



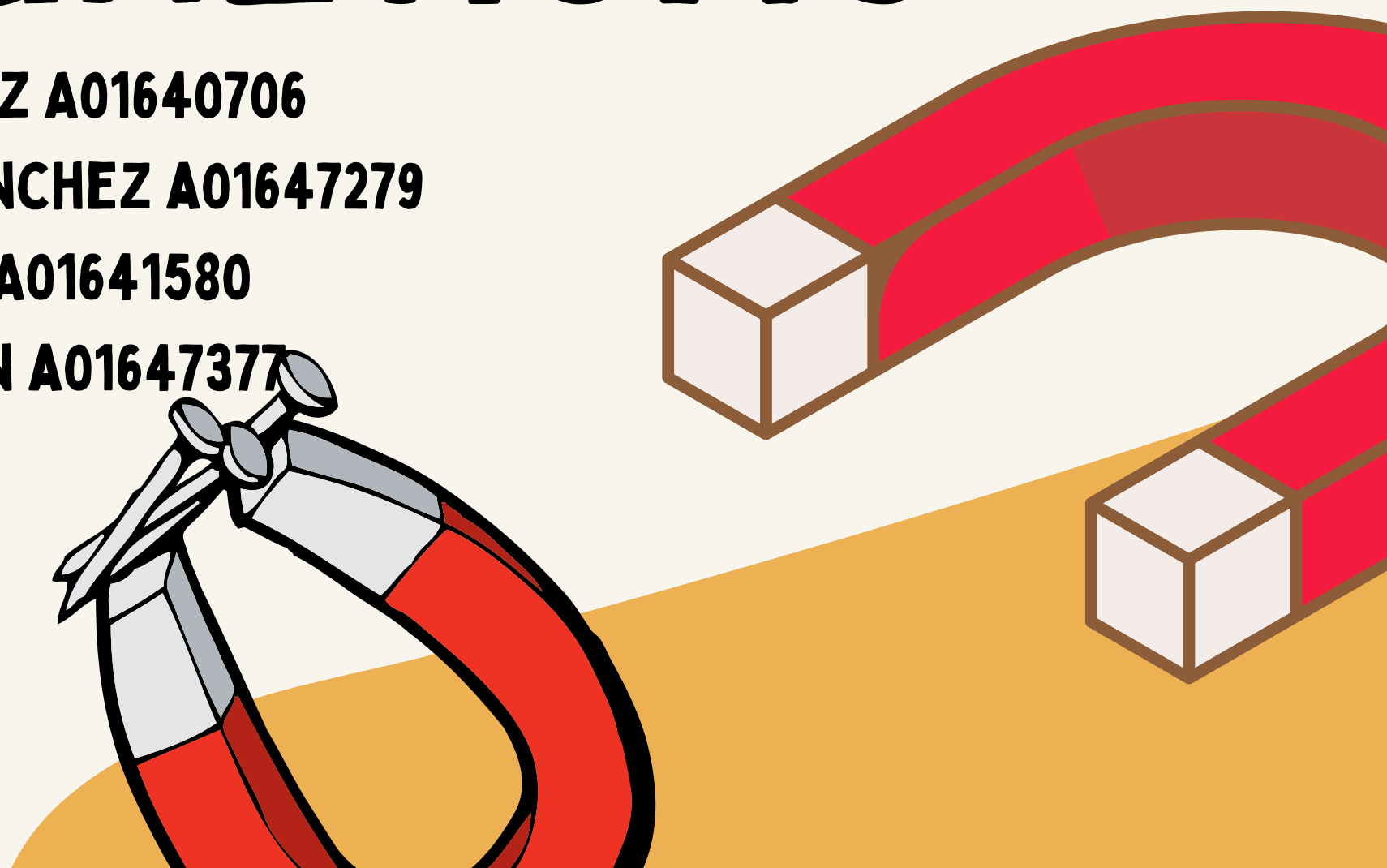
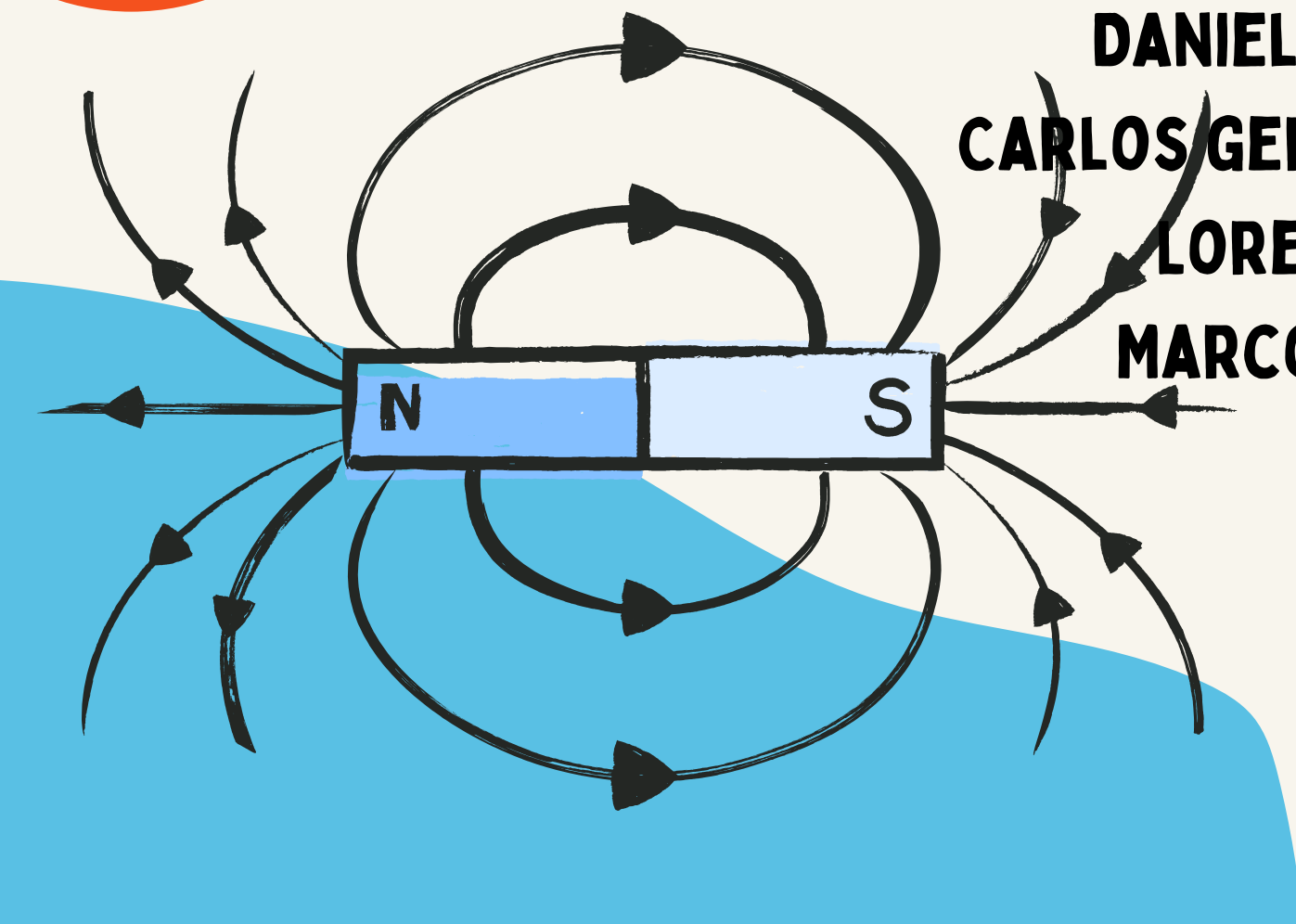
