

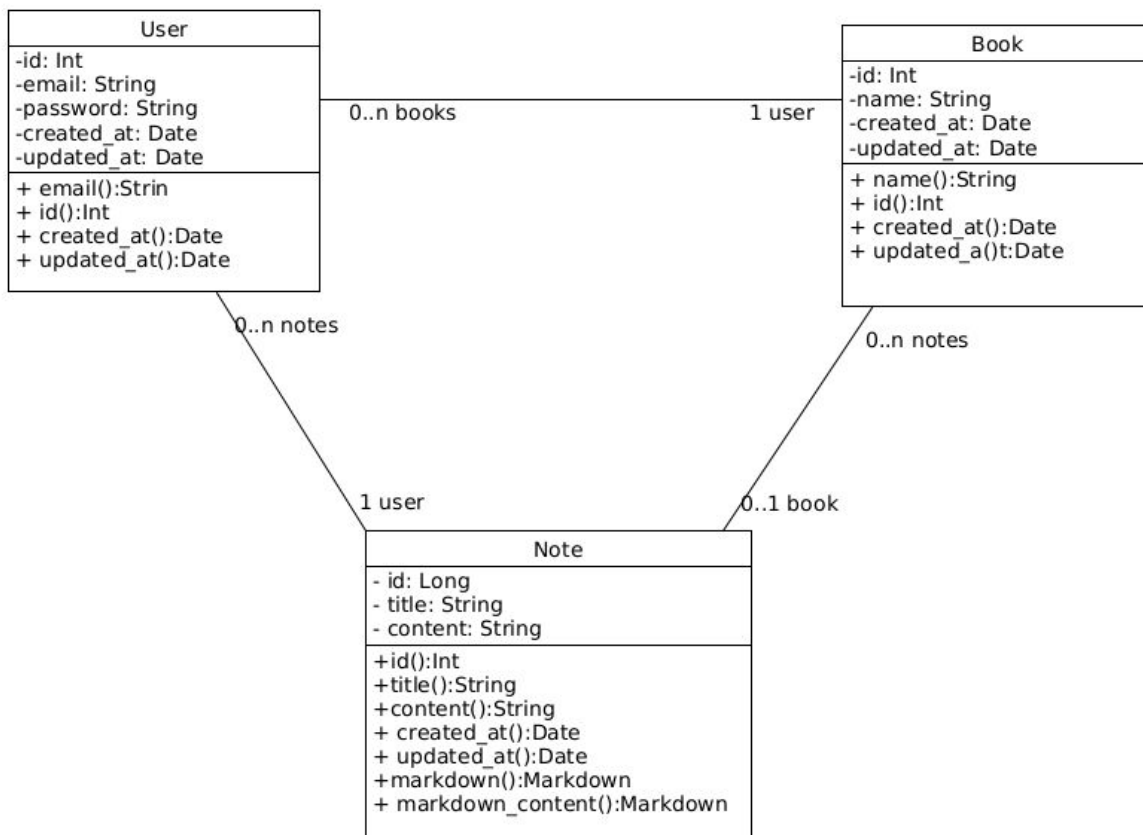
Entrega N°3

Alumna: Cossio Aquino, Ana Mariela
legajo: 143494

- Se utilizó:
Rails 6.1, por esta razón tuve que instalar **nvm install v15.7.0** y **node js**
Yarn → para los estilos: **yarn add bootstrap jquery popper.js**
- Decidí optar por las siguientes gemas:

```
gem 'devise' → Para la autenticación
gem 'hirb' → Para usar en la consola de rails
gem 'rails-i18n' → Para la internacionalización
gem 'will_paginate', '~> 3.1.1' → Para el paginado
gem 'will_paginate-bootstrap4' → para los estilos del paginado
gem 'redcarpet' → Para transformar el texto en markdown
gem 'rubyzip' → Para poder comprimir los archivos
gem 'zip-zip' → Para comprimir los archivos
```

Diseño del modelo:



Restricciones:

El título de una nota se podrá repetir si pertenece a distintos cuadernos:

```
validates_uniqueness_of :title, scope: [:book_id, :user_id]
```

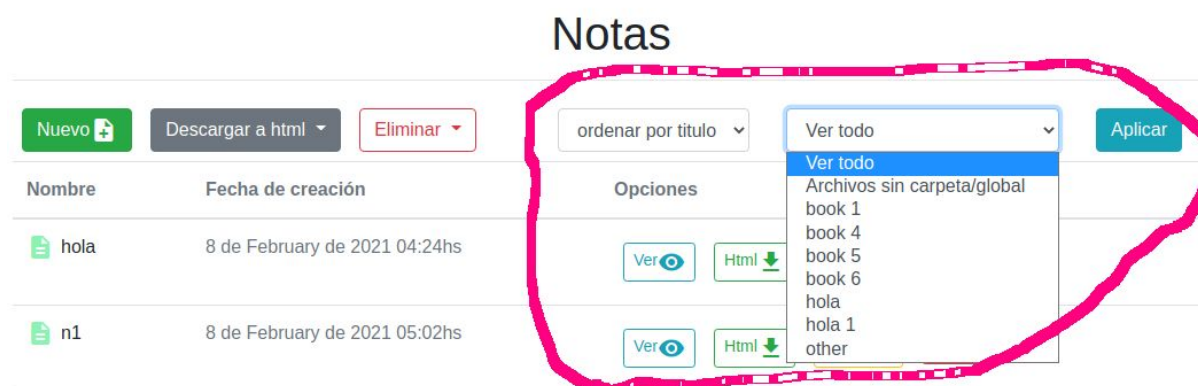
En cambio para el modelo Book, un usuario no puede tener dos cuadernos con el mismo nombre:

```
validates_uniqueness_of :name, scope: :user_id
```

