

Entregable3

October 19, 2025

[2]: # Resumen de negocio - Entregable 3 (Netflix)

Mario Ibañez

****Propósito****

Ofrecer una lectura ejecutiva del catálogo que ayude a decidir ****qué**
→ **promocionar, qué producir o adquirir****, y ****dónde**** enfocarlo. Los insights
→ se eligieron por su impacto directo en ****programación, marketing y**
→ **portfolio****.

¿Por qué estos insights?

1. ****Top 5 géneros**** (con filtro MOVIE/SHOW)

- * ****Para qué sirve:**** priorizar escaparates, campañas y negociaciones por
→ los géneros que más tracción tienen.
- * ****Cómo se lee:**** son los cinco géneros con ****más títulos únicos**** bajo el
→ filtro activo; si cambia de año, tipo o país, el Top se actualiza.
- * ****Uso práctico:**** concentrar portada y promos en los géneros líderes;
→ detectar ****nichos**** donde un empujón curado puede tener alto ROI.

2. ****Evolución género x año (últimos 20)****

- * ****Para qué sirve:**** entender ****tendencia y estacionalidad****; anticipar
→ ventanas de contenido.
- * ****Cómo se lee:**** los círculos más oscuros indican años fuertes para un
→ género; caídas o picos guían la ****planificación de lanzamientos****.
- * ****Uso práctico:**** calendarizar campañas, rescatar catálogos con pico de
→ interés histórico, planear colecciones temáticas.

3. ****Top 5 y "Emergentes 6-10" (Directores / Actores)****

- * ****Para qué sirve:**** gestionar ****talento**** y construir franquicias. Los
→ **"top"** señalan ****caballos ganadores****; los **"emergentes"** revelan ****apuestas**
→ **costo-efectivas****.

* **Cómo se lee:** ranking por **participación en obras únicas** (sin duplicados); **emergente** = posiciones 6-10, con potencial de saltar al Top con visibilidad.

* **Uso práctico:** campañas de **de los creadores de...**, acuerdos de **first-look**, curadurías por talento y **casting** estratégico.

4. **Distribución por clasificación de edad (IMDb/TV)**

* **Para qué sirve:** alinear **segmentación, placement y compliance** (Kids, Teens, Adults).

* **Cómo se lee:** el gráfico de pastel muestra el **mix de certificaciones** del catálogo filtrado; desequilibrios indican **brechas de oferta**.

* **Uso práctico:** reforzar surtido en segmentos débiles, ajustar **controles parentales** y diseñar **rutinas de descubrimiento** por edad.

5. **Mapa por país de producción**

* **Para qué sirve:** orientar **adquisiciones y local originals**; medir diversidad y oportunidad geográfica.

* **Cómo se lee:** mayor intensidad de color = **más títulos únicos** por país; cambios al filtrar revelan **territorios ancla** y **coproducciones** clave.

* **Uso práctico:** priorizar deals en países subrepresentados; campañas de **orgullo local**; ventanas por festividades nacionales.

6. **Tarjetas clave (Títulos, Promedio IMDb, Votos)**

* **Para qué sirve:** tener a la vista un **termómetro de volumen y calidad**.

* **Cómo se lee:** evolución de estas tarjetas bajo filtros indica si el enfoque (género/país/talento) **mejora o no** la propuesta de valor.

* **Uso práctico:** metas de catálogo (volumen) y metas de percepción (calidad/votos) en OKRs trimestrales.

Cómo interpretar y actuar

* **No todo volumen implica valor:** cruzar Top géneros con **Promedio IMDb** y **Votos** permite diferenciar géneros **populares** de géneros **queridos** (alta nota y votos).

