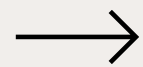
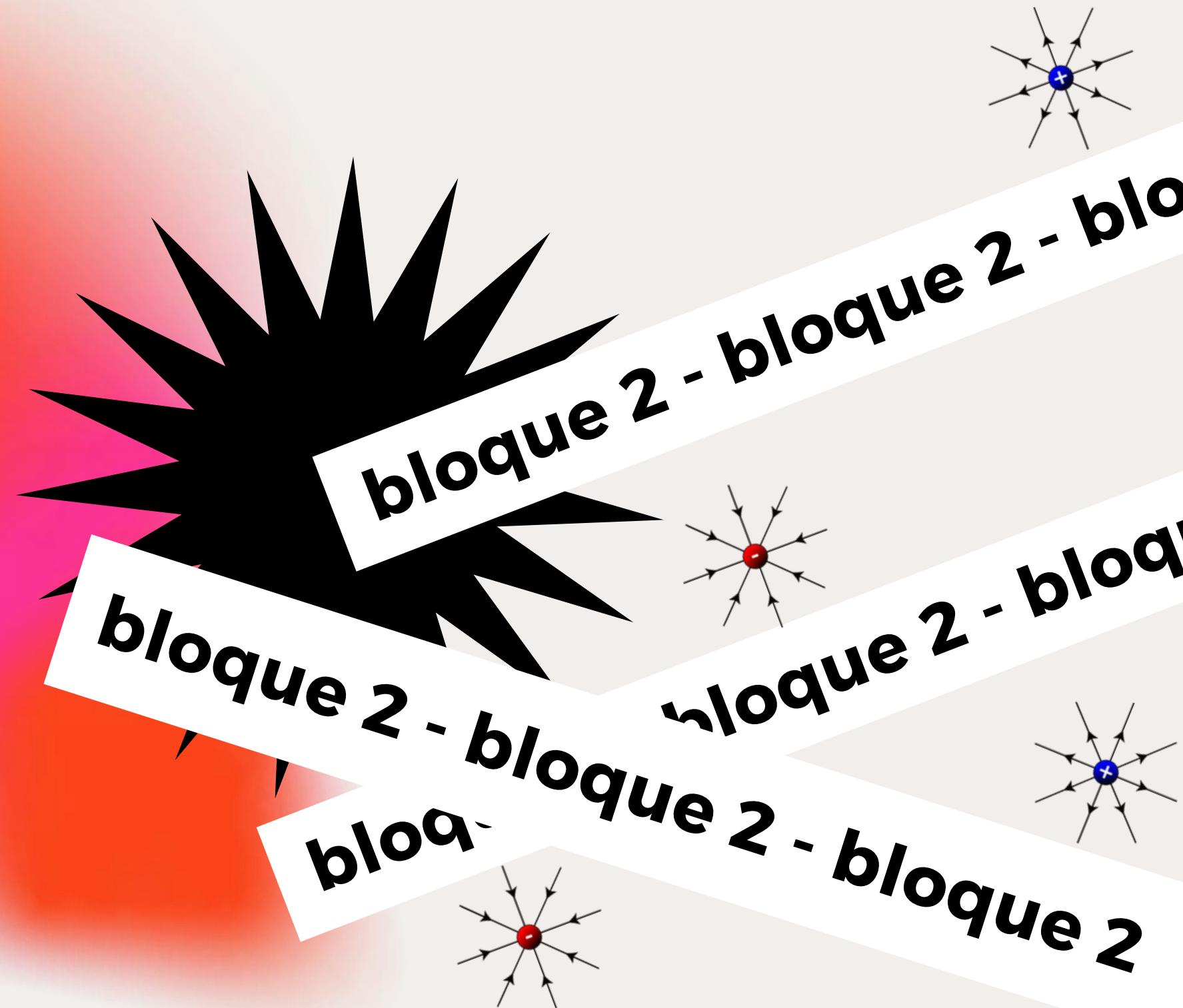