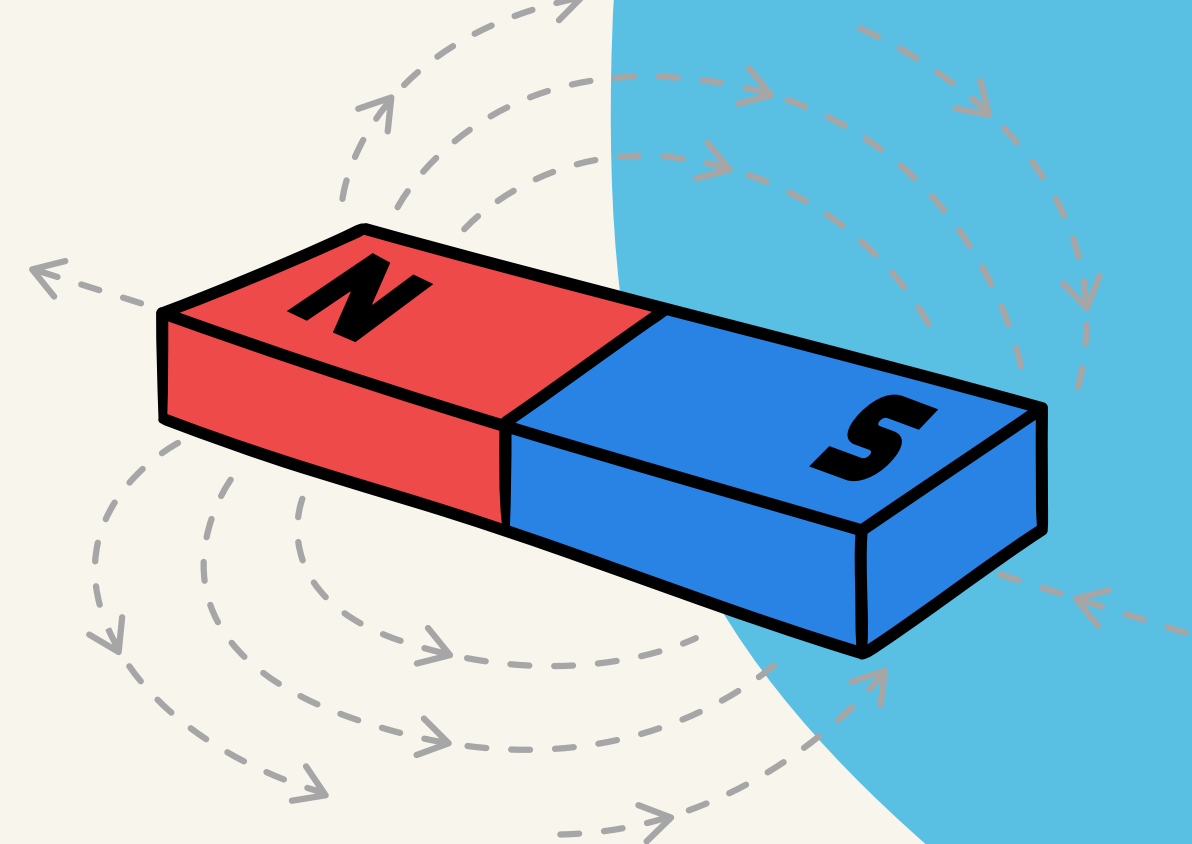
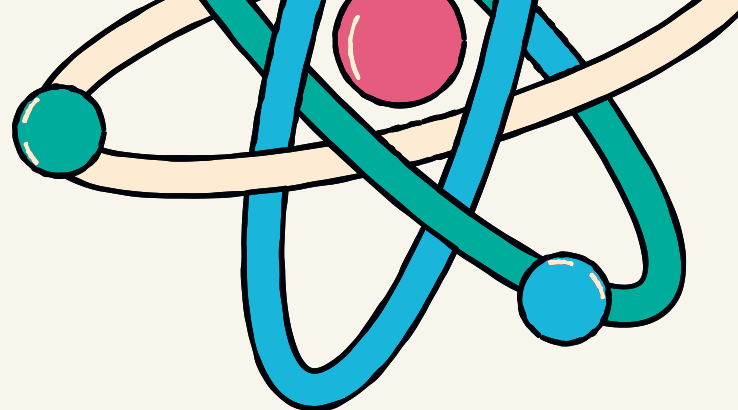
RETO ELECTROMAGNETISMO

