

MANUAL TECNICO DOMITRANSP



Domitransp SAS

MANUAL TECNICO DOMITRANSP

1. Descripción del proyecto:

“Domitransp SAS” es una plataforma web diseñada para simplificar la programación y gestión eficiente de envíos a través de los servicios de logística y transporte ofrecidos por Domitransp SAS. Esta plataforma proporciona a los usuarios la capacidad de programar envíos, realizar un seguimiento de sus pedidos, finanzas y gastos que han hecho y podrán ver su empresa y usuarios conectados a ella.

Tecnologías Utilizadas:

- Ruby on Rails: Marco de desarrollo web.
- Devise: Para la autenticación de usuarios.
- HTML, CSS, JavaScript: Tecnologías web estándar.
- Base de Datos: PostgreSQL.

Dependencias Externas:

- RubyGems para la gestión de gemas.
- Bundler para gestionar dependencias de Ruby

Control de Acceso:

- Se establecen roles de usuario, como administrador y usuario estándar.
- El acceso a ciertas funciones está restringido según el rol del usuario.

Autenticación y Autorización: El sistema utiliza Devise para autenticar usuarios y gestionar la autorización basada en roles. Se implementan políticas de contraseñas seguras.

Modelo Vista Controlador : La arquitectura se organiza en tres componentes principales:

- **Modelo:** Representa los datos y la lógica de negocio.
- **Vista:** Encargada de la presentación y la interfaz de usuario.
- **Controlador:** Gestiona las interacciones del usuario y coordina el flujo de datos.

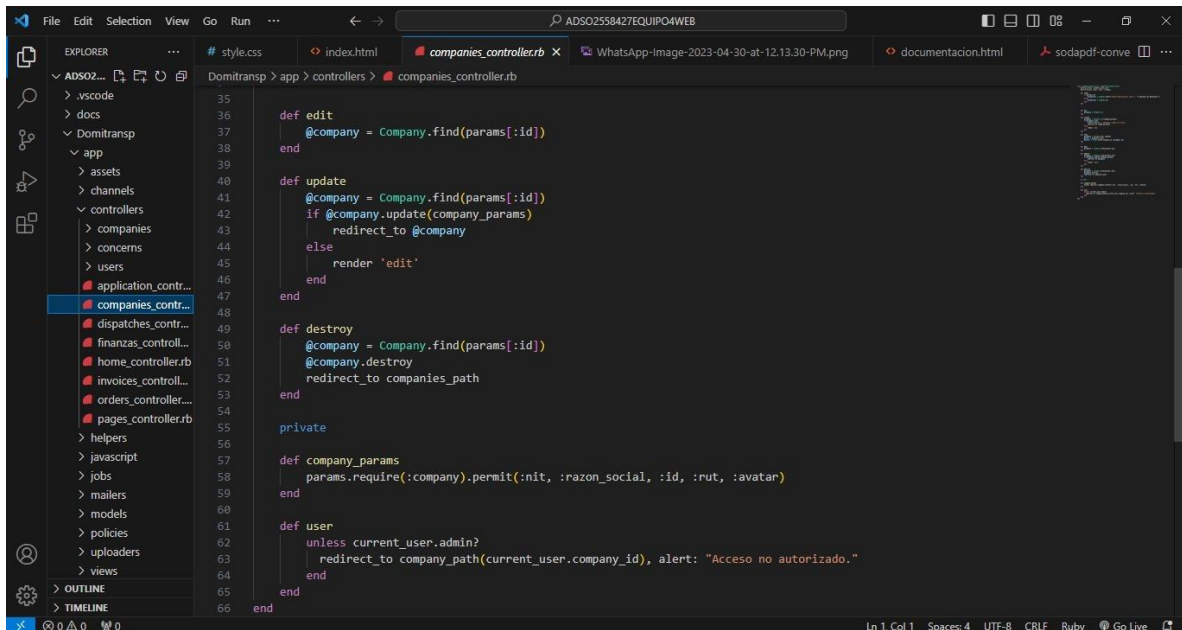
Base de Datos: Se utiliza PostgreSQL como sistema de gestión de bases de datos para almacenar de manera segura y eficiente la información del sistema.

3.codigo fuente:

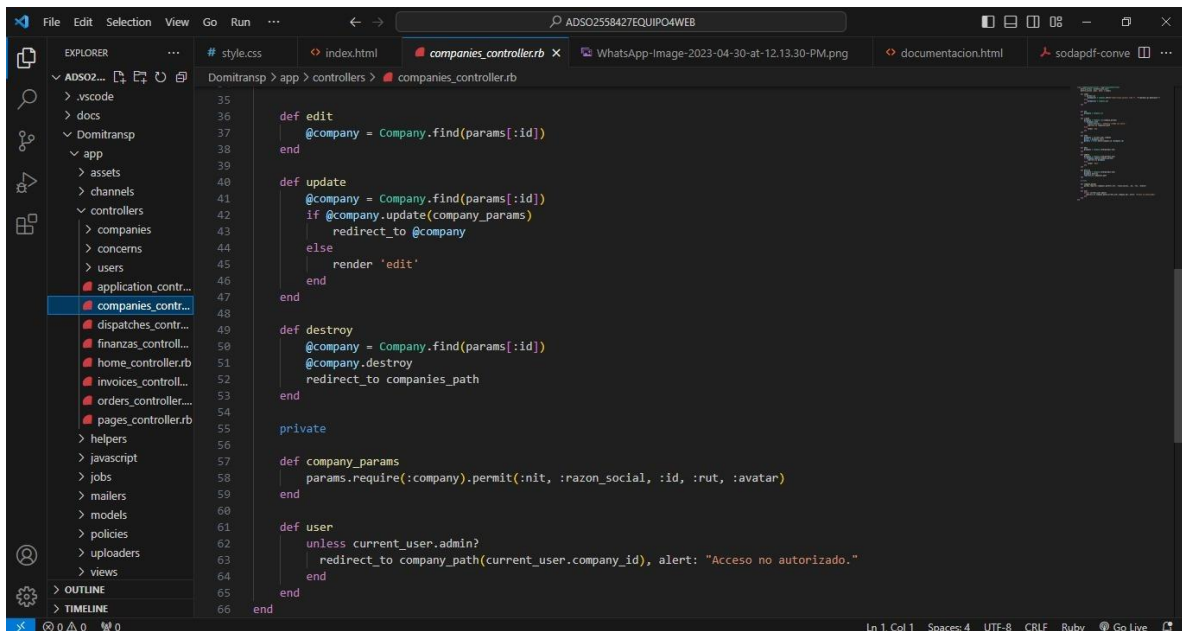
En esta sección podrán ver como esta compuesta nuestra aplicación sus controladores, schema, vistas, modelos y migraciones.

Controladores:

Company:



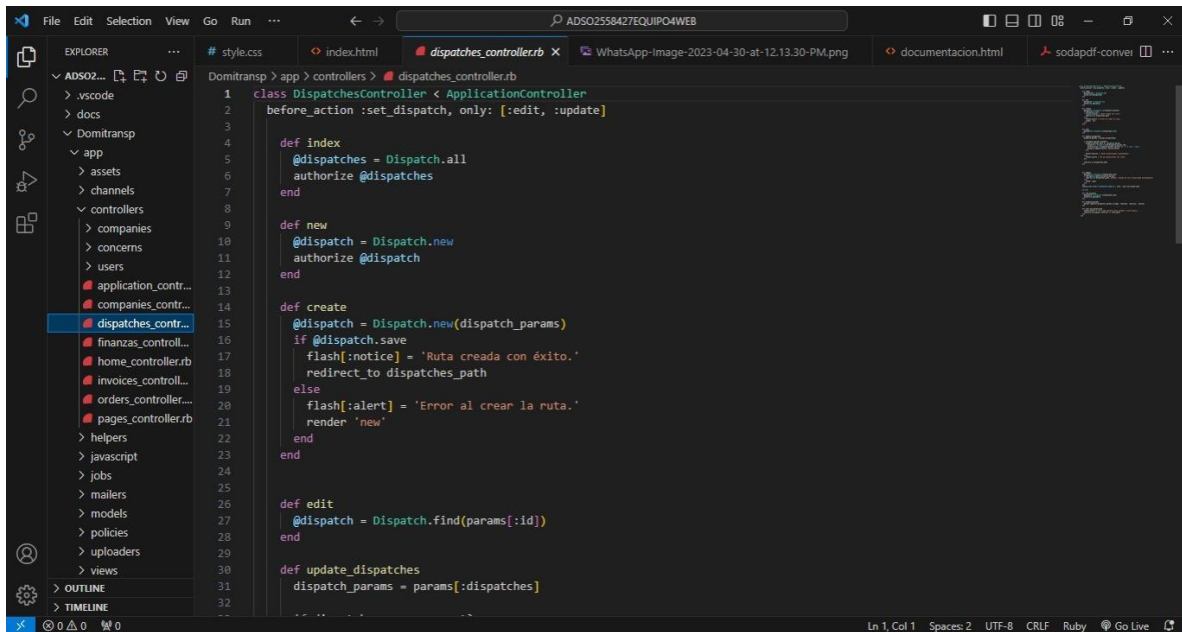
```
35
36
37 def edit
38   @company = Company.find(params[:id])
39 end
40
41 def update
42   @company = Company.find(params[:id])
43   if @company.update(company_params)
44     redirect_to @company
45   else
46     render 'edit'
47   end
48 end
49
50 def destroy
51   @company = Company.find(params[:id])
52   @company.destroy
53   redirect_to companies_path
54 end
55
56 private
57
58 def company_params
59   params.require(:company).permit(:nit, :razon_social, :id, :rut, :avatar)
60 end
61
62 def user
63   unless current_user.admin?
64     redirect_to company_path(current_user.company_id), alert: "Acceso no autorizado."
65   end
66 end
```



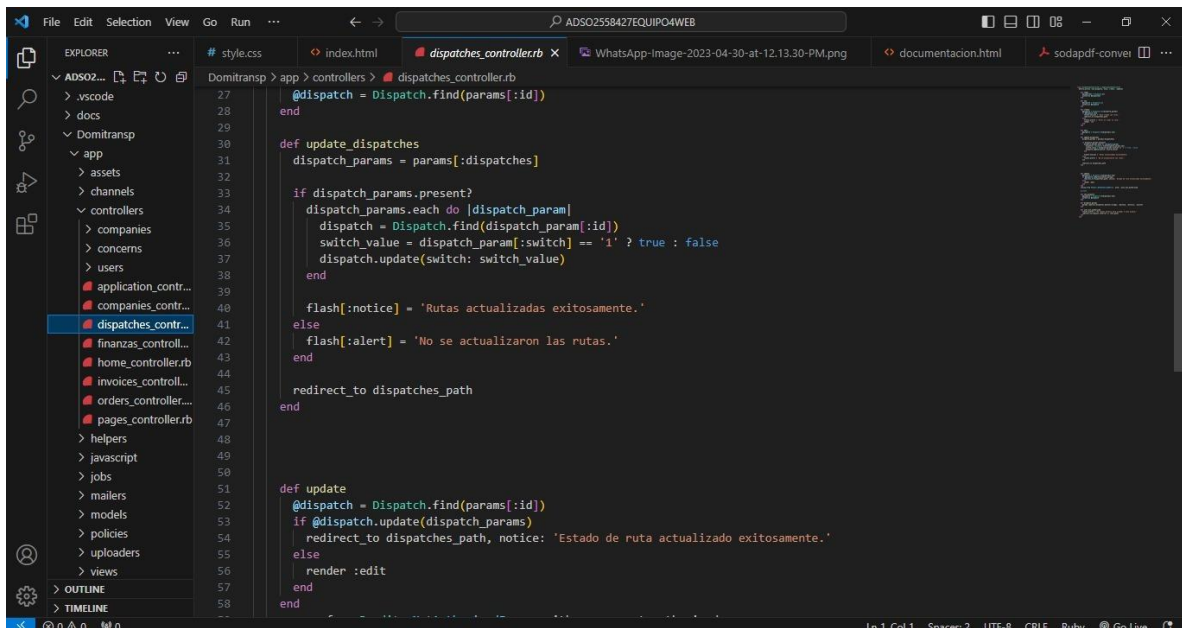
```
35
36
37 def edit
38   @company = Company.find(params[:id])
39 end
40
41 def update
42   @company = Company.find(params[:id])
43   if @company.update(company_params)
44     redirect_to @company
45   else
46     render 'edit'
47   end
48 end
49
50 def destroy
51   @company = Company.find(params[:id])
52   @company.destroy
53   redirect_to companies_path
54 end
55
56 private
57
58 def company_params
59   params.require(:company).permit(:nit, :razon_social, :id, :rut, :avatar)
60 end
61
62 def user
63   unless current_user.admin?
64     redirect_to company_path(current_user.company_id), alert: "Acceso no autorizado."
65   end
66 end
```

Este controlador proporciona funcionalidades para listar, crear, mostrar, editar y eliminar empresas. También incluye medidas de seguridad como la autenticación de usuarios y la autorización para ciertas acciones. La acción `user` verifica si el usuario actual es un administrador antes de permitir el acceso a la acción `index`.

