

ÍNDICE

ÍNDICE.....	1
INTRODUCCIÓN	2
REFERENCIAS	4
MODELO DE DESPLIEGUE.....	6
AMBIENTE DE IMPLEMENTACIÓN	6
RELATIVO A LA PLATAFORMA Y RED.....	6
RELATIVO AL LENGUAJE DE PROGRAMACIÓN.....	7
RELATIVO A BASE DE DATOS Y RESGUARDO.....	7
RELATIVO AL MÓDULO WEB	7
DIAGRAMA DE DESPLIEGUE.....	8
LAYOUT DEL EQUIPAMIENTO.....	10
MODELO DE DISEÑO	11
DIAGRAMA DE CLASES DE DISEÑO – MAPEO A BASE DE DATOS RELACIONALES	11
DIAGRAMA DE ESTADOS	13
CLASE EVENTO.....	13
CLASE ORDENDEINSTALACION.....	15
CLASE ORDENDEREPARACION.....	16
CLASE PEDIDODEPIEZAS.....	17
CLASE PEDIDODEREPARACION.....	18
CLASE VISITA	19

INTRODUCCIÓN

El **Flujo de Trabajo de Diseño** es un flujo de trabajo del Proceso de Desarrollo Unificado. Tiene como principal objetivo refinar la solución y la estructura que satisface los requisitos considerando aspectos de la implementación, creando así un modelo más específico y una abstracción sin costuras de la implementación del sistema.

De esta manera, durante el Diseño se procede a moldear el sistema y encontrar su forma, para que incluya: los requisitos incorporados en el sistema mediante las diferentes funcionalidades, los componentes de código que se compilan e integran en versiones ejecutables y su arquitectura.

En el Diseño se deben tomar decisiones relativas a **cómo** debería el sistema tratar los requisitos de rendimiento y de distribución.

Los principales propósitos del diseño son

- ✚ Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario.
- ✚ Crear una entrada apropiada y un punto de partida para actividades de implementación capturando los requisitos, interfaces o clases.
- ✚ Crear una abstracción para la implementación de manera que ésta sea un refinamiento directo del diseño.

El modelo de diseño tiene las siguientes características

- ✚ Es un modelo físico no genérico y específico para una implementación
- ✚ Los estereotipos que se definen en este momento dependen directamente del lenguaje de programación
- ✚ Es un modelo dinámico centrado en las secuencias de mensajes entre los objetos
- ✚ Se debe mantener durante todo el ciclo de vida del sistema

La etapa de diseño utiliza entre otros, los siguientes diagramas UML

- ▲ Diagrama de Despliegue
- ▲ Diagrama de Clases –mapeo a base de datos relacionales–.
- ▲ Diagrama de Estado

Para comprender en detalle el modelo de diseño se describen a continuación los artefactos utilizados.

El **diagrama de despliegue** es un modelo de objetos que representa la distribución física del sistema, y describe cómo la funcionalidad definida se distribuye entre los nodos de cómputo. En este caso, cada nodo simboliza un recurso de cómputo – normalmente un procesador o dispositivo de hardware –, y éstos se relacionan mediante distintos medios de comunicación.

Agatha – Sistema de Soporte Logístico

Junto al diagrama de despliegue, y conformando el modelo de despliegue, se encuentra la **especificación del ambiente de implementación** que define parte de la arquitectura física del sistema. Esta especificación incluye una descripción del hardware, junto con la plataforma software propuesta para el correcto funcionamiento del sistema. Establece el lenguaje de programación utilizado para el desarrollo del producto, el DBMS y otras especificaciones técnicas relativas al software, tales como políticas de seguridad y respaldo.

Una **clase de diseño** constituye una abstracción sin costuras de una clase o construcción similar en la implementación del sistema, de manera que el lenguaje utilizado es el mismo del lenguaje de programación. Para el caso del sistema en construcción se procede a identificar los atributos requeridos por la clase de diseño y se describen utilizando la sintaxis del lenguaje de programación. Los tipos de atributos disponibles están restringidos por el lenguaje de programación y el sistema de base de datos utilizado.

Para el desarrollo de este sistema se utilizará una base de datos relacional, debido a la capacidad limitada de la memoria primaria. Para ello es necesario determinar qué clases –y cuáles variables o atributos de cada clase – deben almacenarse en forma persistente. Al momento de construir y definir la estructura de la base de datos, cada una de estas clases será representada por al menos una tabla. De esta manera el mapeo a base de datos relacionales de cada clase consiste en la tarea de convertir las clases en tablas que pertenecerán a un DBMS específico.