DANIEL HERNÁNDEZ GUTIÉRREZ A01640706

CARLOS GERARDO GUYONNET SÁNCHEZ A01647279

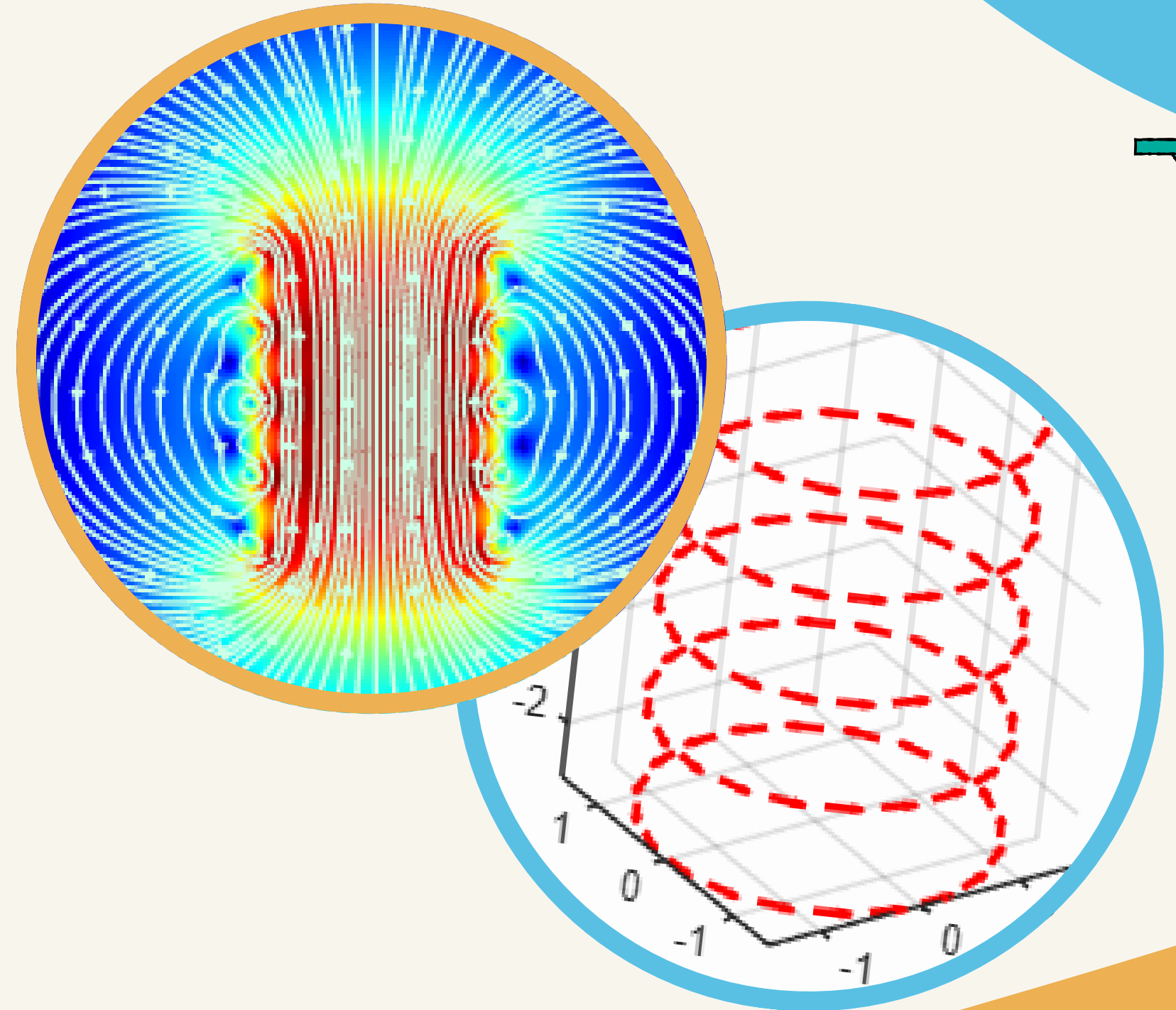
LORENZO ORRANTE ROMÁN A01641580

MARCO NATSUMI RABIELA MUN A01647377



INTRODUCCIÓN

El objetivo principal del experimento con las espiras es mostrar el campo magnético generado por un solenoide donde se podrá identificar las partes donde el campo magnético tiene una mayor intensidad.



MARCO TEÓRICO

01

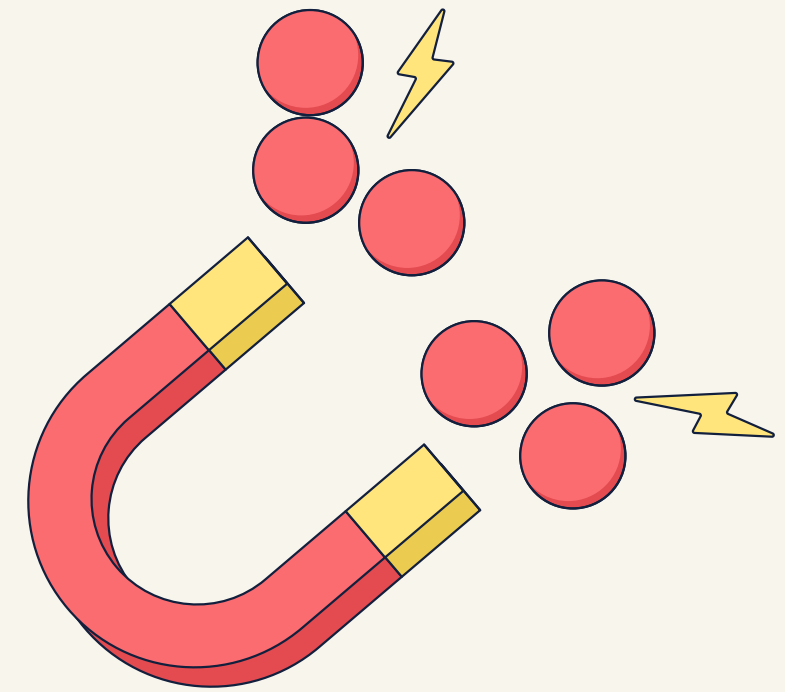
MAGNETISMO

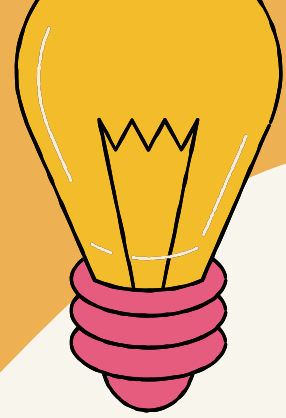
El magnetismo es una manifestación de una de las fuerzas fundamentales de la naturaleza: la fuerza electromagnética. Surge del movimiento de cargas eléctricas, especialmente de electrones en los átomos, y se manifiesta a través de fuerzas entre materiales magnéticos o corrientes eléctricas, es decir, es un fenómeno físico por el que los objetos ejercen fuerzas de atracción o repulsión sobre otros materiales

02

CAMPO MAGNÉTICO

El campo magnético es una región del espacio donde una carga en movimiento (como una corriente eléctrica) experimenta una fuerza magnética. Se trata de un campo vectorial, es decir, en cada punto del espacio tiene una dirección y una magnitud.





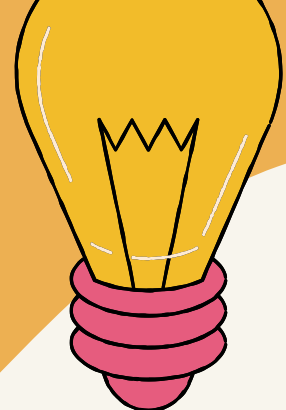
LEY DE BIOT-SAVART

La ley de Biot-Savart describe el campo magnético generado por una corriente eléctrica constante. La dirección, magnitud y sentido dependen de:

- La dirección y magnitud de la corriente
- La forma del conductor
- La distancia en el punto donde se mide el campo y el elemento de la corriente

