

UD8 + UD9

OTROS DIAGRAMAS UML

PRÁCTICA EVALUABLE
D.CLASES Y D.ACTIVIDAD CON UML



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

PRÁCTICA: DIAGRAMA DE CLASES Y ACTIVIDAD CON UML	UD8 + UD9 OTROS DIAGRAMAS UML
Contenidos: <ol style="list-style-type: none"> 1. Justificación de la práctica 2. Instrucciones de la práctica 3. Captura de ejemplo 4. Rúbrica de evaluación 5. Ejercicios: Diagramas de clases: <p>MODELO A Portal de películas y series</p> <p>MODELO B Gestión de una pizzería</p> Diagramas de actividad: <p>MODELO C Montaje de equipos</p> <p>MODELO D Volar en pandemia</p>	
PRÁCTICA EVALUABLE	

1. JUSTIFICACIÓN DE LA PRÁCTICA

Para comprobar que has adquirido los conocimientos y destrezas necesarios para generar diagramas UML propios desde cero, complementaremos la práctica evaluable de Diagramas de Casos de Uso con esta segunda práctica de UML en la que te pediremos que diseñes y entregues:

1. Un tipo de Diagrama de Estructura (estático):
 - Concretamente un **DIAGRAMA DE CLASES (DCL)** como los vistos en la Unidad 8
2. Un tipo de Diagrama de Comportamiento (dinámico):
 - Concretamente un **DIAGRAMA DE ACTIVIDAD (DACT)** como los de la Unidad 9

Por tanto, si quieres obtener la máxima nota en esta práctica, deberás entregar **DOS DIAGRAMAS** en un único documento PDF, dependiendo de estos parámetros personales:

- **Diagrama de Clases (DCL)**
 - Si tu DNI/NIE acaba en número PAR O CERO deberás entregar el **MODELO A**
 - Si tu DNI/NIE acaba en número IMPAR deberás entregar el **MODELO B**
- **Diagrama de Actividad (DACT)**
 - Primer apellido de la A-M deberás entregar el **MODELO C**
 - Primer apellido de la N-Z deberás entregar el **MODELO D**

