Proyecto curso

“FORMACIÓN INICIAL JAVA/DIGITAL”
MovieFLix
[Creación de la gestión de un sistema de vídeo streaming]
[Lucatic]

26/07/2019 (Ed.19)

ÍNDICE DE CONTENIDOS

Planteamiento

Una empresa quiere disponer de una plataforma para **gestionar y administrar** las películas a visionar vía streaming

La vida está muy "achuchada" y la empresa ha contactado con un equipo de trabajo recién salido de un curso. Prometen ser mano de obra barata… pero mu responsable y que quieren aprender... buena gente bro… tíos y tías preparaos'... orgullosos de ello, que rentan mazo, con tolerancia a la frustración, que trabajan en equipo… gente tecnológica, que se come los problemas... que los resuelven... que analizan... que buscan soluciones... que son como perros de presa, que pim-pam-pim-pam que tum pam pam que tumpam quetepetepete… que se adaptan... que son máquinas... son bestias pardas... son shurmanos… son mu buenos… son mu chetos… son OP y lo sabes.

Grupos de Trabajo

Para trabajar se realizarán **cuatro grupos**.

Grupo A	Grupo B	Grupo C	Grupo D
Jose Miguel	Daniel A.	Jorge	Daniel D.
Asiel	Juan Alfonso	Jesús	Ignacio
Javier	Mohibul	Joaquín	Alex
Vladimir	Adrian	Bryan	Erick

Roles de trabajo

Los propios de la metodología SCRUM

- **SCRUM Master**
  - Debe indicarse el primer día
- **Equipo de desarrollo**
- **Product Owner**:
  - Figura realizada por el profesor en última instancia para decisiones finales
  - Desarrollada por el equipo para crear el Product Backlog

Idea de referencia

MovieFlix dispondrá de **catálogos de películas** para streaming. En el portal se comercializan **lotes de películas**. Cada **categoría/catalogo** (policiaca, romántica, aventuras, comedia, animación y thriller) constituye un lote de pelis. Los socios pagan una cuota mensual por cada catálogo (pueden tener 1, 2, …). Durante ese mes pueden ver las películas del catálogo.

- **Básico**: sólo [illegible] a un catálogo
- **Extra**: tiene [illegible] catálogos
- **Premium**: acce[illegible]os los catálogos

Película	Cliente
•Nombre •AñoEstreno •Categoría	•Nombre completo •Fecha Nacimiento •Ciudad residencia •Paquete al que se abona

Objetivos marcados

El **objetivo principal** de este proyecto se centra en tres aspectos:

- Conseguir que el grupo de alumnos sepa **utilizar** y mezclar la **teoría** y las **técnicas** vistas durante el curso relacionadas con Java, BBDD y Pruebas Unitarias.
- Ayudar a que el alumno **desarrolle** y **potencie** las **competencias** marcadas o previstas para este curso:

Análisis y resolución de problemas	Cuenta con mi hacha	Autonomía
Trabajo en equipo	Comunicación	~~Thug Life~~
Flexibilidad	Proactividad	Motivación
Tolerancia a la frustración	Iniciativa	Interés a tope, lo voy a petar
	Soy tu BAE	Capacidad de aprendizaje
	Responsabilidad	

- Aprender y entender cómo funciona una **metodología de trabajo** como **SCRUM**

Entregas a realizar a lo largo del proyecto

a) Cada uno de los sprints marcados en la metodología de Sprint
b) El sprint final viene marcado por la entrega del proyecto.
c) A partir de ese día no se seguirá trabajando en el proyecto.
d) La presentación se realizará junto al siguiente proyecto.

Descripción del proyecto (Tareas de Programación)

El grueso del proyecto **se centrará de forma exclusiva en la zona de administración** y dispondrá de las siguientes tareas:

a. Gestión de categorías

b. Inclusión de nuevas películas

c. Gestión de usuarios

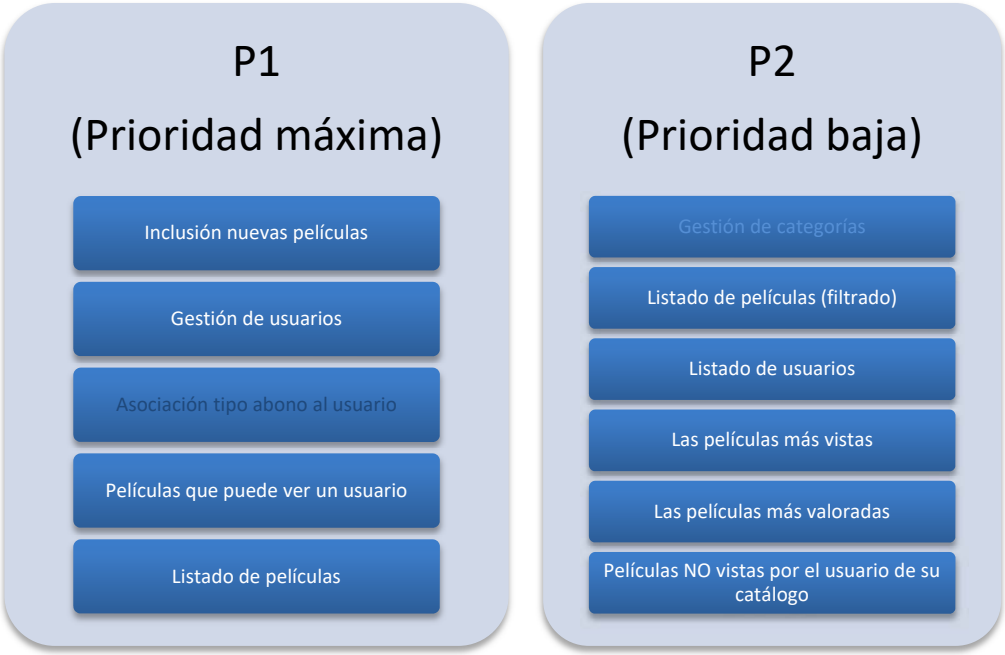
d. Asociación de tipo de abono (Básico, Extra, Premium) para el usuario

