

por Javier Carrasco

Entornos de desarrollo

Bloque 1

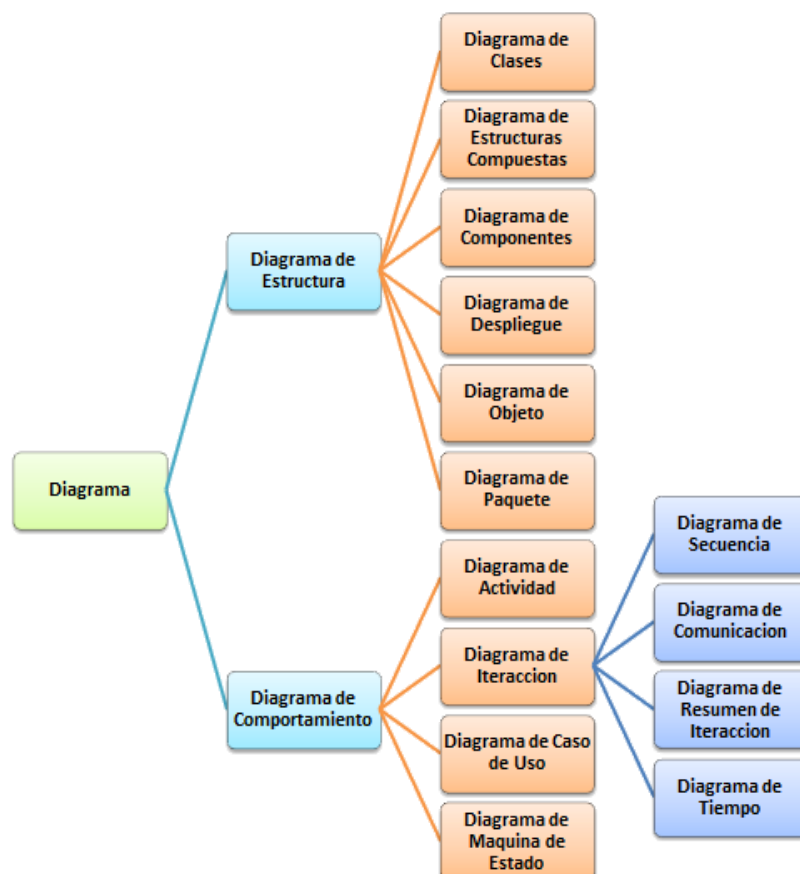
Tema 5: Análisis de una aplicación. Diagramas de casos de uso

1.5.1. Lenguaje Unificado de Modelado (UML *Unified Modeling Language*)

UML es un lenguaje de modelado estándar que permite a los desarrolladores especificar, visualizar, construir y documentar aplicaciones software. Además UML permite hacer estas aplicaciones escalables, seguras y robustas en su ejecución.

Se desarrolló este lenguaje principalmente para dar cabida al análisis y diseño de aplicaciones orientadas a objetos de una forma unificada ya que junto con la generalización del paradigma orientado a objetos, habían surgido diversos lenguajes de modelado. Para unificar todo este trabajo en un solo lenguaje, en 1997 se presentó UML 1.1. Y desde junio del 2015 existe el UML 2.5.

Este lenguaje de modelado tiene 14 tipos divididos en dos subtipos:

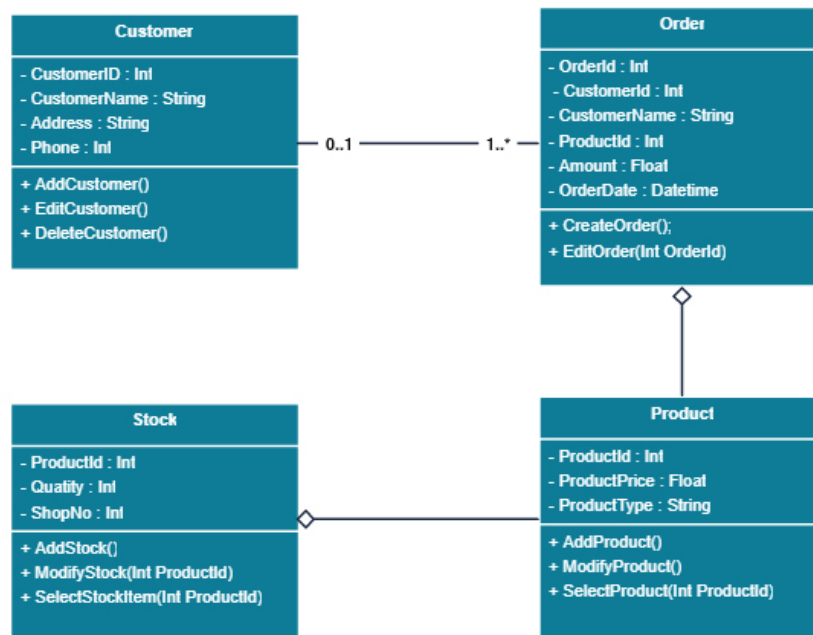


1.5.1.1. Diagramas Estructurales

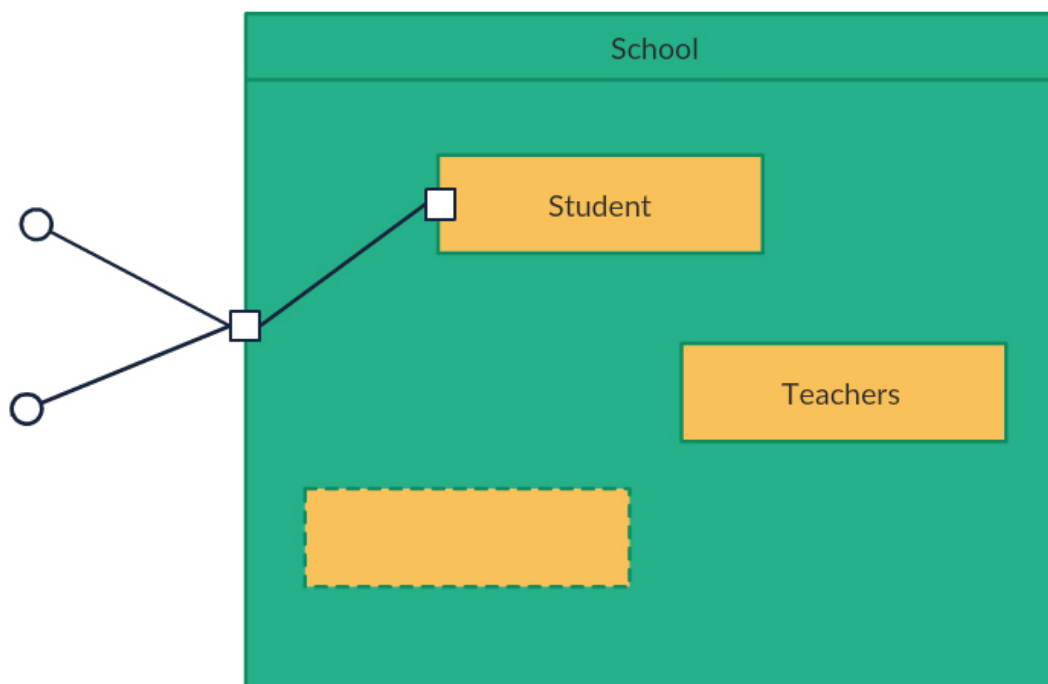
Muestran la estructura estática de un sistema.

