

Trabajo Práctico N° 3: **Casos de Uso.**

PARTE I: Definiciones Generales.

Ejercicio 1.

Definir qué es el desarrollo centrado en el usuario.

El desarrollo centrado en el usuario (DCU) es una metodología de diseño y desarrollo de sistemas que pone al usuario final como eje principal, garantizando que sus necesidades, características y contexto de uso guíen las decisiones de diseño, mediante la participación activa del usuario y la evaluación iterativa de las soluciones propuestas.

Características principales del DCU:

- Involucra a los usuarios de manera activa y continua en el proceso de desarrollo.
- Busca comprender el contexto de uso: quiénes son los usuarios, qué tareas realizan y en qué entorno.
- Promueve la iteración: los prototipos y diseños se prueban, se obtiene retroalimentación y se ajusta.
- El objetivo principal es mejorar la usabilidad, la eficiencia y la experiencia de uso del sistema.

Ejercicio 2.

Definir qué son los casos de uso y describir cómo se utilizan.

Un caso de uso (CU) es una técnica de modelado utilizada en el análisis y diseño de sistemas que describe, de forma estructurada, cómo un actor externo (usuario u otro sistema) interactúa con el sistema para lograr un objetivo específico. Se centra en las funcionalidades que el sistema debe ofrecer y en los resultados esperados desde el punto de vista del usuario.

Los casos de uso:

- Se emplean para capturar y documentar los requerimientos funcionales de un sistema.
- Permiten representar los escenarios de interacción entre usuarios (actores) y el sistema.
- Se usan tanto en la fase de análisis (para entender qué debe hacer el sistema) como en la fase de diseño (para definir cómo se implementarán esas interacciones).
- Facilitan la comunicación entre desarrolladores, analistas y usuarios, ya que son fáciles de comprender sin necesidad de conocimientos técnicos avanzados.
- Generalmente, se representan con diagramas de casos de uso (UML) y con descripciones textuales que detallan pasos, precondiciones, excepciones y resultados.

Ejercicio 3.

Definir qué es un actor y un escenario.

Un actor es una entidad externa al sistema (persona, organización o, incluso, otro sistema) que interactúa con él para cumplir un objetivo. Representa un rol que utiliza el sistema, no necesariamente a una persona en particular.

Un escenario es la descripción concreta de un caso de uso en una situación particular, detallando la secuencia de pasos que sigue un actor para lograr un objetivo dentro del sistema. Puede haber escenarios principales (flujo normal o básico) y alternativos (variaciones, errores o excepciones)

Ejercicio 4.

Definir las relaciones que pueden presentarse en el diagrama de casos de uso. Describir cuándo se utiliza cada una.

Las relaciones que pueden presentarse en el diagrama de casos de uso son:

- Asociación (Association): Es una línea que conecta a un actor con un caso de uso. Se utiliza cada vez que un actor participa o interactúa con un caso de uso.
- Inclusión (Include): Es una relación en la que un caso de uso siempre incluye la funcionalidad de otro. Se utiliza cuando una funcionalidad común se repite en varios casos de uso y conviene modelarla por separado.
- Extensión (Extend): Un caso de uso extiende a otro si agrega comportamiento opcional o condicional. Se utiliza cuando existe un flujo alternativo que sólo ocurre bajo ciertas condiciones.
- Generalización (Generalization): Es una relación jerárquica donde un caso de uso o actor puede heredar comportamiento de otro más general. Se utiliza cuando existen variaciones especializadas de un mismo rol o caso de uso.

Ejercicio 5.

Enumerar los beneficios de modelar requerimientos del sistema con casos de uso.

Los beneficios de modelar requerimientos del sistema con casos de uso son:

- Orientación al usuario: Describen el sistema desde la perspectiva de los usuarios, facilitando la comprensión de las funcionalidades reales que necesitan.
- Claridad y comunicación: Usan un lenguaje sencillo y comprensible tanto para desarrolladores como para usuarios, mejorando la comunicación entre los distintos actores del proyecto.
- Definición precisa de los requerimientos funcionales: Ayudan a identificar qué debe hacer el sistema (funcionalidades principales y alternativas) y qué no debe hacer.
- Identificación de actores y sus interacciones: Permiten reconocer los roles que interactúan con el sistema y cómo se relacionan con él.
- Base para el diseño y desarrollo: Sirven como punto de partida para el diseño de la arquitectura del sistema y la definición de interfaces de usuario.
- Soporte para pruebas y validación: Los escenarios de los casos de uso pueden transformarse en casos de prueba, asegurando que el sistema cumple con lo esperado.
- Reutilización y consistencia: Relacionando casos de uso comunes mediante *include* o *extend* se evita duplicación y se asegura coherencia en el modelo.
- Facilidad de documentación: Los diagramas UML y las descripciones textuales de casos de uso generan documentación clara y estructurada.

