

# Aseguramiento de la calidad del software

## CI-0112 Programación 1

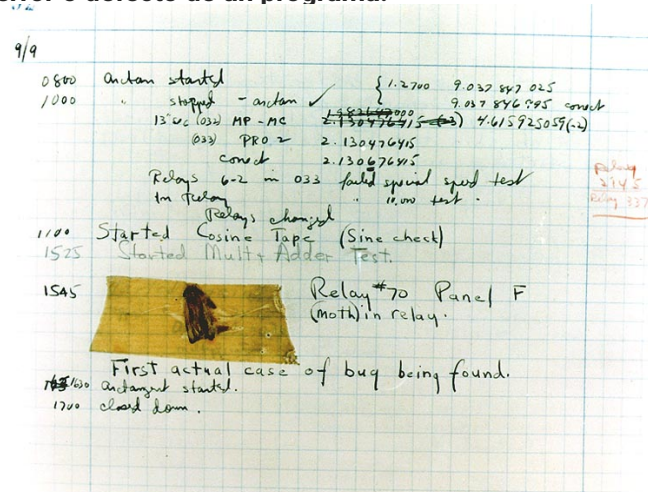
Sivana Hamer - [sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)  
Escuela de Ciencias de la Computación e Informática  
Universidad de Costa Rica  
Licencia: CC BY-NC-SA 4.0



En el proceso de crear software, se tiene que asegurar que se está creando software de cierta calidad por medio de distintas actividades.



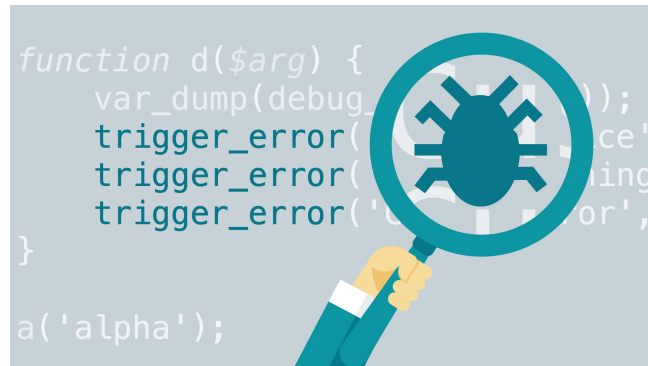
Un bug es un error o defecto de un programa.



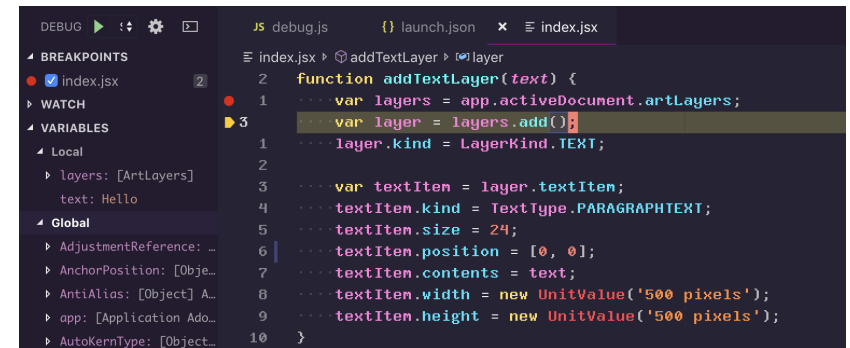
Testing es la actividad de software donde se determina que el código no funciona adecuadamente. Es decir, se encuentran bugs.



Debugging es la actividad de software donde se encuentran y eliminan bugs.



El debugger es una herramienta que ayuda a determinar examinar el código cuando se ejecuta.



Existen muchas maneras de clasificar los defectos:

- Por la **etapa de software** en que se **introdujó**.
- Por la **severidad** o impacto del defecto.
- Por los **errores que provoca**.

Errores de compilación son errores en la **sintaxis** del programa. Es decir, hay un problema gramatical. Provocan que no se pueda compilar un programa.



