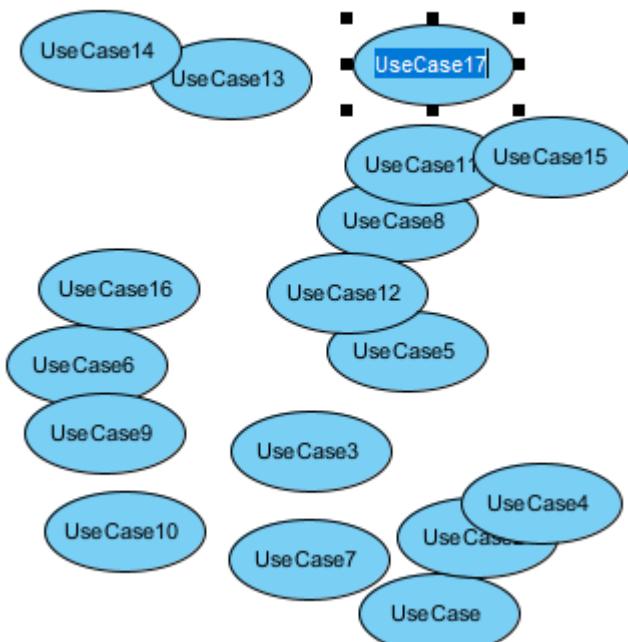


# Consejos para aplicar en la resolución de la Guía de Trabajos Prácticos / Portafolio Individual Obligatorio

---

## Análisis de Sistemas

- \* Modelo de Dominio (MDOM)
- \* Análisis Orientado a Objetos (AOO)
- \* Diagrama de Casos de Usos (CU)
- \* Máquinas de Estado (DES/DAC)
- \* Modelo de Análisis (MANA)



**DISCLAIMER: Antes de poner en práctica los contenidos expuestos en el presente documento, he de aclararle que, según reportes de alumnos que han cursado la materia en cuestión, el criterio de evaluación del profesor puede estar sujeto a cambios de un cuatrimestre al otro.**

### **El porqué de este apunte**

Como alumno he notado que algunas explicaciones y aclaraciones hechas por el profesor no se encuentran presentes en los PDFs provistos, y que probablemente no sean tenidas en cuenta por aquellos alumnos que no estén presentes en las clases.

A su vez, no quería que estas notas fuesen de utilidad únicamente para mí, así que sentí que debía compartirlas con quien pudiera hallar en ellas una ayuda.

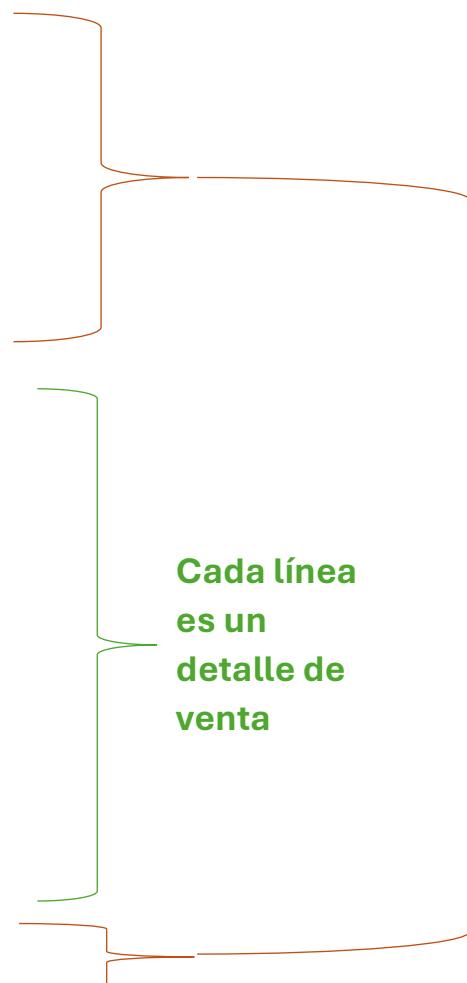
Este documento es una recopilación de comentarios y anotaciones hechas por el docente Fernando La Rosa a lo largo de la cursada del viernes mañana del 1C2025.

