



# Testeo y Métodos de Usabilidad

Prof. Pedro Santana

Testeo del software



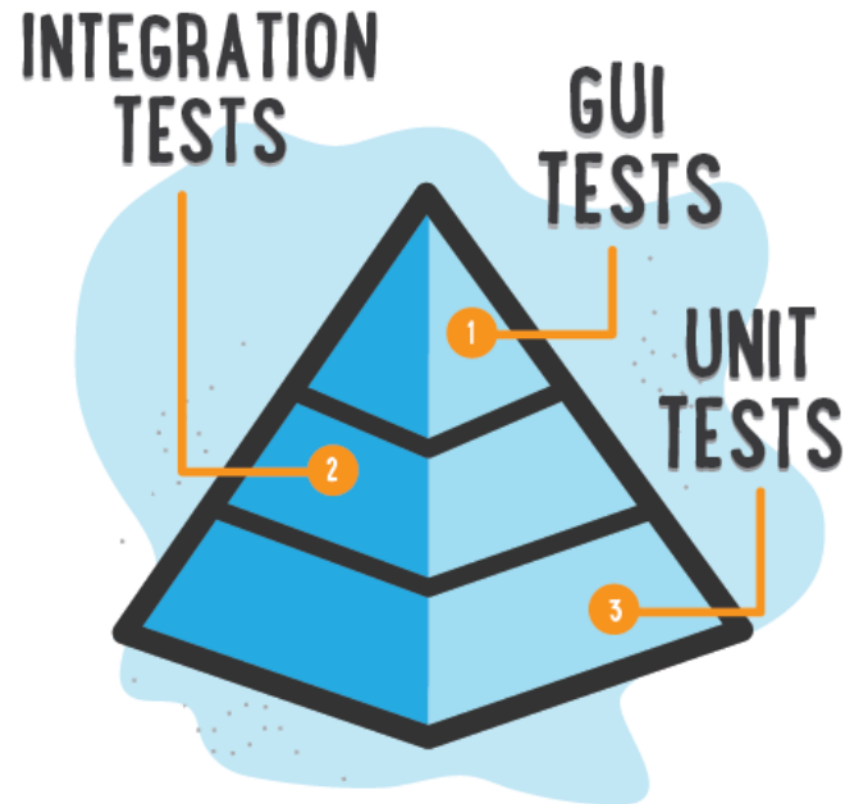
# Prueba de unidad

---

- ▶ Nombre que reciben los procedimientos de pruebas locales a un módulo del sistema.
- ▶ Por definición dichas pruebas cubren la funcionalidad propia del módulo tanto con una perspectiva de caja blanca como de caja negra; pero prestando **poca** o **ninguna** atención a la integración con otros módulos.



- ▶ Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.
- ▶ Luego, con las **Pruebas de Integración**, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.



# Características

---

- ▶ Para que una prueba unitaria sea buena se deben cumplir los siguientes requisitos:
  - ▶ **Automatizable:** no debería requerirse una intervención manual.
  - ▶ **Completas:** deben cubrir la mayor cantidad de código.
  - ▶ **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez.
  - ▶ **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
  - ▶ **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.
- ▶ Aunque estos requisitos no tienen que ser cumplidos al pie de la letra, se recomienda seguirlos o de lo contrario las pruebas pierden parte de su función.



# Limitaciones

---

- ▶ Las pruebas unitarias no descubrirán todos los errores del código.
- ▶ Por definición, sólo prueban las unidades por sí solas, por lo tanto, no descubrirán:
  - ▶ Errores de integración
  - ▶ Problemas de rendimiento
  - ▶ Otros problemas que afectan a todo el sistema en su conjunto.
- ▶ Puede no ser trivial anticipar todos los casos especiales de entradas que puede recibir en realidad la unidad de programa bajo estudio.
- ▶ Las pruebas unitarias sólo son efectivas si se usan en conjunto con otras pruebas de software.



