

ESCUELA:
TECNOLOGÍA



ENTERPRISE ARCHITECT



indra

ANGULAR 2.0

ESCUELA:
TECNOLOGÍA



■ NOMBRE APELLIDO PROFESOR
Javier Martín

■ VER PERFIL COMPLETO:



■ CONTACTO



jmartin@grupoloyal.com



<https://sparxsystems.com/>

© JMA 2019. All rights reserved

INTRODUCCIÓN

© JMA 2019. All rights reserved

Plataforma de modelado integrada

- Modelado de ciclo de vida completo para:
 - Sistemas de negocios y TI
 - Ingeniería de software y sistemas
 - Desarrollo en tiempo real e integrado
- Con capacidades integradas de administración de requisitos, Enterprise Architect ayuda a rastrear especificaciones de alto nivel para análisis, diseño, implementación, prueba y modelos de mantenimiento utilizando UML, SysML, BPMN y otros estándares abiertos.
- Enterprise Architect es una herramienta gráfica multiusuario diseñada para ayudar a los equipos a construir sistemas robustos y fáciles de mantener.
- Con capacidades de control de versiones estrechamente integradas y servidores implementables basados en la nube, Enterprise Architect también permite que los equipos distribuidos globalmente colaboren de manera efectiva en proyectos compartidos.

© JMA 2019. All rights reserved

Características

- Diseñar y construir una amplia gama de sistemas de software
- Análisis de Negocios, modelado de procesos de negocio, administración de requisitos
- Modelado de Sistemas, modelado de arquitectura de sistemas, diseño de componentes, simulación
- Modelado específico de dominio generalizado
- Construir lenguajes de modelado específicos de dominio basados en UML
- Visualizar una amplia gama de sistemas, procesos, datos, actividades y estructuras
- Simular procesos conductuales, máquinas de estado e interacciones
- Colaborar y compartir información y modelos
- Pruebas, control de calidad y verificación/trazabilidad extremo-a-extremo de sistemas complejos
- Administrar tareas de desarrollo, trabajo y planificación

© JMA 2019. All rights reserved

Flexible

- **NIEM:** El Modelo de intercambio de información nacional, National Information Exchange Model (NIEM), provee un marco de trabajo común que se usa para definir como se comparte la información entre sistemas, agencias del gobierno y organizaciones.
- **Estándares abiertos:** Basado en estándares abiertos como UML, BPMN y SysML. Soporta los marcos de trabajo de arquitectura de empresas como TOGAF y UPDM. Herramientas personalizadas integradas para analizar y visualizar el software en ejecución. Simulación avanzada, herramientas de pruebas, repositorios basados en el equipo, control de versiones y más.
- **Probar y depurar:** Desarrollar y visualizar software. Probar y verificar el comportamiento correcto. Simular procesos. Establecer y administrar puntos de prueba. Depurar e inspeccionar software en ejecución. Simulación avanzada, herramientas de prueba, repositorios basado en el equipo, control de versiones y más.
- **Simulación:** Capaz de simular dinámicamente comportamiento y modelos de estado. Confirmar diseño del proceso. Especificar disparadores, eventos, restricciones y más. Especificar sistemas estructurales grandes y complejos. Crear y depurar soluciones embebidas. Construir soluciones específicas del dominio personalizado.

© JMA 2019. All rights reserved

Modelar, gestionar y rastrear

- Capture y rastree requisitos formales para diseñar, construir, implementar y más. Utilice el análisis de impacto para rastrear los cambios propuestos a los requisitos originales. Construye el sistema correcto .
- Rastrear los cambios propuestos
- Las funciones integradas de gestión de requisitos de Enterprise Architect se pueden utilizar para:
 - Definir un modelo de requisitos jerárquicos y organizados
 - Trace la implementación de los requisitos del sistema para modelar elementos
 - Busque y reporte requisitos
 - Realice análisis de impacto de los cambios propuestos en los requisitos

© JMA 2019. All rights reserved

Trazabilidad

- Enterprise Architect proporciona una trazabilidad completa desde los requisitos, los modelos de análisis y diseño, hasta la implementación y la implementación. La verificación efectiva, la validación y el análisis de impacto inmediato son posibles a lo largo de todo el ciclo de vida, utilizando capacidades tales como la matriz de relaciones y la vista de jerarquía de Enterprise Architect.
- En combinación con la asignación integrada de tareas y recursos, los jefes de proyecto y los equipos de control de calidad están equipados con la información correcta para ayudar a entregar proyectos con éxito.

© JMA 2019. All rights reserved

Administrar información compleja

- Enterprise Architect ayuda a los individuos, grupos y organizaciones grandes a modelar y administrar información compleja. Integrando y conectando un amplio rango de información estructural y de comportamiento en forma visual, puede construir un modelo coherente y comprobable de lo que es o que será.
- Las herramientas incorporadas en Enterprise Architect que ayudan a administrar la complejidad incluyen:
 - Diagramas para modelar conceptos de nivel estratégicos y de negocio
 - Perfiles específicos del dominio y patrones del modelo reutilizables
 - Administración de versiones y líneas base para rastrear e integrar cambios
 - Seguridad basada en roles para ayudar a las personas correctas a contribuir en la forma correcta

© JMA 2019. All rights reserved

Documentación

- Enterprise Architect provee generación de documentos y herramientas de reporte con un editor de plantilla WYSIWYG completo.
- Genera informes detallados con la información que se necesita en el formato que la compañía o cliente demanda
- Al hacer clic en un botón produce versiones HTML automáticamente del modelo para una fácil distribución en Internet o la red interna de la organización.

© JMA 2019. All rights reserved

Código fuente de ida y vuelta

- El editor de código fuente incorporado le permite navegar rápidamente desde el modelo directamente al código fuente en el mismo entorno. Las plantillas de generación de código le permiten personalizar el código fuente generado de acuerdo con las especificaciones de su empresa.
- Enterprise Architect admite la generación y la ingeniería inversa del código fuente para muchos lenguajes populares, que incluyen:
 - ActionScript, Ada, C y C ++, DO#, Java, Delphi, Verilog, PHP, VHDL, Python, System C, VB.Net, Visual Basic y más
- Enterprise Architect admite transformaciones avanzadas de Arquitectura controlada por modelo (MDA) utilizando plantillas de transformación fáciles de editar.

© JMA 2019. All rights reserved

Compilar, probar y ejecutar

- Enterprise Architect le permite compilar, probar, depurar, operar y ejecutar scripts de despliegue, todo dentro del entorno de desarrollo de Enterprise Architect. Con la habilidad de generar clases de prueba NUnit y JUnit desde las clases de origen usando transformaciones MDA e integrar el proceso de prueba directamente en Enterprise Architect IDE, ahora pueden integrar UML y el modelado en el proceso de compilar/probar/ejecutar/desplegar.
- Además de las funcionalidades de ejecución y compilar/probar, Enterprise Architect incluye las capacidades de depurar para Java, .NET y Microsoft Native (C++, C y VB). Los depuradores construidos en Enterprise Architect se diseñan específicamente para permitir al desarrollador o evaluador capturar la información del trazo de pila mientras 'recorre' el código en ejecución, realizando inspección en tiempo de ejecución de hilos suspendidos.
- De esta manera pueden usar el historial del trazo de pila final para generar diagramas de secuencia dentro de Enterprise Architect, convirtiendo la ejecución del código actual y las llamadas en diagramas visuales.

© JMA 2019. All rights reserved

Modelado de base de datos

- Las herramientas de modelado de datos y perfiles incorporados de Enterprise Architect extienden UML para proporcionar un trazado intuitivo desde los conceptos de base de datos de tablas y relaciones hasta los conceptos UML de clases y asociaciones. Estas extensiones también permiten modelar claves de base de datos, disparadores, restricciones, RI y otras características de base de datos relacional.
- Enterprise Architect soporta el modelado de esquemas de base de datos y la generación automática de scripts DDL para once destinos DBMS listos para usar:
 - DB2
 - Firebird
 - MS Access
 - MySQL
 - MS SQL Server
 - Oracle
 - PostgreSQL

© JMA 2019. All rights reserved

Ingeniería y simulación de sistemas

- Integrando muchas características de alta calidad para ingenieros de sistemas, las ediciones de Ingeniería de sistemas y Ultimate de Enterprise Architect proveen soporte incorporado para:
 - SysML 1.1, 1.2, 1.3
 - Simulación del modelo paramétrico
 - Generación de código ejecutable
 - Modelo para transformaciones de código para lenguajes de descripción de Hardware y ADA 2005

© JMA 2019. All rights reserved

Modelado del proceso de negocio

- Hay muchos enfoques para el Modelado del proceso de negocio (BPM) usando UML como el lenguaje de modelado base.
- En particular, los diagramas de Actividad, diagramas de Objeto y Perfiles personalizados proveen un poderoso enfoque de modelado para los Analistas de negocio.
- Enterprise Architect complementa UML 2.5 con soporte BPMN y elementos de extensión para análisis, administración de requisitos y administración del proceso (como cambio, característica y elementos de incidencia).
- Las capacidades de modelado del proceso de negocio de Enterprise Architect le permiten:
 - Visualizar el proceso de negocio con un perfil UML para BPMN
 - Generar scripts BPEL ejecutables desde modelos BPMN
 - Validar la corrección de modelos BPMN

© JMA 2019. All rights reserved

Administración del proyecto

- Los administradores del proyecto pueden usar Enterprise Architect para:
 - Asignar recursos a los elementos
 - Medir riesgo y esfuerzo
 - Estimar tamaño del proyecto y complejidad
 - Implementar control de cambios y procedimientos de mantenimiento
- La capacidad de Métricas de casos de uso de Enterprise Architect hace fácil acceder a la complejidad del proyecto, basado en el número y tipo de casos de uso dentro del modelo, el tipo de proyecto y capacidades del entorno de desarrollo.
- Con experiencia, las métricas de casos de uso proveen una gran forma para acceder rápidamente al alcance de un proyecto

© JMA 2019. All rights reserved

INSTALACIÓN

© JMA 2019. All rights reserved

Ediciones

- Professional (Starter Edition)
 - Un entorno de modelado UML con todas las funciones para grupos de trabajo, analistas y desarrolladores. Incluye soporte para proyectos multiusuario e ingeniería de código (importación / exportación / sincronización).
- Corporate (Enterprise Workhorse)
 - Incluye todas las características de la versión Professional, con la capacidad adicional de usar una variedad de DBMS para repositorios de modelos compartidos. También incluye seguridad basada en roles, auditoría y herramientas de fusión de línea base para administrar usuarios concurrentes. Esta edición es para equipos más grandes que requieren un control detallado en entornos de modelado compartidos.
- Unified (Power Tools & Frameworks)
 - Para profesionales y equipos de desarrollo de negocios, sistemas y software que trabajan en tiempo real, soluciones integradas y de sistemas utilizando UML, SysML, DDS, DMN, BPSim y tecnologías relacionadas, esta edición tiene características y herramientas integradas para ayudar a entregar la mejor solución posible.
- Ultimate (Access All Areas)
 - Para los usuarios avanzados y aquellos que trabajan en múltiples dominios, Ultimate Edition ofrece la experiencia completa de Enterprise Architect. Un paquete completo, Ultimate Edition contiene todas las características y el software incluido de las otras ediciones.
- Lite (read only)
 - Es una edición gratuita de solo lectura de Enterprise Architect que las personas, como los patrocinadores del proyecto, pueden usar para revisar el proyecto sin realizar ningún cambio.
- <https://sparxsystems.com/products/ea/compare-editions.html>