## Modelo de Dominio

- Todo pago realizado de manera digital, es decir, cualquier método diferente al pago en efectivo, debe ser asociado a una clase “Pago”.
- Los informes que son producto de una impresión o una visualización de datos del dominio no se representan como clase.
- Toda transacción debe mantener el precio, fecha y hora con el que fue realizada.
- Toda Venta capaz de incluir más de un producto, artículo o reserva, debe tener asociada una clase “Línea de venta”. La misma representa cada ítem vendido y, a su vez, permite tener un seguimiento de la información específica para cada producto.

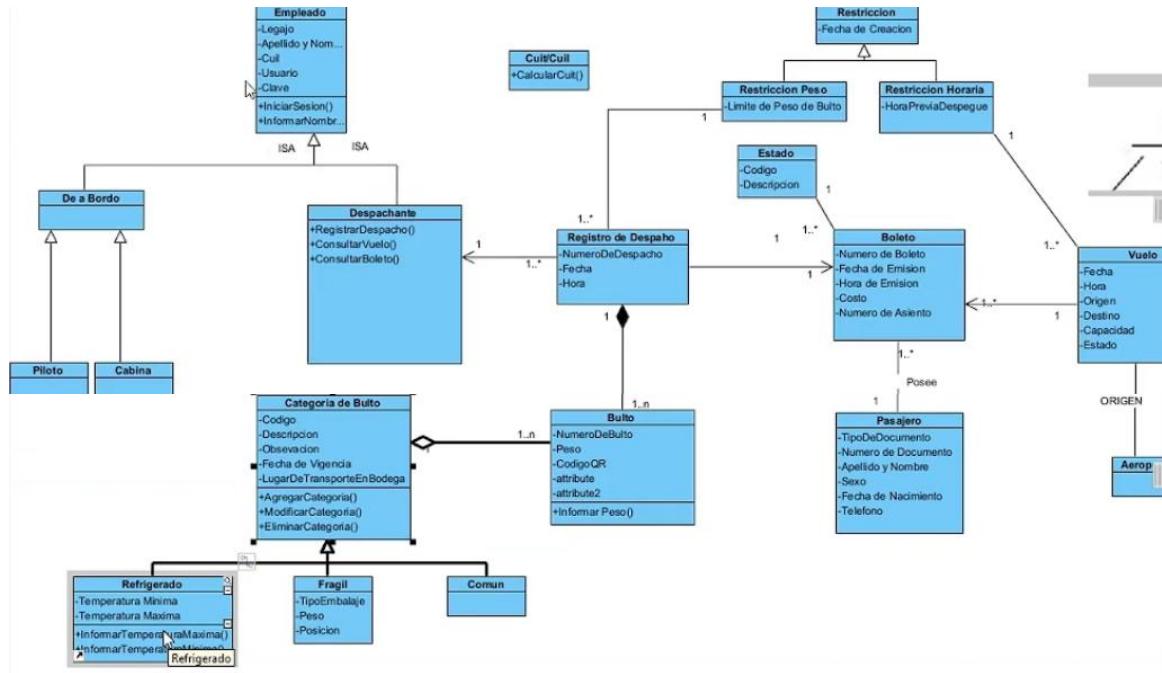
**C**

FACTURA			
Carlos C. Vanhoy Juan Domingo Perón 1946 9º B La Matanza, Pcia. Buenos Aires 2040 - Argentina	FECHA 16 10 20 C.U.I.T. N° 20-293649835-8 Ingresos Brutos: No Exento Inicio de Actividades: 01/2008		
RESPONSABLE MONOTRIBUTO			
Señores: Instituto Patria - SA. Dirección: Rodríguez Sáenz Peña 80 Loc Capital Federal			
IVA	Responsable Inscripto <input type="checkbox"/> Responsable Monotributo <input type="checkbox"/> Exento <input type="checkbox"/> No Responsable <input type="checkbox"/> Cons. Final <input type="checkbox"/>		
C.U.I.T. N°: 20-293649835-8			
Condiciones de Venta:	Contado <input checked="" type="checkbox"/> Cta. Cta. <input type="checkbox"/> REMITO N°: _____		
CANT.	DETALLE	Precio Unit.	IMPORTE
1	Asistencia marcha 17-08 Octubre 2020	1.500	
Pesos mil quinientos			
ORIGINAL BLANCO - DUPLICADO COLOR		TOTAL \$	1500-
1147 teléfono Gratuito C.A.B.A., Área de Defensa y Protección al Consumidor			



