

NUEVAS TECNOLOGÍAS DE DESARROLLO



Ingeniería de Sistemas - Julio 2017

Hola!

Soy Cindy Hernández,

Ingeniera de Sistemas – UPTC

Master en Ingeniería del Software de la
Universidad de los Andes

Maestrante en Tecnologías de la
información para el negocio – Uniandes





VAMOS A CONOCERNOS

- Nombres
- ¿Cuál semestre cursa?
- ¿Qué es lo que más le gusta de formarse como ingeniero de sistemas?



OBJETIVO DEL CURSO

Comprender algunas de las tecnologías de desarrollo web de los últimos tiempos, conocer sus funcionalidades y limitaciones para plantear y desarrollar proyectos innovadores en el campo de las aplicaciones web. ***Se espera que el estudiante domine las aplicaciones Web.***



PROGRAMA

Semana 1

Patrón MVC

Semana 2

Servidores

Semana 3

Modelo - JPA
BD Relacionales y No
Relacionales

Semana 4 a 6

Controlador
Servicios REST, EJB,
Patrones (DAO, DTO)

Semana 7 a 11

Vista - Tecnología Web
HTML, CSS, JavaScript
(Jquery - Angular)

Semana 12 a 13

Python/Django

Semana 14

NodeJs



REGLAS DE JUEGO Y CURSO DE NUEVAS TECNOLOGÍAS DE DESARROLLO



HORA DE INICIO → 6:45 a.m.



Máximo 15 minutos de tardanza los que no cumplan tendrán puntos negativos.



Curso teórico - práctico

Lo más importante no es ser el más duro, lo realmente importante es esforzarse por aprender y realizar entregas de calidad.

Curso 3 créditos → 9 horas de trabajo semanal



3 horas de clase



6 horas de trabajo personal



REGLAS DE JUEGO Y CURSO DE NUEVAS TECNOLOGÍAS DE DESARROLLO

Respeto (hacia los demás y hacia mí)	Respeto (hacia usted mismo y otros)	Respeto (hacia el tema y hacia mí)
LLeque a tiempo.	Levante la mano; haga preguntas	Haga preguntas propias.
No conteste llamadas en clase.	No sea tímido; recuerde que no existen preguntas estúpidas!	Trate de entender “el qué”, “el cómo”, “el porqué” y “el para qué”.
Celulares en silencio.		Los temas difíciles no son aburridos, los temas aburridos no son difíciles.



Se cobran 2 puntos por cada celular que suene en clase.

Esté preparado para corregirme (educadamente).



EVALUACIÓN

CORTE 1 - 40%

Quices y talleres → 15%
Proyecto - Entrega 1 → 15%
Tareas y actividades de clase → 10%

CORTE 2 - 60%

Proyecto - Entrega 2 → 15%
Proyecto - Entrega 3 → 25%
Quices y talleres → 10%
Tareas y actividades de clase → 10%

MANEJO DE PUNTOS

Usados para talleres, tareas y actividades de clase → máximo de puntos en cada aspecto = 100



PROYECTO

APLICACIÓN WEB → Tecnología Java en la parte de Backend y Tecnología Web en la capa de FrontEnd con presencia de una capa intermedia de servicios para comunicar ambos mundos.

Grupos = 4 personas



PROYECTO - Enunciado 1



MarketPlace

El marketplace KoLo es conocido como líder y pionero del país por generar transacciones de comercio online de productos para el hogar, tecnología y artefactos de deportes.

Gestión de Clientes

Carrito de Compras

Proceso de pago para formalizar una compra

Gestión de Productos



PROYECTO - Enunciado 2



Sistema de Reservas

KonLor es un sistema encargado de llevar el control de reservas de una prestigiosa cadena hotelera.

Gestión de Usuarios

Disponibilidad de Hoteles

Proceso de pago para formalizar una reserva

Gestión de Hoteles

ESTILOS Y PATRONES ARQUITECTÓNICOS

Conceptualización



ESTILOS ARQUITECTÓNICOS

Atributos de calidad → **Propiedades** o **cualidades** de calidad que el sistema debe cumplir.

