



## **Algoritmos y Programación 1**

—  
**Cátedra: Kuhn**  
—

***Trabajo Práctico N° 1 | 20/09/2017 | 1<sup>er</sup> Entrega***  
—

### **Alumnos:**

Mauricio Cabrera	101334	maurihuergo@hotmail.com
Aurelien Floutard	101999	aure.flout@hotmail.fr
Elías Urquiza	100714	eliasurquiza07@gmail.com

# Calculador de desempeño académico

95.11/75.02 Algoritmos y Programación I

6 de septiembre de 2017

## 1. Objetivo del TP

El objetivo del presente trabajo consiste en la realización de un conjunto de aplicaciones en modo consola, escritos en lenguaje ANSI-C89, que permitan implementar un sistema de cálculo de métricas sobre el desempeño académico del usuario.

## 2. Alcance del TP

Mediante el presente TP se busca que el estudiante adquiera y aplique conocimientos sobre los siguientes temas:

- Programas en modo consola
- Directivas al preprocesador C
- Juego de caracteres ASCII
- Tipos enumerativos
- Control de flujo
- Salida de datos con formato
- Funciones
- Modularización
- Arreglos
- Cadenas de caracteres

**Fecha de entrega: 20 de septiembre de 2017**

### 3. Introducción

Es común entre los alumnos llevar una cierta medida de cómo el mismo se va desempeñando a lo largo de la carrera. Para esta tarea, se pide desarrollar una aplicación que, cumpliendo con las especificaciones de la sección 4, pueda mostrar una serie de datos según la información provista por el alumno.

### 4. Especificaciones

El programa a desarrollar debe ser interactivo. Debe obtener sus datos a través del flujo de entrada estándar, `stdin`. Deberá consultar al usuario por los siguientes datos: nombre, apellido, padrón, carrera y asignaturas. La cantidad de materias es desconocida y serán ingresadas de a una, con su nombre y nota. Además, debe ser posible compilarlo en diferentes idiomas.

Ante cada error en el ingreso de los datos, el programa debe indicar al usuario, con un mensaje *claro*, la falla.

#### 4.1. Menú principal

Para la toma de datos, el programa se valdrá de un menú, similar al siguiente ejemplo:

```
Elija qué desea hacer:
  R) Modificar registro
      personal
  A) Modificar asignatura
  M) Calcular métrica
      individual (opcional)
5  !) Finalizar
    0) Salir
?
```

Ejemplo 1: Español

```
Velg hva du skal gjøre:
  P) Endre
      personopplysninger
  F) Endre fag
  B) Beregn individuelle
      beregninger
5  !) Slutt
    0) Avslutt
?
```

Ejemplo 2: Idioma 2

Cada uno de los ítems del menú puede o no tener un submenú en el que debe actuar. Nótese que los caracteres de selección pueden ser variados: alfabéticos, símbolos y/o numéricos. A continuación se detallan las operaciones de cada uno de estos ítems.

#### 4.2. Submenú de registro personal

En este menú se pide información para completar lo que llamaremos “registro personal”. Estos datos abarcan: nombre, apellido, padrón y carrera.

La selección del ítem a completar depende de un nuevo menú, a saber:

```
Elija qué desea modificar:
  A) Apellido y nombre
  #) Padron
  C) Carrera
```

```

5      0) Volver
      ?

```

Al seleccionar una opción, se mostrará el dato cargado, se solicitará el nuevo dato, y se registrará el mismo, sin pedido de confirmación.

### 4.3. Submenú de asignaturas

En este menú se presenta la información correspondiente a las asignaturas y se registran cambios y nuevos datos. Un ejemplo de ejecución de este menú es:

```

Usted no tiene asignaturas cargadas.
+ ) Ingresar nueva asignatura
0 ) Volver
?

```

ó bien

```

Opciones:
1 ) AMII (9)
2 ) Química (8)
3 ) Inglés (10)
5  4 ) Probabilidad y estadística (9)
    5 ) Sociología (10)
    + ) Ingresar nueva asignatura
    - ) Eliminar una asignatura (opcional)
10 0 ) Volver
    ?

```

Nótese la particularidad que tiene este menú de modificarse según el estado de ejecución del programa.

### 4.4. Submenú de métricas (opcional)

En este menú se da la opción al usuario de calcular alguna métrica con el estado actual del programa y presenta un menú similar al siguiente:

```

Métricas:
P ) Promedio
M ) Máximo
m ) Mínimo
5  # ) Cantidad de materias
    2 ) Aplazos
    0 ) Volver
?

```

Seleccionada una opción, el programa indica al usuario dicho resultado.

### 4.5. Submenú de finalización

Este submenú no mostrará un nuevo menú. En el mismo se computarán todos las métricas y se imprimirá un mensaje, por una salida estándar diferente a la utilizada en los menús. El mensaje debe respetar el siguiente formato:

```
Nombre y Apellido , Padrón , Carrera , Materias , Promedio , Aplazos
```

Todo el mensaje contenido en una única línea. Luego se volverá al inicio de la aplicación, con todos los datos eliminados.

#### 4.6. Submenú “Salir”

Termina el ciclo de ejecución, sin mostrar mensajes.

### 5. Opcionales

Se consideran puntos opcionales, la posibilidad de eliminar asignaturas ya cargadas de la lista, y la posibilidad de computar cada una de las diferentes métricas a pedido del usuario.

Se puede agregar, como opcional, que el cada uno de los caracteres que se utilizan en las selecciones no varíe con el idioma. De hacer este opcional, se considerará válido si la implementación es correcta, ya que no es correcto mantenerlos a través de hardcodes o modularizaciones pobres.