- **Diagrama de clases:** es el diagrama de construcción principal para cualquier aplicación orientada a objetos. Muestras todas las clases del sistema con sus atributos y métodos y sus relaciones entre ellas.

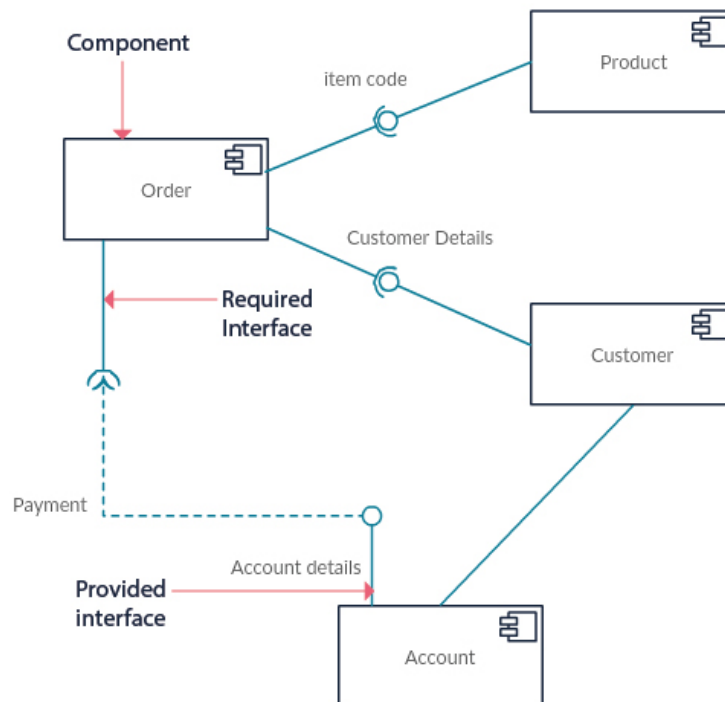
Class Diagram for Order Processing System



- **Diagrama de estructuras compuestas:** Muestran la estructura interna de una clase.

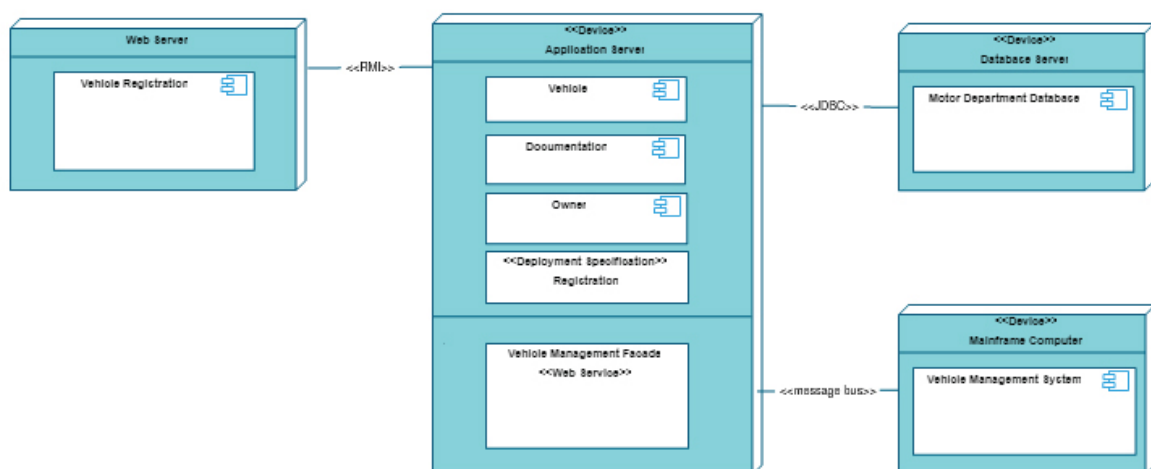


- **Diagrama de componentes:** Muestra la relación estructural entre los componentes del sistema. Se usa principalmente para sistemas complejos con muchos componentes.

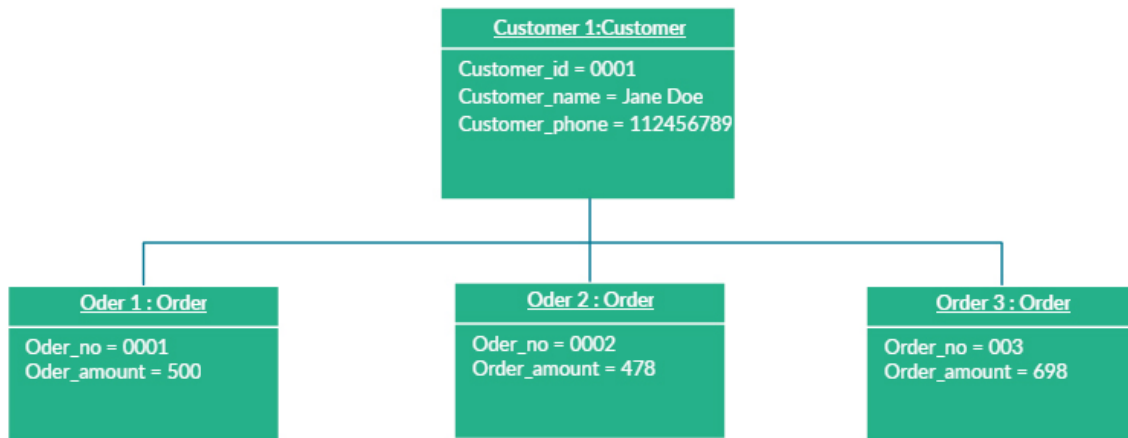


- **Diagrama de despliegue:** Muestra el hardware y el software de un sistema y como está distribuido ese software en el distinto hardware del sistema. Se suele utilizar en sistemas donde el software está distribuido entre distintos componentes hardware.

