

Ejemplo de Descripción Arquitectónica (DA)

Desarrollo de Software
Curso 2021-2022
3º Grado Ingeniería Informática

Dpto. Lenguajes y Sistemas Informáticos
ETSIIT
Universidad de Granada

24 de marzo de 2023



Índice

1. Análisis de requisitos	4
1.1. Requisitos no funcionales	4
1.2. Requisitos funcionales	5
2. Listado y descripción de las partes interesadas en el sistema	8
3. Listado y descripción de las inquietudes o intereses en el sistema	9
3.1. Inquietudes	10
3.1.1. Inquietudes relacionadas con requisitos no funcionales	11
3.1.2. Inquietudes relacionadas con requisitos funcionales	11
4. Descripción detallada del punto de vista contextual	11
4.1. Inquietudes	12
4.1.1. Ámbito del sistema y responsabilidades	12
4.1.2. Identificación de las entidades externas y servicios y datos usados	12
4.1.3. Naturaleza y características de las entidades externas	12
4.1.4. Identificación y responsabilidades de las interfaces externas	13
4.1.5. Naturaleza y características de las interfaces externas	14
4.1.6. Otras interdependencias externas	16
4.1.7. Impacto del sistema en su entorno	17
4.1.8. Completitud, consistencia y coherencia global	17
4.1.9. Inquietudes para cada interesado en el sistema	18
4.2. Modelos: el modelo de contexto	18
4.2.1. Notación	18
4.3. Modelos: escenarios de interacción	20
4.4. Problemas y errores comunes	24
4.5. Lista de verificación	24
5. Descripción detallada del punto de vista funcional	25
5.1. Inquietudes	25
5.1.1. Capacidades funcionales	25
5.1.2. Interfaces externas	26
5.1.3. Estructura interna	27
5.1.4. Filosofía del diseño funcional	28
5.1.5. Inquietudes para cada interesado en el sistema	28
5.2. Modelos	29
5.2.1. Modelo de la estructura funcional	29
5.3. Actividades	35
5.3.1. Identificar los elementos	35
5.3.2. Asignar responsabilidades a los elementos	35
5.3.3. Diseñar las interfaces	37
5.3.4. Diseñar los conectores	37

5.3.5.	Comprobar la trazabilidad funcional	39
5.3.6.	Recorrer escenarios comunes	39
5.3.7.	Analizar las interacciones	40
5.3.8.	Analizar la flexibilidad	40
5.4.	Problemas y errores comunes	40
5.5.	Lista de verificación	40
6.	Descripción detallada de la perspectiva de evolución	41
6.1.	Aplicabilidad	41
6.2.	Inquietudes	41
6.3.	Actividades	42
6.3.1.	Primer paso: Caracterizar las necesidades evolutivas	42
6.3.2.	Segundo paso: Evaluar la facilidad para evolucionar en la actualidad	44
6.3.3.	Tercer paso: Considerar las contrapartidas de la evolución	45
6.3.4.	Cuarto paso: Revisar la arquitectura	45
6.4.	Tácticas arquitectónicas	50
6.5.	Problemas y errores	51
6.6.	Lista de comprobación	51
6.6.1.	Lista de comprobación para captura de los requisitos:	51
6.6.2.	Lista de comprobación para la DA:	52
7.	Descripción detallada de la perspectiva de seguridad	52
7.1.	Aplicabilidad	52
7.1.1.	Actividades	53
7.1.2.	Principales	53

Enunciado: descripción del supuesto práctico

Un minorista ha decidido incluir venta por internet con entrega a domicilio. Al principio, sólo permitirá pago con tarjeta y pedido de productos cuando haya disponibilidad. En el caso de cambios o cancelaciones, habrá que hacerlo por teléfono. Para el comercio es fundamental mantener su buena reputación en todos los aspectos: calidad de los productos, calidad del servicio de pedidos –información clara y accesible del producto, honestidad al hacer la cesta de la compra, sensibilidad a las dificultades de sus compradores (como edad o alguna discapacidad, desconocimiento del idioma local), confidencialidad y seguridad de los datos personales y de pago–, calidad del servicio de reparto –rapidez, sensibilidad a las dificultades de movilidad de los clientes, flexibilidad–, compromiso con el medioambiente.

1. Análisis de requisitos

Dos son los objetivos generales del sistema que se desea construir:

1. Proveer de una tienda en línea para que los clientes puedan realizar compras
2. Mantener la buena reputación del negocio

Los desglosamos a continuación en requisitos funcionales (objetivo 1) y no funcionales (objetivo 2).

1.1. Requisitos no funcionales

El sistema debe ser:

- Seguro: garantizando especialmente la protección de la BD frente a ataques externos y la confidencialidad de datos personales
- Regulable: debe cumplir: Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales; Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico
- Ético:
 - Compromiso/transparencia hacia el cliente: la única forma de retribución será por el importe directo de la compra (ausencia de publicidad en internet, ausencia por defecto de extracción de perfiles de comprador, acuerdos para compartición de datos con otras empresa, etc.)
 - Comercio justo: los productos comercializados cumplen con los criterios de solidaridad establecidos por la Coordinadora Estatal de Comercio Justo (tienda on-line de comercio justo)
 - Compromiso adicional con el medioambiente: en venta de productos perecederos, implantación de una política de venta que permita al cliente su implicación para evitar la pérdida

- Eficiente y escalable: dando respuestas rápidas también en caso de sobrecarga de uso (picos de hasta 10 veces más de la media)
- Evolutivo: en concreto se ampliará para añadir otras formas de pago, gestión de cancelaciones y cambios, y gestión de reservas cuando falta stock
- Disponible y resiliente: sistema en espejo para aumentar disponibilidad; pasarela de pago –sistema bancario externo– con garantía de disponibilidad-resiliencia
- Sobre la interfaz de usuario:
 - Amigable: GUI sencilla de usar, intuitiva, adaptada a distintos dispositivos
 - Accesible: Cumplir con las recomendaciones W3C (video y audio; diseño y estilos Web claros, contraste fuerte en colores; botones/letras grandes o adaptables; reconocimiento de voz; conversión texto a habla; notificaciones y retroalimentación claras; acceso sin ratón)
 - Internacionalizable: castellano e inglés

1.2. Requisitos funcionales

NOTA: Al tratar de identificar los requisitos funcionales, el arquitecto aprecia que la descripción informal del sistema no deja claro si el sistema gestionará la información personal de los clientes, direcciones de entrega, datos bancarios, etc. En una entrevista con el minorista (el cliente o adquiriente del sistema) se aclara y añade lo siguiente a la descripción informal:

ENTREVISTA (adquiriente)

El negocio ya cuenta con una BD donde se registran algunos datos de clientes (nombre, email, dirección) y un sistema de distribución que guarda también en la BD el estado de cada entrega, porque algunos ya hacían pedidos por teléfono o correo electrónico. En cuanto a los datos bancarios, se pide que el sistema no guarde esos datos.

- Consultar catálogo productos
- Consultar la traza de una entrega
- Consultar compras
- Realizar compra
- Alta de usuario
- Gestión de cuentas de usuario
- Gestión del catálogo

Requisitos funcionales que NO son responsabilidad del sistema:

- Cancelar/modificar compra (aún no)
- Pagar con tarjeta (sistema externo)
- Pagar por otros medios (aún no)
- Gestionar direcciones de entrega (entidad externa de la empresa)
- Preparar y entregar pedido (entidad externa de la empresa)
- Registrar traza de entrega (entidad externa de la empresa)

Descripción de los requisitos funcionales

Detalle	Descripción
RF#	1
Nombre	Consultar catálogo productos
Descripción	Permite consultar los productos disponibles (descripción y stock) con distintos criterios de búsqueda
Entrada	Categoría de producto/producto en el catálogo
Procesamiento	Consulta por el criterio de búsqueda en la BD externa al sistema
Salida	Descripción de los productos consultados

Detalle	Descripción
RF#	2
Nombre	Consultar la traza de una entrega
Descripción	Permite consultar el estado de un pedido
Entrada	Pedido en la lista de pedidos (pendientes/entregados/todos; por fechas) del cliente
Procesamiento	[Prerrequisito: acceso identificado] Consulta del pedido en la BD externa al sistema
Salida	Secuencia eventos realizados para la entrega; tiempo aproximado para la entrega si está pendiente

Detalle	Descripción
RF#	3
Nombre	Consultar compras
Descripción	Permite consultar las compras realizadas
Entrada	Compra (todas/pendientes de recibir/recibidas; por fechas) del cliente
Procesamiento	[Prerrequisito: acceso identificado] Consulta de las compras detalladas en la BD externa al sistema
Salida	Detalles de cada compra consultada

Detalle	Descripción
RF#	4
Nombre	Realizar compra
Descripción	Permite realizar un pedido
Entrada	Productos y cantidad solicitada
Procesamiento	[Prerrequisito: acceso identificado] Registro del pedido en la BD externa al sistema; realización del pago (sistema externo); confirmación/cambio dirección de entrega; petición del pedido al sistema externo de preparación y entrega del pedido; actualización del stock en el catálogo
Salida	Confirmación de la compra al usuario
Escenario a destacar	El cliente no realiza el pago (notificación para gestión manual)

Detalle	Descripción
RF#	5
Nombre	Alta de usuario
Descripción	Permite el alta de un usuario (cliente de la tienda) en el sistema
Entrada	Datos del cliente (email, contraseña, nombre, dirección, teléfonos)
Procesamiento	[Prerrequisito: acceso no identificado] Registro de la cuenta en la BD externa al sistema ; [Postrequisito: acceso identificado]
Salida	Confirmación del alta

Detalle	Descripción
RF#	6
Nombre	Gestión de cuentas de usuario
Descripción	Permite baja/modificación de datos de acceso (email, contraseña)
Entrada	Nuevo datos del cliente
Procesamiento	[Prerrequisito: acceso identificado] Actualización de los datos en la BD externa al sistema
Salida	Confirmación de los cambios realizados

Detalle	Descripción
RF#	7
Nombre	Gestión del catálogo
Descripción	Permite añadir/eliminar/modificar productos (descripción y stock) en el catálogo
Entrada	Tipo de cambio; datos nuevos
Procesamiento	[Prerrequisito: acceso identificado como gestor del catálogo] Actualización de los datos en la BD externa al sistema
Salida	Confirmación de los cambios realizados

2. Listado y descripción de las partes interesadas en el sistema

- Arquitecto: A.A.A. (máster en ingeniería informática) es el director gerente de una pequeña empresa A especializada en la actualidad en aplicaciones Web para la promoción de las PYMES. Será el arquitecto software encargado de elaborar la D.A. y supervisar que se desarrolle de forma apropiada.
- Cliente (adquiriente): B.B.B. es el dueño de un negocio B de venta minorista y el que contrata el desarrollo de la tienda Web.
- Desarrollador: C.C.C. es un desarrollador (grado superior de FP en desarrollo de aplicaciones) que trabaja en la empresa A.
- Ingeniero de producción y mantenimiento y webmaster: D.D.D. es una graduada en ingeniería informática que se encarga de dar soporte hardware y middleware a las aplicaciones Web que desarrolla A y de supervisar su mantenimiento.
- Proveedor: La empresa B proporciona la infraestructura hardware a la empresa A.
- Servicio al cliente (helpdesk): C.C.C. también se encarga en la empresa A de proporcionar este servicio a sus clientes.

- Administrador del sistema: Dado la baja carga de administración que se requiere, B.B.B. se encargará de administrar su propio sistema.
- Técnico de pruebas: C.C.C. también realiza las pruebas en las aplicaciones que se desarrollan en la empresa A.
- Usuarios: Existirán cuatro tipos de usuarios:
 - Clientes de la tienda Web.- Como aún no existen, usaremos partes interesadas de proximidad: dos familiares y un amigo de B.B.B. (su padre –de cierta edad–: U1, otro familiar con serios problemas de vista: U2 y un buen amigo con formación en técnicas de mercado: U3), quienes se comprometen a representar a los clientes durante la elaboración de la D.A.
 - Mantenedor del catálogo.- Lo hará el propio B.B.B.
 - Vendedor.- Se encargará de preparar la cesta de entrega una vez que se realiza una compra a través del sistema y de registrar la preparación en el sistema para que se tramite la orden de entrega al sistema interno de distribución
 - Repartidor.- Harán las tareas que ya hacían en el sistema interno de distribución, pero ahora podrán conectar con ese sistema a través de la interfaz de usuario del sistema a implementar

3. Listado y descripción de las inquietudes o intereses en el sistema

Cuando se le presentan a las distintas partes interesadas los requisitos funcionales y no funcionales elaborados por el arquitecto (ver Sección 1), se produce cierto rechazo por parte del cliente y de los usuarios, a pesar de que se han explicado los requisitos en una reunión y aclarado algunos tecnicismos utilizados. El cliente expresa las inquietudes que se detallan en el siguiente cuadro (Rozanski, 2020, Sec. 2.3.2.2.):

INQUIETUD PRINCIPAL: Mantener la reputación del servicio proporcionado

El minorista tiene una gran reputación en la calidad del servicio y la respuesta que da a sus clientes. Esto hace que tenga los siguientes objetivos y aspiraciones para una nueva tienda en línea que quiere construir:

- Los valores, la ética y la reputación del minorista deben reflejarse en la apariencia y el funcionamiento de la tienda en línea y sus procesos de respaldo.
- En todo momento, el sitio Web debe tratar de presentar una cara “humana” al cliente (incluso aquellas partes de la misma que están completamente automatizadas).
- La tienda en línea debe ser fácil de usar para los clientes que tienen una experiencia limitada con los ordenadores y con el comercio electrónico o los que tengan alguna discapacidad.
- La tienda en línea debe ser rápida de cargar y responder a las acciones del cliente independientemente de la velocidad de conexión a Internet.