### 6. Ejemplo de ejecución

```
$ ./mi_aplicacion
Bienvenidx
Elija qué desea hacer:
  R) Modificar registro personal
5  A) Modificar asignatura
  M) Calcular métrica individual
  !) Finalizar
  O) Salir
? R
10 Elija qué desea modificar:
   A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
15 ? N
   No tiene nombre cargado.
   ¿Apellido y nombre? Fulanito Cosme
   Elija qué desea modificar:
20   A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
   ? #
   ¿Padron? 123456
25 Elija qué desea modificar:
   A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
30 ? C
   ¿Codigo de la carrera? 7
   Elija qué desea modificar:
35   A) Apellido y nombre
   #) Padron
   C) Carrera
   O) Volver
   ? O
   Elija qué desea hacer:
```

```

40      R) Modificar registro personal
      A) Modificar asignatura
      M) Calcular métrica individual
      !) Finalizar
      O) Salir
? A
45 Usted no tiene asignaturas cargadas.
Opciones:
    +) Ingresar nueva asignatura
    O) Volver
? +
50 ¿Asignatura (Nota Nombre)? 9 AMII
Opciones:
    1) AMII (9)
    +) Ingresar nueva asignatura
    O) Volver
55 ? 0
Elija qué desea hacer:
    R) Modificar registro personal
    A) Modificar asignatura
    M) Calcular métrica individual
60    !) Finalizar
    O) Salir
? !
Fulanito Cosme,123456,7,1,9,0
Elija qué desea hacer:
65    R) Modificar registro personal
    A) Modificar asignatura
    M) Calcular métrica individual
    !) Finalizar
    O) Salir
70 ? 0
$

```

## 7. Comentarios

Si bien no es necesario utilizar variables globales en este trabajo, con su debida **justificación**, *algunas* pueden ser aceptadas.

Es recomendable tener un diccionario construido con las carreras como cadenas de caracteres y un tipo enumerado; lo mismo que los mensajes de error y el enumerando `status_t`. Este enumerando, en español representa una *condición*.

El nombre del enumerando que representa *estado* en el que se encuentra el programa (menú principal, algún submenú, error, inicio, reinicio, etc.) en inglés es `state_t`.

## 8. Restricciones

La realización del trabajo se encuentra sujeta a las siguientes restricciones:

- Debe realizarse en grupos de 3 (**tres**) integrantes.
- No está permitida la utilización de memoria dinámica.

- No está permitida la utilización de funciones específicas para manejo de archivos. Como funciones de E/S sólo pueden ser utilizadas las funciones de biblioteca `fgetc()`, `getchar()`, `scanf()`, `printf()`, `fprintf()`, `putc()`, `fputc()`, `puts()`, `fputs()` o similares.
- No está permitida la utilización de `gets()`<sup>1</sup>, `fflush(stdin)`<sup>2</sup>, la biblioteca `conio.h`<sup>3</sup> (`#include <conio.h>`), etc.
- Debe recurrirse a la utilización de funciones mediante una adecuada parametrización.
- No está permitido en absoluto tener hard-codings:

```

1 ...
2 char c;
3 ...
4 if (c == 'R')                               /* ; i hard-coded!! */
5     printf("%s", "xxxxxxxx");               /* ; i hard-coded!! */
6 else if (c == 'A')                          /* ; i hard-coded!! */
7     printf("%s", "yyyyyyyy");              /* ; i hard-coded!! */
8 ...

```

sino que debe recurrirse al uso de ETIQUETAS, CONSTANTES SIMBÓLICAS, MACROS, etc.

Los ejemplos no son exhaustivos, sino que existen otros hard-codings y tampoco son aceptados.

- Hay ciertas cuestiones que no han sido especificadas intencionalmente en este Requerimiento, para darle al/la desarrollador/a la libertad de elegir implementaciones que, según su criterio, resulten más convenientes en determinadas situaciones. Por lo tanto, se debe explicitar cada una de las decisiones adoptadas, y el o los fundamentos considerados para las mismas.

## 9. Entrega del Trabajo Práctico

La fecha de entrega del trabajo práctico es: 20 de septiembre de 2017 o antes.

No se requiere entrega en papel. Deberá realizarse una entrega digital, a través del campus de la materia, de un único archivo cuyo nombre debe seguir el siguiente formato:

YYYYMMDD\_apellido1-apellido2-apellido3\_entrega-N.tar.gz

donde YYYY es el año (2017), MM el mes y DD el día en que uno de los integrantes sube el archivo, apellido-1a3 son los apellidos de los integrantes ordenados alfabéticamente, entrega-N indica el número de vez que se envía el trabajo (entrega-1, entrega-2, etc.), y .tar.gz es la extensión, que no necesariamente es .tar.gz.

<sup>1</sup>obsoleta en C99 [3], eliminada en C11 [4] por fallas de seguridad en su uso.

<sup>2</sup>comportamiento indefinido para flujos de entrada ([3],[4]). Definida en estándar POSIX.

<sup>3</sup>biblioteca no estándar, con diferentes implementaciones y licencias, y no siempre disponible.

El archivo comprimido debe contener los siguientes elementos:

- La correspondiente documentación de desarrollo del TP (en formato pdf), siguiendo la numeración siguiente, incluyendo:
  1. Carátula del TP. Incluir una dirección de correo electrónico.
  2. Enunciado del TP.
  3. Estructura funcional de los programas desarrollados.
  4. Explicación de cada una de las alternativas consideradas y las estrategias adoptadas.
  5. Resultados de la ejecución (corridas) de los programas, captura de las pantallas, bajo condiciones normales e inesperadas de entrada.
  6. Reseña sobre los problemas encontrados en el desarrollo de los programas y las soluciones implementadas para subsanarlos.
  7. Bibliografía (ver aparte).
  8. Indicaciones sobre la compilación de lo entregado para generar la aplicación.

**NOTA:** Si la compilación del código fuente presenta mensajes de aviso (warning), notas o errores, los mismos deben ser comentados en un apartado del informe.

