, consultas, reportes, mantenimientos y procesos.





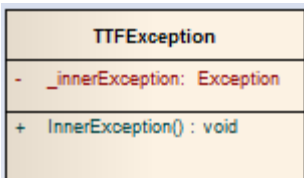
Clases:



Esta clase implementa los valores de la sesión.

Sus propiedades y operaciones:

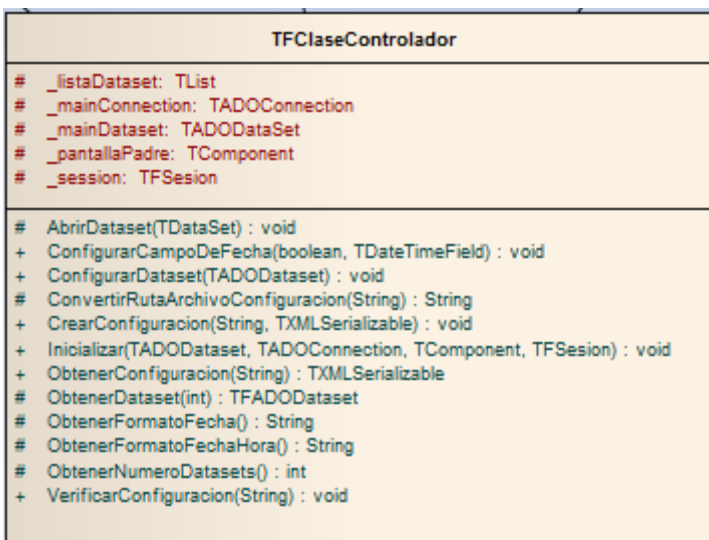
- \_configuracion: Compañía en la que se va a hacer la sesión.
- \_usuario: usuario de la sesión.



Esta clase implementa las excepciones en el sistema.

Sus propiedades y operaciones:

- \_innerException: variable de la excepción.
- InnerException: manejo de la excepción.



Esta clase realiza el control de los procesos en la mayor parte del sistema.

Sus propiedades y operaciones:

- `_listaDataset`: maneja una lista de dataset en el sistema.
- `_mainConnection`: controla la conexión a la base de datos.
- `_mainDataset`: controla el dataset padre en cada pantalla.
- `_pantallaPadre`: controla la pantalla padre (TFForm).
- `_session`: Controla la sesión.
- `AbrirDataset`: procedimiento para la apertura de un dataset para edición.
- `ConfigurarCampoDeFecha`: configura el campo fecha.
- `ConfigurarDataset`: configura el dataset.
- `ConvertirRutaArchivoConfiguracion`: para el manejo de la ruta del archivo de configuración.
- `CrearConfiguracion`: crea la configuración a partir del archivo.
- `Inicializar`: inicializa las premisas necesarias para el controlador.
- `ObtenerConfiguracion`: devuelve la configuración del archivo.
- `ObtenerDataset`: devuelve un dataset específico.
- `ObtenerFormatoFecha`: devuelve la fecha.
- `ObtenerFormatoFechaHora`: devuelve la fecha con la hora.
- `ObtenerNumeroDataSet`: devuelve el numero de un dataset específico.
- `VerificarConfiguracion`: cheque la configuración del archivo de configuración.

TFReportes	
+	<code>AgregarDataset(Boolean, boolean, TFADODataset, TFADODataset) : void</code>
+	<code>Anterior() : void</code>
+	<code>CambiosPendientes() : boolean</code>
+	<code>Cancelar() : void</code>
+	<code>Cerrar() : void</code>
+	<code>Eliminar() : void</code>
+	<code>Excel() : void</code>
+	<code>Imprimir() : void</code>
+	<code>Nuevo() : void</code>
+	<code>PDF() : void</code>
+	<code>Primero() : void</code>
+	<code>Refrescar() : void</code>
+	<code>Salvar() : void</code>
+	<code>SalvarDataset(TFADODataset) : void</code>
+	<code>Siguiente() : void</code>
+	<code>Ultimo() : void</code>
+	<code>VerificarPermiso(String) : boolean</code>
+	<code>Word() : void</code>

Clase para el manejo de los reportes por parte del controlador.

Sus propiedades y operaciones:

- `AgregarDataset`: agrega dataset al inicio de una gestión.
- `Anterior`: para el manejo del anterior registro.
- `CambiosPendientes`: comprueba si existen cambios por parte del usuario.

- Cancelar: para el manejo de cancelar acciones.
- Cerrar: para el manejo de cerrar.
- Eliminar: para el manejo de eliminar registro.
- Excel: par el manejo de exportar a Excel.
- Imprimir: par el manejo de imprimir.
- Nuevo: para el manejo de nuevo registro.
- PDF: para el manejo de exportar a PDF.
- Primero: para el manejo del primer registro.
- Refreshar: para el manejo de refreshar un registro.
- Salvar: para el manejo de guardar el registro.
- SalvarDataset: almacena el dataset en memoria.
- Siguiente: para el manejo del siguiente registro.
- Ultimo: para el manejo del último registro.
- VerificarPermiso: verifica el acceso de un usuario en determinada acción.
- Word: para el manejo de exportar a Word.

TFMantenimientos	
+	AgregarDataset(Boolean, boolean, TFADODataset, TFADODataset) : void
+	Anterior() : void
+	CambiosPendientes() : boolean
+	Cancelar() : void
+	Cerrar() : void
+	Eliminar() : void
+	Excel() : void
+	Imprimir() : void
+	Nuevo() : void
+	PDF() : void
+	Primero() : void
+	Refreshar() : void
+	Salvar() : void
+	SalvarDataset(TFADODataset) : void
+	Siguiente() : void
+	Ultimo() : void
+	VerificarPermiso(String) : boolean
+	Word() : void

Clase para el manejo de los reportes por parte del controlador.

Sus propiedades y operaciones:

- AgregarDataset: agrega dataset al inicio de una gestión.
- Anterior: para el manejo del anterior registro.
- CambiosPendientes: comprueba si existen cambios por parte del usuario.
- Cancelar: para el manejo de cancelar acciones.
- Cerrar: para el manejo de cerrar.
- Eliminar: para el manejo de eliminar registro.
- Excel: par el manejo de exportar a Excel.
- Imprimir: par el manejo de imprimir.
- Nuevo: para el manejo de nuevo registro.

