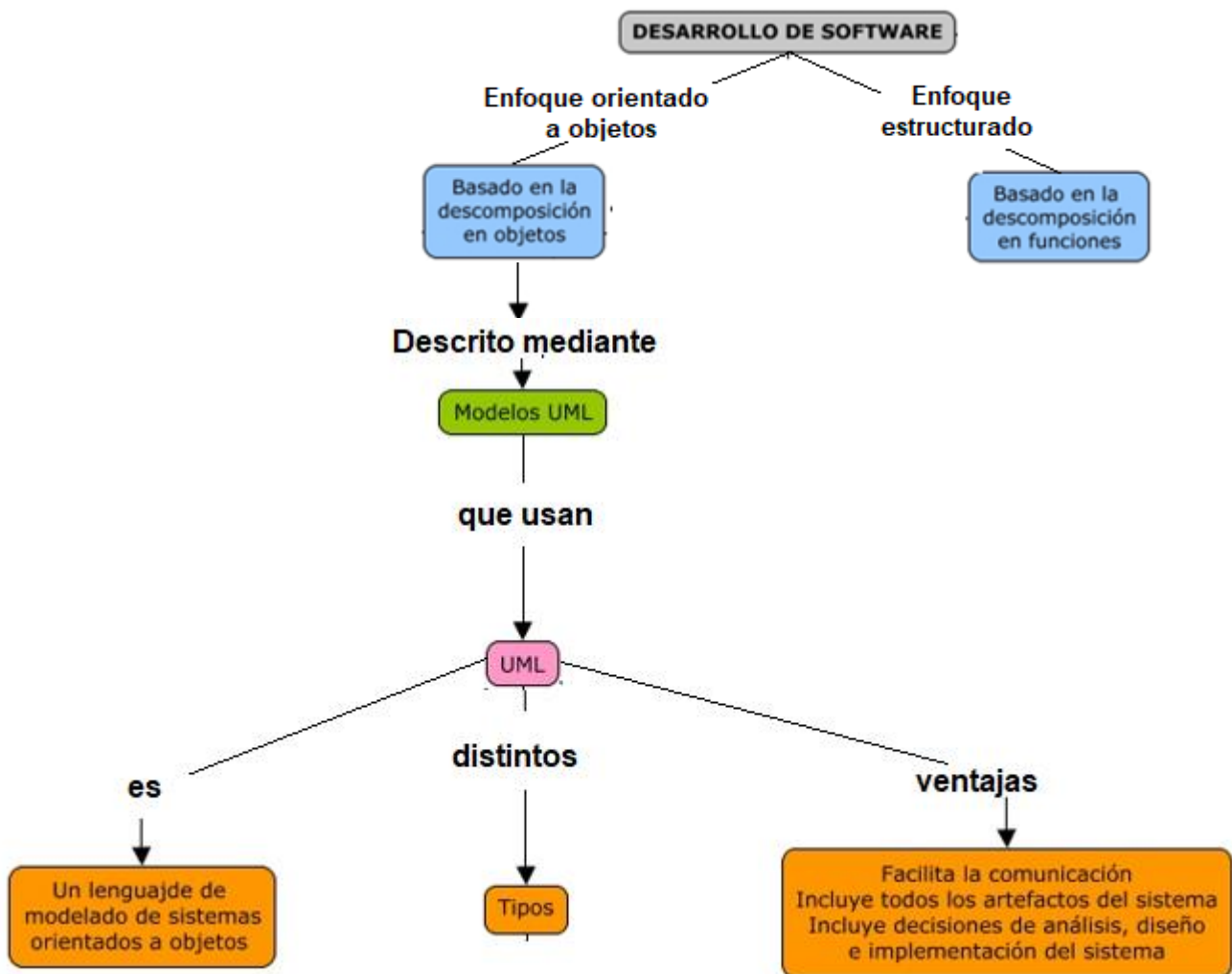


## INDICACIONES PARA RESOLVER LA TAREA 5 (Clase 20/03/2023)

### OBJETIVOS DE LA TAREA

1. Generar **diagramas de clases** valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.
2. Generar **diagramas de comportamiento** valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