El **diagrama de estados** muestra una *máquina de estados*, destacando el flujo de control entre estados. Una máquina de estados es un comportamiento que especifica las secuencias de estados por las que pasa un objeto a lo largo de su vida en respuesta a distintos eventos, junto con sus respuestas a esos eventos.

De esta manera, el diagrama de transición de estados especifica el ciclo de vida de un objeto en términos de estados y transiciones. Cada transición es una secuencia de acciones.

Cada diagrama de estados es una entrada de valor para la implementación correspondiente de la clase de diseño. Un diagrama de estados se compone de:

- ✚ Estados: es una condición o situación en la vida de un objeto durante la cual satisface alguna condición, realiza alguna actividad o espera algún evento.
- ✚ Eventos: especificación de un acontecimiento que ocupa un lugar en el tiempo y en el espacio; en este contexto, es la aparición de un estímulo que puede activar una transición de un estado.
- ✚ Transición: relación entre dos estados que indica que un objeto que esté en el primer estado realizará ciertas acciones y entrará en el segundo estado cuando ocurra un evento y se satisfagan las condiciones especificadas.
- ✚ Actividad: ejecución no atómica en curso, dentro de una máquina de estados.
- ✚ Acción: computación atómica ejecutable que produce un cambio en el estado del modelo o la devolución de un valor.

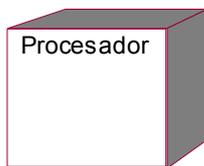
Agatha – Sistema de Soporte Logístico

En resumen, en el Diseño se describe la realización física de los casos de uso centrándose en cómo cumplir con los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación. En el diseño las decisiones estratégicas y tácticas se toman para resolver los requisitos funcionales y de calidad requeridos de un sistema.

Referencias

Antes de comenzar con el desarrollo del flujo de trabajo de diseño, se presenta una breve referencia de la simbología utilizada en los diagramas.

Diagrama de Despliegue



Nodo con capacidad de procesamiento, es decir que puede ejecutar un artefacto o componente



Nodo sin capacidad de procesamiento que en general representa algo que interactúa con el mundo real

Agatha – Sistema de Soporte Logístico

Diagrama de Estados



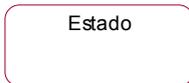
Estado Inicial

Estado a partir del cual comienza el diagrama de estados



Estado Final

Estado que indica la finalización de la actividad del estado de origen



Estado

Condición o situación durante la vida de un objeto, a lo largo de la cual satisface alguna condición o espera algún evento



Se muestra una transición, relación entre dos estados de una máquina de estados. Un evento hace que la transición pueda dispararse. La acción constituye de un procedimiento que puede afectar al objeto.

La sintaxis de la especificación que acompaña a una transición de estados es la siguiente:

[condición] ^ evento / acción

La condición y la acción son opcionales.

MODELO DE DESPLIEGUE

Ambiente de Implementación

Relativo a la plataforma y red

En relación a la red, el sistema se implementará en una red conformada por cinco máquinas relacionadas mediante una topología de estrella. Esta red estará montada sobre cable UTP categoría 5, y contará con un hub centralizador de 16 bocas, rosetas tipo AMP y sus respectivos patchcords.

El cliente de redes a utilizar será el Cliente para Redes Microsoft con IP V4 tipo 3 fijos para cada máquina. Los mismos serán configurados arbitrariamente al momento de la implementación dejando el rango de la última capa libre, de la posición 10 hacia abajo para establecer en este espacio el set de servidor/es.

La máquina que efectuará la tarea de servidor central contará con un sistema operativo Windows 2000 Server SP4, con las políticas de seguridad establecidas para el acceso desde las terminales existentes bajo plataforma Windows 98. A este servidor se encontrará directamente conectado el patchcord del hub, permitiendo el acceso de las máquinas cliente.

El servidor central tendrá instalado un motor de base de datos "Microsoft SQL Server 2000 Enterprise Edition SP4".

El servidor de bases de datos poseerá una instancia del sistema para su manejo.

Se recomienda la compra de un servidor dedicado que actuará como servidor y servidor Web, esto debido a la necesidad de memoria y capacidad mínima disponible a la hora de efectuar las consultas SQL sobre el motor. El mismo deberá contar para su óptimo rendimiento con un procesador Xeon de al menos 2.4 GHz y 1gb de RAM, así como discos SATA II o SCSCI 3. De esta manera se obtiene una performance aceptable de velocidad y asignación de recursos dentro del ordenador.

El disco deberá estar particionado en dos fragmentos. En uno de ellas se instalará el sistema operativo manteniendo el espacio de intercambio mínimo o swap recomendable. En la otra partición se instalarán en forma exclusiva los archivos dedicados al almacenamiento físico de la base de datos, archivos de log y backup diario.

