

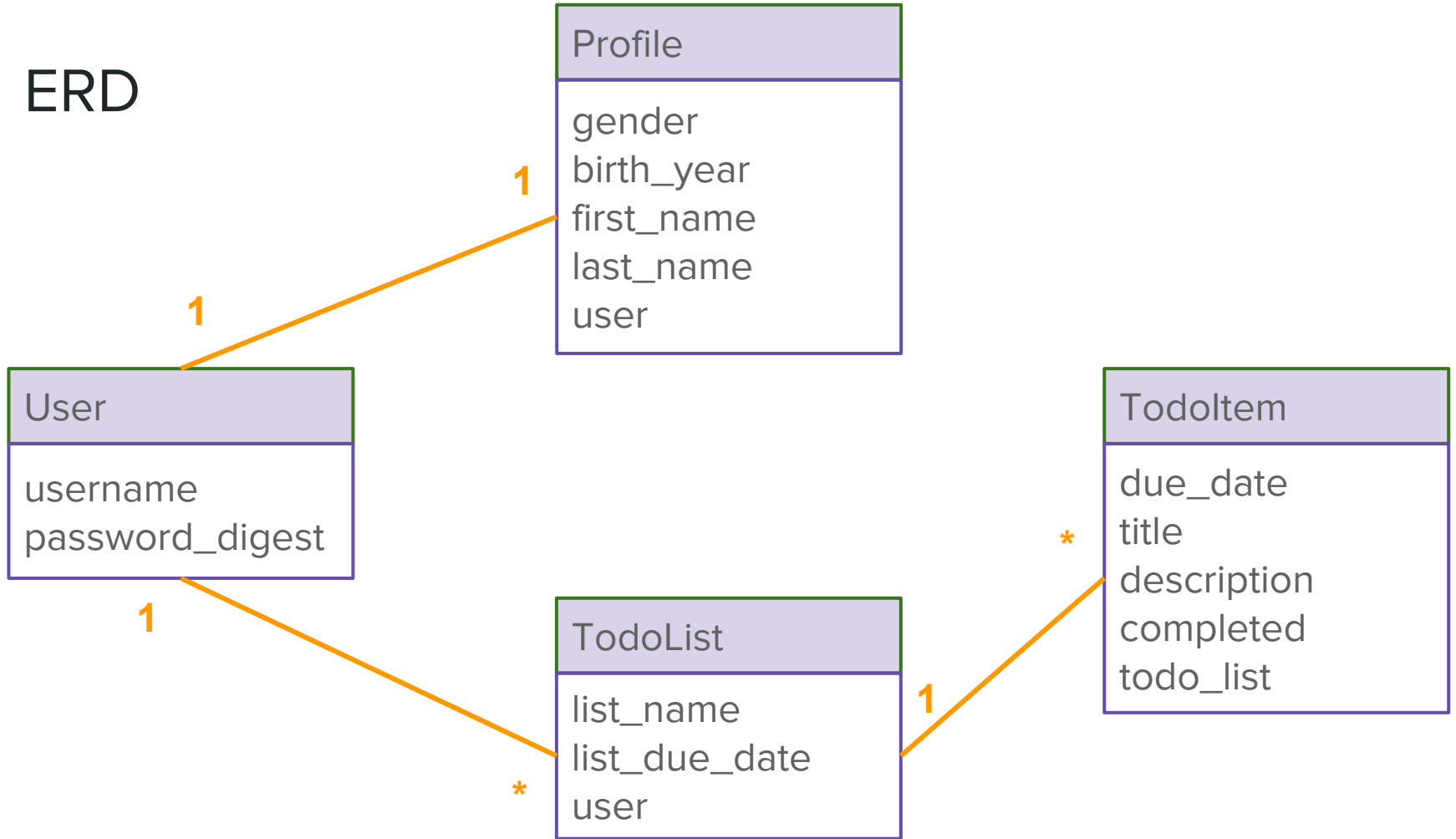
Práctica

Módulo 6

Objetivos

- Nested resources
- Authentication
- Authorization
- Pagination

ERD



Requerimientos funcionales

1. Empezar con la aplicación completada en el módulo 2.
 - a. Esta tarea ya tiene todos los modelos y relaciones que necesitamos (User, TodoList, TodoItem).
 - b. No vamos a utilizar la clase Profile
2. Implementar el acceso a TodoItem como un nested resource de TodoList.
3. Permitir el uso de la aplicación únicamente a los usuarios autenticados.
4. Limitar el acceso a recursos asociados al usuario logeado.
5. Acceder a colecciones ilimitadas de los recursos utilizando paginación.

