

# Introducción a la programación en C

Entrada y salida

---

Bach. Constantino Bolaños Araya

Escuela de Veranillo en HPC, 2018

# **Entrada y salida**

- Entrada: todos los datos que ingresan a un programa, sea desde el usuario o del entorno de ejecución .
- Salida: todos los datos que se obtienen de un programa, por medio de algún periférico (monitor, red, impresora, parlantes, ...) o algún archivo.

**Entrada del / salida al usuario**

- `int getchar()` lee un caracter ingresado por el usuario y lo devuelve como un entero.
- `char *gets(buf)` lee una línea ingresada por el usuario y la guarda en una variable búfer (en este caso `buf`).
- `int scanf(const char *format, ...)` lee la entrada del usuario y extrae valores de esta entrada según el formato provisto

- `int putchar(int c)` pone un caracter en pantalla.
- `int puts(const char *s)` escribe la hilera `s`, seguida de un cambio de línea.
- `int printf(const char *format, ...)` escribe una hilera formateada, llenando los campos con los valores provistos.

## **Entrada y salida de archivos**

# Abrir un archivo

- Para abrir un archivo, se usa la siguiente función:

```
FILE *fopen(const char *filename, const char *mode);
```

- filename es el nombre del archivo que se desea abrir.
- mode puede tener uno de los siguientes valores:
  - 'r': sólo lectura
  - 'w': sólo escritura. Si archivo no existe, se crea. Si existe, se sobrescribe.
  - 'a': sólo escritura. Si archivo no existe, se crea. Si existe, empieza a escribir al final del mismo.
  - 'r+': lectura y escritura.
  - 'w+': similar a 'w', pero también con lectura.
  - 'a+': similar a 'a', pero también con lectura.
- A todos estos modos se les puede agregar la letra 'b', indicando que el tipo de lectura / escritura es en binario.



- Para cerrar un archivo, se usa la función:

```
int fclose(FILE *fp);
```

- Funciones para escritura:
  - `int fputc(int c, FILE *fp)`
  - `int fputs (const char *s, FILE *fp)`
  - `int fprintf(FILE *fp, const char *format, ...)`
- Funciones para lectura:
  - `int fgetc(FILE *fp)`
  - `char *fgets (char *buf, int n, FILE *fp)`
  - `int fscanf(FILE *fp, const char *format, ...)`