Debe existir la posibilidad de acceder a un disco, dedicado exclusivamente al mantenimiento de los backups semanales y/o mensuales. Es recomendable la adquisición de algún dispositivo de grabación de medios ópticos (Grabadora de CD o DVD) para descartar la necesidad de un espacio excesivo de resguardo en la máquina designada para el almacenamiento de los backups.

El resto de las máquinas podrá conservar su configuración original, con la salvedad de contar con una instancia del sistema administrativo.

Agatha – Sistema de Soporte Logístico

Para el correcto funcionamiento de las transacciones con la base de datos se instalará un ADO Update Version, en aquellas máquinas que no posean una versión de los mismos superior a 2.77 en una de sus capas de sistema operativo. Este aspecto es necesario debido a bugs o errores en el desarrollo de los objetos COM y OLE dentro de las versiones anteriores a Windows 2000 Server para el enlace a bases de datos.

Se propone como solución la compra de las licencias pertinentes a *Windows 2000 Server* y *SQL Server 2000 Enterprise Edition*, a fin de salvaguardar la integridad de los datos y respetar los aspectos legales de la implementación del sistema.

Relativo al lenguaje de programación

El sistema se programará íntegramente en el lenguaje de programación 4GL Borland Delphi 7.0 Enterprise Edition.

El sistema será desarrollado en tres capas, con la implementación de objetos tipo ADO para la comunicación de la capa de datos con el motor de bases de datos.

Relativo a base de datos y resguardo

En relación al archivo físico de la base de datos, el mismo se ubicará en el disco destinado para ello, y se encontrará implementado en dos archivos que define el motor de bases de datos elegido. El primer archivo donde se encuentran los datos físicos se denominará "agatha.mdf". El archivo correspondiente a logs de transacción se definirá como "agatha_log.ldf".

Se ejecutará una rutina de backup diferencia automática diaria, programada dentro del agente del motor que se ejecuta como servicio dentro de Windows 2000 Server; de esta manera no existirá la necesidad de tener un usuario utilizando el sistema para realizar el resguardo. Dicha tarea se realizará en horarios pre-convenidos con el responsable del área, procurando su ejecución en horas de inactividad del sistema – para evitar así la disminución de performance del sistema que implica la tarea de backup dentro del motor de bases de datos –

En relación a los backups semanales y/o mensuales, el intervalo entre los mismos se estimará mediante reglas de cálculo basadas en las transacciones efectuadas en períodos de muestra posteriores a la instalación del sistema y según se estime pertinente.

El backup se almacenará en un archivo .zip, generado a partir del backup realizado por el motor de bases de datos.

