Laboratorio de Física Computacional

Campo eléctrico y potencial en el capacitor de placas paralelas

Universidad de Sonora · Semestre 2026-1

Sesiones 1, 2 y 3 — material completo del estudiante

Contents

Part I

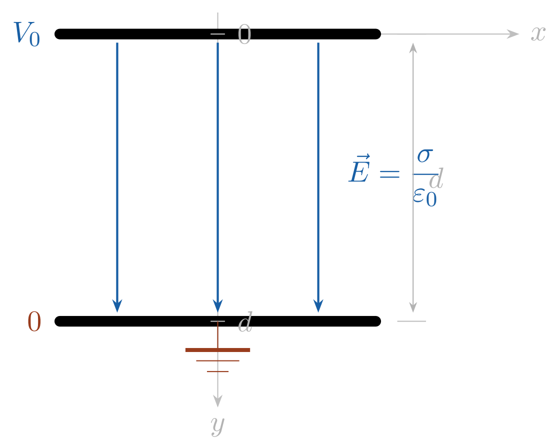
Capacitor en el vacío

1 El sistema físico

Un capacitor de placas paralelas consiste en dos placas conductoras de área A separadas una distancia d, conectadas a una fuente de voltaje V_0. Las placas son paralelas al eje x; el campo eléctrico apunta en la dirección $+y$.

Convención:

- Placa $+$ en $y = 0$, con potencial $V = V_0$.
- Placa $-$ en $y = d$, con potencial $V = 0$ (tierra).



2 Álgebra: campo y potencial

Punto de partida

La Ley de Gauss sobre la placa positiva da un campo uniforme:

$$E = \frac{\sigma}{\varepsilon_0} = \text{constante} \tag{1}$$

Integración

De $E = -dV/dy$ se obtiene $dV = -E\,dy$. Integrando con límites explícitos:

$$\int_{V_0}^{V(y)} dV' = -E \int_0^y dy' \quad \Longrightarrow \quad V(y) - V_0 = -E\,y \tag{2}$$

Condición de frontera

$V(d) = 0$ implica $E = V_0/d$.

$$E = \frac{V_0}{d} \qquad\qquad V(y) = V_0\left(1 - \frac{y}{d}\right) \tag{3}$$

Verificación: $V(0) = V_0 \checkmark \quad V(d) = 0 \checkmark$

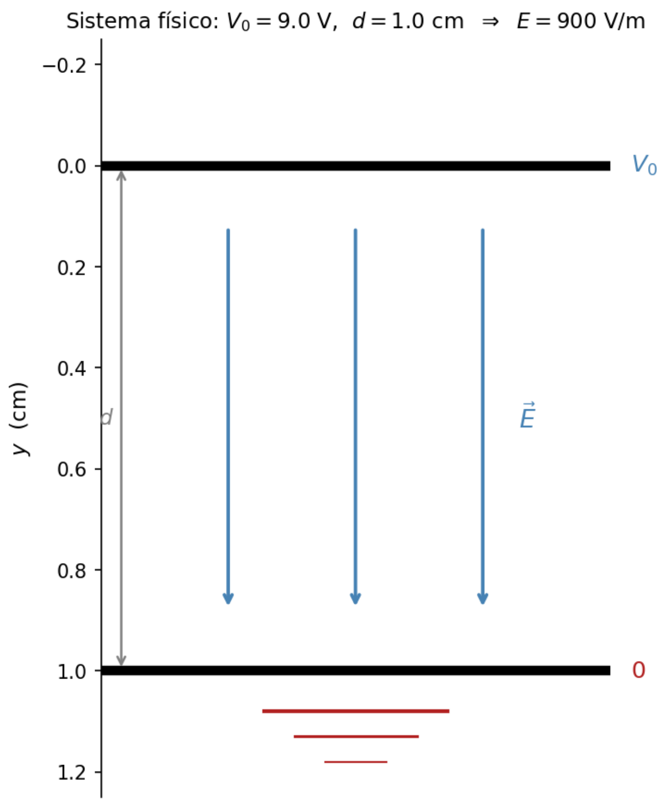
3 Scripts de Python — Capacitor en el vacío

Script 1 — Dibujo del sistema físico

Modifica d (cm) y vuelve a correr. La figura muestra las placas, las flechas de $\vec{E}$ y la cota d.