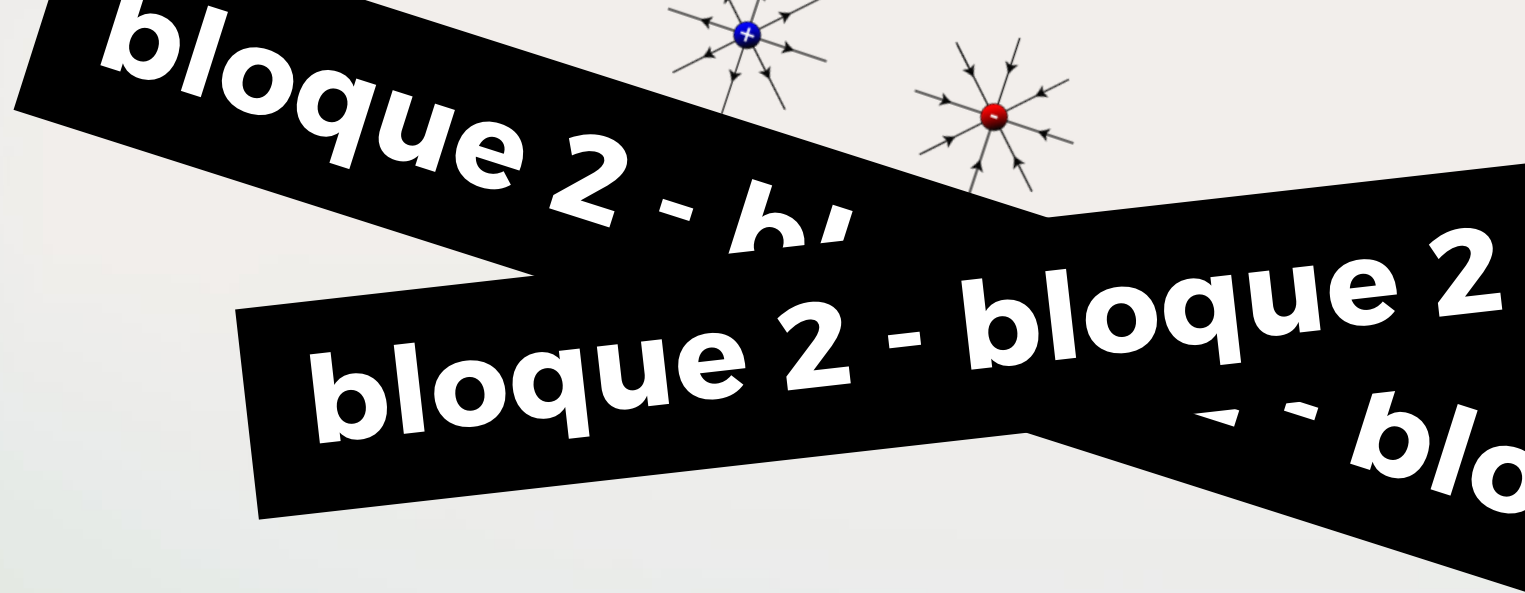


Campos eléctricos asimétricos



ENTREGA FINAL





Miembros del equipo 1

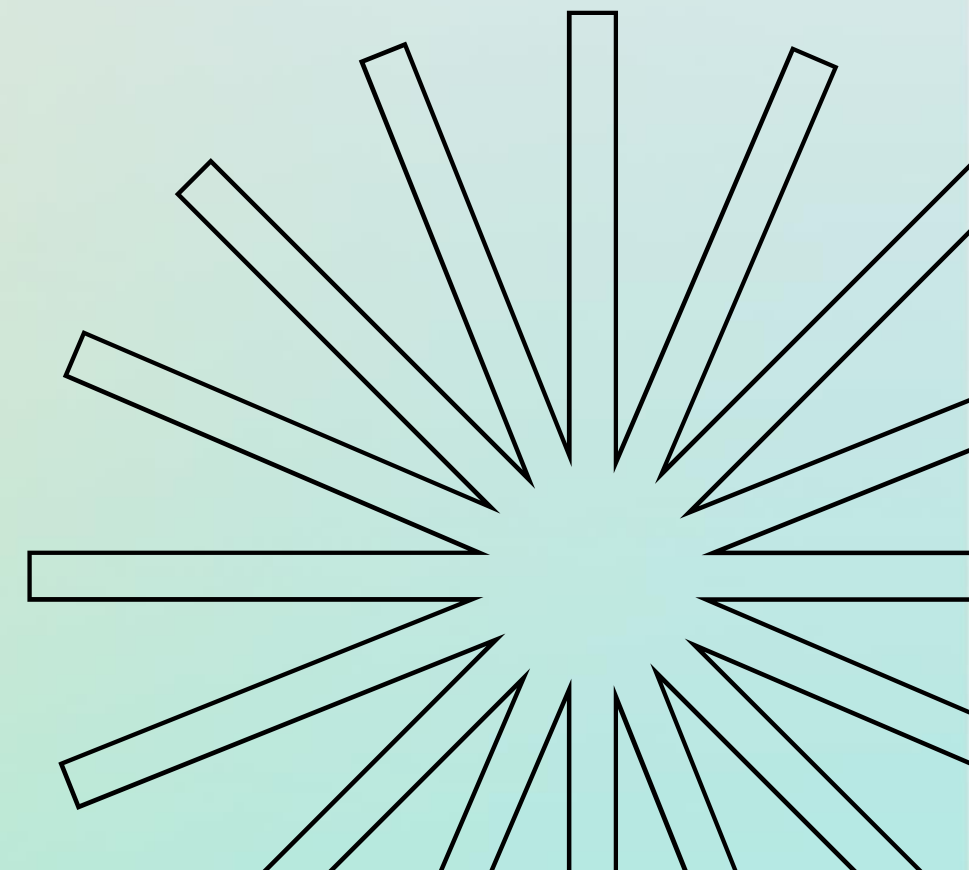
Alexa Serrano Negrete | A01654063

Luis Ernesto Ladrón de Guevara González | A01654419

Isabel Cristina Valdés Luevanos | A01025802

Victor Miguel Ramos Catlin | A01660323

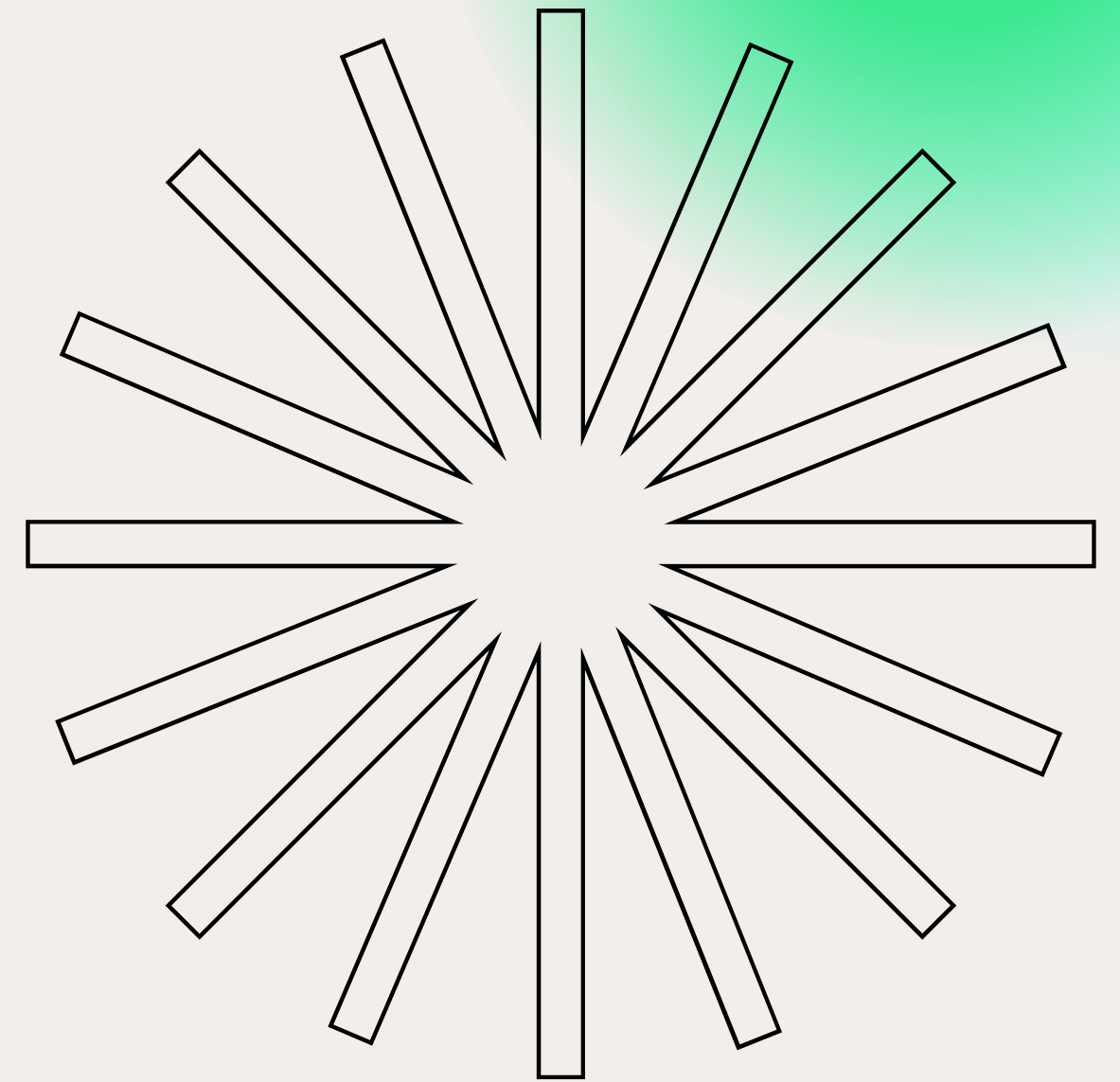
Octavio Augusto Alemán Esparza | A01660702



Índice de contenidos

LA AGENDA DE HOY

- * Definiendo metas, hipótesis y objetivos
- * ¿Qué pasa con la malaria?
- * La *dielectroforesis* y su relación a la malaria
- * ¿Qué pasó en el primer entregable?
- * Una breve explicación del código
- * Analizando los resultados
- * Conclusiones finales



Metas, Hipótesis y Objetivos

EXPECTATIVAS Y RESULTADOS

Objetivos

Crear una simulación computacional de un campo eléctrico como los que se usan en Dielectroforesis para diagnosticar malaria.

