Informe de desarrollo de ESIFLIX

Departamento de Ingeniería Informática
Universidad de Cádiz
Curso 2018/2019

José Correro Barquín
Isabel del Pilar Durán Chumillas
Santiago Godoy Poce
José Manuel Periñán Freire
Fanny Chunyan Vela Díaz

Índice de contenido

1. Listas de requisitos priorizadas del producto (“product backlog”)

1.1 Arquitectura Modelo-Vista-Controlador (MVC)

ESIFLIX sigue una arquitectura denominada Modelo-Vista-Controlador.

El MVC es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de una evolución del modelo cliente-servidor.

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

1.2 . Historias de usuario

1.2.1 *Director*

- **Añadir director:** cuando el dueño de ESIFLIX descubra a un nuevo director cinematográfico, lo dará de alta en el sitio. Pulsará el botón “Añadir” para posteriormente escribir los datos del director en un formulario. Al guardar todos los datos añadidos estarán disponibles.
- **Listar directores**: la página mostrará un listado de todos los directores existentes.
- **Ver director:** se mostrarán en una página los detalles individuales del director seleccionado.
- **Editar director:** cambiaremos los datos del director en una página igual a la de añadir director.
- **Eliminar director**: al borrar el director todos sus datos desaparecerán del sistema.

1.2.2 Producers

- **Añadir productor:** cuando el dueño de ESIFLIX descubra a un nuevo productor cinematográfico, lo dará de alta en el sitio. Pulsará el botón “Añadir” para posteriormente escribir los datos del productor en un formulario. Al guardar todos los datos añadidos estarán disponibles.
- **Listar productores**: la página mostrará un listado de todos los productores existentes.

- **Ver productor:** se mostrarán en una página los detalles individuales del productor seleccionado.
- **Editar productor:** cambiaremos los datos del productor en una página igual a la de añadir productor.
- **Eliminar productor**: al borrar el productor todos sus datos desaparecerán del sistema.

1.2.3 Films

- **Añadir película:** cuando el dueño de ESIFLIX descubra a una nueva película, la dará de alta en el sitio. Pulsará el botón "Añadir" para posteriormente escribir los datos de la película en un formulario. Al guardar todos los datos añadidos estarán disponibles.
- **Listar películas**: la página mostrará un listado de todas las películas existentes.
- **Ver película:** se mostrarán en una página los detalles individuales de la película seleccionada.
- **Editar película:** cambiaremos los datos del productor en una página igual a la de añadir productor.
- **Eliminar película**: al borrar la película todos sus datos desaparecerán del sistema.

1.2.4 Catalog

- **Ver catálogo:** El usuario podrá ver todas las películas publicadas y que pueden ser compradas en ESIFLIX en el catálogo. Esta información deberá estar disponible en todo momento.
- **Mostrar película del catálogo**: El usuario podrá acceder a la información detallada de la película seleccionada.
- **Buscar en el catálogo**: El usuario debe de ser capaz de realizar una búsqueda de películas en el catálogo, tanto con el nombre completo como con parte de este.

1.2.5 Cart y CartItem

- Un cliente será capaz de introducir todos los productos que desee de la tienda en el carrito de la compra.
- Un cliente podrá vaciar el carro o eliminar un artículo individual del mismo.
- Estas acciones se realizarán de forma intuitiva pulsando sobre los botones correspondientes.

1.2.6 User y UserSession

- **Inicio de sesión exitoso:** cuando un usuario que no ha accedido quiere iniciar sesión, el sistema le dirigirá a la página de inicio de sesión. Si el usuario y contraseña proporcionados son correctos, será un inicio de sesión exitoso y se redirigirá al usuario.

- **Inicio de sesión incorrecto**: cuando un usuario que no ha accedido quiere iniciar sesión, el sistema le dirigirá a la página de inicio de sesión. Si el usuario y contraseña proporcionados no son correctos, el sistema redirigirá de nuevo al usuario al formulario con un mensaje de error.2. Planificación del proyecto

1.2.7 Order y OrderItem

- **Facturar**: una vez que el cliente ha terminado de añadir artículos al carrito de la compra, puede proceder a la página de facturación, donde teclea sus datos de contacto, la dirección de envío y la información de la tarjeta de crédito; a continuación se emite el pedido, lo que da inicio a un ujo de procesamiento de pedido que incluye cobrar al cliente y enviar los artículos.
- **Ver pedidos:** el administrador debe poder ver el estado de todos los pedidos, procesados y cerrados. Los pedidos procesados son los que han sido cobrados al cliente pero aún no se han enviado; los pedidos cerrados son los que ya han sido enviados al cliente.
- **Ver pedido:** Antes de proceder al envío, el administrador debe poder ver los detalles de un pedido; se debe añadir una página que muestre la dirección de envío la información de facturación junto con la información de contacto del cliente.
- **Cerrar pedido:** Tras enviar el pedido, éste debe cerrarse; esto se llevará a cabo en la página de detalles del pedido, de forma que con un simple botón se pueda cambiar el estado del pedido a cerrado.
- **Borrar pedidos:** el administrador debe poder borrar los pedidos.

1.2.8 Foro de discusión (ForumPost)

- **Ver Foros:** Los clientes de la tienda podrán hacer uso de un foro, en el que podrán explicar su experiencia con el producto adquirido o formular preguntas y/o quejas con el tema que consideren oportuno.
- **Crear post:** Cualquier cliente podrá iniciar un post o tema en el foro. Para ello tendrá que hacer clic sobre la opción Foro del menú principal y posteriormente hacer clic en el enlace 'Nuevo post'. Se mostrará un breve formulario en el que el cliente introducirá su nombre, una descripción del tema y un desarrollo del mismo. Una vez enviado el formulario y validados todos los campos, se procederá a la publicación del mensaje en el foro y otros usuarios podrán interactuar con él.
- **Borrar post**: Un post solo podrá ser eliminado por el administrador de la tienda on-line. El administrador de la tienda se identificará y podrá eliminar un post o todos los que desee.
- **Responder a un post**: Los clientes podrán responder a los temas abiertos por otros usuarios. Para ello, harán clic sobre el enlace Responder que aparece en la parte inferior izquierda de la página. El cliente rellenará el formulario que aparecerá a continuación con su nombre, con una descripción breve sobre el contenido de su

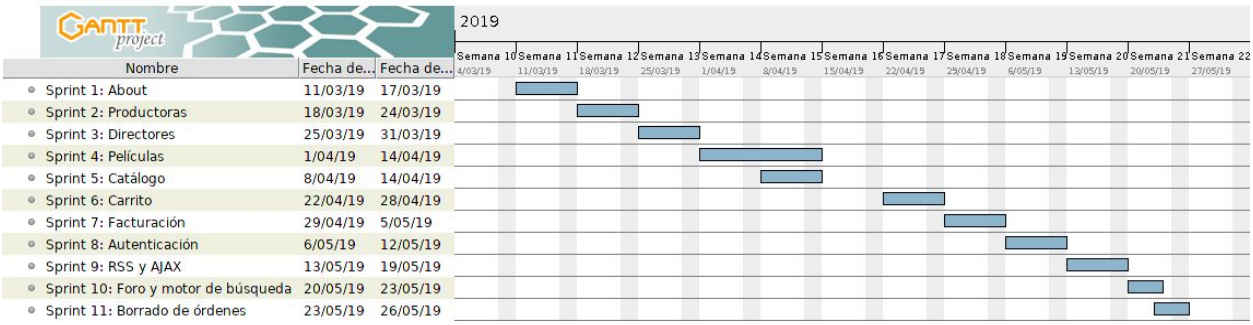
respuesta y finalmente una respuesta más extensa. Al enviar el formulario la respuesta se almacenará y aparecerá debajo del post inicial en el orden que le corresponda.

2. Planificación del proyecto

2.1. Introducción

De acuerdo con SCRUM (metodología de desarrollo utilizada en este proyecto), hemos dividido las diferentes etapas del proyecto en *sprints,* que realizaremos semanalmente.

2.2. Calendario

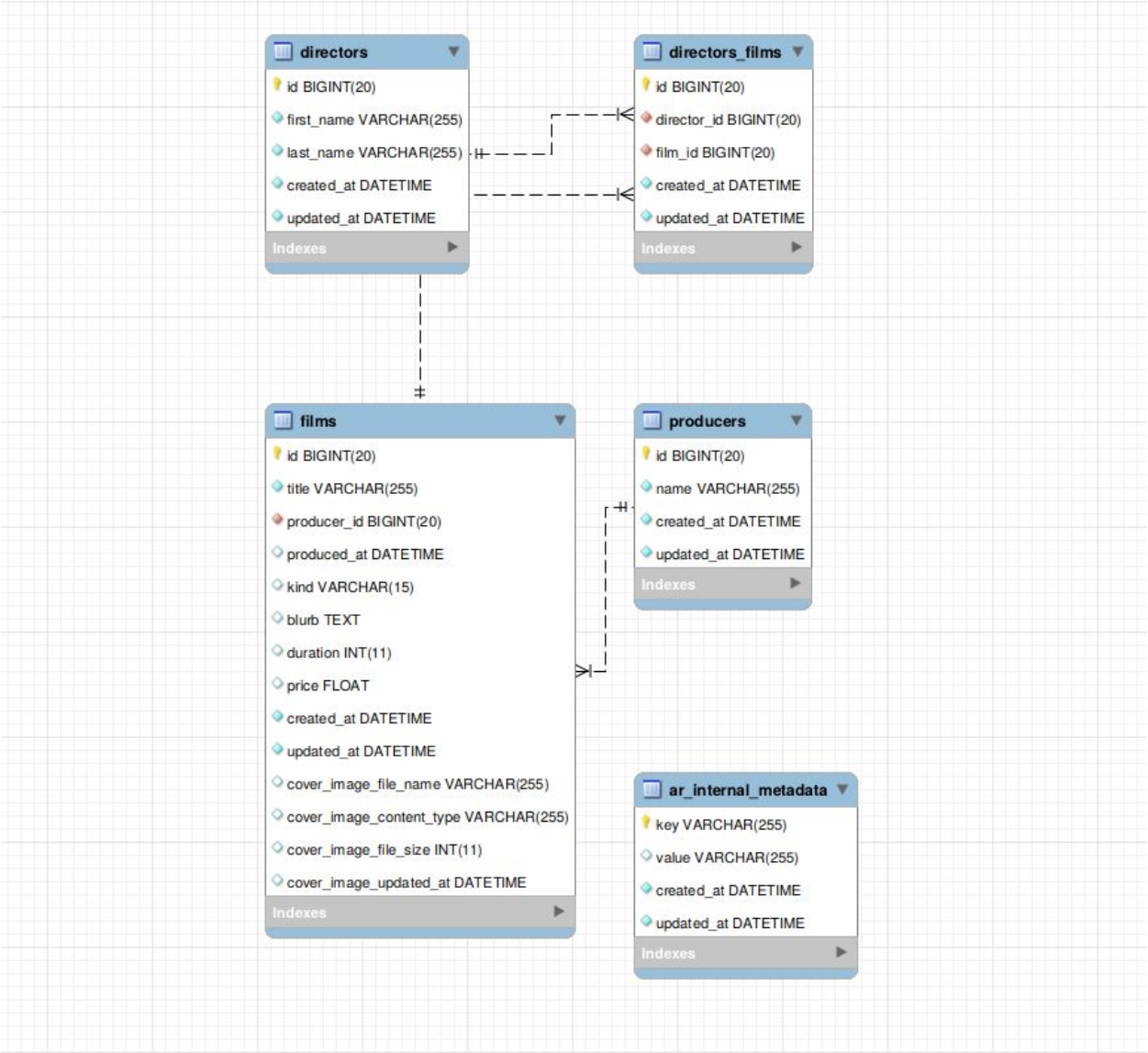


Nombre	Fecha de...	Fecha de...
Sprint 1: About	11/03/19	17/03/19
Sprint 2: Productoras	18/03/19	24/03/19
Sprint 3: Directores	25/03/19	31/03/19
Sprint 4: Películas	1/04/19	14/04/19
Sprint 5: Catálogo	8/04/19	14/04/19
Sprint 6: Carrito	22/04/19	28/04/19
Sprint 7: Facturación	29/04/19	5/05/19
Sprint 8: Autenticación	6/05/19	12/05/19
Sprint 9: RSS y AJAX	13/05/19	19/05/19
Sprint 10: Foro y motor de búsqueda	20/05/19	23/05/19
Sprint 11: Borrado de órdenes	23/05/19	26/05/19

3. Esquema de la base de datos

Vamos a observar la evolución del diagrama E/R de nuestra base de datos a lo largo del curso. Observaremos las nuevas relaciones con las distintas entidades a medida que la tienda online crece.

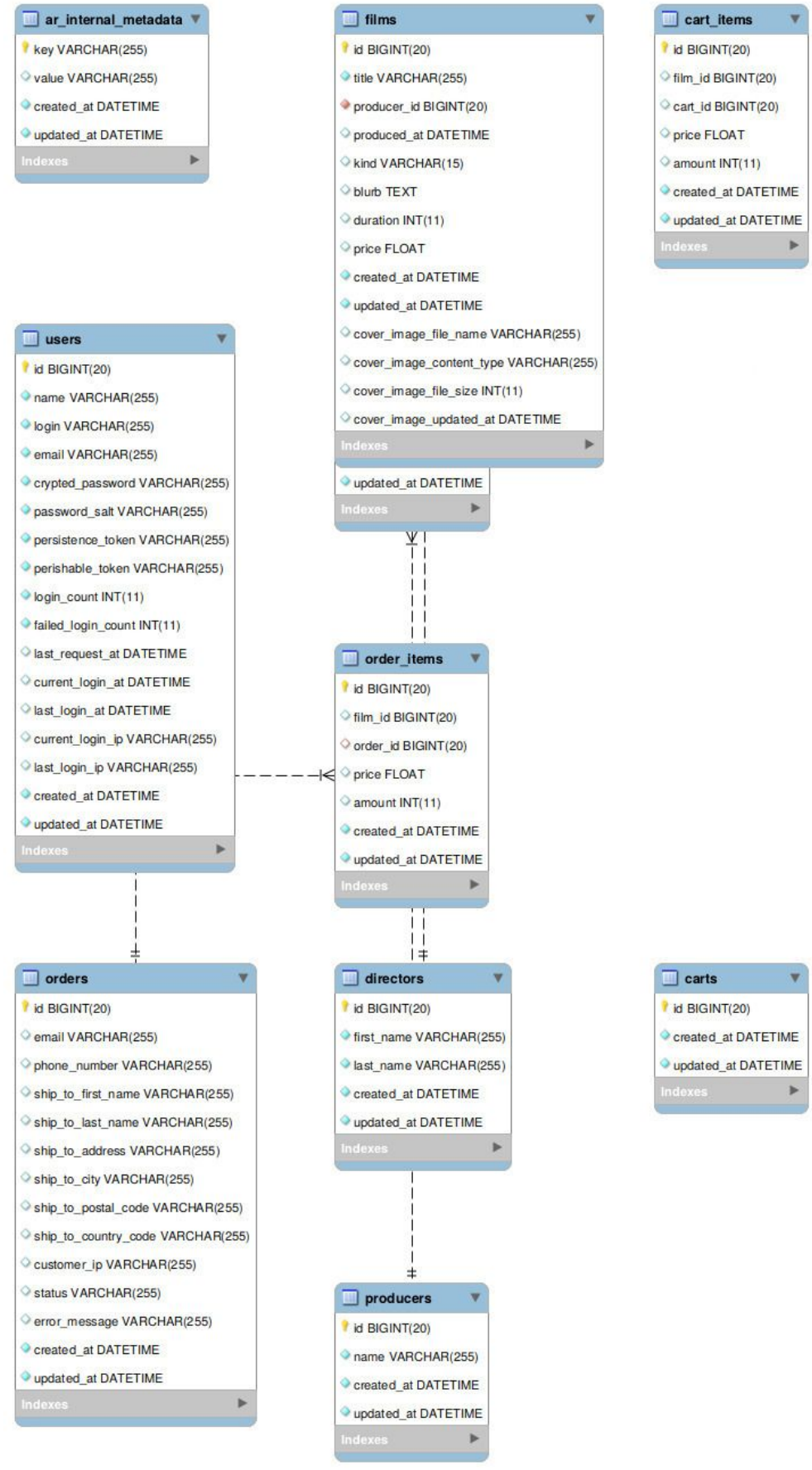
3.1 Diagrama E/R inicial



Es el esquema de la Base de Datos con la que trabajamos en las primeras fases del desarrollo de ESIFLIX. Tendremos la entidad principal, **films**, dos entidades secundarias **directors** y **producers** y la entidad relacion **directors_films**.

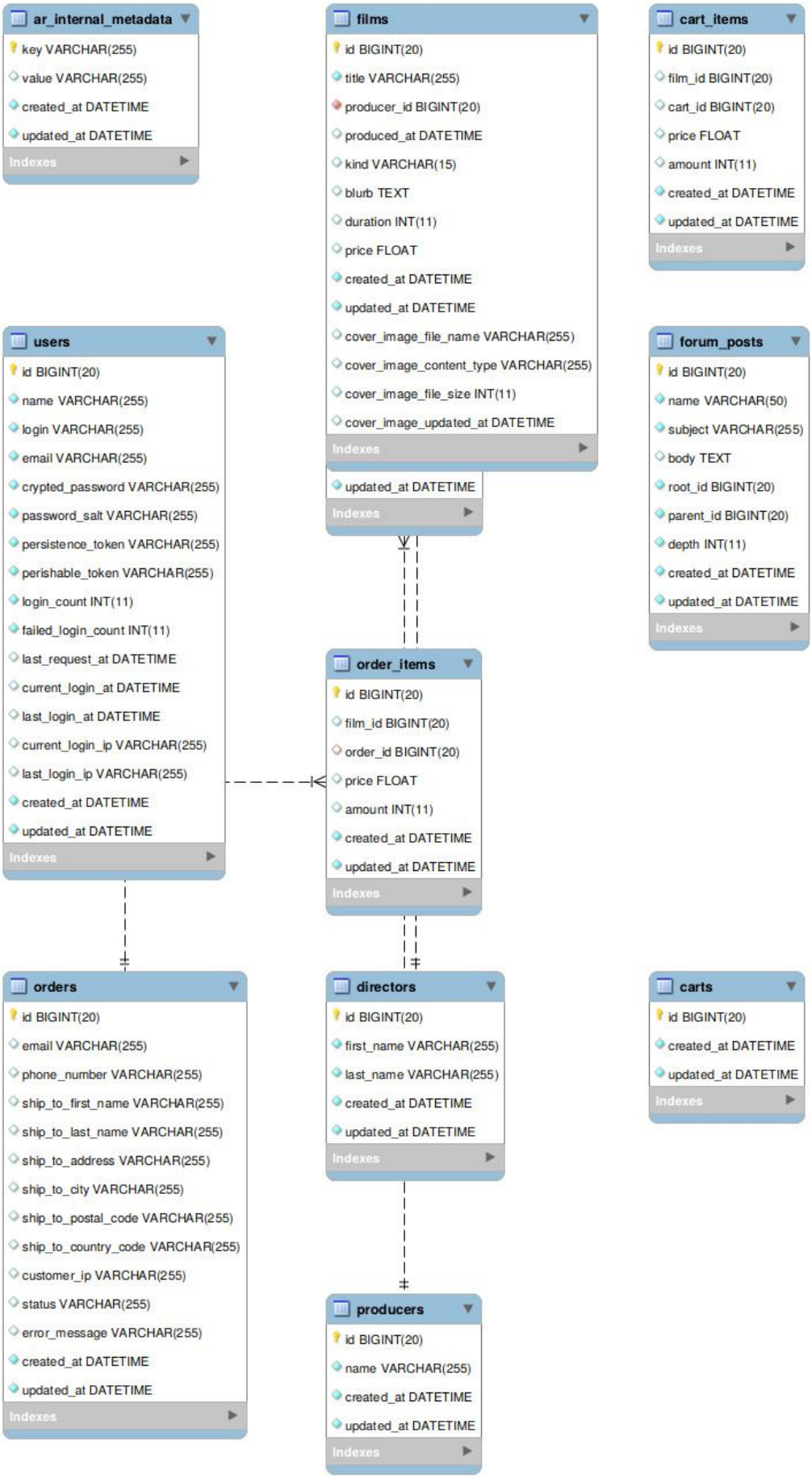
3.2. Diagrama E/R con funcionalidad básica

Tras añadir las funcionalidades básicas obtendremos el siguiente esquema:


ar_internal_metadata
key VARCHAR(255)
value VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
films
id BIGINT(20)
title VARCHAR(255)
producer_id BIGINT(20)
produced_at DATETIME
kind VARCHAR(15)
blurb TEXT
duration INT(11)
price FLOAT
created_at DATETIME
updated_at DATETIME
cover_image_file_name VARCHAR(255)
cover_image_content_type VARCHAR(255)
cover_image_file_size INT(11)
cover_image_updated_at DATETIME
Indexes
cart_items
id BIGINT(20)
film_id BIGINT(20)
cart_id BIGINT(20)
price FLOAT
amount INT(11)
created_at DATETIME
updated_at DATETIME
Indexes
users
id BIGINT(20)
name VARCHAR(255)
login VARCHAR(255)
email VARCHAR(255)
crypted_password VARCHAR(255)
password_salt VARCHAR(255)
persistence_token VARCHAR(255)
perishable_token VARCHAR(255)
login_count INT(11)
failed_login_count INT(11)
last_request_at DATETIME
current_login_at DATETIME
last_login_at DATETIME
current_login_ip VARCHAR(255)
last_login_ip VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
updated_at DATETIME
Indexes
order_items
id BIGINT(20)
film_id BIGINT(20)
order_id BIGINT(20)
price FLOAT
amount INT(11)
created_at DATETIME
updated_at DATETIME
Indexes
orders
id BIGINT(20)
email VARCHAR(255)
phone_number VARCHAR(255)
ship_to_first_name VARCHAR(255)
ship_to_last_name VARCHAR(255)
ship_to_address VARCHAR(255)
ship_to_city VARCHAR(255)
ship_to_postal_code VARCHAR(255)
ship_to_country_code VARCHAR(255)
customer_ip VARCHAR(255)
status VARCHAR(255)
error_message VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
directors
id BIGINT(20)
first_name VARCHAR(255)
last_name VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
carts
id BIGINT(20)
created_at DATETIME
updated_at DATETIME
Indexes
producers
id BIGINT(20)
name VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes

3.3 Diagrama E/R final

Tras añadir las funcionalidades opcionales tenemos el siguiente esquema:


ar_internal_metadata
key VARCHAR(255)
value VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
films
id BIGINT(20)
title VARCHAR(255)
producer_id BIGINT(20)
produced_at DATETIME
kind VARCHAR(15)
blurb TEXT
duration INT(11)
price FLOAT
created_at DATETIME
updated_at DATETIME
cover_image_file_name VARCHAR(255)
cover_image_content_type VARCHAR(255)
cover_image_file_size INT(11)
cover_image_updated_at DATETIME
Indexes
cart_items
id BIGINT(20)
film_id BIGINT(20)
cart_id BIGINT(20)
price FLOAT
amount INT(11)
created_at DATETIME
updated_at DATETIME
Indexes
users
id BIGINT(20)
name VARCHAR(255)
login VARCHAR(255)
email VARCHAR(255)
crypted_password VARCHAR(255)
password_salt VARCHAR(255)
persistence_token VARCHAR(255)
perishable_token VARCHAR(255)
login_count INT(11)
failed_login_count INT(11)
last_request_at DATETIME
current_login_at DATETIME
last_login_at DATETIME
current_login_ip VARCHAR(255)
last_login_ip VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
forum_posts
id BIGINT(20)
name VARCHAR(50)
subject VARCHAR(255)
body TEXT
root_id BIGINT(20)
parent_id BIGINT(20)
depth INT(11)
created_at DATETIME
updated_at DATETIME
Indexes
order_items
id BIGINT(20)
film_id BIGINT(20)
order_id BIGINT(20)
price FLOAT
amount INT(11)
created_at DATETIME
updated_at DATETIME
Indexes
orders
id BIGINT(20)
email VARCHAR(255)
phone_number VARCHAR(255)
ship_to_first_name VARCHAR(255)
ship_to_last_name VARCHAR(255)
ship_to_address VARCHAR(255)
ship_to_city VARCHAR(255)
ship_to_postal_code VARCHAR(255)
ship_to_country_code VARCHAR(255)
customer_ip VARCHAR(255)
status VARCHAR(255)
error_message VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
directors
id BIGINT(20)
first_name VARCHAR(255)
last_name VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes
carts
id BIGINT(20)
created_at DATETIME
updated_at DATETIME
Indexes
producers
id BIGINT(20)
name VARCHAR(255)
created_at DATETIME
updated_at DATETIME
Indexes

4. Contenido del fichero de rutas

/config/routes.rb

```
Rails.application.routes.draw do

  root :to => 'catalog#index'

  get 'about' => 'about#index'
  get 'forum' => 'forum#index'
  get 'checkout' => 'checkout#index'
  get 'admin/director' => 'admin/director#index'
  get 'admin/producer' => 'admin/producer#index'
  get 'admin/film' => 'admin/film#index'
  get 'admin/order' => 'admin/order#index'

  get 'about/index'

  get 'admin/director/new'
  post 'admin/director/create'
  get 'admin/director/edit'
  post 'admin/director/update'
  post 'admin/director/destroy'
  get 'admin/director/show'
  get 'admin/director/show/:id' => 'admin/director#show'
  get 'admin/director/index'

  get 'admin/producer/new'
  post 'admin/producer/create'
  get 'admin/producer/edit'
  post 'admin/producer/update'
  post 'admin/producer/destroy'
  get 'admin/producer/show'
  get 'admin/producer/show/:id' => 'admin/producer#show'
  get 'admin/producer/index'

  get 'admin/film/new'
  post 'admin/film/create'
  get 'admin/film/edit'
  post 'admin/film/update'
  post 'admin/film/destroy'
  get 'admin/film/show'
  get 'admin/film/show/:id' => 'admin/film#show'
```

```
  get 'admin/film/index'

  post 'admin/order/close'
  post 'admin/order/destroy'
  get 'admin/order/show'
  get 'admin/order/show/:id' => 'admin/order#show'
  get 'admin/order/index'

  get 'catalog/show'
  get 'catalog/show/:id' => 'catalog#show'
  get 'catalog/index'
  get 'catalog/latest'
  get 'catalog/search'
  get 'catalog/rss'

  get 'cart/add'
  post 'cart/add'
  get 'cart/remove'
  post 'cart/remove'
  get 'cart/clear'
  post 'cart/clear'

  get 'user_sessions/new'
  get 'user_sessions/create' # for showing failed login screen after
restarting web server
  post 'user_sessions/create'
  get 'user_sessions/destroy'

  get 'checkout/index'
  post 'checkout/submit_order'
  get 'checkout/thank_you'

  get 'user/new'
  post 'user/create'
  get 'user/show'
  get 'user/show/:id' => 'user#show'
  get 'user/edit'
  post 'user/update'

  get 'forum/post'
  post 'forum/create'
  get 'forum/reply'
  get 'forum/destroy'
  post 'forum/destroy'
  get 'forum/show'
  get 'forum/index'
```

```
  get 'password_reset/new'
  post 'password_reset/create'
  get 'password_reset/edit'
  post 'password_reset/update'

  # For details on the DSL available within this file, see
http://guides.rubyonrails.org/routing.html
end
```

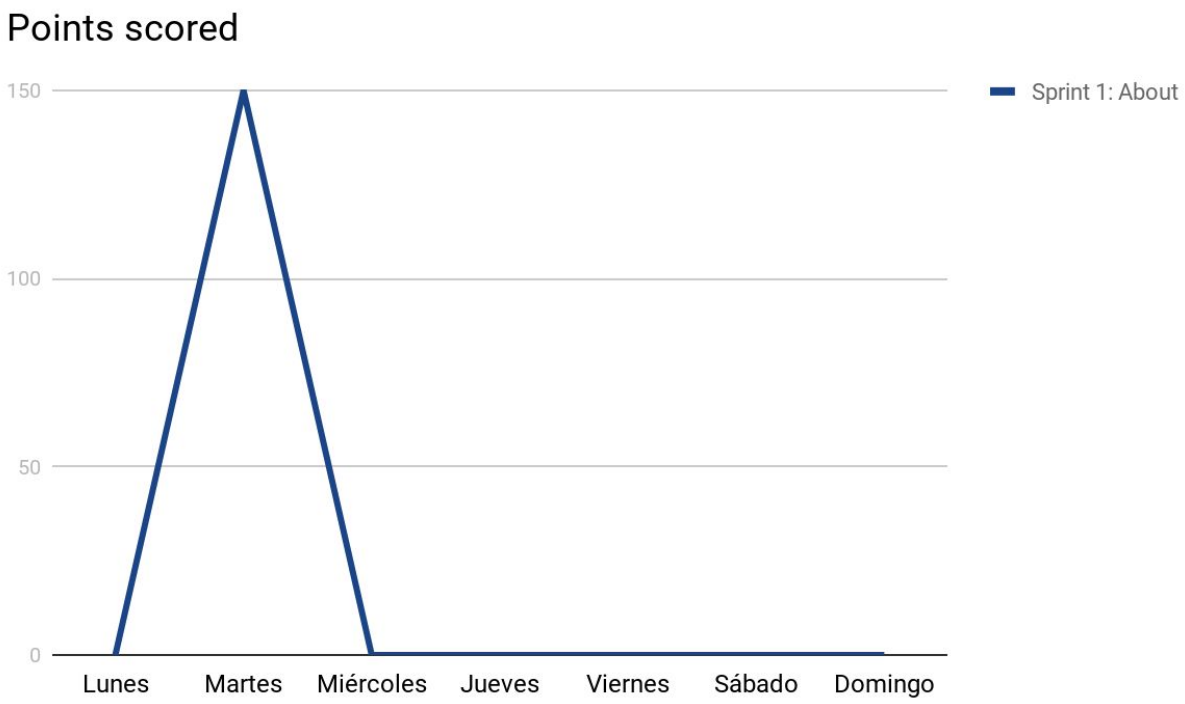
5. Sprints

5.1. Sprint 1

En este sprint nos familiarizamos con *Ruby on Rails* generando la funcionalidad *about* de nuestra tienda online.

5.1.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Vista	0	30	0	0	0	0	0
Controlador	0	30	0	0	0	0	0
Base de datos	0	60	0	0	0	0	0
Layout	0	30	0	0	0	0	0
TOTAL	0	150	0	0	0	0	0



5.1.2. Gemas utilizadas

En este sprint hemos añadido las siguientes gemas:

Gema	Versión
rails	5.1.4
mysql2	0.4.9
puma	3.10.0
sass-rails	5.0.6
uglifier	3.2.0
coffee-rails	4.2.2
turbolinks	5.0.1
jbuilder	2.7

5.1.3. Ficheros de migración

No han sido utilizados ficheros de migración en este sprint.

5.1.4. Modelos ORM

No ha sido creado ningún modelo en este sprint.

5.1.5. Vistas y controladores

Generamos el controlador de *about* con el siguiente comando:

```
rails generate controller about index
```

Se genera un fichero *about_controller.rb* en *app/controllers/* .

A continuación, modificamos dicho fichero de la siguiente forma:

app/controllers/about_controller.rb

```
class AboutController < ApplicationController
  def index
    @page_title = 'Sobre Esiflix'
  end
end
```

Cogemos de miniemporium los siguientes ficheros y los copiamos en nuestro proyecto:

- Hoja de estilos *CSS* - app/assets/stylesheets/style.css
- Vista principal de nuestra tienda - app/views/layouts/application.html.erb
- Vista del *about* - app/views/about/index.html.erb

Los modificamos para adaptarlos a nuestro proyecto:

app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
<head>
  <title><%= @page_title || 'Esiflix' %></title>
  <%= csrf_meta_tags %>
```

```
  <%= stylesheet_link_tag 'application', media: 'all',
'data-turbolinks-track': 'reload' %>
  <%= javascript_include_tag 'application', 'data-turbolinks-track':
'reload' %>
</head>

<body>
  <div id="header">
      <h1 id="logo">Esiflix&trade;</h1>
      <h2 id="slogan">Tu web de películas</h2>
  </div>

  <div id="menu">
      <ul>
      <li><a href="/admin/author">Autores</a> | </li>
      <li><a
href="/admin/publisher">Editoriales</a> | </li>
      <li><a href="/admin/book">Libros</a> | </li>
      <li><a href="/admin/order">Pedidos</a> | </li>
      <li><a href="/">Catálogo</a> | </li>
      <li><a href="/about">Sobre Esiflix</a> </li>
      </ul>
  </div>

  <div id="content">
      <h1><%= @page_title if @page_title %></h1>
      <% if flash[:notice] %>
      <div id="notice"><%= flash[:notice] %></div>
      <% end %>
      <%= yield %>
  </div>

  <% if @cart %>
      <div id="shopping_cart"><%= render :partial => 'cart/cart'
%></div>
  <% end %>

  <div id="footer">
      &copy; 2019 Esiflix
  </div>
</body>
</html>
```

app/views/about/index.html.erb

```
<p>Tienda de películas situada en la muy animada ESI.</p>
<h2>Dirección postal</h2>
<address>
Esiflix<br/>
C.P. 11519<br/>
Avenida Universidad de Cadiz, 10<br/>
</address>
```

La hoja de estilos no se modifica.

5.1.6 Test

Hemos realizado un test del controlador:

test/controllers/about_controller_test.rb

```
require 'test_helper'

class AboutControllerTest < ActionDispatch::IntegrationTest
  test "index" do
    get '/about/index'
    assert_response :success
    assert_template 'about/index'
    assert_equal 'Sobre Esiflix', assigns(:page_title)
    assert_select 'title', 'Sobre Esiflix'
  end
end
```

y lo ejecutamos con el comando:

```
rake test TEST=test/controllers/about_controller_test.rb
```

5.1.7 Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.1.8 Objetivos alcanzados

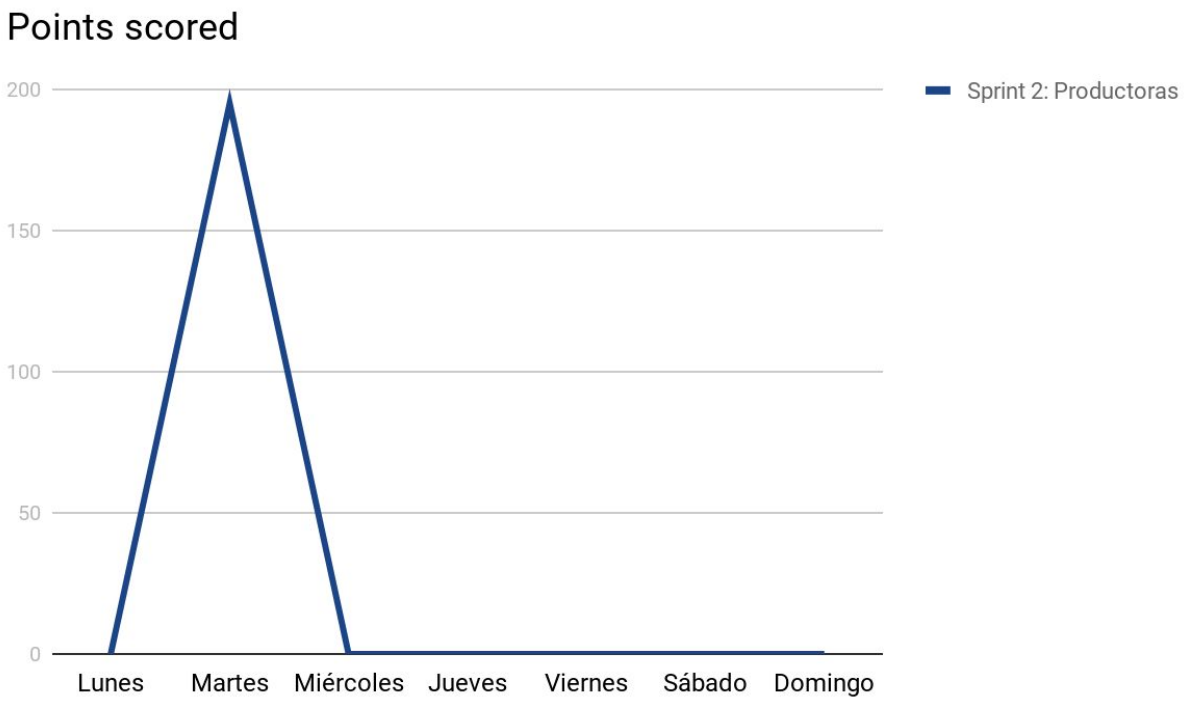
- Se ha realizado el about de la página ESIFLIX

5.2 Sprint 2

En este sprint generamos el CRUD de **Producer** entidad secundaria de **Film.**

5.2.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	20	0	0	0	0	0
Vista y controlador	0	125	0	0	0	0	0
Modelo	0	30	0	0	0	0	0
Test	0	20	0	0	0	0	0
Total	0	195	0	0	0	0	0



5.2.2 Gemas utilizadas

Gema	Versión
rails	5.1.4
mysql2	0.4.9
puma	3.10.0
sass-rails	5.0.6
uglifier	3.2.0
coffee-rails	4.2.2
turbolinks	5.0.1
jbuilder	2.7

5.2.3 Ficheros de migración

Creamos el modelo de la entidad **Producer** con el siguiente comando:

```
rails generate model Producer
```

db/migrate/20190319095120_create_producers.rb

```
class CreateProducers < ActiveRecord::Migration[5.1]
  def change
    create_table :producers do |t|
      t.string :name, :limit => 255, :null => false, :unique => true
      t.timestamps
    end
  end
end
```

Y finalmente ejecutamos el comando de migración a la base de datos:

```
rake db:migrate
```

5.2.4 Modelos ORM

Dejamos el modelo de la siguiente forma:

app/models/producer.rb

```
class Producer < ApplicationRecord
  validates_presence_of :name
  validates_uniqueness_of :name
  validates_length_of :name, :in => 2..255
end
```

5.2.5 Vistas y controladores

Generamos el controlador de **Producer** con el siguiente comando:

```
rails generate controller admin/producer new create edit update
destroy show index
```

Y modificamos el fichero del controlador:

app/controllers/admin/producer_controller.rb

```
class Admin::ProducerController < ApplicationController
  def new
    @producer = Producer.new
    @page_title = 'Crear nueva productora'
  end

  def create
    @producer = Producer.new(producer_params)
    if @producer.save
      flash[:notice] = "Productora #{@producer.name} creada
correctamente."
      redirect_to :action => 'index'
    else
      @page_title = 'Crear nueva productora'
      render :action => 'new'
    end
  end

  def edit
```

```
    @producer = Producer.find(params[:id])
    @page_title = 'Editar productora'
  end

  def update
    @producer = Producer.find(params[:id])
    if @producer.update_attributes(producer_params)
      flash[:notice] = "Productora #{@producer.name} actualizada
correctamente."
      redirect_to :action => 'show', :id => @producer
    else
      @page_title = 'Editar productora'
      render :action => 'edit'
    end
  end

  def destroy
    @producer = Producer.find(params[:id])
    @producer.destroy
    flash[:notice] = "Productora #{@producer.name} eliminada
correctamente."
    redirect_to :action => 'index'
  end

  def show
    @producer = Producer.find(params[:id])
    @page_title = @producer.name
  end

  def index
    @producers = Producer.all
    @page_title = 'Listado de productoras'
  end

  private
    def producer_params
      params.require(:producer).permit(:name)
    end
end
```

Y modificamos las vistas correspondientes a **Producer**:

app/views/admin/producer/edit.html.erb

```
<%= form_tag :action => 'update', :id => @producer do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Actualizar productora' %>
<% end %>

<%= link_to 'Mostrar', :action => 'show', :id => @producer %> |
<%= link_to 'Atrás', :action => 'index' %>
```

app/views/admin/producer/_form.html.erb

```
<% if @producer.errors.any? %>
  <div id="errorExplanation">
  <h2>Esta productora no se puede guardar debido a <%=
pluralize(@producer.errors.count, "error", "errores") %>:</h2>
    <ul>
      <% @producer.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <p><label for="producer_name">Nombre</label><br/>
  <%= text_field 'producer', 'name' %></p>
</div>
```

app/views/admin/producer/index.html.erb

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Editar</th>
    <th>Eliminar</th>
  </tr>

  <%= render :partial => 'producer', :collection => @producers %>
```

```
</table>

<p><%= link_to 'Añadir una nueva productora', :action => 'new' %></p>
```

app/views/admin/producer/new.html.erb

```
<%= form_tag :action => 'create' do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Crear productora' %>
<% end %>

<%= link_to 'Atrás', :action => 'index' %>
```

app/views/admin/producer/_producer.html.erb

```
<tr>
  <td><%= link_to producer.name, :action => 'show', :id => producer
%></td>
  <td><%= link_to 'Editar', :action => 'edit', :id => producer
%></td>
  <td>
    <%= button_to 'Eliminar', { :action => 'destroy', :id => producer
},
        data: { confirm: "¿Está seguro de que quiere eliminar la
productora #{producer.name}?" } %>
  </td>
</tr>
```

app/views/admin/producer/show.html.erb

```
<dl>
  <dt>Nombre</dt>
  <dd><%= @producer.name %></dd>
</dl>

<%= link_to 'Editar', :action => 'edit', :id => @producer %> |
<%= link_to 'Atrás', :action => 'index' %>
```

5.2.6 Test

Se habrán creado los siguientes ficheros de test:

- test/models/producer_test.rb
- test/controllers/admin/producer_controller_test.rb

Los modificamos de la siguiente forma:

test/models/producer_test.rb

```
require 'test_helper'

class ProducerTest < ActiveSupport::TestCase
  test "failing_create" do
      producer = Producer.new
      assert_equal false, producer.save
      assert_equal 2, producer.errors.count
      assert producer.errors[:name]
  end

  test "create" do
      producer = Producer.new(
      :name => 'Warner Bros'
      )
      assert producer.save
  end
end
```

test/controllers/admin/producer_controller_test.rb

```
require 'test_helper'

class Admin::ProducerControllerTest < ActionDispatch::IntegrationTest
  fixtures :producers

  test "new" do
      get '/admin/producer/new'
      assert_response :success
  end

  test "create" do
      num_producers = Producer.count
      post '/admin/producer/create', :params => { :producer => {
```

```
:name => 'Marvel Comics' }}
      assert_response :redirect
      assert_redirected_to :action => 'index'
      assert_equal num_producers + 1, Producer.count
  end

  test "edit" do
      get '/admin/producer/edit', :params => { :id => 1 }
      assert_select 'input' do
      assert_select '[type=?]', 'text'
      assert_select '[name=?]', 'producer[name]'
      assert_select '[value=?]', 'Apress'
      end
  end

  test "update" do
      post '/admin/producer/update', :params => { :id => 1, :producer
=> { :name => 'Apress.com' }}
      assert_response :redirect
      assert_redirected_to :action => 'show', :id => 1
      assert_equal 'Apress.com', Producer.find(1).name
  end

  test "destroy" do
      assert_difference(Producer, :count, -1) do
      post '/admin/producer/destroy', :params => { :id => 1 }
      assert_equal flash[:notice], 'Productora Apress eliminada
correctamente.'
      assert_response :redirect
      assert_redirected_to :action => 'index'
      get '/admin/producer/index'
      assert_response :success
      assert_select 'div#notice', 'Productora Apress eliminada
correctamente.'
      end
  end

  test "show" do
      get '/admin/producer/show', :params => { :id => 1 }
      assert_response :success
      assert_template 'admin/producer/show'
      assert_not_nil assigns(:producer)
      assert assigns(:producer).valid?
      assert_select 'div#content' do
      assert_select 'h1', Producer.find(1).name
      end
  end
```

```
  test "index" do
      get '/admin/producer/index'
      assert_response :success
      assert_select 'table' do
      assert_select 'tr', Producer.count + 1
      end
      Producer.find_each do |a|
      assert_select 'td', a.name
      end
  end
end
```

5.2.7 Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.2.8 Objetivos alcanzados

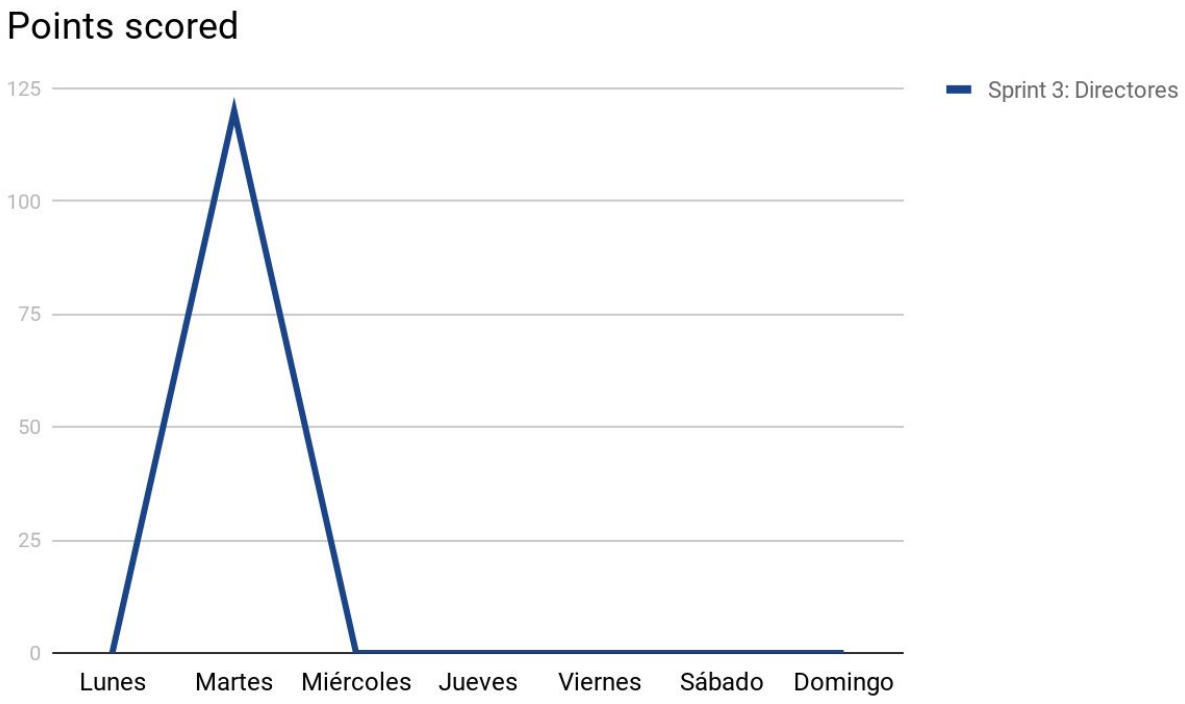
Con este sprint hemos conseguido poder visualizar el CRUD de la entidad secundaria *Producer*.