Los usuarios expresan asimismo las siguientes inquietudes:

INQUIETUDES DE LOS USUARIOS

- U1: Temo que tenga problemas de letra muy pequeña, que algunas páginas no tengan la letra adaptada, quizás en la página de pago; pero sobre todo temo perderme en las páginas y no encontrar la manera de comprar, consultar productos, etc.
- U2: Dudo que el audio para personas con problemas visuales funcione bien, pocos sitios Web lo gestionan bien
- U3: Creo que el sistema debería gestionar perfiles de comprador para facilitar una nueva compra, analizando las compras realizadas para ofrecer los mismos productos y otros nuevos parecidos

3.1. Inquietudes

A partir de las inquietudes expresadas por las partes interesadas y los requisitos elaborados por el arquitecto, se obtienen las siguientes inquietudes, en donde se conectan las inquietudes generales expresadas por el cliente y los usuarios U1 y U2, con los requisitos no funcionales (añadiendo uno) y la inquietud de U3 se incorpora como un requisito funcional más.

3.1.1. Inquietudes relacionadas con requisitos no funcionales

Inquietudes generales

La más importante es la de mantener la reputación del servicio proporcionado. Que se puede desglosar en tres:

- Los valores, la ética y la reputación del minorista deben reflejarse en la apariencia y el funcionamiento de la tienda en línea y sus procesos de respaldo. Para ello se incluyen los requisitos de seguridad, eticidad, eficiencia y disponibilidad.
- En todo momento, el sitio Web debe tratar de presentar una cara “humana” al cliente (incluso aquellas partes de la misma que están completamente automatizadas). Para ello se incluye el requisito de eticidad y los relacionados con la interfaz de usuario.
- La tienda en línea debe ser fácil de usar para los clientes que tienen una experiencia limitada con los ordenadores y con el comercio electrónico o los que tengan alguna discapacidad. Para ello se incluyen los requisitos de interfaz amigable y accesible.
- La tienda en línea debe ser rápida de cargar y responder a las acciones del cliente, independientemente de la velocidad de conexión a Internet. Para ello se incluye (añadido) el requisito de responsividad (ver abajo).

Inquietudes específicas (requisitos no funcionales)

Se añade el siguiente:

- (v2) Responsivo: debe adaptarse al dispositivo usado, considerando en especial los móviles, que suelen tener velocidades de carga bajas

3.1.2. Inquietudes relacionadas con requisitos funcionales

Se añade un requisito funcional más para atender a la inquietud de U3:

Detalle	Descripción
RF#	8-v2
Nombre	Elaboración de perfiles de comprador
Descripción	Mantiene un registro de perfiles de comprador que se actualiza después de cada compra y que es usado para hacer sugerencias al cliente
Entrada	Nueva compra
Procesamiento	Actualización del perfil en la BD externa al sistema
Salida	Sugerencias de compra en la Web

4. Descripción detallada del punto de vista contextual

Describe las relaciones de la tienda Web con las entidades externas (actores), tanto sistemas internos al negocio minorista como sistemas externos con los que deba interactuar.

4.1. Inquietudes

4.1.1. Ámbito del sistema y responsabilidades

Las inquietudes en este ámbito son las que se detallan a continuación ([Rozanski, 2020](#), pág. 42):

Comercio electrónico al por menor

- Presentar el catálogo de productos, con imágenes y descripción de cada producto
- Proporcionar una herramienta de búsqueda flexible (por nombre de producto, tipo, palabra clave, tamaño, etc.
- Aceptar pedidos de productos
- Aceptar pago con tarjeta de crédito (con aprobación asíncrona y notificación al cliente
- Proporcionar interfaces de conexión automática con un sistema de soporte para la entrega del producto

Capacidades excluidas en una primera versión del sistema:

- La capacidad para modificar o cancelar pedidos (sólo se podrá hacer por teléfono)
- La posibilidad de pagar de otra forma distinta a usar tarjeta de crédito
- Posibilidad de reservar hasta disponibilidad

4.1.2. Identificación de las entidades externas y servicios y datos usados

- Usuario del sistema.- Fundamentalmente los clientes, pero también usuarios no identificados, el gestor del catálogo y la webmaster
- Sistema externo bancario para pago con tarjeta
- BD de la tienda.- Usada en la actualidad para información de contacto de clientes que hacen pedidos por teléfono
- Sistema interno de distribución
- Sistema interno de gestión de clientes

4.1.3. Naturaleza y características de las entidades externas

- Usuario del sistema.- Cierta número de clientes pueden presentar alguna limitación (discapacidad) para acceder a la web; otros pueden usar móviles con una velocidad

baja de descarga

- Sistema externo bancario para pago con tarjeta.- La empresa A tiene experiencia con el sistema que será usado. El sistema es proporcionado por una entidad financiera con larga experiencia en comercio electrónico y ofrece todas las garantías de seguridad, velocidad de respuesta, escalabilidad, disponibilidad y resiliencia y legalidad
- BD de la tienda.- Usada en la actualidad para información de contacto de clientes que hacen pedidos por teléfono. La BD se aloja en servidores de la empresa A, junto con otras. La empresa ofrece las garantías de seguridad, confidencialidad, velocidad de respuesta, escalabilidad, disponibilidad y resiliencia y legalidad.
- Sistema interno de distribución.- Es una app Android que se inicia en la actualidad por el vendedor cada vez que se recibe un pedido por teléfono, una vez preparado el mismo. Proporciona, usando la BD de la tienda, las siguientes funciones:
 - Para usuarios vendedores:
 - Crear una orden de entrega, eligiendo el cliente de la BD mientras se prepara la cesta (real)
 - Elegir a la persona encargada del reparto (una sola por ahora), en una lista de repartidores
 - Poner estado de pedido preparado cuando esté terminada la cesta, para que el repartidor pueda enviarla a su destinatario
 - Consultar la traza de una orden de entrega
 - Para usuarios repartidores:
 - Consultar repartos (todos, pendientes, terminados)
 - Conectar una dirección de entrega con el navegador GPS de Android
 - Añadir evento de entrega: salida, carga en medio transporte, llegada al destino, entrega, destinatario ausente
- Sistema interno de gestión de clientes
 - Alta cliente
 - Modificación de dirección de entrega u otros datos de contacto
 - Baja cliente

4.1.4. Identificación y responsabilidades de las interfaces externas

Interfaces de servicios:

Se detallan en las Tablas 1 y 2.

Detalle	Descripción
Nombre	Pago con tarjeta
Semántica	permite hacer un pago con tarjeta de crédito en el sistema bancario para pago con tarjeta
Parámetros	nombre y apellidos, número tarjeta, CVC, mes caducidad, año caducidad, tipo de tarjeta y pin enviado a móvil
Acciones a realizar	cargo en tarjeta
Excepciones (1. error en datos)	devuelve código excepción 1
Excepciones (2. operación no autorizada)	devuelve código excepción 2
Errores: pago interrumpido por cliente (pin no enviado, ventana cerrada ...)	a tratar por el sistema a implementar

Tabla 1: Interfaz de servicio 1

Detalle	Descripción
Nombre	Consulta tipo SQL en BD externa
Semántica	realiza una operación en la BD externa
Parámetros	nombre BD, nombre usuario y contraseña encriptados, código tipo SQL
Acciones a realizar	conectar con SGBD, identificar usuario, hacer operación, desconectar
Excepciones (1. error sintáctico código SQL)	devuelve código excepción 1
Excepciones (2. operación no autorizada)	devuelve código excepción 2
Errores (sistema externo no responde)	a tratar por el sistema a implementar

Tabla 2: Interfaz de servicio 2

Detalle	Descripción
Nombre	Cambio de dirección de entrega en el sistema interno de gestión de clientes
Semántica	cambia la dirección de entrega de un cliente
Contenidos	id cliente, nueva dirección
Acciones a realizar	actualizar BD externa, esperar respuesta, actualizar información app. del repartidor
Volumen	generalmente bajo pero puede tener picos elevados
Tiempos de ocurrencia probables	Menos de 1 min. en volumen bajo; hasta 5 minutos si hay picos

Tabla 3: Interfaz dirigida por eventos I3

Interfaces dirigidas por eventos:

Se detallan en las Tablas 3 y 4.

Detalle	Descripción
Nombre	Petición de entrega al sistema interno de distribución
Semántica	realiza una petición de entrega de un pedido
Contenidos	nombre cliente, código pedido, código repartidor
Acciones a realizar	crear orden de entrega, añadir repartidor, registrar en BD
Volumen	generalmente bajo pero puede tener picos elevados
Tiempos de ocurrencia probables	Menos de 1 min. en volumen bajo; hasta 5 minutos si hay picos

Tabla 4: Interfaz dirigida por eventos I4

4.1.5. Naturaleza y características de las interfaces externas

Las siguientes características se aplican a todas las interfaces externas al sistema:

- Completamente automatizadas
- Transaccionales

Interfaces de servicios:

- pago con tarjeta.- Presenta además las siguientes características, por tratarse de una interfaz de servicios de pago:
 - Escalable: Preparada para aceptar muy grandes volúmenes de datos y muy fuertes picos
 - Eficiente: Respuesta muy rápida
 - Disponible: Muy alta disponibilidad; servidores en espejo
 - Segura y confidencial: datos de tarjeta encriptados, confirmación por pin, protocolo https
 - Tipo: software propietario
- consulta tipo SQL en BD externa.- Presenta además las siguientes características:
 - Escalable: Preparada para aceptar grandes volúmenes de datos y fuertes picos
 - Eficiente: Respuesta rápida
 - Disponible: alta disponibilidad (sistema de alimentación ininterrumpida)
 - Segura y confidencial: acceso identificado a BD, protocolo https, datos de acceso encriptados
 - Tipo: software estándar abierto (JDBC)

Interfaces dirigidas por eventos:

Las siguientes dos interfaces están dirigidas por eventos, ya que no existen exigencias altas de tiempo de respuesta.

- cambio de dirección de entrega en el sistema interno de gestión de clientes.- Presenta además las siguientes características:
- petición de entrega al sistema interno de distribución.- Presenta además las siguientes características:

Además por ser dirigidas por eventos no requieren tampoco de alta disponibilidad. Presenta además las siguientes características:

- Escalable: escalabilidad muy alta por estar dirigidas por eventos
- Segura: protocolo https
- Tipo: software estándar abierto (Java observable/observer)

4.1.6. Otras interdependencias externas

([Rozanski, 2020](#), Cuadro 2.3.16)

Sistema de venta electrónica al por menor

En la Figura 1 se muestra un sistema de comercio electrónico (e-Commerce System) de venta al por menor, que interacciona como varias entidades software externas^a. Este es un ejemplo donde se muestran interdependencias externas entre los sistemas de reparto (Fulfillment System) y de cuentas de cliente (Customer Accounts System). El sistema de reparto tiene su propia lista de direcciones de reparto verificadas para cada cliente y rechazará cualquier petición que se haga a una dirección que no tenga registrada. Sin embargo, esta lista es una copia de la lista del sistema de cuentas de cliente. Cuando un cliente hace un pedido y además actualiza la dirección de envío, el sistema debe tener en cuenta esta dependencia, así como el tiempo para que la lista de direcciones se actualice en el sistema de reparto. Si no, los pedidos podrían ser rechazados porque el sistema de reparto aún no tenga esa dirección registrada.

El impacto que esta dependencia puede tener en la arquitectura es el de permitir el reenvío de la petición al sistema de reparto si ha sido rechazada la anterior, o retrasar las peticiones que conlleven actualización de la dirección de reparto para dar tiempo a que se actualicen las direcciones en el sistema de reparto.

^aEn este diagrama, se utiliza el estereotipo «system» para reflejar el sistema y el resto de “cuadros”, que aparecen sin estereotipo, significa que son entidades externas software.

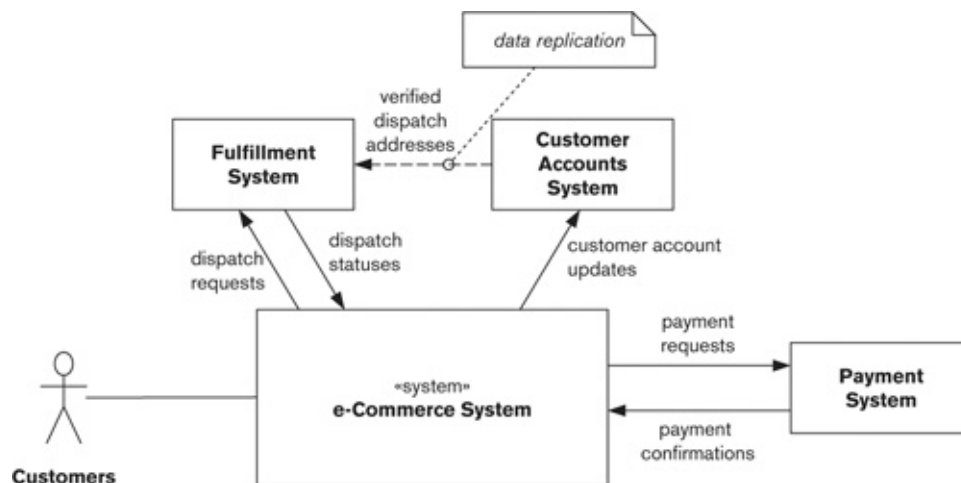


Figura 1: Un diagrama de contexto elaborado con UML. [Fuente: (Rozanski, 2020, Figura 2.26)]

4.1.7. Impacto del sistema en su entorno

Cualquier sistema que dependa del nuestro y que pueda requerir cambios a nivel funcional, de interfaces o mejoras en el rendimiento o en la seguridad

El sistema externo de distribución, que hasta ahora se iniciaba por el vendedor tras preparar un pedido hecho por teléfono, requerirá ahora de una interfaz para lanzarlo desde el nuevo sistema, la que hemos llamado **petición de entrega al sistema interno de distribución**.

El sistema externo de gestión de direcciones de entrega deberá ahora incluir una interfaz para que sea accesible mediante un cliente Web identificado.

El registro de trazas de entrega deberá integrarse en el sistema a desarrollar, ya que está implementado dentro de una clase (TrazaEntrega) que también deberá implementar la consulta de trazas de entrega por cualquier usuario del sistema.

NOTA: No debe confundirse este sistema con otro sistema interno que permitirá la gestión de datos personales y de la cuenta (contraseña, email) en la tienda web.

Cualquier sistema que deba dejar de utilizarse cuando el nuestro entre en explotación

La consulta de la traza de una entrega se rediseñará (en la clase TrazaEntrega) para permitir que la pueda hacer también un cliente y no sólo un repartidor.