- PDF: para el manejo de exportar a PDF.
- Primero: para el manejo del primer registro.
- Refrescar: para el manejo de refrescar un registro.
- Salvar: para el manejo de guardar el registro.
- SalvarDataset: almacena el dataset en memoria.
- Siguiente: para el manejo del siguiente registro.
- Ultimo: para el manejo del último registro.
- VerificarPermiso: verifica el acceso de un usuario en determinada acción.
- Word: para el manejo de exportar a Word.

TFProcesos	
+	AgregarDataset(Boolean, boolean, TFADODataset, TFADODataset) : void
+	Anterior() : void
+	CambiosPendientes() : boolean
+	Cancelar() : void
+	Cerrar() : void
+	Eliminar() : void
+	Excel() : void
+	Imprimir() : void
+	Nuevo() : void
+	PDF() : void
+	Primero() : void
+	Refrescar() : void
+	Salvar() : void
+	SalvarDataset(TFADODataset) : void
+	Siguiente() : void
+	Ultimo() : void
+	VerificarPermiso(String) : boolean
+	Word() : void

Clase para el manejo de los reportes por parte del controlador.  
Sus propiedades y operaciones:

- AgregarDataset: agrega dataset al inicio de una gestión.
- Anterior: para el manejo del anterior registro.
- CambiosPendientes: comprueba si existen cambios por parte del usuario.
- Cancelar: para el manejo de cancelar acciones.
- Cerrar: para el manejo de cerrar.
- Eliminar: para el manejo de eliminar registro.
- Excel: par el manejo de exportar a Excel.
- Imprimir: par el manejo de imprimir.
- Nuevo: para el manejo de nuevo registro.
- PDF: para el manejo de exportar a PDF.
- Primero: para el manejo del primer registro.
- Refrescar: para el manejo de refrescar un registro.
- Salvar: para el manejo de guardar el registro.
- SalvarDataset: almacena el dataset en memoria.

- Siguiente: para el manejo del siguiente registro.
- Ultimo: para el manejo del último registro.
- VerificarPermiso: verifica el acceso de un usuario en determinada acción.
- Word: para el manejo de exportar a Word.

TFConsultas	
+	AgregarDataset(Boolean, boolean, TFADODataset, TFADODataset) : void
+	Anterior() : void
+	CambiosPendientes() : boolean
+	Cancelar() : void
+	Cerrar() : void
+	Eliminar() : void
+	Excel() : void
+	Imprimir() : void
+	Nuevo() : void
+	PDF() : void
+	Primero() : void
+	Refresh() : void
+	Salvar() : void
+	SalvarDataset(TFADODataset) : void
+	Siguiente() : void
+	Ultimo() : void
+	VerificarPermiso(String) : boolean
+	Word() : void

Clase para el manejo de los reportes por parte del controlador.

Sus propiedades y operaciones:

- AgregarDataset: agrega dataset al inicio de una gestión.
- Anterior: para el manejo del anterior registro.
- CambiosPendientes: comprueba si existen cambios por parte del usuario.
- Cancelar: para el manejo de cancelar acciones.
- Cerrar: para el manejo de cerrar.
- Eliminar: para el manejo de eliminar registro.
- Excel: par el manejo de exportar a Excel.
- Imprimir: par el manejo de imprimir.
- Nuevo: para el manejo de nuevo registro.
- PDF: para el manejo de exportar a PDF.
- Primero: para el manejo del primer registro.
- Refresh: para el manejo de refrescar un registro.
- Salvar: para el manejo de guardar el registro.
- SalvarDataset: almacena el dataset en memoria.
- Siguiente: para el manejo del siguiente registro.
- Ultimo: para el manejo del último registro.
- VerificarPermiso: verifica el acceso de un usuario en determinada acción.
- Word: para el manejo de exportar a Word.

### *Especificación de componentes del sistema*

Los componentes a utilizar en el sistema son los que brinda la herramienta Borland Delphi versión 7.0, esto aparte de la adquisición por parte de TicoFrut S.A. de un paquete de componentes licenciados, y que son compatibles con la herramienta de programación la cual se llama Developer Express.VCL. Este paquete de componentes, brinda cierta cantidad de componentes que permiten mejorar el aspecto de las aplicaciones y le adhieren muchas funcionalidades, entre las más importantes y utilizadas en TicoFrut están:

ExpressQuantumGrid 6: este modulo de componentes, se utiliza para el manejo de información en la base de datos por medio de un grid. Este grid permite el despliegue de las columnas de las tablas que el usuario desee. También permite la gestión de los datos que son desplegados en el mismo y permite visualizar cualquier dato que sea almacenado en la base de datos (Incluyendo imágenes). Este componente facilita enormemente el acceso a los datos en la base de datos, ya que solo se necesita crear la conexión, asignar un datagrid con la consulta que se quiere generar y asignarle dicha consulta al componente gris.

