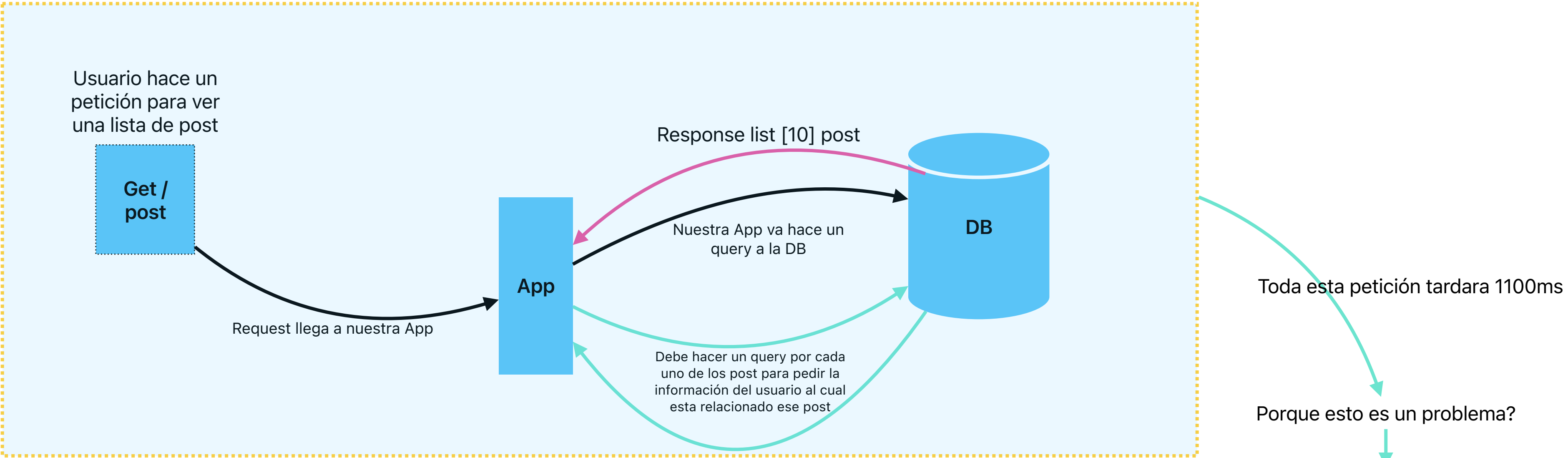


N+1 query problem



```
Post {
  title
  content
  user {
    name
    email
  }
}
```