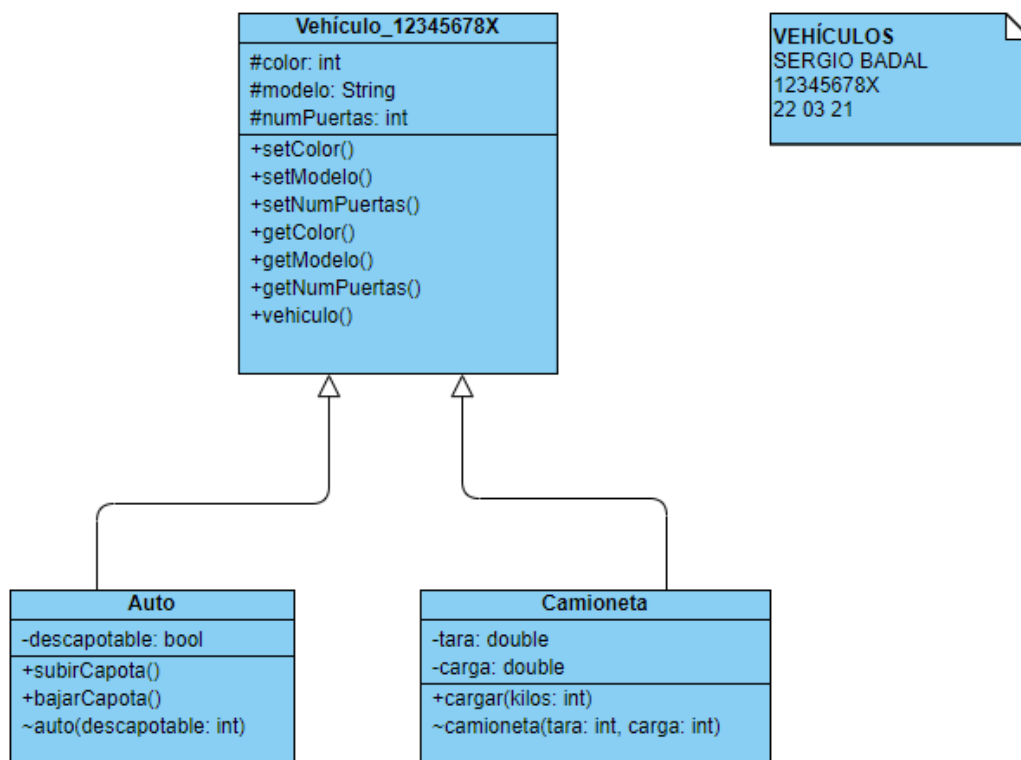
2. INSTRUCCIONES DE LA PRÁCTICA

Deberás realizar los diagramas siguiendo las siguientes indicaciones:

1. Usa la notación UML que hemos visto en las unidades 8 y 9
2. Puedes usar estas dos aplicaciones que hemos visto en clase, optando a la máxima nota solo si usas una distinta para cada uno de los dos diagramas:
 - (a) La aplicación **VISUAL PARADIGM** (online, escritorio o incrustada en tu IDE)
 - (b) La aplicación web **DRAW.IO**
3. Incluye una nota identificativa en cada diagrama con estos datos:
 - Título del diagrama, nombre completo, DNI/NIE y la fecha de creación.
4. Incluye en UNA CLASE (en el DCL) y UNA ACTIVIDAD (en el DACT) tu DNI/NIE al final como hiciste con la práctica evaluable anterior.
 - Ejemplo: CLIENTE_DNICONLETRA, ALUMNO_DNICONLETRA ...
5. Si detectas alguna ambigüedad en el enunciado, incluye todas las notas adicionales que consideres en cada diagrama indicando cómo la resuelves.
6. Entrega los dos diagramas en un único PDF de dos caras llamado **OTROSUML_DNICONLETRA.pdf**
7. Puedes usar herramientas gratis como ilovepdf.com para unir dos PDF.

3. CAPTURA DE EJEMPLO

A continuación te mostramos cómo podría ser **una de las dos caras del PDF**:



4. RÚBRICA DE EVALUACIÓN

[si quieres la máxima nota revisa estos puntos con cuidado antes de la entrega]

DIAGRAMA DE CLASES (DCL)	
CLASES	Hasta 1 punto
ATRIBUTOS/MÉTODOS	Hasta 1 punto
RELACIONES	Hasta 1 punto
CARDINALIDADES	Hasta 1 punto
DIAGRAMA DE ACTIVIDAD (DACT)	
ACTIVIDADES	Hasta 1 punto
NODOS DE CONEXIÓN	Hasta 1 punto
CARRILES	Hasta 1 punto
GUARDAS	Hasta 1 punto
USO DE DOS APLICACIONES	
UN DIAGRAMA ESTÁ HECHO CON DRAW.IO Y EL OTRO CON VPARADIGM	2 puntos
TOTAL	10 puntos

ACLARACIONES IMPORTANTES:

- **No se aceptarán:**
 - Entregas “a mano” o con otros editores no autorizados previamente.
 - Entregas sin los puntos 3 y 4 (nota identificativa y DNI en los diagramas).
 - Entregas que no correspondan con los MODELOS asignados.
- Si la entrega no es en un único PDF de dos caras se restarán **dos puntos** de la nota final.
- **Cualquier entrega que suponga dudas de su originalidad requerirá una entrevista online.**

