

NOTAS Y RECOMENDACIONES PARA PROYECTOS DE DESARROLLO

- Armarse una rutina de trabajo consistente y continua en el tiempo.
- Utilizar de 4 a 8 horas a la semana para realizar el proyecto, preferentemente en 2 sesiones a la semana.
- La complejidad del proyecto debe ser media (no menos de 300 líneas de código por integrante del equipo, o no menos de 30 puntos de función o no menos de 10 clases; y no mas de 1000 líneas de código, 100 puntos de función o 30 clases por integrante).
- Implementar una bitácora donde se detalle por cada sesión de trabajo, lo siguiente:
 - Temas a tratar, módulos a realizar, actividades a hacer.
 - Problemas hallados.
 - Con que se continuará en la próxima sesión, y qué temas quedan abiertos.
 - Utilizar la bitácora siempre leyendo la bitácora de la última sesión antes de comenzar con la sesión siguiente.
- En la selección del Proyecto, tener en cuenta no dejar demasiadas variables desconocidas, por ejemplo, utilizar ambientes de desarrollo ya conocidos, o lenguajes de programación ya conocidos, o encarar sistemas de información o problemas conocidos.
- Es conveniente que nunca todos estos temas sean desconocidos al mismo tiempo.
- El grupo de trabajo no debe ser menor a tres personas ni mayor a cinco. En casos excepcionales se aceptarán trabajos individuales.
- Escribir el anteproyecto inicial basado en este template y luego sobre el mismo documento del proyecto inicial, ir generando nuevas versiones del mismo, cada vez mas completas, hasta tener el proyecto terminado y documentado. Hacer el proyecto "Y LA DOCUMENTACION" al mismo tiempo, no dejar la documentación para el final.
 - Testear frecuentemente el código y armar funciones o módulos de testeo al mismo tiempo que se escriben las funciones o módulos del proyecto.
 - Utilizar herramientas robustas y conocidas, y tecnologías ya probadas y populares, esto es importante para que los errores dependan de quién está haciendo el proyecto y no de problemas externos.
 - Las plataformas de desarrollo NetBeans, Eclipse, Xcode, o Visual Studio, son maduras y robustas, y pueden usarse con múltiples lenguajes de desarrollo.
 - El lenguaje de implementación del proyecto debe ser orientado a objetos:
 - Java
 - Python
 - C/C++
 - Si se propone otro lenguaje Justificar.

- Otras alternativas.
 - Objective-C
 - Ruby
 - Groovy
 - Object Pascal (o Módula u Oberon)
 - CLOS (Common Lisp Object System)
 - SmallTalk (Squeak o GNU SmallTalk)
 - PHP 5 o superior
 - C#
 - Plataforma preferida para desarrollo: La materia prefiere la utilización de soluciones y tecnologías Open Source, Software Libre, FLOSS, o FOSS, como por ejemplo Eclipse o NetBeans
 - Ahora si el desarrollador quiere usar la terminal (consola) y un entorno de caracteres para desarrollar, compilar y probar, no hay problema, siempre que se sienta comodo con él.
 - Hacer backup frecuente del proyecto, por ejemplo después de cada sesión, aún enviándoselo a los integrantes del grupo por email, comprimido, eso puede bastar.
 - Tener múltiples versiones del proyecto y no pisar la versión anterior con la generada en la sesión actual.
 - Balancear la carga de trabajo entre los distintos integrantes del proyecto, que todos realicen trabajos equivalentes, no es necesario que realicen todos las mismas actividades, pero sí que se complementen y que todos conozcan, aunque sea “por arriba” el trabajo realizado por los otros.
 - Cada integrante del equipo debe estar cómodo con la actividad que realiza, para algunos será escribir código, para otros será realizar el análisis y diseño, para otro la interfase visual o el testing del proyecto.
 - Se recomienda a los participantes del proyecto ser pesimistas con las promesas y en la planificación y ser exigentes consigo mismos.
- Se recomienda utilizar la metodología de Programación Extrema para el desarrollo del proyecto (XP).
 - Es altamente recomendable utilizar técnicas de programación extrema (Extreme programming), si no sabe lo que significa búsquelo en Google o Wikipedia:
 - http://en.wikipedia.org/wiki/Extreme_programming
 - http://es.wikipedia.org/wiki/Programación_extrema
- Si no existe infraestructura física para múltiples máquinas y en caso de que el proyecto lo

requiera, entonces se recomienda el uso de máquinas virtuales.

Javier Blanco