



C Piscine

C 10

*Resumen:* Este documento corresponde al enunciado del módulo C 10 de la C Piscine de 42.

*Versión:* 7

# Índice general

I.	Instrucciones	2
II.	Instrucciones sobre la IA	4
III.	Introducción	7
IV.	Ejercicio 00 : display_file	8
V.	Ejercicio 01 : cat	9
VI.	Ejercicio 02 : tail	10
VII.	Ejercicio 03 : hexdump	11
VIII.	Entrega y evaluación	12

# 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 ícono 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. Suspendo. Durante el examen, sin IA, me vuelvo a atascar. Suspendo.

# Capítulo III

## Introducción

Para bien empezar el día, aquí algunas preguntas muy sencillas:

¿Qué pasaría si metiésemos un secador alimentado por corriente continua en una caja hermética de un metro de lado?

¿Se habría detenido el accidente de Chernóbil arrojando antimateria en el reactor cuando se estaba fundiendo?

¿Es posible llorar hasta deshidratarse?

Si todos los humanos desaparecieran de la faz de la tierra, ¿cuánto tiempo pasaría antes de que la última fuente de luz artificial se apagara?

¿Hasta qué punto es peligroso meterse en una piscina durante una tormenta?

¿Desde qué altura habría que tirar un filete para que estuviera cocinado cuando llegara al suelo?

¿Cuándo, en caso de que ocurra alguna vez, el ancho de banda de Internet superará al servicio de mensajería de FedEx?

¿Cuántos tweets diferentes son posibles en nuestro idioma? ¿Cuánto tiempo tardaría la población mundial en leerlos todos en voz alta?

¿Cuál sería el resultado si todos los candidatos al carnet de conducir respondiesen al azar en el test del examen teórico?

¿Cuántos contestarían correctamente a todas las preguntas?

Preguntas sacadas les libro “¿Qué pasaría si... ?” de Randall Munroe.

# Capítulo IV

## Ejercicio 00 : display\_file

	Ejercicio: 00
	display_file
	Directorio de entrega: <i>ex00/</i>
	Archivos a entregar: <b>Makefile y los archivos de tu programa</b>
	Funciones autorizadas: <b>close, open, read, write</b>

- Crea un programa que se llame `ft_display_file` y que muestre en la salida estándar únicamente el contenido del archivo pasado como argumento.
- El directorio de entrega deberá tener un **Makefile** que cumpla las siguientes reglas: **all**, **clean** y **fclean**. El binario se llamará `ft_display_file`.
- La función `malloc` está prohibida. Sólo puedes hacer el ejercicio declarando una tabla de tamaño fijo.
- Todos los archivos pasados como parámetros serán válidos.
- Los mensajes de error tendrán que mostrarse en la salida que les haya sido reservada y seguidos de un salto de línea.
- Si no hay ningún argumento, tu programa deberá mostrar

File name missing.
- Si hay demasiados argumentos, tu programa deberá mostrar

Too many arguments.
- Si el archivo no se puede leer, tu programa deberá mostrar

Cannot read file.

# Capítulo V

## Ejercicio 01 : cat

	Ejercicio: 01
	cat
	Directorio de entrega: <i>ex01/</i>
	Archivos a entregar: <b>Makefile</b> y los archivos de tu programa
	Funciones autorizadas: <b>close</b> , <b>open</b> , <b>read</b> , <b>write</b> , <b>strerror</b> , <b>basename</b>

- Escribe un **programa** que se llame **ft\_cat** y que realice lo mismo que el comando **cat** del sistema.
- No necesitas gestionar las opciones.
- El directorio de entrega tendrá un **Makefile** con las siguientes reglas **all**, **clean** y **fclean**.
- Puedes utilizar la variable **errno** (ver el **man** de **errno**).
- Deberías leer los **man** de todas las funciones autorizadas.
- Solo puedes hacer el ejercicio declarando una tabla de tamaño fijo. Esta tabla tendrá un tamaño limitado a algo menos de unos 30 **ko**. Para probar esta limitación, utiliza el comando **ulimit** de tu shell.

# Capítulo VI

## Ejercicio 02 : tail

	Ejercicio: 02
	tail
	Directorio de entrega: <i>ex02/</i>
	Archivos a entregar: <b>Makefile y los archivos de tu programa</b>
	Funciones autorizadas: <b>close, open, read, write, malloc, free, strerror, basename</b>

- Escriba un programa que se llame **ft\_tail** y que realice lo mismo que el comando **tail**.
- Solo tiene que gestionar la opción **-c**, pero no tiene que gestionar el '+' y el '-'.
- Se realizarán todas las pruebas con la opción **-c**.
- El directorio de entrega tendrá un **Makefile** las siguientes reglas: **all, clean y fclean**.
- Puede utilizar la variable **errno**.

# Capítulo VII

## Ejercicio 03 : hexdump

	Ejercicio: 03
	hexdump
	Directorio de entrega: <i>ex03/</i>
	Archivos a entregar: <b>Makefile</b> y los archivos de tu programa
	Funciones autorizadas: <b>close</b> , <b>open</b> , <b>read</b> , <b>write</b> , <b>malloc</b> , <b>free</b> , <b>strerror</b> , <b>basename</b>

- Escriba un programa que se llame **ft\_hexdump** y que realice lo mismo que el comando **hexdump** del sistema, sin redirección.
- No necesita gestionar la opción **-c**.
- El directorio de entrega tendrá un **Makefile** con una regla **all**, una regla **clean** y una regla **fclean**.
- Puede utilizar la variable **errno** (ver el **man** de **errno**).

# Capítulo VIII

## 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.