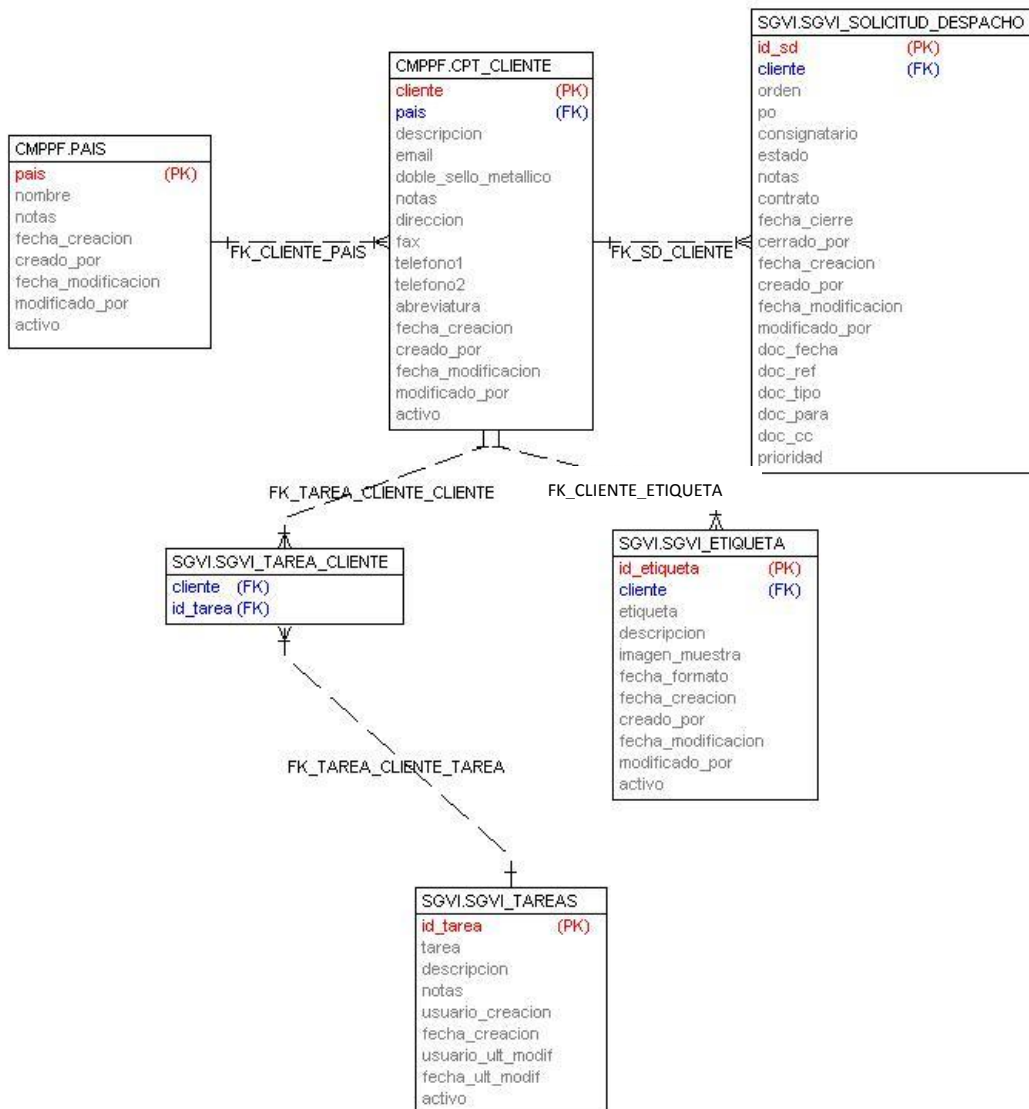
ExpressScheduler 2: este modulo permite la generación y gestión de calendarios o diagramas de Gantt sobre procesos o actividades que se quieran mostrar. Para el sistema de gestión de ventas e inventario, se utilizara para la calendarización de las solicitudes de despacho y las ordenes de producción.

Por otro lado existe el componente DATASET que es el mayormente utilizado para el acceso a la base de datos. En este sistema todas las pantallas que requieran información de la base de datos cuentan con este componente el cual es manejado por el controlador en la interfaz de TFDATOS

### Modelo de la base de datos

La base de datos que se maneja es Oracle 10g. Esta base a utilizar en el sistema de gestión de ventas e inventario está proyectada en 3 esquemas, los cuales son: CMPPF, SGVI y SEGURIDAD.

El modelo de la base de datos se representa en pequeños subdiagramas donde se da una explicación de cada uno de ellos y las relaciones que presentan, y en una especificación de las tablas de manera que se pueda comprender su funcionamiento.



## **Relaciones:**

FK\_CLIENTE\_PAIS: Cada cliente tiene un país de origen.

FK\_CLIENTE\_SD: Cada cliente tiene una cantidad de solicitudes de despacho.

FK\_TAREA\_CLIENTE\_CLIENTE: relaciona cliente con una determinada cantidad de tareas.

FK\_TAREA\_CLIENTE\_TAREA: relaciona una tarea a una determinada cantidad de clientes.

FK\_CLIENTE\_ETIQUETA: establece etiquetas para un cliente.

## **Tablas:**

Tabla cpt\_cliente:

Almacena los clientes y consignatarios compradores de producto a Ticofrut.

Tabla país:

Almacena países que son utilizados para asignarlos a clientes, consignatarios o navieras en el sistema.

Tabla sgvi\_solicitud\_despacho:

Almacena la información referente a una solicitud de despacho en el sistema.

Tabla sgvi\_etiqueta:

Almacena las etiquetas respectivas a cada cliente en el sistema.

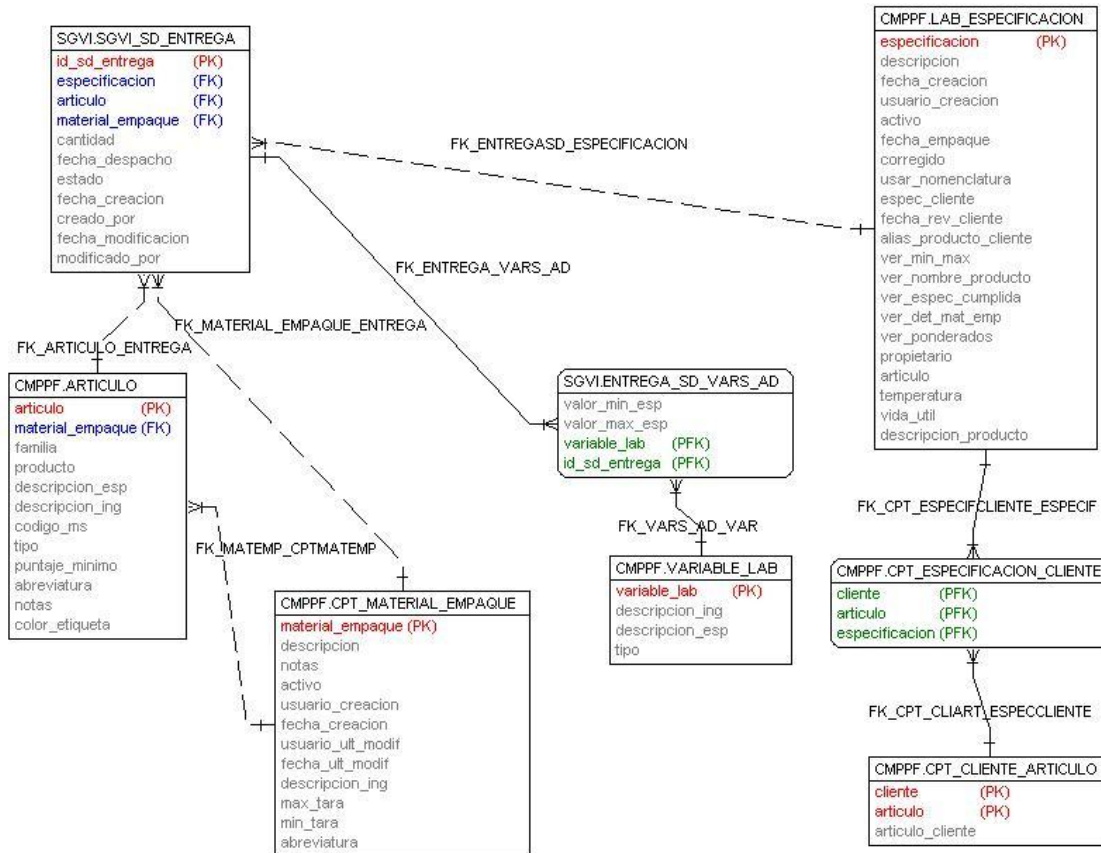
Tabla sgvi\_tarea\_cliente:

Permite ligar clientes a tareas determinadas para las solicitudes de despacho en el sistema.

Tabla sgvi\_tareas:

Permite ligar una tarea específica a una solicitud de despacho en el sistema.





## Relaciones:

FK\_ENTREGASD\_ESPECIFICACION: relaciona una especificación a una entrega determinada.

FK\_ENTREGASD\_VARS\_AD: relaciona una entrega con una determinada cantidad de variables de laboratorio.

FK\_VARS\_AD\_VAR: relaciona una variable a una entrega de producto.

FK\_ARTICULO\_ENTREGA: relaciona el artículo de una entrega.

FK\_MATERIAL\_EMPAQUE\_ENTREGA: relaciona una entrega con el material de empaque de producto seleccionado.

FK\_MATEMP\_CPTMATEMP: relaciona un producto con su material de empaque.

FK\_CPTESPECIFCLIENTE\_ESPECIF: relaciona un cliente con su especificación.

FK\_CPT\_CLIART\_ESPECCLIENTE: relaciona una especificación a un cliente.

## **Tablas:**

Tabla `cpt_cliente_articulo`:

Permite asignar distintos artículos a un determinado cliente.

Tabla `cpt_especificacion_cliente`:

Permite asignar la especificación a un artículo de un determinado cliente.

Tabla `sgvi_entrega_sd_vars_ad`:

Relaciona las variables de laboratorio con una determinada entrega de una solicitud de despacho.

Tabla `variable_lab`:

Almacena las variables de laboratorio en el sistema.

Tabla `cpt_material_empaque`:

Almacena los materiales de empaque para los productos en el sistema.

Tabla `sgvi_sd_entrega`:

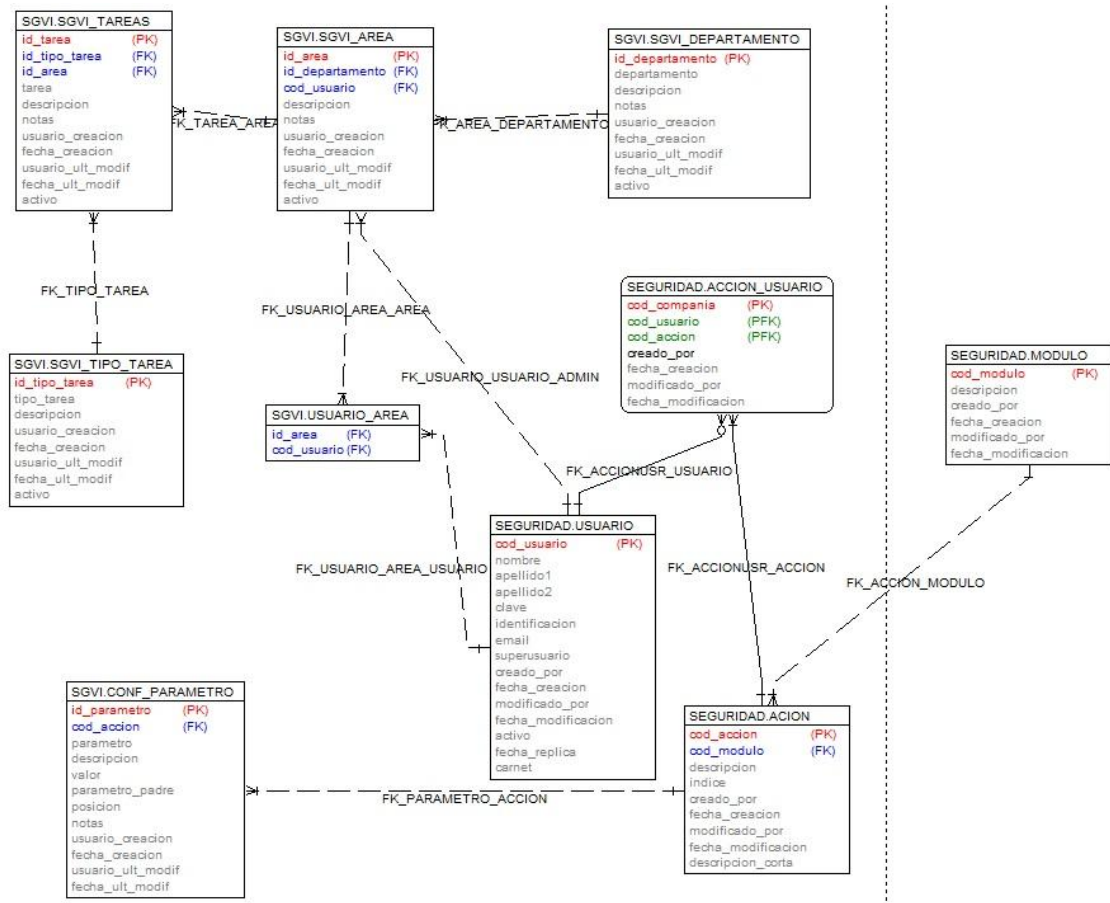
Almacena las entregas de las solicitudes de despacho en el sistema.