5.3 Sprint 3

En este sprint realizaremos la entidad secundaria **Director**.

5.3.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	10	0	0	0	0	0
Vista y controlador	0	60	0	0	0	0	0
Modelo	0	30	0	0	0	0	0
Test	0	20	0	0	0	0	0
TOTAL	0	120	0	0	0	0	0



5.3.2. Gemas utilizadas

En este sprint no hemos añadido nuevas gemas.

5.3.3. Ficheros de migración

Con este comando crearemos las plantillas del modelo, del test de modelo y del fichero de migración de **Director**.

```
$ rails generate model director
```

Modificamos el fichero de migración para generar la estructura de la tabla

db/migrate/20190326094409_create_directors.rb

```
class CreateDirectors < ActiveRecord::Migration[5.1]
  def change
    create_table :directors do |t|
      t.string :first_name, :limit => 255, :null => false, :unique =>
true
      t.string :last_name, :limit => 255, :null => false, :unique =>
```

```
true
      t.timestamps
    end
  end
end
```

Y creamos las tablas en la base de datos con:

```
$ rake db:migrate
```

5.3.4. Modelos ORM

Dejamos el fichero del modelo de la siguiente forma:

app/models/director.rb

```
class Director < ApplicationRecord
  validates_presence_of :first_name, :message => 'Debe introducir un
nombre.'
  validates_presence_of :last_name, :message => 'Debe introducir un
apellido.'

  def name
    "#{first_name} #{last_name}"
  end
end
```

5.3.5. Vistas y controladores

Generamos el controlador de **Director** con el siguiente comando:

```
rails generate controller admin/director new create edit update
destroy show index
```

Y modificamos el fichero del controlador:

app/controllers/admin/director_controller.rb

```
class Admin::DirectorController < ApplicationController
  def new
    @director = Director.new
    @page_title = 'Crear nuevo director'
  end

  def create
```

```
    @director = Director.new(director_params)
    if @director.save
      flash[:notice] = "Director #{@director.name} creado
correctamente."
      redirect_to :action => 'index'
    else
      @page_title = 'Crear nuevo director'
      render :action => 'new'
    end
  end

  def edit
    @director = Director.find(params[:id])
    @page_title = 'Editar director'
  end

  def update
    @director = Director.find(params[:id])
    if @director.update_attributes(director_params)
      flash[:notice] = "Director #{@director.name} actualizado
correctamente."
      redirect_to :action => 'show', :id => @director
    else
      @page_title = 'Editar director'
      render :action => 'edit'
    end
  end

  def destroy
    @director = Director.find(params[:id])
    @director.destroy
    flash[:notice] = "Director #{@director.name} eliminado
correctamente."
    redirect_to :action => 'index'
  end

  def show
    @director = Director.find(params[:id])
    @page_title = @director.name
  end

  def index
    @directors = Director.all
    @page_title = 'Listado de directores'
  end

  private
```

```
  def director_params
    params.require(:director).permit(:first_name, :last_name)
  end
end
```

Y ahora los ficheros correspondientes a las vistas:

app/views/admin/director/_director.html.erb

```
<tr>
  <td><%= link_to director.name, :action => 'show', :id => director
%></td>
  <td><%= link_to 'Editar', :action => 'edit', :id => director
%></td>
  <td>
    <%= button_to 'Eliminar', { :action => 'destroy', :id => director
},
        data: { confirm: "¿Está seguro de que quiere eliminar el
director #{director.name}?" } %>
  </td>
</tr>
```

app/views/admin/director/edit.html.erb

```
<%= form_tag :action => 'update', :id => @director do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Actualizar director' %>
<% end %>

<%= link_to 'Mostrar', :action => 'show', :id => @director %> |
<%= link_to 'Atrás', :action => 'index' %>
```

app/views/admin/director/_form.html.erb

```
<% if @director.errors.any? %>
  <div id="errorExplanation">
  <h2>Este director no se puede guardar debido a <%=
pluralize(@director.errors.count, "error", "errores") %>:</h2>
```

```
    <ul>
      <% @director.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <p><label for="director_first_name">Nombre</label><br/>
  <%= text_field 'director', 'first_name' %></p>
</div>

<div class="field">
  <p><label for="director_last_name">Apellidos</label><br/>
  <%= text_field 'director', 'last_name' %></p>
</div>
```

app/views/admin/director/index.html.erb

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Editar</th>
    <th>Eliminar</th>
  </tr>

  <%= render :partial => 'director', :collection => @directors %>
</table>

<p><%= link_to 'Añadir un nuevo director', :action => 'new' %></p>
```

app/views/admin/director/new.html.erb

```
<%= form_tag :action => 'create' do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Crear director' %>
<% end %>

<%= link_to 'Atrás', :action => 'index' %>
```

app/views/admin/director/show.html.erb

```
<dl>
  <dt>Nombre</dt>
  <dd><%= @director.first_name %></dd>
  <dt>Apellidos</dt>
  <dd><%= @director.last_name %></dd>
</dl>

<%= link_to 'Editar', :action => 'edit', :id => @director %> |
<%= link_to 'Atrás', :action => 'index' %>
```

5.3.6. Test

Crearemos el test correspondiente al modelo de **Director**:

test/models/director_test.rb

```
require 'test_helper'

class DirectorTest < ActiveSupport::TestCase
  test "test_name" do
    director = Director.create(:first_name => 'Christopher',
:last_name => 'Nolan')
    assert_equal 'Christopher Nolan', director.name
  end
end
```

Y el test del controlador:

test/controllers/admin/director_controller_test.rb

```
require 'test_helper'

class Admin::DirectorControllerTest < ActionDispatch::IntegrationTest
fixtures :directors

  test "new" do
    get '/admin/director/new'
```

```
    assert_template 'admin/director/new'
    assert_select 'div#content' do
      assert_select 'h1', 'Crear nuevo director'
      assert_select "form[action=\"/admin/director/create\"]"
    end
  end

  test "create" do
    get '/admin/director/new'
    assert_template 'admin/director/new'
    assert_difference(Director, :count) do
      post '/admin/director/create', :params => { :director => {
:first_name => 'Jon', :last_name => 'Anderson' }}
      assert_response :redirect
      assert_redirected_to :action => 'index'
    end
    assert_equal 'Director Jon Anderson creado correctamente.',
flash[:notice]
  end

  test "failing_create" do
    assert_no_difference(Director, :count) do
      post '/admin/director/create', :params => { :director => {
:first_name => 'Jon' }}
      assert_response :success
      assert_template 'admin/director/new'
      assert_select "div[class=\"field_with_errors\"]"
    end
  end

  test "edit" do
    get '/admin/director/edit', :params => { :id => 1 }
    assert_select 'input' do
      assert_select '[type=?]', 'text'
      assert_select '[name=?]', 'director[first_name]'
      assert_select '[value=?]', 'Joel'
    end
    assert_select 'input' do
      assert_select '[type=?]', 'text'
      assert_select '[name=?]', 'director[last_name]'
      assert_select '[value=?]', 'Spolsky'
    end
  end

  test "update" do
    post '/admin/director/update', :params => { :id => 1, :director
=> { :first_name => 'Joseph', :last_name => 'Anderson' }}
```

```
    assert_response :redirect
    assert_redirected_to :action => 'show', :id => 1
    director = Director.find(1)
    assert_equal 'Joseph', director.first_name
    assert_equal 'Anderson', director.last_name
  end

  test "test_destroy" do
    assert_difference(Director, :count, -1) do
      post '/admin/director/destroy', :params => { :id => 1 }
      assert_equal flash[:notice], 'Director Joel Spolsky eliminado
correctamente.'
      assert_response :redirect
      assert_redirected_to :action => 'index'
      get '/admin/director/index'
      assert_response :success
      assert_select 'div#notice', 'Director Joel Spolsky eliminado
correctamente.'
    end
  end

  test "show" do
    get '/admin/director/show', :params => { :id => 1 }
    assert_template 'admin/director/show'
    assert_equal 'Joel', assigns(:director).first_name
    assert_equal 'Spolsky', assigns(:director).last_name
    assert_select 'div#content' do
      assert_select 'h1', Director.find(1).name
    end
  end

  test "index" do
    get '/admin/director/index'
    assert_response :success
    assert_select 'table' do
      assert_select 'tr', Director.count + 1
    end
    Director.find_each do |a|
      assert_select 'td', a.name
    end
  end
end
```

5.3.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint, es prácticamente similar al de **Producer**.

5.3.8. Objetivos alcanzados

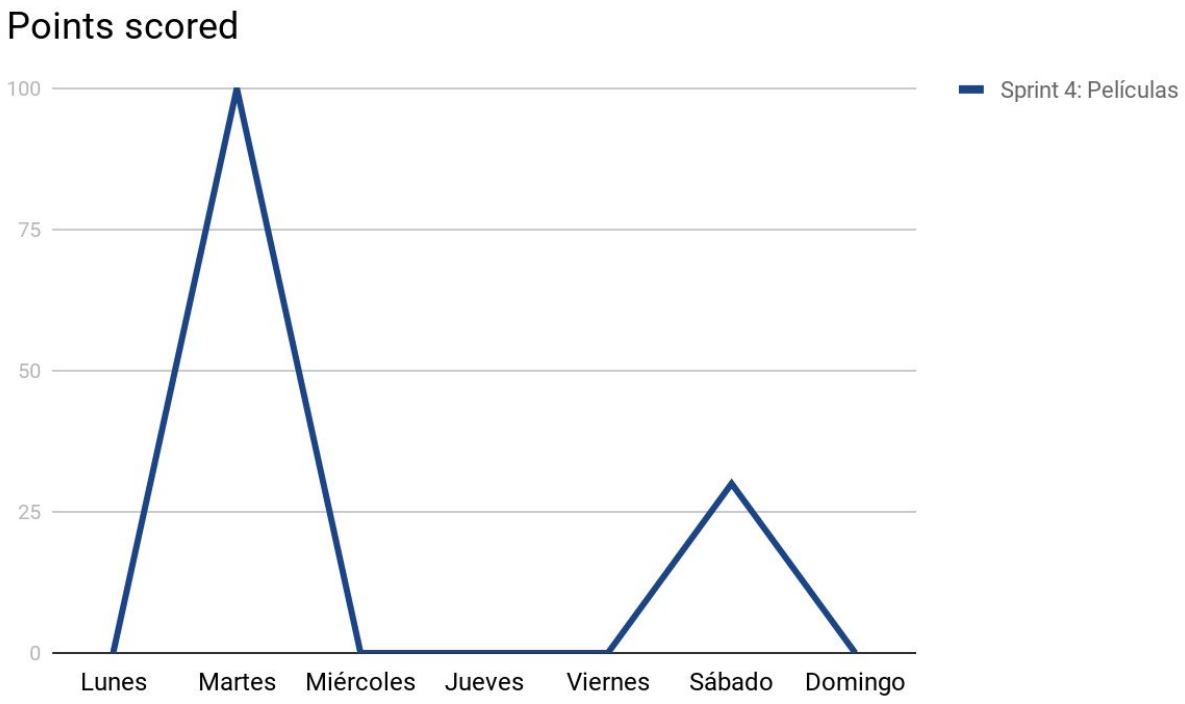
Con este sprint hemos conseguido poder crear el CRUD de **Director**.

5.4 Sprint 4

En este Sprint hemos desarrollado la entidad principal Película y su relación con la entidad Director.

5.4.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	20	0	0	0	0	0
Vista y controlador	0	30	0	0	0	30	0
Modelo	0	30	0	0	0	0	0
Test	0	20	0	0	0	0	0
TOTAL	0	100	0	0	0	30	0



5.4.2. Gemas utilizadas

En este sprint hemos añadido las siguientes gemas:

Gema	Versión
will_paginate	3.0.7
RedCloth	4.2.9
paperclip	4.3.2

5.4.3. Ficheros de migración

Creamos el fichero de migración de **film** y **directors_films**:

```
rails generate migration create_films_and_directors_films
```

db/migrate/20190326115535_create_films_and_directors_films.rb

```
class CreateFilmsAndDirectorsFilms < ActiveRecord::Migration[5.1]
  def up
    create_table :films do |t|
       t.string :title, :limit => 255, :null => false
       t.integer :producer_id, :limit => 8, :null => false
       t.datetime :produced_at
       t.string :kind, :limit => 15
       t.text :blurb
       t.integer :duration
       t.float :price
       t.timestamps
    end

    create_table :directors_films do |t|
      t.integer :director_id, :limit => 8, :null => false
      t.integer :film_id, :limit => 8, :null => false
      t.timestamps
    end

    say_with_time 'Adding foreing keys' do
      # Add foreign key reference to directors_films table
      execute 'ALTER TABLE directors_films ADD CONSTRAINT
fk_directors_films_directors
               FOREIGN KEY (director_id) REFERENCES directors(id) ON
DELETE CASCADE'
      execute 'ALTER TABLE directors_films ADD CONSTRAINT
fk_directors_films_films
               FOREIGN KEY (film_id) REFERENCES films(id) ON DELETE
CASCADE'
      # Add foreign key reference to films table
      execute 'ALTER TABLE films ADD CONSTRAINT fk_films_producers
               FOREIGN KEY (producer_id) REFERENCES producers(id) ON
DELETE CASCADE'
    end
  end

  def self.down
    drop_table :films
    drop_table :directors_films
  end

end
```

Y ahora para la inserción de imágenes añadimos:

db/migrate/20190402114503_add_cover_image_attachment_to_films.rb

```
class AddCoverImageAttachmentToFilms < ActiveRecord::Migration[5.1]
  def up
    add_attachment :films, :cover_image
  end

  def down
    remove_attachment :films, :cover_image
  end
end
```

Y modificamos el modelo de **film**:

app/models/film.rb

```
class Film < ApplicationRecord
  has_and_belongs_to_many :directors
  belongs_to :producer

  #has_many :cart_items
  #has_many :carts, :through => :cart_items

  has_attached_file :cover_image
  validates_attachment :cover_image,
  :content_type => { :content_type => ["image/jpeg", "image/gif",
"image/png"] }

  validates_length_of :title, :in => 1..255
  validates_presence_of :producer
  validates_presence_of :directors
  validates_presence_of :produced_at
  validates_numericality_of :duration, :only_integer => true
  validates_numericality_of :price
  validates_length_of :kind, :in => 1..15

  def director_names
    self.directors.map{|director| director.name}.join(", ")
  end

  #def self.latest(num)
    #all.order("films.id desc").includes(:directors,
:producer).limit(num)
  #end
```

```
end
```

5.4.4. Modelos ORM

Dejamos el fichero del modelo de la siguiente forma:

app/models/film.rb

```
class Film < ApplicationRecord
  has_and_belongs_to_many :directors
  belongs_to :producer

  #has_many :cart_items
  #has_many :carts, :through => :cart_items

  has_attached_file :cover_image
  validates_attachment :cover_image,
  :content_type => { :content_type => ["image/jpeg", "image/gif",
"image/png"] }

  validates_length_of :title, :in => 1..255
  validates_presence_of :producer
  validates_presence_of :directors
  validates_presence_of :produced_at
  validates_numericality_of :duration, :only_integer => true
  validates_numericality_of :price
  validates_length_of :kind, :in => 1..15

  def director_names
    self.directors.map{|director| director.name}.join(", ")
  end

  #def self.latest(num)
    #all.order("films.id desc").includes(:directors,
:producer).limit(num)
  #end
end
```

5.4.5. Vistas y controladores

Generamos el controlador de **Film**:

```
rails generate controller admin/film new create edit update destroy
show index
```

Y modificamos el fichero del controlador:

app/controllers/admin/film_controller.rb

```
# coding: utf-8
class Admin::FilmController < ApplicationController
  def new
    load_data
    @film = Film.new
    @page_title = 'Crear nueva película'
  end

  def create
    @film = Film.new(film_params)
    if @film.save
      flash[:notice] = "Película #{@film.title} creada
correctamente."
      redirect_to :action => 'index'
    else
      load_data
      @page_title = 'Crear nueva película'
      render :action => 'new'
    end
  end

  def edit
    load_data
    @film = Film.find(params[:id])
    @page_title = 'Editar película'
  end

  def update
    @film = Film.find(params[:id])
    if @film.update_attributes(film_params)
      flash[:notice] = "Película #{@film.title} actualizada
correctamente."
      redirect_to :action => 'show', :id => @film
    else
      load_data
      @page_title = 'Editar película'
      render :action => 'edit'
    end
  end
```

```
  def destroy
    @film = Film.find(params[:id])
    @film.destroy
    flash[:notice] = "Película #{@film.title} eliminada
correctamente."
    redirect_to :action => 'index'
  end

  def show
    @film = Film.find(params[:id])
    @page_title = @film.title
  end

  def index
    sort_by = params[:sort_by]
    @films = Film.order(sort_by).paginate(:page => params[:page],
:per_page => 5)
    @page_title = 'Listado de películas'
  end

  private

    def load_data
      @directors = Director.all
      @producers = Producer.all
    end

    def film_params
      params.require(:film).permit(:title, :producer_id,
:produced_at, { :director_ids => [] },
                                   :kind, :blurb, :price, :duration,
:cover_image)
    end
end
```

Y modificamos las vistas correspondientes a **film**:

app/views/admin/film/edit.html.erb

```
<%= form_tag "/admin/film/update?id=#{@film.id}", :multipart => true
do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Actualizar película' %>
```

```
<% end %>

<%= link_to 'Mostrar', :action => 'show', :id => @film %> |
<%= link_to 'Atrás', :action => 'index' %>
```

app/views/admin/film/_form.html.erb

```
<% if @film.errors.any? %>
  <div id="errorExplanation">
  <h2>Esta película no se puede guardar debido a <%=
pluralize(@film.errors.count, "error", "errores") %>:</h2>
    <ul>
      <% @film.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <p><label for="film_title">Título</label><br/>
  <%= text_field 'film', 'title' %></p>
</div>

<div class="field">
  <p><label for="film_producer">Productora</label><br/>
  <%= collection_select :film, :producer_id, @producers, :id, :name
%></p>
</div>

<div class="field">
  <p><label for="film[director_ids][]">Directores</label><br/>
  <%= select_tag 'film[director_ids][]',
options_from_collection_for_select(@directors, :id, :name,
      @film.directors.collect{|director| director.id}), { :multiple
=> true, :size => 5 } %></p>
</div>

<div class="field"><p>
  <label for="film_produced_at">Producida el</label><br/>
  <%= datetime_select 'film', 'produced_at' %></p>
</div>

<div class="field">
```

```
  <p><label for="film_kind">Tipo</label><br/>
  <%= text_field 'film', 'kind' %></p>
</div>

<div class="field">
  <p><label for="film_blurb">Descripción</label><br/>
  <%= text_area 'film', 'blurb' %></p>
</div>

<div class="field">
  <p><label for="film_price">Precio</label><br/>
  <%= text_field 'film', 'price' %></p>
</div>

<div class="field">
  <p><label for="film_duration">Duración</label><br/>
  <%= text_field 'film', 'duration' %></p>
</div>

<div class="field">
  <% if @film.cover_image.exists? then %>
    <dd><%= image_tag @film.cover_image.url %></dd>
  <% else %>
    <p> No hay imagen de portada. Por favor, adjunte una. </p>
  <% end %>
  <p><label for="film_cover_image">Imagen de portada</label><br/>
  <%= file_field 'film', :cover_image %></p>
</div>
```

app/views/admin/film/index.html.erb

```
<table>
  <tr>
    <th><a href="?sort_by=producer_id">Productora</a></th>
    <th><a href="?sort_by=title">Título</a></th>
    <th><a href="?sort_by=kind">Tipo</a></th>
    <th colspan="3"></th>
  </tr>

<% @films.each do |film| %>
  <tr>
    <td><%= h film.producer.name %></td>
    <td><%= h film.title %></td>
    <td><%= h film.kind %></td>
```

```
    <td><%= link_to 'Mostrar', :action => 'show', :id => film %></td>
    <td><%= link_to 'Editar', :action => 'edit', :id => film  %></td>
    <td><%= button_to 'Eliminar', { :action => 'destroy', :id => film
},
                      data: { confirm: "¿Está seguro de que quiere
eliminar la película #{film.title}?" } %>
  </tr>
<% end %>
</table>
<br/>
<%= will_paginate @films, :page_links => false, :link_separator => '
| ',
                          :previous_label => 'Página anterior',
:next_label => 'Página siguiente' %>
<p><%= link_to 'Añadir una nueva película', :action => 'new' %></p>
```

app/views/admin/film/new.html.erb