Ejercicio 1

1. Inicio

- Hacer una copia de la aplicación práctica que está en modulo-6/sources/practica
- Verificar que esté todo en orden:
 - `rake db:migrate`
 - `rspec spec/start_spec.rb`

```
- should respond to #updated_at
rq1.5
  User has a 1:many relationship with TodoItem
  User has a 1:many relationship with TodoItem through TodoList
check database operations
rq1.6
  Cascading delete is represented in model dependencies
  User delete must cascade delete to TodoList and TodoItems
```

```
Finished in 0.37681 seconds (files took 1.81 seconds to load)
30 examples, 0 failures
```

Ejercicio 2 y 3

2. Agregar has_secure_password a **User**

```
$ rspec spec/security_spec.rb -e rq02
```

3. Poblar la base de datos

```
$ rake db:seed
```

```
$ rails c
```

```
$ User.first.todo_lists.count
```

```
=> (N>0)
```


Solucion 2

```
class User < ActiveRecord::Base  
  has_secure_password
```

Solución 3

```
rake db:seed
```

Ejercicio 4



4. Crear controladores para TodoList y TodoItem

```
$ rails g scaffold_controller TodoList list_name list_due_date:date
```

```
$ rails g scaffold_controller TodoItem title due_date:date  
description:text completed:boolean
```

Actualizar config/routes.rb para que:

- Acceder a: los recursos todo_list en la URI /todo_lists
- Acceder a: los recursos todo_item en la URI
/todo_lists/:todo_list_id/todo_items
- Hacer que el action todo_list#index sea la raíz de la aplicación

```
$ rspec spec/nested_resources_spec.rb -e rq04
```

Solución 4

```
Rails.application.routes.draw do
  resources :todo_lists do
    resources :todo_items
  end

  root to: "todo_lists#index"
```

Ejercicio 5

5. Mostrar todo_items como nested resource de todo_lists

1. Copiar la tabla de la página **todo_items#index** y pegarla en el template **todo_lists#show**.
2. Cambiar las referencias globales a **@todo_items** referencias limitadas a **@todo_list.todo_items**:
 - a. de: `<% @todo_items.each do |todo_item| %>`
 - b. a: `<% @todo_list.todo_items.each do |todo_item| %>`
3. Eliminar el link “Edit” para TodoItems.

5. Mostrar `todo_items` como nested resource de `todo_lists`

4. Cambiar los parámetros del **`link_to`** de eliminación para que tenga referencia a **`todo_list`** y **`todo_item`**, como un array de referencias **`[@todo_list, todo_item]`**

- a. `de: <%= link_to 'Show', todo_item %>`
- b. `a: <%= link_to 'Show', [@todo_list, todo_item] %>`
- c. `de: <%= link_to 'Destroy', todo_item, method: :delete, data: ...`
- d. `a: <%= link_to 'Destroy', [@todo_list, todo_item], method: :delete, data: ...`

```
$ rspec spec/nested_resources_spec.rb -e rq05b
```


5. Mostrar todo_items como nested resource de todo_lists

5. Crear un link “New Todo Item” utilizando el helper `link_to` y el route `new_todo_list_todo_item_path(@todo_list)` para producir el link.

```
$ rspec spec/nested_resources_spec.rb -e rq0d
```

Solución 5

```
<p id="notice"><%= notice %></p>
```

```
<p>
  <strong>List name:</strong>
  <%= @todo_list.list_name %>
</p>
```

```
<p>
  <strong>List due date:</strong>
  <%= @todo_list.list_due_date %>
</p>
```

```
<table>
  <thead>
    <tr>
      <th>Title</th>
      <th>Due date</th>
      <th>Description</th>
      <th>Completed</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <%= @todo_list.todo_items.each do |todo_item| %>
      <tr>
        <td><%= todo_item.title %></td>
        <td><%= todo_item.due_date %></td>
        <td><%= todo_item.description %></td>
        <td><%= todo_item.completed %></td>
        <td><%= link_to 'Show', [@todo_list, todo_item] %></td>
        <td><%= link_to 'Destroy', [@todo_list, todo_item], method: :delete, data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <%= end %>
  </tbody>
</table>
```

```
<%= link_to 'New Todo Item', new_todo_list_todo_item_path(@todo_list) %>
```

```
<%= link_to 'Edit', edit_todo_list_path(@todo_list) %> |
<%= link_to 'Back', todo_lists_path %>
```