Tabla `articulo`:

Almacena los distintos artículos o producto que produce TicoFrut.

Tabla `lab_especificacion`:

Almacena las distintas especificaciones con que piden los clientes producto a TicoFrut.



**Relaciones:**

- FK\_TAREA\_AREA: relaciona una tarea específica a un área.
- FK\_AREA\_DEPARTAMENTO: relaciona aéreas a un departamento específico.
- FK\_TIPO\_TAREA: Relaciona una tarea con su tipo de tarea específico.
- FK\_USUARIO\_AREA\_AREA: relaciona el usuario responsable a un área específica.
- FK\_USUARIO\_USUARIO\_ADMIN: relaciona un área con su usuario administrador.
- FK\_USUARIO\_AREA\_USUARIO: relaciona aéreas con usuarios del sistema.
- FK\_ACCIONUSR\_USUARIO: relaciona un usuario con sus acciones en el sistema.
- FK\_ACCIONUSR\_ACCION: relaciona acciones de usuario con acciones del sistema.
- FK\_ACCION\_MODULO: relaciona distintas acciones a un modulo.
- FK\_PARAMETRO\_ACCION: relaciona los parámetros de la aplicación con las acciones del sistema.

## **Tablas:**

Tabla sgvi\_tareas:

Almacenas las tareas o elementos que son pedidos por el cliente cuando solicita un pedido de producto

Tabla sgvi\_tipo\_tarea:

Almacena los tipos de tareas que se pueden incluir en el pedido de una solicitud de despacho.

Tabla sgvi\_area:

Almacena las áreas de los departamentos de Ticofrut en el sistema.

Tabla sgvi\_departamento:

Almacena los departamentos de Ticofrut en el sistema.

Tabla sgvi\_usuario\_area:

Determina los usuarios responsables de las áreas en el sistema.

Tabla usuario:

Almacena los usuarios del sistema.

Tabla acción\_usuario:

Relaciona los usuarios del sistema con las acciones que pueden ejercer en el mismo.

Tabla conf\_parametro:

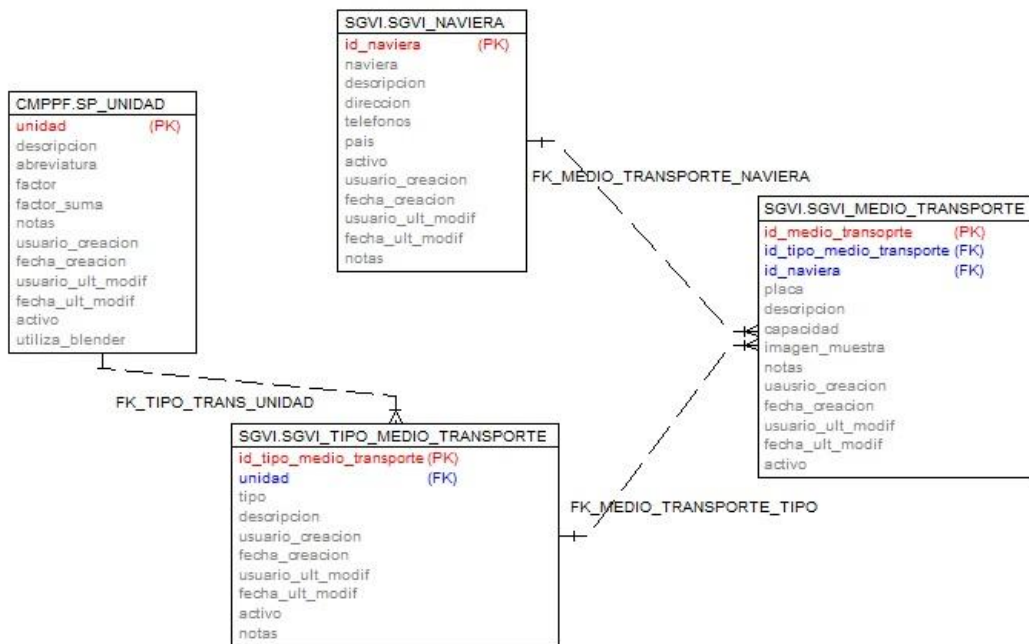
Almacena la configuración de parámetros para las acciones de los usuarios dentro del sistema.

Tabla acción:

Almacenas las acciones que se pueden ejercer dentro del sistema.

Tabla modulo:

Almacena módulos de los distintos sistemas utilizados en Ticofrut.



**Relaciones:**

FK\_TIPO\_TRANS\_UNIDAD: relaciona el tipo de unidad de carga de un medio de transporte con un determinado tipo de medio de transporte.

FK\_MEDIO\_TRANSPORTE\_NAVIERA: relaciona un medio de transporte con su respectiva naviera.

FK\_MEDIO\_TRANSPORTE\_TIPO: relaciona un medio de transporte con su tipo.

**Tablas:**

Tabla sp\_unidad:

Almacena las unidades de peso para el transporte en el sistema

Tabla sgvi\_naviera:

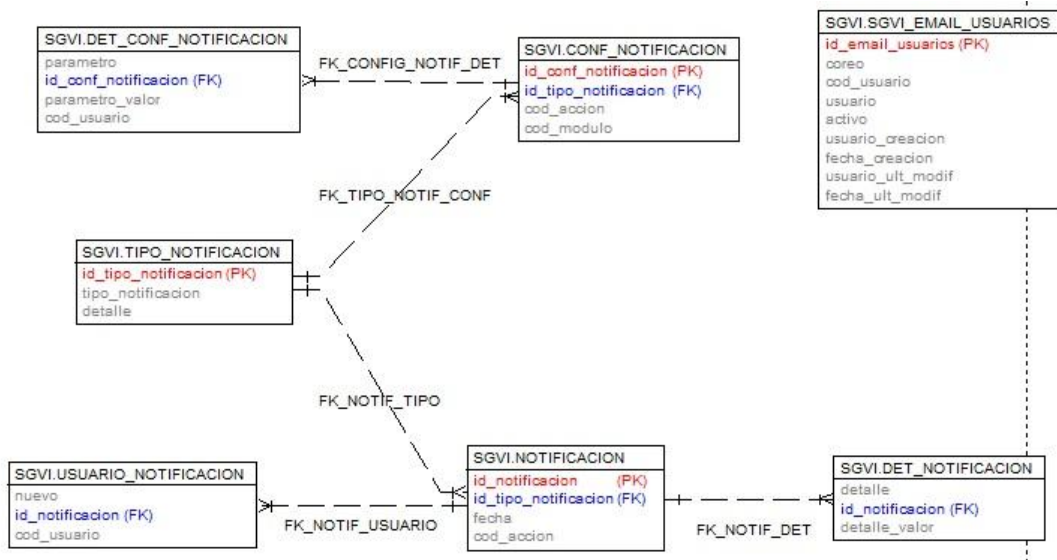
Almacena las navieras que transportan producto a los clientes compradores en el sistema.