Cualquier dato que deba migrarse a nuestro sistema

-

4.1.8. Completitud, consistencia y coherencia global

La inquietud más importante en este aspecto se menciona así: (Rozanski, 2020, Cuadro 2.3.17):

Sistema de venta electrónica al por menor

En los primeros sistemas de comercio electrónico se ponía todo el interés en proporcionar catálogos de productos de gran calidad para hacer los sitios atractivos a los compradores y ser competitivos. Sin embargo, se descuidaba los procesos de pago, reparto o tratamiento de excepciones. Esto finalmente les daba mala reputación en el servicio al cliente, incluso llegando a la quiebra.

Al menos se debe garantizar que los procesos de negocio más importantes tenga una cobertura adecuada, sea por sistemas manuales o automatizados. Del mismo modo, todos los datos que requieran los procesos deben almacenarse en algún lugar (por nuestro sistema o de forma externa) y ser accesibles a todos aquellos sistemas que los necesiten. (Rozanski, 2020, Sec. 2.3.2.6.1)

4.1.9. Inquietudes para cada interesado en el sistema

En la Tabla 5 se muestran las inquietudes que cada una de las partes interesadas en el sistema puede tener desde el punto de vista contextual.

Tipo de interesado	Inquietudes
Cliente (adquiriente)	Ámbito y responsabilidades del sistema; identificación de las entidades externas y servicios y datos usados; impacto de los sistemas en su entorno
Desarrollador	Todas las inquietudes
Ingeniera de producción	Naturaleza y características de las interfaces externas; impacto del sistema en su entorno
Administrador del sistema	Todas las inquietudes
Técnico de pruebas	Todas las inquietudes
Usuarios	Ámbito y responsabilidades del sistema; identificación de las entidades externas y servicios y datos usados; completitud, consistencia y coherencia global

Tabla 5: Inquietudes de los distintos interesados desde el punto de vista contextual, adaptadas al sistema de venta on line de la tienda minorista. [Fuente (adaptada): (Rozanski, 2020, Tabla 2.4)]

4.2. Modelos: el modelo de contexto

4.2.1. Notación

Diagrama UML para las partes interesadas más técnicas Puede verse un ejemplo en la Figura 2. Obsérvese que falta la navegabilidad en las líneas (Rozanski, 2020, Fig. 2.27).

Explicación adicional: los sistemas que no tienen el estereotipo «system» son sistemas externos. Todas las entidades internas están alojadas en servidores de la empresa A.

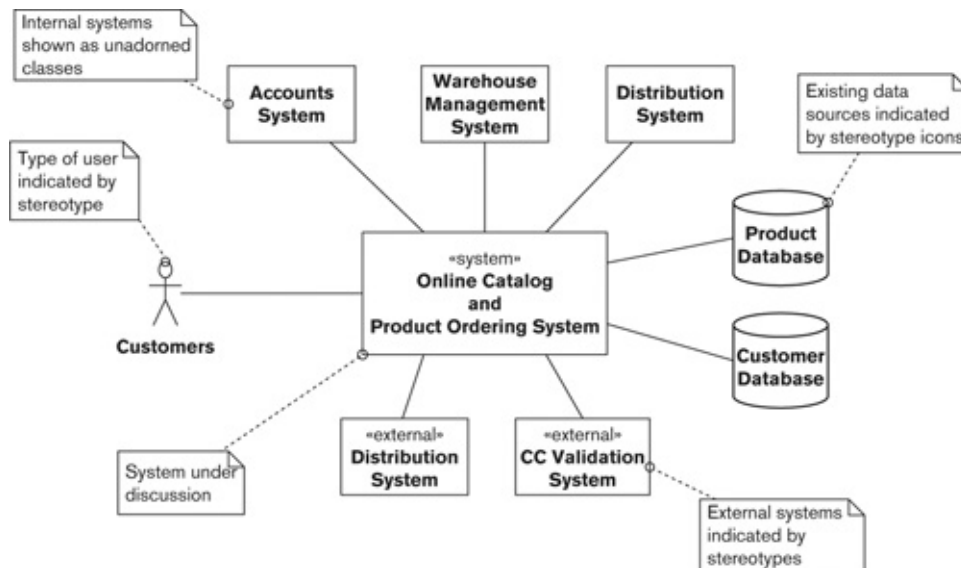


Figura 2: Diagrama de contexto elaborado con UML para las partes interesadas dentro del área de las tecnologías de la información. [Fuente: (Rozanski, 2020, Fig. 2.27)]

Diagrama de cuadros y líneas para las partes interesadas menos técnicas (cliente (adquiriente) y usuarios clientes de la tienda) Puesto que UML no proporciona un soporte específico ni potente para los diagramas de contexto, una alternativa para estas partes interesadas es el uso de diagramas informales con notación de “cuadros y líneas”, que puede ser más simple. Puede verse un ejemplo en la Figura 3 (Rozanski, 2020, pág. 49).

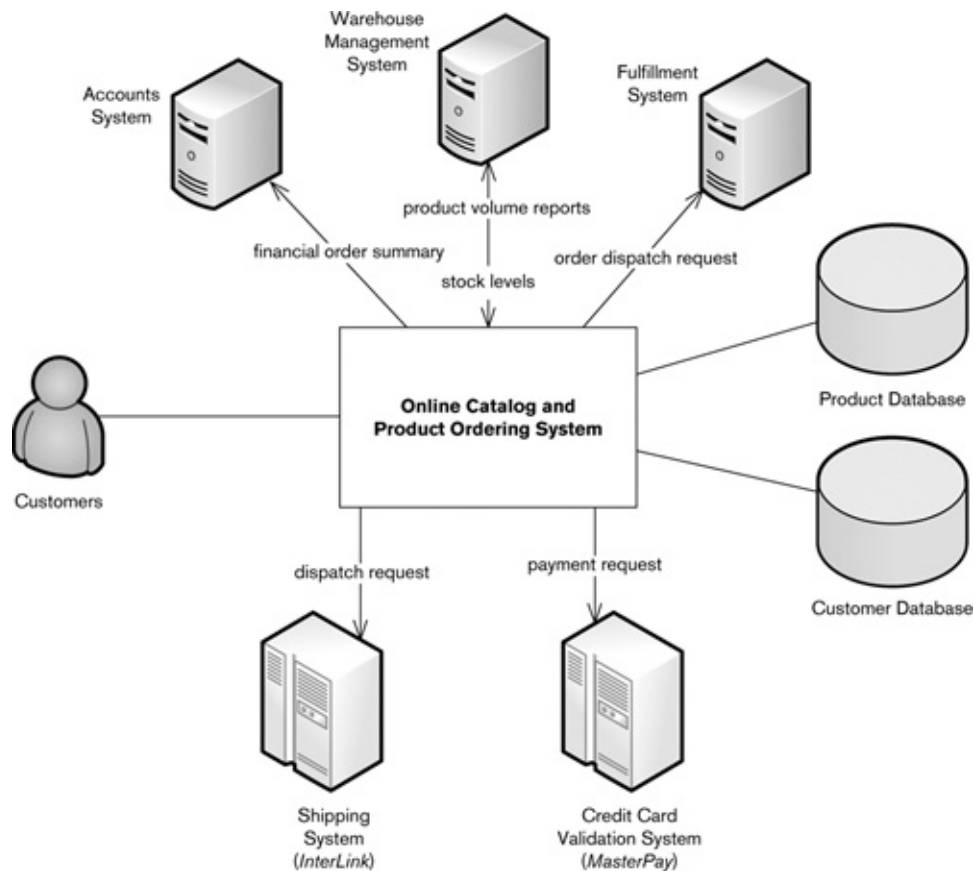


Figura 3: Un diagrama de contexto elaborado de forma libre con “cuadros y líneas”. [Fuente: (Rozanski, 2020, Fig. 2.28)]

4.3. Modelos: escenarios de interacción

Se tiene en cuenta en especial a las partes interesadas no técnicas para identificar los escenarios funcionales de contexto que más puedan aclarar las interacciones entre el sistema y las entidades externas. Luego se consulta con las partes técnicas intentando detectar tanto escenarios fallidos como escenarios “olvidados”.

Utilizamos los dos tipos de notaciones sugeridas: listas de interacción textual y diagramas de secuencia de UML (Rozanski, 2020, Sec. 2.3.2.7.2.1).

ESCENARIO FUNCIONAL DE CONTEXTO 1

- Descripción general: realizar el pago de una compra
- Estado del sistema: el usuario se ha identificado en el sistema y tiene productos en su “carrito de la compra”
- Entorno del sistema: el entorno de implementación funciona normalmente, sin problemas
- Estímulo externo: click sobre botón “realizar compra”
- Respuesta requerida del sistema: el sistema conecta con el sistema externo bancario de pago con tarjeta, que muestra el formulario con los datos de pago

ESCENARIO FUNCIONAL DE CONTEXTO 1 (cont.)

- Estímulo externo: click sobre botón “realizar pago”
- Respuesta requerida del sistema externo: el sistema externo bancario de pago con tarjeta comprueba la viabilidad de la operación y, si es viable, envía pin al móvil del usuario
- Estímulo externo: click sobre botón “introducir pin” con el pin introducido
- Respuesta requerida del sistema externo: el sistema externo bancario de pago con tarjeta comprueba el pin y, si es correcto, realiza el cobro, devolviendo OK al sistema
- Estímulo externo: el sistema externo de pago envía OK
- Respuesta requerida del sistema: Se muestra mensaje de operación correcta y se envía también por email al cliente; se procesa el pedido

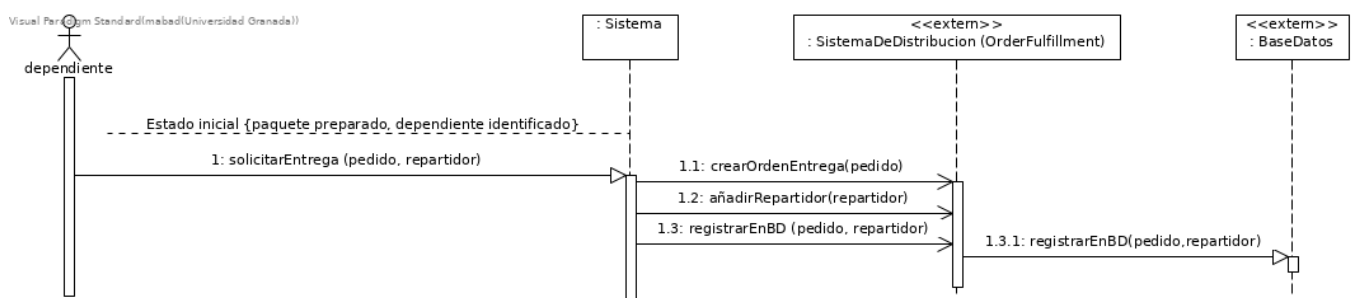


Figura 4: ESCENARIO FUNCIONAL DE CONTEXTO 2: Escenario de petición de entrega descrito mediante un diagrama de secuencias. Se usa mensajería asíncrona (o interacción mediante eventos).

ESCENARIO FUNCIONAL DE CONTEXTO 3

- Descripción general: pago fallido de una compra con excepción
- Estado del sistema: el usuario se ha identificado en el sistema y tiene productos en su “carrito de la compra”
- Entorno del sistema: el entorno de implementación funciona normalmente, sin problemas
- Estímulo externo: click sobre botón “realizar compra”
- Respuesta requerida del sistema: el sistema conecta con el sistema externo bancario de pago con tarjeta, que muestra el formulario con los datos de pago
- Estímulo externo: click sobre botón “realizar pago”
- Respuesta requerida del sistema externo: el sistema externo bancario de pago con tarjeta comprueba la viabilidad de la operación pero no es viable
- Estímulo externo: el sistema externo de pago envía núm. de excepción (1 o 2)
- Respuesta requerida del sistema: Se muestra mensaje de operación no viable y se envía también por email al cliente con link para que pueda volver a intentarlo; no se procesa el pedido

ESCENARIO FUNCIONAL DE CONTEXTO 4

- Descripción general: pago fallido de una compra con error
- Estado del sistema: el usuario se ha identificado en el sistema y tiene productos en su “carrito de la compra”
- Entorno del sistema: el entorno de implementación funciona normalmente, sin problemas
- Estímulo externo: click sobre botón “realizar compra”
- Respuesta requerida del sistema: el sistema conecta con el sistema externo bancario de pago con tarjeta, que muestra el formulario con los datos de pago
- Estímulo externo: NO se hace click sobre botón “realizar pago”
- Respuesta requerida del sistema: El sistema comprueba las compras pendientes de pago hechas hace más de 5 minutos y envía email al cliente con link para que pueda hacer el pago; no se procesa el pedido

ESCENARIO FUNCIONAL DE CONTEXTO 5

- Descripción general: compra recuperada
- Estado del sistema: el sistema tiene compras pendientes de pago
- Entorno del sistema: el entorno de implementación funciona normalmente, sin problemas
- Estímulo externo: el usuario usa el link enviado por email
- Respuesta del sistema: se vuelve a conectar con la página para enviar la orden de pago
- Estímulo externo: click sobre botón “realizar pago”
- Respuesta requerida del sistema externo: el sistema externo bancario de pago con tarjeta comprueba la viabilidad de la operación y, si es viable, envía pin al móvil del usuario
- Estímulo externo: click sobre botón “introducir pin” con el pin introducido
- Respuesta requerida del sistema externo: el sistema externo bancario de pago con tarjeta comprueba el pin y, si es correcto, realiza el cobro, devolviendo OK al sistema
- Estímulo externo: el sistema externo de pago envía OK
- Respuesta requerida del sistema: Se muestra mensaje de operación correcta y se envía también por email al cliente; se procesa el pedido

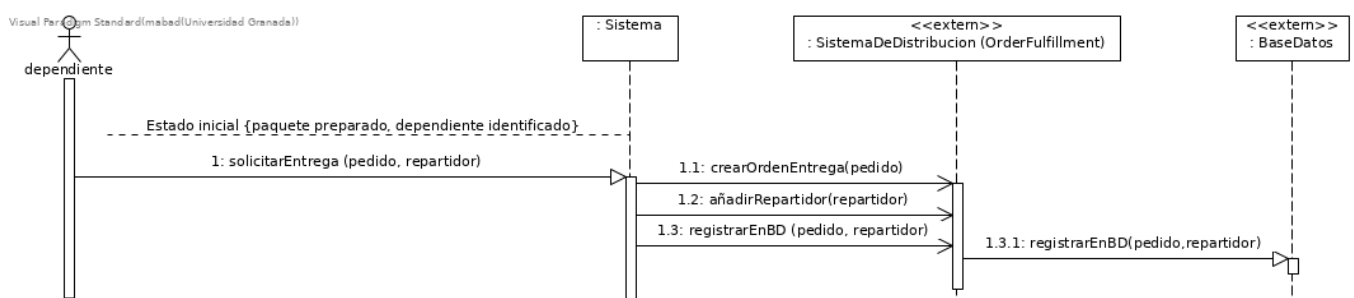


Figura 5: ESCENARIO FUNCIONAL DE CONTEXTO 6. Escenario de activación de solicitud de entrega. Se usa mensajería asíncrona (o interacción mediante eventos). Este escenario (y una función correspondiente) es necesario para evitar inconsistencias en la dirección de entrega debido a las interdependencias entre los sistemas externos de gestión de direcciones de entrega y de distribución.