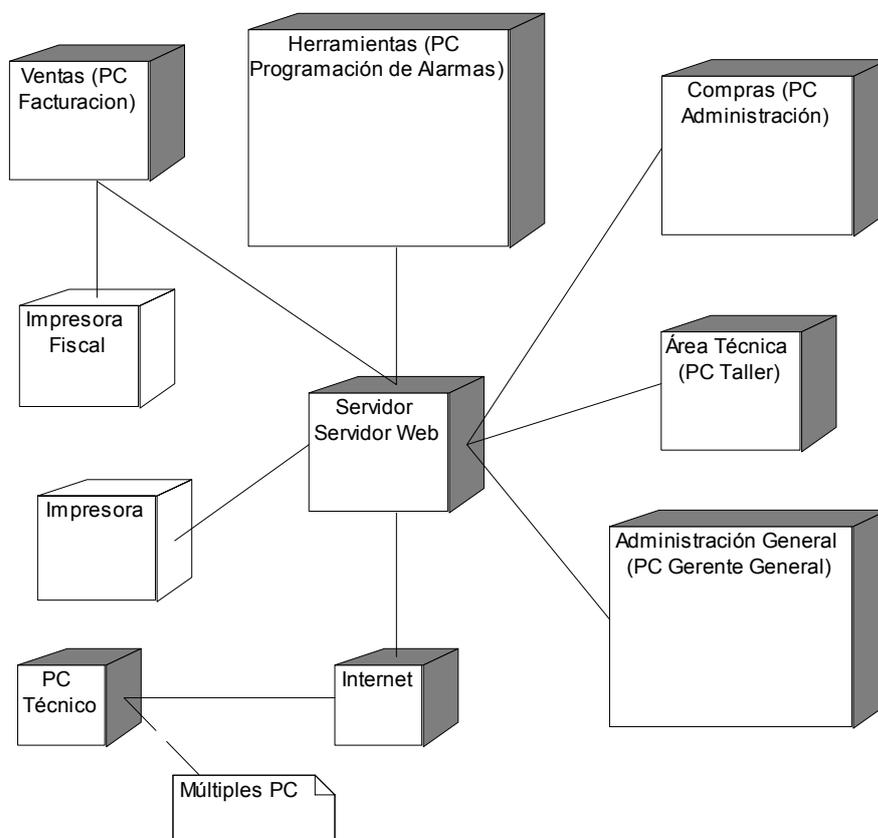
Relativo al Módulo Web

El servidor central deberá también tener habilitados los servicios de Internet Information Services del sistema operativo para poder ejecutar el módulo Web que permita el acceso desde cualquier punto de Internet a la consulta designada para tal efecto.

Diagrama de Despliegue

Como se mencionó anteriormente el diagrama de despliegue muestra la distribución de los recursos computacionales. A continuación se presenta el diagrama de despliegue propuesto que soportará al sistema de información desarrollado.

Diagrama de Despliegue
Agatha - Sistema de Soporte Logístico



Como se observa en el diagrama, se mantiene el equipamiento actual de la empresa y se incorpora la máquina que cumplirá la función de Servidor y Servidor Web.

Las características de cada máquina se detallan a continuación:

La máquina correspondiente a Ventas cuenta con sistema operativo Windows 98. La configuración física de esta máquina es: microprocesador Pentium con 32 Mb de memoria RAM, una lectora CD-ROM, disco rígido fragmentado en 986Mb y 228Mb, y una diskettera 3½.

Agatha – Sistema de Soporte Logístico

La máquina destinada a la Gestión de Herramientas cuenta con un sistema operativo Windows 98. Sus características físicas son: microprocesador Pentium con 32 Mb de memoria RAM, un disco rígido fragmentado en dos partes con 6,86 Gb y 1,86 Gb respectivamente, monitor 15" y diskettera 3½.

La máquina para Compras tiene sistema operativo Windows 2000. Su equipamiento consiste en un microprocesador Pentium III, 262 Mb de Memoria RAM, un disco rígido fragmentado en 3 partes con 9,28Gb, 17Gb y 11,30Gb respectivamente, monitor 15" y diskettera 3½.

La máquina del Área Técnica posee sistema operativo Windows 98. Cuenta con un microprocesador Pentium Celeron, 160 Mb de memoria RAM, un disco rígido fragmentado en 3 partes con 1,65 Gb; 1,26 Gb y 1 Gb respectivamente, un monitor 15" y una diskettera 3½.

La PC destinada a Administración General del Sistema posee sistema operativo Windows XP. Sus características físicas son microprocesador Pentium III de 935 MHz, 320 Mb de memoria RAM, lectora CD-ROM, monitor de 17", diskettera 3½, disco duro fragmentado en siete particiones con 8,74 Gb cada una, cinco discos con 2 Gb y otro con 266Mb.

El equipo que será Servidor y Servidor Web deberá contar con un procesador Xeon de al menos 2.4 GHz y 1gb de RAM, así como discos SATA II o SCSCI 3. El servidor tendrá una conexión a Internet para atender las consultas de eventos y órdenes de trabajo realizadas por los técnicos.

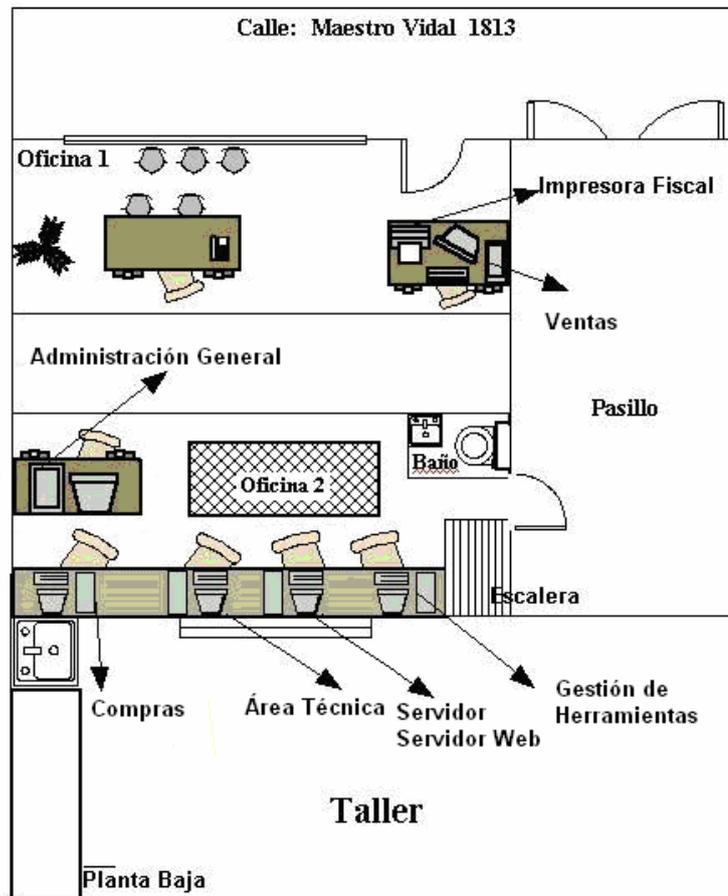
En cuanto a dispositivos periféricos, existe una impresora fiscal conectada a la máquina de Ventas y una impresora Hewlett Packard conectada al Servidor.