**NOTA:** El Informe deberá ser redactado en *correcto* idioma castellano.

- Códigos fuentes en formato de texto plano (.c y .h), *debidamente documentados*.

**NOTA:** Todos los integrantes del grupo deben subir el *mismo* archivo.

**NOTA:** Se debe generar y subir un único archivo (comprimido) con todos los elementos de la entrega digital. **NO usar RAR.** La compresión RAR no es un formato libre, en tanto sí se puede utilizar *ZIP*, *GUNZIP*, u otros (soportados, por ejemplo, por la aplicación de archivo *TAR*).

Si no se presenta cada uno de estos ítems, será rechazado el TP.

## 10. Bibliografía

Debe incluirse la referencia a toda bibliografía consultada para la realización del presente TP: libros, artículos, URLs, etc., citando:

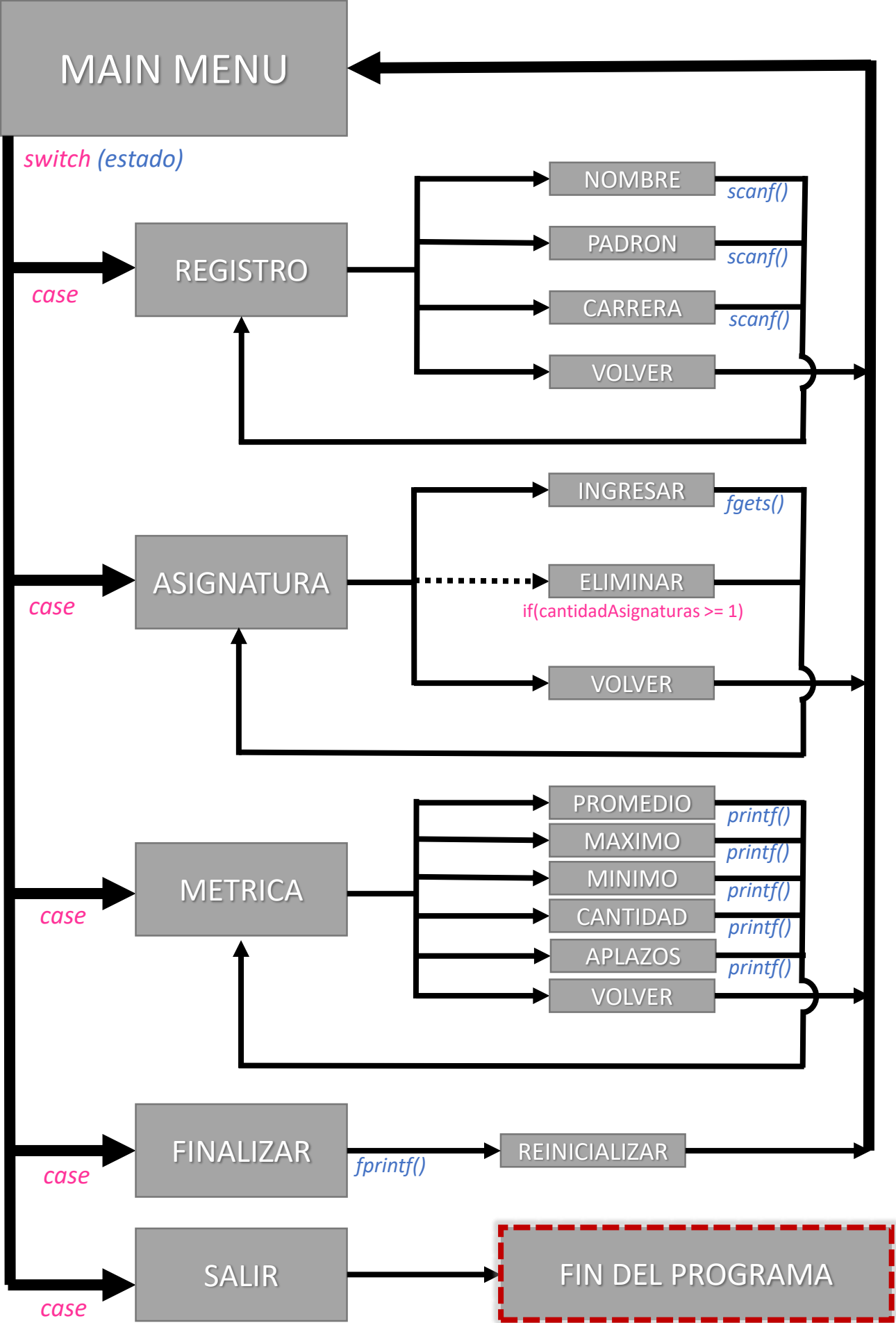
- Denominación completa del material (Título, Autores, Edición, Volumen, etc.).
- Código ISBN del libro (opcional: código interbibliotecario).
- URL del sitio consultado. No poner [Wikipedia.org](http://Wikipedia.org) o [stackexchange.com](http://stackexchange.com), sino que debe incluirse un enlace al artículo, hilo, etc. consultado.

Utilizando  $\text{\LaTeX}$ , la inclusión de citas/referencias es trivial. Los editores de texto gráficos de las suites de ofimática, como LibreOffice Write o MS Word, admiten plugins que facilitan la inclusión.



## Ejemplo de referencias

- [1] B.W. Kernighan y D.M. Ritchie. *The C Programming Language*. 2.<sup>a</sup> ed. Prentice-Hall software series. Prentice Hall, 1988. ISBN: 9780131103627.
- [2] P. Deitel y H. Deitel. *C How to Program*. 7.<sup>a</sup> ed. Pearson Education, 2012. ISBN: 9780133061567.
- [3] ISO/IEC. *Programming Languages – C*. ISO/IEC 9899:1999(E). ANSI, dic. de 1999, págs. 270-271.
- [4] ISO/IEC. *Programming Languages – C*. INCITS/ISO/IEC 9899:2011. INCITS/ISO/IEC, 2012, pág. 305.