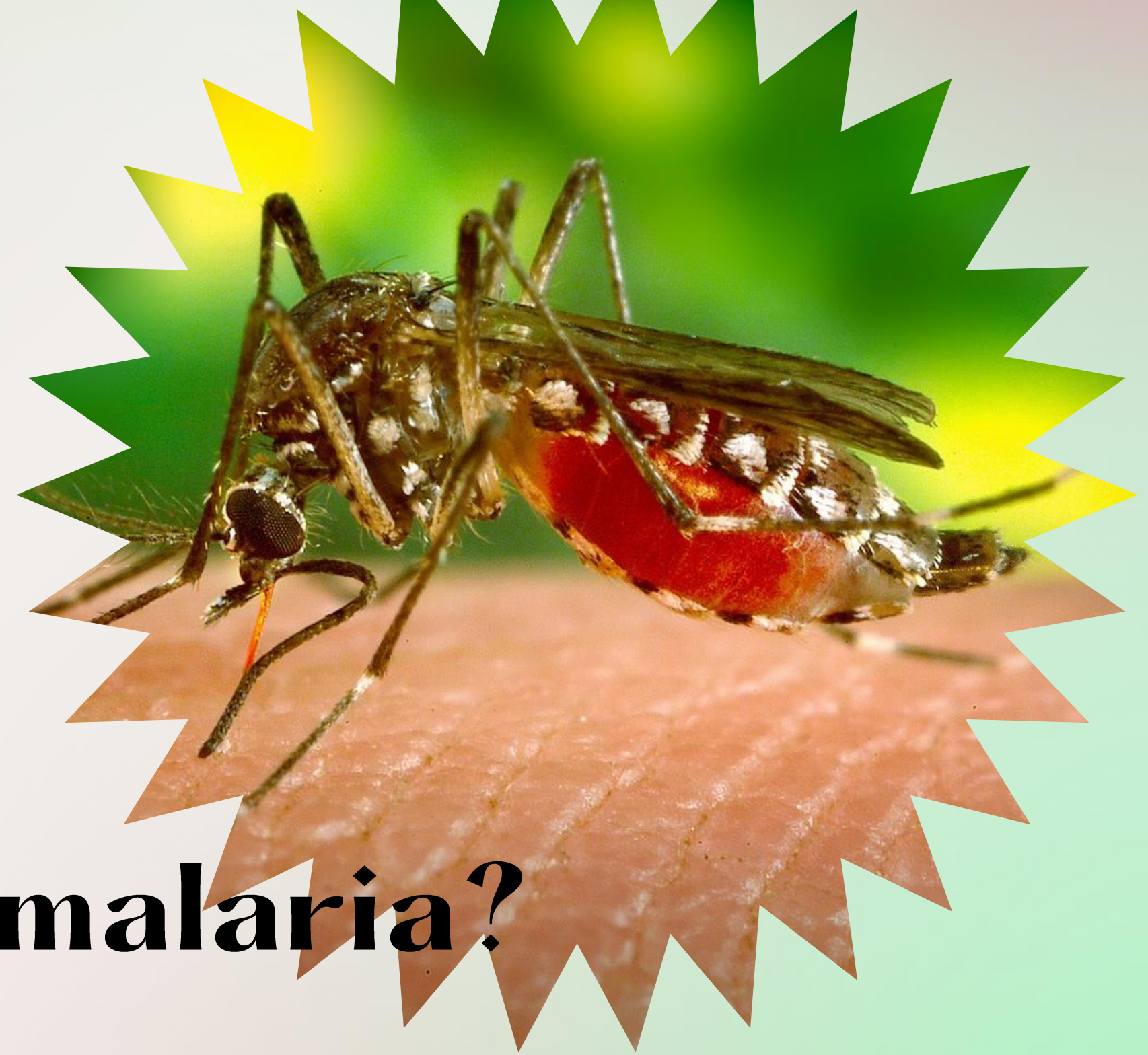
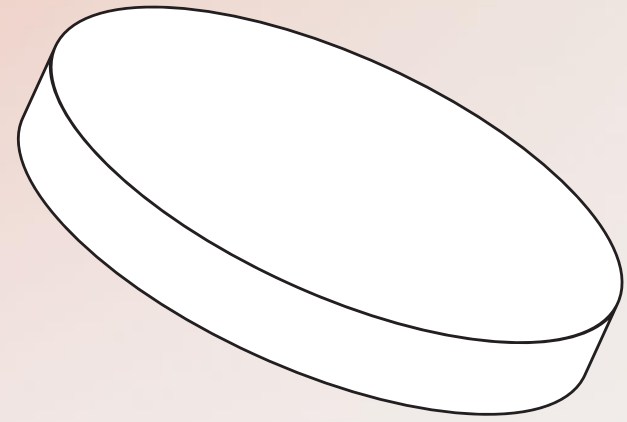
- Representar el campo vectorial generado por dos barras asimétricas en cada punto del plano cartesiano dentro de un rango predeterminado.



Hipótesis

Si la simulación para el cálculo del campo entre las barras es correcta, entonces podrá servir como referencia para un proceso de dielectroforesis.





¿Qué pasa con la malaria?

La malaria es una enfermedad parasitaria que involucra fiebres altas, escalofríos, síntomas similares a los de la gripe y anemia. Es causada por un parásito, y se transmite a los humanos a través de la picadura de mosquitos anofeles infectados. Después de la infección, los parásitos (llamados esporozoítos) viajan a través del torrente sanguíneo hasta el hígado. Allí maduran y producen otra forma de parásitos, llamada merozoítos. Los parásitos ingresan en el torrente sanguíneo e infectan a los glóbulos rojos.

Comprendiendo la dielectroforesis



Detección de la Malaria

La malaria debe reconocerse rápidamente para tratar al paciente a tiempo y prevenir una mayor propagación de la infección en la comunidad a través de los mosquitos locales.

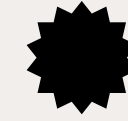
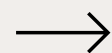
Kasetsirikul, Buranapong, Srituravanich, (n.d.)



Concepto Simplificado

La dielectroforesis, o DEP, es un movimiento de las partículas causado por los efectos de polarización en un campo eléctrico no uniforme.

