

	<b>Taller 4 – ARQUITECTURA DE SOFTWARE I</b> <b>UML</b>	
	<b>Edison Leonardo Neira Espitia</b>	16/08/2016

### Tienda virtual de videos

El proyecto consiste en desarrollar una aplicación empresarial, que simule un modelo de venta de videos por internet. La arquitectura debe ser totalmente orientada a la web (Responsive, multidispositivo).

La primera fase de la implementación consta de la creación de los diseños y plantillas orientados a la conversión de clientes (Compra de videos y registros de clientes) especialmente diseñadas pensando en la experiencia de usuario.

La segunda fase, consta de clientes livianos (Web Browser), lógica de navegación residente en un Web Server Tomcat y una base de datos Oracle. Esta implementación se realizará utilizando Spring View,

La tercera fase, es la implementación de la aplicación realizada en la fase 2, bajo el framework Spring (Spring mvc, Spring boot, etc).

El sistema a desarrollar consiste en un sistema de venta de videos en el que se tienen 3 actores: Cliente, Administrador de la tienda y entidad bancaria (validadora de tarjetas de crédito).

El primero de ellos (Cliente) puede realizar las siguientes operaciones ofrecidas en línea por la tienda:

- Buscar y comprar videos  
Para realizar esta operación el cliente selecciona los items a comprar y debe suministrar:
  - ID Cliente
  - Numero de tarjeta de crédito si es diferente a la ingresada en el momento de registrarse
  - Dirección de envío.
  
- Registrarse ante el sistema  
Este servicio se compone de dos sub-servicios. Es importante notar que cada uno de los dos sub-servicios que se detallan a continuación se ejecuta en invocaciones diferentes, pero ambos hacen parte de la misma transacción. Razón por la que estas invocaciones deben recordar un estado conversacional entre invocaciones. Esto último conlleva a que se implemente un componente con estado.  
Los sub-servicios que componen la operación *registrarse* son:
  - Registra Información Personal: Esta operación consiste en ingresar la siguiente información personal
    - Cedula o identificación.
    - Nombre
    - Apellido
    - Teléfono
    - Correo electrónico

Nota: Esta información debe ir en la tabla de clientes

- Registra Información Financiera: Esta operación consiste en ingresar la siguiente información financiera del cliente
  - Cédula o identificación del cliente.
  - ID tarjeta de crédito.
  - Tipo de tarjeta (visa/master).
  - Fecha de expiración de la tarjeta

Nota: Esta información debe ir en la tabla de historia financiera del cliente.

Para este servicio se debe implementar un componente con estado, que permita la ejecución de este servicio funcional.

- Ver historial de compras

Esta funcionalidad está pensada para que el cliente pueda consultar su historial de compras por fechas, donde el resultado de esta consulta debe ser el detalle de su anterior compra.

El segundo de ellos (Administrador), debe:

- Consultar los clientes de la tienda.
- Modificar el inventario (Adicionar, modificar, consultar, remover)
- Reporte de Ventas realizadas por intervalo de fechas.

Para este actor se debe diseñar un componente sin estado que ofrezca los anteriores servicios, realizados e invocados por el administrador.

El tercero (Entidad Bancaria) debe:

- Validar en línea si la TC es válida o no.
- Validar Saldo al momento de la compra para evitar sobregiros.
- Efectuar Pago.

Para este actor se debe diseñar un componente independiente que ofrezca los anteriores servicios, invocados por el sistema al momento de la compra de videos por parte de un cliente.

## Descripción Entregable – Individual o en Parejas

Se desea que el arquitecto de software (Estudiante/s) diseñe la solución que satisfaga los requerimientos de negocio, por lo cual se requiere construir un documento de diseño que contemple los siguientes entregables:

1. Descripción de actores
  - a. Actores
  - b. Roles
  - c. Objetivos
2. Requerimientos
  - a. Requerimientos funcionales.
  - b. Requerimientos no funcionales.
3. Descripción de casos de uso
4. Diagrama de casos de uso
5. Diagrama de secuencia de los casos de uso arquitecturalmente relevantes.
6. Diagrama de clases.
7. Diagrama de dominio.
8. Diagrama de BPMN de las funcionalidad registrarse en el sistema (Cliente).
9. Diagrama de despliegue.