

## Orientacion a Objetos 2

Dra Alejandra Garrido

Dr. Alejandro Fernandez

Dr. Gustavo Rossi



[garrido, gustavo, alejandro.fernandez]@lifia.info.unlp.edu.ar



Laboratorio de Investigación y Formación en Informática Avanzada  
Facultad de Informática, Universidad Nacional de La Plata  
Calle 50 esq. 115 - Primer piso (1900) La Plata, Argentina  
TE: (0221) 422 8252 <http://www-lifia.info.unlp.edu.ar>



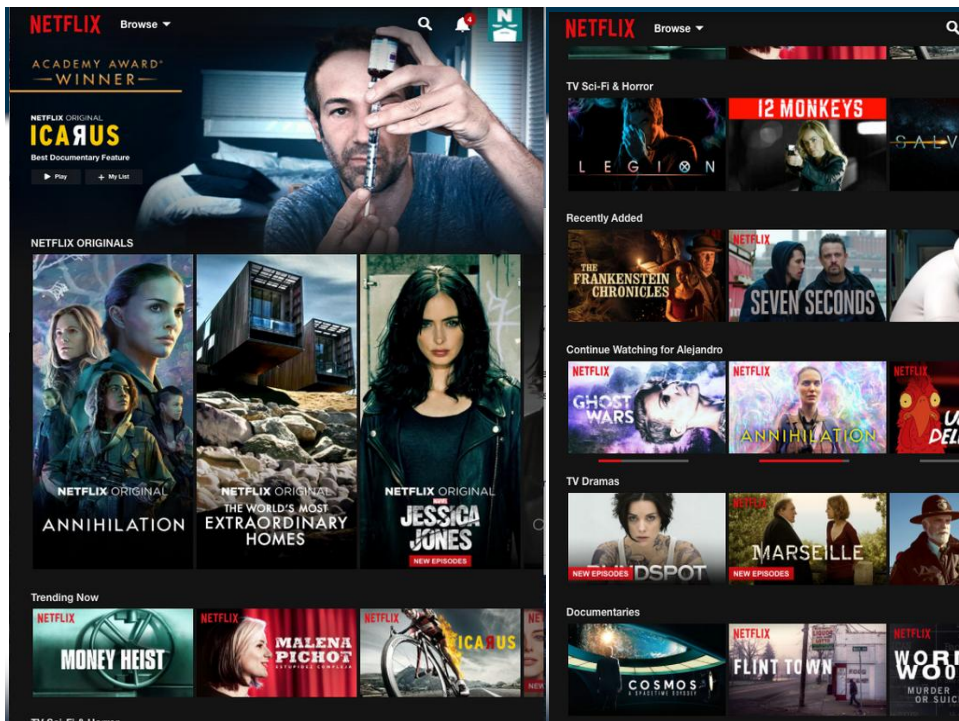
### Práctica

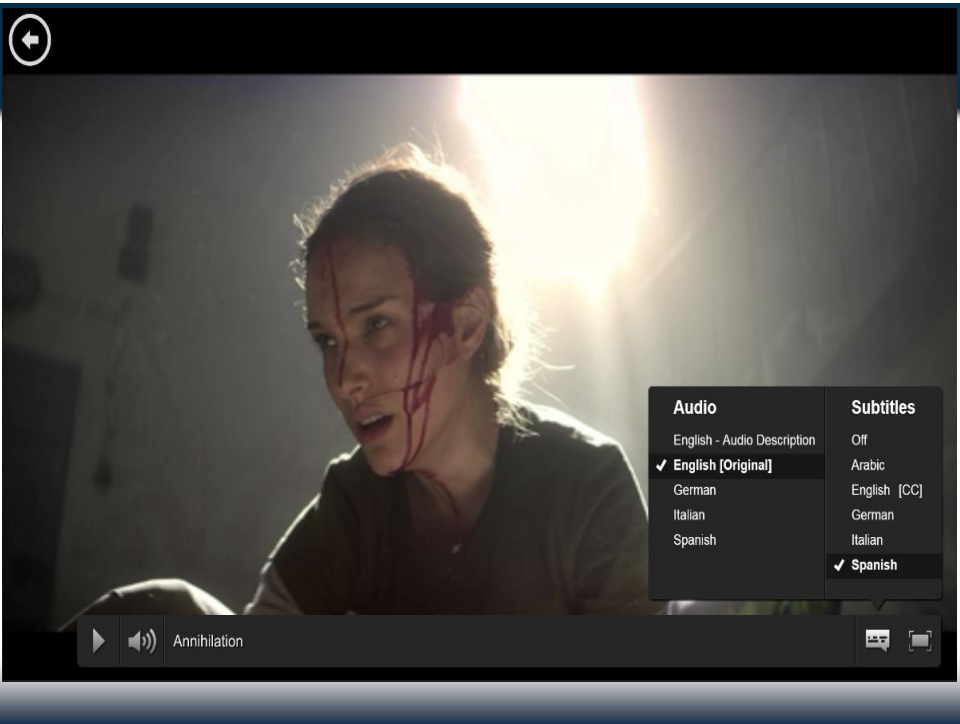
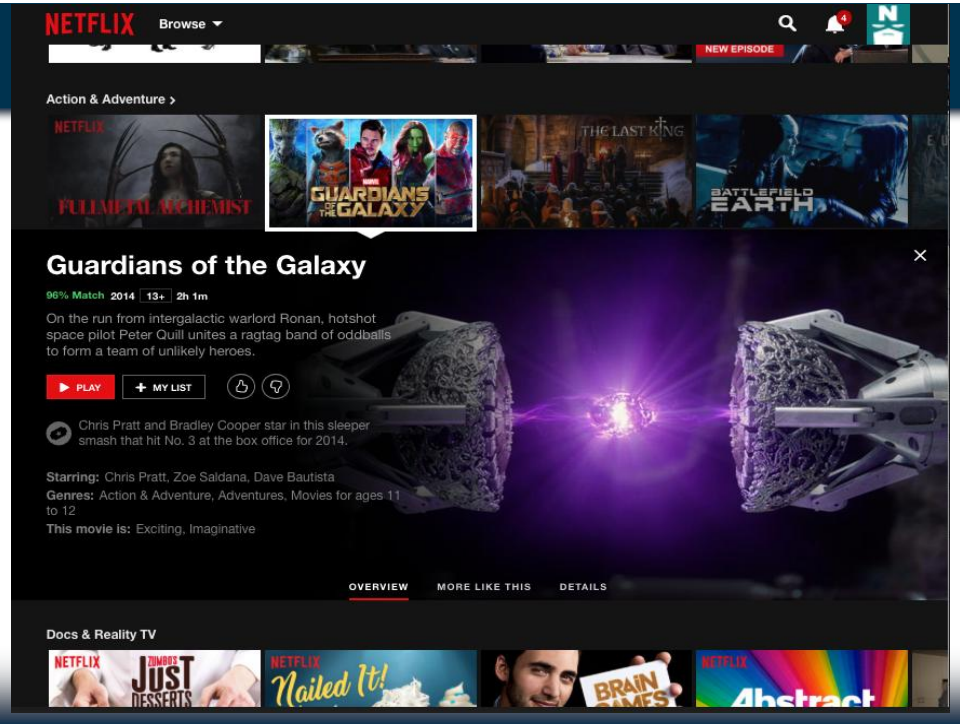
- ✓ Todos los horarios y aulas están publicados en la cartelera
- ✓ La practica empieza esta semana (se puede ir a cualquier horario)
- ✓ Llenar encuesta que esta en Catedras

