



C Piscine

C 11

*Resumen: Este documento corresponde al enunciado del módulo C 11 de la C Piscine de 42.*

*Versión: 8*

# Índice general

I.	Instrucciones	2
II.	Instrucciones sobre la IA	4
III.	Introducción	7
IV.	Ejercicio 00 : ft_foreach	8
V.	Ejercicio 01 : ft_map	9
VI.	Ejercicio 02 : ft_any	10
VII.	Ejercicio 03 : ft_count_if	11
VIII.	Ejercicio 04 : ft_is_sort	12
IX.	Ejercicio 05 : do-op	13
X.	Ejercicio 06 : ft_sort_string_tab	15
XI.	Ejercicio 07 : ft_advanced_sort_string_tab	16
XII.	Entrega y evaluación	17

# Capítulo I

## Instrucciones

- Esta página será la única referencia: no te fíes de los rumores.
- Antes de empezar a hacer los ejercicios deberás registrarte en el proyecto en tu intranet. Una vez lo hayas hecho, obtendrás tu repositorio Git en el cual debes trabajar.
- Los ejercicios han sido ordenados del más sencillo al más complejo. Por lo tanto, debes hacerlos en orden y asegurarte de que el ejercicio es correcto antes de continuar con el siguiente.
- Los ejercicios de Shell se deben ejecutar con `/bin/bash`.
- Los ejercicios de C se deben compilar con `cc` y utilizando las flags `-Wall -Wextra -Werror`.
- Solamente hay que entregar una función `main()` si lo que se pide es un programa. Si se pide una función se puede entregar el `main()` comentado con la batería de tests que hayas hecho.
- Los ejercicios de C se deben escribir de acuerdo a **la Norma**. Puedes encontrarla en la intranet, en el icono de la brújula, FAQ, sección 42, apartado General Pedagogy como **La Norma**.
- Para ayudarte a comprobar si tus ejercicios cumplen **la Norma** puedes ayudarte de **norminette** - un programa que aplica ciertos requisitos de **la Norma** a tu código. Si tienes archivos o funciones adicionales, también deben cumplir con **la Norma**.
- Lee detenidamente los ejemplos. Podrían dar información que no se especifica en el enunciado...
- Asegúrate de que tus directorios y archivos tienen los permisos adecuados.
- Debes respetar **el procedimiento de entrega** para todos tus ejercicios. Solo el trabajo de tu repositorio **Git** será evaluado.
- **No puedes** dejar en tu directorio **ningún** archivo que no se haya indicado de forma explícita en los enunciados de los ejercicios.

- La evaluación de este proyecto consta de dos fases. Primero, las personas con las que compartes piscina se encargarán de evaluar tus ejercicios. A continuación, serán evaluados por un programa que se llama **la Moulinette**.
- Tus funciones no deben terminar de forma inesperada (segfault, bus error, double free) excepto en el caso de comportamientos indefinidos. Si esto sucede, tu proyecto será considerado no funcional y recibirás un 0 durante la evaluación.
- **La Moulinette** es muy estricta a la hora de evaluar y está completamente automatizada. Es imposible discutir con ella sobre tu nota. Por lo tanto, debes mantener el rigor en tu código para evitar cualquier sorpresa.
- **La Moulinette** utiliza **norminette** para comprobar tus ejercicios, aunque este programa no es exhaustivo en la verificación de **la Norma**. Durante las evaluaciones por parte de las personas de tu Piscina se comprobará que **la Norma** es correcta. Si se incumple el proyecto tendrá una puntuación final de **0**, aunque **norminette** no indique ningún error.
- ¿Tienes alguna pregunta? Pregunta a la persona de tu derecha. Si no puede ayudarte, prueba con la persona de tu izquierda.
- ¡Por **Loki**, por **Freyja**! ¡¡Piensa fuerte!!

