

**SISTEMA INTEGRADO DE INFORMACIÓN PARA PRODUCCIÓN
EN INDUSOL**

**JHOANA ANDREA MARÍN VÉLEZ
JUAN CARLOS CALDERÓN GÓMEZ**

**UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2005**

**SISTEMA INTEGRADO DE INFORMACIÓN PARA PRODUCCIÓN
EN INDUSOL**

**JHOANA ANDREA MARÍN VÉLEZ
JUAN CARLOS CALDERÓN GÓMEZ**

**Trabajo de grado para optar al título de Ingeniero de Sistemas y
Telecomunicaciones**

**Presidente del Trabajo
Andrés Mauricio Martínez Hincapié
Ingeniero de Sistemas**

**UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2005**

A mis padres por su apoyo incondicional en la realización de un sueño familiar como lo es volverme un profesional de bien para la sociedad, a mi hermana y mi sobrino un motivo de lucha para la vida.

JUAN CARLOS CALDERÓN GÓMEZ

AGRADECIMIENTOS

La realización de este proyecto, fue posible al apoyo incondicional de los asesores, en especial a Andrés Mauricio Martínez quien dedico gran parte de su tiempo a que este desarrollo fuera posible.

A Hermán Velásquez, quien de una u otra manera participo con su conocimiento en el aporte de ideas valiosas para la culminación de este.

A Omar Antonio Vega, quien con sus valiosos consejos nos guió en la forma correcta de la elaboración de dicho proyecto.

CONTENIDO

RESUMEN	5
ABSTRACT	6
1. ÁREA PROBLEMÁTICA.....	7
2. OBJETIVOS	9
3. JUSTIFICACIÓN.....	10
4. MARCO TEÓRICO	11
4.1 TEORÍAS REFERENCIALES INGENIERILES.	17
4.1.1 Metodología orientada a objetos.....	17
4.1.2 Lenguaje unificado para la construcción de modelos u.m.l.....	42
4.1.3. Herramientas de desarrollo de visual basic.....	59
4.2 ANÁLISIS PREVIO A LOS SISTEMAS DE PRODUCCIÓN MRP I Y MRP II	61
4.2.1 Planificación.	61
4.2.2 Planificación empresarial y de operaciones.	61
4.2.3 ¿Qué es un sistema de producción?.....	62
4.2.4 Función de producción de una empresa.....	72
4.2.5 Elementos de la producción	72
4.2.6 Enfoque jerárquico para el proceso de planificación y control de producción.	75
4.2.7 Introducción a la planificación y control de inventario.....	76
4.3 SISTEMAS MRP I: EL MRP ORIGINARIO.....	80
4.3.1 Beneficios e implicaciones.....	81
4.3.2 Esquema básico del mrp originario	81
4.4. LOS SISTEMAS MRP II	82
4.4.1 El mrp de bucle cerrado.....	83
4.4.2 SALIDAS DEL SISTEMA MRPII	85
4.5 Elaboración del plan maestro de mantenimiento: Naturalmente, las modalidades y procedimientos a utilizar concretamente en la elaboración del plan maestro son función del tipo y problemática de la empresa y del sistema productivo considerado.....	87
4.6 ELABORACIÓN DE LA LISTA DE MATERIALES.....	89
4.7 FACTORES A CONSIDERAR EN LA APLICACIÓN DEL MRPII.....	89
4.8 PROBLEMÁTICA DE LA IMPLEMENTACIÓN DE LOS SISTEMAS MRPII.....	90
4.8.1 Problemas en torno al personal	90
4.8.1 Tipos de kanban y funcionamiento del sistema	93
4.8.2 Reglas	94
4.9 FILOSOFÍA JIT: ADECUACIÓN DEL SUBSISTEMA DE OPERACIONES Y OTROS ASPECTOS DE INTERÉS	94
4.10 MRP y JIT: ¿DIFERENTES O COMPATIBLES?.....	96

4.11 PLANIFICACION Y CONTROL A MUY CORTO PLAZO	96
4.12 FUNCIONES BÁSICAS	97
4.13 INTRODUCCIÓN A LA PROGRAMACIÓN DE OPERACIONES	99
CONCLUSIONES	101
RECOMENDACIONES.....	102
BIBLIOGRAFÍA.....	103
ANEXOS.....	104

RESUMEN

Este proyecto se llevo a cabo en INDUSOL, una empresa con sede en la ciudad de Pereira, se diseñó un software para planificación, programación y control de la producción en la empresa. Se utilizo metodología UML para el desarrollo del análisis y diseño adecuado. El proceso de diseño e implementación se llevó a cabo utilizando la herramienta de desarrollo Visual Basic bajo plataforma Windows.

La realización de este sistema integrado de información se debe a la necesidad de tener un módulo de producción que esté totalmente integrado en línea con la aplicación que actualmente se encuentra en funcionamiento, ya que se presentan problemas tales como: el control de inventario para materias primas y productos terminados (procesos manuales no confiables), el manejo independiente con el departamento de compras (generando procesos innecesarios), y otros diversos inconvenientes como lo son retardo en la planificación y programación de la producción, demoras en el material e inconsistencias en el tipo de material que debe entrar a cada parte del proceso; tiempos ociosos y de inconsistencias en la programación del personal; inconsistencias en los inventarios de materia prima, insumos, producto en proceso y producto terminado; no existen sistemas de control que permitan una adecuada toma de decisiones en el área de producción, entre otros problemas existentes.

Por otro lado, fue importante para la solución de estos problemas realizar un detallado análisis, diseño e implementación que pudiera brindar una rápida documentación a todos los requerimientos y así poder ofrecer a través de la ingeniería del software, un sistema eficaz y eficiente con el fin de proporcionar un sistema confiable y menos costoso.

ABSTRACT

This project I carry out in INDUSOL, a company located in Pereira's city, a software was designed for planning, programming and control of the production in the company. I use methodology UML for the development of the analysis and suitable design. The process of design and implementation was carried out using the tool of Visual development Basic under platform Windows,

The accomplishment of this integrated system of information owes to the need to have a module of production that is totally integrated to line with the application that nowadays one finds in functioning, since they present such problems as: the inventory control for raw materials and products finished (manual not reliable processes), the independent managing with the purchasing department (generating unnecessary processes), and other diverse disadvantages like it are a delay in the planning and programming of the production, delays in the material and inconsistencies in the type of material that must enter to every part of the process; Idle times and of inconsistencies in the programming of the personnel; faults in the inventories of raw material, inputs, product in process and finished product; there do not exist systems of control that allow a suitable capture of decisions in the area of production, between(among) other existing problems.

On the other hand, it was important for the solution of these problems to realize a detailed analysis, design and implementation that could offer a rapid documentation to all the requirements and this way to offer across the software engineering, an effective and efficient system in order to provide a reliable and less costly system.

1. ÁREA PROBLEMÁTICA

INDUSOL es una empresa con sede en la ciudad de Pereira, que tiene como objetivo principal la fabricación de productos referentes a la pintura (vinilos, lacas comerciales, lacas industriales, esmaltes, etc.) y materia prima para la fabricación de la misma (thiner, disolventes en general, xilol, etc.).

En la actualidad cuenta con un sistema de información para controlar los procesos en cuanto a información financiera, gestión comercial (puntos de venta), cuentas por cobrar, cuentas por pagar e inventarios. Dicho sistema de información está representado por un software que fue diseñado a la medida, aplicando los procesos que se llevan a cabo en cualquier empresa colombiana y apoyando tareas específicas que no maneja ningún software comercial, como por ejemplo: Control de estupefacientes en la venta de disolventes y también la orientación de INDUSOL a manejar los clientes de una forma muy convencional, es decir, la creación de grupos (clientes) para tener un mejor control a la hora de aplicar las leyes colombianas que limitan la venta de algunos productos que ofrece la empresa.

INDUSOL en su política de software a la medida, no cuenta con un módulo de producción que este totalmente integrado en línea con la aplicación que actualmente se encuentra en funcionamiento. Por esto, se genera una oportunidad para que un grupo de ingenieros profesionales en el área de los sistemas y las telecomunicaciones analicen, diseñen e implementen una solución, ya que se presentan problemas tales como: el control de inventario para materias primas y productos terminados (procesos manuales no confiables), el manejo independiente con el departamento de compras (generando procesos innecesarios), y otros diversos inconvenientes como lo son retardo en la planificación y programación de la producción, demoras en el material e inconsistencias en el tipo de material que debe entrar a cada parte del proceso; tiempos ociosos y de preparación de máquinas demasiado elevados, inconsistencias en la programación del personal; inconsistencias en los inventarios de materia prima, insumos, producto en proceso y producto terminado; no existen sistemas de control que permitan una adecuada toma de decisiones en el área de producción, entre otros problemas existentes.

Es importante mencionar, que la política de software a la medida de la empresa, es tradición desde sus inicios y considerada uno de los factores del éxito para mantenerse en ascenso constante en el mercado nacional, ya que es considerado

como factor decisivo en la toma de decisiones una información exacta, confiable, oportuna, veraz y muy orientada a las necesidades específicas de la gerencia.

2. OBJETIVOS

OBJETIVO GENERAL

Diseñar un software para planificación, programación y control de la producción en la empresa INDUSOL S.A., ubicada en la ciudad de Pereira.

OBJETIVOS ESPECÍFICOS

- Generar un módulo de planificación.
- Crear un módulo de capacidad.
- Implementar un módulo de programación.
- Generar un módulo de control.
- Definir e implementar un módulo que integre la información de producción con los inventarios y las órdenes de compra.

3. JUSTIFICACIÓN

Dada la situación que genera la política de la empresa, en cuanto a desarrollos a la medida, se entrega la responsabilidad a ingenieros de sistemas y telecomunicaciones, para que justifiquen el porque generar una solución específica para controlar los procesos en el área productiva de INDUSOL, ya que habiendo agotado todas las posibilidades para tratar de instalar un software especializado que se encuentre actualmente posicionado en el mercado, debido a que, como fue especificado en los antecedentes, ningún aplicativo es de fácil integración con los sistemas ya existentes. Cabe anotar, que esto es generado por los muchos de los procesos particulares de la empresa.

INUSOL es una empresa naciente en el mercado regional. Urge de un sistema de producción que se adecue a sus intereses colectivos en cuanto a ordenes productivos que los lineen hacia la calidad de sus productos. El sistema que se plantea pretende satisfacer con los propósitos organizacionales de la Empresa, todo con el fin de sincronizar procesos de control para Centros de Trabajos, Talleres, Herramientas, Operaciones y Materia prima.

Para la facultad de ingeniería de la Universidad de Manizales en su misión de extensión es de vital importancia imponerse en la solución de la problemática regional para las Empresas, ese es el caso de INDUSOL, caso que nos compete.

4. MARCO TEÓRICO

La empresa INDUSOL S.A. realizó un proceso de verificación con los software comerciales que existen en el mercado más cercano, se intentó la instalación y puesta en marcha de algunos de los software abajo mencionados pero sin ningún éxito, puesto que no pueden ser conectados con el sistema actual (inventario y compras en línea) y no manejan (en su mayoría) los controles creados por la misma empresa en su división de Producción.

Se realizó un proceso de investigación para detectar los antecedentes en cuanto a software existente y a continuación aparecen los más importantes, entre los que INDUSOL toma la determinación de tratar de implementar:

- SIIGO1 - Sistema Integrado de Información Gerencial Operativo, software genérico administrativo que permite llevar un registro detallado de las operaciones de la empresa y en general de todos los aspectos relacionados con la administración de negocios; su mercado objetivo son las medianas y pequeñas empresas de los sectores industrial, comercial y de servicios, que utilizan computadores bajo los sistemas operacionales DOS, WINDOWS, LINUX o REDES.

SIIGO se caracteriza principalmente por ser un sistema basado en documentos fuente (Facturas, Recibos de Caja, Cheques, etc.), es decir, que mediante la elaboración de éstos directamente en el computador se actualizan en línea y tiempo real todos los registros de la Empresa en lo referente a: Cartera, Cuentas por Pagar, Inventarios, Costos de Producción, Activos Fijos, Pedidos, Compras, Presupuesto, Contabilidad, Nómina y Ventas, por lo cual no se requiere de expertos para que se empiece a utilizar. SIIGO funciona alrededor del Sistema Contable, permitiendo gran versatilidad en el asentamiento de operaciones, al poderse registrar de dos formas: La primera, mediante la elaboración directa de los documentos en el computador, en cuyo caso el sistema automáticamente realiza la contabilización (Comprobante) y el registro de las cifras en todas las demás áreas (transparente para el usuario); y la segunda, mediante la incorporación directa de Comprobantes de Contabilidad, garantizando por cualquiera de los medios, que todos los registros sean afectados

¹ SIIGO. Qué es SIIGO. [en línea]. Bogotá. Fecha de actualización: 06-02-03. SIIGO / Productos / Qué es SIIGO. Disponible en:
<http://www.siigo.com.co/docs/BPWebsite.asp?TContent=BDDocument.asp%3FAction=View%26NoHeader=1%26NoSubject=1%26ID={B12CE7A5-F9DD-4781-9FC3-FFD06C7F378E}>

simultáneamente; provee así un solo medio de alimentación de información (Comprobantes), lo que asegura Integridad y Consistencia total entre sus diferentes aplicaciones.

- SEVEN-ERP². Gestión Manufactura. SEVEN - ERP Gestión de Manufactura, permite la integración de todos los procesos de producción con la gestión financiera-administrativa y la gestión comercial de su compañía. Los procesos de Manufactura pueden ser documentados mediante el módulo Work Flow. De igual manera soporta la generación de Indicadores de gestión y la aplicación de mejores prácticas en los procesos de manufactura. SEVEN - ERP Gestión de Manufactura, permite la integración de todos los procesos de producción con la gestión financiera-administrativa y la gestión comercial de su compañía. Los procesos de Manufactura pueden ser documentados mediante el módulo Work Flow. De igual manera soporta la generación de Indicadores de Gestión y la aplicación de mejores prácticas en los procesos de manufactura.
- OFIMATICA³. Sistema de Manufactura: Costos, Producción y Control de Piso. AREA DE COSTOS: Formulación: Material Directo, Mano de Obra Directa, Mano de Obra Indirecta, Costos Indirectos de Fabricación. • Cálculo de los Costos Estándar y Reales de la Producción. • Comparativo Costos Estándar vs. Reales. • Valorización Producto en Proceso, Control de Lotes (Órdenes de Producción). ÁREA DE PRODUCCIÓN: Formulación (Ficha Técnica). • Planeación de la Producción. • MPS: Plan Maestro de Manufacturas. • MRP: Planeación Requerimiento Materiales. • CRP: Planeación Requerimiento Capacidad. • Requisiciones Automáticas a Compras. • Entrega de Materiales y Producto Terminado. • Actualización Automática de Inventarios. • Múltiples Simulaciones / Escenarios, Trazabilidad Normas ISO. ÁREA CONTROL DE PISO: Control de la Producción en Planta. • Ordenes de Producción para cada Centro de Trabajo. • Seguimiento Productos por Rutas. Seguimiento a Maquiladores. Generación Automática de Despachos de Producto Terminado a Bodega. • Valorización Productos en Proceso a Real. • Control de Maquinaria y Herramientas. • Control Estándar de Calidad.

² SEVEN-ERP. Gestión Manufactura. [en línea]. Bogotá. Fecha de actualización: 06-04-02 [en línea]. Bogotá. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=511&sid=104>

³ OFIMATICA. [en línea]. Bogotá. Fecha de actualización: 06-04-02 Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=505&sid=104>

- JobBOSS: Formamos parte de su Negocio⁴. Desarrollador: Exact Software. "JobBOSS" - Sistema Líder en administración de producción "MADE TO ORDER". Más de 3.000 compañías en el mundo usan "JobBOSS" para vigilar los costos de producción y ser más lucrativos. Diseñado para cubrir requerimientos de productos bajo diseño - cliente en la cantidad que este solicite, con diseños de ingeniería adecuados a la necesidad de cada tipo de cliente. JobBoss entrega la flexibilidad que el usuario necesita para manejar el mundo real - donde los clientes exigen menor tiempo de proceso desde el comienzo del trabajo hasta que se obtiene el producto optimizando, generando resultados, y obteniendo un mejor precio. Permite ajustarse a los cambios y aumentar el control de todo negocio para conocer el nivel de rentabilidad de cada orden de trabajo. MODULOS: Ventas, Cotizaciones, Producción, Inventarios, Compras, Subcontrataciones, Despachos, Captura de información por código de barras, Capacidad Finita, Programación de producción, compras y materiales.
- ADVANTAGE⁵. Manufactura - Textiles - Confecciones - Calzado - Alimentos y Bebidas. Desarrollador: APPLIED SYSTEMS LTDA. Programación de actividades (Horarios y Calendarios) - Definición de características y configuración para los diferentes tipos de productos, clientes y proveedores -Diseño de reportes -menús personalizados por usuario. • PRODUCCION: Definición de productos en diferentes etapas con los insumos requeridos en cada una de ellas (materias primas, horas hombre, horas equipo, servicios) - Elaboración de ordenes de producción - presupuesto e Insumos requeridos - Programación y Control en la ejecución - Cantidad y costo de los insumos necesarios a través del tiempo (con proyección de costos). • INVENTARIOS: Entradas - Salidas- Transferencias - Manejo de varios almacenes y bodegas - Conteo físico - Composición de productos - Kardex por producto y por almacén - Stocks mínimos y máximos. • COMPRAS Y GASTOS: Pedidos a proveedores - Entregas - Facturas Cuentas por pagar - Comprobantes de egreso - Anticipos - Devoluciones - Elaboración cheques - Movimiento proveedores - Cargue automático a inventarios y a Contabilidad. • FACTURACION Y CARTERA: Pedidos de clientes - Cotizaciones - Control de entregas y facturación - Descargue automático del inventario - Cuentas por cobrar - Recibos de caja - Cartera por edades - Histórico de ventas - Productos por cliente - Utilidades por producto. • CONTABILIDAD: PUC - Comprobantes - Pagos a terceros - IVAS y retenciones - Libro diario -

⁴ JobBOSS: Formamos parte de su Negocio. [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=498&sid=104>

⁵ ADVANTAGE. [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=492&sid=104>

Balances - Conciliación bancaria - Certificados de retención - Centros de costo - Presupuesto - Libros oficiales - Informes DIAN. • NOMINA: Liquidaciones - Apropriaciones - Aportes - Préstamos.

- MACOLA⁶. Manufactura. Desarrollador: Exact Software. Entrega un paquete completo de aplicaciones diseñadas para producción por inventario, producción por orden, ensamblaje por orden y una mezcla de ambientes de producción. Busca optimizar el servicio al cliente, reducir inventarios y tiempos de producción, mejorando la calidad del producto. Incluye inventarios, listas de materiales, rutas de producción, planeación maestra (MS), requerimientos de material (MRP), plan capacidad producción (CRP), programación y control de producción (SFC), desempeño mano de obra, contabilidad costos manufactura, programación procesos finitos, interfaz relojes electrónicos y códigos de barras. El paquete provee planeación de ciclos cerrados y sistemas de ejecución para empresas de manufactura con procesos de cambio repentino.
- Microsoft Business Solution AXAPTA⁷. Sistema Administrativo Integrado e-business y ERP en uno. Desarrollador: MICROSOFT. Business Solution AXAPTA es la mejor Solución Empresarial Integrada, diseñada para cubrir las necesidades de su empresa, totalmente escalable en su versión Enterprise. Completamente funcional a través de Internet, compatible con los productos Microsoft; incluye herramientas OLAP. MÓDULOS: SERIE FINANCIERA: Contabilidad General, Presupuesto Financiero, Cuentas por Cobrar y Pagar, Gestión Bancaria, Consolidación, Flujos de Caja, Apropriaciones. * SERIE COMERCIAL: Gestión de Compras, Gestión de Ventas, Inventarios. • SERIE LOGÍSTICA: Acuerdos Comerciales, Gestión de Almacenes, Warehouse Management, Listas de Materiales, Presupuestos (stock, compra y venta). • SERIE DE PRODUCCIÓN: MRP I, II, III, Planificación Maestra, Control de Piso, Hojas de operación, poderoso Constructor Inteligente de Productos. • SERIE DE PROYECTOS: Gestión de proyectos. * INTELIGENCIA DE NEGOCIOS: Balanced Scorecard I, II y III, Admón. Procesos de Negocios I, II, III, Planificación de Estrategias, Cuestionarios Web. Adicionalmente Microsoft Business Solution AXAPTA incluye CRM (Gestión de Relaciones con Clientes), HRM (Administración de Recursos Humanos), E-Business (Comercio Electrónico), Tecnología Web y Wap, Multiempresa, Multiidioma, Multipaís y Multimoneda. Nuestra metodología Diamond propiedad de Columbus, garantiza el éxito de la

⁶ MACOLA. [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=499&sid=104>

⁷ Microsoft Business Solution AXAPTA. [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=503&sid=104>

implementación en muy corto tiempo. Además se entregan las fuentes del aplicativo.

- SISTEMA CM-UNO⁸. Sistema Comercial Integrado – MRP. Desarrollador: SISTEMAS DE INFORMACION EMPRESARIAL S.A. Nuestra mejor carta de presentación es él haber trabajado exitosamente por varios años bajo condiciones críticas de complejidad de procesos productivos, altos volúmenes de información y gran número de usuarios concurrentes, en reconocidas empresas de diferentes sectores económicos. Maneja plenamente la temática del MRP para orientar a la empresa en que producir / compras, cuando y en que cantidades. Para las empresas con mayor complejidad en sus procesos se tiene él modulo de CRP que ayuda a planear la utilización y capacidad de planta. Para el control de la gestión diaria el sistema ofrece dos módulos transaccionales: Control por Ordenes de Producción y Control de Producción por Procesos. Al ser esta solución totalmente integrada al BackOffice del sistema UNO - Contabilidad CG-UNO, inventarios y compras del sistema comercial CM-UNO, Sistema de nómina(NM-UNO) y sistema de activos fijos(AF-UNO)- se logra un solo entorno que potencializa radicalmente los procesos de control y toma de decisiones en todos los niveles administrativos de la organización.
- System21⁹. Sistema Integrado de Negocios ERP. Desarrollador: Geac Enterprise Solutions. System21 es una solución de clase mundial para empresas productoras y de distribución, permite un control muy efectivo en las áreas de manufactura, costos, distribución, logística, servicio al cliente y contable - administrativa. Permite automatizar los procesos de producción en ambientes continuos o discretos, soporta modelos de producción mixtos por procesos y discretos, combina las operaciones con los insumos y salidas en cada etapa del proceso lo que le permite un control muy preciso de cada operación en la planta. Además de la solución de carácter general cuenta con funcionalidad especializada para los sectores de: alimentos y bebidas, textil-confección, calzado, distribución mayorista y automotriz. Tiene presencia en el mercado Colombiano hace más de 10 años contando en la actualidad con una importante base de clientes. * Ver soluciones integradas a System21: Para Inteligencia de negocios: CORVU - Para monitoreo electrónico de procesos industriales: MOPAC)

⁸ SISTEMA CM-UNO. [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=514&sid=104>

⁹ System21 [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=515&sid=104>

- VISUAL FACTORY.ERP¹⁰ - Sistemas Modular Integrado. Manufactura / Control de Producción / MRPII. VISUAL FACTORY.ERP fue creado con el fin de suministrar una herramienta completa, flexible y de fácil manejo; con el propósito de controlar los inventarios, incrementar la eficiencia de la planta; reduciendo costos y apoyando la gestión administrativa y financiera en la toma de decisiones y la automatización de procesos operativos y de análisis de las diferentes áreas. VISUAL FACTORY.ERP es el Software Colombiano de producción más vendido e instalado en todo tipo de industrias y se encuentra integrado con las demás aplicaciones a nivel financiero, comercial y administrativo. Permite generar resultados altamente confiables traducidos en beneficios para las empresas. VISUAL FACTORY.ERP por ser desarrollado 100% en Colombia, garantiza el cumplimiento de todas las disposiciones legales vigentes y además el usuario cuenta con un servicio de soporte y capacidad de respuesta altamente especializados. Por su modularidad y alto grado de parametrización logra pleno ajuste a las necesidades de cualquier tipo de empresa. VISUAL FACTORY.ERP trabaja con criterio Cliente-Servidor, bajo SQL SERVER u ORACLE para las grandes empresas y bajo Visual FoxPro para la pequeña y mediana empresa. PROGRAMAS CERTIFICADOS PARA MANEJO DE BUENAS PRACTICAS DE MANUFACTURA (BPM)
- VisualProducción¹¹. Control de Producción Empresas Manufactureras. VisualProducción es la solución completa para el total control productivo en las empresas manufactureras. Dentro de sus funciones están: Control de Inventario materia prima, Producto en proceso, Subproductos, Producto terminado, Control de Mano de Obra y Servicios a Terceros, Formulación de Productos (B.O.M), Programación de Producción, Requerimientos de Materiales (MRP), Requerimiento de Recursos (MRPII), Asignación de CIF por actividades o procesos, Costos por actividades (ABC Costing), Inventarios, Automatización. Informes: Costos Reales vs. Estándar, Costos por lote y/o Producto, Capacidad de Planta, Margen de Utilidad, y hasta mas de 1000 informes. El sistema está desarrollado para Intranet y/o Extranet (Tecnología Web). Integración total con los programas de Office 2000/XP y MS Project 2000/2002. Base de datos: SQL 7.0 - 2000. Modulo de Automatización de Planta con Código de Barras para la captura de data en los procesos productivos y sus consumos de Materia Prima.

¹⁰ VISUAL FACTORY.ERP [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=497&sid=104>

¹¹ VisualProducción. [en línea]. Bogotá. Fecha de actualización: 06-04-02. Disponible en: <http://catalogodesoftware.com/ItemsE.asp?iid=517&sid=104>

4.1 TEORÍAS REFERENCIALES INGENIERILES.

Para comprender el proyecto SISTEMA INTEGRADO DE INFORMACIÓN PARA PRODUCCIÓN EN INDUSOL es necesario como mínimo conocer los conceptos teóricos básicos de los siguientes temas:

- Metodología Orientada a Objetos.
- Lenguaje Unificado para la Construcción de Modelos U.M.L.
- Herramienta de Desarrollo Visual Basic.

4.1.1 Metodología orientada a objetos.

La tecnología de Objetos data de los años 60, cuando surge la necesidad de describir y simular fenómenos como sistemas de comunicación, redes neuronales, sistemas administrativos, etc.

En 1961 Krystin Nygaard con la idea de desarrollar un lenguaje de doble propósito (descripción de sistema y simulación programable), crea SIMULA I. Los usuarios descubrieron que también proveía de nuevas y poderosas facilidades cuando era usado para otros propósitos, aparte de la simulación, tales como el prototipo y aplicaciones.

En 1967 se creó SIMULA 67, y en él se implementaron por primera vez los conceptos de clase, objeto y herencia, que en adelante serían elementos centrales en los Lenguajes Orientados A Objetos. SIMULA 67 es una extensión del lenguaje ALGOL 60 y diseñado en 1967 por Ole - Johan Dahl y Krystin Nygaard, de la Universidad de Oslo y del Centro de Computación Noruego (Norsk Regasentral). Sin embargo Simula, así como se le conoce actualmente, es un lenguaje de programación de propósito general, que no es ampliamente utilizado.

En 1970 se crea el SMALLTALK, éste fue el mayor desarrollo de los lenguajes orientado a objetos. El proyecto de investigación se realizó en la Corporación Xerox, Centro de Investigación de Palo Alto (PARC - Palo Alto Research Center) y fue dirigido por Allan Kay. Empezó en los 70's y tuvo como meta, más que el lenguaje de programación, una completa interfacez gráfica y herramienta de desarrollo integradas.

Xerox PARC fue la pionera en el desarrollo y utilización de los componentes estándar de las modernas interfaces gráficas, como ventanas, iconos, mouse, etc.

Smalltalk fue el primer lenguaje orientado a objetos, ya que trató exclusivamente con ellos; los subsecuentes, se ha basado en los conceptos utilizados por él. Smalltalk fue importante no sólo por su lenguaje, sino por las herramientas de

desarrollo disponibles en su ambiente, éstas incluyen visualizadores de clases (class browsers) e inspeccionadores de objetos (object inspectors).

Un visualizador de clases es una poderosa herramienta, que permite editar el código del programa, en una manera mucho más conveniente y estructurada que utilizando editores convencionales. Por la estructura, inherente, bien definida por los programas orientados a objetos, el visualizador de clases es capaz de mostrar, en forma gráfica, el árbol de una clase dada, permitiendo al usuario "apuntar y disparar" método en particular (proceso) a ser editado. Otras ventajas del visualizador es que muchas tareas de programación se realizan sólo por menús, por ejemplo la creación de una nueva clase modificación de la estructura del árbol de herencia, modificación de una estructura de una clase etc. Estas operaciones son mucho más complejas cuando se realizan en un ambiente de edición tradicional.

Herramientas como éstas, son una parte integral de la "promesa" de la Tecnología Orientada a Objetos, ya que puede simplificar la vida del programador, reduciendo el costo y el tiempo de desarrollo.

En los años 80 evoluciona el SMALLTALK y se crea ADA, lo que hizo crecer el interés en el Diseño Orientado a Objetos.

En estos lenguajes la abstracción de los datos tienen gran importancia. Los problemas del mundo real se representan mediante objetos a los cuales se le agrega operaciones cuando es necesario

La TOO se fundamenta en el proceso de construcción y utilización de conocimientos, por lo tanto, objetos y clases son los pasos más importantes en la búsqueda de una nueva revolución que reemplace, esta vez, parte del esfuerzo que implica la organización y utilización del conocimiento, del mismo modo que en la primera, las máquinas reemplazaron el esfuerzo físico del hombre y de los animales, permitiendo el vertiginoso avance del mundo.

Los "Chips de Software" (Objetos y clases altamente reutilizables) serán el motor de la revolución que ya se ha iniciado. Participar en ella garantizará nuestra competitividad en el futuro y solo nos exige todo el esfuerzo de nuestra capacidad creativa, y de la perseverancia de construir productos de alta calidad. La materia más importante del software es la inteligencia.

El Desarrollo de la estructura del pensamiento y la construcción del conocimiento. Desde muy temprana edad, aproximadamente hasta los siete años, en los denominados períodos Senso-motriz y Preparatorio de la inteligencia, en los que se

afirma el esquema del objeto permanente, los seres humanos, a partir de la experiencia de la observación y captación de cosas y hechos del mundo que nos rodea, se construyen conceptos "concretos". Estos nos permiten ser consistentes y razonar. Por ejemplo: Fido es un elemento del mundo real que es conceptualizado como Fido. En él se almacenan sus características (DATOS), por ejemplo: tiene pelos, cuatro patas, dos orejas, dos años de edad, es de color blanco, otros. Junto con los datos también se guardan sus acciones (COMPORTAMIENTO), es decir, lo que puede o no hacer (ladrar, correr, saltar, comer, etc.). Como consecuencia de este hecho (almacenamiento de los datos y comportamiento en una sola unidad conceptual), entre las muchas operaciones de nuestra mente, se puede razonar que Fido no tiene alas, ni plumas, ni puede volar.

A las acciones que, cualquier elemento del mundo real, realiza como reacción frente a un estímulo se llamará Operaciones. Algunos comportamientos complejos son el resultado de la interacción o sucesión de varias operaciones, a éstos se les llamará Procesos. Sin embargo en nuestro enfoque, en la mayoría de los casos y de manera genérica se utilizará el término OPERACIÓN.

Los conceptos se construyen modelando los aspectos y características de cualquier objeto real, extrayendo los detalles (datos y comportamiento) necesarios y descartando lo menos útil. Este proceso, natural en los seres humanos, compuesto por diversas operaciones tales como observar, captar, fijar, comparar, conceptualizar y otros es denominado abstracción.

Cuando se piensa en un auto, no se visualiza cada mínimo detalle que lo describa, lo más probable es que se tenga en mente sólo las principales características físicas y, dependiendo de nuestra experiencia, algunos de los subsistemas importantes como la caja de cambios, o los sistemas de dirección y frenos.

En Tecnología Orientado a Objetos (TO) a los conceptos se les llama objetos, en consecuencia, los OBJETOS son MODELOS de entes del mundo real. Los que se percibe es porque tienen límites que los esperan de su medio ambiente y de los demás (IDENTIDAD). Se encuentran en un ESTADO específico en el mundo.

Un objeto puede ser concreto (material) o abstracto (inmaterial), simple o complejo; pero siempre estará compuesto de DATOS y OPERACIONES.

En los siguientes ejemplos, se trata en cada caso, de reconocer sus componentes y entender porqué son objetos:

Un libro....una factura....una organización.
Una figura en un programa para dibujar.

Una pantalla con la cual un usuario interactúa.
Un campo o nodo en la pantalla de una herramienta CASE.
Un mecanismo o un dispositivo robótico.
Un dibujo de ingeniería completo, un componente de ese dibujo.
Los textos y figuras usadas en el diseño de un periódico.
Un avión...el vuelo de un avión...una reserva de vuelo.
Un icono en pantalla, al que se puede señalar y abrir.
El proceso de llenar un formulario.
El proceso de escribir esta línea etc.

Continuando con el desarrollo de la estructura del pensamiento, se puede notar que aproximadamente a partir de los siete años, en la denominada etapa de las operaciones concretas, el ser humano aprende a CLASIFICAR (separar y agrupar objetos de acuerdo a características y comportamientos comunes) y ORDENAR (establecer criterios de jerarquía que permitan la organización).

Por ejemplo: se conocen 2 animales y se observa que tienen cuatro patas, ladran, muerden, etc. Y a partir de estos conceptos "concretos" se puede formar un nuevo concepto "abstracto" al que se llamara PERRO. Este contiene las características y el comportamiento de todos los semejantes a Fido. En otras palabras, se construyen conceptos tipos o Modelos.

Así se puede construir los modelos GATO, CONEJO, y aún sin la experiencia directa de la observación, TIGRE, ORNITORRINCO, etc. Este proceso es una consecuencia de la función simbólica (imitación diferida, juegos simbólicos, etc.).

De estos modelos se construirán otros, cada vez más abstractos, que contendrán a los anteriores, por ejemplo: CARNÍVOROS, MAMÍFEROS, VERTEBRADOS, ANIMALES, SERES VIVOS, etc. generándose una estructura muy organizada de CLASIFICACIÓN Y ORDEN.

La clasificación es un medio por el cual se organiza el conocimiento. Clasificar y ordenar es una consecuencia de la capacidad de interiorizar esquemas representativos (MODELOS), los cuales permiten: por un lado, reconocer si un concepto (OBJETO) tiene o no las características de dicho esquema; y por otro, incorporarlo o no en el conjunto de los que comparten ese mismo esquema (CLASE). Posteriormente se organiza jerárquicamente en clases y superclases, construyendo en forma progresiva las grandes CLASIFICACIONES y las OPERACIONES DE ORDEN.

Después de los doce años, en la etapa de las Operaciones combinatorias de la inteligencia (Pensamiento formal), el hombre construye SISTEMAS y aún TEORÍAS

como producto del raciocinio de Hipótesis (propuestas de solución a interrogantes o problemas), y de la representación en dimensiones simultáneas.

Un Sistema es un conjunto de componentes (conceptos y modelos, y aún de otros sistemas) que interactúan, en distintos planos, entre sí para alcanzar objetivos específicos. Cada Sistema construido es una manera de solucionar los diferentes problemas que enfrenta el hombre a lo largo de su existencia, por ejemplo el Lenguaje, uno de los sistemas más elaborados, permite satisfacer la necesidad vital de la comunicación.

Cuando un sistema tiene por objetivo explicar un hecho o fenómeno, y es aceptado, mientras no se demuestre lo contrario, u otro sistema lo explique mejor, se le denomina TEORÍA. Por ejemplo, la teoría de la relatividad, la teoría del conocimiento, etc.

En síntesis, se construyen CONCEPTOS, MODELOS y SISTEMAS. Esta es la forma en que los seres humanos captan el mundo real y operan en él. La tecnología de Objetos intenta simular este hecho y sus fundamentos se sostienen en él.

A continuación se explicaran los tres más importantes.

Los fundamentos de la tecnología de objetos (TO).

Simulación. Representación directa de elementos que "maneja" el usuario en el mundo real.

Consiste en recrear el mundo real. No se trata de construir "modelos ideales", sino de representarlo directamente. Por ello, en este enfoque, primero se identifican las características de los elementos del mundo real, los que se organizan en las denominadas ESTRUCTURAS DE DATOS. Luego se captan los comportamientos u operaciones, los cuales se imitan o simulan mediante pequeños programas (procedimientos) a los que en adelante se denominara MÉTODOS. Y así como en la vida real se conceptualiza en una sola unidad (datos y comportamiento), en TO se simula este hecho colocando en una especie de "cápsula" las estructuras de datos y los métodos, a esta unidad se le llamará OBJETO.

También se simula la manera en que los entes del mundo real se comunican entre sí, enviándose señales o "mensajes" a los que responden con una conducta o comportamiento específico, de la misma manera se simulan las clasificaciones, ordenamiento, organización, e incluso la herencia de los seres vivos.

En síntesis, se trata de actuar de la misma manera en que los humanos, a los que en informática, en general, se nos denomina usuarios, "manejamos" y operamos el mundo real.

Encapsulamiento. Utilización del concepto de "Caja Negra" a una potencia mucho mayor. Empaquetar datos y operaciones en forma conjunta se llama ENCAPSULACIÓN. La encapsulación es el resultado (o acto) de ocultar, al usuario, los detalles de la implementación de un objeto. El objeto oculta sus datos a otros objetos y sólo permite acceder a sus datos vía sus propios métodos mediante mensajes específicos, es decir, se crea una "Caja Negra" que solo el constructor del objeto conoce. A esto se llama ocultamiento de información. La encapsulación protege los datos de un objeto de la corrupción. Si todos los programas pudieran acceder a los datos (como actualmente sucede con la tecnología estructurada), fácilmente puede corromperse o perderse. La encapsulación protege los datos del objeto del uso arbitrario y no intencionado. Así la creación está protegida y la competencia garantizada.

Modularidad. El código de un objeto puede ser escrito y se puede mantener independiente del código de otros objetos. Un objeto se puede mover de sistema en sistema, se puede quitar, modificar y volver a colocar sin alterar el sistema general.

Esconder. la información (Information Hiding). Un objeto tiene una interfaz con la que otros objetos se pueden comunicar, pero puede mantener información privada para sí misma que puede cambiar en cualquier momento, sin afectar a los objetos que dependen de ésta.

Reutilización. Capacidad de reutilizar código sin alterarlo, programando solo lo adicional o diferente. Base de la Herencia.

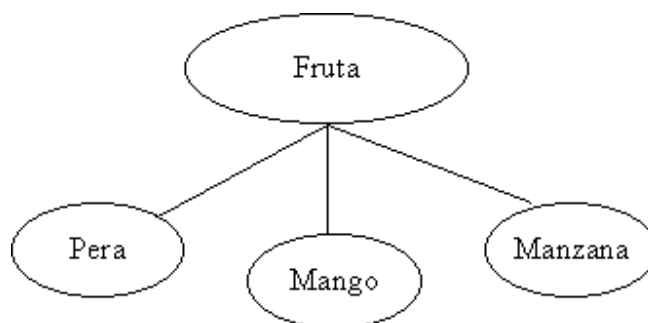
Durante la vida de un sistema, las estructuras de datos se mantienen relativamente estables, mientras que las operaciones sobre dichas estructuras cambian, dependiendo de situaciones. Por ello en el Análisis y Diseño Orientado a Objetos, se da énfasis primordial a los DATOS y al COMPORTAMIENTO de los objetos y tipos de objetos (modelos). Los datos, como son relativamente estables, pueden ser utilizados muchas veces, modificando únicamente aquello que sea necesario para tal o cual situación, en la misma forma se procede con los comportamientos. En cambio, en el Análisis y Diseño Estructurado, se da énfasis a la descomposición funcional orientada al proceso, en consecuencia, cualquier modificación requiere de una reconstrucción, normalmente, a partir de cero.

Programación orientada a objetos. En el mundo de la computación existen varios paradigmas de programación. La mayoría de las personas programan de un modo TOP-Down, en el que la programación va de arriba hacia abajo. El programa tiene una secuencia ordenada que debe seguir, y como quien dice, un principio y un fin. Entonces, ¿Cuál es la filosofía de la programación orientada a objetos?. La respuesta es que en esta filosofía los programas son vistos como objetos.

Primero se va a definir algunos términos desde el punto de vista de programación orientada a objetos:

Objeto. Es una persona, lugar o cosa. En la Figura 1, se aprecia el siguiente ejemplo: Una manzana. Todos los objetos tienen un estado y un comportamiento. Esto es muy importante. Para el ejemplo de la manzana, la manzana (hablando de una manzana normal de color rojo) recién bajada de un árbol, está completa, es dura y de color rojo. Este es su estado. Uno sabe que si se muerde la manzana va a sonar un "crunch", y esto se puede interpretar como un comportamiento. Después de haberla mordido la manzana cambió de estado. Los objetos pueden tener más de un comportamiento y más de un estado.

Figura 1. Representación de un objeto



Clases. Un grupo de objetos con las mismas características. Una clase contiene:

- Una descripción de las variables internas de los objetos de la clase.
- Las operaciones que se pueden aplicar al objeto de la clase.

Ejemplo. Análogo

En este ejemplo se va a analizar la clase Música de Colombia es la clase padre de las clases hija Vallenato y Salsa. En otras palabras las clases Vallenato y Salsa son derivadas de la clase Música de Colombia. Ellas heredan todas las propiedades de su padre. Las clases derivadas pueden adicionar, cambiar u ocultar los miembros

públicos de la clase padre usando las reglas de la herencia. Cuidado... los miembros privados no pueden ser accedidos por otras clases.

Clase padre Música de Colombia

Atributos: Tipo de ritmo, tipo de instrumentos, músicos.

Métodos: Bailada, cantada, tomada.

Clase hija Vallenato (hereda las propiedades de la clase padre)

Atributos: Tipo de ritmo, tipo de instrumentos, músicos.

Métodos: Bailada, cantada. El nuevo método incluye el tocar un instrumento, Tomada.

Clase hija Salsa (hereda las propiedades de las clase padre)

Atributos: Tipo de ritmo, tipo de instrumentos, músicos.

Métodos: Bailada, cantada, tomada.

Herencia. Principio por el cual una clase se puede derivar de otra clase ya existente, heredando las características del padre.

Un ejemplo:

Tengo el objeto fruta, con algunas propiedades inherentes a la fruta. Luego tengo el objeto manzana, que es hijo de fruta y por lo tanto hereda las propiedades de la fruta, pero puede tener otras propiedades propias de la manzana. Este es un ejemplo sencillo pero más o menos ilustra lo que se trata de explicar. El hijo hereda las características del padre, pero tiene otras propias y mejora o empeora, algunas características del papá.

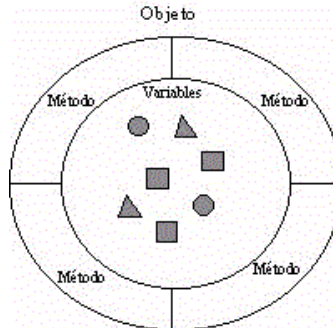
Mensaje. Los objetos se comunican unos con otros mediante mensajes.

Polimorfismo. Principio en el cual objetos de diferentes clases pueden entender el mismo mensaje pero responder de manera distinta.

Una vez que se tienen bien claros estos conceptos se puede entender mejor la filosofía de objetos. Ahora tratamos de encapsular los estados de los programas por medio de objetos, que pueden ser accedidos solamente por medio de operaciones definidas dentro de ellos mismos.

Los programas están compuestos por varios objetos. Cada objeto tiene sus variables internas así como sus métodos bien definidos. Un método es el equivalente a un procedimiento.

Figura 2. Representación de una clase



Modelo orientado a objetos. El Modelo Orientado a Objetos se basa en el paradigma de programación orientado a objetos. Este paradigma ha tenido gran aceptación debido a que es de gran naturalidad buscar objetos en la realidad a modelar.

Estructura de Objetos. El modelo orientado a objetos se basa en encapsular código y datos en una única unidad, llamada objeto. La interfaz entre un objeto y el resto del sistema se define mediante un conjunto de mensajes.

El motivo de este enfoque puede ilustrarse considerando una base de datos de documentos en la que los documentos se preparan usando uno entre varios paquetes software con formateador de texto. Para imprimir un documento debe ejecutarse el formateador correcto en el documento. Bajo un enfoque orientado a objetos, cada documento es un objeto que contiene el texto de un documento y el código que opera sobre el objeto. Todos los objetos del tipo documento responden al mensaje Imprimir, pero lo hacen de forma diferente. Cada documento responde ejecutando el código formateador adecuado. Encapsulando dentro del objeto documento la información acerca de cómo imprimirlo, se puede tener todos los documentos con la misma interfaz externa al usuario (aplicación).

En general, un objeto tiene asociado:

Un conjunto de atributos que contienen datos acerca del objeto. A su vez, cada valor de un atributo es un objeto.

Un conjunto de mensajes a los que el objeto responde.

Un conjunto (puede ser unitario) de métodos, que es un procedimiento o trozo de código para implementar la respuesta a cada mensaje. Un método devuelve un valor (otro objeto) como respuesta al mensaje.

Puesto que la única interfaz externa de un objeto es el conjunto de mensajes al que responde, es posible modificar la definición de métodos y atributos sin afectar a otros objetos.

También es posible sustituir un atributo por un método que calcule un valor.

Ejemplo: Un objeto documento puede contener un atributo de tamaño que contenga el número de bytes de texto en el documento, o bien un método de tamaño que calcule el tamaño del documento leyéndolo y contando el número de bytes.

"La capacidad de modificar la definición de un objeto, sin afectar al resto del sistema, está considerada como una de las mayores ventajas del Modelo de Programación Orientada a Objetos".

Jerarquía de clases. Normalmente en la realidad a modelar existen muchos objetos similares. Por similar se quiere decir que tienen una naturaleza parecida, por lo que son candidatos a poseer atributos, métodos y responder a mensajes comunes en un contexto orientado a objetos (VIC).

Para el caso de los objetos similares, sería un trabajo inútil definir cada uno de ellos por separado. Por tanto, se agrupan para que conformen una clase. A cada uno de estos objetos se le llama instancia de clase. Todos los objetos de una clase comparten una definición común, aunque difieran en los valores asignados a los atributos.

Ejemplo: En un banco, son objetos los clientes, las cuentas y los préstamos.

Una clase incluye:

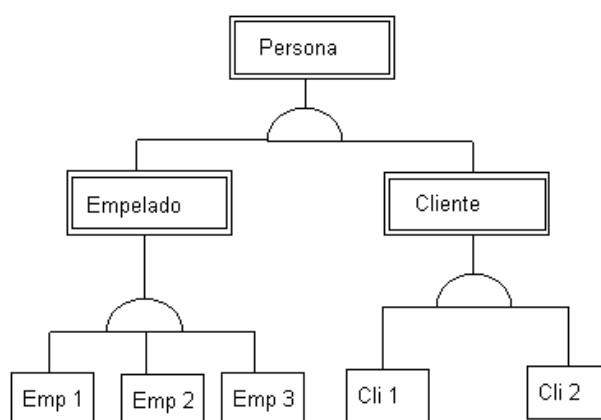
Un atributo con valores en un conjunto cuyo valor es el conjunto de todos los objetos que son instancias de la clase.

Implementación de un método para el mensaje nuevo, el cual crea una nueva instancia de la clase.

Un esquema orientado a los objetos, normalmente requiere un gran número de clases. Sin embargo, a menudo se da el caso que varias clases son similares. Para permitir la representación directa de similitudes entre clases, se utiliza una jerarquía de especialización, como aquella utilizada por el modelo MER extendido.

Ejemplo: En un banco, es de esperarse que los Empleados y los Clientes tengan ciertos atributos comunes como RUC, nombre, dirección y teléfono, sin embargo los Empleados podrían tener un atributo exclusivo, como salario, mientras que los Clientes tendrán atributos como tasa de crédito. Aquí los empleados y clientes pueden ser representados por especializaciones de una clase Persona. Los atributos y los métodos específicos para Empleado se asocian a la clase Empleado, en forma análoga, se actúa para los clientes. Los atributos y métodos comunes para empleados y clientes se asocian a la clase Persona.