- Las restricciones (horarias, de edad, peso, por nombrar ejemplos) y los descuentos/promociones/beneficios se modelan como clase aparte. Lo mismo aplica para aquellos aspectos parametrizables del sistema que responden a reglas del negocio y deben ser debidamente registrados o cargados.
- Si en el enunciado se da una descripción general de un concepto, se puede englobar en una sola clase. En cambio, si se tienen muchas especificaciones o descripciones, requiere un desglose. Por ejemplo, se tiene la clase "Entrada", que, según el enunciado, está asociada a otra clase "Tipo de entrada", cada cual con sus propios beneficios.
- Si se menciona en el enunciado que un concepto "depende de" otro, significa que vincula con una o más clases.

## Análisis Orientado a Objetos



Air Patagonia – AOO (recortado)

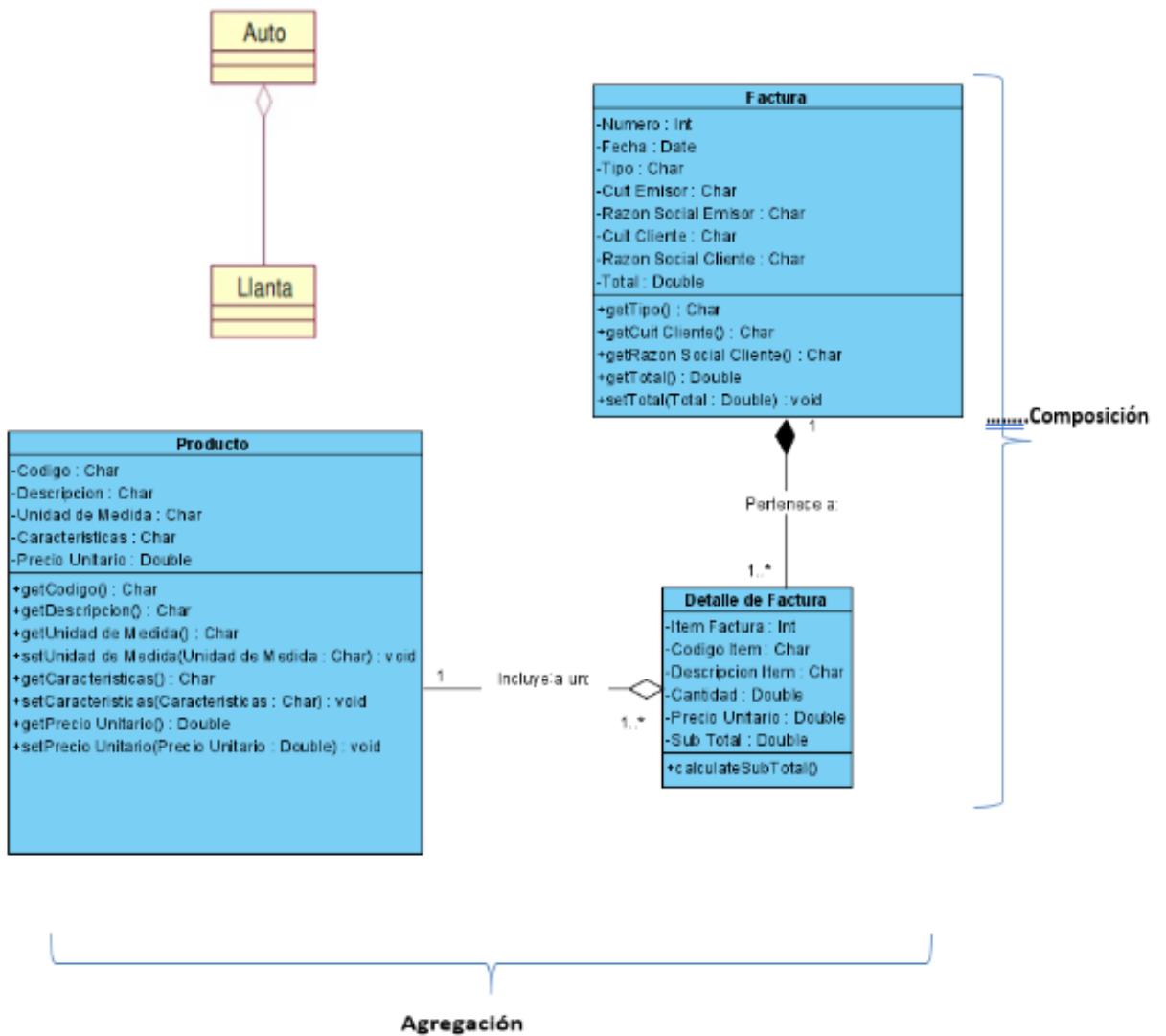
- Cuando hallamos que varias clases comparten una serie de atributos y/o métodos en común, podemos construirlas a partir de una superclase “padre” de la cual heredan estos atributos y/o métodos, teniendo cada una (hijos) sus diferencias propias entre sí. Un ejemplo claro es el de una clase padre “Persona”, que tiene como hijos las subclases “Estudiante” y “Empleado”, pero que a su vez esta última es madre de las clases “Administrador” y “Docente”.  
“Is A” (Es un) representa esta relación.