# Herramientas automatizadas

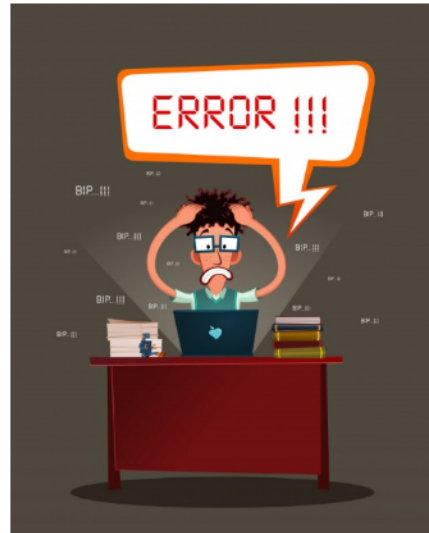
---

- ▶ **JUnit:** Entorno de pruebas para Java creado por Erich Gamma y Kent Beck. Se encuentra basado en SUnit creado originalmente para realizar pruebas unitarias para el lenguaje Smalltalk.
- ▶ **TestNG:** Creado para suplir algunas deficiencias en JUnit.
- ▶ **JUnit:** Basado en anotaciones, como TestNG.
- ▶ **SimpleTest:** Entorno de pruebas para aplicaciones realizadas en PHP.
- ▶ **PHPUnit:** framework para realizar pruebas unitarias en PHP.
- ▶ **PHPUnit:** Versión del framework para lenguajes C/C++.
- ▶ **PHPUnit:** Versión del framework para la plataforma.NET.



# ¡Ensuciémonos las manos!

---



# GitHub Classroom

Your course assignments on GitHub





# GitHub

---

- ▶ GitHub es una plataforma de **desarrollo colaborativo de software** para alojar proyectos utilizando el sistema de control de versiones Git.
- ▶ Git es un software de **control de versiones** diseñado por Linus Torvalds. Pensado para trabajarse a bajo nivel (CLI) pero se han desarrollado varias interfaces de usuario que facilitan su uso.
- ▶ Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de un producto.
- ▶ GitHub aloja los repositorios de código y brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

# GitHub classroom

---

- ▶ Los métodos tradicionales de entrega de prácticas de programación son obsoletos:
  - ▶ USB, .zip por email, cd, etc.
- ▶ Al no contar con un servicio de control de versiones el trabajo puede perderse al dañarse el medio donde se está almacenando o por algún fallo.
- ▶ GitHub ha creado varios servicios para apoyar a estudiantes y profesores en sus actividades educativas, en estos servicios encontramos **Classroom for GitHub**.
  - ▶ Permite crear repositorios para cada alumno.
  - ▶ Facilita el seguimiento y la retroalimentación.
  - ▶ Apoya el aprendizaje de control de versiones.

# Entrega de la tarea

---

- ▶ Cada alumno debe inscribirse al classroom de la materia
  - ▶ <https://classroom.github.com/a/zGuvs8K1>
  - ▶ <https://github.com/ihclab/>



# Contexto de la prueba

---

- ▶ Considerar la siguiente aplicación:



# Contexto de la prueba

---

- ▶ La aplicación calcula la media de un conjunto de enteros.
- ▶ Detrás de la aplicación encontramos referencia a la clase

## **Medias.**

- ▶ Dicha librería contiene los métodos:
  - ▶ `mediaAritmetica()`
  - ▶ `mediaGeometrica()`
  - ▶ `mediaArmonica()`



```
namespace pruebasunitarias {  
    class Medias {  
        /**  
         * Calcula y regresa la media aritmética  
         */  
        public static double mediaAritmetica(params int[] vals) { }  
  
        /**  
         * Calcula y regresa la raíz enésima =  $x^{(1/n)}$   
         */  
        private static double raizEnesima(double x, int n) { }  
  
        /**  
         * Usa raizEnesima para calcular y regresar la media geométrica  
         */  
        public double mediaGeometrica(params int[] vals) {}  
  
        /**  
         * Este método no está implementado  
         */  
        public static double mediaArmonica(params int[] vals) { }  
    }  
}
```

---



# Almacenar los casos de prueba

---

## ► Problema 01

- Crear y almacenar los casos de prueba en un archivo de texto plano.