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

# ---
d  = 1.0   # separacion entre placas [cm]  <-- modifica
V0 = 9.0   # voltaje de la fuente     [V]

# ---
E0 = V0 / (d * 1e-2)   # E = V0/d [V/m]

# ---
fig, ax = plt.subplots(figsize=(5, 6))
ax.set_xlim(0, 1)
ax.set_ylim(-0.25, d + 0.25)
ax.invert_yaxis()
ax.set_ylabel("$y$ (cm)", fontsize=11)
ax.set_xticks([])
ax.spines[['top','right','bottom']].set_visible(False)

# placas
ax.axhline(0, color='black', lw=5, solid_capstyle='round')
ax.axhline(d, color='black', lw=5, solid_capstyle='round')

# etiquetas
ax.text(1.04, 0, r'$V_0$', va='center', fontsize=12,
        color='steelblue', transform=ax.get_yaxis_transform())
ax.text(1.04, d, '$0$', va='center', fontsize=12,
        color='firebrick', transform=ax.get_yaxis_transform())

# simbolo tierra
for i, w in enumerate([0.18, 0.12, 0.06]):
    ax.plot([0.5-w, 0.5+w], [d+0.08+i*0.05]*2,
            color='firebrick', lw=2-i*0.5)

# flechas E
for xf in [0.25, 0.50, 0.75]:
    ax.annotate("", xy=(xf, d*0.88), xytext=(xf, d*0.12),
                arrowprops=dict(arrowstyle='->', color='steelblue', lw=1.8))
ax.text(0.82, d/2, r'$\vec{E}$', va='center', fontsize=13, color='steelblue')

```

```python
# cota d
ax.annotate("", xy=(0.04, d), xytext=(0.04, 0),
            arrowprops=dict(arrowstyle='<->', color='gray', lw=1.2))
ax.text(0.01, d/2, '$d$', va='center', ha='center', fontsize=11, color='gray')