```
<%= form_tag "/admin/film/create", :multipart => true do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Crear película' %>
<% end %>

<%= link_to 'Atrás', :action => 'index' %>
```

app/views/admin/film/show.html.erb

```
<dl>
  <dt>Título</dt>
  <dd><%= @film.title %></dd>
  <dt>Productora</dt>
  <dd><%= @film.producer.name %></dd>
  <dt>Producida el</dt>
  <dd><%= @film.produced_at.strftime("%d/%m/%Y a las %I:%M%p")
%></dd>
  <dt>Directores</dt>
  <dd><%= @film.director_names %></dd>
  <dt>Tipo</dt>
  <dd><%= @film.kind %></dd>
  <dt>Descripción</dt>
  <%= RedCloth.new(@film.blurb).to_html.html_safe if @film.blurb %>
```

```
  <dt>Precio</dt>
  <dd><%= @film.price %></dd>
  <dt>Duración</dt>
  <dd><%= @film.duration %></dd>
  <dt>Imagen de portada</dt>
  <% if @film.cover_image.exists? then %>
    <dd><%= image_tag @film.cover_image.url %></dd>
  <% else %>
    <dd><%= image_tag '/missing.png' %></dd>
    <p> No hay imagen de portada. Por favor, adjunte una. </p>
  <% end %>
</dl>

<%= link_to 'Editar', :action => 'edit', :id => @film %> |
<%= link_to 'Atrás', :action => 'index' %>
```

5.4.6. Test

Se genera el test del modelo de **Película**

test/models/film_test.rb

```
require 'test_helper'

class FilmTest < ActiveSupport::TestCase
  fixtures :directors, :producers, :films, :directors_films

  test "failing_create" do
      film = Film.new
      assert_equal false, film.save
      assert_equal 8, film.errors.count
      assert film.errors[:title]
      assert film.errors[:producer]
      assert film.errors[:directors]
      assert film.errors[:produced_at]
      assert film.errors[:kind]
      assert film.errors[:blurb]
      assert film.errors[:duration]
      assert film.errors[:price]
  end

  test "create" do
      film = Film.new(
```

```
      :title => 'Ruby on Rails',
      :directors => Director.all,
      :producer_id => Producer.find(1).id,
      :produced_at => Time.now,
      :kind => '123-123-123-1',
      :blurb => 'A great book',
      :duration => 375,
      :price => 45.5
      )
  assert film.save
  end

  test "has_many_and_belongs_to_mapping" do
      apress = Producer.find_by_name("Apress");
      count = apress.films.count
      film = Film.new(
      :title => 'Pro Rails E-Commerce 8th Edition',
      :directors =>
[Director.find_by_first_name_and_last_name('Joel', 'Spolsky'),
                 Director.find_by_first_name_and_last_name('Jeremy',
'Keith')],
      :produced_at => Time.now,
      :kind => '123-123-123-x',
      :blurb => 'E-Commerce on Rails',
      :duration => 400,
      :price => 55.5
      )
      apress.films << film
      apress.reload
      film.reload
      assert_equal count + 1, apress.films.count
      assert_equal 'Apress', film.producer.name
  end

  test "has_many_and_belongs_to_many_directors_mapping" do
      film = Film.new(
      :title => 'Pro Rails E-Commerce 8th Edition',
      :directors =>
[Director.find_by_first_name_and_last_name('Joel', 'Spolsky'),
                 Director.find_by_first_name_and_last_name('Jeremy',
'Keith')],
      :producer_id => Producer.find_by_name("Apress").id,
      :produced_at => Time.now,
      :kind => '123-123-123-x',
      :blurb => 'E-Commerce on Rails',
      :duration => 400,
      :price => 55.5
```

```
      )
      assert film.save
      film.reload
      assert_equal 2, film.directors.count
      assert_equal 2,
Director.find_by_first_name_and_last_name('Joel',
'Spolsky').films.count
  end
end
```

5.4.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.4.8. Objetivos alcanzados

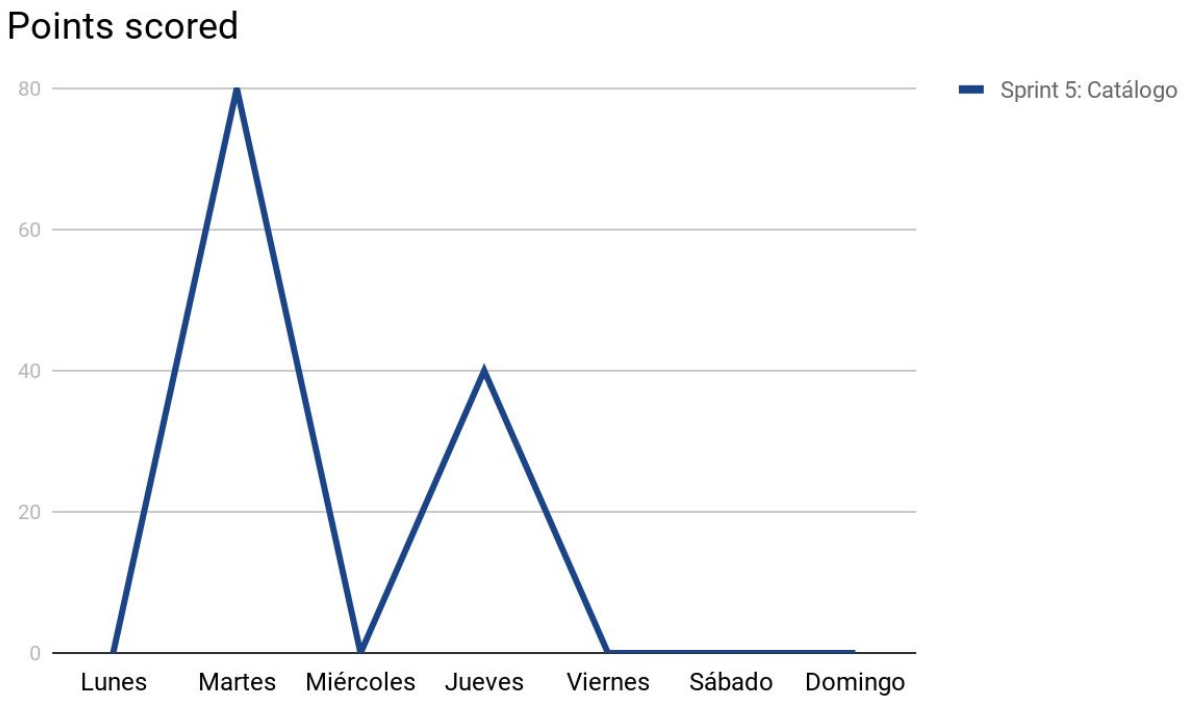
Se ha realizado la entidad principal y se ha efectuado con éxito la relación con las entidades secundarias. Se ha creado el CRUD y se ha añadido también la inserción de imágenes y de paginación.

5.4 Sprint 5

En este Sprint hemos desarrollado el catálogo de nuestra tienda donde los clientes podrán ver nuestras películas a partir de la información ya almacenada anteriormente. Añadimos además el test correspondiente.

5.5.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	20	0	0	0	0	0
Vista y controlador	0	30	0	20	0	0	0
Modelo	0	30	0	0	0	0	0
Test	0	0	0	20	0	0	0
TOTAL	0	80	0	40	0	0	0



5.5.2. Gemas utilizadas

En este Sprint no hemos añadido ninguna gema nueva.

5.5.3. Ficheros de migración

Tampoco hemos añadido ningún fichero de migración nuevo.

5.5.4. Modelos ORM

Tampoco ha sido necesario crear ningún modelo nuevo, todo lo necesario para la información de la base de datos está ya creado.

5.5.5. Vistas y controladores

Generamos el controlador del catálogo:

```
rails generate controller Catalog index show latest
```

Modificamos el fichero:

app/controller/catalog_controller.rb

```
# coding: utf-8
class CatalogController < ApplicationController
```

```
  before_action :initialize_cart, :except => :show
  #before_action :require_no_user

  def show
    @film = Film.find(params[:id])
    @page_title = @film.title
  end

  def index
    @films = Film.order("films.id desc").includes(:directors,
:producer).paginate(:page => params[:page], :per_page => 5)
    @page_title = 'Catálogo'
  end

  def latest
    @films = Film.latest 5 # invoques "latest" method to get the five
latest films
    @page_title = 'Últimas películas'
  end
end
```

En el fichero film.rb añadimos el siguiente método:

app/models/film.rb

```
def self.latest(num)
    all.order("films.id desc").includes(:directors,
:producer).limit(num)
  end
```

Y ahora realizaremos las vistas relacionadas con el catálogo:

app/views/catalog/_films.html.erb

```
<dl id = 'films'>
  <% for film in @films %>
    <dt>
      <%= link_to film.title, :action => 'show', :id => film %>
      <%= link_to '+', :controller => 'cart', :action => 'add', :id
=> film %>
    </dt>
    <% for director in film.directors %>
```

```
      <dd><%= director.last_name %>, <%= director.first_name %></dd>
    <% end %>
    <dd><%= sprintf("Duración: %s. Precio: %.2f €",
pluralize(film.duration, "minuto"), film.price) %> </dd>
    <dd><small>Productora: <%= film.producer.name %></small></dd>
  <% end %>
</dl>
```

app/views/catalog/index.html.erb

```
<%= render :partial => 'films' %>
<%= will_paginate @films, :page_links => false, :link_separator => '
| ',
                          :previous_label => 'Página anterior',
:next_label => 'Página siguiente' %>
```

app/views/catalog/latest.html.erb

```
<%= render :partial => 'films' %>
```

app/views/catalog/show.html.erb

```
<h2>por <%=@film.director_names %></h2>
<% if @film.cover_image.exists? then %>
  <dd><%= image_tag @film.cover_image.url %></dd>
<% else %>
  <p>Imagen de portada no disponible.</p>
<% end %>
<dl>
  <dt>Precio</dt>
  <dd><%= sprintf("%0.2f €", @film.price) %></dd>
  <dt>Duración</dt>
  <dd><%= pluralize(@film.duration, "minuto") %></dd>
  <dt>Productora</dt>
  <dd><%= @film.producer.name %></dd>
  <dt>Descripción</dt>
  <%= RedCloth.new(@film.blurb).to_html.html_safe if @film.blurb %>
</dl>
```

```
<p><%= link_to 'Catálogo', :action => 'index' %> </p>
```

5.5.6. Test

Generamos los test de integración ejecutando el comando:

```
rails generate integration_test film_administration_test
```

Lo modificamos:

test/integration/film_administration_test.rb

```
require 'test_helper'

class FilmAdministrationTest < ActionDispatch::IntegrationTest

  test "film_aministration" do
    producer = Producer.create(:name => 'Films of Ruby')
    director = Director.create(:first_name => 'John', :last_name =>
'Anderson')
    george = new_session_as(:george)

    new_film_ruby = george.add_film :film => {
      :title => 'A new Film of Ruby',
      :producer_id => producer.id,
      :director_ids => [director.id],
      :produced_at => Time.now,
      :kind => '123-123-123-X',
      :blurb => 'A new Film of Ruby',
      :duration => 325,
      :price => 45.5
    }

    george.list_films
    george.show_film new_film_ruby

    george.edit_film new_film_ruby, :film => {
      :title => 'A very new Film of Ruby',
      :producer_id => producer.id,
      :director_ids => [director.id],
      :produced_at => Time.now,
```

```
      :kind => '123-123-123-X',
      :blurb => 'A very new Film of Ruby',
      :duration => 350,
      :price => 50
    }

    bob = new_session_as(:bob)
    bob.delete_film new_film_ruby
  end

  private

  module FilmTestDSL
    attr_writer :name

    def add_film(parameters)
      director = Director.first
      producer = Producer.first
      get '/admin/film/new'
      assert_response :success
      assert_template 'admin/film/new'
      assert_select 'select#film_producer_id' do
        assert_select "option[value=\"#{producer.id}\"]",
producer.name
      end
      assert_select "select[name=\"film[director_ids][]\"]" do
        assert_select "option[value=\"#{director.id}\"]",
director.name
      end
      post '/admin/film/create', :params => parameters
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'admin/film/index'
      page = Film.all.count / 5 + 1
      get "/admin/film/index/?page=#{page}"
      assert_select 'td', parameters[:film][:title]
      film = Film.find_by_title(parameters[:film][:title])
      return film;
    end

    def edit_film(film, parameters)
      get "/admin/film/edit?id=#{film.id}"
      assert_response :success
      assert_template 'admin/film/edit'
      post "/admin/film/update?id=#{film.id}", :params => parameters
      assert_response :redirect
```

```
      follow_redirect!
      assert_response :success
      assert_template 'admin/film/show'
    end

    def delete_film(film)
      post "/admin/film/destroy?id=#{film.id}"
      assert_response :redirect
      follow_redirect!
      assert_template 'admin/film/index'
    end

    def show_film(film)
      get "/admin/film/show/#{film.id}"
      assert_response :success
      assert_template 'admin/film/show'
    end

    def list_films
      get '/admin/film/index'
      assert_response :success
      assert_template 'admin/film/index'
    end
  end

  def new_session_as(name)
    open_session do |session|
      session.extend(FilmTestDSL)
      session.name = name
      yield session if block_given?
    end
  end
end
```

Creamos también el test de integración:

app/test/integration/browsing_and_searching_test.rb

```
require 'test_helper'

class BrowsingAndSearchingTest < ActionDispatch::IntegrationTest
  fixtures :producers, :directors, :films, :directors_films

  test "browse" do
```

```
    jill = new_session_as :jill
    jill.index
    jill.second_page
    jill.film_details 'Pride and Prejudice'
    jill.latest_films
  end

  module BrowsingTestDSL
    include ERB::Util
    attr_writer :name

    def index
      get '/catalog/index'
      assert_response :success
      assert_select 'dl#films' do
        assert_select 'dt', :count => 5
      end
      assert_select 'dt' do
        assert_select 'a', 'The Idiot'
      end
      check_film_links
    end

    def second_page
      get '/catalog/index?page=2'
      assert_response :success
      assert_template 'catalog/index'
      assert_equal Film.find_by_title('Pro Rails E-Commerce'),
                   assigns(:films).last
      check_film_links
    end

    def film_details(title)
      @film = Film.where(:title => title).first
      get "/catalog/show/#{@film.id}"
      assert_response :success
      assert_template 'catalog/show'
      assert_select 'div#content' do
        assert_select 'h1', @film.title
        assert_select 'h2', "por #{@film.directors.map{|a|
a.name}.join(", ")}"
      end
    end

    def latest_films
      get '/catalog/latest'
      assert_response :success
```

```
      assert_template 'catalog/latest'
      assert_select 'dl#films' do
        assert_select 'dt', :count => 5
      end
      @films = Film.latest(5)
      @films.each do |a|
        assert_select 'dt' do
          assert_select 'a', a.title
        end
      end
    end
  end

  def check_film_links
    for film in assigns :films
      assert_select 'a' do
        assert_select '[href=?]', "/catalog/show/#{film.id}"
      end
    end
  end

  def new_session_as(name)
    open_session do |session|
      session.extend(BrowsingTestDSL)
      session.name = name
      yield session if block_given?
    end
  end
end
```

5.5.7 Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.5.8. Objetivos alcanzados

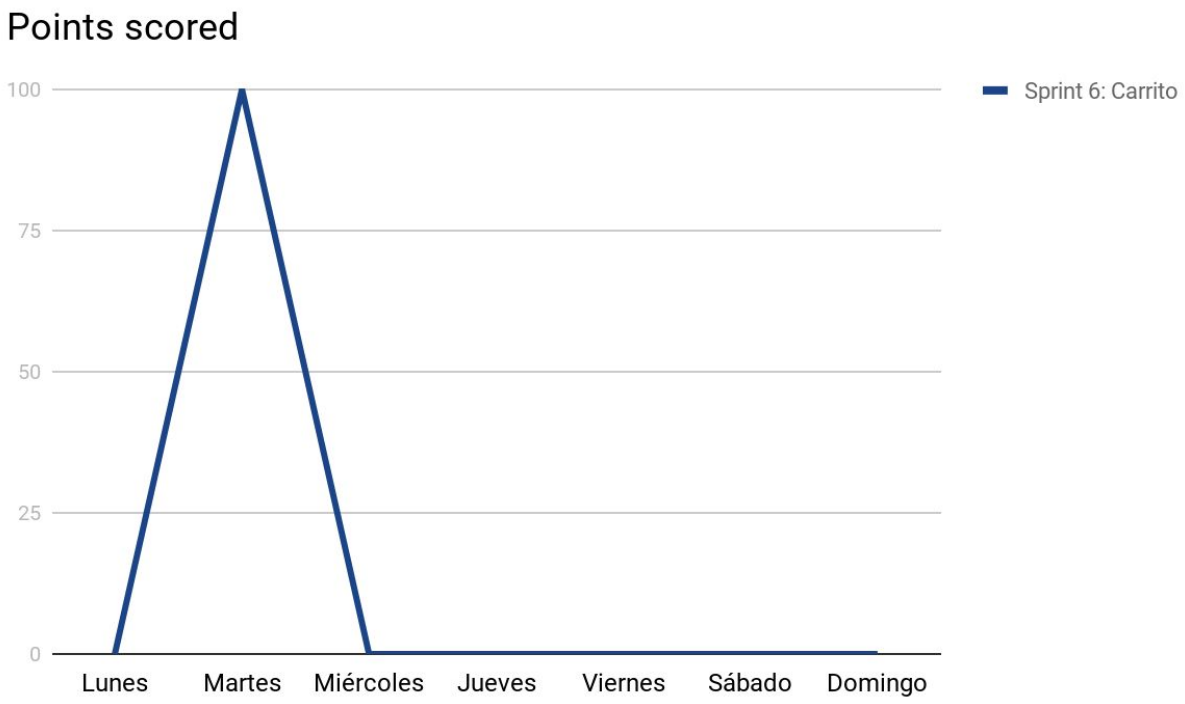
En este Sprint hemos creado el catálogo para nuestras películas y además, listamos las últimas películas añadidas.

5.4 Sprint 6

En este Sprint crearemos el carrito de la compra para posibilitar la adquisición de películas en nuestra tienda online. Dicha funcionalidad nos permitirá añadir películas a la lista que queramos comprar, eliminarla y vaciar el carrito.

5.6.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	20	0	0	0	0	0
Vista y controlador	0	30	0	0	0	0	0
Modelo	0	30	0	0	0	0	0
Test	0	20	0	0	0	0	0
TOTAL	0	100	0	0	0	0	0



5.6.2. Gemas utilizadas

En este sprint hemos añadido las siguientes gemas:

Gema	Versión
jquery-rails	4.0.5

5.6.3. Ficheros de migración

Creamos el modelo de la entidad **Cart** y **CartItem** con los siguientes comandos:

```
rails generate model Cart
```

```
rails generate model CartItem
```

db/migrate/20190423104800_create_carts.rb

```
class CreateCarts < ActiveRecord::Migration[5.1]
  def change
    create_table :carts do |t|
      t.timestamps
    end
  end
end
```

db/migrate/20190423105000_create_cart_items.rb

```
class CreateCartItems < ActiveRecord::Migration[5.1]
  def change
    create_table :cart_items do |t|
      t.integer :film_id, :limit => 8
      t.integer :cart_id, :limit => 8
      t.float :price
      t.integer :amount
      t.timestamps
    end
  end
end
```

Y realizamos el comando: `rake db:migrate`

5.6.4. Modelos ORM

Los ficheros correspondiente al modelo quedarían de la siguiente forma:

/app/models/cart.rb

```
class Cart < ApplicationRecord
  has_many :cart_items
  has_many :films, :through => :cart_items

  def add(film_id)
    items = cart_items.where(film_id: film_id)
    film = Film.find film_id
    if items.size < 1
      ci = cart_items.create :film_id => film_id, :amount => 1,
:price => film.price
    else
      ci = items.first
      ci.update_attribute :amount, ci.amount + 1
    end
    ci
  end

  def remove(film_id)
    ci = cart_items.where(film_id: film_id).first
    if ci.amount > 1
      ci.update_attribute :amount, ci.amount - 1
    else
      CartItem.destroy ci.id
    end
    ci
  end

  def total
    sum = 0
    cart_items.each do |item| sum += item.price * item.amount end
    sum
  end
end
```

/app/models/cart_item.rb

```
class CartItem < ApplicationRecord
  belongs_to :cart
  belongs_to :film
end
```

5.6.5. Vistas y controladores

Tras ejecutar: `rails generate controller Cart` editaremos:

El fichero del controlador del carrito:

/app/controllers/cart_controller.rb

```
class CartController < ApplicationController
  before_action :initialize_cart

  def add
    @film = Film.find params[:id]
    @page_title = 'Añadir artículo'
    if request.post?
      @item = @cart.add params[:id]
      flash[:cart_notice] = "Añadido <em>#{@item.film.title}</em>."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'add', :template =>
'cart/add'
    end
  end

  def remove
    @film = Film.find params[:id]
    @page_title = 'Eliminar artículo'
    if request.post?
      @item = @cart.remove params[:id]
      flash[:cart_notice] = "Eliminado <em>#{@item.film.title}</em>."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'remove', :template =>
'cart/remove'
    end
  end

  def clear
    @page_title = 'Vaciar carrito'
    if request.post?
      @cart.cart_items.destroy_all
      flash[:cart_notice] = "Carrito vaciado."
      redirect_to :controller => 'catalog'
    else
```

```
      render :controller => 'cart', :action => 'clear', :template =>
'cart/clear'
    end
  end
end
```

Ahora vamos a mostrar los ficheros correspondientes a la vista:

/app/views/cart/_cart.html.erb

```
<% if flash[:cart_notice] %>
<%= render :partial => 'cart/cart_notice' %>
<% end %>

<h3>Su carrito de la compra</h3>
<p>
  <strong>
    <% unless controller.controller_name == 'checkout' %>
      <%= link_to 'Proceder a la facturación', :controller =>
'checkout' %>
    <% end %>
  </strong>
</p>
<ul>
  <% for item in @cart.cart_items %>
    <li id="cart_item_<%= item.film.id %>">
      <%= render :partial => 'cart/item', :object => item %>
    </li>
  <% end %>
</ul>
<p id='cart_total'><strong>Total: <%= sprintf "%0.2f €", @cart.total
%></strong></p>
<% unless @cart.cart_items.empty? %>
  <p id='clear_cart_link'>
  <b><%= link_to 'Vaciar carrito', :controller => 'cart', :action =>
'clear' %></b>
  </p>
<% end %>
```

/app/views/cart/_cart_notice.html.erb