## ► Diseño

- Usar un archivo delimitado por : que incluya un identificador único para cada caso de prueba, valores de entrada y resultados esperados.

## ► Solución

- 0001:mediaAritmetica:2 4 8:4.66
- 0002:mediaAritmetica:1 5:3.00
- 0003:mediaAritmetica:1 2 4 8 16 32:10.50



# Leer los casos de prueba

---

## ▶ Problema 02

- ▶ Se desea leer cada caso de prueba del archivo de texto donde se almacenan.

## ▶ Diseño

- ▶ Iterar a través de cada línea del archivo de casos de prueba usando un ciclo **while** con el objeto ***System.IO.StreamReader***.

## ▶ Solución

- ▶ [Actividad para alumnos.](#)





# Procesado (parsing) del caso de prueba

---

## ▶ Problema 03

- ▶ Se desea procesar los campos individuales del caso de prueba.

## ▶ Diseño

- ▶ Usar el método ***String.Split()***, pasándole como argumento el carácter delimitador y almacenando en un ***arreglo de strings*** el valor resultante.

## ▶ Solución

- ▶ **Actividad para alumnos.**



# Conversión de tipos de datos

---

## ▶ Problema 04

- ▶ Convertir las entradas y los resultados esperados del caso de prueba de **string** al tipo de datos esperado por los métodos a probar.

## ▶ Diseño

- ▶ Actividad para alumnos.

## ▶ Solución

- ▶ Actividad para alumnos.



# Determinar el resultado del caso de prueba

---

## ▶ Problema 05

- ▶ Se desea determinar cuando un caso de prueba de unidad tiene éxito o falla.

## ▶ Diseño

- ▶ Actividad para alumnos.

## ▶ Solución

- ▶ Actividad para alumnos.



# Validar valores nulos de entrada

---

- ▶ Problema 06

- ▶ Desea probar el correcto manejo de valores de entrada NULL.

- ▶ Diseño

- ▶ Actividad para alumnos.

- ▶ Solución

- ▶ Actividad para alumnos.



# Validar excepciones en el resultado

---

- ▶ Problema 07

- ▶ Desea probar métodos que generen una excepción.

- ▶ Diseño

- ▶ Actividad para alumnos.

- ▶ Solución

- ▶ Actividad para alumnos.



# Actividades complementarias

---

## ▶ Actividad 01

- ▶ Registro de los resultados (éxito o falla).
- ▶ Color verde para éxito.
- ▶ Color rojo para falla.

## ▶ Actividad 02

- ▶ Determinar el tiempo que llevó ejecutar el caso de prueba.

## ▶ Actividad 03

- ▶ Guardar los resultados en un archivo de texto, que incluya en el nombre la fecha y hora para identificar la ejecución de una prueba.



# Entrega de la tarea

---

- ▶ Por cada Problema deben hacer un **Git Commit**, en la descripción de los cambios realizados deben indicar el **problema o actividad** que están agregando.



# ¿Cómo usar GitHub?

---

- ▶ Línea de comandos.

- ▶ Git for Windows: <https://gitforwindows.org/>
- ▶ Linux: `sudo apt-get install git`
- ▶ Git OSX installer: <https://sourceforge.net/projects/git-osx-installer/>

- ▶ GitHub Desktop

- ▶ <https://desktop.github.com/>

- ▶ SourceTree

- ▶ <https://www.sourcetreeapp.com/>

- ▶ TortoiseGit

- ▶ <https://tortoisegit.org/>

- ▶ GitHub Extension for Visual Studio

- ▶ <https://visualstudio.github.com/>