Figura 3. Representación grafica de la clase persona



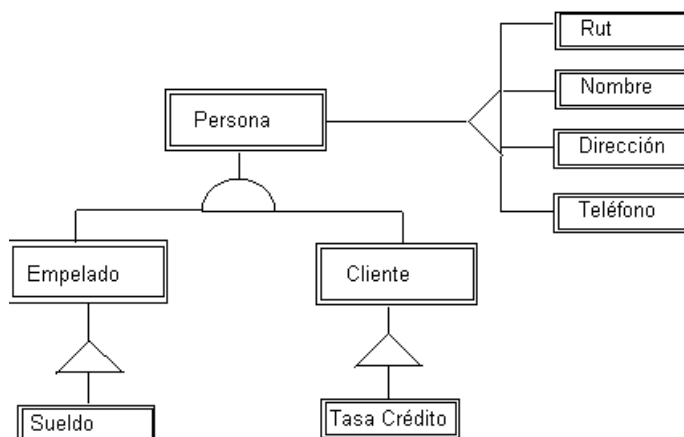
Observación. Se denotaran las clases por medio de rectángulos con línea doble, mientras que las instancias de las mismas u objetos, se representarán por rectángulos. El semicírculo denota la relación ES_UN, o ESPECIALIZACION de, que se utiliza para construir la jerarquía de clases (VIC).

Herencia. Una propiedad interesante para efectos de modelación, es la herencia. Esta se manifiesta cuando se tiene una clase que es una especialización, de otra que se encuentra en un nivel superior en la jerarquía de clases. La clase subordinada (en un nivel inferior) hereda los métodos, mensajes y atributos de la clase superior.

Cuando una clase que hereda de otra necesita incluir un atributo extra, éste se le asocia, y no afecta a la clase superior. Por otro lado, es común que en una clase que constituye una especialización, se desee modificar alguno de los métodos heredados para adecuar la respuesta a algún mensaje. En este caso se modifica el comportamiento de esta clase, y el método nuevo se le asocia. Ejemplo:

Retornando al caso del banco, podemos observar que como atributos para la clase persona se tiene ruc, nombre, dirección y teléfono. A su vez los empleados tienen un sueldo, y los clientes una tasa de crédito.

Figura 4. Representación grafica de la clase persona con herencias



En este caso, un objeto de la clase persona tiene como atributos RUC, Nombre, Dirección y Teléfono, un Empleado, hereda todos los anteriores y agrega el atributo Sueldo, mientras que el Cliente agrega el atributo Tasa Crédito.

Observación: El triángulo del diagrama denota la relación COMPUESTO_DE o INTEGRADO_POR, para indicar que los atributos de una clase pertenecen a las clases que se indican. En algunos formalismos para modelar clases y objetos existe la cardinalidad, para indicar el número de veces que un atributo puede aparecer en una clase.

En este caso no se han incluido los métodos pertenecientes a cada clase ni los mensajes a los que responde cada método. Esto se ha obviado por quedar fuera del ámbito de nuestro interés.

Identidad de objeto. En este modelo no necesitamos detectar los atributos que constituirán el identificador de un objeto. Aquí cada objeto conserva su identidad independientemente de su estado.

El estado de un objeto está dado por los valores que en un momento tengan sus atributos. En el caso de las otras técnicas, habíamos identificado un elemento de la realidad en base al valor de algún conjunto de sus atributos, por ejemplo el nombre de una persona o la patente de un auto. Aquí podemos utilizar el mismo

criterio para fines de búsqueda de información, pero a diferencia de los sistemas como los relacionales, aunque modifiquemos el nombre de un objeto persona, éste seguirá siendo el mismo objeto.

Análisis y diseño orientado a objetos. Durante los últimos años ha ido creciendo en forma considerable el análisis y diseño orientado a objetos. Se han publicado numerosos libros y muchas organizaciones están listas para implementar la práctica de esta nueva tecnología.

De un tiempo para acá ha venido presentándose un interés creciente en el campo del análisis orientado a objetos (AOO) y el diseño orientado a objetos (DOO). Este interés es debido a que la programación orientada a objetos (POO) se ha impuesto debido a sus enormes ventajas, pero las metodologías de análisis y diseño tradicional no son aplicables. Con la publicación de numerosos libros, los métodos se han estabilizado y ahora las organizaciones pueden moverse con tranquilidad a esta nueva tecnología.

Hay que tener en claro la idea de que un buen análisis puede acortar considerablemente la fase de desarrollo de programas, por ello, no se debe escatimar horas en organizar y estructurar la aplicación en cuestión. Este mensaje no sólo va dirigido para el analista - programador, sino también para algunos jefes de proyecto o gerente de empresas de desarrollo que se ponen literalmente nerviosos cuando ven que sus programadores están sentados mirando el techo, "sin hacer nada", y por consiguiente, no produciendo. Este es un gran error, sin duda. Que el programador tenga en claro que lo que va hacer es básico para que la aplicación sea realmente atractiva y competitiva, sobre todo para el usuario, que es el objetivo principal.

Vamos a ver en los párrafos siguientes en qué medida cambia el análisis y diseño a objetos, respecto del tradicional. Primeramente, desglosemos las diferentes partes en que se divide cada uno de ellos:

Análisis de Aplicación

- Análisis Funcional
- Análisis Orientado a Objetos (AOO)
- Análisis de Datos
- Análisis de las Reglas del Negocio

Diseño de Aplicación

- Esquema de Base de Datos
- Estructura de Clases (DOO)
- Interfaces del Usuario

Cuando enfocamos cualquier Aplicación hecha en un entorno gráfico, como un lenguaje visual, debemos tener presente como punto de referencia la interfaz del usuario que manejará los datos. El punto de vista del usuario es fundamental. Por tanto, cualquier suposición pasará siempre el tamiz de la opinión del mismo. Ahora, no debemos caer tampoco en una dependencia absoluta e intentemos reconducir al usuario por donde más nos interese. Es tarea del analista educar al usuario e informarle en un lenguaje sencillo de las posibilidades del sistema a implantar.

Análisis de la aplicación. Al llegar a nuestras manos la aplicación, debemos tomar los datos básicos que nos pueda darnos una idea del mismo.

Objetivo. Dar un comprobante de compra al cliente.

Cliente y usuario. Comercio de ventas de ropa, dedicado principalmente a la venta de ropa de caballeros. Los usuarios serán los vendedores de tienda con ninguna experiencia en el manejo de programas informáticos.

Hardware y software. Programa hecho en Power Builder en una máquina Pentium a 200 Mhz. sobre Windows 95.

Primera impresión. Parece un programa sencillo en su dinámica, pero difícil porque tiene que ser muy intuitivo para el usuario.

Primer paso. Hablar con el gerente de la tienda para hacerle ver la implantación del sistema, que tal vez resulte con problemas al principio.

En este punto se estaría en disposición de recoger toda la información posible sobre los productos que comercializa la tienda y cómo trabaja normalmente, haciendo especial hincapié en el deseo de mejora, sin olvidar los posibles temores. Haciendo un pequeño esquema de lo que sería el proceso de mecanización, conseguido a través del tradicional análisis funcional, y luego se comparará con al análisis orientado a objetos

Análisis funcional. Una de las técnicas utilizadas normalmente para el análisis funcional es el que busca y especifica las distintas funciones de trabajo del usuario final. Según esto, se puede desglosar el ejemplo en los siguientes puntos:

- El dependiente entrega el producto
- El cliente paga el producto

- El dependiente hace un recibo

Estas serían las funciones que describen el objetivo del programa. Ahora bien, de aca en adelante se mostrara como puede realizarse un Análisis Orientado a Objetos.

Análisis orientado a objetos (AOO). Como su propio nombre indica, lo que nos importa en este análisis es distinguir cuáles serán los objetos que van a ser parte de la aplicación. Tal vez ésta es la tarea más complicada del analista. En un primer momento, no debemos intentar enfocar con rigurosidad los objetos que nos puedan hacer falta en nuestra aplicación. Lo que haremos, un Brain Storming (tormenta de ideas que depuremos con posterioridad).

El primer paso que debemos hacer en un AOO será explicar con mayor claridad los puntos funcionales anteriormente definidos. Utilizamos para ello los escenarios.

Análisis mediante escenarios. Esta determinación viene a analizar la realidad como si de una película se tratase, viendo qué escenario se producen. Normalmente derivarán de los puntos funcionales.

- El dependiente toma el producto del almacén y lo entrega al cliente

- El dependiente comprueba el precio del producto, se lo dice al cliente y éste lo paga.

- El dependiente rellena un recibo con los datos del producto y lo entrega al cliente.

Determinación de objetos. La programación orientada a objetos se basa en la definición de clases. Una clase no es más que una abstracción de un objeto, o lo que es lo mismo, las características comunes de un grupo de objetos. Elegir qué conjunto de objetos son susceptibles de crear una abstracción, o sea, una clase, será la labor del analista. Por tanto buscaremos las clases que componen la aplicación, que luego se convertirá en objetos.

Una forma fácil de localizar objetos es encontrar en la aplicación:

Cosas

Lugares

Personas o funciones

Eventos

Conceptos

Organizaciones

Según esto, en nuestro ejemplo podemos conseguir los siguiente objetos:

Cosas Recibí (ticket)

Conceptos Productos

Lugares Almacén

Personas Dependiente, Clientes

Se podría, por lo tanto, a partir de aquí, crear unas clases candidatas, es decir que posiblemente serán las clases utilizadas en la aplicación. Cuando definimos las clases tenemos que indicar cuáles van a ser los métodos y sus propiedades, así como los términos específicos de la programación orientada a objetos. Para aquellos que no están acostumbrados a esta terminología lo podemos comparar con estos procedimientos y variables que deben contener un módulo, donde se pueda tomar la licencia de igualar clase con módulo, procedimiento con método y variable con propiedad.

Análisis de las reglas del negocio. Es interesante que algunas aplicaciones técnicas puedan documentar todos los algoritmos empleados por el usuario. Estas son las reglas del negocio. Una de las particularidades del trabajo del analista es que debe saber "un poco de todo" y el usuario siempre supone que nosotros sólo sabemos ese poco, si no estamos al corriente de en qué consiste su trabajo. Es aquí donde debemos recopilar todas las peculiaridades del "modos operandi" del usuario.

En la aplicación, por ejemplo, se hablará de tallas grandes, medianas y pequeñas; de una codificación de almacén, etc. En otros programas, tal vez tengamos que trabajar siempre en dólares en vez de soles , o siempre en centímetros, por lo que debemos tener unos algoritmos de conversión que será la regla del negocio. Una idea interesante. Si éstas son muchas, es crear una tabla con dichas reglas y acudir a ella cuando haga falta.

Diseño orientado a objetos. El enfoque particular del análisis orientado a objetos, modela la forma en que las personas comprenden y procesan la realidad a través de los conceptos que adquieren. Estos conceptos se pueden implantar por diversos medios, entre los que están las máquinas, computadoras y personas. Así la implantación puede incluir válvulas, tuberías, dispositivos de medición, circuitos, compuertas, controladores, flujo de materiales, reglas de dirección, etc.

El objetivo del diseño se restringe al software de aplicación.

Conceptos básicos: Asociación a partir de esquemas con código oo. La producción de diseños de un Lenguaje de Programación Orientado a Objetos (LPOO) con los esquemas de análisis OO presentados, es muy directa. Los

lenguajes de programación OO tienen una estructura similar a nuestra forma de pensar. Este es uno de los beneficios de los LPOO.

La conversión de términos del AOO a las construcciones del LPOO se pueden describir de manera básica mediante la siguiente tabla:

Tabla 1. Conversiones del AOO a las construcciones del LPOO

Término en el esquema de objetos	Asociación en el LPOO
Tipo de objeto	Clase
Objeto	Objeto
Función básica	Campo
Jerarquía de generalización	jerarquía de clase/subclase con
	Herencia
Función calculada (o atributo calculado)	Operación
Tipo de objeto calculado	Operación
Operación	Operación
Regla de activación	tipo de solicitud soportado por un planificador de eventos
Tipo de evento	No tiene equivalente, soportado por un planificador de eventos
Esquema de eventos	Método o planificador de evento

Asociación a partir de los tipos de objetos. Al implantar los tipos de objetos como clases del LPOO, el diseñador también debe garantizar que se disponga de dos tipos de operaciones fundamentales para cada clase, las que construyen instancias de clases y las que destruyen instancias de clases.

Construcción de un objeto. Cuando una clase construye sus instancias, asigna cierto espacio para almacenarlas y configura los valores iniciales de los atributos o campos del objeto. Al hacer esto, no sólo da existencia al objeto sino que cumple con todos los requisitos. Por ejemplo para construir un objeto empleado, una operación de la clase empleado debe primero asignar espacio para el objeto.

Destrucción de un objeto. La destrucción de un objeto también se debe adherir a las restricciones definidas en el esquema de objetos. Por ejemplo anterior, cada objeto empleado tiene una asociación obligatoria con un objeto organización. Al

eliminar el objeto organización con el que está ligado un empleado, se viola la restricción de cardinalidad de empleado.

Principales herramientas y tecnologías.

Case Computer-Aided Software Engineering
(Ingeniería del Software Asistido por Computadora)

Las herramientas Case. Software que se utiliza en una o cualquiera de las fases de desarrollo de sistemas de información, incluyendo análisis, diseño y programación. Por ejemplo las herramientas de diagramación ayudan en las fases de análisis y diseño mientras que los generadores de aplicaciones aceleran la fase de programación.

Las herramientas CASE proporcionan métodos automáticos para diseñar y documentar las técnicas tradicionales de Programación Orientada a Objetos. La meta es proveer un lenguaje para describir un sistema complejo, que sea suficiente para generar todos los programas necesarios.

Case integrate case. (Cases Integrados) Un I-Case se refiere a un conjunto de herramientas Case que soportan todas las fases del ciclo de vida incluyendo la generación de código, con un repositorio único y consistente.

Programación Visual. La programación Visual es una forma de case que expresa el diseño de programa con gráfico, color y posiblemente sonido. Los objetos se representan en forma visual y pueden ser concebidos como si fueran máquinas físicas que pasan de un estado a otro.

La programación visual permite a los desarrolladores, ingresar, comprender "pensar con", ejecutar y manipular programas, usando fundamentalmente notaciones gráficas.

Máquina de Inferencia. Es un software que hace de hechos y reglas usando técnicas de inferencia lógica. Una máquina de inferencia opera con una colección de reglas acerca de un área del conocimiento. Selecciona las reglas y el encadenamiento hacia adelante (razonamiento orientado a ingreso input), el encadenamiento hacia atrás (razonamiento orientado a metas) o ambos permite a las computadoras hacer deducciones complejas sin un programa de aplicación. Es una técnica principal usada en Inteligencia Artificial.

Generadores de Código. Siempre que sea posible, los programadores deben ser generados automáticamente a partir de diseños de alto nivel, especificaciones o

imágenes en una pantalla de un CASE, El código puede generarse a partir de las tablas de decisión, reglas diagramas de acción, diagramas de eventos, diagramas de transición de estado, representaciones de objetos y sus propiedades y relaciones, etc. Los generadores de código producen códigos libres de errores de sintaxis a partir de diseños de alto nivel, cuadro o especificaciones.

Metodología basada en Repositorio. El desarrollo basado en el software se basará cada vez más en lo repositorios. Un repositorio continuamente creciendo, almacena modelos especificaciones, diseños o construcciones reusables, a partir de los cuales se construyen software.

Actualmente, la metodología para el desarrollo de sistemas está basada en la construcción de un conjunto comprensivo de conocimientos de repositorio, donde el repositorio produce generadores de código. Una creciente colección de constructores reusables se almacenará en el repositorio.

Una metodología basada en repositorio está diseñada para aprovechar al máximo las herramientas CASE-I y para maximizar la reusabilidad. Las metodologías basadas en repositorios son muy importantes en la etapa de codificación, para lograr un desarrollo acelerado de la aplicación.

Ingeniería de la información. La ingeniería de la información emplea la metodología basada en repositorio a una empresa completa. La ingeniería de la información emplea modelamiento integrado y técnicas de diseño de la empresa como un todo (o para un área grande de la empresa) y no para un solo proyecto.

Base de datos orientado a objetos. Una base de datos orientada a objetos es una base de datos inteligente. Soporta los paradigmas orientados a objetos, almacenamientos de métodos y datos, más que simplemente datos. Esta diseñada para ser físicamente eficiente en el almacenamiento de datos complejos. Un repositorio en un base de datos orientada a objetos (aún cuando las herramientas CASE no soporten análisis, diseño y programación orientada a objeto).

Tecnología cliente/servidor. Es un software que ejecuta en múltiples computadoras tales como sistemas servidoras LAN, sistemas cooperativos, procesamiento distribuido y computadoras en paralelo. La tecnología cliente/servidor puede describirse como sigue: Un cliente es un módulo de software que solicita una operación. Un servidor es un módulo de software que atiende el requerimiento, cliente/servidor y puede ejecutarse en máquinas separadas dispuestas en casi cualquier configuración.

Lenguajes no procedurales. Con los Lenguajes Procedurales (tal como Cobol o C), nosotros instruimos al computador cómo debe ejecutar una secuencia de operaciones. Con los Lenguajes No-Procedurales, definimos los resultados requeridos. Un intérprete o compilador genera el código.

Los lenguajes No-Procedurales definen qué es lo que se quiere y no cómo debe ser programado. Muchos lenguajes de cuarta generación tales como FOCUS o NATURAL contienen elementos procedurales y no-procedurales.

Métodos basados en matemáticas formales. Las técnicas matemáticas han sido utilizadas para crear especificaciones de probable exactitud y proporcionan la corrección de programas. Los métodos formales son utilizados para garantizar que los programas realicen su función libre de errores. Los métodos formales son apropiados para crear los bloques de construcción (clases) de los Sistemas Orientados al Objeto.

Inferencia lógica. Una inferencia lógica opera con una serie de reglas del área de conocimiento.

Selecciona reglas y las encadena en forma efectiva para realizar un razonamiento inferencial. Se puede utilizar encadenamiento hacia delante (razonamiento dirigido por INPUTS) o encadenamiento reverso (razonamiento dirigido por OBJETIVOS) o ambos.

Esto capacita a un computador a hacer deducciones complejas sin un programa de aplicación. Es la técnica fundamental utilizada en software de Inteligencia Artificial.

Una Inferencia Lógica es un software que hace las deducciones a partir de hechos y reglas, utilizando técnicas de Inferencia Lógica.

Librerías de clases. Una librería de clases contiene implementaciones reusables de tipo de objeto. Su objetivo consiste en conseguir el máximo grado de reusabilidad en el desarrollo de software. El software con librería de clases debe ayudar al programador a encontrar, adaptar y reusar las clases que necesita para la aplicación que él desarrolle.

Beneficios de la tecnología orientada a objetos. Muchos de los beneficios son alcanzados únicamente cuando el Análisis y Diseño son utilizados con herramientas CASE OO, basados en repositorios que generan códigos.

Reusabilidad. Las clases son diseñadas de tal manera que ellas puedan ser reutilizadas en muchos sistemas. Para maximizar el reuso las clases deben ser

construidas de manera que puedan ser personalizadas. Un repositorio debería ser cargado con una colección de clases reusables. Un objetivo permanente de las técnicas OO, es conseguir reusabilidad masiva en la construcción de software.

Estabilidad. Las clases diseñadas para el reuso repetido, llegan a ser estables de la misma manera que los microprocesadores y otros chips que son bastante estables. Las aplicaciones serán construidas utilizando chips de software.

El diseñador piensa de comportamiento de objeto, no en niveles de detalle. El encapsulamiento oculta los detalles y hace fácil el uso de clases complejas. Las clases son semejantes a las cajas negras. El desarrollador utiliza la caja negra sin mirar su interior. El tiene un entendimiento del comportamiento de la caja negra y cómo comunicarse con ella.

Construcción de objetos de complejidad creciente. Los objetos se construyen fuera de los objetos. Una buena manera de fabricar es construir tomando una lista de materiales de partes y subpartes existentes. Esto posibilita construir componentes de software complejos y los mismos se utilizarán para construir otros bloques de software más complejos.

Confiabilidad. EL software construido a partir de una librería de clases estables, es probable que se encuentre libre de errores, respecto a construir software desde el inicio. Cada método en una clase es en sí mismo simple y diseñado para ser confiable.

Verificación de correcciones. El Diseño OO con técnica formal para la creación de métodos, puede generar potencialmente software de alta confiabilidad. Técnicas para verificar y garantizar la operación correcta de una clase, probablemente estén disponibles en nuevas generaciones de herramientas CASE OO.

Diseño rápido. Las aplicaciones son creadas tomando componentes pre-existentes. Muchos componentes son construidos de tal forma que, puedan ser observados, personalizados, para un diseño particular. Los componentes pueden ser vistos, customizados y enlazados en la pantalla de la herramienta CASE.

Nuevos Mercados de Software. Las compañías de software, deberían proporcionar librerías de clases para áreas específicas, fácilmente adaptables a las necesidades de la organización. La era de los paquetes monolíticos esta siendo reemplazada por software que incorpora clases y encapsula paquetes de diferentes vendedores.

Diseño de alta calidad. Los diseños son a menudo de alta calidad, ya que ellos se construyen a partir de componentes que han sido aprobados y refinados repetidamente.

Integridad. Las estructuras de Datos pueden ser utilizadas solamente con métodos específicos. Esto es particularmente importante en sistemas distribuidos y sistemas CLIENTE/SERVIDOR, donde usuarios desconocidos pueden tratar de acceder al sistema.

Facilidad de Programación. Los programas son construidos utilizando pequeñas plazas de software las cuales son generalmente fáciles de crear.

Fácil Mantenimiento. Los programas de mantenimiento generalmente cambiarán los métodos correspondientes a una clase. Cada clase realiza sus operaciones independientemente de otras clases.

Creatividad. Implementadores hábiles en poderosas herramientas CASE OO laborando sobre estaciones de trabajo, encuentran que puede generar rápidamente muchas ideas. Las herramientas estimulan la creación e implementan las invenciones. La genialidad individual puede ser más creativa.

Ciclo de Vida Dinámico. Los objetivos de desarrollo de un sistema, a menudo cambian durante la implementación. Las herramientas CASE OO, hacen los cambios durante el ciclo de vida rápidamente.

Esto permite a los diseñadores de sistemas satisfacer mejor a los usuarios finales, adaptarse a los cambios, refinar los objetivos y mejorar constantemente el diseño durante la implementación.

Refinamiento durante la Construcción. Las personas creativas cambian constantemente el diseño de su trabajo mientras se está implementando. Esto conduce a más y mejores resultados. Los trabajos creativos objetivos, son una y otra vez refinados. Las herramientas CASE OO (Orientado a Objetos) proporcionan a los constructores de software la capacidad para refinar el diseño durante la implementación.

Modelamiento más realístico. El AOO modela la empresa o área de negocio de una manera más coherente y minuciosa que los métodos tradicionales de análisis. El análisis se traslada directamente al diseño e implementación. En técnicas convencionales, el entorno del problema cambia cuando vamos del análisis al diseño y del diseño a la programación. Con técnicas de OO Análisis, Diseño e implementación, utiliza el mismo paradigma y lo refinan sucesivamente.

Mejor comunicación entre profesionales de informática y los usuarios finales. Los usuarios finales entienden mejor el paradigma OO. Ellos piensan en términos de eventos, objetos y políticas de negocios que describen el comportamiento de los objetos. Las metodologías OO estimulan el mejor entendimiento, cuando el usuario final y los desarrolladores comparten un modelo común.

Modelos Inteligentes de la Empresa. Los modelos de la empresa debería escribir las reglas del negocio con las cuales el ejecutivo desearía administrarla. Esto debería ser expresado en términos de eventos y de cómo éstos cambian el estado de los objetos del negocio. El diseño de la aplicación debería ser derivando automáticamente como sea posible, el modelo del negocio.

Especificaciones y Diseños Declarativos. La especificación y el diseño, construido con la formalidad de las herramientas CASE, deberían ser declarativas tanto como sea posible, fijando explícitamente lo que es solicitado. Esto permite al diseñador pensar en el usuario antes que en el computador.

Interfase gráfica seductiva al usuario. Se debería utilizar interfaces gráficas para usuarios, tal que ésta apunte al icono que relacione al objeto.

Independencia de diseño. Las clases son diseñadas independientemente de plataforma de operación, hardware o software.

Las clases emplean requerimientos y respuestas de forma. Esto permite que ellos sean utilizados con múltiples sistemas operativos, DBMS, manejadores de redes, interfaces gráficas para usuarios, etc.

Interoperatividad. Software de diferentes vendedores pueden trabajar juntos. Un vendedor puede utilizar clase de otros vendedores. La interoperatividad de software de diferentes vendedores es uno de los objetivos más importantes de los estándares de la OO. Software desarrollado independientemente en lugares separados, deberían ser capaces de trabajar juntos y presentarse como una unidad simple al usuario.

Computación cliente / servidor. En el sistema Cliente / Servidor, las clases en el software cliente deberían enviar sus requerimientos a las clases de software servidor y recibir respuestas. Una clase servidor puede ser utilizada por muchos clientes. Esto puede acceder al software únicamente a través de los métodos (así los datos se protegen de corrupciones).

Computación masivamente distribuida. Redes alrededor del mundo emplearán directorios de software de objetos accesibles. El diseño orientado al objeto, es la clave para la computación masivamente distribuida. Las clases en una máquina interactuarán con cualquier otra, sin necesidad de saber dónde residen. Ellas envían y reciben mensajes en formatos estándares.

Computación paralela. La velocidad de las máquinas, pueden ser ampliamente mejoradas mediante la instalación de computadoras en paralelo. Se pueden tener procesamientos simultáneos y concurrentes en múltiples chips de procesadores (eventualmente, un chip puede tener muchos procesadores). Objetos en diferentes procesadores se ejecutarán simultáneamente, cada uno de ellos actuando independientemente.

Alto nivel de automatización de bases de datos. Las estructuras en Base de Datos OO, están ligadas a métodos que toman acciones automáticas. Una Base de Datos OO, tiene su inteligencia construida en la forma de métodos, mientras que otras bases de datos no.

Performance de Máquinas. Las Bases de Datos Orientada a Objetos han demostrado una mayor performance que las bases de datos relacionales para ciertas aplicaciones con estructuras de datos más complejas. Las bases de datos OO, la computación concurrente y el diseño OO prometen mayores saltos en la performance de las máquinas LAN'S basadas en sistemas Cliente/Servidor. Emplearán servidores de Base de Datos concurrentes y orientadas al objeto.

Migración. Existiendo o no aplicaciones orientadas a objetos, ellos pueden ser preservados convenientemente con una cobertura OO, comunicándose entre ellos mediante mensajes estándares OO.

Mejores herramientas CASE. Las herramientas Case utilizarán técnicas gráficas para diseñar las clases y sus interacciones, y para utilizar objetos existentes adaptados en nuevas aplicaciones. Las herramientas deberían facilitar el modelamiento en términos de eventos, triggers (iniciadores), estado de los objetos, etc. Las herramientas de los CASE OO generan códigos tan pronto como una clase sea definida y permitirá al diseñador probar y utilizar el método creado. Las herramientas deberán ser diseñadas para estimular la máxima creatividad y continuo refinamiento del diseño durante la construcción.

Industriales de Librerías de Clases. Las compañías de software comercializarán librerías para diferentes áreas de aplicación. Las librerías de clases independientes de las aplicaciones, serán también importantes y éstas serán proporcionadas como facilidades de herramientas CASE (VIC).

Librerías de Clases Corporativas. Las corporaciones, crearán sus propias librerías de clases que reflejen sus estándares internos y requerimientos de aplicación. La identificación TOP-DOWN de los OBJETOS del negocio, es un aspecto importante de la ingeniería de la Información.

4.1.2 Lenguaje unificado para la construcción de modelos u.m.l.

Elementos notacionales de UML. A Continuación se revisan en niveles de complejidad los diversos elementos notacionales que presenta el Unified Modeling Language. Estos elementos pretenden ser un lenguaje común para el modelamiento de cualquier sistema. Esta descripción no pretende ser exhaustiva en terminos sintácticos, semánticos y de presentación de los elementos de la notación. Debe entenderse como una guía inicial al tema.

Se agrupan los conceptos básicos por tipo de diagrama:

- Diagrama de Estructura Estática
- Diagrama de Casos de Uso
- Diagrama de Secuencia
- Diagrama de Colaboración
- Diagrama de Estados
- Diagrama de Actividades
- Diagrama de Implementación

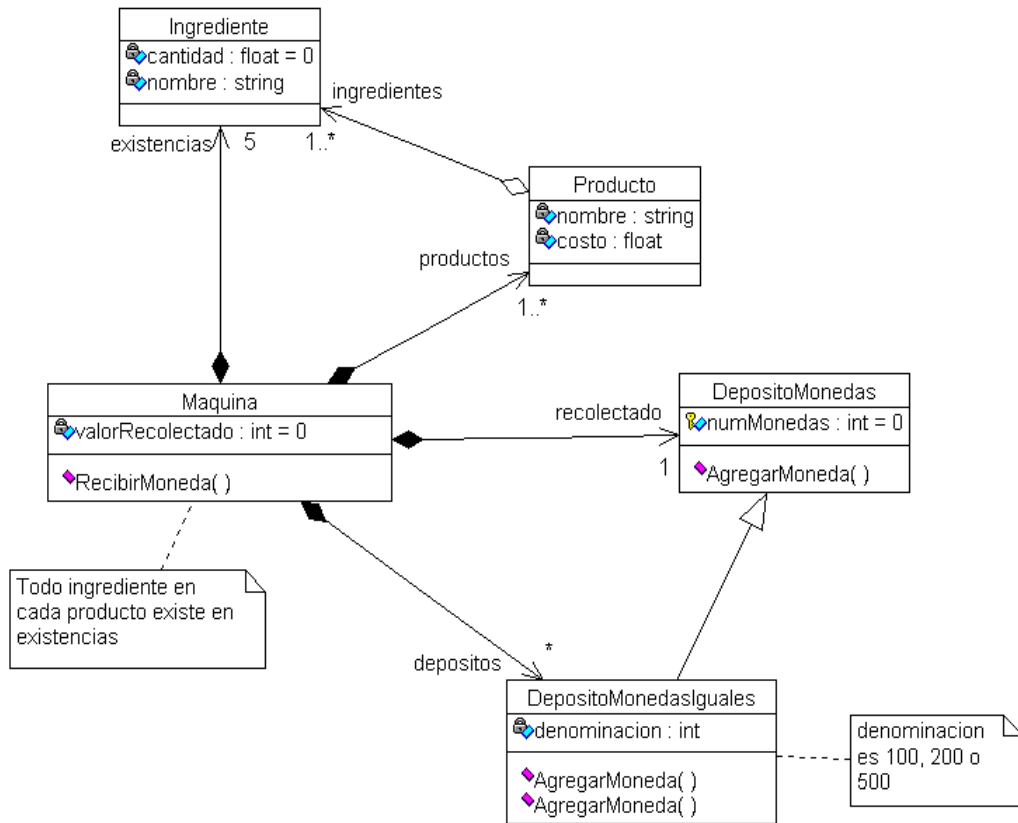
Los conceptos avanzados por tipo de diagrama son:

- Diagrama de Estructura Estática
- Diagrama de Secuencia
- Diagrama de Colaboración
- Diagrama de Estados

Elementos básicos en un diagrama de estructura estática. Un diagrama de estructura estática muestra el conjunto de clases y objetos importantes que hacen parte de un sistema, junto con las relaciones existentes entre estas clases y objetos. Muestra de una manera estática la estructura de información del sistema y la visibilidad que tiene cada una de las clases, dada por sus relaciones con las demás en el modelo.

Supongamos el modelamiento de una máquina de café. Un diagrama de estructura estática inicial podría ser:

Figura 5. Representación grafica de la construcción de un modelo en UML



Clase: Representada por un rectángulo con tres divisiones internas, son los elementos fundamentales del diagrama. Una clase describe un conjunto de objetos con características y comportamiento idéntico. En el ejemplo se encuentran las clases **Ingrediente**, **Producto**, **Maquina**, **DepositoMonedas** y **DepositoMonedasIguales**. Los tres compartimientos estándares alojan el nombre de la clase, sus atributos y sus mensajes, respectivamente.




Atributo: Identifican las características propias de cada clase. Generalmente son de tipos simples, ya que los atributos de tipos compuestos se representan mediante asociaciones de composición con otras clases. La sintaxis de un atributo es:

visibility name : type-expression = initial-value { property-string }

Donde visibility es uno de los siguientes:

- + public visibility
- # protected visibility
- private visibility

type-expression es el tipo del atributo con nombre name. Puede especificarse como se ve un valor inicial y un conjunto de propiedades del atributo.

En el caso del ejemplo, la clase Ingrediente tiene dos atributos: uno denominado cantidad, de tipo float y con valor inicial 0; y el atributo nombre de tipo string sin valor inicial. En este caso, la herramienta utilizada ha cambiado la representación de la visibilidad, utilizando el símbolo  para indicar visibilidad privada, el símbolo  para visibilidad protegida y el símbolo  para indicar visibilidad pública.

Operación: El conjunto de operaciones describen el comportamiento de los objetos de una clase. La sintaxis de una operación en UML es
visibility name (parameter-list) : return-type-expression { property-string }

Cada uno de los parámetros en parameter-list se denota igual que un atributo. Los demás elementos son los mismos encontrados en la notación de un atributo.

Asociación (rol, multiplicidad, cualificador)

Una asociación en general es una línea que une dos o más símbolos. Pueden tener varios tipos de adornos, que definen su semántica y características. Los tipos de asociaciones entre clases presentes en un diagrama estático son:

- Asociación binaria
- Asociación n-aria
- Composición
- Generalización
- Refinamiento

Cada asociación puede presentar algunos elementos adicionales que dan detalle a la relación, como son:

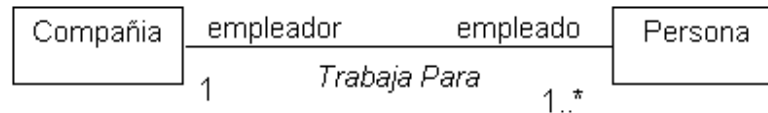
Rol: Identificado como un nombres al los finales de la linea, describe la semántica de la relación en el sentido indicado. Por ejemplo, la asociación de composición entre Maquina e Ingrediente recibe el nombre de existencias, como rol en ese sentido

Multiplicidad: Describe la cardinalidad de la relación. En el ejemplo anterior se utilizan 1, 1 ..*, 5, *, como indicadores de multiplicidad.

Asociación binaria: Se identifica como una línea sólida que une dos clases. Representa una relación de algún tipo entre las dos clases, no muy fuerte (es decir, no se exige dependencia existencial ni encapsulamiento).

En posible ejemplo es la relación entre una compañía y sus empleados

Figura 6. Relación entre una compañía y sus empleados



En este caso la relación recibe el nombre genérico Trabaja Para, la compañía tiene uno o más instancias de la clase Persona denominadas empleado y cada empleado conoce su empleador (en este caso único).

Composición: Es una asociación fuerte, que implica tres cosas: dependencia existencial. El elemento dependiente desaparece al destruirse el que lo contiene y, si es de cardinalidad 1, es creado al mismo tiempo.

Hay una pertenencia fuerte. Se puede decir que el objeto contenido es parte constitutiva y vital del que lo contiene

Los objetos contenidos no son compartidos, esto es, no hacen parte del estado de otro objeto.

Se denota dibujando un rombo relleno del lado de la clase que contiene a la otra en la relación. En el ejemplo inicial de esta hoja se presentan varios ejemplos de relaciones de composición entre Maquina y Producto, Maquina y DepositoMonedas y Maquina y DepositoMonedasIguales.

Existe también una relación de composición menos fuerte (no se exige dependencia existencial, por ejemplo) que es denotada por un rombo sin rellenar en uno de los extremos. Un ejemplo puede encontrarse entre Producto e Ingrediente.

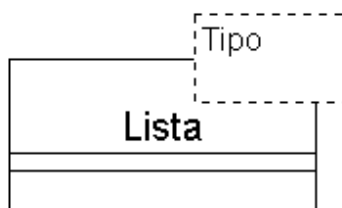
Generalización: La relación de generalización denota una relación de herencia entre clases. Se representa dibujando un triángulo sin rellenar en el lado de la superclase. La subclase hereda todos los atributos y mensajes descritos en la superclase. En el ejemplo se encuentra una generalización entre DepositoMonedas (superclase) y DepositoMonedasIguales (subclase).

Clase Paramétrica: Una clase paramétrica representa el concepto de clase genérica en los conceptos básicos OO o de template en C++. Se dibuja como una

clase acompañada de un rectángulo en la esquina superior derecha, con los parámetros del caso.

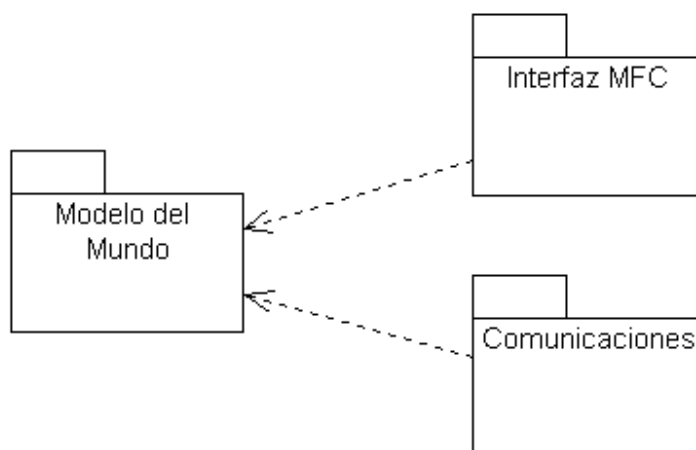
Por ejemplo, la clase Lista que utiliza un parámetro formal Tipo se vería de la siguiente manera

Figura 7. Representación gráfica una clase parametrica



Paquete: Un paquete es una forma de agrupar clases (u otros elementos en otro tipo de diagramas) en modelos grandes. Pueden tener asociaciones de dependencia o de generalización entre ellos. Un ejemplo puede ser el siguiente:

Figura 8. Representación gráfica de un paquete



En este caso existen tres paquetes (que se muestran varios en este caso, con su contenido encapsulado), con dos de ellos dependiendo del Modelo del Mundo.

Dependencia: Denota una relación semántica entre dos elementos (clases o paquetes, por el momento) del modelo. Indica que cambiar el elemento independiente puede requerir cambios en los dependientes. Se muestra como una línea punteada direccional, indicando el sentido de la dependencia. Puede tener por medio de estereotipos una explicación del tipo de dependencia presentada.

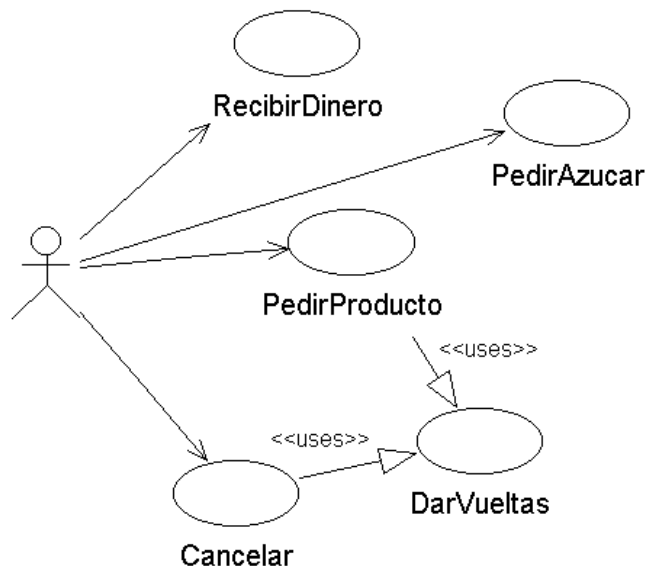
En el ejemplo anterior pueden verse dos relaciones de dependencia hacia el paquete Modelo del Mundo.

Nota: Es un comentario dentro de un diagrama. Puede estar relacionado con uno o más elementos en el diagrama mediante líneas punteadas. Pueden representar aclaraciones al diagrama o restricciones sobre los elementos relacionados (cuando el texto se encuentra entre '['y ']'). Se representa mediante un rectángulo con su borde superior derecho doblado.

En el ejemplo inicial de esta hoja se encuentran dos notas: Una relacionada con la clase máquina y otra con el depósito de monedas iguales.


Conceptos de un diagrama de casos de uso. Un diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones). Se muestra como ilustración los casos de uso de la máquina de café

Figura 9. Representación gráfica de los diagramas de casos de uso.



Caso de uso: Se representa en el diagrama por una elipse, denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema. Va acompañado de un nombre significativo. En el caso del ejemplo se tienen como casos de uso de la cafetera RecibirDinero, PedirAzucar, PedirProducto, DarVueltas y Cancelar.

Actor: Es un usuario del sistema, que necesita o usa algunos de los casos de uso.

Se representa mediante un , acompañado de un nombre significativo, si es necesario.

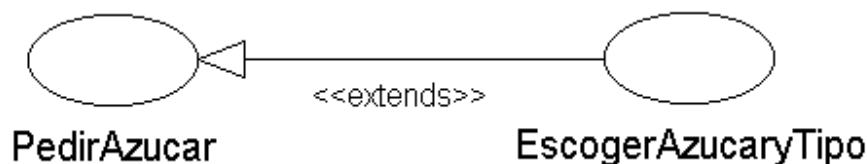
Relaciones en un diagrama de casos de uso: Entre los elementos de un diagrama de Casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas entre ellos (del elemento dependiente al independiente)

Communica (communicates). Relación entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado. En el diagrama de ejemplo, todas las líneas que salen del actor denotan este tipo de relación.

Usa (uses). Relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. En el caso del ejemplo el caso de uso Cancelar incluye en su comportamiento DarVueltas; y PedirProducto incluye también DarVueltas

Extiende (extends). Relación entre dos casos de uso, denota cuando un caso de uso es una especialización de otro. Por ejemplo, podría tenerse un caso de uso que extienda la forma de pedir azucar, para que permita escoger el tipo de azuacr (normal, dietético moreno) y además la cantidad en las unidades adecuadas para cada caso (cucharaditas, bolsitas o cucharaditas, respectivamente). Un posible diagrama se muestra a continuación.

Figura 10. Representación grafica de relaciones en diagramas de casos de uso

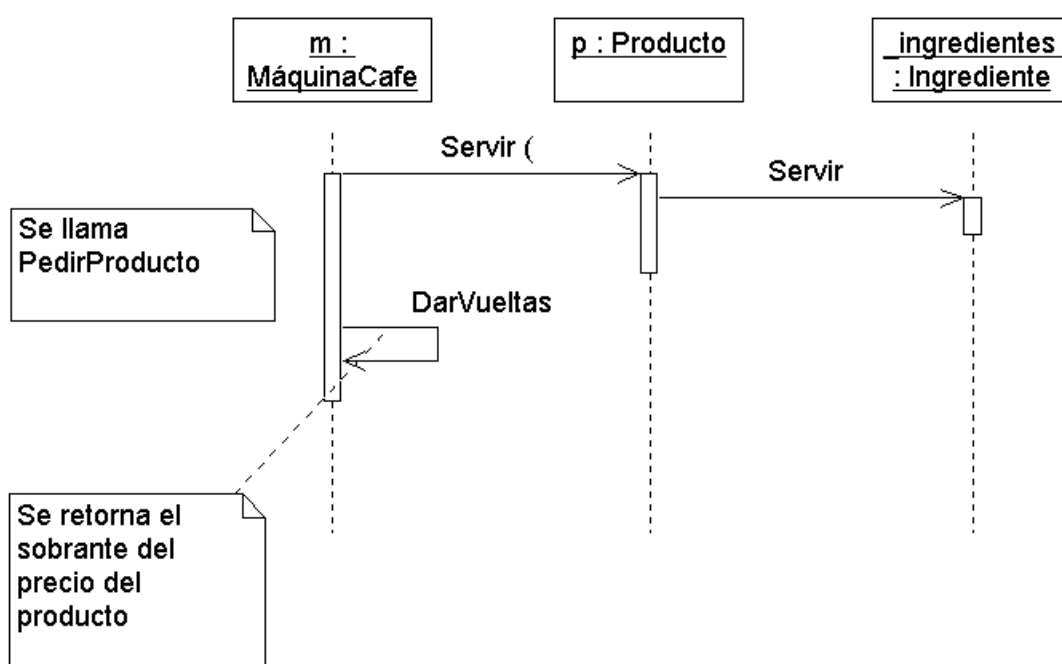


Conceptos básicos en un diagrama de secuencia: Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del

tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación.

A continuación se muestra un ejemplo de diagrama de secuencia, que da detalle al caso de uso PedirProducto del ejemplo de la cafetera.

Figura 11. Representación gráfica de los diagramas de secuencia



Línea de vida de un objeto. Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos (véase activación). El rectángulo de encabezado contiene el nombre del objeto y el de su clase, en un formato nombreObjeto: nombreClase. Por ejemplo, el objeto m, instancia de la clase MáquinaCafe envía dos mensajes seguidos para dar respuesta a la operación PedirProducto: Servir al objeto p de la clase Producto y DarVueltas a sí mismo.

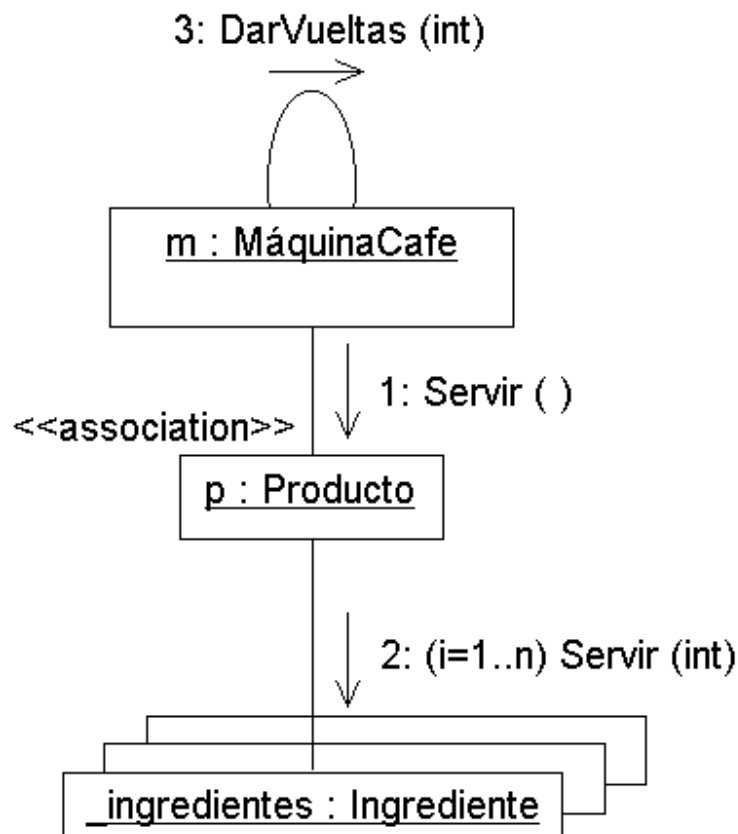
Activación. Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto. En el ejemplo anterior el objeto _ingredientes se encuentra activado mientras ejecuta el método correspondiente al mensaje Servir; el objeto p

se encuentra activo mientras se ejecuta su método Servir (que ejecuta `_ingredientes.Servir`) y el objeto `m` se encuentra activo mientras se ejecuta `p.Servir` y `DarVueltas`.

Mensaje. El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta. En el ejemplo anterior el objeto `m` envía el mensaje `Servir` al objeto `p` y un poco más adelante en el tiempo el objeto `m` se envía a sí mismo el mensaje `DarVueltas`.

Conceptos básicos en un diagrama de colaboración. Un diagrama de colaboración es una forma de representar interacción entre objetos, alterna al diagrama de secuencia. A diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales, ...) y ciclos en la ejecución. Se toma como ejemplo el caso de uso `PedirProducto` ya descrito como diagrama de secuencia.

Figura 12. Representación gráfica del diagrama de colaboración



Objeto. Un objeto se representa con un rectángulo, que contiene el nombre y la clase del objeto en un formato nombreObjeto: nombreClase.

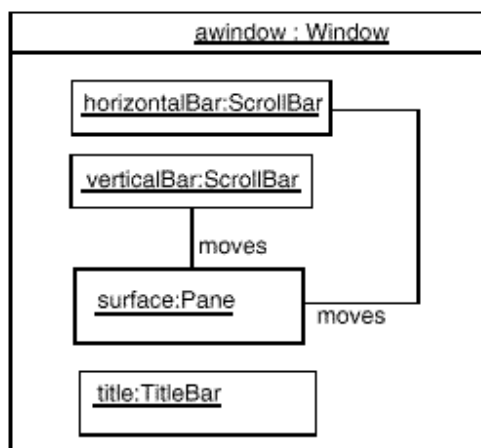
Enlaces. Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos. Esta acompañada por un número que indica el orden dentro de la interacción y por un estereotipo que indica que tipo de objeto recibe el mensaje. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Los estereotipos indican si el objeto que recibe el mensaje es un atributo (association y se asume por defecto), un parámetro de un mensaje anterior, si es un objeto local o global.