PARTE II: Ejercitación.**Ejercicio 1: Alquiler de Mobiliario.**

Suponga que trabaja en una consultora, la cual ha sido, recientemente, contactada por una empresa de alquiler de mobiliario para eventos para la realización de una app.

De las diferentes entrevistas, se ha obtenido la siguiente información:

El gerente nos dijo que resulta fundamental tener una aplicación móvil que nos permita manejar la agenda de la empresa, sabiendo qué disponibilidad tenemos y permitiendo que nuestros clientes alquilen a través de la app. Para esta primera versión de la app, el gerente nos pidió que sea posible dar de alta los diferentes mobiliarios, así como la posibilidad de que los usuarios puedan realizar una reserva de alquiler desde sus dispositivos. Para el detalle de cómo se realiza la carga de los muebles, el gerente nos sugirió hablar con el encargado del departamento de mobiliario. El encargado de mobiliario nos comentó que, de cada mueble, se debe cargar código de inventario, tipo de mueble, fecha de creación, fecha de último mantenimiento, estado (libre, de baja, alquilado) y el precio de alquiler. Además, no pueden existir códigos repetidos. Para que el encargado pueda dar de alta el mobiliario, debe autenticarse en el sistema. El registro de los usuarios de carga no debe modelarse.

El encargado del departamento de alquileres nos comentó acerca de las reservas de los alquileres. Por una política comercial de la marca, una reserva tiene que incluir, como mínimo, 3 muebles. La reserva debe tener una fecha, lugar del evento, cantidad de días y mobiliario junto a su cantidad. Para realizar una reserva, se debe abonar el 20% del total del alquiler. El pago de la reserva se realiza, únicamente, con tarjeta de crédito validando número de tarjeta y fondos a través de un servicio del banco. Luego de efectuado el pago, se emite un número de reserva único que será, luego, utilizado por el cliente para hacer efectivo el alquiler.

Ejercicio 2: Posgrado.

Suponga que trabaja en el área de sistemas de la Facultad de Informática y se le solicitó la automatización del pago de carreras de posgrado.

Inicialmente, se coordinó una reunión con el director del posgrado y se obtuvo la siguiente información: Ya que no se desea seguir cobrando el dinero en la secretaría, es necesario que los alumnos puedan pagar las carreras vía web. Como el director de posgrado no realiza tareas administrativas, nos recomendó hablar con el secretario académico. De la entrevista con el secretario académico, se obtuvo la siguiente información: Es necesario cargar las carreras a un sistema. En esta primera versión del sistema, sólo se nos pidió esta funcionalidad, sin la modificación ni eliminación. De cada carrera, se conoce: nombre de la carrera (no puede repetirse), duración en años (a partir de la consulta del estatuto de posgrado se obtuvo que como máximo son 5 años), costo y cantidad máxima de cuotas para el pago. La carga de las carreras no la realiza el secretario académico, sino un empleado administrativo.

Al preguntarle por la dinámica del sistema, el secretario académico nos derivó con el jefe del área administrativa, con el cual hicimos otra entrevista y pudimos obtener la siguiente información: El requerimiento fue que el alumno ingrese a la web de posgrado y pueda registrarse ingresando: nombre, apellido, nombre de usuario (único) y contraseña (más de 6 dígitos). Cualquier alumno previamente registrado, puede iniciar sesión con su nombre de usuario y contraseña, habilitándose la inscripción a alguna de las carreras. Para ejemplificar esta funcionalidad, nos otorgaron acceso al sistema SIGEF, el cual realiza funcionalidades similares para las carreras de grado. Para inscribirse, el alumno deberá seleccionar la carrera, ingresar la cantidad de cuotas a pagar, ingresar el número de tarjeta y, en caso de que la tarjeta sea válida y tenga fondos, se hará efectivo el cobro y la inscripción. La tarjeta de crédito se valida a través de un servicio del banco con el cual la universidad tiene convenio. Luego de efectuado el cobro, el sistema debe imprimir dos comprobantes, uno de inscripción y otro de pago. La única forma que tiene el alumno de pagar es con tarjeta de crédito.

Ejercicio 3: Contratos.

