A. Crear categorías (20) y tags (150)

```
// 20 categorías tipo Pinterest
WITH [
  'Art', 'Design', 'Technology', 'Food', 'Travel', 'Fitness', 'Fashion', 'DIY',
  'Photography', 'Education', 'Gaming', 'Music', 'Movies', 'Quotes', 'Home Decor',
  'Nature', 'Business', 'Sports', 'Memes'
] AS categories
UNWIND range(0, size(categories)-1) AS i
CREATE (:Category {
  id_category: 'CAT-' + toString(i+1),
  name: categories[i]
});

// 150 tags variados (se mezclan palabras reales + sufijo para que no choquen)
WITH [
 'travel','food','design','tech','art','fitness','music','gaming','coding','nature',
 'pets','fashion','quotes','diy','coffee','books','photography','cars','sports','movies',
 'series','anime','cyberpunk','rpg','open world','scifi','romance','streetwear',
 'architecture','data science','ai','ml','python','javascript','cooking','mexico',
 'japan','europe','usa','beach','mountains','city','startup','productivity','study',
 'selfcare','motivation','memes','cats','dogs','luxury','minimalism','colorful',
 'black&white','abstract','landscape','portrait','interior','exterior','3d','flat'
] AS baseTags
UNWIND range(1,150) AS i
WITH i, baseTags[toInteger(rand()*size(baseTags))] AS base
CREATE (:Tag {name: base + '_' + toString(i)});
```

B. Crear usuarios, boards, pins y comentarios

```
// 300 usuarios
UNWIND range(1,300) AS i
CREATE (:User {
  id_user: 'USER-' + toString(i),
  name: 'Usuario ' + toString(i),
  profile_picture: 'https://api.dicebear.com/7.x/avataaars/svg?seed=' + toString(i)
});

// 200 boards
UNWIND range(1,200) AS i
CREATE (:Board {
  id_board: 'BOARD-' + toString(i),
  title: 'Board ' + toString(i),
```

```
  description: 'Colección de pins ' + toString(i),
  created_at: datetime() - duration({days: toInteger(rand()*365)})
});

// 600 pins
UNWIND range(1,600) AS i
CREATE (:Pin {
  id_pin: 'PIN-' + toString(i),
  title: 'Pin ' + toString(i),
  description: 'Descripción del pin ' + toString(i),
  url_image: 'https://example.com/pin' + toString(i) + '.jpg',
  created_at: datetime() - duration({days: toInteger(rand()*365)})
});

// 300 comentarios
UNWIND range(1,300) AS i
CREATE (:Comment {
  id_comment: 'CMT-' + toString(i),
  text: 'Comentario ' + toString(i),
  created_at: datetime() - duration({days: toInteger(rand()*365)})
});
```

C. Crear relaciones del modelo Pinterest

1. Usuarios creadores de boards y pins

```
// Cada board tiene 1 creador
MATCH (u:User)
WITH collect(u) AS users
MATCH (b:Board)
WITH b, users[toInteger(rand()*(size(users)-1))] AS creator
MERGE (creator)-[:CREATES]->(b);

// Cada pin tiene 1 creador
MATCH (u:User)
WITH collect(u) AS users
MATCH (p:Pin)
WITH p, users[toInteger(rand()*(size(users)-1))] AS creator
MERGE (creator)-[:CREATES]->(p);
```

2. Categoría de cada board

```
MATCH (c:Category)
WITH collect(c) AS cats
```

```
MATCH (b:Board)
WITH b, cats[toInteger(rand()*(size(cats)-1))] AS cat
MERGE (b)-[:IN_CATEGORY]->(cat);
```

3. Pins dentro de boards

```
MATCH (b:Board)
WITH collect(b) AS boards
MATCH (p:Pin)
WITH p, boards[toInteger(rand()*(size(boards)-1))] AS board
MERGE (board)-[:CONTAINS]->(p);
```

4. Tags de cada pin (2–4 tags por pin)

```
MATCH (t:Tag)
WITH collect(t) AS tags
MATCH (p:Pin)
WITH p, tags, toInteger(2 + rand()*3) AS numTags   // 2 a 4 tags
UNWIND range(1, numTags) AS i
WITH p, tags[toInteger(rand()*(size(tags)-1))] AS tag
MERGE (p)-[:HAS_TAG]->(tag);
```

5. Likes de usuarios a pins (5–15 likes por pin)

```
MATCH (u:User)
WITH collect(u) AS users
MATCH (p:Pin)
WITH p, users, toInteger(5 + rand()*10) AS numLikes   // 5 a 15 likes
UNWIND range(1, numLikes) AS i
WITH p, users[toInteger(rand()*(size(users)-1))] AS u
MERGE (u)-[:LIKES {
  date: datetime() - duration({days: toInteger(rand()*30)})
}]->(p);
```

6. Follows entre usuarios (5–15 follows por usuario)

```
MATCH (u:User)
WITH collect(u) AS users
UNWIND users AS u1
WITH u1, users, toInteger(5 + rand()*10) AS numFollows
UNWIND range(1, numFollows) AS i
WITH u1, users[toInteger(rand()*(size(users)-1))] AS u2
WHERE u1 <> u2
MERGE (u1)-[:FOLLOWS {
```

```
  since: datetime() - duration({days: toInteger(rand()*365)})
}]->(u2);
```

7. Comentarios (WROTE y ON)

```
MATCH (u:User)
WITH collect(u) AS users
MATCH (p:Pin)
WITH users, collect(p) AS pins
MATCH (c:Comment)
WITH c,
    users[toInteger(rand()*(size(users)-1))] AS u,
    pins[toInteger(rand()*(size(pins)-1))] AS p
MERGE (u)-[:WROTE]->(c)
MERGE (c)-[:ON]->(p);
```

D. Verificar que ya superaste el mínimo

```
// Número total de nodos
MATCH (n)
RETURN count(n) AS totalNodos;

// Número total de relaciones
MATCH ()-[r]->()
RETURN count(r) AS totalRelaciones;
```

E. Actualizar los pines que ya existen con URL de imagen real

```
// Actualizar TODOS los pins existentes para que tengan imagen real (Picsum)
MATCH (p:Pin)
SET p.url_image = 'https://picsum.photos/seed/' + p.id_pin + '/400/300';
```