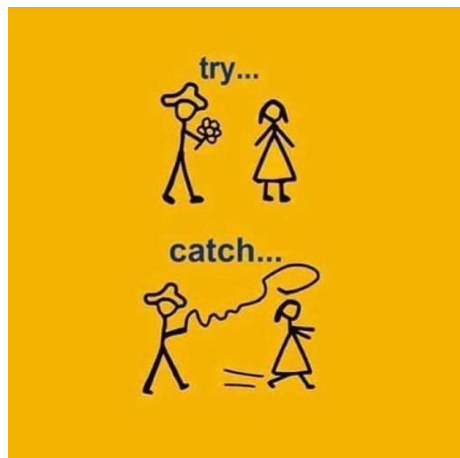
## Recomendaciones para errores de compilación

- Crear el software incrementalmente.
- Utilizar la información que sale en el error.
- Buscar cerca, usualmente antes, de la zona donde indica el error.
- Seguir buenas prácticas de programación.
- Ir arreglando error por error, iniciando en el primero.
- Entender el programa.
- En caso de estar pegado, compilar el código por partes (debugging by bisection).
- Usar internet para buscar el error. Muchos errores tiene información en foros y StackOverflow.

Errores de ejecución son errores que suceden cuando se ejecutan los programas. También son llamados excepciones.

```
input
Compilation failed due to following error(s).
main.c: In function 'main':
main.c:14:12: warning: division by zero [-Wdiv-by-zero]
    int b=a/0;
           ^
```

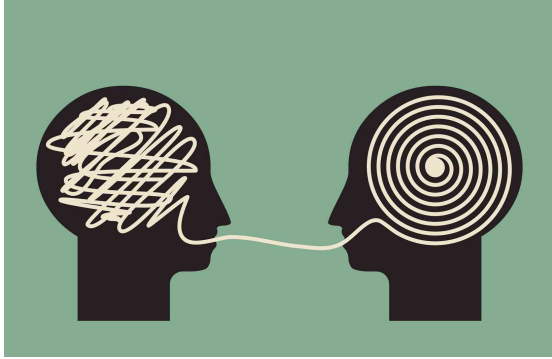
Un bloque try es donde una excepción puede ocurrir. Un bloque catch maneja la excepción.



## Recomendaciones para errores de ejecución

- Usar logs (prints) o el debugger.
- Usar tries y excepts.
- Asegurarse que los ciclos tengan condiciones para finalizar.
- Manejar los datos de entrada **siempre**.
- Revisar si se están usando valores validos.
- Considerar siempre si las operaciones matemáticas se indefinen con algunos valores.

Errores lógicos son errores de la lógica del programa que producen resultados no deseados.



## Recomendaciones para errores lógicos

- Usar logs o debuggers.
- KISS (Keep it simple stupid).
- Revisar el flujo de ejecución del programa paso a paso.
- Probar a mano y utilizar diagramas.
- Alejarse de la computadora por unos minutos.
- Explicar el programa a otra persona (rubber ducking).
- Preguntar por ayuda.

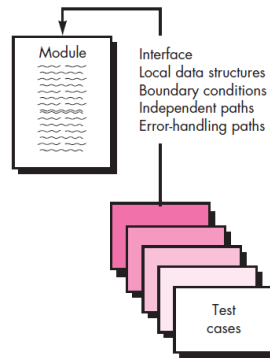
“If you are absolutely sure that everything in your program is right, and if it still doesn’t work, then one of the things that you are absolutely sure of is wrong.”

(Eck, 2020)

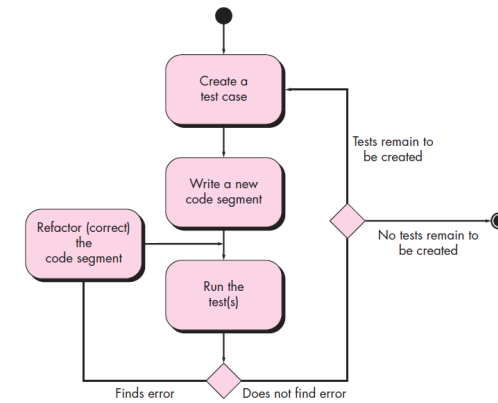
## Se puede probar el software de dos maneras...

- **Manualmente:** Se prueba a mano, ya sea l@s programador@s, usuari@s o testers.
- **Automaticamente:** Se crean o utilizan programas que prueban el software.

Las pruebas de unidad verifica una unidad (unit) de software, por ejemplo un método o componente.