$$\vec{B} = \frac{\mu_0 I}{4\pi} \int \frac{d\vec{l} \times \vec{r}}{r^3}$$



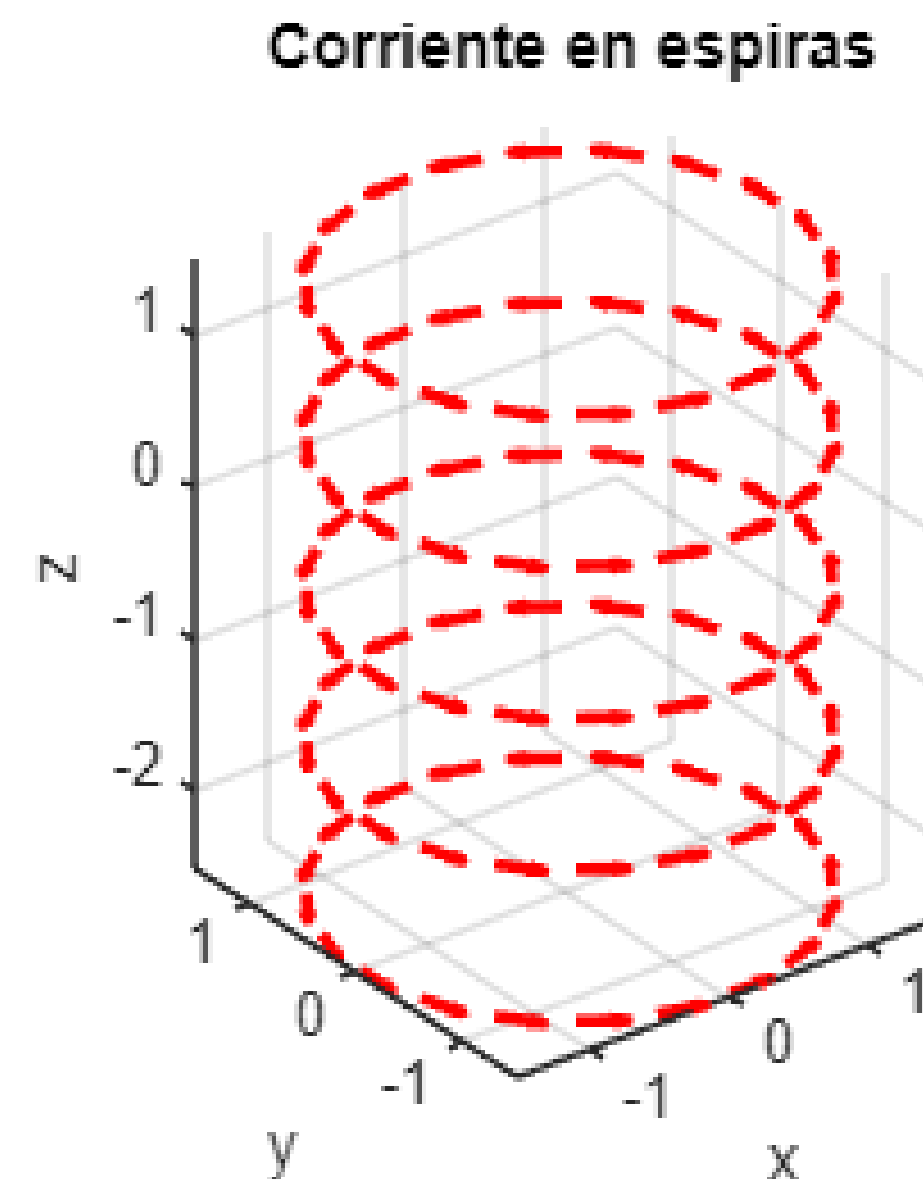


ESTABLECIMIENTO DE VARIABLES Y GRAFICA DE ESPIRAS

En el inicio del código establecimos los puntos que van a haber en cada espira, la derivada de theta y su ángulo, la separación entre cada espira, valores de unidades establecidas para el cálculo del campo magnético, etc. Se establece un ciclo for para graficar cada punto de la espira y la cantidad de espiras que van a ser graficadas y será graficada con quiver para mostrar la dirección del campo magnético con flechas.

```
N = 20;  
s = 1;  
ne = 5;  
dtheta = (2*pi)/N;  
angulo = 0:dtheta:(2*pi - dtheta);  
radio = 1.5;  
h = 1;  
I = 300;  
mo = 4*pi*1e-7;  
km = mo * I / (4*pi);  
rw = 0.2;
```

```
for i = 1:ne  
    x(s:s + N - 1) = radio * cos(angulo);  
    y(s:s + N - 1) = radio * sin(angulo);  
    z(s:s + N - 1) = -ne/2 * h + (i - 1) * h;  
    dx(s:s + N - 1) = -y(s:s + N - 1) * dtheta;  
    dy(s:s + N - 1) = x(s:s + N - 1) * dtheta;  
    s = s + N;  
end
```



CÁLCULO DEL CAMPO MAGNÉTICO



Tras graficar el solenoide, pasamos a definir el área de la gráfica del campo magnético, y creamos vectores de ceros para almacenar las componentes del campo magnético.

CICLO FOR

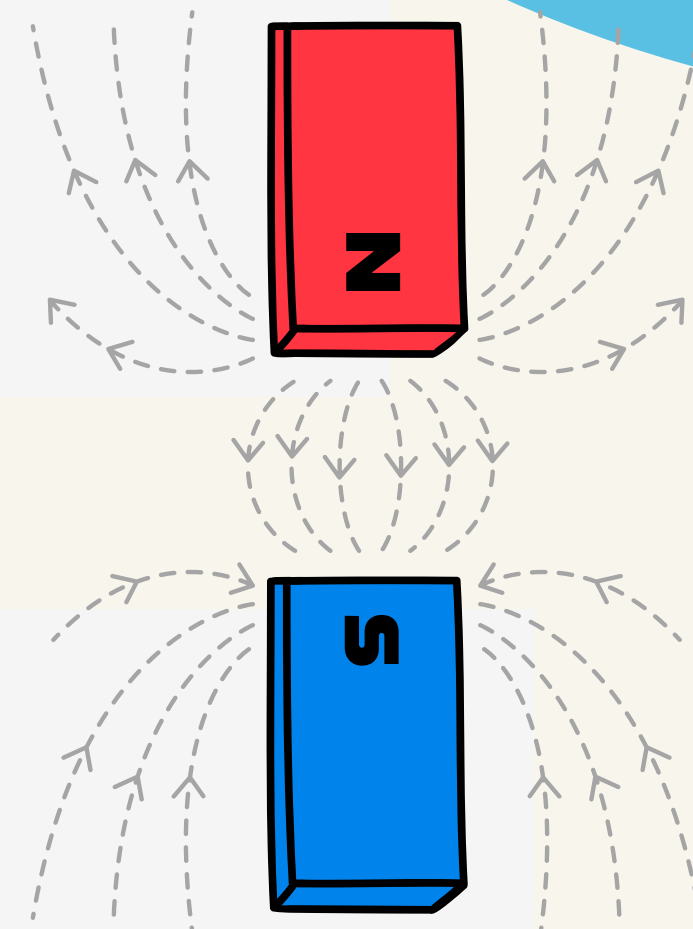
Ahora pasamos a calcular el campo magnético para después graficarlo. Esto se hizo mediante un triple ciclo for que primero calcula las distancias de cada componente entre cada punto a evaluar, y después mediante la Ley de Biot-Savart calculamos las componentes del campo magnético en cada punto del campo.

```
xcampo = -5:0.1:5; ycampo = xcampo; zcampo = xcampo;
```

```
lx = length(xcampo);  
ly = length(ycampo);  
lz = length(zcampo);
```

```
Bx = zeros(lx, ly, lz);  
By = zeros(lx, ly, lz);  
Bz = zeros(lx, ly, lz);
```

```
tic  
for i = 1:lx  
    for j = 1:ly  
        for k = 1:lz  
            for l = 1:ne*N  
                rx = xcampo(i) - x(l);  
                ry = ycampo(j) - y(l);  
                rz = zcampo(k) - z(l);  
  
                r = sqrt(rx^2 + ry^2 + rz^2 + rw^2);  
                r3 = r^3;  
  
                Bx(i, j, k) = Bx(i, j, k) + km * (dy(l) * rz - dz(l) * ry) / r3;  
                By(i, j, k) = By(i, j, k) + km * (dz(l) * rx - dx(l) * rz) / r3;  
                Bz(i, j, k) = Bz(i, j, k) + km * (dx(l) * ry - dy(l) * rx) / r3;  
            end  
        end  
    end  
end  
toc
```