© JMA 2019. All rights reserved

Requisitos del sistema

- Windows
 - Microsoft® Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Vista, Windows Server 2016, Windows Server 2012 or Windows Server 2008. (32 bit and 64 bit operating systems supported *)
 - 2GB of RAM
 - 800MB of available hard-disk space
 - 1280*720 or higher resolution display
- Linux
 - Linux Operating System (kernel 2.4 or later)
 - Wine ** 1.8 (minimum), 1.9x or later (recommended) (Or an equivalent CrossOver version)
 - Microsoft Data Access Components (MDAC) 2.8
 - 2GB of RAM
 - 800MB of available hard-disk space
 - 1280*720 or higher resolution display

© JMA 2019. All rights reserved

Requisitos del sistema

- Mac OS X
 - Mac OS 10.8 or later
 - Wine 1.8 (minimum) (Or CrossOver Mac 15.0.1, or later)
 - Microsoft Data Access Components (MDAC) 2.8
 - Intel® based processor
 - 2GB of RAM (128 MB or higher recommended)
 - 800MB of available hard-disk space
 - 1280*720 or higher resolution display
- Bases de datos compatibles (DBMS) para usar como repositorio de modelos de Enterprise Architect (Para alojar un modelo en un repositorio DBMS basado en servidor requiere una edición corporativa o superior)
 - Firebird (*.feap)
 - MariaDB
 - MySQL
 - MS SQL Server (including Express and Azure)
 - MS Access (*.eap)
 - Oracle
 - PostgreSQL

© JMA 2019. All rights reserved

Instalación

- Descargue el último instalador para la edición de prueba del sitio web de Sparx Systems.
 - <https://sparxsystems.com/products/ea/trial/request.html>
- Cuando finalice la descarga, haga doble clic en el archivo instalador para comenzar la instalación.
- Haga clic en el botón Siguiente .
- Lea el Acuerdo de licencia, haga clic en la casilla 'Acepto los términos del Acuerdo de licencia' y haga clic en el botón Siguiente.
- Lea la 'Información del archivo Léame' y haga clic en el botón Siguiente.
- Haga clic en el botón Siguiente para instalar la edición de prueba en la ubicación predeterminada. Si necesita instalar en una ubicación especial, haga clic en el botón Cambiar y proporcione una ruta diferente, haga clic en el botón Aceptar y luego en el botón Siguiente.
- Haga clic en el botón Instalar y observe que la barra 'Estado' indica el progreso de la instalación de Enterprise Architect.
- El diálogo cambia automáticamente.
- Haga clic en el botón Finalizar y verifique que el icono de Enterprise Architect se muestre en el escritorio de su estación de trabajo.

© JMA 2019. All rights reserved

Persistencia de los proyectos

- Los modelos de Enterprise Architect se almacenan en bases de datos relacionales estándar. Este enfoque se ha incorporado a Enterprise Architect desde la primera versión.
- El uso de una base de datos relacional proporciona una gran ventaja sobre otras herramientas que tienen un alcance y una capacidad limitados al depender de un sistema de almacenamiento de modelo basado en un archivo de texto simple.
- Con soporte para dos formatos de repositorio basados en archivos (MS JET y Firebird) más una serie de servidores DBMS estándar de la industria (como SQL Server, MySQL y Oracle) más almacenamiento basado en la nube, Enterprise Architect proporciona acceso rápido, escalable, flexible y transparente a los modelos en una amplia gama de escenarios de implementación.

© JMA 2019. All rights reserved

Repositorios basados en servidor

- A medida que el repositorio de Enterprise Architect se hace más grande, o el número de usuarios concurrentes aumenta o la política de la organización lo dicta, podría ser más apropiado usar un sistema de administración de bases de datos (DBMS) para almacenar el repositorio. Para usar un repositorio basado en DBMS, se deberá usar las ediciones Corporate, Unified o Ultimate. Enterprise Architect tiene una función para transferir un repositorio basado en archivos a uno basado en servidor.
- Para configurar un proyecto en un repositorio DBMS:
 - Configurar el software DBMS y crear una base de datos.
 - Crear las tablas requeridas en su base de datos, ejecutando un script suministrado en el sitio web de Sparx Systems.
 - <https://www.sparxsystems.com/resources/repositories/index.html>
 - (Para ciertos productos DBMS) Configurar un controlador ODBC para permitir la conexión al repositorio.
 - Conectarse al repositorio.

© JMA 2019. All rights reserved

INTERFAZ DE USUARIO

© JMA 2019. All rights reserved

Herramientas de interfaz de usuario

- **Ribbon**
 - Enterprise Architect muestra un conjunto de cintas en la parte superior de la pantalla, como una forma moderna y contemporánea de acceder a las potentes funciones de la herramienta. Cada cinta proporciona acciones con un enfoque particular, y los comandos en las cintas se agrupan por función en paneles.
 - **Portales**
 - Los portales proporcionan una agrupación lógica de alto nivel de las herramientas e instalaciones de Enterprise Architect, para formar un punto de entrada en características específicas que puede necesitar aplicar. Proporcionan acceso a herramientas comunes, búsquedas personalizadas, diseños de ventanas y Perspectivas. Esto facilita la configuración de un entorno de trabajo dentro del sistema y lleva a cabo sus tareas en una secuencia fluida.
 - **Barras de herramientas**
 - Enterprise Architect proporciona un conjunto de barras de herramientas que contienen accesos directos convenientes a una amplia gama de funciones comunes que puede realizar en su proyecto como un todo a través de las unidades de modelado individuales. Las barras de herramientas se muestran inicialmente en el banner de la barra de herramientas en la parte superior del área de trabajo, pero puede mostrarlas al pie del área de trabajo, arrastrarlas y acoplarlas dentro del área de trabajo, o flotarlas sobre la aplicación. Esto es útil cuando utiliza un determinado conjunto de funciones en un área en particular. También puede acoplar barras de herramientas al borde del área de trabajo de Enterprise Architect arrastrándolas por la barra de título y colocándolas contra el borde apropiado.
-

© JMA 2019. All rights reserved

Herramientas de interfaz de usuario

- Windows
 - A medida que se trabaja en el proyecto, creando y revisando estructuras de modelos, se puede utilizar una amplia gama de ventanas del sistema que ayudan a localizar y enfocarse en tipos particulares de objetos o propiedades de objetos. Para facilitar aún más sus tareas de modelado, puede arreglar estas ventanas en un lugar conveniente de su pantalla, moverlas, configurarlas como pestañas invocables o ocultarlas automáticamente. Se puede acceder a estas ventanas a través de las opciones de cinta o mediante combinaciones de teclas del teclado.
- Perspectivas
 - Enterprise Architect proporciona una amplia gama de lenguajes y tecnologías de modelado integrados, y puede llevar tiempo identificar la herramienta o combinación de herramientas correctas para cumplir con su requisito o escenario exacto de modelado. La función Perspectivas ayuda a filtrar rápida y efectivamente la capacidad de modelado general a un conjunto de tecnologías que son relevantes para un dominio.
- Espacios de trabajo
 - A medida que realiza el trabajo en su modelo, abre varias ventanas, diagramas y vistas, y probablemente necesitará volver a estas mismas vistas en una sesión de trabajo posterior. Al final de su sesión de trabajo, puede capturar su entorno de trabajo actual como un conjunto de trabajo, que utiliza para volver a ese entorno exacto más tarde, ya sea cuando lo desee o haciendo que el sistema lo vuelva a crear automáticamente la próxima vez que inicie sesión en Arquitecto Empresarial.

© JMA 2019. All rights reserved

Modelado

- El modelado es la práctica de construir una representación gráfica de algo real con el propósito de estudiar, documentar, razonar, probar o comunicarlo a otros.
- Los modelos resultantes se pueden comparar con las máquinas que funcionan y, a menudo, proporcionan una visión poderosa de la forma en que se estructuran o se comportan las cosas en el mundo real.
- Enterprise Architect se usa generalmente para modelar sistemas complejos de empresas, negocios, ingeniería, tiempo real y software, y es una plataforma multidisciplinaria que facilita a los equipos de personas que trabajan juntas para describir, visualizar, construir y probar sistemas complejos mediante el uso de modelos.
- La plataforma de modelado de Enterprise Architect se basa en el lenguaje de modelado unificado (UML),

© JMA 2019. All rights reserved

Componentes estructurales

- **Proyecto**
 - Un proyecto es un archivo único o almacenamiento basado en repositorio para uno o más modelos.
- **Modelos**
 - Los modelos son el nivel superior de un proyecto de Enterprise Architect. Un proyecto puede tener uno o varios modelos contenidos en él.
- **Vistas**
 - Las vistas son el segundo nivel dentro de un modelo y definen un punto de vista específico del sistema que se está modelando, por ejemplo, una vista de Caso de uso, una Vista de requisitos o una Vista dinámica (de comportamiento). Las vistas son simplemente paquetes que tienen un significado conceptual adicional.
- **Paquetes**
 - Un paquete es un contenedor de elementos del modelo y se representa en la ventana del navegador como un icono de "carpeta".
 - Un paquete contiene los bloques de construcción del modelo, como diagramas, elementos y otros paquetes.

© JMA 2019. All rights reserved

Componentes estructurales

- **Diagramas**
 - Un diagrama es una representación visual de los elementos del modelo, sus atributos y características, cómo están conectados y / o interactúan entre sí
 - Los diferentes tipos de diagramas muestran diferentes aspectos del modelo y las relaciones entre los elementos.
- **Elementos**
 - Los elementos son los componentes básicos de los modelos. Representan tanto construcciones estructurales como clases e interfaces, como construcciones de comportamiento como actividades, acciones y estados.
- **Conectores**
 - Los conectores representan los diversos tipos de relación entre elementos dentro de un modelo, incluidas las relaciones de comportamiento, asociaciones, relaciones taxonómicas y similares.

© JMA 2019. All rights reserved

Componentes estructurales

- Valores etiquetados
 - Los valores etiquetados son propiedades personalizables, definidas por el perfil o por el usuario que generalmente están relacionadas con un estereotipo y definen atributos y características adicionales de un elemento. Son uno de los medios fundamentales para extender UML a áreas más específicas de dominio.
- Notas
 - Las notas son las descripciones estándar comunes basadas en el lenguaje de para qué sirve un elemento, diagrama, característica o relación y cómo se usa dentro del modelo. Las notas a menudo se usan como un primer paso para obtener el significado y el uso de un elemento, que luego se refina en algo más concreto y especificado con precisión.
- Información de referencia
 - Una variedad de tipos básicos de información que se utiliza en un modelo particular, por ejemplo, estereotipos o valores etiquetados.