* **Emergentes secundarios:** el bloque 6-10 indica **resiliencia y crecimiento**; darles portada puede moverlos al Top con coste menor que fichajes premium.

```

* Ajuste por país: si un género es fuerte global pero débil en ciertos
↳países, localizar campañas y explorar adquisiciones selectivas.
* Ventanas por año: los picos en la matriz ayudan a reestrenar
↳colecciones (aniversarios, sagas) y planear temporalidad de marketing.
* Clasificación de edad: un pastel muy cargado a adultos sugiere reforzar
↳Kids/Teens para ampliar base y mejorar retención familiar.

---

## Recomendaciones de acción

1. Portada basada en datos: ocupar el 60-70% del hero con Top géneros/
↳talento bajo filtro (MOVIE/SHOW) y reservar slots tácticos para
↳emergentes.
2. Apuestas de contenido: pipeline trimestral de originals/licencias en
↳géneros con hueco (por país), midiendo impacto en Promedio IMDb y
↳Votos.
3. Talento como ancla de curación: carruseles "De [Director/Actor]" y
↳campañas cruzadas; negociar continuidad donde haya señales de franquicia.
4. Segmentación por edad: diversificar catálogo en segmentos deficitarios y
↳ajustar UX de descubrimiento (rutas Kids/Teens/Adults).
5. Operación continua: revisar mensualmente Top/Emergentes y el mix
↳geográfico, ajustar promociones y priorizar compras según gaps.

---

Cierre
El tablero resume las preguntas de negocio críticas: ¿qué géneros empujan
↳el consumo?, ¿qué talento conviene destacar o asegurar?, ¿dónde hay
↳oportunidades por país y edad? Con estos bloques, las decisiones de
↳programación, marketing y adquisiciones pueden ser más rápidas,
↳comparables y accionables, alineando catálogo y demanda en cada mercado.

```

```
Cell In[2], line 18
```

```
2. Evolución género × año (últimos 20)
```

```
SyntaxError: invalid character '×' (U+00D7)
```

```
[1]: !jupyter nbconvert --to pdf Entregable 3.ipynb
```

This application is used to convert notebook files (*.ipynb)
to various other formats.

WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options

=====

The options below are convenience aliases to configurable class-options, as listed in the "Equivalent to" description-line of the aliases.

To see all configurable class-options for some <cmd>, use:

<cmd> --help-all

--debug

set log level to logging.DEBUG (maximize logging output)

Equivalent to: [--Application.log_level=10]

--show-config

Show the application's configuration (human-readable format)

Equivalent to: [--Application.show_config=True]

--show-config-json

Show the application's configuration (json format)

Equivalent to: [--Application.show_config_json=True]

--generate-config

generate default config file

Equivalent to: [--JupyterApp.generate_config=True]

-y

Answer yes to any questions instead of prompting.

Equivalent to: [--JupyterApp.answer_yes=True]

--execute

Execute the notebook prior to export.

Equivalent to: [--ExecutePreprocessor.enabled=True]

--allow-errors

Continue notebook execution even if one of the cells throws an error and include the error message in the cell output (the default behaviour is to abort conversion). This flag is only relevant if '--execute' was specified, too.

Equivalent to: [--ExecutePreprocessor.allow_errors=True]

--stdin

read a single notebook file from stdin. Write the resulting notebook with default basename 'notebook.*'

Equivalent to: [--NbConvertApp.from_stdin=True]

--stdout

Write notebook output to stdout instead of files.

Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]

--inplace

Run nbconvert in place, overwriting the existing notebook (only relevant when converting to notebook format)

Equivalent to: [--NbConvertApp.use_output_suffix=False]

--NbConvertApp.export_format=notebook --FilesWriter.build_directory=

--clear-output

Clear output of current file and save in place, overwriting the existing notebook.

Equivalent to: [--NbConvertApp.use_output_suffix=False]

--NbConvertApp.export_format=notebook --FilesWriter.build_directory=

--ClearOutputPreprocessor.enabled=True]

`--coalesce-streams`
 Coalesce consecutive stdout and stderr outputs into one stream (within each cell).

