

Actividades de la Unidad 1

Francisco Javier Sueza Rodríguez

5 de diciembre de 2022

Centro: IES Aguadulce
Ciclo Formativo: Desarrollo Aplicaciones Web (Distancia)
Asignatura: Entornos de Desarrollo
Tema: Tema 1 - Desarrollo de Software y Entornos de Desarrollo

1. Caso Práctico

Nos piden desarrollar una aplicación web que permita visualizar películas a través del navegador.

Por lo tanto el cliente quiere que el software que le vamos a desarrollar esté optimizado para cualquier navegador, sobre todo Edge, Firefox, Chrome y Safari. Necesitan que la web sea responsive para que se adapte a todo tipo de dispositivos móviles y de escritorio.

Los usuarios pagarán una suscripción que podrá ser semanal, mensual, anual o por día, por lo que el sistema registrará la información de los usuarios y el tipo de suscripción que tienen activa. Por lo tanto, al tratar datos de carácter personal, el sistema debe cumplir las disposiciones recogidas en la Ley Orgánica de Datos Personales. La aplicación debe proporcionar las herramientas necesarias para que los usuarios paguen (por lo que debe usar una pasarela segura para la realización de los pagos).

El sistema debe permitir a los usuarios buscar y consultar la información sobre las películas. El resultado de la búsqueda debe aparecer en pantalla rápido sin demorarse más de 2 segundos. Debe almacenar información sobre las películas que se pueden visualizar. Se tendrán en cuenta por tanto las leyes referentes a los derechos de autor. El sistema debe permitir a un usuario ver todas las películas que quiera durante el periodo de tiempo que tenga contratado en su suscripción.

Nuestro cliente no sabe si añadirá algo más adelante porque es la primera vez que va a trabajar con un sistema así, pero lo que sí tiene claro es que quiere ir viendo como va quedando la web, por lo que le proponemos ir mostrándole el proyecto conforme se vaya desarrollando cada una de las funcionalidades de la aplicación y está conforme con ello.

2. Solución

Teniendo en cuenta el caso práctico definido en el punto anterior, responde a las siguientes preguntas:

1. **Indica qué tipo de lenguaje se tendría que utilizar en este caso concreto atendiendo al proceso de traducción del código.**

Respuesta:

En este caso, sería necesario un **lenguaje de programación interpretado**, ya que es el tipo de lenguaje que comprenden los navegadores web. En este caso, ese lenguaje debería ser **Javascript**, para la parte de Frontend, y podríamos usarlo también para la parte de Backend, junto las librerías Node y Express.

2. **¿Qué tipo de código intervendría en este desarrollo? Justifica la respuesta.**

Respuesta:

El código que podríamos encontrar en la aplicación sería **código fuente** y **código ejecutable**. No podríamos encontrar código objeto ya que los lenguajes interpretados no lo generan, siendo la conversión entre código fuente y ejecutable directa y realizada línea por línea.

3. **¿Qué tipo de software deberíamos utilizar si necesitáramos separar el funcionamiento del ordenador de los componentes hardware instalados? Pon un ejemplo de este tipo de software (aunque no fuese el caso concreto del supuesto práctico descrito anteriormente).**

Respuesta:

El software que necesitaríamos usar es una **máquina virtual**, que son las aplicaciones que crean una capa de abstracción entre los componentes de la computadora y el software. Un ejemplo sería

un **entorno de ejecución**, como el de Java, que nos permite ejecutar una aplicación hecha en este lenguaje en cualquier sistema.

4. Indica el tipo de recursos hardware que necesitará nuestra aplicación durante su ejecución.⁴⁴

Respuesta:

Los recursos hardware que necesitaría usar nuestra aplicación son los siguientes:

- **CPU:** para procesar todas las instrucciones de nuestra aplicación.
- **Memoria RAM:** para ir almacenando las instrucciones que se tiene que ejecutar.
- **Interfaces de Entrada/Salida:** para permitir la introducción de datos y la lectura de los resultados.
- **Tarjeta Gráfica:** para representar la información en pantalla, tanto la interface de la aplicación como para reproducir las películas.
- **Tarjeta de Sonido:** especialmente para la reproducción de las películas.
- **Tarjetas de Red:** ya sea ethernet o wifi para realizar la conexión con la base de datos, la pasarela de pago,...
- **Almacenamiento interno:** para almacenar información sobre las películas, datos en la cache, etc...

5. Explica cómo sería el proceso de traducción de nuestro código fuente en este caso concreto.

Respuesta:

En nuestro caso el proceso de traducción sería el siguiente:

- **Codificación:** escribiríamos el código en un archivo de código fuente en el lenguaje seleccionado, en este caso, Javascript.
- **Interpretación:** el interprete de Javascript, incluido en todos los navegadores web, traducirá línea por línea el código fuente en código ejecutable.
- **Ejecución:** el código ejecutable sería ejecutado por el ordenador.