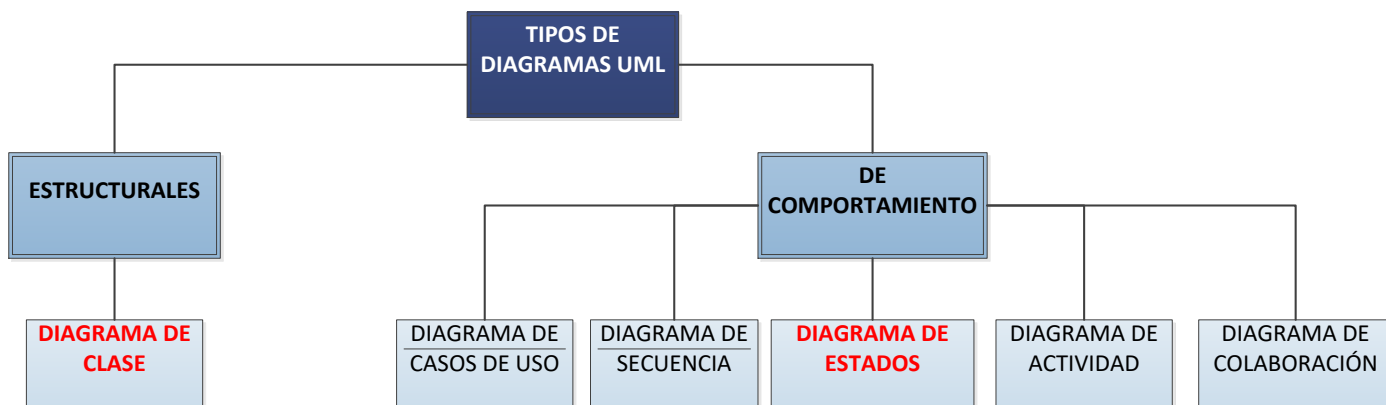
### CONCEPTOS TEÓRICOS BÁSICOS



### ¿QUÉ ES UML?

**UML** (*Unified Modeling Language* o *Lenguaje Unificado de Modelado*) es un conjunto de herramientas que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

- Porque permite utilizar un lenguaje común que facilita la comunicación entre el equipo de desarrollo.
- Con UML podemos documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, ...) por lo que se dispone de documentación que trasciende al proyecto.
- Hay estructuras que trascienden lo representable en un lenguaje de programación, como las que hacen referencia a la arquitectura del sistema, utilizando estas tecnologías podemos incluso indicar qué módulos de software vamos a desarrollar y sus relaciones, o en qué nodos hardware se ejecutarán cuando trabajamos con sistemas distribuidos.
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose modelos precisos, no ambiguos y completos.

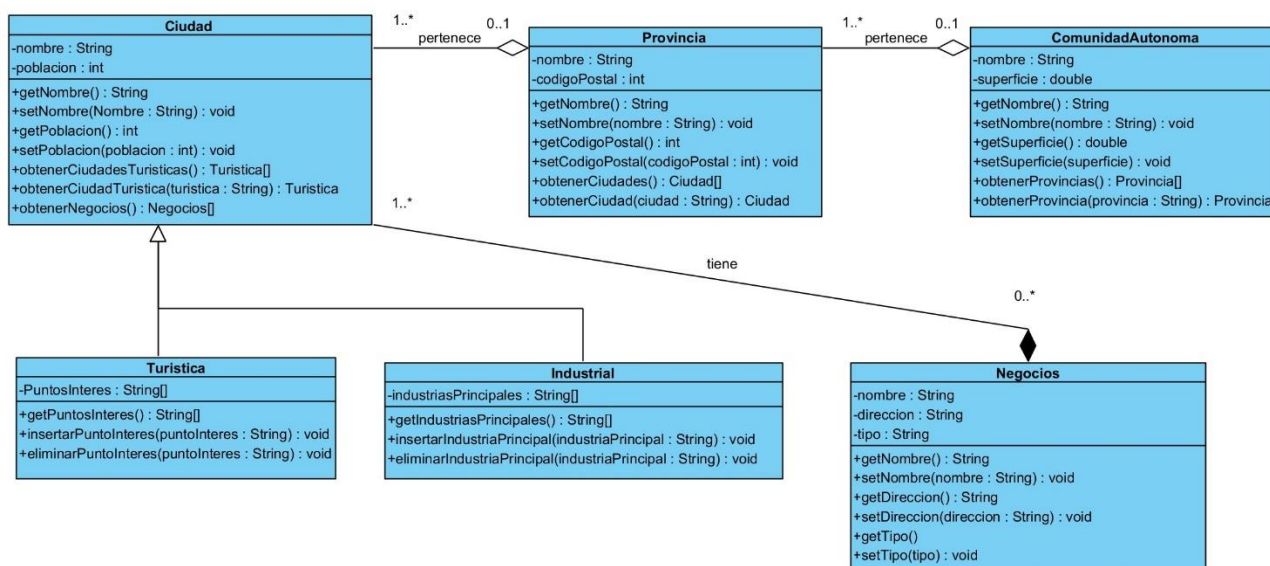


## PASOS DE LA TAREA

### Parte 1: Diagramas estructurales

**Ejercicio 1:** Elabora el siguiente diagrama de clases mediante el programa Visual Paradigm. Debes crear el proyecto VP-UML.