```
<p id='cart_notice'><%= flash[:cart_notice].html_safe if flash[:cart_notice] %></p>
```

/app/views/cart/add.html.erb

```
<strong>Por favor, confirme la agregación de <em><%= @film.title %></em> al carrito
de la compra.</strong>
<br><br>
<div style="float: left; width: auto;">
<%= button_to 'Cancelar', { :controller => 'catalog', :action => 'index' }, :method
=> :get %>
</div>
<div style="float: left; width: auto;">
<%= button_to 'Confirmar', :action => 'add', :id => params[:id] %>
</div>
<br>
```

/app/views/cart/clear.html.erb

```
<strong>Por favor, confirme el vaciado del carrito de la compra.</strong>
<br><br>
<div style="float: left; width: auto;">
<%= button_to 'Cancelar', { :controller => 'catalog', :action => 'index' }, :method
=> :get %>
</div>
<div style="float: left; width: auto;">
<%= button_to 'Confirmar', :action => 'clear', :id => params[:id] %>
</div>
<br>
```

/app/views/cart/remove.html.erb

```
<strong>Por favor, confirme la eliminación de <em><%= @film.title %></em> del
carrito de la compra.</strong>
<br><br>
<div style="float: left; width: auto;">
<%= button_to 'Cancelar', { :controller => 'catalog', :action => 'index' }, :method
=> :get %>
</div>
<div style="float: left; width: auto;">
<%= button_to 'Confirmar', :action => 'remove', :id => params[:id] %>
</div>
<br>
```

5.6.6. Test

Realizamos el test para el controlador del carrito de la compra:

/test/controllers/cart_controller_test.rb

```
 require 'test_helper'

class CartControllerTest < ActionDispatch::IntegrationTest
  fixtures :directors, :producers, :films

  test "add" do
    assert_difference(CartItem, :count) do
      post '/cart/add', :params => { :id => 4 }
    end
    assert_response :redirect
    assert_redirected_to :controller => 'catalog'
    assert_equal 1, Cart.find(@request.session[:cart_id]).cart_items.size
  end

  test "remove" do
    post '/cart/add', :params => { :id => 4 }
    assert_equal [Film.find(4)], Cart.find(@request.session[:cart_id]).films

    post '/cart/remove', :params => { :id => 4 }
    assert_equal [], Cart.find(@request.session[:cart_id]).films
  end

  test "clear" do
    post '/cart/add', :params => { :id => 4 }
    assert_equal [Film.find(4)], Cart.find(@request.session[:cart_id]).films

    post '/cart/clear'
    assert_response :redirect
    assert_redirected_to :controller => 'catalog'
    assert_equal [], Cart.find(@request.session[:cart_id]).films
  end
end
```

5.6.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.6.8. Objetivos alcanzados

En este Sprint hemos creado el carrito de la compra para comprar películas. Además, podremos añadir al carrito, borrar artículos y vaciar todo el carrito.

5.4 Sprint 7

En este Sprint crearemos la facturación de pedidos. Dicha funcionalidad nos permitirá realizar el pago de los productos que queramos adquirir.

5.7.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	20	0	0	0	0	0	0
Vista y controlador	30	0	0	0	0	0	0
Modelo	30	0	0	0	0	0	0
Test	40	0	0	0	0	0	0
TOTAL	120	0	0	0	0	0	0

5.7.2. Gemas utilizadas

En este sprint hemos añadido las siguientes gemas:

Gema	Versión
countries	2.1.2
country_select	3.1.1

activemerchant	1.73.0

5.7.3. Ficheros de migración

Creamos los siguientes ficheros de migración:

db/migrate/20190430140523_create_orders.rb

```
class CreateOrders < ActiveRecord::Migration[5.1]
  def change
    create_table :orders do |t|
      # contact information
      t.string :email
      t.string :phone_number
      # shipping address
      t.string :ship_to_first_name
      t.string :ship_to_last_name
      t.string :ship_to_address
      t.string :ship_to_city
      t.string :ship_to_postal_code
      t.string :ship_to_country_code
      # private fields
      t.string :customer_ip
      t.string :status
      t.string :error_message
      t.timestamps
    end
  end
end
```

db/migrate/20190501101220_create_order_items.rb

```
class CreateOrderItems < ActiveRecord::Migration[5.1]
  def self.up
    create_table :order_items do |t|
      t.integer :film_id, :limit => 8
      t.integer :order_id, :limit => 8
      t.float :price
      t.integer :amount
      t.timestamps
    end

    say_with_time 'Adding foreing keys' do
```

```
      # Add foreign key reference to order_items table
      execute 'ALTER TABLE order_items ADD CONSTRAINT
fk_order_items_orders
              FOREIGN KEY (order_id) REFERENCES orders(id) ON DELETE
CASCADE'
    end
  end

  def self.down
    drop_table :order_items
  end
end
```

Utilizamos el comando *rake db:migrate* para que surta efecto.

5.7.4. Modelos ORM

Creamos los modelos ***order*** y ***order_item***.

app/models/order.rb

```
# coding: utf-8
class Order < ApplicationRecord
  require "active_merchant/billing/rails"

  attr_accessor :card_type, :card_number, :card_expiration_month,
:card_expiration_year,
                :card_verification_value

  has_many :order_items
  has_many :films, :through => :order_items

  validates_presence_of :order_items,
                        :message => '¡Su carrito de la compra está
vacío! ' +
                                    'Por favor, agregue al menos una
película antes de realizar el pedido.'
  validates_format_of :email, :with =>
/\A([^@\s]+)@((?:[-a-z0-9]+\.)+[a-z]{2,})\Z/i, :message => 'Debe
introducir un correo válido.'
  validates_length_of :phone_number, :in => 7..20, :message => 'Debe
introducir un número de teléfono válido.'

  validates_length_of :ship_to_first_name, :in => 2..255, :message =>
'Nombre no válido.'
```

```
  validates_length_of :ship_to_last_name, :in => 2..255, :message =>
'Apellido no válido.'
  validates_length_of :ship_to_address, :in => 2..255, :message =>
'Dirección no válida.'
  validates_length_of :ship_to_city, :in => 2..255, :message =>
'Ciudad no válida.'
  validates_length_of :ship_to_postal_code, :in => 2..255, :message
=> 'Código postal no válido.'
  validates_length_of :ship_to_country_code, :in => 2..255, :message
=> 'Código de país no válido.'

  validates_length_of :customer_ip, :in => 7..15
  validates_inclusion_of :status, :in => %w(open processed closed
failed)

  validates_inclusion_of :card_type, :in => ['Visa', 'MasterCard',
'American Express', 'Discover'], :on => :create
  validates_length_of :card_number, :in => 13..19, :on => :create
  validates_inclusion_of :card_expiration_month, :in => %w(1 2 3 4 5
6 7 8 9 10 11 12), :on => :create
  validates_inclusion_of :card_expiration_year, :in => %w(2017 2018
2019 2020 2021 2022), :on => :create
  validates_length_of :card_verification_value, :in => 3..4, :on =>
:create

  def total
    sum = 0
    order_items.each do |item|
      sum += item.price * item.amount
    end
    sum
  end

  def amount
    sum = 0
    order_items.each do |item|
      sum += item.amount
    end
    sum
  end

  def process
    begin
      raise 'No puede procesarse de nuevo un pedido ya cerrado' if
self.closed?
      active_merchant_payment
    rescue => e
```

```
      logger.error("Pedido #{id} fallido debido a una excepción:
#{e}.")
      self.error_message = "Excepción generada: #{e}"
      self.status = 'failed'
    end
    save!
    self.processed?
  end

  def active_merchant_payment
    ActiveMerchant::Billing::Base.mode = :test
    ActiveMerchant::Billing::AuthorizeNetGateway.default_currency =
'EUR'
    ActiveMerchant::Billing::AuthorizeNetGateway.wiredump_device =
STDERR
    ActiveMerchant::Billing::AuthorizeNetGateway.wiredump_device.sync
= true
    self.status = 'failed' # order status by default

    # the card verification value is also known as CVV2, CVC2, or CID
    creditcard = ActiveMerchant::Billing::CreditCard.new(
      :brand              => card_type,
      :number             => card_number,
      :month              => card_expiration_month,
      :year               => card_expiration_year,
      :verification_value => card_verification_value,
      :first_name         => ship_to_first_name,
      :last_name          => ship_to_last_name
    )

    # buyer information
    shipping_address = {
      :first_name => ship_to_first_name,
      :last_name  => ship_to_last_name,
      :address1   => ship_to_address,
      :city       => ship_to_city,
      :zip        => ship_to_postal_code,
      :country    => ship_to_country_code,
      :phone      => phone_number,
    }

    # order information
    details = {
      :description      => 'Esiflix online store purchase',
      :order_id         => self.id,
      :email            => email,
      :ip               => customer_ip,
```

```
      :billing_address  => shipping_address,
      :shipping_address => shipping_address
    }

    if creditcard.valid? # validating the card automatically detects
the card type
      gateway = ActiveMerchant::Billing::AuthorizeNetGateway.new( #
use the test account
        :login    => '3CFzaE62a',
        :password => '2DwH3Nbp2Ca223Q5'
        # the statement ":test = 'true'" tells the gateway not to
process transactions
      )

      # Active Merchant accepts all amounts as integer values in
cents
      response = gateway.purchase(self.total * 100, creditcard,
details)

      if response.success?
        self.status = 'processed'
      else
        self.error_message = response.message
      end
    else
      self.error_message = 'Tarjeta de crédito no válida'
    end
  end

  def processed?
    self.status == 'processed'
  end

  def failed?
    self.status == 'failed'
  end

  def closed?
    self.status == 'closed'
  end

  def close
    self.status = 'closed'
    save!
  end
end
```

app/models/order_item.rb

```
# coding: utf-8
class OrderItem < ApplicationRecord
  belongs_to :order
  belongs_to :film

  def validate
    errors.add(:amount, "debe ser uno o más") unless amount.nil? ||
amount > 0
    errors.add(:price, "debe ser un número positivo") unless
price.nil? || price > 0.0
  end
end
```

5.7.5. Vistas y controladores

Generamos los controladores y las vistas con los siguientes comandos:

- *rails generate controller admin/order index show _navigation*
- *rails generate controller checkout index thank_you*

Los modificamos para que queden de la siguiente forma:

app/controllers/admin/order_controller.rb

```
class Admin::OrderController < ApplicationController
  def close
    order = Order.find(params[:id])
    order.close
    flash[:notice] = "El pedido ##{order.id} ha sido cerrado."
    redirect_to :action => 'index'
  end

  def show
    @order = Order.find(params[:id])
    @page_title = "Mostrando pedido ##{@order.id}"
  end

  def index
    @status = params[:id]
    if @status.blank?
```

```
      conditions = nil
      @status = 'all'
      @page_title = 'Listando todos los pedidos'
    else
      conditions = "status = '#{@status}'"
      @estado = 'abierto' if @status == 'open'
      @estado = 'procesado' if @status == 'processed'
      @estado = 'cerrado' if @status == 'closed'
      @estado = 'fallido' if @status == 'failed'
      @page_title = "Listando pedidos #{@estado.pluralize}"
    end
    @orders = Order.where(conditions).paginate(:page =>
params[:page], :per_page => 10)
  end
end
```

app/controllers/checkout_controller.rb

```
# coding: utf-8
class CheckoutController < ApplicationController
  before_action :initialize_cart, :only => :index

  def index
    @order = Order.new
    @page_title = 'Facturación'
    if @cart.films.empty?
      flash[:notice] = '¡Su carrito de la compra está vacío! ' +
                       'Por favor, agruegue al menos una película
antes de proceder a la facturación.'
      redirect_to :controller => 'catalog'
    end
  end

  def submit_order
    @cart = Cart.find(params[:cart][:id]) # Search the cart from the
cart id hidden field of the form
    @order = Order.new(order_params)
    @order.ship_to_country_code = @order.ship_to_country_code.upcase
    @order.customer_ip = request.remote_ip
    @order.status = 'open'
    @page_title = 'Facturación'
    populate_order

    if @order.save
      if @order.process
```

```
        flash[:notice] = 'Su pedido ha sido realizado y se procesará
inmediatamente.'
        session[:order_id] = @order.id
        @cart.cart_items.destroy_all # empty shopping cart
        redirect_to :action => 'thank_you'
      else
        flash[:notice] = "Error al realizar el pedido
'#{@order.error_message}'."
        render :action => 'index'
      end
    else
      render :action => 'index'
    end
  end

  def thank_you
    @page_title = 'Gracias.'
  end

  private

  def populate_order
    for cart_item in @cart.cart_items
      order_item = OrderItem.new(:film_id => cart_item.film_id,
                                 :price => cart_item.price,
                                 :amount => cart_item.amount)
      @order.order_items << order_item
    end
  end

  def order_params
      params.require(:order).permit(:email, :phone_number,
:ship_to_first_name, :ship_to_last_name, :ship_to_address,
:ship_to_city, :ship_to_postal_code, :ship_to_country_code,
:card_type,
      :card_expiration_month, :card_expiration_year, :card_number,
:card_verification_value)
  end
end
```

app/views/admin/order/index.html.erb

```
<% if @orders == [] %>
  <% if @status == 'all' %>
    <h2><%= "No hay pedidos." %></h2>
  <% else %>
    <h2><%= "No hay pedidos con el estado '#{@estado}'." %></h2>
  <% end %>
<% else %>
  <table>
    <tr>
      <th>ID</th>
      <th>Estado</th>
      <th>Precio total</th>
      <th>Cantidad</th>
      <th>Creado el</th>
      <th>Actualizado el</th>
      <th></th>
    </tr>
  <% for order in @orders %>
    <% estado = 'procesado' if order.status == 'processed'
       estado = 'fallido' if order.status == 'failed'
       estado = 'abierto' if order.status == 'open'
       estado = 'cerrado' if order.status == 'closed' %>
    <tr>
      <td align="center"><%= order.id %></td>
      <td align="center"><%= estado.capitalize %></td>
      <td align="center"><%= order.total %></td>
      <td align="center"><%= order.amount %></td>
      <td align="center"><%= order.created_at.strftime("%d-%m-%Y
%H:%M") %></td>
      <td align="center"><%= order.updated_at.strftime("%d-%m-%Y
%H:%M") %></td>
      <td><%= link_to 'Mostrar', :action => 'show', :id => order
%></td>
    </tr>
  <% end %>
  </table>

  <% if @orders.total_pages > 1 %>
    <br/>
    <%= 'Ver página:' %>
  <% end %>

  <%= will_paginate @orders, :page_links => true, :link_separator =>
' ', :container => false,
                              :previous_label => '', :next_label => ''
```

```
%>
  <p></p>
<% end %>

<%= render :partial => 'navigation' %>
```

app/views/admin/order/show.html.erb

```
<h2>Información de contacto</h2>
<dl>
  <dt>Correo electrónico</dt>
  <dd><%= @order.email %></dd>
  <dt>Teléfono</dt>
  <dd><%= @order.phone_number %></dd>
</dl>

<h2>Dirección de envío</h2>
<dl>
  <dt>Nombre</dt>
  <dd><%= @order.ship_to_first_name %></dd>
  <dt>Apellidos</dt>
  <dd><%= @order.ship_to_last_name %></dd>
  <dt>Dirección</dt>
  <dd><%= @order.ship_to_address %></dd>
  <dt>Ciudad</dt>
  <dd><%= @order.ship_to_city %></dd>
  <dt>Código postal</dt>
  <dd><%= @order.ship_to_postal_code %></dd>
  <dt>País</dt>
  <dd><%=
ISO3166::Country.find_country_by_alpha2(@order.ship_to_country_code).
name %></dd>
</dl>

<h2>Detalles</h2>
<% for item in @order.order_items %>
  <%= link_to item.film.title, :controller => 'film', :action =>
'show', :id => item.film.id %>
  <%= pluralize(item.amount, "película", "películas") %>, <%=
item.price * item.amount %> € <br/>
<% end %>

<p><strong>Total: <%= @order.total %> €</strong></p>

<h2>Estado</h2>
```

```
<dl>
  <dt>Estado</dt>
  <% estado = 'procesado' if @order.status == 'processed'
     estado = 'fallido' if @order.status == 'failed'
     estado = 'abierto' if @order.status == 'open'
     estado = 'cerrado' if @order.status == 'closed' %>
  <dd><%= estado.capitalize %></dd>
  <% if @order.failed? %>
    <dt>Error</dt>
    <dd><%= @order.error_message %></dd>
  <% end %>
</dl>

<% if !@order.closed? %> <p></p> <% end %>
<%= button_to 'Cerrar pedido', {:action => 'close', :id => @order},
    data: { confirm: "¿Está seguro de que quiere cerrar el pedido
##{@order.id}?" } if @order.processed? %>
<%= render :partial => 'navigation' %>
```

app/views/admin/order/_navigation.html.erb

```
<p>
  <strong>Listar: <%= link_to 'todos', :action => 'index', :id => ''
%>,
  <%= link_to 'abiertos', :action => 'index', :id => 'open' %>,
  <%= link_to 'procesados', :action => 'index', :id => 'processed'
%>,
  <%= link_to 'cerrados', :action => 'index', :id => 'closed' %>,
  <%= link_to 'fallidos', :action => 'index', :id => 'failed'
%></strong>
</p>
```

app/views/checkout/index.html.erb

```
<% if @order.errors.any? %>
  <div id="errorExplanation">
  <h2>Este pedido no se puede guardar debido a <%=
pluralize(@order.errors.count, "error", "errores") %>:</h2>
    <ul>
      <% @order.errors.full_messages.each do |msg| %>
        <% if msg.include?('Order items ') %>
          <li><%= msg.gsub('Order items ', '') %></li>
        <% else %>
          <li><%= msg %></li>
```

```
        <% end %>
      <% end %>
    </ul>
  </div>
<% end %>

<p><em>Su pedido se muestra en el carrito de la compra a la
derecha.</em></p>
<%= form_tag :action => 'submit_order' do %>
  <div id="checkout">
    <fieldset>
      <legend>Información de contacto</legend>
      <p><label for="order_email">Correo electrónico</label><br/>
      <%= text_field :order, :email %></p>
      <p><label for="order_phone_number">Teléfono</label><br/>
      <%= text_field :order, :phone_number %></p>
    </fieldset>

    <fieldset>
      <legend>Dirección de envío</legend>
      <p><label for="order_ship_to_first_name">Nombre</label><br/>
      <%= text_field :order, :ship_to_first_name %></p>
      <p><label for="order_ship_to_last_name">Apellidos</label><br/>
      <%= text_field :order, :ship_to_last_name %></p>
      <p><label for="order_ship_to_address">Dirección</label><br/>
      <%= text_field :order, :ship_to_address %></p>
      <p><label for="order_ship_to_city">Ciudad</label><br/>
      <%= text_field :order, :ship_to_city %></p>
      <p><label for="order_ship_to_postal_code">Código
postal</label><br/>
      <%= text_field :order, :ship_to_postal_code %></p>
      <p><label for="order_ship_to_country_code">País</label><br/>
      <%= country_select(:order, :ship_to_country_code,
priority_countries: ['ES']) %></p>
    </fieldset>

    <fieldset>
      <legend>Información de facturación</legend>
      <p><label for="order_card_type">Tipo de tarjeta de
crédito</label><br/>
      <select name="order[card_type]" id="order_card_type">
      <%= options_for_select(["Visa", "MasterCard", "American
Express", "Discover"], @order.card_type) %>
      </select></p>
      <p><label for="order_card_expiration_month">Fecha de
caducidad</label><br/>
      <select name="order[card_expiration_month]">
```

```
      <%= options_for_select(%w(1 2 3 4 5 6 7 8 9 10 11 12),
@order.card_expiration_month) %>
      </select>
      <select name="order[card_expiration_year]">
      <%= options_for_select(%w(2017 2018 2019 2020 2021 2022),
@order.card_expiration_year) %>
      </select></p>
      <p><label for="order_card_number">Número de
tarjeta</label><br/>
      <%= text_field :order, :card_number %></p>
      <p>
        <label for="order_card_verification_value">
        <abbr title="Card Verification Value">CVV</abbr>
        <abbr title="Card Validation Check">CVC</abbr>
        </label><br/>
        <%= text_field :order, :card_verification_value %>
      </p>
    </fieldset>

    <div class="field">
      <%= hidden_field :cart, :id %>
    </div>

    <p><%= submit_tag "Realizar pedido" %></p>
  </div>
<% end %>
```

app/views/checkout/thank_you.html.erb

```
Para futuras referencias use el numero de factura <%=
session[:order_id] %>
```

5.7.6. Test

Vamos a realizar el test del modelo de pedido:

/test/models/order_test.rb

```
require 'test_helper'

class OrderTest < ActiveSupport::TestCase
  test "create_valid_order" do
    order = Order.new(
```

```
      # Contact information
      :email => 'email@email.com',
      :phone_number => '666112233',
      # Shipping address
      :ship_to_first_name => 'Firstname',
      :ship_to_last_name => 'Lastname',
      :ship_to_address => 'Address',
      :ship_to_city => 'City',
      :ship_to_postal_code => '00000',
      :ship_to_country_code => 'ES',
      # Billing information
      :card_type => 'Visa',
      :card_number => '4007000000027',
      :card_expiration_month => '12',
      :card_expiration_year => '2022',
      :card_verification_value => '000'
    )

    # Private information
    order.customer_ip = '127.0.0.1'
    order.status = 'open'

    order.order_items << OrderItem.new(:film_id => 1, :price =>
155.25, :amount => 3)

    assert order.save
    assert order.process
    order.reload
    assert_equal 1, order.order_items.size
    assert_equal 155.25, order.order_items[0].price
    assert_equal order.status, 'processed'
    order.close
    assert order.closed?
  end

  test "validations" do
    order = Order.new
    assert_equal false, order.save
    assert_equal 16, order.errors.size

    # An order must have at least one order item
    assert order.errors[:order_items]
    # Contact information
    assert order.errors[:email]
    assert order.errors[:phone_number]
    # Shipping address
    assert order.errors[:ship_to_first_nane]
```