© JMA 2019. All rights reserved

UML

© JMA 2019. All rights reserved

Introducción

- El lenguaje de modelado unificado (UML) se ha convertido rápidamente en el estándar de facto para construir software orientado a objetos.
- La especificación de Object Management Group (OMG) establece:
 - *"El lenguaje de modelado unificado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo en software. El UML ofrece una forma estándar de escribir los planos de un sistema, incluyendo elementos conceptuales como procesos comerciales y funciones del sistema, así como cosas concretas, como declaraciones de lenguaje de programación, esquemas de bases de datos y componentes de software reutilizables".*
- El punto importante a tener en cuenta aquí es que UML es un 'lenguaje' para especificar y no un método o procedimiento. El UML se usa para definir un sistema de software; para detallar los artefactos en el sistema, para documentar y construir: es el lenguaje en el que está escrito el plan. El UML se puede usar de varias maneras para soportar una metodología de desarrollo de software (como el Proceso Racional Unificado), pero en sí mismo no especifica esa metodología o proceso.

© JMA 2019. All rights reserved

Modelos

- UML define la notación y la semántica para los siguientes dominios:
 - El modelo de interacción de usuario o casos de uso: describe los límites y la interacción entre el sistema y los usuarios. Corresponde en algunos aspectos a un modelo de requisitos.
 - El modelo de interacción o comunicación: describe cómo los objetos en el sistema interactuarán entre sí para realizar el trabajo.
 - El estado o modelo dinámico: los gráficos de estado describen los estados o condiciones que las clases asumen con el tiempo. Los gráficos de actividad describen los flujos de trabajo que implementará el sistema.
 - El modelo lógico o de clase: describe las clases y los objetos que conformarán el sistema.
 - El modelo de componentes físicos: describe el software (y, a veces, los componentes de hardware) que componen el sistema.
 - El modelo de implementación física: describe la arquitectura física y la implementación de componentes en esa arquitectura de hardware.

© JMA 2019. All rights reserved

Modelos de comportamiento UML

- Los diagramas de comportamiento de UML representan los elementos de un sistema que dependen del tiempo y que transmiten los conceptos dinámicos del sistema y cómo se relacionan entre sí.
- Los elementos en estos diagramas se parecen a los verbos en un lenguaje natural y las relaciones que los conectan típicamente transmiten el paso del tiempo.
- Por ejemplo, un diagrama de comportamiento de un sistema de reserva de vehículos puede contener elementos como hacer una reserva, alquilar un automóvil y proporcionar detalles de la tarjeta de crédito.
- Los modeladores experimentados mostrarán la relación con los elementos estructurales en estos diagramas.

© JMA 2019. All rights reserved

Modelos de comportamiento UML

- Diagramas de casos de uso
 - Los diagramas de casos de uso capturan los casos de uso y las relaciones entre los actores y el sistema; describen los requisitos funcionales del sistema, la forma en que los operadores externos interactúan en los límites del sistema y la respuesta del sistema.
- Diagramas de actividad
 - Los diagramas de actividad modelan los comportamientos de un sistema y la forma en que estos comportamientos están relacionados en un flujo general del sistema.
- Diagramas de StateMachine
 - Los diagramas de StateMachine ilustran cómo un elemento puede moverse entre estados, clasificando su comportamiento de acuerdo con los desencadenantes de transición y los guardias de restricción.
- Diagramas de tiempo
 - Los diagramas de tiempo definen el comportamiento de diferentes objetos dentro de una escala de tiempo, proporcionando una representación visual de objetos que cambian de estado e interactúan con el tiempo.

© JMA 2019. All rights reserved

Modelos de comportamiento UML

- Diagramas de secuencia
 - Los diagramas de secuencia son representaciones estructuradas del comportamiento como una serie de pasos secuenciales a lo largo del tiempo. Se utilizan para representar el flujo de trabajo, el paso de mensajes y cómo los elementos en general cooperan con el tiempo para lograr un resultado.
- Diagramas de comunicación
 - Los diagramas de comunicación muestran las interacciones entre elementos en tiempo de ejecución, visualizando las relaciones entre objetos.
- Diagramas generales de interacción
 - Los diagramas de descripción general de interacción visualizan la cooperación entre los diagramas de interacción (sincronización, secuencia, comunicación y otros diagramas de descripción general de interacción) para ilustrar un flujo de control que cumple un propósito global.

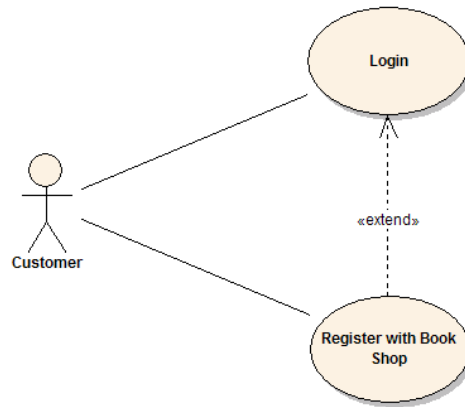
© JMA 2019. All rights reserved

Modelo de caso de uso

- Un modelo de caso de uso describe la funcionalidad propuesta de un nuevo sistema.
- Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema.
- Esta interacción es una sola unidad de trabajo significativo, como Crear cuenta o Ver detalles de la cuenta.
- Cada caso de uso describe la funcionalidad que se construirá en el sistema propuesto, que puede incluir la funcionalidad de otro caso de uso o extender otro caso de uso con su propio comportamiento.

© JMA 2019. All rights reserved

Modelo de caso de uso



© JMA 2019. All rights reserved

Descripción de un caso de uso

- **Comentarios** generales y notas que describen el caso de uso.
- **Requisitos:** los requisitos funcionales formales de las cosas que un caso de uso debe proporcionar al usuario final. Estos corresponden a las especificaciones funcionales que se encuentran en las metodologías estructuradas y forman un contrato en el que el caso de uso realiza alguna acción o proporciona algún valor al sistema.
- **Restricciones:** las reglas y limitaciones formales con las que opera un caso de uso, que definen qué se puede y qué no se puede hacer. Éstas incluyen:
 - Condiciones previas que ya deben haberse producido o estar vigentes antes de ejecutar el caso de uso
 - Condiciones posteriores que deben ser ciertas una vez que se complete el caso de uso
 - Invariantes que siempre deben ser ciertas a lo largo del tiempo que opera el caso de uso

© JMA 2019. All rights reserved

Descripción de un caso de uso

- **Escenarios:** descripciones formales y secuenciales de los pasos tomados para llevar a cabo el caso de uso, o el flujo de eventos que ocurren durante una instancia de caso de uso. Estos pueden incluir múltiples escenarios, para atender circunstancias excepcionales y rutas de procesamiento alternativas. Por lo general, se crean en texto y corresponden a una representación textual del diagrama de secuencia.
- **Diagramas de escenarios:** diagramas de secuencia para representar el flujo de trabajo; similar a los escenarios pero representado gráficamente.
- **Atributos adicionales,** como fase de implementación, número de versión, clasificación de complejidad, estereotipo y estado.

© JMA 2019. All rights reserved

Elementos del caso de uso

- Los casos de uso generalmente están relacionados con los 'actores', que son entidades humanas o de máquinas que usan o interactúan con el sistema para realizar un trabajo significativo que les ayuda a lograr un objetivo. El conjunto de casos de uso a los que un actor tiene acceso define su rol general en el sistema y el alcance de su acción.
- Un Caso de Uso puede incluir la funcionalidad de otro como parte de su procesamiento normal. Generalmente se asume que los casos de uso incluidos se llamarán cada vez que se ejecute el camino base. Un ejemplo puede ser listar un conjunto de órdenes de clientes de las cuáles poder elegir antes de modificar una orden seleccionada; en este caso, el Caso de Uso <listar órdenes> se puede incluir en el Caso de Uso <modificar orden> cada vez que éste se ejecute.
- Un caso de uso puede ser incluido por uno o más casos de uso, por lo que ayuda a reducir la duplicación de la funcionalidad al factorizar el comportamiento común en casos de uso que se reutilizan muchas veces.
- Un caso de uso puede extender el comportamiento de otro, generalmente cuando se encuentran circunstancias excepcionales. Por ejemplo, si un usuario debe obtener la aprobación de alguna autoridad superior antes de modificar un tipo particular de pedido de un cliente, entonces el Caso de uso <obtener aprobación> podría extender opcionalmente el Caso de uso normal de <modificar pedido>.

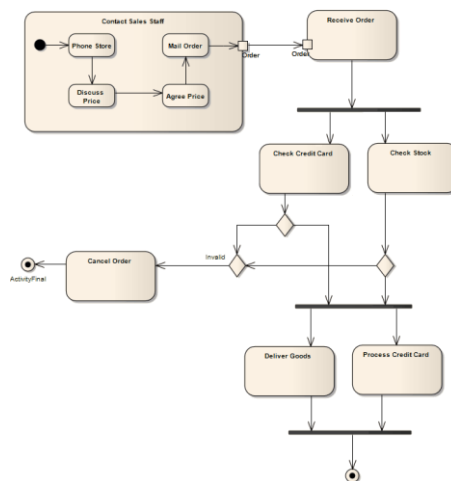
© JMA 2019. All rights reserved

Diagrama de actividad

- Los diagramas de actividad se utilizan para modelar los comportamientos del sistema y la forma en que estos comportamientos se relacionan en un flujo general del sistema (es decir, las interacciones de elementos dinámicos).
- Las rutas lógicas que sigue un proceso, basadas en diversas condiciones, procesamiento concurrente, acceso a datos, interrupciones y otras distinciones de rutas lógicas, se utilizan para construir un proceso, sistema o procedimiento.
- Los diagramas de actividad se utilizan para mostrar cómo se construyen los diferentes flujos de trabajo en el sistema, cómo comienzan y las posibles rutas de decisión que se pueden tomar de principio a fin.
- También pueden ilustrar dónde puede ocurrir el procesamiento paralelo en la ejecución de algunas actividades.