Vamos a ver 1 lista de posts -> 10 = N

Después vamos a ver N Query pidiendo el detalle del User

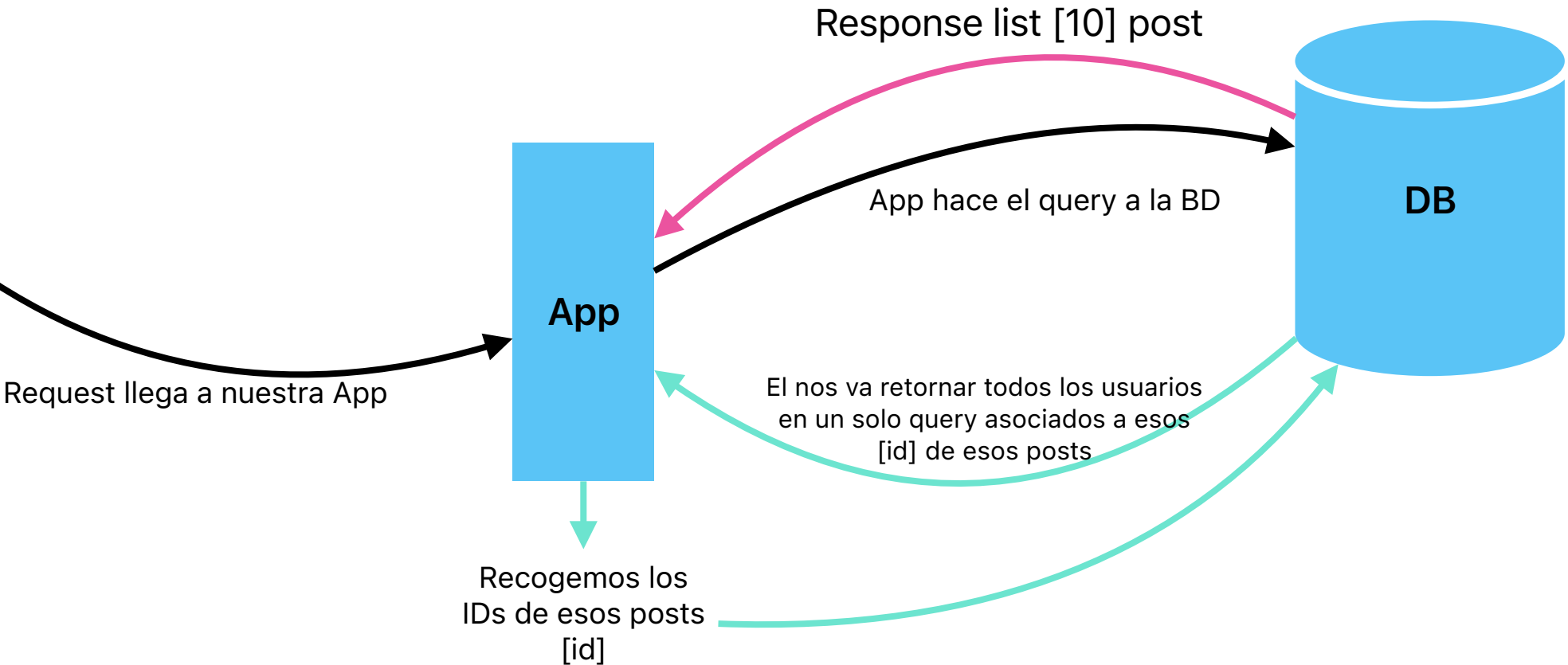
N+1 query problem

Supongamos que cada request cuesta 100ms

Después vamos hacer N peticiones que en este caso serían 10 * 100 darían 1000ms

Ahora para solucionar esto podemos pasar de N+1 queries -> a solo 2 query a la BD

```
Post {
  title
  content
  user {
    name
    email
  }
}
```