(química.es, 2021)

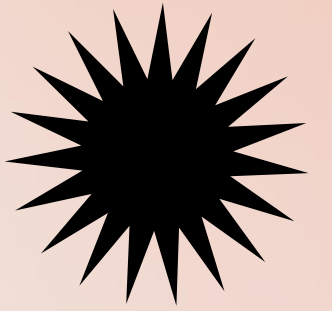


Definición explícita

De esta manera, la DEP se define también como un mecanismo de transporte electrocinético, que ocurre en la presencia de un campo eléctrico no homogéneo.

(Ozuna, Lapizco, Rito, Collado & Martínez, 2007).

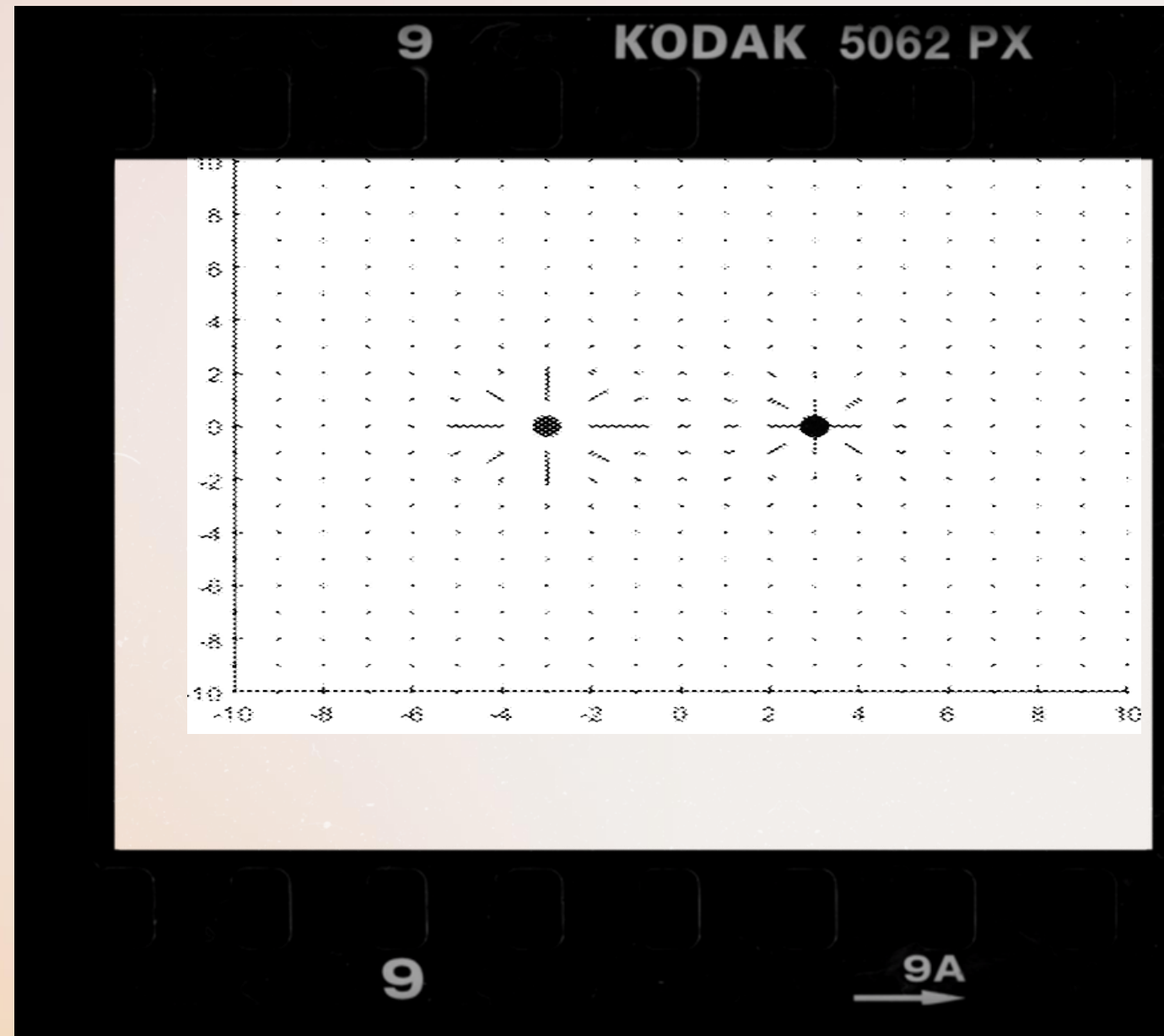


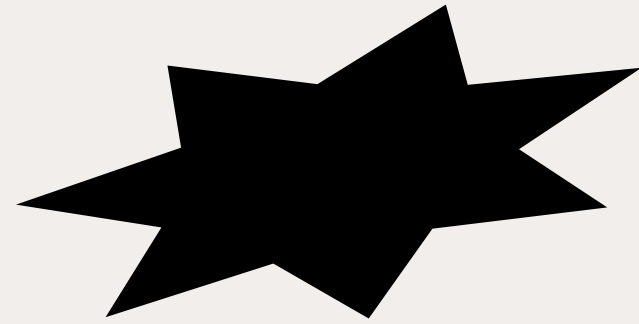


Etapa 1.