© JMA 2019. All rights reserved

Diagrama de actividad



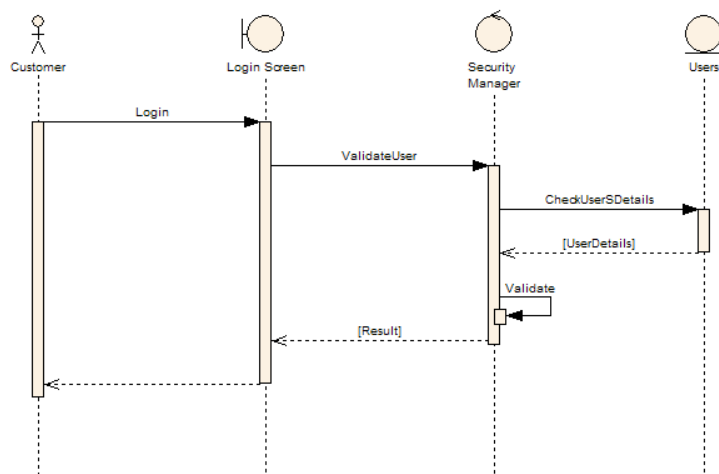
© JMA 2019. All rights reserved

Diagramas de secuencia

- Los diagramas de secuencia proporcionan una representación gráfica de las interacciones de objetos a lo largo del tiempo. Por lo general, muestran a un usuario o actor, y los objetos y componentes con los que interactúan en la ejecución de un caso de uso. Un diagrama de secuencia generalmente representa un solo 'escenario' de caso de uso o flujo de eventos.
- Los diagramas de secuencia son una excelente manera de documentar escenarios de uso y capturar los objetos requeridos al principio del análisis y verificar el uso de los objetos más adelante en el diseño. Los diagramas muestran el flujo de mensajes de un objeto a otro y, como tales, corresponden a los métodos y eventos admitidos por una clase / objeto.
- El siguiente ejemplo de diagrama de secuencia muestra al usuario o actor a la izquierda iniciando un flujo de eventos y mensajes que corresponden al escenario de Caso de uso. Los mensajes que pasan entre objetos se convierten en operaciones de clase en el modelo final.

© JMA 2019. All rights reserved

Diagramas de secuencia



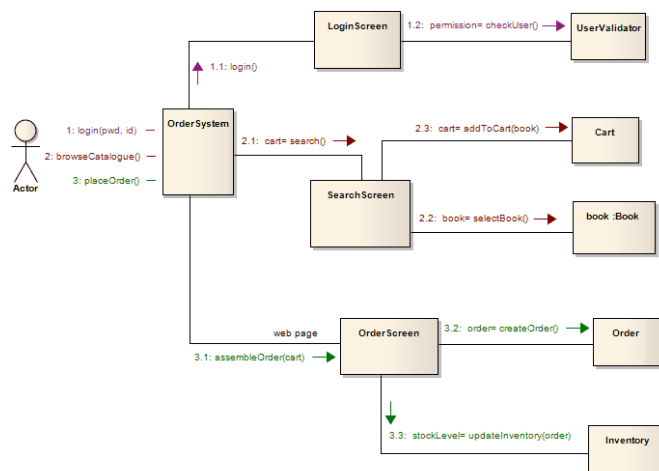
© JMA 2019. All rights reserved

Diagrama de comunicación

- Un diagrama de comunicación es un diagrama que muestra las interacciones entre elementos en tiempo de ejecución de la misma manera que un diagrama de secuencia.
- Sin embargo, los diagramas de comunicación se utilizan para visualizar las relaciones entre objetos, mientras que los diagramas de secuencia son más efectivos para visualizar el procesamiento a lo largo del tiempo.
- Los diagramas de comunicación emplean asociaciones ordenadas y etiquetadas para ilustrar el procesamiento. La numeración es importante para indicar el orden y el anidamiento del procesamiento.

© JMA 2019. All rights reserved

Diagrama de comunicación



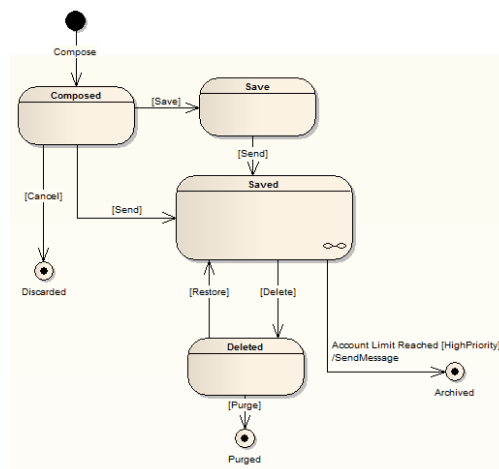
© JMA 2019. All rights reserved

StateMachines

- Las StateMachines (se conocía anteriormente como diagramas de estado) ilustran cómo un elemento (a menudo una Clase) puede moverse entre Estados, clasificando su comportamiento de acuerdo con los desencadenantes de transición y protectores de restricción.
- Los gráficos de estado se utilizan para detallar las transiciones o cambios de estado por los que puede pasar un objeto en el sistema. Muestran cómo se mueve un objeto de un estado a otro y las reglas que rigen ese cambio. Los gráficos de estado generalmente tienen una condición de inicio y fin.
- Una tabla StateMachine es una variante al diagrama StateMachine. Muestra la información de StateMachine en forma de tabla y es un método para especificar el comportamiento discreto de un sistema de transición de estado finito; es decir, a qué estado se mueve StateMachine y las condiciones bajo las cuales tiene lugar la transición.

© JMA 2019. All rights reserved

Diagrama StateMachine



© JMA 2019. All rights reserved

Tabla StateMachine

State \ Trigger		Event1	Event2	Event3	Event4	<None>
		E0	E1	E2	E3	E4
Initial	S0					S1
State1	S1				S2	
State2	S2	S6	[Guard] S4			
	SubState1	S3	S4			
	SubState2	S4		[Cond] S2		
	SubState3	S5				
State3	S6					S7
Final	S7					

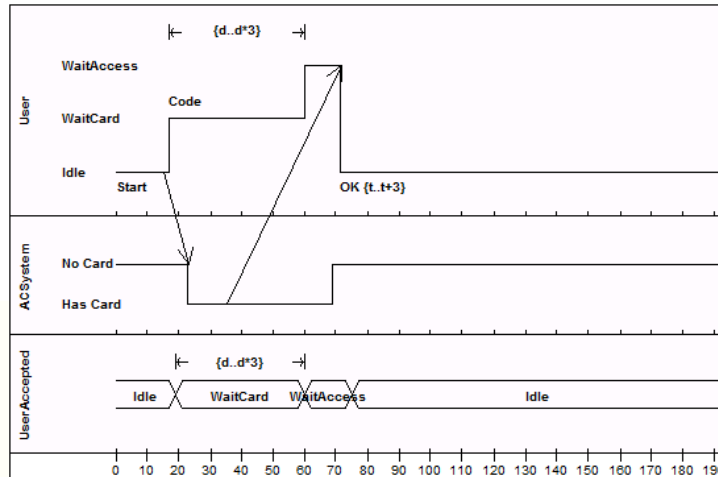
© JMA 2019. All rights reserved

Diagrama de tiempo

- Un diagrama de tiempo define el comportamiento de diferentes objetos dentro de una escala de tiempo. Proporciona una representación visual de objetos que cambian de estado e interactúan con el tiempo. Puedes usarlo para:
 - Definir componentes de software integrados o basados en hardware; por ejemplo, los utilizados en un sistema de inyección de combustible o un controlador de microondas
 - Especificar procesos de negocio basados en el tiempo.

© JMA 2019. All rights reserved

Diagrama de tiempo



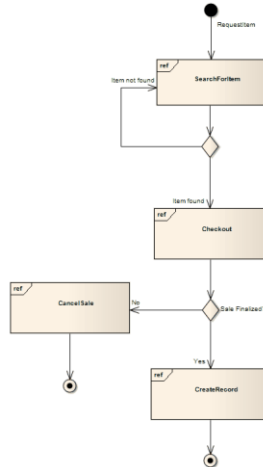
© JMA 2019. All rights reserved

Diagrama general de interacción

- Los diagramas de visión general de interacción visualizan la cooperación entre otros diagramas de interacción para ilustrar un flujo de control que cumple un propósito global. Como los diagramas de descripción general de interacción son una variante de los diagramas de actividad, la mayor parte de la notación del diagrama es la misma, al igual que el proceso de construcción del diagrama.
- Los puntos de decisión, las bifurcaciones, las uniones, los puntos de inicio y los puntos de finalización son los mismos. Sin embargo, en lugar de elementos de actividad, se utilizan elementos rectangulares. Hay dos tipos de estos elementos:
 - Los elementos de interacción muestran un diagrama de interacción en línea, que puede ser cualquiera de los cuatro tipos (secuencia, sincronización, comunicación o descripción general de la interacción)
 - Los elementos de aparición de interacción son referencias a un diagrama de interacción existente: están representados visualmente por un marco, con ref en el espacio del título del marco; el nombre del diagrama se indica en el contenido del marco

© JMA 2019. All rights reserved

Diagrama general de interacción



© JMA 2019. All rights reserved

Modelos estructurales UML

- Los diagramas estructurales de UML representan los elementos de un sistema que son independientes del tiempo y que transmiten los conceptos de un sistema y cómo se relacionan entre sí.
- Los elementos en estos diagramas se parecen a los sustantivos en un lenguaje natural y las relaciones que los conectan siempre muestran relaciones estructurales o semánticas.
- Por ejemplo, un diagrama estructural de un sistema de reserva de vehículos puede contener elementos como Automóvil, Reserva y Tarjeta de crédito, y puede contener conectores que unen estos elementos.
- Los modeladores experimentados también mostrarán las relaciones con los elementos de comportamiento.

© JMA 2019. All rights reserved

Modelos estructurales UML

- Diagrama de clase
 - Los diagramas de clases capturan la estructura lógica del sistema, las clases y los objetos que componen el modelo, describiendo lo que existe y qué atributos y comportamiento tiene.
- Diagrama de estructura compuesta
 - Los diagramas de estructura compuesta reflejan la colaboración interna de clases, interfaces y componentes (y sus propiedades) para describir una funcionalidad.
- Diagrama de componentes
 - Los diagramas de componentes ilustran las piezas de software, los controladores integrados y los que componen un sistema, y su organización y dependencias.
- Diagrama de implementación
 - Los diagramas de implementación muestran cómo y dónde se implementará el sistema; es decir, su arquitectura de ejecución.
- Diagrama de objetos
 - Los diagramas de objetos representan instancias de objetos de clases y sus relaciones en un punto en el tiempo.
- Diagrama de paquetes
 - Los diagramas de paquetes representan la organización de los elementos del modelo en paquetes y las dependencias entre ellos.
- Diagrama de perfil
 - Los diagramas de perfil son aquellos creados en un paquete «profile», para extender elementos UML, conectores y componentes.

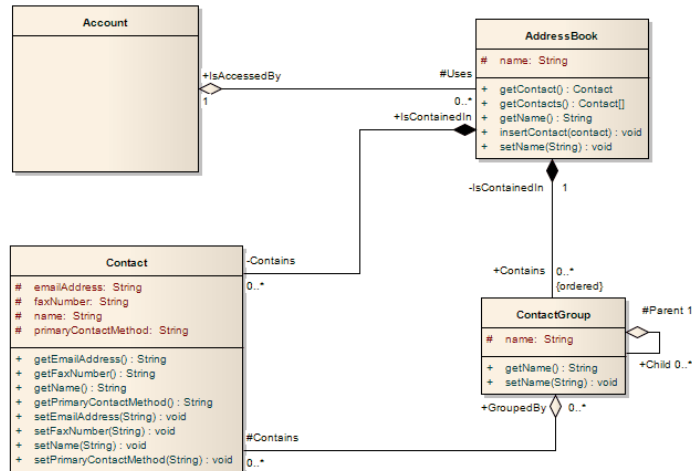
© JMA 2019. All rights reserved

Diagrama de clase

- El diagrama de clase captura la estructura lógica del sistema, las clases, y las cosas que componen el modelo.
- Es un modelo estático, que describe lo que existe y qué atributos y comportamiento tiene, en lugar de cómo se hace algo.
- En un diagrama de clase, se puede ilustrar las relaciones entre clases e interfaces utilizando generalizaciones, agregaciones y asociaciones, que son valiosas para reflejar la herencia, la composición o el uso, y las conexiones, respectivamente.