5. EJERCICIOS

MODELO A: PORTAL DE PELÍCULAS Y SERIES (DAWFLIX)	ENUNCIADO
<p>Se requiere desarrollar un DIAGRAMA DE CLASES que refleje ciertos aspectos de la gestión de DAWFLIX, un portal de películas y series que se quiere lanzar este verano.</p> <p>Los usuarios del portal se clasifican en libres y abonados, almacenando para todos ellos los apellidos, nombre, documento (NIF/NIE), fecha de alta, y usuario y clave de acceso al portal. También se requiere indicar el método “cambioDatos”, al que le pasaremos nombre, apellidos y documento, el método “cambioClave”, al que le pasaremos la nueva clave, y el método “bajaTotal”.</p> <p>El sistema almacena películas y series de esta manera:</p> <ul style="list-style-type: none">• Las series compuestas por, como mínimo, una temporada.• De las películas y de las series queremos almacenar su código (“codSerie” y “codPeli”) y su título (“titSerie” y “titPeli”).• De las temporadas nos basta con el número de temporada y una breve descripción. <p>Los usuarios considerados como libres podrán realizar pedidos que estarán formados por varias películas y/o varias temporadas, almacenando la fecha y el código de cada pedido (“codPedido”). De este subtipo de usuarios solo queremos almacenar su antigüedad (en meses) para poder ofrecerles descuentos más adelante según el tiempo que llevan con nosotros y un método que nos permita tramitar que no quiere recibir publicidad (“noQuierePubli”).</p> <p>Los abonados funcionan de otra manera. Cada mes se genera una suscripción mensual de la que queremos almacenar el mes, año y un campo llamado “renovarAuto” que nos diga si se renueva automáticamente con los mismos títulos o no (sí/no). Cada suscripción está formada por una lista de películas y un conjunto de temporadas que el abonado selecciona, pudiendo marcar un máximo de 6 películas y 3 temporadas por suscripción. En este caso, como ocurre con los usuarios libres, no hay un número mínimo de temporadas ni de películas.</p> <p>De los abonados nos interesan: el número de abonado (“codAbonado”), el día de pago (el día del mes en que se le carga en cuenta), y las IPs desde dónde se puede acceder a la cuenta, aunque de momento solo podemos poner una IP en el campo “ipDeAcceso”. También queremos dos métodos: “congelarCuenta” y “bajaComoAbonado”.</p> <p>De cada pedido de los usuarios libres y de cada suscripción mensual, se necesita generar un único pago, que en ambos casos podrá ser de tipo PayPal (almacenando la cuenta de PayPal) o de tipo tarjeta (almacenando el número de tarjeta).</p> <p>Los métodos ser públicos y, si no se indica, no tienen parámetros ni de entrada ni de salida.</p> <p><i>Pista: Una posible solución tiene 11 clases.</i></p>	

Indicaciones:

- *No es necesario que incluyas los setters, getters, constructores y destructores, ni que incluyas tipos de datos en los atributos, ya que aún no sabemos en qué lenguaje se va a implementar.*
- *Escoge adecuadamente el tipo de relaciones que puede haber entre cada clase. Recuerda que había cuatro posibilidades y que, normalmente, solo es posible una de ellas en cada caso.*

¡IMPORTANTE!

1. Crea tu propio diagrama siguiendo las indicaciones vistas en clase
2. Usa únicamente los editores sugeridos
3. Asegúrate de que una de las clases tiene tu NIF/NIE al final
4. Añade un objeto -Note- con los datos que te indicamos en la práctica
5. Revisa la rúbrica antes de exportarlo
6. Expórtalo a PDF

