

Examen de Julio de IS2

- **Esquema de matriculacion de practicas**

- *Configurar pom para utilizar cobertura*

Configuramos el pom para añadir el plugin de site y con el el de cobertura

- *Pasos y ejecuciones que debes hacer si es la primera vez que ejecutas maven y quieres obtener los ficheros de cobertura*

Esta no la tenia muy clara, pero pense que seria que cuando ejecutas `mvn site` no se compila el codigo o algo asi, asi que puse que se debe ejecutar `mvn compile` para que se compile todo previamente.

- **Esquema de hudson, pruebas cada 15 min.**

- *Que indica el esquema? y que decision tomarias si fueras el director del proyecto?*

Lo de explicar el esquema lo sabias, era lo de que en un momento tenias X pruebas y como era todo azul pues todas pasaban correctamente y llegaba un punto que se añadian mas pruebas y tambien se pasaban correctamente. En cuanto a la segunda pregunta, como todo avanza correctamente no hay necesidad de ninguna decision a tomar por parte del administrador.

- **Caja blanca**

El esquema es el mismo de las transparencias, pero en la condicion de salida habia && en vez de ||. Esquema en Google docs. En cuanto a la segunda parte el stub era de la propia clase en la que estaba implementado el metodo valida pin porque el metodo que queremos controlar es el que controla el input del usuario (obtener pin) el cual tendremos que sobrescribir en el Stub. Para falsear la introduccion de varios datos por el usuario lo que hice fue meter un array por el constructor del stub con los datos que introduciria el *usuario* que en realidad podemos controlar, y un contador para saber por que datos vamos. De esta manera definimos el pin bueno como el "3333" por ejemplo y para que salga por la condicion del contador > 2 tendríamos que meter 3 fallos en el array, ["1234", "2222", "4444", "5555"]. Las condiciones de comprobacion de pin en el metodo comprobar pin tambien tendremos que programarlas nosotros sobrescribiendo ese metodo.

El codigo java esta en los ficheros java adjuntos.

- **Responde verdadero o falso:**

1. MsProject sirve para monitorizar y no para controlar: puse que verdadero porque no es una herramienta de control de proyectos, sino que sirve al administrador del

proyecto para visualizar la informacion y tomar decisiones al respecto, por lo que solo es una herramienta de monitorizacion (ademas de que no hace estimaciones).

2. MsProject puede trabajar con varias agendas y con diferentes niveles de abstraccion: verdadero porque podemos poner un WBS estructurado a distintos niveles y cerrar y abrir esos niveles para centrarnos en una parte concreta de la planificacion.

3. ¿? pero puse falso

4. ¿? pero puse falso

- **DBUnit con Stub. Si estamos realizando pruebas de integracion en nuestro proyecto, realizaremos pruebas con DBUnit.**

Se podrian necesitar stubs para sustituir dependencias no controladas que no fuesen bases de datos, por ejemplo un servicio web que nos diera la cotizacion en bolsa o el tiempo NTP. Cuando estamos haciendo pruebas de integracion lo que no debemos de utilizar son stubs que sustituyan las partes que estamos integrando. Si estamos integrando dos modulos de la aplicacion debemos de poder probar las interfaces que los conectan y si funcionan realmente bien, por ello no se deben utilizar los stubs que quizas se implementaron anteriormente para sustituir esa funcionalidad. El caso de DBUnit es mas necesario en las pruebas de unidad en las cuales pruebas una parte del codigo que se encarga de realizar las operaciones con la base de datos, con lo cual en las pruebas de integracion esas pruebas existiran porque aseguran el correcto funcionamiento de las unidades que se van a integrar.

- **Indica todos los pasos que se deben hacer para calcular cuantas iteraciones se realizaran por cada fase de UP.** Debemos realizar una estimacion completa, es decir.

Primero estimamos el tamaño del proyecto mediante el metodo que deseemos, proyectos pasados o COCOMO por ejemplo. Con este tamaño pasamos a estimar el esfuerzo necesario para realizar el proyecto y el tiempo en el que se realizara, por ejemplo con Construx. Hecho esto necesitamos las estimaciones de distribucion del tiempo en un proyecto UP, que estan definidas en las transparencias (yo puse el ejemplo de la primera fase que es un 10%). Calculamos el tiempo que nos toca hacer en esa fase y el tiempo de timeboxing que tenemos por iteracion en nuestra empresa. Dividimos el tiempo total de la fase entre el tiempo de cada iteracion y obtenemos el numero de iteraciones que se haran en esa fase, y asi con todas. Por ejemplo:

Las estimaciones de tiempo nos dan 100 dias de proyecto, para la fase de inicio tenemos un 10% con lo cual seran 10 dias. Si nuestro timeboxing es de 7 dias, dividimos $10/7$ y nos da 1,4, pero como no podemos poner un numero decimal de iteraciones, pues para no quedarnos cortos, 2 iteraciones tendremos que realizar en la fase de inicio.