4.4. Problemas y errores comunes

Transcribo aquí los errores comunes, aunque algunos no detectados ahora pueden detectarse más adelante. Se analizan con las distintas partes interesadas para detectar y corregir ahora el máximo número de ellos. Se pone sólo un guión (-) cuando se haya considerado el problema y se crea que está bajo control.

- Entidades externas ausentes o incorrectas: -
- Dependencias implícitas entre entidades externas no consideradas, y que pueden afectar al propio sistema: detectada la dependencia entre los sistemas externos de gestión de direcciones de entrega y de distribución
- Descripciones imprecisas o ausentes de una interfaz externa: -
- Nivel de detalle inapropiado: -
- Degeneración del entorno, o efectos progresivos de cambios no controlados que pueden no apreciar las partes interesadas: -
- Contexto o ámbito implícito o asumido: -
- Complicación de interacciones: -
- Abuso de jergas tecnológicas que pueden no entender las partes interesadas: -

4.5. Lista de verificación

- ¿Has consultado con todas las partes interesadas quiénes están interesados en el punto de vista contextual (probablemente sean todos)? Sí
- ¿Has identificado todas las entidades externas al sistema y sus responsabilidades más relevantes? Sí
- ¿Comprendes bien la naturaleza de cada interfaz con cada entidad externa y está documentada en un nivel apropiado de detalle? Sí
- ¿Has considerado las posibles dependencias entre las entidades externas con las que hay que interactuar? ¿Están documentadas estas dependencias implícitas en la DA? Sí
- ¿Ilustra de forma adecuada el diagrama de contexto todas las interfaces entre el sistema y su entorno con las suficientes definiciones aclaratorias en el diagrama? Sí
- ¿Han acordado formalmente todas las partes interesadas con los contenidos del modelo contextual? ¿Está documentado en algún lugar? Sí
- ¿Está situado el modelo contextual bajo algún sistema formal de control de cambios? Sí

- ¿Se sigue el proceso de control de cambios? ¿Se consulta a las partes interesadas para que den su consentimiento formal? Sí
- ¿Se coloca en modelo contextual en algún lugar fácilmente accesible por todos, tal como una carpeta compartida? Sí
- ¿Has identificado todas las capacidades o requerimientos básicos del sistema y están documentados en el nivel de detalle apropiado? Sí
- ¿Es la definición del ámbito consistente internamente? Sí
- ¿Identifica el entorno cualquier posible restricción tecnológica como por ejemplo plataformas de uso exigidas? Sí
- ¿Está el entorno especificado en un nivel apropiado de detalle, equilibrando brevedad con claridad y completitud? Sí
- ¿Has explorado un conjunto de escenarios realistas de interacciones entre el sistema y actores externos? Sí
- ¿Les parece claro el contexto, el ámbito y posibles implicaciones a otros equipos con los que debes interactuar? Sí
- ¿Has comprobado si en el modelo contextual existe alguna información que parezca obvia y debería ser explícitamente descrita pero se ha omitido? Sí
- Tienen los procesos de negocios más importantes una cobertura adecuada, tanto por los sistemas o los procesos manuales definidos? Sí
- ¿Están todos los datos requeridos para los procesos de negocio principales almacenados en algún lugar, interna o externamente? Sí
- ¿Está formulada de forma coherente la solución global? Sí

5. Descripción detallada del punto de vista funcional

Describe los elementos funcionales que forman parte de la tienda Web, una vez en ejecución, así como sus responsabilidades, interfaces e interacciones más importantes.

5.1. Inquietudes

5.1.1. Capacidades funcionales

Las capacidades funcionales se pueden obtener directamente de la última versión de los requisitos funcionales:

- Consultar catálogo productos (Tabla 1.2)

- Consultar la traza de una entrega (Tabla 1.2)
- Consultar compras (Tabla 1.2)
- Realizar compra (Tabla 1.2)
- Alta de usuario (Tabla 1.2)
- Gestión de cuentas de usuario (Tabla 1.2)
- Gestión del catálogo (Tabla 1.2)
- (v2) Elaboración de perfiles de comprador (Tabla 3.1.2)

Funciones que NO son responsabilidad del sistema:

- Cancelar/modificar compra (aún no)
- Pagar con tarjeta (sistema externo)
- Pagar por otros medios (aún no)
- Registrar datos cliente y direcciones de entrega (entidad externa de la empresa)
- Preparar y entregar pedido (entidad externa de la empresa)
- Registrar traza de entrega (entidad externa de la empresa)

5.1.2. Interfaces externas

Se han descrito las siguientes interfaces, dentro del punto de vista del contexto:

- Pago con tarjeta 1
- Consulta tipo SQL en BD externa 2
- Cambio de dirección de entrega en el sistema 3
- Petición de entrega al sistema interno de distribución 4

Todas son interfaces que se inician en el sistema y van hacia las entidades externas.

5.1.3. Estructura interna

Dado que la empresa A está especializada en comercio electrónico, construirá el sistema con software propio, la mayor parte del cual ya está desarrollado. Se asumen un estilo arquitectónico general de tipo “Cliente/Servidor”. La mayoría de la funcionalidad del lado del servidor está implementada en Java –servlets y java server pages (JSP)–, mientras que para el lado del cliente se usa sobre todo Java Script. En el lado del servidor, se asume como estilo arquitectónico general el de “Abstracción de datos y organización OO” (a partir de ahora hablaremos simplemente de estilo o enfoque OO).

Para mantener la máxima cohesión, se usará un mismo módulo para encapsular todo el software necesario de cada capacidad funcional completa (en el servidor de aplicaciones). Para mantener un mínimo acoplamiento, se unirán en el mismo módulo funciones que tengan una fuerte interacción. Se usa la siguiente tabla 2x2 para representar la interacción entre las distintas funciones, de forma que a partir de ella se decidirá el número final de módulos:

- Consultar catálogo productos (Tabla 1.2)
- Consultar la traza de una entrega (Tabla 1.2)
- Consultar compras (Tabla 1.2)
- Realizar compra (Tabla 1.2)
- Alta de usuario (Tabla 1.2)
- Gestión de cuentas de usuario (Tabla 1.2)
- Gestión del catálogo (Tabla 1.2)
- (v2) Elaboración de perfiles de comprador (Tabla 3.1.2)

Según las interdependencias de la Tabla 6, uniremos en un mismo paquete distintas funcionalidades, de la siguiente forma:

- M1: Catálogo de productos
 - Consultar catálogo productos (Tabla 1.2)
 - Gestión del catálogo (Tabla 1.2)
- M2: Traza de entrega
 - Consultar la traza de una entrega (Tabla 1.2)
- M3: Compra
 - Consultar compras (Tabla 1.2)
 - Realizar compra (Tabla 1.2)

	Cons. cat.	Cons. traza	Cons. compr.	Com- prar	Alta	Gest. cuent.	Gest. cat.	Elab. perfil
Cons. cat.	x						x	
Cons. traza		x						
Cons. compr.			x	x				
Comprar			x	x				
Alta					x	x		
Gest. cuent.					x	x		
Gest. cat.	x						x	
Elab. perfil								x

Tabla 6: Tabla 2x2 de interacción entre las distintas capacidades funcionales

- M4: Usuario
 - Alta de usuario (Tabla 1.2)
 - Gestión de cuentas de usuario (Tabla 1.2)
- M5: Perfil de compra
 - (v2) Elaboración de perfiles de comprador (Tabla 3.1.2)

5.1.4. Filosofía del diseño funcional

Además de la máxima cohesión y bajo acoplamiento, también en la estructura en paquetes descrita se da importancia a otros principios de diseño:

- Coherencia: se deduce de la aplicación de los principios de máxima cohesión y bajo acoplamiento
- Consistencia: las decisiones de diseño se aplican teniendo en cuenta a todo el sistema en su conjunto

Estos principios facilitarán la extensibilidad y flexibilidad funcional, atributos de calidad necesarios para este sistema, pues el cliente, B.B.B. desde el principio habla de cierta funcionalidad a añadir en el futuro (v.g. la cancelación o modificación de una compra).

5.1.5. Inquietudes para cada interesado en el sistema

En la Tabla 7 se asocian las inquietudes con las distintas partes interesadas.

Tipo de interesado	Inquietudes
Cliente (adquiriente)	Capacidades funcionales básicas, interfaces externas, en especial las que conectan con otros elementos internos a la empresa (v.g. sistema de distribución)
Desarrollador	Estructura interna y cualidades de diseño básicas; capacidades funcionales e interfaces externas
Ingeniero de producción y mantenimiento y webmaster	Estructura interna y cualidades de diseño básicas, interfaces externas y capacidades funcionales
Servicio al cliente	Capacidades funcionales básicas e interfaces externas; gestión de funciones configurables
Administrador del sistema	Filosofía de diseño funcional básica, interfaces externas y estructura interna
Técnico de pruebas	Estructura interna y cualidades de diseño básicas; capacidades funcionales e interfaces externas
Usuarios	Capacidades funcionales básicas e interfaces externas

Tabla 7: Inquietudes de los distintos tipos de partes interesadas en el punto de vista funcional. [Adaptado de fuente: (Rozanski, 2020, Tab. 2.6)]

5.2. Modelos

5.2.1. Modelo de la estructura funcional

Se debe utilizar un modelo donde se indiquen los siguientes componentes:

- Elementos funcionales
- Interfaces (de los elementos funcionales del sistema)
- Conectores (entre los elementos funcionales del sistema)

Las entidades externas NO deben formar parte de este modelo.

Rozansky (Rozanski, 2020) propone dos diagramas, uno de componentes y otro de cuadros y líneas. Se usará un diagrama de componentes UML en donde los conectores se representarán como dependencias UML y flujos de información (ver Figura 6). También se usa un diagrama de cuadros y líneas para hacer este punto de vista más sencillo de entender a las partes interesadas menos técnicas (ver Figura 7).

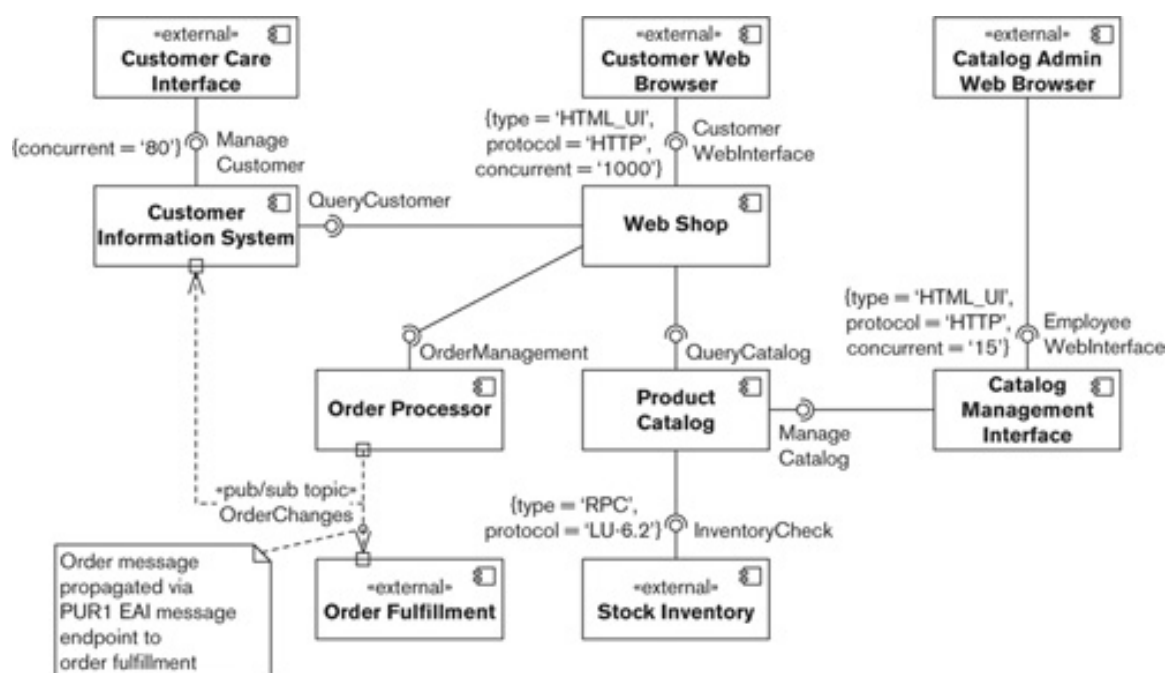


Figura 6: Ejemplo de diagrama de componentes UML para modelar una tienda Web. [Fuente: (Rozanski, 2020, Fig. 2.30)]