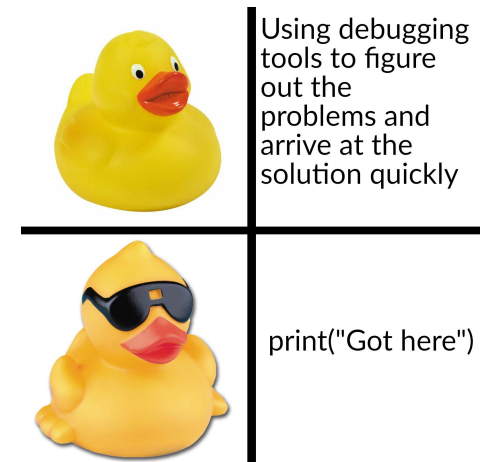


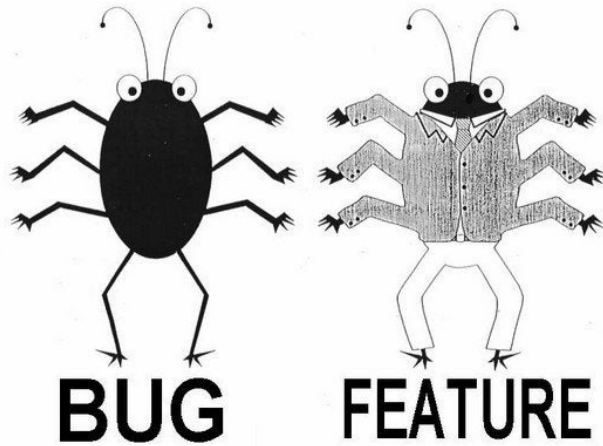
En Test driven development se diseñan las pruebas antes de crear el software.



JUnit es un framework que permite hacer prubeas unitarias en Java.

**J**Unit





## Referencias II

- D. J. Barnes and M. Kölling, *Objects First with Java: A Practical Introduction Using BlueJ*, sixth edition ed. Boston: Pearson, 2017.
- A. Mikhailau. Software quality management: What, where, and how? [Image]. [Online]. Available: <https://www.scnsoft.com/software-testing/quality-management-optimization>
- M. Monkey. Itech7409 software testing assignment help. [Image]. [Online]. Available: <https://www.moodlemonkey.com/solution/itech7409-software-testing-assignment-help/>
- JUnit. Junit. [Online]. Available: <https://junit.org/>
- Mr\_recci. Bug vs feature. [Image]. [Online]. Available: [https://www.reddit.com/r/ProgrammerHumor/comments/1w1u3o/bug\\_vs\\_feature/](https://www.reddit.com/r/ProgrammerHumor/comments/1w1u3o/bug_vs_feature/)
- R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. New York: McGraw-Hill Higher Education, 2010.

## Referencias I

- N. S. W. Center. (2012) First computer bug, 1945. [Image]. [Online]. Available: [https://commons.wikimedia.org/wiki/File:First\\_Computer\\_Bug,\\_1945.jpg](https://commons.wikimedia.org/wiki/File:First_Computer_Bug,_1945.jpg)
- K. Shots. (2019) Extendscript debugger 1.1.0 released. [Image]. [Online]. Available: <https://medium.com/adobetech/extendscript-debugger-1-1-0-released-5f2e1a068e89>
- javatpoint. Compile time vs runtime. [Image]. [Online]. Available: <https://www.javatpoint.com/compile-time-vs-runtime>
- T. irrational investor. The fallacy of logical reasoning. [Image]. [Online]. Available: [https://medium.com/@akshay\\_m/the-fallacy-of-logical-reasoning-fafa567bc617](https://medium.com/@akshay_m/the-fallacy-of-logical-reasoning-fafa567bc617)
- A. B. Downey and C. Mayfield, *Think Java: How to Think Like a Computer Scientist*, second edition ed., 2020.
- D. J. Eck, *Introduction to Programming Using Java*, eighth edition ed. Geneva (NY): Hobart and William Smith Colleges, Department of mathematics and computer science, 2020.

## Referencias III

- I. Sommerville, *Software Engineering*, tenth edition, global edition ed., ser. Always Learning. Boston Columbus Indianapolis New York San Francisco Hoboken Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo: Pearson, 2016.
- Oracle. Lesson: Exceptions. [Online]. Available: <https://docs.oracle.com/javase/tutorial/essential/exceptions/>
- ShubhamBadal. try... catch... [Image]. [Online]. Available: [https://www.reddit.com/r/ProgrammerHumor/comments/82hdxu/try\\_catch/](https://www.reddit.com/r/ProgrammerHumor/comments/82hdxu/try_catch/)
- arcanewright. Duck, duck, debug. [Image]. [Online]. Available: [https://www.reddit.com/r/ProgrammerHumor/comments/nn0rhj/duck\\_duck\\_debug/](https://www.reddit.com/r/ProgrammerHumor/comments/nn0rhj/duck_duck_debug/)