## **Explicación de alternativas consideradas y estrategias adoptadas**

### **Etapas Iniciales de Desarrollo:**

Primeramente, se diseñó una estructura “base” del programa, que en sí constaba únicamente de un menú principal, con un while-loop infinito que utiliza un switch(estado) para alternar entre distintos mensajes según el estado en que se encontrara (con la idea de eventualmente reemplazar estos mensajes por funciones pertenecientes al submenú respectivo a la opción) y la última opción permite salir del while-loop con un “break;”, permitiendo este que main(void) retorne un valor de salida exitosa.

Dicha estructura está todavía presente en el programa final, aunque fue modificada para dar inicio a variables y matrices que son utilizadas en todo el programa.

Después de trasladar las constantes definidas hasta ese momento a un main.h, los integrantes del grupo nos dividimos la tarea de encargarnos de un sub-menú cada uno y trabajar equitativamente en main.h y main.c según fuese necesario.

Durante el desarrollo del primer sub-menú, el sub-menú Registro, se decidió por trabajar utilizando punteros (mejor dicho, utilizando un leve entendimiento de los mismos pero lo suficiente para permitir pasaje y modificación de variables por referencia). Sin embargo, durante el desarrollo del mismo, se decidió cambiar el diseño del programa por uno que utilizara una estructura de datos para simplificar el pasaje de variables y la impresión de los datos al final. Por este motivo, uno de los integrantes trabajó en la estructura y la incorporación de la misma al menú principal en main.c y el código original del sub-menú registro fue modificado para incorporar este nuevo acercamiento al desarrollo del programa.

[Para capturas de pantalla del código original, refiérase a la sección 6 – Reseña Sobre Problemas Encontrados y Soluciones Implementadas ]

### **Etapas Media de Desarrollo:**

Con la nueva estructura de datos, procedimos a incorporar lo necesario para que todos los sub-menús planeados pudieran ser desarrollados, y, al mismo tiempo, empezamos a trasladar porciones de main.h a un nuevo archivo llamado espanol.h, el cual formaría a ser parte de 3 archivos de cabecera encargados de los 3 idiomas distintos que incluye el programa final. Alrededor de este momento en la etapa de desarrollo, el sub-menú Asignaturas seguía encontrando nuevos problemas y nuevas soluciones a los mismos, los cuales son desarrollados en mayor detalle en la Sección 6, pero que, en resumen, correspondían a problemas para poder modificar el menú con cada asignatura ingresada, poder validar los datos, y además poder modificar y borrar las asignaturas no deseadas.

Para no perder tiempo de desarrollo, terminamos el sub-menú Métricas antes que el sub-menú Asignaturas y empezamos a migrar las funciones que hasta ese momento estaban contenidas en main.c a un nuevo archivo de código fuente “funciones.c”.

## Etapas Finales de Desarrollo:

Con el programa casi terminado, empezamos la traducción parcial del programa a los otros dos idiomas planeados: inglés y francés.

Para ello, realizamos copias de `espanol.h` y modificamos las constantes ahí contenidas.

Para entonces, el sub-menú asignaturas estaba completo salvo por una gran limitación que no le permitía el ingreso de mas de 10 materias. A pesar de esto, se incorporó el mismo al programa, se compilaron ejecutables de ensayo y se realizaron pruebas en busca de errores/"Bugs".

Finalmente, se resolvió el problema del menu asignaturas y se hicieron los ultimos chequeos para asegurarse que lo que imprime el sub-menú finalizar saliera por `stderr` y no por `stdout`.

Con el programa terminado, prosiguió una etapa de testeo y se dio por finalizado el desarrollo del mismo.

## Corriendo el programa en condiciones normales (Español)

```

¡Bienvenido!
Elija que desea hacer:
    R) Registro Personal
    A) Asignaturas
    M) Calcular métrica individual
    !) Finalizar
    Ø) Salir
? R
Menu Registro. Elija una opción para modificar los datos ingresados.
    A) Apellido y Nombre
    #) Padrón
    C) Carrera
    Ø) Volver
? A
Ingrese Apellido: Cabrera
Ingrese Nombre: Mauricio
Queda registrado lo siguiente: Cabrera, Mauricio
    A) Apellido y Nombre
    #) Padrón
    C) Carrera
    Ø) Volver
? .
ERROR: Opción ingresada no válida, intenta de nuevo:
#
Ingrese su padrón: 101334
Queda registrado lo siguiente: 101334
    A) Apellido y Nombre
    #) Padrón
    C) Carrera
    Ø) Volver
? C
Ingrese el número correspondiente a su carrera: 16
Queda registrado lo siguiente: Número de Carrera Inválido.
    A) Apellido y Nombre
    #) Padrón
    C) Carrera
    Ø) Volver
? C
Ingrese el número correspondiente a su carrera: 7
Queda registrado lo siguiente: Ingeniería Electrónica
    A) Apellido y Nombre
    #) Padrón
    C) Carrera
    Ø) Volver
? Ø
Elija que desea hacer:
    R) Registro Personal
    A) Asignaturas
    M) Calcular métrica individual
    !) Finalizar
    Ø) Salir
? M
Métrica:
    P) Promedio
    M) Máximo
    m) Mínimo
    #) Cantidad de materias
    A) Aplazos
    Ø) Volver
? Ø
Elija que desea hacer:
    R) Registro Personal
    A) Asignaturas
    M) Calcular métrica individual