© JMA 2019. All rights reserved

Diagrama de clase



© JMA 2019. All rights reserved

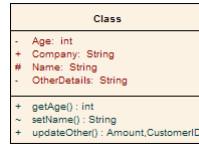
Modelo de clase

- Una clase es una construcción UML estándar utilizada para detallar el patrón a partir del cual se producirán los objetos en tiempo de ejecución. Una clase es una especificación: un objeto, una instancia de una clase.
- Las clases pueden heredarse de otras clases (es decir, heredan todo el comportamiento y el estado de sus padres y añaden nuevas funciones propias), tienen otras clases como atributos, delegan responsabilidades a otras clases e implementan interfaces abstractas.
- El modelo de clase está en el núcleo del desarrollo y diseño orientado a objetos: expresa tanto el estado persistente del sistema como el comportamiento del sistema. Una clase encapsula el estado (atributos) y ofrece servicios para manipular ese estado (comportamiento).
- Un buen diseño orientado a objetos limita el acceso directo a los atributos de clase y ofrece servicios que manipulan los atributos en nombre de la persona que llama.
- Esta ocultación de datos y exposición de servicios garantiza que las actualizaciones de datos solo se realicen en un lugar y de acuerdo con reglas específicas: para sistemas grandes, la carga de mantenimiento del código que tiene acceso directo a elementos de datos en muchos lugares es extremadamente alta.

© JMA 2019. All rights reserved

Modelo de clase

- La clase se representa a continuación:

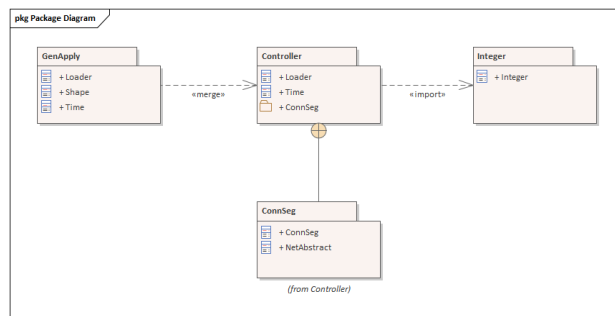


- Tenga en cuenta que la clase tiene tres áreas distintas:
 - El nombre de la clase (y estereotipo si se aplica)
 - El área de atributos de clase (es decir, elementos de datos internos)
 - El comportamiento, tanto privado como público.
- Los atributos y métodos pueden marcarse como
 - Privado, lo que indica que no son visibles para quienes llaman fuera de la clase
 - Protegidos, solo son visibles para los niños de la clase
 - Público, son visibles para todos

© JMA 2019. All rights reserved

Diagrama de paquete

- Los diagramas de paquetes representan la organización de los elementos del modelo en paquetes y las dependencias entre ellos, incluidas las importaciones de paquetes y las extensiones de paquetes. También proporcionan una visualización de los espacios de nombres correspondientes.



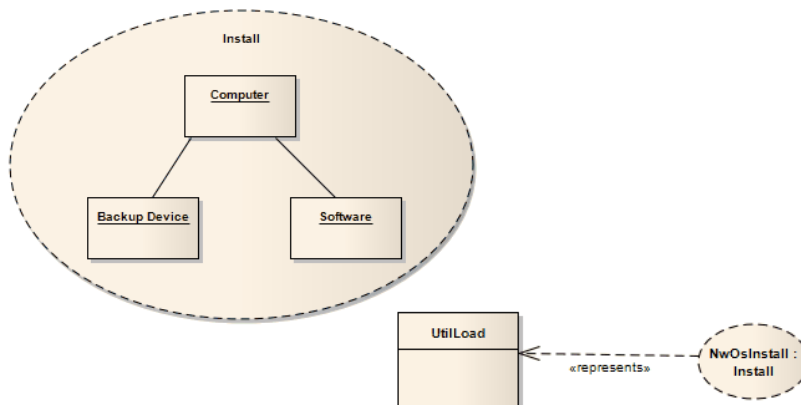
© JMA 2019. All rights reserved

Diagrama de estructura compuesta

- Un diagrama de estructura compuesta refleja la colaboración interna de clases, interfaces o componentes (y sus propiedades) para describir una funcionalidad. Los diagramas de estructura compuesta son similares a los diagramas de clase, pero mientras que los diagramas de clase modelan una vista estática de las estructuras de clase, incluidos sus atributos y comportamientos, los diagramas de estructura compuesta modelan un uso específico de la estructura. Puede usarlos para expresar arquitecturas de tiempo de ejecución, patrones de uso y las relaciones de los elementos participantes, que podrían no reflejarse en diagramas estáticos.
- En un diagrama de estructura compuesta, se accede a las clases como partes o instancias de tiempo de ejecución que cumplen una función particular. Estas partes pueden tener multiplicidad, si el rol desempeñado por la clase requiere múltiples instancias. Los puertos definidos por la clase de una parte deben representarse en la estructura compuesta, de modo que todas las partes conectadas proporcionen las interfaces requeridas especificadas por el puerto. Existe una gran flexibilidad, y la consiguiente complejidad, que viene con el modelado de estructuras compuestas. Para optimizar su modelado, considere crear colaboraciones para representar patrones reutilizables que respondan a sus problemas de diseño.

© JMA 2019. All rights reserved

Diagrama de estructura compuesta



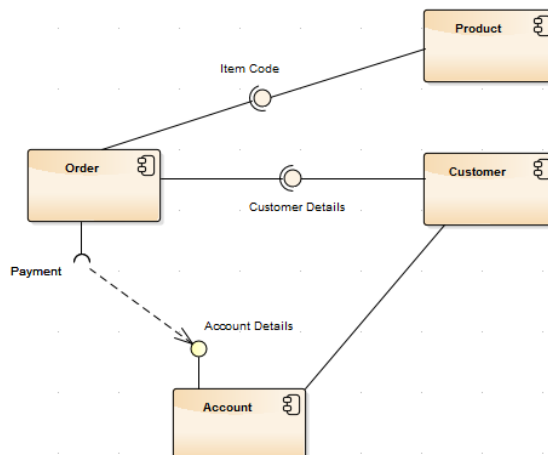
© JMA 2019. All rights reserved

Diagrama de componentes

- El modelo de componentes ilustra los componentes de software que se utilizarán para construir el sistema. Estos pueden ser construidos a partir del modelo de clase y escritos desde cero para el nuevo sistema, o pueden ser traídos de otros proyectos y proveedores de terceros. Los componentes son agregaciones de alto nivel de piezas de software más pequeñas y proporcionan un enfoque de bloques de construcción de 'caja negra' para la construcción de software.
- Un diagrama de componentes ilustra las piezas de software, los controladores integrados y los que componen un sistema, y su organización y dependencias. El diagrama de componentes muestra la relación entre los componentes de software, sus dependencias, comunicación, ubicación y otras condiciones.
- Un diagrama de componentes tiene un mayor nivel de abstracción que un diagrama de clase; generalmente un componente es implementado por una o más Clases (u Objetos) en tiempo de ejecución. Son bloques de construcción, contruidos para que eventualmente un componente pueda abarcar una gran parte de un sistema.

© JMA 2019. All rights reserved

Diagrama de componentes



© JMA 2019. All rights reserved

Diagrama de objeto

- Un diagrama de objeto está estrechamente relacionado con un diagrama de clase, con la distinción de que representa instancias de objetos de clases y sus relaciones en un momento determinado. Los diagramas de objetos no revelan arquitecturas que varían de sus diagramas de clase correspondientes, pero reflejan la multiplicidad y los roles que las clases instanciadas podrían servir. Son útiles para comprender un diagrama de clase complejo, al crear diferentes casos en los que se aplican las relaciones y las clases
- Esto puede parecer similar a un diagrama de Estructura compuesta, que también modela el comportamiento en tiempo de ejecución; la diferencia es que los diagramas de objetos ejemplifican los diagramas de clase estáticos, mientras que los diagramas de estructura compuesta reflejan arquitecturas de tiempo de ejecución diferentes de sus contrapartes estáticas.
- Un diagrama de objetos también puede ser un tipo de diagrama de comunicación (que también modela las conexiones entre objetos, pero además secuencia eventos a lo largo de cada ruta).

© JMA 2019. All rights reserved

Diagrama de objeto

Diagrama de clases

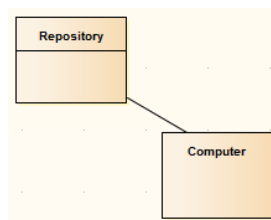
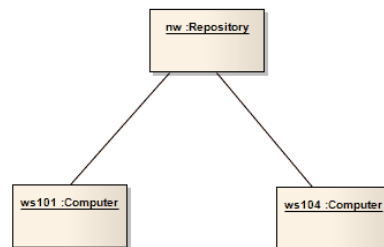


Diagrama de objetos



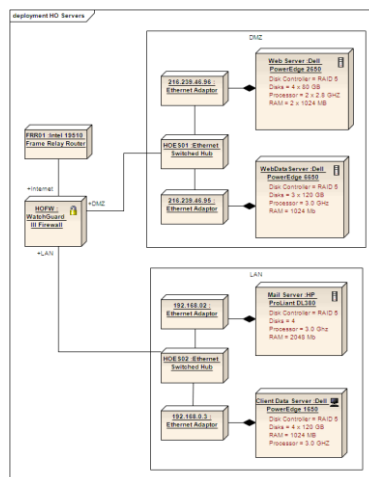
© JMA 2019. All rights reserved

Diagrama de implementación

- Un diagrama de implementación o despliegue muestra cómo y dónde se implementará el sistema; es decir, su arquitectura de ejecución.
- Los dispositivos de hardware, los procesadores y los entornos de ejecución de software (artefectos del sistema) se reflejan como nodos, y la construcción interna se puede representar incrustando o anidando nodos. Las relaciones de implementación indican la implementación de artefactos, y las relaciones de manifiesto revelan la implementación física de componentes. Como los artefactos se asignan a los nodos para modelar la implementación del sistema, la asignación se guía por el uso de especificaciones de implementación.
- Un diagrama de implementación también puede indicar que un Nodo tiene un Estado, o mostrar una instancia de un Nodo con un valor de tiempo de ejecución real para el estado, que representa una condición o escenario específico.