MODELO B: GESTIÓN DE UNA PIZZERÍA (PIZZA DAW)	ENUNCIADO
<p>Se requiere desarrollar un DIAGRAMA DE CLASES que refleje ciertos aspectos de la gestión de PIZZA DAW, una pizzería que quiere abrir este verano.</p> <p>El sistema gestiona los pedidos que se tramitan desde la pizzería, pudiendo ser estos de tres subtipos: recoger, llevar (a domicilio), local.</p> <p>De cualquier pedido se desea conocer el código del pedido, nombre y apellidos (del cliente) y la fecha, y se necesitan métodos (sin parámetros) para cancelar un pedido, duplicar un pedido y un tercero para congelar un pedido al que le pasaremos el número de minutos en los que el pedido estará marcado como “en espera”.</p> <p>Además, cualquier pedido (sea del subtipo que sea) puede estar compuesto por:</p> <ul style="list-style-type: none"> • Una o varias pizzas personalizadas • Cero varias observaciones • Una o ninguna promoción de un catálogo de promos preexistente <p>De las observaciones nos basta con un campo descripción (“quiere masa fina”, por ejemplo), del catálogo de promos el código de la promo (“2x1”, por ejemplo) y la validez (“enero2022”, por ejemplo).</p> <p>Cada pizza personalizada está compuesta por una pizza del catálogo de pizzas y por cero o varios extras de un catálogo de extras preexistente, con un máximo de 3, teniendo que guardar de cada extra solo el ingrediente y la cantidad.</p> <p>Para los “pedidos de local” almacenaremos la mesa, si está servido (sí/no) y los métodos “entregarBeeper” y “recogerBeeper” para registrar cuándo se le ha dado al cliente el avisador y cuándo lo ha devuelto.</p> <p>En los pedidos para llevar almacenamos la dirección y un campo para saber si está entregado (sí/no).</p> <p>En los pedidos para recoger, necesitamos los campos “prepagado” (sí/no) y “recogido” (sí/no) y un método llamado “banearClientePorNoRecoger” para registrar cuando un cliente hace un pedido y no pasa a recogerlo.</p> <p>De cada pedido se necesita generar un único pago, que en ambos casos podrá ser de tipo PayPal (almacenando la cuenta de PayPal) o de tipo tarjeta (almacenando el número de tarjeta).</p> <p>Nos indica el cliente que todos los métodos deben ser públicos y, si no se indica, no tienen parámetros ni de entrada ni de salida.</p> <p><i>Pista: Una posible solución tiene 12 clases.</i></p>	

Indicaciones:

- *No es necesario que incluyas los setters, getters, constructores y destructores, ni que incluyas tipos de datos en los atributos, ya que aún no sabemos en qué lenguaje se va a implementar.*
- *Escoge adecuadamente el tipo de relaciones que puede haber entre cada clase. Recuerda que había cuatro posibilidades y que, normalmente, solo es posible una de ellas en cada caso.*

¡IMPORTANTE!

1. Crea tu propio diagrama siguiendo las indicaciones vistas en clase
2. Usa únicamente los editores sugeridos
3. Asegúrate de que una de las clases tiene tu NIF/NIE al final
4. Añade un objeto -Note- con los datos que te indicamos en la práctica
5. Revisa la rúbrica antes de exportarlo
6. Expórtalo a PDF