UNA BREVE INTRODUCCIÓN.

La primera entrega constó de un código de 157 líneas donde el usuario tenía la libertad de ingresar sus propios datos para todas las variables. El caso de prueba para este código constó de determinar un rango de -10 a 10, y el código creó un espacio bidimensional con una longitud de 21 tanto en el eje x como en el y.





Breve vista al código 1



Función *genDipoloPar*

Esta función genera un dipolo tomando en cuenta la posición simétrica a de las partículas con respecto al origen. Previamente, los valores fueron asignados de forma predeterminada dentro de la sección de arguments.

```
function genDipoloPar(k, a, scale, delta, minval, maxval,  
qpos, qneg)
```

```
arguments
```

```
    k (1,1) double = 1
```

```
    a (1,1) double = 2
```

```
    scale (1,1) double = 1
```

```
    delta (1,1) double = 1
```

```
    minval (1,1) double = -10
```

```
    maxval (1,1) double = 10
```

```
    qpos (1,1) double = 1
```

```
    qneg (1,1) double = -1
```

```
end
```

```
% DIBUJO DEL DIPOLO
```

```
if k == 0
```

```
    k = 8.99e9;
```

```
end
```



```
qposplt = scatter([-a], [0]);  
qposplt.SizeData = 100;  
qposplt.MarkerFaceColor = 'r';  
qposplt.MarkerEdgeColor = 'r';
```

```
hold on;  
malla_x = minval:delta:maxval;  
malla_y = minval:delta:maxval;
```

```
% MESHGRID  
[x, y] = meshgrid(malla_x, malla_y);
```

```
% ECUACIONES X  
E1x = ((k*(qpos).*(a+x))./((a+x).^2 + y.^2).^(3/2));  
class(E1x)  
E2x = ((k*(qneg).*(x-a))./((x-a).^2 + y.^2).^(3/2));
```

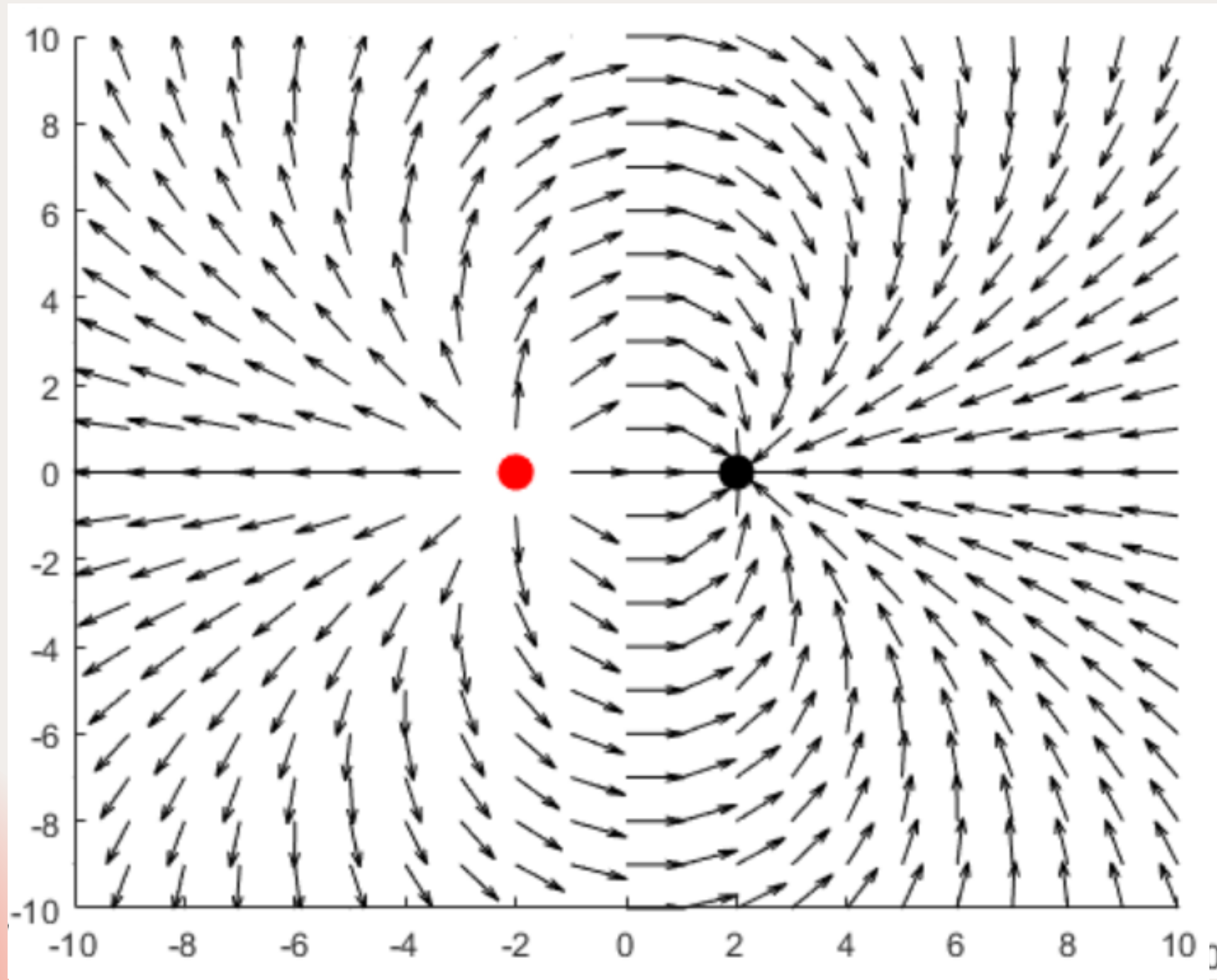
```
% ECUACIONES Y  
E1y = ((k*(qpos).*y)./((a+x).^2 + y.^2).^(3/2));  
E2y = ((k*(qneg).*y)./((x-a).^2 + y.^2).^(3/2));
```

```
qnegplt = scatter([a], [0]);  
qnegplt.SizeData = 100;  
qnegplt.MarkerFaceColor = 'k';  
qnegplt.MarkerEdgeColor = 'k';  
axis([minval maxval minval maxval]);
```

```
Ex = (E1x + E2x);  
Ey = (E1y + E2y);  
Module = sqrt(Ex.^2 + Ey.^2);  
quiver(x, y, Ex./Module, Ey./Module, scale,  
'Color','k', 'linewidth', 0.5);  
hold off
```

```
end
```