- La composición es una relación donde la no existencia del todo invalida la parte. Es decir, una clase requiere de otra para tener sentido. Es el ejemplo de las líneas/detalles de venta, que no existen sin que haya una venta propiamente dicha.

La relación es un “Pertenece a un”.

- La agregación es una relación donde una clase está formada por objetos de otra clase, pero cuya existencia no es requisito para que tenga sentido el objeto incluido. La no existencia del todo no invalida la parte.

La relación es un “Incluye a un” o “Tiene a un”.



Ejemplo de una “venta” con “detalle de venta” donde se aplica tanto composición como agregación.

## Diagrama de Estados

- Antes de empezar a modelar, es necesario plantear si tiene sentido estudiar los posibles estados del objeto escogido.
- Siempre hay que dejar una nota donde se especifica cuál es el objeto de análisis
- Un objeto de análisis puede tener o no un estado final. El ejemplo más común es el de una puerta automática: siempre vuelve al estado inicial (cerrada).
- Comprender y aplicar estados que engloben subestados dotan a la resolución de mayor complejidad.

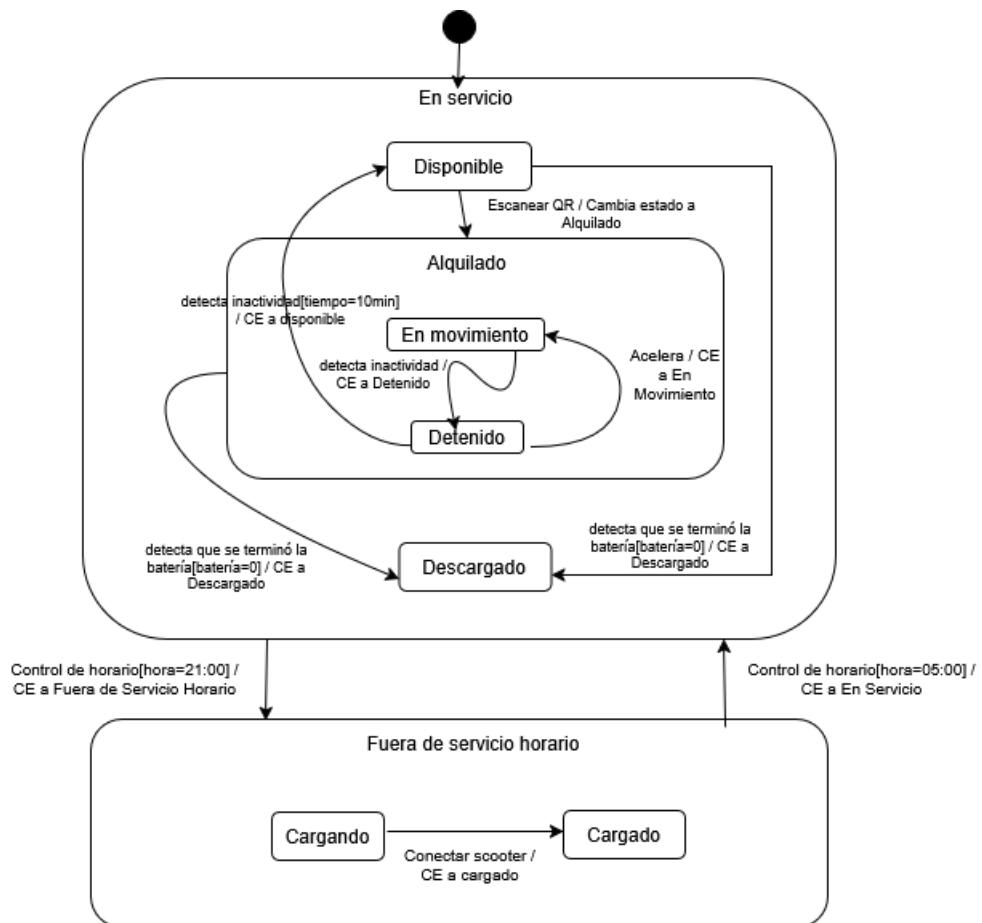


Diagrama de Estados - Alquiler de Scooters

## Diagrama de Actividades

- El nodo de decisión no lleva nombre y es precedido por una actividad de verificación cuyo resultado se bifurca a partir del rombo. El nodo de decisión NO es un if, es un Branch.
- Si estoy modelando un proceso que se llama "tal cosa", debe existir la actividad "tal cosa" en el diagrama de actividades. 🚨🔥

