# ANEXO · TAREAS DEL ALUMNADO

## INFORMES PDF CON TYPST: UNA PLANTILLA + LOS DATOS

Nombre: _____ Fecha: _____

Piensa en un formulario preimpreso y la tinta. El formulario es la **plantilla**: tamaño de página, cabecera, pie, fuentes; se escribe una vez. La tinta son los **datos**: filas y totales. Hoy el informe de NoteBox se convierte en un PDF A4 que nadie dibuja a mano.

## PASO 1 · HOLA, INFORME (8 MIN)

Crea la carpeta `reports/` en tu proyecto y el fichero `reports/notes-report.typ`:

```
#let notes = csv("../static/data/notes.csv", row-type: dictionary)

= Notes report

There are #notes.len() notes. The first one is *#notes.first().title*,
with #notes.first().words words.
```

Compila en el terminal, **desde la raíz del proyecto**:

```
typst compile --root . reports/notes-report.typ notes-report.pdf
```

| OK | HAZ | ESTÁ HECHO CUANDO… |
|---|---|---|
| ☐ | Crea el fichero y compílalo. | El PDF dice There are 36 notes. The first one is **Complaint from table 7**, with 419 words. |

Si algo falla:

- **`would escape the project root.`** Te falta `--root .` o estás dentro de `reports/`. Typst solo lee ficheros dentro de su raíz.

## PASO 2 · LA PLANTILLA (7 MIN)

Guarda `template.typ` de Classroom dentro de `reports/`. Mientras lo leemos juntos, apunta qué banda es cada parte (page header, page footer, report header). Después sustituye el título y el párrafo de `notes-report.typ` por:

```
#import "template.typ": report

#let notes = csv("../static/data/notes.csv", row-type: dictionary)

#show: report.with(
  title: "Notes report",
  subtitle: [Folder: All · #notes.len() notes],
)
```

| OK | HAZ | ESTÁ HECHO CUANDO… |
|---|---|---|
| ☐ | Compila otra vez. | Una página A4 con page header, el título, «Folder: All · 36 notes» y el pie con la fecha y **1 / 1**. |

## PASO 3 · LA TABLA, BANDA A BANDA (17 MIN)

Mira la página de Figma **Design spec · the PDF**. Compila después de cada paso (`typst watch` en un segundo terminal te ahorra escribir).

| OK | HAZ | ESTÁ HECHO CUANDO… |
|---|---|---|
| ☐ | **1 · Column header y detalle** (con el profesor): `table.header` y una fila por nota, con el valor calculado de minutos. | La cabecera de la tabla se repite en la página 2. |
| ☐ | **2 · Grupos** (inténtalo tú primero): por cada carpeta, un group header, sus filas de detalle y un group footer con el recuento y los subtotales. | Cada carpeta empieza con su nombre y acaba con «N notes in …». |
| ☐ | **3 · Report footer** (tú solo): un `table.footer` con los totales generales, como último argumento de la tabla. | Los totales salen **una sola vez**, al final. |
| ☐ | **4 · Comprueba**. | Dos páginas. Última línea: **Total · 36 notes · 9213 · 58**. Group footers: Personal 1640 · 12, School 5217 · 28, Shopping 286 · 7, Work 2070 · 11. |

Si algo falla:

- **`cannot add integer and string`.** Te falta un `int()` alrededor de `n.words`: los valores del CSV son texto.
- **`expected content, found integer`.** Una celda tiene que ser contenido o texto: envuelve los números en `str(...)`.
- **Las columnas se desplazan.** Te falta `.flatten()`, o un group footer tiene tres celdas en lugar de cuatro (cuenta el `colspan`).
- **`unknown variable: report`.** Falta la línea `#import` o la plantilla está en otra carpeta.

## PASO 4 · UN PARÁMETRO (5 MIN)

Un informe de una sola carpeta, sin tocar el fichero. Encima de la tabla:

```
#let folder = sys.inputs.at("folder", default: "All")
#let rows = notes.filter(n => folder == "All" or n.folder == folder)
```

| OK | HAZ | ESTÁ HECHO CUANDO… |
|---|---|---|
| ☐ | Cambia `notes` por `rows` en el subtítulo, la lista de carpetas, los grupos y el footer, y escribe `Folder: #folder` en el subtítulo. | Sin `--input` sigue saliendo el informe completo. |
| ☐ | Compila con `typst compile --root . --input folder=School reports/notes-report.typ school.pdf`. | `school.pdf`: una página, **12 notes · 5217 words · 28 min**. |

## PREGUNTAS DE LA SESIÓN

**1** Con `row-type: dictionary`, ¿cómo es cada fila? ¿De qué tipo es el valor de `words`?

**2** ¿Por qué el `table.footer` lleva `repeat: false`? ¿Qué pasaría sin él?

**3** ¿Quién podría ejecutar el segundo comando del paso 4 en lugar de ti?

## ENTREGA

En la tarea de Classroom **UT6 · PDF report**: `notes-report.pdf` (todas las carpetas, dos páginas) y `notes-report.typ`. Una línea: ¿qué tres cosas están en la plantilla, y por qué no en el fichero del informe?

## PARA CASA

- Escribe la versión PDF de **tu propio** informe con el mismo `template.typ`: tu fichero de datos, un column header, una agrupación con group footers, un valor calculado, un report footer y un parámetro `--input` que se vea en el subtítulo. La lista está en la página de Figma **Design spec · the PDF**.
- Cambia una cosa de la plantilla (el nombre de la app en el page header) y comprueba que tus dos informes cambian.
- Ejercicio de transferencia (un párrafo): un gimnasio imprime cada mes un extracto para cada uno de sus 300 socios: las visitas del mes, los minutos de cada visita y el total de minutos. Di qué va en la plantilla, qué son los datos, cuál es el parámetro y cuántas veces se ejecuta el comando.