Tabla sgvi\_tipo\_medio\_transporte:

Almacena los tipos de medios de transporte para el traslado de producto en el sistema.

Tabla `sgvi_medio_transporte`:

Almacena los medios de transporte utilizados para el traslado de producto a los clientes.



## Relaciones:

**FK\_CONFIG\_NOTIF\_DET**: relaciona el detalle de la configuración de notificaciones, con la configuración de notificaciones.

**FK\_TIPO\_NOTIF\_CONF**: relaciona el tipo de notificación con la configuración de la notificación.

**FK\_NOTIF\_TIPO**: relaciona una notificación con su tipo correspondiente.

**FK\_NOTIF\_USUARIO**: relaciona un usuario con sus notificaciones correspondientes.

**FK\_NOTIF\_DET**: relaciona el detalle de la notificación con su notificación respectiva.

## Tablas:

Tabla `det_conf_notificacion`:

Especifica el detalle en cuanto a la configuración de la notificación en el sistema.

Tabla conf\_notificacion:

Maneja la configuración de las notificaciones en el sistema.

Tabla sgvi\_email\_usuarios:

Almacena los correos electrónicos de los usuarios del sistema para las notificaciones.

Tabla tipo\_notificacion:

Especifica el tipo de notificación para los usuarios en el sistema.

Tabla usuario\_notificacion:

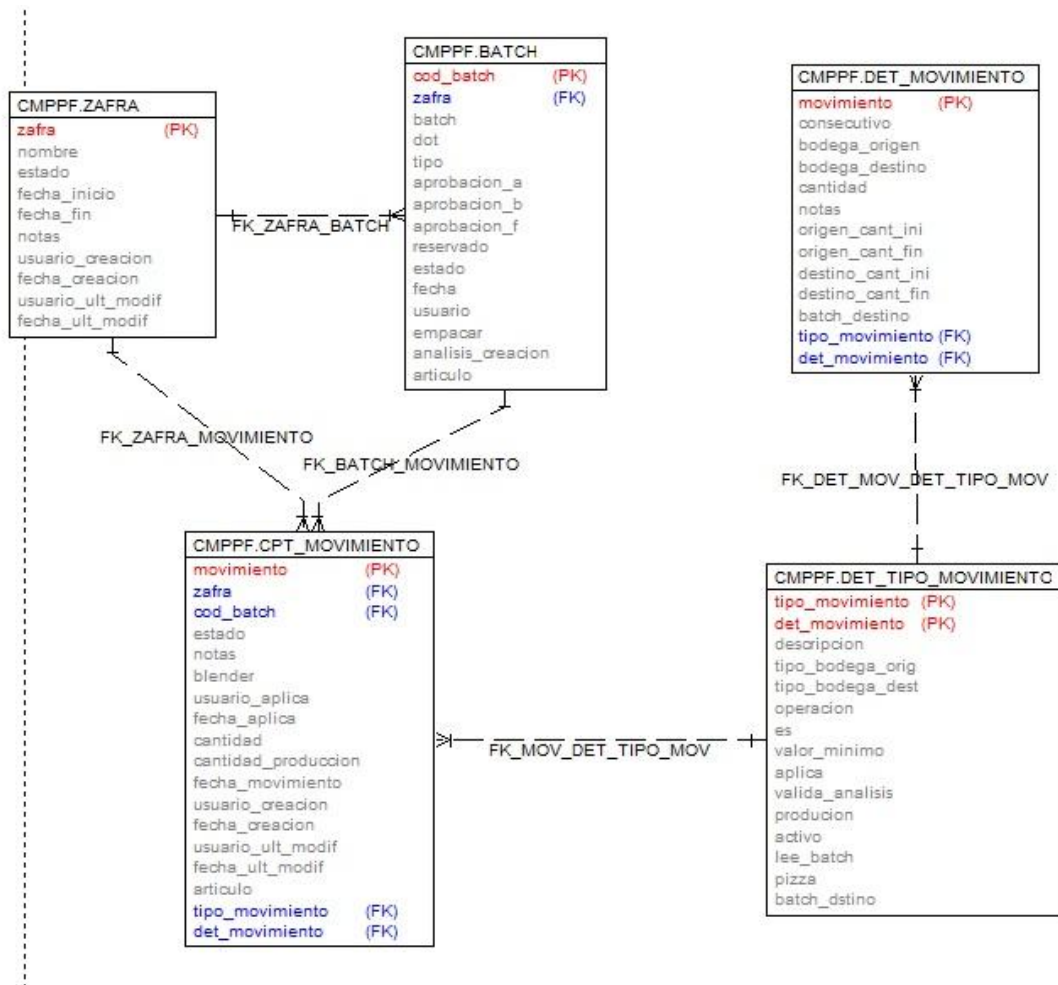
Almacena las notificaciones por usuario en el sistema.

Tabla notificación:

Almacena las notificaciones que se les envían a los usuarios en el sistema.

Tabla det\_notificacion:

Especifica el detalle de las notificaciones en el sistema.



## Relaciones:

FK\_ZAFRA\_BATCH: relaciona un Batch con su zafra de producción respectiva.

FK\_ZAFRA\_MOVIMIENTO: relaciona un determinado movimiento en bodega con la zafra correspondiente al producto.