Flujo de mensajes. Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace.

Marcadores de creación y destrucción de objetos. Puede mostrarse en la gráfica cuáles objetos son creados y destruidos, agregando una restricción con la palabra new o delete, respectivamente, cercana al rectángulo del objeto

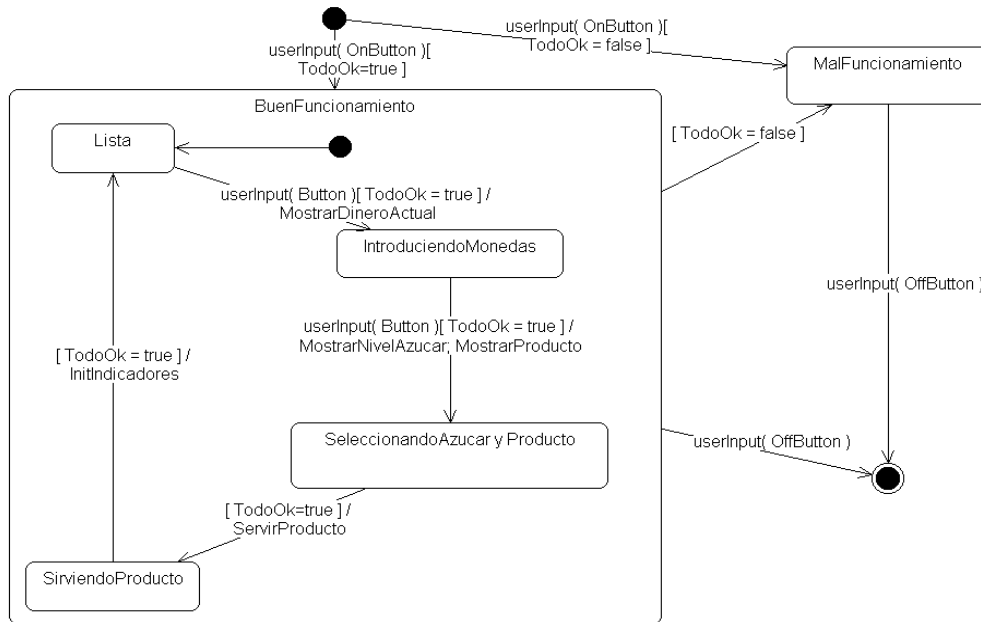
Objeto compuesto. Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. Un ejemplo es el siguiente objeto ventana

Figura 13. Representación gráfica del objeto ventana.



Conceptos básicos en un diagrama de estados. Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Un ejemplo en el caso de la cafetera son los estados posibles para la clase MaquinaCafe:

Figura 14. Representación gráfica de un diagrama de estados



Estado. Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se representa mediante un rectángulo con los bordes redondeados, que puede tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado (entry, exit o do, respectivamente). En el caso del ejemplo anterior, se tienen cuatro estados (EnFuncionamiento, SinCambio, SinIngredientes, MalFuncionamiento), en los cuales se desarrollan ciertas acciones al entrar; por ejemplo, al entrar al estado SinIngredientes se debe realizar la acción "Indicador SinIngredientes en On".

Se marcan también los estados iniciales y finales mediante los símbolos

y  , respectivamente.

Eventos. Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- Condición que toma el valor de verdadero o falso
- Recepción de una señal de otro objeto en el modelo
- Recepción de un mensaje

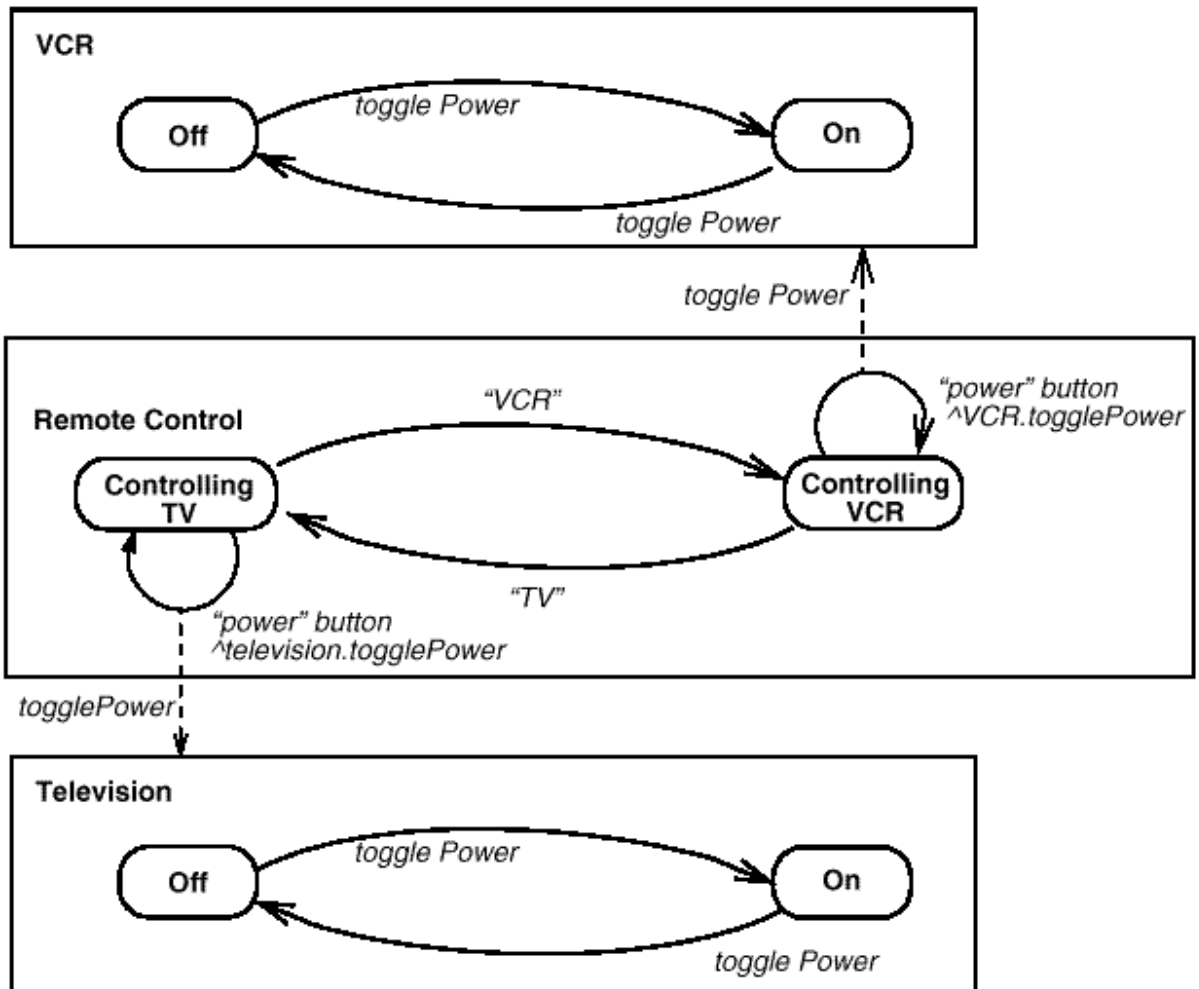
Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular.

El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre.

En el caso del ejemplo anterior se encuentra nombrado en varias transiciones el evento `userInput`, que recibe como parámetro un `Button`, para indicar el botón que ha sido presionado por el usuario de la máquina de café.

Envío de mensajes. Además de mostrar y transición de estados por medio de eventos, puede representarse el momento en el cual se envían mensajes a otros objetos. Esto se realiza mediante una línea punteada dirigida al diagrama de estados del objeto receptor del mensaje. Si tomamos como ejemplo un control remoto que puede enviar órdenes de encender o apagar al televisor o a la videograbadora se puede obtener un diagrama de estados como el siguiente:

Figura 15. Representación gráfica del envío de mensajes entre estados.



Los tres aparatos tienen diagramas de estados separados y algunas de las transiciones del control remoto causan el envío de mensajes (*togglePower*) a los otros aparatos.

Transición simple. Una transición simple es una relación entre dos estados que indica que un objeto en el primer estado puede entrar al segundo estado y ejecutar ciertas operaciones, cuando un evento ocurre y si ciertas condiciones son satisfechas. Se representa como una línea sólida entre dos estados, que puede venir acompañada de un texto con el siguiente formato:

event-signature `'[guard-condition]'` action-expression `'^` send-clause
 event-signature es la descripción del evento que da lugar a la transición, guard-condition son las condiciones adicionales al evento necesarias para que la transición ocurra, action-expression es un mensaje al objeto o a otro objeto que se

ejecuta como resultado de la transición y el cambio de estado y send-clause son acciones adicionales que se ejecutan con el cambio de estado, por ejemplo, el envío de eventos a otros paquetes o clases.

En el caso del ejemplo inicial de esta hoja se tiene una transición entre los estados `IntroduciendoMoneda` y `SeleccionadoAzucaryProducto` que tiene una transición con el siguiente detalle:

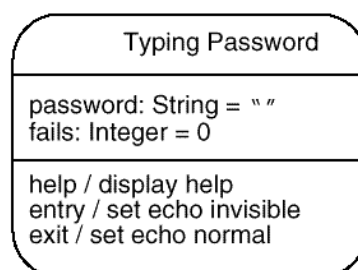
```
userInput( Button ) | [TodoOk=true} / MostrarNivelAzucar, MostrarProducto
```

El evento que dispara el cambio de estado es `userInput(Button)`. Se requiere como condición adicional que no se haya detectado ninguna falla (`TodoOk = true`) y se ejecuta `MostrarNivelAzucar` y `MostrarProducto`, que deberían ser ejecutables por el objeto al cual pertenece el diagrama.

Transición interna. Es una transición que permanece en el mismo estado, en vez de involucrar dos estados distintos. Representa un evento que no causa cambio de estado. Se denota como una cadena adicional en el compartimiento de acciones del estado.

Supongamos el estado de una interfaz pidiendo password al usuario. En este caso puede tenerse una transición interna que muestre una ayuda al usuario. Esta transición se muestra en el siguiente diagrama con la cadena "help / display help" dentro del cuerpo del estado.

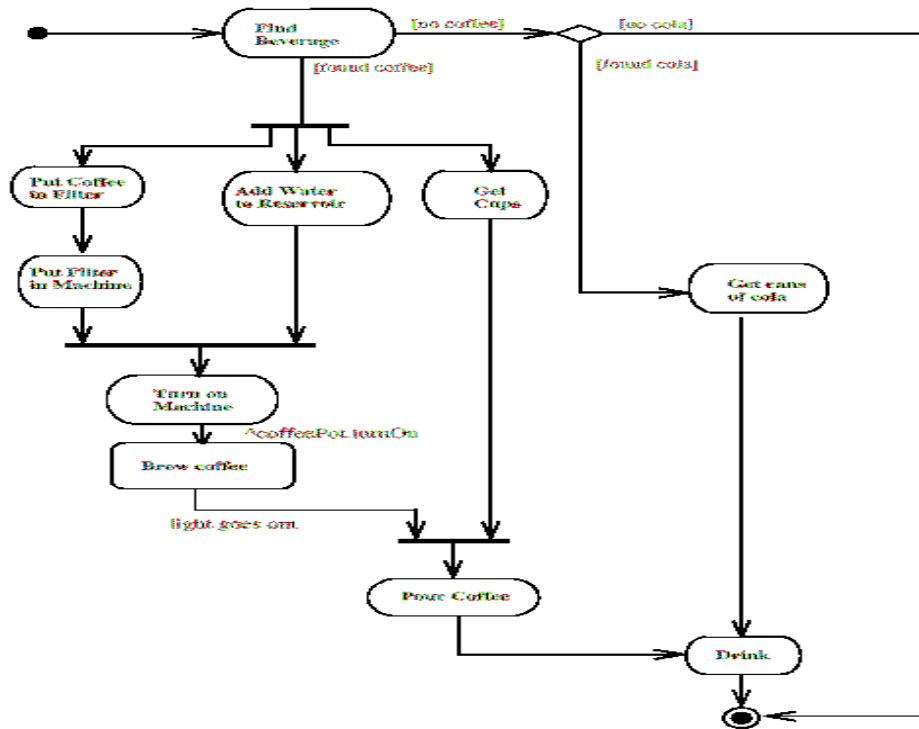
Figura 16. Representación grafica de una transición interna



Conceptos de un diagrama de actividades. Un diagrama de actividades es un caso especial de un diagrama de estados en el cual casi todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Sirven para representar transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos. Se presenta a continuación un ejemplo de diagrama de

actividades para un mensaje de un objeto. Generalmente modelan los pasos de un algoritmo.

Figura 17. Representación gráfica de un diagrama de actividades



Estado de acción. Representa un estado con acción interna, con por lo menos una transición que identifica la culminación de la acción (por medio de un evento implícito). No deben tener transiciones internas ni transiciones basadas en eventos (Si este es el caso, represéntelo en un diagrama de estados). Permite modelar un paso dentro del algoritmo.

Se representan por un rectángulo con bordes redondeados.

Transiciones. Las flechas entre estados representan transiciones con evento implícito. Pueden tener una condición en el caso de decisiones.

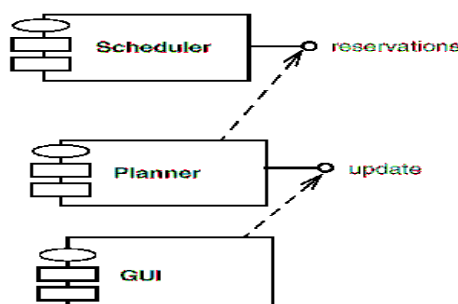
Decisiones. Se representa mediante una transición múltiple que sale de un estado, donde cada camino tiene un label distinto. Se representa mediante un diamante al cual llega la transición del estado inicial y del cual salen las múltiples transiciones de los estados finales. Un ejemplo se ve en la figura cuando no hay cafe y se toma una decisión entre hay cola o no hay cola.

Conceptos en un diagrama de implementación. Un diagrama de implementación muestra la estructura del código (Diagrama de componentes) y la estructura del sistema en ejecución (Diagrama de ejecución).

Diagrama de Componentes. Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. Los componentes software tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución. Se consideran en este tipo de diagramas solo tipos de componentes. Instancias específicas se encuentran en el diagrama de ejecución.

Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también contención de entre componentes software e interfaces soportadas. Un ejemplo es el siguiente:

Figura 18. Representación gráfica del diagrama de componentes



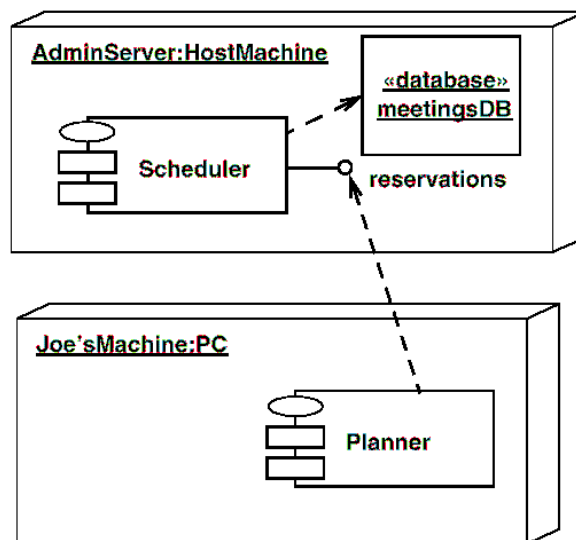
En este caso tenemos tres componentes, GUI dependiendo de la interfaz update provista por Planner, Planner dependiendo de la interfaz reservations provista por Scheduler.

Diagrama de Ejecución. Un diagrama de ejecución muestra la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software, procesos y objetos que se ejecutan en ellos. Instancias de los componentes software representan manifestaciones en tiempo de ejecución del código. Componentes que solo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes.

Un diagrama de ejecución es un grafo de nodos conectados por asociaciones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (un caso particular de un objeto). Las instancias de componentes software pueden estar unidos por relaciones de dependencia, posiblemente a interfaces.

Un ejemplo de diagrama de ejecución es el siguiente:

Figura 19. Representación gráfica del diagrama de ejecución.



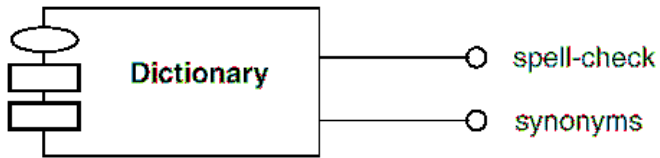
En este caso se tienen dos nodos, AdminServer y Joe'sMachine. AdminServer contiene la instancia del componente Scheduler y un objeto activo (proceso) denominado meetingsDB. En Joe'sMachine se encuentra la instancia del componente software Planner, que depende de la interfaz reservations, definida por Scheduler.

Nodos. Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Pueden representarse instancias o tipos de nodos. Se representa como un cubo 3D en los diagramas de implementación.

Componentes. Un componente representa una unidad de código (fuente, binario o ejecutable) que permite mostrar las dependencias en tiempo de compilación y ejecución. Las instancias de componentes de software muestran unidades de software en tiempo de ejecución y generalmente ayudan a identificar sus dependencias y su localización en nodos. Pueden mostrar también que interfaces implementan y qué objetos contienen. Su representación es un rectángulo atravesado por una elipse y dos rectángulos más pequeños.

Un ejemplo de componente que implementa dos interfaces es

Figura 20. Representación gráfica de un componente.



4.1.3. Herramientas de desarrollo de visual basic.

Basándonos en la página oficial de Microsoft "Desarrolle aplicaciones utilizando el lenguaje de programación más fácil de leer y de escribir que existe. La compilación en segundo plano proporciona información al instante y señala los errores con un subrayado ondulante. La implementación lado a lado acaba con los conflictos entre versiones y la herencia permite reutilizar el código de cualquier lenguaje basado en Windows". Tomamos la Herramientas de desarrollo más adecuada: Visual Basic.

El Control de excepciones estructurado proporciona un código de control de errores más elegante y fácil de mantener. Incorpore recursos, componentes y código de más de 3 millones de programadores de Visual Basic de todo el mundo. Utilice componentes del gran mercado de proveedores de controles para crear completas aplicaciones basadas en Windows.

Con Windows Forms, los programadores pueden crear aplicaciones para Windows que aprovechan las completas características de la interfaz de usuario del sistema operativo Windows. Esta versión incluye todas las herramientas de programación rápida de aplicaciones que los programadores esperan de Microsoft, como la creación, con arrastrar y colocar, de aplicaciones para Windows que aprovechan totalmente las bases de datos y los servicios Web XML. Con la herencia visual, los programadores pueden simplificar enormemente la creación de aplicaciones basadas en Windows, centralizando en formularios primarios la lógica común y la interfaz de usuario para toda la solución. Utilizando delimitadores y acoplamiento de controles, los programadores pueden generar formularios redimensionables automáticamente, mientras el editor de menús in situ permite crear menús de manera visual directamente desde el Diseñador de Windows Forms.

Con el Diseñador de Web Forms y el Diseñador XML, los programadores pueden utilizar las características de Microsoft IntelliSense® y la capacidad para completar etiquetas; o bien, pueden elegir el editor WYSIWYG (lo que ve es lo que se imprime) para crear aplicaciones Web interactivas arrastrando y colocando

elementos. Con sólo algunos pasos sencillos, los programadores pueden diseñar, programar, depurar y distribuir eficaces servicios Web XML que reduzcan el tiempo de programación al tener acceso y encapsular los procesos empresariales desde cualquier plataforma.

Visual Basic proporciona el centro de control más avanzado para los programadores. La página de inicio ofrece a los programadores un portal donde obtener información con un solo clic sobre proyectos utilizados recientemente, preferencias, actualizaciones y la comunidad MSDN Online. Las características de Microsoft IntelliSense, el cuadro de herramientas y la lista de tareas proporcionan mejoras de productividad importantes, mientras que las ventanas de ocultación automática y la compatibilidad con varios monitores ayudan a los programadores a maximizar el estado real de la pantalla y personalizar su entorno de desarrollo.

Con los asistentes para aplicaciones, las plantillas de proyectos y el código fuente de ejemplo, los programadores pueden crear rápidamente aplicaciones basadas en Windows, Internet y dispositivos con una inversión inicial mínima. La Ayuda dinámica y Microsoft Developer Network (MSDN®) proporcionan ayuda para la tarea que se ejecuta en ese momento, para que los programadores nunca carezcan de la información necesaria acerca de la plataforma Microsoft. Las macros de Visual Studio, al igual que las de Visual Basic para Aplicaciones (VBA) en Office, permiten la automatización de las tareas rutinarias en el IDE, aumentando aún más la productividad general de los programadores de Visual Basic.

El lenguaje de Visual Basic se ha actualizado, simplificado y modernizado. Con Visual Basic, ahora tiene acceso a un conjunto de herramientas mucho más completo y eficaz que en versiones anteriores de Visual Basic. Para satisfacer la fuerte demanda de los clientes, Visual Basic ofrece un amplio conjunto de nuevas características, como capacidades de diseño completamente orientado a objetos, subprocesamiento libre y acceso directo a Microsoft Framework. Asimismo, el lenguaje de Visual Basic se ha optimizado, eliminando palabras clave obsoletas que se habían heredado, mejorando la seguridad de tipos y revelando las construcciones de bajo nivel que los programadores avanzados necesitan.

Ahora, Visual Basic se integra plenamente con otros lenguajes de Microsoft Visual Studio®. No sólo se pueden programar componentes de aplicaciones en diferentes lenguajes de programación, sino que ahora, las clases se pueden heredar de clases escritas en otros lenguajes utilizando la herencia entre lenguajes. Con el depurador unificado, ahora se pueden depurar aplicaciones en varios lenguajes, independientemente de si se ejecutan localmente o en equipos remotos. Por último, independientemente del lenguaje que se utilice, Framework proporciona un amplio conjunto de interfaces de programación de aplicaciones (API) para el sistema operativo Microsoft Windows e Internet.

Los programadores de Visual Basic pueden unirse a la mayor comunidad de programadores, formada por más de 3 millones de programadores de Visual Basic de todo el mundo. Los programadores pueden usar componentes del inmenso mercado de controles de otros fabricantes para crear aplicaciones completas basadas en y pueden aprender Visual Basic rápidamente de los muchos libros y ejemplos disponibles.

4.2 ANÁLISIS PREVIO A LOS SISTEMAS DE PRODUCCIÓN MRP I Y MRP II

Conocido ya que productos o servicios se van a elaborar y mediante que procesos, hemos decidido la capacidad a largo plazo, los equipos necesarios y la localización de la actividad productiva y distribución en planta, del equipo y del factor humano. Hecho esto, el marco de referencia que nos indica donde queremos llegar, como y con que medios, se determina una ESTRATEGIA DE OPERACIONES. A partir de aquí, se hace necesario para medio y corto plazo:

- Concretar objetivos
- Decidir, (planificación) respecto a que productos o servicios a elaborar
- Determinar que artículo o ítem hay que producir y en que momento (programación)
- Ver que actividades deberán desarrollarse en las distintas unidades productivas, y en que momento, con el objeto de cumplir los requerimientos de componentes (programación a muy corto plazo)
- Tener en cuenta los recursos disponibles, es decir considerar la problemática de la capacidad, de forma que se elaboren planes y programas factibles.
- Considerar las necesidades de materiales, tanto de productos terminados como de materia prima y componentes intermedios para la fabricación de acuerdo con la planificación y programación elaborada (planificación, gestión y control inventario).

4.2.1 Planificación.

Es proyectar el futuro deseado, medios necesarios y actividades a desarrollar para conseguirlo.

Hecho hasta aquí, podemos pasar a la fase de ejecución y hacer los controles necesarios, que permitirán detectar y corregir las posibles desviaciones entre resultados obtenidos y los distintos objetivos marcados.

4.2.2 Planificación empresarial y de operaciones.

La planificación estratégica, en la que se establecen los objetivos, las estrategias y los planes globales a largo plazo, normalmente son entre 3 y 5 años. Esta actividad

es desarrollada por la alta Dirección, que se ocupa de problemas de gran amplitud, tanto en términos de actividad organizativa como de tiempo.

La planificación operativa, donde se concretan los planes estratégicos y objetivos a un elevado grado de detalles. Así se establecen las tareas a desarrollar para que se cumplan los objetivos y planes a largo plazo. En esa etapa las actividades son un poco más limitadas y van de 18 meses o un año varias semanas.

La planificación adaptativa, pretende eliminar las posibles divergencias entre los resultados y los objetivos relacionados con ellos.

4.2.3 ¿Qué es un sistema de producción?

Un sistema de producción es un conjunto de actividades que un grupo humano (por ejemplo, la familia campesina) organiza, dirige y realiza, de acuerdo a sus objetivos, cultura y recursos, utilizando prácticas en respuesta al medio ambiente físico.

De esta definición se desprenden algunas conclusiones o consecuencias:

Para conocer un sistema de producción, se debería partir de la observación de sus COMPONENTES: las actividades que allí se realizan, los medios y recursos con que cuenta, las cantidades y características de las personas que en él viven o trabajan, las propiedades del suelo o clima, etc.

Como en el sistema hay ORGANIZACION y hay RELACIONES, se debería además tratar de entender las propiedades o proporciones en que estos componentes están presentes; el rol o función que cada uno cumple y las interacciones que suceden entre los componentes. Por ejemplo, cómo se distribuye la mano de obra entre los diferentes rubros y actividades del predio; cómo se distribuyen los ingresos entre consumo, producción y ahorro; cómo la producción de un rubro contribuye a la generación de productos para el autoconsumo y para la venta, etc.

Finalmente, se necesitará comprender la DINAMICA del sistema de producción, es decir, su comportamiento a través del tiempo. Por ejemplo, cómo se distribuye la mano de obra a través del año; cuáles son los meses de mayor actividad y cuáles los de mayor escasez.

El Sistema de producción también se ha definido la administración de operaciones como la administración de los sistemas productivos o sistemas de transformación, que son los que convierten los insumos en bienes o servicios. Los insumos para el sistema son: Energía, materiales, mano de obra, capital e información. Estos se convierten en bienes o servicios mediante la tecnología del proceso. Las

operaciones de cada tipo de industria varían dependiendo del ramo, al igual que sus insumos.

Un sistema de producción proporciona una estructura que facilita la descripción y la ejecución de un proceso de búsqueda. Un sistema de producción consiste de:

- Un conjunto de facilidades para la definición de reglas.

- Mecanismos para acceder a una o más bases de conocimientos y datos.

- Una estrategia de control que especifica el orden en el que las reglas son procesadas, y la forma de resolver los conflictos que pueden aparecer cuando varias reglas coinciden simultáneamente.

- Un mecanismo que se encarga de ir aplicando las reglas.

- Configuraciones productivas.

- Conceptos y tipologías fundamentales.

Los sistemas de producción son sistemas que están estructurados a través de un conjunto de actividades y procesos relacionados, necesarios para obtener bienes y servicios de alto valor añadido para el cliente, con el empleo de los medios adecuados y la utilización de los métodos más eficientes.

En las empresas, ya sean de servicio o de manufactura, estos sistemas representan las configuraciones productivas adoptadas en torno al proceso de conversión y/o transformación de unos inputs (materiales, humanos, financieros, informativos, energéticos, etc.) en unos outputs (bienes y servicios) para satisfacer unas necesidades, requerimientos y expectativas de los clientes, de la forma más racional y a la vez, más competitiva posible.

Si se estudia el contexto empresarial, podrá encontrarse que existen distintos sistemas de producción en las empresas manufactureras y de servicio, respondiendo como es lógico, a características propias de sus procesos y funcionamiento. Así mismo, si se revisa apropiadamente la literatura sobre Administración de la Producción y las Operaciones, se encontrará con cierta diversidad de tipologías respecto a la forma de clasificar las configuraciones productivas. Esto se debe, fundamentalmente, a la variedad de enfoque con que los autores tratan estos temas en sus trabajos, que lejos de clarificar añaden mayor complejidad a dicha problemática. La gran diversidad de procesos existentes y los potenciales criterios de clasificación a considerar hacen que sea

difícil encontrar una clasificación exhaustiva que de manera unívoca contemple cada caso concreto.

Woodward (1965), fue probablemente el primer autor en tipificar los sistemas productivos. Descubrió que las tecnologías de fabricación se podían encuadrar en tres grandes categorías: producción artesanal o por unidad (producción discreta no-repetitiva), producción mecanizada o masiva (producción discreta repetitiva), y la producción de proceso continuo. Cada categoría incluye un método distinto de obtener los productos, siendo las principales diferencias, el grado de estandarización y automatización, tipo de proceso y la repetitividad de la producción. La tipología de Woodward distingue entre fabricación unitaria, de pequeños lotes, de grandes lotes, la producción en serie y aquellos procesos de transformación de flujo continuo. La propuesta de Woodward ha marcado pautas en la comunidad de autores. Gousty y Kieffer (1988), sobre la base de otros criterios, como complejidad e incertidumbre, proponen una nueva tipología para los sistemas industriales, delimitando los principales componentes que configuran la problemática de los sistemas de producción.

Hopeman (1991), Companys (1986), Díaz (1993) y Schroeder (1992), entre otros, optan por diferenciar los sistemas de producción en dos grandes grupos básicos: sistemas continuos e intermitentes. Otros, como Chase, Aquilano y Jacob (2000), Ochoa y Arana (1996) y Heizer y Render (1997), prefieren clasificarlos en: repetitivos y no-repetitivos. Los primeros, se refieren a la continuidad en sí del proceso de producción, y los segundos, a la repetitividad o recurrencia del producto y su proceso. Monks (1992), propone otra clasificación de sistemas de producción, identificando el sistema continuo (operaciones de flujo), sistema intermitente (operaciones de flujo y por lotes), sistema de trabajo interno (por lotes o trabajos únicos) y proyecto (trabajos únicos). Además, este autor añade que los sistemas productivos son frecuentemente clasificados según destino de la producción, ya sean fabricantes de bienes almacenables (tales como equipos) o fabricantes de bienes por pedido. Otra clasificación muy común, se basa en el sector de actividad, presentándose dos tipos extremos: sistemas de manufactura, encargados de la fabricación y/o montaje de bienes materiales, y sistemas de prestación de servicios.

Por su parte Womack, Jones y Roos (1991), y Doll y Vonderembse (1992), proponen otra clasificación de los sistemas productivos: producción «craft» (craft system), producción en masa (industrial system), y producción con mínimo desperdicio (post-industrial system). Esta clasificación es adecuada para algunos propósitos, tal como explicar las diferencias entre los nuevos sistemas de producción (sistemas de mínimo desperdicio) y los tradicionales (Miltenburg, 1995). También resulta útil para reflejar la evolución y los cambios de paradigmas

ocurridos en fabricación desde el modelo inicial de producción artesanal (craft model) hasta el modelo más actual denominado producción ajustada (lean manufacturing).

Por su parte Gorostegui (1991), ofrece una clasificación que difiere de las anteriores, clasificándolos según varias características propias, tales como: el destino del producto (por encargo /para el mercado), la razón de producir (por órdenes /almacén), la tipificación del producto (producción estándar /producción en serie) y la dimensión temporal del producto (intermitente /continua). En esta misma línea, Acevedo (1987), propone una clasificación sobre la base de una matriz morfológica que contempla la clasificación del sistema de producción de acuerdo a tres características fundamentales: relación producción-consumo, que considera la respuesta que debe dar el sistema hacia el entorno, ya sea por entrega directa o contra almacén; forma en que se ejecuta la producción; y elemento a optimizar. Al igual que Gorostegui, se combinan características que se refieren a dimensiones externas e internas.

El problema fundamental de estas formas de clasificar el sistema de producción, radica en que aunque son útiles desde el punto de vista de contextualización y caracterización de las unidades de producción, no resultan muy útiles para la realización de análisis competitivo y estratégico en fabricación, ya que, entre otras cosas, al ser demasiado amplias y genéricas, no logran identificar una cantidad finita y discreta de opciones efectivas de sistemas de producción que reflejen las distintas formas existentes de producir los bienes y/o servicios. Además, no tratan en su proceder la interrelación estratégica del binomio «producto-proceso», omitiendo así, las implicaciones potenciales que representa para la empresa la elección de uno u otro sistema de producción, expresadas en términos de las diferentes dimensiones técnicas y empresariales que componen un sistema de producción.

Las clasificaciones muy amplias, no facilitan la formulación de decisiones y acciones precisas y la realización de trade-offs entre ellas y mucho menos, permiten especificar los detalles de la estrategia de fabricación. En tal sentido, la estrategia de fabricación necesita una forma de clasificación del sistema de producción distinta, mucho más desagregada, discreta, detallada, que facilite entrar en detalles en la composición interna del sistema de producción, que permita comparar sus desempeños específicos y para este fin, la clasificación fundamentada en la tipología existente de los procesos de producción, además de ser la más usualmente empleada por investigadores y practicantes, resulta la más apropiada para los propósitos de análisis competitivo y de la estrategia de fabricación.

Quizás la clasificación basada en la tipología existente de procesos de producción más difundida sea la propuesta por Hayes y Wheelwright (1984; pp. 176-179), la cual resulta similar en muchos aspectos a la establecida por Woodward (1965), pero con énfasis básico en las pautas que siguen los flujos de trabajo en la fábrica.

Ellos arribaron a cinco tipos de configuraciones productivas bien definidas: proyecto, taller de trabajo (job-shop), lotes o flujo en línea desacoplado, línea de ensamblaje (también denominada en serie, repetitiva o de producción en masa) y proceso continuo. Asimismo, destacaron que estas últimas cuatro varían entre dos extremos en lo que a desplazamiento de materiales se refiere, la configuración orientada hacia el producto y la configuración orientada hacia el proceso, cuya diferencia más evidente es la distribución en planta, por producto, para la primera y por procesos, para la segunda.

Buffa (1968), fue sin duda otro de los primeros autores en ofrecer una tipología de sistemas de producción más acorde a lo antes referido. Partiendo de las dicotomías existentes en relación al layout físico de los sistemas productivos, o sea layout por producto vs. proceso, lineal vs. funcional ó continuo vs. intermitente, Buffa destaca que la mayoría de los sistemas productivos son realmente combinaciones de estos estados extremos y en tal sentido, ofrece una clasificación basada en cinco tipos de sistemas diferentes, correspondiendo los dos primeros a sistemas continuos y los restantes a sistemas intermitentes, ellos son: (1) sistemas de distribución para productos de inventario, (2) sistemas de producción-distribución para productos estandarizados de alto volumen, (3) taller de trabajo cerrado para productos de inventario, (4) taller de trabajo abierto para productos a medida y (5) proyectos de gran envergadura. Cada uno se distingue y diferencia por sus características propias y problemáticas específicas. Los dos primeros se refieren a productos planeados para inventarios, diferenciándose en el alcance de sus operaciones y el grado de control gerencial, los tres restantes se refieren a operaciones intermitentes mayormente dedicadas a obtener productos sobre diseño, a la medida, según requerimientos de clientes. A lo anterior, Buffa añade que dichos sistemas pueden no aparecer en sus formas "puras", sino que comúnmente aparecen como sistemas "mezclados". A esto Hill (1997) añade, que aunque puedan existir sistemas híbridos, orientados a reflejar mejor las necesidades de la fábrica, siempre se deberán clasificar por aquel que predomine, el "proceso base" o también denominado "proceso raíz".

En esta misma línea, Miltenburg (1995) subraya que son dos, entre otros, los factores principales que determinan la amplitud de las similitudes y diferencias entre los sistemas de producción existentes; ellos son: el tipo de producto que se fabrica y los outputs provistos al mercado. Cada empresa fabrica un tipo de producto diferente y provee diferentes outputs de fabricación a sus clientes. A esto Hill (1993, 1997), añade que la coincidencia entre las dimensiones de mercado y

producto con las características del proceso es un requisito esencial para evitar incompatibilidades de enfoque y ser competitivos en manufactura.

Concretando esta parte, un creciente número de autores, entre los que destacan Buffa (1984), Hayes y Wheelwright (1984), Miltenburg (1995), Hill (1993,1997), Cribillers (1997), Domínguez et al. (1998), Hax y Majluf (1999) y Cuatrecasas (1999), han preferido utilizar, de forma general, la clasificación de sistemas de producción fundamentada en la tipología de procesos productivos. El proceso es considerado el factor de mayor relevancia al identificar o caracterizar cualquier sistema de fabricación. Esta relevancia se fundamenta en el hecho de que cada proceso se caracteriza por tener un patrón de flujo material y layout que lo hacen diferente. Asimismo, existe una indisoluble interrelación entre producto y proceso, binomio esencial para análisis estratégico. Tal es así, que el producto y el proceso transitan por similares ciclos de vida compartidos, en los cuales el proceso adopta configuraciones específicas según sea la naturaleza del producto y la fase de su desarrollo en el mercado.

Cada sistema de producción, caracterizado esencialmente por su proceso productivo, conlleva un conjunto de implicaciones para la empresa, en cuanto al comportamiento apropiado de las diferentes dimensiones de fabricación y empresariales (Hill, 1997). Según este enfoque, y haciendo un análisis más detallado de los distintos trabajos y literatura consultada, se ha encontrado que los autores han aceptado por lo general, la existencia de ocho tipologías de sistemas o configuraciones productivas bien definidas: Proyecto, Job-Shop, Lotes (Batch), Línea acompañada por Equipo, Línea acompañada por Obrero, Configuración Continua, Just in Time y Sistema Flexible de Fabricación.

Configuración por Proyecto. Producción generalmente de productos únicos de cierta complejidad que requieren gran cantidad de inputs. Estos deben fabricarse en un lugar definido debido a que es difícil o casi imposible transportarlos una vez terminados. Como resultado, y a diferencia de cualquier otro proceso productivo, los recursos que comprende deben trasladarse al lugar de operación, ya que aquí no existe flujo del objeto de trabajo, sino que son los recursos técnicos y humanos quienes acuden al lugar de trabajo. Las actividades y recursos se gestionan como un todo. Su coordinación adquiere carácter crítico. Existe un connotado interés por el control de los costos y las fechas de terminación.

Configuración de Taller (Job-shop). El sistema de producción Job-Shop fabrica muchos productos diferentes en volúmenes que varían entre la unidad y pocas unidades de cada producto. Consiste en una fabricación no en serie, de lotes pequeños, para pedidos únicos o de pequeñas cantidades. Por lo regular implica productos adaptados, diseñados a la medida del cliente y de naturaleza muy poco

repetitiva. Se requieren operaciones poco especializadas, las cuales son realizadas por un mismo obrero o por un grupo pequeño de ellos, los cuales tienen la responsabilidad de terminar todo o casi todo el producto. Como se fabrican productos muy diferentes, los recursos son flexibles y versátiles. El flujo material es irregular, aleatorio y varía considerablemente de un pedido al siguiente. Se requiere que el fabricante interprete el diseño y las especificaciones del trabajo, así como que aplique capacidades del alto nivel en el proceso de conversión. En la producción Job-Shop lo que se trata es de obtener un "producto a medida" del cliente.

Configuración por Lotes. El sistema de flujo en lotes produce menos variedad de producto en volúmenes más elevados que el caso anterior. El mayor volumen se debe a un aumento de la repetitividad en ciertos artículos que se hacen dominantes. Estos productos se fabrican en lotes, que representan unos pocos meses de requerimientos de clientes. En este caso se requieren más operaciones, y éstas son más especializadas, por lo que difícilmente un mismo operario pueda dominarlas todas con una eficiencia aceptable. En tal sentido, el trabajo se divide en diferentes etapas tecnológicas, en las cuales los lotes sufren distintas operaciones. Así la instalación se suele dividir en secciones o talleres, en los cuales se agrupan los equipos con funciones similares. Se suele emplear una combinación de layouts celulares y funcionales. Los layouts celulares se utilizan cuando es efectivo en cuanto a costos disponer el equipo en células, para producir familias de productos. Como hay muchos productos, el equipo y utillaje son mayormente flexibles, de propósito general. El flujo material es desconectado aunque regular, variable de un pedido a otro, aunque existen pautas de flujo para familias de productos y para grandes lotes. Es el sistema más utilizado.

Configuración en Línea Acompasada por el Equipo (LAE). El equipo y procesos están organizados en una línea o líneas especializadas para producir un pequeño número de productos diferentes o familias de productos. Estos sistemas se usan sólo cuando el diseño del producto es estable y el volumen es lo suficientemente elevado para hacer un uso eficiente de una línea especializada con capacidades dedicadas. Se fabrica a una tasa constante, con un flujo automatizado e intensivo en capital. Los operarios realizan tareas relativamente simples a un ritmo determinado por la velocidad de la línea. El control del ciclo productivo está automatizado, existe alta estandarización y una elevada eficiencia en todo el proceso.

Configuración en Línea Acompasada por Operarios (LAO). Se utiliza cuando el número de productos diferentes es demasiado elevado y los volúmenes de producción demasiado variables para el sistema en línea con flujo acompasado por el equipo. En este sistema, la línea es más flexible que en el caso anterior, y puede funcionar con una variedad de velocidades. La tasa de producción depende del

producto particular que se fabrique, del número de operarios asignados a la línea y de la eficacia del trabajo en equipo de los operarios. Aunque los productos sean algo diferentes, son técnicamente homogéneos, usando la misma instalación, personal y la misma secuencia de estaciones de trabajo, aunque alguno de ellos pueda no pasar por alguna que no le es necesaria. El ciclo de productivo está controlado por los operarios a diferencia de la LAE donde dicho control está automatizado, esto hace que sea más flexible y versátil que el anterior.

Configuración de Flujo Continuo. Este sistema es similar al de línea en flujo acompasado por el equipo. Sin embargo, es más automatizado, más intensivo en capital y menos flexible. Cada máquina y equipo están diseñados para realizar siempre la misma operación y preparados para aceptar de forma automática el trabajo suministrado por la máquina precedente. Está diseñado para fabricar un producto o una familia limitada de productos en volúmenes muy elevados. El diseño del producto es muy estable, a menudo es un producto genérico o «commodity». El flujo material es continuo sincronizado, integrado a través de toda la instalación como si fuera un gran proceso tecnológico. Este rígido sistema, se basa en un proceso muy automatizado, costoso y especializado en la obtención de un producto estándar, donde la homogeneidad es total y absoluta, funcionando continuamente con mínima intervención del personal de línea. Generalmente precisa laborar las 24 horas para procurar ser un sistema costeable y eficiente.

Sistema de Producción JIT. Es importante distinguir entre el sistema de producción JIT y las técnicas JIT. Las técnicas denominadas JIT incluyen el control estadístico de la calidad, reducción de los tiempos de cambio de útiles (SMED), polivalencia de los trabajadores, versatilidad de los equipos, estandarización de operaciones, el enfoque de la producción mediante «arrastre» (Kanban), layout celular, mantenimiento autónomo, implicación de todo el personal en las decisiones gerenciales, resolución continua de problemas control automático de defectos, etc. Estas técnicas se usan en el sistema de producción JIT, pero también se usan en otros sistemas. El sistema de producción JIT es mucho más que un agregado de técnicas JIT. Surgido en Toyota Motor Co., es un sistema de flujo lineal (virtual o físico) que fabrica muchos productos en volúmenes bajos a medios. Por su diseño, el sistema JIT fuerza la eliminación de todos los innecesarios ("desperdicios"), y a partir de aquí, impone la mejora continua. Esto conduce naturalmente a costos inferiores, mejoras en la calidad y entregas más rápidas. El sistema JIT es el más difícil de diseñar, implantar y gestionar de todos, y pueden existir diferentes niveles de implantación del mismo.

Sistema Flexible de Fabricación (FMS). El sistema FMS consiste en un grupo de máquinas controladas por computadoras y sistemas automáticos de manejo, carga y descarga de material, todo ello controlado por un computador supervisor. Un

FMS puede funcionar sin atención de personal durante largos periodos. Las máquinas, el sistema de manipulación de materiales y las computadoras son muy flexibles, versátiles, lo que permite a un sistema FMS fabricar muchos productos diferentes en bajos volúmenes. Por ser sumamente costoso, se emplea comúnmente en situaciones en las que no pueden utilizarse sistemas de producción en línea de flujo más simples y baratos. Por lo general, se desarrolla en un entorno CIM (manufactura integrada por computador).

Las seis primeras modalidades de sistemas de producción se han denominado sistemas tradicionales ó clásicos y están fundamentados por los enfoques de gestión craft y producción en masa, que van desde la búsqueda de habilidades y capacidades individuales basadas en la funcionalidad del proceso y la pericia del operario, hasta la consecución de alta productividad y eficiencia a través de la optimización de las operaciones y economías de escala. Las dos últimas, Just in Time (JIT) y Sistemas Flexibles de Fabricación (FMS), han surgido producto de un nuevo enfoque de gestión de la producción denominado «lean production» o producción ajustada, surgido en los últimos años y que se basa en la producción con mínimo desperdicio, que busca la eliminación de aquellas actividades que no añaden valor, así como los consumos innecesarios de recursos, que se consideran como despilfarro. Este enfoque ha dado lugar a estos nuevos sistemas productivos, orientados a la obtención de pequeños a medianos volúmenes con alta variedad de productos, empleando para ello un layout de flujo lineal (en lugar de funcional), que resulta más efectivo y eficiente. Se trata de una combinación apropiada de las bondades de sus predecesores. Ambos sistemas, híbridos por naturaleza, están dotados de eficiencia y flexibilidad, y sus diferencias básicas radican en el grado de intensidad tecnológica utilizado en sus operaciones y procesos.

Otros autores como Hill (1997), describen modalidades adicionales de sistemas de producción híbridos que, aunque no sean tan completos como los antes descritos, sí contribuyen por igual a que las empresas ofrezcan un proceso de fabricación que refleje mejor sus necesidades en términos de poder respaldar las características de sus mercados. Entre estos sistemas híbridos destacan la Fabricación Celular (basada en la tecnología de grupo), las Líneas de Transferencia (o líneas transfer) y los Centros Maquinadores.

La aparición de las configuraciones híbridas, resultantes de combinar aspectos de los sistemas básicos o clásicos, ha sido un proceso evolutivo natural en la gestión de la producción en una economía competitiva. Por lo general y mucho más en los tiempos actuales de alta rivalidad competitiva, las empresas tienden a presentar una combinación de procesos y configuraciones en fabricación a fin de tratar de reflejar y cubrir mejor las diversas necesidades y requerimientos de los productos que proveen y venden. Claro está, se debe prestar mucha atención a estas

combinaciones de características para evitar incompatibilidades y disfunciones operativas, y debido también, al hecho de que la elección que puedan hacer estará siempre limitada por la dimensión de ingeniería (el proceso deberá poder cumplir con las especificaciones del producto) y por las propias limitaciones técnicas, tecnológicas y empresariales que restringen las posibles opciones.

Estos sistemas de producción, clásicos y modernos, se diferencian entre sí por el comportamiento descrito en las diversas dimensiones técnicas y empresariales, propias del diseño del sistema así como de su funcionamiento, tales como, y por citar algunos ejemplos: la repetitividad de las operaciones y trabajos, el nivel de continuidad o intermitencia en el flujo material, el tipo de producción predominante, el mix de producto con que se opera (volumen-variedad), la estructura espacial utilizada, la estructura temporal de la producción, la propia naturaleza del producto que se fabrica y comercializa (estándar, especial ó adaptado), el nivel de especialización de las capacidades, nivel de estandarización de productos, el grado de automatización incorporado, así como las dimensiones de competencia /mercado que se proveen al cliente final, entre otras.

En este trabajo no se ha pretendido abordar, ni mucho menos, todos los tipos de clasificación existentes de sistemas de producción, sino más bien ofrecer una panorámica terminológica sobre el tema y un marco conceptual que constituya una reflexión y punto de partida para futuras investigaciones en este campo.

4.2.3 Clasificación de sistemas productivos.

- Sistema de producción por encargo. Este se basa en el encargo o pedido de uno o más productos o servicios. La empresa que lo utiliza sólo produce después de haber recibido el contrato o encargo de un determinado producto o servicio, aquí se llevan a cabo tres actividades:
 - Plan de producción: Relación de materia prima, mano de obra y proceso de producción.
 - Arreglo físico: Se concentra en el producto.
 - Previsibilidad de la producción: Cada producto exige un plan de producción específico.
 - Sistema de producción por lotes: Lo utilizan las empresas que producen una cantidad limitada de un tipo de producto o servicio por vez. También se llevan a cabo las tres actividades que el sistema anterior:
 - Plan de producción: Se realiza anticipadamente en relación a las ventas.
 - Arreglo físico: se caracterizan por máquinas agrupadas en baterías del mismo tipo.