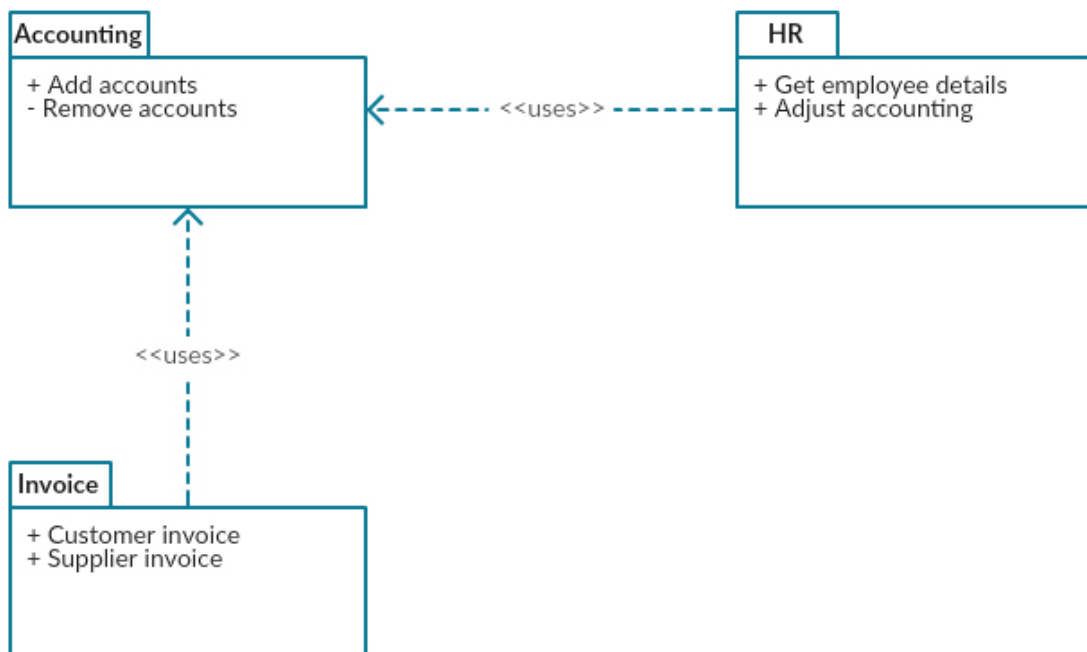
Deployment Diagram For a Vehicle Registration System



- **Diagrama de objetos:** Similar al diagrama de clases pero con objetos con datos de ejemplo.



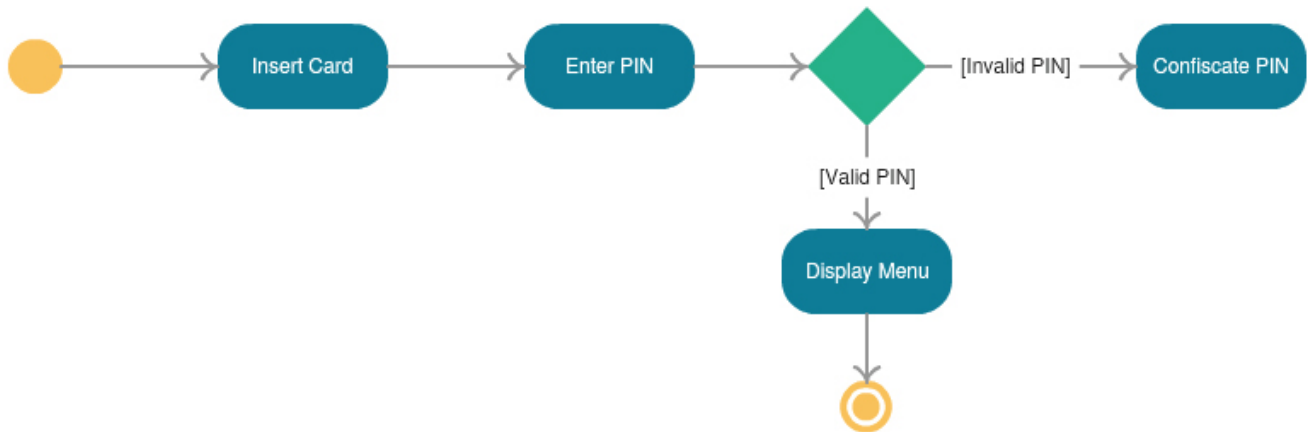
- **Diagrama de paquetes:** Muestras las dependencias entre los distintos paquetes del sistema.



1.5.1.2. Diagramas de Comportamiento

Muestran el comportamiento dinámico de los objetos del sistema.

- **Diagrama de actividad:** Muestra el flujo de trabajo en un modo gráfico. Puede usarse para describir el flujo de trabajo en un negocio o el flujo operacional de los componentes de un sistema.



- **Diagrama de interacción:** Incluyen distintos tipos de diagramas:

Diagrama de Secuencia: Muestra como los objetos interactúan entre ellos y el orden de esas interacciones.

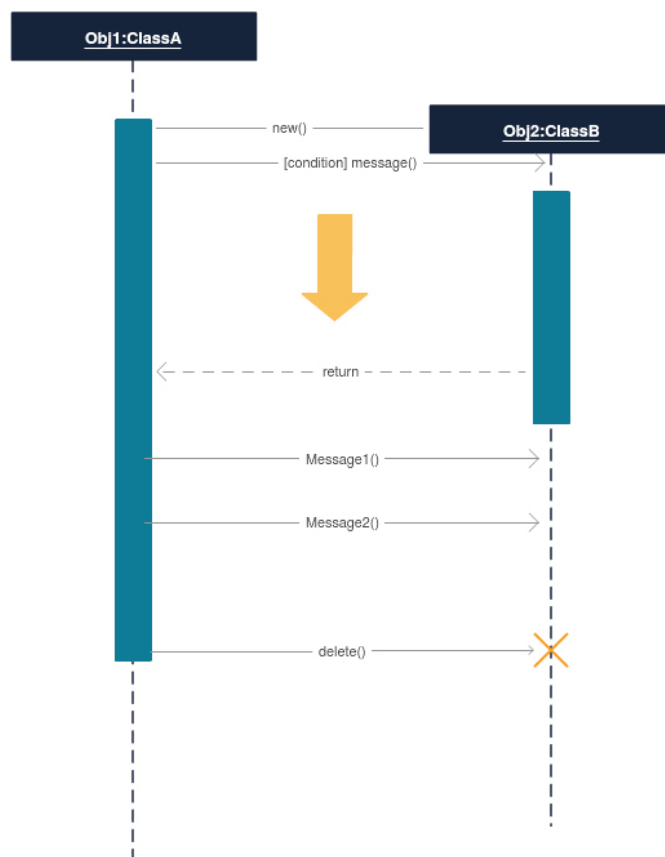


Diagrama de comunicación: En UML 1 se llamaba diagrama de colaboración. Es parecido al diagrama de secuencia pero se centra en los mensajes pasados entre objetos.