#### INDICACIONES



#### INDICACIONES EJERCICIO 1

Para descargar e instalar la herramienta CASE Visual Paradigm, debéis acceder dentro del contenido del tema, al apartado

### Anexo I.- Descarga e instalación de Visual Paradigm.

Encontraréis ejemplos de diagramas de clase resueltos, tanto en el enunciado de esta tarea, como en el **Anexo II.- Generación del diagrama de clases de un problema dado.**

En los siguientes enlaces, también podéis encontrar información de interés:

- <https://dokumen.tips/documents/manual-de-visual-paradigm.html?page=13>
- <https://ocw.unican.es/pluginfile.php/2168/course/section/1988/is1-p01-trans.pdf>
- [https://www.youtube.com/watch?v=yVRM8MOnhqY&ab\\_channel=YoAndroide](https://www.youtube.com/watch?v=yVRM8MOnhqY&ab_channel=YoAndroide)
- [https://www.youtube.com/watch?v=JERoVM3umL8&ab\\_channel=FabianFranco](https://www.youtube.com/watch?v=JERoVM3umL8&ab_channel=FabianFranco)
- [https://www.youtube.com/watch?v=3BXdiMFLQ1Y&ab\\_channel=CynthiaPacheco](https://www.youtube.com/watch?v=3BXdiMFLQ1Y&ab_channel=CynthiaPacheco)

**Ejercicio 2:** Importación del proyecto VP-UML en un proyecto de NetBeans. (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).