- Previsibilidad de la producción: Debe ser constantemente replaneado y actualizado.
- Sistema de producción continua: Lo utilizan las empresas que producen un determinado producto sin modificaciones por un largo período, el ritmo de producción es rápido y las operaciones se ejecutan sin interrupciones. Dentro de este sistema se realizan los tres pasos:
 - Plan de producción: Se elabora generalmente para períodos de un año, con subdivisiones mensuales. Este sistema lo utilizan fabricantes de papel, celulosa, de automóviles, electrodomésticos.
 - Arreglo físico: Se caracteriza por máquinas y herramientas altamente especializadas, dispuestas en formación lineal y secuencial.
 - Previsibilidad de la producción: El éxito de este sistema depende totalmente del plan detallado de producción, el que debe realizarse antes que se inicie la producción de un nuevo producto.

4.2.4 Función de producción de una empresa.

Puede definirse como el proceso de transformación de los factores que ella toma de su entorno, en productos que generan valor agregado. Todo proceso de producción puede subdividirse en tres fases:

- **Insumos:** implica la adquisición, recepción y almacenamiento de materias primas. Pueden ser materiales o personas.
- **Procesos:** Conjunto de operaciones a través de las cuales los factores se transforman en productos. Incluye planta, maquinaria y trabajo. Es decir, la tecnología de los activos productivos de materiales indirectos y el conocimiento.
- **Productos:** Bienes físicos y/o servicios entregados del productor al consumidor.

4.2.5 Elementos de la producción

Los elementos de producción están conformados por:

- **Capital:** En sistema de producción el capital es el que designa un conjunto de bienes y una cantidad de dinero de los que se puede obtener, en el futuro, una serie de ingresos. En general, los bienes de consumo y el dinero empleado en satisfacer las necesidades actuales se representan en los sistemas de producción. Por lo tanto, una empresa considerará como capital

la tierra, los edificios, la maquinaria, los productos almacenados, las materias primas que se posean, así como las acciones, bonos y los saldos de las cuentas en los bancos. No se consideran como capital, en el sentido tradicional, las casas, el mobiliario o los bienes que se consumen para el disfrute personal, ni tampoco el dinero que se reserva para estos fines.

Se pueden distinguir varias clases de capital. Una clasificación muy común distingue entre capital fijo y capital circulante. El capital fijo incluye medios de producción más o menos duraderos, como la tierra, los edificios y la maquinaria. El capital circulante se refiere a bienes no renovables, como las materias primas o la energía, así como los fondos necesarios para pagar los salarios y otros pasivos que se le puedan exigir a la empresa.

La cuantía de los rendimientos depende de las ganancias de productividad resultantes del aumento de capital utilizado en el proceso productivo. Estos aumentos de capital alargan el proceso de producción (en lugar de producir bienes de consumo hay que esperar a terminar de producir los bienes de capital: maquinaria, etcétera), por lo que aumenta el tiempo necesario para obtener rendimientos.

Por lo tanto, se considera que la cantidad de dinero que se ahorra, y en consecuencia la cantidad de capital creado, dependían del equilibrio entre el deseo de una satisfacción inmediata, derivada del consumo presente, y el deseo de obtener ganancias en el futuro, derivadas de un proceso de producción más largo.

- **Mano de obra:** En los sistemas de producción es necesaria la mano de obra como un elemento fundamental de la producción, igualmente es el esfuerzo humano realizado para asegurar un beneficio económico en la organización.

En la industria, la mano de obra tiene una gran variedad de funciones, que se pueden clasificar de la siguiente manera: producción de materias primas, como en la minería y en la agricultura; producción en el sentido amplio del término, o transformación de materias primas en objetos útiles para satisfacer las necesidades humanas; distribución, o transporte de los objetos útiles de un lugar a otro, en función de las necesidades humanas; las operaciones relacionadas con la gestión de la producción.

- **Materiales:** En el sistema de producción utiliza materiales para desarrollar su función esencial, la de transformación de insumos para obtener bienes o servicios; Cada operación del proceso requiere materiales y suministros a tiempo en un punto en particular, el eficaz manejo de materiales. Se

asegura que los materiales serán entregados en el momento y lugar adecuado, así como, la cantidad correcta. El manejo de materiales debe considerar un espacio para el almacenamiento.

El manejo de materiales puede llegar a ser el problema de la producción ya que agrega poco valor al producto, consume una parte del presupuesto de manufactura. Este manejo de materiales incluye consideraciones de movimiento, lugar, tiempo, espacio y cantidad. El manejo de materiales debe asegurar que las partes, materias primas, material en proceso, productos terminados y suministros se desplacen periódicamente de un lugar a otro.

En una época de alta eficiencia en los procesos industriales las tecnologías para el manejo de materiales se han convertido en una nueva prioridad en lo que respecta al equipo y sistema de manejo de materiales. Pueden utilizarse para incrementar la productividad y lograr una ventaja competitiva en el mercado. Aspecto importante de la planificación, control y logística por cuanto abarca el manejo físico, el transporte, el almacenaje y localización de los materiales.

Riesgos de un manejo ineficiente de materiales:

Sobrestadía: La sobrestadía es una cantidad de pago exigido por una demora, esta sobrestadía es aplicada a las compañías si no cargan o descargan sus productos dentro de un periodo de tiempo determinado.

Desperdicio de tiempo de máquina: Una máquina gana dinero cuando está produciendo, no cuando está ociosa, si una maquina se mantiene ociosa debido a la falta de productos y suministros, habrá y ineficiencia es decir no se cumple el objetivo en un tiempo predeterminado. Cuando trabajen los empleados producirán dinero y si cumplen el objetivo fijado en el tiempo predeterminado dejaran de ser ineficientes.

Lento movimiento de los materiales por la planta: Si los materiales que se encuentran en la empresa se mueven con lentitud, o si se encuentran provisionalmente almacenados durante mucho tiempo, pueden acumularse inventarios excesivos y esto nos lleva a un lento movimiento de materiales por la planta.

Todos han perdido algo en un momento o en otro: Muchas veces en los sistemas de producción por lote de trabajo, pueden encontrarse mal colocados partes, productos e incluso las materias primas. Si esto ocurre, la producción se va

a inmovilizar e incluso los productos que se han terminado no pueden encontrarse cuando así el cliente llegue a recógenos.

Un mal sistema de manejo de materiales puede ser la causa de serios daños a partes y productos: Muchos de los materiales necesitan almacenarse en condiciones específicas (papel en un lugar cálido, leche y helados en lugares frescos y húmedos). El sistema debería proporcionar buenas condiciones, si ellas no fueran así y se da un mal manejo de materiales y no hay un cumplimiento de estas normas, el resultado que se dará será en grandes pérdidas, así como también pueden resultar daños por un manejo descuidado.

Un mal manejo de materiales puede dislocar seriamente los programas de producción: En los sistemas de producción en masa, si en una parte de la línea de montaje le faltaran materiales, se detiene toda la línea de producción del mal manejo de los materiales que nos lleva a entorpecer la producción de la línea así como que el objetivo fijado no se llegue a cumplir por el manejo incorrecto de los materiales.

Desde el punto de vista de la mercadotecnia, un mal manejo de materiales puede significar clientes inconformes: La mercadotecnia lo forma un conjunto de conocimientos donde está el aspecto de comercialización, proceso social y administrativo.

4.2.6 Enfoque jerárquico para el proceso de planificación y control de producción.

Las actividades productivas, la planificación y control deben seguir un enfoque jerárquico que permita la coordinación entre objetivos – planes – actividades, de los niveles estratégicos, tácticos y operativos. O sea, cada uno va a proseguir su perseguir su propia meta, pero siempre teniendo en cuenta los del nivel superior, de los cuales depende, y los de nivel inferior que restringen.

Los objetivos estratégicos de la empresa tiene en cuenta otros factores, las previsiones de demanda a largo plazo, marcaran el Plan de Ventas, donde se indicaran las cifras de demanda que la empresa debería alcanzar para cumplir las metas de la firma. Este plan, mas los objetivos citados van a establecer el Plan de Producción a Largo Plazo, surge a partir e las necesidades de recursos mas lo ingresos previstos por ventas. Este conjunto de planes conforma la base del Plan Estratégico o Plan de Empresas, que deberá tener en cuenta la situación del sector, la competitividad y previsiones sobre las condiciones económicas en general.

Luego caemos en los que se denomina Planificación Agregada, esta fase trata de establecer, todavía en unidades agregadas para periodos normalmente mensuales, los valores de las principales variables productivas, teniendo en cuenta la capacidad disponible e intentando que permita cumplirse el Plan a Largo Plazo al menor costo posible. Esta etapa finaliza con el establecimiento de dos planes agregados: el de producción y el de capacidad.

El Programa Maestro de Producción (PMP) se obtiene como el grado de detalles del Plan Agregado, que permite la coordinación de la Planificación estratégica y de la Operativa, dado que no es suficiente para llevar a cabo esta última, por lo que las distintas familias se descompondrán en productos concretos y los periodos pasaran de meses a semanas.

En la cuarta etapa se llevara a cabo la Programación Detallada de los Componentes que integran los distintos productos y la Planificación Detallada de la Capacidad requerida por los mismos.

Deberá conseguirse que se cumpla el PMF (fabricación), él cual si existen problemas irresolubles de disponibilidad respecto a la capacidad existente deberá ser reajustado. El resultado de este proceso, por lo que respecta a producción, es la obtención del denominado Plan de materiales.

Aquí entramos en la última fase, que implicara la Ejecución y Control del Plan de materiales. Tendremos por un lado, Programa de Operaciones en los centros de trabajo (CT) que tengan en cuenta las prioridades de fabricación, y por otro lado, las Acciones de Compras de materia prima, y componentes que se adquieren en el exterior. También será necesario realizar un control de la capacidad, de tipo detallado (control I/O), que proporcionara retroalimentación a este nivel y a los niveles superiores.

No debemos dudar que la aproximación jerárquica es condición necesaria, pero no suficiente para lograr la integración, la cual contempla, en sentido vertical, de lo global a lo concreto.

4.2.7 Introducción a la planificación y control de inventario.

Los inventarios o stocks, son considerados como una inversión, es cualquier recurso ocioso almacenado en espera de ser utilizado.

- Cuales son las razones por los que las empresas provocan mantenimiento de stocks?

- Por qué las empresas inmovilizan con frecuencia enormes cantidades de dinero en recursos ociosos?

Razones que justifican la existencia de inventarios:

- **Hacer frente a la demanda de productos finales:** Si la demanda de los clientes fuese conocida con certeza y la producción coincidiese exactamente en fecha y cantidad no sería necesario almacenar productos finales.
- **Evitar interrupciones en el proceso productivo:** Las empresas se protegen de eventuales paradas no deseadas, acumulando una cierta cantidad de inventarios.

Estas son:

- **Falta de suministro externo:** se pueden dar por retrasos en las entregas y o recepción de pedidos.
- **Falta de suministro interno:** por averías de equipos, mala calidad de componentes elaborados, etc.
- Cuando en un determinado momento existe la necesidad de un artículo concreto, y este no se encuentra disponible, se dice que se producido una ruptura de stock, esto puede darse tanto en productos finales como en suministros externos e internos. El inventario que se mantiene para hacer frente a dicha eventualidad se denomina stock de seguridad (SS).
- La naturaleza del proceso de producción: Dado que cualquier etapa del proceso productivo requiere un determinado tiempo para su realización, existirá en permanencia una cierta cantidad de productos en curso. Si todas las fases estuviesen perfectamente sincronizadas, es decir, todos los componentes que salen de una etapa entrasen en la siguiente sin esperas intermedias, el stock se reduciría al mínimo.
- Nivelar el flujo de producción: Cuando nos encontramos con una demanda variable, una posible solución es fabricar por encima de la demanda en épocas bajas y almacenar el exceso de producción para emplearlo en aquellos momentos en los que la demanda supera la capacidad de la firma.
- Obtener ventajas económicas.

- **Falta de acoplamiento entre producción:** Es la causa típica de las empresas agrícolas, en las que la producción se obtiene en un periodo determinado, y su consumo se realiza a lo largo del año.
- **Ahorro y especulación:** Cuando se prevé un alza en los precios, puede ser interesante adquirirlos antes de que este se produzca y almacenarlos hasta el momento de su consumo (ahorro) o venta (especulación), en un momento posterior a la subida.
- Cuestiones fundamentales para la planificación de materiales
- Cuando (en que momento) deben realizarse los distintos pedidos de material?
- Cuanto debe pedirse de cada material al emitir un pedido? O lo que es lo mismo.
- Cual debe ser el tamaño de los lotes a solicitar?

Las respuestas van a depender de los factores que a continuación se describen,

- Características de la demanda
- Planificación de inventarios de ciclo único o monoperiodico, que se trata de un producto cuya demanda se produce una sola vez, y por lo tanto los ítems necesarios para la elaboración se almacenan en un solo periodo.
- El método de Planificación Multiperiodica, es el método frecuente, cuando la demanda se mantiene a lo largo del tiempo, ya sea continua o discontinua, regular o irregular.
- Tipos de demanda. Demanda Independiente: será aleatoria en función de las condiciones de mercado, no esta relacionada directamente con la de otros artículos. Es el caso de los productos terminados adquiridos por los clientes o piezas de repuestos (calculo de la demanda por estimación).

Demanda Dependiente: dependen de otros artículos almacenados, es el caso de un automóvil, cuyo consumo dependerá del número de unidades a fabricar del producto final. (Calculo de la demanda es directo).

- Costes relacionados con los inventarios. El hecho de mantener un stock provoca gastos a la empresa pero en el momento de su falta provocan costes.
- Tiempo de suministro (TS). Es el intervalo de tiempo que transcurre entre el momento en que se solicita un pedido, y el instante de su llegada, entendida ésta, como el momento en que esta disponible para ser utilizado.

Este concepto se aplica tanto al suministro externo, como al interno. En caso de SE, la empresa determina el TS en base a la experiencia con el proveedor. Para el caso de SI, en el que el pedido solicitado, es fabricado por la propia empresa, los tiempos consumidos desde que se detecta la necesidad de aquel, hasta que esta disponible, se encuentran los siguientes componentes:

- Tiempo de confección del pedido: Es aquel necesario para elaborar la documentación y enviarla al CT, incluye datos como tamaño, ruta, fechas previstas, material necesario, etc.
- Tiempo de desplazamiento o transporte: Incluye traslado de materiales hasta el CT, dentro de él y el envío hasta el almacén de pedido.
- Tiempo de cola: tiempo de espera en el CT, hasta que otros pedidos de mayor importancia sean entregados.
- Tiempo de preparación: del CT para ejecutar el pedido.
- Tiempo de ejecución: del pedido.
- Tiempo de espera: Que transcurre desde que se han finalizado las operaciones hasta que el lote se traslada desde el CT hasta el almacén.
- Tiempo de inspección: consumido para realizar dicha actividad sobre el lote en cuestión.

La tendencia actual es reducir los TS, con el objeto de incrementar la rapidez de respuesta al mercado, disminuir los inventarios, etc. Dos vías fundamentales pueden ser: simplificar las listas de materiales y eliminar o reducir los componentes de TS que no generen valor añadido.

El objetivo del MRP I es dar un enfoque más objetivo, sensible y disciplinado a determinar los requerimientos de materiales de la empresa. Para ello el sistema trabaja con dos parámetros básicos: tiempos y capacidades.

4.3 SISTEMAS MRP I: EL MRP ORIGINARIO

La meta fundamental a alcanzar por la empresa, es disponer del stock necesario justo en el momento en que va a ser utilizado. El énfasis debe ponerse más en él cuando pedir que en el cuanto, lo cual hace que sea más necesario una técnica de programación de inventarios que de gestión de los mismos, el objetivo básico, pues, no es vigilar los niveles de stocks como se hace en la gestión clásica, sino asegurar su disponibilidad en la cantidad deseada, en el momento y lugar adecuados.

Este sistema da por órdenes las compras dentro de la empresa, resultantes del proceso de planificación de necesidades de materiales.

Ámbito: Mediante este sistema se garantiza la prevención y solución de errores en el aprovisionamiento de materias primas, el control de la producción y la gestión de stocks.

La utilización de los sistemas MRP conlleva una forma de planificar la producción caracterizada por la anticipación, tratándose de establecer qué se quiere hacer en el futuro y con qué materiales se cuenta, o en su caso, se necesitaran para poder realizar todas las tareas de producción.

Es un sistema que puede determinar de forma sistemática el tiempo de respuesta (aprovisionamiento y fabricación) de una empresa para cada producto.

El MRP (planificación de las necesidades de materiales) es más que una simple técnica de gestión de inventarios. Este tipo de sistema sigue un enfoque jerárquico y nacen como una técnica informatizada de gestión de stocks de fabricación y de programación de la producción, capaz de generar el Plan de materiales a partir de un PMP.

El sistema MRP calculará las cantidades de producto terminado a fabricar, los componentes necesarios y las materias primas a comprar para poder satisfacer la demanda del mercado, obteniendo los siguientes resultados:

- El plan de producción especificando las fechas y contenidos a fabricar.
- El plan de aprovisionamiento de las compras a realizar a los proveedores
- Informes de excepción, retrasos de las órdenes de fabricación, los cuales repercuten en el plan de producción y en los plazos de entrega de producción final.

4.3.1 Beneficios e implicaciones

Los beneficios más significativos son:

- Satisfacción del cliente
- Disminución del stock
- Reducción de las horas extras de trabajo
- Incremento de la productividad
- Menores costos, con lo cual, aumento en los beneficios
- Incremento de la rapidez de entrega
- Coordinación en la programación de producción e inventarios
- Rapidez de detección de dificultades en el cumplimiento de la programación
- Posibilidad de conocer rápidamente las consecuencias financieras de nuestra planificación

4.3.2 Esquema básico del mrp originario

MRP, es un sistema de planificación de componentes de fabricación que, mediante un conjunto de procedimientos lógicamente relacionados, traduce un

Programa Maestro de Producción, PMP, en necesidades reales de componentes, con fechas y cantidades. A su vez no permite conocer que actividad ha de desarrollar cada unidad productiva en cada momento de tiempo para fabricar los pedidos planificados en el orden establecido, ni tampoco si se cuenta o no con la capacidad suficiente de hacerlo.

En cuanto a las características del sistema, se podría resumir en:

- Está orientado a los productos, a partir de las necesidades de estos, planifica las de componentes necesarios.
- Es prospectivo, pues la planificación se basa en las necesidades futuras de los productores.

Realiza un decalaje de tiempo de las necesidades de ítems en función de los tiempos de suministro, estableciendo las fechas de emisión y entrega de los pedidos. Con respecto a este tema, hay que recordar que el sistema MRP toma al TS como un dato fijo, por lo que es importante que este sea reducido al mínimo antes de aceptarlo.

No tiene en cuenta las restricciones de capacidad, por lo que no asegura que el plan de pedidos sea viable.

Es una base de datos integrada que debe ser empleada por las diferentes áreas de la empresa.

Lista de materiales: Es una descripción clara y precisa de la estructura que caracteriza la obtención de un determinado producto, mostrando claramente:

- Los componentes que lo integran
- Las cantidades necesarias de cada una de ellos para formar una unidad del producto en cuestión
- La secuencia en que los distintos componentes se combinan para obtener el artículo final.

Aunque existen diversas formas de expresar la Lista de Materiales, la mas clara, es la de la estructura en forma de árbol, con diferentes niveles de fabricación y montaje. La codificación por niveles facilita la explosión de las necesidades a partir del elemento final, y su lógica es la siguiente:

- Nivel 0: los productos finales no usados, en general, como componentes de otros productos, es el nivel mas complejo de la lista.
- Nivel 1: los componentes unidos directamente a un elemento de nivel 0.
- ETC.

Respecto de los elementos de nivel 0, hay que decir que no siempre se tratara de productos finales propiamente dicho. En el caso de múltiples productos finales, que son en realidad opciones de un numero reducido de modelos, se colocaran en el nivel 0 los subconjuntos complejos representativos de cada uno de estos. Cuando se da este caso, las Listas de materiales se denominan modulares.

4.4. LOS SISTEMAS MRP II

Lo definiremos como una ampliación del MRP de bucle cerrado que, de forma integrada y mediante un proceso informatizado on-line, con una base de datos única para toda la empresa, participa en la planificación estratégica, programa la producción, planifica los pedidos de los diferentes ítem componentes, programa las prioridades y las actividades a desarrollar por los diferentes talleres, planifica y controla la capacidad disponibles y necesaria y gestiona los inventarios. Además, partiendo de los outputs obtenidos, realiza cálculos de costes y desarrolla estados financieros en unidades monetarias.

Sus características son similares al anterior y además:

- Participa en la planificación estratégica, en el cálculo de costes y en el desarrollo de estados financieros.

- Permite planificar, programar, gestionar y controlar todos los recursos de la empresa manufacturera.
- Debe ser capaz de convertir en unidades monetarias las cifras derivadas de la explotación en unidades físicas.

4.4.1 El mrp de bucle cerrado

Este sistema parte de un Plan Agregado de Producción elaborado del Sistema, el cual será convertido en un PMP por el módulo de Programación Maestra. Este último será el punto de partida para la planificación de la capacidad a medio plazo mediante una técnica aproximada. Si el plan resultante es viable, el Programa Maestro pasara a servir de input al modulo MRP. Los Planes de Pedidos a proveedores de MRP irán destinados a la gestión de compras, mientras que los pedidos a talle servirán para la planificación de Capacidad (CRP). Si el plan a corto plazo deducido de CRP es viable, los pedidos pasaran a formar parte un la gestión de Talleres, en la que el sistema controlara las prioridades y programar las operaciones (normalmente con Lista de Expedición). La situación en los talleres de capacidad a corto plazo servirá al sistema para controlar la capacidad, normalmente usando el Análisis Input/Output. El término bucle cerrado implica que no solo se incluye cada uno de esos elementos en el Sistema Global, sino que también hay retroalimentación para mantener planes validos en todo momento. De acuerdo con lo anterior las características del MRP de bucle cerrado podrían resumirse de la siguiente forma:

Características del mrp de bucle cerrado

- Es prospectivo. Ya que la planificación esta basada en el Plan Agregado de Producción
- Incluye la PMP. La Planificación de necesidades de materiales, la Planificación de capacidad a corto y medio plazo, Control de la Capacidad y la Gestión de talleres.
- Trata de forma integrada todos los aspectos que contempla. Dado que la base de datos y el sistema son únicos para todas las áreas de la empresa.
- Actúa en tiempo real. Usando terminales on-line, aunque algunos de los procesos se producirían en batch (ejemplo, la explosión de materiales)

- Tiene capacidad de simulación. De forma que permite determinar que ocurriría si se produjeran determinados cambios en las circunstancias de partida.
- Actúan de la cúspide hacia abajo. Pues el proceso ha de comenzar en el Plan Agregado de Producción.

Funciones del sistema

Las dos grandes funciones de este sistema se dividen en directas, que son aquellas que el MRPII desarrolla en los procesos y transacciones realizadas por el sistema; e indirectas, en estas encuadran aquellas otras que muestran el efecto de las funciones directas sobre otras áreas de la empresa.

Funciones directas

- Formalización informatizada del proceso de planificación empresarial. Desarrolla parte mecánica de este y proporciona a los decidores la información para la valoración y selección de las alternativas.
- Elaboración de planes a largo y medio plazo. A través del Plan Agregado, el MRPII concreta su participación en la planificación estratégica de la empresa, tanto en desarrollo del plan de empresas como la validación del plan de producción a largo plazo. Además desarrolla el Plan Agregado de producción a medio plazo.
- Calculo de costes. Permite determinar los costes estándar unitarios, tanto de operaciones y centros de trabajo como de los distintos items.
- Programación maestra de la producción. Desarrolla la conversión del Plan de Producción en PMP para la planificación operativa, permitiendo, además, determinar la viabilidad del mismo en términos de capacidad.
- Planificación y control de la capacidad a medio, corto y muy corto plazo. Permite establecer la validez de los Planes de Producción. Además, el análisis Input/Output proporciona información crucial para la adopción de medidas correctoras de los elementos determinantes del Sistema de planificación y Control de la Capacidad.
- Gestión de Inventarios. Permite el desarrollo de una gestión de Stocks, dado que mantiene los registros de inventarios y permite la determinación del tamaño óptimo de lote para los ítems que desee.
- Planificación de las necesidades de materiales. Está construido alrededor del modelo MRP, por lo que incluye las funciones de este.
- Programación de Proveedores. Ya que los programas pueden ser enviados por medios informáticos, este elimina gran parte del trabajo administrativo, dando grandes beneficios a la actividad compras.

- Presupuestación. Puede elaborar presupuestos de compras, de ventas y de inventarios proyectados, que son fundamentalmente en cualquier empresa industrial.
- Gestión de talleres. Aconseja los pedidos a emitir al taller en función de sus fechas de emisión y entrega planificadas, como así también el informe de producción.
- Simulador de la actividad empresarial. Permite simular desde los efectos del cambio de un componente en el coste final de un ítem, hasta los de un cambio en las ventas previstas o en un objetivo sobre la capacidad, materiales o inventarios futuros.

Funciones indirectas

- Apoyo a la fijación de objetivos, estrategias y políticas. La elaboración por MRPII de los planes a largo plazo permitirán a la Alta Dirección la comprobación de la validez de los objetivos, estrategias y políticas trazados.
- Información básica para la toma de decisiones. Por un lado las salidas MRPII que aportan información válida para la toma de decisiones. Y por otro lado la capacidad de simulación, que permitirá lograr importante información en situaciones de futuro probable o incierto.
- Información básica al Subsistema Comercial. La posibilidad de determinar las fechas de entregas de pedidos, de conocer las entregas a realizar en determinados periodos a los clientes en cartea, etc, serán de gran ayuda para la programación de la distribución física, la consecución de altos niveles de servicios, etc.
- Información básica a Contabilidad y Finanzas. Son consecuencia de las funciones que, en costes y presupuestación desarrolla el Sistema.

4.4.2 SALIDAS DEL SISTEMA MRPII

Para la planificación a medio y largo plazo. Diversos informes sobre el Plan de Empresa, las previsiones de ventas, el Plan de Ventas, etc.

Para la programación de Sobre costes. Costes unitarios y reales de un ítem o de un CT, costes estándar y reales y globales de un pedido o de un CT.

Proveedores y presupuesto de compras. Básicamente expresaran el comportamiento pasado de los proveedores, los programas de pedidos de estos últimos y los pedidos a proveedores por ítems.

Sobre el presupuesto de ventas y a los inventarios proyectados. Incluirán los resultados de las actividades desarrolladas por el Sistema en esta campo,

presupuesto de ventas, informes de valoración del inventario actual y del resultante de la planificación, etc.

Sobre la programación maestra. Recoge toda información empleada para la obtención de PMP, incluyendo informes de cambios y desviaciones del PMP.

Sobre la gestión de capacidad. Entre ellos, informes de cargas planificadas por RRP, informes de cargas derivadas del PMP, elaborado por el CRP, etc.

Sobre la gestión de talleres. Abarcan toda la información resultante del procesamiento de pedidos en los CT, como la necesaria para la actividad del programador.

Sobre la función de compras. También muy numerosos, permiten obtener información sobre la situación de los pedidos en curso de un ítem o un proveedor.

Otras salidas. Básicamente incluye los listados de cualquiera de los diferentes registros de la base de datos con diversas ordenaciones, como las informaciones derivadas de las transacciones.

VENTAJAS

- Aportaciones a la dirección y gestión de la empresa.
- Permite una gestión anticipada, permitiendo además simular las consecuencias de cualquier evento sobre dichos programas.
- Facilita la integración, el consenso de criterios y un aunamiento de esfuerzos para alcanzar el mismo objetivo.
- Impacto sobre la exactitud de los datos empleados y las información generadas.
- El MRPII cuenta con sistemas muy avanzados de detección de errores en la introducción de datos, así como de salidas para la retroalimentación con vistas a determinar divergencias.
- Emplea una base de datos única, y su consecuencia se reduce al número de empleados dedicados a estas tareas y disminuye la probabilidad de error. Además obliga a disponer de unos procedimientos claros y detallados de forma que cualquier persona realice la misma tarea de la misma forma.
- Impacto sobre los inventarios.
- La programación permite a los sistemas MRP acercarse al objetivo de disponer de los stocks necesarios justo a tiempo, por lo que se eliminan en gran medida los stock de seguridad y se aumentan la rotación de los inventarios. Se puede hablar de reducciones de la inversión en inventarios de entre el 10% y el 50%.

- Impacto sobre la información y el nivel de servicio a clientes.
- Gracias a la capacidad de programación se pueden conocer las fechas de emisión y entrega con mucha antelación, por lo que se puede proporcionar al cliente una fecha prácticamente exacta de entrega de su pedido.
- Las mejoras de los niveles de servicio son del 26% y de las entregas de los pedidos en la fecha prometida desde el 90 al 97%.
- Impacto sobre la productividad del trabajo. MRPII puede lograr importantes mejoras en la productividad del trabajo, siendo las más importantes las conseguidas en la mano de obra directa.
- Gracias a la integración de la gestión de las diversas áreas en un sistema computarizado como MRPII, se puede lograr reducir en parte del trabajo administrativo al disminuir la documentación empleada y los pasos de esta.
- Impacto sobre compras. Reducción de papeleo mantenido por el personal de compras mayor tiempo disponible para comprar, pues, al conocer las necesidades y sus fechas con mayor anticipación, puede negociarse con los proveedores, consiguiendo contratos anuales y comunicándoles las necesidades futuras de la empresa.
- Impacto sobre los costes de transporte.
- Los retrasos y urgencias en el cumplimiento de las fechas de entrega, como también la descoordinación entre producción e inventarios, hacen que se eleven en muchos casos los costes de transporte.
- La mejora que en ambos puntos consiguen MRPII evitara muchas de estas urgencias, logrando reducción en el coste del transporte hasta en un 15%.
- Reducción de la obsolescencia y aumento de la productividad del Departamento Técnico.
- Mejora de la posición competitiva de la empresa.
- Mejora del grado de satisfacción de los clientes.
- Mejor control de los inventarios y estimación de los costes. Mayor calidad y exactitud de la presupuestación.

INCONVENIENTES

- Alto coste
- Dificultad de implementación
- Defectos técnicos

4.5 Elaboración del plan maestro de mantenimiento: Naturalmente, las modalidades y procedimientos a utilizar concretamente en la elaboración del plan maestro son función del tipo y problemática de la empresa y del sistema productivo considerado.

El punto de partida es la información del plan empresarial. Esta información proviene, en general, de los puntos distintos de la empresa y puede tener formatos

diferentes, su sintetización en un único esquema (tanto referido al mantenimiento como al tiempo) es lo que se denomina Plan de Demanda de Mantenimiento.

A partir del plan de demanda se establece un plan maestro agregado tentativo lo que, dada nuestra estructura de planificación, no ofrecerá dificultades mayores pues bastará con utilizar el plan elaborado en el ciclo anterior de planificación sometiéndolos a dos tipos de retoques:

a.-Añadir la programación de mantenimiento de las máquinas adquiridas o por adquirirse.

b.-Corregir, en su caso, el programa inicial de mantenimiento si se consideran que se van a producir desviaciones importantes entre los planificado y los reales.

Las necesidades de carga se comparan con las disponibilidades, capacidades globales, y se determinan las diferencias, las cuales comprenden no solo una cuantía sino también una posición temporal. En el caso de que las diferencias sean insignificantes el plan es factible; en caso contrario, es preciso proceder a la modificación del plan tentativo hasta lograr la factibilidad. Los procedimientos utilizados para la modificación del plan cubren un amplio espectro, desde los automáticos, basados en la programación matemática, hasta los manuales prueba y error. En general, un planificador entrenado, mediante la ayuda de un sistema informático, puede obtener buenos resultados como se ve frecuentemente en la práctica.

Es muy importante mencionar que para elaborar el Programa Maestro de Mantenimiento se considere los tipos de mantenimiento preventivo y predictivo, así mismo considerar las siguientes preguntas:

a.- ¿Qué tipo de mantenimiento vamos a realizar?

b.- ¿Cuándo vamos a realizar dicho mantenimiento?

La predicción de mantenimiento es una predicción de cara a las necesidades de los equipos (clientes). El hecho de que se realice el mantenimiento adecuado depende de:

a.- Levantamiento de información del comportamiento de los equipos.

b.-El grado de incertidumbre existente en la previsión realizada, de manera que se puedan atender mantenimientos inesperados.

4.6 ELABORACIÓN DE LA LISTA DE MATERIALES

Esto establece:

a.-¿Qué materiales necesitamos para realizar el mantenimiento requerido por el Programa General de Mantenimiento?

b.-¿Cuándo se necesitan los materiales, para cuadrar con el programa maestro de mantenimiento de reposición de almacén?

El proceso es análogo al anterior, salvo el nivel de detalle empleado. A partir de la información comercial se establece el plan de la demanda de producción (detallado), que permite la construcción del plan maestro detallado del mismo y a su vez elaborar el plan maestro de mantenimiento detallado.

Utilizando los datos técnicos se determinan las necesidades de carga. Los datos técnicos son análogos a los empleados para determinar la factibilidad del plan agregado, salvo que su nivel de agregación sea menor, coherente con el utilizado plan maestro detallado. Genéricamente se ha denominado macro ciclos y macro lista por constituir un tipo de información de la misma naturaleza para mayor claridad lo llamaremos con el nombre de lista de materiales.

La participación de los aspectos ligados a componentes de procedencia exterior será más importante aquí, sobretodo si se desea tener una estabilidad razonable en los programas sucesivos comunicados a los proveedores. El proceso del plan maestro detallado de mantenimiento, es el que se utilizará como punto de partida de MRP II.

Estas necesidades se comparan con el stock disponible. En caso de no contar con el material en stock se programa la compra de acuerdo a los Lead Time de cada repuesto.

4.7 FACTORES A CONSIDERAR EN LA APLICACIÓN DEL MRPII

Son miles las empresas que en los últimos años se han propuesto a instalar un sistema MRP, pero el porcentaje de las mismas que lo han hecho con éxito no es elevado. De estos fracasos ocurridos puede deducirse que la mayor parte han sido debido a una serie de factores relacionados con la puesta en marcha, así como un conjunto de pre-requisitos necesarios para un buen funcionamiento del sistema. La mayor parte de los autores especializados coinciden en resaltar la importancia de los siguientes elementos:

a. Exactitud en los datos de entrada, tanto el programa maestro de

mantenimiento, como la lista de materiales y el registro de inventarios deben responder a la realidad y mantenerse al día.

b. El programa maestro de mantenimiento debe ser realista en tres sentidos, pues su ejecución va a depender de la disponibilidad de materiales, de tiempo y de capacidad de recursos. La falta de componentes suele ser un síntoma de la existencia de problemas en algunos de los siguientes procesos: planificación de inventarios (cobrimiento insuficiente de las necesidades netas o tiempo de suministro real superior al previsto), compras (retrasos, calidad, etc.) y/o fabricación (defectos retrasos, falta temporal de capacidad, etc.). En algunos casos será posible actuar sobre los problemas de forma que puedan corregirse y sea posible cumplir el programa maestro de mantenimiento; en otros casos ello no será factible y será convenientemente alterar el programa de forma que corresponda a la realidad. Mientras que con los métodos tradicionales esto resultaba difícil o imposible dada la cantidad de elementos e interrelaciones, el MRPII lo permite con relativa facilidad.

c. Del mismo modo, la lista de materiales que guía la explosión de necesidades debe responder la estructura del producto debiendo reflejar cualquier cambio realizable al mismo. En cuanto al registro de inventarios, es necesario tomar conciencia de que los ficheros existentes no suelen estar adaptados a su utilización directa por el ordenador, al cual no deben pasar los errores existentes, además suelen ser bastantes menos precisos.

4.8 PROBLEMÁTICA DE LA IMPLEMENTACIÓN DE LOS SISTEMAS MRPII

Si algún factor ha contribuido a atenuar el ritmo de crecimiento del número de firmas usuarias de MRP II , a pesar de los cuantiosos beneficios que se podría haber obtenido, éste es sin duda, el alto número de empresas que han fracasado en su intento, o bien lo han abandonado o no han llegado a desarrollarlo lo suficiente como para convertirse en empresas tipo A. Estos altos índices de fracaso se deben, por una parte, al desarrollo de un proceso de implementación inadecuado en el que no se resuelven los importantes problemas que se presentan. Por otra parte, se da el hecho de que "todas las empresas que fallan en la implementación de sus sistemas MRP suelen continuar cometiendo los mismos fallos en sucesivas implementaciones".

4.8.1 Problemas en torno al personal

Uno de los requisitos fundamentales para el éxito del MRP II es que se eduque a todas las personas que desarrollan su actividad en la empresa. Así, según el estudio realizado por encuestadoras americanas un 23% de las empresas

consideraban que su fallo más importante en la implementación del MRP fue la insuficiente educación al personal. Resulta evidente que MRP es un sistema de comunicación, debido a ello es necesario que los usuarios empleen un vocabulario común, que cambien la rutina contraída con los procedimientos anteriores, los "vicios" y costumbres adquiridos con el tiempo, así como la toma de decisiones unilateral sin fijarse en las posibles repercusiones sobre el resto de las áreas.

Otro de los problemas fundamentales relativos al personal, que los estudios consideran causa importante de fracasos en la implementación de sistemas MRP, es la actitud ante el cambio. Existe miedo a la pérdida del puesto de trabajo, al ver que el ordenador desarrolla en todo o parte su tarea. Es importante acabar con este tipo de sentimientos ante el cambio, lo que implica de un lado hacer sentir al personal que el nuevo sistema es "su sistema" y no el de la Alta Dirección, para ello es conveniente y necesario el acercamiento de la alta dirección en forma continua hacia el personal de rango inferior pues ello influirá a la importancia que se debe tomar a la implantación del MRP.

Falta de exactitud en los datos. Uno de los preparativos más delicados en la implementación de un sistema MRP es el de depuración y corrección de todos los datos con los que el Sistema trabaja. Es necesaria una gran exactitud en las listas de materiales, programa maestro, registro de inventario, etc., que MRP II ha de utilizar, puesto que, en caso contrario se partiría de datos incorrectos y el Sistema estaría abocado al fracaso; el levantamiento de información depende muchas veces de la veracidad de los datos del personal y del apoyo que brinde.

Apoyo de la Alta Dirección. Se trata de un aspecto fundamental en la implementación del sistema MRP. El 20% según el estudio lo consideró como el más importante de todos los problemas acaecidos. Se suele delegar totalmente las responsabilidades en un director de proyecto con conocimientos técnicos del tema, lo cual es totalmente insuficiente, es necesario, un compromiso claro y continuado de la Alta Dirección con el proyecto y que este apoyo se haga patente.

Otros problemas. Existen otros problemas importantes como:

- Problemas en torno al software y hardware.
- Fallos en la elaboración y desarrollo del plan.
- Problemas Técnicos.
- Problemas de tipo Organizacional.

4.9 FILOSOFÍA JIT, "JUSTO A TIEMPO"

El JIT pretende que los clientes sean servidos justo en el momento preciso, exactamente en la cantidad requerida, con productos de máxima calidad y mediante un proceso de producción que utilice el mínimo inventario posible y que se encuentre libre de cualquier tipo de despilfarro o coste innecesario. El JIT es algo más que un método de planificación y control de la producción, es considerado como una verdadera filosofía, un proceso de mejora continua.

Básicamente el JIT tiene dos estrategias básicas:

Eliminar toda actividad innecesario fuente de despilfarro, por lo que intenta desarrollar el proceso de producción utilizando un mínimo de personal materiales, espacio y tiempo.

Fabricar lo que se necesite, en el momento en que se necesite y con la máxima calidad posible.

La teoría de los 5 ceros

Cero defectos. La calidad bajo la filosofía JIT significa un proceso de producción sin defectos en el que esta se incorpora al producto cuando se fabrica.

Cero averías o cero tiempo inoperativo. Evitar cualquier retraso por fallo de los equipos durante las horas de trabajo.

Cero stocks. Compara a la empresa con un barco que navega tranquilamente por un río plagado de rocas(problemas), un nivel adecuado de los inventarios (nivel de agua), podrá conseguir que la empresa (navegue) placidamente. La filosofía JIT lucha contra cualquier política de empresa que implique mantener altos inventarios al considerar a los stocks como el derroche más dañino. Disimulan diversos problemas, como incertidumbre en las entregas de los proveedores, paradas de maquinas, ruptura de stocks, etc.

Cero plazos. Elimina al máximo todos los tiempos no directamente indispensables, en particular los tiempos de espera, de preparaciones y de tránsito.

Cero papel o cero burocracia. El JIT en su lucha continua por la sencillez y eliminación de costes superfluos, entabla una batalla permanente contra la fábrica oculta. En este sentido intenta eliminar cualquier burocracia de la empresa.

4.8 EJECUCIÓN Y CONTROL: EL SISTEMA KANBAN

La norma de fabricar los componentes y enviarlos a donde se necesitan, empujando así el material a lo largo de las líneas de producción de acuerdo con el plan de materiales, caracteriza al MRP como un sistema de empuje, es mejor anticipar las necesidades antes de que estas se produzcan. En estos sistemas cualquier desviación con respecto a la programación da lugar a problemas. Por actuar de forma centralizada y por trabajar con tamaños de lotes y tiempos TS supuestamente constantes y predeterminados, cualquier cambio en la programación inicial puede dar lugar a una serie de dificultades. Por esto se emplea la reprogramación y en su caso el mantenimiento de cierto nivel de inventarios de seguridad. Estos sistemas de empuje la labor de control de la producción se concreta en intentar mantenerla dentro del programa.

En el sistema de arrastre utilizada por JIT ya no es el proceso anterior el que decide suministrar los componentes al proceso siguiente, sino que será el proceso siguiente el que le retira al anterior las piezas necesarias en la cantidad justa y en el preciso momento en que las necesite. Esto hace que el programa de producción solo sea comunicado como orden de fabricación al puesto de montaje final, desencadenando este todo el proceso de producción a medida que retira los componentes necesarios para montar los productos finales.

El sistema KANBAN es un sistema de arrastre basado en la utilización de una serie de tarjetas, que dirigen y controlan la producción entre los distintos CT.

4.8.1 Tipos de kanban y funcionamiento del sistema

Antes de poner en funcionamiento un sistema KANBAN es necesario realizar en la planta de producción una serie de transformaciones físicas.

Fijar el diagrama de flujos de forma que cada elemento provenga de una solo lugar y tengo solo un camino definido a lo largo de la ruta de producción.

Al suministrarse los almacenes, cada CT debe contar con una zona donde depositar sus inputs y almacenar los outputs o items elaborados.

Cualquier puesto de ensamblaje, deberá dividir su zona de inputs con lugares determinados para cada uno de ellos, operación similar ocurrirá con la zona de outputs.

La instalación de uno o mas buzones servirán para la recogida de los KANBANS.
Tipos de KANBAN

KANBAN de transporte o de movimiento que se mueven entre dos puestos de trabajo e indican las cantidades de producto a retirar del proceso anterior.

KANBAN de producción, se mueven dentro del puesto de trabajo y funcionan como orden de fabricación.

4.8.2 Reglas

- Regla 1. El proceso posterior recogerá del anterior, del lugar adecuado, los productos necesarios en las cantidades precisas.
- Regla 2. El proceso precedente deberá fabricar sus productos en las cantidades recogidas por el proceso siguiente. Las reglas 1 y 2 permitirán que la planta de producción funcione como una línea de transporte ideal, utilizando el KANBAN como medio de conexión de todos los procesos.
- Regla 3. Los productos defectuosos nunca deben pasar al proceso siguiente, dado que cualquier producto defectuoso que sea encontrado por el proceso siguiente difícilmente podrá ser sustituido a tiempo, pudiendo originar una parada de la línea de producción.
- Regla 4. El número de KANBANS debe disminuirse, dado que cuando un contenedor está lleno de piezas debe tener adherido un KANBAN, la cantidad máxima de inventario entre dos puestos de trabajo coincidirá con la siguiente formulación:
$$\text{INVENTARIO MÁXIMO} = \text{capacidad del contenedor} \times \text{número de KANBAN puestos en circulación}$$

4.9 FILOSOFÍA JIT: ADECUACIÓN DEL SUBSISTEMA DE OPERACIONES Y OTROS ASPECTOS DE INTERÉS

Reducción de los tiempos de preparación y de fabricación (sist. SMED) SMED (single minute exchange of die), nace como un conjunto de conceptos y técnicas que pretenden reducir los tiempos de preparación hasta poderlos expresar en minutos utilizando solo un dígito, es decir, realizar cualquier preparación de máquinas en tiempo inferior a los 10 minutos. El final de estos procesos de mejora debería consistir en la eliminación misma de la necesidad de preparación, que puede implicar dos aspectos: estandarizar los componentes para que puedan ser utilizados en distintos productos o bien fabricar las piezas necesarias al mismo tiempo, ya sea en la misma máquina o en máquinas diferentes.

El tiempo de espera puede tener dos orígenes. En primer lugar desequilibrios en el tiempo de producción, y por otro lado el tamaño de los lotes a procesar. El tiempo de transporte ocurre cuando se utilizan grandes lotes, estos tiempos se convierten en un componente importante del tiempo total de fabricación. Lo que el JIT propone en el caso de transporte es la conexión directa con una línea de montaje, no diseña las secciones por funciones, sino en líneas multiprocesos. Para el tiempo

de espera, una de las propuestas del JIT es reducir estos tiempos por estandarización de la ruta de operaciones, entrenando a los trabajadores.

Al observar la propuesta para disminuir los tiempos antes mencionados, es fundamental la reducción de la dimensión de los lotes de procesamiento. Por otra parte es evidente que cualquier disminución del tamaño del lote permitirá reducir proporcionalmente el tiempo de ejecución del mismo. Se concluye con que la disminución del tamaño del lote se convierte en la llave de cualquier proceso que intente reducir el plazo de fabricación.

Algunas de las ventajas de este sistema se enumeran a continuación:

Se producen reducción importante del nivel de inventarios, mayor flexibilidad a la empresa para adaptar la producción a las fluctuaciones y modificaciones de la demanda.

Aumente la tasa de productividad

Los plazos de fabricación y de entrega son muy cortos.

Al trabajar con lotes pequeños, los problemas de calidad son mas rápidamente detectados y afectan a menor número de piezas.

Estandarización de las operaciones

Los objetivos se detallan a continuación:

Alta productividad por utilizar el mínimo de trabajadores posibles y eliminar todas las tareas o movimientos inútiles

Equilibrar todos los procesos en términos de tiempo de producción. Para ello se basa en la utilización del concepto de ciclo de fabricación.

Utilizar la mínima cantidad posible de trabajo en curso, a la cual denomina cantidad estándar e trabajo en curso.

Puede ocurrir que la ruta estándar de cada trabajador no siga la secuencia lógica de operaciones a realizar al producto, será entonces necesario mantener cierta cantidad de trabajo en curso entre máquinas para que éstas y los distintos trabajadores puedan operar de forma simultánea. A esta cantidad, que debe ser calculada para que sea la mínima, se la denomina cantidad estándar de trabajo en curso, que no incluye la existencia en los almacenes de productos terminados.

4.10 MRP y JIT: ¿DIFERENTES O COMPATIBLES?

Pese a las diferencias entre ambos enfoques, éstos no son irreconciliables, todo lo contrario, recogiendo las potencialidades de cada uno de ellos, su utilización con junta no sólo es posible sino que puede llegar a ser enormemente beneficiosa para las empresas:

Aunque utilicen técnicas distintas, ambos métodos comparten objetivos básicos, como la minimización de los costes, la reducción de inventarios o la consecución del máximo servicio al cliente.

No todas las empresas desarrollan un tipo de producción al que podríamos denominar puro, ya sea por tipo repetitivo o por lotes. En ambos casos no está claro cual sería el mejor método de gestión a aplicar.

Cada día parece estar mas claro que ambos sistemas pueden llegar a ser complementarios.

Algunos directivos encuestados indican que existe la necesidad de simultanear ambas técnicas (MRP JIT), ninguna de las dos cumple hoy todas las necesidades, pues existen artículos y/o áreas de la fábrica donde se pueden aplicar JIT y otras donde se consiguen mejores resultados con técnicas de MRPII. El óptimo se alcanza integrando ambos sistemas.

Muchas empresas que han optado por un sistema JIT, cuando ya estaban utilizando un sistema MRP, han seguido utilizándolo para planificar los programas de entregas de los proveedores y la operación final de montaje.

Empresas que sin reunir las condiciones óptimas para la aplicación del JIT, han conseguido resultados enormemente positivos con la aplicación de un sistema híbrido MRP-JIT, que le permite aprovechar las ventajas de ambos, utilizando MRP para labores de planificación y el JIT para la ejecución y el control.