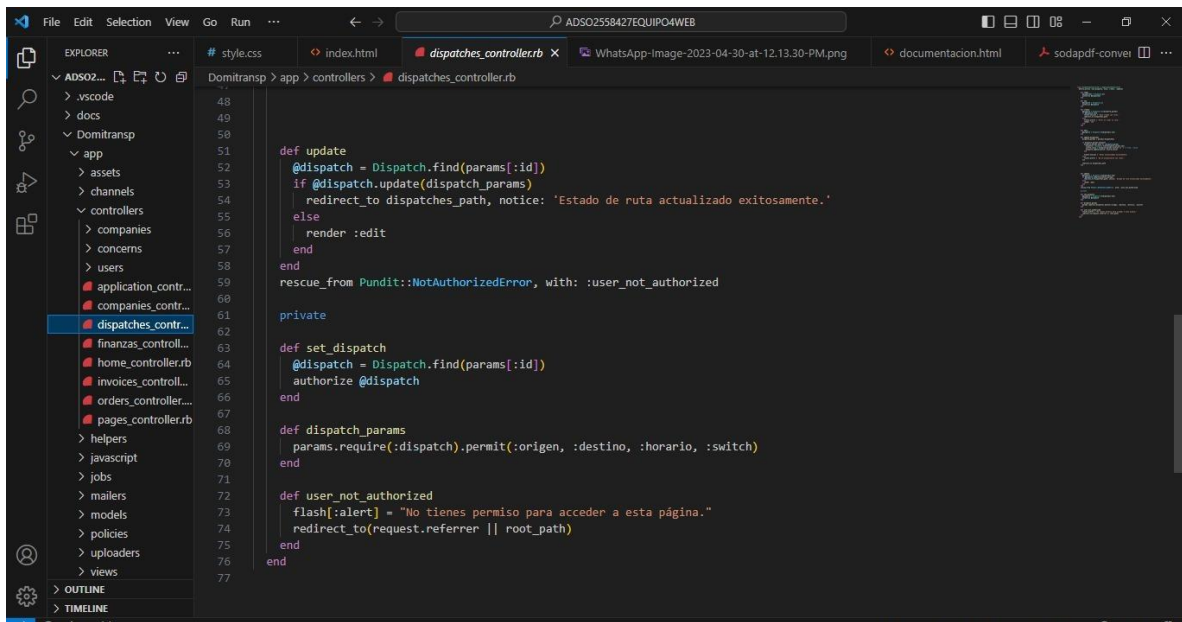
dispatch



```
1 class DispatchesController < ApplicationController
2   before_action :set_dispatch, only: [:edit, :update]
3
4   def index
5     @dispatches = Dispatch.all
6     authorize @dispatches
7   end
8
9   def new
10    @dispatch = Dispatch.new
11    authorize @dispatch
12  end
13
14  def create
15    @dispatch = Dispatch.new(dispatch_params)
16    if @dispatch.save
17      flash[:notice] = 'Ruta creada con éxito.'
18      redirect_to dispatches_path
19    else
20      flash[:alert] = 'Error al crear la ruta.'
21      render :new
22    end
23  end
24
25  def edit
26    @dispatch = Dispatch.find(params[:id])
27  end
28
29  def update_dispatches
30    dispatch_params = params[:dispatches]
```



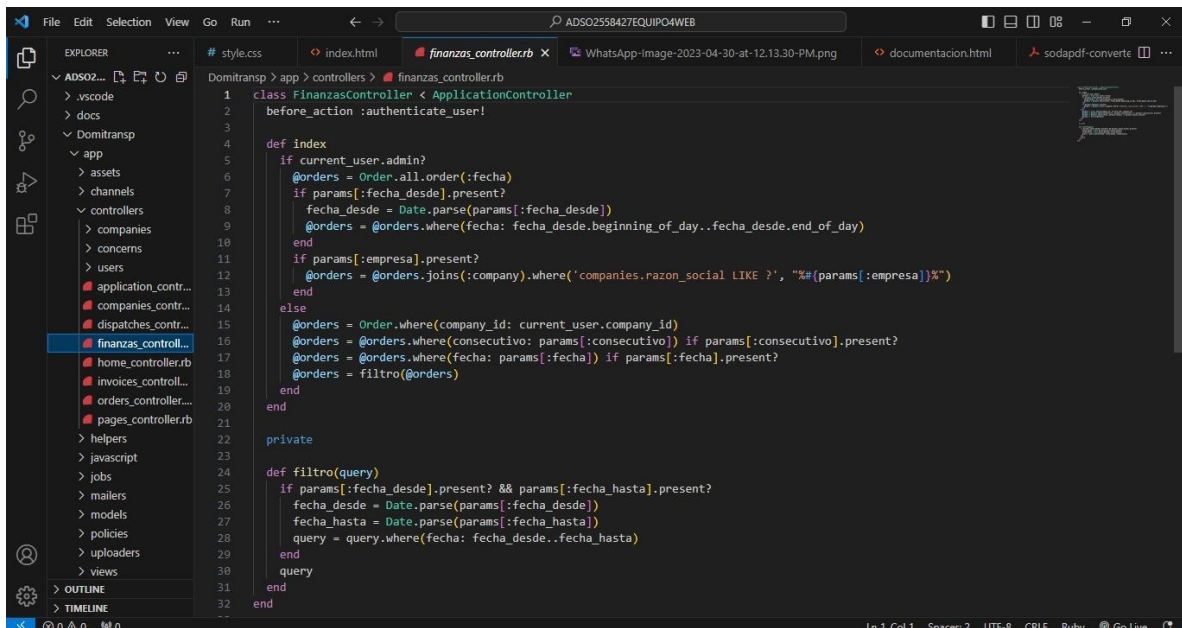
```
27 @dispatch = Dispatch.find(params[:id])
28 end
29
30 def update_dispatches
31   dispatch_params = params[:dispatches]
32
33   if dispatch_params.present?
34     dispatch_params.each do |dispatch_param|
35       dispatch = Dispatch.find(dispatch_param[:id])
36       switch_value = dispatch_param[:switch] == '1' ? true : false
37       dispatch.update(switch: switch_value)
38     end
39     flash[:notice] = 'Rutas actualizadas exitosamente.'
40   else
41     flash[:alert] = 'No se actualizaron las rutas.'
42   end
43   redirect_to dispatches_path
44 end
45
46 def update
47   @dispatch = Dispatch.find(params[:id])
48   if @dispatch.update(dispatch_params)
49     redirect_to dispatches_path, notice: 'Estado de ruta actualizado exitosamente.'
50   else
51     render :edit
52   end
53 end
```



```
48
49
50
51
52 def update
53   @dispatch = Dispatch.find(params[:id])
54   if @dispatch.update(dispatch_params)
55     redirect_to dispatches_path, notice: 'Estado de ruta actualizado exitosamente.'
56   else
57     render :edit
58   end
59 end
60 rescue_from Pundit::NotAuthorizedError, with: :user_not_authorized
61
62 private
63
64 def set_dispatch
65   @dispatch = Dispatch.find(params[:id])
66   authorize @dispatch
67 end
68
69 def dispatch_params
70   params.require(:dispatch).permit(:origen, :destino, :horario, :switch)
71 end
72
73 def user_not_authorized
74   flash[:alert] = "No tienes permiso para acceder a esta página."
75   redirect_to(request.referrer || root_path)
76 end
77 end
```

Este controlador utiliza la gema Pundit para manejar la autorización. Proporciona funcionalidades para listar, crear, mostrar, editar y actualizar rutas de envío. Además, incluye medidas de seguridad como la autenticación de usuarios y la autorización para ciertas acciones.

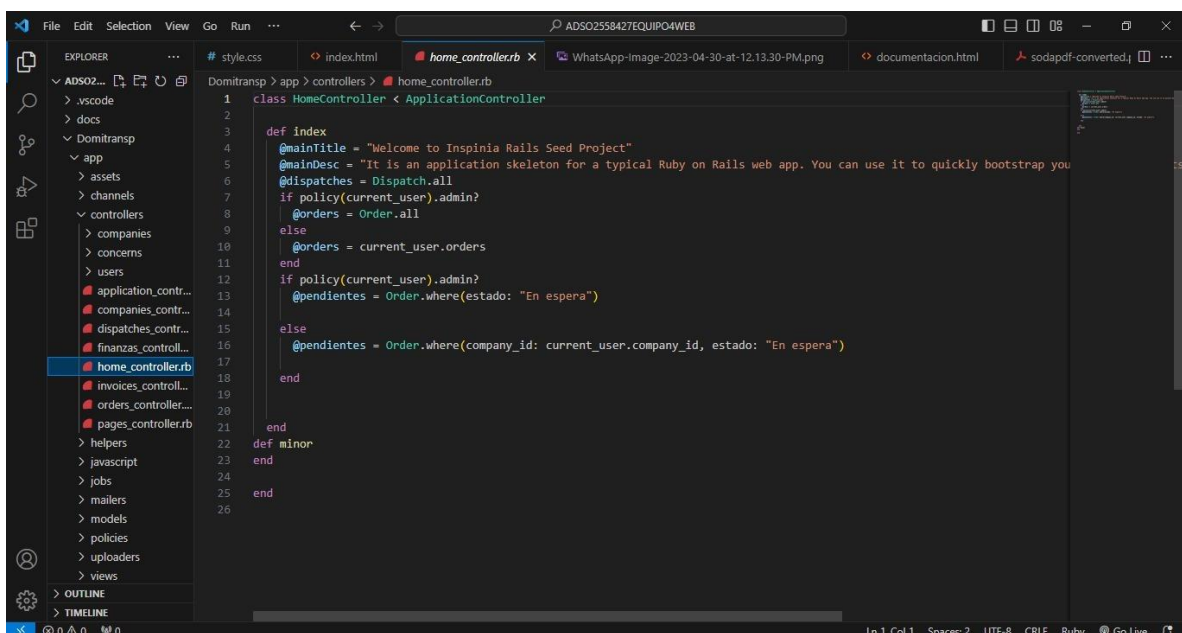
Finanzas:



```
1 class FinanzasController < ApplicationController
2   before_action :authenticate_user!
3
4   def index
5     if current_user.admin?
6       @orders = Order.all.order(:fecha)
7       if params[:fecha_desde].present?
8         fecha_desde = Date.parse(params[:fecha_desde])
9         @orders = @orders.where(fecha: fecha_desde.beginning_of_day..fecha_desde.end_of_day)
10      end
11      if params[:empresa].present?
12        @orders = @orders.joins(:company).where('companies.razon_social LIKE ?', "%#{params[:empresa]}%")
13      end
14    else
15      @orders = Order.where(company_id: current_user.company_id)
16      @orders = @orders.where(consecutivo: params[:consecutivo]) if params[:consecutivo].present?
17      @orders = @orders.where(fecha: params[:fecha]) if params[:fecha].present?
18      @orders = filtro(@orders)
19    end
20  end
21
22  private
23
24  def filtro(query)
25    if params[:fecha_desde].present? && params[:fecha_hasta].present?
26      fecha_desde = Date.parse(params[:fecha_desde])
27      fecha_hasta = Date.parse(params[:fecha_hasta])
28      query = query.where(fecha: fecha_desde..fecha_hasta)
29    end
30    query
31  end
32 end
```

Este controlador proporciona funcionalidades para mostrar información financiera según el rol del usuario (administrador o no administrador). Además, incluye la capacidad de filtrar los resultados por fecha, empresa y otros criterios específicos. La acción `filtro` es un método privado que ayuda a aplicar filtros adicionales según los parámetros proporcionados.

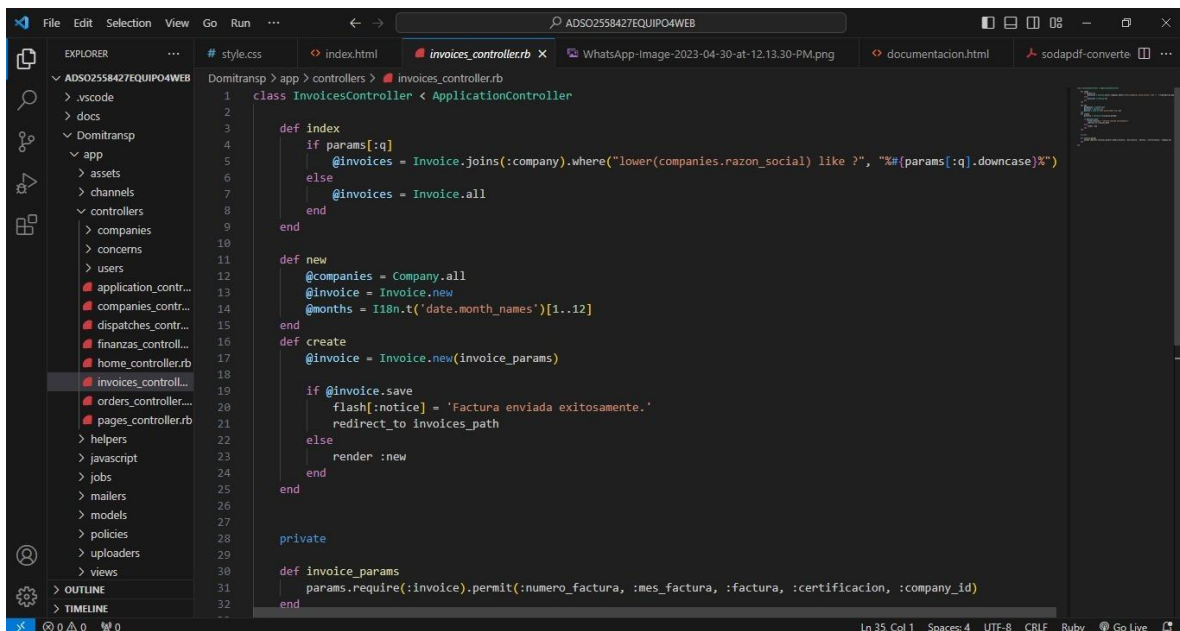
Home:



```
1 class HomeController < ApplicationController
2
3   def index
4     @mainTitle = "Welcome to Inspinia Rails Seed Project"
5     @mainDesc = "It is an application skeleton for a typical Ruby on Rails web app. You can use it to quickly bootstrap your application."
6     @dispatches = Dispatch.all
7     if policy(current_user).admin?
8       @orders = Order.all
9     else
10      @orders = current_user.orders
11    end
12    if policy(current_user).admin?
13      @pendientes = Order.where(estado: "En espera")
14    else
15      @pendientes = Order.where(company_id: current_user.company_id, estado: "En espera")
16    end
17  end
18
19  def minor
20  end
21
22 end
```

Este controlador proporciona funcionalidades para mostrar información en la página de inicio. Además, incluye la capacidad de diferenciar entre administradores y usuarios normales para mostrar información específica para cada tipo de usuario. La acción `minor` parece ser una acción adicional sin lógica específica proporcionada en el código.