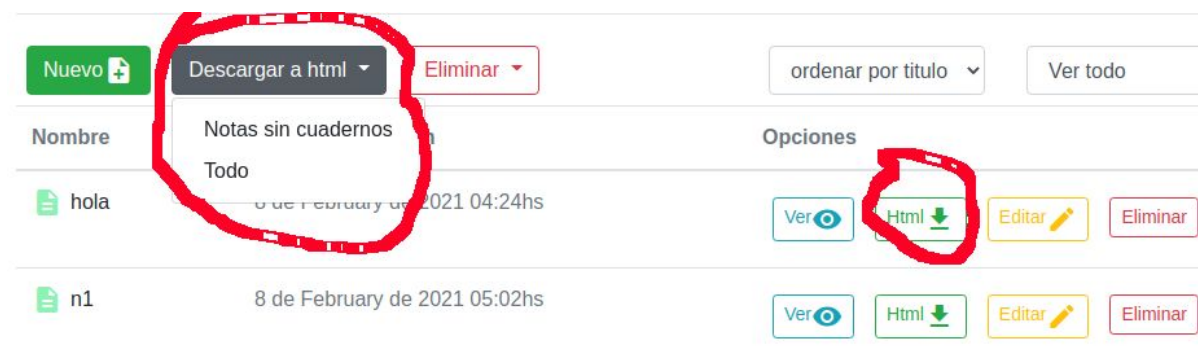
Y el nombre “global”, está reservado, por ende no se puede utilizar para los cuadernos.

Acerca de los filtros

Para que las notas puedan filtrarse por cuadernos y además ordenarse alfabéticamente, utilice un objeto NotesFilter y acompañado de un presenters, para tener los controladores lo más limpios posibles.



Además en dicha sección se podrá descargar en forma grupal en formato .zip, las notas que **no tienen** una carpeta asociada o descargar todas las notas posea el usuario logueado o las notas de forma individual.



Para las siguientes opciones:

- Descargar todas las notas que pertenezcan a la carpeta **global** en formato .zip
- Descargar todas las notas de **un cuaderno** en particular en formato .zip
- El usuario quiere descargar todas sus notas de un usuario en formato .zip

Se utilizó los concerns, dado que esto me permitió abstraer el código de los controladores, cuando se quiere descargar un archivo zip con las múltiples notas.

```
require "active_support/concern"
require 'rubygems'
require 'zip'
module Download
  extend ActiveSupport::Concern
  def to_html(name_book="all",notes)
    FileUtils.mkdir_p("tmp/#{current_user.id}")
    FileUtils.mkdir_p("tmp/#{current_user.id}/#{name_book}")
    zipfile = Zip::File.open("tmp/#{current_user.id}.zip", Zip::File::CREATE)
    zipfile.mkdir("#{name_book}")
    notes.each do |note|
      File.write("tmp/#{current_user.id}/#{name_book}/#{note.title_for_download}.html",note.markdown)
      zipfile.add("#{name_book}/#{note.title_for_download}.html",
"tmp/#{current_user.id}/#{name_book}/#{note.title_for_download}.html" )
    end
    zipfile.close
    (Dir.new("tmp/#{current_user.id}/#{name_book}").children().map{|d|
File.delete("tmp/#{current_user.id}/#{name_book}/#{d}")}
    Dir.rmdir("tmp/#{current_user.id}/#{name_book}")
    Dir.rmdir("tmp/#{current_user.id}/")
    send_data(File.read("tmp/#{current_user.id}.zip"), filename: "#{name_book}.zip",
type: 'application/zip' )
    File.delete("tmp/#{current_user.id}.zip")
  end
end
```

De esta manera en el controlador de las notas solo se deberá realizar lo siguiente:

```
class NotesController < LoggedInController
  def download_global
    name_book="global"
    notes=Note.notes_global(current_user.id)
    if !notes.empty?
      to_html(name_book,notes)
    else
      flash[:alert] = I18n.t(:error_download)
      redirect_to logged_in_books_path and return
    end
  end
end
```

```

        end
    end

    def download_all
        name_book="all"
        notes=current_user.notes
        if !notes.empty?
            to_html(name_book,notes)
        else
            flash[:alert] = I18n.t(:error_download)
            redirect_to logged_in_books_path and return
        end
    end
end
end
end

```

En el controlador de los cuadernos:

```

class BooksController < LoggedInController
    def download
        name_book=@book.name
        notes=@book.notes
        if !notes.empty?
            to_html(name_book,notes)
        else
            flash[:alert] = I18n.t(:error_download)
            redirect_to logged_in_books_path and return
        end
    end

end
end
end

```