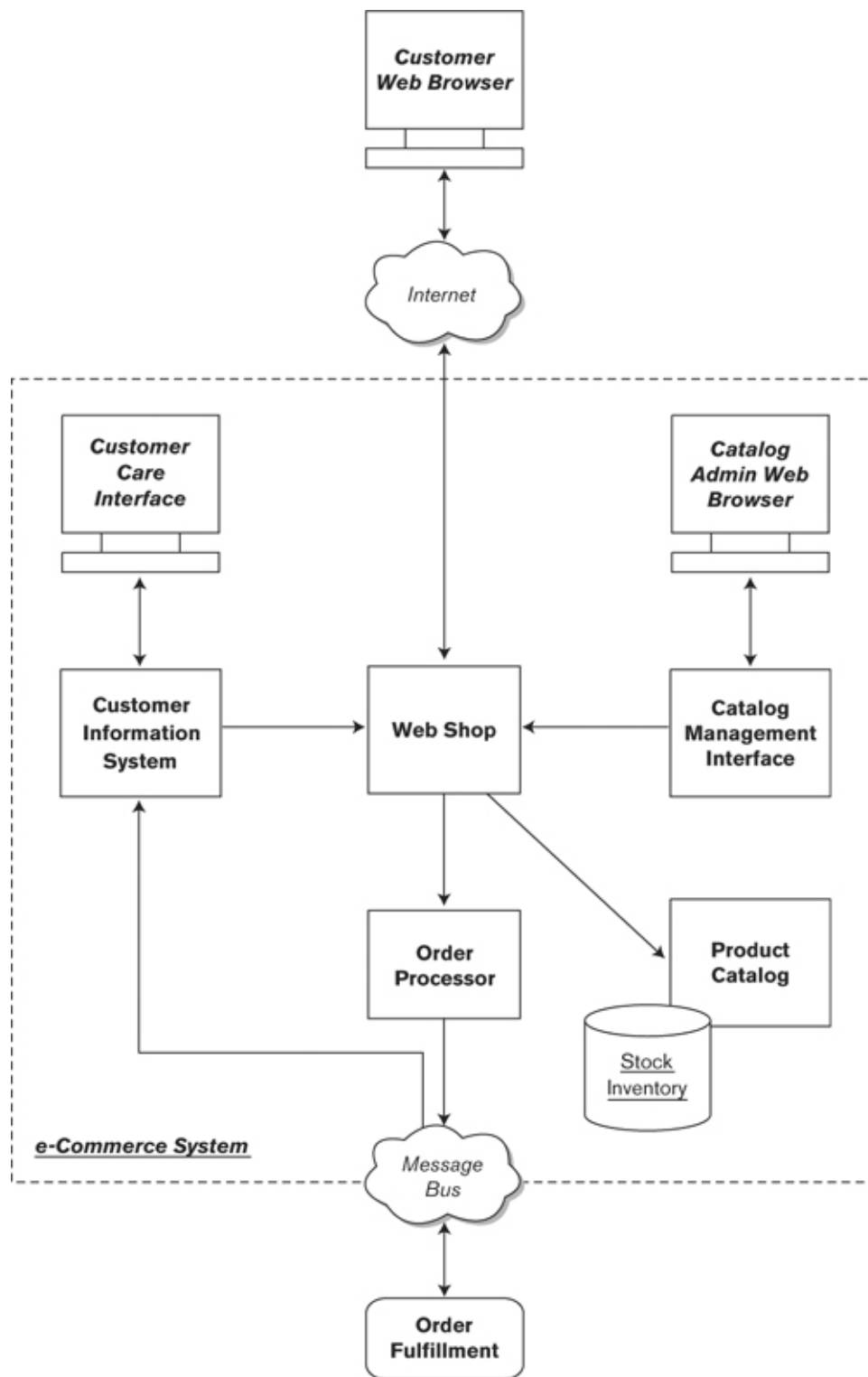


Figura 7: Ejemplo de diagrama de “cuadros y líneas” equivalente al diagrama de componentes UML para modelar una tienda Web que aparece en la Figura 6. Se ha usado una notación propia: las interfaces de usuario externas se representan por un icono que simboliza el monitor del ordenador, y los sistemas externos de respaldo se representan por rectángulos con esquinas redondeadas. Los almacenes de datos se representan por un icono que simboliza un tambor de discos y las interfaces funcionales (Internet, el bus de mensajes –Message Bus–) se representan mediante una nube. El ámbito del sistema incluye a todos los elementos dentro del rectángulo de línea discontinua. [Fuente: (Rozanski, 2020, Fig. 2.31)]

El diagrama de la Figura 6 se describe a continuación (en ([Rozanski, 2020](#), Cuadro 2.3.19)).

Tienda Web dentro de un entorno software ya existente en la empresa

La Figura 6 muestra un diagrama de componentes UML para describir la estructura funcional de un sistema sencillo. El sistema proporciona una tienda Web (Web Shop) que usan los clientes para compras a partir de un catálogo en línea y que será parte de un entorno software ya existente en la empresa.

El modelo muestra que el sistema interactúa con cinco entidades externas: los navegadores Web de los tres tipos de usuario más importantes (personal de servicio al cliente, clientes y administrados del catálogo de productos), y dos sistemas externos (sistema de reparto e inventario). El sistema en sí está compuesto por cinco componentes principales, unidos por distintos tipos de conectores (incluyendo HTML sobre HTTP, mensajería por publicación/subscripción y una interfaz externa LU 6.2 (desarrollada por IBM).

Los clientes hacen el pedido a través de la tienda Web (Web Shop), que interactúa a su vez con el gestor de pedidos (Order Processor), el catálogo de productos (Product Catalog) y el sistema de información del cliente (Customer Information System). Los administradores del catálogo mantienen el catálogo a través de su interfaz basada en la Web (Catalog Management Interface) y los que forman parte del equipo de servicio al cliente mantienen la información del cliente (Customer Information System) mediante un programa cliente de interfaz dedicada (Customer Care Interface). Cuando se desea consultar el stock de un producto determinado, el catálogo de productos (Product Catalog) obtiene la información del inventario (Stock Inventory), que es un sistema que ya existe.

Hay también cierta información en cuanto a la naturaleza de las interacciones entre componentes. Sabemos que pueden acceder al sistema simultáneamente hasta 1000 clientes, 80 personas del servicio al cliente y 15 administradores del catálogo. Además, la interacción entre el catálogo de productos y el inventario tiene lugar mediante un protocolo específico (que seguramente era un tecnología que ya existía por ser el inventario parte del software actual de la empresa). Podemos asumir según el ejemplo que la comunicación entre componentes que no ha sido descrita se lleva a cabo con algún procedimiento RPC (que será definido en algún otro lugar de la DA de forma clara).

Uno de los aspectos interesantes que se reflejan en este diagrama es la cantidad de deducciones no obvias que se pueden hacer a partir de él. Las responsabilidades de los componentes no está claras, tampoco los detalles de las interfaces ni los detalles de la interacción entre los distintos componentes. Esto nos hace entender la necesidad de añadir descripciones de texto y otros diagramas que modelen el sistema de forma complementaria, por ejemplo, las interacciones entre componentes puede mostrarse modelando escenarios del sistema.

Un diagrama de clases Dado que ya se conoce que se usará OO (Java) en el lado del servidor, se propone un diagrama de clases para el modelo funcional del sistema en el lado del servidor. Dado el enfoque OO, los módulos propuestos en la Sección 5.1.3, serán clases.

Al empezar a elaborar este diagrama se aprecia la ausencia de un requisito funcional, el de procesar una orden de entrega, y se añade a continuación:

Detalle	Descripción
RF#	9-v3
Nombre	Procesar un pedido
Descripción	Registra la elaboración de la cesta (real) de envío de cada compra y ordena su entrega al sistema de distribución
Entrada	Nueva cesta
Procesamiento	[Prerrequisito: acceso identificado como vendedor, pedido registrado como no preparado por el sistema de distribución] Se registra en la BD cada uno de los productos que se van atendiendo; se cierra la cesta; se espera hasta que la dirección de entrega del pedido coincida con la del sistema de distribución; se envía orden de entrega al sistema de distribución
Salida	Confirmación del procesamiento del pedido
Escenario 1 a destacar	Cesta vacía ¹ (la devolución del dinero es por ahora manual)
Escenario 2 a destacar	Cesta incompleta ² (la devolución del dinero es por ahora manual)

Por tanto, las capacidades funcionales actualizadas son ahora:

- Consultar catálogo productos (Tabla 1.2)
- Consultar la traza de una entrega (Tabla 1.2)
- Consultar compras (Tabla 1.2)
- Realizar compra (Tabla 1.2)
- Alta de usuario (Tabla 1.2)
- Gestión de cuentas de usuario (Tabla 1.2)
- Gestión del catálogo (Tabla 1.2)
- (v2) Elaboración de perfiles de comprador (Tabla 3.1.2)
- (v3) Procesar un pedido (Tabla 5.2.1)

Dado que el nuevo requisito funcional (RF9) es bastante independiente en funcionalidad al resto, se pondrá en una clase aparte, para no reducir la cohesión del sistema.

La Figura 8 muestra el diagrama de clases para el sistema (funcionalidad de lado del servidor).

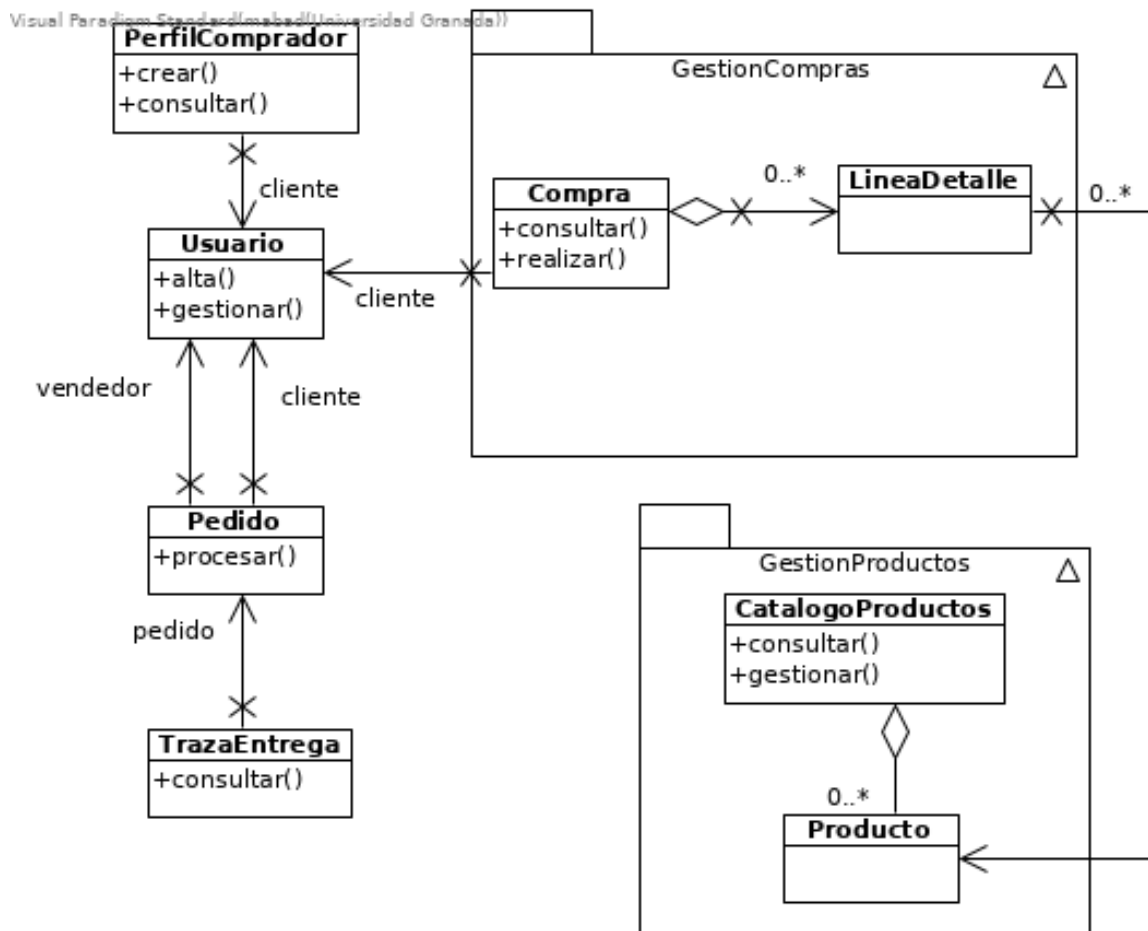


Figura 8: Ejemplo de diagrama de clases de análisis en UML para modelar el sistema del lado del servidor en una tienda Web.

5.3. Actividades

5.3.1. Identificar los elementos

Para elaborar el diagrama de clases, hemos partido de las capacidades funcionales y agrupado las mismas identificando los conceptos más importantes para deducir a partir de éstos las entidades (clases). Asimismo hemos aplicado sobre todo la composición para agrupar clases en paquetes. Otras tareas de refinamiento, como descomposición, generalización (herencia), etc. se dejarán para la fase de diseño.

5.3.2. Asignar responsabilidades a los elementos

A cada clase se le han asignado las capacidades que están dentro de un mismo módulo según la estructura interna (Sección 5.1.3).

Las responsabilidades de dos elementos del diagrama de componentes pueden verse a continuación (ver (Rozanski, 2020, Cuadro 2.3.20)):

Responsabilidades de dos elementos de la aplicación de comercio electrónico de venta al por menor antes descrita

- Elemento: Tienda Web
 - Proporcionar a los clientes una interfaz HTML accesible desde el navegador Web
 - Gestionar todos los estados relacionados con la sesión de la interfaz del cliente
 - Interaccionar con otras partes del sistema para permitir que el cliente pueda ver el catálogo de productos y el stock, comprar y consultar información sobre su cuenta
- Elemento: Sistema de Información al Cliente
 - Gestionar toda la información persistente relacionada con los clientes del sistema
 - Proporcionar una interfaz sólo-consulta al cliente que le permita consultar la información a la que tenga acceso
 - Proporcionar una interfaz programable de gestión de información que pueda usarse para crear aplicaciones de gestión de información de clientes
 - Proporcionar una interfaz dirigida por eventos de manejo de mensajes que acepte las líneas de detalle de los pedidos hechos por los clientes y los cambios de estado realizados sobre dichos pedidos

Las de los otros tres componentes del sistema, procesamiento de pedidos, interfaz para gestión del catálogo y catálogo de productos, se describen a continuación:

- Elemento: Procesamiento de pedidos
 - Proporcionar a los vendedores una interfaz HTML accesible desde el navegador Web
 - Gestionar todos los estados relacionados con la sesión de la interfaz del vendedor
 - Interaccionar con la BD para registrar el procesamiento de la cesta y transmitir la orden de entrega al sistema de distribución
- Elemento: Catálogo de productos
 - Gestionar (alta/baja/modificación/stock) de cada producto del sistema
 - Registrar cada transacción en la BD

- Elemento: Interfaz para gestión del catálogo
 - Permite que el administrador del catálogo (B.B.B., el adquiriente del sistema) pueda acceder al elemento “Catálogo de productos” para administrar los productos

5.3.3. Diseñar las interfaces

En este apartado hay que diseñar las interfaces entre los componentes del sistema. Las que tienen más relevancia en la arquitectura son las interfaces entre el servidor y los clientes (navegadores web).