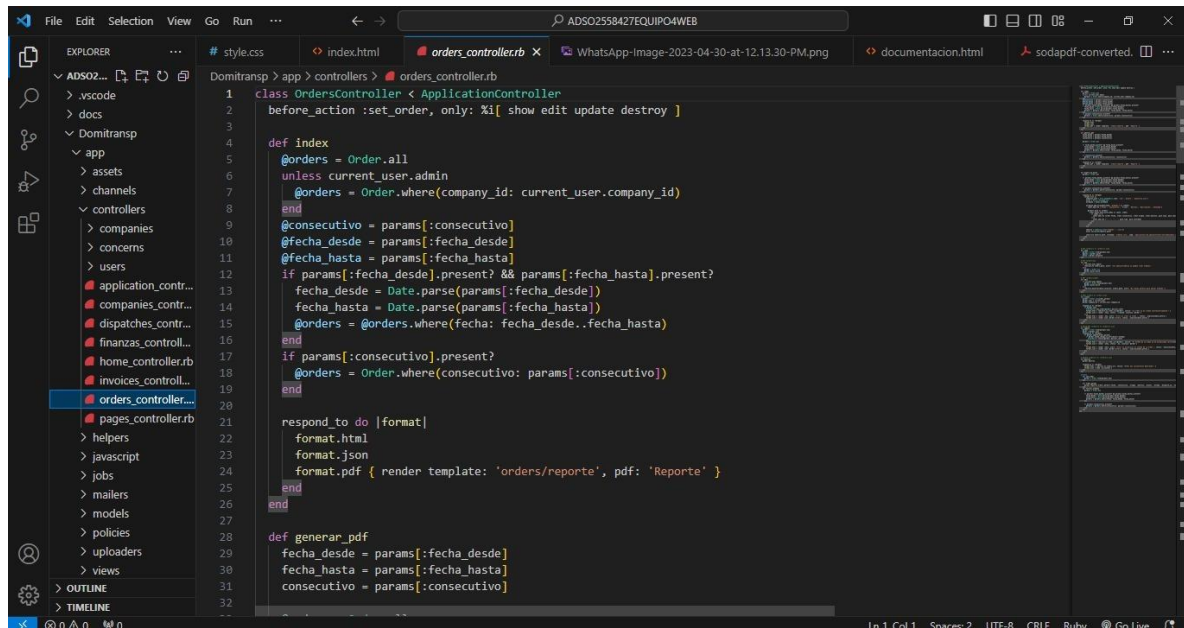
Invoices:



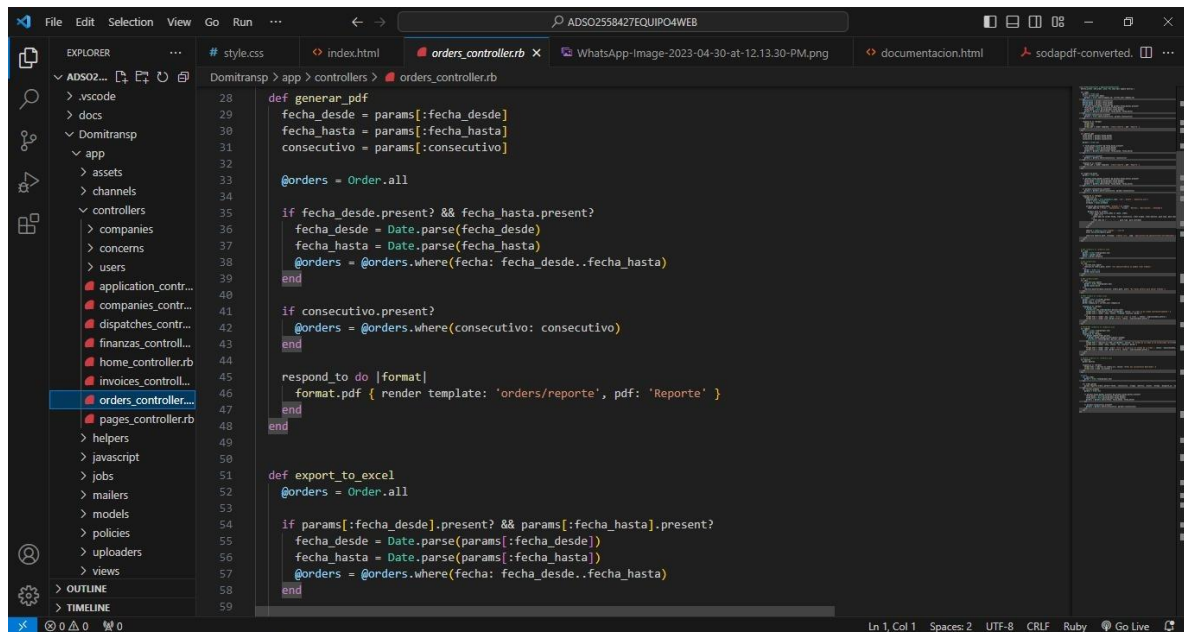
```
1 class InvoicesController < ApplicationController
2
3
4   def index
5     if params[:q]
6       @invoices = Invoice.joins(:company).where("lower(companies.razon_social) like ?", "%#{params[:q].downcase}%")
7     else
8       @invoices = Invoice.all
9     end
10  end
11
12  def new
13    @companies = Company.all
14    @invoice = Invoice.new
15    @months = I18n.t('date.month_names')[1..12]
16  end
17
18  def create
19    @invoice = Invoice.new(invoice_params)
20
21    if @invoice.save
22      flash[:notice] = 'Factura enviada exitosamente.'
23      redirect_to invoices_path
24    else
25      render :new
26    end
27  end
28
29  private
30
31  def invoice_params
32    params.require(:invoice).permit(:numero_factura, :mes_factura, :factura, :certificacion, :company_id)
33  end
34 end
```

Este controlador proporciona funcionalidades para listar, crear y mostrar facturas. La acción `index` puede filtrar las facturas por nombre de empresa. La acción `new` renderiza el formulario para crear una nueva factura y recupera todas las empresas disponibles. La acción `create` crea una nueva factura en la base de datos.

Orders:



```
1 class OrdersController < ApplicationController
2   before_action :set_order, only: %i[ show edit update destroy ]
3
4   def index
5     @orders = Order.all
6     unless current_user.admin
7       @orders = Order.where(company_id: current_user.company_id)
8     end
9     @consecutivo = params[:consecutivo]
10    @fecha_desde = params[:fecha_desde]
11    @fecha_hasta = params[:fecha_hasta]
12    if params[:fecha_desde].present? && params[:fecha_hasta].present?
13      fecha_desde = Date.parse(params[:fecha_desde])
14      fecha_hasta = Date.parse(params[:fecha_hasta])
15      @orders = @orders.where(fecha: fecha_desde..fecha_hasta)
16    end
17    if params[:consecutivo].present?
18      @orders = Order.where(consecutivo: params[:consecutivo])
19    end
20
21    respond_to do |format|
22      format.html
23      format.json
24      format.pdf { render template: 'orders/reporte', pdf: 'Reporte' }
25    end
26  end
27
28  def generar_pdf
29    fecha_desde = params[:fecha_desde]
30    fecha_hasta = params[:fecha_hasta]
31    consecutivo = params[:consecutivo]
```



```
28  def generar_pdf
29    fecha_desde = params[:fecha_desde]
30    fecha_hasta = params[:fecha_hasta]
31    consecutivo = params[:consecutivo]
32
33    @orders = Order.all
34
35    if fecha_desde.present? && fecha_hasta.present?
36      fecha_desde = Date.parse(fecha_desde)
37      fecha_hasta = Date.parse(fecha_hasta)
38      @orders = @orders.where(fecha: fecha_desde..fecha_hasta)
39    end
40
41    if consecutivo.present?
42      @orders = @orders.where(consecutivo: consecutivo)
43    end
44
45    respond_to do |format|
46      format.pdf { render template: 'orders/reporte', pdf: 'Reporte' }
47    end
48  end
49
50  def export_to_excel
51    @orders = Order.all
52
53    if params[:fecha_desde].present? && params[:fecha_hasta].present?
54      fecha_desde = Date.parse(params[:fecha_desde])
55      fecha_hasta = Date.parse(params[:fecha_hasta])
56      @orders = @orders.where(fecha: fecha_desde..fecha_hasta)
57    end
58  end
59 end
```



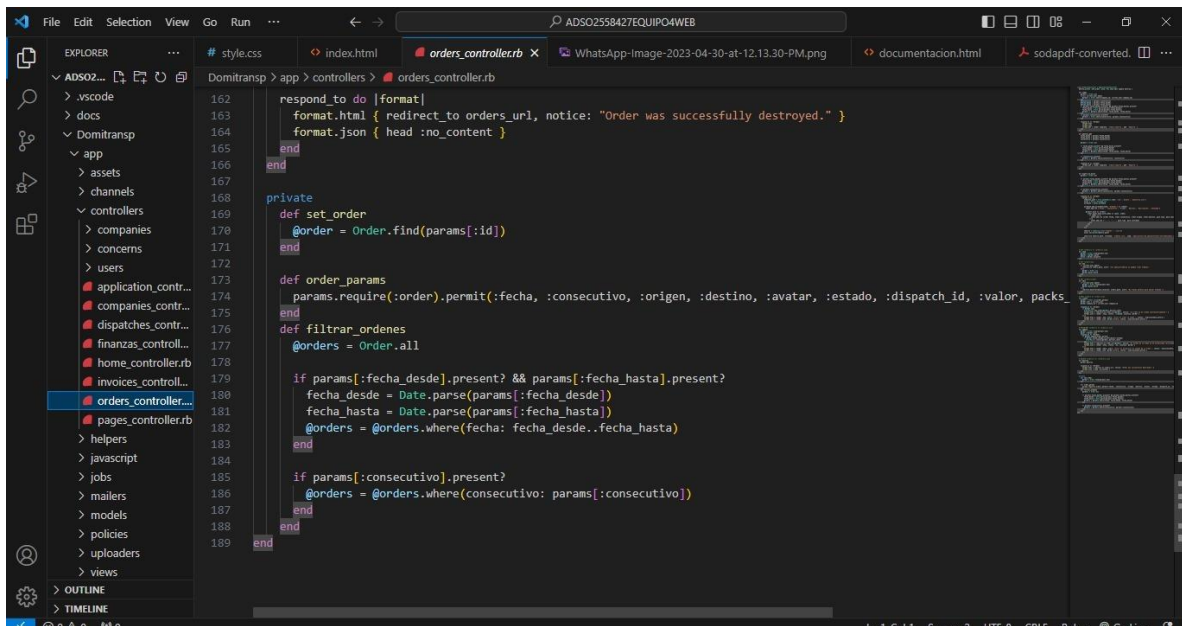
```
# style.css
index.html
orders_controller.rb
WhatsApp-Image-2023-04-30-at-12.13.30-PM.png
documentacion.html
sodapdf-converted.

Domitransp > app > controllers > orders_controller.rb
52 @orders = Order.all
53
54 if params[:fecha_desde].present? && params[:fecha_hasta].present?
55   fecha_desde = Date.parse(params[:fecha_desde])
56   fecha_hasta = Date.parse(params[:fecha_hasta])
57   @orders = @orders.where(fecha: fecha_desde..fecha_hasta)
58 end
59
60 if params[:consecutivo].present?
61   @orders = @orders.where(consecutivo: params[:consecutivo])
62 end
63
64 respond_to do |format|
65   format.xlsx do
66     template_path = File.join(Rails.root, 'lib', 'assets', 'plantilla.xlsx')
67     excel = Axlsx::Package.new
68     workbook = excel.workbook
69
70     workbook.add_worksheet(name: 'Ordenes') do |sheet|
71       sheet.add_row ['Fecha', 'Consecutivo', 'Origen', 'Destino', 'Descripción', 'Cantidad']
72
73       @orders.each do |order|
74         order.packs.each_with_index do |pack, index|
75           if index == 0
76             sheet.add_row [order.fecha, order.consecutivo, order.origen, order.destino, pack.tipo, pack.cantidad]
77           else
78             sheet.add_row ['', '', '', '', pack.tipo, pack.cantidad]
79           end
80         end
81       end
82     end
83   end
84 end
```

```
81 end
82 end
83
84 tmpfile = Tempfile.new(['ordenes', '.xlsx'])
85 excel.serialize(tmpfile.path)
86
87 send_file tmpfile.path, filename: 'ordenes.xlsx', type: 'application/vnd.openxmlformats-officedocument.spreadsheetml-workbook'
88 end
89 end
90
91 # GET /orders/1 or /orders/1.json
92 def show
93   @order = Order.find(params[:id])
94   @packs = @order.packs
95   @ruta = @order.dispatch
96 end
97
98 # GET /orders/new
99 def new
100   if current_user.admin?
101     redirect_to orders_path, alert: "Los administradores no pueden crear ordenes."
102   else
103     @order = Order.new
104     @order.packs.build
105   end
106 end
107
108 # GET /orders/1/edit
```

```
110 end
111
112 # GET /orders/1/edit
113 def edit
114   if current_user.admin?
115     @order = Order.find(params[:id])
116     @order.packs.build
117   else
118     redirect_back(fallback_location: orders_path, alert: "No tienes permiso para editar órdenes.")
119   end
120 end
121
122 # POST /orders or /orders.json
123 def create
124   @order = Order.new(order_params)
125   @order.user = current_user
126   @order.company_id = current_user.company_id
127
128   respond_to do |format|
129     if @order.save
130       OrderMailer.new_order(@order).deliver_later
131       format.html { redirect_to order_url(@order), notice: "La orden se ha creado satisfactoriamente." }
132       format.json { render :show, status: :created, location: @order }
133     else
134       format.html { render :new, alert: "Error al crear la orden.", status: :unprocessable_entity }
135       format.json { render json: @order.errors, status: :unprocessable_entity }
136     end
137   end
138 end
139
140 # PATCH/PUT /orders/1 or /orders/1.json
141 def update
```

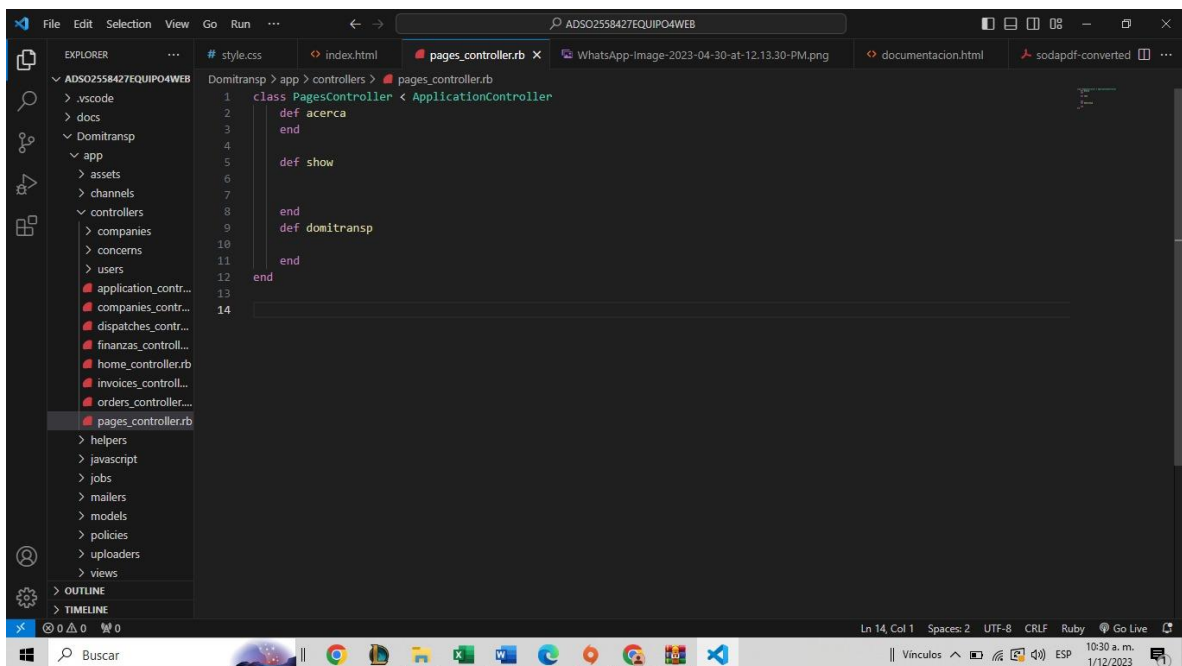
```
142   @order = Order.find(params[:id])
143   @user = @order.user
144   respond_to do |format|
145     if @order.update(order_params)
146       if @order.saved_change_to_attribute?(:estado)
147         OrderMailer.status(@order).deliver_later
148       end
149       format.html { redirect_to order_url(@order), notice: "El estado de la orden se ha actualizado correctamente." }
150       format.json { render :show, status: :ok, location: @order }
151     else
152       format.html { render :edit, alert: "Error al actualizar el estado de la orden.", status: :unprocessable_entity }
153       format.json { render json: @order.errors, status: :unprocessable_entity }
154     end
155   end
156 end
157
158 # DELETE /orders/1 or /orders/1.json
159 def destroy
160   @order.destroy
161
162   respond_to do |format|
163     format.html { redirect_to orders_url, notice: "Order was successfully destroyed." }
164     format.json { head :no_content }
165   end
166 end
167
168 private
169 def set_order
170   @order = Order.find(params[:id])
```



```
File Edit Selection View Go Run ...
ADSO2558427EQUIPO4WEB
EXPLORER
ADSO2...
> .vscode
> docs
> Domitransp
  app
    assets
    channels
    controllers
      companies
      concerns
      users
      application_contr...
      companies_contr...
      dispatches_contr...
      finanzas_controll...
      home_controller.rb
      invoices_controll...
      orders_controller...
      pages_controller.rb
    helpers
    javascript
    jobs
    mailers
    models
    policies
    uploaders
    views
  OUTLINE
  TIMELINE
Domitransp > app > controllers > orders_controller.rb
162 respond_to do |format|
163   format.html { redirect_to orders_url, notice: "Order was successfully destroyed." }
164   format.json { head :no_content }
165 end
166
167
168 private
169 def set_order
170   @order = Order.find(params[:id])
171 end
172
173 def order_params
174   params.require(:order).permit(:fecha, :consecutivo, :origen, :destino, :avatar, :estado, :dispatch_id, :valor, :packs_
175 end
176 def filtrar_ordenes
177   @orders = Order.all
178
179   if params[:fecha_desde].present? && params[:fecha_hasta].present?
180     fecha_desde = Date.parse(params[:fecha_desde])
181     fecha_hasta = Date.parse(params[:fecha_hasta])
182     @orders = @orders.where(fecha: fecha_desde..fecha_hasta)
183   end
184
185   if params[:consecutivo].present?
186     @orders = @orders.where(consecutivo: params[:consecutivo])
187   end
188 end
189 end
```

