

Internet y Negocio Electrónico

Departamento de Ingeniería Informática
Universidad de Cádiz
Curso 2016/2017

Informe de desarrollo UCASport

Grupo 6

Martínez Gavira, Andrés
Pérez Peregrino, Luis Fernando
Rodríguez Cárdenas, Antonio José
Ruiz Rondán, Antonio
Vázquez Crespo, José Manuel

9 de junio de 2017

ÍNDICE

1. Listas de requisitos priorizadas del producto (“product backlog”)	7
1.1. Arquitectura Modelo-Vista-Controlador (MVC)	7
1.2. Historias de usuario	7
1.2.1. Fabricante (Manufacturer)	7
1.2.2. Proveedor (Supplier)	7
1.2.3. Artículo (Article)	8
1.2.3. Catálogo (Catalog)	8
1.2.4. Carrito de la Compra (Cart y CartItem)	8
1.2.5. Seguridad (User y UserSession)	9
1.2.6. Facturación (Order y OrderItem)	9
1.2.7. Foro de discusión (ForumPost)	9
2. Planificación del proyecto	11
2.1. Introducción	11
2.2. Calendario	11
3. Esquema de la base de datos	13
3.1. Diagrama E/R-Inicial	13
3.2. Diagrama E/R con funcionalidad básica	14
3.3. Diagrama ER final	15
4. Contenido del fichero de rutas	17
5. Sprints	19
5.1. Sprint 1 (Generar About)	19
5.1.1. Sprint backlog y diagrama burndown	19
5.1.2. Gemas utilizadas	20
5.1.3. Ficheros de migración	20
5.1.4. Modelos ORM	20
5.1.5. Vistas y controladores	20
5.1.6. Test	22
5.1.7. Dificultades encontradas	22
5.1.8. Objetivos alcanzados	23
5.2. Sprint 2 (Gestión de Fabricantes)	24
5.2.1. Sprint backlog y diagrama burndown	24
5.2.2. Gemas utilizadas	24
5.2.3. Ficheros de migración	25
5.2.4. Modelos ORM	25
5.2.5. Vistas y controladores	25
5.2.6. Test	28

5.2.7. Dificultades encontradas	30
5.2.8. Objetivos alcanzados	30
5.3. Sprint 3 (Gestión de Proveedores)	31
5.3.1. Sprint backlog y diagrama burndown	31
5.3.2. Gemas utilizadas	31
5.3.3. Ficheros de migración	32
5.3.4. Modelos ORM	32
5.3.5. Vistas y controladores	33
5.3.6. Test	35
5.3.7. Dificultades encontradas	37
5.3.8. Objetivos alcanzados	37
5.4 Sprint 4 (Gestión de Artículos)	38
5.4.1. Sprint backlog y diagrama burndown	38
5.4.2. Gemas utilizadas	39
5.4.3. Ficheros de migración	39
5.4.5. Vistas y controladores	41
5.4.6. Test	45
5.4.7. Dificultades encontradas	48
5.4.8. Objetivos alcanzados	48
5.5. Sprint 5 (Creación del catálogo)	49
5.5.1. Sprint backlog y diagrama burndown	49
5.5.2. Gemas utilizadas	49
5.5.3. Ficheros de migración	50
5.5.4. Modelos ORM	50
5.5.5. Vistas y controladores	50
5.5.6. Test	52
5.5.7. Dificultades encontradas	53
5.5.8. Objetivos alcanzados	54
5.6. Sprint 6 (Carrito de la compra)	55
5.6.1. Sprint backlog y diagrama burndown	55
5.6.2. Gemas utilizadas	56
5.6.3. Ficheros de migración	56
5.6.4. Modelos ORM	57
5.6.5. Vistas y controladores	57
5.6.6. Test	60
5.6.7. Dificultades encontradas	61
5.6.8. Objetivos alcanzados	61
5.7. Sprint 7 (Facturación y pedidos)	62
5.7.1. Sprint backlog y diagrama burndown	62
5.7.2. Gemas utilizadas	63
5.7.3. Ficheros de migración	63
5.7.4. Modelos ORM	64

5.7.5. Vistas y controladores	67
5.7.6. Test	72
5.7.7. Dificultades encontradas	75
5.7.8. Objetivos alcanzados	75
5.8. Sprint 8 (Autenticación)	76
5.8.1. Sprint backlog y diagrama burndown	76
5.8.2. Gemas utilizadas	77
5.8.3. Ficheros de migración	77
5.8.4. Modelos ORM	78
5.8.5. Vistas y controladores	78
5.8.6. Test	84
5.8.7. Dificultades encontradas	87
5.8.8. Objetivos alcanzados	88
5.9. Sprint 9 (RSS y AJAX)	89
5.9.1. Sprint backlog y diagrama burndown	89
5.9.2. Gemas utilizadas	90
5.9.3. Ficheros de migración	90
5.9.4. Modelos ORM	90
5.9.5. Vistas y controladores	90
5.9.6. Tests	94
5.9.7. Dificultades encontradas	96
5.9.8. Objetivos alcanzados	97
5.10. Sprint 10 (Foro de discusión y motor de búsqueda)	98
5.10.1. Sprint backlog y diagrama burndown	98
5.10.2. Gemas utilizadas	98
5.10.3. Ficheros de migración	99
5.10.4. Modelos ORM	99
5.10.5. Vistas y Controladores	99
5.10.6. Test	103
5.10.7. Dificultades encontradas	105
5.10.8. Objetivos alcanzados	105
5.11. Sprint 11 (Borrado de órdenes)	106
5.11.1. Sprint backlog y diagrama burndown	106
5.11.2. Gemas utilizadas	106
5.11.3. Ficheros de migración	107
5.11.4. Modelos ORM	107
5.11.5. Vistas y Controladores	107
5.11.6. Test	108
5.11.7. Dificultades encontradas	108
5.11.8. Objetivos alcanzados	109

6. Equipo	111
6.1. Componentes	111
6.2. Entorno tecnológico	112
7. Conclusiones	113
8. Referencias	115

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

1.1. Arquitectura Modelo-Vista-Controlador (MVC)

UCASport sigue la arquitectura MVC, una evolución del modelo cliente-servidor y la más simple de todas las versiones de la arquitectura n-capas [Kappel, 2006]. Siendo $n = 3$ en este caso.

- Modelo: Representa la estructura de datos. Las clases que componen la capa de modelo poseen funciones de recuperación, inserción y actualización de la información almacenada en la base de datos.
- Vista: Es la capa más externa y por tanto más cercana al usuario.
- Controlador: Es la capa que actúa de intermediaria entre las vistas y los modelos, ayudando a procesar y dirigir las peticiones de usuario y generando las páginas pertinentes.

1.2. Historias de usuario

1.2.1. Fabricante (Manufacturer)

- Añadir fabricante: cuando el dueño de la tienda sepa de la existencia de un nuevo fabricante, debe poderlo dar de alta en el sitio. Tras hacer un click, teclear los detalles del fabricante en un formulario y enviarlo, los datos del fabricante deben estar disponibles.
- Listar fabricantes: la interfaz de administración debe tener una página principal que funcione como lista de todos los fabricantes.
- Ver fabricante: debe haber páginas que muestren los detalles de cada fabricante individual.
- Editar fabricante: cambiar los datos de un fabricante debe ser tan sencillo como darlo de alta.
- Eliminar fabricante: los fabricantes deben poder ser eliminados del sistema.

1.2.2. Proveedor (Supplier)

- Añadir proveedor: cuando el dueño de la tienda sepa de la existencia de un nuevo proveedor, debe poderlo dar de alta en el sitio. Tras hacer un click, teclear los detalles del proveedor en un formulario y enviarlo, los datos del proveedor deben estar disponibles.

- Listar proveedores: la interfaz de administración debe tener una página principal que funcione como lista de todos los proveedores.
- Ver proveedor: debe haber páginas que muestren los detalles de cada proveedor individual.
- Editar proveedor: cambiar los datos de un proveedor debe ser tan sencillo como darlo de alta.
- Eliminar proveedor: los proveedores deben poder ser eliminados del sistema.

1.2.3. Artículo (Article)

- Añadir artículo: cuando el dueño de la tienda sepa de la existencia de un nuevo artículo, debe poderlo dar de alta en el sitio. Tras hacer un click, teclear los detalles del artículo en un formulario y enviarlo, los datos del artículo deben estar disponibles.
- Listar artículos: la interfaz de administración debe tener una página principal que funcione como lista de todos los artículos.
- Ver artículo: debe haber páginas que muestren los detalles de cada artículo individual.
- Editar artículo: cambiar los datos de un artículo debe ser tan sencillo como darlo de alta.
- Eliminar artículo: los artículos deben poder ser eliminados del sistema.

1.2.3. Catálogo (Catalog)

- Ver catálogo: El usuario, en cualquier momento, debe de ser capaz de ver en el catálogo de artículos, todos aquellos artículos que se encuentran publicados y pueden ser comprados en la tienda.
- Mostrar artículo del catálogo: El usuario en cualquier momento, debe de ser capaz de poder acceder a la información detallada de un determinado artículo.
- Buscar en el catálogo: El usuario debe de ser capaz de realizar una búsqueda de artículos en el catálogo, por el nombre o parte de éste.
- RSS del catálogo: El usuario podrá suscribirse al RSS del catálogo, el cual le mantendrá informado con los últimos artículos que se han puesto a la venta en la tienda.

1.2.4. Carrito de la Compra (Cart y CartItem)

- Un cliente podrá introducir en su carrito de la compra tantos productos como desee adquirir de la tienda.
- Un cliente podrá eliminar artículos del carrito o de vaciar el carro por completo.

- Las anteriores acciones se podrán realizar de manera sencilla e intuitiva con una pulsación sobre los enlaces correspondientes.

1.2.5. Seguridad (User y UserSession)

- Inicio de sesión correcto: cuando un usuario que no ha iniciado sesión intenta acceder a una funcionalidad administrativa, el sistema le debe dirigir a la página de inicio de sesión; si el usuario y contraseña proporcionados son correctos, tras iniciar sesión el usuario debe quedar situado en la página de acceso a la funcionalidad que solicitó.
- Inicio de sesión incorrecto: cuando un usuario que no ha iniciado sesión intenta acceder a una funcionalidad administrativa, el sistema le debe dirigir a la página de inicio de sesión; si el usuario y contraseña proporcionados no son correctos, el sistema debe dirigir al usuario formulario de inicio de sesión junto a un mensaje de error.

1.2.6. Facturación (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 flujo 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.

1.2.7. 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

En la planificación del proyecto, dado que hemos usado una metodología ágil de desarrollo como SCRUM, nos hemos basado en los conocidos como sprints (etapas de desarrollo). La descripción del trabajo realizado en cada sprint puede encontrarse en la sección 5.

2.2. Calendario

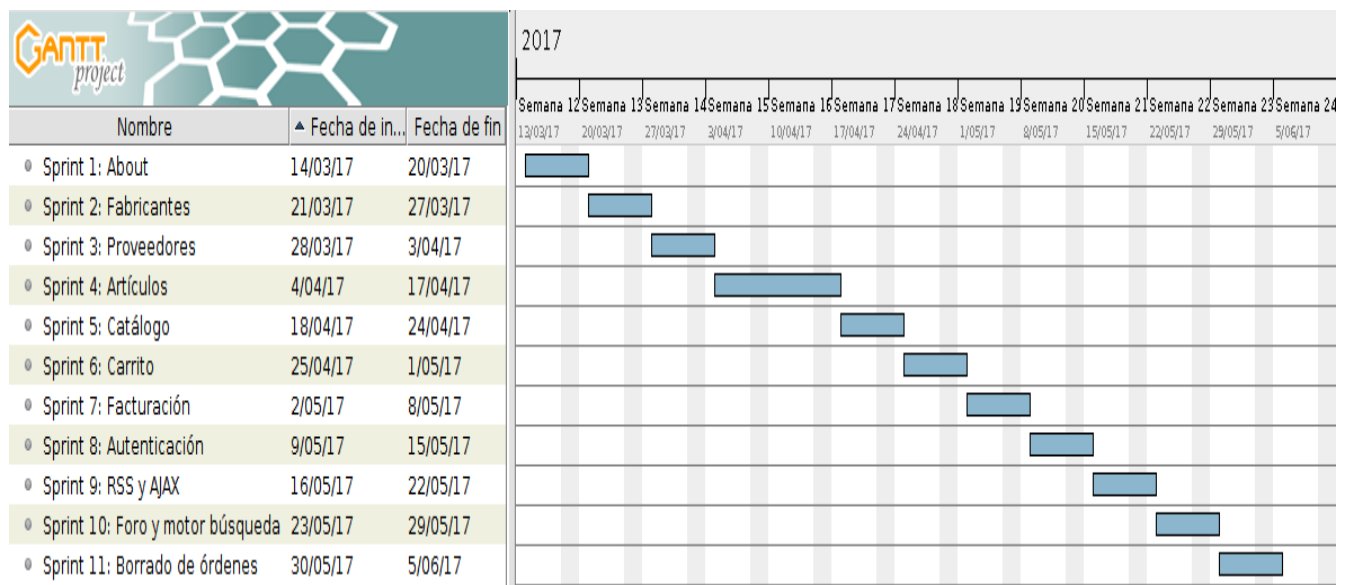


Figura 2.2. Calendario de las distintas etapas del proyecto

3. Esquema de la base de datos

En esta sección observaremos la evolución del diagrama E/R que representa el esqueleto de nuestra Base de Datos, el cual ha sido utilizado como apoyo para la elaboración de los ficheros de migración del proyecto.