```

```

Elija que desea hacer:
    0) MateriaA (7)
    -> Eliminar una asignatura.
    +> Ingreso de asignatura.
    !> Salir.
? +

Ingrese la asignatura: MateriaB
Ingrese la nota: 5

Elija que desea hacer:
    0) MateriaA (7)
    1) MateriaB (5)
    -> Eliminar una asignatura.
    +> Ingreso de asignatura.
    !> Salir.
? +

Ingrese la asignatura: MatericaC
Ingrese la nota: 9

Elija que desea hacer:
    0) MateriaA (7)
    1) MateriaB (5)
    2) MatericaC (9)
    -> Eliminar una asignatura.
    +> Ingreso de asignatura.
    !> Salir.
? +

Ingrese la asignatura: MateriaD
Ingrese la nota: 10

Elija que desea hacer:
    0) MateriaA (7)
    1) MateriaB (5)
    2) MatericaC (9)
    3) MateriaD (10)
    -> Eliminar una asignatura.
    +> Ingreso de asignatura.
    !> Salir.
? +

Ingrese la asignatura: MateriaF
Ingrese la nota: 6

Elija que desea hacer:
    0) MateriaA (7)
    1) MateriaB (5)
    2) MatericaC (9)
    3) MateriaD (10)
    4) MateriaF (6)
    -> Eliminar una asignatura.
    +> Ingreso de asignatura.
    !> Salir.
? !
Elija que desea hacer:
    R) Registro Personal
    A) Asignaturas
    M) Calcular métrica individual
    !> Finalizar
    0) Salir
? M

Métrica:
    P) Promedio
    M) Máximo

```

```

Metrica:
  P> Promedio
  M> Maximo
  m> Minimo
  #> Cantidad de materias
  A> Aplazos
  Ø> Volver
? P
El promedio es de: 7.4
Metrica:
  P> Promedio
  M> Maximo
  m> Minimo
  #> Cantidad de materias
  A> Aplazos
  Ø> Volver
? M
La asignatura con la mejor nota es: MateriaD <10>
Metrica:
  P> Promedio
  M> Maximo
  m> Minimo
  #> Cantidad de materias
  A> Aplazos
  Ø> Volver
? m
La asignatura con la peor nota es: MateriaB <5>
Metrica:
  P> Promedio
  M> Maximo
  m> Minimo
  #> Cantidad de materias
  A> Aplazos
  Ø> Volver
? A
La cantidad de aplazos es: 0
Metrica:
  P> Promedio
  M> Maximo
  m> Minimo
  #> Cantidad de materias
  A> Aplazos
  Ø> Volver
? Ø
Elija que desea hacer:
  R> Registro Personal
  A> Asignaturas
  M> Calcular métrica individual
  !> Finalizar
  Ø> Salir
? !
Mauricio Cabrera, 101334, Ingeniería Electrónica, 5, 7.40, 0
Elija que desea hacer:
  R> Registro Personal
  A> Asignaturas
  M> Calcular métrica individual
  !> Finalizar
  Ø> Salir
?

```

(Nótese que hay un par de fallos por el tema de la acentuación y el reconocimiento del mismo)

## Corriendo el programa en condiciones normales (Inglés)

```

Welcome!
Please, pick an option:
    R) Personal Registry
    C) Courses
    M) Calculate Individual Metrics
    !) End Session
    0) Exit
? R
Registry Menu. Pick an option to modify the logged data.
    N) Full Name
    #) College ID
    C) Career
    0) Go Back
? N
Insert Surname: Cabrera
Insert Name: Mauricio
Registered:: Cabrera, Mauricio
    N) Full Name
    #) College ID
    C) Career
    0) Go Back
? #
Insert College ID: 101334
Registered:: 101334
    N) Full Name
    #) College ID
    C) Career
    0) Go Back
? C
Insert your career's corresponding number: 13
Registered:: Invalid Career Number.
    N) Full Name
    #) College ID
    C) Career
    0) Go Back
? +
ERROR: Invalid option. Please, try again:
213
ERROR: Invalid option. Please, try again:
?
ERROR: Invalid option. Please, try again:
Please, pick an option:
    R) Personal Registry
    C) Courses
    M) Calculate Individual Metrics
    !) End Session
    0) Exit
? C
Please, pick an option:
    +) Add a Course
    !) Exit
? +
Insert your course: Analisis
Insert your course grade: 7
Please, pick an option:
    0) Analisis (? )
    -) Delete Course
    +) Add a Course
    !) Exit
? +
Insert your course: Quimica
Insert your course grade: 5

```



```

Insert your course: Quimica
Insert your course grade: 5

Please, pick an option:
    0) Analisis (7)
    1) Quimica (5)
    -) Delete Course
    +) Add a Course
    !) Exit
? +

Insert your course: Algebra
Insert your course grade: 2

Please, pick an option:
    0) Analisis (7)
    1) Quimica (5)
    2) Algebra (2)
    -) Delete Course
    +) Add a Course
    !) Exit
? +

Insert your course: Fisica
Insert your course grade: 10

Please, pick an option:
    0) Analisis (7)
    1) Quimica (5)
    2) Algebra (2)
    3) Fisica (10)
    -) Delete Course
    +) Add a Course
    !) Exit
? !
Please, pick an option:
    R) Personal Registry
    C) Courses
    M) Calculate Individual Metrics
    !) End Session
    0) Exit
? M

Pick which metric to measure:
    A) Average
    M) Maximum
    m) Minimum
    #) Amount of Courses
    F) Failed Courses
    0) Go Back
? A
Your average is: 6.0

Pick which metric to measure:
    A) Average
    M) Maximum
    m) Minimum
    #) Amount of Courses
    F) Failed Courses
    0) Go Back
? M
The course with the highest note is: Fisica (10)

Pick which metric to measure:
    A) Average
    M) Maximum
    m) Minimum