Este controlador sigue las convenciones de Ruby on Rails y proporciona funcionalidades para listar, crear, mostrar, editar y eliminar órdenes. También incluye acciones para generar informes en PDF y exportar a Excel.

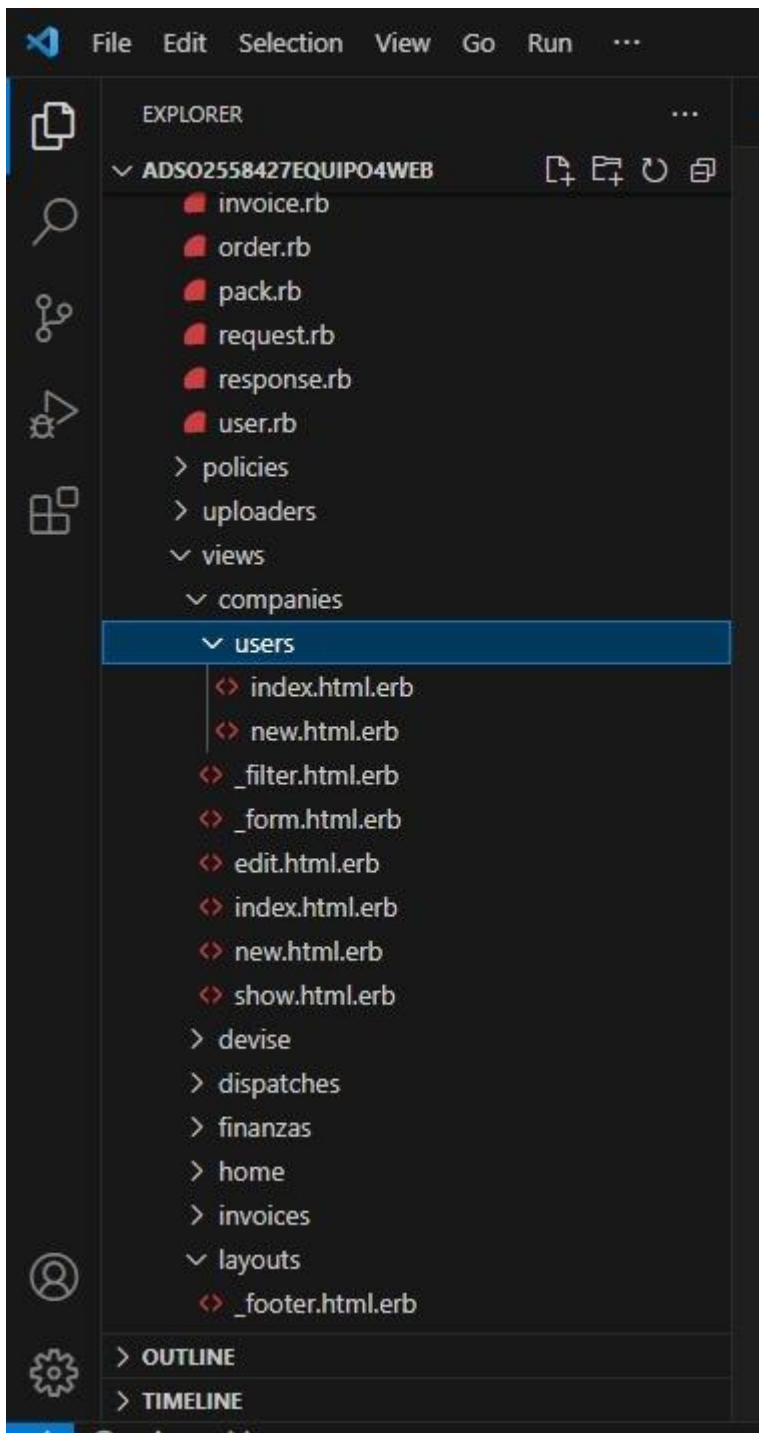
Pages:



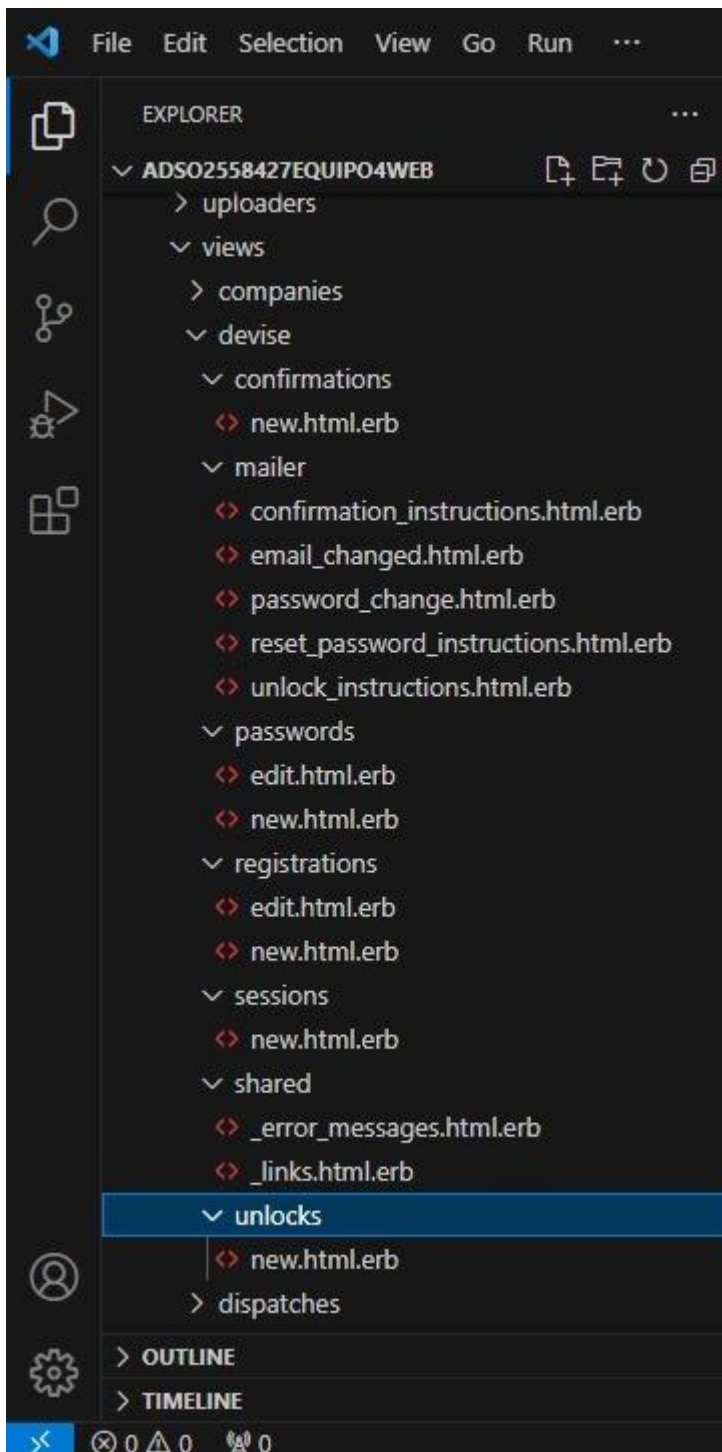
```
File Edit Selection View Go Run ...
ADSO2558427EQUIPO4WEB
EXPLORER
ADSO2558427EQUIPO4WEB
> .vscode
> docs
> Domitransp
  app
    assets
    channels
    controllers
      companies
      concerns
      users
      application_contr...
      companies_contr...
      dispatches_contr...
      finanzas_controll...
      home_controller.rb
      invoices_controll...
      orders_controller...
      pages_controller.rb
    helpers
    javascript
    jobs
    mailers
    models
    policies
    uploaders
    views
  OUTLINE
  TIMELINE
Domitransp > app > controllers > pages_controller.rb
1 class PagesController < ApplicationController
2   def acerca
3   end
4
5   def show
6
7
8   end
9   def domitransp
10
11
12 end
13
14
```

Este controlador proporciona acciones para renderizar páginas específicas.

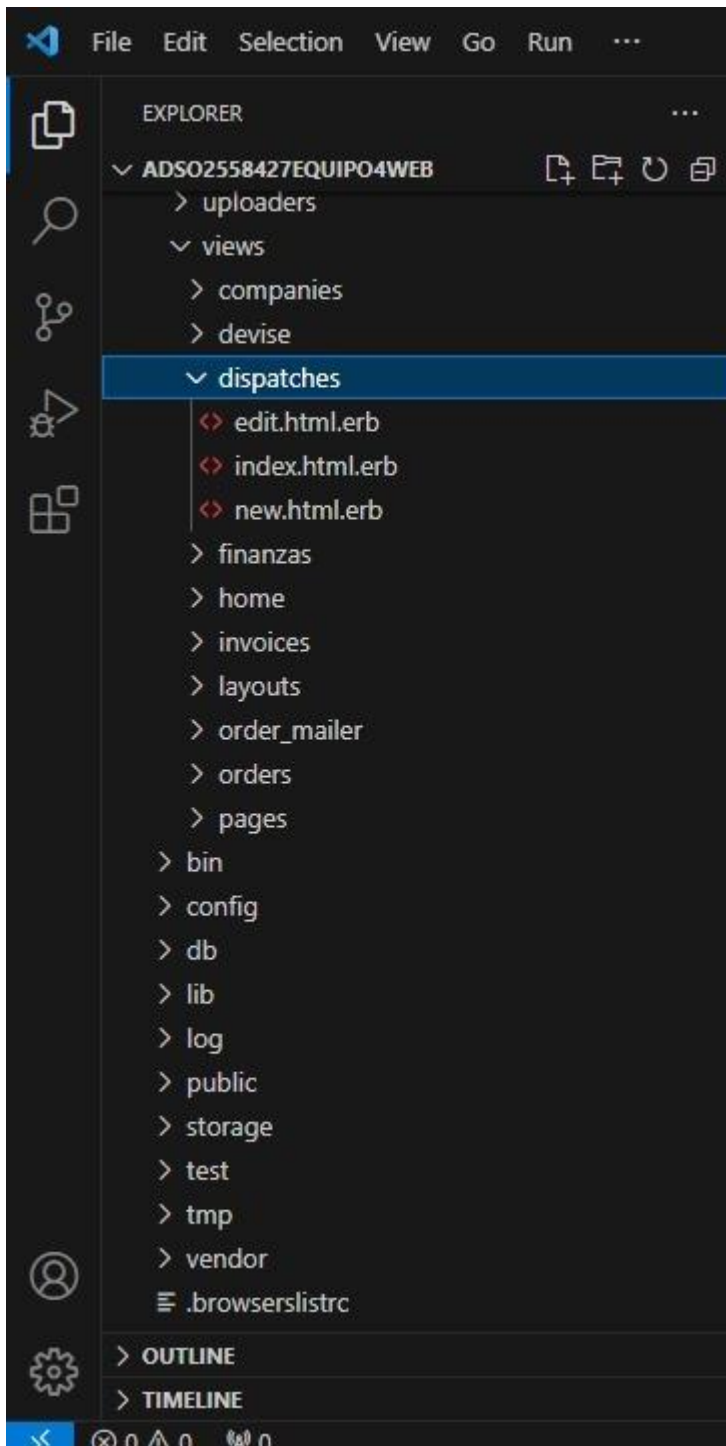
Vistas:



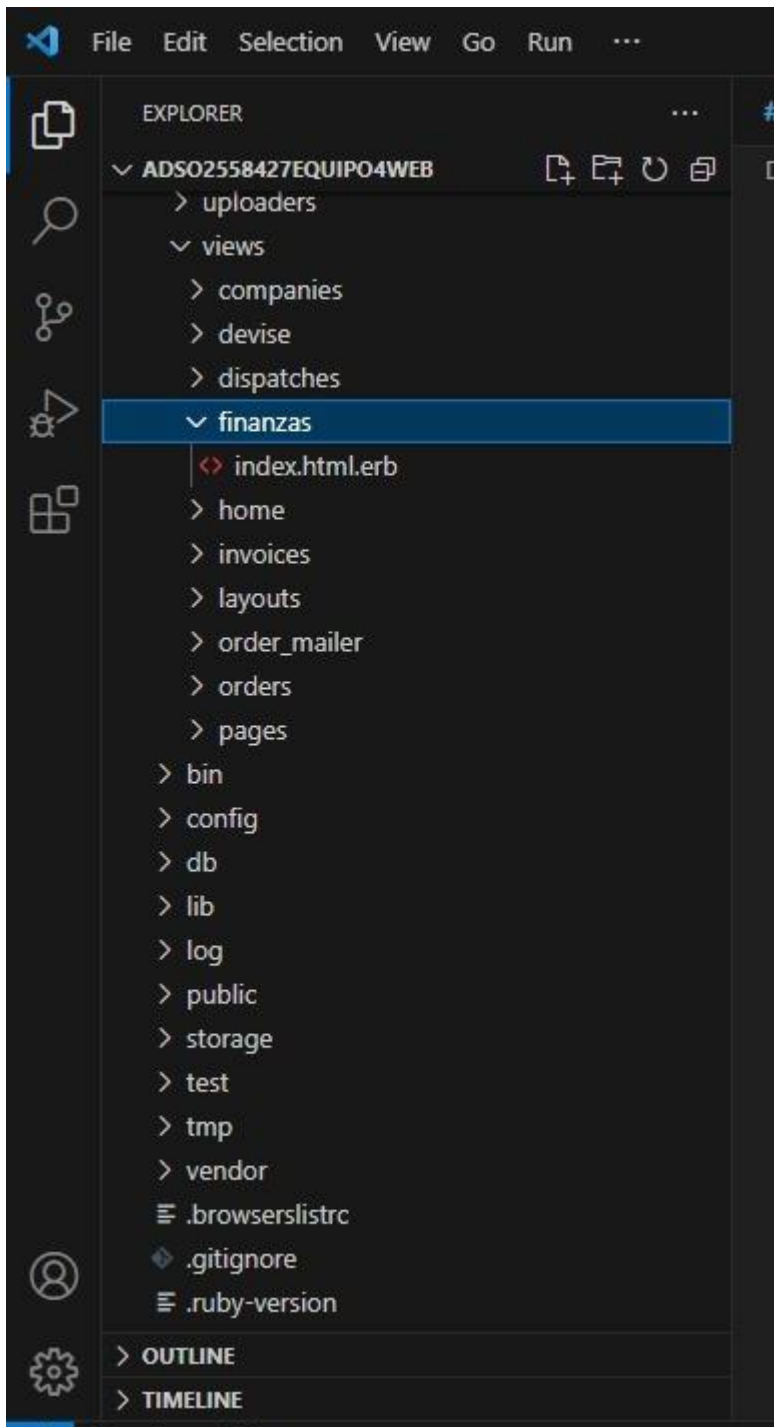
Vistas de company y user



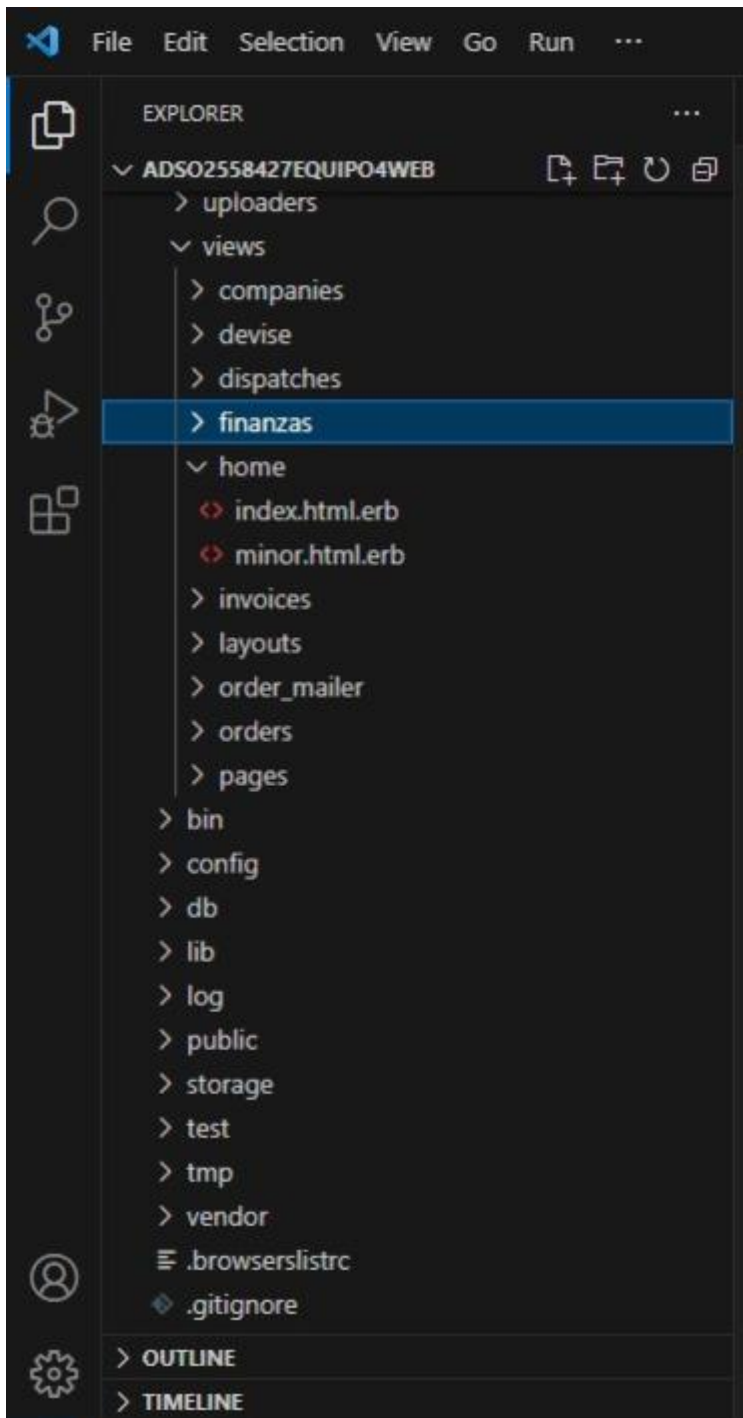
Devise



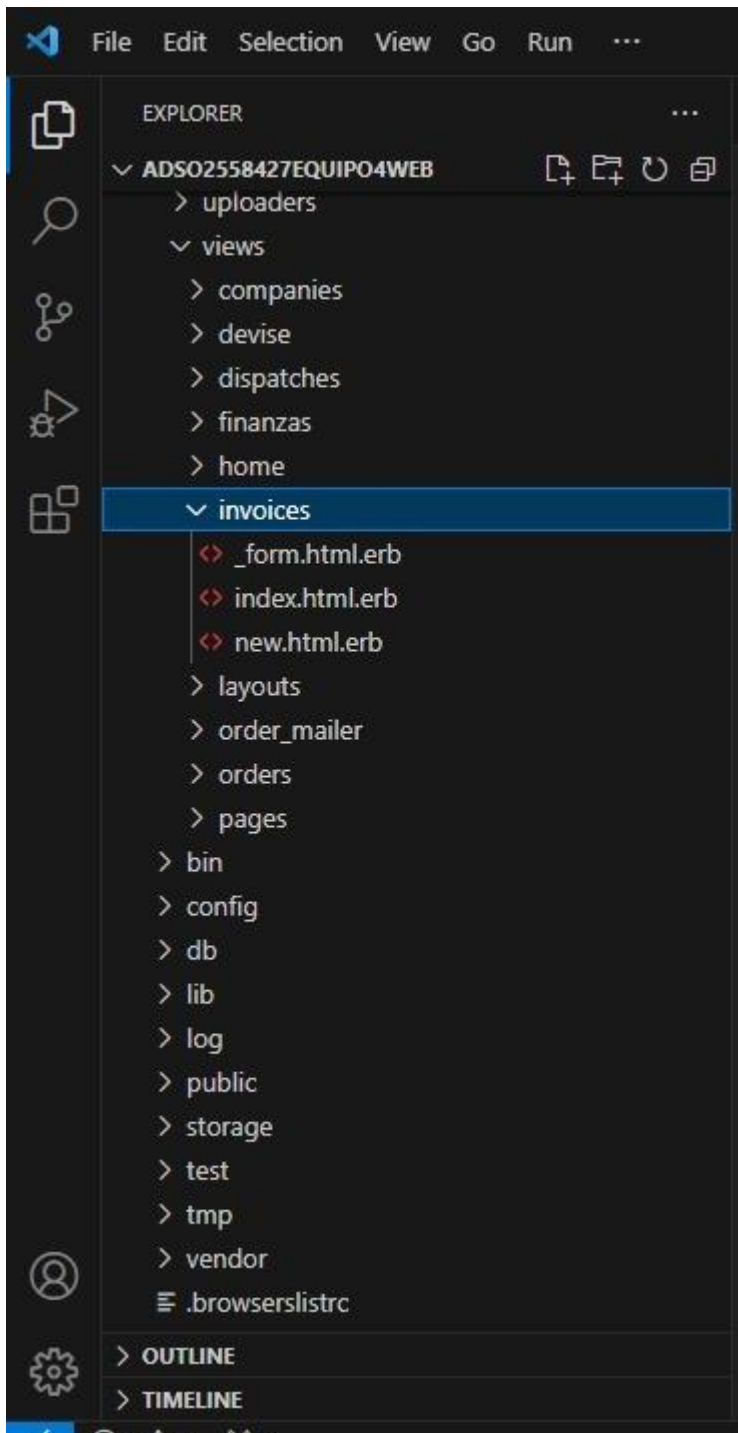
Dispatch



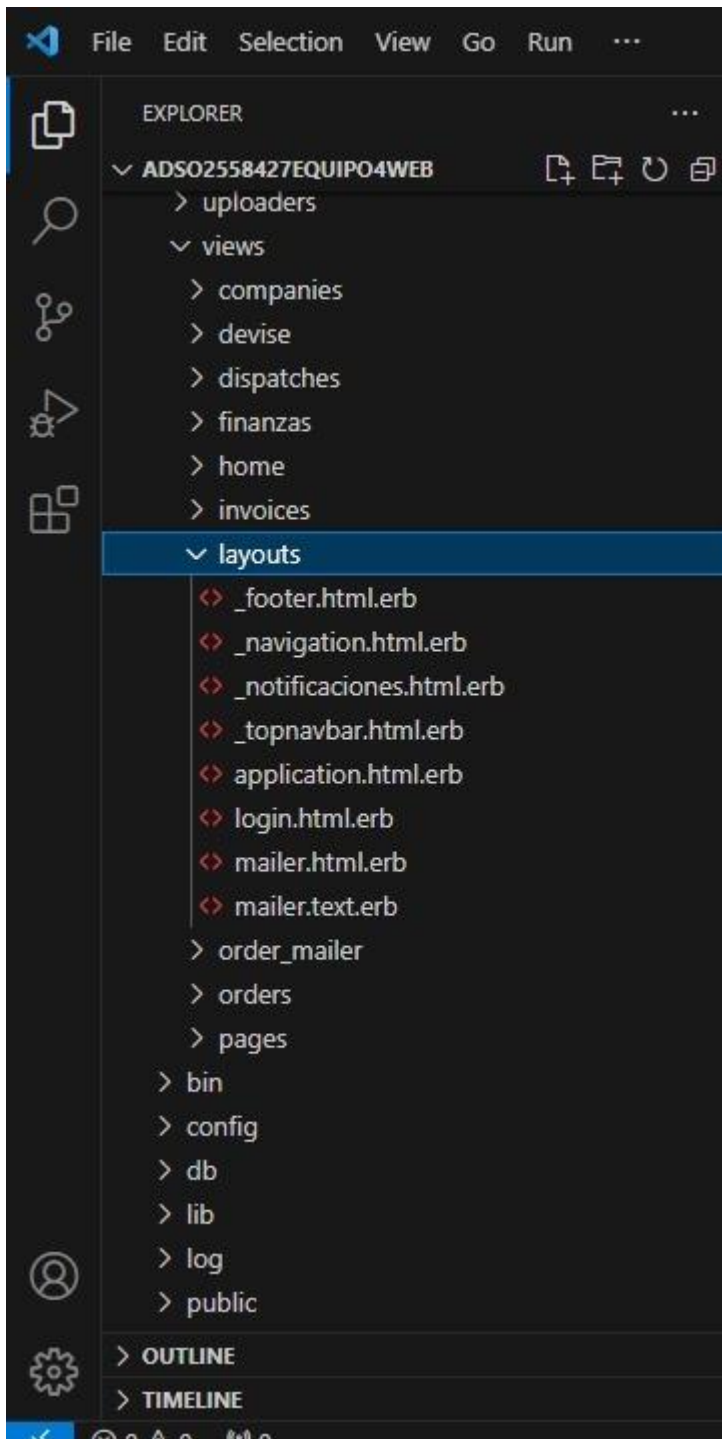
Finanzas



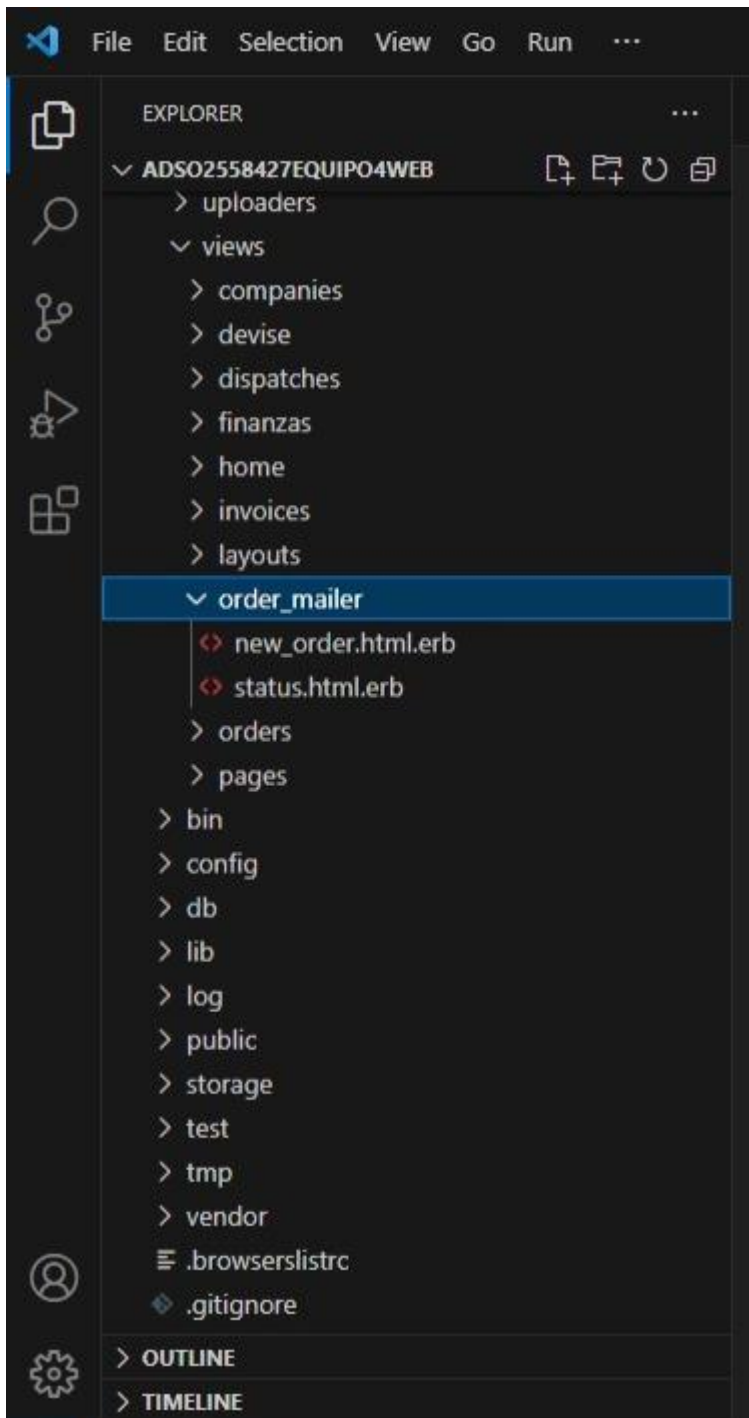
Home



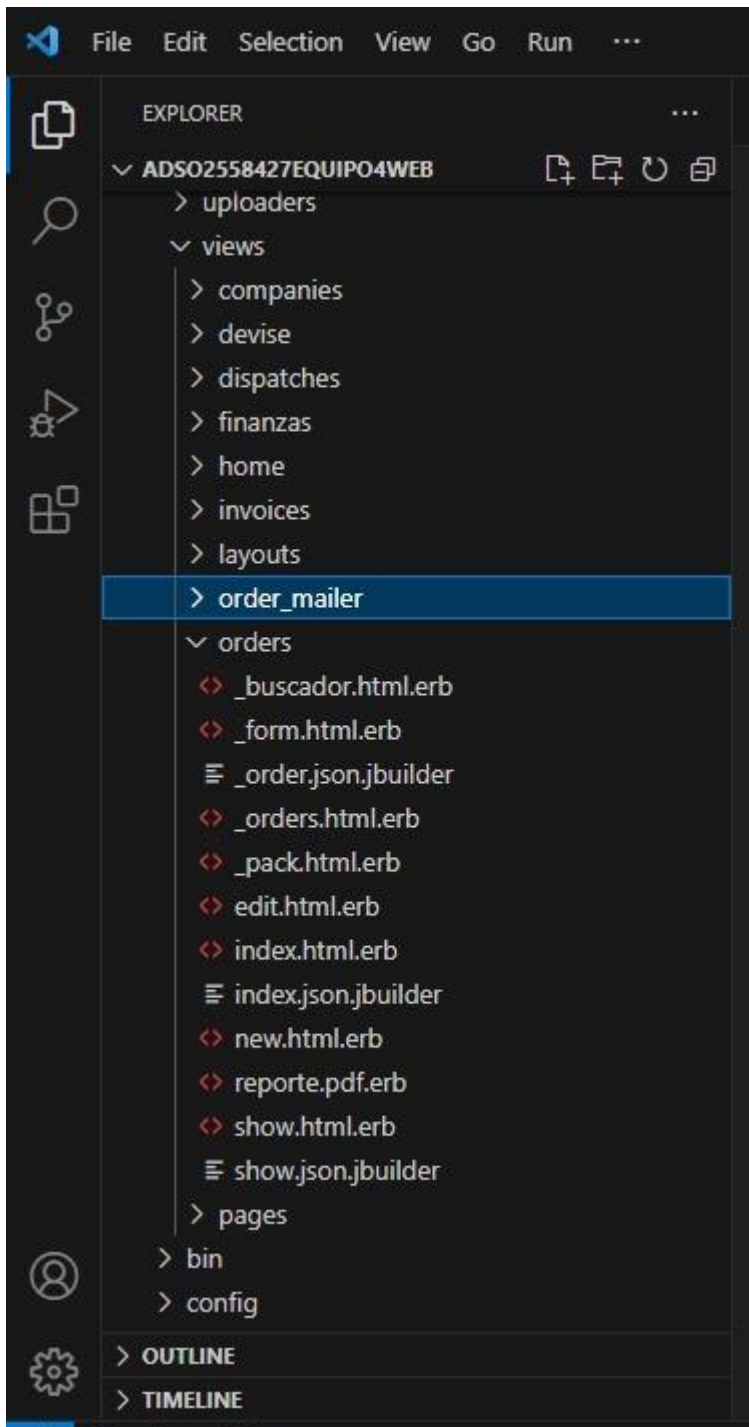
Invoice



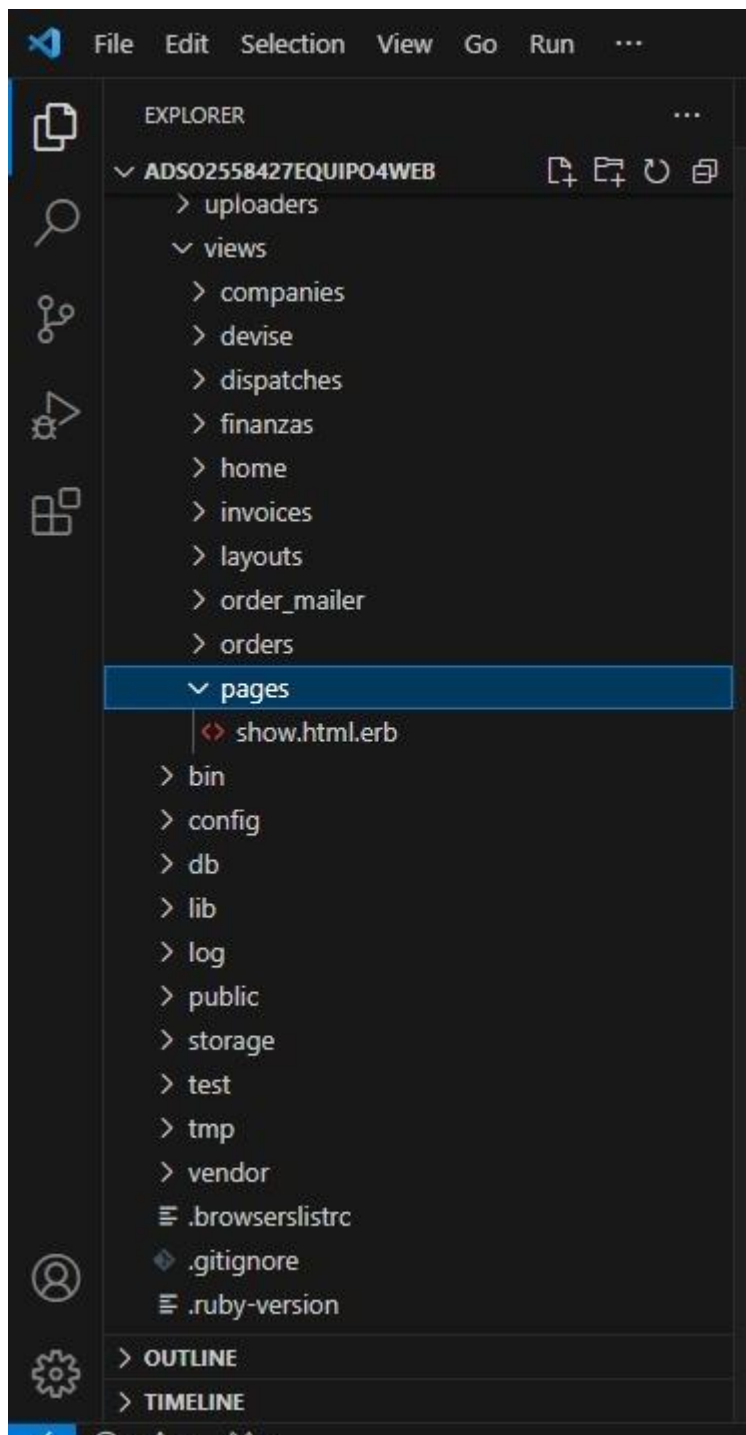
Layouts



Orden_mailer



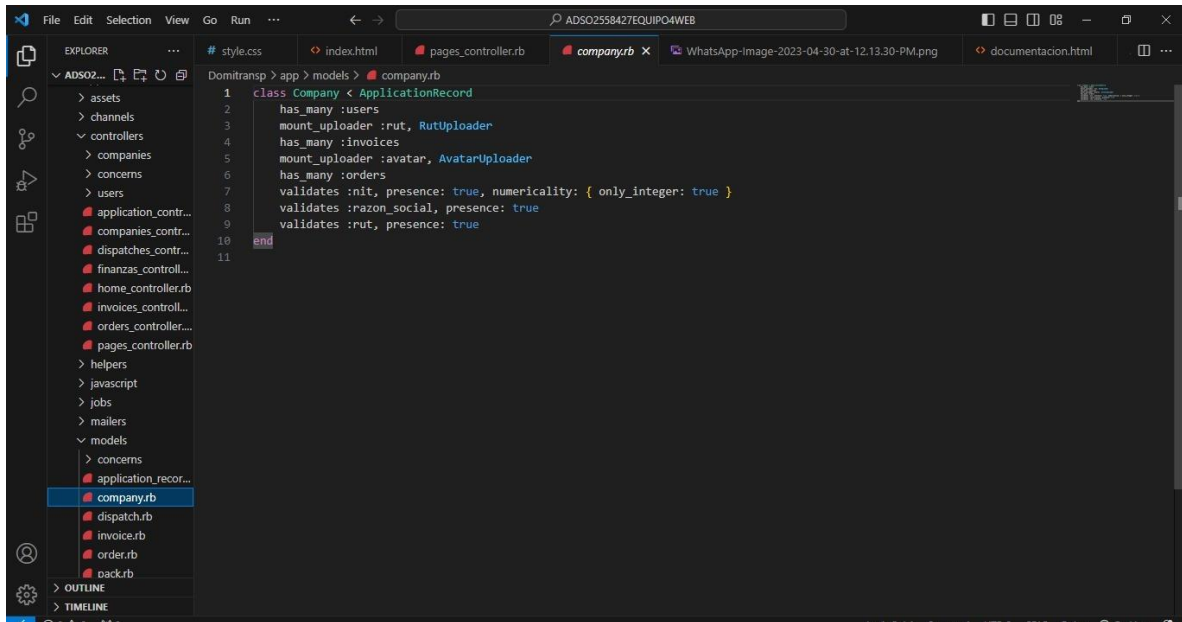
Order



Pages

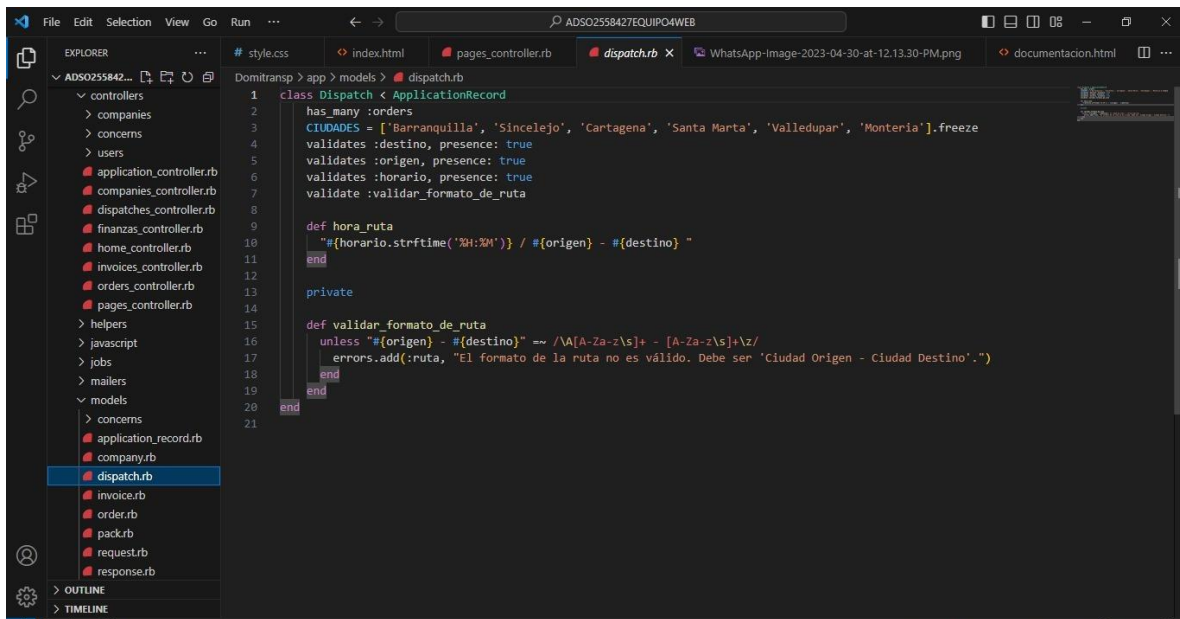
MODELOS:

Company



- **has_many:** Define las asociaciones de uno a muchos con otros modelos (en este caso, users, invoices y orders).
- **mount_uploader:** Se utiliza para montar un cargador de archivos (uploader) para las imágenes de RUT y avatar.
- **validates:** Establece validaciones para los atributos de la empresa, como la presencia del NIT, la razón social y el RUT, así como la restricción numérica para el NIT.