3.1. Diagrama E/R-Inicial

En las primeras etapas de desarrollo hemos trabajado con la entidad principal, las dos entidades secundarias y la entidad relacion. Por lo tanto, observamos que en el diagrama inicial tenemos las siguientes entidades:

- Article
- Manufacturer
- Supplier

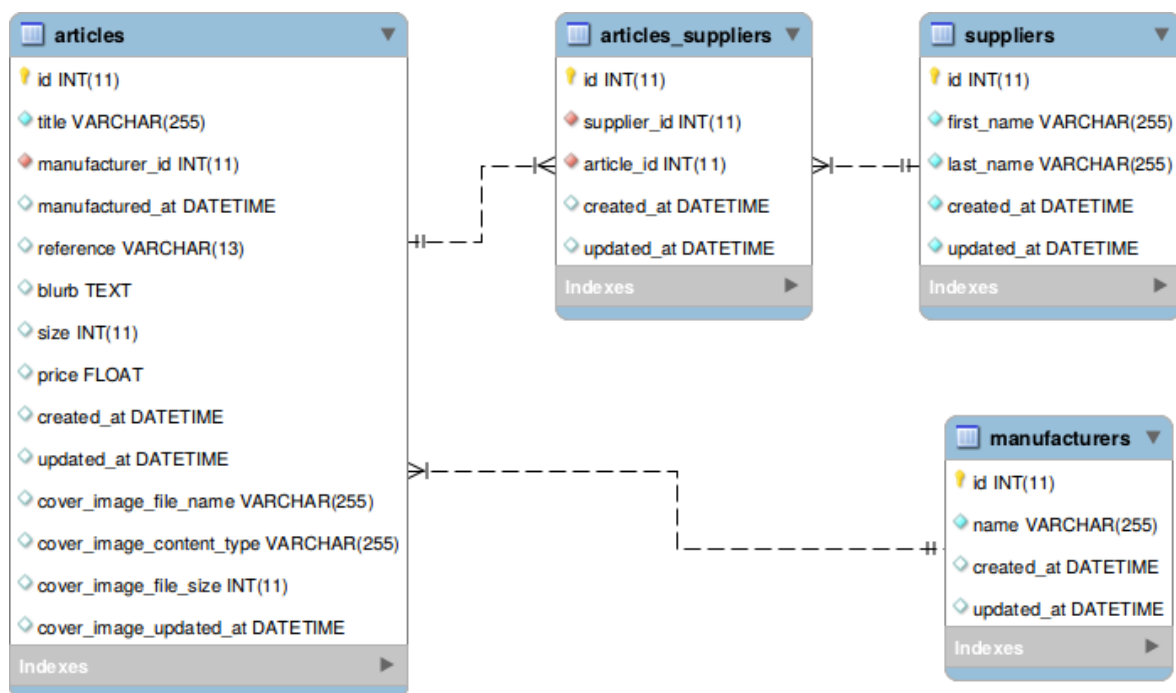


Figura 3.1. Diagrama E-R Inicial

3.2. Diagrama E/R con funcionalidad básica

Posteriormente y hasta la octava iteración, implementamos toda la funcionalidad básica, el carrito de la compra, los pedidos y la seguridad, por lo cual, al diagrama inicial añadimos las siguientes entidades:

- Article
- Manufacturer
- Supplier
- Cart
- Order
- User

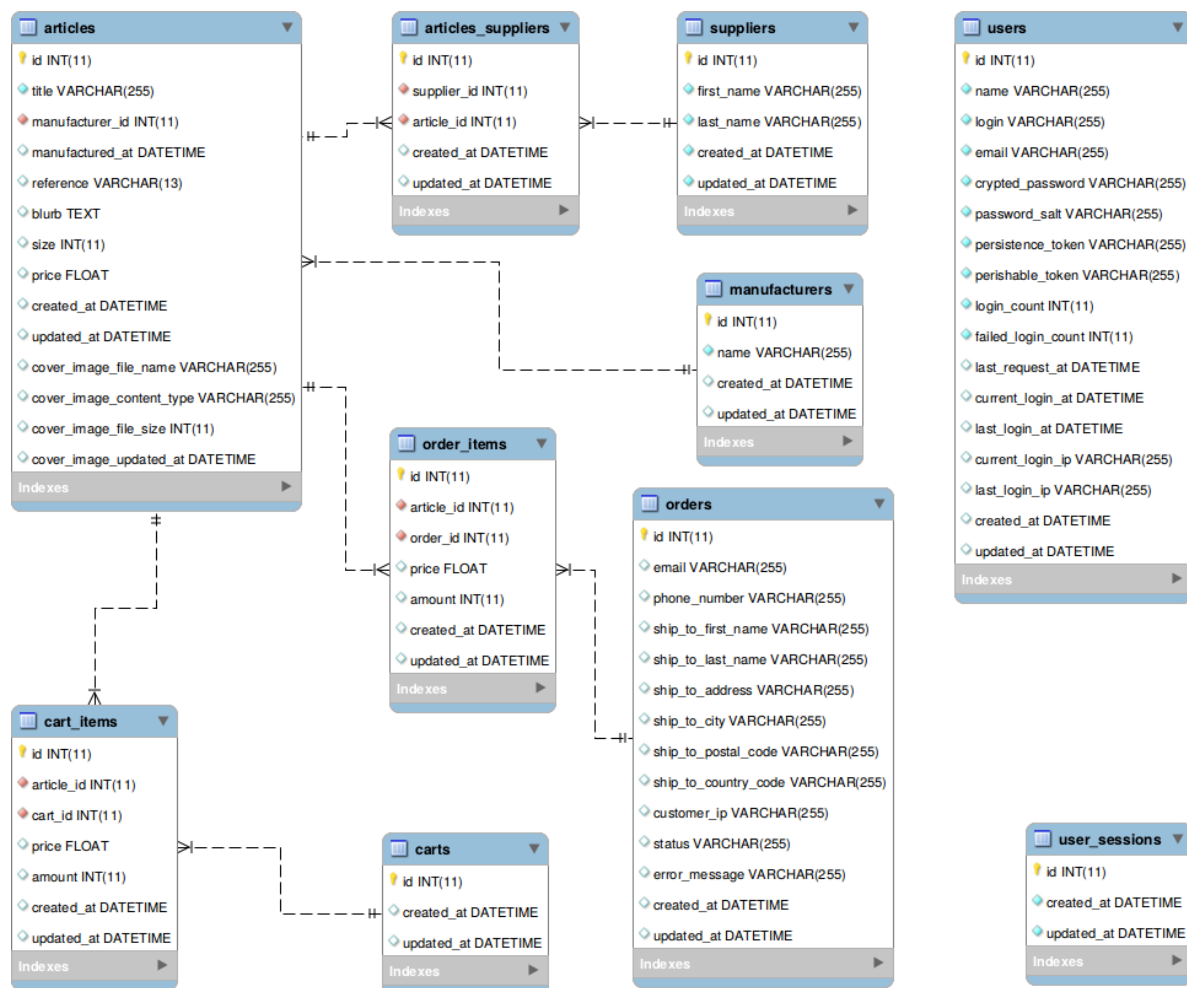


Figura 3.2. Diagrama E-R con funcionalidad básica

3.3. Diagrama ER final

Por último, hemos extendido en la décima iteración el diagrama de E-R con la inclusión del foro, añadiendo la entidad ForumPost:

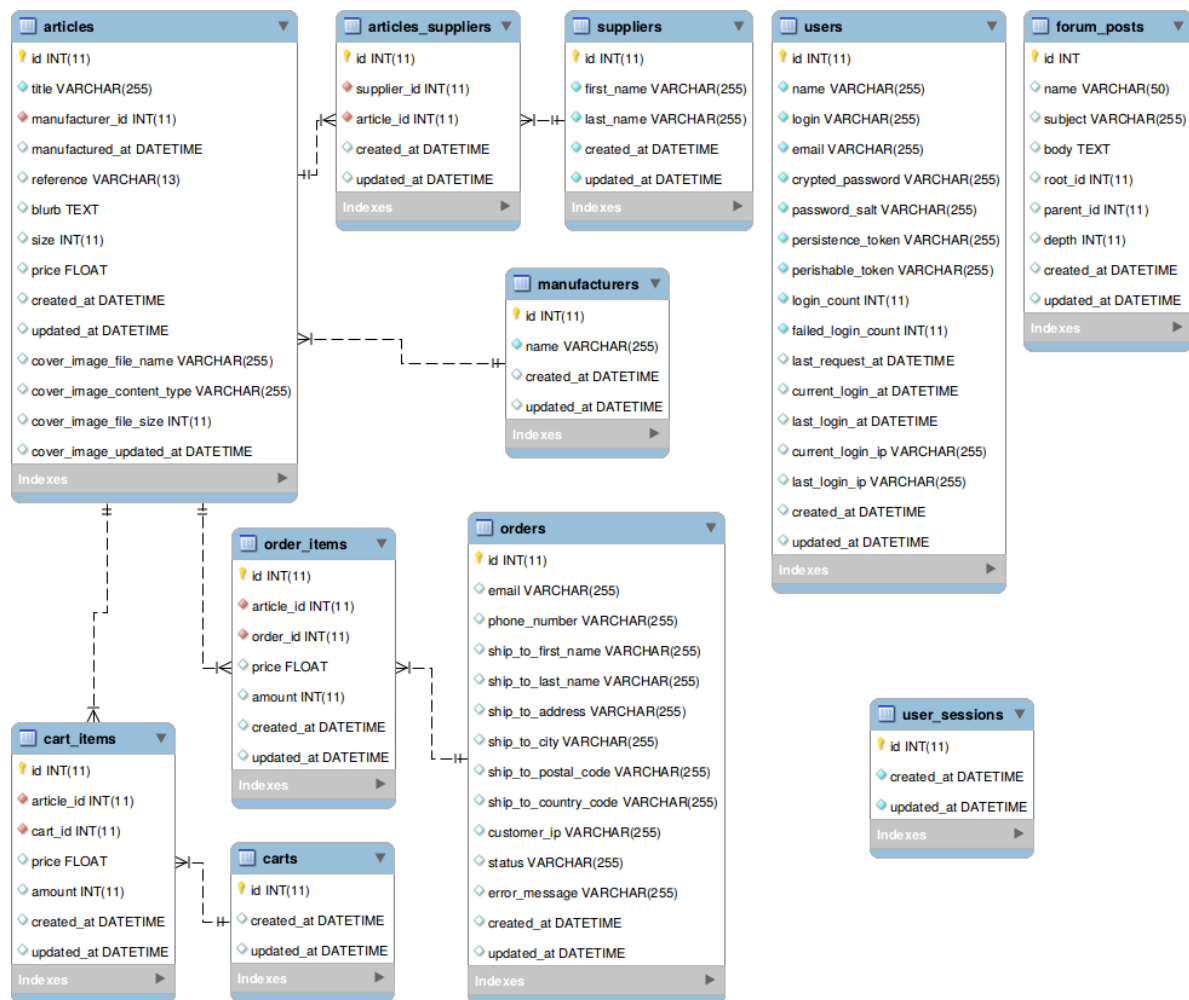


Figura 3.3. Diagrama E-R Final

Así pues, nuestro esquema entidad/relación final está compuesto por:

- Article
- Manufacturer
- Supplier
- Cart
- Order
- User
- ForumPost

4. Contenido del fichero de rutas

El estado final en el que ha quedado nuestro fichero de rutas es:

Fichero: config/routes.rb

Rails.application.routes.draw do

```
  root :to => 'catalog#index'

  get 'about' => 'about#index'
  get 'catalog' => 'catalog#index'
  get 'checkout' => 'checkout#index'
  get 'admin/manufacturer' => 'admin/manufacturer#index'
  get 'admin/supplier' => 'admin/supplier#index'
  get 'admin/article' => 'admin/article#index'
  get 'admin/order' => 'admin/order#index'
  get 'forum' => 'forum#index'

  get 'about/index'

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

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

  get 'admin/article/new'
  post 'admin/article/create'
  get 'admin/article/edit'
  post 'admin/article/update'
  post 'admin/article/destroy'
  get 'admin/article/show'
  get 'admin/article/show/:id' => 'admin/article#show'
  get 'admin/article/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/rss'
  get 'catalog/search'

  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'
post 'user_sessions/create'
get 'user_sessions/destroy'

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

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

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

end
```

5. Sprints

5.1. Sprint 1 (Generar About)

El objetivo de esta iteración era generar la vista about de nuestra web y familiarizarnos con el framework de desarrollo Ruby on Rails.