```
    assert order.errors[:ship_to_last_name]
    assert order.errors[:ship_to_address]
    assert order.errors[:ship_to_city]
    assert order.errors[:ship_to_postal_code]
    assert order.errors[:ship_to_country_code]
    #Billing information
    assert order.errors[:card_type]
    assert order.errors[:card_number]
    assert order.errors[:card_expiration_month]
    assert order.errors[:card_expiration_year]
    assert order.errors[:card_verification_value]
    # Private information
    assert order.errors[:customer_ip]
    assert order.errors[:status]
  end
end
```

Y crearemos el fichero para el test de integración de facturación con el comando:

```
rails generate integration_test Checkout
```

y editamos el fichero que hemos obtenido:

/test/integration/checkout_test.rb

```
require 'test_helper'

class CheckoutTest < ActionDispatch::IntegrationTest
  fixtures :producers, :directors, :films

  test "empty_cart_shows_error_message" do
    get '/checkout'
    assert_response :redirect
    assert_redirected_to :controller => 'catalog'
    assert_equal flash[:notice], '¡Su carrito de la compra está
vacío! ' +
                                 'Por favor, agruegue al menos una
película antes de proceder a la facturación.'
  end

  test "submitting_order" do
    post '/cart/add', :params => { :id => 1 }
    get '/checkout'
    assert_response :success
    assert_select 'legend', 'Información de contacto'
    assert_select 'legend', 'Dirección de envío'
    assert_select 'legend', 'Información de facturación'
```

```
    post '/checkout/submit_order', :params => { :cart => { :id =>
Cart.last.id }, :order => {
      # Contact information
      :email => 'email@email.com',
      :phone_number => '666112233',
      # Shipping address
      :ship_to_first_name => 'Firstname',
      :ship_to_last_name => 'Lastname',
      :ship_to_address => 'Address',
      :ship_to_city => 'City',
      :ship_to_postal_code => '00000',
      :ship_to_country_code => 'Country',
      # Billing information
      :card_type => 'Visa',
      :card_number => '4007000000027',
      :card_expiration_month => '12',
      :card_expiration_year => '2022',
      :card_verification_value => '000'
    }}

    assert_response :redirect
    assert_redirected_to '/checkout/thank_you'
  end
end
```

5.7.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.7.8. Objetivos alcanzados

Se ha conseguido con éxito la facturación y la gestión de los pedidos.

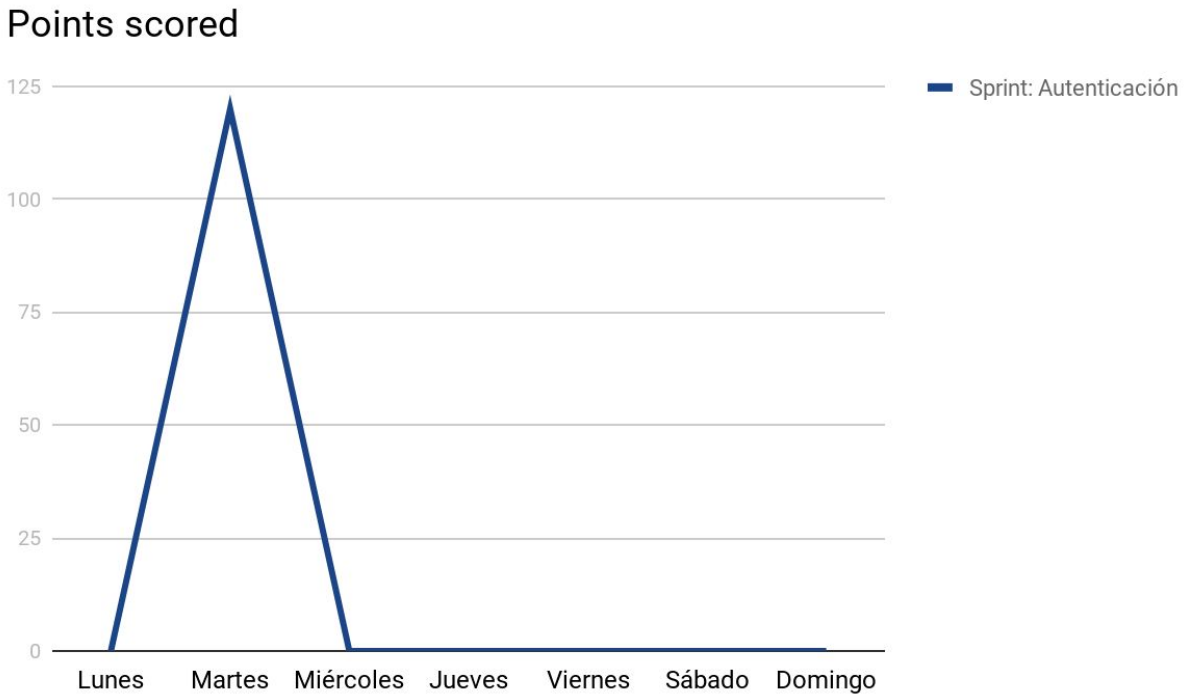
5.4 Sprint 8

En este Sprint implementaremos la autenticación de usuarios.

5.8.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	20	0	0	0	0	0

Vista y controlador	0	60	0	0	0	0	0
Modelo	0	20	0	0	0	0	0
Test	0	20	0	0	0	0	0
TOTAL	0	120	0	0	0	0	0



5.8.2. Gemas utilizadas

En este sprint hemos añadido las siguientes gemas:

Gema	Versión
authlogic	3.6.1

5.8.3. Ficheros de migración

Creamos los ficheros de migración de usuarios.

db/migrate/20190507110000_ create_users.rb

```
class CreateUsers < ActiveRecord::Migration[5.1]
  def change
    create_table :users do |t|
      t.string :name, :null => false, :unique => true
      t.string :login, :null => false, :unique => true
      t.string :email, :null => false, :unique => true
      t.string :crypted_password, :null => false
      t.string :password_salt, :null => false
      t.string :persistence_token, :null => false
      t.string :perishable_token, :null => false # optional, used for reset password
functionality

      #t.string :single_access_token, :null => false # optional, see Authlogic::Session::Params

      # magic fields (all optional, see Authlogic::Session::MagicColumns)
      t.integer :login_count, :null => false, :default => 0
      t.integer :failed_login_count, :null => false, :default => 0
      t.datetime :last_request_at
      t.datetime :current_login_at
      t.datetime :last_login_at
      t.string :current_login_ip
      t.string :last_login_ip
      t.timestamps
    end
  end
end
```

5.8.4. Modelos ORM

Creamos los modelos ***user*** y ***user_session***.

app/models/user.rb

```
# coding: utf-8
class User < ApplicationRecord
  acts_as_authentic do |a|
    a.validate_login_field = true
    a.validate_password_field = true
    a.validates_length_of_password_field_options = { minimum: 4 }
    a.require_password_confirmation = true
    a.validates_length_of_password_confirmation_field_options = {
minimum: 4 }
```

```
    a.logged_in_timeout = 5.minutes # default is 10.minutes
    # default encryption system uses "SCrypt" and requires 'scrypt'
gem
    # for using previous encryption system uncomment next statement
    # a.crypto_provider = Authlogic::CryptoProviders::Sha512
  end

  validates_presence_of :name, :login, :email, :password,
:password_confirmation, :message => 'no puede estar vacío'
  validates_length_of :name, :in => 3..225, :message => 'es demasiado
corto (mínimo 3 caracteres)'
  validates_uniqueness_of :name, :login, :email, :message => 'ya
existe'
end
```

app/models/user_session.rb

```
class UserSession < Authlogic::Session::Base
  logout_on_timeout true # default if false
end
```

5.8.5. Vistas y controladores

Generamos los controladores y las vistas con los siguientes comandos:

- *rails generate controller user new show edit _form*
- *rails generate controller user_sessions new _form*
- *rails generate controller admin/authenticated*

Los modificamos para que queden de la siguiente forma:

app/controllers/user_controller.rb

```
class UserController < ApplicationController
  before_action :require_user, :only => [:show, :edit, :update]

  def new
    @user = User.new
    @page_title = 'Crear nueva cuenta'
    if current_user
      flash[:notice] = 'Sólo se puede crear una cuenta.'
      redirect_to :controller => 'about', :action => 'index'
    else
```

```
      # only when there are no accounts it allows to create a new
one, unique in the system
      redirect_to :controller => 'user_sessions', :action => 'new'
unless User.count == 0
    end
  end

  def create
    @user = User.new(user_params)
    if @user.save # the new user has been logged in automatically
      flash[:notice] = "Cuenta #{@user.name} creada correctamente.
Sesión iniciada."
      redirect_to :action => 'show'
    else
      @page_title = 'Crear nueva cuenta'
      render :action => :new
    end
  end

  def edit
    @user = current_user
    @page_title = 'Editar cuenta'
  end

  def update
    @user = current_user
    if @user.update_attributes(user_params)
      flash[:notice] = "Cuenta #{@user.name} actualizada
correctamente."
      redirect_to :action => 'show'
    else
      @page_title = 'Editar cuenta'
      render :action => 'edit'
    end
  end

  def show
    @user = current_user
    @page_title = @user.name
  end

  private
    def user_params
      params.require(:user).permit(:name, :login, :email, :password,
:password_confirmation)
    end
end
```

app/controllers/user_sessions_controller.rb

```
# coding: utf-8
class UserSessionsController < ApplicationController
  before_action :require_no_user, :only => [:new, :create]

  def new
    @user_session = UserSession.new
    @page_title = 'Iniciar sesión'
  end

  def create
    @user_session = UserSession.new(user_session_params)
    @user_session.remember_me = false # just in case
    if @user_session.save
      flash[:notice] = "Sesión iniciada correctamente."
      redirect_back_or_default :controller => '/admin/director',
:action => :index # default login route
    else
      render :action => :new
    end
  end

  def destroy
    if current_user_session # only for an authenticated user
      current_user_session.destroy
      flash[:notice] = "Sesión cerrada correctamente."
    end
    redirect_to :controller => :catalog, :action => :index # logout
route
  end

  private
    def user_session_params
      params.require(:user_session).permit(:login, :password,
:remember_me)
    end
end
```

app/controllers/admin/authenticated_controller.rb

```
class Admin::AuthenticatedController < ApplicationController
  before_action :require_user
end
```

app/views/user/new.html.erb

```
<%= form_tag :action => 'create' do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Crear cuenta' %>
<% end %>
```

app/views/user/show.html.erb

```
<dl>
  <dt>Nombre</dt>
  <dd><%= @user.name %></dd>

  <dt>Usuario</dt>
  <dd><%= @user.login %></dd>

  <dt>Correo electrónico</dt>
  <dd><%= @user.email %></dd>

  <dt>Número de sesiones</dt>
  <dd><%= @user.login_count %></dd>

  <dt>Última petición</dt>
  <dd><%= @user.last_request_at %></dd>

  <% if @user.last_login_at %>
    <dt>Sesión anterior</dt>
    <dd><%= @user.last_login_at %></dd>
  <% end %>

  <dt>Sesión actual</dt>
  <dd><%= @user.current_login_at %></dd>

  <% if @user.last_login_ip %>
    <dt>IP de sesión anterior</dt>
    <dd><%= @user.last_login_ip %></dd>
  <% end %>

  <dt>IP de sesión actual</dt>
  <dd><%= @user.current_login_ip %></dd>
</dl>

<%= link_to 'Editar', :action => 'edit' %>
```

app/views/user/edit.html.erb

```
<%= form_tag :action => 'update', :id => @user do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Actualizar cuenta' %>
<% end %>

<%= link_to 'Atrás', :action => 'show' %>
```

app/views/user_sessions/_form.html.erb

```
<% if @user_session.errors.any? %>
  <div id="errorExplanation">
  <h2>Este usuario no puede iniciar sesión debido a <%=
pluralize(@user_session.errors.count, "error", "errores") %>:</h2>
    <ul>
      <% @user_session.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <p><label for="login">Usuario</label><br/>
  <%= text_field 'user_session', 'login' %></p>
</div>

<div class="field">
  <p><label for="password">Contraseña</label><br/>
  <%= password_field 'user_session', 'password' %></p>
</div>

<!-- <div class="check_box">
  <p><label for="remember_me">Recordarme</label><br/>
  <%= check_box 'user_session', 'remember_me' %></p>
</div> -->
```

app/views/user_sessions/new.html.erb

```
<%= form_tag :action => 'create' do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Iniciar sesión' %>
```

```
<% end %>
```

app/views/user_sessions/_form.html.erb

```
<% if @user_session.errors.any? %>
  <div id="errorExplanation">
  <h2>Este usuario no puede iniciar sesión debido a <%=
pluralize(@user_session.errors.count, "error", "errores") %>:</h2>
    <ul>
      <% @user_session.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <p><label for="login">Usuario</label><br/>
  <%= text_field 'user_session', 'login' %></p>
</div>

<div class="field">
  <p><label for="password">Contraseña</label><br/>
  <%= password_field 'user_session', 'password' %></p>
</div>

<!-- <div class="check_box">
  <p><label for="remember_me">Recordarme</label><br/>
  <%= check_box 'user_session', 'remember_me' %></p>
</div> -->
```

Además de esto, debemos modificar el resto de controladores de *admin*, sustituyendo la clase de la que heredan (*ApplicationController*) por *Admin::AuthenticatedController*.

Finalmente, modificamos el *layout*.

app/views/layouts/application.html.erb

```
<div id="header">
    <h1 id="logo">Esiflix&trade;</h1>
    <h2 id="slogan">Tu tienda on-line de películas</h2>
```

```
  <% if current_user %>
      <p id="loginlogout">
        Identificado como <%= current_user.login %>
        (<%= link_to "Editar cuenta", :controller => '/user', :action
=> :show %>)
        <br/>
        (<%= link_to "Cerrar sesión", :controller =>
'/user_sessions', :action => :destroy %>)
      </p>
    <% else %>
      <p id="loginlogout">
        <% if User.count == 0 %>
          (<%= link_to "Crear nueva cuenta", :controller => '/user',
:action => :new %>)
        <% else %>
          (<%= link_to "Iniciar sesión", :controller =>
'/user_sessions', :action => :new %>)
        <% end %>
      </p>
    <% end %>
  </div>

  <div id="menu">
```

5.8.6. Test

Realizamos los test de integración de usuario y de autenticación.

test/integration/user_test.rb

```
# coding: utf-8
require 'test_helper'

class UserTest < ActionDispatch::IntegrationTest

  def setup
  end

  test "user_account" do
      george = new_session_as(:george)
      user_account = george.creates_user_account(:user => { :name =>
'George Smith', :login => 'george',
                                                :email =>
'george@emporium.com', :password => 'gold',
```

```
                                                :password_confirmation =>
'gold' })
      george.shows_user_account user_account
      george.edits_user_account(user_account, :user => { :name =>
'George Jackson', :login => 'george',
                                                :email =>
'george@emporium.com', :password => 'silver',
                                                :password_confirmation =>
'silver' })
  end

  private

  module UserTestDSL
      include ERB::Util
      attr_writer :name

      def creates_user_account(parameters)
      user_name = parameters[:user][:name]
      get '/user/new'
      assert_response :success
      assert_template 'user/new'
      assert_select 'div#content' do
      assert_select 'h1', 'Crear nueva cuenta'
      assert_select 'input#user_name'
      end
      post '/user/create', :params => parameters
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'user/show'
      assert_select 'div#content' do
      assert_select 'h1', user_name
      assert_select 'dt', 'Nombre'
      assert_select 'dd', user_name
      end
      assert_equal flash[:notice], "Cuenta #{user_name} creada
correctamente. Sesión iniciada."
      assert_select 'div#notice', "Cuenta #{user_name} creada
correctamente. Sesión iniciada."
      return User.find_by_login(parameters[:user][:login])
      end

      def shows_user_account(user_account)
      get "/user/show/?id=#{user_account.id}"
      assert_response :success
      assert_template 'user/show'
```

```
      assert_select 'div#content' do
      assert_select 'h1', user_account.name
      assert_select 'dt', 'Nombre'
      assert_select 'dd', user_account.name
      end
      end

      def edits_user_account(user_account, parameters)
      get "/user/edit?id=#{user_account.id}"
      assert_response :success
      assert_template 'user/edit'
      assert_select 'div#content' do
      assert_select 'h1', 'Editar cuenta'
      assert_select 'input#user_name'
      end
      post "/user/update?id=#{user_account.id}", :params =>
parameters
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'user/show'
      user_name = parameters[:user][:name]
      assert_select 'div#content' do
      assert_select 'h1', user_name
      assert_select 'dt', 'Nombre'
      assert_select 'dd', user_name
      end
      assert_equal flash[:notice], "Cuenta #{user_name} actualizada
correctamente."
      assert_select 'div#notice', "Cuenta #{user_name} actualizada
correctamente."
      end
  end

  def new_session_as(name)
      open_session do |session|
      session.extend(UserTestDSL)
      session.name = name
      yield session if block_given?
      end
  end
end
```

test/integration/authentication_test.rb

```
require 'test_helper'

class AuthenticationTest < ActionDispatch::IntegrationTest

  def setup
      User.create(:name => 'George Smith',
                  :login => 'george',
                  :email => 'george@emporium.com',
                  :password => 'cheetah',
                  :password_confirmation => 'cheetah')
  end

  test "successful_login" do
      george = new_session_as(:george)
      george.tries_to_go_to_admin
      george.logs_in_succesfully("george", "cheetah")
  end

  test "failed_login" do
      harry = new_session_as(:harry)
      harry.tries_to_go_to_admin
      harry.fails_login("harry", "micky")
  end

  private

  module AuthenticationTestDSL
      include ERB::Util
      attr_writer :name

      def tries_to_go_to_admin
      get '/admin/film/new'
      assert_response :redirect
      assert_redirected_to '/user_sessions/new'
      end

      def logs_in_succesfully(login, password)
      post_login(login, password)
      assert_response :redirect
      assert_redirected_to '/admin/film/new'
      end

      def fails_login(login, password)
      post_login(login, password)
      assert_response :success
      assert_template 'user_sessions/new'
      assert_select 'div#content' do
```

```
      assert_select 'div#errorExplanation'
      assert_select 'li', 'Login is not valid'
      end
      end

      private

      def post_login(login, password)
      post '/user_sessions/create', :params => { :user_session => {
:login => login, :password => password }}
      end
  end

  def new_session_as(name)
      open_session do |session|
      session.extend(AuthenticationTestDSL)
      session.name = name
      yield session if block_given?
      end
  end
end
```

5.8.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.8.8. Objetivos alcanzados

Hemos integrado la autenticación de usuarios en la tienda online.

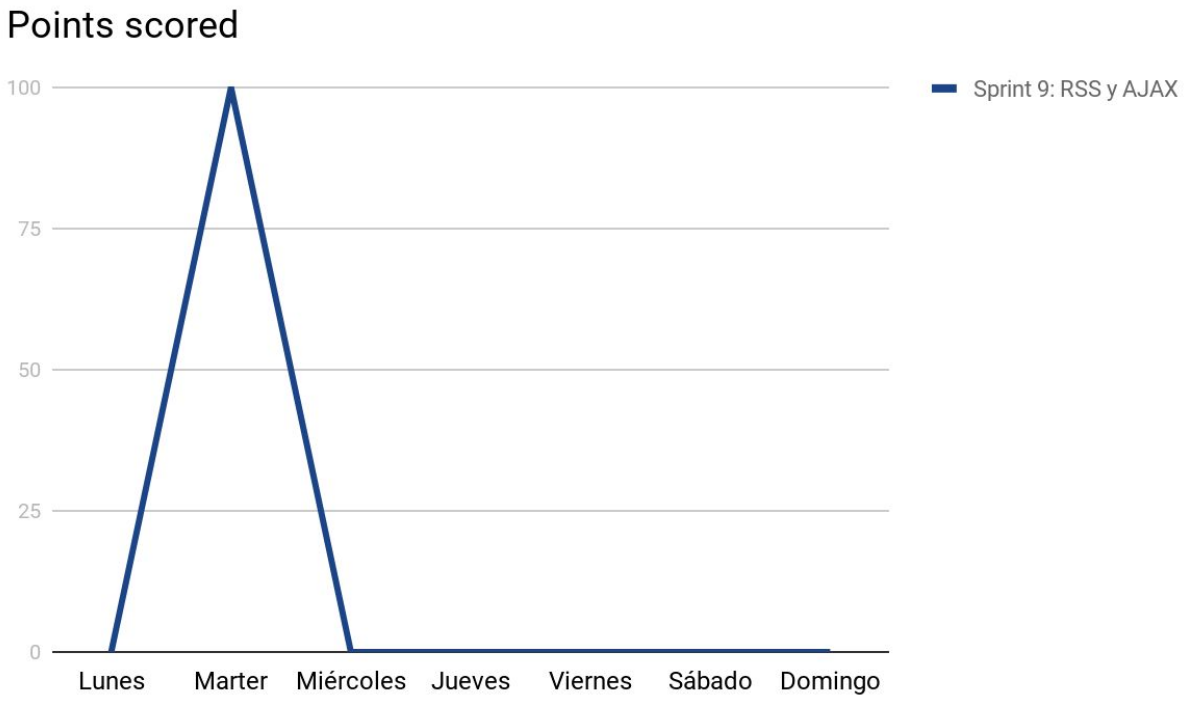
5.4 Sprint 9 (RSS y AJAX)

En este Sprint implementaremos el RSS (Really Simple Syndication) y AJAX (Asynchronous JavaScript And XML) en nuestra tienda online.

5.9.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	0	0	0	0	0	0
Vista y controlador	0	60	0	0	0	0	0

Modelo	0	10	0	0	0	0	0
Test	0	30	0	0	0	0	0
TOTAL	0	100	0	0	0	0	0



5.9.2. Gemas utilizadas

jquery-rails, versión 4.0.5: Biblioteca jQuery.

jquery-ui-rails, versión 4.2.1: Animaciones de jQuery.

5.9.3. Ficheros de migración

No se han añadido ficheros de migración en este sprint.

5.9.4. Modelos ORM

No se han añadidos ficheros de modelos en este sprint.

5.9.5. Vistas y controladores

Para el RSS, añadimos el método ***rss*** al controlador **app/controllers/catalog_controller.rb** y creamos una nueva vista:

app/controllers/catalog_controller.rb