```

```
? #
The amount of courses you've taken is: 4
Pick which metric to measure:
    A) Average
    M) Maximum
    m) Minimum
    #) Amount of Courses
    F) Failed Courses
    0) Go Back
? F
Amount of failed courses: 1
Pick which metric to measure:
    A) Average
    M) Maximum
    m) Minimum
    #) Amount of Courses
    F) Failed Courses
    0) Go Back
? 0
Please, pick an option:
    R) Personal Registry
    C) Courses
    M) Calculate Individual Metrics
    !) End Session
    0) Exit
? !
Mauricio Cabrera, 101334, Invalid Career Number., 4, 6.00, 1
Please, pick an option:
    R) Personal Registry
    C) Courses
    M) Calculate Individual Metrics
    !) End Session
    0) Exit
? _
```

## Corriendo el programa en condiciones normales(Francés)

```

Bienvenue !
Choisissez ce que vous voulez faire :
    R) Register personnel
    M) Matipres
    C) Calcul de données personnelles
    !) Finaliser
    Ø) Sortir
? R
Menu register, Choisissez une option pour modifier les données rentrées.
    N) Nom et prénom
    #) Padron
    D) Département
    Ø) Sortir
? N
Rentrer votre nom: Cabrera
Rentrer votre prénom: Mauricio
Donnée enregistrée: Cabrera, Mauricio
    N) Nom et prénom
    #) Padron
    D) Département
    Ø) Sortir
? #
Rentrer votre padron: hola
ERREUR: Le padron ne peut pas être lu.
Option invalide. Merci de réessayer : 101334
Donnée enregistrée: 101334
    N) Nom et prénom
    #) Padron
    D) Département
    Ø) Sortir
? D
Rentrer le numéro correspondant à votre département.: ?
Donnée enregistrée: Ingenieur Electronique
    N) Nom et prénom
    #) Padron
    D) Département
    Ø) Sortir
? Ø
Choisissez ce que vous voulez faire :
    R) Register personnel
    M) Matipres
    C) Calcul de données personnelles
    !) Finaliser
    Ø) Sortir
? M
Choisissez ce que vous voulez faire :
    +) Ajouter une matipre
    !) Sortir.
? +
Entrer le matipre: Algoritmos
Entrer le note: 9
Choisissez ce que vous voulez faire :
    Ø) Algoritmos (9)
    -) Supprimer une matipre.
    +) Ajouter une matipre
    !) Sortir.
? +
Entrer le matipre: Algebra
Entrer le note: 2
Choisissez ce que vous voulez faire :
    Ø) Algoritmos (9)
    1) Algebra (2)

```

```

Entrer le matibre: Algebra
Entrer le note: 2

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    -) Supprimer une matibre.
    +) Ajouter une matibre
    !) Sortir.
? +

Entrer le matibre: Organica
Entrer le note: 10

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    -) Supprimer une matibre.
    +) Ajouter une matibre
    !) Sortir.
? +

Entrer le matibre: Fisica
Entrer le note: 5

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    3) Fisica (5)
    -) Supprimer une matibre.
    +) Ajouter une matibre
    !) Sortir.
? -

Entrer le matibre: 2
Entrer le note: Geometria
ERREUR: La note n'est pas valide. Merci de réessayer5

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    3) Fisica (5)
    4) 2 (5)
    -) Supprimer une matibre.
    +) Ajouter une matibre
    !) Sortir.
? 4

Entrer le matibre: Geometria
Entrer le note: 6

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    3) Fisica (5)
    4) 2 (5)
    5) Geometria (6)
    -) Supprimer une matibre.
    +) Ajouter une matibre
    !) Sortir.
? -

Entrer le matibre: 3

```

```

Entrer le matiPre: Geometria
Entrer le note: 6

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    3) Fisica (5)
    4) 2 (5)
    5) Geometria (6)
    -> Supprimer une matiPre.
    +> Ajouter une matiPre
    !> Sortir.
? -

Entrer le matiPre: 3
Entrer le note: 4

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    3) Fisica (5)
    4) 2 (5)
    5) Geometria (6)
    6) 3 (4)
    -> Supprimer une matiPre.
    +> Ajouter une matiPre
    !> Sortir.
? +

Entrer le matiPre: j
Entrer le note: 6

Choisissez ce que vous voulez faire :
    0) Algoritmos (9)
    1) Algebra (2)
    2) Organica (10)
    3) Fisica (5)
    4) 2 (5)
    5) Geometria (6)
    6) 3 (4)
    7) j (6)
    -> Supprimer une matiPre.
    +> Ajouter une matiPre
    !> Sortir.
? !

Choisissez ce que vous voulez faire :
    R) Registre personnel
    M) Matipres
    C) Calcul de données personnelles
    !> Finaliser
    0) Sortir
? C

Données :
    %> Moyenne
    M> Maximum
    m> Minimum
    #> Nombre de matipres
    E> Echec
    0> Sortir
? %

La moyenne est de : 5.9

Données :
    %> Moyenne

```

```
? %  
La moyenne est de : 5.9  
Données :  
  %> Moyenne  
  M> Maximum  
  m> Minimum  
  #> Nombre de matièrres  
  E> Echec  
  Ø> Sortir  
? M  
La matièrre ou vous avez la meilleure bonne note est : Organica <10>  
Données :  
  %> Moyenne  
  M> Maximum  
  m> Minimum  
  #> Nombre de matièrres  
  E> Echec  
  Ø> Sortir  
? m  
La matièrre ou vous avez la moins bonne note est : Algebra <2>  
Données :  
  %> Moyenne  
  M> Maximum  
  m> Minimum  
  #> Nombre de matièrres  
  E> Echec  
  Ø> Sortir  
? E  
La nombre de matièrre(s) échouée(s) : 1  
Données :  
  %> Moyenne  
  M> Maximum  
  m> Minimum  
  #> Nombre de matièrres  
  E> Echec  
  Ø> Sortir  
? _
```

## Transición al uso de “structs”:

Durante la construcción del sub-menú registro, la idea original era utilizar punteros para dos variables e incluir dos vectores “nombre[]” y “apellido[]” y asimismo una matriz “carreras[][]” tal que el usuario modificara lo que se encontraba en las direcciones de memoria de las variables “padrón” y “num\_carrera”. El sub-menú registro funcionaba adecuadamente para esta situación, aunque se encontraba en un estado poco desarrollado respecto al desplazamiento por el menú.