Como los lenguajes de programación son distintos y la conexión no es síncrona por RPC sino basada en mensajería (REST), se usará para describirlas un enfoque orientado a datos.

De cada interfaz detallaremos los siguientes aspectos:

- Nombre (con la que se invoca)
- Semántica
- Tipo
- Prerrequisitos
- Entrada
- Salida
- Efecto

La mayoría de las interfaces surgen a partir de las capacidades funcionales, pues la mayor parte de ellas requieren software tanto en el lado del servidor como en el del cliente. Por eso de cada una de ellas describimos una interfaz. Una excepción es la capacidad definida a partir del RF8: elaboración de perfiles de compra, para la que no hace falta la intervención del usuario y se realiza por completo en el servidor. Sin embargo, hay una interfaz, consultarPerfilComprador, que sí describimos, relacionada con ese requisito funcional. No se ha considerado requisito funcional como tal porque sólo afecta al contenido Web (dinámico) que ve el cliente, según su perfil.

Sí se podría haber considerado ese contenido Web como un requisito de datos o información.

5.3.4. Diseñar los conectores

Todos los conectores entre cliente y servidor serán por mensajería HTTP (servicios REST), usarán páginas java del lado del servidor (JSP) cuando los datos se puedan pasar directamente a la BD o servlets (aplicación completa java del lado del servidor), cuando se requiera cierto procesamiento. Del lado del cliente, se usará en consultas pesadas, javascript y XML (AJAX) y en operaciones más ligeras, sólo javascript.

Detalle	Descripción
Nombre	consultarCatalogo
Semántica	muestra los productos del catálogo según algún criterio de búsqueda (categoría/novedades/todos)
Tipo	REST (mensajería): GET
Prerrequisitos	Estar identificado
Entrada	criterio de búsqueda
Salida	Listado productos

Detalle	Descripción
Nombre	nuevoProducto/modificarProducto/eliminarProducto
Semántica	operaciones de gestión de productos en el catálogo (hacia el servidor)
Tipo	REST (mensajería): PUT
Prerrequisitos	Estar identificado como gestor catálogo
Entrada	nombre (si es nuevoProducto) o cod. Producto (si es modificarProducto o eliminarProducto)
Salida	operación realizada

Detalle	Descripción
Nombre	consultarTraza
Semántica	muestra la traza de un envío
Tipo	REST (mensajería): GET
Prerrequisitos	Estar identificado como repartidor o cliente
Entrada	cód. pedido
Salida	información de la traza

Detalle	Descripción
Nombre	consultar compras
Semántica	se muestran las compras realizadas
Tipo	REST (mensajería): GET
Prerrequisitos	Estar identificado como repartidor o cliente
Entrada	criterio (todas/filtro por fecha/filtro por precio)
Salida	información de las compras seleccionadas

Detalle	Descripción
Nombre	comprar
Semántica	se envía el contenido del carrito para registrar la compra y se hace el pago con tarjeta
Tipo	REST (mensajería): PUT
Prerrequisitos	Estar identificado como cliente
Entrada	carrito compra (productos elegidos y cantidades)
Salida	confirmación compra

Detalle	Descripción
Nombre	altaUsuario
Semántica	se envían datos de usuario y se registra el alta
Tipo	REST (mensajería): PUT
Prerrequisitos	No estar identificado como cliente
Entrada	datos personales, email, contraseña, dirección postal
Salida	confirmación alta

Detalle	Descripción
Nombre	modificarDatoCuenta/bajaUsuario
Semántica	se envían datos a cambiar
Tipo	REST (mensajería): PUT
Prerrequisitos	Estar identificado como cliente
Entrada	dato a cambiar o nada si es para darse de baja
Salida	confirmación cambio o baja; si es baja, vuelta a acceso no identificado

Detalle	Descripción
Nombre	consultarPerfilComprador
Semántica	se personaliza la Web del cliente con productos del perfil
Tipo	REST (mensajería): GET
Prerrequisitos	Estar identificado como cliente
Entrada	-
Salida	datos de productos según perfil

Detalle	Descripción
Nombre	procesarPedido
Semántica	se informa de los detalles en la confección de la cesta a enviar
Tipo	REST (mensajería): PUT
Prerrequisitos	Estar identificado como vendedor
Entrada	-
Salida	confirmación procesamiento registrado y pendiente de entrega

5.3.5. Comprobar la trazabilidad funcional

Puede consultarse la Tabla 6 y la estructura interna elaborada a partir de ella (Sección 5.1.3).

5.3.6. Recorrer escenarios comunes

Por parte del desarrollador y la técnico de pruebas se revisan posibles escenarios de interacción entre cliente y servidor que deben ser recorridos pero se considera que ninguno es necesario tratarlo a nivel arquitectónico sino más adelante en la fase de diseño.

5.3.7. Analizar las interacciones

Se ha considerado en enfoque OO que, por centrarse en conceptos del mundo real que se trata de virtualizar, reduce de forma natural las interacciones entre conceptos (bajo acoplamiento) y aumenta las internas a un mismo concepto (alta cohesión).

5.3.8. Analizar la flexibilidad

La estructura OO es muy apropiada para las ampliaciones del sistema que ya se conocen: cancelar/modificar una compra y usar otros medios de pago. En general, cuando se trabaja con conceptos lo más cerca de la realidad que sea posible más fácil se hace tanto el mantenimiento como la evolución del sistema.

5.4. Problemas y errores comunes

Se pone sólo un guión (-) cuando se haya considerado el problema y se crea que está bajo control.

- Interfaces pobremente definidas: -
- Responsabilidades no bien entendidas: -
- Infraestructuras modeladas como elementos funcionales: -
- Vista sobrecargada: -
- Diagramas sin definiciones de elementos: el diagrama de clases no presenta conectores ni interfaces, pero estos ya se han puesto en el diagrama de componentes y además han sido descritos de forma textual
- Dificultades para reconciliar las necesidades de distintas partes interesadas: -
- Nivel de detalle erróneo: -
- “Elementos divinos”: -
- Demasiadas dependencias: -

5.5. Lista de verificación

- ¿Tiene menos de 15 a 20 elementos de nivel superior? Sí
- ¿Tienen todos los elementos un nombre, responsabilidades claras e interfaces claramente definidas? Sí
- ¿Tienen lugar todas las interacciones de elementos a través de interfaces y conectores bien definidos que unan las interfaces? Sí, aunque se describen de forma textual
- ¿Exhiben los elementos un nivel apropiado de cohesión? Sí

- ¿Exhiben los elementos un nivel apropiado de acoplamiento? Sí
- ¿Ha identificado los escenarios de uso importantes y los ha utilizado para validar la estructura funcional del sistema? No, se considera que los que restan es mejor dejarlo para la fase de diseño. Los escenarios más importantes, relacionados con la interacción con entidades externas, han sido considerados en el punto de vista del contexto
- ¿Ha verificado la cobertura funcional de su arquitectura para asegurarse de que cumple con los requisitos funcionales? Sí
- ¿Ha definido y documentado un conjunto apropiado de principios de diseño arquitectónico y su arquitectura cumple con estos principios? Sí
- ¿Ha considerado cómo es probable que la arquitectura haga frente a posibles escenarios de cambio en el futuro? Sí
- ¿La presentación de la vista tiene en cuenta las preocupaciones y capacidades de todos los grupos de partes interesadas? ¿Actuará la vista como un vehículo de comunicación eficaz para todos estos grupos? Sí, se han realizado tres modelos distintos para lograrlo

6. Descripción detallada de la perspectiva de evolución

6.1. Aplicabilidad

Los dos puntos de vista descritos hasta ahora se ven afectados por esta perspectiva.

Desde el punto de vista del contexto, y observando el diagrama UML de la Figura 1, vemos que hay tres entidades externas, y dos de ellas que interaccionan entre sí. Se podrían añadir otras, incorporar alguna de las externas al sistema y/o cambiar las interfaces.

Pero el que más probablemente se verá afectado será el punto de vista funcional, pues la adición de nueva funcionalidad al sistema suele ser bastante más frecuente que cambios que afecten también al contexto, a no ser que sean impuestos por el propio contexto.

Observando el diagrama UML de la Figura 6, los cambios podrían incluir nuevos componentes funcionales, o la adición de funcionalidad dentro de componentes concretos. También podrían afectar a las interfaces de los distintos componentes y a los conectores entre componentes. Podría también modificarse (aumentarse) la funcionalidad en los clientes Web y no sólo en el núcleo más importante de la aplicación, que se ejecuta en el servidor.

6.2. Inquietudes

Las inquietudes en la perspectiva de evolución se refieren a los criterios de calidad que se van a considerar para llevar a cabo la evolución del sistema.

Aunque la lista propuesta (Rozanski, 2020, Sec. 2.3.2.10.2) no es ni suficiente ni necesaria, hemos considerado que a priori y sin conocer otras posibles opciones, todos los criterios propuestos deben usarse para nuestro sistema:

- Considerar el sistema como un producto para estimar la evolución en su conjunto
- Magnitud del cambio.- Ser capaz de apreciar la magnitud del cambio, evitará grandes costos (bien en el primer desarrollo, por exceso en la estimación o bien en el mantenimiento, por defecto en la estimación de la misma)
- Dimensión del cambio.- A nivel funcional, de plataforma, de integración –contextual–, crecimiento de uso
- Probabilidad del cambio.- Ser muy precisos en la apreciación de que los cambios sean realmente necesarios
- Temporización del cambio.- Estimar el momento probable para realizarlo
- Elegir el momento del cambio equilibrando los costes de desarrollo (mayores a mayor flexibilidad del sistema a desarrollar) y los costes de mantenimiento (mayores a mayor velocidad en el desarrollo –conseguida con una mayor simplicidad del sistema a desarrollar–)
- Considerar los cambios impuestos por factores externos (fin de la vida útil, cambio en interfaces externas, cambios de regulación, cambios en la empresa, en la población de usuarios o perfiles)
- Preservar el conocimiento mediante alguna documentación para cuando cambie el equipo de mantenimiento
- Probar los cambios (automatizar las pruebas, hacer una buena gestión de la configuración, abstraer procesos repetibles, etc.

6.3. Actividades

Se proponen una secuencia de actividades compuesta por cuatro pasos ([Rozanski, 2020](#), Fig. 2.37).

6.3.1. Primer paso: Caracterizar las necesidades evolutivas

Se hace una relectura de las capacidades funcionales, detectando indicios de distintos tipos:

- Funciones diferidas: Aparecen en la lista de requisitos/capacidades clasificadas como “aún no”
 - Cancelar/modificar compra (aún no)
 - Pagar por otros medios (aún no)
- Lagunas en los requisitos:
 - Consultar catálogo usando otros criterios de búsqueda

- Requisitos vagos o indefinidos: -
- Requisitos abiertos:
 - Elaboración de perfiles de comprador, cambio en el método de elaboración

Los describimos en las tablas 6.3.1 a 6.3.1.

Detalle	Descripción
RE#	1
Nombre	Cancelar compra
Tipo de cambio:	funcional
Magnitud del cambio (Alta/Media/Baja):	Alta
Probabilidad del cambio (Alta/Media/Baja)	Alta
Escala de tiempo de los cambios	De varios meses a 2 años, según crecimiento del sistema

Detalle	Descripción
RE#	2
Nombre	Modificar compra
Tipo de cambio:	funcional
Magnitud del cambio (Alta/Media/Baja):	Alta
Probabilidad del cambio (Alta/Media/Baja)	Alta
Escala de tiempo de los cambios	2 años, según crecimiento del sistema

Detalle	Descripción
RE#	3
Nombre	Pagar por otros medios
Tipo de cambio:	de integración
Magnitud del cambio (Alta/Media/Baja):	Media
Probabilidad del cambio (Alta/Media/Baja)	Baja
Escala de tiempo de los cambios	Imprecisa según crecimiento del sistema

Priorización La tabla 8 muestra el orden (columna 7^a) de los distintos requisitos evolutivos, de mayor prioridad (1) a menor (5).

³Alta:3, media: 2, baja: 1

⁴Alta:0,8, media: 0,5, baja: 0,2

⁵ $magnitud_{absoluta} * probabilidad_{cambio}$

Detalle	Descripción
RE#	4
Nombre	Consultar catálogo de productos usando otros criterios de búsqueda
Tipo de cambio:	funcional
Magnitud del cambio (Alta/Media/Baja):	Baja
Probabilidad del cambio (Alta/Media/Baja)	Media
Escala de tiempo de los cambios	1-2 años

Detalle	Descripción
RE#	5
Nombre	Elaboración de perfiles de comprador, cambio en el método de elaboración
Tipo de cambio:	funcional
Magnitud del cambio (Alta/Media/Baja):	Media
Probabilidad del cambio (Alta/Media/Baja)	Alta
Escala de tiempo de los cambios	1-2 años

Requisito Evolutivo	Magnitud absoluta ³	Prob. de cambio ⁴	Magnitud relativa ⁵	Tiempo	Puntuación	Orden
RE1	3	0,8	2,4	10	0,24	1
RE2	3	0,8	2,4	24	0,1	2
RE3	2	0,2	0,4	60	0,07	4
RE4	1	0,5	0,5	18	0,028	5
RE5	2	0,8	1,6	18	0,09	3

Tabla 8: Priorización de ls requisitos de evolución

6.3.2. Segundo paso: Evaluar la facilidad para evolucionar en la actualidad

Los revisamos según orden de prioridad:

- RE1 (Cancelar compra): **NO**, no parece teer un coste razonable incluir esta funcionalidad en pocos meses si el uso del sistema creciera mucho. Los cambios serían simples pero afectarían a muchos elementos del sistema y a sus interfaces y también a las entidades externas: varias clases del servidor, interfaz de usuario (software en el lado del cliente), BD externa, sistema externo de distribución. Sería mejor incluir este requisito evolutivo como un requisito funcional más.
- RE2 (Modificar compra): **SÍ**, si se considera que se puede incluir en una revisión del sistema a realizar en dos años. Se requerirían cambios parecidos a los especificados en el requisito anterior, pero se dispone de más tiempo.
- RE5 (Elaboración de perfiles de comprador: cambio en el método de elaboración): **SÍ**, los cambios sólo afectarían a un método, o varios pero perfectamente encapsulados

dentro de la clase PerfilComprador (ver Figura 8.