Suponga que trabaja en un grupo en el área de sistemas de una organización y está por comenzar un nuevo proyecto para desarrollar un sistema que depende del departamento contable.

El sistema deberá administrar los contratos realizados con terceros. En una de las reuniones con el jefe de departamento nos dijo que él no usará el sistema pero que recibirá listados del personal contratado, ya que deberá firmarlos para elevarlos a las autoridades.

Para obtener más información generamos una reunión con el empleado de mesa de entradas. Nos contó que el problema que tienen actualmente es que realizan todas las minutas a mano, por lo cual desean automatizar esta tarea. Las minutas son el paso previo a un contrato. Para confeccionar una minuta, el empleado de mesa de entradas debe ingresar nombre y número de CUIT de una persona a contratar, tipo de contrato, fecha de comienzo, duración y monto, a lo que el sistema le asociará un número de minuta automáticamente. Nos recomendó leer la reglamentación vigente acerca de contratos, de la que obtuvimos que los montos de los mismos no pueden superar los \$25.000 y que la duración debe ser como máximo de 6 meses. Una vez confeccionada la minuta por parte del empleado de mesa de entradas, la misma queda pendiente de aprobación.

El que puede aprobar una minuta es el empleado de rendiciones. Realizamos una reunión con él y nos contó que su tarea consiste en evaluar las minutas para determinar su aprobación. También nos dijo que en otro trabajo que tiene usan un sistema llamado MiMiNuTa, al que nos puede dar acceso para ver cómo hacen esa tarea. Después del análisis de este sistema, se concluyó que, para aprobar una minuta, se necesitaría ingresar un número de minuta y que el sistema muestre los datos de la misma para poder aprobarla. Nos dijo que no puede aprobar la minuta si la persona a contratar tiene 3 contratos vigentes (minutas aprobadas) ni tampoco si el CUIT de la persona a contratar está inhabilitado por la AFIP. Actualmente, se comunica telefónicamente con la AFIP para realizar esta verificación, pero sabe que ésta provee un servicio para aplicaciones que permite hacer la verificación en línea.

Esto último nos obligó a generar una reunión con el administrador de servidores de la AFIP. Nos dijo que, para poder conectarnos con un servidor de la AFIP, el sistema debe mandar un token (código que identificará de manera única a nuestra aplicación) y CUIT; si el token es correcto, el servidor responde si el CUIT está habilitado o no.

Por último, el empleado de rendiciones será el responsable de imprimir los listados con las minutas aprobadas, es decir, un listado con el personal contratado para poder dárselo al jefe de departamento para que lo firme.

Ejercicio 4: Máquina de Reciclados.

Se desea modelar un sistema que controle una máquina de reciclado.

La máquina cuenta con un recipiente externo donde se depositan los materiales a reciclar (vidrio, papel, plástico o aluminio). La persona coloca lo que desea reciclar en el recipiente y, luego, presiona la opción “reciclar”. El sistema detecta el tipo de material y registra el peso; por último, imprime un recibo con el monto total que se le debe pagar a la persona por lo reciclado. Si el sistema no detecta, correctamente, el tipo de material, aborta el proceso retornando el producto.

El sistema también es utilizado por un operador a través de una consola que se encuentra bajo llave. El operador puede solicitar un listado con los tipos de materiales reciclados en un período de fechas determinado, detallando, además, el total abonado por dicho material. También, el operador puede actualizar los montos a pagar por kilo de cada tipo de material que la máquina puede reciclar. Para esto, el sistema, primero, solicita el tipo de material a actualizar. Luego, se pide el nuevo monto. Si el operador acepta, se actualiza el material seleccionado.

Ejercicio 5: Sistema de Alarmas.

Una empresa de seguridad ha solicitado el desarrollo de un sistema de alarma inteligente para el hogar.

Este sistema se basa en una red de sensores que monitorean la actividad tanto dentro como alrededor de la vivienda. Los residentes de la casa interactúan con el sistema a través de un panel de control físico, desde el cual pueden armar y desarmar la alarma. Además, tienen la capacidad de activar una alarma silenciosa en situaciones de emergencia. Para hacerlo, deben ingresar un código especial; si el código es correcto, el sistema registra el evento y se comunica con un servidor central para notificar el incidente, permitiendo, así, que se tomen las medidas adecuadas.