5.1.1. Sprint backlog y diagrama burndown

No se requirió mucho tiempo para terminar este paso, así que nuestro backlog es bastante simple.

Tareas	L	M	X	J	V	S	D
Base de Datos	0	120	0	0	0	0	0
Vista y Controlador	0	40	0	0	0	0	0
Layout	0	20	20	0	0	0	0
Total	0	180	20	0	0	0	0

Tabla 5.1: Sprint 1 backlog (minutos)

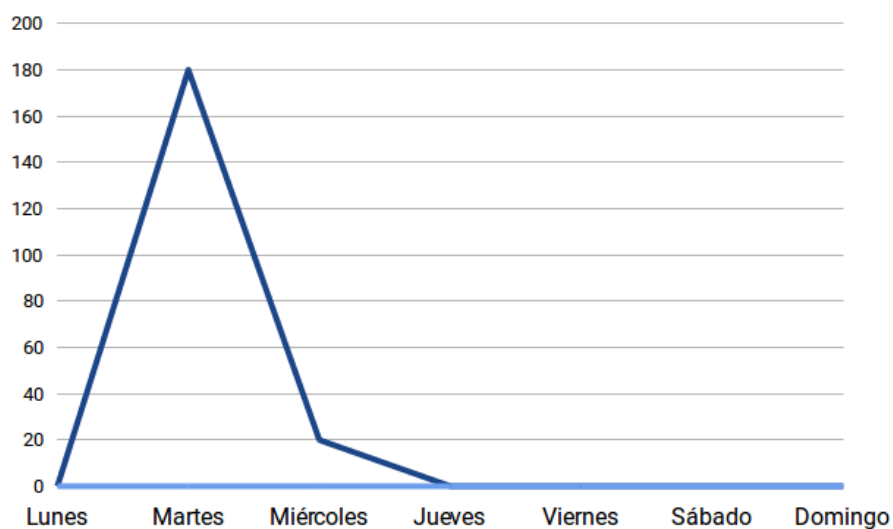


Figura 5.1: Diagrama burndown sprint 1

5.1.2. Gemas utilizadas

Las gemas más relevantes que forman parte del proyecto son las siguientes:

- rails , versión 4.2.5: Framework de Ruby on Rails.
- mysql2 , versión 0.4.2: Controlador de MySQL.

Además se incluyen otras gemas instaladas por omisión:

- sass-rails, versión 5.0.6: Utiliza SCSS para la hoja de estilos.
- uglifier, versión 2.7.2: Comprime los archivos de JavaScript.
- coffee-rails, versión 4.1.1: Permite utilizar CoffeeScript en Rails.
- turbolinks, versión 2.5.3: Acelera el seguimiento de enlaces.
- jbuilder, versión 2.3.2: Construye JSON APIs con facilidad.
- sddoc, versión 0.4.1 : Genera documentación de API.
- spring, versión 1.6.1: Permite mantener la aplicación en segundo plano durante el desarrollo.

5.1.3. Ficheros de migración

En este sprint no hemos añadido ficheros de migración en db/migrate

5.1.4. Modelos ORM

En este sprint no ha sido necesario crear ningún modelo en app/models

5.1.5. Vistas y controladores

Hemos generado el controlador para la vista about, de la siguiente forma:

```
ine@gii ~/projects/eshop6 $ /bin/bash --login
ine@gii ~/projects/eshop6 $ rvm use 2.2.4
Using /home/ine/.rvm/gems/ruby-2.2.4
ine@gii ~/projects/eshop6 $ rails generate controller about index
```

Hemos modificado el fichero del controlador, para que se muestre el título de la página:

Fichero: app/controllers/about_controller.rb

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

Posteriormente añadimos la hoja de estilo del proyecto de ejemplo miniemporium localizada en **app/assets/stylesheets/style.css** y adaptamos la plantilla de nuestra página modificando el archivo, que pasa a contener lo siguiente:

Fichero: app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
<head>
  <title><%= @page_title || 'UCASport' %></title>
  <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track' => true %>
  <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
  <%= csrf_meta_tags %>
</head>
<body>
  <div id="header">
    <h1 id="logo">UCASport</h1>
    <h2 id="slogan">Expertos en zapatillas deportivas</h2>
  </div>

  <div id="menu">
    <ul>
      <li><a href="/admin/supplier">Proveedores</a> &nbsp; &nbsp; </li>
      <li><a href="/admin/manufacturer">Fabricantes</a> &nbsp; &nbsp; </li>
      <li><a href="/admin/article">Artículos</a> &nbsp; &nbsp; </li>
      <li><a href="/admin/order">Pedidos</a> &nbsp; &nbsp; </li>
      <li><a href="/">Catálogo</a> &nbsp; &nbsp; </li>
      <li><a href="/about">Sobre UCASport</a> &nbsp; &nbsp; </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; <%= Time.current.year %> UCASport
  </div>
</body>
</html>
```

Finalmente, modificamos la vista about#index, desde su fichero de vista.

Fichero: app/views/about/index.html.erb

```
<p>Tienda online UCASport, situada en la ESI de Puerto Real</p>
<h2>Dirección de Correo</h2>
<address>
  UCASport<br/>
  CP 11510<br/>
  Puerto Real, Cádiz<br/>
</address>
```

5.1.6. Test

En esta ocasión hemos hecho un test de controlador.

Fichero: test/controllers/about_controller_test.rb

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

class AboutControllerTest < ActionController::TestCase
  test "index" do
    get :index
    assert_response :success
    assert_template 'about/index'
    assert_equal 'Sobre UCASport', assigns(:page_title)
    assert_select 'title', 'Sobre UCASport'
    assert_select 'h1', 'Sobre UCASport'
  end
end
```

Pasamos a ejecutarlo:

```
ine@gii ~/projects/eshop6 $ rake test TEST=test/controllers/about_controller_test.rb
Run options: --seed 22942

# Running:

.

Finished in 3.241217s, 0.3085 runs/s, 1.5426 assertions/s.

1 runs, 5 assertions, 0 failures, 0 errors, 0 skips
```

5.1.7. Dificultades encontradas

Para nosotros Ruby y Ruby on rails es algo completamente nuevo, por este motivo hemos tenido problemas para adaptarnos. Tuvimos dificultades con la integración de la base de datos y con las gemas.

Con ayuda del libro *Beginning Ruby on Rails ECommerce*, recomendado por el profesor, y el código de *miniemporium* adaptándolo a nuestra tienda conseguimos familiarizarnos algo más rápido a Ruby y Ruby on Rails. Del mismo modo, la ayuda del profesor fue fundamental para entender las convenciones de Ruby on Rails, que también eran completamente nuevas para nosotros.

5.1.8. Objetivos alcanzados

Con este sprint hemos conseguido:

- Familiarizarnos poco a poco con el framework, de todas formas todavía queda mucho por delante.
- Poder visualizar el about de nuestra tienda online.

5.2. Sprint 2 (Gestión de Fabricantes)

En esta iteración el objetivo ha sido generar la entidad secundaria **Manufacturer**, que corresponde a los fabricantes de la entidad principal **Article**. Es a su vez la primera implementación de un CRUD y migración de la base de datos, con lo que este sprint se complica con respecto al anterior.

5.2.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros Migración	0	30	0	0	0	0	0
Vista y Controlador	0	115	0	0	0	0	30
Modelo	0	20	0	0	0	0	0
Test	0	0	0	0	0	0	20
Total	0	180	0	0	0	0	50

Tabla 5.2: Sprint 2 backlog (minutos)

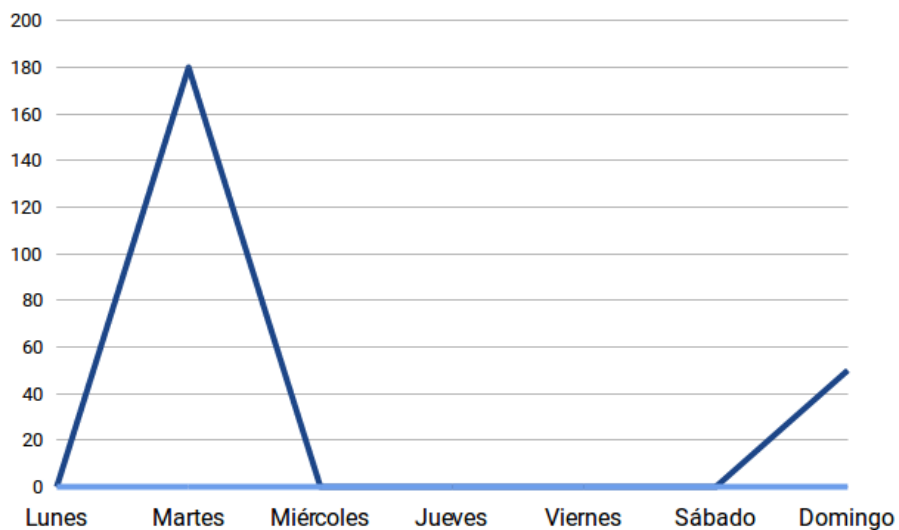


Figura 5.2: Diagrama burndown sprint 2

5.2.2. Gemas utilizadas

En esta iteración no hemos utilizados gemas nuevas.

5.2.3. Ficheros de migración

Seguidamente vamos a introducir el comando que permite crear las plantillas del modelo, del test del modelo y del fichero de migración de Manufacturer.

```
ine@gii ~/projects/eshop6 $ rails generate model Manufacturer
```

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

Fichero: db/migrate/20170321100602_create_manufacturers.rb

```
class CreateManufacturers < ActiveRecord::Migration
  def change
    create_table :manufacturers do |t|
      t.string :name, :limit => 255, :null => false, :unique => true
      t.timestamps
    end
  end
end
```

Para que se creen convenientemente las tablas en la base de datos:

```
ine@gii ~/projects/eshop6 $ rake db:migrate
```

5.2.4. Modelos ORM

Dejamos el fichero del modelo de la siguiente forma:

Fichero: app/models/manufacturer.rb

```
class Manufacturer < ActiveRecord::Base
  has_many :articles
  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 manufacturer, dentro de admin:

```
ine@gii ~/projects/eshop6 $ rails generate controller admin/manufacturer new create edit update destroy show index
```

A continuación, modificamos el fichero del controlador:

Fichero: app/controllers/admin/manufacture_controller.rb

```
class Admin::ManufacturerController < ApplicationController
  def new
    @manufacturer = Manufacturer.new
    @page_title = 'Crear nuevo fabricante'
  end

  def create
    @manufacturer = Manufacturer.new(manufacturer_params)
    if @manufacturer.save
      flash[:notice] = "Fabricante #{@manufacturer.name} ha sido creado exitosamente."
      redirect_to :action => 'index'
    else
      @page_title = 'Crear nuevo fabricante'
      render :action => 'new'
    end
  end

  def edit
    @manufacturer = Manufacturer.find(params[:id])
    @page_title = 'Editar fabricante'
  end

  def update
    @manufacturer = Manufacturer.find(params[:id])
    if @manufacturer.update_attributes(manufacturer_params)
      flash[:notice] = "Fabricante #{@manufacturer.name} ha sido actualizado exitosamente."
      redirect_to :action => 'show', :id => @manufacturer
    else
      @page_title = 'Editar Fabricante'
      render :action => 'edit'
    end
  end

  def destroy
    @manufacturer = Manufacturer.find(params[:id])
    @manufacturer.destroy
    flash[:notice] = "Fabricante borrado correctamente #{@manufacturer.name}."
    redirect_to :action => 'index'
  end

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

  def index
    @manufacturers = Manufacturer.all
    @page_title = 'Listado de fabricantes'
  end

  private
  def manufacturer_params
    params.require(:manufacturer).permit(:name)
  end
end
```

Y seguidamente los ficheros correspondientes a las vistas:

Fichero: app/views/admin/manufacturer/_form.html.erb

```
<% if @manufacturer.errors.any? %>
<div id="errorExplanation">
<h2><%= pluralize(@manufacturer.errors.count, "error") %> prohíben guardar este fabricante:</h2>
<ul>
  <% @manufacturer.errors.full_messages.each do |msg| %>
    <li><%= msg %></li>
  <% end %>
</ul>
</div>
<% end %>

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

Fichero: app/views/admin/manufacturer/_manufacturer.html.erb

```
<tr>
<td><%= link_to manufacturer.name, :action => 'show', :id => manufacturer %></td>
<td><%= link_to 'Editar', :action => 'edit', :id => manufacturer %></td>
<td>
  <%= button_to 'Borrar', { :action => 'destroy', :id => manufacturer },
    data: { confirm: "Estás seguro que quieres borrar el fabricante #{manufacturer.name}?" } %>
</td>
</tr>
```