Equivalent to: `[--NbConvertApp.use_output_suffix=False`
`--NbConvertApp.export_format=notebook --FilesWriter.build_directory=`
`--CoalesceStreamsPreprocessor.enabled=True]`

`--no-prompt`
 Exclude input and output prompts from converted document.
 Equivalent to: `[--TemplateExporter.exclude_input_prompt=True`
`--TemplateExporter.exclude_output_prompt=True]`

`--no-input`
 Exclude input cells and output prompts from converted document.
 This mode is ideal for generating code-free reports.
 Equivalent to: `[--TemplateExporter.exclude_output_prompt=True`
`--TemplateExporter.exclude_input=True`
`--TemplateExporter.exclude_input_prompt=True]`

`--allow-chromium-download`
 Whether to allow downloading chromium if no suitable version is found on the system.
 Equivalent to: `[--WebPDFExporter.allow_chromium_download=True]`

`--disable-chromium-sandbox`
 Disable chromium security sandbox when converting to PDF..
 Equivalent to: `[--WebPDFExporter.disable_sandbox=True]`

`--show-input`
 Shows code input. This flag is only useful for dejavu users.
 Equivalent to: `[--TemplateExporter.exclude_input=False]`

`--embed-images`
 Embed the images as base64 dataurls in the output. This flag is only useful for the HTML/WebPDF/Slides exports.
 Equivalent to: `[--HTMLExporter.embed_images=True]`

`--sanitize-html`
 Whether the HTML in Markdown cells and cell outputs should be sanitized..
 Equivalent to: `[--HTMLExporter.sanitize_html=True]`

`--log-level=<Enum>`
 Set the log level by value or name.
 Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
 Default: 30
 Equivalent to: `[--Application.log_level]`

`--config=<Unicode>`
 Full path of a config file.
 Default: ''
 Equivalent to: `[--JupyterApp.config_file]`

`--to=<Unicode>`
 The export format to be used, either one of the built-in formats
 ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf']
 or a dotted object name that represents the import path for an

```

        ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme distributed
    as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be sanitized.This
    should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the
                                results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    Overwrite base name use for output files.
                                Supports pattern replacements '{notebook_name}'.
    Default: '{notebook_name}'
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                to output to the directory of each notebook.
To recover
                                previous default behaviour (outputting to the
current
                                working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
                                This defaults to the reveal CDN, but can be any url pointing to a

```

copy

of reveal.js.

For speaker notes to work, this must be a relative path to a local copy of reveal.js: e.g., "reveal.js".

If a relative path is given, it must be a subdirectory of the current directory (from which the server is run).

See the usage documentation

(<https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-slideshow>)

for more details.

Default: ''

Equivalent to: [--SlidesExporter.reveal_url_prefix]

--nbformat=<Enum>

The nbformat version to write.

Use this to downgrade notebooks.

Choices: any of [1, 2, 3, 4]

Default: 4

Equivalent to: [--NotebookExporter.nbformat_version]

Examples

The simplest way to use nbconvert is

```
> jupyter nbconvert mynotebook.ipynb --to html
```

Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'python', 'qtpdf', 'qtpng', 'rst', 'script', 'slides', 'webpdf'].

```
> jupyter nbconvert --to latex mynotebook.ipynb
```

Both HTML and LaTeX support multiple output templates. LaTeX includes

'base', 'article' and 'report'. HTML includes 'basic', 'lab' and 'classic'. You can specify the flavor of the format used.

```
> jupyter nbconvert --to html --template lab mynotebook.ipynb
```

You can also pipe the output to stdout, rather than a file

```
> jupyter nbconvert mynotebook.ipynb --stdout
```

PDF is generated via latex

```
> jupyter nbconvert mynotebook.ipynb --to pdf
```

You can get (and serve) a Reveal.js-powered slideshow

```
> jupyter nbconvert myslides.ipynb --to slides --post serve
```

Multiple notebooks can be given at the command line in a couple of different ways:

```
> jupyter nbconvert notebook*.ipynb
> jupyter nbconvert notebook1.ipynb notebook2.ipynb
```

or you can specify the notebooks list in a config file, containing::

```
c.NbConvertApp.notebooks = ["my_notebook.ipynb"]
```

```
> jupyter nbconvert --config mycfg.py
```

To see all available configurables, use `--help-all`.

```
[NbConvertApp] WARNING | pattern 'Entregable' matched no files
[NbConvertApp] WARNING | pattern '3.ipynb' matched no files
```