El sistema cuenta con un perfil de administrador, que posee las mismas funciones que un usuario estándar, pero con privilegios adicionales. El administrador tiene la capacidad de actualizar la configuración del sistema de alarma y modificar los códigos de acceso. Cada 15 segundos, el sistema realiza una verificación de todos los sensores. Si uno de los sensores se activa, el sistema registra los datos del evento (fecha, hora, lugar de la vivienda, sensor) y activa la alarma. De inmediato, se inicia un proceso de comunicación con el servidor central para reportar el incidente. Para establecer contacto con el servidor central, el sistema marca un número telefónico. Una vez que la central acepta la conexión, solicita al sistema el identificador de la alarma y los datos del evento. A continuación, el servidor devuelve un tiempo estimado de llegada de un móvil que el sistema debe almacenar. Si la línea telefónica se encuentra ocupada o la central rechaza la conexión, el sistema debe seguir insistiendo hasta que la conexión se concrete.

Ejercicio 6: Préstamos Personales.

Se necesita modelar un subsistema web de administración de préstamos personales.

El sistema forma parte de un sistema mayor que implementa un servicio de homebanking para clientes del banco. Para todas las operaciones que realiza el cliente, se asume que está debidamente autenticado.

Un cliente del banco puede solicitar un préstamo personal vía web. Cada cliente puede solicitar un máximo de 3 préstamos y hasta un monto total de \$30.000. Para realizar la solicitud, el cliente ingresa el motivo, selecciona la cuenta de dónde se descontará, automáticamente, la cuota mensualmente e ingresa el monto del préstamo. El sistema debe verificar que el cliente no figure en el Veraz (banco de datos con información de deudores). Para realizar la comprobación, el sistema envía un código de seguridad para validar la identidad de la aplicación del banco. Una vez validada la identidad, la aplicación le envía el nombre y apellido de la persona y su número de CUIL/CUIT para determinar si existe o no como deudor. En caso de que todo sea válido, el sistema registra el préstamo correspondiente, genera un identificador del préstamo, un código de verificación y un comprobante con los datos del préstamo.

El cliente podrá adelantar cuotas a partir del sexto mes de otorgado el préstamo. Para esto, el sistema muestra un listado de préstamos vigentes donde se debe seleccionar el que se desea pagar. A continuación, el sistema solicita ingresar la cantidad de cuotas a abonar. Luego, el sistema muestra un listado de cuentas del cliente, donde se deberá seleccionar una de ellas para realizar el pago. Si hay saldo suficiente, el monto se deberá descontar de la cuenta del cliente. En caso de no poseer saldo, se le informa al cliente.

El cliente podrá realizar la cancelación total del préstamo a partir del noveno mes. Para ello, se debe presentar personalmente en el banco con su DNI para acreditar su identidad. Cuando se introduce el DNI, el sistema lista los préstamos para que se seleccione cuál desea pagar. Luego, se verifica que la cuenta asociada originalmente posea saldo suficiente para alcanzar el monto total adeudado. De ser así, se registrará dicha cancelación y emitirá un comprobante con los datos de la operación.

Ejercicio 7: Pago de Impuestos y Servicios.

Se desea modelar un sistema de pago electrónico de impuestos y servicios en efectivo.

Cuando un cliente llega para realizar un pago, el empleado o el gerente de la sucursal ingresa el código de pago electrónico y el sistema se conecta con la central de cobro para recuperar los datos de la factura (empresa, nro. de cliente, 1era fecha de vencimiento, 2da fecha de vencimiento, recargo y monto original). Una vez recuperados los datos, el sistema debe verificar los vencimientos para determinar el monto a cobrar. Teniendo esto en cuenta, cuando el 2do vencimiento está vencido, se debe informar que la factura no se puede cobrar por dicho motivo. Cuando el 1er vencimiento está vencido, hay que aplicar el recargo al monto original. Si la factura no está vencida, se cobra el monto original.

Una vez al día, el gerente de la sucursal debe registrar en la central de cobro los pagos que hicieron los clientes. Para esto, el sistema requiere la clave maestra y, de ser correcta, recupera las transacciones de los impuestos y servicios cobrados en el día, se conecta a la central de cobro y se las envía. Cuando la central confirma la recepción exitosa, el sistema las registra como enviadas. Este último paso es importante porque no deben enviarse dos veces las transacciones. Si el gerente intenta enviar una segunda vez, el sistema no debe permitirlo.

Finalmente, el gerente puede ver las estadísticas de los impuestos y servicios cobrados. Para esto, se ingresa la clave maestra, un rango de fechas sobre las cuales debe calcularse las estadísticas y el sistema debe mostrar los montos y la cantidad de cobros realizados, agrupando por empresa.