4.11 PLANIFICACION Y CONTROL A MUY CORTO PLAZO

Hasta aquí se analizó los niveles de proceso de Planificación y Control de la Producción, partiendo de los Objetivos y Planes Estratégicos a Largo Plazo se llegó a la obtención del Plan de Materiales, ya a corto plazo, pasando por la elaboración de los Planes Agregado y Maestro de Producción y de los respectivos Planes de Capacidad. Por último queda el desarrollo de las actividades de Ejecución y Control ya en el último nivel de proceso.

Por un lado los pedidos contenidos en el Plan de Materiales no son independientes, existiendo entre ellos una relación de prioridad que es imprescindible respetar. Por otro lado las rutas de los ítems pueden pasar por diferentes CT, con el agravante de que diferentes pedidos de distintos ítems pueden requerir en su ruta operaciones que han de realizarse en las mismas instalaciones.

Teniendo en cuenta todo esto, que pedidos deberá elaborar cada CT?, en que orden deben realizarse?, Cuales son las fechas de comienzo y finalización de cada operación?

Por un lado, la asignación de los trabajos a las maquinas y su posterior secuenciación en las mismas están condicionadas por la capacidad disponible de cada una y por las disponibilidades de materiales en el momento de emisión de los pedidos. Por otro lado, la forma en que se haga conllevará importantes consecuencias, pues esta influirá en el tiempo total empleado en la realización de los trabajos, en el volumen de la producción en proceso, en la eficiencia, en los costes, en las fechas de terminación de los pedidos establecidos en el Plan de Materiales y por consiguiente en el nivel de servicio a clientes.

Considerando todo lo anterior, se llegará a un Programa de Operaciones para un horizonte de entre unas horas y varias semanas (en función del caso), el cual es imprescindible para que cada trabajador o cada responsable de una instalación, sepa en cada momento que ha de hacer para que se logre el Plan de Materiales y con ellos, el Programa Maestro, el Plan Agregado y los Planes y Objetivos Estratégicos de la empresa.

La Planificación y Control a muy corto plazo englobará actividades como cantidades de ítems realmente obtenidas tras las operaciones, la eficiencia y utilización de los CT, fechas de entrega, tiempos de suministros, evolución de las colas de espera y con ellas la adecuación de las capacidades y cargas planificadas a las reales, etc.; y que para el caso concreto de las empresas fabriles, identificaremos como Gestión de Talleres. Dichas actividades están encaminadas a programar, controlar y evaluar las operaciones de producción a muy corto plazo, para lograr el cumplimiento del Programa Maestro con la capacidad disponible y con la mayor eficiencia posible.

4.12 FUNCIONES BÁSICAS

Evaluación y control de los pedidos a fabricar del plan de materiales estableciendo los que han de emitirse en cada momento y elaborando la información necesaria para su emisión, tras comprobar la disponibilidad de los materiales que necesita.

Establecer las prioridades entre los pedidos ordenándolos por CT y asignándolos previamente a cada uno de ellos si fuera preciso, obteniendo así el Programa de Operaciones.

Rastrear la evolución de los pedidos en curso a través de los CT, estableciendo la situación de los mismos al final de cada jornada y controlando las cantidades de items, así se elaborará el Informe de Producción diario.

Controlar el desarrollo de las operaciones en los CT, estableciendo los tiempo empleados y desperdiciados.

Controlar la capacidad de cada CT, mediante la comparación de la carga y capacidad planificadas con las reales, estableciendo la evolución prevista de la cola de espera y las medidas de ajuste de capacidad necesarias a muy corto plazo para mantenerla en los niveles deseados.

Proporcionar realimentación al Sistema de Planificación y Control de Capacidad, así como a los niveles superiores de Planificación de la Producción.

Existen dos metas que resumen los criterios de muchas compañías: nivel de servicio y costes mínimos, este último implicará

Mantener el menor volumen de inventarios posibles

Emplear la menor cantidad de recursos posibles

Una vez que, de acuerdo con el Plan de Materiales, se ha comprobado la llegada de la fecha de emisión de un pedido, se lleva a cabo el proceso de revisión y autorización, en función de:

Disponibilidad de materiales para la elaboración del pedido.

Disponibilidad de capacidad en el correspondiente CT.

Pasado el proceso de revisión, y concedida la autorización, se procedería a la confección del pedido, que contendrá toda la información precisa sobre el mismo: ítem, cantidad, ruta que ha de seguir, materiales a obtener del almacén, etc.

Seguidamente, el pedido sería emitido, retirándose del almacén los materiales a emplear y trasladándolos al CT por donde ha de pasar en primer lugar. El pedido pasaría a considerarse en curso de fabricación, permaneciendo como tal hasta el momento en que se finalice y se reciba en almacén tras la preceptiva inspección.

4.13 INTRODUCCIÓN A LA PROGRAMACIÓN DE OPERACIONES

La Programación de Operaciones tiene como función determinar que operaciones se van a realizar sobre los distintos pedidos, durante cada momento del HP (Horizonte de Planificación), en cada CT, de forma que, con la capacidad disponible en cada uno de ellos, se cumplan las fechas de entrega planificadas, empleando el menor volumen de recursos e inventarios posibles.

El HP dependerá de las características del proceso productivo y su entorno, variando entre unas horas y varias semanas.

Configuración Continua, las empresas que trabajan este tipo de configuraciones suelen fabricar para inventario, empleando siempre las mismas instalaciones para obtener el mismo producto, con una disposición de las máquinas en cadena; la operaciones que realiza cada una de ellas es siempre la misma y no ha de ser determinada.

Configuración por lotes, donde un CT es empleado para obtener diferentes pedidos de distintos artículos. Esto implica que tras obtener un cierto lote, se parará el CT y se preparará para la producción del siguiente. No obstante, podemos encontrarnos con dos situaciones en función de las características de los procesos:

Distribución en línea, para la fabricación de pocos productos en lotes homogéneos de gran tamaño.

Distribución por funciones, donde la empresa fabrica bajo pedido, para obtener lotes de pequeño tamaño de una gran variedad de productos y componentes, de forma que las máquinas se agrupan en los CT en base a la función que desarrollan.

Configuración Job-Shop, donde cada lote puede diferir notable en términos de materiales necesarios, tiempo de procesamiento en cada CT, necesidades de preparación, etc.

Si la ruta de obtención de los distintos items es fija, el problema se centrará, en la determinación de la secuencia de paso de los pedidos, aunque ahora habrá que determinarla para cada equipo. Si, por el contrario, algunos items tienen rutas alternativas, de forma que una operación puede realizarse en varios equipos diferentes, será necesario proceder a una asignación concreta antes de establecer la secuencia de paso.

Carga de Talleres, asignación de los pedidos a los CT, indicando qué operaciones se realizarán en cada uno de ellos.

Secuenciación, establecimiento de la prioridad de paso de los pedidos en los diferentes CT para cumplir las fechas de entrega planificadas con la menor cantidad de inventarios y recursos.

Programación Detallada, determinación de los momentos de comienzo y fin de las actividades de cada CT, así como de las operaciones de cada pedido para la secuenciación realizada.

CONCLUSIONES

Se Diseñó un software para planificación, programación y control de la producción en la empresa INDUSOL S.A., ubicada en la ciudad de Pereira. Todo bajo los resultados del estudio de las plataformas exigidas, entre las que tenemos visual basic y access 2003.

Se realizó previo análisis a los sistemas actuales para lograr el entendimiento entre las teorías de MRP y el modo de trabajo de la Fábrica, creando un módulo de capacidad.

Se Implementó un módulo de programación utilizando la creación de centros de trabajo que manejan herramientas, operaciones y talleres. Permitiendo la creación de un módulo para el control, todo esto mediante rutas de operación, las cuales se pueden configurar por medio del software.

Se Definió e implementó un módulo que integre la información de producción con los inventarios y las órdenes de compra. Todo mediante un estudio de las bases de datos actuales, permitiendo la Generación de un módulo para la planificación.

RECOMENDACIONES

Se recomienda a la fábrica de pinturas INDUSOL, que el sistema actual con el que trabaja la fábrica, sea migrado a una aplicación web, para facilitar el mantenimiento, administración y control remoto para la programación y control de lotes de producción dentro de la fábrica.

BIBLIOGRAFÍA

GARCIA GONZALES, Jose Antonio; DOMÍNGUEZ MACHUCA, Migue Angel; RUIZ JIMÉNEZ, Antonio; ALVAREZ GIL, Maria Jose. Dirección de operaciones. Aspectos Tácticos y operativos en la producción y los servicios. España. McGRAW-HILL. 503 p.

RUMBAUGH, James; BLAHA, Michael; PREMERLANI, William; EDDY, Frederick; LORENSEN, William. Modelado y diseño orientado a objetos. España. Prentice Hall. 643 p.

MEYER, Bertrand. Reusable software. Great Britain. 514 p.

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. El proceso unificado de desarrollo de software. España. Addison Wesley. 438 p

BRUEGGE, Bernd y H. Dutoit, Allen. Ingenieria del software orientado a objetos. México. 576 p.

SOMMERVILLE, Ian. Ingeniería de software. España. Addison Wesley. 712 p

ANEXOS

ANÁLISIS Y DISEÑO DEL SISTEMA INTEGRADO DE INFORMACIÓN PARA PRODUCCIÓN EN INDUSOL

1. SITUACIÓN ACTUAL DEL PROBLEMA

Actualmente, INDUSOL cuenta con un sistema de información para controlar los procesos en cuanto a información financiera, gestión comercial (puntos de venta), cuentas por cobrar, cuentas por pagar e inventarios. Dicho sistema de información está representado por un software que fue diseñado a la medida, aplicando los procesos que se llevan a cabo en cualquier empresa colombiana y apoyando tareas específicas que no manejan ningún software comercial, como por ejemplo: Control de estupefacientes en la venta de disolventes y también la orientación de INDUSOL a manejar los clientes de una forma muy convencional, es decir, la creación de grupos (clientes) para tener un mejor control a la hora de aplicar las leyes colombianas que limitan la venta de algunos productos que ofrece la empresa.

INDUSOL en su política de software a la medida, no cuenta con un módulo de producción que esté totalmente integrado en línea con la aplicación que actualmente se encuentra en funcionamiento, dicho software a la medida requiere como necesidad básica para su implementación un análisis, diseño e implementación basados en modelos funcionales y dinámicos.

Se presentan problemas tales como: el control de inventario para materias primas y productos terminados (procesos manuales no confiables), el manejo independiente con el departamento de compras (generando procesos innecesarios), y otros diversos inconvenientes como lo son retardo en la planificación y programación de la producción, demoras en el material e inconsistencias en el tipo de material que debe entrar a cada parte del proceso; tiempos ociosos y de inconsistencias en la programación del personal; inconsistencias en los inventarios de materia prima, insumos, producto en proceso y producto terminado; no existen sistemas de control que permitan una adecuada toma de decisiones en el área de producción, entre otros problemas existentes.

Lo anterior creo la necesidad de sistematizar y crear un sistema integrado de información para la producción y de mejorar el sistema que se viene empleando, el cual será presentado de forma mas detallada a continuación:

LÍNEAS DE PRODUCCIÓN:

Las diferentes líneas de producción que se manejan actualmente en la empresa son:

- * Línea decorativa
- * Línea de lacas automotrices
- * Línea de disolventes
- * Línea de lacas industriales
- * Línea de materias primas terminadas para la fabricación de la misma (thiner, disolventes en general, xilol, etc.).

FABRICACIÓN: Se lleva a cabo en el nivel de producción, de acuerdo a la solicitud recibida por los clientes, aunque se pretende mantener un equilibrio.

Dependiendo del producto se realiza cada operación por separado; de acuerdo a los colores y la pigmentación, pues se debe evitar la lavada de las ollas. Luego se sacan unas muestras que posteriormente pasan al laboratorio para ser revisadas, se analiza su cubrimiento, secado, dureza, viscosidad. Concluido este proceso, se realiza el empaque, el almacenaje y el inventario.

PROGRAMACIÓN DE PRODUCCIÓN: Normalmente este proceso debe ser medido puesto que depende de muchos factores, como lo son la adquisición de materias primas para la elaboración de los productos pues a veces estas se escasean y esto genera retardos en la producción.

En este momento el funcionamiento y comercialización de la empresa sé esta llevando a cabo en su mayor parte a nivel de comunicaciones telefónicas y fax, en formatos de papel, lo cual es un punto en contra no permitiendo la evolución de la empresa y más hoy en día con tantas herramientas creadas por la tecnología, pero a su vez implicando un costo elevado en su implementación.

VENTAS: Se realizan a través del comité de ventas en asocio con el departamento de mercadeo.

El proceso se inicia realizando un estudio acerca de las distintas épocas del año determinando cual será la que provoque una mayor demanda y consumo del producto, con el fin de establecer la intensidad de la producción logrando cubrir de una manera plena el mercado e implementando como se distribuirán las ventas en el resto del año.

También se encarga de planear estrategias útiles para una buena comercialización del producto, para que llegue con satisfacción al cliente, creando una sensación de seguridad acerca de la calidad en el producto adquirido.

Para lo anterior se utiliza un asesor de mercadeo, quién es el encargado de llevar las pautas necesarias de un buen convencimiento hacia el posible cliente o comprador.

El mercado se distribuye a nivel de occidente y centro del país, en 9 departamentos y 27 ciudades

Las ventas también se clasifican de acuerdo al tipo de cliente, ya que pueden ser potenciales, particulares y a crédito.

Venta a potenciales: se realiza después de tener una investigación detallada del estado del cliente, comprobando su capacidad de pago y endeudamiento, posterior a ello se dispone a recibir la orden del pedido la cual se dirige hacia productos terminados; por último se realiza el despacho del pedido incluyendo una factura y recibiendo del cliente la respectiva consignación.

Venta a particulares: se realiza de contado y se expide una factura.

Venta a crédito: se reciben y se almacenan en un archivo los datos personales del cliente, se recibe la orden de compra, se hace el estudio en producción de la cantidad de productos terminados y se envía el pedido.

COMPRAS: De acuerdo al reporte recibido por parte del nivel de producción, se realiza un inventario de los insumos o materias primas, para así determinar el número de productos específicos.

Luego se envía el pedido al proveedor y este devuelve una remisión y factura de los elementos solicitados y la fábrica se dispone hacer la respectiva consignación.

Hay épocas del año en que la adquisición de estas materias primas se escasea por diferentes razones, es cuando se generan diferentes problemáticas en la empresa sobretodo en el ámbito de producción.

REQUERIMIENTOS: Base de datos para clientes potenciales a nivel nacional, particulares y aquellos que hacen parte del sistema de crédito.

Base de datos que contenga el inventario de insumos y de productos terminados.

Un software para planificación, programación y control de la producción en la empresa.

Implicaciones externas a ser consideradas: La empresa actualmente cuenta con proyección a nivel nacional, esto implica una mayor agilidad en las comunicaciones, requiriendo implementación en redes, para afianzar la comercialización.

Lista de características obligatorias: Realizar la implementación del sistema hasta que este en perfecto funcionamiento.

12. METODOLOGÍA

Fase 1: Recolección de información.

La persona que estaba a cargo de la producción de la empresa, se presto para hacer un recorrido por toda la planta y así realizó una breve narración de todos los procesos realizados en dicho lugar.

Además, como etapa inicial también se realizó una toma de información respecto a los requerimientos basados en las mejores técnicas que nos presenta la ingeniería del software.

Fase 2: Análisis y diseño del sistema.

En este proyecto se aplica el Análisis y diseño orientada a Objetos y se desarrolla en el lenguaje Unificado de construcción de modelos Unificado (UML) y los diagramas fueron desarrollados en el Programa Rational Rose.

Lenguaje Unificado de Modelado: UML es "un lenguaje de modelado. Es la notación (principalmente gráfica) de que se valen los métodos para expresar los diseños"¹²

La metodología OMT y la notación UML, permiten mostrar el proyecto en módulos simples y fáciles de comprender, enlazándose unos con otros los cuales van formando un modulo que hará referencia a todo el sistema general.

El programa Rational Rose es un software especializado para modelar sistemas bajo diferentes metodologías, en este caso se utilizó para graficar los diagramas del lenguaje de modelado UML.

¹² FOWLER, Martín. UML gota a gota. México. Addison Wesley Longman de México S.A. de C.V., 1999, 1 p.

Se canalizaron todos los requerimientos depurados y transformados en casos de uso que permitieron ver un entorno global del sistema.

Se tomaron los lineamientos de los casos de uso para generar un modelo de análisis que describe el sistema y sus subsistemas de forma eficaz, veraz y estructurada.

Se generó un modelo el diseño del sistema contemplando las plataformas requeridas y basados en el modelo de análisis generado en la etapa anterior.

Fase 3: Creación de la base de datos y las diferentes reglas del sistema.

Se realizó un estudio de la base de datos implementada para la administración de la información de las demás dependencias en la fábrica (contabilidad, cartera, inventario, facturación, pedidos) se estableció un modelo que integrara el sistema de producción con dichos datos ya existentes.

Fase 4: Creación de la interfaz de usuario y conexión con la base de datos.

En esta fase se realizó una interfaz amigable y fácil de manejar con el fin de que los usuarios del sistema se sientan a gusto manipulándolo, y así, en el momento de ponerlo en funcionamiento, no cause inconvenientes en los procesos manuales

Fase 5: Implementación de la herramienta de desarrollo

En esta fase del sistema, se Implementó una herramienta de desarrollo con cada una de las especificaciones indicadas en el diseño del sistema, permitiendo comunicación directa y en línea con los módulos que afecta.

Fase 6: Pruebas

Las pruebas son el proceso de encontrar diferencias entre el comportamiento esperado, especificado por los modelos del sistema, y el comportamiento observado del sistema¹³.

Se aplicaron las técnicas de prueba de ingeniería de software y métricas de desarrollo del sistema, que están definidas en los estándares internacionales; con el fin de generar una aplicación segura y confiable.

Fase 7: Implantación y control

Después de un estudio de la situación de la empresa en cuanto a redes, servidores y equipos de cómputo, Se implementa un sistema de capacitación para el personal

¹³ BRUEGGE, Bernd y DUTOIT, Allen. Ingeniería del software orientada a objetos. Mexico: Prentice Hall. Vol, no.1; 2002p. 327-328

de la planta con el fin de poner en marcha el sistema y proceder a la implantación de este.

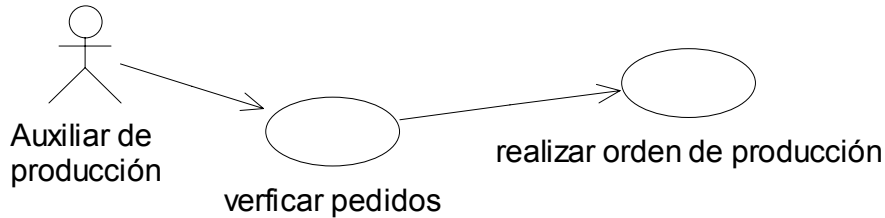
Fase 8: Evaluación

Esta es la fase final del proyecto, en el cual se sacan las conclusiones y resultados finales del sistema.

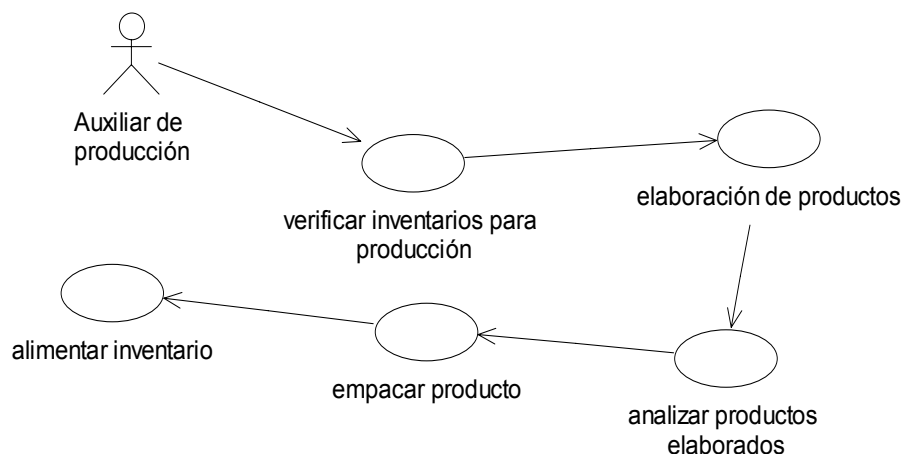
2. CASOS DE USO DEL PROBLEMA: CARACTERÍSTICAS

Estos casos de uso se indican en la parte inicial para establecer el área problemática desde el punto de vista de la metodología en el ciclo de vida. Todo con el fin de iniciar mejor la comprensión de las características en general.

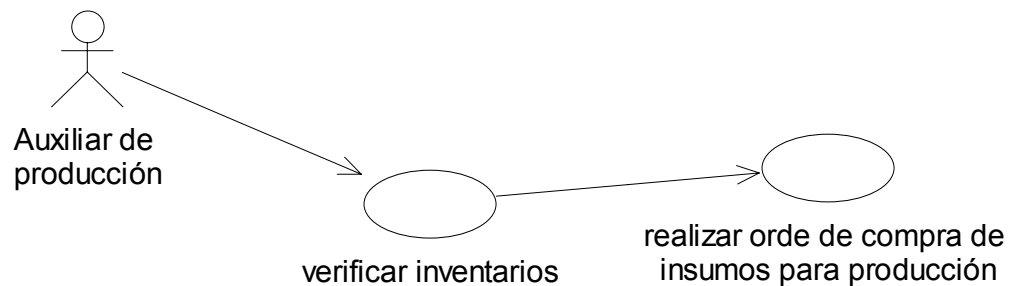
Caso de Uso-01	Programar producción
Descripción	Para realizar la programación de la producción es necesario verificar los pedidos realizados por los clientes, de acuerdo a la demanda de los productos en el mercado se genera una orden de producción.
Precondición	Muchas veces ocurre que no puede programarse la producción por falta de materia prima
Secuencia Normal	Verificar pedidos Verificar estadísticas Verificar productos a elaborar
Postcondición	Capacidad de las instalaciones
Excepciones	Ninguna



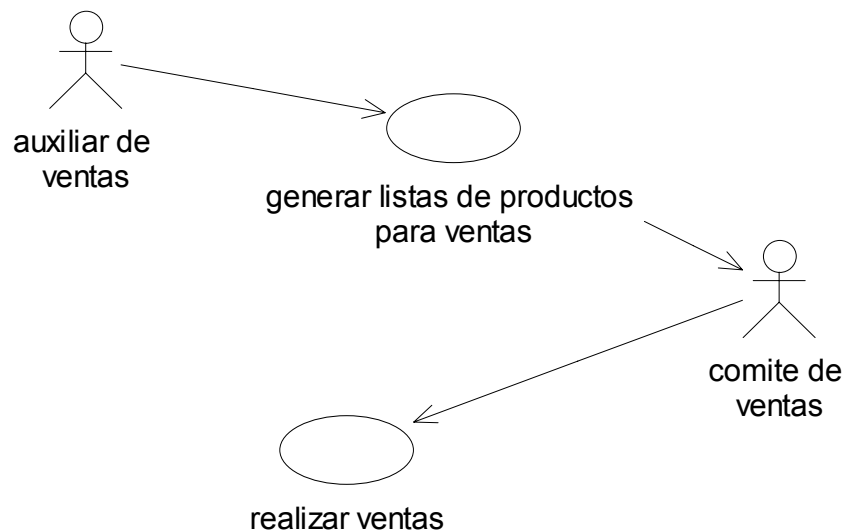
Caso de Uso-02	Fabricar productos
Descripción	En este proceso se elaboran los diferentes productos ofrecidos por la empresa, las diferentes líneas de producción que se manejan son: * Línea decorativa * Línea de lacas automotrices * Línea de disolventes * Línea de lacas industriales * Línea de materias primas terminadas para la fabricación de la misma (thiner, disolventes en general, xilol, etc.).
Precondición	Tener la materia prima necesaria para la elaboración de los productos
Secuencia Normal	Dependiendo del producto se realiza cada operación por separado, de acuerdo a los colores y la pigmentación, Luego se sacan unas muestras que posteriormente pasan al laboratorio para ser revisadas, se analiza su cubrimiento, secado, dureza, viscosidad.
Postcondición	Concluido este proceso, se realiza el empaque, el almacenaje y el inventario. Se debe evitar la lavada de las ollas.
Excepciones	Problemas en la obtención de materias primas



Caso de Uso-03	Realizar compras de insumos
Descripción	Para la elaboración de los productos se hace necesario la obtención de materias primas primordiales en estos procesos, de manera que la fabrica debe realizar las compras pertinentes para cada caso
Precondición	Que estos insumos estén disponibles en el mercado, pues a veces escasean
Secuencia Normal	De acuerdo al reporte recibido por parte del supervisor de fabricación. El jefe de producción realiza un inventario de los insumos o materias primas, para así determinar el número de productos específicos.
Postcondición	Luego se envía el pedido al proveedor y este devuelve una remisión y factura de los elementos solicitados y la fábrica se dispone hacer la respectiva consignación.
Excepciones	Problemas en la adquisición o transporte de materias primas



Caso de Uso-04	Realizar ventas
Descripción	Se realizan a través del comité de ventas en asocio con el departamento de mercadeo.
Precondición	
Secuencia Normal	El proceso se inicia realizando un estudio acerca de las distintas épocas del año determinando cual será la que provoque una mayor demanda y consumo del producto, con el fin de establecer la intensidad de la producción logrando cubrir de una manera plena el mercado e implementando como se distribuirán las ventas en el resto del año.
Postcondición	Se utiliza un asesor de mercadeo, quién es el encargado de llevar las pautas necesarias de un buen convencimiento hacia el posible cliente o comprador. El mercado se distribuye a nivel de occidente y centro del país, en 9 departamentos y 27 ciudades
Excepciones	Ninguna



4. DICCIONARIO DE DATOS

Clase: Mgr_operaciones		
Descripción: administra los sucesos. Protocolos y pasos necesarios par la fabricación de un producto		
Atributo	Descripción	Tipo de dato
*código		Texto
Nombre	Nombre de la operación	Texto
Detalle	Detalle de las operaciones	Texto
Tiempo_reparación	Tiempo de reparación	Texto
Tiempo_ejecución	Tiempo de ejecución	Texto

Clase: Mgr_inventarios		
Descripción: esta clase indica el registro de las existencias de mercancía		
Atributo	Descripción	Tipo de dato
* Producto	Nombre del producto	Texto
* Costo	Valor del producto	Moneda
* Ubicación	Donde se ubica dentro del inventario	Texto
* Origen		Texto
* Contador	Numero asignado dentro del inventario	Auto numérico
Cantidad	Cantidad de existencias	Número
Usuario	Usuario registrado en el sistema	Texto
Proveedor	Quien provee el producto	Texto
Precio venta	Precio al que se vende el producto	Moneda
Fecha	Fecha de inventario	Fecha
Descuento	Descuento que se aplica al producto	Número
Cantidad fija	Manejo de stoks mínimos de bodegaje	Número
Cantidad separada	Manejo de separación de mercancía	Número

Clase: Mgr_Proveedores		
Descripción: esta clase indica los proveedores de materias primas a la empresa		
Atributo	Descripción	Tipo de dato
* Nit	Numero de identificación del proveedor	Texto
Código	Código asignado al proveedor	Texto
Id	Numero de identificación	Número
Nombre	Nombre del proveedor	Texto
Dirección	Dirección del proveedor	Texto
Teléfono	Teléfono del proveedor	Texto
Ciudad	Ciudad donde se encuentra el proveedor	Texto
Departamento	Departamento donde se encuentra el proveedor	Texto
Url	Dirección web	Texto
Correo electrónico	Correo del proveedor	Texto
Por comisión	Porcentaje sobre comisiones	Número
Categoría	Tipo de proveedor	Texto
Contacto del contacto	Contacto dentro de la empresa que provee	Texto
Teléfono del contacto en la empresa	Teléfono del contacto	Texto
Contacto ca	Contacto alternativo	Texto
Teléfono contca	Telefono contacto alternativo	Texto
Ntc_pos		Texto
Ntc_iso		Texto
Cert_pos		Texto
Cert_dne		Texto
Ven_dne		Fecha/Hora
F_pago		Texto
U_compra	Última compra	Fecha/Hora
Val_u_compra	Valor última compra	Número
Reteiva	Reteiva que se aplica	Texto
Retefuente	Retefuete que se aplica	Texto
Reteica		Texto
Fecha creación	Fecha de creación del proveedor	Fecha/Hora
Usuario	Usuario que administra los	Texto

	proveedores	
Activo	Si es usuario activo o no	Número
Observaciones	Observaciones que se hacen sobre el proveedor	Memo
Cheques devueltos	Cheques que se devuelven al proveedor	Número
Valor cheques	Valor de los cheques que se manejan al proveedor	Moneda

Clase: Mgr_productos		
Descripción: Tabla que administra toda la información de los productos		
Atributo	Descripción	Tipo de dato
* Código	Código del producto	Texto
Clasificación	Tipo de clasificación del producto	Número
Activo	Si es un producto activo o no	Número
Detalle	Descripción del producto	Texto
Unidad	Unidad en la que se maneja el producto	Texto
Usuario	Usuario que administra el producto	Texto
Iva	Iva que se aplica al producto	Texto
Descargar	Descargas que se hace sobre el sistema en los productos	Texto
Rentabilidad	Cual es la rentabilidad el producto	Número
Máximo	Tope máximo del producto	Número
Mínimo	Tope mínimo del producto	Número
Porcentaje_iva	Porcentaje sobre el iva del producto	Número
Cantidad	Cantidad que se maneja del producto	Número
Con_cuota		Texto
Centro_costos	Centro de costos al que pertenece	Texto
Nodo_padre	Familia a la que pertenece el producto	Texto
Preciocontado	Precio que se maneja de	Moneda

	contado del producto	
Preciocredito	Precio que se maneja a crédito del producto	Moneda

Clase: mgr_Item_Factura_Compra		
Descripción: Ítems que se manejan dentro de la factura de compra, acá estará la descripción de cada uno de sus ítems de compra		
Atributo	Descripción	Tipo de dato
* Factura	Número de la factura	Número
* Producto	Nombre del producto	Texto
* Proveedor	Nombre del proveedor	Texto
* Contador	Numero del contador asignado dentro del sistema	Autonumérico
Cantidad	Cantidad que se esta comprando	Número
Costo	Valor asignado de la compra	Moneda
Anulado	Si este es anulado o no, pues no se puede eliminar	Texto
Usuario	Usuario que administra el sistema	Texto
Ubicación	Ubicación del producto dentro del inventario	Texto
Descuento	Descuento que se aplica a la venta	Número
Precio de venta	Cual es el precio de venta	Moneda
Costo us	Precio de venta en dólares	Moneda
Subtotal	Subtotal de la venta	Moneda
Valor descuento	Descuento que se da dentro de la venta	Moneda
IVA	Iva aplicado a la venta	Moneda
Valor IVA	Valor que aplica a la venta	Moneda
Total	Total de la venta	Moneda

Clase: Mgr_ubicación		
Descripción: manejo de ubicaciones en bodega		
Atributo	Descripción	Tipo de dato
* Código	Código de la ubicación	Texto
Descripción	Descripción de dicha ubicación	Texto
Tipo	Tipo de ubicación	Texto
Usuario	Usuario que administra las ubicaciones	Texto

Clase: Mgr_centros_trabajo		
Descripción: En esta clase se configuran todos los centros de trabajo a utilizar en el sistema de producción de la fábrica		
Atributo	Descripción	Tipo dato
Código	Código del centro de trabajo	Texto
Nombre	Nombre del centro de trabajo	Texto
Descripción	Descripción del centro de costo	Texto
Cod_herra	Código de herramienta a utilizar dentro del centro de costos	Texto
Cod_operacion	Código de operaciones a utilizar dentro del centro de costos	Texto
Tiempo_operacion	Tiempos de operación dentro del centro de costos	Texto

Clase: Mgr_tipo_herramientas		
Descripción: manejar las herramientas y maquinaria necesarias para cada operación		
Atributo	Descripción	Tipo de dato
*código		Texto
Nombre	Nombre del tipo de herramientas	Texto
Descripción	Descripción del tipo de herramientas	Texto

Clase: Mgr_prod_clasificación		
Descripción: clasificación de productos, ayuda a diferenciar entre materia prima e insumos		
Atributo	Descripción	Tipo de dato
*código		Numero
Descripción	Descripción de la clasificación del producto	Texto
Usuario	Usuario que administra el sistema	Texto

Clase : Mgr_producción		
Descripción: utilizado para almacenar las ordenes de producción		
Atributo	Descripción	Tipo de dato
*Nro producción		Número
Fecha	Fecha de producción	Fecha/Hora
Usuario	Usuario que administra el sistema	Texto
Cod_Formula	Codigo de la formula	Texto
Descargado	1 si se descargo a inventario de lo contrario 0	Numero
Costo_mano_obra	Valor de la mano de obra	Moneda
Otros_costos		Moneda
Gastos_aplicados		Moneda
Costos_insumos		Moneda
Costos_empaques		Moneda

Clase: Mgr_item_prod_formula		
Descripción: almacena la formula necesaria para la elaboración de los productos		
Atributo	Descripción	Tipo de dato
*cód_formula		Texto
Cod_familiia	Código de la familia a la que pertenece el producto	Texto
Uso	Si es insumo o producto	Numero
Cantidad	Cantidad a usar dentro de la fórmula	Numero
Orden	Orden de producción	Numero

Porcentaje_pertenencia	Porcentaje que se usa dentro de la fórmula	Numero
------------------------	--	--------

Clase: Mgr_taller		
Descripción: asocia son los diferentes citios de producción que hay dentro de la planta		
Atributo	Descripción	Tipo de dato
*codigo	Quien provee el producto	Texto
*nombre	Producto que se provee	Texto

Clase: Mgr_produc_formula		
Descripción: productos asociados a la fórmula		
Atributo	Descripción	Tipo de dato
*cod_formula	Codigo de la formula	Texto
*descripcion	Descripcion de la formula	Texto
Tiempo	Tiempo de preparación de la fórmula	Texto

Clase: Mgr_facturas		
Descripción: factura de ventas en la empresa		
Atributo	Descripción	Tipo de dato
*factura	Numero de la factura	Texto
*tipo	Tipo de factura	Texto
Nit	Nit de la empresa	Texto
Sucursal	Sucursal que expide factura	Texto
Cod_ciudad	Ciudad en la que se realiza	Texto
Total	Total de la venta	Número
Saldo	Saldo de la venta	Número
Fecha	Fecha que se realiza	Texto
Fecha_pago	Fecha en que se paga	Texto
Anulado	Si es anualda o no	Texto
Observaciones	Observaciones de la factura	Texto
Vendedor	Quien realiza la venta	Texto
Usuario	Usuario que administra el sistema	Texto

Subtotal	Subtotal de la factura	Número
Iva	Iva facturado	Número
Descuento	Descuento de la factura	Número
Comision	Comision sobre la venta	Número
Nro_pedido	Nro de pedido	Texto
Origen	Origen de la venta	Texto
Por_comision	Comision de venta	Número
Cod_pago	Numero para pago	Texto
Retención	Retención de fuente	Número
Retención_iva	Retención de iva	Número
Total_pagar	Total a pagar	Número
Total_letras	Total a pagar	Texto
Dias	Dias de pago	Texto
Obs_anulado	Observación si es anulado	Texto
Tambores	Tambores que se venden	Texto
Centro_costos	Centro de costos que la genera	Texto
Fecha_cancelación	Fecha de cancelación de la factura	Texto
Valor_excentos	Valor por excentos	Número
Despachado	Si es despachada o no	Texto
Fecha_entrega	Fecha en la que se entrega	Texto
Nit-org	Nit de la empresa	Texto
Vendedor	Nombre del vendedor	Texto

Clase: Mgr_proveedores_productos		
Descripción: asocia productos con proveedores		
Atributo	Descripción	Tipo de dato
*proveedor	Quien provee el producto	Texto
*producto	Producto que se provee	Texto
Preferencia	Cual es el nivel de importancia dentro de la producción	Numero
Usuario	Usuario que administra el sistema	Texto
Precio	Precio del producto	Moneda

Clase: Mgr_pedidos		
Descripción: se relacionan todos los pedidos hechos a la fabrica		
Atributo	Descripción	Tipo de dato
*Nro_pedido	Nro del pedido	Texto
Nit	Nit de la fabrica	Texto
Sucursal	Sucursal de la fábrica	Texto
Fecha	Fecha de realización	Texto
Anulado	Si es anulado	Texto
Observaciones	Observaciones del pedido	Texto
Vendedor	Quien realiza el pedido	Texto
Usuario	Quien administra el sistema	Texto
Condiciones de pago	Condiciones para pago del pedido	Texto
Autorizado	Si el pedido es autorizado o no	Texto
Facturado	Valor del pedido	Numero
Dias	Dias de entrega	Texto

Clase: Mgr_prod_familia		
Descripción: productos que se relacionan dentro de una familia de productos		
Atributo	Descripción	Tipo de dato
Codigo	Codigo del producto por familia	Texto
Descripcion	Descripcion del producto	Texto
Nivel	Nivel sobre el cual estara el producto	Texto
Nodo_padre	Nodo padre al que pertenece el producto	Texto
Clasificación	Clasificación del producto	Texto
Unidad	Unidad en la que se maneja	Texto
Producto_usable	Que tipo de uso tendra el producto	Texto
Can_prod_min	Cantidad mínima	Número
Can_prod_max	Cantidad máxima	Número
Stock_min	Stock minimo	Número
Stock_max	Stock máximo	Número

Can_inventario	Cantidad del inventario	Número
Inventario	Inventario al que pertenece	Texto
Iva	Iva	Número
Porcentaje_iva	Porcentaje del iva	Número
Centro_costos	Centro de costos al que pertenece	Texto
Con_cuota	Cuota	Número

Clase: Mgr_empleados		
Descripción: maneja los empleados de la planta		
Atributo	Descripción	Tipo de dato
*cedula		Texto
Nombre		Texto
Apellido		Texto
Dirección		Texto
Teléfono		Texto

Clase : Mgr_herramientas		
Descripción: Administra las diferentes herramientas que se manejan en la planta de producción		
Atributo	Descripción	Tipo de dato
*código	Código de la herramienta	Texto
Nombre	Nombre de la herramienta	Texto
Descripción	Descripcion de esta	Texto
Voltaje	Voltaje que maneja	Texto
Material	Material del que se constituye	Texto
Costo	Costo de esta	Texto
Factura	Factura de compra	Texto
Cod_taller	Taller al que se asocia	Texto
Cod_tipo_herramienta	Clasificación de esta	Texto

Clase : Mgr_item_producción		
Descripción: Items que se producen dentro de la planta		
Atributo	Descripción	Tipo de dato
*Nro producción	Nro de la produccion	Número
Cod_familia	Codigo de la familia a la que se asocia	Texto

Uso	Porcentaje de uso dentro de la fórmula	Número
Cantidad_ideal	Cantidad ideal de uso	Número
Cantidad_real	Cantidad real de uso	Número
Cantidad_formula	Cantidad de la formula a producir	Número
Exceso	Porcentaje de exceso dentro de la producción	Número
Valor	Valor de la producción	Número
Orden	Orden asociada a la producción	Número
Porcentaje_pertenencia	Que tanto influye el insumo dentro de la fórmula	Número

Clase Mgr_lotes		
Descripción: Lote de producción dentro de la planta		
Atributo	Descripción	Tipo de dato
*numero de lote		Texto
Cod_formula		Texto
Cant_prod		Texto
Porcentaje1		Texto
Porcentaje2		Texto
Nro_produccion		Texto

Modelo Dinámico
Diagramas de comportamiento
Diagrama de estados

Diagrama de estados: Configuración centros de trabajo

En este diagrama se puede mostrar cuales podrían ser los estados de transición a la hora de configurar un centro de trabajo, el cual se encargara de integrar diferentes departamentos de la fabrica a la hora de trabajar en la producción de un compuesto.

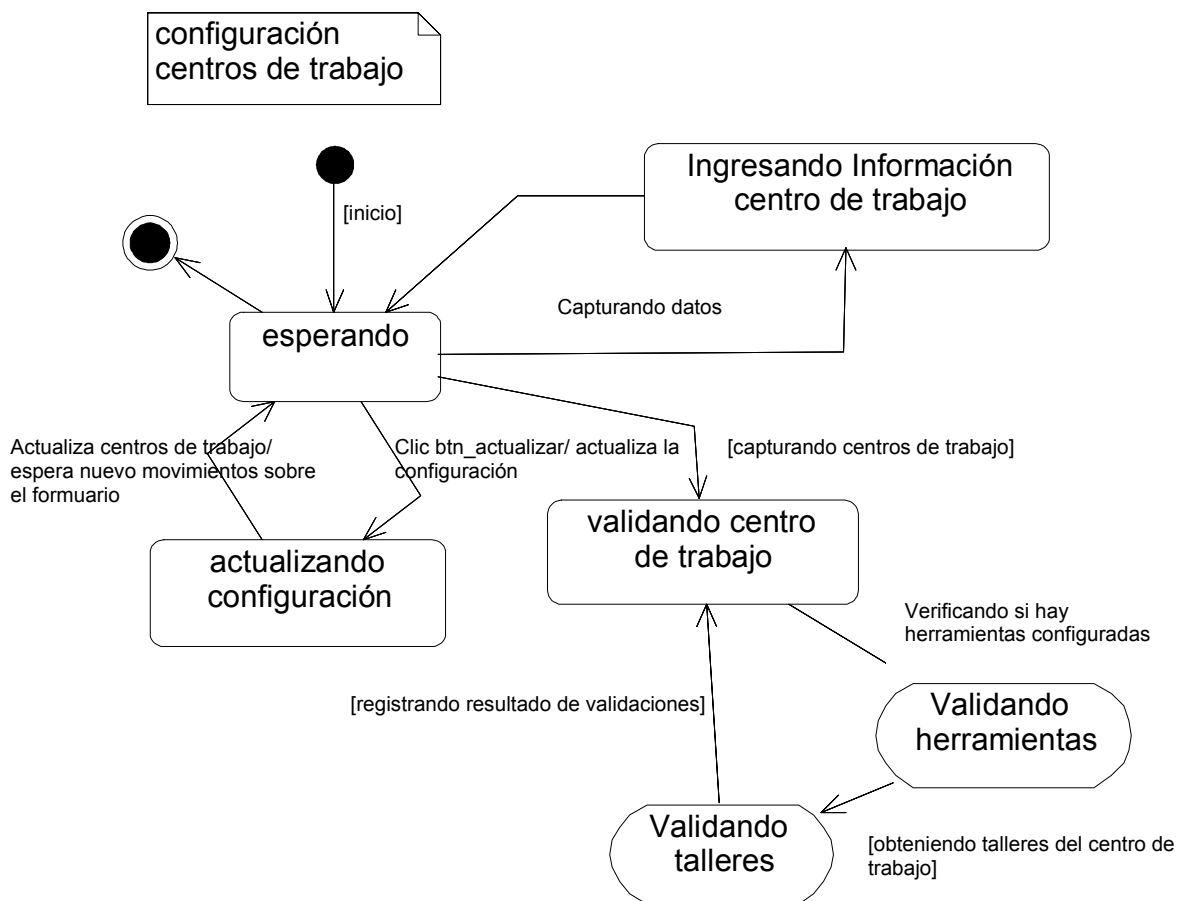


Diagrama de estado: Configuración de orden de producción

La orden de producción es la que da el parte de producción por lotes, o por cantidades de formulas a realizar.

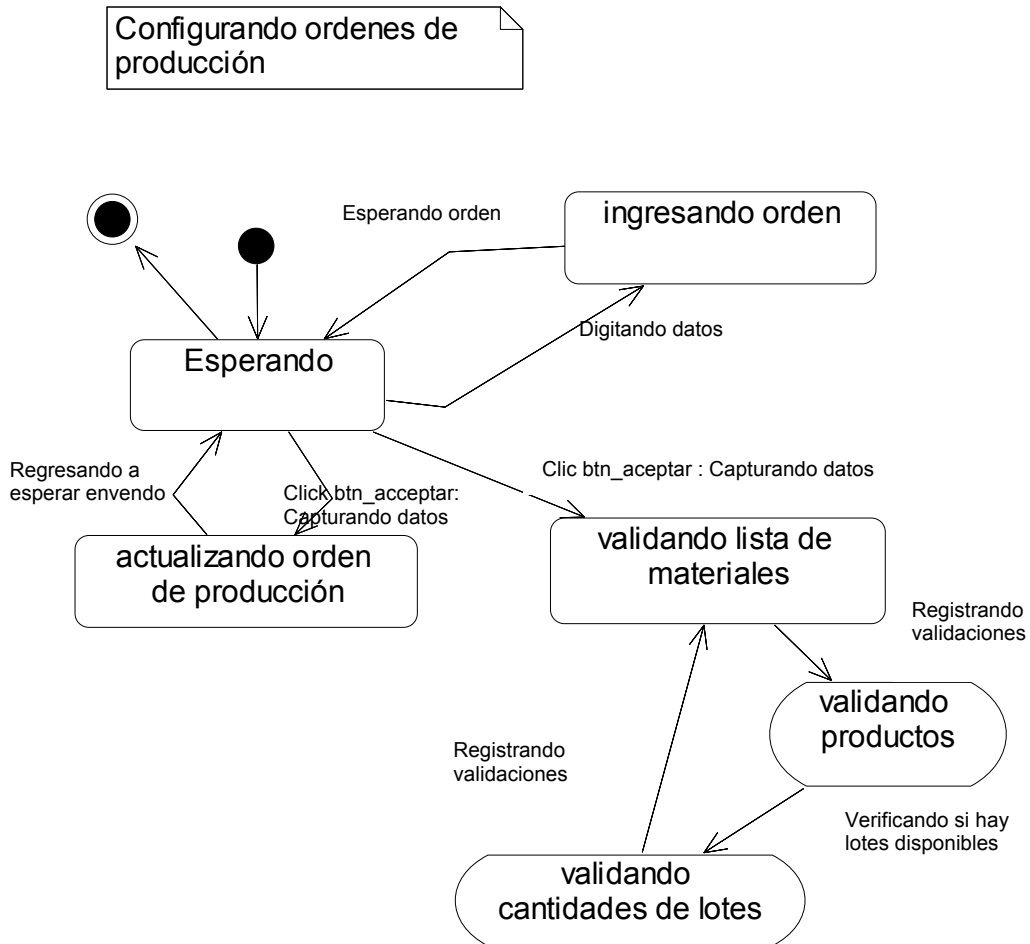


Diagrama de estado: Operaciones

Configura todas las operaciones que se realizar en el momento de realizar la elaboración de los diferentes de la empresa.

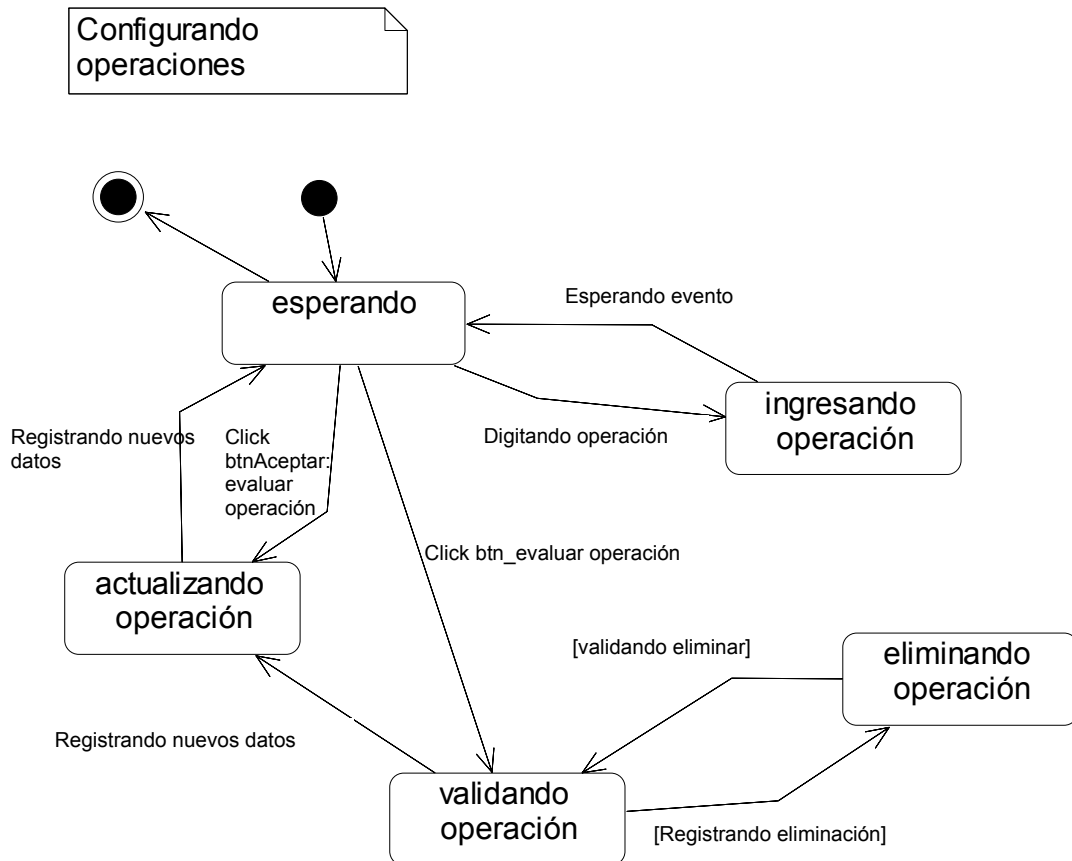


Diagrama de estado: Configurando rutas de producción

Es donde se muestra cuales serán las diferentes trayectorias de producción dentro de la empresa.