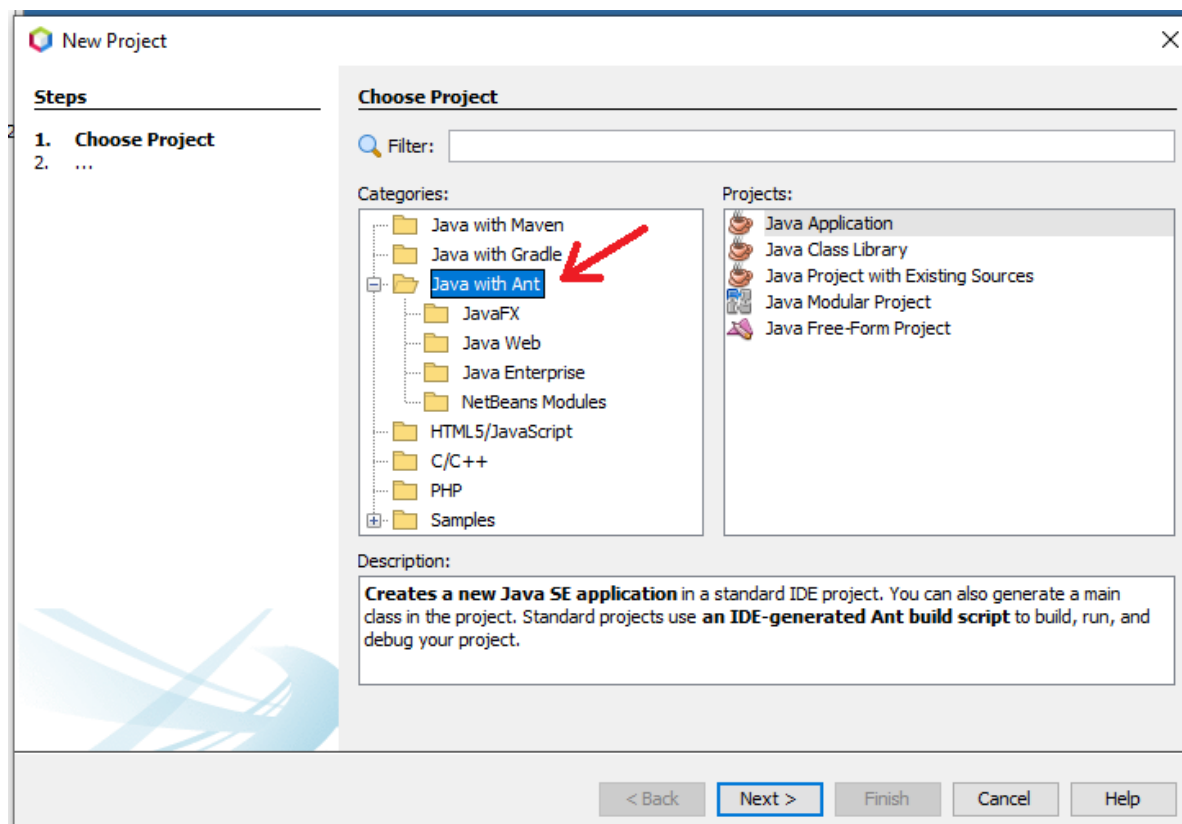
### INDICACIONES EJERCICIO 2

1º Antes de hacer este apartado hay que integrar Visual Paradigm con NetBeans. Tenéis indicaciones de cómo hacerlo en el [Anexo I.- Descarga e instalación de Visual Paradigm](#).

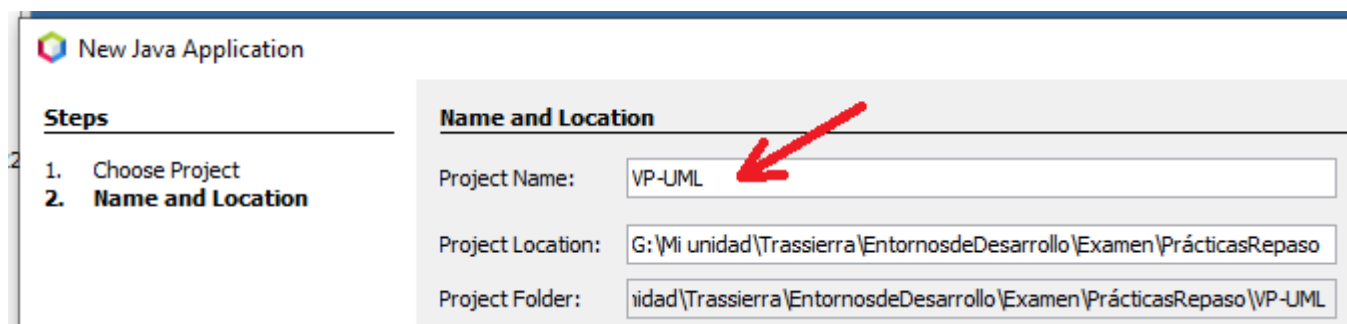
Y también os dejo el siguiente vídeo:

- [https://www.youtube.com/watch?v=teg5NrprjA&list=PL\\_G4cw7yREsfEfPhq0SMI6fGKTeeAy8JX&ab\\_channel=J.RoMo](https://www.youtube.com/watch?v=teg5NrprjA&list=PL_G4cw7yREsfEfPhq0SMI6fGKTeeAy8JX&ab_channel=J.RoMo)

2º Desde Netbeans hay que crear un proyecto con el mismo nombre que el diagrama de clases creado y en él importaremos nuestro proyecto



Recuerda el nombre:



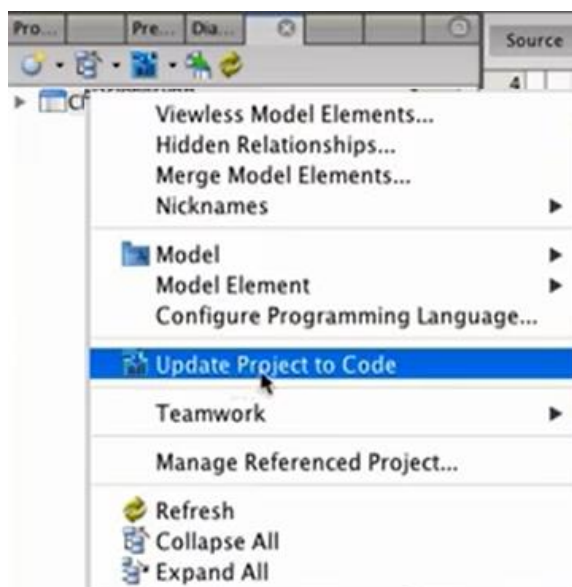
Vídeo de ayuda de cómo hacer la importación:

[https://www.youtube.com/watch?v=k5l8DmVjKbw&ab\\_channel=J.RoMo](https://www.youtube.com/watch?v=k5l8DmVjKbw&ab_channel=J.RoMo)

**Ejercicio 3:** Generación del código a partir del diagrama de clases realizado.

### INDICACIONES EJERCICIO 3

En este apartado debes generar el código JAVA a partir del diagrama UML importado.