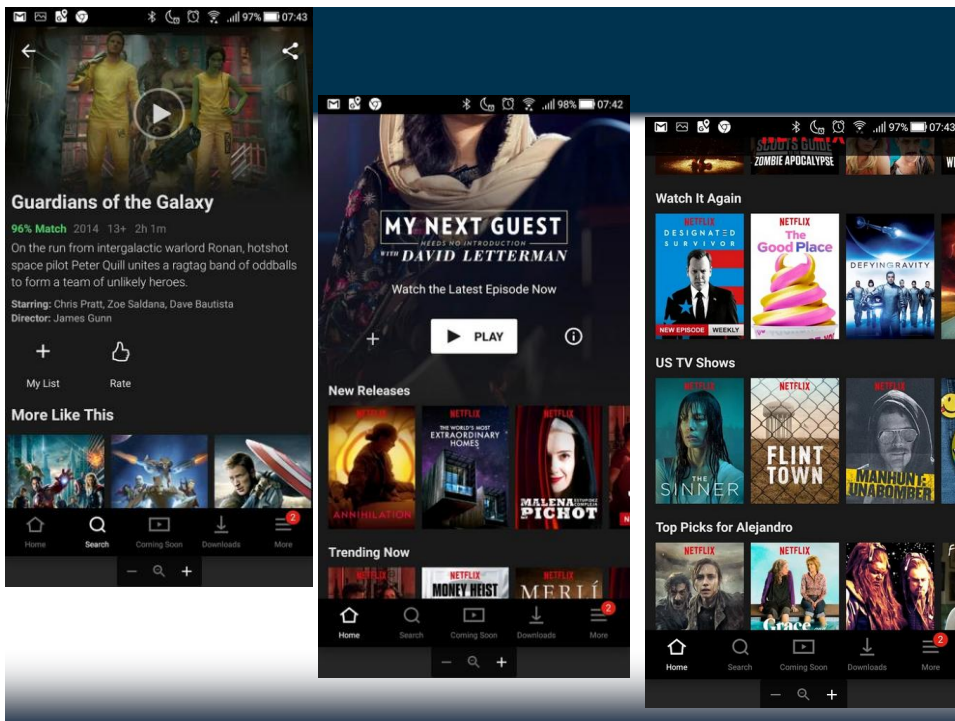


## “Netflix” gratuito

- ✓ Tenemos que desarrollar un software tipo Netflix pero libre y gratuito
- ✓ Se nutre de videos de youtube, vimeo y otros
- ✓ Tiene que poder competir con Netflix







## Problemas

- ✓ Empezamos de cero?
- ✓ Como programamos el reproductor?
- ✓ Como recomendamos? Las formas de recomendar dependen de que?
- ✓ Como reproducimos?
- ✓ "Almacenamos" los videos? Almacenamos algo? Donde?
- ✓ Queremos reproducir en smart tv...como? Otro soft?
- ✓ Como evolucionamos (e.g. Netflix que empezo mandando DVDs por correo como un videoclub!!!)



## Problemas en el diseño con objetos

- ✓ Encontrar las clases adecuadas
- ✓ Identificar correctamente las responsabilidades (metodos)
- ✓ Diseñar para el cambio (incluso para el no previsto)
- ✓ Buscar diseños modulares
- ✓ Reusar código o diseño existente



## Ejemplo

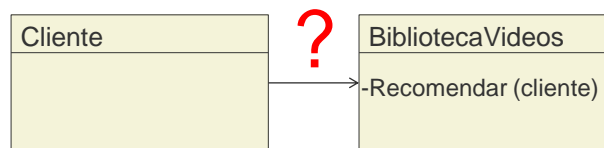
- ✓ Recomendaciones de Netflix
- ✓ Luego de un extenso análisis (y comprensión del dominio) llegamos a un diseño que incluye esta clase

BibliotecaVideos
-Recomendar (cliente)



## Evolución de la aplicación

- ✓ No hay una sola forma de recomendar
- ✓ El algoritmo de recomendación puede que dependa del “perfil” del cliente, o del momento del año, o de la publicidad o....



## Solucion?

- ✓ Hacer diseños mas sensibles a la evolución
- ✓ Como me doy cuenta que necesito en este caso particular?
- ✓ Con mas experiencia... mia, o de otros

Patrones de Diseño



## Las aplicaciones son interactivas

- ✓ En la Web en un smartphone, tableta etc, lo esencial es la interaccion
- ✓ Como construimos interfaces interactivas? Siguiendo que principios?
- ✓ Como hacemos para que la funcionalidad no dependa de la caracteristica del dispositivo? (tablet, Web, etc)

Nociones de construccion de aplicaciones interactivas



## Como estructuramos sistemas "grandes"?

- ✓ Los sistemas rara vez son "pequeños" programas ejecutandose en un equipo
- ✓ Habitualmente la funcionalidad esta distribuida (como en una app Web o mobile)
- ✓ Como organizamos las componentes de tales sistemas?
- ✓ Como "consumimos" informacion o funcionalidad de otros sistemas?