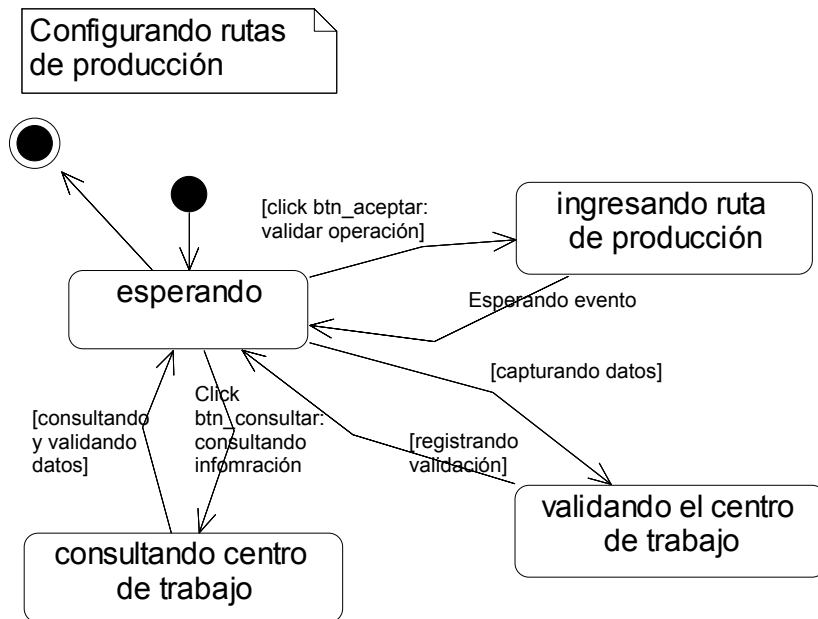


Diagrama de estado: Configuración de empleados

Permite el registro de los empleados que estarán trabajando sobre el sistema.

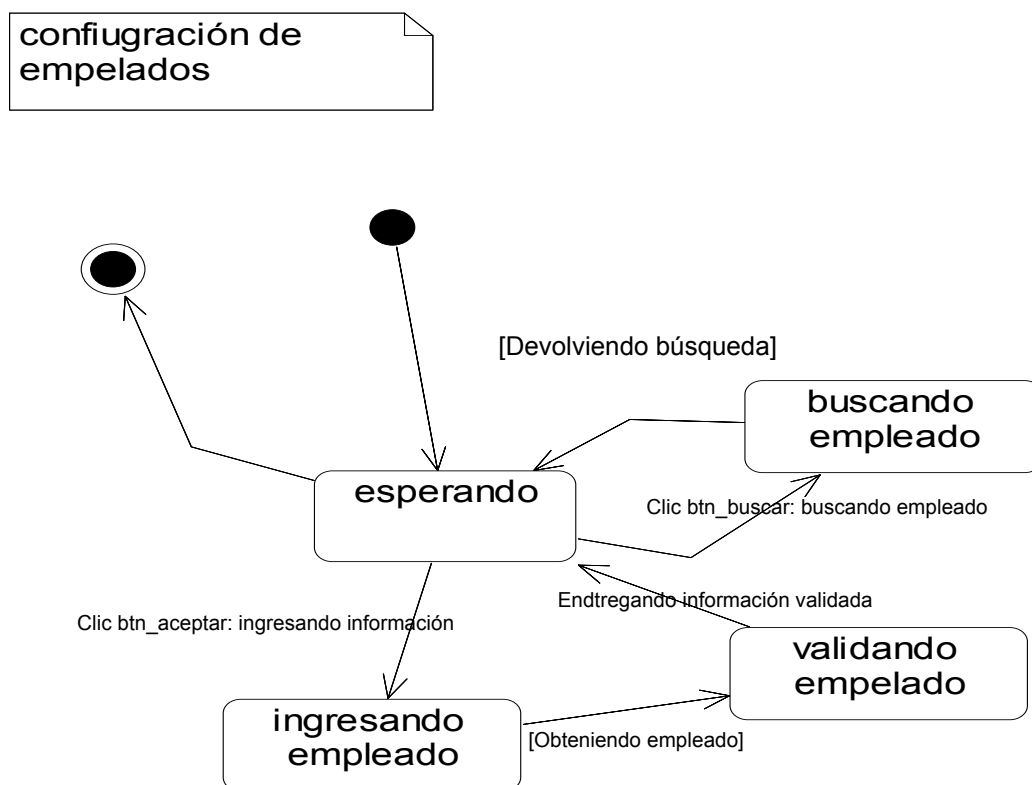


Diagrama de estados: Configurando herramientas

Permite ver las diferentes herramientas con las que se elaboraran los productos dentro de la empresa.

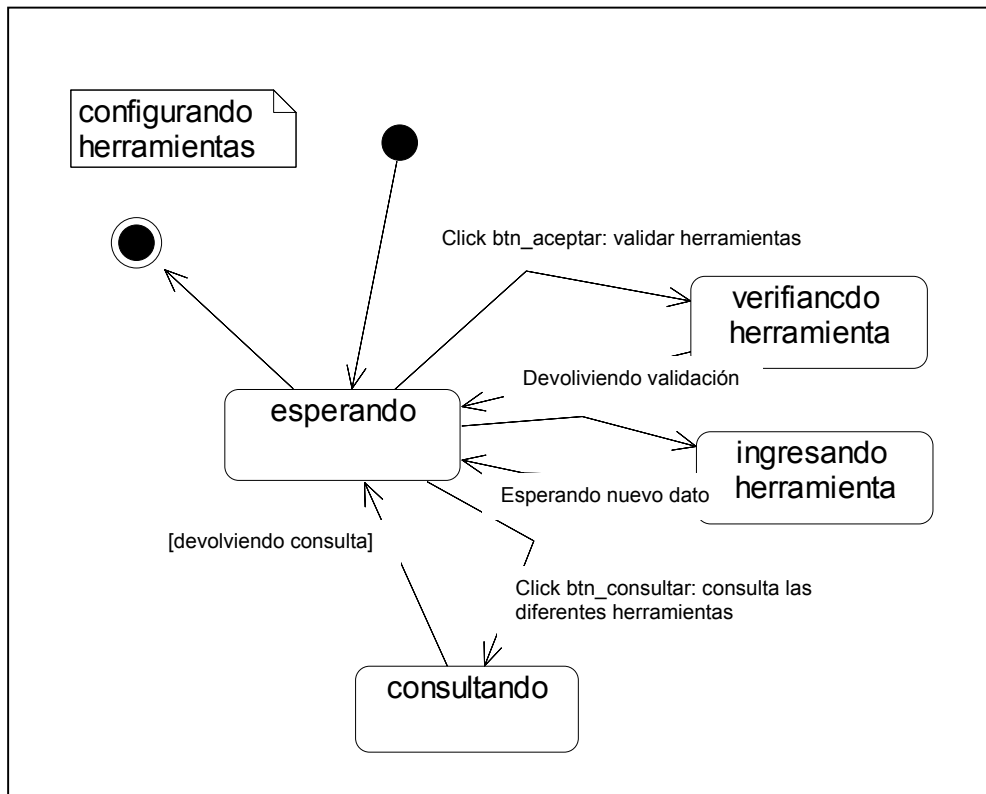


Diagrama de estado: Configurando producción

Muestra como se configurara la producción de los determinados productos de la empresa.

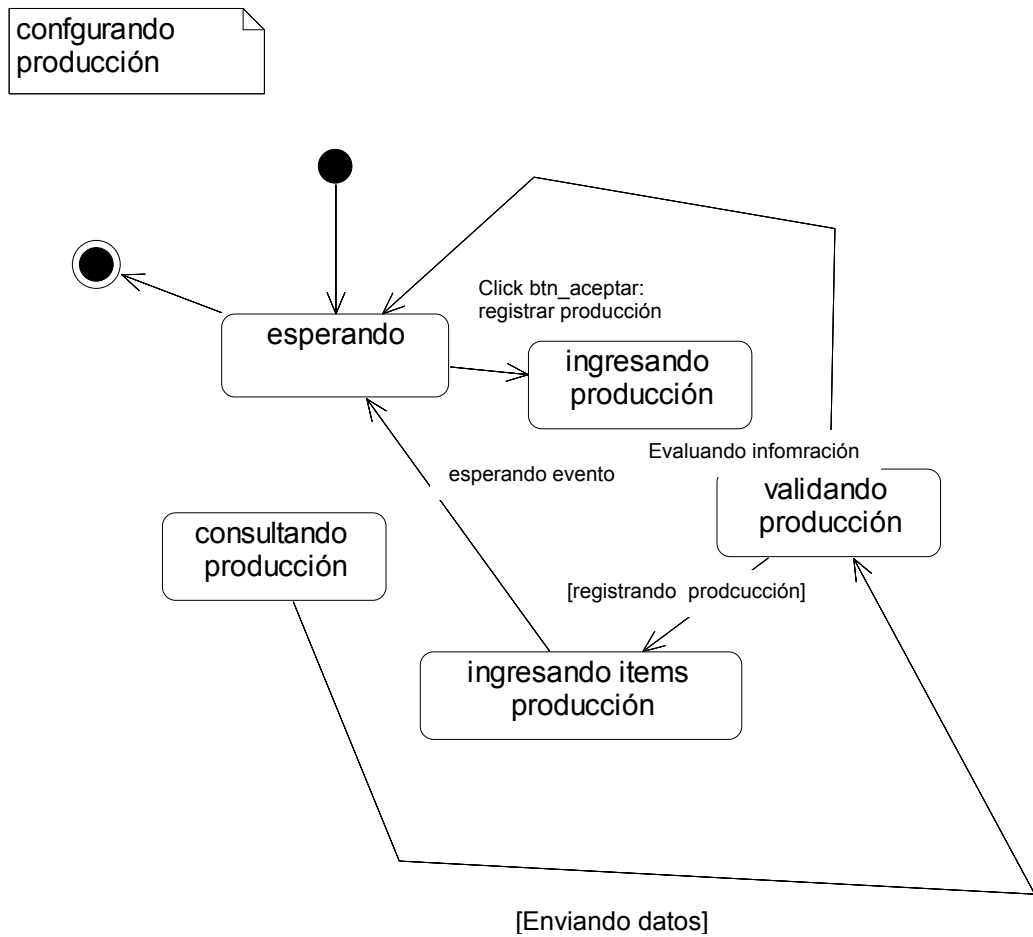


Diagrama de secuencia: Centros de trabajo

Permite configurar el centro de trabajo sobre el cual se va a realizar la producción de determinado producto en la fábrica.

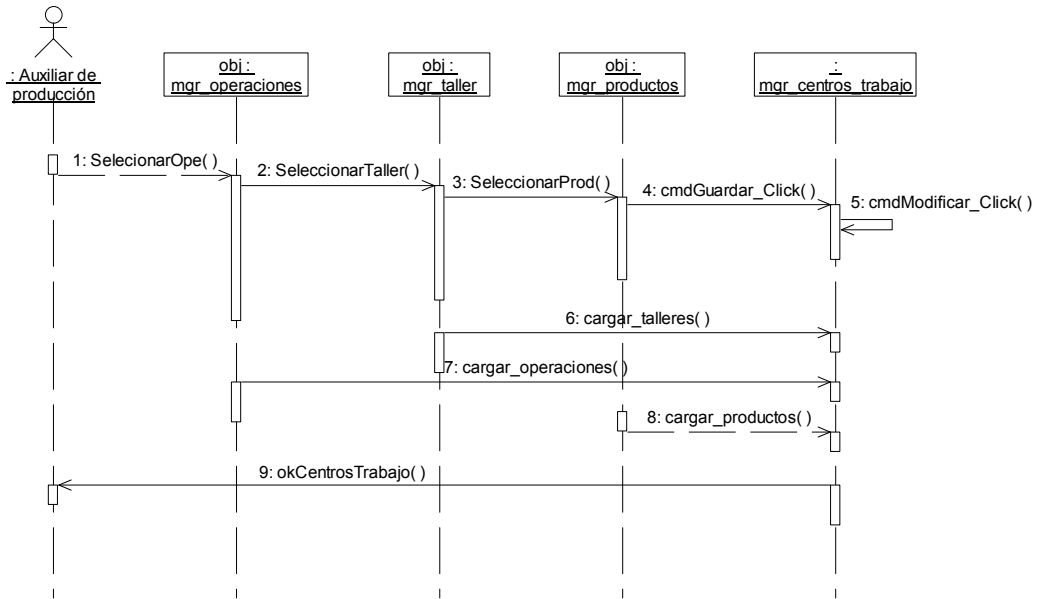


Diagrama de secuencia: Configurar rutas de producción

Una ruta de producción será los pasos a realizar para poder llegar a producir determinado producto.

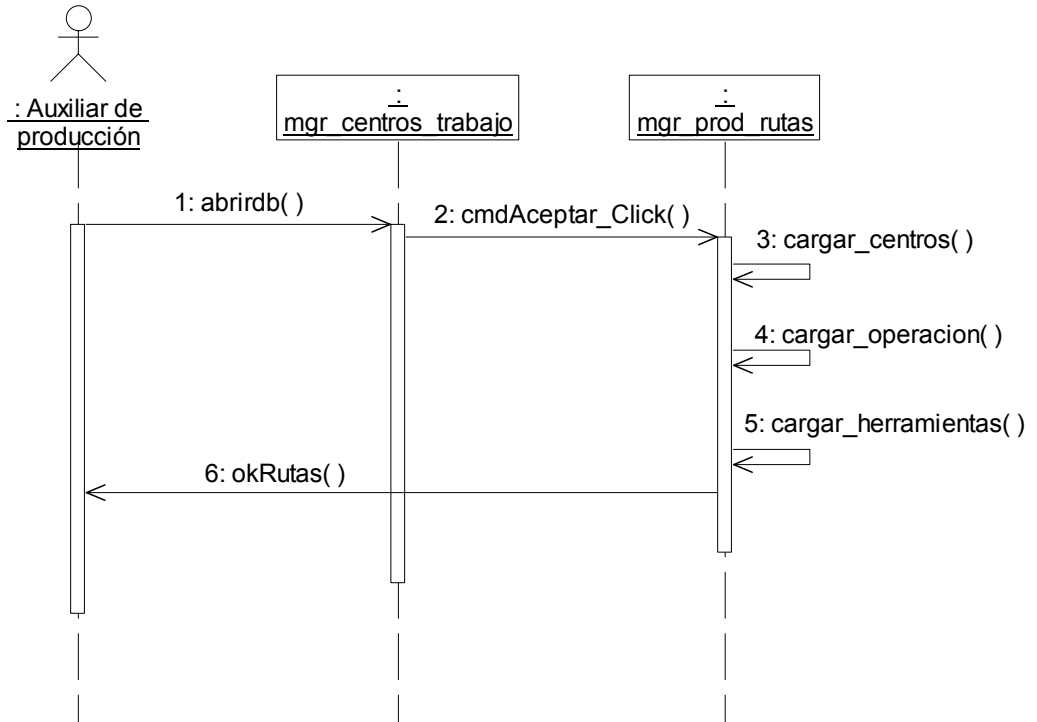


Diagrama de secuencia: Configurar Herramientas

Se muestra cuales son las herramientas que se trabajaran dentro de la producción de determinados productos.

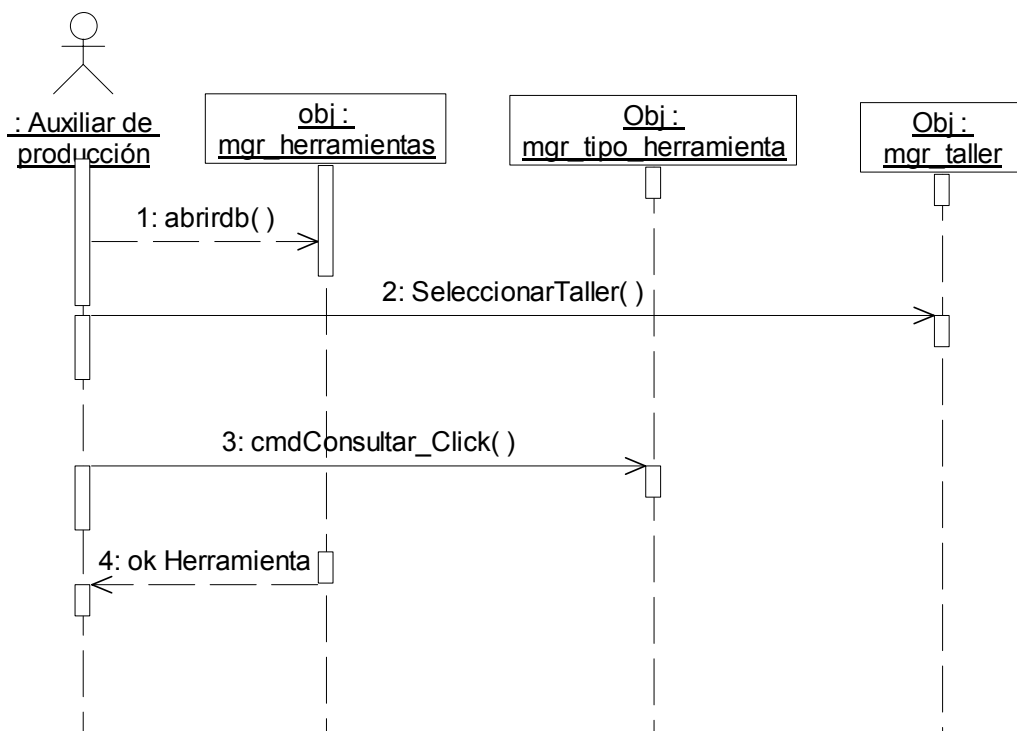
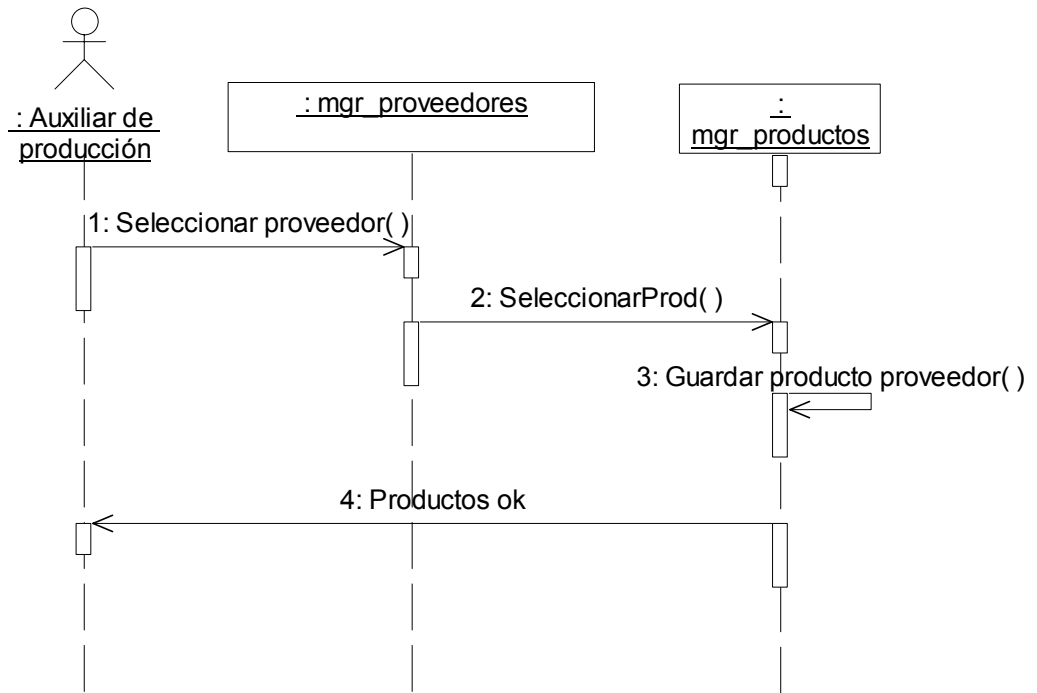


Diagrama de secuencia: Configurar productos por proveedores

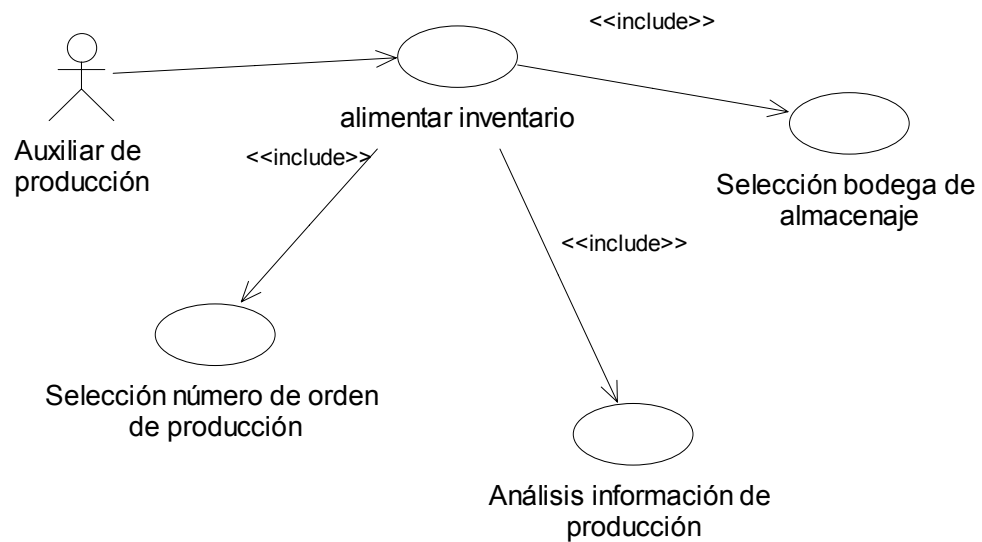
Se muestra como se configurarán los proveedores según los productos que provee a la planta de producción.



Modelo Funcional
Diagrama de Casos de Uso

Caso de uso: Alimentar inventario

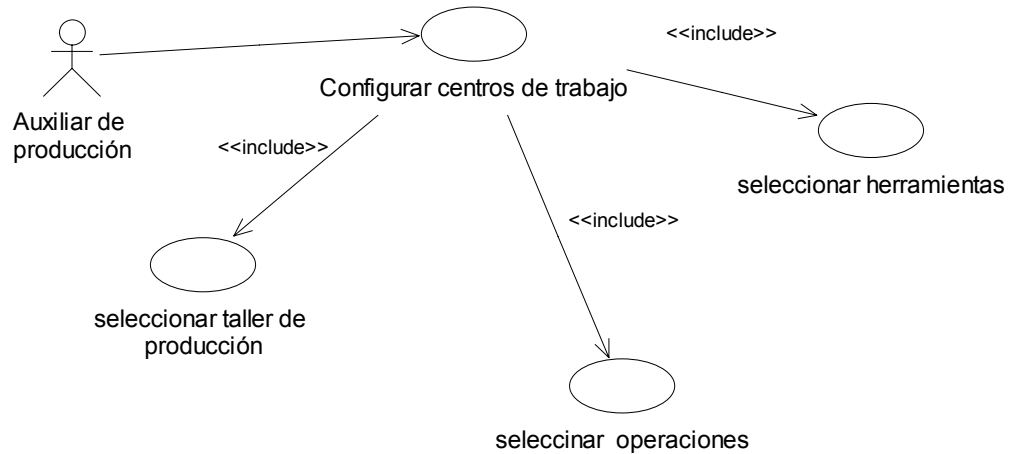
Muestra la forma como se alimenta el inventario dentro de la fábrica.



Caso de Uso-01	Alimentar inventario
Descripción	Cada que termina un proceso de producción, se realiza esta operación de acuerdo a las cantidades que se elaboraron
Precondición	Se debe realizar control de calidad
Secuencia Normal	1. El auxiliar de producción alimenta el inventario 2. Selecciona el número de orden de producción 3. Verificar que esté el producto almacenado en bodega
Postcondición	Seleccionar bodega de almacenaje
Excepciones	Problemas en producción

Caso de uso: Configurar centros de trabajo

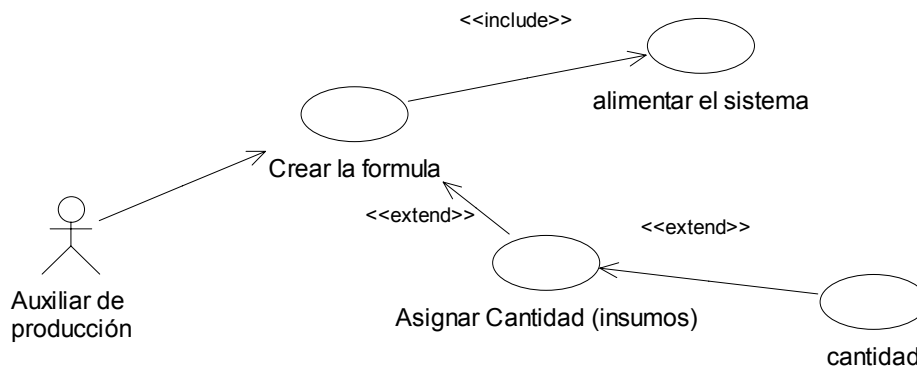
Según este caso de uso, así se manejan los centros de trabajo dentro de la fábrica, mostrando así las dependencias que se necesitan para poder configurar dicho centro.



Caso de Uso-02	Configurar Centros de trabajo
Descripción	Para crear un determinado producto, es necesario crear un centro de trabajo, que determina los elementos a utilizar de forma que suministre la información sobre herramientas, talleres y operaciones.
Precondición	Cada producto debe tener determinada la receta o fórmula para su elaboración.
Secuencia Normal	<ol style="list-style-type: none"> 1. El auxiliar de producción configura el centro de trabajo que se va a utilizar. 2. El sistema se encarga de asignar el taller de producción necesario. 3. El sistema se encarga de seleccionar las herramientas para la producción de determinado producto. 4. El sistema se encarga de asignar las operaciones para la producción de determinado producto.
Postcondición	Esta información queda almacenada en la base de datos y es utilizada a la hora de generar un lote de producción.
Excepciones	Ninguna

Caso de uso: Configuración de fórmulas

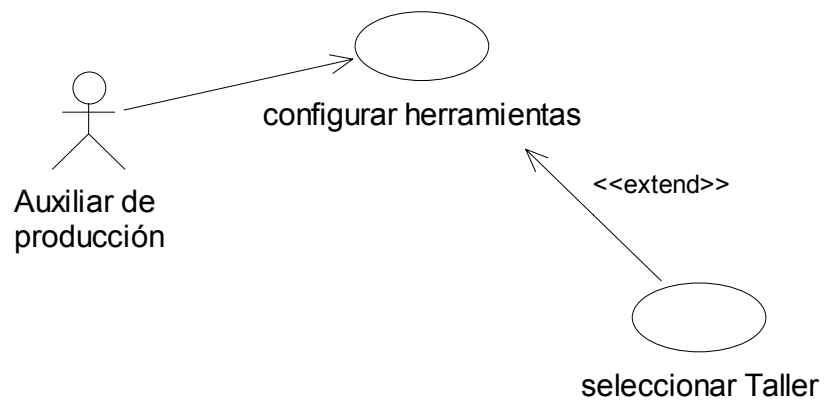
Las formulas que se manejan dentro de la fábrica las cuales son las que permiten la elaboración de los diferentes productos



Caso de Uso-03	Configurar formulas de producción
Descripción	En este proceso se llevan a cabo la aplicación de formulas necesarias para la fabricación de determinados productos
Precondición	Determinar el producto a fabricar
Secuencia Normal	Crear la formula Alimentar el sistema Asignar la cantidad de insumos
Postcondición	
Excepciones	Ninguna

Caso de uso: Configuración de herramientas

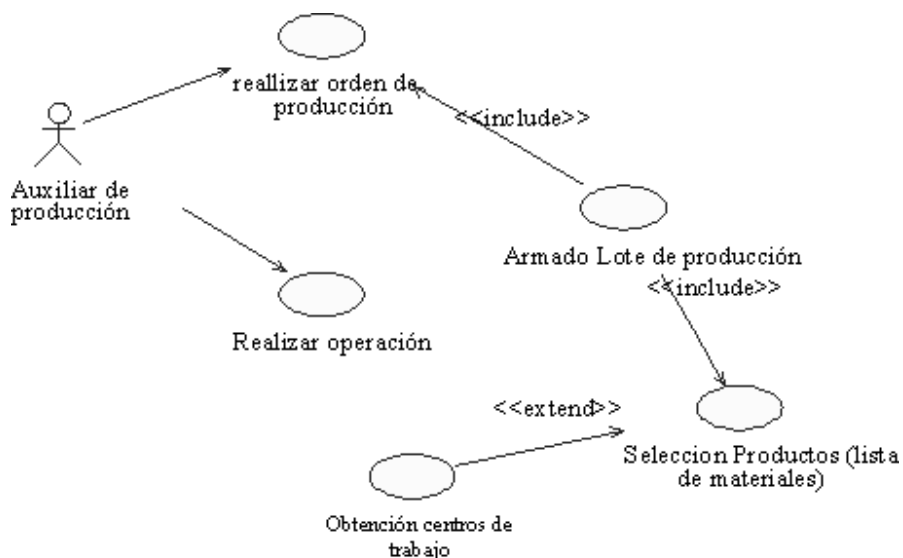
Muestra la configuración de las diferentes herramientas según el taller de producción.



Caso de Uso-04	Configurar herramientas
Descripción	Proceso por el cual se identifican las herramientas que van a ser utilizadas en el centro de trabajo.
Precondición	Cada centro de trabajo tiene sus respectivas herramientas
Secuencia Normal	El sistema configura las herramientas necesarias para la fabricación de determinado producto Seleccionar el taller que se va utilizar para este proceso Esta información queda almacenada en la base de datos
Postcondición	y es utilizada a la hora de generar un lote de producción.
Excepciones	Alguna de las herramientas del centro de trabajo este fallando

Caso de uso: Realizar orden de producción

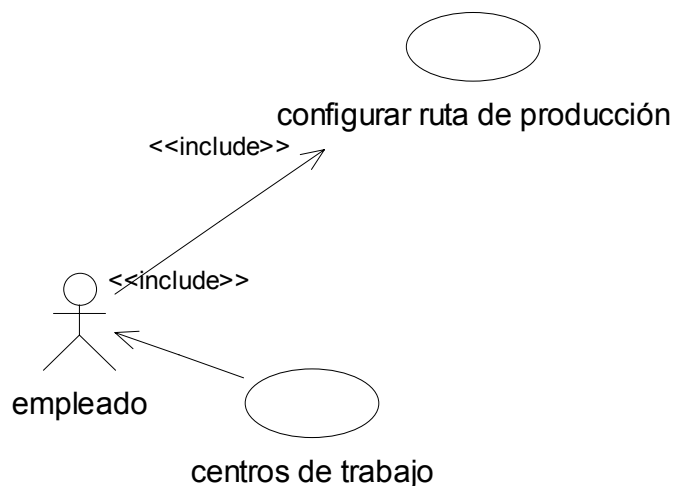
Muestra la manera de cómo se realizan las ordenes de producción



Caso de Uso-05	Realizar orden de producción
Descripción	Para realizar las ordenes de producción, el auxiliar de producción genera una orden de manera que se puedan seguir los demás procedimientos para la producción.
Precondición	Seleccionar los productos que se van a realizar .
Secuencia Normal	El auxiliar de producción realiza la orden de producción Al realizar la operación se obtiene el centro de trabajo Selecciona la lista de materiales Se arma el lote de producción
Postcondición	Tener las formulas o recetas de fabricación en el sistema
Excepciones	Ninguna

Caso de uso: Configuración de rutas de producción

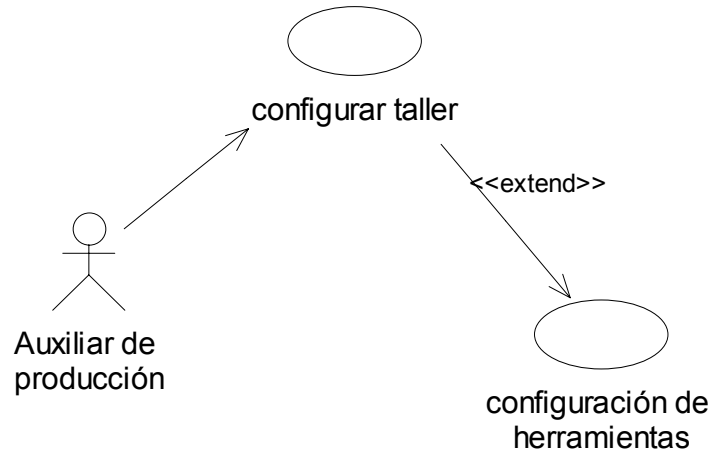
Muestra como se configura la ruta de producción dentro del sistema



Caso de Uso-06	Configurar rutas de producción
Descripción	Para este proceso el empleado con la ruta ya generada, se dispone a configurar la ruta como tal.
Precondición	Ya se ha recibido la orden de producción
Secuencia Normal	El empleado determina la ruta de producción el sistema esta en la capacidad de proveer la información sobre los centros de trabajo que se van a utilizar Esta información queda almacenada en la base de datos
Postcondición	y es utilizada a la hora de generar un lote de producción.
Excepciones	Ninguna

Caso de uso: Configuración de talleres

Muestra como se configura los diferentes talleres de la fábrica

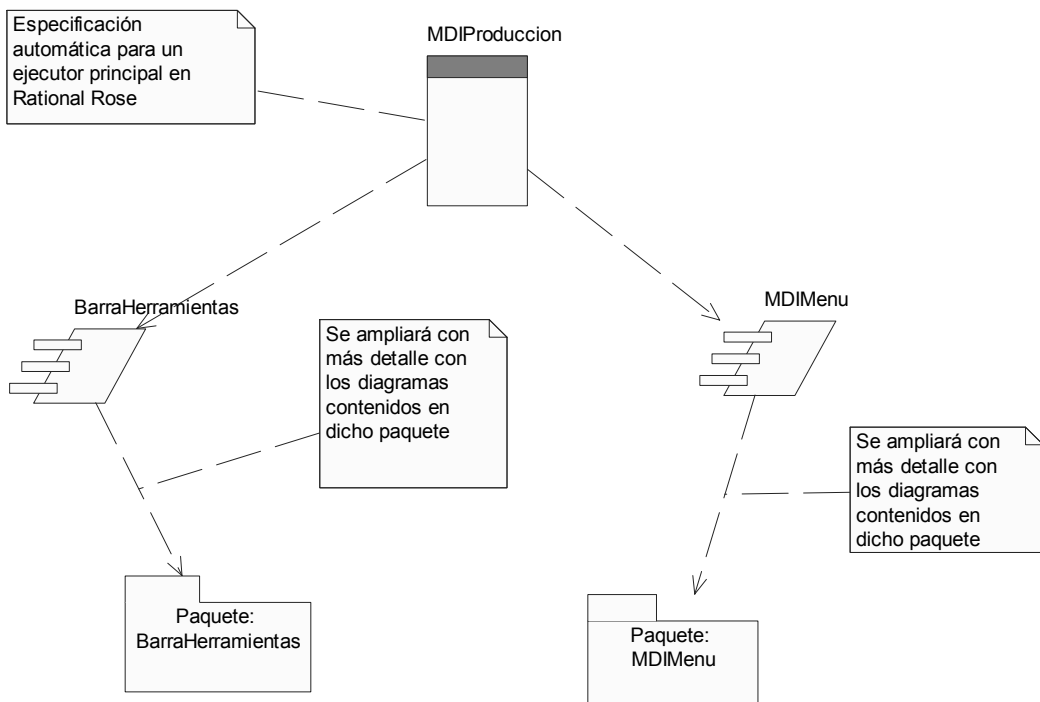


Caso de Uso-07	Configurar talleres
Descripción	El auxiliar de producción se encarga de configurar los talleres de acuerdo a la producción
Precondición	Que exista un lote de producción ya establecido
Secuencia Normal	El auxiliar de producción configura el taller El sistema predefine las herramientas que van a ser utilizadas
Postcondición	Esta información queda almacenada en la base de datos y es utilizada a la hora de generar un lote de producción.
Excepciones	Que exista alguna falla mecánica en las herramientas que están configuradas en los centros de trabajo

DISEÑO

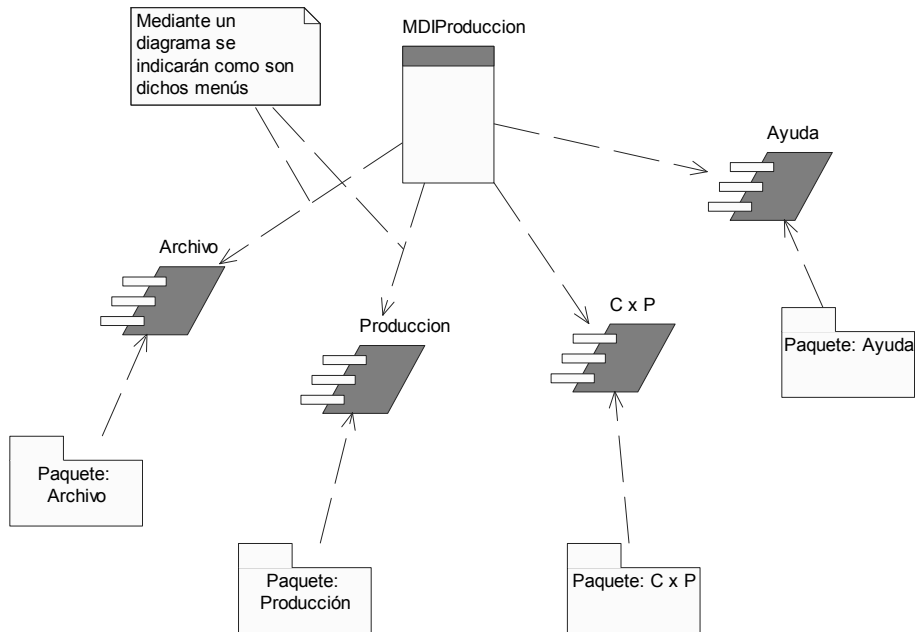
Diagramas de despliegues para especificar como estarán compuestos los elementos de la aplicación en cuanto a Diseño

Construcción de la aplicación principal. MDI GENERAL

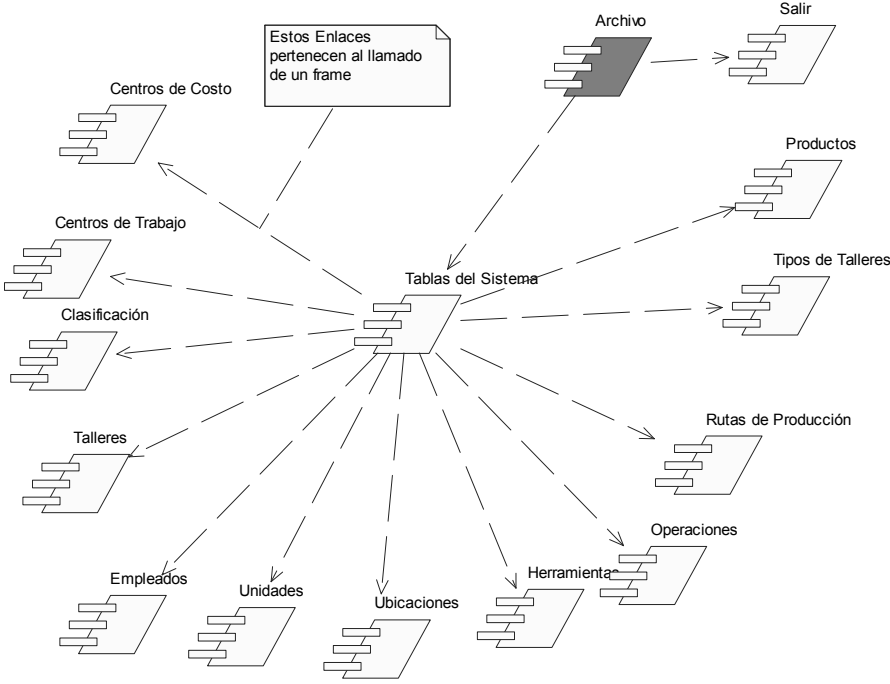


Paquete Barra de herramientas

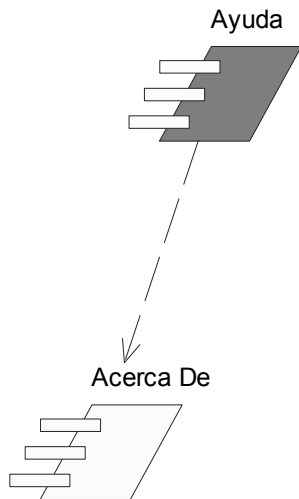
Este paquete describe graficamente como va a estar configurada la barra de herramientas que manejará el sistema.



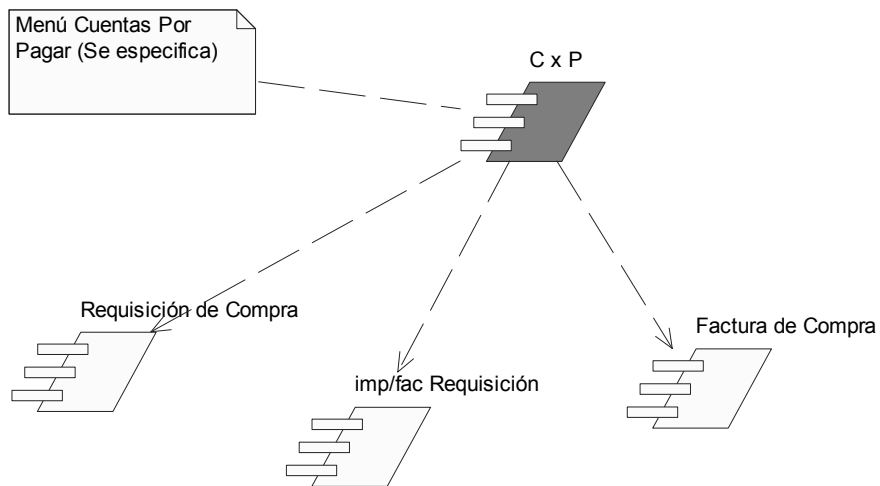
Paquete archivo: Dentro del paquete archivo, encontraremos acceso a la administración de las tablas que alimentarán el sistema



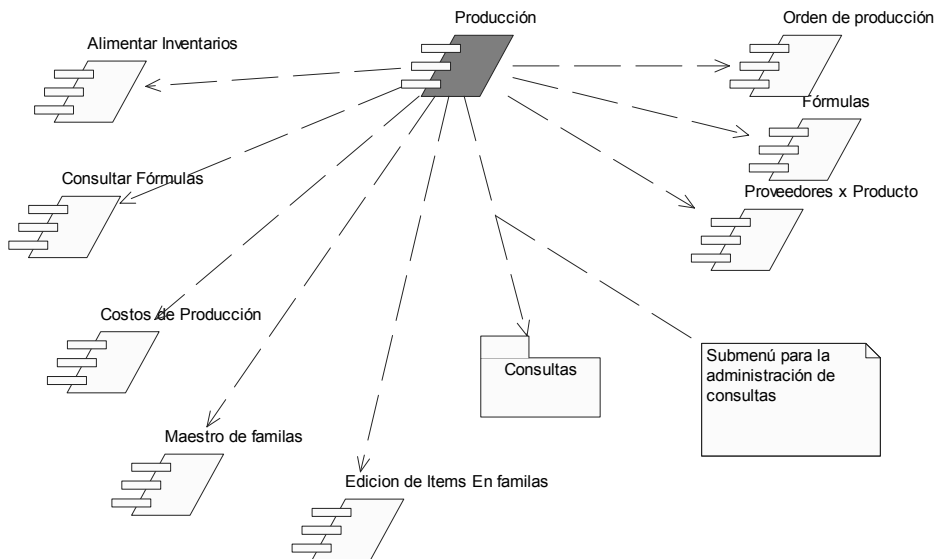
Paquete Ayuda: paquete informativo, es un acerca de, con una breve descripción del producto



Paquete cuentas por pagar(**C x P**): Este paquete maneja lo que son las requisiciones de compra, el manejo de requisiciones de producción, y las facturas de compra.



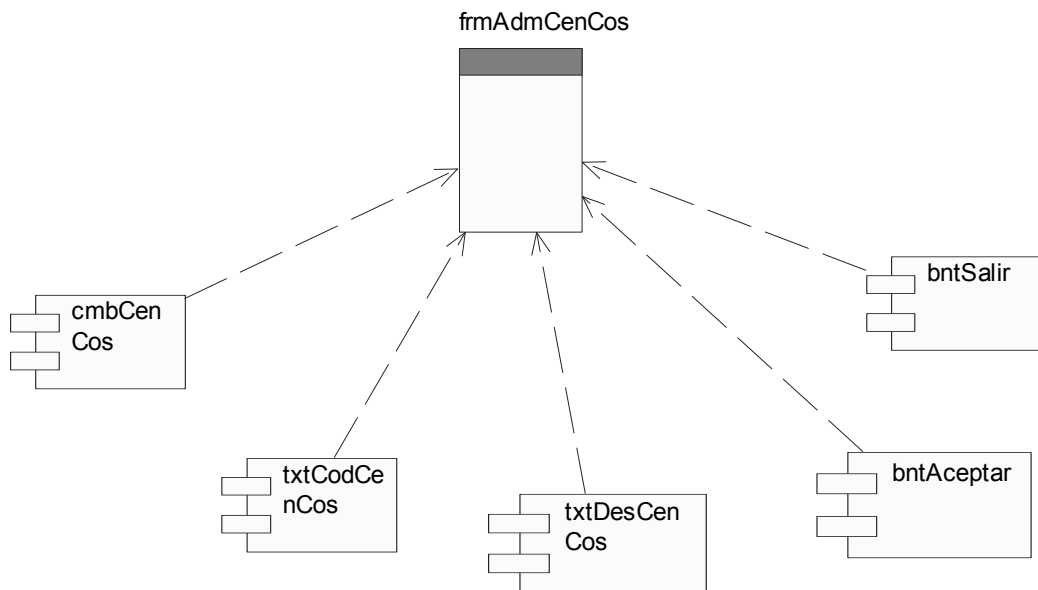
Paquete producción: Administra el sistema para la parte de producción, una de las partes más importantes del sistema, pues acá se podrán configurar todo lo que tenga que ver con inventarios, maestro de operaciones y demás.



Construcción de los formularios de la aplicación.