Vídeos de ayuda

- [https://www.youtube.com/watch?v=k5l8DmVjKbw&ab\\_channel=J.RoMo](https://www.youtube.com/watch?v=k5l8DmVjKbw&ab_channel=J.RoMo)

**Ejercicio 4:** Mediante el proceso de ingeniería inversa, genera el diagrama de clases a partir del siguiente [proyecto Java](#). Si al realizar este proceso faltan relaciones, tendrás que establecerlas a mano.

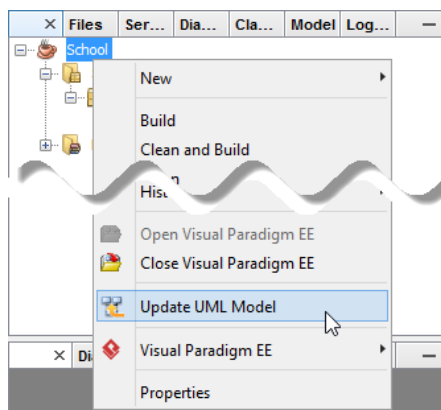
### INDICACIONES EJERCICIO 4

**¿Qué es la ingeniería inversa?** → Permite el proceso inverso al que hemos realizado en el apartado anterior. Consistiría en, a partir de un código, generar el diagrama UML correspondiente.

En este apartado podemos descargarnos el proyecto JAVA y debemos generar el diagrama de clases asociado.

1º Una vez descargado el proyecto, se importa desde Netbeans

2º



Enlace de interés para seguir los pasos: [https://www.visual-paradigm.com/support/documents/vpuserguide/2381/2384/66565\\_reverseengin.html](https://www.visual-paradigm.com/support/documents/vpuserguide/2381/2384/66565_reverseengin.html)