6. ¿En qué fase del ciclo de vida se realizarían las Beta Test?

Respuesta:

Las Beta Test se realizan en la **fase de explotación**, ya que este tipo de prueba necesita que el software este ya instalado y configurado en el sistema del cliente, además de que estas pruebas van enfocadas en probar la aplicación tal y como va a ser usada por los usuarios finales.

7. ¿Qué herramientas necesitaríamos usar nosotros como desarrolladores para desarrollar el proyecto solicitado?

Respuesta:

Las principales herramientas que necesitaremos serán las utilidades **CASE**, que nos facilitarán el proceso de desarrollo, haciéndolo más ágil y permitiendo el desarrollo software de más calidad, ya que favorecen la creación de aplicaciones fáciles de mantener y con componentes altamente reutilizables. Según la etapa del proceso de desarrollo, las herramientas que utilizaremos serán las siguientes:

- **U-CASE:** estas herramientas nos ayudarán en la fase de **planificación y análisis de requisitos**. La mayoría de estas herramientas se apoyan en la creación de **diagramas UML** que nos permiten tener una visión de los requisitos del sistema más gráfica y rápida de entender. Algunos ejemplos son **Jira** o **Openproj**.
- **M-CASE:** estas herramientas nos ayudarán durante la etapa de **análisis y diseño**. Estas herramientas también se basan en la **creación de diagramas UML**, como diagramas de clases, de casos de uso, de entidad/relación y otros muchos. Algunos ejemplos son **ArgoUML** y **Use Case Maker**.
- **L-CASE:** este tipo de herramientas nos facilitan el proceso de codificación en si mismo, agrupando diferentes funcionalidades bajo un mismo entorno. En este tipo de herramientas se incluye unas de la más importantes para un desarrollador, como son los **IDEs**. Estas herramientas incluyen muchas funcionalidades útiles durante el proceso de codificación como detección de errores, autocompletado, integración automática de librerías, etc... La elección del IDE estará un poco condicionada por las preferencias del desarrollador, aunque también por el lenguaje usado y algunos otros requisitos de la aplicación. Algunos de los más empleados son **VSCode**, **Netbeans** o **IntelliJ IDEA**.

También podríamos incluir los **sistemas de control de versiones**. Una herramienta imprescindible para el desarrollo de software facilita el desarrollo colaborativo, llevando un registro de todos los cambios y permitiendo volver a versiones anteriores, entre otras cosas. Además, permiten integrar aplicaciones para la realización de **pruebas de integración**, que se ejecutarán con cada cambio que se realice en el código. El software de control de versiones más empleado en la actualidad es **Git** y su plataformas online **Github** o **Gitlab**, pero también podemos encontrar otros como **CVS** o **Subversion**.

Además de estas herramientas, cabe destacar otras que aunque no necesariamente entran dentro de las categorías mencionadas deberemos usar para el desarrollo. Estas herramientas son:

- **Librerías y Frameworks:** se necesitará usar un conjunto de librerías o frameworks que nos facilitarán el desarrollo de la aplicación, centrándose en aspectos concretos de ésta. Para pa parte de Frontend se podría, por ejemplo, usar **Reactjs**, una librería enfocada en el desarrollo de la interfaz mediante componentes y que ademas de acelerar el proceso de codificación, nos permite crear componentes altamente reutilizables. Podría usarse alternativamente algún framework como **Angular**. También se deberán usar librerías que nos faciliten la realización de las pruebas, como puede ser **testing library**, para la realización de test unitarios de la interfaz o **Jest**, para testear la lógica de la aplicación.
- **Navegadores:** necesitaremos tener como mínimo los navegadores que se solicitan en el caso práctico para comprobar que el funcionamiento de la aplicación es correcto en todos y que visualmente no hay discordancias entre un navegador u otro. En nuestro caso, tendremos que probar la aplicación Edge, Firefox, Chrome y Safari.
- **Máquina Virtual:** ésta será solo necesaria en caso de que no tengamos acceso a todos los sistemas operativos en los que se ejecutan los diferentes navegadores, Edge (Windows) y Safafi (MacOS) principalmente. En la práctica, hay frameworks como **Selenium** que nos permiten testear una aplicación en diferentes navegadores si tener que instalar el sistema operativo en cuestión, aunque podría ser también una opción. Además dentro de este apartado podríamos incluir los containers, como **Docker**, que nos permiten agregar una alto grado de portabilidad a la aplicación.