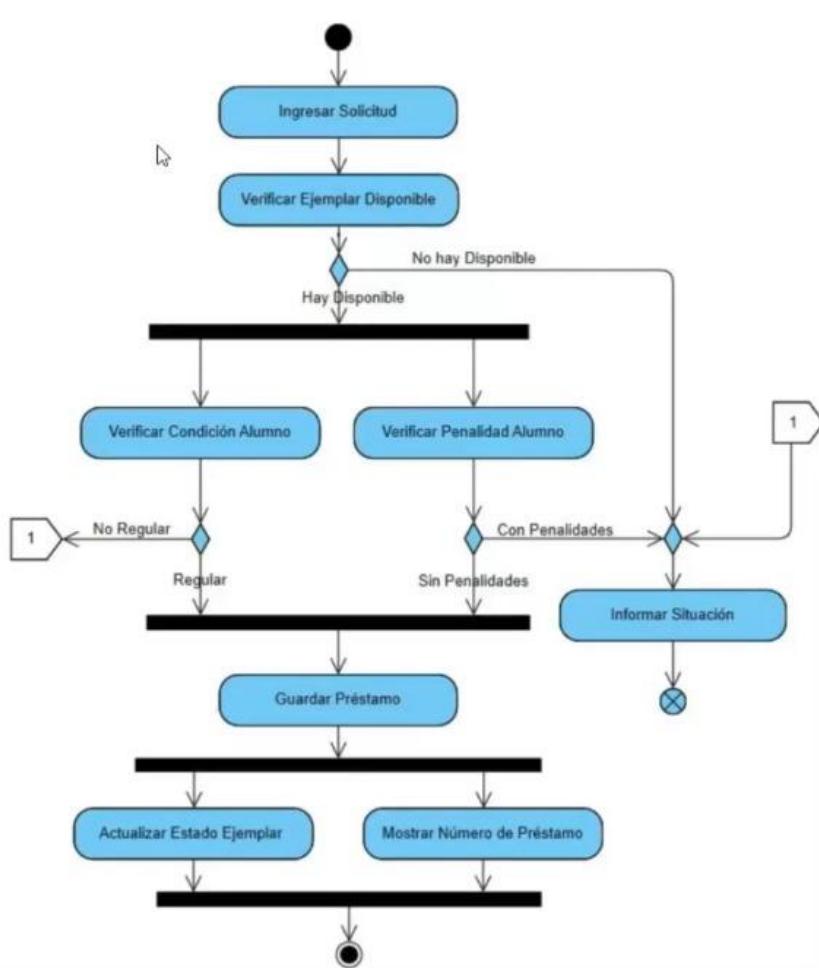
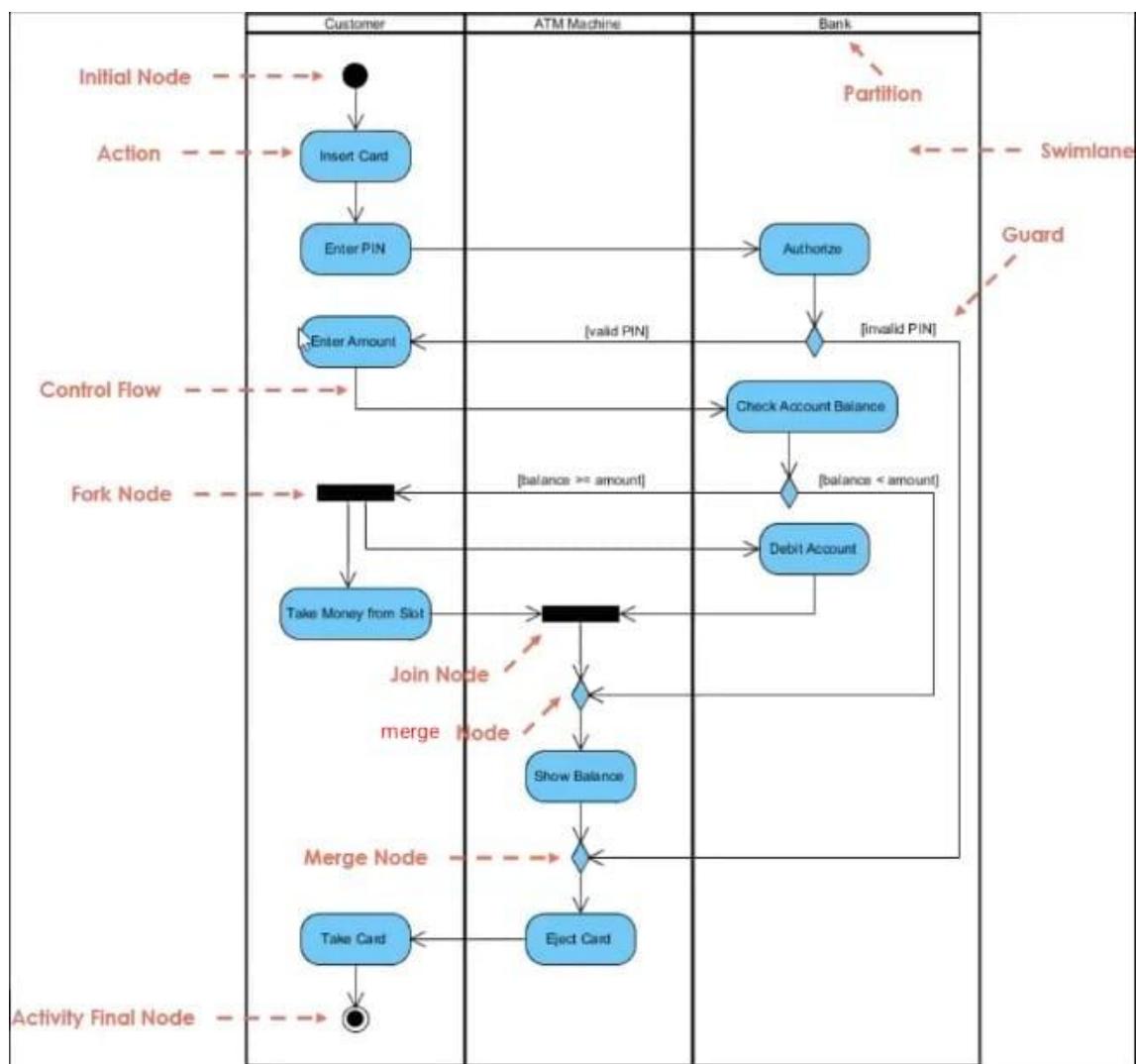


Diagrama de Actividades: pedir un préstamo de un libro de la biblioteca (Ejemplo extraído de una clase virtual)

- Utilizar swimlanes, en la medida de lo posible, con tal de tener una separación marcada entre las acciones que realiza cada actor.
- Una actividad no puede tener dos flechas que van hacia ella. En este caso, se coloca un nodo de tipo merge, que recibe varias salidas y las junta en una sola.