ax.set_title(rf"Sistema físico: $V_0={V0}$ V,  $d={d}$ cm"
             rf"  $\Rightarrow$  $E={E0:.0f}$ V/m", fontsize=11)
plt.tight_layout()
plt.savefig("s1_dibujo_vacio.png", dpi=150, bbox_inches='tight')
plt.show()
print(f"E = V0/d = {E0:.1f} V/m")
```

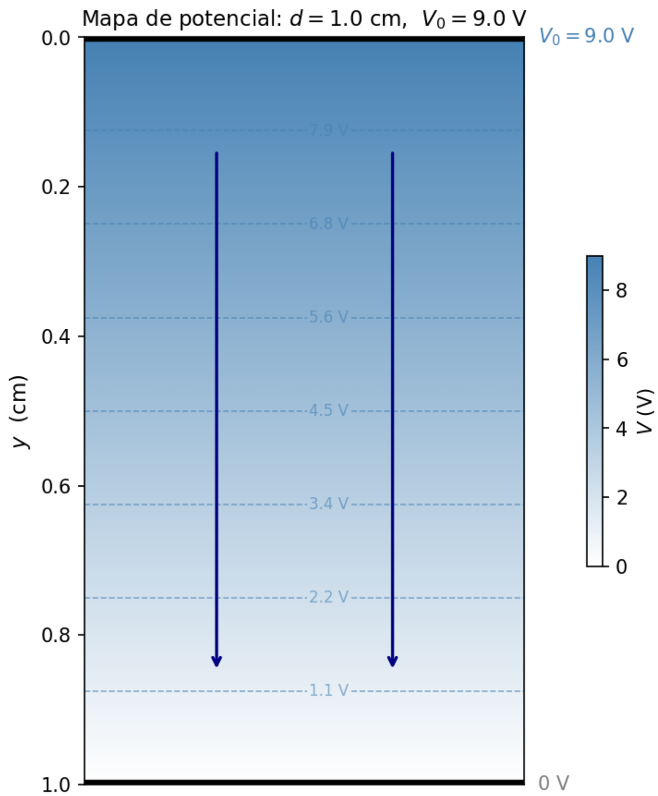


Script 2 — Mapa de color y curvas de nivel de $V(y)$

Modifica d y V0. El color azul indica V_0; el blanco indica 0 V. Las líneas punteadas son equipotenciales.

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors

# ---
d  = 1.0   # separacion entre placas [cm]  <-- modifica
V0 = 9.0   # voltaje de la fuente      [V]

# ---
```

```python
y = np.linspace(0, d, 300)
V = V0 * (1 - y / d)
x = np.linspace(0, 1, 10)
Y, X = np.meshgrid(y, x)
V2D  = V0 * (1 - Y / d)

cmap = mcolors.LinearSegmentedColormap.from_list(
    'azul_blanco', ['white', 'steelblue'])

fig, ax = plt.subplots(figsize=(5, 6))
im = ax.imshow(V2D.T, origin='upper', extent=[0,1,d,0],
               cmap=cmap, vmin=0, vmax=V0, aspect='auto')
niveles = np.linspace(0, V0, 9)[1:-1]
cs = ax.contour(X, Y, V2D, levels=niveles,
                colors='steelblue', linewidths=0.8,
                linestyles='--', alpha=0.6)
ax.clabel(cs, fmt='%.1f V', fontsize=8, inline=True)
ax.axhline(0, color='black', lw=5)
ax.axhline(d, color='black', lw=5)
for xf in [0.3, 0.7]:
    ax.annotate("", xy=(xf, d*0.85), xytext=(xf, d*0.15),
                arrowprops=dict(arrowstyle='->', color='navy', lw=1.6))
ax.text(1.03, 0, f'$V_0={V0}$ V', va='center', fontsize=10,
        color='steelblue', transform=ax.get_yaxis_transform())
ax.text(1.03, d, '0 V', va='center', fontsize=10,
        color='gray', transform=ax.get_yaxis_transform())
fig.colorbar(im, ax=ax, fraction=0.03, pad=0.12).set_label('$V$ (V)')
ax.set_ylabel('$y$  (cm)', fontsize=11)
ax.set_xticks([])
ax.set_title(rf"Mapa de potencial: $d={d}$ cm,  $V_0={V0}$ V", fontsize=11)
plt.tight_layout()
plt.savefig("s2_mapa_vacio.png", dpi=150, bbox_inches='tight')
plt.show()
```

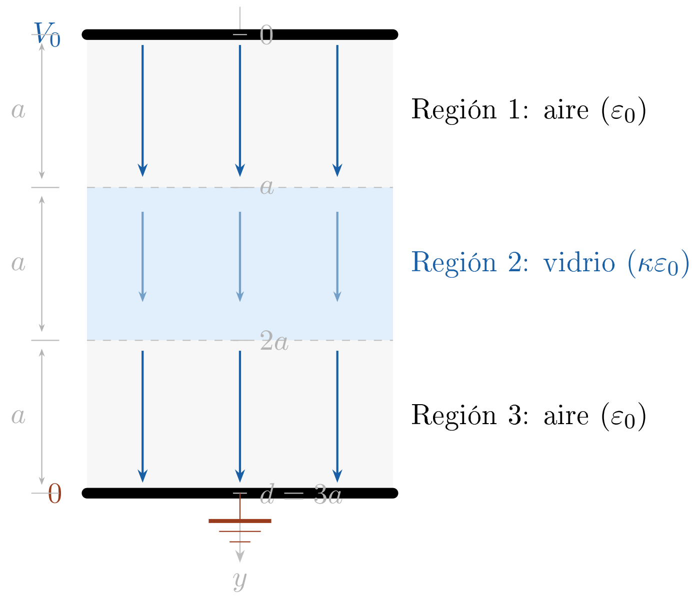

Mapa de potencial: $d = 1.0$ cm, $V_0 = 9.0$ V
$V_0 = 9.0$ V
0 V
7.9 V
6.8 V
5.6 V
4.5 V
3.4 V
2.2 V
1.1 V
0.0
0.2
0.4
0.6
0.8
1.0
y (cm)
V (V)
0
2
4
6
8