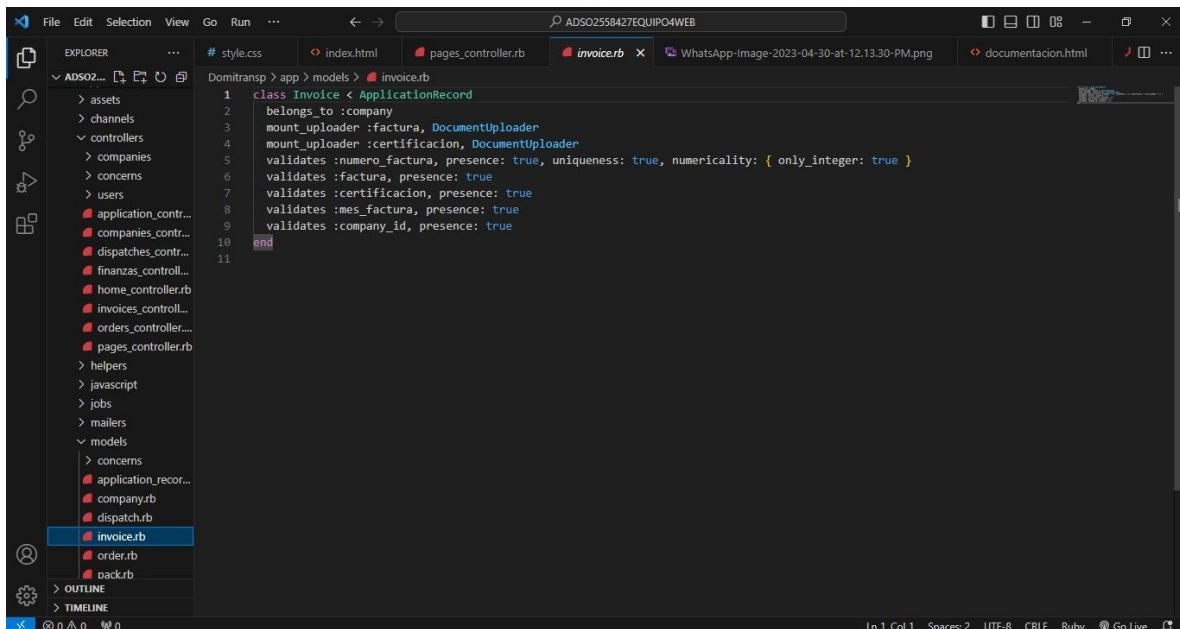
Dispatch



```
1 class Dispatch < ApplicationRecord
2   has_many :orders
3   CIUDADES = ['Barranquilla', 'Sinlejo', 'Cartagena', 'Santa Marta', 'Valledupar', 'Monteria'].freeze
4   validates :destino, presence: true
5   validates :origen, presence: true
6   validates :horario, presence: true
7   validate :validar_formato_de_ruta
8
9   def hora_ruta
10     "#{horario.strftime('%H:%M')} / #{origen} - #{destino} "
11   end
12
13   private
14
15   def validar_formato_de_ruta
16     unless "#{origen} - #{destino}" =~ /\A[A-Za-z\s]+ - [A-Za-z\s]+\Z/
17       errors.add(:ruta, "El formato de la ruta no es válido. Debe ser 'Ciudad Origen - Ciudad Destino'.")
18     end
19   end
20 end
```

Este modelo establece la estructura de datos y las validaciones asociadas con las rutas de envío . Además, define un método `hora_ruta` para obtener la hora y el formato de la ruta.

