|  |  |  |  |
| --- | --- | --- | --- |
|  | En proceso | Puede mejorar | Alcanzado |
| Prueba los **métodos** más importantes de los **objetos** principales del modelo. | Se prueban los métodos principales, pero hay un grave error que se explicará más adelante |  |  |
| Las pruebas están **atomizadas** (se prueban unidades de código acotadas). | No prueban unidades de código acotadas. |  |  |
| En el **código** de las pruebas automatizadas se utilizan **nombres claros y significativos**. |  |  | Se entiende perfectamente. |
| Las distintas pruebas están **agrupadas** en **clases o categorías**. | No están agrupadas en clases o en categorías, son básicamente lo mismo. |  |  |
| Emplea **métodos** de prueba variados (***that*, *notThat*, *equals*, *notEquals*, etc.**) | No utiliza ningún método variado |  |  |
| Las pruebas **detectan fallos inyectados** en el código o producto de la **variación en el lote de datos**. | No se pueden comprobar las pruebas |  |  |

Quiero aclarar un error sumamente **grave**, en cada caso de prueba que hicieron se entiende lo que se busca además de que lleva un comentario explicándolo, esto me parece perfecto pero al verificar en cada caso usan un *assert.that(true),* esto significa que no comprueba ningún método y cada prueba siempre devolverá un **ok** ignorando los métodos que estés probando.

Dentro de las comillas va el método que estas comprobando, ejemplo:

*assert.that(colectivo.estaLleno())*

Es un modelo simple y sencillo que no contiene gran variedad de métodos y objetos, esto hace que todos los métodos que contiene sean “importantes” logrando que se utilicen esencialmente en los casos de prueba (tomo en cuenta lo que realmente buscaban ignorando el **error** principal que aclaré, fuera del ámbito de las pruebas necesita mejorar bastante ya que hay **cuestiones/preguntas** que no responde el modelo y menos lo hará en los casos de prueba.

Lo último que aclararé es que los nombres de los métodos y variables del código son entendibles, pero no me gustó nada que haya un método que te informe todo sobre el colectivo, sugiero que hagan métodos distintos para saber cada cosa del colectivo.

Además, estos métodos tienen que retornar un valor **booleano** o **numérico** no un **texto**, esto beneficia porque se puede utilizar el método dentro de los métodos assert.