En este momento, por decisión de la mayoría del grupo y tras consultar si era un método viable, pasamos a utilizar una estructura de datos “usuario\_t” tal que el sub-menú Asignaturas fuera más fácil, concluimos, de pasar y recibir datos tal como lo requiere el programa.

Aunque una solución utilizando punteros y arreglos era igual de viable como en el caso del sub-menú Registro, los miembros del grupo concluyeron que utilizar una estructura sería mas sencillo de escribir y la legibilidad del código sería mayor.

Sin Structs (Únicamente referente al Sub-menú Registro):

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
#include "main.h"

int menu_registro(int *ptr_padron, int *ptr_num_carrera, char carrera[][30], char apellido[], char nombre[]);

void imprimir_carrera(int fila, char carrera[][30]);

int main(void)
{
    /*Falta Eliminar Hardcodeo*/
    estado main estado = MAIN_MENU;
    int input_i = 0;
    char apellido[50];
    char nombre[50];
    int padron, num_carrera;
    char carrera[12][30] = {{ING_0}, {ING_1}, {ING_2}, {ING_3}, {ING_4}, {ING_5}, {ING_6}, {ING_7}, {ING_8}, {ING_9}, {ING_10}, {ING_11}};

```

```

        case MENU_REGISTRO:
        {
            input_i = menu_registro(&padron, &num_carrera, carrera, apellido, nombre);

            if(input_i == EXIT_SUCCESS)
                estado = MAIN_MENU;
            else
            {
                return EXIT_FAILURE;
            }

            printf("PRUEBA: PADRON: %i CARRERA: %i\n", padron, num_carrera);

            break;
        }

```

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
void imprimir_carrera(int fila, char carrera[][30])
{
    printf("%s\n", carrera[fila]);
}

int menu_registro(int *ptr_padron, int *ptr_num_carrera, char carrera[][30], char apellido[], char nombre[])
{
    int input_i = 0;
    puts(MSG_REGISTRO);
    while(1)
    {
        printf("1) %s\n2) %s\n3) %s\n0) %s\n", REGISTRO_OPCION_NOMBRE, REGISTRO_OPCION_PADRON, REGISTRO_OPCION_CARRERA, OPCION_VOLVER);

        if(scanf("%i", &input_i) != 1)
        {
            fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_OPCIONES);
            return EXIT_FAILURE;
        }
        while(getchar() != '\n');

        if(input_i == 1)
        {
            printf("%s: ", REGISTRO_ING_APELLIDO);
            if(scanf("%s", apellido) != 1)
            {
                fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_REG_NOMBRE);
                return EXIT_FAILURE;
            }
            while(getchar() != '\n');

            printf("%s: ", REGISTRO_ING_NOMBRE);

            if(scanf("%s", nombre) != 1)
            {
                fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_REG_NOMBRE);
                return EXIT_FAILURE;
            }
            while(getchar() != '\n');

            printf("%s: %s, %s\n", REGISTRO_ING_AVIS0, apellido, nombre);
        }
        else if(input_i == 2)
        {
            printf("%s: ", REGISTRO_ING_PADRON);
            if(scanf("%i", ptr_padron) != 1)
            {
                fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_REG_PADRON);
                return EXIT_FAILURE;
            }
            while(getchar() != '\n');
            printf("%s: %i\n", REGISTRO_ING_AVIS0, *ptr_padron);
        }
        else if(input_i == 3)
        {
            printf("%s: ", REGISTRO_ING_CARRERA);
            if(scanf("%i", ptr_num_carrera) != 1)
            {
                fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_REG_CARRERA);
                return EXIT_FAILURE;
            }
            while(getchar() != '\n');

            printf("%s: ", REGISTRO_ING_AVIS0);
            imprimir_carrera(*ptr_num_carrera, carrera);
        }
        else if(input_i == 0)
        {
            break;
        }
        else
        {
            fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_OPCIONES);
            return EXIT_FAILURE;
        }
    }

    return EXIT_SUCCESS; /* Solo se sale exitosamente si se activa el break; */
}

```

209,1 Final

[La versión con Structs se puede encontrar en el código fuente funciones.c]

## Usar GitHub

Puesto que queríamos trabajar en el proyecto a través de internet, tuvimos que buscar un servicio que nos permitiera realizar esto sin que hubiera problemas al editar el código entre nosotros. Por ejemplo, evitar que alguien trabajara en algo solo para que accidentalmente otro compañero lo sobre-escribiera con su versión. Para esto, nos decidimos a usar un repositorio de GitHub y a aprender algunos comandos Git.

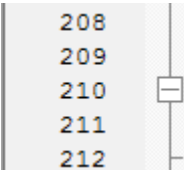


Primeramente, nos pareció un tanto complicado aprender los comandos y hasta el momento seguimos sin tener el tema muy bien afianzado. Sin embargo, utilizando el navegador y arreglándonos, pudimos aprovechar la capacidad de Git para separar el trabajo en ramas (“Branches”) en las que cada uno trabajaba y mediante “Pull Requests” unir todo en la rama maestra.