Part II

Capacitor con dieléctrico parcial

4 El sistema físico con dieléctrico

Insertamos una lámina de vidrio ($\kappa = 2.25$) centrada entre las placas. El espacio queda dividido simétricamente en tres regiones de igual grosor $a = d/3$:



5 Álgebra: campo y potencial con dieléctrico

Condición de frontera en la interfaz

Sin carga libre superficial, $\vec{D}$ normal es continuo:

$$\varepsilon_0 E_1 = \kappa \varepsilon_0 E_2 \quad \Longrightarrow \quad E_2 = \frac{E_1}{\kappa}, \quad E_3 = E_1 \tag{4}$$

Despeje de E_1

La integral total de E debe dar V_0:

$$V_0 = E_1 a + \frac{E_1}{\kappa} a + E_1 a = E_1 a \left(2 + \frac{1}{\kappa}\right) \tag{5}$$

$$E_1 = E_3 = \frac{V_0}{a\left(2 + \frac{1}{\kappa}\right)} \qquad E_2 = \frac{E_1}{\kappa} \tag{6}$$

Potencial a trozos

$$V(y) = \begin{cases} V_0 - E_1\, y & 0 \le y \le a \\ (V_0 - E_1 a) - \dfrac{E_1}{\kappa}(y-a) & a \le y \le 2a \\ \left(V_0 - E_1 a - \dfrac{E_1}{\kappa} a\right) - E_1(y-2a) & 2a \le y \le 3a \end{cases} \tag{7}$$

Verificación: $V(0) = V_0 \checkmark \quad V(3a) = V_0 - E_1 a(2 + 1/\kappa) = 0 \checkmark$

6 Scripts de Python — Capacitor con dieléctrico

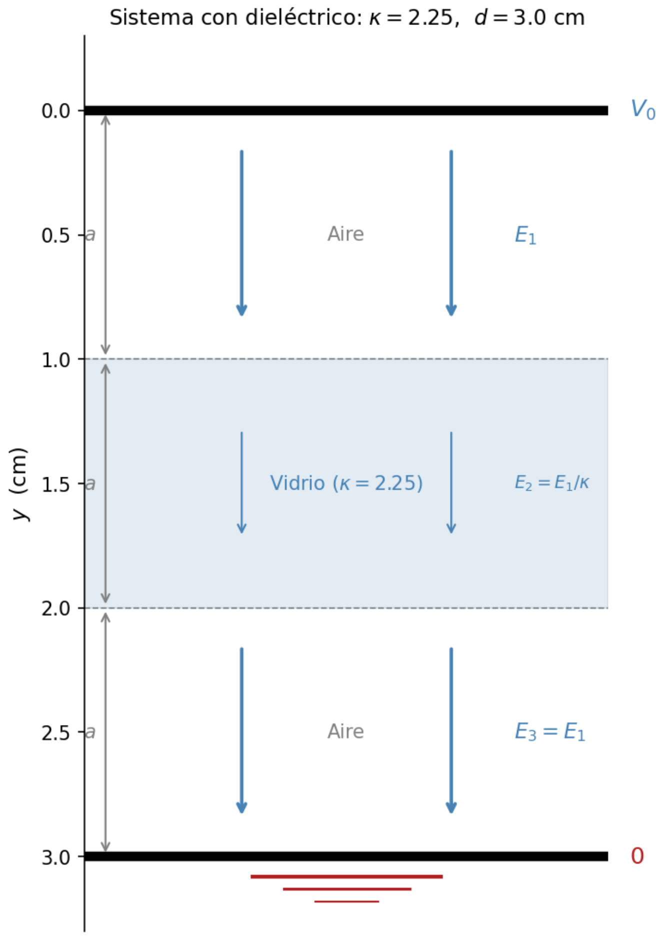
Script 3 — Dibujo del sistema con dieléctrico

Modifica kappa y observa cómo cambia la longitud de las flechas en cada región.