Las máquinas conectadas al Servidor Web deberán contar con un explorador Internet Explorer 5.5 en adelante o Mozilla Firefox 8.0 en adelante para poder realizar las consultas Web proporcionadas al técnico.

El sistema se podrá ejecutar en todos los nodos presentados anteriormente, siempre que los mismos se encuentren conectados a la red y al Servidor. Es necesario aclarar que todas las opciones del sistema estarán disponibles en cualquier máquina de la red, siendo estas opciones restringidas de acuerdo a los permisos de acceso y perfil del usuario logueado.

Layout del equipamiento

A continuación se muestra la distribución física del equipamiento recomendado en el ambiente de implementación.



MODELO DE DISEÑO

Diagrama de Clases de Diseño – Mapeo a Base de Datos Relacionales

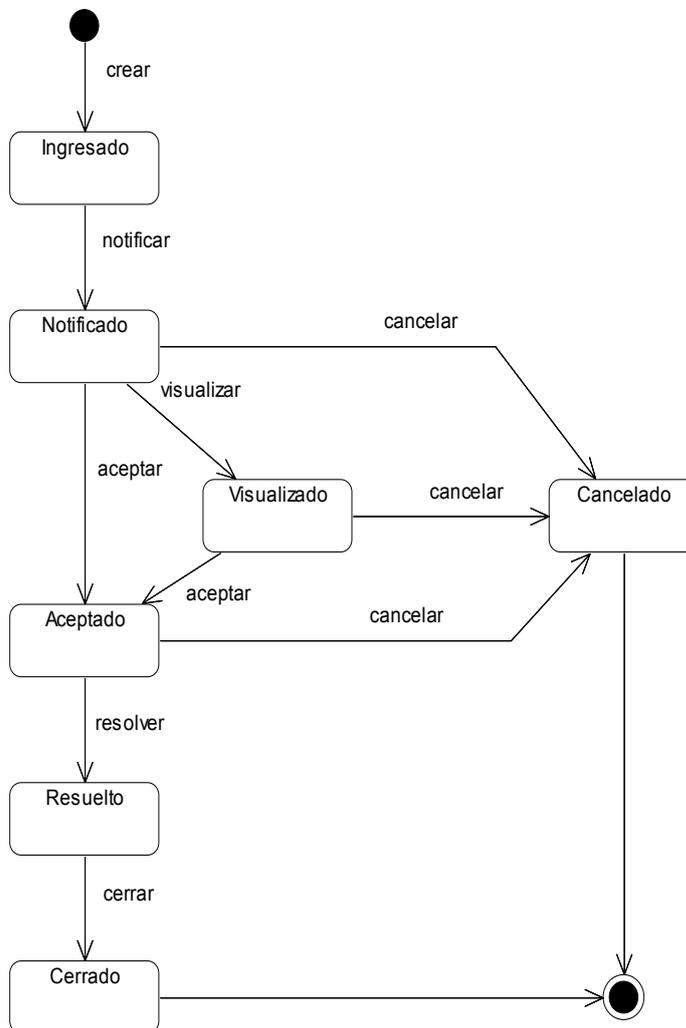
A continuación se presenta el diagrama en el cual se muestra el mapeo de las clases persistentes del diagrama de clases a las tablas de la base de datos relacionales. En este caso sólo se trasladan a las tablas los atributos de las clases, que se convierten en las columnas correspondientes.

Diagrama de Estados

En la presente sección se muestran los diagramas de estado de aquellas clases cuyos objetos atraviesan por diferentes estados a lo largo de su ciclo de vida dentro del sistema.