Fichero: app/views/admin/manufacturer/edit.html.erb

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

<%= link_to 'Show', :action => 'show', :id => @manufacturer %> |
<%= link_to 'Back', :action => 'index' %>
```

Fichero: app/views/admin/manufacturer/index.html.erb

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

<%= render :partial => 'manufacturer', :collection => @manufacturers %>
</table>

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

Fichero: app/views/admin/manufacturer/new.html.erb

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

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

Fichero: app/views/admin/manufacturer/show.html.erb

```
<dl>
  <dt>Name</dt>
  <dd><%= @manufacturer.name %></dd>
</dl>

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

5.2.6. Test

Vamos a preparar el fichero de tests, en primer lugar el del modelo:

Fichero: test/models/manufacturer_test.rb

```
require 'test_helper'

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

  test "create" do
    manufacturer = Manufacturer.new(
      :name => 'Joma'
    )
    assert manufacturer.save
  end
end
```

Y a continuación el del controlador:

Fichero: test/controllers/admin/manufacture_controller_test.rb

```
require 'test_helper'

class Admin::ManufacturerControllerTest < ActionController::TestCase

  fixtures :manufacturers

  test "new" do
    get :new
    assert_response :success
  end
end
```

```

end

test "create" do
  num_manufacturers = Manufacturer.count
  post :create, :manufacturer => { :name => 'New Balance' }
  assert_response :redirect
  assert_redirected_to :action => 'index'
  assert_equal num_manufacturers + 1, Manufacturer.count
end

test "edit" do
  get :edit, :id => 1
  assert_select 'input' do
    assert_select '[type=?]', 'text'
    assert_select '[name=?]', 'manufacturer[name]'
    assert_select '[value=?]', 'Adidas'
  end
end

test "update" do
  post :update, :id => 1, :manufacturer => { :name => 'Adidas.com' }
  assert_response :redirect
  assert_redirected_to :action => 'show', :id => 1
  assert_equal 'Adidas.com', Manufacturer.find(1).name
end

test "destroy" do
  assert_difference(Manufacturer, :count, -1) do
    post :destroy, :id => 1
    assert_equal flash[:notice], 'Fabricante borrado correctamente Adidas.'
    assert_response :redirect
    assert_redirected_to :action => 'index'
    get :index
    assert_response :success
    assert_select 'div#notice', 'Fabricante borrado correctamente Adidas.'
  end
end

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

test "index" do
  get :index
  assert_response :success
  assert_select 'table' do
    assert_select 'tr', Manufacturer.count + 1
  end

  Manufacturer.find_each do |a|
    assert_select 'td', a.name
  end
end
end

```

Vamos a ejecutar ambos tests:

```
ine@gii ~/projects/eshop6 $ rake test TEST=test/models/manufacturer_test.rb
Run options: --seed 20656

# Running:

..

Finished in 0.528788s, 3.7822 runs/s, 7.5645 assertions/s.

2 runs, 4 assertions, 0 failures, 0 errors, 0 skips

ine@gii ~/projects/eshop6 $ rake test TEST=test/controllers/admin/manufacture_controller_test.rb
Run options: --seed 49689

# Running:

.....

Finished in 0.975148s, 7.1784 runs/s, 31.7901 assertions/s.

7 runs, 31 assertions, 0 failures, 0 errors, 0 skips
```

5.2.7. Dificultades encontradas

Al no estar muy familiarizados con el framework, volvimos a tener problemas con la integración de la base de datos y con las gemas; ésto se solucionó con los comandos:

- bundle install
- rake db:migrate

También, debimos consultar [miniemporium](#) para aclararnos dudas puntuales de las vistas y los controladores.

5.2.8. Objetivos alcanzados

Con este sprint hemos conseguido poder visualizar el CRUD (Create, Read, Update and Delete) de nuestra entidad secundaria, manufacturer.

5.3. Sprint 3 (Gestión de Proveedores)

El objetivo de esta iteración es tener el CRUD de Proveedores.

5.3.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros Migración	0	60	0	0	0	0	0
Vista y Controlador	0	80	0	0	0	0	0
Modelo	0	50	0	0	0	0	0
Test	0	40	50	0	0	0	0
Total	0	230	50	0	0	0	0

Tabla 5.3: Sprint 3 backlog (minutos)

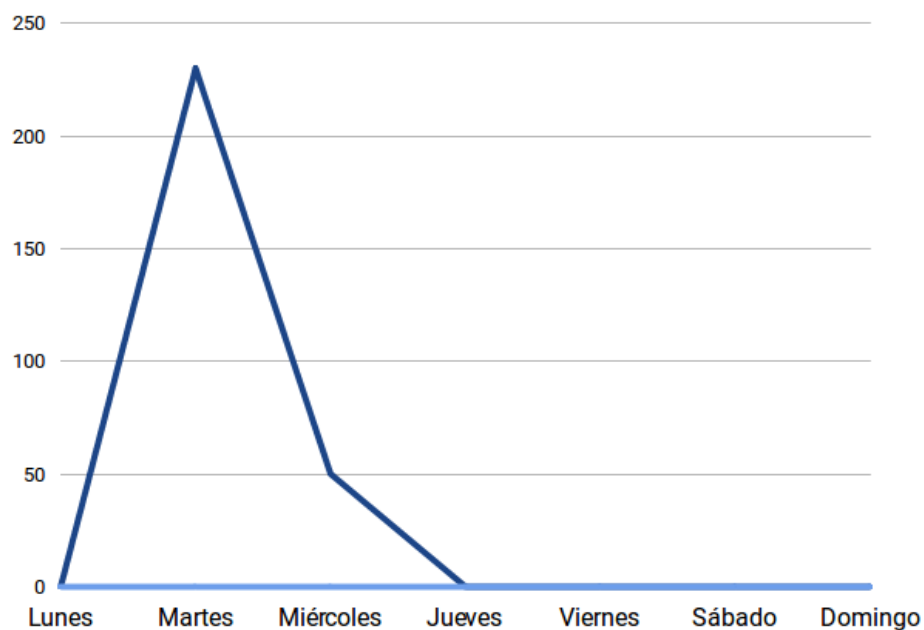


Figura 5.3: Diagrama burndown

5.3.2. Gemas utilizadas

En esta iteración no hemos utilizados gemas nuevas.

5.3.3. Ficheros de migración

Vamos a introducir el comando que permite crear las plantillas del modelo, del test del modelo y del fichero de migración de Supplier.

```
ine@gii ~/projects/eshop6 $ rails generate model Supplier
```

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

Fichero: db/migrate/20170328092851_create_suppliers.rb

```
class CreateSuppliers < ActiveRecord::Migration
  def change
    create_table :suppliers do |t|
      t.string :first_name, :limit => 255, :null => false
      t.string :last_name, :limit => 255, :null => false
      t.timestamps null: false
    end
  end
end
```

Para que se creen convenientemente las tablas en la base de datos:

```
ine@gii ~/projects/eshop6 $ rake db:migrate
```

5.3.4. Modelos ORM

Dejamos el fichero del modelo de la siguiente forma:

Fichero: app/models/supplier.rb

```
class Supplier < ActiveRecord::Base
  has_and_belongs_to_many :articles
  validates_presence_of :first_name, :last_name
  validates_length_of :first_name, :in => 2..255
  validates_length_of :last_name, :in => 2..255

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


5.3.5. Vistas y controladores

Generamos el controlador supplier, dentro de admin:

```
ine@gii ~/projects/eshop6 $ rails generate controller admin/supplier new create edit update destroy show index
```

A continuación, modificamos el fichero del controlador:

Fichero: app/controllers/admin/supplier_controller.rb

```
class Admin::SupplierController < ApplicationController
  def new
    @supplier = Supplier.new
    @page_title = 'Crear nuevo proveedor'
  end

  def create
    @supplier = Supplier.new(supplier_params)
    if @supplier.save
      flash[:notice] = "Proveedor #{@supplier.name} fue correctamente creado."
      redirect_to :action => 'index'
    else
      @page_title = 'Crear nuevo proveedor'
      render :action => 'new'
    end
  end

  def edit
    @supplier = Supplier.find(params[:id])
    @page_title = 'Editar proveedor'
  end

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

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

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

  def index
    @suppliers = Supplier.all
    @page_title = 'Listado de proveedores'
  end
end
```

```

private
def supplier_params
  params.require(:supplier).permit(:first_name, :last_name)
end
end

```

Y seguidamente los ficheros correspondientes a las vistas:

Fichero: app/views/admin/supplier/_form.html.erb

```

<% if @supplier.errors.any? %>
<div id="errorExplanation">
<h2><%= pluralize(@supplier.errors.count, "error") %> prohíben guardar este proveedor:</h2>
<ul>
  <% @supplier.errors.full_messages.each do |msg| %>
    <li><%= msg %></li>
  <% end %>
</ul>
</div>
<% end %>

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

<div class="field">
<p><label for="supplier_last_name">Apellidos</label><br/>
<%= text_field 'supplier', 'last_name' %></p>
</div>

```

Fichero: app/views/admin/supplier/_supplier.html.erb

```

<tr>
<td><%= link_to supplier.name, :action => 'show', :id => supplier %></td>
<td><%= link_to 'Editar', :action => 'edit', :id => supplier %></td>
<td>
  <%= button_to 'Borrar', { :action => 'destroy', :id => supplier },
    data: { confirm: "Estás seguro que quieres eliminar al proveedor #{supplier.name}?" } %>
</td>
</tr>

```

Fichero: app/views/admin/supplier/index.html.erb

```

<table>
<tr>
  <th>Nombre y Apellidos</th>
  <th>Editar</th>
  <th>Borrar</th>
</tr>

<%= render :partial => 'supplier', :collection => @suppliers %>
</table>

<p><%= link_to 'Añadir nuevo proveedor', :action => 'new' %></p>

```

Fichero: app/views/admin/supplier/new.html.erb

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

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

Fichero: app/views/admin/supplier/show.html.erb

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

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

Fichero: app/views/admin/supplier/edit.html.erb

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

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

5.3.6. Test

Vamos a preparar el fichero de tests, en primer lugar el del modelo:

Fichero: test/models/supplier_test.rb

```
require 'test_helper'

class SupplierTest < ActiveSupport::TestCase
  test "test_name" do
    supplier = Supplier.create(:first_name => 'Pepe', :last_name => 'Mateo')
    assert_equal 'Pepe Mateo', supplier.name
  end
end
```

Y a continuación el del controlador:

Fichero: test/controllers/admin/supplier_controller_test.rb

```
require 'test_helper'

class Admin::SupplierControllerTest < ActionController::TestCase
  fixtures :suppliers
end
```

```

test "new" do
  get :new
  assert_template 'admin/supplier/new'
  assert_select 'div#content' do
    assert_select 'h1', 'Crear nuevo proveedor'
    assert_select "form[action=\"/admin/supplier/create\"]"
  end
end

test "create" do
  get :new
  assert_template 'admin/supplier/new'
  assert_difference(Supplier, :count) do
    post :create, :supplier => { :first_name => 'Pepe', :last_name => 'Mateo' }
    assert_response :redirect
    assert_redirected_to :action => 'index'
  end
  assert_equal 'Proveedor Pepe Mateo fue correctamente creado.', flash[:notice]
end

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

test "edit" do
  get :edit, :id => 1
  assert_select 'input' do
    assert_select '[type=?]', 'text'
    assert_select '[name=?]', 'supplier[first_name]'
    assert_select '[value=?]', 'Pepe'
  end
  assert_select 'input' do
    assert_select '[type=?]', 'text'
    assert_select '[name=?]', 'supplier[last_name]'
    assert_select '[value=?]', 'Mateo'
  end
end

test "update" do
  post :update, :id => 1, :supplier => { :first_name => 'Gabriel', :last_name => 'Mateo' }
  assert_response :redirect
  assert_redirected_to :action => 'show', :id => 1
  assert_equal 'Gabriel', Supplier.find(1).first_name
end

test "test_destroy" do
  assert_difference(Supplier, :count, -1) do
    post :destroy, :id => 1
    assert_equal flash[:notice], 'Proveedor correctamente eliminado Pepe Mateo.'
    assert_response :redirect
    assert_redirected_to :action => 'index'
  end
  get :index
  assert_response :success
  assert_select 'div#notice', 'Proveedor correctamente eliminado Pepe Mateo.'
end

test "show" do
  get :show, :id => 1

```

```

assert_template 'admin/supplier/show'
assert_equal 'Pepe', assigns(:supplier).first_name
assert_equal 'Mateo', assigns(:supplier).last_name
assert_select 'div#content' do
  assert_select 'h1', Supplier.find(1).name
end
end

test "index" do
  get :index
  assert_response :success
  assert_select 'table' do
    assert_select 'tr', Supplier.count + 1
  end
  Supplier.find_each do |a|
    assert_select 'td', a.name
  end
end
end

```

Vamos a ejecutar ambos tests:

```

ine@gii ~/projects/eshop6 $ rake test TEST=test/models/supplier_test.rb
Run options: --seed 40385

# Running:

.

Finished in 0.484517s, 2.0639 runs/s, 2.0639 assertions/s.