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

# ---
d_cm  = 3.0    # separacion total [cm]  (d = 3a)
V0    = 9.0    # voltaje             [V]
kappa = 2.25   # constante dielectrica del vidrio

# ---
a   = d_cm / 3.0
a_m = a * 1e-2
E1  = V0 / (a_m * (2 + 1/kappa))
E2  = E1 / kappa

fig, ax = plt.subplots(figsize=(5, 7))
ax.set_xlim(0, 1)
ax.set_ylim(-0.3, d_cm + 0.3)
ax.invert_yaxis()
ax.set_ylabel("$y$ (cm)", fontsize=11)
ax.set_xticks([])
ax.spines[['top','right','bottom']].set_visible(False)

# regiones
ax.axhspan(0,   a,    color='white',     alpha=0.0)
ax.axhspan(a,   2*a,  color='steelblue', alpha=0.15)
ax.axhspan(2*a, d_cm, color='white',     alpha=0.0)

```

```python
# interfaces
for yi in [a, 2*a]:
    ax.axhline(yi, color='gray', lw=0.8, ls='--')

# etiquetas regiones
ax.text(0.5, a/2,     'Aire',     ha='center', va='center',
        fontsize=10, color='gray')
ax.text(0.5, a+a/2,  f'Vidrio ($\\kappa={kappa}$)', ha='center',
        va='center', fontsize=10, color='steelblue')
ax.text(0.5, 2*a+a/2,'Aire',    ha='center', va='center',
        fontsize=10, color='gray')

# placas
ax.axhline(0,     color='black', lw=5, solid_capstyle='round')
ax.axhline(d_cm, color='black', lw=5, solid_capstyle='round')

# etiquetas potencial
ax.text(1.04, 0,     r'$V_0$', va='center', fontsize=12,
        color='steelblue', transform=ax.get_yaxis_transform())
ax.text(1.04, d_cm, '$0$',    va='center', fontsize=12,
        color='firebrick', transform=ax.get_yaxis_transform())

# tierra
for i, w in enumerate([0.18, 0.12, 0.06]):
    ax.plot([0.5-w, 0.5+w], [d_cm+0.08+i*0.05]*2,
            color='firebrick', lw=2-i*0.5)

# flechas E (más largas en aire, más cortas en vidrio)
for xf in [0.3, 0.7]:
    ax.annotate("", xy=(xf, a*0.85),     xytext=(xf, a*0.15),
                arrowprops=dict(arrowstyle='->', color='steelblue', lw=1.8))
    ax.annotate("", xy=(xf, a+a*0.72),  xytext=(xf, a+a*0.28),
                arrowprops=dict(arrowstyle='->', color='steelblue', lw=1.0))
    ax.annotate("", xy=(xf, 2*a+a*0.85),xytext=(xf, 2*a+a*0.15),
                arrowprops=dict(arrowstyle='->', color='steelblue', lw=1.8))

# labels E
ax.text(0.82, a/2,      r'$E_1$', va='center', fontsize=11, color='steelblue')
ax.text(0.82, a+a/2,    r'$E_2=E_1/\kappa$', va='center',
        fontsize=9, color='steelblue')
ax.text(0.82, 2*a+a/2, r'$E_3=E_1$', va='center', fontsize=11,
        color='steelblue')