- RE3 (Pagar por otros medios): Dado que es un requisito con temporización imprecisa, no se puede decir nada sobre su viabilidad económica.
- RE4 (Consultar catálogo de productos usando otros criterios de búsqueda): **SÍ**, incluso en un año sería posible un cambio a coste razonable pues requeriría muy pequeños cambios aunque afectarían tanto al cliente Web (interfaz de usuario) como al servidor (clase CatalogoProductos).

6.3.3. Tercer paso: Considerar las contrapartidas de la evolución

Lo mejor sería incluir ya como requisito funcional el RE#1 dados los elevados costes de evolución y la alta probabilidad de tener que hacerlos después.

En cuanto a los otros requisitos, lo más conveniente es:

- Software cliente: Dar mucha flexibilidad en la elaboración de la interfaz de usuario de forma que se puedan incluir/modificar/quitar opciones en la página. La empresa de desarrollo usa un gestor de contenidos para comercio electrónico, y dados los cambios previstos parece adecuado usarlo en el desarrollo de esta tienda Web.
- Software servidor: Usar OO y un buen diseño da la suficiente flexibilidad implícita para poder incorporar los nuevos requisitos a un coste razonable. Parece eso más adecuado que el coste que podría derivarse de hacer un sistema especialmente flexible.

6.3.4. Cuarto paso: Revisar la arquitectura

De los 5 requisitos evolutivos, el primero se ha decidido impementarlo por completo desde el principio, pasando por tanto a ser un requisito funcional más.

Se describe este nuevo requisito funcional (RF10, Tabla 5.2.1), en sustitución del RE1.

Se revisan ahora los distintos puntos de vista:

Punto de vista de contexto:

RE1: Cancelar compra

Como consecuencia de la conversión a nuevo requisito funcional, se añadirá una nueva interfaz (del sistema externo de pago): Tabla 9.

Esta nueva interfaz no afecta a los diagramas de contexto (Figuras 1, 2 y 3), por no especificar las interfaces con detalle.

Detalle	Descripción
RF#	10-v4
Nombre	Cancelar compra
Descripción	Permite cancelar una compra
Entrada	identificador de la compra
Procesamiento	[Prerrequisito: acceso identificado como cliente o como vendedor] Comprobación de estado; (si no entregado) realización de la devolución (sistema externo); (si no entregado) anulación del pedido en el sistema externo de preparación y entrega del pedido; actualización del stock en el catálogo; registro de la cancelación en la BD;
Salida	Confirmación de la cancelación
Escenarios a destacar	El pedido ya ha sido entregado. La cancelación la hará el vendedor cuando lo compruebe
Excepción: cancelación interrumpida por cliente (pin no enviado, ventana cerrada ...)	Registro del error

Detalle	Descripción
Nombre	Devolución pago con tarjeta
Semántica	Permite hacer la devolución de un pago con tarjeta de crédito en el sistema bancario para pago con tarjeta
Parámetros	id. comercio, id. transacción, fecha transacción y pin enviado a móvil
Acciones a realizar	devolución en tarjeta
Excepciones (1. error en datos)	devuelve código excepción 1
Excepciones (2. operación no autorizada)	devuelve código excepción 2
Errores: pago interrumpido por cliente (pin no enviado, ventana cerrada ...)	a tratar por el sistema a implementar

Tabla 9: Interfaz de servicio I5v2

RE2: Modificar compra

Sí se espera que pueda ser implementado en el futuro. Se necesitaría una nueva interfaz con el sistema de pagos que permita hacer e una misma transacción un abono y cargo a una tarjeta de crédito, pero esto no afectaría al diagrama de contexto.

RE3: Pagar por otros medios

El RE3 no se va a considerar en absoluto, dada su vaguedad en el tiempo y probabilidad posible de implementación.

RE4: Consultar catálogo de productos usando otros criterios de búsqueda

Aunque se espera implementar en el futuro, el punto de vista contextual no se ve afectado en absoluto.

RE5: Elaboración de perfiles de comprador: cambio en el método de elaboración)

Aunque se espera implementar en el futuro, el punto de vista contextual tampoco se ve afectado en el caso de que se implementara.

Punto de vista funcional:

En el lado del cliente, los cambios no afectarían a la programación, ya que se va a utilizar un gestor de contenidos para dar máxima flexibilidad a la programación de la GUI.

RE1: Cancelar compra

Se actualiza la tabla de interacciones (Tabla ??):

Según las nuevas interdependencias de la Tabla ??, sólo el paquete 3 se ve afectado:

- M3v2: Compra
 - Consultar compras (Tabla 1.2)
 - Realizar compra (Tabla 1.2)
 - Cancelar compra (Tabla 6.3.4)

Las capacidades funcionales actualizadas son ahora:

- Consultar catálogo productos (Tabla 1.2)
- Consultar la traza de una entrega (Tabla 1.2)
- Consultar compras (Tabla 1.2)
- Realizar compra (Tabla 1.2)
- Alta de usuario (Tabla 1.2)
- Gestión de cuentas de usuario (Tabla 1.2)

	Cons. cat.	Cons. traza	Cons. compr.	Com- prar	Cancelar compra	Alta	Gest. cuent.	Gest. cat.	Elab. perfil
Cons. cat.	x							x	
Cons. traza		x							
Cons. compr.			x	x	x				
Comprar			x	x	x				
Cancelar compra			x	x	x				
Alta						x	x		
Gest. cuent.						x	x		
Gest. cat.	x							x	
Elab. perfil									x

Tabla 10: Tabla 2x2 de interacción entre las distintas capacidades funcionales -v2

- Gestión del catálogo (Tabla 1.2)
- (v2) Elaboración de perfiles de comprador (Tabla 3.1.2)
- (v3) Procesar un pedido (Tabla 5.2.1)
- (v4) Cancelar compra (Tabla 6.3.4)

Se añadirá además una interfaz interna (Tabla 6.3.4):

Detalle	Descripción
Nombre	cancelar compra
Semántica	se envía el código de la compra a cancelar y se hace la devolución a la tarjeta
Tipo	REST (mensajería): PUT
Prerrequisitos	Estar identificado como cliente
Entrada	Código operación de compra
Salida	confirmación compra

RE2: Modificar compra

En el lado del servidor, afectaría sólo a la clase *Compra* (ver nuevo diagrama de clases de análisis en Figura 8).

RE3: Pagar por otros medios

El RE3 no se va a considerar en absoluto, dada su vaguedad en el tiempo y probabilidad posible de implementación.

RE4: Consultar catálogo de productos usando otros criterios de búsqueda

Afectaría a la clase CatalogoProductos, aunque no al diagrama de clases de análisis.

RE5: Elaboración de perfiles de comprador: cambio en el método de elaboración)

Afectaría a la clase PerfilComprador, aunque no al diagrama de clases de análisis.

Diagrama de clases de análisis El diagrama de clases del servidor para recoger todos los requisitos nuevos (RF10 y el requisito evolutivo RE2) quedaría como aparece en la Figura 9.

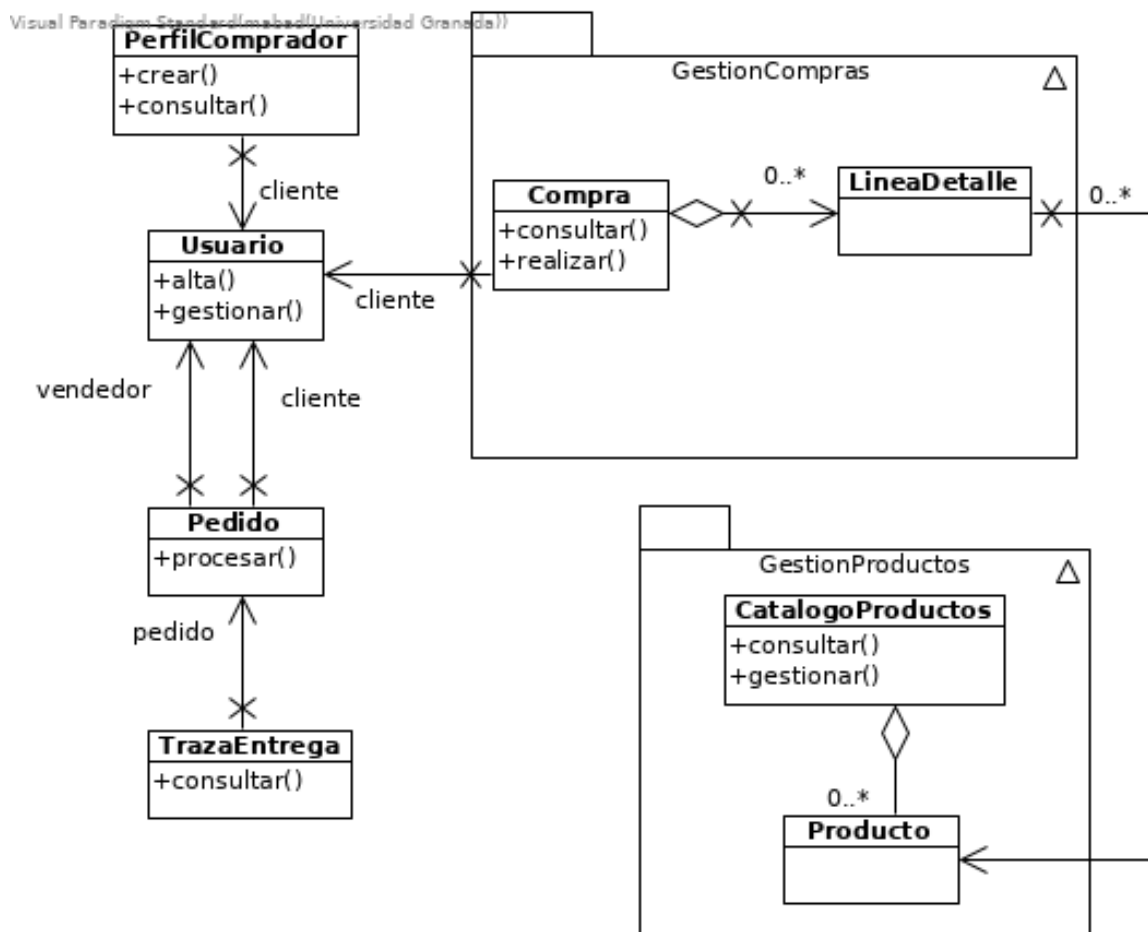


Figura 9: Ejemplo de diagrama de clases de análisis en UML para modelar el sistema del lado del servidor en una tienda Web (v2).

6.4. Tácticas arquitectónicas

Se resumen aquí las tácticas arquitectónicas adoptadas:

- Aislar los cambios.- En los casos en los que, a pesar de usar el enfoque OO y un buen diseño, para permitir encapsulación y alta cohesión, los cambios requeridos impliquen a muy diversas partes del sistema (como el caso del RE1), y cuando las probabilidades de tener que implementarlo a corto plazo son muy altas, se ha optado por incorporar su implementación desde el principio, transformando lo que el adquirente, B.B.B., expresó con futuroble, a un requisito funcional más.
- Aplicar abstracción.- Las clases de análisis *CatalogoProductos* y *PerfilComprador* se convertirán en clases abstractas en el diagrama de clases de diseño, de forma que sea fácil incorporar los requisitos evolutivos RE4 y RE5 respectivamente, creando nuevas subclases concretas.

- Usar un gestor de contenidos para la GUI.- Para dar máxima flexibilidad a la GUI y dado que hay otros requisitos evolutivos con bastante probabilidad de que tengan que implementarse y que afectarán al diseño de las páginas Web, se ha optado por el uso de un gestor de contenidos “ligero”⁶ de forma que la adición de nueva funcionalidad no requiera programar nada desde el lado del cliente.

6.5. Problemas y errores

Transcribo aquí los errores comunes. Se analizan con las distintas partes interesadas para detectar y corregir ahora el máximo número de ellos. Se pone sólo un guión (-) cuando se haya considerado el problema y se crea que está bajo control.