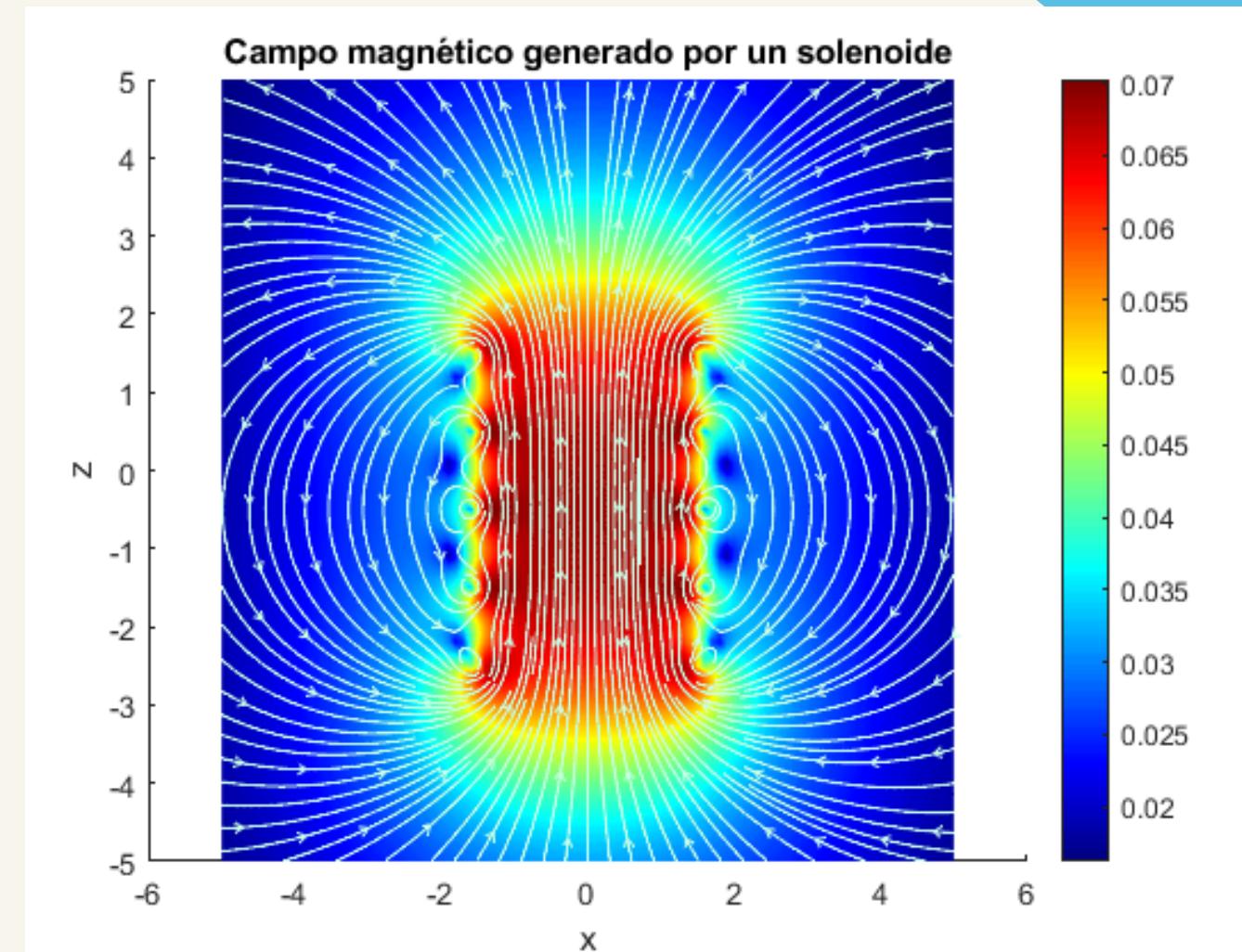
GRAFICA DEL CAMPO MAGNÉTICO

```
% Graficar plano XZ
Bmag = sqrt(Bx.^2 + By.^2 + Bz.^2);
centery = round(ly / 2);
Bx_xz = squeeze(Bx(:, centery, :));
Bz_xz = squeeze(Bz(:, centery, :));
Bxz = squeeze(Bmag(:, centery, :));

figure(2);
hold on;
pcolor(xcampo, zcampo, (Bxz').^(1/3)); shading interp; colormap jet; colorbar;
h1 = streamslice(xcampo, zcampo, Bx_xz', Bz_xz', 3);
set(h1, 'Color', [0.8 1.0 0.9]);

xlabel('x');
ylabel('z');
title('Campo magnético generado por un solenoide');
```

Se empieza calculando la magnitud del campo magnético ya que este se va a utilizar para representar visualmente el campo en 2D usando centery y squeeze.



Para que podamos identificar la intensidad del campo magnético usamos pcolor con la que vamos a graficar un mapa de colores para que la magnitud en cada parte de la gráfica sea identificable con los colores.

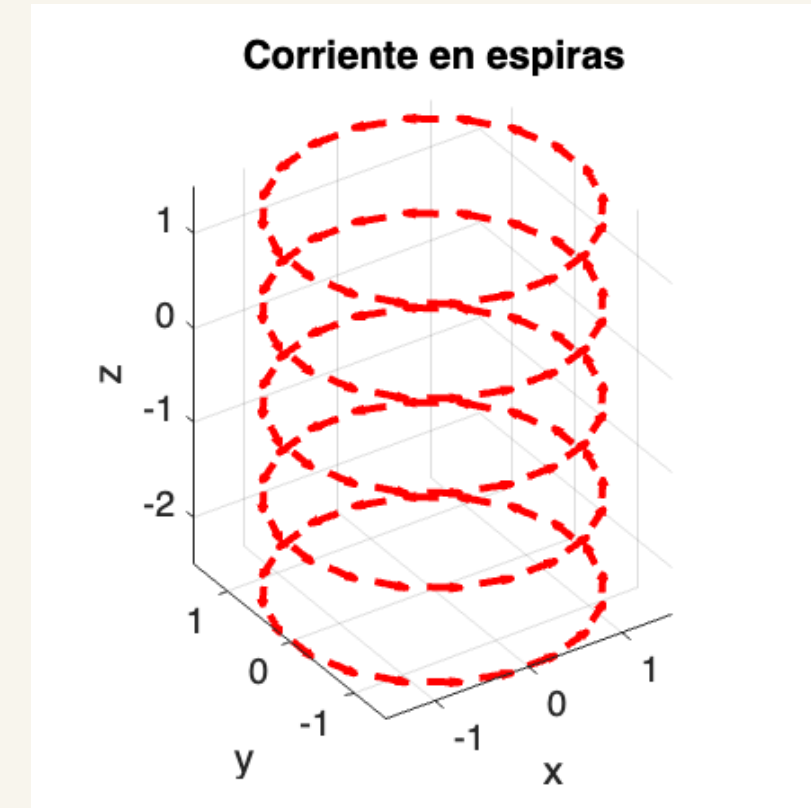
FUNCIÓN ESPIRAS

```
function [Px, Py, Pz, dx, dy, dz] = espiras(ne, N, radio)
    dtheta = (2*pi)/N;
    angulo = 0:dtheta:(2*pi - dtheta);
    s = 1;
    h = 1;
    for i = 1:ne
        Px(s:s + N - 1) = radio * cos(angulo);
        Py(s:s + N - 1) = radio * sin(angulo);
        Pz(s:s + N - 1) = -ne/2 * h + (i - 1) * h;
        dx(s:s + N - 1) = -Py(s:s + N - 1) * dtheta;
        dy(s:s + N - 1) = Px(s:s + N - 1) * dtheta;
        s = s + N;
    end

    dz = zeros(1, N * ne);

    figure(1);
    quiver3(Px, Py, Pz, dx, dy, dz, 0.5, '-r', 'LineWidth', 2);
    view(-34, 33);
    title('Corriente en espiras');
    xlabel('x'); ylabel('y'); zlabel('z');
    axis equal;
    grid on;
end
```

La función de espiras genera un conjunto de espiras circulares, modelando la posición y dirección del alambre. Simula un solenoide hecho por varias vueltas circulares por donde circula la corriente.



Los parámetros de entrada son ne (número de espiras), N (puntos discretos por cada espira) y el radio de cada espira.