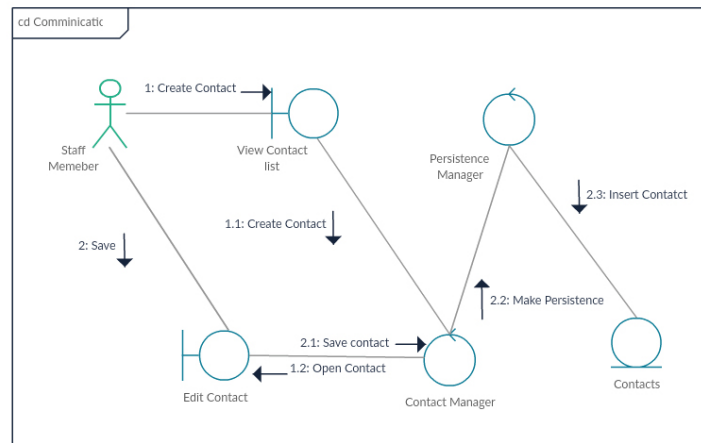


Diagrama de resumen de interacción: Es similar al diagrama de actividad, en este último se mostraba la secuencia de los procesos y en el resumen de interacción se muestra la secuencia de los diagramas de interacción. Dado que hay varios tipos de diagramas de interacción, cualquiera de ellos puede aparecer en este resumen.

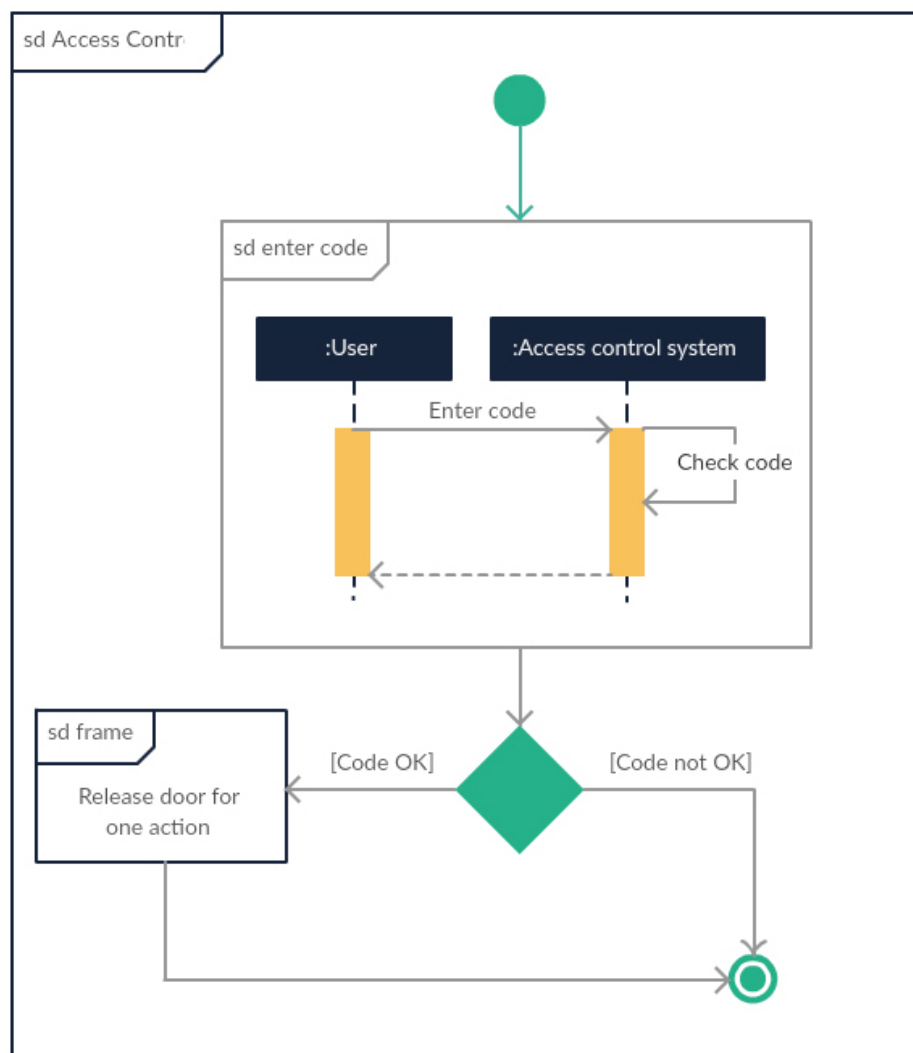
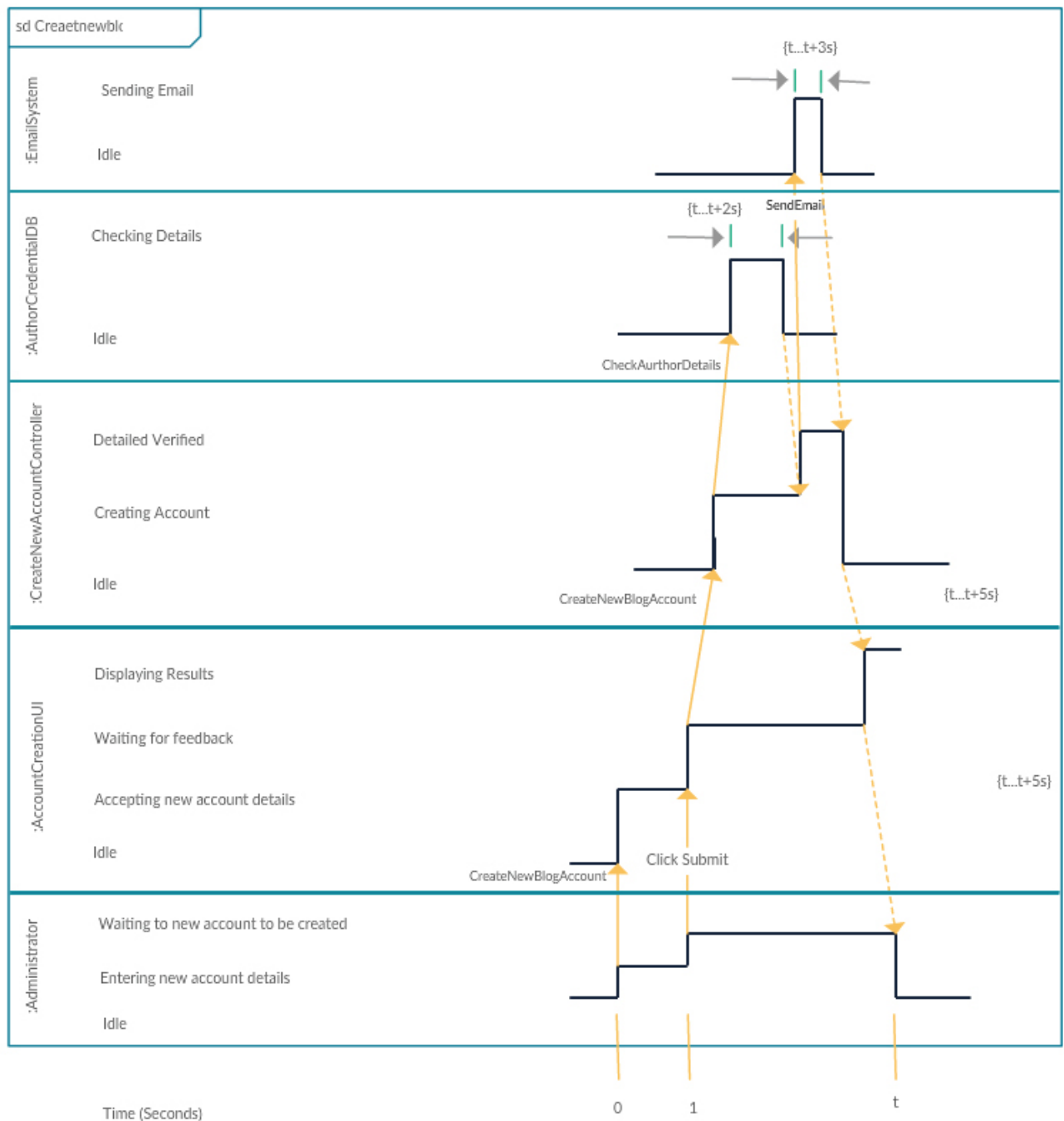
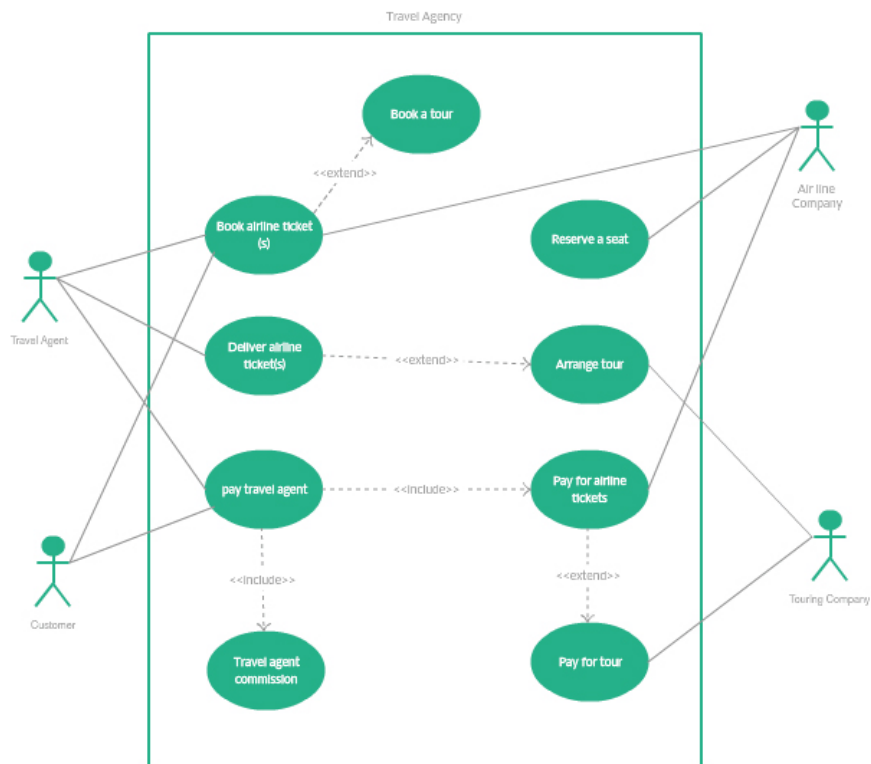