Diagrama de Implementación Formulario: Administración centro de costos

Diagrama de despliegue



Diseño visual del formulario



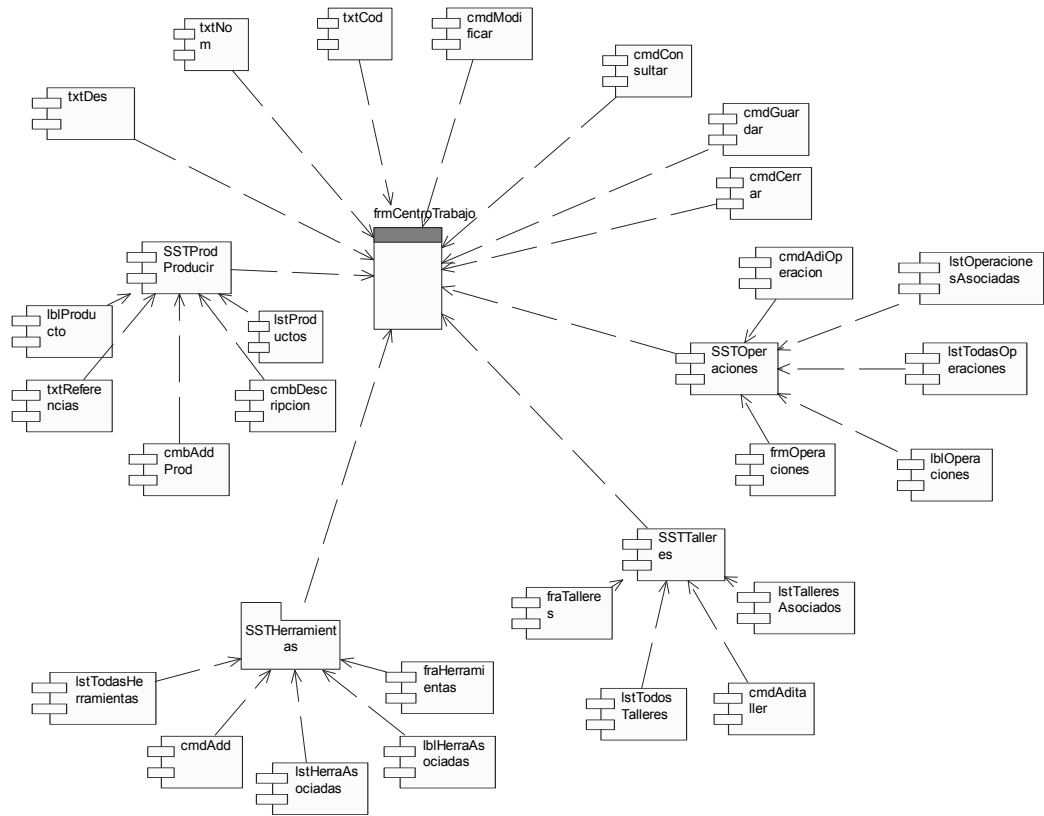
Algoritmos

```
Function Insertar_datos()  
    Dim rs As ADODB.Recordset  
    Set rs = New ADODB.Recordset  
    numero = Trim(txtCodCenCos.Text)  
    nombre = Trim(txtDesCenCos.Text)  
  
    If numero <> "" And nombre <> "" Then  
        sql = "select * from centro_costos where numero ='" &  
numero & "'" &  
        AbrirDB ("M")  
        rs.Open sql, connM, adOpenDynamic, adLockOptimistic  
        If rs.RecordCount > 0 Then  
            MsgBox "el codigo ya existe"  
        Else  
            rs.AddNew  
            rs("numero") = numero  
            rs("nombre") = nombre  
            rs.Update  
            limpiar_forma  
            LeerCentroCostos  
        End If  
        rs.Close  
    Else  
        MsgBox "Faltan datos para la inserción"  
    End If  
End Function
```

```
btnSalir:  
Unload Me
```

Diagrama de Implementación Formulario: administración centros de trabajo

Diagrama de despliegue



Diseño visual del formulario

The image shows a Windows-style application window titled 'Form1' with the main title 'Administración Centros de Trabajo'. The interface includes a 'Código' dropdown menu, 'Nombre' and 'Descripción' text input fields. Below these are three tabs: 'Herramientas', 'Talleres', and 'Operaciones'. The 'Operaciones' tab is selected and contains a sub-section titled 'Productos a producir'. This sub-section has a 'Datos de los Productos' area with 'Referencias' and 'Descripción' fields, a list box, and '>>' and '<<' buttons. Below this is a 'Capacidad por hora de Produccion' section with a 'Capacidad X hora' field. At the bottom, there is a toolbar with icons for undo, save, delete, and help.

Algoritmos

```
Private Sub bntAceptar_Click()  
    If cmbCenCos.ListIndex = 0 Then  
        Insertar_datos  
    End If  
    'If chkCenCos.Enabled = True Then  
    '    Borrar_datos  
    'End If  
End Sub  
Function Borrar_datos()  
    numero = txtCodCenCos.Text  
    If cmbCenCos.Text <> "" And cmbCenCos.Text > 0 Then  
        sql = "delete from centro_costos where numero='" &  
numero & "'" &  
        Set rs = New ADODB.Recordset  
        rs.Open sql, connM, adOpenDynamic, adLockOptimistic
```

```

        limpiar_forma
        LeerCentroCostos
    End If
End Function

Function Insertar_datos()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    numero = Trim(txtCodCenCos.Text)
    nombre = Trim(txtDesCenCos.Text)

    If numero <> "" And nombre <> "" Then
        sql = "select * from centro_costos where numero =" &
numero & ""
        AbrirDB ("M")
        rs.Open sql, connM, adOpenDynamic, adLockOptimistic
        Set rs = New ADODB.Recordset
        rs.Open connM, adOpenStatic, adLockReadOnly
        If rs.RecordCount > 0 Then
            MsgBox "el codigo ya existe"
        Else
            rs.AddNew
            rs("numero") = numero
            rs("nombre") = nombre
            rs.Update
            limpiar_forma
            LeerCentroCostos
        End If
        rs.Close
    Else
        MsgBox "Faltan datos para la inserción"
    End If
End Function

Function limpiar_forma()
    cmbCenCos.Text = ""
    txtCodCenCos.Text = ""
    txtDesCenCos.Text = ""
End Function

Private Sub bntSalir_Click()
    Unload Me
End Sub

Private Sub cmbCenCos_Click()

```

```

        txtCodCenCos.Text = ""
        txtDesCenCos.Text = ""
        If cmbCenCos <> "Nuevo_centro_de_costos" Then
            mostrar_sel
        End If
    End Sub
Function mostrar_sel()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    Dim nomCenCos As String
    nomCenCos = cmbCenCos.Text
    sql = "SELECT * FROM CENTRO_COSTOS where nombre= '" &
nomCenCos & "'"
    AbrirDB ("M")
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    txtCodCenCos.Text = rs("numero")
    txtDesCenCos.Text = rs("NOMBRE")
    rs.Close
End Function

Private Sub Command1_Click()

End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
    Me.Width = 5535
    Me.Height = 4260
    LeerCentroCostos
End Sub
Function LeerCentroCostos()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    cmbCenCos.Clear
    sql = "SELECT * FROM CENTRO_COSTOS ORDER BY nombre"
    AbrirDB ("M")
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    cmbCenCos.Clear
    cmbCenCos.AddItem "Nuevo_centro_de_costos"
    While Not rs.EOF
        cmbCenCos.AddItem rs("nombre")
        rs.MoveNext
    End While
End Function

```

```

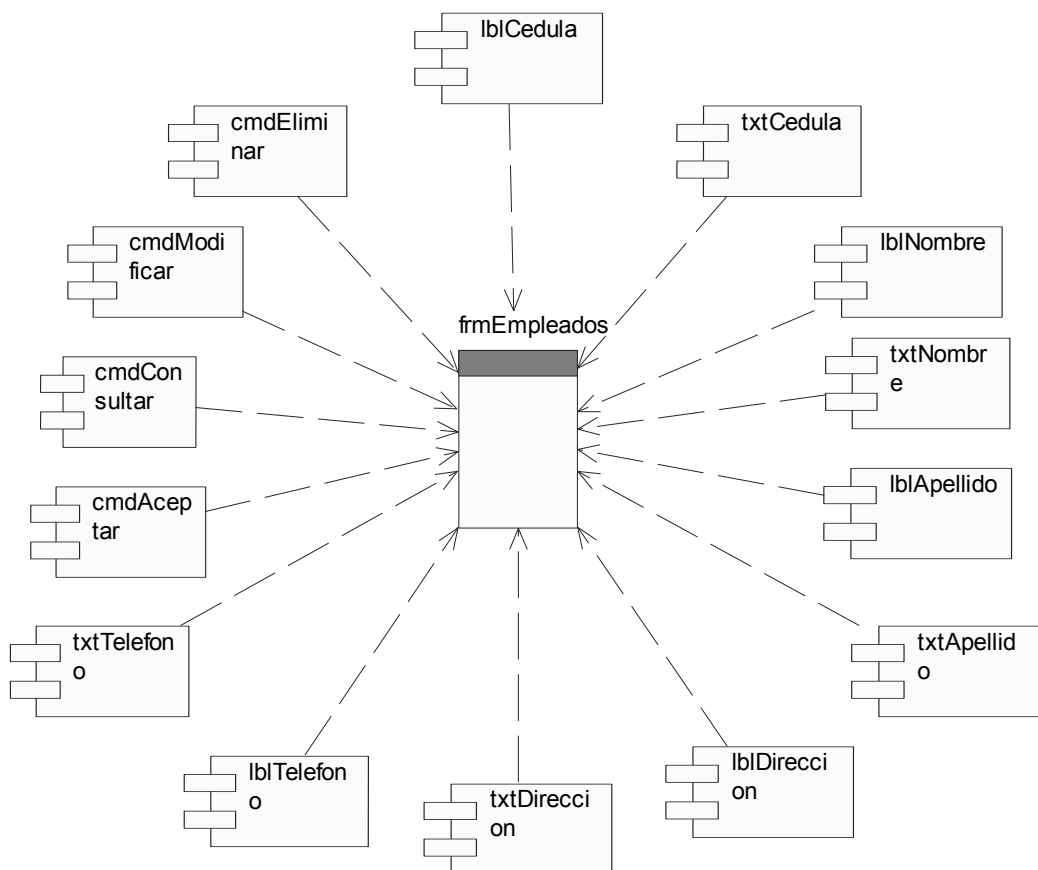
Wend
rs.Close
End Function

Private Sub txtCodCenCos_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
txtDesCenCos.SetFocus
Else
chequeodigitos KeyAscii
End If
End Sub

```

Diagrama de Implementación Formulario: administración de empleados

Diagrama de despliegue



Diseño visual del formulario



The image shows a Windows application window titled "frmEmpleados" with a standard Windows title bar (minimize, maximize, close buttons). The main content area has a title "Administración de Empleados". Below the title, there is a form with the following fields:

- Cédula:** A dropdown menu.
- Nombre:** A text box.
- Apellido:** A text box.
- Dirección:** A text box.
- Telefono:** A text box.

At the bottom of the form, there is a toolbar with five icons: a floppy disk (save), a printer (print), a pair of scissors (delete), a circular arrow (refresh), and a dark octagon (stop).

Algoritmos

```
Private Sub cmdAceptar_Click()  
Dim rs As ADODB.Recordset  
Set rs = New ADODB.Recordset  
AbrirDB ("M")  
sql = "select * from mgr_empleados where cedula='" &  
txtCedula.Text & "'" &  
rs.Open sql, connM, adOpenDynamic, adLockOptimistic  
If rs.RecordCount = 0 Then  
    rs.AddNew  
        rs("cedula") = txtCedula.Text  
        rs("nombre") = txtNombre.Text  
        rs("apellido") = txtApellido.Text  
        rs("direccion") = txtDireccion.Text  
        rs("telefono") = txtTelefono.Text  
    rs.Update  
  
    txtCedula.Text = ""  
    txtNombre.Text = ""  
    txtApellido.Text = ""  
    txtDireccion.Text = ""  
    txtTelefono.Text = ""  
Else
```

```

    MsgBox "La cedula ya se encuentra registrada"
End If
End Sub

```

```

Private Sub cmdConsultar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
AbrirDB ("M")
sql = "select * from mgr_empleados where cedula='" &
txtCedula.Text & "'"
rs.Open sql, connM, adOpenStatic, adLockReadOnly
If rs.RecordCount > 0 Then
    txtCedula.Text = rs("cedula")
    txtNombre.Text = rs("nombre")
    txtApellido.Text = rs("apellido")
    txtDireccion.Text = rs("direccion")
    txtTelefono.Text = rs("telefono")

```

```

Else
    MsgBox "La cedula no se encuentra registrada"
End If

```

```

End Sub

```

```

Private Sub cmdEliminar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
AbrirDB ("M")
sql = "select * from mgr_empleados where cedula='" &
txtCedula.Text & "'"
rs.Open sql, connM, adOpenDynamic, adLockOptimistic
If rs.RecordCount > 0 Then
    rs.Delete
    txtCedula.Text = ""
    txtNombre.Text = ""
    txtApellido.Text = ""
    txtDireccion.Text = ""
    txtTelefono.Text = ""

```

```

Else
    MsgBox "La cedula no se encuentra registrada"
End If

```

```

End Sub

```



```

Private Sub cmdModificar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
AbrirDB ("M")
sql = "select * from mgr_empleados where cedula='" &
txtCedula.Text & "'"
rs.Open sql, connM, adOpenDynamic, adLockOptimistic
If rs.RecordCount > 0 Then
    rs("cedula") = txtCedula.Text
    rs("nombre") = txtNombre.Text
    rs("apellido") = txtApellido.Text
    rs("direccion") = txtDireccion.Text
    rs("telefono") = txtTelefono.Text
    rs.Update

    txtCedula.Text = ""
    txtNombre.Text = ""
    txtApellido.Text = ""
    txtDireccion.Text = ""
    txtTelefono.Text = ""
Else
    MsgBox "La cedula ya se encuentra registrada"
End If
End Sub

Private Sub txtApellido_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtDireccion.SetFocus
Else
    chequeoletras KeyAscii
End If
End Sub

Private Sub txtCedula_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtNombre.SetFocus
Else
    chequeodigitos KeyAscii
End If
End Sub

Private Sub txtDireccion_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    txtTelefono.SetFocus
End If
End Sub

```

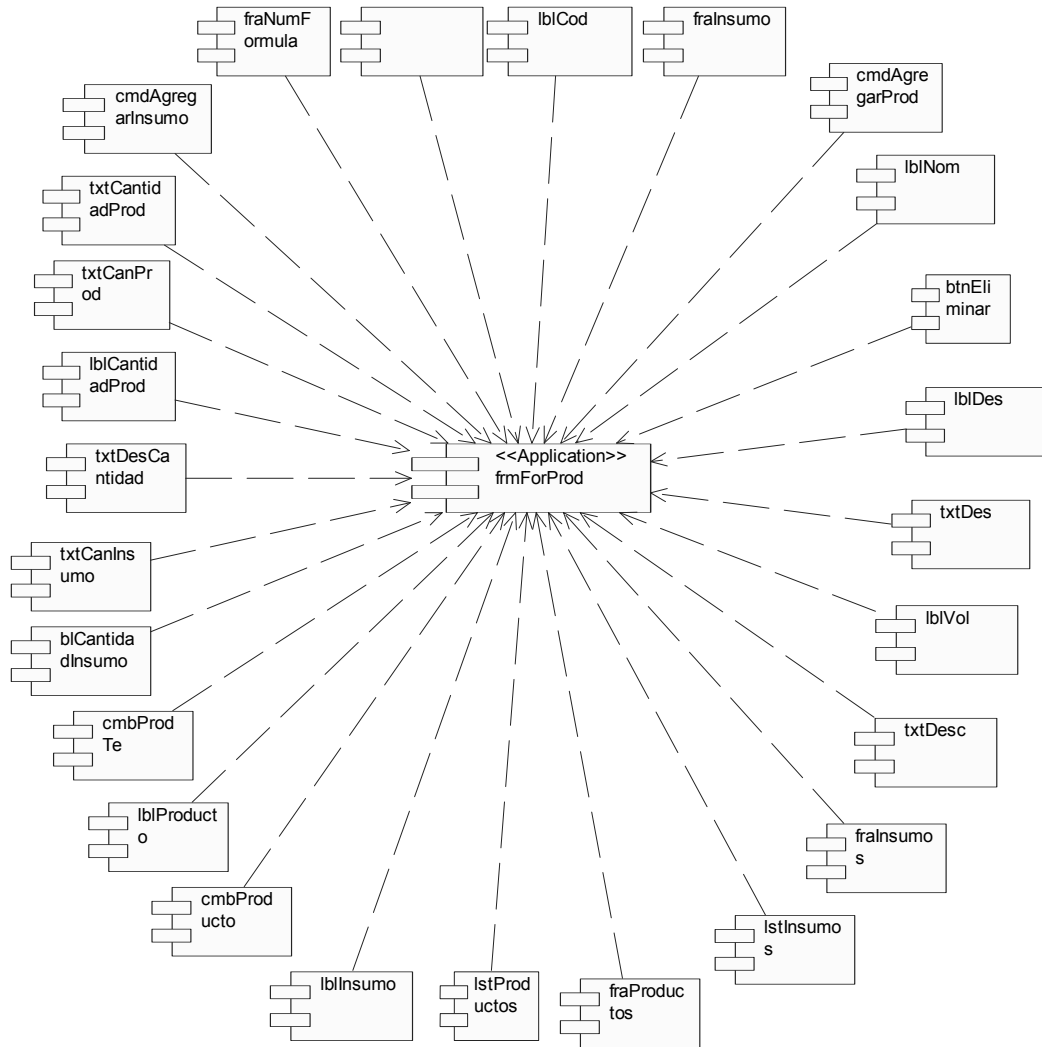
```
End Sub
```

```
Private Sub txtNombre_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    txtApellido.SetFocus  
Else  
    chequeoletras KeyAscii  
End If  
End Sub
```

```
Private Sub txtTelefono_KeyPress(KeyAscii As Integer)  
    chequeodigitos KeyAscii  
End Sub
```

Diagrama de Implementación Formulario: Administración de fórmulas

Diagrama de despliegue



Diseño visual del formulario

Formulario de Administración de Fórmulas. Campos: Numero de la formula, Descripción, Insumos, Productos, Insumo, Producto, Cantidad, Porcentaje de pertenencia.

Algoritmos

Dim ban_nuevo As Boolean

```
Function insertar_insumos_nuevo()  
    Dim rs As ADODB.Recordset  
    Set rs = New ADODB.Recordset  
    Dim rs2 As ADODB.Recordset  
    Set rs2 = New ADODB.Recordset  
    i = 0
```

```

        If ((cmbProducto.Text <> "") And (txtCanInsumo.Text <>
"")) Then
            While (i < lstInsumos.ListCount)
                sql = "select * from mgr_item_produc_formula
where cod_formula='" & txtNumFormula.Text & "'"
                rs.Open sql, connM, adOpenDynamic,
adLockOptimistic
                x = 0
                x = InStr(lstInsumos.List(i), ",")
                lon1 = Len(lstInsumos.List(i))
                insumo = Left(lstInsumos.List(i), x - 1)
                lon = Len(insumo)
                cantidad = Right(lstInsumos.List(i), lon1 - lon -
1)

                sql = "select * from mgr_prod_familia where
descripcion='" & insumo & "'"
                rs2.Open sql, connM, adOpenStatic, adLockReadOnly
                If rs2.RecordCount > 0 Then
                    cod_insumo = rs2("codigo")
                End If
                rs.AddNew
                    rs("cod_formula") = txtNumFormula.Text
                    rs("cod_familia") = cod_insumo
                    rs("uso") = 0
                    rs("cantidad") = cantidad
                    rs("orden") = i + 1
                rs.Update
                rs.Close
                rs2.Close
                i = i + 1
            Wend
        End If
    End Function
Function insertar_producto_nuevo()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    Dim rs2 As ADODB.Recordset
    Set rs2 = New ADODB.Recordset
    ' AbrirDB ("M")
    i = 0

    If ((cmbProdTer.Text <> "") And (txtCanProd.Text <> "")
And (txtPorcentaje.Text <> "")) Then
        While (i < lstProductos.ListCount)

```

```

        sql = "select * from mgr_item_produc_formula
where cod_formula='" & txtNumFormula.Text & "'"
        rs.Open sql, connM, adOpenDynamic,
adLockOptimistic
        x = 0

        x = InStr(lstProductos.List(i), ",")
        lon1 = Len(lstProductos.List(i))
        insumo = Left(lstProductos.List(i), x - 1)
        lon = Len(insumo)

        cadena = Right(lstProductos.List(i), lon1 - lon -
1)
        Y = InStr(cadena, "%")
        cantidad = Left(cadena, Y - 1)
        lon1 = Len(cadena)
        lon = Len(cadena)
        porcen = Right(cadena, lon1 - Y)
        sql = "select * from mgr_prod_familia where
descripcion='" & insumo & "'"
        rs2.Open sql, connM, adOpenStatic, adLockReadOnly
        If rs2.RecordCount > 0 Then
            cod_insumo = rs2("codigo")
        End If
        rs.AddNew
            rs("cod_formula") = txtNumFormula.Text
            rs("cod_familia") = cod_insumo
            rs("uso") = 1
            rs("cantidad") = cantidad
            rs("orden") = i + 1
            rs("PORCENTAJE_PERTENENCIA") = porcen
        rs.Update
        rs.Close
        rs2.Close
        i = i + 1
    Wend
End If
End Function
Private Sub btnAplicar_Click()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    If txtNumFormula.Text <> "" And txtDesc.Text <> "" And
lstInsumos.ListCount > 0 And lstProductos.ListCount > 0 Then

```

```

        SQL_2 = "delete from MGR_PRODUC_FORMULA where
COD_FORMULA='" & cmbCodFor.Text & "'"
        rs.Open sql, connM, adOpenDynamic, adLockOptimistic

        sql = "select * from mgr_produc_formula where
cod_formula='" & txtNumFormula.Text & "'"
        rs.Open sql, connM, adOpenStatic, adLockOptimistic
        If rs.RecordCount = 0 Then
            rs.AddNew
                rs("COD_FORMULA") = txtNumFormula.Text
                rs("DESCRIPCION") = txtDesc.Text
                rs.Update
                btnNuevo.Caption = "Nuevo"
                ban_nuevo = False
                rs.Close
                txtNumFormula.Visible = False
                insertar_insumos_nuevo
                insertar_producto_nuevo

                cmbCodFor.Visible = True
                cmbNomFor.Visible = True
                txtNumFormula.Visible = False
            Else
                rs.Close
                MsgBox "El codigo ya existe"
            End If
        Else
            MsgBox "Faltan datos para la inserción"
        End If

End Sub

```

```

Function limpiar()
    txtNumFormula.Visible = False
    btnNuevo.Enabled = True
    btnEliminar.Enabled = True
    cmbCodFor.Visible = True
    cmbNomFor.Visible = True
    cmbCodFor.Text = ""
    cmbNomFor.Text = ""
    cmbProducto.Text = ""
    cmbProdTer.Text = ""
    txtCanInsumo.Text = ""

```

```

        txtCanProd.Text = ""
        txtPorcentaje.Text = ""
        lstInsumos.Clear
        lstProductos.Clear
        txtDesCantidad.Text = ""
        txtCantidadProd.Text = ""
    End Function

Private Sub btnCancelar_Click()
    txtNumFormula.Visible = False
    cmbCodFor.Visible = True
    cmbNomFor.Visible = True
    ban_nuevo = False
    btnNuevo.Caption = "Nuevo"
    LeerFormulas
    txtDesc.Text = ""
    lstInsumos.Clear
    lstProductos.Clear
End Sub

Private Sub btnEliminar_Click()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    sql = "delete from MGR_PRODUC_FORMULA where
COD_FORMULA='" & cmbCodFor.Text & "'"
    rs.Open sql, connM, adOpenDynamic, adLockOptimistic
    num = cmbCodFor.Text
    i = 0
    While i < cmbCodFor.ListCount
        If cmbCodFor.List(i) = num Then
            pos = i
            cmbCodFor.RemoveItem (pos)
            cmbNomFor.Text = ""
            txtDesc.Text = ""
            lstInsumos.Clear
            lstProductos.Clear
        End If
        i = i + 1
    Wend

    i = 0
    nom = cmbNomFor.Text
    While i < cmbNomFor.ListCount
        If cmbNomFor.List(i) = nom Then
            pos2 = i

```



```

        cmbNomFor.RemoveItem (pos2)
        cmbNomFor.Text = ""
    End If
    i = i + 1
Wend
End Sub

Private Sub btnNuevo_Click()
    If ban_nuevo = False Then
        btnNuevo.Caption = "Guardar"
        cmbCodFor.Visible = False
        cmbNomFor.Visible = False
        cmbNomFor.Text = ""
        txtNumFormula.Visible = True
        txtNumFormula.Text = ""
        cmbProducto.Text = ""
        cmbProdTer.Text = ""
        txtCanInsumo.Text = ""
        txtDesCantidad.Text = ""
        txtCanProd.Text = ""
        txtCantidadProd.Text = ""
        txtPorcentaje.Text = ""
        lstInsumos.Clear
        lstProductos.Clear
        txtDesc.Text = ""
        ban_nuevo = True
    Else
        Dim rs As ADODB.Recordset
        Set rs = New ADODB.Recordset
        If txtNumFormula.Text <> "" And txtDesc.Text <> ""
And lstInsumos.ListCount > 0 And lstProductos.ListCount > 0
Then
            sql = "select * from mgr_produc_formula where
cod_formula='" & txtNumFormula.Text & "'"
            rs.Open sql, connM, adOpenStatic,
adLockOptimistic
            If rs.RecordCount = 0 Then
                rs.AddNew
                rs("COD_FORMULA") = txtNumFormula.Text
                rs("DESCRIPCION") = txtDesc.Text
                rs.Update
                btnNuevo.Caption = "Nuevo"
                ban_nuevo = False
                rs.Close
                txtNumFormula.Visible = False
            End If
        End If
    End If
End Sub

```

```

        insertar_insumos_nuevo
        insertar_producto_nuevo

        cmbCodFor.Visible = True
        cmbNomFor.Visible = True
        txtNumFormula.Visible = False
        LeerFormulas
    Else
        MsgBox "El codigo ya existe"
        rs.Close
    End If
Else
    MsgBox "Faltan datos para la inserción"
End If

End If
End Sub

Private Sub btnSalir_Click()
    Unload Me
End Sub

Private Sub cmbCodFor_Click()
    Dim codFor As String

    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset

    Dim rs2 As ADODB.Recordset
    Set rs2 = New ADODB.Recordset
    lstInsumos.Clear
    lstProductos.Clear
    codFor = cmbCodFor.Text
    sql = "select descripcion from MGR_PRODUC_FORMULA where
cod_formula='" & codFor & "'"
    AbrirDB ("M")
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    cmbNomFor.Text = rs("descripcion")
    txtDesc.Text = rs("descripcion")
    'lstProductos.AddItem (txtDesc.Text)

    While Not rs.EOF
        cmbNomFor.AddItem rs("descripcion")
        rs.MoveNext
    Wend

```

```

rs.Close
'buscar_productos (codFor)
buscar_insumos (codFor)

End Sub
Function buscar_productos(codigo As String)
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
sql = "select descripcion from MGR_PROD_FAMILIA where
codigo in (select COD_Familia from mgr_item_produc_formula
where COD_FORMULA= '" & Nivel & "')"
rs.Open sql, connM, adOpenStatic, adLockReadOnly
lstInsumos.Clear
If rs.RecordCount > 0 Then
While Not rs.EOF
lstProductos.AddItem rs("Descripcion")
rs.MoveNext
Wend
rs.Close
End If
End Function
Private Sub cmbNomFor_Click()
Dim nomfor As String
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
nomfor = cmbNomFor.Text
sql = "select * from MGR_PRODUC_FORMULA where
descripcion='" & nomfor & "'"
AbrirDB ("M")
rs.Open sql, connM, adOpenStatic, adLockReadOnly
cmbCodFor.Text = rs("COD_FORMULA")
txtDesc.Text = rs("DESCRIPCION")
buscar_insumos (rs("cod_formula"))

While Not rs.EOF
cmbCodFor.AddItem rs("cod_formula")
rs.MoveNext
Wend
rs.Close

End Sub

Function buscar_insumos(Nivel As String)
Dim rs As ADODB.Recordset

```

```

Set rs = New ADODB.Recordset
Dim rs2 As ADODB.Recordset
Set rs2 = New ADODB.Recordset
lstInsumos.Clear
lstProductos.Clear

sql = "SELECT MGR_PROD_FAMILIA.DESCRIPCION,
MGR_ITEM_PRODUC_FORMULA.CANTIDAD, "
sql = sql & "MGR_ITEM_PRODUC_FORMULA.COD_FORMULA, "
sql = sql &
"MGR_ITEM_PRODUC_FORMULA.PORCENTAJE_PERTENENCIA, "
sql = sql & "MGR_ITEM_PRODUC_FORMULA.USO "
sql = sql & "FROM MGR_PROD_FAMILIA INNER JOIN
MGR_ITEM_PRODUC_FORMULA ON "
sql = sql & "MGR_PROD_FAMILIA.CODIGO =
MGR_ITEM_PRODUC_FORMULA.COD_FAMILIA "
sql = sql & "WHERE MGR_ITEM_PRODUC_FORMULA.COD_FORMULA=" &
& Nivel & "'"

rs.Open sql, connM, adOpenStatic, adLockReadOnly
While Not rs.EOF
    If (rs("uso") = 0) Then
        lstInsumos.AddItem rs("descripcion") & "," &
rs("cantidad")
    End If
    If (rs("uso") = 1) Then
        lstProductos.AddItem rs("descripcion") & "," &
rs("cantidad") & "%" & rs("PORCENTAJE_PERTENENCIA")
    End If
    rs.MoveNext
Wend
rs.Close
End Function

Private Sub Command4_Click()
    Unload Me
End Sub

Private Sub cmbProdTer_Click()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    sql = "select descripcion from mgr_prod_unidad where
codigo in (select unidad from mgr_prod_familia where
descripcion='" & cmbProdTer.Text & "'"")
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    txtCantidadProd.Text = rs("descripcion")

```

```

End Sub

Private Sub cmbProducto_Click()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    sql = "select descripcion from mgr_prod_unidad where
codigo in (select unidad from mgr_prod_familia where
descripcion='" & cmbProducto.Text & "')"
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    txtDesCantidad.Text = rs("descripcion")

End Sub

Private Sub Command1_Click()

End Sub

Private Sub Command5_Click()
End Sub

Private Sub cmdAgregarInsumo_Click()
    band = False
    i = 0
    If ((cmbProducto.Text <> "") And (txtCanInsumo.Text <>
"")) Then
        While (i < lstInsumos.ListCount)

            x = InStr(lstInsumos.List(i), ",")
            z = Left(lstInsumos.List(i), x - 1)
            If (cmbProducto.Text = z) Then
                band = True
            End If
            i = i + 1
        Wend
        If (band = False) Then
            lstInsumos.AddItem (cmbProducto.Text) & "," &
(txtCanInsumo.Text)
            i = 0
        Else
            MsgBox ("Este insumo ya se encuentra en la lista
de insumos")
        End If
    Else
        MsgBox ("Faltan datos en cantidad")
    End If
End Sub

```

```

        lstInsumos.Refresh

End Sub

Private Sub Command6_Click()
End Sub

Private Sub cmdAgregarProd_Click()
    band = False
    i = 0
    If ((txtCanProd.Text <> "") And (txtPorcentaje.Text <>
"")) Then
        While (i < lstProductos.ListCount)
            x = InStr(lstProductos.List(i), ",")
            z = Left(lstProductos.List(i), x - 1)
            If (cmbProdTer.Text = z) Then
                band = True
            End If
            i = i + 1
        Wend
        If (band = False) Then
            lstProductos.AddItem (cmbProdTer.Text) & "," &
(txtCanProd.Text) & "%" & (txtPorcentaje.Text)
            i = 0
        Else
            MsgBox ("Este producto ya se encuentra en la
lista de productos")
        End If
    Else
        MsgBox ("DIGITE PORCENTAJE O CANTIDAD")
    End If

End Sub

Private Sub Form_Load()
    Me.Width = 8070
    Me.Height = 8190
    ban_nuevo = False
    LeerFormulas
    cargarProductos
    cargarProdTer
End Sub
Function cargarProdTer()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset

```

```

        cmbProdTer.Clear
        sql = "SELECT * FROM MGR_PROD_FAMILIA where nivel=3 and
clasificacion=3 or clasificacion=2 ORDER BY descripcion"
        AbrirDB ("M")
        rs.Open sql, connM, adOpenStatic, adLockReadOnly
        While Not rs.EOF
            cmbProdTer.AddItem rs("descripcion")
            rs.MoveNext
        Wend
        rs.Close
End Function
Function LeerFormulas()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    cmbCodFor.Clear
    sql = "SELECT * FROM MGR_PRODUC_FORMULA ORDER BY
cod_formula"
    AbrirDB ("M")
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    While Not rs.EOF
        cmbCodFor.AddItem rs("COD_FORMULA")
        rs.MoveNext
    Wend
    rs.Close

    cmbNomFor.Clear
    sql = "select * from MGR_PRODUC_FORMULA order by
DESCRIPCION"
    AbrirDB ("M")
    rs.Open sql, connM, adOpenStatic, adLockReadOnly
    While Not rs.EOF
        cmbNomFor.AddItem rs("DESCRIPCION")
        rs.MoveNext
    Wend
    rs.Close
End Function

Function cargarProductos()
    Dim nomProd As String
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset

    sql = "select descripcion from MGR_PROD_FAMILIA WHERE
NIVEL =3 AND CLASIFICACION=1 order by DESCRIPCION"
    AbrirDB ("M")

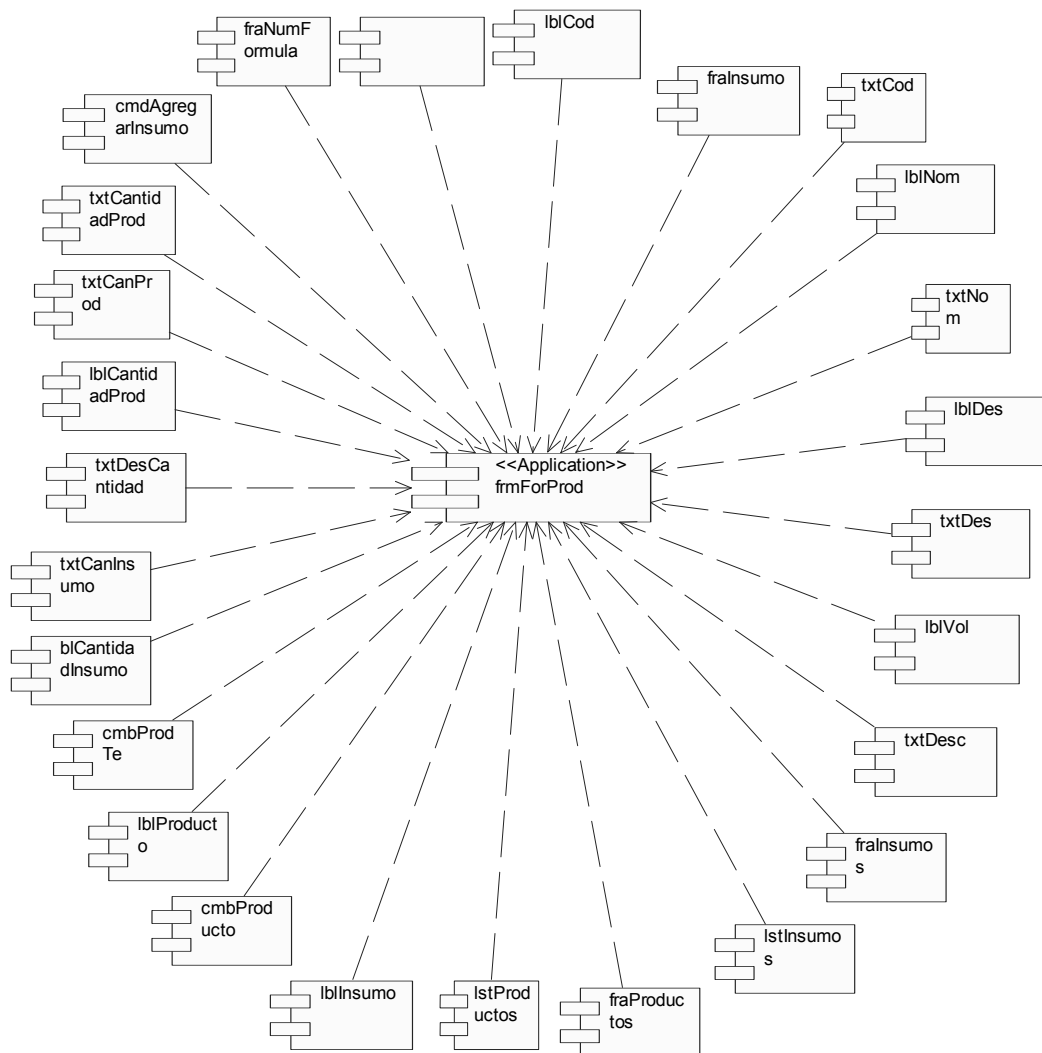
```

```
cmbProducto.Clear
rs.Open sql, connM, adOpenStatic, adLockReadOnly
While Not rs.EOF
    cmbProducto.AddItem rs("descripcion")
    rs.MoveNext
Wend
rs.Close
End Function
```

```
Private Sub lstInsumos_DblClick()
    'pos = lstInsumos.
    'lstInsumos.RemoveItem (pos)
End Sub
```


Diagrama de Implementación Formulario: Administración de herramientas

Diagrama de despliegues



Diseño visual del formulario

Algoritmos

```
Private Sub Command3_Click()  
End Sub
```

```
Private Sub Command4_Click()  
Dim rs As ADODB.Recordset  
Set rs = New ADODB.Recordset  
AbrirDB ("M")  
If txtCod.Text <> "" Then  
    sql = "select * from mgr_herramientas where codigo='" &  
txtCod.Text & "'" &  
    rs.Open sql, connM, adOpenDynamic, adLockOptimistic  
    If rs.RecordCount > 0 Then  
        rs.Delete  
        txtCod.Text = ""  
        Text2.Text = ""  
        Text3.Text = ""  
        Text4.Text = ""
```

```

        Text5.Text = ""
        Text6.Text = ""
        Text7.Text = ""
    Else
        MsgBox "el codigo digitado no existe"
    End If
Else
    MsgBox "Falta el codigo de la herramienta"
End If
rs.Close
End Sub

Private Sub cmdAceptar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Dim rs2 As ADODB.Recordset
Set rs2 = New ADODB.Recordset

AbrirDB ("M")

If txtCod.Text <> "" And txtNom.Text <> "" And cmbTaller.Text
<> "" Then
    sql = "select * from mgr_herramientas where codigo='" &
txtCod.Text & "'"
    rs.Open sql, connM, adOpenDynamic, adLockOptimistic
    If rs.RecordCount = 0 Then
        rs.AddNew
            rs("codigo") = txtCod.Text
            rs("nombre") = txtNom.Text
            rs("descripcion") = txtDes.Text
            rs("voltaje") = txtVol.Text
            rs("material") = txtMat.Text
            rs("costo") = txtCos.Text
            rs("factura") = txtFac.Text
        SQL2 = "select * from mgr_taller where
descripcion='" & cmbTaller.Text & "'"
        rs2.Open SQL2, connM, adOpenStatic,
adLockOptimistic
        If rs2.RecordCount > 0 Then
            rs("cod_taller") = rs2("codigo")
        End If
        rs2.Close

        SQL2 = "select * from mgr_tipo_herramienta where
nombre='" & cmbTipoHerramienta.Text & "'"

```

```

        rs2.Open SQL2, connM, adOpenStatic,
adLockOptimistic
        If rs2.RecordCount > 0 Then
            rs("cod_tipo_herramienta") = rs2("codigo")
        End If
        rs2.Close
    rs.Update
    txtCod.Text = ""
    txtNom.Text = ""
    txtDes.Text = ""
    txtVol.Text = ""
    txtMat.Text = ""
    txtCos.Text = ""
    txtFac.Text = ""
    cmbTaller.Text = ""
    cmbTipoHerramienta = ""
Else
    MsgBox "el codigo digitado ya existe"
End If
rs.Close
Else
    MsgBox "Digte el codigo y el nombre de la herramienta"
End If
End Sub

Private Sub cmdConsultar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Dim rs2 As ADODB.Recordset
Set rs2 = New ADODB.Recordset

AbrirDB ("M")
If txtCod.Text <> "" Then
    sql = "select * from mgr_herramientas where codigo='" &
txtCod.Text & "'"
    rs.Open sql, connM, adOpenStatic, adLockOptimistic
    If rs.RecordCount > 0 Then
        txtCod.Text = rs("codigo")
        txtNom.Text = rs("nombre")
        txtDes.Text = rs("descripcion")
        If rs("voltaje") <> "" Then
            txtVol.Text = rs("voltaje")
        End If
        If rs("material") <> "" Then
            txtMat.Text = rs("material")
        End If
    End If
End Sub

```

```

End If

If rs("costo") <> "" Then
    txtCos.Text = rs("costo")
End If
If rs("factura") <> "" Then
    txtFac.Text = rs("factura")
End If
SQL2 = "select * from mgr_taller where codigo='" &
rs("cod_taller") & "'"
rs2.Open SQL2, connM, adOpenStatic, adLockOptimistic
If rs2.RecordCount > 0 Then
    cmbTaller.Text = rs2("descripcion")
End If
rs2.Close

SQL2 = "select * from mgr_tipo_herramienta where
codigo='" & rs("cod_tipo_herramienta") & "'"
rs2.Open SQL2, connM, adOpenStatic, adLockOptimistic
If rs2.RecordCount > 0 Then
    cmbTipoHerramienta.Text = rs2("nombre")
End If
rs2.Close
Else
    MsgBox "el codigo digitado no existe"
End If
Else
    MsgBox "Digte un codigo"
End If
End Sub

Private Sub cmdEliminar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
sql = "select * from mgr_herramientas where codigo='" &
txtCod.Text & "'"
rs.Open sql, connM, adOpenDynamic, adLockOptimistic
If rs.RecordCount > 0 Then
    rs.Delete
    txtCod.Text = ""
    txtNom.Text = ""
    txtDes.Text = ""
    txtVol.Text = ""
    txtMat.Text = ""
    txtCos.Text = ""

```

```

        txtFac.Text = ""
        cmbTaller.Text = ""
        cmbTipoHerramienta = ""

Else
    MsgBox "_El codigo no se encuentra registrado"
End If
rs.Close

End Sub

Private Sub cmdModificar_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Dim rs2 As ADODB.Recordset
Set rs2 = New ADODB.Recordset

AbrirDB ("M")
If txtCod.Text <> "" And txtNom.Text <> "" Then
    sql = "select * from mgr_herramientas where codigo='" &
txtCod.Text & "'"
    rs.Open sql, connM, adOpenDynamic, adLockOptimistic
    If rs.RecordCount > 0 Then
        rs("codigo") = txtCod.Text
        rs("nombre") = txtNom.Text
        rs("descripcion") = txtDes.Text
        rs("voltaje") = txtVol.Text
        rs("material") = txtMat.Text
        rs("costo") = txtCos.Text
        rs("factura") = txtFac.Text
        SQL2 = "select * from mgr_taller where descripcion='"
& cmbTaller.Text & "'"
        rs2.Open SQL2, connM, adOpenStatic, adLockOptimistic
        If rs2.RecordCount > 0 Then
            rs("cod_taller") = rs2("codigo")
        End If
        rs2.Close
        SQL2 = "select * from mgr_tipo_herramienta where
descripcion='" & cmbTipoHerramienta.Text & "'"
        rs2.Open SQL2, connM, adOpenStatic, adLockOptimistic
        If rs2.RecordCount > 0 Then
            rs("cod_tipo_herramienta") = rs2("codigo")
        End If
        rs2.Close
        rs.Update
    End If
End If

```

```

        txtCod.Text = ""
        txtNom.Text = ""
        txtDes.Text = ""
        txtVol.Text = ""
        txtMat.Text = ""
        txtCos.Text = ""
        txtFac.Text = ""
        cmbTaller.Text = ""
        cmbTipoHerramienta = ""
    Else
        MsgBox "el codigo digitado no existe"
    End If
    rs.Close
Else
    MsgBox "Digte el codigo y el nombre de la herramienta"
End If
End Sub

Private Sub Form_Load()
    Me.Width = 7905
    Me.Height = 5130
    cargar_taller
    cargar_tipo_herramientas
End Sub
Function cargar_tipo_herramientas()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
AbrirDB ("M")
sql = "select * from mgr_tipo_herramienta"
rs.Open sql, connM, adOpenStatic, adLockOptimistic
cmbTipoHerramienta.Clear
If rs.RecordCount > 0 Then
    While Not rs.EOF
        cmbTipoHerramienta.AddItem rs("nombre")
        rs.MoveNext
    Wend
Else
    MsgBox "No existen tipos de herramientas definidos"
End If
rs.Close

End Function
Function cargar_taller()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset

```

```

AbrirDB ("M")
sql = "select * from mgr_taller"
rs.Open sql, connM, adOpenStatic, adLockOptimistic
cmbTaller.Clear
If rs.RecordCount > 0 Then
    While Not rs.EOF
        cmbTaller.AddItem rs("descripcion")
        rs.MoveNext
    Wend
Else
    MsgBox "No existen talleres definidos"
End If
rs.Close
End Function

```

```

Private Sub txtCod_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtNom.SetFocus
    Else
        chequeodigitos KeyAscii
    End If

```

End Sub

```

Private Sub txtCos_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtFac.SetFocus
    Else
        chequeodigitos KeyAscii
    End If

```

End Sub

```

Private Sub txtDes_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtVol.SetFocus
    Else
        chequeoletras KeyAscii
    End If

```

End Sub

```

Private Sub txtFac_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then

```



```

        txtDes.SetFocus
    Else
        chequeodigitos KeyAscii
    End If

End Sub

Private Sub txtMat_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtCos.SetFocus
    Else
        chequeoletras KeyAscii
    End If

End Sub

Private Sub txtNom_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtDes.SetFocus
    Else
        chequeoletras KeyAscii
    End If

End Sub

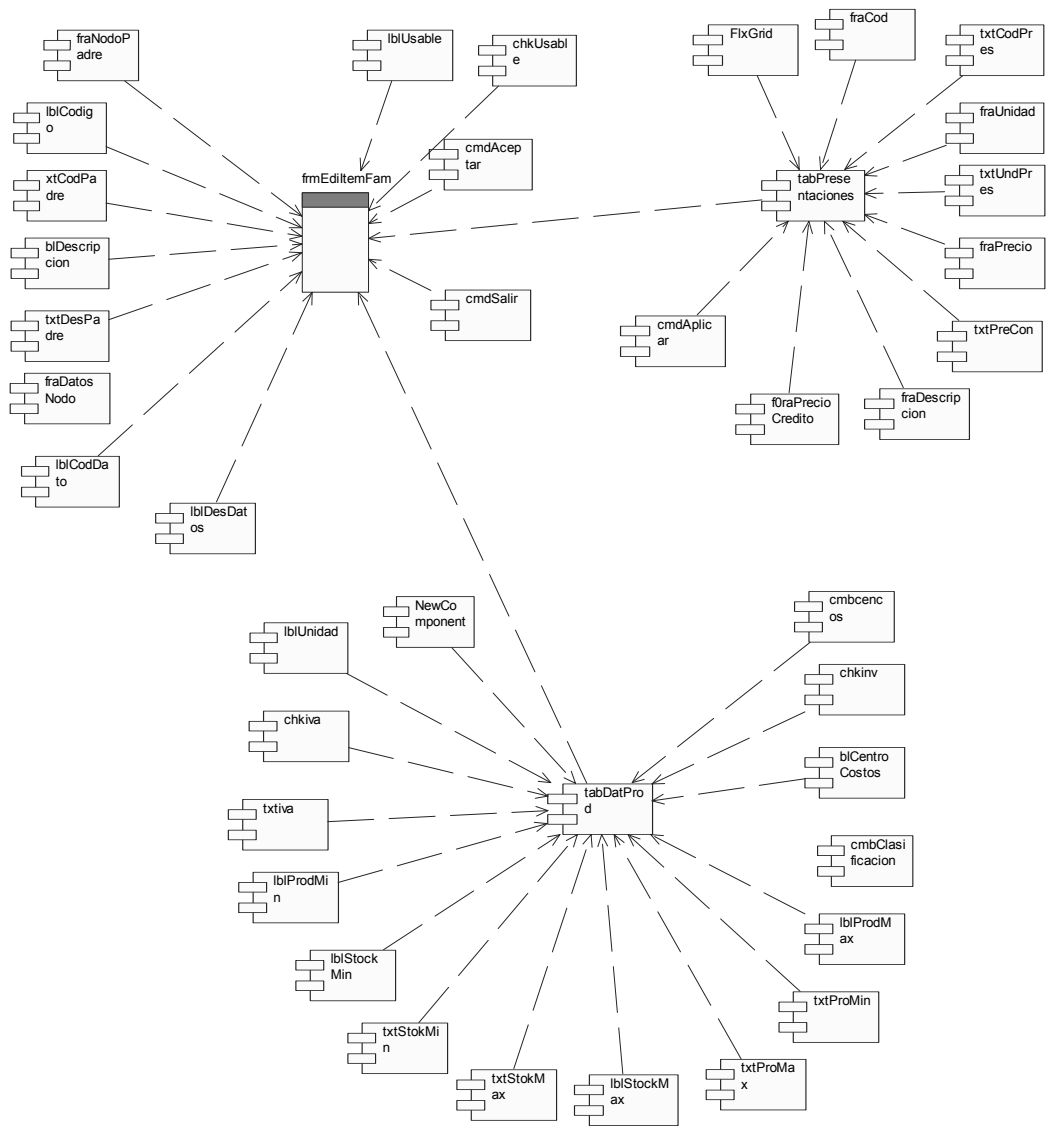
Private Sub txtVol_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtMat.SetFocus
    Else
        chequeodigitos KeyAscii
    End If

End Sub

```

Diagrama de Implementación Formulario: Administración de familias por productos

Diagrama de despliegues



Diseño visual del formulario

EDICION DE ITEMS A UNA FAMILIA

Administrar Familias en Productos

NODO PADRE

CODIGO DESCRIPCION

DATOS DEL NODO

CODIGO DESCRIPCION

Usable

Datos del producto		Presentaciones	
UNIDAD	CLASIFICACION		
<input type="checkbox"/> IVA	Centro de costos		
<input type="checkbox"/> INVENTARIO			
PRODUCCION MINIMA	PRODUCCION MAXIMA	STOK MIN.	STOK MAX.