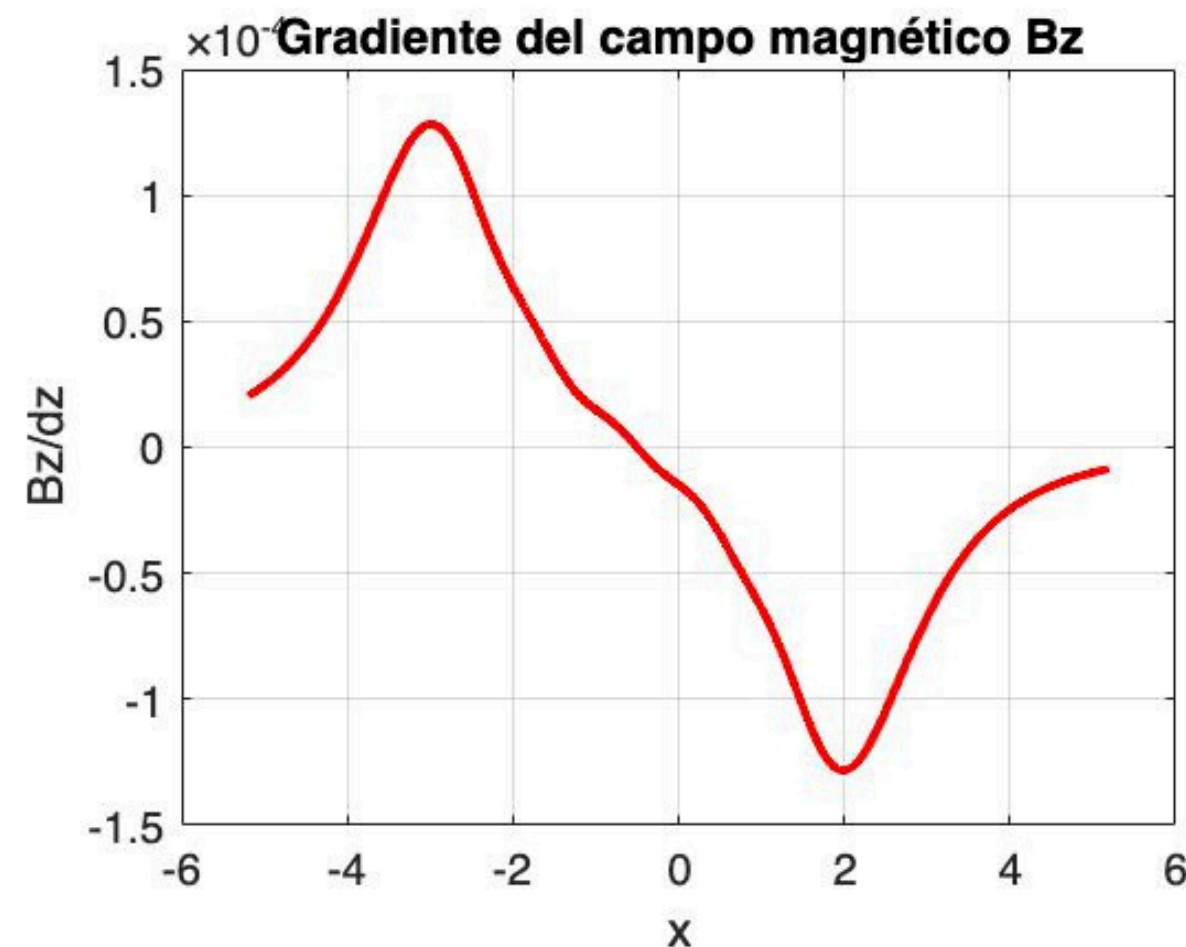
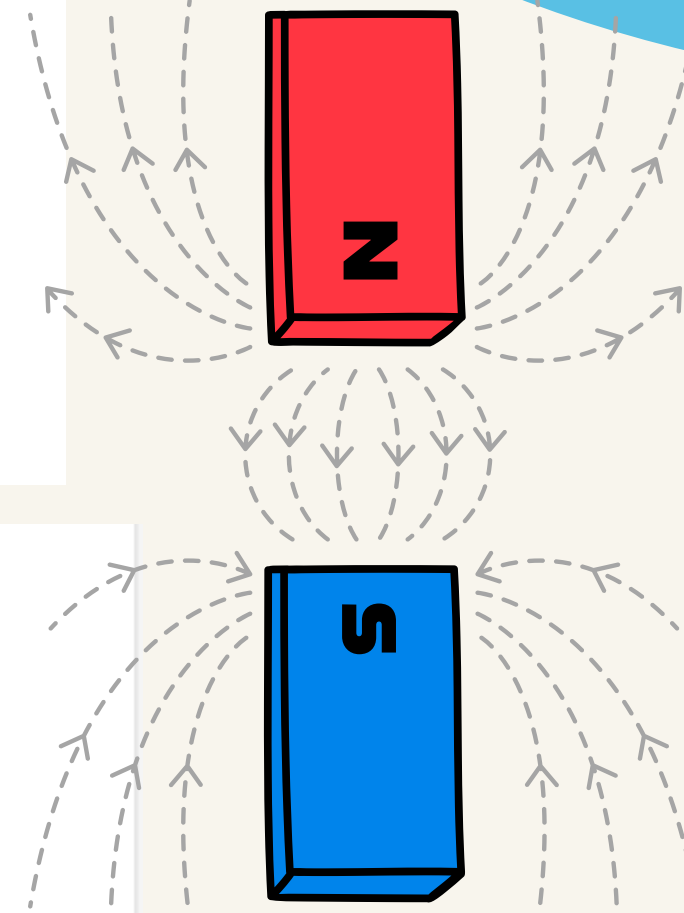
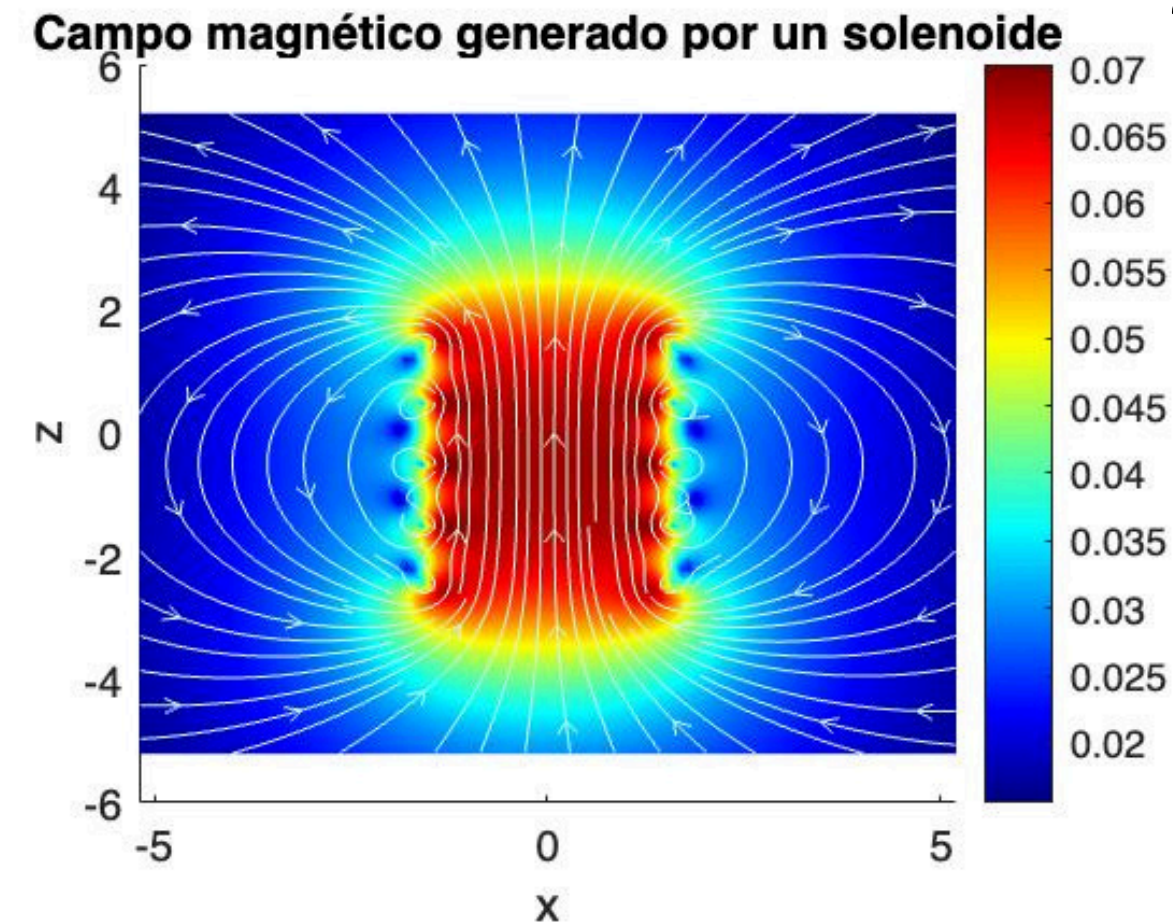
Los parámetros de salida son Px, Py y Pz (coordenadas 3D de los puntos por donde pasa la espira) y dx, dy y dz (dirección de la corriente en cada punto).