- Las barras de sincronización, una vez abiertas, siempre tienen un cierre.
- Las actividades precedidas de barras de sincronización representan acciones que pueden realizarse en un orden indistinto, pero cuya resolución es necesaria para continuar con el flujo del programa.



Ejemplo de DAC de un sistema de cajero automático, aplicando todos los elementos vistos en la materia.

## Diagrama de Casos de Uso

- Las funcionalidades de “Registrar cuenta” o “Crear usuario” pueden ser omitidas. En el caso de que se desee incluirlas, deben ser llevadas a cabo por un actor principal conocido como “Invitado”. Favor de no utilizar el nombre “Anónimo”, como comúnmente se hace en la materia anterior, Introducción a la Gestión de Requisitos.
- Para que un actor pueda seleccionar un atributo o un valor de un CU, ese valor o atributo debe estar asociado a una clase conceptual del modelo de dominio.
- Para que un atributo o valor esté en una clase conceptual, primero debe de haber sido registrada por un caso de uso o actor.
- La mayoría de casos de uso tiene un impacto en la base de datos del sistema.
- Todo aquello que está fuera del dominio, no aplica dentro del modelo de casos de uso.
- Para que algo sea un actor, debe actuar autónomamente.
- Los pasos de un CU no son Casos de Uso propios. Es importante pensar en si lo que estamos por escribir es realmente un CU o simplemente otro paso del flujo normal/alternativo.
- El pago con tarjeta es una funcionalidad propia.
- Un CU "Include" es una funcionalidad obligatoria que se ejecuta siempre que se realice un CU Base. Es el ejemplo de un sistema de ventas donde se tiene la opción “Pago con tarjeta” como único método de pago admitido o donde este se toma como flujo principal.
- Un CU “Extend” se ejecuta opcionalmente durante el flujo de un CU base. En el caso de efectivamente ser ejecutado, se hace de manera inmediata. Es decir, no se puede tener un CU Extend que se realiza X tiempo luego de la finalización del CU base.

## Especificación de Caso de Uso

- La Especificación de un Caso de Uso es un ping-pong entre el actor y el sistema: el actor hace, el sistema responde.
- Si se tiene más de un método de pago posible, conviene escoger uno para el flujo normal y representar el/los otro/s en los flujos alternativos.
- A la hora de describir en un flujo los descuentos asociados a una venta, es recomendable especificar que, de existir, el sistema los aplica. De esta forma, nos deshacemos de tener que plantear un flujo normal donde se aplican descuentos y otro alternativo donde esto no sucede.
- No se recomienda escribir “El sistema verifica ...”.
- El actor siempre recibe feedback del sistema a través de la interfaz gráfica (“El sistema muestra ...”).
- Cuando sucede que el sistema debe comunicarse con un sistema externo para realizar una determinada acción, puede expresarse la situación de la siguiente manera:  
[Se incluye comunicación con el sistema de tarjetas de crédito a fines de procesar el pago]

## Modelo de Clases de Análisis

- La clase INTERFAZ no tiene inteligencia. Solo puede recibir mensajes para comunicarlos al exterior o permite ingresar información desde el exterior hasta nuestro caso de uso, pero no tiene ningún tipo de autonomía.

Por otro lado, la clase ENTIDAD solo permite mantener almacenados datos requeridos para el caso de uso y realizar sobre ellas operaciones CRUD (Create, Read, Update, Delete).

La clase CONTROLADOR es la única que tiene inteligencia.

- En el diagrama de clases de análisis, se modela generalmente el flujo principal, aunque se pueden agregar elementos propios de un flujo alternativo a fines de mostrar un panorama completo, siempre y cuando se especifique que estos pertenecen a dicho camino alterno.

- No puedo tener ninguna clase ENTIDAD en el Modelo de Clases de Análisis que no esté previamente definida en el Modelo de Dominio. No obstante, puedo tener clases en el MDOM que no figuren como clase ENTIDAD en el MANA.

- Clases Entidad: son las clases del MDOM. Son aquellas importantes del dominio. Representan información persistente.