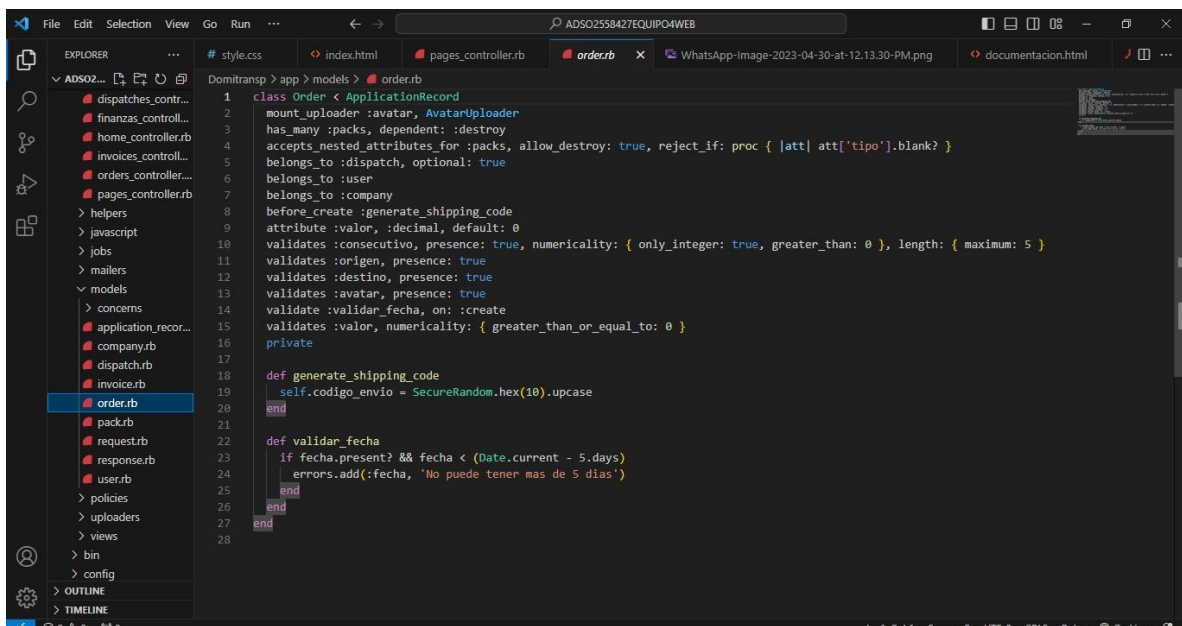
Invoice:



```
1 class Invoice < ApplicationRecord
2   belongs_to :company
3   mount_uploader :factura, DocumentUploader
4   mount_uploader :certificacion, DocumentUploader
5   validates :numero_factura, presence: true, uniqueness: true, numericality: { only_integer: true }
6   validates :factura, presence: true
7   validates :certificacion, presence: true
8   validates :mes_factura, presence: true
9   validates :company_id, presence: true
10 end
```

Este modelo establece la estructura de datos y las validaciones asociadas con las facturas en tu aplicación. Además, utiliza el montaje de cargadores de archivos para gestionar los documentos de factura y certificación

Order:

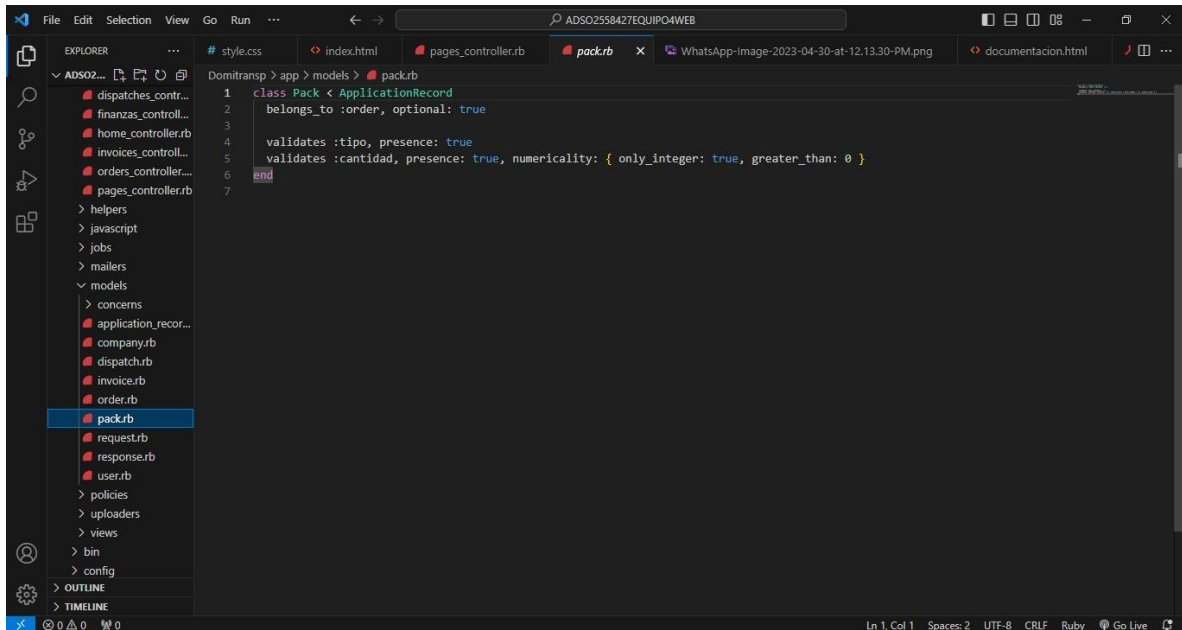


The screenshot shows a code editor with the following content:

```
1 class Order < ApplicationRecord
2   mount_uploader :avatar, AvatarUploader
3   has_many :packs, dependent: :destroy
4   accepts_nested_attributes_for :packs, allow_destroy: true, reject_if: proc { |att| att['tipo'].blank? }
5   belongs_to :dispatch, optional: true
6   belongs_to :user
7   belongs_to :company
8   before_create :generate_shipping_code
9   attribute :valor, :decimal, default: 0
10  validates :consecutivo, presence: true, numericality: { only_integer: true, greater_than: 0 }, length: { maximum: 5 }
11  validates :origen, presence: true
12  validates :destino, presence: true
13  validates :avatar, presence: true
14  validate :validar_fecha, on: :create
15  validates :valor, numericality: { greater_than_or_equal_to: 0 }
16  private
17
18  def generate_shipping_code
19    self.codigo_envio = SecureRandom.hex(10).upcase
20  end
21
22  def validar_fecha
23    if fecha.present? && fecha < (Date.current - 5.days)
24      errors.add(:fecha, 'No puede tener mas de 5 dias')
25    end
26  end
27 end
```

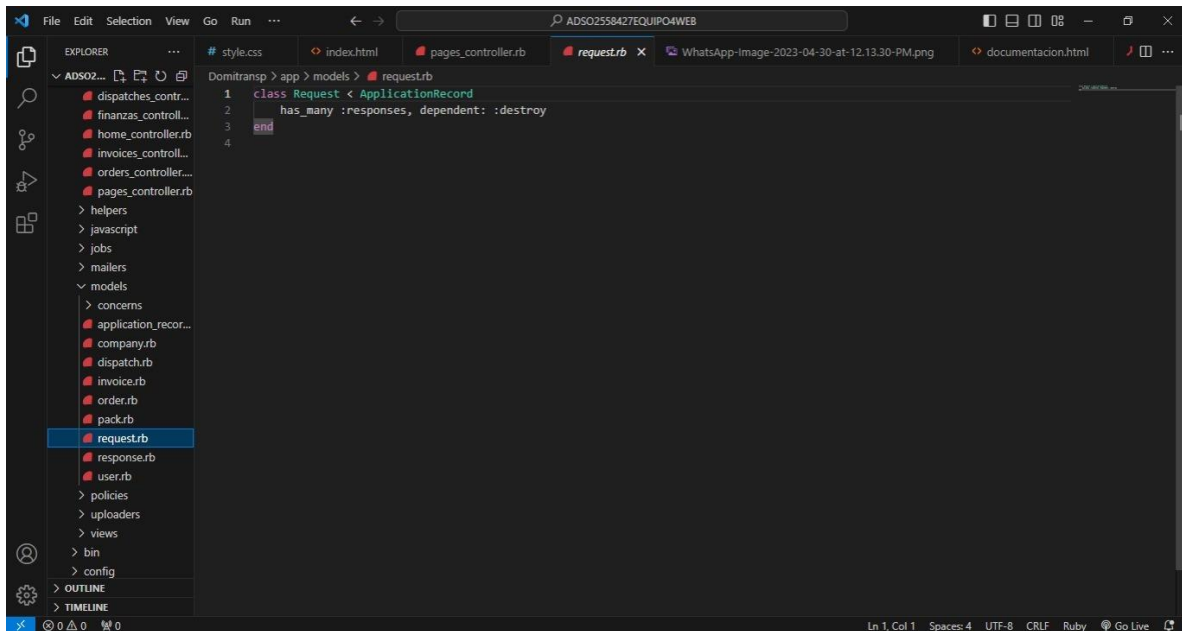
Este modelo establece la estructura de datos, las relaciones y las validaciones asociadas con las órdenes. También incluye un atributo virtual para el valor de la orden y un método callback para generar un código de envío único antes de crear la orden.

Pack:



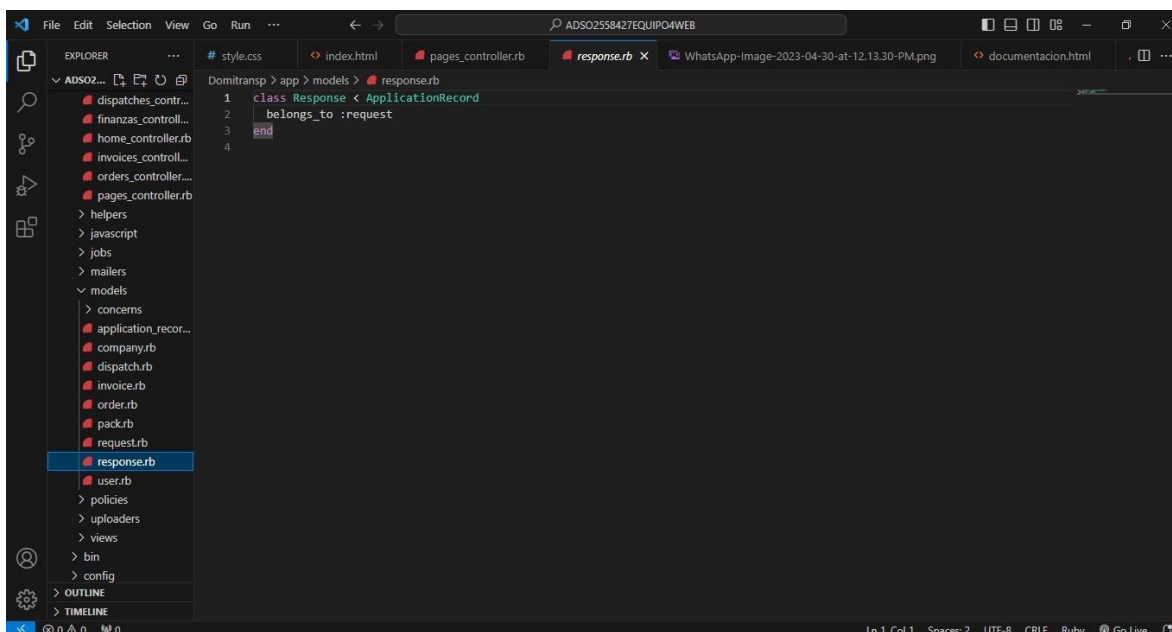
Este modelo establece la estructura de datos y las validaciones asociadas con los paquetes. Cada paquete pertenece a una orden y tiene un tipo y una cantidad asociados.

Request:



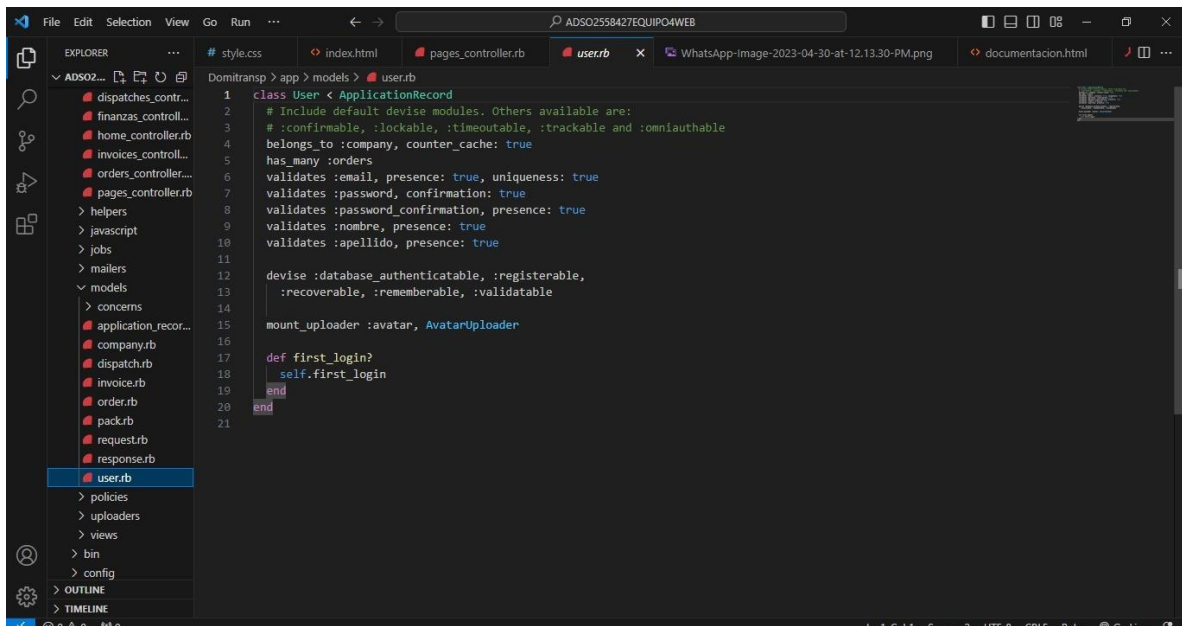
Este modelo establece que una solicitud puede tener múltiples respuestas (**has_many :responses**). La opción **dependent: :destroy** indica que, si se elimina una solicitud, también se deben eliminar todas sus respuestas asociadas.