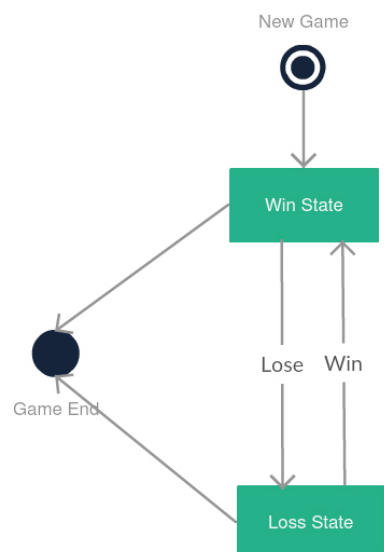
Diagrama de tiempo: Es similar al diagrama de secuencia. Muestra el comportamiento de los objetos en un momento determinado. Si solo hay un objeto el diagrama es directo, pero si hay más de un objeto representará las interacciones entre los objetos en ese marco de tiempo.



- **Diagrama de casos de uso:** Es el diagrama más conocido de los diagramas de comportamiento. Proporciona una visión general de los actores involucrados en un sistema, las funciones necesarias para estos actores y como estas funciones interactúan.



- **Diagrama de máquina de estado:** Es similar al diagrama de actividad aunque la notación y el uso cambian un poco. Es muy útil para describir el comportamiento de los objetos que actúan de forma diferente en función del estado en el que se encuentran.



1.5.2. Diagrama de casos de uso

Los diagramas de casos de uso los encontramos enmarcados dentro de los **diagramas de comportamiento** como puede verse en la clasificación de diagramas UML.

El objetivo o propósito de un diagrama de casos de uso es tratar de representar las posibles interacciones entre usuario y sistema, así como entre diferentes partes del propio sistema o incluso relaciones con otros sistemas, obteniendo así un punto de vista general de éste. En definitiva, permite definir los requisitos funcionales del sistema desde un punto de vista del usuario, especialmente útil para aquellos sistemas con un alto grado de interacción usuario/sistema, y guía el proceso de desarrollo.

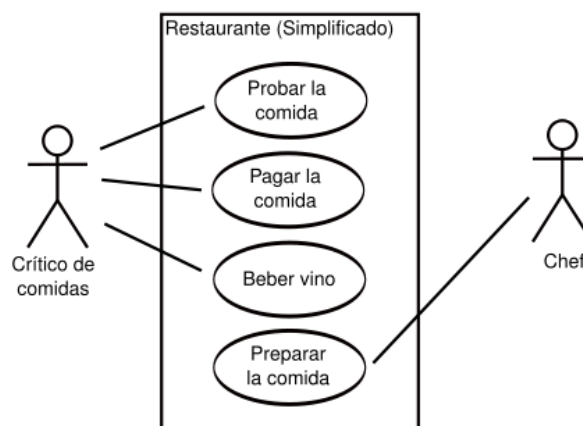
Los diagramas de casos de uso utilizarán actores y casos de uso para representar el uso de los servicios del sistema y sus reacciones. Es posible utilizar varios actores que representen diferentes roles dentro del sistema, lo que permite también tener una visión general de los distintos roles dentro del sistema y sus permisos de uso.

1.5.2.1. Elementos de un diagrama de casos de uso

Este tipo de diagrama es uno de los más sencillo de representar, tanto por los elementos que intervienen, como por la lógica que se debe aplicar en su desarrollo.

Sistema

Es el rectángulo que define los límites del sistema. Dentro de este se encuentran los casos de uso, y los actores deberán representarse fuera de los límites del mismo.



Caso de uso

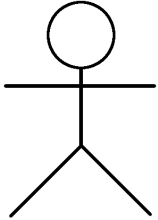


Los casos de uso representan lo que un actor quiere que el sistema haga. Éstos ejecutarán una serie de acciones con un resultado. Deben ser siempre iniciados por un actor y desarrollados, o pensados, desde el punto de vista del mismo.