1 runs, 1 assertions, 0 failures, 0 errors, 0 skips

ine@gii ~/projects/eshop6 $ rake test TEST=test/controllers/admin/supplier_controller_test.rb
Run options: --seed 58218

# Running:

.....

Finished in 1.008706s, 7.9310 runs/s, 42.6289 assertions/s.

8 runs, 43 assertions, 0 failures, 0 errors, 0 skips

```

5.3.7. Dificultades encontradas

En este sprint no hemos encontrado ninguna dificultad, ya que el proceso para gestionar al proveedor es prácticamente idéntico al de fabricante.

5.3.8. Objetivos alcanzados

Con este sprint hemos conseguido poder visualizar el CRUD (Create, Read, Update and Delete) de Proveedor.

5.4 Sprint 4 (Gestión de Artículos)

El objetivo de esta iteración es tener el CRUD de Artículos, nuestra entidad principal y su asociación con la entidad secundaria Manufacturer, cuya relación es de uno a muchos. Y también con la asociación con la entidad secundaria Supplier, cuya relación es de muchos a muchos. Para este Sprint hemos contado con dos semanas debido a las fiestas de Semana Santa. En esta ocasión, además del test del modelo, tuvimos que realizar los correspondientes tests de integración.

5.4.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D	L	M	X	J	V	S	D
Ficheros Migración	0	60	0	0	0	0	0	0	0	0	0	0	0	0
Vista y Controlador	0	0	30	20	0	0	0	0	0	0	0	20	0	0
Modelo	0	40	0	20	0	0	0	0	0	0	0	0	0	0
Test	0	70	20	20	0	0	0	0	0	0	0	0	10	0
Total	0	170	50	60	0	0	0	0	0	0	0	20	10	0

Tabla 5.4: Sprint 4 backlog (minutos)

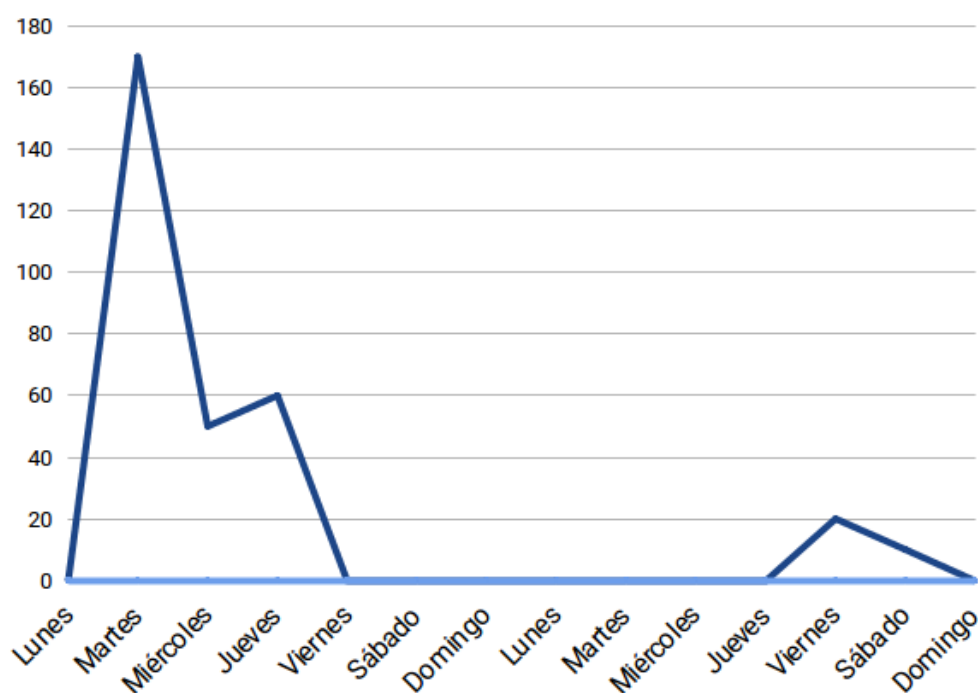


Figura 5.4: Diagrama burndown

5.4.2. Gemas utilizadas

En esta iteración si hemos utilizados gemas nuevas. Son las siguientes:

- will_paginate, versión 3.0.7: Paginación de las consultas.
- RedCloth, versión 4.2.9 : Añade estilo al texto plano para html en la descripción de los artículos
- paperclip, versión 4.3.2 : Adjunta imágenes.

5.4.3. Ficheros de migración

Con el siguiente comando vamos a crear el correspondiente fichero de migración y posteriormente el fichero de modelo de Article, pero en esta ocasión lo haremos de forma que cuando cree el modelo, no cree el fichero de migración, por haberlo creado ya anteriormente (--skip-migration).

```
ine@gii ~/projects/eshop6 $ rails generate migration create_articles_and_articles_suppliers
```

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

Fichero: db/migrate/20170404092937_create_articles_and_articles_suppliers.rb

```
class CreateArticlesAndArticlesSuppliers < ActiveRecord::Migration
  def up
    create_table :articles do |t|
      t.string :title, :limit => 255, :null => false
      t.integer :manufacturer_id, :null => false
      t.datetime :manufactured_at
      t.string :reference, :limit => 13, :unique => true
      t.text :blurb
      t.integer :size
      t.float :price
      t.timestamps
    end

    create_table :articles_suppliers do |t|
      t.integer :supplier_id, :null => false
      t.integer :article_id, :null => false
      t.timestamps
    end

    say_with_time 'Adding foreign keys' do
      # Add foreign key reference to articles_suppliers table
      execute 'ALTER TABLE articles_suppliers ADD CONSTRAINT fk_articles_suppliers_suppliers
        FOREIGN KEY (supplier_id) REFERENCES suppliers(id) ON DELETE CASCADE'
      execute 'ALTER TABLE articles_suppliers ADD CONSTRAINT fk_articles_suppliers_articles
        FOREIGN KEY (article_id) REFERENCES articles(id) ON DELETE CASCADE'
      # Add foreign key reference to manufacturers table
      execute 'ALTER TABLE articles ADD CONSTRAINT fk_articles_manufacturers
        FOREIGN KEY (manufacturer_id) REFERENCES manufacturers(id) ON DELETE CASCADE'
    end
  end
end
```

```

def self.down
  drop_table :articles
  drop_table :articles_suppliers
end
end

```

Para la inserción de imágenes de artículos añadimos el siguiente fichero dentro de db/migrate:

Fichero: db/migrate/20130630150644_add_cover_image_attachment_to_articles.rb

```

class AddCoverImageAttachmentToArticles < ActiveRecord::Migration
  def up
    add_attachment :articles, :cover_image
  end

  def down
    remove_attachment :articles, :cover_image
  end
end

```

Creamos ahora el fichero del modelo:

```

ine@gii ~/projects/eshop6 $ rails generate model Article --skip-migration

```

Ahora editamos dicho fichero convenientemente:

Fichero: app/models/article.rb

```

class Article < ActiveRecord::Base
  has_and_belongs_to_many :suppliers
  belongs_to :manufacturer

  #Las 2 líneas siguientes habrá que descomentarlas al implementar el carrito
  #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 => 3..255, :message => 'debe ser una cadena entre 3 y 255 caracteres'
  validates_presence_of :manufacturer
  validates_presence_of :suppliers, :message => 'debe seleccionar al menos un proveedor'
  validates_presence_of :manufactured_at
  validates_numericality_of :size, :only_integer => true, :greater_than => 30, :less_than => 50, :message => 'debe ser un número mayor que 30 y menor que 50'
  validates_numericality_of :price, :message => 'debe ser un número'
  validates_length_of :reference, :in => 1..13, :message => 'debe ser una cadena entre 1 y 13 caracteres'
  validates_format_of :reference, :with => /[0-9\-\XX]{13}/, :message => 'formato incorrecto'
  validates_uniqueness_of :reference, :message => 'ya existe un artículo con esa misma referencia'

  def supplier_names
    self.suppliers.map{|supplier| supplier.name}.join(", ")
  end
end

```



```
#Las 3 líneas siguientes habrá que descomentarlas al implementar el catálogo
#def self.latest(num)
# all.order("articles.id desc").includes(:suppliers, :manufacturer).limit(num)
#end
end
```

Para finalizar:

```
ine@gii ~/projects/eshop6 $ rake db:migrate
```

5.4.5. Vistas y controladores

Generamos el controlador supplier, dentro de admin:

```
ine@gii ~/projects/eshop6 $ rails generate controller admin/article new create edit update destroy show index
```

A continuación, modificamos el fichero del controlador:

Fichero: app/controllers/admin/article_controller.rb

```
class Admin::ArticleController < ApplicationController
  def new
    load_data
    @article = Article.new
    @page_title = 'Crear nuevo artículo'
  end

  def create
    @article = Article.new(article_params)
    if @article.save
      flash[:notice] = "El artículo #{@article.title} ha sido correctamente creado."
      redirect_to :action => 'index'
    else
      load_data
      @page_title = 'Crear nuevo artículo'
      render :action => 'new'
    end
  end

  def edit
    load_data
    @article = Article.find(params[:id])
    @page_title = 'Editar artículo'
  end

  def update
    @article = Article.find(params[:id])
    if @article.update_attributes(article_params)
      flash[:notice] = "El artículo #{@article.title} ha sido correctamente actualizado."
      redirect_to :action => 'show', :id => @article
    else
      load_data
      @page_title = 'Editar artículo'
      render :action => 'edit'
    end
  end
end
```

```

    end
  end

  def destroy
    @article = Article.find(params[:id])
    @article.destroy
    flash[:notice] = "Borrado correctamente el artículo #{@article.title}."
    redirect_to :action => 'index'
  end

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

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

  private

  def load_data
    @suppliers = Supplier.all
    @manufacturers = Manufacturer.all
  end

  def article_params
    params.require(:article).permit(:title, :manufacturer_id, :manufactured_at, { :supplier_ids => [] },
                                     :reference, :blurb, :price, :size, :cover_image)
  end
end

```

Y seguidamente los ficheros correspondientes a las vistas:

Fichero: app/views/admin/article/_form.html.erb

```

<% if @article.errors.any? %>
  <div id="errorExplanation">
    <h2><%= pluralize(@article.errors.count, "error") %> prohibido guardar este artículo:</h2>
    <ul>
      <% @article.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

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

<div class="field">
  <p><label for="article_manufacturer">Fabricante</label><br/>
  <%= collection_select :article, :manufacturer_id, @manufacturers, :id, :name %></p>
</div>

<div class="field">
  <p><label for="article[supplier_ids][]">Proveedores</label><br/>
  <%= select_tag 'article[supplier_ids][]', options_from_collection_for_select(@suppliers, :id, :name,

```

```

    @article.suppliers.collect{|supplier| supplier.id}), { :multiple => true, :size => 5 } %></p>
</div>

<div class="field"><p>
  <label for="article_manufactured_at">Fecha de Fabricación</label><br/>
  <%= datetime_select 'article', 'manufactured_at' %></p>
</div>

<div class="field">
  <p><label for="article_reference">Referencia</label><br/>
  <%= text_field 'article', 'reference' %></p>
</div>

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

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

<div class="field">
  <p><label for="article_size">Talla</label><br/>
  <%= text_field 'article', 'size' %></p>
</div>

<div class="field">
  <% if @article.cover_image.exists? then %>
    <dd><%= image_tag @article.cover_image.url %></dd>
  <% else %>
    <p>No existe imagen para el artículo. Por favor añada una. </p>
  <% end %>
  <p><label for="article_cover_image">Imagen</label><br/>
  <%= file_field 'article', :cover_image %></p>
</div>

```

Fichero: app/views/admin/article/edit.html.erb

```

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

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

```

Fichero: app/views/admin/article/index.html.erb

```

<table>
  <tr>
    <th><a href="?sort_by=manufacturer_id">Fabricante</a></th>
    <th><a href="?sort_by=title">Nombre</a></th>
    <th><a href="?sort_by=reference">Referencia</a></th>
    <th colspan="3"></th>
  </tr>

  <% @articles.each do |article| %>
    <tr>
      <td><%= link_to article.manufacturer.name, controller: "manufacturer", :action => 'show', :id =>

```

```

article.manufacturer %></td>
<td><%= link_to article.title, :action => 'show', :id => article %> </td>
<td><%= h article.reference %></td>
<td><%= link_to 'Mostrar', :action => 'show', :id => article %></td>
<td><%= link_to 'Editar', :action => 'edit', :id => article %></td>
<td><%= button_to 'Borrar', { :action => 'destroy', :id => article },
      data: { confirm: "Seguro que quieres borrar el artículo #{article.title}?" } %></td>
</tr>
<% end %>
</table>
<br/>
<%= will_paginate @articles, :page_links => false, :link_separator => ' | ',
      :previous_label => 'Anterior', :next_label => 'Siguiente' %>
<p><%= link_to 'Añadir nuevo artículo', :action => 'new' %></p>

```

Fichero: app/views/admin/article/new.html.erb

```

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

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

```

Fichero: app/views/admin/article/show.html.erb

