

## NORMAS GENERALES PARA LAS ACTIVIDADES 2 DE LA UNIDAD DE TRABAJO 3

- Condiciones de entrega
    - La entrega de todas las actividades se hará a través de la herramienta Aules.
    - Cada actividad se nombrará con el nombre y primer apellido del alumno del alumno y la cadena “-actividades1-UT3”. Por ejemplo, el alumno con *Pepe Pérez*, nombrará sus actividades de la siguiente *PepePérez-actividades1-UT3*.
  - Condiciones de corrección
    - Si se detecta copia en alguna actividad se suspenderá automáticamente la unidad de trabajo a todos los alumnos implicados.
  - Calificación
    - Las actividades etiquetadas en verde puntuarán dentro del apartado de procedimientos, que supone un 15% de la nota de la UT3.
-

## Ejercicio 1. Completa las siguientes cuestiones

1.- Atendiendo a los tipos de ciclos de vida vistos en clase, completa la siguiente tabla, rellenando cada casilla con un "SI" o un "NO", dependiendo de si el tipo de ciclo de vida satisface la pregunta de la izquierda:

Requisito	Cascada	Prototipo	Iterativo	Incremental	Espiral
¿Los requerimientos están bien definidos o conocidos?	SI	SI	NO	SI	NO
¿Pueden cambiar frecuentemente los requerimientos?	NO	SI	SI	SI	SI
¿El sistema es complejo?	NO	NO	NO	NO	SI
¿Se puede empezar a probar pronto el sistema?	NO	SI	SI	SI	NO

2.- A continuación, se plantean varios supuestos de desarrollo de aplicaciones. Indica para cada uno qué tipo de ciclo de vida usarías y por qué. Indica también cuál (o cuáles) no usarías en ningún caso, y por qué.

a) Quieres hacer un pequeño programa que envíe un e-mail automáticamente a tus amigos el día de su cumpleaños. Sólo lo vas a usar tú, y sólo necesitas un fichero con las fechas y nombres de los cumpleaños, con los e-mails, y luego el programa que tome ese fichero y envíe el e-mail a quien corresponda

Optaría por el modelo en cascada al ser un proyecto pequeño, para mí, y que no se requiere de mucha inversión o incremental, a pesar de que es un proyecto muy simple y sólo para mí, las instrucciones son simples y concisas, puedo modificarlo y añadir nuevas versiones sobre lo anteriormente escrito.

En este proyecto no usaría nunca el modelo en cascada en caso de querer en un futuro ampliar la agenda de cumpleaños o felicitaciones, ya que para aumentar la lista de los amigos o cambiar las fechas tendría que volver a empezar de nuevo desde el principio.

b) Una pequeña empresa te ha encargado un programa para gestionar el inventario de su almacén. El programa tiene unos requisitos claros (poder dar de alta o de baja productos en una lista), pero los jefes no tienen muy claro qué apariencia quieren que tenga el programa, y qué funcionalidades.

Optaría por el modelo iterativo, ya que me permite realizar mejoras del programa y el cliente al no tener clara la apariencia ni las funcionalidades, no me hace falta para empezar que me lo diga desde el principio, si no que o puedo modificar en el transcurso.

En este proyecto no usaría nunca el modelo en cascada ni basado en prototipos ya que, al no tener del todo clara la apariencia ni las funcionalidades del proyecto, tendría que volver a empezar de nuevo desde el principio, además es posible que no se tenga que realizar una única entrega al cliente.

c) Un amigo te propone un programa para instalar en un móvil y monitorear la actividad física de una persona en un gimnasio (calorías quemadas, km recorridos, etc.). No tiene demasiado claro cómo hacerlo, ni lo que vais a tardar, ni lo que los gimnasios pueden andar buscando exactamente. Le propones entregarle el programa con todo lo que te ha comentado y luego ir refinando.

Al tratarse de un proyecto, con un entorno incierto, donde las funcionalidades se van introduciendo poco a poco, optaría por un modelo incremental, ya que puedo mantener las versiones anteriores y perfilar la aplicación poco a poco, integrando funcionalidades.

En cambio, nunca usaría nunca el modelo en cascada ni basado en prototipos ya que, al no saber las funcionalidades del proyecto, tendría que volver a empezar de nuevo desde el principio, además es posible que no se tenga que realizar una única entrega al cliente.

3.- Elabora una tabla comparativa de las ventajas e inconvenientes de los siguientes tipos de ciclos de vida, atendiendo a lo que hay en las transparencias de clase y añadiendo más ventajas e inconvenientes que tú supongas que tienen:

	<b>Cascada</b>	<b>Incremental</b>
<b>Ventajas</b>	<p>Realiza un buen funcionamiento en equipos débiles y productos maduros, por lo que se requiere de menos dinero y herramientas para hacerlo funcionar de manera óptima.</p> <p>Es un modelo fácil de implementar y entender.</p> <p>Promueve una metodología de trabajo efectivo: diseñar antes que codificar.</p>	<p>Se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.</p> <p>También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del software. El modelo proporciona todas las ventajas del modelo en Cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.</p> <p><input type="checkbox"/> Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.</p>
<b>Inconvenientes</b>	<p>En la vida real, un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lleve al fracaso.</p> <p>Cualquier error de diseño detectado en la etapa de prueba conduce al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.</p> <p>Una etapa determinada del proyecto no se puede llevar a cabo a menos de que se haya culminado la etapa anterior.</p>	<p>No es recomendable para casos de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido y/o de alto índice de riesgos.</p> <p>Requiere de mucha planeación, tanto administrativa como técnica.</p> <p>Requiere de metas claras para conocer el estado del proyecto.</p>

4.- Indica si son verdaderas o falsas las siguientes afirmaciones, y por qué:

- El modelo de CV en cascada tiene las etapas de especificación de requisitos, análisis, diseño, implementación, pruebas e instalación y mantenimiento. El incremental no.

Falso. En el modelo incremental no hay fase de instalación ni mantenimiento, entre el desarrollo y las pruebas continuas se ubican las versiones y modificaciones de la aplicación.

- El modelo incremental necesita conocer de antemano todos los requisitos

Falso. Se adapta a entornos inciertos, pero con los requisitos claros (las funcionalidades deben estar claras desde el principio), donde se pueden ir añadiendo funcionalidades poco a poco.

- El prototipado rápido es incompatible con el CV en cascada

Falso. Permite mayor flexibilidad, integración y opciones el prototipado rápido.

- El radio de la espiral en el modelo en espiral nos indica cuánto hemos avanzado en el proyecto

Verdadero. En cada vuelta se seguirán los siguientes pasos: determinar lo que se quiere lograr, determinar las posibles alternativas para lograrlo, analizando sus riesgos, los resultados finales y seleccionando la mejor, seguir con la alternativa seleccionada en el paso anterior y por último establecer qué se tiene terminado.