Tener en cuenta que, cada vez que el sistema debe conectarse a la central, debe enviarle un token (código que identifica al sistema). Una vez que la central valida el token, el sistema envía el requerimiento para recuperar los datos de la factura o el requerimiento para registrar los pagos del día según corresponda.

Ejercicio 8: Gimnasio.

Se desea modelar con casos de uso un sistema web para el manejo de turnos de un gimnasio.

Las personas que desean solicitar un turno para concurrir al gimnasio, primero, deben registrarse. Para ello, deben ingresar el DNI, el nombre, el apellido y el email que será utilizado como nombre de usuario (por lo tanto, no puede repetirse). Una vez que el usuario se registra, el sistema genera una contraseña que es enviada al correo de la persona.

Para solicitar un turno, el cliente del gimnasio, previa autenticación, debe ingresar: fecha, hora y actividad. Si hay cupo para esa actividad en ese día y a esa hora, el sistema registra el turno, de lo contrario, muestra un mensaje de cupo no disponible.

Por último, cuando un cliente llega al gimnasio, debe registrar su llegada; para esto, la secretaria del gimnasio solicita el número de DNI del cliente para hacer el registro. El gimnasio se maneja con un sistema externo que suma puntos a aquellos clientes que cumplen con las reservas realizadas. Es necesario que, cuando se registre la asistencia del cliente al gimnasio, se haga la suma correspondiente, para lo cual el sistema se conecta con el servidor externo, envía el DNI del cliente y la actividad que va a realizar y el servidor se encarga de sumar los puntos.

Ejercicio 9: Carga de Empleados.

Se desea modelar un sistema de carga de empleados de la Facultad de Informática.

Se ha realizado una reunión con el jefe de recursos humanos, el cual nos detalló lo siguiente: cuando un nuevo empleado es contratado, debe ir a la oficina de recursos humanos y presentar a la secretaria, quien realiza la carga, su DNI, apellido, nombre, edad y domicilio. Según el reglamento interno de la facultad, para poder llevar a cabo el alta, el sistema debe verificar que el nuevo agente no figure en la base de sumariados, de ser así se informa dicha situación y se cancela la carga. La base de sumariados es una base interna del sistema.

Además, según la ley 1231/6 de empleo, el sistema debe permitir asociar una obra social a un agente, para lo cual la secretaria deberá ingresar el DNI. Para poder efectuar la consulta, el sistema debe conectarse con el servidor y enviar el DNI del empleado. En caso de morosidad, el sistema deberá informar dicha situación permitiendo al empleado regularizar su situación en los próximos 30 días e imprimiendo un código de inicio de trámite. Si no es moroso, el sistema le asocia la obra social e imprime un carnet. También se ha tenido una reunión con el jefe del área de comunicación visual, quien nos ha dado acceso al manual de estilo de la organización para que lo consultemos y respetemos cuestiones estéticas.

Ejercicio 10: Alquiler de Bicicletas.

Modelar un sistema municipal de alquiler de bicicletas. El alquiler y devolución están automatizados con un sistema que cuenta con un módulo que retiene y libera las bicicletas, un lector de tarjetas y un display para mostrar información al usuario.

Los usuarios deben estar registrados y contar con una tarjeta para los pagos. Para registrarse, un usuario concurre a una dependencia municipal donde un empleado o el responsable del área lo da de alta, pidiéndole los datos personales y el número de tarjeta.

Cuando un usuario quiere alquilar una bicicleta, apoya su tarjeta en un lector y, si la misma está registrada y no tiene otra bicicleta alquilada, el sistema libera la bicicleta y espera 45 segundos para que la retire. Pasado ese tiempo y no retirada la bicicleta, se retiene la bicicleta y se cancela la operación. Si el usuario retiró la bicicleta, se registra identificador de la bicicleta y la fecha de retiro para poder realizar el cobro cuando la devuelva.

Para devolver la bicicleta, el usuario apoya la tarjeta en el lector y, si la misma pertenece a un usuario registrado, libera el receptor durante 45 segundos. Si el usuario no pone la bicicleta, se cancela la operación. Si se detecta la bicicleta, se procede al cobro del alquiler. El sistema calcula el monto y se conecta a un servidor de pago pasando el monto, el nombre del usuario y el número de tarjeta. De haber algún problema con el alquiler o devolución de la bicicleta, debe informarse en el display.

El responsable del área puede actualizar el precio del alquiler ingresando el monto de la hora. También puede hacer un seguimiento de una bicicleta introduciendo un rango de fechas y el identificador de la bicicleta.