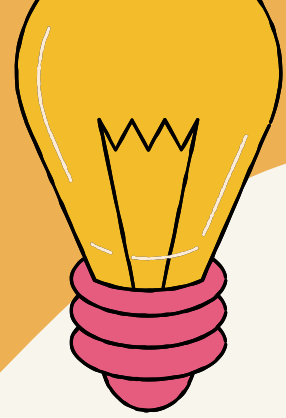
FUNCIÓN CAMPO_B

La función campoB calcula el campo magnético de un solenoide usando la ley de Biot-Savart y genera, según la opción seleccionada, un mapa del campo en el plano x-z o el perfil del gradiente axial dB_z/dz . Estas gráficas permiten analizar cómo varía el campo en el espacio.

PARÁMETROS

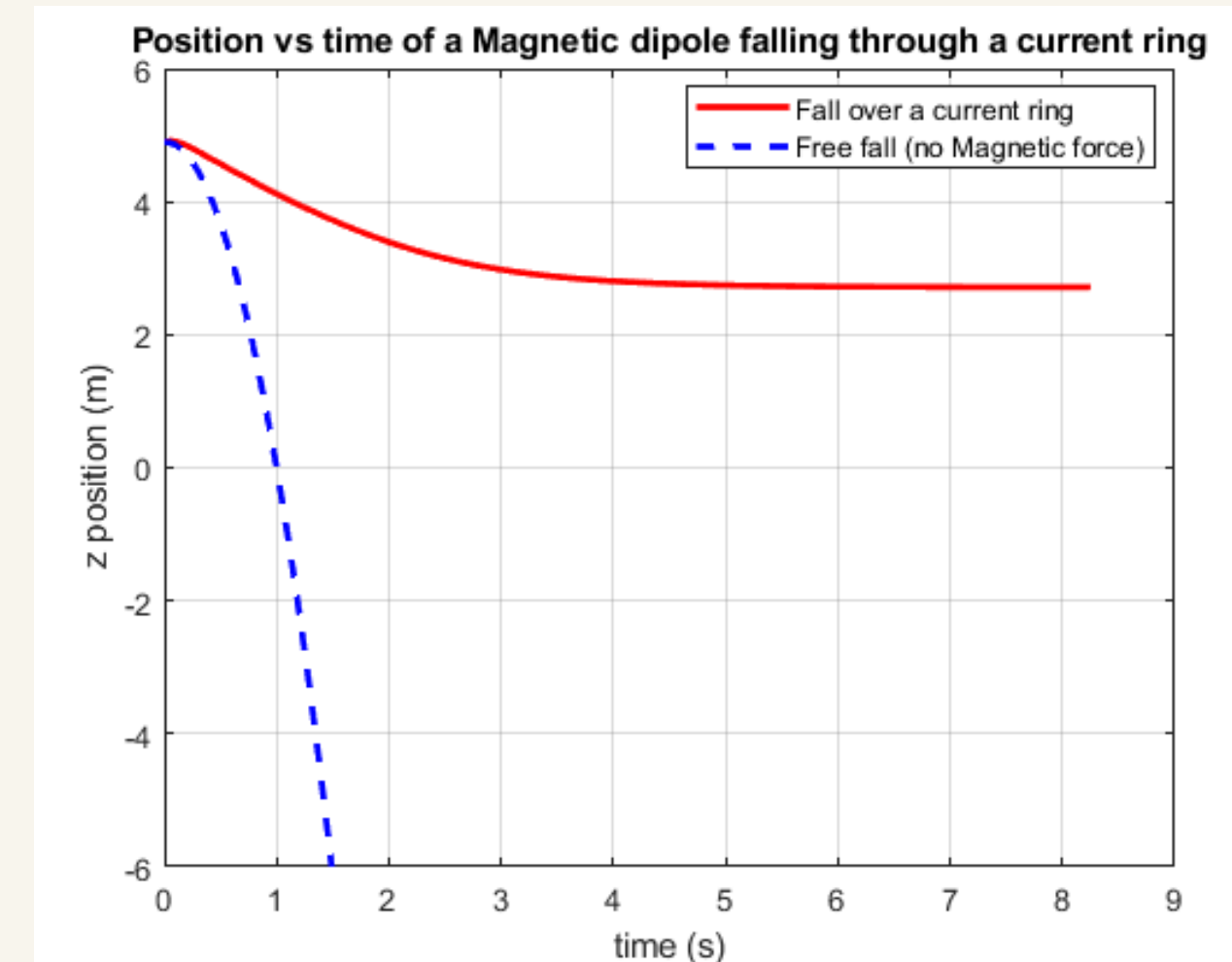
- ds: paso para el vector de posiciones z, controla resolución espacial.
- km: constante proporcional, relacionada con $\mu_0 * I / (4\pi)$.
- Px, Py, Pz: vectores con las coordenadas de los elementos de corriente.
- dx, dy: componentes del vector de corriente para cada elemento.
- ne: número de elementos por espira.
- N: número de espiras.
- rw: parámetro regularizador para evitar división por cero en r^\wedge .
- plot_option: si es true, genera una gráfica tipo mapa de flujo (campo completo). Si es false, genera solo el perfil del gradiente del campo en el eje central.





TRAYECTORIA

La trayectoria muestra en que dirección se va a trasladar el imán a través de un solenoide en comparación a la trayectoria que seguiría si no fuera afectado por un campo magnético. En un ciclo while se va graficando punto por punto las dos cinemáticas mostrando su posición en el eje z y después se grafica en el mismo plot. Azul siendo la cinemática sin campo magnético y rojo la cinemática a través del solenoide.