Clase Evento

Diagrama de Transición de Estados
Clase Evento



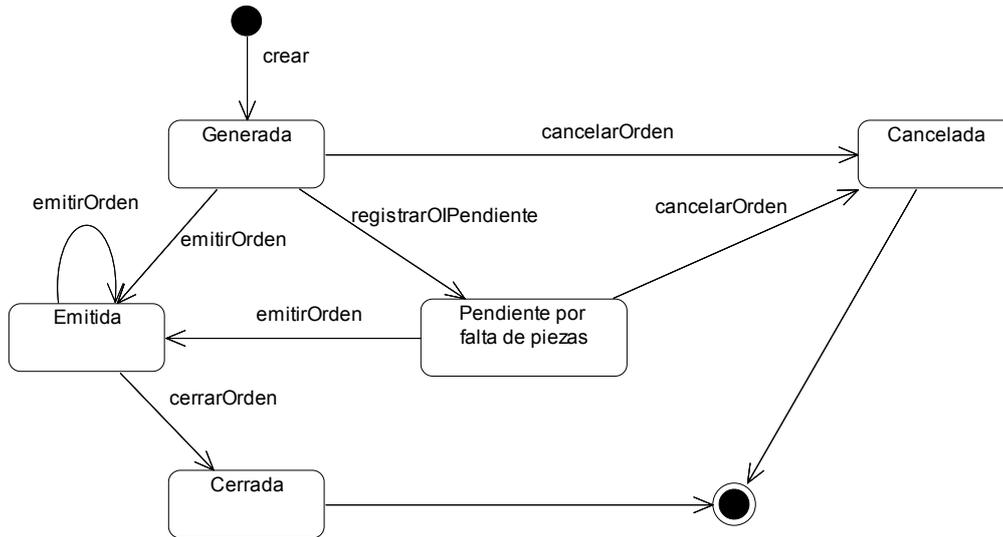
Casos de Uso relacionados

En el siguiente cuadro se presenta la relación entre los métodos que provocan los cambios de estado en la clase Evento y los casos de uso en los cuales se ejecutan o invocan dichos métodos.

Método	Caso de Uso	
	Nro.	Nombre
crear()	30.	Registrar evento
	10.	Registrar orden de instalación
	9.	Registrar orden de reparación
	1.	Registrar visita a domicilio
	5.	Registrar cancelación de visita
notificar()	12.	Consultar evento [El evento se muestra por primera vez al usuario]
visualizar()	12.	Consultar evento [El usuario accede por primera vez a la consulta en detalle del evento]
	13.	Registrar avance de evento [El usuario accede a la consulta en detalle del evento y selecciona el nuevo estado del evento]
aceptar()	13.	Registrar avance de evento [El usuario aceptó el evento]
resolver()	13.	Registrar avance de evento [El usuario resolvió el evento]
cerrar()	13.	Registrar avance de evento [El usuario cerró el evento]
cancelar()	13.	Registrar avance de evento [El usuario canceló y rechazó el evento]