# cotas
for y0, y1, lab in [(0,a,'a'), (a,2*a,'a'), (2*a,d_cm,'a')]:
    ax.annotate("", xy=(0.04, y1), xytext=(0.04, y0),
                arrowprops=dict(arrowstyle='<->', color='gray', lw=1.0))
    ax.text(0.01, (y0+y1)/2, f'${lab}$', va='center',
            ha='center', fontsize=10, color='gray')

ax.set_title(rf"Sistema con dieléctrico: $\kappa={kappa}$,  $d={d_cm}$ cm",
             fontsize=11)
plt.tight_layout()
plt.savefig("s3_dibujo_dielectrico.png", dpi=150, bbox_inches='tight')
plt.show()
```

```
print(f"E1 = {E1:.1f} V/m  |  E2 = {E2:.1f} V/m  |  E2/E1 = {E2/E1:.4f}")
```



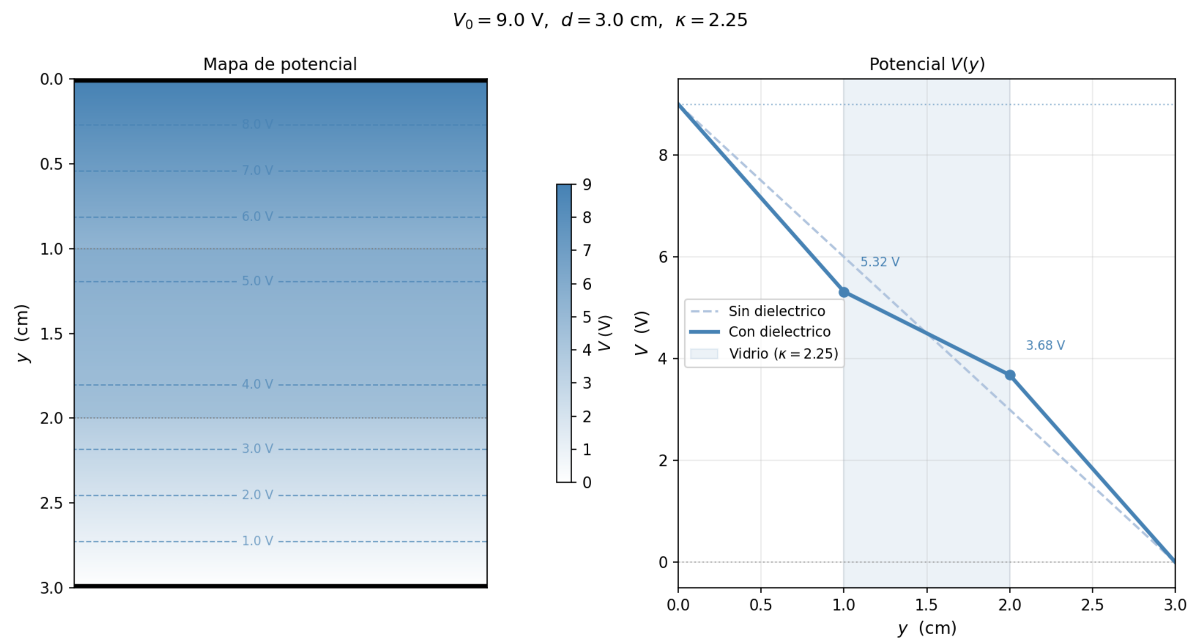
Script 4 — Mapa de color y $V(y)$ con dieléctrico

Compara la curva con y sin dieléctrico. Las equipotenciales se aprietan dentro del vidrio porque $E_2 < E_1$.

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

# ---
V0    = 9.0     # voltaje [V]
d_cm  = 3.0     # separacion total [cm]
kappa = 2.25    # constante dielectrica

# ---
a   = d_cm / 3.0
a_m = a * 1e-2
E1  = V0 / (a_m * (2 + 1/kappa))
E2  = E1 / kappa
```

```python
Va  = V0 - E1 * a_m
V2a = Va - E2 * a_m

y  = np.linspace(0, d_cm, 500)
V  = np.where(y <= a,
       V0  - E1 * y * 1e-2,
     np.where(y <= 2*a,
       Va  - E2 * (y-a) * 1e-2,
       V2a - E1 * (y-2*a) * 1e-2))

# malla 2D: Y varia en filas, X en columnas
nx, ny = 10, 400
xv = np.linspace(0, 1, nx)
yv = np.linspace(0, d_cm, ny)
X2D, Y2D = np.meshgrid(xv, yv)          # shape (ny, nx)
V2D = np.where(Y2D <= a,
       V0  - E1 * Y2D * 1e-2,
     np.where(Y2D <= 2*a,
       Va  - E2 * (Y2D-a) * 1e-2,
       V2a - E1 * (Y2D-2*a) * 1e-2))   # shape (ny, nx)

cmap = mcolors.LinearSegmentedColormap.from_list(
    'azul_blanco', ['white', 'steelblue'])

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(11, 6))
fig.suptitle(rf"$V_0={V0}$ V,  $d={d_cm}$ cm,  $\kappa={kappa}$",
             fontsize=12)

# mapa de color
im = ax1.imshow(V2D, origin='upper', extent=[0,1,d_cm,0],