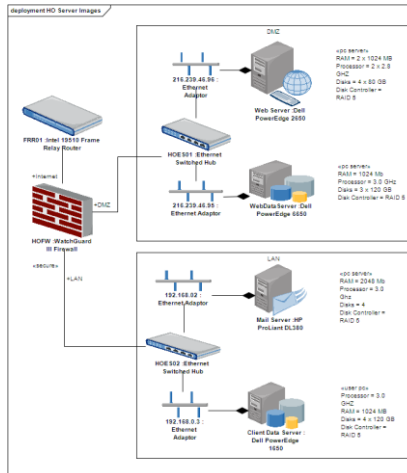
© JMA 2019. All rights reserved

Diagrama de implementación



© JMA 2019. All rights reserved

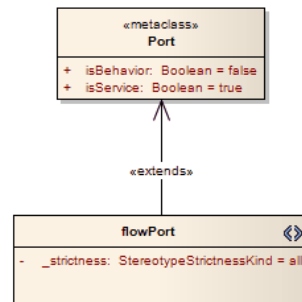
Diagrama de implementación



© JMA 2019. All rights reserved

Diagrama de perfil

- Un diagrama de perfil es cualquier diagrama creado en un paquete de «perfil».
- Los perfiles proporcionan un medio para extender el UML. Se basan en estereotipos adicionales y valores etiquetados que se aplican a elementos UML, conectores y sus componentes.
- Un perfil es una colección de tales extensiones que juntas describen algún problema particular de modelado y facilitan construcciones de modelado en ese dominio.



© JMA 2019. All rights reserved

MODELOS DE BASES DE DATOS

© JMA 2019. All rights reserved

Introducción

- El poder de la ingeniería basada en modelos es la capacidad de visualizar, analizar y diseñar todos los aspectos de un sistema. Poder ver y administrar información y datos junto con otros modelos de un sistema proporciona una gran claridad y reduce la posibilidad de error.
 - Enterprise Architect tiene un amplio soporte para la disciplina de modelado de datos, que va desde la representación de información en un modelo conceptual hasta la generación de objetos de base de datos.
 - Ya sea que esté generando objetos de base de datos desde el modelo UML o el DBMS heredado de ingeniería inversa en un modelo para análisis, las características de la herramienta le permitirán ahorrar tiempo y recursos valiosos del proyecto.
 - Enterprise Architect admite el modelado de información en las capas conceptuales, lógicas y físicas. Usando una serie de características estándar, estos modelos pueden interconectarse, proporcionando trazabilidad.
 - Los modelos lógicos y físicos también se pueden generar automáticamente utilizando un motor de transformación totalmente personalizable.
 - Los sistemas heredados pueden importarse, analizarse y compararse utilizando la potente instalación de ingeniería inversa.
-

© JMA 2019. All rights reserved

Modelo conceptual de datos

- Un modelo de datos conceptual es la forma más abstracta de modelo de datos. Es útil para comunicar ideas a una amplia gama de partes interesadas debido a su simplicidad. Por lo tanto, la información específica de la plataforma, como los tipos de datos, índices y claves, se omite de un modelo de datos conceptual. También se excluyen otros detalles de implementación, como procedimientos y definiciones de interfaz.
- Este es un ejemplo de un modelo de datos conceptual, representado con dos de las anotaciones admitidas por Enterprise Architect.



- Usando la notación Entidad-Relación (ER), representamos los conceptos de datos 'Clientes' y 'Direcciones de Clientes' como Entidades con una relación de 1 a muchos entre ellos. Podemos representar exactamente la misma información semántica usando las clases y asociaciones UML.

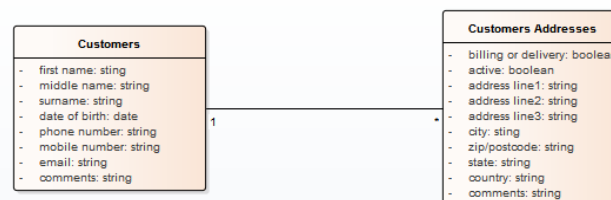


- El uso de notación UML o ER para representar conceptos de datos en su proyecto depende de la experiencia y las preferencias de las partes interesadas involucradas. La estructura detallada de los conceptos de datos ilustrados en un modelo de datos conceptual se define mediante el modelo de datos lógico.

© JMA 2019. All rights reserved

Modelo de datos lógicos

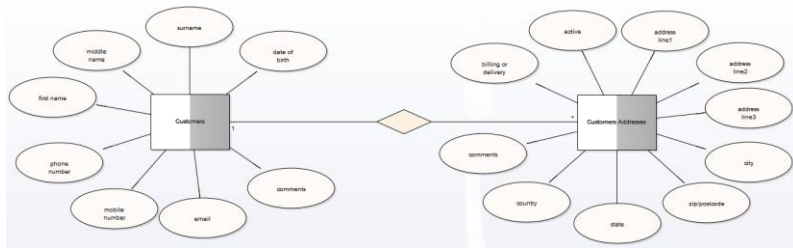
- Los modelos de datos lógicos ayudan a definir la estructura detallada de los elementos de datos en un sistema y las relaciones entre los elementos de datos. Refinan los elementos de datos introducidos por un modelo de datos conceptual y forman la base del modelo de datos físicos.
- En Enterprise Architect, un modelo de datos lógico generalmente se representa utilizando la notación de clase UML (Entidad).



© JMA 2019. All rights reserved

Diagramas de entidad-relación (ERD)

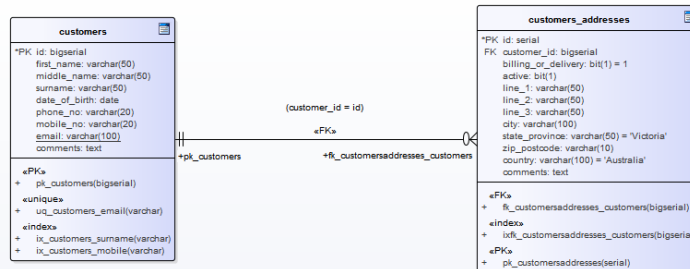
- Un modelo de entidad-relación (ERM) es una representación abstracta y conceptual de datos. El modelado de relación de entidad es un método de modelado de base de datos, utilizado para producir un tipo de esquema conceptual o modelo de datos semántico de un sistema, a menudo una base de datos relacional, y sus requisitos de forma descendente. Los diagramas creados por este proceso se denominan Diagramas de entidad-relación, Diagramas ER o ERD.
- Los diagramas de relación de entidad en Enterprise Architect se basan en los bloques de construcción ERD de Chen: las entidades (tablas) se representan como rectángulos, los atributos (columnas) se representan como elipses (unidos a su entidad) y las relaciones entre las entidades se representan como conectores en forma de diamante.



© JMA 2019. All rights reserved

Modelos de datos físicos

- Un modelo de datos físicos representa visualmente la estructura de datos implementada por un esquema de base de datos relacional. Además de proporcionar una abstracción visual de la estructura de la base de datos, un beneficio importante de definir un modelo de datos físicos es que puede derivar automáticamente el esquema de la base de datos del modelo.
- Esto es posible debido a la riqueza de metadatos capturados por un modelo de datos físicos y su asignación cercana a aspectos del esquema de la base de datos, como tablas de la base de datos, columnas, claves primarias y claves externas.



© JMA 2019. All rights reserved

Generar DDL

- Una vez que se ha definido un modelo físico y se han modelado los objetos, Enterprise Architect puede generar el lenguaje de definición de base de datos (DDL) para una variedad de objetos, incluyendo tablas de base de datos, vistas, funciones, secuencias y procedimientos.
- Este es un mecanismo que ahorra tiempo y reduce los errores que pueden introducirse al hacerlo manualmente en otras herramientas. La ingeniería avanzada se rige por un conjunto de plantillas que definen cómo las construcciones UML se convierten en los objetos en el DBMS objetivo. Se proporcionan plantillas estándar para todos los DBMS compatibles, y se pueden editar para personalizar la forma en que se genera el DDL. En el caso de que un DBMS no sea compatible de forma inmediata, se puede crear un nuevo conjunto de plantillas utilizando las existentes como punto de partida y referencia.
- Cuando se genera el DDL, la salida se puede dirigir a un archivo (o una serie de archivos, uno para cada objeto) o al motor de ejecución DDL. El motor de ejecución le permite ejecutar el DDL inmediatamente, apuntando a una base de datos en vivo a través de una conexión ODBC. Si dirige la salida a un archivo, puede ejecutar el DDL en una base de datos en vivo más adelante.

© JMA 2019. All rights reserved

Importar esquema de base de datos

- Enterprise Architect se comunica con el DBMS a través de ODBC, puede importar un esquema a través de esta conexión y puede sincronizar el modelo con la base de datos en vivo si se cambia.
- Una vez que el esquema está en Enterprise Architect, los objetos de la base de datos se pueden rastrear a otros elementos, asegurando la integridad del diseño y la arquitectura.
- Cuando los sistemas se dirigen a múltiples DBMS, todos estos pueden ser modificados en un modelo y los elementos y tipos de datos se pueden comparar entre estos modelos.
- El sofisticado motor de informes puede producir documentación de alta calidad, que incluye diccionarios de datos, diagramas y relaciones con otros modelos, como los requisitos de arquitectura e información, y en última instancia con los objetivos y los impulsores del negocio.

© JMA 2019. All rights reserved

MODELADO UX

© JMA 2019. All rights reserved

Introducción

- La experiencia del usuario y el diseño de la interfaz de usuario se han modelado tradicionalmente en una variedad de herramientas que están separadas de otras disciplinas, lo que lleva a una desconexión entre estos modelos y los otros modelos de análisis y tecnología.
 - Enterprise Architect permite modelar una amplia gama de interfaces y plataformas de usuario, incluido el software del cliente, sitios web y páginas, y dispositivos móviles como teléfonos y tabletas. Utiliza representaciones convincentes de los dispositivos físicos y las plataformas para hacer que estos modelos sean atractivos y útiles para los usuarios.
 - Los elementos en estos modelos también se pueden rastrear a otros elementos en el repositorio, incluidos los principios de diseño, los requisitos, los casos de uso y las historias de usuarios, las preocupaciones de las partes interesadas, los modelos de información, la arquitectura y los modelos de diseño.
 - Los diagramas de StateMachine también se pueden crear para representar los estados importantes de la interfaz de usuario, y estos se pueden rastrear hasta los modelos de prueba.
-

© JMA 2019. All rights reserved

Diagramas de interfaz de usuario

- El diagrama de interfaz de usuario es un tipo de diagrama extendido que proporciona un conjunto de cajas de herramientas de wireframe con una rica paleta de elementos de interfaz de usuario para dispositivos Android y Apple, así como para páginas web y cuadros de diálogo. También hay una función para modelar interfaces de usuario de Win32®, con una caja de herramientas que contiene una amplia gama de controles, como casillas de verificación, controles de giro, controles de árbol y muchos más.
- Un wireframe o prototipo no es más que un boceto donde se representa visualmente, de una forma muy sencilla y esquemática la estructura de una página web.
- Los dispositivos que puede modelar incluyen:
 - Teléfonos y tabletas Android
 - iPhones y iPad de Apple
 - Teléfonos Windows 8.1
 - Páginas web
 - Diálogos de pantalla

© JMA 2019. All rights reserved

Diseño del interfaz de usuario

- **Diseño de información**
 - El diseño de información es la ubicación en la presentación y priorización de información de manera que facilite el entendimiento. El diseño de información es un área del diseño gráfico pensada para representar información de manera efectiva para una comunicación clara. Para sitios web, los elementos informativos deben estar ordenados de manera que reflejen los objetivos y tareas del usuario.
- **Diseño de navegación**
 - El sistema de navegación provee un conjunto de elementos en pantalla que permiten al usuario moverse entre páginas del mismo sitio web. El diseño de navegación debe comunicar la relación entre los enlaces que contiene de forma que los usuarios entiendan las opciones que tienen para navegar el sitio. A menudo, los sitios web contienen múltiples sistemas de navegación, tales como navegación global, navegación local, navegación suplementaria, navegación contextual y navegación de cortesía.
 - Este diseño permite dar una vista de manera global, de la ubicación y desplazamiento a través de las diferentes páginas de un sitio web.
- **Diseño de interfaz**
 - El diseño de la interfaz de usuario incluye seleccionar y ordenar los elementos de la interfaz para permitir a los usuarios interactuar con la funcionalidad del sistema. El objetivo es facilitar la usabilidad y eficiencia tanto como sea posible. Elementos comunes que se encuentran en el diseño de interfaces son los botones de acción, campos de texto, cajas de verificación, botones radiales y menús desplegables.