# Capítulo II

## Instrucciones sobre la IA

### Contexto

La Piscina de C es intensa. Es tu primer gran desafío en 42: una inmersión profunda en la resolución de problemas, la autonomía y la comunidad.

Durante esta etapa, tu objetivo principal es construir unas bases sólidas, a través del esfuerzo, la repetición y, sobre todo, mediante el **aprendizaje entre pares**.

En la era de la IA, los atajos son fáciles de encontrar. Sin embargo, es importante considerar si el uso que haces de la IA te está ayudando realmente a crecer, o si simplemente te está impidiendo desarrollar habilidades reales.

La Piscina también es una experiencia humana y, por ahora, nada puede reemplazar eso. Ni siquiera la IA.

Para obtener una visión más completa de nuestra postura sobre la IA como herramienta de aprendizaje, como parte del plan de estudios de TIC (Tecnologías de la Información y la Comunicación) y como una expectativa creciente en el mercado laboral, consulta las preguntas frecuentes *FAQ* disponibles en la intranet.

### ● Mensaje principal

- 👉 Construir fundamentos sólidos sin atajos.
- 👉 Desarrollar de forma real habilidades técnicas y transversales.
- 👉 Experimentar el aprendizaje entre pares de forma real, empezar a aprender a aprender y a resolver nuevos problemas.
- 👉 El proceso de aprendizaje es más importante que el resultado.
- 👉 Aprender sobre los riesgos asociados a la IA y desarrollar prácticas de control efectivas y medidas de mitigación para evitar errores comunes.

## ● Reglas para la piscina:

- Aplica la lógica y el razonamiento a las tareas asignadas, especialmente antes de recurrir a la IA.
- No deberías pedir respuestas directas a la IA.
- Aprende sobre el enfoque global de 42 sobre la IA.

## ● Resultados de esta etapa:

Durante esta etapa fundamental, obtendrás los siguientes resultados:

- Obtener Una base sólida técnica y en programación.
- Comprender por qué y cómo la IA puede ser peligrosa durante esta fase.

## ● Comentarios y ejemplos:

- Si, sabemos que la IA existe. Y sí, puede resolver tus proyectos. Pero estás aquí para aprender, no para demostrar que la IA ha aprendido. No pierdas tu tiempo (o el nuestro) solo para demostrar que la IA puede resolver el problema dado. y que puede resolver tus proyectos, pero estás aquí para aprender. Te recomendamos que inviertas 26 días de aprendizaje de calidad para resolver los retos que hemos pensado para ti.
- Aprender en 42 no va sólo de saber la respuesta. Se trata de desarrollar la capacidad de encontrar una. La IA te da la respuesta directamente, pero eso te impide construir tu propio razonamiento. Y el razonamiento requiere tiempo, esfuerzo y conlleva fracasos. El camino hacia el éxito no debe ser fácil. no tiene nada que ver con saber la respuesta. En 42 aprendemos desarrollando la capacidad para encontrarla. La IA te dará la respuesta directa, lo que impide que desarrolles tu propio razonamiento. Razonar requiere tiempo, esfuerzo y equivocarse muchas veces.
- Piensa que durante los exámenes, la IA no está disponible sin internet, sin smartphones, etc. Te darás cuenta rápidamente si has confiado demasiado en la IA durante tu proceso de aprendizaje. Recuerda que durante los exámenes no tendrás acceso a la IA ni a internet. Te vas a enfrentar a una hoja en blanco donde vas a tener que escribir tu propio código.
- El aprendizaje entre pares te expone a diferentes ideas y enfoques, mejorando tus habilidades interpersonales y tu capacidad de pensar de forma diferente. Eso es mucho más valioso que simplemente chatear con un bot. Así que, ¡que no te supere la timidez!: ¡habla, haz preguntas y aprende con el resto de la piscina!

- Sí, la IA formará parte del plan de estudios, tanto como herramienta de aprendizaje como tema en sí mismo. Incluso tendrás la oportunidad de crear tu propio software de IA. Para aprender más sobre nuestro enfoque progresivo, puedes consultar la documentación disponible en la intranet.