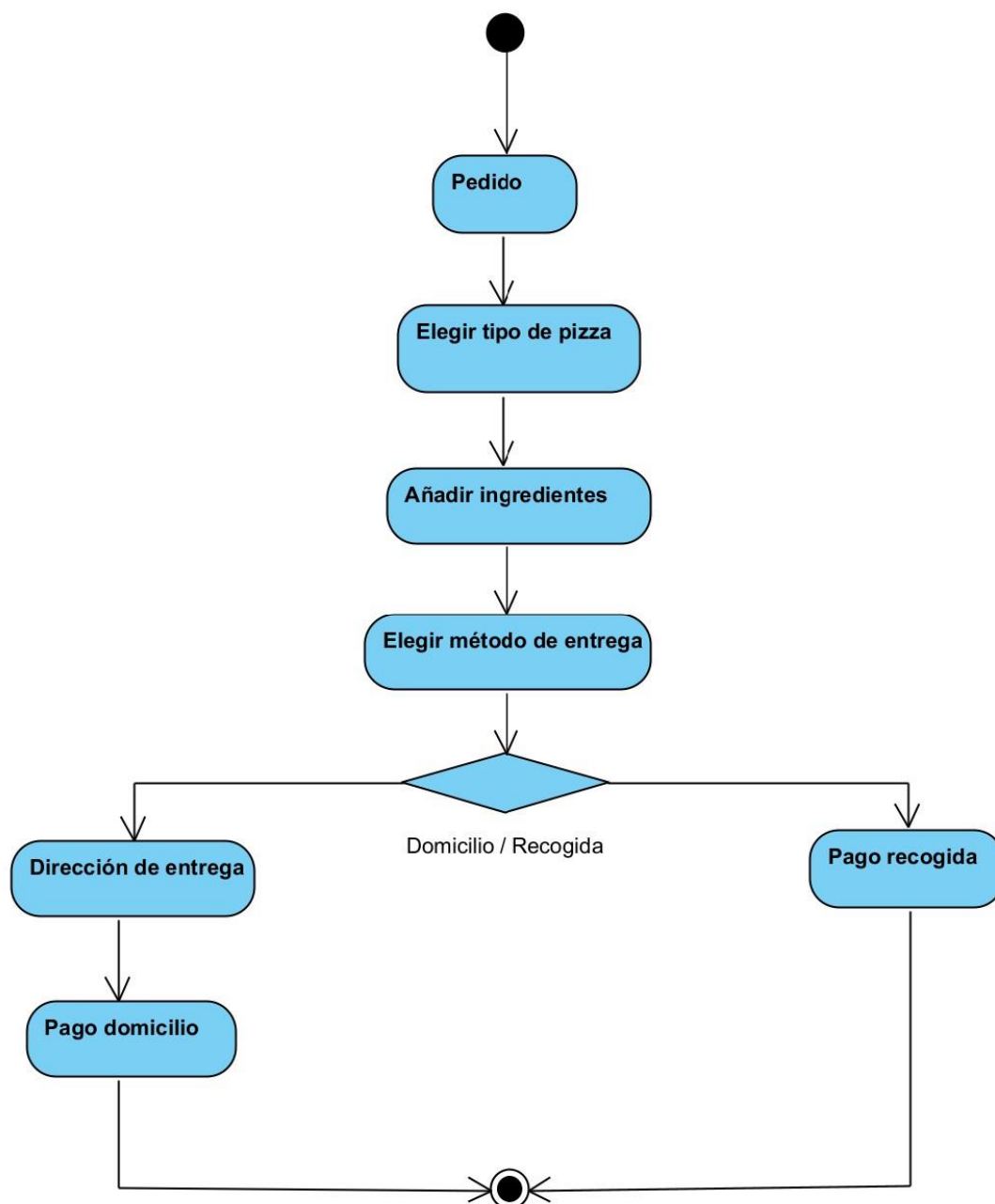
## Parte 2: Diagramas de comportamiento

**Ejercicio 5:** Indica los distintos tipos de diagramas de comportamiento. Define qué son y para qué sirven los **diagramas de secuencia**.

### INDICACIONES EJERCICIO 5

Esta pregunta es teórica.

**Ejercicio 6:** Identifica el tipo de diagrama e interpreta lo que está representando sobre pedidos de pizza.



### INDICACIONES EJERCICIO 6

De nuevo se trata de una pregunta teórica.

**Ejercicio 7:** Realiza el **diagrama de estados** para la siguiente operatoria de un ascensor:

El ascensor dispone de los siguientes estados: inactivo, subiendo, bajando, puertas abiertas y puertas cerradas.

La posición inicial parte del estado inactivo, en el cual se mantiene hasta que se pulsa un botón, pasando al siguiente estado subiendo o bajando, dependiendo de la dirección del piso en el que se ha pulsado el botón.

Cuando llega al piso, vuelve a la posición de inactivo, y abre las puertas, cambiando a este estado. A continuación, cierra las puertas, cambiando a dicho estado, y vuelve a estar inactivo a la espera de que se pulse un botón.

#### INDICACIONES EJERCICIO 7

Enlace que explica cómo hacer diagramas de estado con visual paradigm+

<https://youtu.be/lwCX5yrAqXo>

Vídeo explicativo de un ejemplo de diagrama de estado:

[https://www.youtube.com/watch?v=vPylavs\\_xcE&ab\\_channel=GaRbyRuiz](https://www.youtube.com/watch?v=vPylavs_xcE&ab_channel=GaRbyRuiz)

**Ejercicio 8:** Realiza el diagrama de **casos de uso** para el siguiente supuesto:

Vamos a realizar la gestión de una tienda online, donde tendremos tres actores “Cliente”, “Cliente Premium” y “Administrador”.

El cliente puede realizar diversas acciones, como buscar productos, añadir productos al carrito, realizar un pedido, consultar el histórico de pedidos, realizar el seguimiento de los pedidos y cancelar un pedido. Cuando realiza un pedido debe elegir el método de pago, el método de envío, revisar el pedido y confirmar el pedido. También puede aplicar un cupón en el pedido (de manera opcional). Cuando cancela un pedido puede solicitar la devolución del mismo (de manera opcional).

El cliente premium, además de todas las acciones que puede realizar un cliente, puede canjear los puntos obtenidos por cada pedido realizado.

El administrador puede gestionar productos, consultar pedidos y gestionar clientes.

#### INDICACIONES EJERCICIO 8

Enlace que explica cómo hacer diagramas de casos de uso con visual paradigm+

[https://www.youtube.com/watch?v=x4EdxcH5wxw&ab\\_channel=LuisFelipedeJes%C3%BAsHern%C3%A1ndezQuintanar](https://www.youtube.com/watch?v=x4EdxcH5wxw&ab_channel=LuisFelipedeJes%C3%BAsHern%C3%A1ndezQuintanar)

Ejemplos de diagramas de casos de uso resueltos

[http://www.lsi.us.es/~javierj/cursos\\_ficheros/metricaUML/CasosUsoUML.pdf](http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/CasosUsoUML.pdf)