© JMA 2019. All rights reserved

Ventajas de usar Wireframes

- Rápidos y baratos de crear: Como son bocetos esquemáticos son rápidos de crear y tienen un coste muy bajo. Esto permite realizar múltiples versiones hasta encontrar la adecuada sin que ello suponga un problema de tiempo o dinero.
- Detectar y corregir los problemas antes: Al ser sencillos y rápidos de realizar te permiten exponerlos rápidamente a feedback y resolver problemas básicos relacionados con la usabilidad y funcionalidades propuestas.
- Mejoras sencillas: En poco tiempo podrás mostrar los primeros Wireframes a compañeros, clientes o repasarlos tu mismo para repasar las mejoras que se puedan realizar en el diseño, el posicionamiento de los elementos o la estructura de los contenidos.
- Mejor usabilidad: Planear previamente la estructura y los elementos del interfaz de usuario te permitirá ofrecer una mejor usabilidad al no improvisar sobre la marcha y haber definido previamente estos elementos.

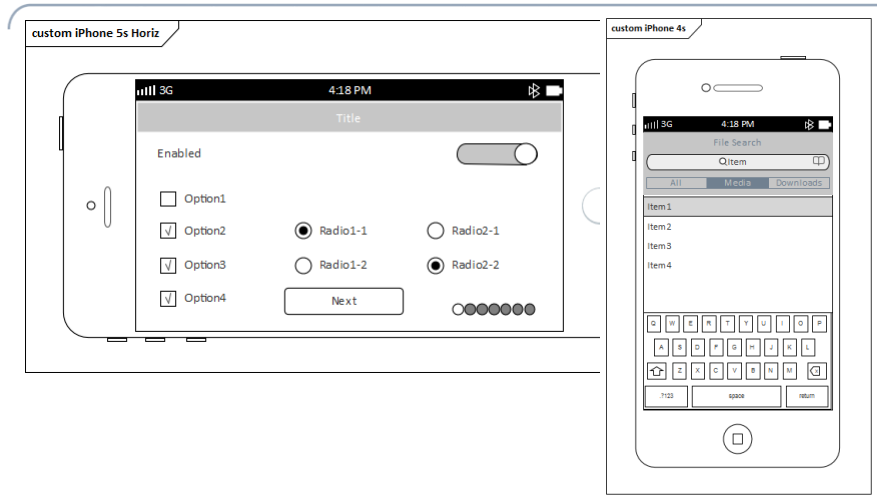
© JMA 2019. All rights reserved

Android Wireframe



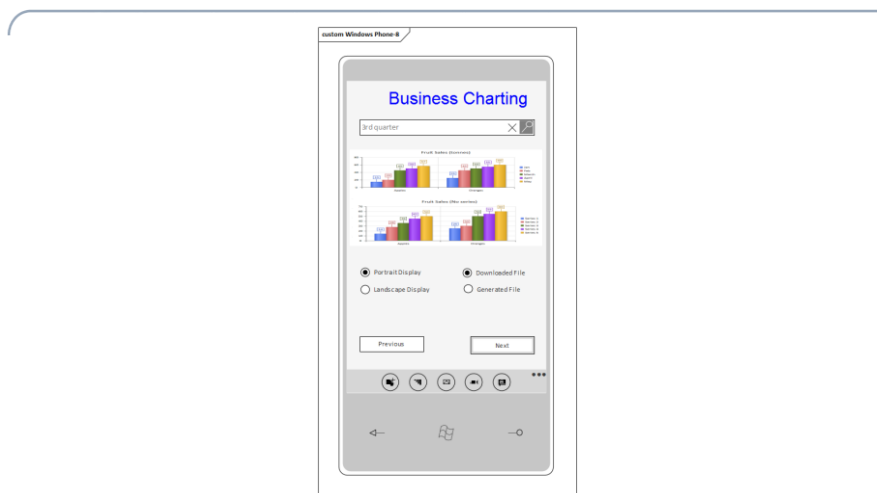
© JMA 2019. All rights reserved

iPhone / iPad Wireframe



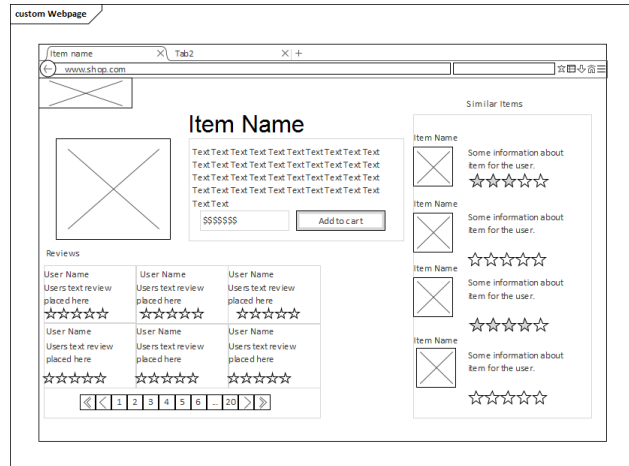
© JMA 2019. All rights reserved

Windows Phone Wireframe



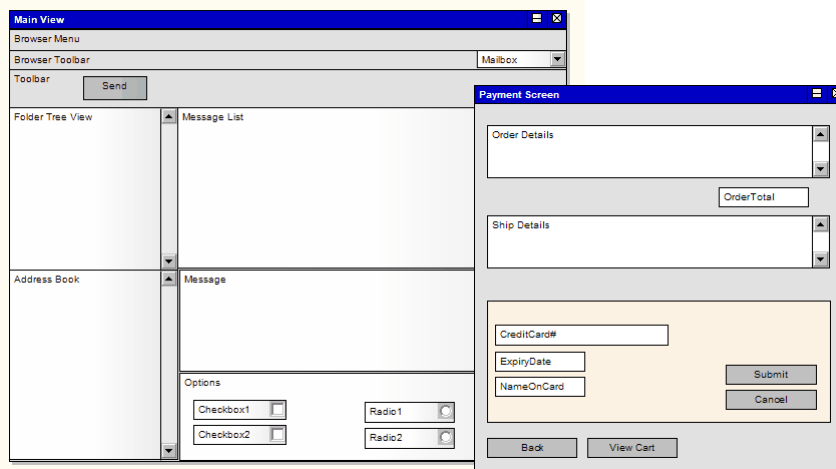
© JMA 2019. All rights reserved

Web Wireframe



© JMA 2019. All rights reserved

Windows Wireframe



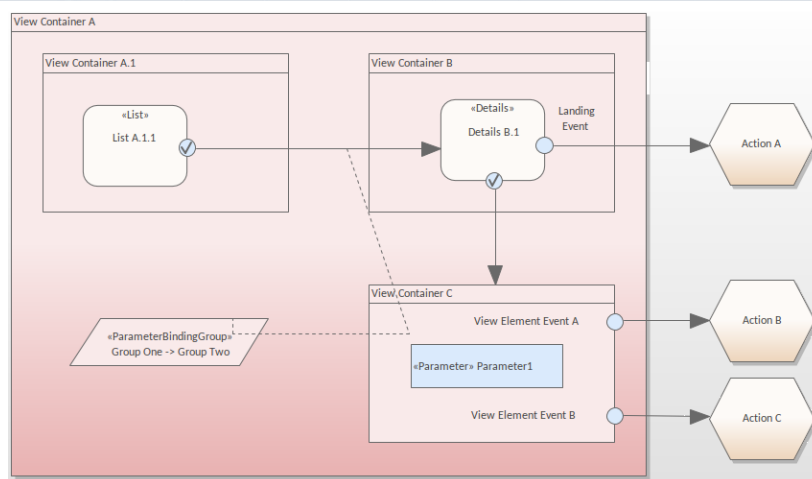
© JMA 2019. All rights reserved

Tecnología MDG para IFML

- El lenguaje de modelado de flujo de interacción (IFML) proporciona a los arquitectos de sistemas, ingenieros de software y desarrolladores de software herramientas que admiten la descripción independiente de la plataforma de interfaces gráficas de usuario para aplicaciones a las que se accede o se implementa en sistemas como computadoras de escritorio, computadoras portátiles, PDA, teléfonos móviles, y tabletas. El lenguaje fue desarrollado por el Object Management Group. La especificación IFML (versión 1.0. Febrero de 2015) está disponible en el sitio web de OMG.
- En Enterprise Architect, se puede modelar los flujos de interacción de aplicaciones de forma rápida y sencilla mediante el uso de la tecnología MDG integrada con el instalador de Enterprise Architect. Las instalaciones de IFML se proporcionan en forma de:
 - Once patrones de modelo IFML para entrada de información, interacción y wireframe, búsquedas y aplicaciones de escritorio
 - Dos tipos de diagramas IFML: diagrama IFML y diagrama de modelo de dominio IFML
 - Páginas de 'Essential Concepts', 'Core' and 'Extensions' de IFML en el Toolbox

© JMA 2019. All rights reserved

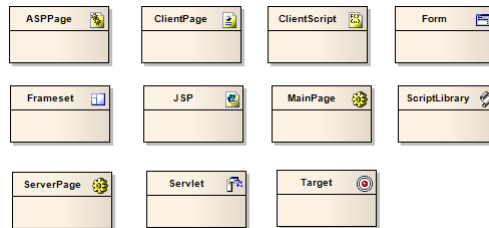
IFML



© JMA 2019. All rights reserved

Estereotipos web

- Enterprise Architect admite una serie de estereotipos para el modelado de páginas web, cuyos elementos gráficos se muestran con un icono gráfico en lugar del formato habitual de "estereotipo". Estos estereotipos solo son compatibles con elementos de clase. Estos son los diversos íconos gráficos y sus estereotipos asociados:



© JMA 2019. All rights reserved

MODELOS DE REQUISITOS

© JMA 2019. All rights reserved

Modelos de requisitos

- El campo de la ingeniería de requisitos es una de las disciplinas más críticas en el ciclo de vida de desarrollo de soluciones y tiene un impacto documentado en el éxito de los proyectos.
- El desarrollo de requisitos consiste en todas las actividades y tareas asociadas con el descubrimiento, la evaluación, el registro, la documentación y la validación de los requisitos para un proyecto en particular. Los requisitos se descubren, analizan, especifican y verifican. Enterprise Architect tiene una amplia gama de herramientas y características para ayudar al Analista a medida que desarrolla los requisitos.
- La pieza central para el desarrollo de requisitos es el Administrador de especificaciones, a través del cual el Analista de requisitos puede ingresar, ver y administrar requisitos en forma de texto como en una hoja de cálculo. Las propiedades de requisitos como Estado, Prioridad y Autor se pueden editar en línea, y los filtros se pueden aplicar para restringir la visualización a requisitos particulares.