Clase OrdenDelInstalacion

Diagrama de Transición de Estados
Clase OrdenDelInstalacion



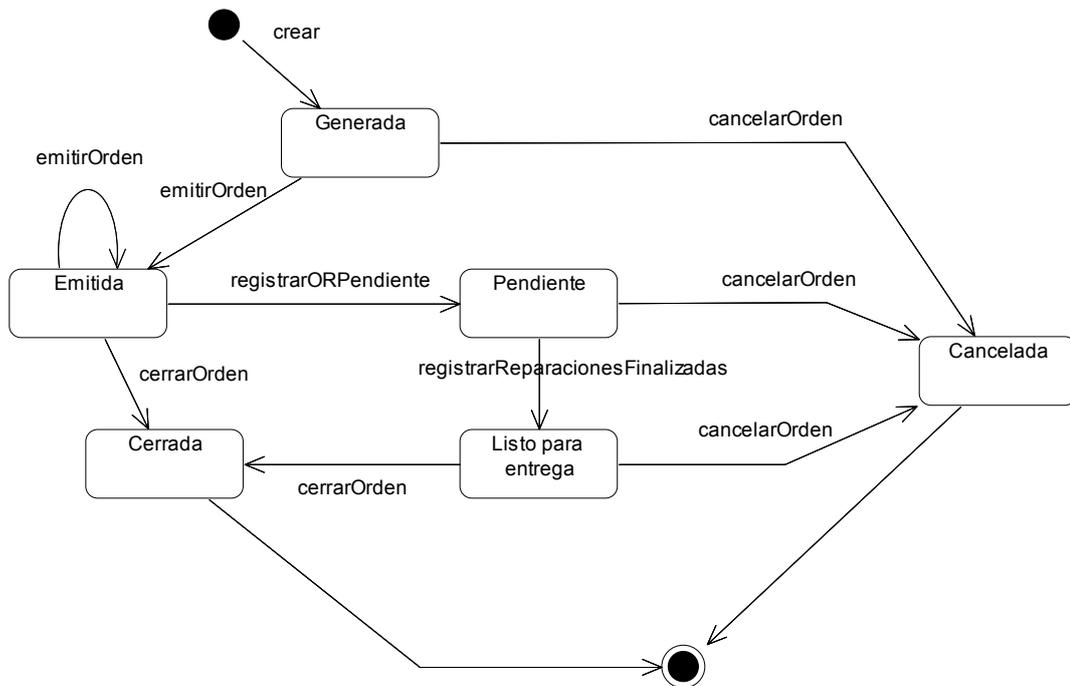
Casos de Uso relacionados

En el siguiente cuadro se presenta la relación entre los métodos que provocan los cambios de estado en la clase OrdenDelInstalacion y los casos de uso en los cuales se ejecutan o invocan dichos métodos.

Método	Caso de Uso	
	Nro.	Nombre
crear()	10.	Registrar orden de instalación
emitirOrden()	14.	Emitir orden de trabajo [Hay stock suficiente de piezas]
cerrarOrden()	27.	Registrar cierre de orden de trabajo
registrarOIPendiente()	14.	Emitir orden de trabajo [No hay stock de alguna pieza de la orden]
cancelarOrden()	28.	Registrar cancelación de orden de trabajo

Clase OrdenDeReparacion

Diagrama de Transición de Estados
Clase OrdenDeReparacion



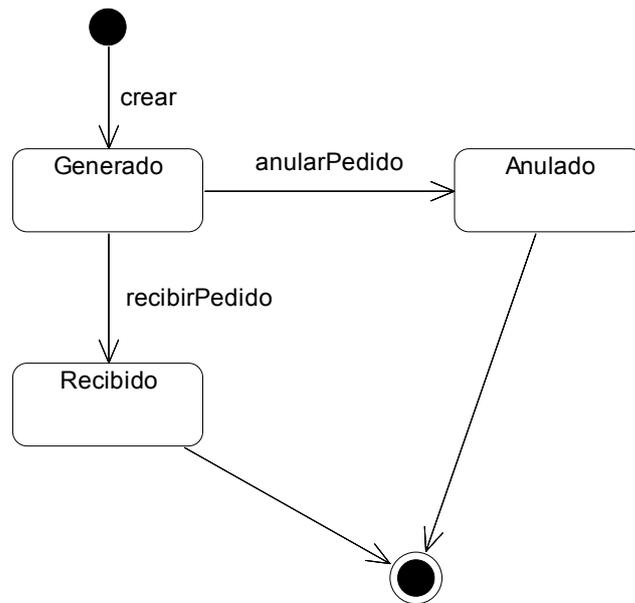
Casos de Uso relacionados

En el siguiente cuadro se presenta la relación entre los métodos que provocan los cambios de estado en la clase OrdenDeReparacion y los casos de uso en los cuales se ejecutan o invocan dichos métodos.

Método	Caso de Uso	
	Nro.	Nombre
crear()	9.	Registrar orden de reparación
emitirOrden()	14.	Emitir orden de trabajo
cerrarOrden()	27.	Registrar cierre de orden de trabajo
registrarORPendiente()	22.	Generar pedido de reparación
registrarReparacionesFinalizadas()	24.	Registrar reparaciones de piezas
cancelarOrden()	28.	Registrar cancelación de orden de trabajo