5. Create new item

Observemos como está formado el route:

new_todo_list_todo_item GET /todo_lists/:todo_list_id/todo_items/new(.:format)
todo_items#new

Queremos que:

1. Invocar al action `todo_items#new` cuando creamos un nuevo `TodoItem`
2. Debemos proveer `todo_items#new` un parámetro `todo_list_id` que es obtenido pasando `@todo_list` en la invocación.
3. Note que `TodolItems` ahora se despliega en el action `todo_items#show`, navegando en un `todo_list` específico.



localhost:3000/todo_lists/2503

List name: List 2

List due date: 2016-11-22

Title	Due date	Description	Completed	
Task 1	2016-11-22	very important task	TEST false	Show Destroy
Task 2	2016-11-25	do something else	TEST true	Show Destroy
Task 3	2016-11-25	learn Action Pack	TEST true	Show Destroy
New Todo Item Edit Back				

Ejercicio 6

6. Modificar los controladores para responder a la creación de un TodoItem.

Modificar el controlador de TodoItem para funcionar como un nested resource implementando los siguientes pasos. Notar que las vistas con URIs TodoItem no van a funcionar hasta que todos los cambios hayan sido realizados y los links de los forms hayan sido cambiados para tener las referencias `todo_list->todo_item`.

Los test unitarios, sin embargo, podrán hacer llamadas al backend para determinar que todas las URIs han sido implementadas correctamente - antes de ir a la modificación de vistas.

a. Reemplazar todas las URIS que referencian a `todo_item`

En las vistas creadas por el scaffold, reemplazar todas las URIS que hagan referencia a `todo_item` sólo, ya que solamente existen las rutas a `todo_item`, pasando por un `todo_list`.

```
rake routes
```

```
/todo_list/:todo_list_id/todo_items
```

```
/todo_list/:todo_list_id/todo_items/:id
```


b. Eliminar el action `todoItem#index`

Tanto la vista como el método en el controlador.

c. Agregar el helper `set_todo_list` en `todo_items_controller.rb`

El método debe ser privado, y setear una variable de instancia `@todo_list` que obtenga el registro en la base de datos que matchee con el parámetro `params[:todo_list_id]`

```
$ rails c  
> list_id=TodoList.first.id  
> @todo_list=TodoList.find(list_id)
```

d. Actualizar el método `set_todo_item` para obtener solo los `todo_items` que pertenezcan a `@todo_list`.

```
$ rails c  
> list_id=TodoList.first.id  
> @todo_list=TodoList.find(list_id)  
> item_id=@todo_list.todo_items.first.id  
> @todo_item=@todo_list.todo_items.find(item_id)
```

e. Invocar al método `set_todo_list` antes de ejecutar cualquier action utilizando `before_action`

f. Actualizar el action `todo_item#new` para retornar una nueva instancia de `TodoItem` relacionada a `@todo_list`.

```
$ rails c  
> @todo_list=TodoList.first  
> @todo_item=@todo_list.todo_items.new
```

g. Actualizar el action `todo_item#create` para crear un `TodoItem` basado en `todo_items_params` como antes, con la diferencia que ahora esta nueva instancia está relacionada a `@todo_list`.

```
$ rails c
> @todo_list=TodoList.first
> @todo_item=@todo_list.todo_items.new(title:"my item")
> @todo_list.save()
```

h. Actualizar el HTML redirect de los actions `todo_item#create`, `todo_item#update`, y `todo_item#destroy` para volver a `todo_list#show`.

Ejercicio 7



```
rspec spec/nested_resources_spec.rb -e rq07a
```

a. Actualizar los enlaces de `todo_items#show` para incluir al `TodoList` del cual es miembro el `TodoItem`.

- Cambiar el link Edit que hace referencia a `edit_todo_item` que no existe en el `link_to` al nuevo path `edit_todo_list_todo_item_path`: este route requiere que se le pase `@todo_list` y `'todo_item'` como parámetros separados:

```
from: <%= link_to 'Edit', edit_todo_item_path(@todo_item) %> |  
to:   <%= link_to 'Edit', edit_todo_list_todo_item_path(@todo_list, @todo_item) %> |
```