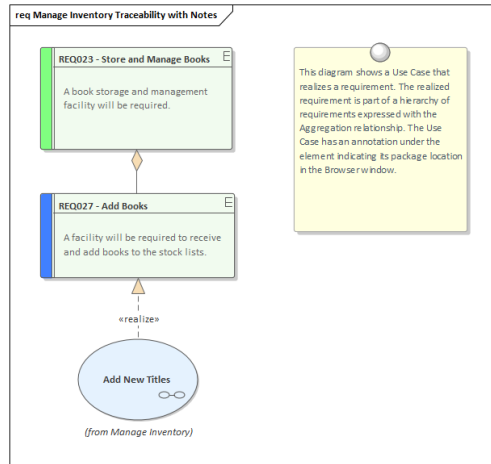
© JMA 2019. All rights reserved

Diagrama de requisitos

- El diagrama de requisitos proporciona una representación visual de cómo se relacionan los requisitos entre sí y con otros elementos del modelo, incluidos los controladores comerciales, las restricciones, las reglas comerciales, los casos de uso, las historias de usuario, los componentes de diseño y más.
- El diagrama es uno de los tipos de diagrama extendido de Enterprise Architect y para los analistas que están acostumbrados a trabajar con requisitos en una herramienta basada en texto, proporcionará una representación gráfica de los requisitos bienvenida y convincente.

© JMA 2019. All rights reserved

Diagrama de requisitos



© JMA 2019. All rights reserved

Administrador de especificaciones

- El Administrador de especificaciones es la herramienta central para trabajar con los requisitos y proporciona un procesador de textos o una hoja de cálculo como interfaz para ingresar, mantener y ver los requisitos.
- Se pueden crear nuevos requisitos con nombres y descripciones detalladas y propiedades como Estado y Prioridad se pueden agregar desde listas desplegables.
- Los requisitos existentes se pueden ver y administrar en una vista conveniente y cambiarlos en el Administrador de especificaciones los cambiará en todos los demás lugares del repositorio, como diagramas y ventanas.
- Es la herramienta perfecta para aquellos analistas que se sienten más cómodos trabajando con texto en lugar de diagramas y que están acostumbrados a trabajar en un procesador de textos o una hoja de cálculo.
- Tiene la ventaja adicional de que los requisitos son parte de un modelo y se pueden rastrear a otros elementos, incluidos los controladores empresariales.

© JMA 2019. All rights reserved

Administrador de especificaciones

Item

1 REQ019 - Manage Inventory

The system MUST include a complete inventory management facility to store and track stock of books for the on-line bookstore.

1.1 REQ122 - Inventory Reports

Inventory reports are required that detail the available stock for each item including back orders. Future stock level reports should be able to predict the quantity of stock at a specified future date.

1.2 REQ023 - Store and Manage Books

A book storage and management facility will be required.

1.2.1 REQ022 - Order Books

A book order facility will be required to allow on-line ordering from major stockist's.

1.2.2 REQ021 - List Stock Levels

A facility will exist to list current stock levels and to manually update stock quantities if physical checking reveals inconsistencies.

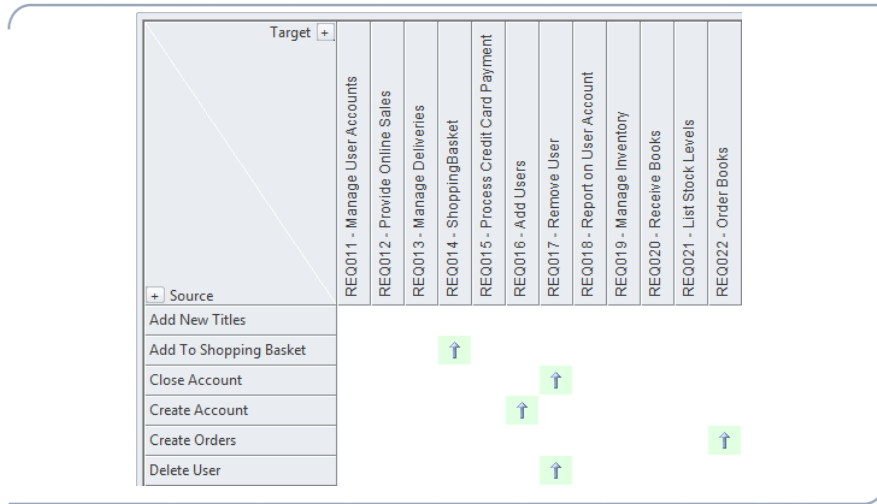
© JMA 2019. All rights reserved

Matriz de relaciones

- La matriz de relaciones es una herramienta para visualizar la forma en que los requisitos están relacionados entre sí y con otros elementos del modelo en una matriz visualmente atractiva o una hoja de cálculo como una vista.
- Se puede usar para ver la relación entre las partes interesadas y sus requisitos, cómo se relacionan los casos de uso con los requisitos comerciales o los requisitos funcionales, cómo se relacionan las capacidades con los controladores comerciales, qué componentes implementan un conjunto de requisitos y más.
- Se puede definir rápidamente cualquier número de matrices y luego guardarlas para verlas en talleres o incluirlas en la documentación generada automáticamente desde el modelo o exportarlas a un archivo de hoja de cálculo. Cuando se crea una matriz, se pueden ver las conexiones colocando los requisitos en un eje de la matriz y los elementos conectados en el otro eje.

© JMA 2019. All rights reserved

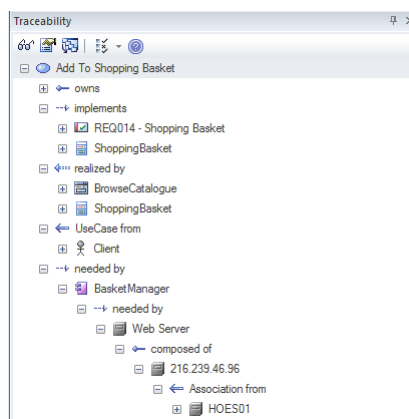
Matriz de relaciones



© JMA 2019. All rights reserved

Ventana de trazabilidad

- La ventana Trazabilidad le da al modelador una vista jerárquica de las conexiones de elementos, lo que permite visualizar y consultar la trazabilidad a medida que los elementos se recorren en el modelo.
- Esta herramienta es particularmente poderosa porque un modelador a menudo elige ocultar las relaciones del diagrama, pero al seleccionar un elemento en el diagrama y ver sus conexiones en la ventana de trazabilidad, se mostrarán todas sus relaciones.



© JMA 2019. All rights reserved

PUBLICACIÓN

© JMA 2019. All rights reserved

Publicación modelo

- Enterprise Architect tiene un motor de documentación sofisticado y flexible que lo ayuda a crear y difundir documentación de alta calidad y totalmente personalizable directamente desde el modelo, en formato de documento o página web. Una razón fundamental para construir modelos es comunicar ideas y conceptos, y crear una vista compartida de un sistema o proceso. Si bien muchos usuarios podrán ver los modelos dentro de Enterprise Architect, existen circunstancias en las que se requiere documentación visible o impresa en la pantalla.
 - La generación de documentos basada en modelos es una poderosa herramienta de productividad porque cuando se cambia un modelo, la documentación puede regenerarse automáticamente, ahorrando tiempo y recursos y reduciendo la posibilidad de errores. Las plantillas incorporadas cuidadosamente diseñadas y versátiles lo ayudan a generar documentación de alta calidad de manera inmediata.
 - La gestión de proyectos y la información estratégica también se pueden generar en una serie de cuadros y gráficos que proporcionan una representación visualmente convincente de datos importantes del proyecto. Los gráficos son configurables y pueden incluirse en la documentación o mostrarse visualmente en diagramas, o incorporarse a una presentación de gestión.
-

© JMA 2019. All rights reserved

Características

- Generar documentos
 - Generación con un clic de documentos de estilo RTF, DOCX y PDF.
- Informes web
 - Implemente su proyecto a través de Internet o intranet para una comunicación de equipo rápida y fácil.
- Artículo Joomla!
 - Publica tu modelo como un artículo en una instalación local de Joomla!.
- Charting
 - Haga que sus informes cobren vida con cuadros y gráficos actualizados al segundo de su modelo.
- Documentos virtuales
 - Personalice sus informes a nivel de modelo definiendo la estructura y el contenido de su informe a través de un diagrama.
- Plantillas de documentos
 - Adapte sus informes para darles su propia apariencia corporativa única, asegurando que toda la documentación se vea igual.

© JMA 2019. All rights reserved

Diagramas del tablero de instrumentos

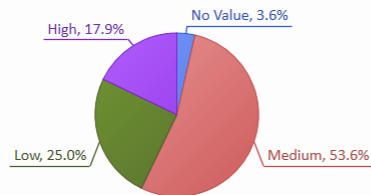
- Los diagramas del tablero (Extended | Dashboard) permiten crear cuadros y gráficos de alta calidad para mostrar la información del repositorio de una manera visualmente atractiva, como la proporción de Prioridades de requisitos en un gráfico circular.
- Hay una página de la caja de herramientas de cuadros y gráficos preconfigurados, pero el usuario es libre de crear y guardar cualquier cantidad de datos de origen de gráficos desde cualquier parte del repositorio.
- Los cuadros y gráficos proporcionan valiosa información resumida que ayuda en la gestión de los requisitos.
- Los informes de alto nivel y el estado del proyecto se pueden rastrear y documentar fácilmente utilizando los numerosos gráficos y elementos de informes disponibles que se vinculan estrechamente con el contenido y el estado del modelo.

© JMA 2019. All rights reserved

Diagramas del tablero de instrumentos

dash Functional Requirements Dashboard Priority

Elements by Priority



This diagram shows a Pie Chart element depicting element priority for all the requirements in a selected package. It provides a useful summary for a requirements manager and is dynamically updated when the Priority changes and the diagram is reopened. There are a range of other pre-defined charts and user defined charts can also be added. A filter has been added to exclude all elements other than Requirements.