```
# coding: utf-8
class CatalogController < ApplicationController
  before_action :initialize_cart, :except => :show
  #before_action :require_no_user
  def show
        @film = Film.find(params[:id])
        @page_title = @film.title
  end
  def index
        @films = Film.order("films.id desc").includes(:directors, :producer).paginate(:page
=> params[:page], :per_page => 5)
        @page_title = 'Catálogo'
  end
  def latest
        @films = Film.latest 5 # invoques "latest" method to get the five latest films
        @page_title = 'Últimas películas'
  end

  def rss
        latest
        render :layout => false
        response.headers["Content-Type"] = "application/xml; version=1.0; charset=utf-8"
  end
end
```

app/views/catalog/rss.builder

```
xml.instruct!
xml.rss "version" => "2.0", "xmlns:dc" => "http://purl.org/dc/elements/1.1/" do
  xml.channel do
        xml.title @page_title
        xml.description "Esiflix: Tu tienda on-line de películas"
        xml.link url_for :action => 'index', :only_path => false
        xml.language "en-us"
        xml.ttl "60"

        for film in @films do
        xml.item do
        xml.title film.title
        xml.description "#{film.title} by #{film.director_names}"
        xml.link url_for :action => "show", :id => film, :only_path => false
        xml.guid url_for :action => "show", :id => film, :only_path => false
        xml.pubDate film.created_at.to_s :long
        xml.author film.director_names
```

```
        end
        end
  end
end
```

Para Ajax, modificamos el controlador **app/controllers/cart_controller.rb** y las vistas **app/views/cart/_cart.html.erb**, **app/views/cart/_item.html.erb** y **app/views/catalog/_films.html.erb**, y creamos tres nuevas vistas:

app/controllers/cart_controller.rb

```
# coding: utf-8
class CartController < ApplicationController
  before_action :initialize_cart
  def add
    @film = Film.find params[:id]
    @page_title = 'Añadir artículo'
    if request.post?
      @item = @cart.add params[:id]
      flash[:cart_notice] = "Añadido <em>#{@item.film.title}</em>."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'add', :template => 'cart/add'
    respond_to do |format|
      format.js { @item = @cart.add params[:id]
        flash.now[:cart_notice] = "Añadido <em>#{@item.film.title}</em>"
        render :controller => 'cart', :action => 'add_with_ajax' }
      format.html { if request.post?
        @item = @cart.add params[:id]
        flash[:cart_notice] = "Añadido <em>#{@item.film.title}</em>."
        redirect_to :controller => 'catalog'
      else
        render :controller => 'cart', :action => 'add', :template => 'cart/add'
      end }
    end
  end


  def remove
    @film = Film.find params[:id]
    @page_title = 'Eliminar artículo'
    if request.post?
      @item = @cart.remove params[:id]
      flash[:cart_notice] = "Eliminado <em>#{@item.film.title}</em>."
```

```
    redirect_to :controller => 'catalog'
   else
    render :controller => 'cart', :action => 'remove', :template => 'cart/remove'
   respond_to do |format|
    format.js { @item = @cart.remove params[:id]
     flash.now[:cart_notice] = "Eliminado <em>#{@item.film.title}</em>"
     render :controller => 'cart', :action => 'remove_with_ajax' }
    format.html { if request.post?
     @item = @cart.remove params[:id]
     flash[:cart_notice] = "Eliminado <em>#{@item.film.title}</em>."
     redirect_to :controller => 'catalog'
    else
     render :controller => 'cart', :action => 'remove', :template => 'cart/remove'
    end }
   end
  end

  def clear
   @page_title = 'Vaciar carrito'
   if request.post?
    @cart.cart_items.destroy_all
    flash[:cart_notice] = "Carrito vaciado."
    redirect_to :controller => 'catalog'
   else
    render :controller => 'cart', :action => 'clear', :template => 'cart/clear'
   respond_to do |format|
    format.js { @cart.cart_items.destroy_all
     flash.now[:cart_notice] = "Carrito vaciado"
     render :controller => 'cart', :action => 'clear_with_ajax' }
    format.html { if request.post?
     @cart.cart_items.destroy_all
     flash[:cart_notice] = "Carrito vaciado."
     redirect_to :controller => 'catalog'
    else
     render :controller => 'cart', :action => 'clear', :template => 'cart/clear'
    end }
   end
  end
end
```

app/views/cart/_cart.html.erb

```
<% if flash[:cart_notice] %>
<%= render :partial => 'cart/cart_notice' %>
<% end %>
```

```
<h3>Su carrito de la compra</h3>
<p>
  <strong>
    <% unless controller.controller_name == 'checkout' %>
      <%= link_to 'Proceder a la facturación', :controller => 'checkout' %>
    <% end %>
  </strong>
</p>
<ul>
  <% for item in @cart.cart_items %>
    <li id="cart_item_<%= item.film.id %>">
      <%= render :partial => 'cart/item', :object => item %>
    </li>
  <% end %>
</ul>
<p id='cart_total'><strong>Total: <%= sprintf "%0.2f €", @cart.total %></strong></p>
<% unless @cart.cart_items.empty? %>
  <p id='clear_cart_link'>
  <b><%= link_to 'Vaciar carrito', :controller => 'cart', :action => 'clear' %></b>
  <b><%= link_to 'Vaciar carrito', :controller => 'cart', :action => 'clear', :remote => true
%></b>
  </p>
<% end %>
```

app/views/cart/_item.html.erb

```
<%= link_to item.film.title, :action => 'show', :controller => 'catalog', :id => item.film.id %>

<%= pluralize item.amount, "película", "películas" %>, <%= sprintf "%0.2f €", item.price *
item.amount %>

(<%= link_to '<b>-</b>'.html_safe, :controller => 'cart', :action => 'remove', :id => item.film
%>)
(<%= link_to '<b>-</b>'.html_safe, :controller => 'cart', :action => 'remove', :id => item.film,
:remote => true %>)
```

app/views/catalog/_films.html.erb

```
<dl id = 'films'>
  <% for film in @films %>
    <dt>
      <%= link_to film.title, :action => 'show', :id => film %>
      <%= link_to '+', :controller => 'cart', :action => 'add', :id => film %>
      <%= link_to '+', :controller => 'cart', :action => 'add', :id => film, :remote => true %>
```

```
    </dt>
    <% for director in film.directors %>
      <dd><%= director.last_name %>, <%= director.first_name %></dd>
    <% end %>
    <dd><%= sprintf("Duración: %s. Precio: %.2f €", pluralize(film.duration, "minuto"),
film.price) %> </dd>
    <dd><small>Productora: <%= film.producer.name %></small></dd>
  <% end %>
</dl>
```

app/views/cart/add_with_ajax.js.erb

```
jQuery.noConflict ();
jQuery('#shopping_cart').html("<%=j render :partial => 'cart/cart' %>");
jQuery("#cart_item_<%=@item.film.id
%>").css({'background-color':'#7ef'}).animate({'background-color':'#def'}, 3000);
jQuery('#cart_notice').hide('fade',{}, 3000);
```

app/views/cart/clear_with_ajax.js.erb

```
jQuery.noConflict();
jQuery('#shopping_cart').html("<%=j render :partial => 'cart/cart' %>");
jQuery('#cart_notice').hide('fade',{},3000);
```

app/views/cart/remove_with_ajax.js.erb

```
jQuery.noConflict();
jQuery('#shopping_cart').html("<%=j render :partial => 'cart/cart' %>");
jQuery ("#cart_item_<%= @item.film.id %>").css({'background-color':'#7ef'}). animate
({'background-color':'#def'},
3000);
jQuery("#cart_item_<%= @item.film.id %>a"
).css({'background-color':'#7ef'}).animate({'background-color':'#def'},3000);
jQuery('#cart_notice').hide('fade',{},3000);
```

5.9.6. Test

Añadimos el test ***read_rss*** a ***test/integration/browsing_and_searching_test.rb***:

test/integration/browsing_and_searching_test.rb

```
require 'test_helper'
```

```
class BrowsingAndSearchingTest < ActionDispatch::IntegrationTest
 fixtures :producers, :directors, :films, :directors_films
 test "browse" do
        jill = new_session_as :jill
        jill.index
        jill.second_page
        jill.film_details 'Pride and Prejudice'
        jill.latest_films
        jill.reads_rss
 end

 module BrowsingTestDSL
        include ERB::Util
        attr_writer :name
        def index
        get '/catalog/index'
        assert_response :success
        assert_select 'dl#films' do
        assert_select 'dt', :count => 5
        end
        assert_select 'dt' do
        assert_select 'a', 'The Idiot'
        end
        check_film_links
        end
        def second_page
        get '/catalog/index?page=2'
        assert_response :success
        assert_template 'catalog/index'
        assert_equal Film.find_by_title('Pro Rails E-Commerce'),
                assigns(:films).last
        check_film_links
        end
        def film_details(title)
        @film = Film.where(:title => title).first
        get "/catalog/show/#{@film.id}"
        assert_response :success
        assert_template 'catalog/show'
        assert_select 'div#content' do
        assert_select 'h1', @film.title
        assert_select 'h2', "por #{@film.directors.map{|a| a.name}.join(", ")}"
        end
        end
        def latest_films
        get '/catalog/latest'
        assert_response :success
        assert_template 'catalog/latest'
```

```
        assert_select 'dl#films' do
        assert_select 'dt', :count => 5
        end
        @films = Film.latest(5)
        @films.each do |a|
        assert_select 'dt' do
        assert_select 'a', a.title
        end
        end
        end
  end
  def check_film_links
        for film in assigns :films
        assert_select 'a' do
        assert_select '[href=?]', "/catalog/show/#{film.id}"
        end
        end
  end
  def new_session_as(name)
        open_session do |session|
        session.extend(BrowsingTestDSL)
        session.name = name
        yield session if block_given?
        end
  end

  def reads_rss
        get "/catalog/rss"
        assert_response :success
        assert_template "catalog/rss"
        assert_match "application/xml", response.headers["Content-Type"]

        assert_select 'channel' do
        assert_select 'item', :count => 5
        end

        @films = Film.latest(5)
        @films.each do |film|
        assert_select 'item' do
        assert_select 'title', film.title
        end
        end
  end
end
```

Modificamos **test/controllers/cart_controller_test.rb**:

test/controllers/cart_controller_test.rb

```
require 'test_helper'
class CartControllerTest < ActionDispatch::IntegrationTest
  fixtures :directors, :producers, :films
  test "add" do
    assert_difference(CartItem, :count) do
      post '/cart/add', :params => { :id => 4 }
    end
    assert_response :redirect
    assert_redirected_to :controller => 'catalog'
    assert_equal 1, Cart.find(@request.session[:cart_id]).cart_items.size
  end

  test "add_xhr" do
    assert_difference(CartItem, :count) do
      post '/cart/add', :params => { :id => 4 }, :xhr => true
    end
    assert_response :success
    assert_select_jquery :html, '#shopping_cart' do
      assert_select 'li#cart_item_4', /Pro Rails E-Commerce 4th Edition/
    end
    assert_equal 1, Cart.find(@request.session[:cart_id]).cart_items.size
  end

  test "remove" do
    post '/cart/add', :params => { :id => 4 }
    assert_equal [Film.find(4)], Cart.find(@request.session[:cart_id]).films
    post '/cart/remove', :params => { :id => 4 }
    assert_equal [], Cart.find(@request.session[:cart_id]).films
  end

  test "remove_xhr" do
    assert_difference(CartItem, :count) do
      post '/cart/add', :params => { :id => 4 }, :xhr => true
    end

    assert_response :success
    post '/cart/remove', :params => { :id => 4 }, :xhr => true
    assert_select_jquery :html, '#shopping_cart' do
      assert_select 'li#cart_item_4', false
    end
  end

  test "clear" do
```

```
    post '/cart/add', :params => { :id => 4 }
    assert_equal [Film.find(4)], Cart.find(@request.session[:cart_id]).films
    post '/cart/clear'
    assert_response :redirect
    assert_redirected_to :controller => 'catalog'
    assert_equal [], Cart.find(@request.session[:cart_id]).films
  end

  test "clear_xhr" do
    assert_difference(CartItem, :count) do
      post '/cart/add', :params => { :id => 4 }, :xhr => true
    end

    assert_response :success
    post '/cart/clear', :xhr => true
    assert_select_jquery :html, '#shopping_cart' do
      assert_select 'li#cart_item_4', false
    end
  end
end
```

5.9.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.9.8. Objetivos alcanzados

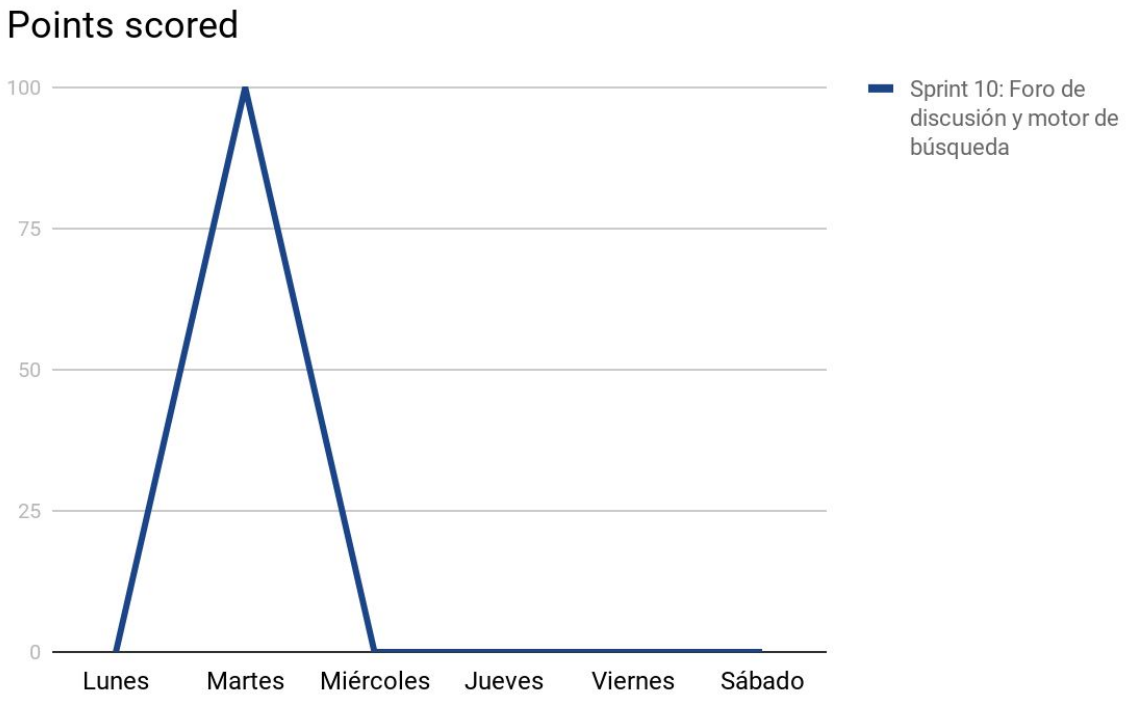
Se han añadido funcionalidades de AJAX y RSS que funcionan correctamente y hacen que nuestra tienda sea más atractiva para el usuario.

5.4 Sprint 10 (Foro de discusión y motor de búsqueda)

5.10.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	0	0	0	0	0	0
Vista y controlador	0	60	0	0	0	0	0
Modelo	0	10	0	0	0	0	0

Test	0	30	0	0	0	0	0
TOTAL	0	100	0	0	0	0	0



5.10.2. Gemas utilizadas

No se añaden nuevas gemas en esta iteración.

5.10.3. Ficheros de migración

Se ha tenido que generar un chero de migración la estructura del foro y sus mensajes:

/eshop5/db/migrate/20190522181101_create_forum_posts.rb

```
class CreateForumPosts < ActiveRecord::Migration[5.1]
  def change
    create_table :forum_posts do |table|
      table.column :name, :string, :limit => 50, :null => false
      table.column :subject, :string, :limit => 255, :null => false
      table.column :body, :text

      table.column :root_id, :integer, :limit => 8, :null => false,
:default => 0
      table.column :parent_id, :integer, :limit => 8, :null => false,
```

```
:default => 0
      table.column :depth, :integer, :null => false, :default => 0

      table.timestamps
    end
  end
  def self.down
    drop_table :forum_posts
  end
end
```

5.10.4. Modelos ORM

Luego hemos generado el modelo ForumPost:

/eshop5/app/models/forum_post.rb

```
class ForumPost < ApplicationRecord
  validates_length_of :name, :within => 2..50, :message => "El nombre
debe tener entre 2 y 50 caracteres."
  validates_length_of :subject, :within => 5..255, :message => "El
asunto debe contener entre 5 y 255 caracteres."
  validates_length_of :body, :within => 5..5000, :message => "El
cuerpo del post debe estar entre 5 y 5000 caracteres."
end
```

5.10.5. Vistas y controladores

A continuación Ficheros del controlador y las vistas del foro:

/eshop5/app/controllers/forum_controller.rb

```
class ForumController < ApplicationController
  #before_action :require_user, :only => :destroy

  def post
    @post = ForumPost.new(:parent_id => 0, :root_id => 0, :depth =>
0)
    @page_title = 'Publicar un post'
  end
```

```
  def create
    @post = ForumPost.new(post_params)
    if @post.save
      @post.root_id = @post.id if @post.root_id == 0
      @post.save
      flash[:notice] = 'Post creado correctamente.'
      redirect_to :action => 'index'
    else
      if @post.parent_id == 0
        @page_title = 'Publicar un post'
      else
        @page_title = "Responder a
'#{ForumPost.find(@post.parent_id).subject}'"
      end
      render :action => 'post'
    end
  end

  def reply
    reply_to = ForumPost.find(params[:id])
    @post = ForumPost.new(:parent_id => reply_to.id, :root_id =>
reply_to.root_id,
                          :depth => reply_to.depth + 1)
    @page_title = "Responder a '#{reply_to.subject}'"
    render :action => 'post'
  end

  def destroy
    @post = ForumPost.find(params[:id])
    @posts = ForumPost.where(root_id: @post.root_id)
    @posts.each do |post|
      post.destroy
    end
    flash[:notice] = "El post '#{@post.subject}' fue borrado al
completo correctamente."
    redirect_to :action => 'index'
  end

  def show
    @post = ForumPost.find(params[:id])
    @page_title = "'#{@post.subject}'"
  end

  def index
    @posts = ForumPost.order('root_id desc,
created_at').paginate(:page => params[:page], :per_page => 10)
```

```
    @page_title = 'Foro'
  end

  private
    def post_params
      params.require(:post).permit(:name, :subject, :body, :root_id,
:parent_id, :lft, :rgt, :depth)
    end
end
```

/eshop5/app/views/forum/_form.html.erb

```
<% if @post.errors.any? %>
  <div id="errorExplanation">
  <h2>Este post no puede ser guardado debido a <%=
pluralize(@post.errors.count, "error", "errores")%></h2>
    <ul>
      <% @post.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <%= hidden_field :post, :root_id %>
</div>

<div class="field">
  <%= hidden_field :post, :parent_id %>
</div>

<div class="field">
  <%= hidden_field :post, :depth %>
</div>

<div class="field">
  <p><label for="post_name">Nombre</label><br/>
  <%= text_field 'post', 'name' %></p>
</div>

<div class="field">
```

```
  <p><label for="post_subject">Tema</label><br/>
  <%= text_field 'post', 'subject' %></p>
</div>

<div class="field">
  <p><label for="post_body">Cuerpo</label><br/>
  <%= text_area 'post', 'body' %></p>
</div>
```

/eshop5/app/views/forum/index.html.erb

```
<% if @posts.size > 0 %>
  <div><%= link_to 'Nuevo post', :action => 'post' %></div>
  <p></p>
  <table><%= display_like_a_tree @posts %>
  </table>
  <p></p>
<% else %>
  <p>No hay posts todavía. <%= link_to 'Sé el primero en poner un
post.', :action => 'post' %></p>
<% end %>

<%= will_paginate @posts, :page_links => false, :link_separator => '
| ',
                          :previous_label => 'Página anterior',
:next_label => 'Página siguiente' %>
```

/eshop5/app/views/forum/post.html.erb

```
<%= form_tag :action => 'create' do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Crear post' %>
<% end %>

<%= link_to 'Atrás', :action => 'index' %>
```

/eshop5/app/views/forum/show.html.erb

```
<dl>
  <dt>Nombre</dt>
  <dd><%= h @post.name %></dd>
  <dt>Tema</dt>
  <dd><%= h @post.subject %></dd>
  <dt>Cuerpo</dt>
  <dd><%= h @post.body %></dd>
</dl>

<%= link_to 'Responder', :action => 'reply', :id => @post %> |
<%= link_to 'Atrás', :action => 'index' %>
```

...y creamos el siguiente helper:

/eshop5/app/helpers/forum_helper.rb

```
# coding: utf-8
module ForumHelper
  def display_like_a_tree(posts)
    content = ''
    for post in posts
      url = link_to "#{h post.subject}", {:action => 'show', :id =>
post.id}
      button = button_to "Borrar", {:action => 'destroy', :method =>
'post', :id => post}, data: {confirm: "¿Está seguro de que quiere
borrar el post #{post.subject}?"}
      margin_left = post.depth*30
      content << %(
      <tr><td><div style ="margin-left:#{margin_left}px">
      #{url} by #{h post.name} &middot; #{post.created_at.strftime
"%H:%M:%S %d-%m-%Y"}</div></td>)
      content << %(<td>#{button}</td>) unless post.parent_id != 0
      content  << %(</tr>)
    end
    content.html_safe
  end
end
```

...ya tenemos todo lo necesario para tener un foro en nuestra web.

Vamos a detallar los cheros que tenemos que modificar para tener un motor de búsqueda en el catálogo de nuestra web:

Añadimos al controlador de catálogo:

/eshop5/app/controllers/catalog_controller.rb

```
# coding: utf-8
class CatalogController < ApplicationController
  before_action :initialize_cart, :except => :show
  #before_action :require_no_user

  def show
    @film = Film.find(params[:id])
    @page_title = @film.title
  end

  def index
    @films = Film.order("films.id desc").includes(:directors,
:producer).paginate(:page => params[:page], :per_page => 5)
    @page_title = 'Catálogo'
  end

  def latest
    @films = Film.latest 5 # invoques "latest" method to get the five
latest films
    @page_title = 'Últimas películas'
  end

  def rss
    latest
    render :layout => false
    response.headers["Content-Type"] = "application/xml; version=1.0;
charset=utf-8"
  end

  def search
    @page_title = "Buscar"
    if params[:commit] == "Buscar" || params[:q]
      @films = Film.where 'title LIKE ?', "%#{params[:q]}%"
      unless @films.size > 0
        flash.now[:notice] = "No se ha encontrado la película."
      end
    end
  end
end
```

Añadimos también a app/views/catalog/index.html.erb

/eshop5/app/views/catalog/index.html.erb