Analizando los resultados

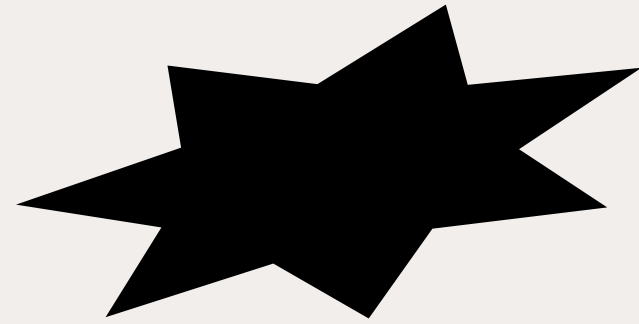


Resultados generales

Esta gráfica representa las partículas respecto al origen. Se aprecia de esta manera porque el dipolo de los vectores está normalizado sobre el modulo. Esta parte representa cómo se ve con respecto a dos puntos únicamente, no dos barras. Las ecuaciones de x y y se hicieron sin el ángulo, modificando así el Código que se hizo durante la primera entrega.



genDipoloPar()



Función *genDipoloBar*

Esta primera función sirve para generar el dipolo considerando la distancia entre las barras a generar. En esta función, hay ciertos valores previamente asignados en los argumentos.

```
function genDipoloBar(k, a, lenBarPos, lenBarNeg, scale,  
delta, minval, maxval, qpos, qneg)
```

```
% ARGUMENTOS
```

```
arguments
```

```
    k (1,1) double = 1
```

```
    a (1,1) double = 2
```

```
    lenBarPos (1,1) double = 1
```

```
    lenBarNeg (1,1) double = 1
```

```
    scale (1,1) double = 1
```

```
    delta (1,1) double = 1
```

```
    minval (1,1) double = -10
```

```
    maxval (1,1) double = 10
```

```
    qpos (1,1) double = 1
```

```
    qneg (1,1) double = -1
```

```
end
```

Breve vista al código 2




```

% DIBUJO DEL DIPOLO
if mod(lenBarPos, 2) == 0 % Impar
    posYChar = -(lenBarPos-
1)/2:delta:(lenBarPos-1)/2;
else % Par
    posYChar = -(lenBarPos-
1)/2:delta:(lenBarPos-1)/2;
end

if mod(lenBarNeg, 2) == 0 % Impar
    negYChar = -(lenBarNeg-
1)/2:delta:(lenBarNeg-1)/2;
else % Par
    negYChar = -(lenBarNeg-
1)/2:delta:(lenBarNeg-1)/2;
end

posXChar = ones(1, length(posYChar)) .* (-a);
negXChar = ones(1, length(negYChar)) .* (a);

qposplt = scatter(posXChar, posYChar);
qposplt.SizeData = 100;
qposplt.MarkerFaceColor = 'r';
qposplt.MarkerEdgeColor = 'r';
hold on:

```

```

qnegplt = scatter(negXChar, negYChar);
qnegplt.SizeData = 100;
qnegplt.MarkerFaceColor = 'k';
qnegplt.MarkerEdgeColor = 'k';
axis([minval maxval minval maxval]);

if k == 0
    k = 8.99e9;
end

malla_x = minval:delta:maxval;
malla_y = minval:delta:maxval;

% MESHGRID
[x, y] = meshgrid(malla_x, malla_y);
E1x = 0; E1y = 0; E2x = 0; E2y = 0;

for i = 1:lenBarPos
    E1x = E1x + ((k*(qpos).*(a+x))./((a+x).^2 + (y -
posYChar(i)).^2).^3/2));
    E1y = E1y + ((k*(qpos).*(y -
posYChar(i)))./((a+x).^2 + (y -
posYChar(i)).^2).^3/2));
end