Response:



Este diseño refleja una relación de uno a muchos, donde una solicitud puede tener varias respuestas, y cada respuesta está vinculada a una sola solicitud.

User:



Este modelo establece una relación de pertenencia a una empresa (**belongs_to :company**) y una relación de uno a muchos con las órdenes (**has_many :orders**). Además, utiliza el módulo Devise para la autenticación y la gestión del usuario. Las validaciones se han configurado para garantizar la integridad de los datos del usuario.

Schema de nuestro proyecto:

```
create_table "codes", force: :cascade do |t|
  t.string "code"
  t.bigint "user_id", null: false
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.index ["user_id"], name: "index_codes_on_user_id"
end
```

```
create_table "companies", force: :cascade do |t|
  t.string "nit"
  t.string "razon_social"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "rut"
  t.integer "users_count"
  t.string "avatar"
end
```

```
create_table "dispatches", force: :cascade do |t|
  t.time "horario"
  t.boolean "switch"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "origen"
  t.string "destino"
end
```

```
create_table "invoices", force: :cascade do |t|
  t.integer "numero_factura"
  t.string "mes_factura"
  t.string "factura"
```

→

e.css

<> index.html

pages_controller.rb

schema.rb X

WhatsApp-

transp > db > schema.rb

```
end

create_table "invoices", force: :cascade do |t|
  t.integer "numero_factura"
  t.string "mes_factura"
  t.string "factura"
  t.string "certificacion"
  t.bigint "company_id", null: false
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.index ["company_id"], name: "index_invoices_on_company_id"
end

create_table "orders", force: :cascade do |t|
  t.date "fecha"
  t.integer "consecutivo"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "avatar"
  t.string "destino"
  t.string "origen"
  t.string "estado", default: "En espera"
  t.string "codigo_envio"
  t.decimal "valor"
  t.bigint "dispatch_id"
  t.bigint "company_id", null: false
  t.bigint "user_id", null: false
  t.index ["company_id"], name: "index_orders_on_company_id"
  t.index ["dispatch_id"], name: "index_orders_on_dispatch_id"
  t.index ["user_id"], name: "index_orders_on_user_id"
end
```



```
→ ADSO2558427EQUIPO4WEB
e.css  <> index.html  pages_controller.rb  schema.rb x  WhatsApp
transp > db > schema.rb

  t.index ["user_id"], name: "index_orders_on_user_id"
end

create_table "packs", force: :cascade do |t|
  t.bigint "order_id", null: false
  t.string "tipo"
  t.string "cantidad"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.index ["order_id"], name: "index_packs_on_order_id"
end

create_table "requests", force: :cascade do |t|
  t.string "title"
  t.string "name"
  t.string "subject"
  t.string "email"
  t.string "code"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "estado", default: "En espera"
end

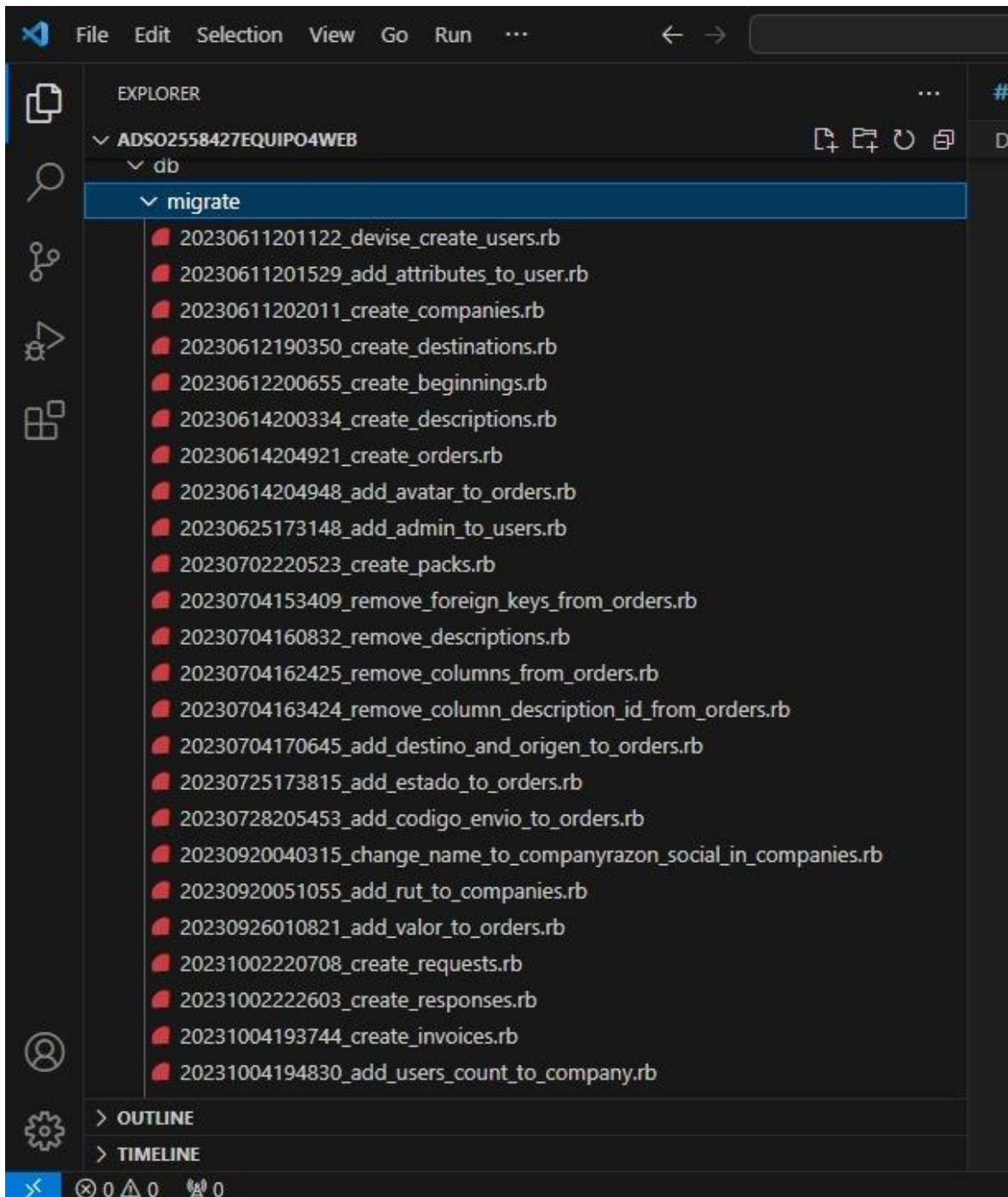
create_table "responses", force: :cascade do |t|
  t.text "content"
  t.bigint "request_id", null: false
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "rol", default: "U"
  t.index ["request_id"], name: "index_responses_on_request_id"
end
```

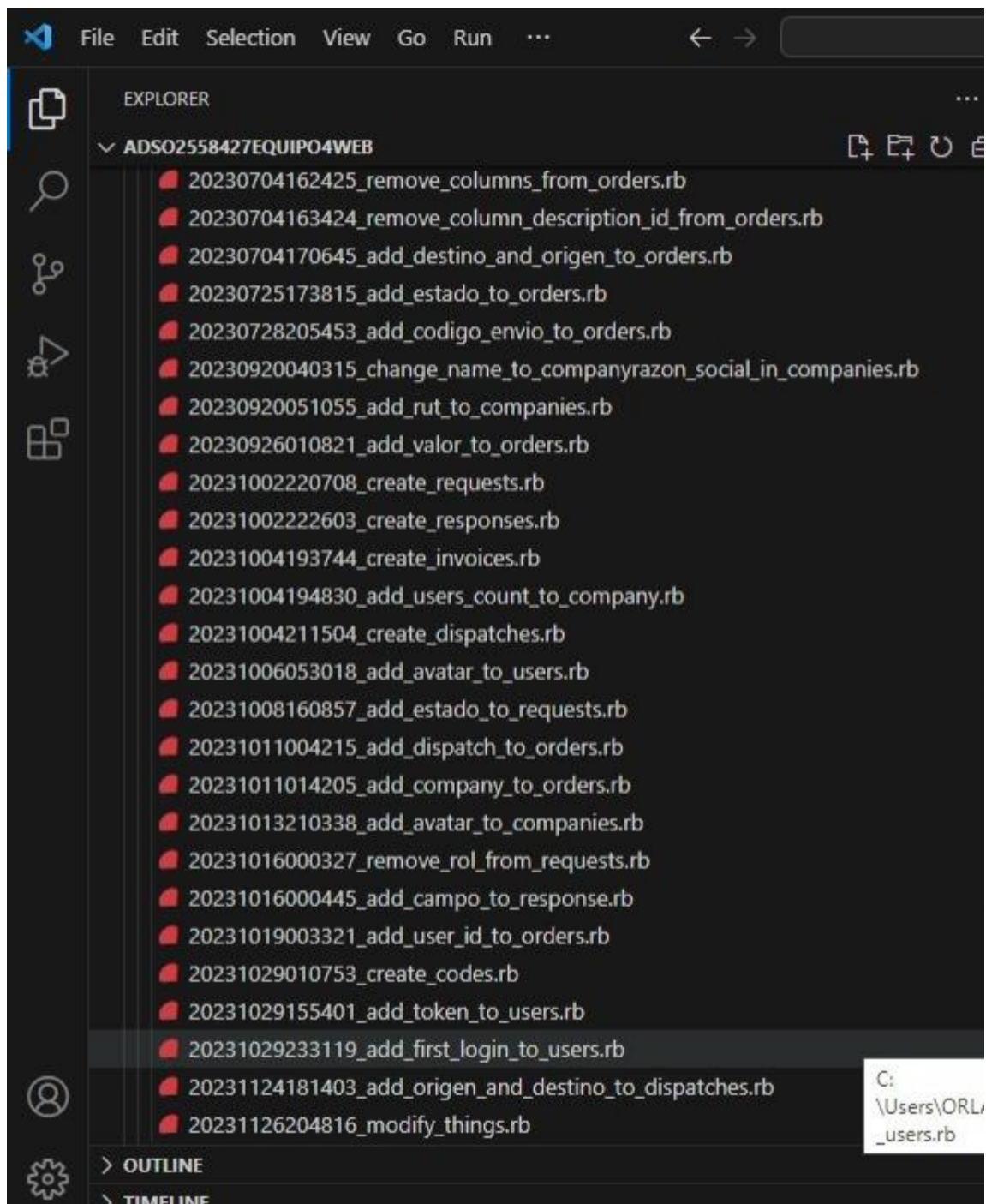
```
itransp > db > schema.rb
  t.string "rol", default: "U"
  t.index ["request_id"], name: "index_responses_on_request_id"
end

create_table "users", force: :cascade do |t|
  t.string "email", default: "", null: false
  t.string "encrypted_password", default: "", null: false
  t.string "reset_password_token"
  t.datetime "reset_password_sent_at"
  t.datetime "remember_created_at"
  t.bigint "company_id"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "nombre"
  t.string "apellido"
  t.boolean "admin"
  t.string "avatar"
  t.string "token"
  t.boolean "first_login"
  t.index ["company_id"], name: "index_users_on_company_id"
  t.index ["email"], name: "index_users_on_email", unique: true
  t.index ["reset_password_token"], name: "index_users_on_reset_password_token", unique: true
end

add_foreign_key "codes", "users"
add_foreign_key "invoices", "companies"
add_foreign_key "orders", "companies"
add_foreign_key "orders", "dispatches"
add_foreign_key "orders", "users"
add_foreign_key "packs", "orders"
add_foreign_key "responses", "requests"
end
```

Migraciones:





Gemas utilizadas:

... ← → 🔍 ADSO2558427EQUIPO

yle.css <> index.html 📄 pages_controller.rb 📄 Gemfile

itransp > 📄 Gemfile

```
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64_mingw,  
  
gem 'mimemagic', '0.3.10'  
gem "devise", "~> 4.9"  
  
gem 'simple_form'  
gem 'carrierwave', '>= 3.0.0.rc', '< 4.0'  
  
gem 'wicked_pdf'  
gem 'wkhtmltopdf-binary'  
  
gem 'yaml_db'  
  
gem 'pg'  
  
gem 'jquery-rails', '~> 4.4'  
  
gem 'webpacker', '~> 5.x'  
  
gem "pundit", "~> 2.3"  
  
gem 'axlsx'  
  
gem 'zip'  
  
gem 'i18n'  
  
gem 'aws-sdk-ses', '~> 1.6'  
gem 'toastr-rails'  
  
gem 'fog', :require => 'fog/aws'
```



```
le.css    index.html    pages_controller.rb    Gemfile    WhatsApp-Image-2023-04-30-at-12.13.30-PM.png    documenta
ransp > Gemfile

# Use Active Storage variant
# gem 'image_processing', '~> 1.2'

# Reduces boot times through caching; required in config/boot.rb
gem 'bootsnap', '>= 1.4.2', require: false

group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platforms: [:mri, :mingw, :x64_mingw]
end

group :development do
  # Access an interactive console on exception pages or by calling 'console' anywhere in the code.
  gem 'web-console', '>= 3.3.0'
  gem 'listen', '>= 3.0.5', '< 3.2'
  # Spring speeds up development by keeping your application running in the background. Read more: https://github.com/rails/spring
  gem 'spring'
  gem 'spring-watcher-listen', '~> 2.0.0'
end

group :test do
  # Adds support for Capybara system testing and selenium driver
  gem 'capybara', '>= 2.15'
  gem 'selenium-webdriver'
  # Easy installation and use of web drivers to run system tests with browsers
  gem 'webdrivers'
end

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64_mingw, :jruby]
```

Ln 1 Col 1 Spaces: 2 UTF-8 CRLF B