```
<%= render :partial => 'films' %>
<%= will_paginate @films, :page_links => false, :link_separator => '
| ',
                            :previous_label => 'Página anterior',
:next_label => 'Página siguiente' %>
<p><%= link_to 'Últimos articulos', :action => 'latest' %> | <%=
link_to 'RSS', :action => 'rss' %> | <%= link_to 'Buscar', :action =>
'search' %></p>
```

Por último creamos las siguientes vistas:

/eshop5/app/views/catalog/_search_box.html.erb

```
<%= form_tag({:action => 'search'},{:method => 'get'}) %>
<%= text_field_tag :q %>
<%= submit_tag 'Buscar' %>
</form>
```

/eshop5/app/views/catalog/search.html.erb

```
<p>
  Introduzca el título (o parte del título) de la película que quiere
buscar.
</p>

<%= render :partial => 'search_box' %>

<% if @films %>
    <p>
      Resultado de la búsqueda '<%= params[:q] %>':
      <%= pluralize @films.size, 'resultado', 'resultados' %>
    </p>
    <%= render(:partial => 'films') %>
<% end %>
```

Ya tenemos por tanto disponible un motor de búsqueda dentro de nuestro catálogo de artículos.

5.10.6. Test

Para el foro hemos realizado una prueba de integración:

/eshop5/test/integration/forum_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'

class ForumTest < ActionDispatch::IntegrationTest

  test "forum" do
    jill = new_session_as(:jill)
    george = new_session_as(:george)
    post = jill.post_to_forum :post => {
      :name => 'Bookworm',
      :subject => 'Downtime',
      :body => 'Emporium is down again!'
    }
    george.view_forum
    jill.view_forum
    george.view_post post
    george.reply_to_post(post, :post => {
      :name => 'George',
      :subject => 'Rats!',
      :body => 'Rats!!!!!!!!'
    })
    george.delete_post(post)
  end

  private

  module ForumTestDSL
    attr_writer :name

    def post_to_forum(parameters)
      get '/forum/post'
      assert_response :success
      assert_template 'forum/post'
      post '/forum/create', :params => parameters
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'forum/index'
      return ForumPost.find_by_subject(parameters[:post][:subject])
    end
```

```
    def view_forum
      get '/forum'
      assert_response :success
      assert_template 'forum/index'
      assert_select 'div#content' do
        assert_select 'h1', 'Foro'
        assert_select 'a', 'Nuevo post'
      end
    end

    def view_post(post)
      get "/forum/show/?id=#{post.id}"
      assert_response :success
      assert_template 'forum/show'
      assert_select 'div#content' do
        assert_select 'h1', "\'#{post.subject}\'"
      end
    end

    def reply_to_post(post, parameters)
      get "/forum/reply/?id=#{post.id}"
      assert_response :success
      assert_select 'div#content' do
        assert_select 'h1', "Responder a \'#{post.subject}\'"
      end
      post '/forum/create', :params => { :post => { :name =>
parameters[:post][:name],
                                                  :subject =>
parameters[:post][:subject],
                                                  :body =>
parameters[:post][:body], :parent_id => post.id }}
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'forum/index'
      assert_select 'table' do
        assert_select 'a', parameters[:post][:subject]
      end
    end

    def delete_post(post)
      num_posts = ForumPost.count
      post "/forum/destroy/?id=#{post.id}"
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'forum/index'
```

```
      assert_select 'div#notice', "El post \'#{post.subject}\' fue
borrado al completo correctamente."
      assert_equal ForumPost.count, num_posts - 1
    end
  end

  def new_session_as(name)
    open_session do |session|
      session.extend(ForumTestDSL)
      session.name = name
      yield session if block_given?
    end
  end
end
```

5.10.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

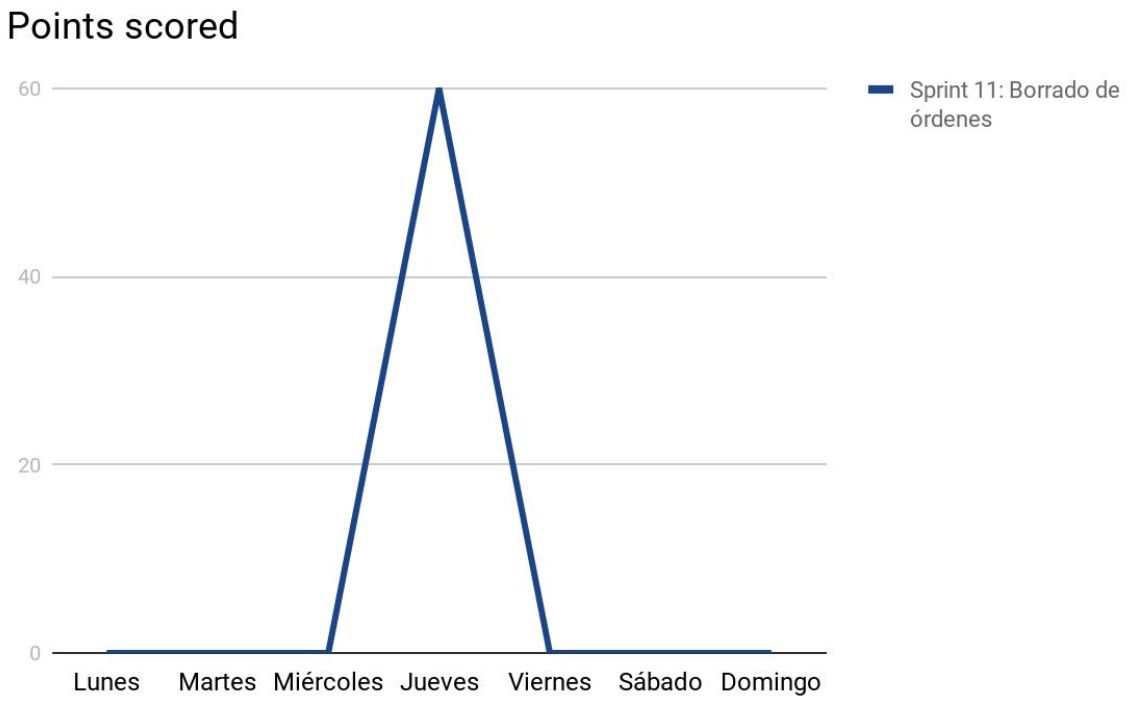
5.10.8. Objetivos alcanzados

Se han añadido el motor de búsqueda y el foro de discusión con lo que los usuarios podrán buscar sus películas favoritas y comunicarse entre ellos a través del foro.

5.4 Sprint 11 (Restauración de contraseña y borrado de órdenes)

5.11.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	0	0	0	0	0	0
Vista y controlador	0	0	0	40	0	0	0
Modelo	0	0	0	10	0	0	0
Test	0	0	0	10	0	0	0
TOTAL	0	0	0	60	0	0	0



5.11.2. Gemas utilizadas

No se añaden nuevas gemas en esta iteración.

5.11.3. Ficheros de migración

No ha sido necesario.

5.11.4. Modelos ORM

No ha sido necesario.

5.11.5. Vistas y controladores

Añadimos el método destroy a app/controllers/admin/order_controller.rb

/eshop5/app/controllers/admin/order_controller.rb

```
class Admin::OrderController < Admin::AuthenticatedController
  def close
    order = Order.find(params[:id])
    order.close
    flash[:notice] = "El pedido ##{order.id} ha sido cerrado."
    redirect_to :action => 'index'
  end
```

```
  def show
    @order = Order.find(params[:id])
    @page_title = "Mostrando pedido ##{@order.id}"
  end

  def index
    @status = params[:id]
    if @status.blank?
      conditions = nil
      @status = 'all'
      @page_title = 'Listando todos los pedidos'
    else
      conditions = "status = '#{@status}'"
      @estado = 'abierto' if @status == 'open'
      @estado = 'procesado' if @status == 'processed'
      @estado = 'cerrado' if @status == 'closed'
      @estado = 'fallido' if @status == 'failed'
      @page_title = "Listando pedidos #{@estado.pluralize}"
    end
    @orders = Order.where(conditions).paginate(:page =>
params[:page], :per_page => 10)
  end

  def destroy
    @order = Order.find(params[:id])
    @order.destroy
    flash[:notice] = "La orden ##{@order.id} fue borrada
correctamente."
    redirect_to :action => 'index'
  end
end
```

y añadimos las líneas en negrita a la vista index de órdenes:

/eshop5/app/views/admin/order/index.html.erb

```
<% if @orders == [] %>
  <% if @status == 'all' %>
    <h2><%= "No hay pedidos." %></h2>
  <% else %>
    <h2><%= "No hay pedidos con el estado '#{@estado}'." %></h2>
  <% end %>
<% else %>
  <table>
```

```
    <tr>
      <th>ID</th>
      <th>Estado</th>
      <th>Precio total</th>
      <th>Cantidad</th>
      <th>Creado el</th>
      <th>Actualizado el</th>
      <th></th>
    </tr>
  <% for order in @orders %>
    <% estado = 'procesado' if order.status == 'processed'
       estado = 'fallido' if order.status == 'failed'
       estado = 'abierto' if order.status == 'open'
       estado = 'cerrado' if order.status == 'closed' %>
    <tr>
      <td align="center"><%= order.id %></td>
      <td align="center"><%= estado.capitalize %></td>
      <td align="center"><%= order.total %></td>
      <td align="center"><%= order.amount %></td>
      <td align="center"><%= order.created_at.strftime("%d-%m-%Y
%H:%M") %></td>
      <td align="center"><%= order.updated_at.strftime("%d-%m-%Y
%H:%M") %></td>
      <td><%= link_to 'Mostrar', :action => 'show', :id => order
%></td>
      <td><%= button_to 'Borrar', {:action => 'destroy', :id =>
order},
     data: {confirm: "¿Estás seguro? La orden ##{order.id} se
borrará para siempre."}%></td>
    </tr>
  <% end %>
  </table>

  <% if @orders.total_pages > 1 %>
    <br/>
    <%= 'Ver página:' %>
  <% end %>

  <%= will_paginate @orders, :page_links => true, :link_separator =>
' ', :container => false,
                              :previous_label => '', :next_label => ''
%>
  <p></p>
<% end %>

<%= render :partial => 'navigation' %>
```

Para la restauración de la contraseña primero hacemos el controlador:

/eshop5/app/controllers/password_reset_controller.rb

```
# coding: utf-8
class PasswordResetController < ApplicationController
  before_action :require_no_user

  def new
    @page_title = 'Restaurar contraseña'
  end

  def create
    @user = User.find_by_email(params[:email])
    if @user
      @user.reset_perishable_token!
      NotifierMailer.password_reset_instructions(@user).deliver_now
      flash[:notice] = 'Las instrucciones para restaurar su
contraseña se les han sido enviadas. ' +
                       'Por favor, compruebe su email.'
      redirect_to :controller => 'user_sessions', :action => 'new'
    else
      @page_title = 'Restaurar contraseña'
      flash[:notice] = 'No se ha encontrado ningún usuario con ese
email.'
      render :action => :new
    end
  end

  def edit
    load_user_using_perishable_token
    @page_title = 'Editar contraseña'
  end

  def update
    @user = User.find_by_id(params[:user][:id])
    @user.password = params[:user][:password]
    @user.password_confirmation =
params[:user][:password_confirmation]
    if @user.save
      flash[:notice] = 'Contraseña actualizada correctamente. Usuario
identificado.'
      redirect_to :controller => 'user', :action => 'show'
    else
      @page_title = 'Editar contraseña'
      render :action => :edit
```

```
    end
  end

  private
    def load_user_using_perishable_token
      @user = User.find_using_perishable_token(params[:id])
      unless @user
        flash[:notice] = 'Su cuenta no ha sido localizada. ' +
                         'Si no puedes usar directamente la url del
email, ' +
                         'prueba a copiarla y pegarla en su navegador
o ' +
                         'reinicia el proceso de restauración de
contraseña.'
        redirect_to :controller => 'user_sessions', :action => 'new'
      end
    end
end
```

Y las vistas:

/eshop5/app/views/password_reset/_form.html.erb

```
<% if @user.errors.any? %>
  <div id="errorExplanation">
  <h2>Esta cuenta no puede ser creada debido a <%=
pluralize(@user.errors.count, "error", "errores")%></h2>
    <ul>
      <% @user.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

<div class="field">
  <%= hidden_field :user, :id %>
</div>

<div class="field">
  <p><label for="user_password">Contraseña</label><br/>
  <%= password_field 'user', 'password' %></p>
</div>

<div class="field">
```

```
  <p><label for="user_password_confirmation">Confirmar
contraseña</label><br/>
  <%= password_field 'user', 'password_confirmation' %></p>
</div>
```

/eshop5/app/views/password_reset/edit.html.erb

```
<%= form_tag :action => 'update', :id => @user do %>
  <%= render :partial => 'form' %>
  <%= submit_tag 'Actualizar contraseña' %>
<% end %>
```

/eshop5/app/views/password_reset/new.html.erb

```
Rellene el siguiente formulario y se les enviarán las instrucciones
para restaurar su contraseña:

<%= form_tag :action => 'create' do %>
  <div class="field">
    <p><label for="email">Correo electrónico</label><br/>
    <%= text_field_tag 'email' %></p>
  </div>
  <%= submit_tag 'Enviar correo electrónico' %>
<% end %>

<%= link_to 'Catálogo', :controller => 'catalog' %>
```

5.11.6. Test

Se ha desarrollado un test para comprobar la contraseña:

/eshop5/test/integration/password_reset_test.rb

```
# coding: utf-8
require File.dirname(__FILE__) + '/../test_helper'

class PasswordResetTest < ActionDispatch::IntegrationTest

  def setup
```

```
    User.create(:name => 'George Smith', :login => 'george', :email
=> 'george@emporium.com',
                :password => 'cheetah', :password_confirmation =>
'cheetah')
  end

  test "forgot_password" do
    george = new_session_as(:george)
    george.goes_to_forgot_password_form
    george.sends_mistaken_email(:email => 'george@bookshop.com')
    george.sends_correct_email(:email => 'george@emporium.com')
    george.resets_mismatched_password(:user => { :id =>
User.first.id, :password => 'gold',

:password_confirmation => 'silver' })
    george.resets_correct_password(:user => { :id => User.first.id,
:password => 'gold',
                                             :password_confirmation
=> 'gold' })
  end

  private

  module BrowsingTestDSL
    include ERB::Util
    attr_writer :name

    def goes_to_forgot_password_form
      get '/password_reset/new'
      assert_response :success
      assert_template 'password_reset/new'
      assert_select 'div#content' do
        assert_select 'h1', 'Restaurar contraseña'
        assert_select "input#email"
      end
    end

    def sends_mistaken_email(email)
      post_email(email)
      assert_response :success
      assert_template 'password_reset/new'
      assert_select 'div#content' do
        assert_select 'h1', 'Restaurar contraseña'
      end
      assert_equal flash[:notice], 'No se ha encontrado ningún
usuario con ese email.'
      assert_select 'div#notice', 'No se ha encontrado ningún usuario
```

```
con ese email.'
    end

    def sends_correct_email(email)
      post_email(email)
      mail = ActionMailer::Base.deliveries.last
      assert_equal ['george@emporium.com'], mail.to
      assert_equal 'noreply@esiflix.es', mail[:from].value
      assert_equal 'Instrucciones para restaurar contraseña',
mail.subject
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'user_sessions/new'
      assert_select 'div#content' do
        assert_select 'h1', 'Iniciar sesión'
      end
      assert_equal flash[:notice], 'Las instrucciones para restaurar
su contraseña se les han sido enviadas. Por favor, compruebe su
email.'
      assert_select 'div#notice', 'Las instrucciones para restaurar
su contraseña se les han sido enviadas. Por favor, compruebe su
email.'
      assert_select 'input#user_session_login'
    end

    def resets_mismatched_password(parameters)
      user = User.find_by_id(parameters[:user][:id])
      edit_password_reset_url = url_for :controller =>
'password_reset', :action => 'edit'
      edit_password_reset_url += "?id=#{user.perishable_token}"
      get edit_password_reset_url
      assert_response :success
      assert_template 'password_reset/edit'
      assert_select 'div#content' do
        assert_select 'h1', 'Editar contraseña'
        assert_select 'input#user_password'
        assert_select 'input#user_password_confirmation'
      end
      post '/password_reset/update', :params => parameters
      assert_response :success
      assert_template 'password_reset/edit'
      assert_select 'div#content' do
        assert_select 'h1', 'Editar contraseña'
        assert_select 'h2', 'Esta cuenta no puede ser creada debido a
1 error'
        assert_select 'li', "Password confirmation doesn\'t match
```

```
Password"
        assert_select 'input#user_password'
        assert_select "div[class=\"field_with_errors\"]"
      end
    end

    def resets_correct_password(parameters)
      user = User.find_by_id(parameters[:user][:id])
      edit_password_reset_url = url_for :controller =>
'password_reset', :action => 'edit'
      edit_password_reset_url += "?id=#{user.perishable_token}"
      get edit_password_reset_url
      assert_response :success
      assert_template 'password_reset/edit'
      assert_select 'div#content' do
        assert_select 'h1', 'Editar contraseña'
        assert_select 'input#user_password'
        assert_select 'input#user_password_confirmation'
      end
      post '/password_reset/update', :params => parameters
      assert_response :redirect
      follow_redirect!
      assert_response :success
      assert_template 'user/show'
      assert_equal flash[:notice], 'Contraseña actualizada
correctamente. Usuario identificado.'
      assert_select 'div#content' do
        assert_select 'h1', user.name
        assert_select 'div#notice', 'Contraseña actualizada
correctamente. Usuario identificado.'
        assert_select 'dt', 'Nombre'
        assert_select 'dd', user.login
      end
    end

    def post_email(email)
      post '/password_reset/create', :params => email
    end
  end

  def new_session_as(name)
    open_session do |session|
      session.extend(BrowsingTestDSL)
      session.name = name
      yield session if block_given?
    end
  end
```

```
end
```

5.11.7. Dificultades encontradas

No hemos tenido problemas relevantes en la realización de las tareas de este sprint.

5.11.8. Objetivos alcanzados

Hemos conseguido que el administrador del sitio pueda borrar las órdenes que él desee y que los usuarios puedan restablecer su contraseña cuando se olviden de ella.

6. Equipo

En este punto vamos a desglosar los componentes del equipo, con sus respectivas funciones dentro del equipo de trabajo, así como, el entorno tecnológico utilizado por cada uno de ellos.

6.1 Componentes

- **Santiago Godoy Poce**
  - Coordinador
  - Analista-desarrollador
  - Contribución al proyecto:
    - Planificación del trabajo en grupo.
    - División de tareas.
    - Desarrollo de los controladores y modelos.

- **Fanny Chunyan Vela Diaz**
  - Desarrolladora
  - Documentación
  - Contribución al proyecto:
    - Elaboración del documento del proyecto.
    - Desarrollo de las vistas y funcionalidad de las mismas.
    - Colaboración en el diseño de los controladores

- **José Correro Barquín**
  - Ingeniero de pruebas
  - Documentación
  - Contribución al proyecto:
    - Desarrollo de los test de unidad y de integración.
    - Elaboración del documento del proyecto.
    - Colaboración en el diseño de los modelos.

- **Isabel del Pilar Durán Chumillas**
  - Desarrolladora
  - Documentación
  - Contribución al proyecto:
    - Elaboración del documento del proyecto.

        - Colaboración en el diseño y estructura de la base de datos.
        - Identificación de los principales requisitos de usuarios.

- **José Manuel Periñán Freire**
    - Desarrollador
    - Administrador de base de datos
    - Contribución al proyecto:
        - Elaboración del documento del proyecto.
        - Diseño y verificación de la estructura de la base de datos.
        - Colaboración en el diseño de los controladores y las vistas.

6.2 Entorno tecnológico

El desarrollo de la aplicación de la aplicación Esiflix se ha realizado en el siguiente entorno:

- **Portátil Asus con las siguientes especificaciones técnicas:**
    - Procesador: Intel Core i5, 7ª generación
    - Memoria: 8gb de RAM
    - Sistema Operativo: Ubuntu, Linux 18.04 LTS

- **Portátil Dell con las siguientes especificaciones técnicas:**
    - Procesador: Intel Core i7, 8ª generación
    - Memoria: 16gb de RAM
    - SIstema Operativo: Antergos 5.0.5-arch

- **Portátil Lenovo con las siguientes especificaciones técnicas:**
    - Procesador: Intel Core i7 7º generación
    - Memoria: 8GB de RAM
    - Sistema Operativo: Ubuntu, Linux 18.04 LTS

- **Portátil ASUS con las siguientes especificaciones técnicas:**
    - Procesador: Intel Core i5, 4ª generación
    - Memoria: 8gb de RAM
    - Sistema Operativo: Ubuntu, Linux 18.04 LTS

- **Portátil Samsung con las siguientes especificaciones técnicas:**
    - Procesador: Intel Core i5, 2ª generación
    - Memoria: 6gb de RAM
    - Sistema Operativo: Ubuntu, Linux 18.04 LTS

El software utilizado para el desarrollo de la aplicación ha sido el siguiente:

- Ruby on Rails como framework.
    - Versión 2.4.1 de Ruby.
    - Versión 5.4.1 de Rails.
- Mysql y mysql-workbench como gestor de base de datos
    - Versión 5.7 server y client.
- Sublime Text, Emacs y Kate como editores de texto.
- Google documents como herramienta para desarrollar este documento.

7. Conclusiones

Gracias a este proyecto nos hemos familiarizado con el lenguaje de programación Ruby, así como con el framework Ruby on Rails. Hemos aprendido a desarrollar de forma rápida, gratificante y satisfactoria una tienda online, lo cual nos parece muy útil en un tiempo donde está en auge el comercio electrónico, por lo que seguiremos desarrollando nuestros conocimientos con las herramientas empleadas en la asignatura.

8. Referencias

[1] **E. Turban, D. King, J. Lang.**, *Introduction to electronic commerce. Pearson Education, 2011.*

[2] **Adigital**, *Libro blanco del comercio electrónico, libroblanco.adigital.org*

[3] **Aspastore**, J.R., *Al día en comercio electrónico. MC GrawHill, 2001.*

[4] **E. Grafeuille**. *Una introducción a Scrum. Mountain Goat Software, LLC, 2008.*