- Priorización incorrecta de las dimensiones: -
- Cambios que nunca suceden: -
- Impactos de la evolución en criterios críticos de calidad: -
- Dependencia excesiva de hardware o software específico: El adquiriente considera que pueden depender demasiado de una empresa especializada en aplicaciones Web, se queda conforme tras considerar el siguiente posible problema
- Ambientes de desarrollo perdidos: La empresa documenta de forma sistemática la incorporación de tecnología externa y además se compromete a guardar imágenes de disco para cada nuevo lanzamiento de la aplicación. Además usa un sistema de control de versiones a disposición del adquiriente

6.6. Lista de comprobación

6.6.1. Lista de comprobación para captura de los requisitos:

- ¿Has considerado qué dimensiones evolutivas son más importantes para tu sistema? Sí
- ¿Confías en que has realizado un análisis suficiente para confirmar que tu priorización de las dimensiones evolutivas es válida? Sí
- ¿Has identificado cambios específicos particulares que se requerirán y la magnitud de cada uno? Sí
- ¿Has evaluado la probabilidad de que cada uno de esos cambios sea realmente necesario? Sí

⁶En el sentido de que no gestionará la parte del lado del servidor (ni aún menos la BD, que es externa al sistema), sino sólo los links para conectar con el mismo

6.6.2. Lista de comprobación para la DA:

- ¿Has realizado una evaluación arquitectónica para establecer si tu arquitectura es lo suficientemente flexible como para satisfacer las necesidades evolutivas del sistema?
Sí
- Cuando el cambio es probable, ¿tu diseño arquitectónico contiene el cambio en la medida de lo posible? Sí
- ¿Has considerado elegir un estilo arquitectónico inherentemente orientado al cambio?
Si es así, ¿has evaluado los costos de hacerlo? Sí
- ¿Has cambiado los costos de tu apoyo a la evolución por las necesidades del sistema en su conjunto? Sí
¿Alguna propiedad de calidad crítica se ve afectada negativamente por el diseño que has adoptado? El gestor de contenidos hace algo menos eficiente el sistema
- ¿Has diseñado la arquitectura para acomodar sólo aquellos cambios que crees que serán necesarios? Sí
- ¿Puedes recrear sus entornos de desarrollo y prueba de manera confiable? Sí el de desarrollo, no el de prueba
- ¿Puedes construir, probar y lanzar de manera confiable y repetible el sistema, incluida la capacidad de revertir los cambios si salen mal? Sí construcción y lanzamiento, no pruebas
- ¿Es tu enfoque evolutivo elegido la opción más barata y menos arriesgada de entregar el sistema inicial y la evolución futura requerida? Creo que sí

7. Descripción detallada de la perspectiva de seguridad

7.1. Aplicabilidad

En este documento sólo se han descrito dos puntos de vista: contextual y funcional. Un punto de vista muy importante a nivel de seguridad es el informacional o de datos. Aunque en este DA parcial no se ha incluido, sí que se abordará la seguridad también con respecto a los datos permanentes (bases de datos).

Desde la perspectiva contextual, hay que recordar que el acceso al sistema es por parte de actores. No hay otros sistemas que accedan al nuestro, salvo en respuesta a peticiones, es decir, nuestro sistema no proporciona servicios a otros sistemas.

Desde la perspectiva funcional, habrá que proteger todos los subsistemas que procesen datos sensibles, los cuales se especificarán más adelante, en el apartado de actividades.

Desde la perspectiva de la información, hay que proteger el acceso a los datos sensibles en sí y a sus lugares físicos de almacenamiento.

7.1.1. Actividades

Paso 1: Identificación de los recursos

Partiendo del punto de vista funcional, los tipos de recursos que pueden identificarse a partir de los diagramas, en función del tipo de acceso son:

- Gestión de las cuentas de usuarios, y de los clientes en especial
- Gestión del catálogo de productos, y de los precios en especial
- Gestión de las compras

Partiendo del punto de vista de la información, los tipos de recursos que pueden identificarse según su sensibilidad son:

- Registros de cuentas de clientes
- Catálogo de productos
- Precios de los productos
- Registro de operaciones realizadas por clientes (compras)

Los principales pueden también clasificarse (en roles) en función de los recursos a los que pueden acceder:

7.1.2. Principales

- Cliente
- Gestor de catálogo
- Vendedor (quien prepara un pedido)
- Repartidor (quien entrega un pedido)
- Administrador
- Superusuario

La Figura 10 muestra la Tabla 25-2 Rozanski (2011) en donde se identifican los recursos en la Tienda Web.

Resource	Sensitivity	Owner	Access Control
Customer account records	Personal information of value for identity theft or invasion of privacy	Customer Care Group	No direct data access
Descriptive product catalog entries	Defines what is for sale and its description; if maliciously changed, could harm the business	Stock Management Group	No direct data access
Pricing product catalog entries	Defines pricing for catalog items; if maliciously or accidentally modified, could harm the business or allow fraud	Pricing Team in Stock Management Group	No direct data access
Business operations on customer account records	Needs to be controlled to protect data access and integrity	Customer Care Group	Access to individual record or all records by authenticated principal
Descriptive catalog operations	Needs to be controlled to protect data access and integrity	Stock Management Group	Access to catalog modification operations by authenticated principal
Pricing catalog modification operations	Needs to be controlled to protect data access and integrity	Pricing Team	Access to price modification operations by authenticated principal, with accountability of changes
...

Figura 10: Tabla 25-2 donde se identifican algunos de los recursos sensibles en el sistema de la Tienda Web. [Fuente:([Rozanski, 2011](#), Tabla 25-2)]

Paso 2: Definición de la política de seguridad

La Figura 11 muestra la Tabla 25-3 [Rozanski \(2011\)](#) en donde se identifican las políticas de control de acceso en la Tienda Web.

	User Account Records	Product Catalog Records	Pricing Records	User Account Operations	Product Catalog Operations	Price Change Operations
Data administrator	Full with audit	Full with audit	Full with audit	All with approval and audit	All with audit	All with approval from a product price administrator
Catalog clerk	None	None	None	All	Read-only operations	None
Catalog manager	None	None	None	Read-only operations with audit	All	All with audit
Product price administrator	None	None	None	None	Read-only operations	All with audit
Customer care clerk	None	None	None	All with audit	Read-only operations	None
Registered customer	None	None	None	All on own record	Read-only operations	None
Unknown Web-site user	None	None	None	None	Read-only operations	None

Figura 11: Tabla 25-3 donde se muestra parte del resultado de definir la política de seguridad para el sistema de Tienda Web. [Fuente:([Rozanski, 2011](#), Tabla 25-3)]

En este paso se han descrito 7 tipos distintos de principales según el control de acceso y 5 tipos distintos de recursos sensibles.

Paso 3: Identificación de las amenazas del sistema

Para elaborarlo, respondemos a las siguientes preguntas:

- ¿Quién es probable que quiera saltarse la política de seguridad?:
 - Hackers que quieran conseguir información referente a tarjetas de crédito
 - Un vendedor que quiera cambiar el stock de un producto
- ¿Qué motivación tiene el atacante para atacar el sistema?
 - Hackers: uso fraudulento de una tarjeta de crédito
 - Vendedores: apropiación indebida de productos de la tienda
- ¿Cómo tratará de saltarse la política de seguridad?
 - Hackers:

- Obteniendo datos de la base de datos (no es posible en la variante de nuestro sistema a la propuesta de Rozansky)
 - Obteniendo los datos desde la interfaz Web
 - Obteniendo los datos de un sistema externo
- Vendedor: robando contraseña
- ¿Cuáles son las principales características del atacante (sofisticación, compromiso, recursos, etc.)?
 - Hackers: mucha experiencia, atacan a otros sistemas parecidos, dedicación quasi-profesional
 - Vendedores: ninguna experiencia y baja motivación
- ¿Cuáles son las consecuencias de que se salte la política de esta forma?
 - Hackers: la reputación del negocio bajará tanto que lo haría caer en bancarrota
 - Vendedores: disminución (baja) de las ganancias para el negocio

Para el objetivo de conseguir detalles de tarjeta de crédito, el cuadro 7.1.2 muestra un árbol de ataque (ver (Rozanski, 2020, Cuadro 2.3.31)).

EJEMPLO: Árbol de ataque

Este es un posible árbol de ataque con el objetivo de extraer los datos de la tarjeta de crédito del cliente de un sitio web de comercio electrónico:

Objetivo: obtener los datos de la tarjeta de crédito del cliente.

1. Extraer detalles de la base de datos del sistema
 - a) Acceder a la base de datos directamente
 - 1) Descifrar / adivinar contraseñas de bases de datos
 - 2) Romper / adivinar las contraseñas del sistema operativo que permiten evitar la seguridad de la base de datos
 - 3) Explotar una vulnerabilidad conocida en el software de la base de datos
 - b) Acceder a los detalles a través de un miembro del personal de administración de la base de datos
 - 1) Sobornar a un administrador de base de datos (DBA)
 - 2) Realizar ingeniería social por teléfono / correo electrónico para engañar al DBA para que revele detalles
2. Extraer detalles de la interfaz web
 - a) Configurar un sitio web ficticio y enviar por correo electrónico a los usuarios la URL para engañarlos para que introduzcan los detalles de la tarjeta de crédito
 - b) Descifrar / adivinar contraseñas para cuentas de usuario y extraer detalles de la interfaz web del usuario
 - c) Enviar a los usuarios un programa troyano por correo electrónico para grabar pulsaciones de teclas / interceptar el tráfico web
 - d) Atacar el servidor de nombres de dominio para secuestrar el nombre de dominio y use el ataque de sitio ficticio de 2.1
 - e) Atacar el software del servidor del sitio directamente para tratar de encontrar lagunas en su seguridad o configuración o para aprovechar una vulnerabilidad conocida en el software
3. Encontrar detalles fuera del sistema
 - a) Realizar ingeniería social por teléfono / correo electrónico para que el personal de servicio al cliente revele los detalles de la tarjeta
 - b) Dirigir un ataque de ingeniería social a los usuarios mediante el uso de detalles públicos del sitio para hacer contacto (ver también 2.1)

Paso 4: Diseño de los mecanismos de seguridad a implementar

El cuadro 7.1.2 muestra un ejemplo de medidas de seguridad a adoptar para las amenazas identificadas en el árbol de ataque del ejemplo 7.1.2, en un sistema de comercio electrónico (ver (Rozanski, 2020, Cuadro 2.3.32)).

EJEMPLO: Diseño de medidas de seguridad a adoptar

- Aislar las máquinas de la base de datos de la red pública utilizando la tecnología de firewall de red
- Aislar las partes sensibles a la seguridad del sistema de la red pública utilizando la tecnología de firewall de red
- Analizar las rutas en el sistema para verificar si hay vulnerabilidades posibles
- Organizar pruebas de penetración para ver si los expertos pueden encontrar formas de entrar en el sistema
- Identificar una estrategia de detección de intrusos que permita reconocer las violaciones de seguridad
- Capacitar al personal de administración y servicio al cliente (de hecho, probablemente todo el personal) para evitar ataques de ingeniería social y acatar estrictos procedimientos de protección de la privacidad de la información del cliente
- Diseñar el sitio Eeb para que una cantidad mínima de información del usuario (idealmente, ninguna) sea visible públicamente
- Diseñar el sitio Web para que la información confidencial (por ejemplo, números de tarjetas de crédito) nunca se muestren en su totalidad (por ejemplo, muestre solo los últimos cuatro dígitos para permitir que los usuarios legítimos identifiquen sus tarjetas en las listas)
- Aplicar de forma rutinaria actualizaciones software relacionadas con la seguridad a todo el software de terceros utilizado en el sistema
- Revisar el código del sistema para detectar vulnerabilidades de seguridad utilizando herramientas de análisis e inspección experta
- Recordar constantemente a los usuarios las precauciones de seguridad que deben tomar (por ejemplo, no revelar las contraseñas a nadie, incluido su personal; verificar las URL antes de ingresar información, etc.)

Paso 5: Evaluación de los riesgos de seguridad

La Figura 12 muestra la Tabla 25-4 [Rozanski \(2011\)](#) en donde se muestra como ejemplo la evaluación de tres de los riesgos identificados en el sistema de la Tienda Web.

Risk	Estimated Cost	Estimated Likelihood	Notional Cost
Attacker gains direct database access	\$8,000,000	0.2%	\$16,000
Web-site flaw allows free orders to be placed and fulfilled	\$800,000	4.0%	\$32,000
Social-engineering attack on a customer service representative results in hijacking of customer accounts	\$4,000,000	1.5%	\$60,000
...

Figura 12: Tabla 25-4 donde se muestra la evaluación de tres de los riesgos identificados en el sistema de la Tienda Web. [Fuente:([Rozanski, 2011](#), Tabla 25-4)]

Tácticas arquitectónicas

Problemas y errores comunes

Lista para capturar todos los requerimientos de seguridad:

- ¿Has identificado los recursos sensibles del sistema? Sí
- ¿Has identificado los conjuntos de principales que necesitan acceder a los recursos? Sí
- ¿Has identificado las necesidades que tiene el sistema de garantizar la integridad de la información? Sí
- ¿Has identificado las necesidades de disponibilidad del sistema? Sí
- ¿Has establecido una política de seguridad para definir las necesidades de seguridad del sistema, junto con los principales y los permisos de acceso que tiene cada uno a cada recurso, y cuándo debe comprobarse la integridad de la información? Sí
- ¿Es la política de seguridad lo más simple posible? Sí
- ¿Has recorrido un modelo formal de amenazas para identificar los riesgos de seguridad del sistema? Sí
- ¿Has considerado tanto las amenazas externas como las internas al sistema? Sí
- ¿Has considerado cómo el entorno de despliegue del sistema alterará las amenazas del sistema? No

- ¿Has recorrido escenarios de ejemplos junto con las partes interesadas en el sistema de forma que puedan entender la política de seguridad planificada y los riesgos del sistema? No
- ¿Has revisado los requisitos de seguridad con expertos externos en seguridad? No

La otra lista proporcionada es directamente aplicable en la DA:

- ¿Has abordado cada amenaza del modelo de amenazas con la profundidad necesaria? Sí
- ¿Has usado lo más posible las tecnologías de seguridad de terceras partes? Sí
- ¿Has realizado un diseño global integrado de la solución dada para garantizar la seguridad? No
- ¿Has considerado los principios estándares de seguridad a la hora de diseñar la infraestructura de seguridad? Sí
- ¿Es tu infraestructura de seguridad lo más simple posible? Sí
- ¿Has definido una forma de identificar brechas de seguridad y de recuperar el sistema frente a ellas? No
- ¿Has aplicado los resultados de la perspectiva de seguridad a todos los puntos de vista afectados? No
- ¿Han revisado tu diseño de seguridad expertos externos en seguridad? No

Referencias

Nick Rozanski. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives, Second Edition*. Addison-Wesley Professional, 2011. URL <https://learning.oreilly.com/library/view/software-systems-architecture/9780132906135/>.

Nick Rozanski. Software systems architecture: Working with stakeholders using viewpoints and perspectives, second edition. Resumen/traducción para clase de Desarrollo de Software (Tema 2), prof. M. Mar Abad Grau. Technical report, 2020.