## Lectura de una lectura tipo char o int (Sub-menú Asignaturas)

Como el menú debe cambiar conforme a la cantidad de asignaturas ingresadas, no se podía preseleccionar un caso para cada asignatura ya que las mismas son reguladas por el usuario, por lo que se decidió que las opciones para poder sobrescribir una asignatura tenían que ser números. A su vez, las otras opciones (agregar asignatura, eliminar asignatura y salir del sub-menú) tenían que poder distinguirse claramente. Por este motivo, usamos variables del tipo char (más específicamente: '+', '-', '!') para identificar estas opciones. Sin embargo, esto trajo un problema ya que ahora el programa tenía que tomar un input e interpretarlo como uno del tipo char o tipo int según lo que quisiera el usuario.

La primera solución que se nos ocurrió en esto fue guardar lo escrito en una variable de tipo char, por lo que la selección de las opciones no-numéricas no era un problema. Pero esto dio lugar a problema ya que, al ingresar un número, el programa no lo reconocería como tal sino que lo interpretaría como un carácter,. Por lo cual, al compararlo no se tiene en cuenta el valor que representa sino su equivalente en el código ASCII, para esto al valor seleccionado se le restaba el valor del '0' en ASCII (que se le asigna el valor 48) y se guardaba en una variable entera.



```

208     NUMERO=ELEC-CERO;
209     if (NUMERO>=CANT_MAT || CANT_MAT==0)
210     {
211         fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_ELEC);
212     }

```

(Imagen del programa con la solución temporal)

Con este método el programa podía funcionar con un ingreso de hasta 10 materias sin ningún problema aparente. Pero si se superaba este límite, podía ocurrir que algún carácter no seleccionado se tome como una variable cualquiera. Para esto encontramos la solución con la función “atoi()”, la cual traduce una cadena de caracteres en una variable entera. Por ejemplo, cuando el usuario ingresa un carácter numérico como “4” la función “atoi()”, convierte ese ‘4’ de un carácter a un número y si se ingresa otro tipo de carácter como, por dar otro ejemplo, ‘a’, la función “atoi()”, devuelve un cero, por lo que solamente hay que verificar que, cuando el usuario ingrese un 0, el programa lo interprete como tal.

```

258     NUMERO = atoi(ELEC); /*Modificacion para que se guarde el numero ingresado en vez de su equivalente en ASCII*/
259     if(NUMERO >= CANT_MAT || CANT_MAT == 0 || NUMERO == 0 && (ELEC[0] != ASCII_CERO))
260     {
261         fprintf(stderr, "%s: %s\n", ERR_PREFIJO, ERR_ELEC);
262     }

```

(Imagen de la función con la solución definitiva)

## Problema con los idiomas

Al realizar los primeros test, el programa parecía funcionar correctamente. Sin embargo, al probar los distintos idiomas incluidos, descubrimos que había un error de compilación al utilizar el idioma inglés. Al revisarlo mejor descubrimos que en el sub-menú “métricas”, en la definición de las variables donde especificamos cual es el caracter que hay que ingresar para elegir una opción, la letra ‘A’ se utilizaba para acceder a dos opciones distintas. Gracias al mensaje de error de compilación del gcc pudimos identificarlo rápidamente y solucionarlo a tiempo.

```

elias@Urquiza-Debian: ~/fiuba/algo1/tp1/algo1-tp1-electronica-master
Archivo Editar Ver Buscar Terminal Ayuda
elias@Urquiza-Debian:~/fiuba/algo1/tp1/algo1-tp1-electronica-master$ gcc -ansi -Wall -pedantic -o academica funciones.c main.c
funciones.c: In function 'metrica':
funciones.c:471:4: error: duplicate case value
     case APLAZOS:
     ^~~~~
funciones.c:424:4: error: previously used here
     case PROMEDIO:
     ^~~~~
elias@Urquiza-Debian:~/fiuba/algo1/tp1/algo1-tp1-electronica-master$

```

[Lectura del error en la consola (Superior) y Motivo del error (Inferior)]

```

english.h (~/fiuba/algo1...lectronica-master) - VIM
Archivo Editar Ver Buscar Terminal Ayuda

/* Metrics */
#define MSJ_METRICA "\nPick which metric to measure:"
#define METRICA_OPCION_PROMEDIO "Average"
#define METRICA_OPCION_PROMEDIO_CHAR 'A'
#define METRICA_OPCION_MAXIMO "Maximum"
#define METRICA_OPCION_MAXIMO_CHAR 'M'
#define METRICA_OPCION_MINIMO "Minimum"
#define METRICA_OPCION_MINIMO_CHAR 'm'
#define METRICA_OPCION_CANTIDAD "Amount of Courses"
#define METRICA_OPCION_CANTIDAD_CHAR '#'
#define METRICA_OPCION_APLAZOS "Failed Courses"
#define METRICA_OPCION_APLAZOS_CHAR 'A'
#define METRICA_OPCION_VOLVER "Go Back"
#define METRICA_OPCION_VOLVER_CHAR '0'

86,1 90%

```

## **Bibliografía**

- <https://stackoverflow.com/questions/2001626/fgets-function-in-c>
- <https://stackoverflow.com/questions/22608160/atom-change-indentation-mode>
- H. M. Deitel y P. J. Deitel. Como programar en C/C++ y Java. 4ª Edición. Pearson Educación, 2004, ISBN 970-26-0531-8, pág 295 y 296.
- [Repositorio del Grupo](#)

## Indicaciones de Compilación

Para compilar el programa se puede escribir lo siguiente en la consola:

```
$ gcc -ansi -Wall -pedantic -o nombrePrograma main.c funciones.c
```

O bien:

```
$ gcc -ansi -Wall -pedantic -c -o funciones.o funciones.c
```

```
$ gcc -ansi -Wall -pedantic -c -o main.o main.c
```

```
$ gcc -ansi -Wall -pedantic -o nombrePrograma main.o funciones.o
```

Posteriormente, se puede correr el programa utilizando `./nombrePrograma`, siendo “nombrePrograma” el nombre que usted desee asignarle al mismo.