#### ✓ Buenas prácticas:

Me atasco en un nuevo concepto. Le pregunto a alguien cercano cómo lo ha abordado. Hablamos durante 10 minutos y, de repente, todo encaja. Lo entiendo. No entiendo algo concreto del proyecto y no sé cómo continuar. Le pregunto a otra persona de la piscina cómo lo ha abordado, hablamos sobre el tema y, si es necesario, incluso utilizamos otros métodos (papel y boli, dibujos, metáforas, etc.) hasta conseguir entenderlo.

#### ✗ Mala práctica:

Utilizo la IA en secreto, copio un código que parece correcto. Durante la evaluación entre pares, no puedo explicar nada. Suspenso. Durante el examen, sin IA, me vuelvo a atascar. Suspenso.

# Capítulo III

## Introducción

Cita sacada de la película V de Vendetta:

¡Vedà! A primera Vista, un humilde Veterano de VaudeVille,  
en el papel de Víctima y Villano por Vicisitudes de la Vida.  
Este que Ves, ya no mas Velo de Vanidad, es un Vestigio de  
la Vox populi, ahora Vacua, desVanecida. Sin embargo, esta Valerosa  
Visión de una extinta Vejación se siente reViVida y ha hecho Voto de  
Vencer el Vil Veneno de estas Víboras en aVanzada que Velan por los  
Violentos Viciosos y por la Violación de la Voluntad. El único Veredicto  
es Venganza; Vendetta, como Voto, y no en Vano, pues la Valía y Veracidad  
de esta un día Vindicará al Vigilante y al Virtuoso. La verdad, esta  
Vichyssoise de Verborrea se está VolViendo muy Verbosa así que solo añadiré  
que es un Verdadero placer conocerte y que puedes llamarme V.




Evite las aliteraciones. Siempre.



# Capítulo IV

## Ejercicio 00 : ft\_foreach

	Ejercicio: 00
	ft_foreach
	Directorio de entrega: <i>ex00/</i>
	Archivos a entregar: <b>ft_foreach.c</b>
	Funciones autorizadas: Ninguna

- Escribe una función **ft\_foreach** que, para una tabla de enteros determinada, aplique una función sobre cada uno de los elementos de la tabla. Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:


```
void ft_foreach(int *tab, int length, void(*f)(int));
```

- Por ejemplo, para mostrar todos los enteros de la tabla, se podrá llamar a la función **ft\_foreach** de la siguiente forma:

```
ft_foreach(tab, 1337, &ft_putnbr);
```

# Capítulo V

## Ejercicio 01 : ft\_map


	Ejercicio: 01
ft_map	
Directorio de entrega: <i>ex01/</i>	
Archivos a entregar: <b>ft_map.c</b>	
Funciones autorizadas: <b>malloc</b>	

- Escribe una función **ft\_map** que, para una tabla de enteros determinada, aplique una función sobre cada uno de los elementos de la tabla (por orden) y retorne una tabla con todos los valores de retorno.
- Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
int      *ft_map(int *tab, int length, int(*f)(int));
```

# Capítulo VI

## Ejercicio 02 : ft\_any

	Ejercicio: 02
ft_any	
Directorio de entrega: <i>ex02/</i>	
Archivos a entregar: <b>ft_any.c</b>	
Funciones autorizadas: <b>Ninguna</b>	


- Escribe una función **ft\_any** que devuelva 1 si, al menos uno de los elementos de la tabla pasados a la función **f** devuelve algo diferente de 0, si no devolverá 0.
- Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
int ft_any(char **tab, int(*f)(char*));
```

- La tabla terminará con un puntero NULL.

