ANEXO

B. CODIGO INTEGRADO EN PYTHON

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt


# ----------------------------

# 1. Datos base de patentes

# ----------------------------

data = {

  "Patente": ["P1","P2","P3","P4","P5","P6","P7","P8","P9","P10"],

  "Año": [2010,2012,2014,2015,2017,2018,2020,2021,2023,2025],

  "Cp": [3,5,8,12,6,9,15,7,11,4],

  "Colaboracion": [

    "Sin colaboración","Intra-sectorial","Universidad–Empresa","Universidad–Empresa",

    "Intra-sectorial","Universidad–Empresa","Universidad–Empresa",

    "Intra-sectorial","Universidad–Empresa","Sin colaboración"

  ]

}


df = pd.DataFrame(data)


# ----------------------------

# 2. Mapear Ip
```

```
# ----------------------------
map_ip = {
   "Sin colaboración": 0,
   "Intra-sectorial": 0.5,
   "Universidad–Empresa": 1
}

df["Ip"] = df["Colaboracion"].map(map_ip)

# ----------------------------
# 3. Normalizar Cp
# ----------------------------
Cmax = df["Cp"].max()
Cmin = df["Cp"].min()

df["Cp_norm"] = (df["Cp"] - Cmin) / (Cmax - Cmin)

# ----------------------------
# 4. Producto
# ----------------------------
df["Producto"] = df["Cp_norm"] * df["Ip"]

# ----------------------------
# 5. IVCT por año
```

```
# ----------------------------
ivct_year = df.groupby("Año").agg({
    "Producto": "sum",
    "Patente": "count"
}).rename(columns={"Patente": "N_pat"}).reset_index()

ivct_year["IVCT"] = ivct_year["Producto"] / ivct_year["N_pat"]

# ----------------------------
# 6. Datos Scopus simulados
# ----------------------------
years = list(range(2010, 2026))

scopus = [20,25,30,35,42,50,60,75,90,110,130,150,180,210,240,280]

df_sci = pd.DataFrame({
    "Año": years,
    "Scopus": scopus
})

# ----------------------------
# 7. Patentes por año
# ----------------------------
pat_year = df.groupby("Año").size().reset_index(name="Patentes")
```

```
# ----------------------------
# 8. Unir todo
# ----------------------------
df_all = pd.merge(df_sci, pat_year, on="Año", how="left")
df_all = pd.merge(df_all, ivct_year[["Año","IVCT"]], on="Año", how="left")

df_all["Patentes"] = df_all["Patentes"].fillna(0)
df_all["IVCT"] = df_all["IVCT"].fillna(0)

# ----------------------------
# 9. Gráfica integrada
# ----------------------------
fig, ax1 = plt.subplots()

# Líneas principales
ax1.plot(df_all["Año"], df_all["Scopus"], marker='o', label="Scopus")
ax1.plot(df_all["Año"], df_all["Patentes"], marker='s', label="Patentes")

# Brecha (relleno)
ax1.fill_between(df_all["Año"], df_all["Scopus"], df_all["Patentes"], alpha=0.2)

ax1.set_xlabel("Año")
ax1.set_ylabel("Producción")
```

```
ax1.legend(loc="upper left")
ax1.grid()

# Segundo eje: IVCT
ax2 = ax1.twinx()
ax2.plot(df_all["Año"], df_all["IVCT"], linestyle='--', marker='d', label="IVCT")
ax2.set_ylabel("IVCT")

# Leyenda combinada
lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines1 + lines2, labels1 + labels2, loc="upper right")

plt.title("Integración Ciencia–Tecnología: Scopus vs Patentes vs IVCT")

plt.show()

# ----------------------------
# 10. Tabla final
# ----------------------------
print("\nTabla integrada:\n")
print(df_all.round(3))
```

```
import numpy as np
import matplotlib.pyplot as plt


# ----------------------------
# 1. Datos base de patentes
# ----------------------------
data = {
    "Patente": ["P1","P2","P3","P4","P5","P6","P7","P8","P9","P10"],
    "Año": [2010,2012,2014,2015,2017,2018,2020,2021,2023,2025],
    "Cp": [3,5,8,12,6,9,15,7,11,4],
    "Colaboracion": [
        "Sin colaboración","Intra-sectorial","Universidad–Empresa","Universidad–Empresa",
        "Intra-sectorial","Universidad–Empresa","Universidad–Empresa",
        "Intra-sectorial","Universidad–Empresa","Sin colaboración"
    ]
}

df = pd.DataFrame(data)


# ----------------------------
# 2. Mapear Ip
# ----------------------------
map_ip = {
```

```
    "Sin colaboración": 0,
    "Intra-sectorial": 0.5,
    "Universidad–Empresa": 1
}

df["Ip"] = df["Colaboracion"].map(map_ip)

# ----------------------------
# 3. Normalizar Cp
# ----------------------------
Cmax = df["Cp"].max()
Cmin = df["Cp"].min()

df["Cp_norm"] = (df["Cp"] - Cmin) / (Cmax - Cmin)

# ----------------------------
# 4. Producto
# ----------------------------
df["Producto"] = df["Cp_norm"] * df["Ip"]

# ----------------------------
# 5. IVCT por año
# ----------------------------
```

```
ivct_year = df.groupby("Año").agg({
    "Producto": "sum",
    "Patente": "count"
}).rename(columns={"Patente": "N_pat"}).reset_index()

ivct_year["IVCT"] = ivct_year["Producto"] / ivct_year["N_pat"]

# ----------------------------
# 6. Datos Scopus simulados
# ----------------------------
years = list(range(2010, 2026))

scopus = [20,25,30,35,42,50,60,75,90,110,130,150,180,210,240,280]

df_sci = pd.DataFrame({
    "Año": years,
    "Scopus": scopus
})

# ----------------------------
# 7. Patentes por año
# ----------------------------
pat_year = df.groupby("Año").size().reset_index(name="Patentes")
```

```
# ----------------------------

# 8. Unir todo

# ----------------------------

df_all = pd.merge(df_sci, pat_year, on="Año", how="left")

df_all = pd.merge(df_all, ivct_year[["Año","IVCT"]], on="Año", how="left")


df_all["Patentes"] = df_all["Patentes"].fillna(0)

df_all["IVCT"] = df_all["IVCT"].fillna(0)


# ----------------------------

# 9. Gráfica integrada

# ----------------------------

fig, ax1 = plt.subplots()


# Líneas principales

ax1.plot(df_all["Año"], df_all["Scopus"], marker='o', label="Scopus")

ax1.plot(df_all["Año"], df_all["Patentes"], marker='s', label="Patentes")


# Brecha (relleno)

ax1.fill_between(df_all["Año"], df_all["Scopus"], df_all["Patentes"], alpha=0.2)


ax1.set_xlabel("Año")
```

```
ax1.set_ylabel("Producción")

ax1.legend(loc="upper left")

ax1.grid()


# Segundo eje: IVCT

ax2 = ax1.twinx()

ax2.plot(df_all["Año"], df_all["IVCT"], linestyle='--', marker='d', label="IVCT")

ax2.set_ylabel("IVCT")


# Leyenda combinada

lines1, labels1 = ax1.get_legend_handles_labels()

lines2, labels2 = ax2.get_legend_handles_labels()

ax1.legend(lines1 + lines2, labels1 + labels2, loc="upper right")


plt.title("Integración Ciencia–Tecnología: Scopus vs Patentes vs IVCT")


plt.show()


# ------------------------------

# 10. Tabla final

# ------------------------------

print("\nTabla integrada:\n")

print(df_all.round(3))
```