Patrones Arquitecturales (\*)



## Como desarrollamos?

- ✓ Cuanto tiempo nos puede llevar diseñar un sistema ultra-flexible?
- ✓ Es aceptable para los clientes?
- ✓ Hay alternativas?

## Metodos Agiles



## Somos Agiles...y entonces?

- ✓ Los metodos agiles suelen buscar resultados “rapidamente”, para chequear requerimientos con los clientes
- ✓ Pero los diseños entonces quedan “feos”

## Refactoring





## Que pasa con la “correctitud” del sistema

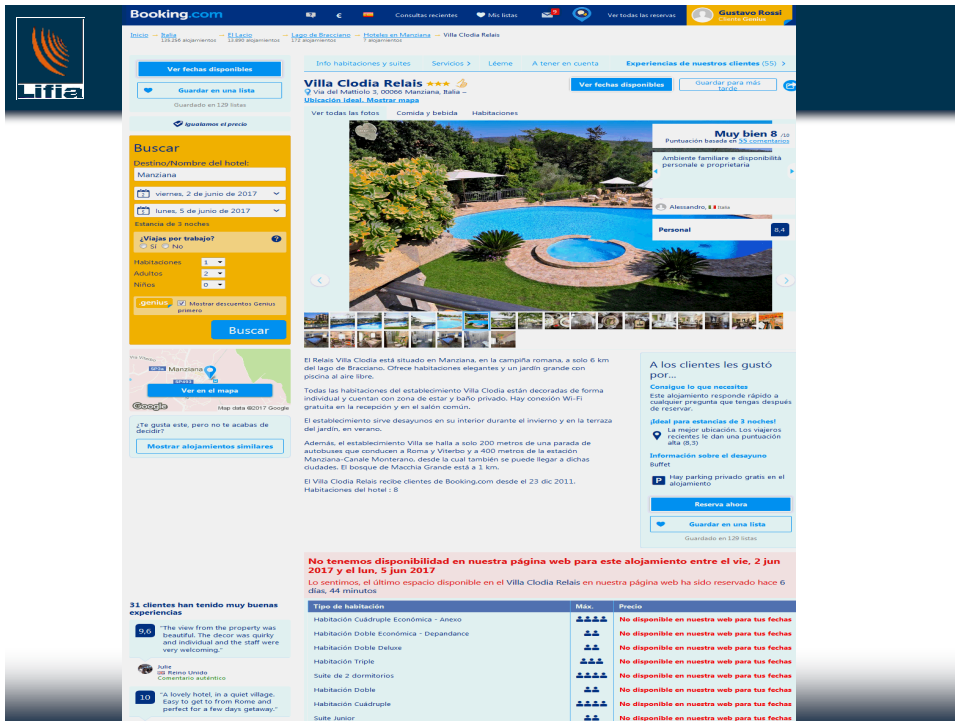
- ✓ Somos agiles y ademas...
- ✓ Nuestros diseños son modulares (soportan el cambio)
- ✓ Pero como sabemos que el sistema funciona? Esperamos hasta...cuando?

## Test Driven Development (TDD)



## Interaccion, Interfaz

- ✓ Nuestros sistemas son usables?
- ✓ Un usuario “promedio” consigue llevar a cabo las tareas que el sistema soporta? (e.g. una transferencia en home banking, una compra en un e-commerce, un remate en e-bay)



## Interfaz, interacción...

- ✓ Que guías tenemos para organizar la interfaz, la interacción?
- ✓ Como nos aseguramos que el “diseño” de la interfaz sea usable?

## Patrones de Interfaz/interacción



## Interfaces, Persistencia...

- ✓ Queremos que nuestro sistema muestre información en distintas formas (barras, estadísticas, mapas....)
- ✓ Y queremos guardar los datos en diferentes formatos....
- ✓ Pero todo esto no tiene que ver con nuestro problema....
- ✓ Como nos “conectamos” con otras clases

## Frameworks