MODELO C: MONTAJE DE EQUIPOS	ENUNCIADO
<p>[Caso ficticio] El gigante Microsoft, tras recibir numerosas incidencias por fallos en sus productos, se ha puesto firme y ha decidido cambiar su política de licencias, de manera que ahora solo dará licencias de software a equipos que tengan todos sus componentes autorizados por ellos. Con esto, cuando alguien quiere adquirir un equipo con una serie de componentes (memoria de esta marca y esta capacidad, disco duro de esta marca y esta capacidad...) debe remitir esa lista a Microsoft para que valide los componentes y, solo entonces, le venderá la licencia.</p> <p>Se pide un DIAGRAMA DE ACTIVIDAD con carriles que refleje este proceso desde el punto de vista del cliente, la tienda y Microsoft.</p> <p>El cliente solicita un equipo con su lista de componentes, en paralelo, a una tienda (por ejemplo APP) y a la web de Microsoft de manera que la tienda recibe la solicitud y espera a que Microsoft le conteste con la licencia compatible.</p> <p>Por su lado, Microsoft recibe la solicitud del cliente (al mismo tiempo que la recibe la tienda), consulta la viabilidad y decide si es compatible o no. Si es compatible, se comunica con la tienda para que esta comience con el montaje del equipo, en caso contrario, envía un mensaje al usuario de "equipo denegado".</p> <p>Solo cuando la tienda ha recibido la solicitud del equipo y la confirmación de viabilidad de Microsoft, da el siguiente paso que es comprobar que dispone de los componentes que pide el cliente. Si dispone de los componentes monta el equipo y se lo entrega al cliente y acaba el proceso y, si no los tiene, envía un mensaje al usuario de "equipo denegado".</p> <p>No te preocupes por cuándo recibe la tienda la licencia para instalar el software autorizado. No es relevante.</p> <p>Deben usarse carriles y no es necesario indicar cómo pasan los objetos de un proceso a otro. <i>Pista: Una posible solución tiene 8 actividades.</i></p> <p>Indicaciones:</p> <ul style="list-style-type: none"> • Recuerda que de un mismo proceso solo puede salir o entrar una flecha, por lo que tienes que escoger adecuadamente los nexos o uniones entre procesos para poder implementar más de una flecha de entrada o de salida. <p>¡IMPORTANTE!</p> <ol style="list-style-type: none"> 1. Crea tu propio diagrama siguiendo las indicaciones vistas en clase 2. Usa únicamente los editores sugeridos 3. Asegúrate de que una de las actividades tiene tu NIF/NIE al final 4. Añade un objeto -Note- con los datos que te indicamos en la práctica 5. Revisa la rúbrica antes de exportarlo 6. Expórtalo a PDF 	

MODELO D: VOLAR EN PANDEMIA	ENUNCIADO
<p>[Caso ficticio] Las agencias de viajes ya no pueden vender billetes de avión sin una autorización previa del Ministerio de Sanidad que debe solicitar el propio cliente.</p> <p>Se pide un DIAGRAMA DE ACTIVIDAD con carriles que refleje este proceso desde el punto de vista del cliente, la agencia de viajes y el Ministerio de Sanidad.</p> <p>El cliente, cuando quiere reservar un vuelo, debe hacer dos solicitudes en paralelo:</p> <ul style="list-style-type: none">• Una solicitud de vuelo a la agencia de viajes• Una solicitud de Pasaporte Sanitario (PS) al Ministerio de Sanidad <p>Cuando la agencia recibe la petición queda en espera hasta que el Ministerio le envíe el visto bueno (vuelo autorizado).</p> <p>Por su parte, el Ministerio recibe la solicitud de PS, consulta la viabilidad de la petición (registro de PCR, vacunas, estado de la pandemia en el país destino, legislación actual...) y decide si autorizarlo o no. Si lo autoriza, envía a la agencia un mensaje de "vuelo autorizado" y, en caso contrario, informa directamente al cliente con "billete denegado".</p> <p>Una vez la agencia recibe la solicitud de vuelo y el mensaje positivo del Ministerio, comprueba la disponibilidad y, si hay asientos, emite el billete y se lo envía al cliente. Este lo recibe y acaba el proceso. En caso contrario, informa directamente al cliente con "billete denegado".</p> <p>Deben usarse carriles y no es necesario indicar cómo pasan los objetos de un proceso a otro. <i>Pista: Una posible solución tiene 8 actividades.</i></p> <p>Indicaciones:</p> <ul style="list-style-type: none">• <i>Recuerda que de un mismo proceso solo puede salir o entrar una flecha, por lo que tienes que escoger adecuadamente los nexos o uniones entre procesos para poder implementar más de una flecha de entrada o de salida.</i> <p>¡IMPORTANTE!</p> <ol style="list-style-type: none">1. Crea tu propio diagrama siguiendo las indicaciones vistas en clase2. Usa únicamente los editores sugeridos3. Asegúrate de que una de las actividades tiene tu NIF/NIE al final4. Añade un objeto -Note- con los datos que te indicamos en la práctica5. Revisa la rúbrica antes de exportarlo6. Expórtalo a PDF	