e. Informes

1. Listado de películas
2. Listado de usuarios
3. Listado de películas (filtrado por categoría)
4. Las películas más vistas
5. Dado un usuario, indicar las películas que puede ver
6. Dado un usuario, indicar las películas NO vistas (de su catálogo)
7. Las películas más valoradas

ERROR: JDWP Unable to get JNI 1.2 environment, jvm->GetEnv() return code = -2
JDWP exit error AGENT_ERROR_NO_JNI_ENV(183): [util.c:840]

Las tareas a realizar pueden dividirse en dos grupos dependiendo de su prioridad.



Para realizar la carga inicial de las películas se leerán datos de un fichero que contendrá, en formato texto, al menos 25 títulos de películas.

Trabajo a realizar a lo largo del proyecto

Cada grupo de alumnos/as debe **desarrollar** y **programar** las distintas partes marcadas en la "Descripción del Proyecto".

Esas tareas pueden ser de dos tipos:

- **Tareas de Programación**: definen las acciones que debe realizar el grupo a lo largo del proyecto para crear la aplicación.
- **Tareas Transversales**: definen las acciones no relacionadas de forma directa con la programación pero que permiten la mejora competencial o la calidad del trabajo realizado.

El equipo debe **usar la metodología SCRUM** y crear un backlog de productos (revisable y ampliable durante el proyecto), un backlog de tareas por cada sprint y los cuatro tipos de reuniones (reuniones diarias, planificación de sprint, revisión y presentación de producto).

La pauta básica para todo el proyecto es la misma:

Ante la duda… simplifica

Descripción del proyecto Web (Tareas transversales)

Hay una serie de tareas a pedir a lo largo del proyecto (las 6 son de Prioridad Máxima)

Salvo la herramienta de gestión usada por el Scrum Master, el resto de tareas las debe realizar TODO el equipo

- **Documentación Colaborativa**: Generar una documentación "formal/informal"
  - Posibles elementos: tareas realizadas, explicación breve de procesos y/o arquitectura, trabajo Sprint realizado, pantallazos de product backlog, etc.
- **Documentación JavaDoc**: Generada a partir de los scripts. Puede incluirse en la documentación colaborativa. Al menos una por integrante
- **Uso de repositorios**: para compartir las versiones de código.

  Ideas: GitHub, GitLab, Bitbucket.

- **Herramientas de Seguimiento de Proyectos y Gestión**: que permiten gestionar proyectos Scrum, tiempos, etc. (o cualquier acción que se considere necesaria).
  - Como mínimo se usará Jira y Slack.
- **Pruebas de calidad**: El código de un método/clase estará completo cuando pase estas pruebas.

Para Clases

- Todos los paquetes van en minúsculas
- No se puede usar en el import .*
- En el import no pueden existir clases no utilizadas
- El código debe estar correctamente formateado
- Todo el código tiene que tener una cabecera en Javadoc que incluya: nombre de la clase, descripción, fecha, versión y autor.
- Toda clase (model) debe incluir setter/getter, constructor (default) y toString

Para métodos

- Las llaves de inicio del método están en la misma línea que la cabecera
- Todos los nombres de variables son descriptivos
- Usar sistema de logging para la información del programador

- **Pruebas unitarias**: define, de forma previa, varias pruebas unitarias.
  - Tienes que usarlas en todos los sprint. Pueden ser sencillas.
  - Cada miembro del equipo debe hacer una prueba unitaria al menos.
  - Utiliza JUnit.

Normas a cumplir

Hay una serie de normas e ideas que debes implementar en tu proyecto

- Uso de **interfaces**
- Uso de capas de **Servicios**, **Controladores** y **Datos** (o parecidas)
- Uso de **Excepciones**
- Dividir en distintos **paquetes**
- Utilizar el **patrón DAO** para la BBDD
- Cada "**pantalla gráfica**" debe ser un método que muestre esa información
- Utilizar una clase externa (y static) para leer datos

¿Cómo se valorará el proyecto?

Para valorar el proyecto se tendrán en cuenta tres aspectos

a. **Valoración técnica**: se revisará la calidad de la solución, elementos empleados, técnicas utilizadas, empleo de tecnologías, aplicación de los conocimientos vistos en clase

b. **Materiales aportados**: tanto el código como la documentación si se ha elaborado, valorando la calidad de la misma, la aportación al proyecto, la implicación del equipo en la misma, etc.

c. **Valoración competencial**: tomando como referencia algunas ideas basadas en competencias profesionales como el trabajo en equipo, análisis y resolución de problemas, flexibilidad, etc.

d. **Directrices SCRUM**: haber seguido los parámetros indicados por la tecnología, seguir las necesidades del cliente, etc.