```

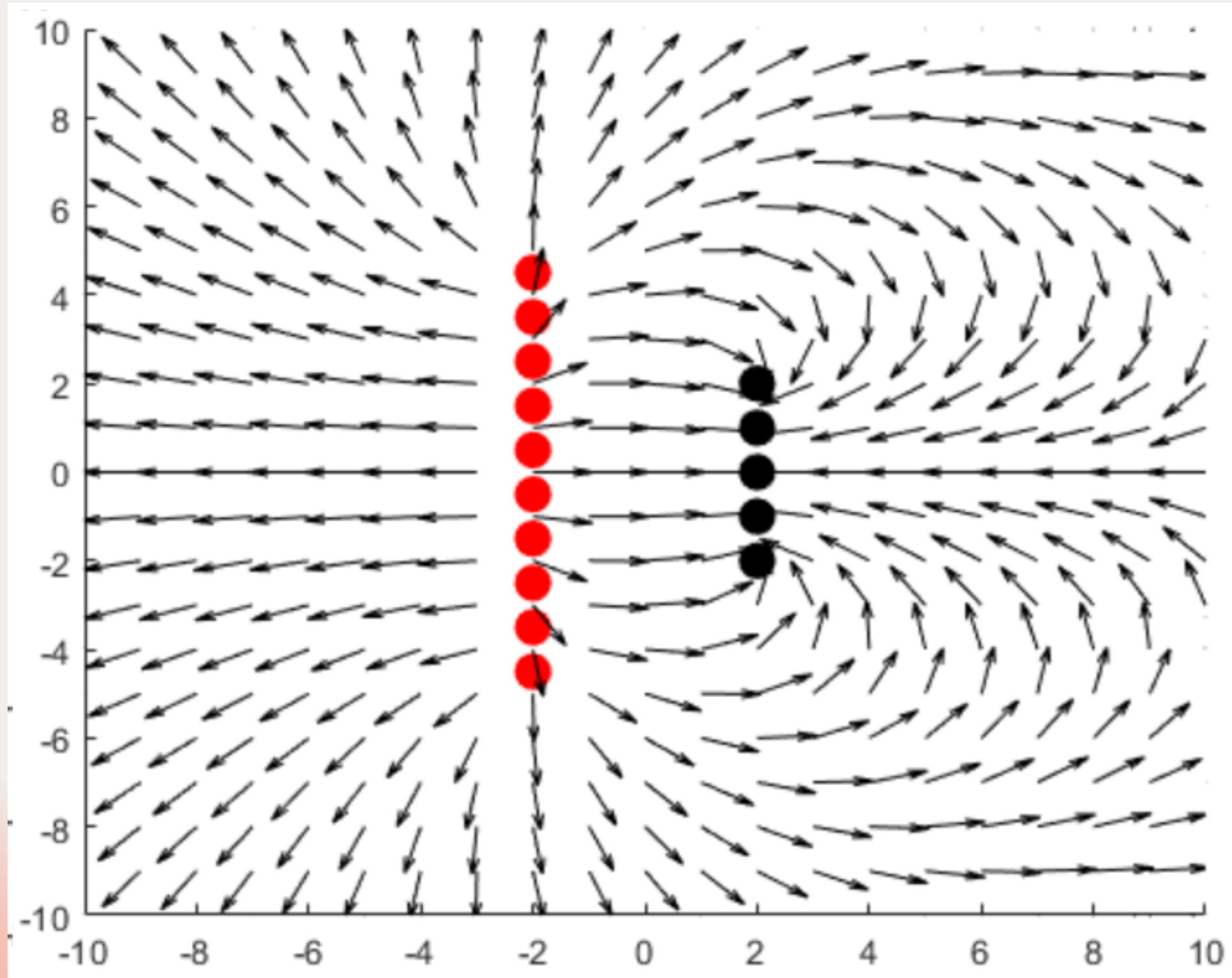
```

for i = 1:lenBarNeg
    E2x = E2x + ((k*(qneg).*(x-
a))./((x-a).^2 + (y -
negYChar(i)).^2).^3/2));
    E2y = E2y + ((k*(qneg).*(y -
negYChar(i)))./((x-a).^2 + (y -
negYChar(i)).^2).^3/2));
end

Ex = (E1x + E2x);
Ey = (E1y + E2y);
Module = sqrt(Ex.^2 + Ey.^2);
quiver(x, y, Ex./Module,
Ey./Module, scale, 'Color','k',
'linewidth', 0.5);
hold off
end

```

Analizando los resultados



Resultados generales

Esta función generó el dipole basado en la barra. Adiferencia de la segunda función, este código parte de que se conoce la posición de x , pero no de y . El Código inicia definiendo si la longitud es par o impar y a partir de ahí lo grafica acorde al eje. A partir de aquí, ambos códigos son muy similares, a excepción del loop que está aquí presente, pues esta itera cada una de las cargas y las agrega a la suma total del componente.



genDipoloBar()

¡Gracias!

Los objetivos del Proyecto se cumplieron y la hipótesis fue correcta. Se generaron dos funciones que generaban dos campos eléctricos. A parte de esto, el trabajo en equipo, así como la comunicación efectiva fueron competencias clave que promovieron el Desarrollo óptimo del Proyecto.