FUNCIÓN ACCELERACIÓN

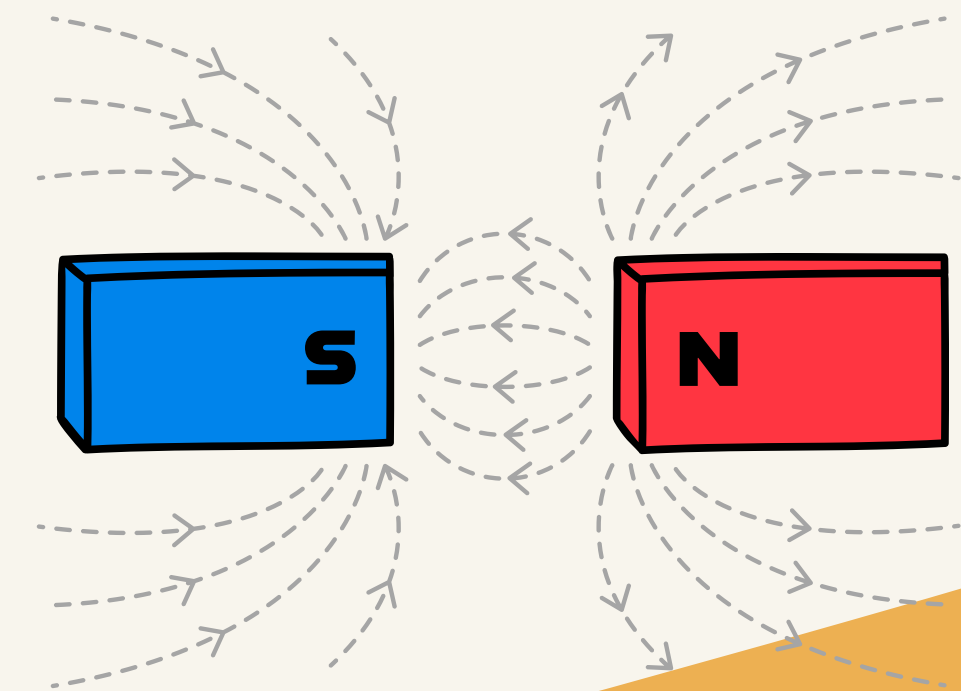
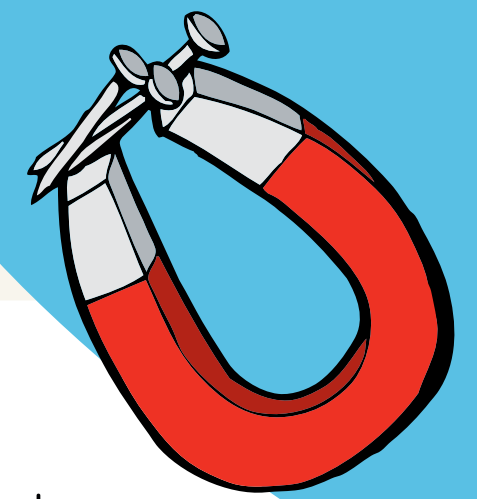
La función para la aceleración usa la fuerza magnética calculada en el campo magnético en el eje z para calcular la aceleración del objeto mientras pasa a través del campo magnético en el solenoide.

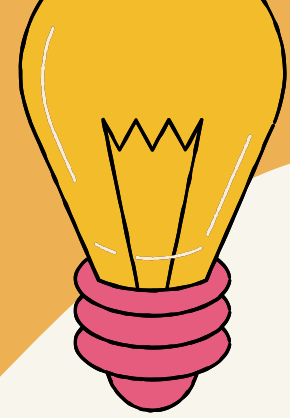
```
function a = a_total(z, v, Bz, z_axis, mag, gamma, m)
    delta = 0.005;
    Bz_forward = interp1(z_axis, Bz, z + delta, 'linear', 'extrap');
    Bz_backward = interp1(z_axis, Bz, z - delta, 'linear', 'extrap');
    dBz_dz = (Bz_forward - Bz_backward) / (2 * delta);

    % Calcula fuerza magnética y total
    Fm = -mag * dBz_dz;
    Ff = -gamma * v; % Nueva fuerza de fricción
    F = Fm + Ff - m * 9.81;
    a = F / m;
end
```

PARÁMETROS

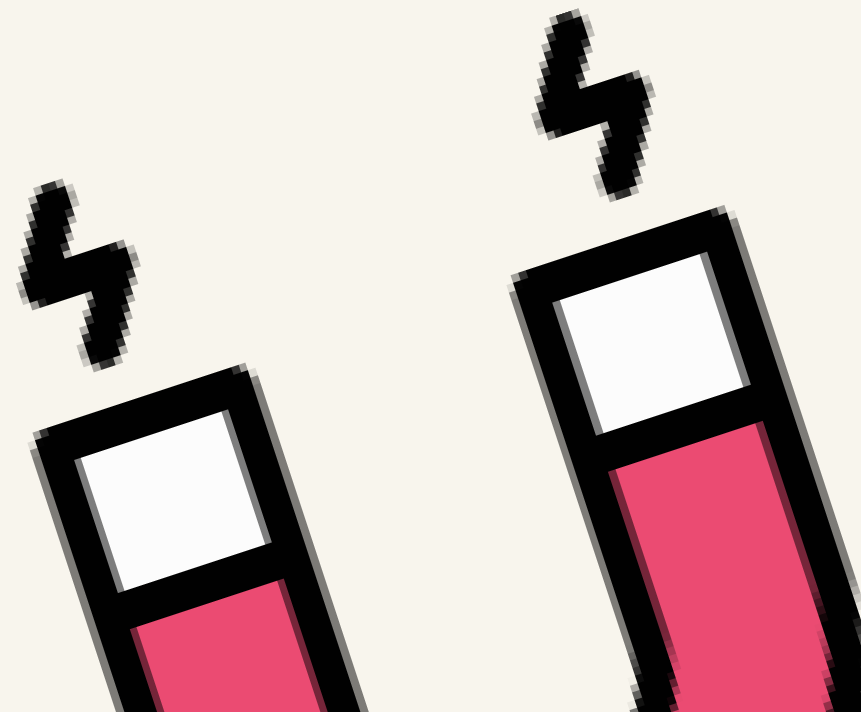
- z: El punto en z actual del objeto.
- v: La velocidad actual del objeto.
- Bz: El campo magnético en z.
- z_axis: Los intervalos donde se va a aplicar la función en z
- mag: La magnitud de la fuerza magnética.
- gamma: Coeficiente para calcular la fuerza de fricción.
- m = Masa del objeto





FUNCIÓN RUNGE KUTTA

La función de Runge Kutta hace el método de Runge Kutta de cuarto orden donde con el que vamos a calcular la posición en z y la velocidad siguiendo los pasos donde k1 es la pendiente inicial, k2 es la pendiente al punto medio usando la dirección inicial, k3 es la pendiente al punto medio usando la corrección de k2 y la k4 es la pendiente al final del paso.



$$\begin{aligned} k_1 &= f(x_n, y_n) && \text{pendiente inicial} \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) && \text{pendiente al punto medio usando la dirección inicial} \\ k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) && \text{pendiente al punto medio usando la corrección de } k_2 \\ k_4 &= f(x_n + h, y_n + h \cdot k_3) && \text{pendiente al final del paso} \end{aligned}$$
$$\rightarrow y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

```
function [z_next, v_next] = metodoRungeKutta(z, v, dt, a_func)
    k1z = v;
    k1v = a_func(z, v);

    k2z = v + 0.5*dt*k1v;
    k2v = a_func(z + 0.5*dt*k1z, v + 0.5*dt*k1v);

    k3z = v + 0.5*dt*k2v;
    k3v = a_func(z + 0.5*dt*k2z, v + 0.5*dt*k2v);

    k4z = v + dt*k3v;
    k4v = a_func(z + dt*k3z, v + dt*k3v);

    z_next = z + (dt / 6) * (k1z + 2*k2z + 2*k3z + k4z);
    v_next = v + (dt / 6) * (k1v + 2*k2v + 2*k3v + k4v);
end
```




¡GRACIAS!