                cmap=cmap, vmin=0, vmax=V0, aspect='auto')
ax1.axhspan(a, 2*a, color='steelblue', alpha=0.15)
cs = ax1.contour(X2D, Y2D, V2D,
                 levels=np.linspace(0, V0, 10)[1:-1],
                 colors='steelblue', linewidths=0.9,
                 linestyles='--', alpha=0.7)
ax1.clabel(cs, fmt='%.1f V', fontsize=8, inline=True)
for yi in [a, 2*a]:
    ax1.axhline(yi, color='gray', lw=0.8, ls=':')
ax1.axhline(0,    color='black', lw=5)
ax1.axhline(d_cm, color='black', lw=5)
fig.colorbar(im, ax=ax1, fraction=0.03, pad=0.14).set_label('$V$ (V)')
ax1.set_ylabel('$y$  (cm)', fontsize=11)
ax1.set_xticks([])
ax1.set_title('Mapa de potencial', fontsize=11)

# V(y) comparacion
V_ref = V0 * (1 - y / d_cm)
ax2.plot(y, V_ref, color='lightsteelblue', lw=1.5,
         ls='--', label='Sin dielectrico')
ax2.plot(y, V, color='steelblue', lw=2.5,
         label='Con dielectrico')
ax2.axvspan(a, 2*a, color='steelblue', alpha=0.10,
            label=f'Vidrio ($\\kappa={kappa}$)')
for yi, vi in [(a, Va), (2*a, V2a)]:
```

```python
    ax2.plot(yi, vi, 'o', color='steelblue', ms=6, zorder=5)
    ax2.annotate(f'{vi:.2f} V', xy=(yi, vi),
                 xytext=(yi+0.1, vi+0.5), fontsize=8, color='steelblue')
ax2.axhline(V0, color='steelblue', ls=':', lw=1, alpha=0.5)
ax2.axhline(0,  color='gray',      ls=':', lw=1, alpha=0.5)
ax2.set_xlabel('$y$  (cm)', fontsize=11)
ax2.set_ylabel('$V$  (V)',  fontsize=11)
ax2.set_xlim(0, d_cm)
ax2.set_ylim(-0.5, V0 + 0.5)
ax2.legend(fontsize=9)
ax2.grid(alpha=0.25)
ax2.set_title('Potencial $V(y)$', fontsize=11)

plt.tight_layout()
plt.savefig("s4_mapa_dielectrico.png", dpi=150, bbox_inches='tight')
plt.show()
```



Part III

Actividades, problemas y planeación

7 Actividades de exploración

A1. Efecto de d (Script 1 y 2). Prueba $d = 0.5$, 1, 2, 4 cm con $V_0 = 9$ V fijo. Registra E en cada caso. ¿Qué relación encuentras entre E y d?

