EJEMPLO DE ACTIVIDAD DE APRENDIZAJE

NOMBRE DEL MÓDULO	**Programación orientada a objetos**
NOMBRE DE LA ACTIVIDAD DE APRENDIZAJE	Construcción de una unidad de prueba para verificar un resultado esperado
DURACIÓN DE LA ACTIVIDAD	4 horas
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN QUE INCLUYE
1. Construye unidades de prueba para verificar el correcto funcionamiento de la codificación realizada, de acuerdo a exigencias técnicas de confiabilidad.	1.1 Realiza pruebas para detectar problemas previos a la codificación de una unidad de *software* con una herramienta de *software* disponible en el mercado.
METODOLOGÍAS SELECCIONADAS	Detección de fallas

DESCRIPCIÓN DE LAS TAREAS QUE REALIZAN DOCENTES Y ESTUDIANTES, Y LOS RECURSOS QUE SE UTILIZAN EN CADA UNA DE LAS SIGUIENTES ETAPAS

PREPARACIÓN DE LA ACTIVIDAD

Docente:

- Prepara el laboratorio con puestos de trabajo y *software* de trabajo instalado en el computador.
- Elabora una presentación (PPT) del caso de pruebas con uso de herramientas de depuración.
- Genera una guía con el planteamiento y los alcances del trabajo a realizar.

Recursos:

- Laboratorio con puestos de trabajo y energía disponible.
- Computador.
- NetBeans.
- MySQL.
- MySQLConnector/J.
- Java y otros *software* actualizados.
- Manuales de uso y referencia de *software* utilizado.

DESCRIPCIÓN DE LAS TAREAS QUE REALIZAN DOCENTES Y ESTUDIANTES, Y LOS RECURSOS QUE SE UTILIZAN EN CADA UNA DE LAS SIGUIENTES ETAPAS:	
EJECUCIÓN	**Docente:** › Explica y contextualiza la actividad de la clase. › Expone la importancia de las pruebas de caja negra sobre una aplicación o los componentes de la aplicación, ya sea de escritorio o web. › Muestra los criterios para crear un plan de pruebas, reconociendo la importancia que posee tanto en la planificación del proyecto sobre los requisitos funcionales y no funcionales como en la calidad esperada de la solución. › Propone un ejercicio para realizar pruebas con JUnit a partir de un plan de pruebas, ilustrando cada una de las alternativas y casos en que una aplicación puede generar errores, excepciones o situaciones de borde, aplicando la orientación a objeto obtenida por el estándar O.R.M. (Mapeo Objeto-Relacional). › Entrega guía de procedimiento de trabajo, los archivos a probar, los manuales y las guías de referencia de los *software* involucrados, en medio manual o digital, para construir una prueba de resultados. **Estudiantes:** › Observan la presentación realizada para realizar el procedimiento de la clase. › Revisan la guía y los manuales entregados. › Analizan el caso y las alternativas de uso del lenguaje para su solución. › Activan los programas de entorno necesarios para trabajar y desarrollan las sentencias en función del objetivo. › Explican la relación entre las sentencias utilizadas y el objetivo esperado de solución. › Reconocen errores y corrigen.
CIERRE	**Estudiantes:** › En un plenario dan cuenta de la experiencia de la actividad realizada y las dificultades más comunes. › Formulan conclusiones del trabajo. **Docente:** › Retroalimenta la actividad de clase, comentando las soluciones y aclarando los errores del procedimiento. › Profundiza el funcionamiento de los *assert* y los *fail* que ocurren dentro de una prueba unitaria, teniendo en cuenta cómo se compara la orientación a objetos con un resultado esperado. › Aclara que JUnit es también un medio de controlar las pruebas de regresión necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple con los requerimientos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.