Cada caso de uso es una operación completa desarrollada por el sistema. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema.

Se representan mediante un óvalo, o elipse, y están nombrados mediante un verbo que identifique la acción, por ejemplo, "Imprimir", "Pedir cuenta", "Pagar la comida", etc. También se pueden encontrar diagramas de casos de uso identificado los casos de uso como CU1, CU2, CU1.2, etc.

Actor



Los actores debemos entenderlos como algo o alguien ajeno al sistema y que interactúa con el sistema. Además, son los encargados de iniciar las acciones sobre los casos de uso del sistema.

Los actores además pueden definir, o identificar, los roles de los usuarios u otros sistemas que actúan con nuestro sistema.

Relaciones

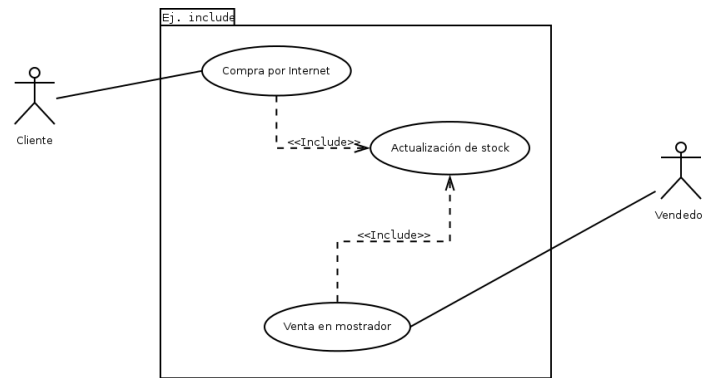
Las relaciones entre actores y casos de uso las representaremos mediante una línea continua, e indican el inicio de una acción.

Las relaciones entre casos de uso pueden ser de cuatro tipos, comunicación, incluye, extiende y generaliza. Fijate en la siguiente tabla.

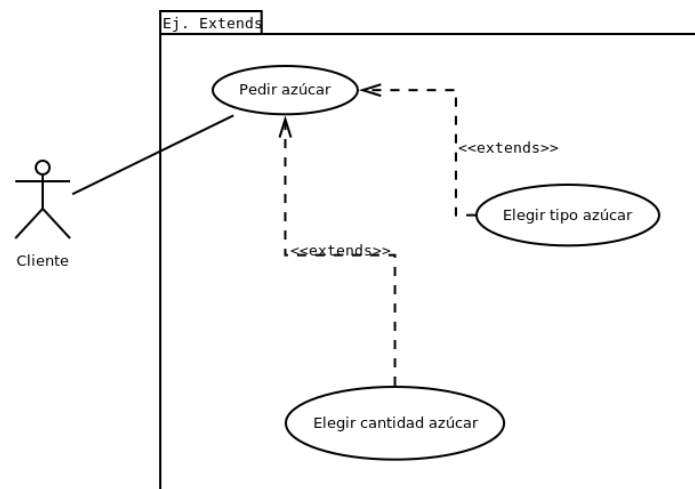
Relación	Representación	Significado
Comunica	_____	Comunica a los actores con los casos de uso, una línea simple sin puntas de flecha.
Incluye	— — —> <<include>>	Esta relación indica que un caso de uso base incluye el comportamiento de otro. Esto se utiliza para casos de uso comunes. La flecha apuntará hacia el caso de uso común.
Extiende	<<extend>> — — —<	Esta relación permite añadir funciones de otros casos de uso en puntos concretos, llamado puntos de extensión. La flecha apunta hacia el caso de uso base.
Generaliza	—————>	Se utiliza para indicar que un caso de uso (sub-caso) hereda de un caso de uso (super-caso) más general. Este tipo de relación puede llevar a confusión y no se recomienda mucho su uso.

Veamos a continuación una serie de ejemplos para reforzar los conceptos de los distintos tipos de relaciones.

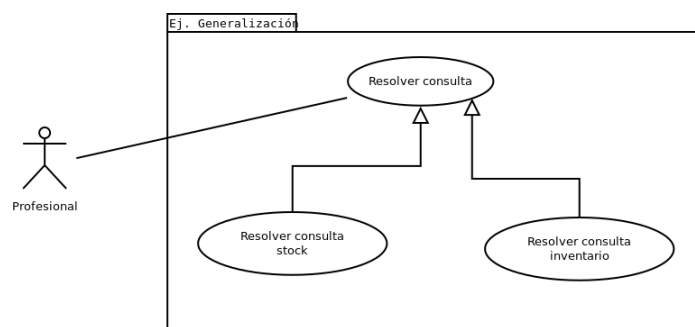
El siguiente ejemplo muestra una **relación de inclusión**, en este caso tenemos un sistema en el que un cliente puede realizar una compra a través de Internet, y un vendedor puede realizar una venta desde el mostrador. Como se puede ver, ambos casos conllevan la actualización de stock, que es la misma operación desde un punto u otro.



Veamos ahora la **relación extiende**, en este caso vemos como un caso de uso necesita de la especialización de otro, pero no tiene por qué ser obligatorio.



En este último ejemplo podemos ver una **generalización**, vemos un caso de uso del cual heredan dos sub-casos.



1.5.2.2. Ejemplo clásico

Veamos ahora como obtener un diagrama de casos de uso a partir de los requerimientos del usuario. Debemos tener en cuenta, que para conseguir la información debemos mantener una estrecha relación de comunicación Cliente/Desarrollador.