Clase PedidoDePiezas

Diagrama de Transición de Estados
Clase PedidoDePiezas



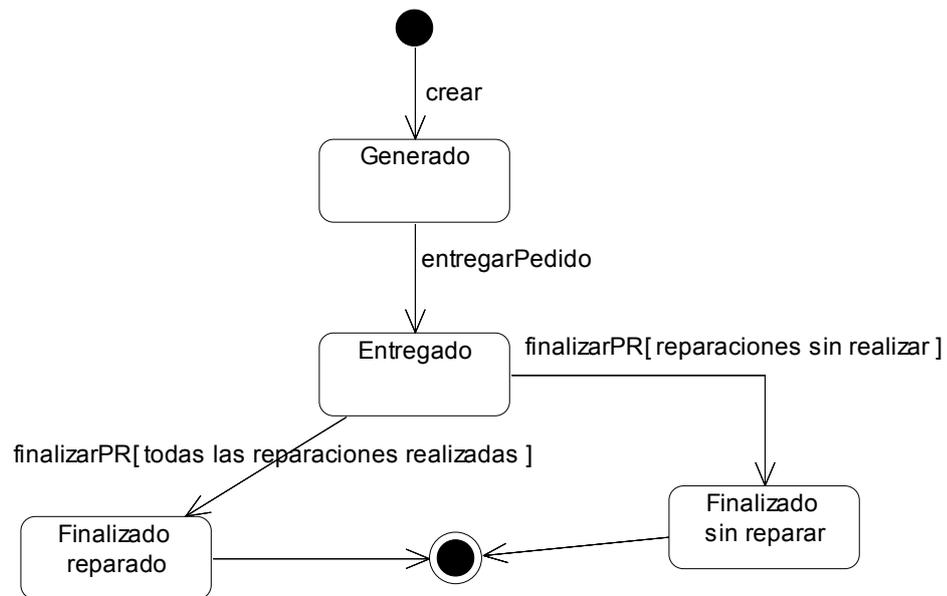
Casos de Uso relacionados

En el siguiente cuadro se presenta la relación entre los métodos que provocan los cambios de estado en la clase PedidoDePiezas y los casos de uso en los cuales se ejecutan o invocan dichos métodos.

Método	Caso de Uso	
	Nro.	Nombre
crear()	16.	Registrar pedido de piezas
recibirPedido()	17.	Registrar ingreso de piezas
anularPedido()	18.	Anular pedido de piezas

Clase PedidoDeReparacion

Diagrama de Transición de Estados
Clase PedidoDeReparacion



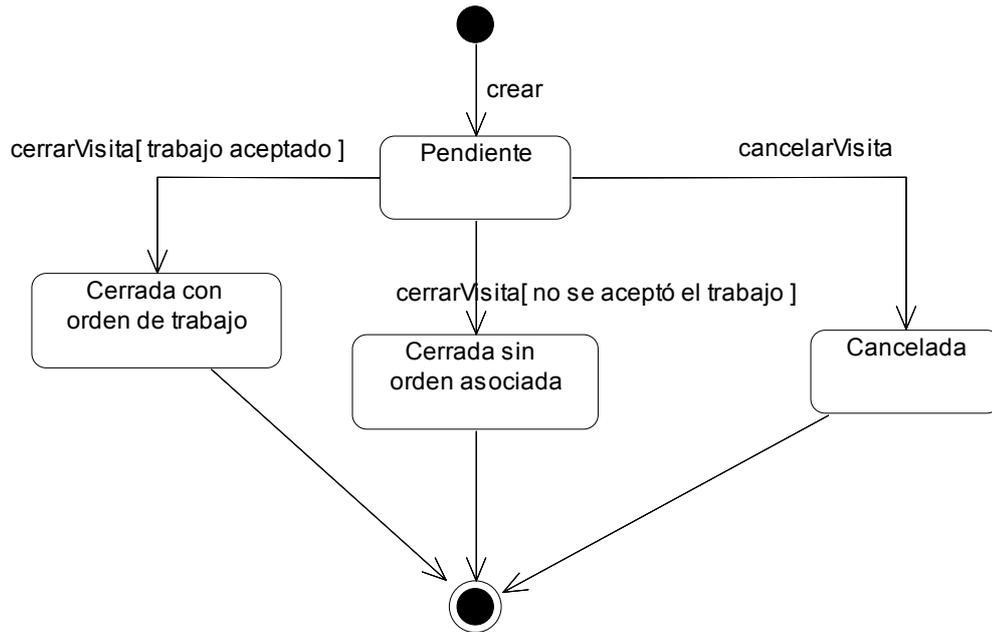
Casos de Uso relacionados

En el siguiente cuadro se presenta la relación entre los métodos que provocan los cambios de estado en la clase PedidoDeReparacion y los casos de uso en los cuales se ejecutan o invocan dichos métodos.

Método	Caso de Uso	
	Nro.	Nombre
crear()	22.	Generar pedido de reparación
entregarPedido()	23.	Registrar entrega de pedido de reparación
finalizarPR()	24.	Registrar reparaciones de piezas

Clase Visita

Diagrama de Transición de Estados
Clase Visita



Casos de Uso relacionados

En el siguiente cuadro se presenta la relación entre los métodos que provocan los cambios de estado en la clase Visita y los casos de uso en los cuales se ejecutan o invocan dichos métodos.

Método	Caso de Uso	
	Nro.	Nombre
crear()	1.	Registrar visita a domicilio
cerrarVisita()	4.	Registrar cierre de visita
cancelarVisita()	5.	Registrar cancelación de visita