

Entornos de Desarrollo

UT01: Desarrollo de Software

Ingeniería del software

Objetivos

- Afianzar los conceptos básicos del desarrollo de software

Elementos necesarios

- Ordenador con acceso a Internet
- LibreOffice Writer

Instrucciones

Esta actividad se hará de forma individual. Responde sobre el propio documento.

Tiempo estimado: 2 sesiones

01: Los problemas del desarrollo de software

Investiga sobre el Standish Group 2015 Chaos Report y contesta a las siguientes cuestiones. Puedes obtener un resumen del informe en español en

<http://www.laboratorioti.com/2016/05/16/informe-del-caos-2015-chaos-report-2015-bien-mal-fueron-los-proyectos-ano-2015/>

1. ¿Quiénes son Standish Group?

- Son una empresa encargada de estudiar una cantidad de proyectos de desarrollo software y determinar el porcentaje de éxito obtenido a lo largo del año.

2. ¿Qué es el Chaos Report?

- Es un estándar donde se determina lo que es el éxito de un proyecto, y el porcentaje obtenido a lo largo del año

3. ¿Qué porcentaje de proyectos software han sido exitosos, fracasados y no completamente exitosos según el informe de 2015?

Exitosos	Fallidos	No completados
29%	19%	52%

4. ¿Crees que el tamaño de los proyectos afecta en el éxito de los mismos?

- Si porque cuanto mas pequeño es el proyecto mas sencillo resulta, de dirigir, estructurar y llevar a cabo.

5. ¿Usar un ciclo de desarrollo en cascada o Agile (como SCRUM) afecta en la resolución exitosa de un proyecto? ¿Guarda esto alguna relación con el tamaño de los proyectos? Justifica la respuesta Si, porque los proyectos en cascada no pueden avanzar sin que el anterior campo de el visto bueno, mientras que en el agil, todos los campos van avanzando por su cuenta y retroalimentandose entre si, por esa razon la resolucion de problemas es mas llevadera. No, pueden desarrollar un proyecto en una metodologia o otra, pero el tamaño por si solo influye tanto en una como en otra.

Actividad 02: Proceso de desarrollo

Haz una analogía entre cada una de las fases del desarrollo de software del modelo en cascada y las fases por las que pasas la construcción de una casa de diseño. Por ejemplo:

- Especificación de requisitos: El arquitecto determina lo que el cliente quiere para la casa, es decir los requisitos: tipo de diseño, elementos, materiales, etc.
1. Especificación de requisitos(Análisis):El arquitecto escucha todos los requisitos que debe tener la casa y comienza a hacer un esquema básico de como debe finalizar el proyecto
 2. Diseño: El diseñador coge toda la información del arquitecto, y esboza una estructura básica que tiene que seguir el proyecto para finalizar con éxito, esto lo comprueba con el arquitecto que es quien tiene que dar el visto bueno
 3. Implementación: Los albañiles implementan el diseño y construyen la casa conforme los diseños que les han proporcionado, una vez acabado el diseñador revisará el proyecto para dar el visto bueno.
 4. Verificación: La casa piloto es presentada al cliente, y este dará su visto bueno o no al arquitecto, que en caso negativo tendrá que volver a hacer el esquema básico con el cual volver a empezar
 5. Mantenimiento: El arquitecto da las llaves de la casa al cliente, y le ofrece un mantenimiento de la misma, junto con el acuerdo de poder mejorar la casa

Actividad 03: Arquitectura del software

1. ¿Qué es un antipatrón de software? Explica algún ejemplo
 - Un antipatrón es las pautas a evitar cuando se diseña un proyecto
 - Dar a conocer funcionalidades antes de ser funcionales
2. ¿Qué son los Front-End y Back-End developers? ¿Con qué modelo de arquitectura de software crees que encajan estos perfiles? Justifica la respuesta y cita las fuentes consultadas:
 - Los front-End developers son aquellos que desarrollan la parte que visualizará el usuario, en cambio los back-End developers son aquellos que se encargan del desarrollo de la parte funcional que queda oculta al usuario
 - Esta arquitectura se utiliza para el desarrollo web
<https://eonomiatic.com/desarrollador-web/>
3. ¿Qué es la escalabilidad de un sistema software? ¿Cómo crees que el diseño arquitectónico en capas contribuye a este factor? Justifica la respuesta.
 - La escalabilidad de un sistema software es la capacidad de adaptarse a la demanda de rendimiento con el crecimiento de usuarios que lo utilizan.
 - El diseño arquitectónico nos permite escalar mejor un software puesto que al no saber exactamente la magnitud del software, dividimos el mismo en partes con un propósito específico, así de esta manera cada parte del software se encarga de una tarea diferente y reparten la carga del software
4. Define lo que es una arquitectura cliente-servidor desde el punto de vista de las arquitecturas software. Contrasta el modelo en 3 capas con una arquitectura clásica cliente-servidor. ¿Cuáles son las ventajas y los inconvenientes de cada uno de ellos?
 - Cliente-servidor. Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones. Arquitectura de tres niveles. Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se

encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

5. ¿Qué es un framework? ¿Qué ventajas le ves al uso de frameworks? ¿Qué frameworks de desarrollo podemos encontrar hoy en día en las plataformas Java, .Net, Javascript y PHP que sigan la arquitectura MVC?
- Es un software que nos permite desarrollar software de manera mas sencilla pues nos evita tener que introducir código de manera repetitiva, nos proporciona herramientas específicas del lenguaje que utilicemos.

Nombre	Lenguaje
Spring MVC	Java
ASP. NET MVC	.NET
Kendo	JavaScript
Laravel	PHP

Fuentes consultadas:

- <https://www.hostinger.com/tutorials/best-php-framework>
- <https://www.bbconsult.co.uk/blog/top-10-javascript-mvc-frameworks>
- <https://dotnet.microsoft.com/apps/aspnet/mvc>
- <https://www.jrebel.com/blog/java-mvc-frameworks-comparison>