La máquina de café

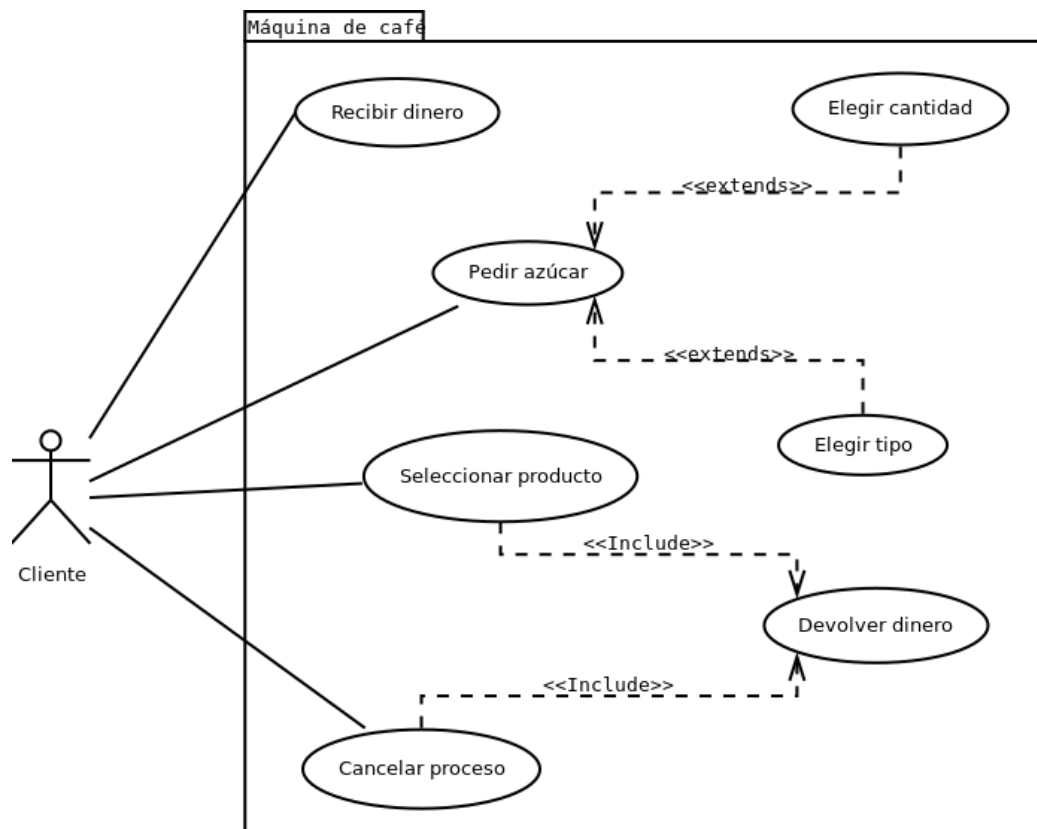
Se nos pide desarrollar el software para el funcionamiento de una máquina expendedora de café. Tras hablar con el cliente, llegamos a la siguiente especificación de requisitos.

Para que un cliente pueda obtener un café de dicha máquina, en primer lugar deberá introducir el dinero suficiente para iniciar el proceso. El siguiente paso será seleccionar el azúcar, este nuevo modelo, además de elegir la cantidad de azúcar, nos permite seleccionar el tipo (blanca, morena, etc.), pero por defecto siempre será blanca.

Una vez establecido el nivel y tipo de azúcar, el cliente seleccionará el producto (café solo, cortado, con leche, etc.).

La máquina deberá devolver el cambio una vez finalizado y entregado el producto, pero en caso que el usuario cancele la operación, antes de seleccionar producto, también se devolverá el importe en cuestión.

Como puede verse, el enunciado contiene bastantes detalles, pero el diagrama debe ser desarrollado desde el punto de vista del usuario, es decir, lo que se espera, en este caso, de la máquina.



1.5.3. Software específico

Si hacemos una búsqueda por Internet sobre *software* para dibujar diagramas de casos de uso, o en su defecto UML, encontraremos una gran variedad, la mayoría de ellos propietarios. Pero, veremos que destacan dos entre ellos, uno es **Visio**, comprado por Microsoft en el año 2000, es un programa de dibujo vectorial que permite realizar diferentes tipos de diagramas, entre ellos, UML.



El otro que encontraremos es **Dia**, el sustituto natural de Visio, creado inicialmente dentro del proyecto **GNOME**, es un *software* completamente modular y bajo licencia GNU. Y, aunque su última versión estable es de 2011, es el que utilizaremos para nuestros diagramas.

Como ya se ha mencionado, es totalmente gratuito, y podemos descargarnos la última versión desde <http://dia-installer.de/>, bastará con seleccionar la versión de nuestro sistema operativo. En Linux se puede incluso instalar desde los repositorios oficiales de las distribuciones, por ejemplo, desde Ubuntu podemos ejecutar el siguiente comando para su instalación y ya lo tendremos.

```
javier@desktop:~$ sudo apt-get install dia
```

No vamos a detallar la instalación ya que este software es un editor vectorial y no necesita una configuración especial durante el proceso de instalación.

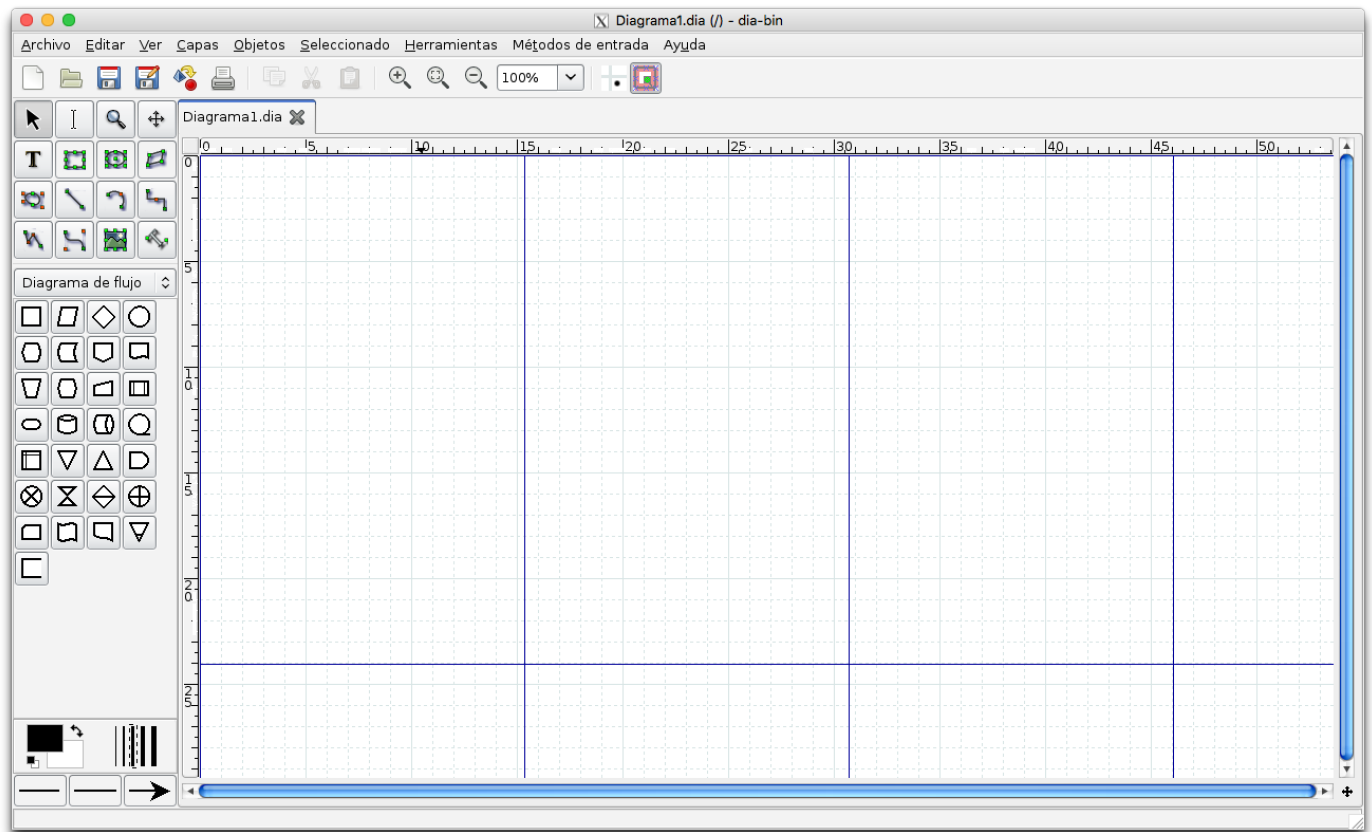
Tras la instalación, debo puntualizar una cosa en la versión para MAC. Se necesita XQuartz para que funcione, que podéis descargar desde <https://www.xquartz.org/>, pero puede producirse un fallo al intentar lanzar Dia, y es que no aparezca.