N+1 -> 2

Pasamos de 1100ms a solamente 200ms

```
Terminal: Local (2) x Local x + u
Post Create (0.1ms) INSERT INTO "posts" ("title", "content", "published", "user_id", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?) [{"title", "Deserunt at laboriosam commodi."}, {"content", "Numquam et edit. Eum accusamu
s nobis. Enim magni quia."}, {"published", 1}, {"user_id", 9}, {"created_at", "2023-10-14 19:42:08.485112"}, {"updated_at", "2023-10-14 19:42:08.485112"}]
TRANSACTION (0.0ms) RELEASE SAVEPOINT active_record_2
TRANSACTION (0.0ms) SAVEPOINT active_record_2
User Create (0.1ms) INSERT INTO "users" ("email", "name", "auth_token", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?) [{"email", "velda@wolff.example"}, {"name", "Thaddeus Turner"}, {"auth_token", "xxxxx"}, {"created_at", "
2023-10-14 19:42:08.487621"}, {"updated_at", "2023-10-14 19:42:08.487621"}]
TRANSACTION (0.0ms) RELEASE SAVEPOINT active_record_2
TRANSACTION (0.0ms) SAVEPOINT active_record_2
Post Create (0.1ms) INSERT INTO "posts" ("title", "content", "published", "user_id", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?) [{"title", "Ipsa porro architecto necessitatibus."}, {"content", "Sequi sed omnis. Vel ma
iores beatae. Vel qui quia."}, {"published", 1}, {"user_id", 10}, {"created_at", "2023-10-14 19:42:08.488716"}, {"updated_at", "2023-10-14 19:42:08.488716"}]
TRANSACTION (0.0ms) RELEASE SAVEPOINT active_record_2
Started GET "/posts" for 127.0.0.1 at 2023-10-14 14:42:08 -0500
Processing by PostsController#index as HTML
Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."published" = ? AND (title like '%X%') [{"published", 1}]
[active_model_serializers] User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 1}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 2}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 3}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 4}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 5}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 6}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 7}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 8}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.1ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 9}, [{"LIMIT", 1}]
[active_model_serializers] User Load (0.0ms) SELECT "users".* FROM "users" WHERE "users"."id" = ? LIMIT ? [{"id", 10}, [{"LIMIT", 1}]
[active_model_serializers] Rendered ActiveRecord::Serializer::CollectionSerializer with ActiveRecordSerializers::Adapter::Attributes (10.82ms)
Completed 200 OK in 34ms (Views: 18.8ms | ActiveRecord: 0.7ms | Allocations: 29081)
TRANSACTION (0.1ms) ROLLBACK TO SAVEPOINT active_record_1
TRANSACTION (0.5ms) rollback transaction
```

```
blogapi • app • controllers • posts_controller.rb • PostsController • index
1 class PostsController < ApplicationController
2   # manejamos las excepciones que pueden ocurrir donde vamos a recibir en formato json los errores de la pruebas
3   # Se debe tener en cuenta el orden de las excepciones.
4   rescue_from Exception do |e|
5     render json: { errors: e.message }, status: :internal_server_error # representa un 500 servidor falle
6   end
7
8   rescue_from ActiveRecord::RecordInvalid do |e|
9     render json: { errors: e.message }, status: :unprocessable_entity
10  end
11  # Codeium: Refactor Explain Docstring
12  # /posts
13  def index
14    # return posts published
15    @posts = Post.where(published: true)
16    if params[:search].nil? || params[:search].present?
17      @posts = PostsSearchService.search(@posts, params[:search])
18    end
19    render json: @posts.includes(:user), status: :ok You, Yesterday • Uncommitted changes
20  end
21 end

Terminal: Local x Local (2) x + u
Post Create (0.1ms) INSERT INTO "posts" ("title", "content", "published", "user_id", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?) [{"title", "Atque et quia voluptates."}, {"content", "Aut iste et. Iusto et at. Est disti
nctio et."}, {"published", 1}, {"user_id", 9}, {"created_at", "2023-10-14 19:51:52.573511"}, {"updated_at", "2023-10-14 19:51:52.573511"}]
TRANSACTION (0.0ms) RELEASE SAVEPOINT active_record_2
TRANSACTION (0.0ms) SAVEPOINT active_record_2
User Create (0.1ms) INSERT INTO "users" ("email", "name", "auth_token", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?) [{"email", "albert@schaefer.test"}, {"name", "Tama Roberts"}, {"auth_token", "xxxxx"}, {"created_at", "20
23-10-14 19:51:52.576199"}, {"updated_at", "2023-10-14 19:51:52.576199"}]
TRANSACTION (0.0ms) RELEASE SAVEPOINT active_record_2
TRANSACTION (0.0ms) SAVEPOINT active_record_2
Post Create (0.1ms) INSERT INTO "posts" ("title", "content", "published", "user_id", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?) [{"title", "Delectus ut queraet eos."}, {"content", "Molestiae consequuntur nihil. Itaque
provident magni. Non iusto ut."}, {"published", 1}, {"user_id", 10}, {"created_at", "2023-10-14 19:51:52.577384"}, {"updated_at", "2023-10-14 19:51:52.577384"}]
TRANSACTION (0.0ms) RELEASE SAVEPOINT active_record_2
Started GET "/posts" for 127.0.0.1 at 2023-10-14 14:51:52 -0500
Processing by PostsController#index as HTML
Post Load (0.2ms) SELECT "posts".* FROM "posts" WHERE "posts"."published" = ? AND (title like '%X%') [{"published", 1}]
User Load (0.2ms) SELECT "users".* FROM "users" WHERE "users"."id" IN (?, ?, ?, ?, ?, ?, ?, ?, ?) [{"id", 1}, {"id", 2}, {"id", 3}, {"id", 4}, {"id", 5}, {"id", 6}, {"id", 7}, {"id", 8}, {"id", 9}, {"id", 10}]
[active_model_serializers] Rendered ActiveRecord::Serializer::CollectionSerializer with ActiveRecordSerializers::Adapter::Attributes (2.05ms)
Completed 200 OK in 42ms (Views: 25.8ms | ActiveRecord: 0.4ms | Allocations: 30191)
TRANSACTION (0.1ms) ROLLBACK TO SAVEPOINT active_record_1
TRANSACTION (0.4ms) rollback transaction
```