



C Piscine

C 09

*Resumen: Este documento corresponde al enunciado del módulo C 09 de la C Piscine de 42.*

*Versión: 5*

# Índice general

<b>I.</b>	<b>Instrucciones</b>	<b>2</b>
<b>II.</b>	<b>Instrucciones sobre la IA</b>	<b>4</b>
<b>III.</b>	<b>Introducción</b>	<b>7</b>
<b>IV.</b>	<b>Ejercicio 00 : libft</b>	<b>9</b>
<b>V.</b>	<b>Ejercicio 01 : Makefile</b>	<b>10</b>
<b>VI.</b>	<b>Ejercicio 02 : ft_split</b>	<b>12</b>
<b>VII.</b>	<b>Entrega y evaluación</b>	<b>13</b>

# 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!!



Para este módulo, la Norminette debe ser ejecutada con el flag `-R CheckForbiddenSourceHeader`. La Moulinette también lo utilizará.

# 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

Fragmento de Proyecto Hail Mary, de Andy Weir

Toc, toc, toc, toc

-¿Qué?- Me vuelvo a mirarlo.

Señala el símbolo Å que todavía tengo en la mano y luego a mí.

Luego otra vez el Å y otra vez a mí. Lo hace de una forma casi frenética.

-Oh, lo siento- digo. Sostengo el dígito en alto y digo-: Tres.

Hace las manos de jazz. Le devuelvo el gesto.

Eh. Ya que estamos en el tema...

Me quedo quieto un momento, así sabrá que hay un corte en la conversación.

Entonces hago las manos de jazz y digo.

-Sí.

Repito el gesto.

-Sí.

-El vuelve a hacerlo y dice:

-Pp.

Anoto y grabo las frecuencias en mi portátil.

-Está bien. Tenemos ‘‘sí’’ en nuestro vocabulario ahora- digo.

Toc, toc, toc.

Miro. Una vez que sabe que tiene mi atención hace las manos de jazz y dice:

-Pp.

El mismo acorde que antes. -Sí -digo-. Lo tenemos cubierto.

Levanta un dedo un momento. Entonces cierra dos de sus puños y los golpea juntos.

-pp.

-¿Qué?

Ohhh -digo. Soy profesor.

¿Qué le enseñaría a alguien que acaba de aprender la palabra ‘‘sí’’?

‘‘Eso es ‘‘no’’.

Al menos eso espero.

Cierro los puños y los golpeo juntos.

-No.

-Pp dice. Verifico el portátil. Acaba de decir sí.

Espera. ¿Significa eso que no es no? ¿Es eso otro sí? Ahora estoy confundido.

¿No? -pregunto.

-No- dice en eridiano.

Entonces «sí»? -No, sí.



¿Sí? -No. No. -¿Sí, sí?


¡No! -Cierra un puño hacia mí, claramente frustrado.

Basta de esta rutina de Abbott y Costello interespecies.

Por desgracia, este enunciado no tiene nada que ver con el aprendizaje del eridiano.

# Capítulo IV

## Ejercicio 00 : libft

	Ejercicio: 00
libft	
Directorio de entrega: <i>ex00/</i>	
Archivos a entregar: <code>libft_creator.sh</code> , <code>ft_putchar.c</code> , <code>ft_swap.c</code> , <code>ft_putstr.c</code> , <code>ft_strlen.c</code> , <code>ft_strcmp.c</code>	
Funciones autorizadas: <code>write</code>	

- Crea tu biblioteca `ft`. Se llamará `libft.a`.
- Un script shell denominado `libft_creator.sh` compilará correctamente los archivos fuente y creará tu biblioteca.
- Esta biblioteca debe incluir todas las funciones que se indican a continuación:


```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

- Ejecutaremos el comando siguiente:

```
sh libft_creator.sh
```

# Capítulo V

## Ejercicio 01 : Makefile

	Ejercicio: 01
Makefile	
Directorio de entrega: <i>ex01/</i>	
Archivos a entregar: <b>Makefile</b>	
Funciones autorizadas: <b>Ninguna</b>	

- Crea el **Makefile** que compile una librería **libft.a**.
- Tu **makefile** debe mostrar cada comando que ejecute.
- Tu **makefile** no debe ejecutar ningún comando inútil.
- El **Makefile** irá a buscar los archivos fuente al directorio **srcs**.
- Estos archivos fuente serán: **ft\_putchar.c**, **ft\_swap.c**, **ft\_putstr.c**, **ft\_strlen.c**, **ft\_strcmp.c**
- El **Makefile** irá a buscar los archivos headers al directorio **includes**.
- Estos archivos headers serán: **ft.h**
- Tendrás que compilar los archivos C utilizando **cc** y las opciones flag **-Wall -Wextra -Werror**, en ese orden.
- La librería estará en la raíz del ejercicio.
- Los archivos **.o** tendrán que estar al lado de tus archivos **.c** correspondientes.
- El **Makefile** también tendrá que implementar reglas **clean**, **fclean**, **re**, la regla **all** y, por supuesto, **libft.a**.
- Ejecutar solo **make** debe ser equivalente a **make all**.
- La regla **all** tendrá que hacer lo mismo que **make libft.a**.
- La regla **clean** tendrá que retirar todos los archivos temporales que se hayan generado.


- La regla `fclean` hará lo mismo que un `make clean` y borrará también el binario creado al ejecutar `make all`.
- La regla `re` hará lo mismo que un `make fclean` seguido de un `make all`.
- Tu makefile no debe compilar los archivos inútilmente.
- Solo recogeremos tu Makefile y lo probaremos con nuestros archivos.



¡Ten cuidado con los wildcards!

# Capítulo VI

## Ejercicio 02 : ft\_split

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

- Escribe una función que divida una cadena de caracteres en función de otra cadena de caracteres.
- Habrá que utilizar cada carácter de la cadena **charset** como separador.
- La función devuelve una tabla donde cada uno de tus elementos contiene la dirección de una cadena de caracteres comprendida entre dos separadores. El último elemento de la tabla tendrá que ser igual a 0 para indicar el final de la tabla.
- Tu tabla no puede tener cadenas vacías. Saca las conclusiones pertinentes.
- La cadena que se transmitirá no será modificable.
- El prototipo de la función deberá ser el siguiente:

```
char **ft_split(char *str, char *charset);
```

# Capítulo VII

## 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.