Para solucionar este problema en la instalación de MAC, una vez instalado Dia, desde la **carpeta Aplicaciones** buscamos Dia y con el botón derecho sobre el icono le decimos **Mostrar contenido del paquete**. Seguimos la siguiente ruta **Contents > Resources > bin** y editamos el archivo **dia** que encontraremos ahí. En la línea 39 añadimos `export DISPLAY=:0` para que quede como se puede ver a continuación.

```
1. startx=`which startx`
2. if [[ "" == $startx ]]; then
3.   if [[ ! -e /opt/X11/bin/startx ]]; then
4.     osascript -e 'tell app "System Events" to display dialog "X11
      (XQuartz) is not installed or not running. Would you like to
      visit xquartz.macosforge.org now in order to download and
      install XQuartz?" with icon 0'
5.   if [[ $? -eq 0 ]]; then
6.     open http://xquartz.macosforge.org
7.     exit
8.   fi
9. fi
10. fi
11. export DISPLAY=:0
12. osascript -e 'tell app "XQuartz" to launch'
13. for i in `seq 1 30`; do
```

Añadiendo esta línea se corrige el fallo, si hecho esto sigue sin aparecer, deberéis reiniciar, pero no debería ocurrir.

El aspecto de Dia es el que podéis ver en la siguiente captura de pantalla. Como podréis observar, el diseño de la aplicación es sencillo y fácil de utilizar.



Cuando guardemos diagramas con Dia estos se guardarán en ficheros con extensión **.dia**. En ocasiones nos será útil exportar los diagramas, bastará con utilizar la opción **Archivo > Exportar** y en la sección **Exportar opciones** elegir el tipo de archivo por extensión al que queremos exportar (PNG, JPG, GIF, TIFF, etc.).



Ahora nos centraremos en los aspectos de Dia que nos interesan para crear diagramas de casos de uso. En primer lugar, los diagramas de casos de uso forman parte de UML, debemos seleccionar la **opción UML** del desplegable que encontramos entre las herramientas y los elementos de diagramas.

Además, nos interesan las siguientes herramientas para empezar a realizar nuestros primeros diagramas de casos de uso.



La herramienta **Paquete grande** nos permite definir los límites del sistema.



La herramienta **Actor**, define los actores.



La herramienta **Casos de uso**, la cual nos permitirá crear los casos de uso que necesitemos en nuestro diagrama.



La herramienta **Dependencia** nos permite crear las relaciones de inclusión y extensión, para ello, una vez creada la flecha, haciendo

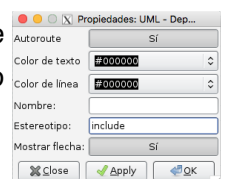
doble clic sobre ella indicaremos si es una u otra en la opción **Estereotipo**.



La herramienta **Generalización** permitirá crear generalizaciones en el diagrama.



La relación entre actor y caso de uso puede crearse utilizando la herramienta **Línea (L)**.



Ejercicios propuestos

1.5.3.1. Diseña el diagrama de casos de uso que se adapte a la siguiente especificación de requisitos. Utiliza **Dia** para su elaboración.

Un cliente nos pide desarrollar el funcionamiento de una máquina expendedora de snacks según se detalla a continuación.

La máquina expendedora tiene los artículos codificados mediante una numeración de dos cifras, siendo una distribución de cinco baldas con ocho elementos cada una de ellas. La primera balda, la de arriba del todo, los artículos serán el 11, 12, 13... 18. La segunda balda 21, 22, 23... 28 y así hasta la quinta balda.

Debemos tener en cuenta que la máquina no aceptará el dinero hasta que el cliente introduzca el código del producto que desea. El cliente puede cancelar la operación mientras no pulse el botón de confirmación, recuperando así el importe introducido.

Tras pulsar el botón de confirmación, la máquina moverá la espiral del artículo seleccionado para que este caiga y pueda ser recogido por el cliente, momento en el que se devolverá el cambio de la operación si fuese necesario.

1.5.3.2. Diseña el diagrama de casos de uso que se adapte a la siguiente especificación de requisitos. Utiliza **Dia** para su elaboración.

Se desea diseñar el funcionamiento de una App para hacer Sudokus. Para iniciar una partida, el jugador deberá seleccionar la opción "Start Game", a continuación seleccionará el nivel de dificultad. Una vez seleccionada, empezará el juego.

Ya en la partida, el jugador deberá seleccionar una casilla libre para indicar donde quiere colocar un número. Seguidamente, seleccionará un número del 1 al 9, y así hasta completar el Sudoku.

Durante la partida, el jugador dispone del botón "Check" para comprobar si va por buen camino, la App mostrará el mensaje "Todo bien, te quedan X celdas" o "Error en la celdas iluminadas". En este último caso la App iluminará las celdas erróneas.

Una vez completado el Sudoku correctamente, aparecerá el mensaje "Bien hecho!" y se mostrarán las estadísticas de tiempo de resolución.

1.5.3.3. Diseña el diagrama de casos de uso que se adapte a la siguiente especificación de requisitos. Utiliza **Dia** para su elaboración.

Se desea diseñar un módulo de consulta de facturas. Dicho módulo podrá ser accedido tanto por los clientes, como por los comerciales. Dicho acceso deberá producirse bajo la validación de credenciales (usuario y contraseña básicamente).

Cuando un cliente accede podrá realizar consultas sobre sus propias facturas estableciendo rangos de fechas (cliente+fechas) o por importes (cliente+importe).

Cuando un comercial accede puede realizar consultas de facturas puede filtrar por cliente (sólo cliente), por cliente y por rango de fechas (cliente+fechas) o por importes (cliente+importe).

Una vez terminada la resolución de la consulta, el módulo mostrará en pantalla el resultado. 1.5.3.4. Realiza el diagrama de casos de uso del ejercicio 1.4.3.2 del tema anterior sobre el centro cultural, utilizando Dia.

1.5.4. Para profundizar

Si quieres saber más sobre UML y diagramas de casos de uso puedes consultar las siguientes referencias:

- [UML](#)
- [UML 2.5.1.](#)
- [Creating UML Use Case Diagrams](#)
- [Dia Diagram Editor](#)