

# Informática Aplicada I

## Técnicas de debugging

Alonso Ramírez  
Alejandro Guerrero

CIMAT

September 17, 2014

## Dudas?

- Dudas de `if,else,for, while`?
- Dudas de `funciones`?
- Dudas de `Arreglos`?
- Dudas de `Apuntadores`?
- Dudas en general?

## Debugging: Qué es?

Son estrategias que sirven para encontrar un error en el código en tiempo de **compilación** o **ejecución**, popularmente llamado **bug**

## Cuáles son?

Son los errores que el compilador detecta al intentantar compilar el programa. Usualmente son más simples de encontrar y corregir pues se refieren comúnmente a la sintaxis utilizada.

## Errores comunes

- Olvidar poner un ; al final de una instrucción.
- Olvidar incluir una librería.
- Escribir mal los parámetros de una función.

## El compilador

Para este tipo de errores el **compilador** se encarga de mostrarnos los **errores** y las **alertas (warnings)** que tiene nuestro código.

## Warnings comunes

Sirven para prevenir **algunos** posibles errores que tendremos en tiempo de ejecución y que causaran un mal funcionamiento de nuestro programa, como por ejemplo

- Que una función tipo int no regrese nada.
- Que declaremos una variable que nunca utilizamos.

## Ejemplos ilustrativos

Analicemos algunos casos usuales (ver archivos en carpeta "ilustrativos").

## Tiempo de compilación

- El compilador es tu amigo, hacerle caso.
- Al momento de abrir {},[],() siempre abrir y cerrarlos inmediatamente y LUEGO escribir el código dentro de ellos.

## Tiempo de ejecución

- Asegurarse de siempre **inicializar** variables antes de usarlas.
- Agrupar el código en **funciones concretas** y probar cada una antes de continuar con el código.
- Imprimir las variables paso por paso siguiendo el **flujo del programa** (combinación printf-return por ejemplo).

## Tiempo de compilación

Poner en práctica lo anterior con ejercicios (ver carpeta "compilacion").



## Tiempo de ejecución

Poner en práctica lo anterior con ejercicios (ver carpeta "ejecucion").