# Capítulo VII

## Ejercicio 03 : ft\_count\_if


	Ejercicio: 03
	ft_count_if
	Directorio de entrega: <i>ex03/</i>
	Archivos a entregar: <b>ft_count_if.c</b>
	Funciones autorizadas: Ninguna

- Escribe una función `ft_count_if` que devuelva el número de elementos de la tabla que, cuando sean pasados a la función `f`, no devuelvan 0.
- Se aplicará esta función siguiendo el orden de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
int ft_count_if(char **tab, int length, int(*f)(char*));
```

# Capítulo VIII

## Ejercicio 04 : ft\_is\_sort


	Ejercicio: 04
	ft_is_sort
Directorio de entrega: <i>ex04/</i>	
Archivos a entregar: <b>ft_is_sort.c</b>	
Funciones autorizadas: Ninguna	

- Escribe una función `ft_is_sort` que devuelva 1 si la tabla está ordenada y 0 en el caso contrario.
- La función pasada como parámetro devolverá un entero negativo si el primer argumento es inferior al segundo, 0 si son iguales y un entero positivo en cualquier otro caso.
- El prototipo de la función deberá ser el siguiente:

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

# Capítulo IX

## Ejercicio 05 : do-op

	Ejercicio: 05
do-op	
Directorio de entrega: <i>ex05/</i>	
Archivos a entregar: Todos los archivos necesarios para su programa	
Funciones autorizadas: write	

- Crea un programa que se llame do-op.
- El programa deberá ser ejecutado con tres argumentos: do-op valor1 operador valor2
- Ejemplo:

```
$> ./do-op 42 "+" 21
63
$>
```

- Deberías usar una tabla de punteros a funciones para llamar a la función que corresponda a un operador.
- En el caso de un operador desconocido, tu programa debe mostrar 0.
- Si el número de argumentos no es correcto, do-op no muestra nada.
- Tu programa debe aceptar y mostrar el resultado con los siguientes operadores: '+', '-', '/', '\*', y '%'
- Tu programa debe tratar el valor el valor como int.
- En el caso de una división por 0 tu programa debe mostrar:

```
Stop : division by zero
```

- En el caso de un módulo por 0 tu programa debe mostrar:


```
Stop : modulo by zero
```

- He aquí un ejemplo de pruebas de la Moulinette:

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

# Capítulo X

## Ejercicio 06 : ft\_sort\_string\_tab

	Ejercicio: 06
	ft_sort_string_tab
	Directorio de entrega: <i>ex06/</i>
	Archivos a entregar: <b>ft_sort_string_tab.c</b>
	Funciones autorizadas: Ninguna


- Escribe la función `ft_sort_string_tab` que ordene por código ASCII las cadenas de caracteres.
- `tab` se terminará con un `NULL`.
- La clasificación se realizará intercambiando los punteros de la tabla.
- El prototipo de la función deberá ser el siguiente:

```
void ft_sort_string_tab(char **tab);
```



# Capítulo XI

## Ejercicio 07 : ft\_advanced\_sort\_string\_tab

	Ejercicio: 07
	ft_advanced_sort_string_tab
	Directorio de entrega: <i>ex07/</i>
	Archivos a entregar: <b>ft_advanced_sort_string_tab.c</b>
	Funciones autorizadas: Ninguna

- Escribe la función `ft_advanced_sort_string_tab` que ordene, en función del retorno de la función pasada como parámetro.
- La clasificación se realizará intercambiando los punteros de la tabla.
- `tab` se terminará con un `NULL`.
- El prototipo de la función deberá ser el siguiente:

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



La llamada a `ft_advanced_sort_string_tab()` con un segundo parámetro `ft_strcmp` dará el mismo resultado que `ft_sort_string_tab()`.

# Capítulo XII

## Entrega y evaluación

Entrega tu proyecto en tu repositorio `Git` como de costumbre. Solo el trabajo entregado en el repositorio será evaluado durante la defensa. No dudes en comprobar varias veces los nombres de los archivos para verificar que sean correctos.



Sólo necesitas entregar los archivos requeridos por el enunciado de este proyecto.