A2. Efecto de V_0 (Script 2). Fija $d = 1$ cm y prueba $V_0 = 3$, 6, 9, 12 V. ¿Cambia la forma de $V(y)$? ¿Cambia E?

A3. Potencial no lineal. En el Script 2 reemplaza la línea de V por:

$$V = V_0\left(1 - \left(\frac{y}{d}\right)^2\right)$$

Calcula $E = -dV/dy$ con numpy.gradient y grafica. ¿Sigue siendo uniforme?

A4. Efecto de κ (Script 4). Prueba $\kappa = 1.0$, 2.25, 5.0, 100. ¿Qué ocurre con $V(y)$ dentro del vidrio cuando $\kappa \to \infty$?

A5. Dieléctrico asimétrico (Script 3 y 4). Modifica el script para que el vidrio ocupe $b = d/2$ y el aire $a = d/4$ en cada lado. Recalcula E_1, E_2 y $V(y)$ y verifica numéricamente.

8 Problemas numéricos

P1. Un capacitor tiene $A = 50\,\mathrm{cm}^2$, $d = 2\,\mathrm{mm}$ y $V_0 = 12\,\mathrm{V}$.

a) Calcula $C = \varepsilon_0 A/d$.
b) Calcula $q = CV_0$.
c) Calcula $E = V_0/d$.
d) Calcula la energía almacenada $U = \frac{1}{2}CV_0^2$.

P2. Para el capacitor del Problema 1, inserta vidrio ($\kappa = 2.25$) en la mitad central del espacio (geometría simétrica $a = b = a$, $d = 3a$).

a) Calcula E_1, E_2, E_3.
b) Calcula $V(a)$ y $V(2a)$.
c) ¿Cuánto cambia la energía almacenada?

P3. Demuestra que la capacitancia efectiva del sistema aire/vidrio/aire simétrico es:

$$C' = \frac{\varepsilon_0 A}{a\left(2 + \frac{1}{\kappa}\right)}$$

¿Es mayor o menor que $C_0 = \varepsilon_0 A/d$? ¿Por qué?

P4. Ejemplo resuelto. Con $V_0 = 9$ V, $d = 3$ cm, $\kappa = 2.25$, $a = 1$ cm:

$$\begin{aligned} E_1 &= \frac{9}{0.01 \cdot (2 + 1/2.25)} = 368.2\,\mathrm{V/m} \\ E_2 &= 368.2/2.25 = 163.6\,\mathrm{V/m} \\ V(a) &= 9 - 368.2 \times 0.01 = 5.32\,\mathrm{V} \\ V(2a) &= 5.32 - 163.6 \times 0.01 = 3.68\,\mathrm{V} \end{aligned}$$

Verifica estos valores con el Script 4.

9 Planeación de las sesiones

Sesión	Tiempo	Contenido	Tipo
1	0:00–0:10	Introducción: pila de 9 V, convención de tierra	Discusión
	0:10–0:30	Ley de Gauss $\to E$, integral $\to V(y)$	Álgebra
*	0:30–0:45	Script 1: dibujo del sistema, modificar d	Python
	0:45–1:05	Script 2: mapa de color, curvas de nivel	Python
	1:05–1:30	Actividades A1, A2, A3, discusión	Exploración
2	0:00–0:10	Repaso. ¿Qué cambia con dieléctrico?	Discusión
	0:10–0:35	Condición D continua, despeje de E_1	Álgebra
*	0:35–0:50	Script 3: dibujo con tres regiones	Python
	0:50–1:15	Script 4: mapa de color, comparación	Python
	1:15–1:30	Actividades A4, A5	Exploración
3	0:00–0:20	Problemas P1 y P2 en clase	Álgebra
*	0:20–0:50	Actividad libre: cada estudiante elige parámetros	Python
	0:50–1:20	Problema P3 (capacitancia efectiva), P4 verificación	Discusión
	1:20–1:30	Cierre: conexión con el problema 30-9 del Resnick	Discusión