- Clases interfaz: Representa la comunicación entre dos sistemas. Se utiliza para modelar la interacción con un sistema externo

Para el ejemplo de una Estación de Servicio, podríamos tener las interfaces: lector de tarjetas, sistema de tarjetas de crédito, horquilla, manguera, impresora de ticket

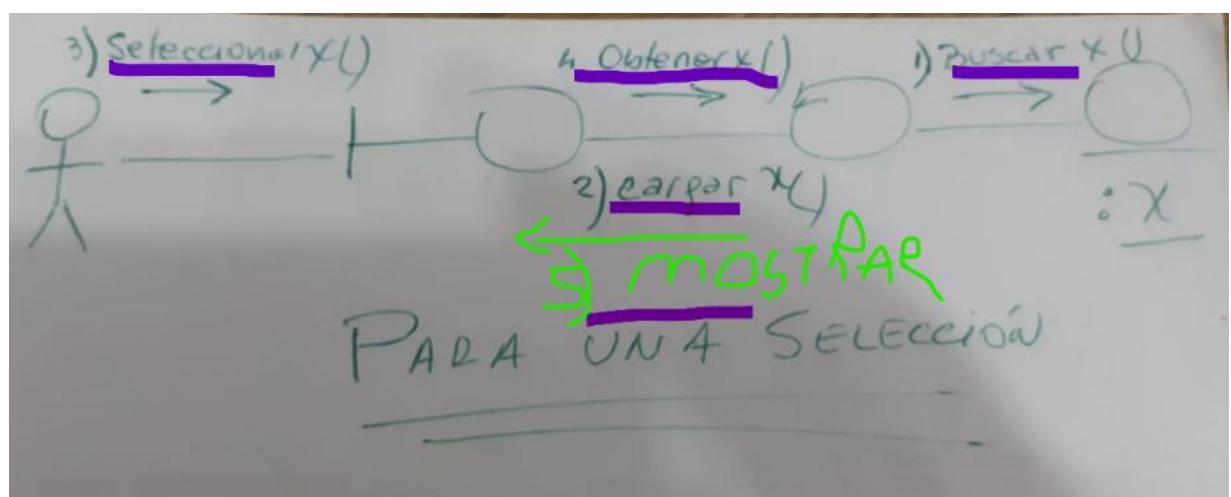
Otros ejemplos pueden ser: interfaz de Sistema de Correo Electrónico, interfaz WhatsApp, Interfaz Facebook.

- Clase gestor: es aquella encargada de enviar mensajes a las clases interfaz y entidades, a la vez que a sí mismo. Realiza las operaciones lógicas del flujo que se está modelando, séase verificar, calcular, solicitar información a un sistema externo, etc. A su vez, permite realizar operaciones CRUD sobre las entidades.

## Modelo de Comunicación de Análisis

- En el contexto del dictado de la materia, vamos a tener siempre una clase control/gestora con el mismo nombre que el caso de uso, y en general, no siempre, deberíamos tener una clase interfaz con el mismo nombre (que representa la interfaz gráfica con la que interactúa el actor).
- Existe uno para cada flujo del CU. En el contexto de la materia, modelamos solo el flujo principal.
- En el modelo de comunicación de análisis (basándonos en la especificación del caso de uso) solamente se grafican aquellas clases que intervienen en el flujo analizado. No puede haber una clase que no reciba ningún mensaje.
- Los casos de uso extend/include los representamos con una clase interfaz.
- Cada vez que a un sistema externo se le pregunta algo a través de una interfaz, el mismo debe responder siempre.
- No se registra nada hasta que el actor confirma la correcta ejecución del caso de uso
- Siempre se guarda primero la venta antes de la línea de venta, el despacho antes del bulto, la inscripción antes del detalle de inscripción, la reserva antes del detalle. Sin despacho/inscripción/venta, no hay bulto/detalle/línea de venta.
- Las clases entidad solamente pueden recibir mensajes del gestor.
- Las clases Interfaz pueden recibir mensajes del gestor o del actor y solo pueden enviar mensajes al gestor.
- La clase gestor envía un mensaje a clases entidad (la respuesta de éstas últimas está implícita) e Interfaces (siempre tiene respuesta).

- El gestor es como un mozo en un restaurante. El cliente no interactúa directamente con el cocinero, sino que usa como intermediario al mozo.



Ejemplo del profesor