Save Cancel

Algoritmos

```
Option Explicit
Private SQLS As String
Private RsTempf As Recordset

Private Sub FAMILIAS_DblClick()
    If optNuevo.Value Then
        MsgBox "nuevo"
    Else
        If FAMILIAS.SelectedItem.Text <> "" Then
            If FAMILIAS.SelectedItem.Key <> "P0" Then
                gDesc = Trim(Left(FAMILIAS.SelectedItem.Text,
                InStr(FAMILIAS.SelectedItem.Text, "(") - 1))
                frmEdiItemFam.txtDes =
                Trim(Left(FAMILIAS.SelectedItem.Text,
                InStr(FAMILIAS.SelectedItem.Text, "(") - 1))
                Unload Me
            End If
        End If
    End If
End Sub

Private Sub Form_Load()
    Me.Width = 5850
    Me.Height = 7695
    'FrmAdmConfInt.seleccion = ""
    cargar_arbol ("0")
End Sub

Private Sub salir_Click()
    'FrmAdmConfInt.seleccion = ""
    Unload Me
End Sub

Public Sub cargar_arbol(cod_raiz As String)
    FAMILIAS.Nodes.Clear
    Dim RsTempo As ADODB.Recordset
    Set RsTempo = New ADODB.Recordset
    Dim RsTempf As ADODB.Recordset
    Set RsTempf = New ADODB.Recordset

    AbrirDB ("M")
```

```

        SQLS = "Select * from mgr_prod_familia where nodo_padre =
'" & cod_raiz & "'"
        RsTempf.Open SQLS, connM, adOpenStatic, adLockReadOnly
        If RsTempf.RecordCount > 0 Then
            'agregue el nodo padre
            SQLS = "Select * from mgr_prod_familia where codigo =
'" & cod_raiz & "'"
            RsTempo.Open SQLS, connM, adOpenStatic, adLockReadOnly
            If RsTempo.RecordCount > 0 Then
                FAMILIAS.Nodes.Add , , "P" & RsTempo("codigo"),
RsTempo("descripcion") & " (" & RsTempo("codigo") & ")"
                While Not RsTempf.EOF
                    'inserte priemro el nodo raiz

                    agregar_subnodo RsTempf("CODIGO"),
RsTempf("DESCRIPCION"), "P" & RsTempo("codigo")
                    RsTempf.MoveNext
                Wend
                FAMILIAS.Nodes("P" & RsTempo("codigo")).Expanded =
True
            End If
            RsTempo.Close
        End If
        RsTempf.Close
        connM.Close
    End Sub

```

```

Private Sub agregar_subnodo(cod_fam As String, Desc As
String, Padre As String)
    Dim RsAdd As ADODB.Recordset
    Set RsAdd = New ADODB.Recordset
    AbrirDB ("M")
    FAMILIAS.Nodes.Add Padre, tvwChild, "P" & cod_fam, Desc &
" (" & cod_fam & ")"

    SQLS = "Select * from mgr_prod_familia where nodo_padre =
'" & cod_fam & "'"
    RsAdd.Open SQLS, connM, adOpenStatic, adLockReadOnly
    If RsAdd.RecordCount > 0 Then
        While Not RsAdd.EOF
            agregar_subnodo RsAdd("CODIGO"),
RsAdd("DESCRIPCIÓN"), "P" & cod_fam
            RsAdd.MoveNext
        Wend
    End If

```

```

RsAdd.Close
'connM.Close
End Sub

Private Sub MOSTRARPO()
End Sub

```

Diagrama de Implementación Formulario: Administración de cierre de producción

Diagrama de despliegue

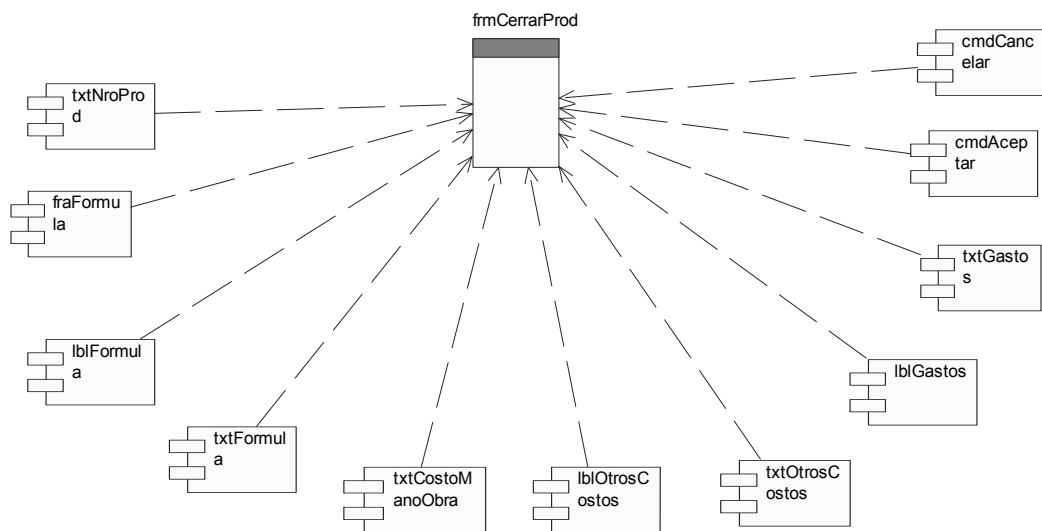
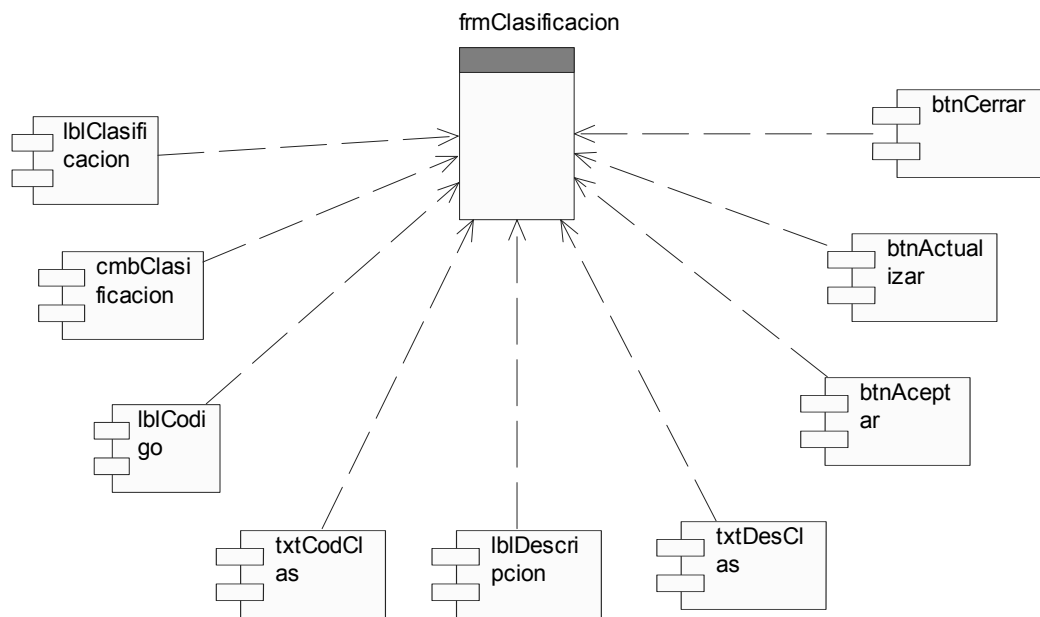


Diagrama de Implementación Formulario: Administración de clasificación de productos

Diagrama de despliegue



Diseño visual del formulario



Algoritmos

```
Private Sub btnAceptar_Click()  
    If cmbClasificacion.ListIndex = 0 Then  
        Insertar_datos  
    End If  
    '    If chkClas.Value = True Then  
    '        Borrar_datos  
    '    End If  
End Sub
```

```
Private Sub btnActualizar_Click()  
    Dim rs As ADODB.Recordset  
    Set rs = New ADODB.Recordset  
    Dim codigo As String  
    Dim descripcion As String  
  
    codigo = Trim(txtCodClas.Text)  
    descripcion = Trim(txtDesClas.Text)  
  
    If codigo <> "" And descripcion <> "" Then  
        sql = "select * FROM MGR_PROD_CLASIFICACION where  
codigo =" & codigo & ""  
        AbrirDB ("M")  
        rs.Open sql, connM, adOpenDynamic, adLockOptimistic  
        If rs.RecordCount > 0 Then  
            rs("codigo") = codigo  
            rs("descripcion") = descripcion  
            rs.Update  
            limpiar_forma  
            LeerClasificaciones  
            rs.Close  
        Else  
            MsgBox "DIGITE EL CODIGO Y LA DESCRIPCION"  
        End If  
    End If  
End Sub
```

```
Private Sub cmbClasificacion_Click()  
    txtCodClas.Text = ""  
    txtDesClas.Text = ""  
  
    If txtCodClas.Locked = True Then  
        txtCodClas.Locked = False  
    End If
```



```

        If cmbClasificacion <> "NUEVA_CLASIFICACION" Then
            mostrar_sel
        End If
    End Sub
Function mostrar_sel()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Dim desClas As String
desClas = cmbClasificacion.Text
sql = "SELECT * FROM MGR_PROD_CLASIFICACION where
descripcion= '" & desClas & "'"
AbrirDB ("M")
rs.Open sql, connM, adOpenStatic, adLockReadOnly
txtCodClas.Text = rs("codigo")
txtDesClas.Text = rs("descripcion")
rs.Close
End Function

Private Sub Command2_Click()
Unload Me
End Sub
Function Borrar_datos()
Dim rs As Recordset
Set rs = New ADODB.Recordset
Dim codigo As String
Dim descripcion As String
descripcion = cmbClasificacion.Text

If cmbClasificacion.Text <> "" And txtCodClas.Text > 0
Then
AbrirDB ("M")

sql = "delete from MGR_PROD_CLASIFICACION where
descripcion ='" & descripcion & "'"
Set rs = New ADODB.Recordset
rs.Open sql, connM, adOpenDynamic, adLockOptimistic
MsgBox "Datos borrados"
limpiar_forma
LeerClasificaciones
End If

End Function

```

```

Function Insertar_datos()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset
    Dim codigo As String
    Dim descripcion As String

    codigo = Trim(txtCodClas.Text)
    descripcion = Trim(txtDesClas.Text)

    If codigo <> "" And descripcion <> "" Then
        sql = "select * from MGR_PROD_CLASIFICACION where
codigo =" & codigo & ""
        AbrirDB ("M")
        rs.Open sql, connM, adOpenDynamic, adLockOptimistic
        If rs.RecordCount > 0 Then
            MsgBox "el codigo ya existe"
        Else
            rs.AddNew
            rs("codigo") = codigo
            rs("descripcion") = descripcion
            rs.Update
            limpiar_forma
            LeerClasificaciones
        End If
        rs.Close
    Else
        MsgBox "DIGITE EL CODIGO Y LA DESCRIPCION"
    End If
End Function

Function limpiar_forma()
cmbClasificacion.Text = ""
txtCodClas.Text = ""
txtDesClas.Text = ""
End Function

Private Sub Form_Load()
    LeerClasificaciones
    Me.Width = 4875
    Me.Height = 4425
End Sub

Function LeerClasificaciones()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset

```

```

        cmbClasificacion.Clear
        sql = "SELECT * FROM MGR_PROD_CLASIFICACION ORDER BY
DESCRIPCION"
        AbrirDB ("M")
        rs.Open sql, connM, adOpenStatic, adLockReadOnly
        cmbClasificacion.Clear
        cmbClasificacion.AddItem "NUEVA_CLASIFICACION"
        While Not rs.EOF
            cmbClasificacion.AddItem rs("descripcion")
            rs.MoveNext
        Wend
        rs.Close
    End Function

Private Sub txtCodClas_Change()

End Sub

Private Sub txtCodClas_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        txtDesClas.SetFocus
    Else
        chequeodigitos KeyAscii
    End If
End Sub

```

Diagrama de Implementación Formulario: consulta de fórmulas
Diagrama de despliegue

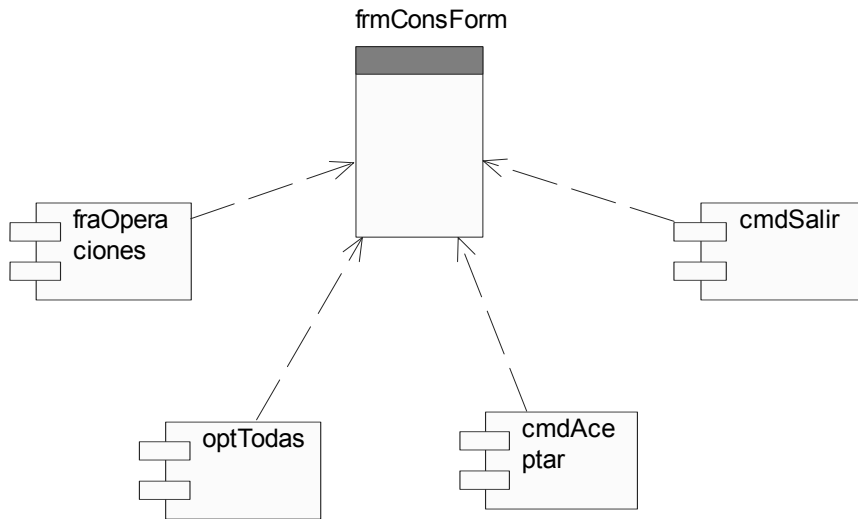


Diagrama de Implementación Formulario: Consulta de proveedores
Diagrama de despliegue

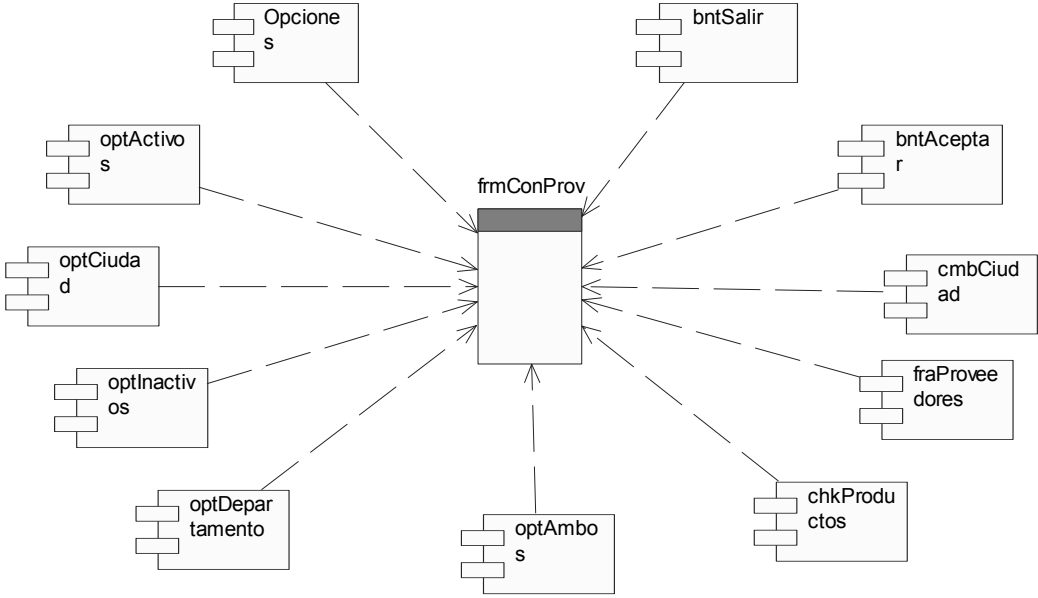


Diagrama de Implementación Formulario: Costos de producción

Diagrama de despliegue

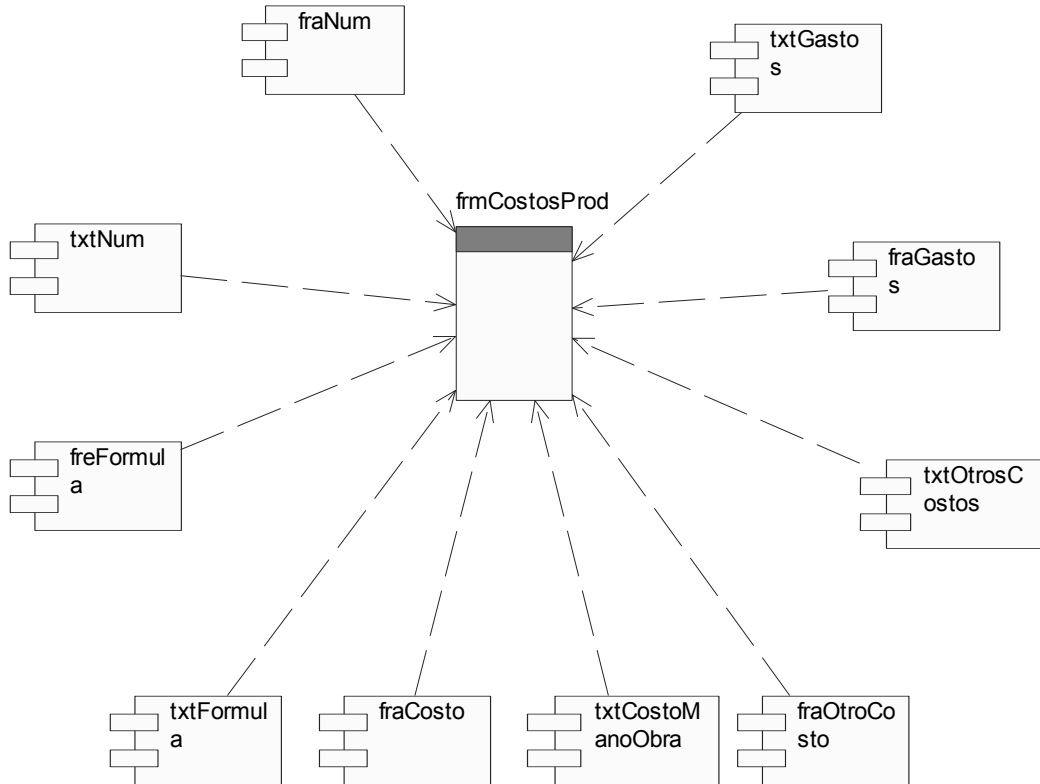
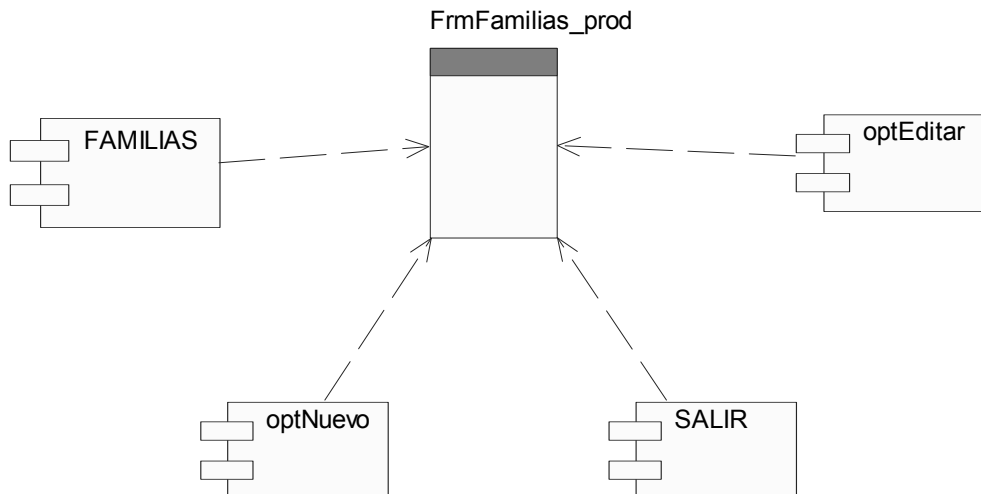
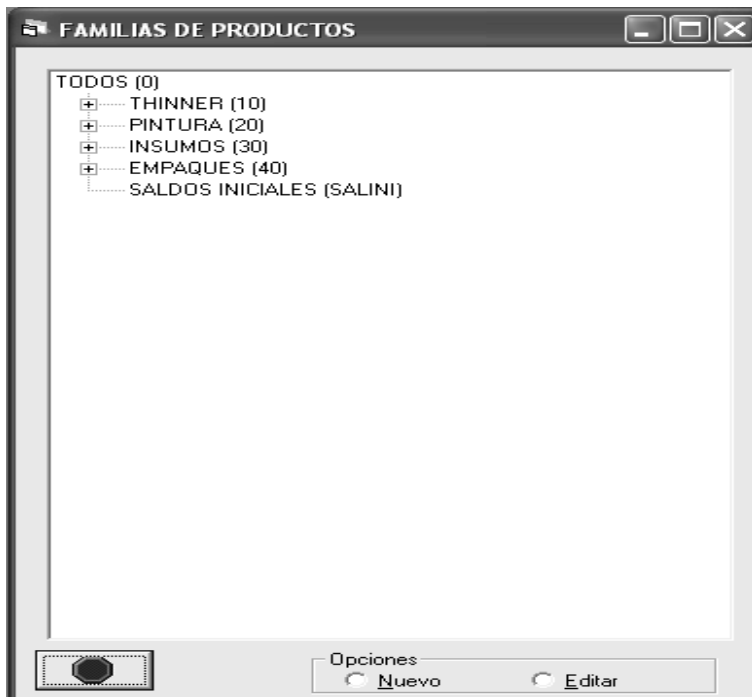


Diagrama de Implementación Formulario: Familias de productos

Diagrama de despliegue



Diseño visual del formulario



Algoritmos

```
Option Explicit
Private SQLS As String
Private RsTempf As Recordset

Private Sub FAMILIAS_DblClick()
    If optNuevo.Value Then
        MsgBox "nuevo"
    Else
        If FAMILIAS.SelectedItem.Text <> "" Then
            If FAMILIAS.SelectedItem.Key <> "P0" Then
                gDesc = Trim(Left(FAMILIAS.SelectedItem.Text,
                InStr(FAMILIAS.SelectedItem.Text, "(") - 1))
                frmEdiItemFam.txtDes =
                Trim(Left(FAMILIAS.SelectedItem.Text,
                InStr(FAMILIAS.SelectedItem.Text, "(") - 1))
                Unload Me
            End If
        End If
    End If
End Sub

Private Sub Form_Load()
    Me.Width = 5850
    Me.Height = 7695
    'FrmAdmConfInt.seleccion = ""
    cargar_arbol ("0")
End Sub

Private Sub salir_Click()
    ' FrmAdmConfInt.seleccion = ""
    Unload Me
End Sub

Public Sub cargar_arbol(cod_raiz As String)
    FAMILIAS.Nodes.Clear
    Dim RsTempo As ADODB.Recordset
    Set RsTempo = New ADODB.Recordset
    Dim RsTempf As ADODB.Recordset
    Set RsTempf = New ADODB.Recordset

    AbrirDB ("M")
```



```

        SQLS = "Select * from mgr_prod_familia where nodo_padre =
'" & cod_raiz & "'"
        RsTempf.Open SQLS, connM, adOpenStatic, adLockReadOnly
        If RsTempf.RecordCount > 0 Then
            'agregue el nodo padre
            SQLS = "Select * from mgr_prod_familia where codigo =
'" & cod_raiz & "'"
            RsTempo.Open SQLS, connM, adOpenStatic, adLockReadOnly
            If RsTempo.RecordCount > 0 Then
                FAMILIAS.Nodes.Add , , "P" & RsTempo("codigo"),
RsTempo("descripcion") & " (" & RsTempo("codigo") & ")"
                While Not RsTempf.EOF
                    'inserte priemro el nodo raiz

                    agregar_subnodo RsTempf("CODIGO"),
RsTempf("DESCRIPCION"), "P" & RsTempo("codigo")
                    RsTempf.MoveNext
                Wend
                FAMILIAS.Nodes("P" & RsTempo("codigo")).Expanded =
True
            End If
            RsTempo.Close
        End If
        RsTempf.Close
        connM.Close
    End Sub

```

```

Private Sub agregar_subnodo(cod_fam As String, Desc As
String, Padre As String)
    Dim RsAdd As ADODB.Recordset
    Set RsAdd = New ADODB.Recordset
    AbrirDB ("M")
    FAMILIAS.Nodes.Add Padre, tvwChild, "P" & cod_fam, Desc &
" (" & cod_fam & ")"

    SQLS = "Select * from mgr_prod_familia where nodo_padre =
'" & cod_fam & "'"
    RsAdd.Open SQLS, connM, adOpenStatic, adLockReadOnly
    If RsAdd.RecordCount > 0 Then
        While Not RsAdd.EOF
            agregar_subnodo RsAdd("CODIGO"),
RsAdd("DESCRIPCIÓN"), "P" & cod_fam
            RsAdd.MoveNext
        Wend
    End If

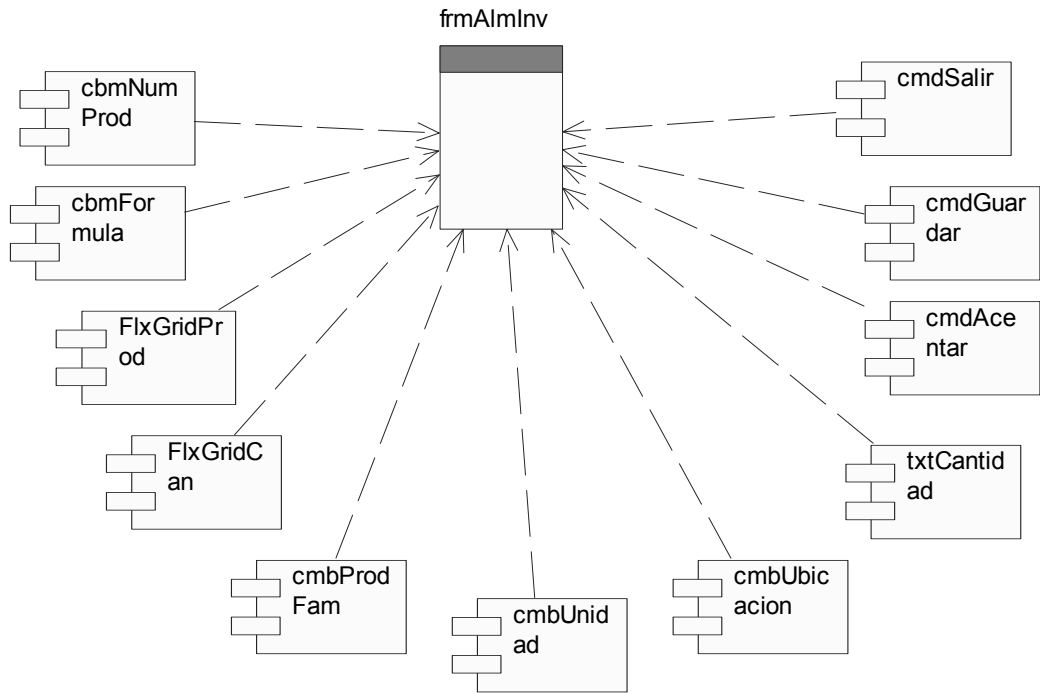
```

```
RsAdd.Close
'connM.Close
End Sub

Private Sub MOSTRARPO()
End Sub
```

Diagrama de Implementación Formulario: Registro de inventario

Diagrama de despliegue



PRUEBAS Y RESULTADOS

Formulario Clasificación de los productos:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
CmbClasificacion	Seleccionar dato	Selecciona y consulta lo datos relacionados bien	
TxtCodClas	Escribir dato erroneo	Permite el dato malo	C01: Revisar escritura de datos
txtDesClas	Escribir datos	Permite los datos digitados	

Formulario Administración de empleados:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
cmdCedula	Escribir datos incorrectos	No acepta datos malos	
txtNombre	Escribir datos erroneos	Acepta datos erroneos	C01: Revisar escritura de datos
TxtDireccion	Escribir cualquier dato	Permite escritura de datos varios	
txtApellido	Escribir cualquier dato	Permite escritura de datos varios	
txtTelefono	Escribir dato alfanumerico	Solo permite escritura de numeros	

Formulario Administración de herramienta:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
cmbCod	Escribir dato incorrecto	No acepta dato incorrecto	
txtNom	Escribir información	Acepta cualquier información	
txtDes	Escribir información	Aceptar cualquier información	
txtVol	Escribir dato alfanumérico	Acepta información correcta	
TxtMat	Escribir dato	Acepta información correcta	
TxtCos	Escribir datos numerico	Acepta dato correcto	

TxtFac	Escribir datos aleatorios	Error de lectura de datos	C01: Revisar escritura de datos
CmbTaller	Seleccionar información	Muestra información correcta	
cmbTipoHerramienta	Seleccionar información	Muestra información correcta	

Formulario Administración de Operaciones:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
cmbCod	Seleccionar información	Muestra información correcta	
txtNom	Digitar inforamción incorrecta	Valida correctamente la información	
CmbProd	Seleccionar información	Muestra información correcta	
txtDes	Digitar información verdadera	Aceptar la información	
txtTiempoPreparacion	Digita información	Valida bien información	
txtTiempoEjecucion	Digita información	Valida bien información	
cmbCentroTrabajo	Seleccionar información	Muestra información correcta	

Formulario Administración de rutas:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
cmbCodigo	Seleccionar información	Muestra información correcta	
TxtDescripcion	Digitar información	Valida bien la información digitada	
LstTodosCT	Lista todas las opciones	Permite la escogencia de la información necesaria	
LstCTAsociados	Lista todas las opciones	Permite la escogencia de la información necesaria	
LstTalleres	Lista todas las opciones	Permite la escogencia de la información necesaria	

LstOperaciones	Lista todas las opciones	Permite la escogencia de la información necesaria	
----------------	---------------------------------	---	--

Formulario Administración de talleres:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
CmbCodigo	Seleccionar información	Muestra información correcta	
TxtDescripcion	Digital información	Acepta la información digitada	
TxtDiasLaborales	Digital días	Valida los días	

Formulario Administración de rutas:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
cmbCodigo	Seleccionar información	Muestra información correcta	
TxtDescripcion	Digital información	Acepta la información digitada	
LstTodosCT	Lista todas las opciones	Permite la escogencia de la información necesaria	

Formulario Administración de fórmulas:

OBJETO	ACCIÓN	RESULTADO	CODIGO DE TAREA
cmbCodFor	Seleccionar información	Muestra información correcta	
CmbNomFor	Seleccionar información	Muestra información correcta	
TxtDesc	Digital información	Acepta la información digitada	
LstInsumos	Lista todas las opciones	Permite la escogencia de la información	

		necesaria	
LstProductos	Lista todas las opciones	Permite la escogencia de la información necesaria	
cmbProducto			
txtCanInsumo	Digitar información	Acepta la información digitada	
TxtDesCantidad	Digitar información	Acepta la información digitada	
cmbProdTer			
TxtCanProd	Digitar información	Acepta la información digitada	
TxtCantidadProd	Digitar información	Acepta la información digitada	
txtPorcentaje			

FASE DE IMPLEMENTACIÓN E IMPLANTACIÓN

MÉTRICAS DE DESARROLLO

1. Facilidad de operación.

¿Requiere el sistema copias de seguridad y de recuperación fiables?	Valoración Pregunta
	5

La aplicación se diseña para operación sin atención.

2. Comunicación de los datos. Los datos o información de control que la aplicación utiliza se envía o recibe a través de los facilidades de comunicación.

¿Se requiere de comunicación de datos?	Valoración Pregunta
	2

Impresión o entrada de datos remota

3. Función distribuida. "Distribuida" significa que los componentes (o los datos) de la aplicación están distribuidos en dos o más procesadores diferentes (esto incrementa el factor anterior).

¿Existen funciones de procesamiento distribuido?	Valoración Pregunta
	0

La aplicación no ayuda a la transferencia de datos o a la función de procesamiento entro los componentes del sistema.

4. Rendimiento. Referido a la importancia de respuesta dentro de todo el sistema.

¿Es crítico el rendimiento?	Valoración Pregunta
	3

Análisis y diseño de las consideraciones del rendimiento son estándar. No se precisan requerimientos especiales por parte del usuario.

5. Configuración utilizada masivamente. Referente a la importancia del entorno. Esto es, si hay restricciones de memoria o del hardware.

¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	Valoración Pregunta
	3

La aplicación corre en una maquina estándar sin restricciones de operación.

6. Tasas de transacción. Una alta llegada de transacciones provoca problemas más allá de los de las características.

Pregunta	Valoración Pregunta
	3

Las tasas son tales que las consideraciones de análisis de rendimiento son estándares.

7. Entrada de datos On-line. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?

¿Requiere el sistema entrada de datos interactiva?	Valoración Pregunta
	0

Hasta el 15% de las transacciones tienen entrada interactiva.

8. Diseño para la eficiencia de usuario final.

Pregunta	Valoración Pregunta
	2

No se especifican requerimientos especiales

9. Actualización on-line.

¿Se actualizan los archivos maestros de forma interactiva?	Valoración Pregunta
	4

Además es esencial la protección contra la pérdida de datos.

10. Complejidad del procesamiento. Esto es, complejidad interna más allá de la media en lo referente a la entrada, salida o lógica de procesamiento. ¿Qué características tiene la aplicación?

- Mucho procesamiento matemático y lógico
- Procesamiento complejo de las entradas
- Procesamiento complejo de las salidas
- Muchas excepciones de procesamiento, muchas transacciones incompletas y mucho procesamiento de las transacciones.
- Procesamiento de seguridad y/o control sensitivo.

¿Son complejas las entradas, las salidas, los archivos o las peticiones? y ¿Es complejo el procesamiento interno?	Valoración Pregunta
	2

Se aplican dos elementos. Procesamiento complejo de las entradas y salidas (verificaciones, restricciones de las entradas y generación de reportes).

11. Utilizable en otras aplicaciones. El código se diseña para que sea compartido o utilizable por otras aplicaciones.

¿Se ha diseñado el código para ser reutilizado?	Valoración Pregunta
	5

Además, la aplicación se "empaquetó" y documentó con el propósito de la fácil reutilización.

12. Facilidad de instalación.

¿Están incluidas en el diseño la conversión y la instalación?	Valoración Pregunta
	0

No se requieren por parte del usuario facilidades especiales de conversión e instalación.

13. Puestos múltiples.

¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	Valoración Pregunta
	0

El usuario no requiere la consideración de más de un puesto.

14. Facilidad de Cambio. Esfuerzo específico de diseño para facilitar cambios futuros.

¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	Valoración Pregunta
	0

No hay requerimientos especiales del usuario para minimizar o facilitar el cambio.

$$PF = \text{Cuenta_total} * [0,65 + 0,01*.(Fi)],$$

METRICAS TECNICAS	VALORACION	PF
Facilidad de operación	5	4.7
Comunicación de los datos.	2	1.88
Función distribuida	0	0
Rendimiento	3	r.82
Configuración utilizada masivamente	3	2.82
Tasas de transacción	3	2.82
Entrada de datos On-line	0	0
Diseño para la eficiencia de usuario final	2	1.88
Actualización on-line	4	3.76
Complejidad del procesamiento	2	1.88
Utilizable en otras aplicaciones	5	4.7
Facilidad de instalación	0	0
Facilidad de Cambio	0	0
TOTAL	29	27.26

MANUAL TÉCNICO

SISTEMA INTEGRADO DE INFORMACIÓN PARA PRODUCCIÓN EN INDUSOL

Esta solución se encarga de administrar la información de producción de la fábrica INDUSOL, registrándola en una base de datos de MS ACCES de Microsoft en un diseño en estrella. Posteriormente se utilizó un componente de Visual Basic, que permite realizar consultas mucho más eficaces al motor de la base de datos, presentando una manera más eficiente a la hora de realizar consultas y mostrar los respectivos resultados.

Recomendación técnicas generales

Plataforma de funcionamiento

Sistema Operativo	Windows
Versión	98 o superior

Plataforma de desarrollo

Lenguaje	Visual Basic
Versión	6.0
Reportes	Cristal Report 4.6

Plataforma de datos

Base de datos	MS Access 2003
---------------	-----------------------

Recomendación previas

1. Para instalar la aplicación entre a la carpeta instalar y ejecute el archivo setup.exe
2. El código fuente se encontrará en la carpeta fuentes
3. Crear la carpeta c:\sae\datos y descargar allí los siguientes archivos.
reportman.mdb
manageDB.mdb
4. Los directorios que encontrará en el cd-room serán:
instalador
fuentes
bases de datos

MANUAL DE USUARIO

SISTEMA INTEGRADO DE INFORMACIÓN PARA PRODUCCIÓN EN INDUSOL

Menú Archivo / Tablas del sistema / centro de costos:

El usuario al ingresar en este menú, se encontrara con el siguiente formulario:



The screenshot shows a software window titled "Administración Centros de Costo". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Below the title bar, the text "Administración Centros de Costo" is displayed. The main area of the window contains a dropdown menu at the top. Below the dropdown menu, there are two input fields: "Código" and "Descripción". At the bottom of the window, there are three buttons: a trash can icon, a floppy disk icon, and a power button icon.

En el cual, se podrán realizar las 4 operaciones básicas como son las de ingresar, consultar, modificar, eliminar centro de costo.

Menú Archivo / Tablas del sistema / centros de trabajo:

Bajo este menú, el usuario podrá configurar los diferentes centros de trabajo que se manejarán en el sistema de producción, acá el usuario deberá asociar herramientas, talleres, operaciones y productos a producir.

The screenshot shows a software interface for managing work centers. The window title is "Form1" and the main heading is "Administración Centros de Trabajo".

Fields for data entry:

- Código:
- Nombre:
- Descripción:

Navigation tabs:

- Herramientas
- Talleres
- Operaciones

Active tab: **Productos a producir**

Section: **Datos de los Productos:**

- Referencias: >>
- Descripción: X

Section: **Capacidad por hora de Produccion:**

- Capacidad X hora:

Toolbar at the bottom contains icons for save, print, and other functions.

Menú Archivo / Tablas del sistema / clasificaciones:

En este menú, el usuario podrá administrar los diferentes tipos de clasificaciones que hay para los productos dentro de las fábricas, como lo son, materia prima, o producto terminado, además podrá utilizar las 4 operaciones básicas de ingresar, consultar, modificar, eliminar del sistema



The image shows a screenshot of a software window titled "Clasificación de productos". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is titled "Clasificación de Productos" and contains a form with the following elements:

- A dropdown menu labeled "Clasificaciones" with a downward arrow.
- A text input field labeled "Código".
- A larger text input field labeled "Descripción".
- A toolbar at the bottom with four icons: a floppy disk (Save), a magnifying glass (Search), a checkmark (Confirm/Save), and a stop sign (Cancel/Close).

Menú Archivo / Tablas del sistema / empleados:

Acá el usuario podrá administrar los diferentes usuarios de la planta de producción, consultando, eliminando, modificando uno a uno de las diferentes personas que laboran en la planta.



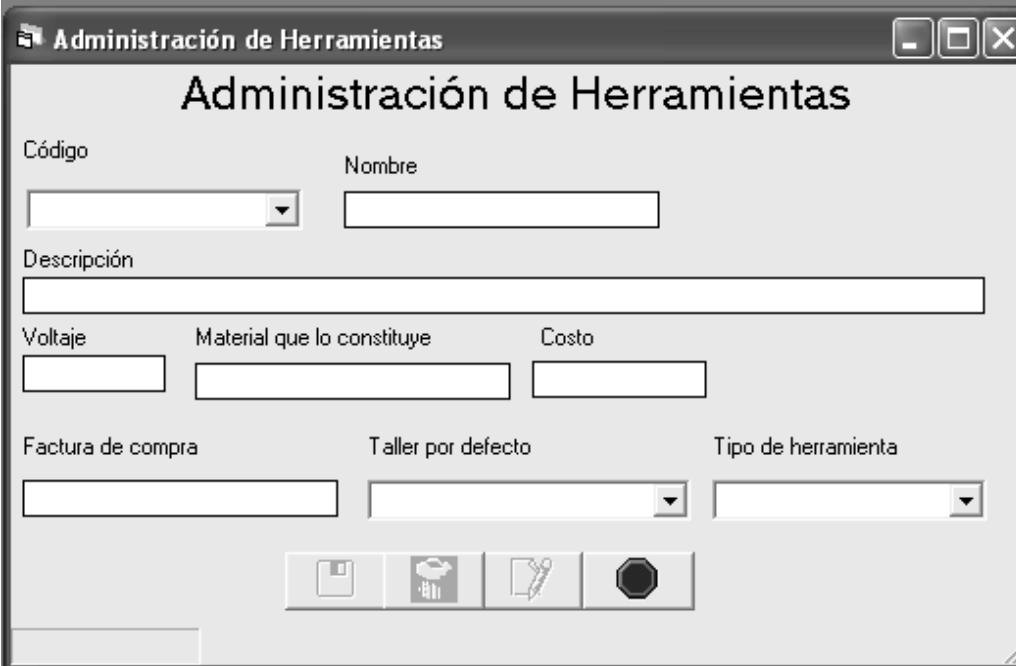
The image shows a screenshot of a software application window titled "frmEmpleados". The window contains a form titled "Administración de Empleados". The form has the following fields:

- Cédula: A dropdown menu.
- Nombre: A text input field.
- Apellido: A text input field.
- Dirección: A text input field.
- Telefono: A text input field.

At the bottom of the form, there is a toolbar with four icons: a floppy disk (save), a trash can (delete), a pencil (edit), and a stop sign (cancel).

Menú Archivo / Tablas del sistema / Herramientas:

Dentro del sistema, se podrán administrar las diferentes herramientas que se manejan dentro de la planta de producción.







Administración de Herramientas

Código Nombre

Descripción

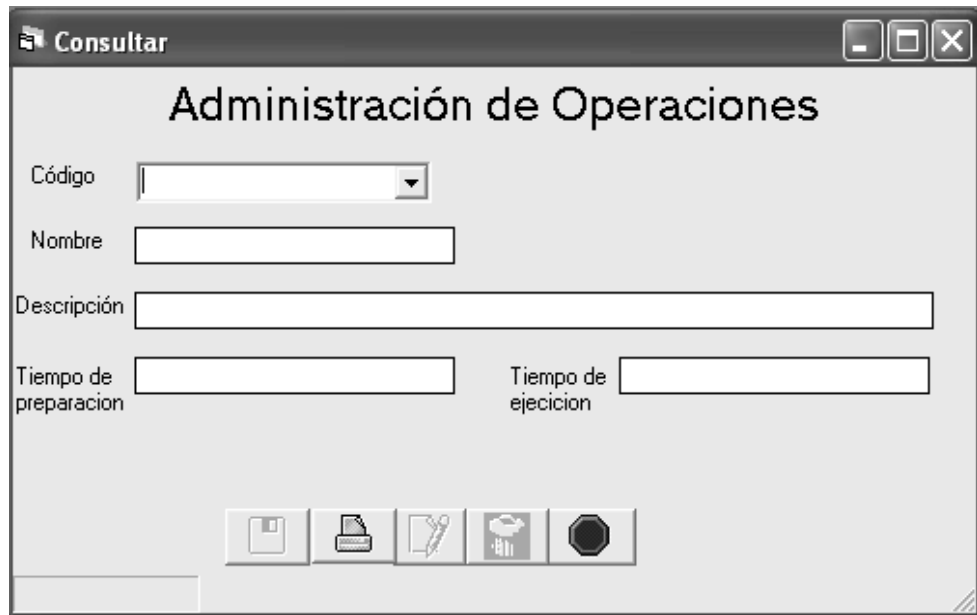
Voltaje Material que lo constituye Costo

Factura de compra Taller por defecto Tipo de herramienta

Menú Archivo / Tablas del sistema / Operaciones:

Dentro del sistema, se podrán administrar las diferentes operaciones de producción que se manejan dentro de la planta de producción.



Consultar






Administración de Operaciones

Código

Nombre

Descripción

Tiempo de preparacion Tiempo de ejecicion

Icons:     

Menú Archivo / Tablas del sistema / Rutas de producción:
Dentro de las rutas de producción, se manejarán los diferentes momentos que tendrá que asumir un producto a la hora de su elaboración.

The screenshot shows a window titled "Rutas de producción" with a subtitle "Administración de Rutas". The interface includes a "Código" dropdown menu, a "Descripción:" text field, and a "Configuración" section. The "Configuración" section is divided into "Todos Los Centros" (a list with items like "El Quinto", "CT", "uno dos tres", "vinilo", "zaq", "nombre 1" and navigation arrows), "CT Asociados" (a large empty box with a ">>" button), "Lista de Talleres" (a text field), and "Lista de Operaciones" (another text field). Below the configuration is an "Observaciones:" text field. At the bottom, there is a toolbar with icons for save, edit, print, and a stop sign.

Menú Archivo / Tablas del sistema / Talleres:

Dentro de los talleres, se podrán configurar los diferentes estados que puede asumir un taller en determinado momento.



The image shows a screenshot of a software application window titled "Talleres". The main content area is titled "Administración de Talleres" and features two tabs: "Administración" (which is selected and highlighted with a dashed border) and "Configuración". Below the tabs, there are two input fields: "Código" with a dropdown arrow and "Descripcion del taller" with a text box. At the bottom of the window, there is a toolbar with four icons: a floppy disk (save), a printer, a pair of scissors (cut), and a stop sign (stop).

Menú Archivo / Producción / Fórmulas:

Dentro de las formulas, se podrán configurar las diferentes formulas de producción, acá se manejaran los insumos, productos y cantidades a producir

The screenshot shows a window titled "Formula de producción" with the main heading "Administración de Fórmulas". The interface includes the following elements:

- Numero de la formula:** A section with two dropdown menus.
- Insumo/Insumo:** A section containing a "Descripción" text field.
- Insumos:** A large empty rectangular box for listing inputs.
- Productos:** A large empty rectangular box for listing products.
- Insumo:** A dropdown menu for selecting an input.
- Producto:** A dropdown menu for selecting a product.
- Cantidad:** Two input fields for quantity, one for the selected input and one for the selected product.
- Porcentaje de pertenencia:** An input field for the percentage of ownership.
- Buttons:** A button with a document icon and a plus sign is located below the "Insumo" and "Producto" dropdowns. A toolbar at the bottom contains icons for a document with an asterisk, a trash can, a close (X) button, and a stop button.

Producción / adición de ítems en familias

Permite administrar la creación de familias de producción, en este formulario, el usuario podrá seleccionar cual es la unidad del producto a crear, que tipo de clasificación maneja, tiene iva y de cuanto, si va al inventario, a que centro de costos pertenece, y que tipo de presentación maneja el producto creado.

The screenshot shows a software window titled "EDICION DE ITEMS A UNA FAMILIA" with a subtitle "Administrar Familias en Productos". The window is divided into several sections:

- NODO PADRE:** Contains two input fields labeled "CODIGO" and "DESCRIPCION".
- DATOS DEL NODO:** Contains two input fields labeled "CODIGO" and "DESCRIPCION", and a checkbox labeled "Usable".
- Datos del producto / Presentaciones:** A tabbed interface with two tabs: "Datos del producto" (selected) and "Presentaciones".
 - UNIDAD:** A dropdown menu.
 - CLASIFICACION:** A dropdown menu.
 - IVA:** A checkbox with an adjacent input field.
 - INVENTARIO:** A checkbox.
 - Centro de costos:** A dropdown menu.
 - PRODUCCION MINIMA:** An input field.
 - PRODUCCION MAXIMA:** An input field.
 - STOK MIN. STOK MAX.:** Two input fields.

At the bottom of the window, there are two buttons: a save icon (floppy disk) and a stop icon (octagon).