FK\_BATCH\_MOVIMIENTO: relaciona un movimiento en bodega con el Batch de producto.

FK\_MOV\_DET\_TIPO\_MOV: relaciona un movimiento con su tipo de movimiento.

FK\_DET\_MOV\_DET\_TIPO\_MOV: relaciona un tipo de movimiento con su correspondiente detalle.



## **Tablas:**

Tabla zafra:

Almacena las zafra realizadas por año en Ticofrut.

Tabla movimiento:

Almacena los movimientos de producto en inventario dentro del sistema.

Tabla Batch:

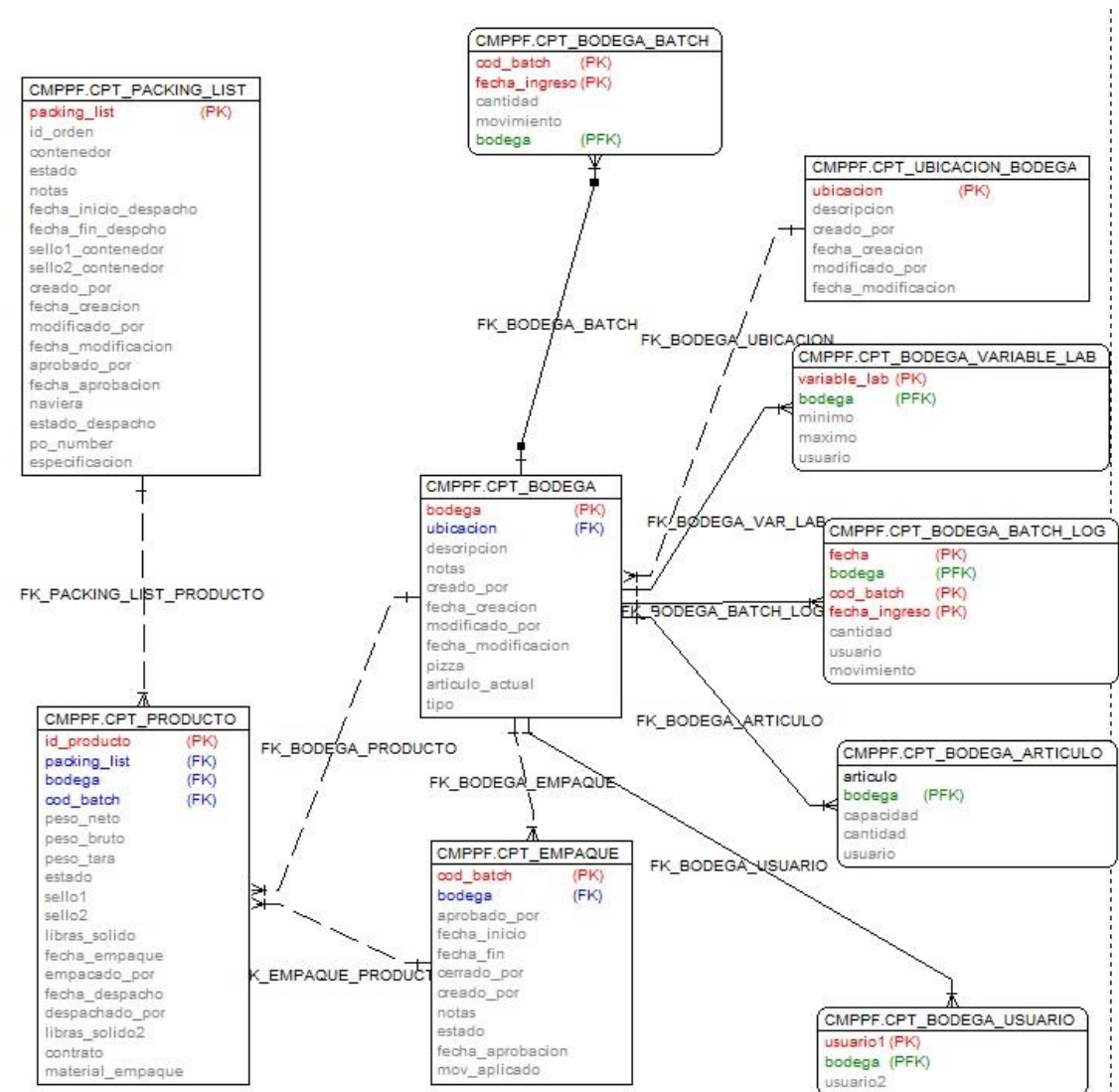
Almacena los Batch de producto en el sistema.

Tabla det\_movimiento:

Especifica detalles del movimiento dentro del inventario.

Tabla det\_tipo\_movimiento:

Especifica el detalle del tipo de movimiento en el sistema.



## Relaciones:

FK\_PACKING\_LIST\_PRODUCTO: relaciona productos a un correspondiente Packing List.

FK\_EMPAQUE\_PRODUCTO: relaciona un determinado producto con su empaque.

FK\_BODEGA\_PRODUCTO: relaciona un producto con su bodega de almacenaje correspondiente.

FK\_BODEGA\_EMPAQUE: relaciona una bodega con los materiales de empaque del producto.

FK\_BODEGA\_USUARIO: relaciona las bodegas con sus usuarios responsables.

FK\_BODEGA\_ARTICULO: relaciona bodegas con artículos de las ventas.

FK\_BODEGA\_BATCH\_LOG: relaciona una bodega con sus accesos correspondientes.

FK\_BODEGA\_VAR\_LAB: asocia las variables de laboratorio con las bodegas que almacenan el producto correspondiente.

FK\_BODEGA\_UBICACION: determina la ubicación de una determinada bodega.

FK\_BODEGA\_BATCH: asocia un Batch a una bodega determinada.

## **Tablas:**

Tabla cpt\_bodega\_batch:

Relaciona un Batch con la bodega a que pertenece en el sistema.

Tabla cpt\_bodega:

Almacena las bodegas de almacenaje de producto en el sistema.

Tabla cpt\_ubicacion\_bodega:

Almacena las ubicaciones de las bodegas en el sistema.

Tabla cpt\_bodega\_variable\_lab:

Especifica las variables de laboratorio acorde a la bodega de almacenaje de producto en el sistema.

