

# **Relación de ejercicios 3**

## **TEMA 3. Métodos de desarrollo ágiles**

1. Las metodologías ágiles están más orientadas a proyectos pequeños de corta duración con entregas frecuentes, donde el cliente es parte del equipo de desarrollo, se esperan cambios durante el proyecto por lo que la arquitectura la van definiendo y mejorando en base a ello, y además se hace énfasis en los aspectos humanos, ya sean individuales o del trabajo en equipo.

Las metodologías tradicionales se pueden utilizar en proyectos de cualquier tamaño, pero son más efectivos en proyectos grandes con equipos dispersos. Tienen un contrato prefijado, donde el cliente interactúa con el equipo de desarrollo mediante reuniones, promoviendo que la arquitectura se defina de manera temprana, esperando con esto que no ocurran grandes cambios durante el proyecto.

2.
  - a. SCRUM.
  - b. Standup diario, Retrospectivas, Planificación en Sprints/Iteraciones, Revisión de Sprints/Iteraciones, Iteraciones cortas.
  - c. Se realiza poco diseño del software, con frecuencia se utilizan para sacar dinero al cliente a través de la falta de definición de un entregable, solo funciona con desarrolladores experimentados, requiere demasiado cambio cultural para adaptarlo, puede ser muy ineficiente, requiere de encuentros a intervalos frecuentes con un enorme coste para los clientes.
  - d. Procesos y prácticas inconsistentes entre equipos  
Cultura organizacional reñida con valores ágiles  
Resistencia de la organización general al cambio  
Falta de habilidades / experiencia con métodos ágiles  
No hay suficiente participación de liderazgo  
Apoyo administrativo y patrocinio inadecuados

## **SEMINARIO 3. Historias de Usuario**

3. Los casos de uso son mucho más específicos a la hora de modelar la interacción entre un actor y el sistema, con un texto muy detallado, asociado a diagramas, siendo de difícil comprensión incluso para personal técnico. Los escenarios y las historias de usuario son mucho más sencillas que los casos de uso, sin embargo, los escenarios explican una situación en la que el usuario interactúa con el sistema utilizando varias funciones que necesite, pero sin entrar en mucho detalle. Las historias de usuario van un paso más a la hora de simplificar esto. Consisten en explicar una funcionalidad del sistema en una o dos frases, muy fáciles de leer y comprender y que se pueden utilizar para planificar a la hora de realizar el desarrollo.

4. Se utilizan para añadir alguna que otra característica a la HU, y son las pruebas que confirman que la HU ha sido correctamente desarrollada. Expresando con ello muchos detalles obtenidos de la conversación con el cliente, los desarrolladores lo usan para mejorar el proceso sin necesidad de codificar nada.

5.

- a. Como administrador quiero añadir películas

**Pruebas de Aceptación**

- Añadir una película con caracteres especiales en el título
- Añadir una película que ya esté en el sistema y de error diciendo que ya existe
- Añadir una película sin duración y que de error diciendo que necesita una duración mayor que 0 segundos.

**Tareas**

- Crear la tabla de películas en la base de datos
- Interfaz de usuario para añadir películas
- Interacción del usuario con la interfaz y llamada a las operaciones sobre la base de datos con la información introducida.

- b. Como cliente quiero consultar películas

**Pruebas de Aceptación**

- Buscar cualquier película muestra todas las coincidencias posibles
- Consultar una película, muestra toda su información.
- Buscar un título que no exista y dirá no se encuentran resultados para su búsqueda

**Tareas**

- Interfaz de la barra de búsqueda
- Interfaz de los datos de una película de cara al usuario
- Operaciones SQL para obtener todos los datos necesarios.

- c. Como cliente quiero comprar una entrada a una película

**Pruebas de Aceptación**

- No permita seleccionar horarios no disponibles
- No permita seleccionar asientos no disponibles
- Si la tarjeta no tiene saldo comunicarlo y no vender la entrada

**Tareas**

- Interfaz de horarios
- Interfaz de asientos disponibles

- Pasarela de pago
- Operaciones SQL necesarias

d. Como cliente quiero filtrar películas por horario

**Pruebas de Aceptación**

- Poner un horario que salgan todas las películas de ese horario
- Poner un tramo horario y que salgan todas las películas posibles en ese tramo horario

**Tareas**

- Crear las operaciones SQL correspondientes
- Crear la función filtrar en la interfaz
- Interacción con la función de filtrado por horario

e. Como usuario quiero devolver entradas

**Pruebas de Aceptación**

- Devolver una entrada cuando la película ya ha comenzado, dar error y decir devolución no posible
- Devolver una entrada antes de que comience la película, devolver el dinero y confirmar acción
- Devolver una entrada que no existe, dar error

**Tareas**

- Operaciones SQL correspondientes
- Interfaz de devolución
- Conexiones con las entidades financieras para devolver pago

f. Como administrador quiero quitar una película del catálogo

g. Como usuario quiero poder reservar dos o más entradas de la misma película

h. Como administrador quiero poder modificar los datos de una película

i. Como usuario quiero ver las películas de estreno

j. Como usuario quiero filtrar películas para encontrar las que se emiten en 3D

6. Por que al final, los usuarios de la aplicación son los que la van a usar y no los desarrolladores, entonces es mejor determinar las funciones y objetivos desde su punto de vista, para conseguir un producto final con un valor que cumpla todas las necesidades que estos van a tener.