- Cambiar el link Back `link_to` q
 - de : `<%= link_to 'Back', todo_items_path %>`
 - to: `<%= link_to 'Back', @todo_list %>`

```
from: <%= link_to 'Back', todo_items_path %>  
to:   <%= link_to 'Back', @todo_list %>
```

b. Actualizar los links en `todo_items#edit` para incluir los miembros `TodoList/TodoItem`

```
from: <%= link_to 'Show', @todo_item %> |  
to:    <%= link_to 'Show', [@todo_list, @todo_item] %> |
```

```
from: <%= link_to 'Back', todo_items_path %>  
to:    <%= link_to 'Back', @todo_list %>
```


c. Actualizar los parametros del los form_for del partial todo_items/_form.html.erb

```
from: <%= form_for(@todo_item) do |f| %>  
to:   <%= form_for([@todo_list, @todo_item]) do |f| %>
```

```
rspec spec/nested_resources_spec.rb -e rq07c
```

d. Actualizar los links de `todo_items#new` para incluir al `TodoList`

```
from: <%= link_to 'Back', todo_items_path %>  
to:   <%= link_to 'Back', @todo_list %>
```

```
rspec spec/nested_resources_spec.rb -e rq07d
```

e. Hacer que se muestre el campo completed (boolean, checkbox) solo cuando se está editando. No debe aparecer en la creación de un Item.

Se puede saber si un objeto está persistido con el método persisted?, y también si es nuevo con el método new_record?

```
rspec spec/nested_resources_spec.rb -e rq07e
```

Solución 6 y 7

Ejercicio 8

Verificar que funcione el bcrypt con rails c, para proceder a crear el controller de sesiones.

```
rspec spec/security_spec.rb -e rq08
```

```
$ rails c
> user=User.where(username:"rich").first
> user.authenticate("wrongpassword")
=> false

> user=User.where(username:"rich").first
> user.authenticate("123abc")
=> #<User id: 277, username: "rich", password_digest: "$2a...

> user=User.where(username:"rich").first
> user.authenticate("123abc").todo_lists.count
=> 49 #seed data randomly generated
```

Ejercicio 9

- a. Utilizar rails g controller para crear un session controller con los actions new, create y destroy.
- b. Cambiar los routes generados en el routes.rb de las siguiente manera:

```
get 'sessions/new'  
get 'sessions/create'  
get 'sessions/destroy'
```



```
resources :sessions, only: [:new, :create, :destroy]
```

- c. Mapear GET /login a sessions#new y DELETE /logout a sessions#destroy

```
get "/login" => "sessions#new", as: "login"  
delete "/logout" => "sessions#destroy", as: "logout"
```

```
rspec spec/security_spec.rb -e rq09
```


Ejercicio 10

Implementar Sessions Controller & View

1. Editor view/sessions/new.html.erb

```
<h1>Login</h1>
<%= form_for(:user, url: sessions_path) do |f| %>
  <%= f.label :username %>

  <%= f.text_field :username %>

  <%= f.label :password %>

  <%= f.password_field :password %>

  <%= f.submit "Login" %>

<% end %>
```

rspec spec/security_spec.rb -e rq10b

Implementar el método create

- Obtener el nombre de usuario y password del formulario.
- Encontrar al usuario en la base de datos por su username
- Autenticar al usuario con el password provisto.
 - Si autentica: guardar user.id en la sesión, y enviar al root_path
 - Si no autentica: volver al login_path y mostrar un mensaje flash.alert

```
rspec spec/security_spec.rb -e rq10c
```

Ejercicio 11

```
rspec spec/authentication_spec.rb -e rq11
```

Autenticación

- En ApplicationController:
 - Crear el helper method **logged_in?** que verifique si un usuario está logeado.
 - Crear el helper `current_user` que encuentre la instancia de User asociada con la sesión.
 - Definir el método **ensure_login** que redirige a la página de login en caso de que el usuario no esté logeado.
 - Hacer que **ensure_login** se llame en un `before_action`
- En Session Controller
 - Agregar una excepción al `before_action` para no llamar a `ensure_login` al ingresar a `/login`.
- `application.html.erb`
 - Agregar información del usuario logeado:

```
<% if logged_in? %>

Logged in as <%= current_user.username %> | <%= link_to "Logout", logout_path, method:
:delete %>

<% end %>
```

Ejercicio 12

```
rspec spec/authentication_spec.rb -e rq12
```

Actualizar el controller de TodoList para limitar el acceso a las listas de tareas.

En `current_user` se encuentra el usuario logeado. Este usuario solo debe tener acceso a sus listas de tareas.

```
$ rails c  
> user_id=User.where(username:"rich").first.id  
> current_user=User.find(user_id)
```

Este cambio mas que nada implica:

Cambiar todos los `TodoList.XXXX` por `current_user.todo_lists.XXXX`

Ejercicio 13



Agregar paginación!

Agregar una línea al controller.

Agregar una línea al view.