Tabla cpt\_bodega\_batch\_log:

Almacena los ingresos a las bodegas en el sistema.

Tabla cpt\_bodega\_articulo:

Especifica la bodega a la que pertenece un artículo en el sistema.

Tabla cpt\_empaque:

Especifica el empaque del batch y su bodega en el sistema.

Tabla cpt\_bodega\_usuario:

Designa el usuario o usuarios encargados de una bodega determinada en el sistema.

Tabla cpt\_producto:

Almacena los productos que procesa y distribuye TicoFrut.

Tabla cpt\_packing\_list:

Almacena las lista de empaque de producto que contiene los pedidos de los clientes.

## Conclusiones y comentarios

Para el desarrollo de las 2 primeras fases que comprenden el proyecto SGVI, se cumplieron los requerimientos de la primera fase, con algunas variantes en lo que respecta a funcionalidad y componentes utilizados para el proceso de algunos mantenimientos, específicamente en pantallas como mantenimiento de clientes y consignatarios y mantenimiento de medios de transporte de producto, en donde se solicitó, incluir detalles en cuanto a despliegue de consignatarios según clientes seleccionados y viceversa; y detalle de medios de transporte, según tipos de medios de transporte seleccionados y viceversa. También se solicitó la inclusión de una programación de entregas, esto para evitar incluir mucha información en la creación de una solicitud de despacho.

Se resolvió el problema para la asignación de varios productos a solicitudes de despacho y se implementó la asignación de una prioridad para cada orden de pedido, lo que permite ahora llevar un mejor orden de las solicitudes a procesar. Se implementó un calendario (tipo Gantt) en donde se puede observar el estado de las solicitudes de despacho y sus fechas de entrega, y el cual da acceso a la asignación de productos.

Para la segunda fase se concluyó el proceso de asignación de producto según especificaciones del cliente, a los Packing List, y el proceso de creación de órdenes de producción. Se implementó el proceso de cálculo de libras sólido para los tambores y bins de producto y se gestionó un módulo para la aprobación de muestras y producto con especificaciones claves por parte del departamento de Calidad. Se espera finalizar con lo respectivo al manejo de inventario y reportes sobre la existencia de producto en bodegas.

En resumen se entrega el sistema con el manejo de los mantenimientos requeridos para la elaboración de solicitudes de despacho y programaciones de entrega, la gestión de procesos en cuanto a creación de solicitudes de despacho, creación de órdenes de producción y asignación de producto a pedidos, y se trabaja en el proceso de movimientos en inventario, actualización del mismo y en la elaboración de los informes en cuanto a inventario estado del producto.

Cabe destacar que con la evolución de este sistema, se ha mostrado un interés por otros clientes en la implementación de un sistema similar para el manejo de sus procesos de ventas, producción y facturación como es el caso de la venta de Pellets (Residuo de la naranja) y el negocio de Fruta Fresca (Venta de naranjas por unidad), los cuales realizan un proceso similar, en el que varían únicamente los productos a vender, los clientes y algunos procesos en cuanto a inventario.

Por último, puedo decir que el trabajo desarrollado como práctica de especialidad ha sido una gran experiencia para mi persona, me ha permitido enfrentarme al proceso laboral, a conocer el manejo del negocio en esta compañía, ha conocer nuevas personas y trabajar en equipo. He adquirido conocimiento en nuevas tecnologías, en el trato hacia el cliente, y poner en práctica los conocimientos adquiridos en años atrás durante el estudio de la carrera.

## Glosario de términos

<b>Batch</b>	Cantidad de producto con su especificación
<b>Requisito</b>	Característica que se desea que posea un sistema o software
<b>Software</b>	Anglicismo relacionado al equipamiento lógico o soporte lógico de un computador digital
<b>Auditoria</b>	Controla la modificación y el acceso a los datos por parte de un usuario
<b>Especificación</b>	Conjunto de parámetros llamados variables de laboratorio que definen las características precisas de un producto
<b>Solicitud de Despacho</b>	Documento que especifica un pedido de productos por un cliente
<b>Orden de Producción</b>	Documento que se crea con el fin de realizar el procesamiento de nuevo producto para una solicitud dada.
<b>Packing List</b>	Documento comercial que tiene por objeto detallar el contenido de las mercancías que contiene cada bulto
<b>ERP</b>	Planificación de recursos empresariales
<b>SGVI</b>	Sistema de gestión de ventas e inventario
<b>CPT</b>	Sistema actual que se encarga del área de producción en su totalidad
<b>Tambor</b>	Tipo de empaque similar a un estañon que contiene una bolsa con el producto
<b>Consignatario</b>	Persona a quien va destinado un cargamento o mercancías
<b>Consecutivo</b>	Lo que sigue o sucede sin interrupción

<b>Libras Sólido</b>	Es una unidad en la que se conoce también el almacenaje del producto, esto es para ciertos reportes
<b>Muestra</b>	Pequeña cantidad de un producto que es obtenida para su análisis
<b>D-Limoneno</b>	Sustancia natural que se extrae del aceite de la cascara de los cítricos
<b>Termógrafo</b>	Instrumento de registro electrónico que monitorea y reporta los diversos cambios en las condiciones del medio ambiente en el tiempo
<b>Form</b>	Clase de plantilla de una pantalla.
<b>Dataset</b>	Conjunto de datos de una tabla en base de datos.
<b>Login</b>	Forma de acceder a una sesión en un sistema.
<b>Logout</b>	Forma de salir de una sesión en un sistema.
<b>Framework</b>	Estructura que implementa conjunto de sistemas o subsistemas.
<b>Grid</b>	Componente de base de datos para mostrar información de una o varias tablas.
<b>Delphi</b>	Versión moderna del lenguaje de programación Pascal.

## Bibliografía

[1] Rosales Rodríguez, Aryuris. Antecedentes de la Empresa Ticofrut S.A. (2006). San Carlos, Alajuela, Costa Rica.

[2] Juan Carlos Rojas Benavides, Eduardo Méndez Araya. Documento de Requisitos del Sistema. (2009). San Carlos, Alajuela, Costa Rica.

[3] Wilson Araya Rojas, Eduardo Méndez Araya. Documento de Alcances del Sistema (2009). San Carlos, Alajuela, Costa Rica.