Interoperabilidad

Usabilidad

Escalabilidad

Requerimientos arquitecturalmente significativos → Identificados principalmente en reuniones con los **stakeholders y documentación del negocio.**

Identificados y claros

insumo

Arquitecto → diseña la visión general de arquitectura.



ESTILOS ARQUITECTÓNICOS



¿Por dónde comenzar?

¿Cómo favorecer los
atributos de calidad
importantes?

¿Alguién habrá hecho
algo similar?

ESTILOS VS PATRON

ESTILO DE ARQUITECTURA [Fair2010]	PATRÓN DE ARQUITECTURA [Bass2013]
Es un tipo de patrón que ocurre a nivel de la arquitectura de software.	Es un paquete de decisiones de diseño que se encuentra repetidamente en la práctica.
Es aplicable a elementos de arquitectura como componentes, conectores y módulos.	Tiene un conjunto de propiedades conocidas que permiten su reutilización.
Define un lenguaje compuesto de elementos de arquitectura y restricciones de diseño.	Describe una clase de arquitectura; en general no se inventa, se descubre durante la práctica.

ESTILOS ARQUITECTURA

FAMILIA MÓDULO

Capas

El software requiere ser dividido para poder ser desarrollado por separado. Poca interacción entre las partes para favorecer portabilidad, modificación y reuso.

FAMILIA COMPONENTE - CONECTOR

Broker

Cómo construir software
cuyos componentes estarán
distribuidos.

Se desea soportar interoperabilidad
de componentes heterogéneos y
distribuidos.

Orientado a servicios

FAMILIA ASIGNACIÓN

Muchas aplicaciones con grandes
conjuntos de datos, es necesario
ordenarla y procesarla de forma
distribuida y paralela.

A TENER EN CUENTA

- Los patrones y/o estilos establecen la forma general que tendrá la arquitectura.
- Ésta forma general tiene un impacto significativo en los atributos de calidad que se verán favorecidos y los que no.
- El arquitecto debe conocer diferentes estilos para aplicarlos rápidamente a problemas comunes.
- Los estilos ofrecen un vocabulario unificado y facilitan la interacción entre arquitectos.

MVC

MVC

- Patrón que pertenece a la familia componente-conector de los estilos arquitectónicos.
- Separa la lógica de negocio de la interfaz de usuario.
- Facilita la evolución por separado de ambos aspectos.
- Incrementa reutilización y flexibilidad.

Descrito por
primera vez en
1979 para
Smalltalk.

M V C

MODELO

- Información
- Base de datos
- Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

VISTA

- Representación visual
- Elementos que el usuario recibe y con los que puede interactuar.

CONTROLADOR

- Intermediario entre las capas Modelo y Vista
- Responsable de actualizar el Modelo cuando el usuario manipula la Vista.

MVC - Ejemplo



Mesero y cocinero esperan un evento del **cliente**.

El cliente solicita un plato y bebida específica → **el mesero** captura la orden y solicita **al cocinero** prepararla.

Cliente y mesero esperan una respuesta del **cocinero**.

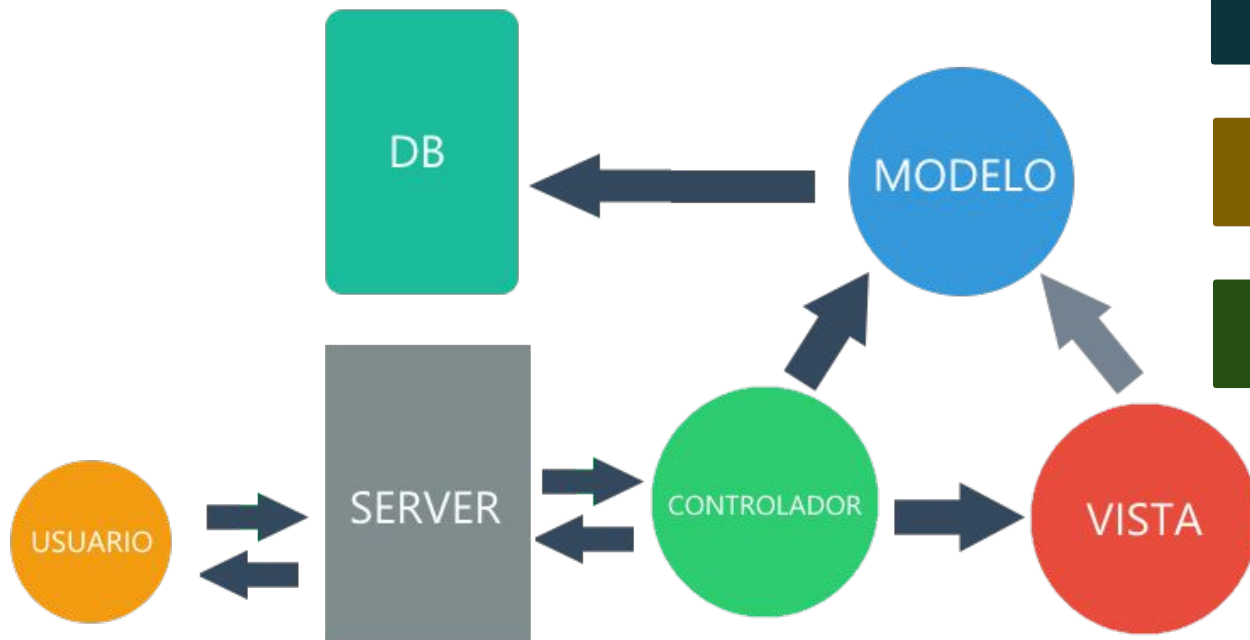
MVC - Ejemplo



El **cocinero** interpreta y prepara lo solicitado por el **mesero**.

El **cocinero** devuelve al mesero el plato solicitado (datos) → el **mesero** hace entrega del plato al cliente

MVC



SEPARACIÓN DE RESPONSABILIDADES

CÓDIGO INDEPENDIENTE Y DESACOPLADO

REUTILIZACIÓN DE CÓDIGO



EJERCICIO - 10 Puntos

Algoritmo de Luhn → usado para validar las tarjetas de crédito

Dado un número entero positivo de hasta 16 dígitos, devuelve true si es un número de tarjeta de crédito válido y false si no lo es.

1. Si hay un número par de dígitos, doble cada otro dígito comenzando con el primero, y si hay un número impar de dígitos, doble todos los demás dígitos comenzando con el segundo. Otra forma de pensar en ello es, de derecha a izquierda, doblar cada dígito desde el segundo hasta el último dígito.

EJEMPLO

→ **1714** => [1*, 7, 1*, 4] => [2, 7, 2, 4]

12345 => [1, 2*, 3, 4*, 5] => [1, 4, 3, 8, 5]

891 => [8, 9*, 1] => [8, 18, 1]



EJERCICIO - 10 Puntos

2. Si un número duplicado resultante es mayor que 9, reemplázelo con la suma de sus propios dígitos, o 9 se resta de él.



EJEMPLO

$$[8, 18^*, 1] \Rightarrow [8, (1+8), 1] \Rightarrow [8, 9, 1]$$

0

$$[8, 18^*, 1] \Rightarrow [8, (18-9), 1] \Rightarrow [8, 9, 1]$$

3. Se suman todos los dígitos



$$[8, 9, 1] \Rightarrow 8+9+1 \Rightarrow 18$$

4. Finalmente, el total de la suma se divide en 10. Si el residuo es igual a cero, el número de la tarjeta de crédito es válido, de lo contrario no lo es.



$$18 \% 10 \Rightarrow 8.$$

8 no es igual a 0, entonces 891 no es un número de tarjeta de crédito válido

BIBLIOGRAFIA

- [Bass2013] Bass, L. Clements, P., Kazman, R., “Software Architecture in Practice”, Addison-Wesley, Third Edition, 2013.
- [Fair2010] Fairbanks, G. Just Enough Software Architecture: A Risk-Driven Approach. 2010.