```

<dl>
  <dt>Nombre</dt>
  <dd><%= @article.title %></dd>
  <dt>Fabricante</dt>
  <dd><%= @article.manufacturer.name %></dd>
  <dt>Fecha de fabricación</dt>
  <dd><%= @article.manufactured_at.strftime("%d/%m/%Y at %I:%M%p") %></dd>
  <dt>Proveedores</dt>
  <dd><%= @article.supplier_names %></dd>
  <dt>Referencia</dt>
  <dd><%= @article.reference %></dd>
  <dt>Descripción</dt>
  <%= RedCloth.new(@article.blurb).to_html.html_safe if @article.blurb %>
  <dt>Precio</dt>
  <dd><%= @article.price %></dd>
  <dt>Talla</dt>
  <dd><%= @article.size %></dd>
  <dt>Imagen</dt>
  <% if @article.cover_image.exists? then %>
    <dd><%= image_tag @article.cover_image.url %></dd>
  <% else %>
    <dd><%= image_tag '/missing.png' %></dd>
  <p> No existe imagen para el artículo. Por favor añada una. </p>
  <% end %>
</dl>

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

```

5.4.6. Test

Vamos a preparar el fichero de tests, en primer lugar el del modelo:

Fichero: test/models/article_test.rb

```
require 'test_helper'

class ArticleTest < ActiveSupport::TestCase
  fixtures :suppliers, :manufacturers, :articles, :articles_suppliers

  test "failing_create" do
    article = Article.new
    assert_equal false, article.save
    assert_equal 8, article.errors.count
    assert article.errors[:title]
    assert article.errors[:manufacturer]
    assert article.errors[:suppliers]
    assert article.errors[:manufactured_at]
    assert article.errors[:reference]
    assert article.errors[:blurb]
    assert article.errors[:size]
    assert article.errors[:price]
  end

  test "create" do
    article = Article.new(
      :title => 'Zapatillas Running',
      :suppliers => Supplier.all,
      :manufacturer_id => Manufacturer.find(1).id,
      :manufactured_at => Time.now,
      :reference => '123-123-123-1',
      :blurb => 'Unas zapatillas rojas',
      :size => 42,
      :price => 90.5
    )
    assert article.save
  end

  test "has_many_and_belongs_to_mapping" do
    adidas = Manufacturer.find_by_name("Adidas");
    count = adidas.articles.count
    article = Article.new(
      :title => 'Gorra',
      :suppliers => [Supplier.find_by_first_name_and_last_name('Pepe', 'Mateo'),
        Supplier.find_by_first_name_and_last_name('Toni', 'Gavira')],
      :manufactured_at => Time.now,
      :reference => '123-123-123-x',
      :blurb => 'Una gorra roja y una raya amarilla',
      :size => 40,
      :price => 55.5
    )
    adidas.articles << article
    adidas.reload
    article.reload
    assert_equal count + 1, adidas.articles.count
    assert_equal 'Adidas', article.manufacturer.name
  end

  test "has_many_and_belongs_to_many_suppliers_mapping" do
    article = Article.new(
      :title => 'Gorra',
```

```

:suppliers => [Supplier.find_by_first_name_and_last_name('Pepe', 'Mateo'),
  Supplier.find_by_first_name_and_last_name('Toni', 'Gavira')],
:manufacturer_id => Manufacturer.find_by_name("Adidas").id,
:manufactured_at => Time.now,
:reference => '123-123-123-x',
:blurb => 'Una gorra roja y una raya amarilla',
:size => 40,
:price => 55.5
)
assert article.save
article.reload
assert_equal 2, article.suppliers.count
assert_equal 4, Supplier.find_by_first_name_and_last_name('Pepe', 'Mateo').articles.count
assert_equal 3, Supplier.find_by_first_name_and_last_name('Toni', 'Gavira').articles.count
end
end

```

Crearemos ahora el fichero para el test de Integración de administración de la entidad principal:

```
ine@gii ~/projects/eshop6 $ rails generate integration_test ArticleAdministration
```

Y modificamos el fichero correspondiente:

```

Fichero: test/integration/article_administration_test.rb

require 'test_helper'

class ArticleAdministrationTest < ActionDispatch::IntegrationTest

  test "article_aministration" do
    manufacturer = Manufacturer.create(:name => 'Fabricante de UCASport')
    supplier = Supplier.create(:first_name => 'Pepe', :last_name => 'Gonzalez')
    george = new_session_as(:george)

    new_article_ruby = george.add_article :article => {
      :title => 'Zapatillas Joma',
      :manufacturer_id => manufacturer.id,
      :supplier_ids => [supplier.id],
      :manufactured_at => Time.now,
      :reference => '123-123-123-X',
      :blurb => 'Nuevas zapatillas Joma',
      :size => 40,
      :price => 45.5
    }

    george.list_articles
    george.show_article new_article_ruby

    george.edit_article new_article_ruby, :article => {
      :title => 'Zapatillas Joma - año 2017',
      :manufacturer_id => manufacturer.id,
      :supplier_ids => [supplier.id],
      :manufactured_at => Time.now,
      :reference => '123-123-123-X',
      :blurb => 'Nuevas zapatillas Joma. Edición Limitada',
      :size => 42,
    }
  end
end

```

```

    :price => 50
  }

  bob = new_session_as(:bob)
  bob.delete_article new_article_ruby
end

private

module ArticleTestDSL
  attr_writer :name

  def add_article(parameters)
    supplier = Supplier.first
    manufacturer = Manufacturer.first
    get '/admin/article/new'
    assert_response :success
    assert_template 'admin/article/new'
    assert_select 'select#article_manufacturer_id' do
      assert_select "option[value=\"#{manufacturer.id}\"]", manufacturer.name
    end
    assert_select "select[name=\"article[supplier_ids][]\"]" do
      assert_select "option[value=\"#{supplier.id}\"]", supplier.name
    end
    post '/admin/article/create', parameters
    assert_response :redirect
    follow_redirect!
    assert_response :success
    assert_template 'admin/article/index'
    page = Article.all.count / 5 + 1
    get "/admin/article/index/?page=#{page}"
    assert_select 'td', parameters[:article][:title]
    article = Article.find_by_title(parameters[:article][:title])
    return article;
  end

  def edit_article(article, parameters)
    get "/admin/article/edit?id=#{article.id}"
    assert_response :success
    assert_template 'admin/article/edit'
    post "/admin/article/update?id=#{article.id}", parameters
    assert_response :redirect
    follow_redirect!
    assert_response :success
    assert_template 'admin/article/show'
  end

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

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

  def list_articles
    get '/admin/article/index'
    assert_response :success
    assert_template 'admin/article/index'
  end
end

```

```
end

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

Probamos que todo va bien:

```
ine@gii ~/projects/eshop6 $ rake test TEST=test/models/article_test.rb
Run options: --seed 20804

# Running:

....

Finished in 1.510454s, 2.6482 runs/s, 11.2549 assertions/s.

4 runs, 17 assertions, 0 failures, 0 errors, 0 skips

ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/article_administration_test.rb
Run options: --seed 39823

# Running:

.

Finished in 1.743649s, 0.5735 runs/s, 0.0000 assertions/s.

1 runs, 0 assertions, 0 failures, 0 errors, 0 skips
```

5.4.7. Dificultades encontradas

En este sprint la principal dificultad la hemos tenido en establecer la relación entre la entidad principal y las secundarias. Del mismo modo el test de integración ha sido novedoso para nosotros y tuvimos algún problema para entender su funcionamiento al principio.

5.4.8. Objetivos alcanzados

Con este sprint hemos conseguido poder visualizar el CRUD (Create, Read, Update and Delete) de Artículo y su relación con las entidades secundarias. Asimismo se ha añadido la inserción de imágenes y la paginación del listado.

5.5. Sprint 5 (Creación del catálogo)

Ahora que ya tenemos los CRUD necesarios para nuestra tienda, lo que nos hace falta es tener un catálogo operativo para que los clientes puedan ver nuestros artículos a partir de la información almacenada gracias a lo creado anteriormente. Vamos a crear el catálogo junto a su test correspondiente.

5.5.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Vista y Controlador	0	100	0	0	0	0	0
Test	0	40	0	0	0	0	0
Total	0	140	0	0	0	0	0

Tabla 5.5: Sprint 5 backlog (minutos)

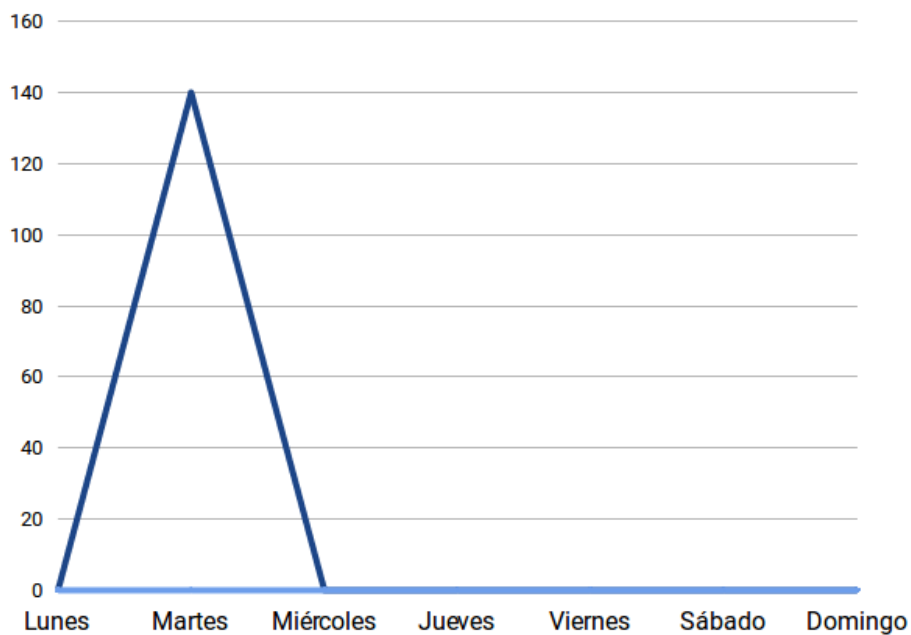


Figura 5.5: Diagrama burndown

5.5.2. Gemas utilizadas

En este sprint no hemos añadido gemas nuevas.

5.5.3. Ficheros de migración

En este sprint no hemos añadido ficheros de migración.

5.5.4. Modelos ORM

En este sprint no ha sido necesario crear ningún modelo, ya que la información de la base de datos que vamos a mostrar ya está gestionada en otros modelos (article, manufacturer y supplier).

5.5.5. Vistas y controladores

Generamos el controlador y lo modificamos:

```
ine@gii ~/projects/eshop6 $ rails generate controller Catalog index show latest
```

El fichero debe quedar:

Fichero: app/controllers/catalog_controller.rb

```
class CatalogController < ApplicationController
  #Descomentar la siguiente línea al implementar el carrito
  #before_filter :initialize_cart, :except => :show

  #Descomentar la siguiente línea al implementar la autenticación
  #before_filter :require_no_user

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

  def index
    @articles = Article.order("articles.id desc").includes(:suppliers, :manufacturer).paginate(:page => params[:page],
:per_page => 5)
    @page_title = 'Catálogo'
  end

  def latest
    @articles = Article.latest 5
    @page_title = 'Últimos artículos'
  end
end
```

Hay que acordarse de descomentar las últimas líneas del Fichero: **app/models/article.rb** (**apartado 5.4.3**), donde está definido el método latest:

Fichero: app/models/article.rb (añadir o descomentar el siguiente método)

```
...
...
...
def self.latest(num)
  all.order("articles.id desc").includes(:suppliers, :manufacturer).limit(num)
end
```

Y ahora las vistas:

Fichero: app/views/catalog/_articles.html.erb

```
<dl id = 'articles'>
  <% for article in @articles %>
    <dt>
      <%= link_to article.title, :action => 'show', :id => article %>
    </dt>
    <% for supplier in article.suppliers %>
      <dd><%= supplier.last_name %>, <%= supplier.first_name %></dd>
    <% end %>
    <dd><%= sprintf("Precio: %.2f €, Talla: %d", article.price, article.size)
    %> </dd>
    <dd><small>Fabricante: <%= article.manufacturer.name %></small></dd>
  <% end %>
</dl>
```

Fichero: app/views/catalog/index.html.erb

```
<%= render :partial => 'articles' %>
<%= will_paginate @articles, :page_links => false, :link_separator => ' | ',
      :previous_label => 'Anterior', :next_label => 'Siguiente' %>
```

Fichero: app/views/catalog/latest.html.erb

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

Fichero: app/views/catalog/show.html.erb

```
<h2>Distribuido por: <%= @article.supplier_names %></h2>
<% if @article.cover_image.exists? then %>
  <dd><%= image_tag @article.cover_image.url %></dd>
<% else %>
  <p>No hay imagen disponible.</p>
<% end %>
<dl>
  <dt>Precio</dt>
  <dd><%= sprintf("%.2f", @article.price) %></dd>
  <dt>Talla</dt>
  <dd><%= @article.size %></dd>
  <dt>Fabricante</dt>
  <dd><%= @article.manufacturer.name %></dd>
```

```
<dt>Descripción</dt>
<%= RedCloth.new(@article.blurb).to_html.html_safe if @article.blurb %>
</dl>

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

5.5.6. Test

Vamos a realizar un test de integración, primero crearemos el fichero correspondiente:

```
ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/browsing_and_searching_test.rb
```

El fichero queda de la siguiente forma:

```
Fichero: test/integration/browsing_and_searching_test.rb

require 'test_helper'

class BrowsingAndSearchingTest < ActionDispatch::IntegrationTest
  fixtures :manufacturers, :suppliers, :articles, :articles_suppliers

  test "browse" do
    jill = new_session_as :jill
    jill.index
    jill.second_page
    jill.article_details 'Articulo_9'
    jill.latest_articles
  end

  module BrowsingTestDSL
    include ERB::Util
    attr_writer :name

    def index
      get '/catalog/index'
      assert_response :success
      assert_select 'dl#articles' do
        assert_select 'dt', :count => 5
      end
      assert_select 'dt' do
        assert_select 'a', 'Articulo_10'
      end
      check_article_links
    end

    def second_page
      get '/catalog/index?page=2'
      assert_response :success
      assert_template 'catalog/index'
      assert_equal Article.find_by_title('Articulo_1'),
        assigns(:articles).last
      check_article_links
    end

    def article_details(title)
      @article = Article.where(:title => title).first
      get "/catalog/show/#{@article.id}"
      assert_response :success
    end
  end
end
```

```

assert_template 'catalog/show'
assert_select 'div#content' do
  assert_select 'h1', @article.title
  assert_select 'h2', "Distribuido por:#{@article.suppliers.map{|a| a.name}.join(", ")}"
end
end

def latest_articles
  get '/catalog/latest'
  assert_response :success
  assert_template 'catalog/latest'
  assert_select 'dl#articles' do
    assert_select 'dt', :count => 5
  end
  @articles = Article.latest(5)
  @articles.each do |a|
    assert_select 'dt' do
      assert_select 'a', a.title
    end
  end
end
end

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

```

Y finalmente ejecutamos el test:

```

ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/browsing_and_searching_test.rb
Run options: --seed 49760

# Running:

.

Finished in 2.617550s, 0.3820 runs/s, 0.0000 assertions/s.

1 runs, 0 assertions, 0 failures, 0 errors, 0 skips

```

5.5.7. Dificultades encontradas

En este sprint no hemos encontrado dificultades. Lo más complicado, por destacar algo, ha sido la creación de las vistas.

5.5.8. Objetivos alcanzados

Con este sprint hemos conseguido crear el catálogo de nuestros artículos y el listado de los últimos artículos añadidos. Además, hemos realizado todos los tests de integración correspondientes.

5.6. Sprint 6 (Carrito de la compra)

Una vez creado el catálogo es necesario añadir una funcionalidad para que podamos seleccionar los artículos mostrados para poder comprarlos. Podremos añadir y eliminar artículos al carrito, así como vaciar de una sola vez dicho carrito. Para ello vamos a necesitar integrar en nuestro proyecto jQuery.

5.6.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de Migración	0	30	0	0	0	0	0
Modelo	0	60	0	0	0	0	0
Vista y Controlador	0	40	40	40	0	0	0
Test	0	10	10	30	0	0	0
Total	0	140	50	70	0	0	0

Tabla 5.6: Sprint 6 backlog (minutos)

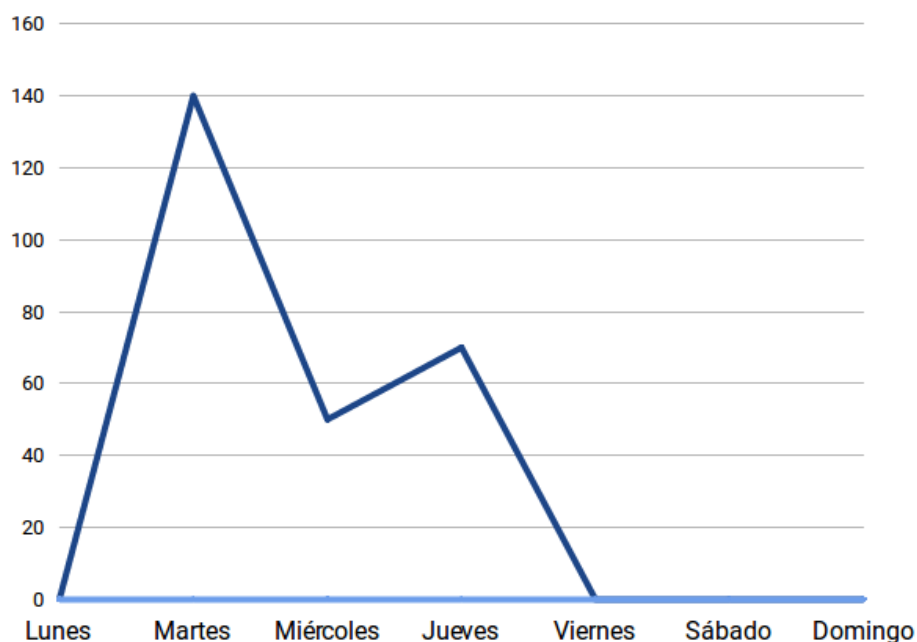


Figura 5.6: Diagrama burndown

5.6.2. Gemas utilizadas

En este sprint hemos necesitado añadir la gema:

- jquery-rails, versión 4.0.5

5.6.3. Ficheros de migración

Vamos a introducir el comando que permite crear las plantillas del modelo, del test del modelo y del fichero de migración del carrito de la compra:

```
ine@gii ~/projects/eshop6 $ rails generate model Cart
ine@gii ~/projects/eshop6 $ rails generate model CartItem
```

Editamos pues los ficheros de migración:

Fichero: db/migrate/20170427122245_create_carts.rb

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

Fichero: db/migrate/20170427122257_create_cart_items.rb

```
class CreateCartItem < ActiveRecord::Migration
  def change
    create_table :cart_items do |t|
      t.integer :article_id
      t.integer :cart_id
      t.float :price
      t.integer :amount
      t.timestamps
    end
  end
end
```

Para que se creen convenientemente las tablas en la base de datos:

```
ine@gii ~/projects/eshop6 $ rake db:migrate
```


5.6.4. Modelos ORM

Los ficheros correspondiente al modelo quedarían de la siguiente forma:

Fichero: app/models/cart.rb

```
class Cart < ActiveRecord::Base
  has_many :cart_items
  has_many :articles, :through => :cart_items

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

  def remove(article_id)
    ci = cart_items.where(article_id: article_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
```

Fichero: app/models/cart_item.rb

```
class CartItem < ActiveRecord::Base
  belongs_to :cart
  belongs_to :article
end
```

5.6.5. Vistas y controladores

Tras ejecutar:

```
ine@gii ~/projects/eshop6 $ rails generate controller Cart
```

Editamos el fichero del controlador del carrito:

Fichero: app/controllers/cart_controller.rb

```
class CartController < ApplicationController
  before_filter :initialize_cart

  def add
    @article = Article.find params[:id]
    @page_title = 'Añadir Artículo'
    if request.post?
      @item = @cart.add params[:id]
      flash[:cart_notice] = "Artículo <em>#{@item.article.title}</em> añadido satisfactoriamente al carrito."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'add', :template => 'cart/add'
    end
  end

  def remove
    @article = Article.find params[:id]
    @page_title = 'Eliminar Artículo'
    if request.post?
      @item = @cart.remove params[:id]
      flash[:cart_notice] = "Artículo <em>#{@item.article.title}</em> eliminado correctamente del carrito."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'remove'
    end
  end

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

Seguidamente vamos a mostrar los ficheros correspondientes a las vistas:

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

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

<h3>Tu carrito de la Compra</h3>
<ul>
  <% for item in @cart.cart_items %>
    <li id="cart_item_<%= item.article.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 %>

```

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

```

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

```

Fichero: app/views/cart/_item.html.erb

```

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

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

(<%= link_to '<b>-</b>', :controller => 'cart', :action => 'remove', :id => item.article.id %>)

```

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

```

<strong>Por favor confirma que quieres añadir <em><%= @article.title %></em> a tu 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>

```

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

```

<strong>Por favor confirma que quieres vaciar el 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>

```

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

```

<strong>Por favor confirma para eliminar <em><%= @article.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>

```

Hay que acordarse de descomentar 2 líneas que indicamos del Fichero: **app/models/article.rb** (apartado 5.4.3):

Fichero: app/models/article.rb (añadir o descomentar estas 2 líneas)

```
...
...
...
has_many :cart_items
has_many :carts, :through => :cart_items
...
...
...
```

Asimismo también en el fichero **app/controllers/catalog_controller.rb** (apartado 5.5.5) tenemos que descomentar la línea que indicamos:

Fichero: app/controllers/catalog_controller.rb (añadir o descomentar esta línea)

```
...
...
...
before_filter :initialize_cart, :except => :show
...
...
...
```

5.6.6. Test

Vamos con el controlador del carrito de la compra:

Fichero: test/controllers/cart_controller_test.rb

```
require 'test_helper'

class CartControllerTest < ActionController::TestCase
  fixtures :suppliers, :manufacturers, :articles

  test "add" do
    assert_difference(CartItem, :count) do
      post :add, :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 :add, :id => 4
    assert_equal [Article.find(4)], Cart.find(@request.session[:cart_id]).articles

    post :remove, :id => 4
    assert_equal [], Cart.find(@request.session[:cart_id]).articles
  end
end
```

```
test "clear" do
  post :add, :id => 4
  assert_equal [Article.find(4)], Cart.find(@request.session[:cart_id]).articles

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

Vamos a ejecutarlo:

```
ine@gji ~/projects/eshop6 $ rake test TEST=test/controllers/cart_controller_test.rb
Run options: --seed 4843

# Running:

...

Finished in 0.654624s, 4.5828 runs/s, 18.3311 assertions/s.

3 runs, 12 assertions, 0 failures, 0 errors, 0 skips
```

5.6.7. Dificultades encontradas

En este sprint no hemos encontrado dificultades destacables.

5.6.8. Objetivos alcanzados

Ya tenemos nuestro carrito de la compra creado, acumula los artículos con su precio pudiendo añadir, eliminar y borrar todo el carrito.

5.7. Sprint 7 (Facturación y pedidos)

En esta iteración se han desarrollado dos CRUDs, el primero de ellos para gestionar el proceso de facturación y conexión con la pasarela de pago, y el segundo que permite listar, consultar y cerrar las órdenes que son creadas en las facturaciones.

Como paso previo hemos tenido que registrarnos en el entorno de test de la pasarela de pago Authorize.Net.

5.7.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de Migración	0	70	0	0	0	0	0
Modelo	0	40	30	0	0	0	0
Vista y Controlador	0	30	40	30	20	0	0
Test	0	30	20	0	0	0	0
Total	0	170	90	30	20	0	0

Tabla 5.7: Sprint 7 backlog (minutos)

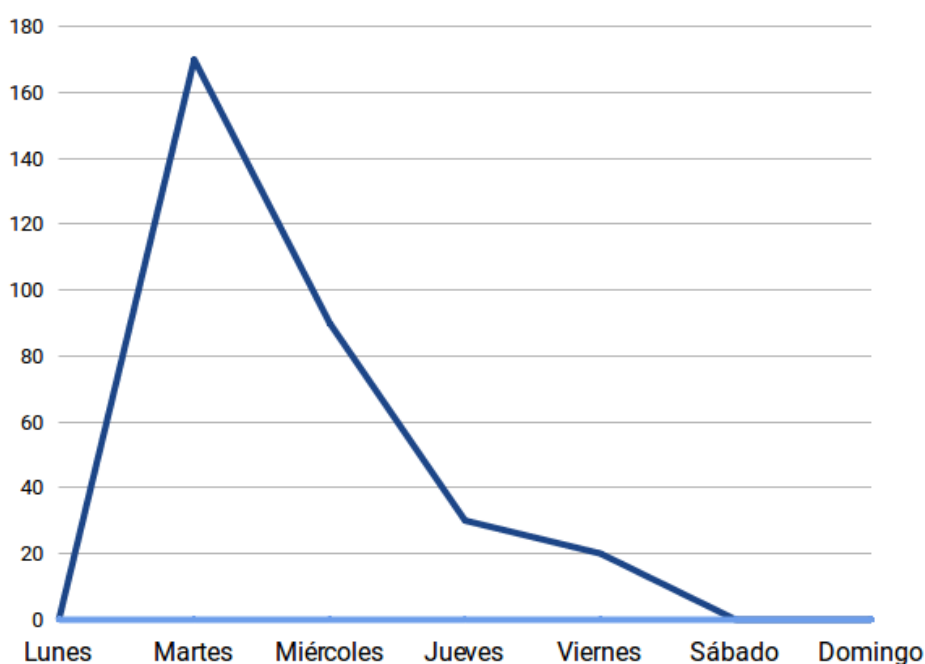


Figura 5.7: Diagrama burndown

5.7.2. Gemas utilizadas

En este sprint hemos necesitado añadir las gemas:

- activemerchant, versión 1.56.0
- countries, versión 0.9.3
- country_select, versión 1.3.1

5.7.3. Ficheros de migración

Vamos a introducir el comando que permite crear las plantillas del modelo, del test del modelo y del fichero de migración de pedido:

```
ine@gii ~/projects/eshop6 $ rails generate model Order
ine@gii ~/projects/eshop6 $ rails generate model OrderItem
```

Editamos pues los ficheros de migración:

Fichero: db/migrate/20170502091305_create_orders.rb

```
class CreateOrders < ActiveRecord::Migration
  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
```

Fichero: db/migrate/20170502091320_create_order_items.rb

```
class CreateOrderItems < ActiveRecord::Migration
  def self.up
    create_table :order_items do |t|
      t.integer :article_id
      t.integer :order_id
      t.float :price
      t.integer :amount
      t.timestamps
    end
  end
end
```

```

end

say_with_time 'Adding foreign 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

```

Para que se creen convenientemente las tablas en la base de datos:

```
ine@gii ~/projects/eshop6 $ rake db:migrate
```

5.7.4. Modelos ORM

Los ficheros correspondiente al modelo quedarían de la siguiente forma:

Fichero: app/models/order.rb

```

class Order < ActiveRecord::Base
  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 :articles, :through => :order_items

  validates_presence_of :order_items,
    :message => 'Tu carrito de la compra está vacío! ' +
      'Por favor añade al menos un artículo al carrito antes de enviar la orden.'
  validates_format_of :email, :with => /\A([^\s]+)@((?:[-a-z0-9]+\.)+[a-z]{2,})\Z/i, :message => 'El formato del
e-mail no es correcto'
  validates_length_of :phone_number, :in => 7..20, :message => 'La longitud del número de teléfono debe ser entre 7 y
20 caracteres'

  validates_length_of :ship_to_first_name, :in => 2..255, :message => 'La longitud del nombre debe ser entre 2 y 255
caracteres'
  validates_length_of :ship_to_last_name, :in => 2..255, :message => 'La longitud de los apellidos debe ser entre 2 y
255 caracteres'
  validates_length_of :ship_to_address, :in => 2..255, :message => 'La longitud de la dirección debe ser entre 2 y 255
caracteres'
  validates_length_of :ship_to_city, :in => 2..255, :message => 'La longitud de la ciudad debe ser entre 2 y 255
caracteres'
  validates_length_of :ship_to_postal_code, :in => 2..255, :message => 'La longitud del código postal debe ser entre 2 y
255 caracteres'
  validates_length_of :ship_to_country_code, :in => 2..255, :message => 'La longitud del código de país debe ser entre
2 y 255 caracteres'

  validates_length_of :customer_ip, :in => 7..15, :message => 'La longitud de la dirección ip del cliente debe ser entre 7
y 15 caracteres'
  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, :message => 'La longitud del número de tarjeta debe
ser entre 13 y 19 caracteres'
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, :message => 'La longitud del código de
verificación de la tarjeta debe ser entre 3 y 4 caracteres'

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 'Una orden cerrada no puede ser procesada de nuevo' if self.closed?
    active_merchant_payment
  rescue => e
    logger.error("La orden #{id} ha fallado debido a que se ha lanzado la excepción: #{e}.")
    self.error_message = "Excepción lanzada: #{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 => 'Compras en UCASport',
  :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 => '637vLCZkhM',
    :password => '3Cwe2ZnF7478up6r'
    # the statement ":test = 'true'" tells the gateway to 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

```

Fichero: app/models/order_item.rb

```

class OrderItem < ActiveRecord::Base
  belongs_to :order
  belongs_to :article

  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

Para obtener ficheros de controladores de Checkout y admin/Order, ejecutar:

```
ine@gii ~/projects/eshop6 $ rails generate controller Checkout index submit_order thank_you
ine@gii ~/projects/eshop6 $ rails generate controller admin/order show index close
```

Editamos ambos controladores convenientemente:

Fichero: app/controllers/checkout_controller.rb

```
class CheckoutController < ApplicationController
  before_filter :initialize_cart, :only => :index

  def index
    @order = Order.new
    @page_title = 'Pago'
    if @cart.articles.empty?
      flash[:notice] = 'Tu carrito de la compra está vacío! ' +
        'Por favor añade al menos un artículo al carrito antes de enviar la orden.'
      redirect_to :controller => 'catalog'
    end
  end

  def submit_order
    @cart = Cart.find(params[:cart][:id])
    @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 = 'Pago'
    populate_order

    if @order.save
      if @order.process
        flash[:notice] = 'Tu pedido ha sido recibido y será procesado inmediatamente.'
        session[:order_id] = @order.id
        @cart.cart_items.destroy_all # empty shopping cart
        redirect_to :action => 'thank_you'
      else
        flash[:notice] = "Error al enviar tu orden '#{@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(:article_id => cart_item.article_id,
        :price => cart_item.price,
        :amount => cart_item.amount)
    end
  end
end
```

```

    @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

```

Y también...

Fichero: app/controllers/admin/order_controller.rb

```

class Admin::OrderController < ApplicationController
  def close
    order = Order.find(params[:id])
    order.close
    flash[:notice] = "La orden ##{order.id} ha sido cerrada."
    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?
      @status = ''
      conditions = nil
    else
      conditions = "status = '#{@status}'"
    end
    @orders = Order.where(conditions).paginate(:page => params[:page], :per_page => 10)

    if @status == 'open'
      @status = 'abiertos'
    elsif @status == 'processed'
      @status = 'procesados'
    elsif @status == 'closed'
      @status = 'cerrados'
    elsif @status == 'failed'
      @status = 'fallidos'
    end

    @page_title = "Mostrando pedidos #{@status}"
  end
end

```

Seguidamente vamos a mostrar los ficheros correspondientes a las vistas:

Fichero: app/views/checkout/index.html.erb

```

<% if @order.errors.any? %>
  <div id="errorExplanation">
    <h2><%= pluralize(@order.errors.count, "error") %> prohíben guardar esta orden:</h2>
  </div>
</%>

```

```

<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>Tu pedido será mostrado en el carro 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">Email</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, ['ES'], :iso_codes => true) %></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 "Enviar pedido" %></p>
</div>
<% end %>

```

Fichero: app/views/checkout/thank_you.html.erb

Para futuras referencias este es el número de factura: <%= session[:order_id] %>

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

```

<p>
  <strong>Listar pedidos: <%= 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>

```

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

```

<% if @orders == [] %>
  <h2><%= "No hay pedidos con estado: '#{@status}'." %></h2>
<% else %>
  <table>
    <tr>
      <th>ID</th>
      <th>Estado</th>
      <th>Precio total</th>
      <th>Cantidad</th>
      <th>Creado el</th>
      <th>Actualizado el</th>
    </tr>
    <% for order in @orders %>
      <tr>
        <td align="center"><%= order.id %></td>
        <td align="center"><%= order.status[0].capitalize + order.status[1..order.status.length-1] %></td>
        <td align="center"><%= order.total %></td>
        <td align="center"><%= order.amount %></td> <!-- amount -->
        <td align="center"><%= order.created_at.strftime("%d-%m-%Y %I:%M") %></td>
        <td align="center"><%= order.updated_at.strftime("%d-%m-%Y %I:%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' %>
```

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

```
<h2>Información de contacto</h2>
<dl>
  <dt>E-mail</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><%= Country.find_country_by_alpha2(@order.ship_to_country_code).name %></dd>
</dl>

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

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

<h2>Estado del pedido</h2>
<dl>
  <dt>Estado</dt>
  <dd><%= @order.status[0].capitalize + @order.status[1..@order.status.length-1] %></dd>
  <% if @order.failed? %>
    <dt>Error</dt>
    <dd><%= @order.error_message %></dd>
  <% end %>
</dl>

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

5.7.6. Test

Vamos con el test del modelo de pedido:

Fichero: 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 => '3',
      :card_expiration_year => '2018',
      :card_verification_value => '000'
    )

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

    order.order_items << OrderItem.new(:article_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_name]
    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

```

Crearemos ahora el fichero para el test de Integración de facturación:

```
ine@gii ~/projects/eshop6 $ rails generate integration_test Checkout
```

Y modificamos el fichero correspondiente:

Fichero: test/integration/checkout_test.rb

```

require 'test_helper'

class CheckoutTest < ActionDispatch::IntegrationTest

  fixtures :manufacturers, :suppliers, :articles

  test "empty_cart_shows_error_message" do
    get '/checkout'
    assert_response :redirect
    assert_redirected_to :controller => 'catalog'
    assert_equal flash[:notice], 'Tu carrito de la compra está vacío! ' +
      'Por favor añade al menos un artículo al carrito antes de enviar la orden.'
  end

  test "submitting_order" do
    post '/cart/add', :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', :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 => '3',
      :card_expiration_year => '2018',
      :card_verification_value => '000'
    }
  end
end

```

```

}

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

```

Vamos a ejecutar primero el del modelo de pedido:

```

ine@gii ~/projects/eshop6 $ rake test TEST=test/models/order_test.rb
Run options: --seed 44704

# Running:

opening connection to apitest.authorize.net:443...
opened
starting SSL for apitest.authorize.net:443...
SSL established
< - "POST /xml/v1/request.api HTTP/1.1\r\nContent-Type: text/xml\r\nAccept-Encoding: gzip;q=1.0,deflate;q=0.6,identity;q=0.3\r\nAccept: */*\r\nUser-Agent:
Ruby\r\nConnection: close\r\nHost: apitest.authorize.net\r\nContent-Length: 1399\r\n\r\n"
< - "<?xml version='1.0' encoding='UTF-8'>\n<createTransactionRequest
xmlns='\"AnetApi/xml/v1/schema/AnetApiSchema.xsd\"'\n>\n<merchantAuthentication>\n<name>637vLCZkhM</name>\n<transactionKey>3Cwe2ZnF7478up6r</transactionKey>
\n</merchantAuthentication>\n<refid>980190963</refid>\n<transactionRequest>\n<transactionType>authCaptureTransaction</transactionType>\n<amount>465.75</amount>
\n<payment>\n<creditCard>\n<cardNumber>40070000000027</cardNumber>\n<expirationDate>03/2018</expirationDate>\n<cardCode>000</cardCode>\n</creditCard>\n</pay
ment>\n<order>\n<invoiceNumber>980190963</invoiceNumber>\n<description>Compras en
UCASport</description>\n</order>\n<customer>\n<email>email@email.com</email>\n<customer>\n<billTo>\n<firstName>Firstname</firstName>\n<lastName>Lastname</las
tName>\n<address>Address</address>\n<city>City</city>\n<state>n/a</state>\n<zip>00000</zip>\n<country>ES</country>\n<phoneNumber>666112233</phoneNumber>\n</
billTo>\n<shipTo>\n<firstName>Firstname</firstName>\n<lastName>Lastname</lastName>\n<address>Address</address>\n<city>City</city>\n<state>\n<zip>00000</zip>\n<c
ountry>ES</country>\n</shipTo>\n<customerIP>127.0.0.1</customerIP>\n<cardholderAuthentication>\n<authenticationIndicator>\n<cardholderAuthenticationValue>\n</ca
rdholderAuthentication>\n<transactionSettings>\n<userFields>\n<userField>\n<name>x_currency_code</name>\n<value>EUR</value>\n</userField>\n</userFields>\n</trans
actionRequest>\n</createTransactionRequest>\n"
-> "HTTP/1.1 200 OK\r\n"
-> "Cache-Control: private\r\n"
-> "Content-Type: application/xml; charset=utf-8\r\n"
-> "Server: Microsoft-IIS/7.5\r\n"
-> "Access-Control-Allow-Origin: *\r\n"
-> "Access-Control-Allow-Methods: PUT,OPTIONS,POST,GET\r\n"
-> "Access-Control-Allow-Headers: x-requested-with,cache-control,content-type,origin,method,SOAPAction\r\n"
-> "X-Cnection: close\r\n"
-> "X-Powered-By: ASP.NET\r\n"
-> "Content-Length: 999\r\n"
-> "Date: Fri, 05 May 2017 09:48:18 GMT\r\n"
-> "Connection: close\r\n"
-> "\r\n"
reading 999 bytes...
-> "\xEF\xBB\xBF<?xml version='1.0' encoding='utf-8'>\n<createTransactionResponse xmlns:xsi='\"http://www.w3.org/2001/XMLSchema-instance\"'
xmlns:xsd='\"http://www.w3.org/2001/XMLSchema\"'\n
xmlns='\"AnetApi/xml/v1/schema/AnetApiSchema.xsd\"'\n>\n<refid>980190963</refid>\n<messages>\n<resultCode>Ok</resultCode>\n<message>\n<code>I00001</code>\n<text>Successful.</
text>\n</message>\n</messages>\n<transactionResponse>\n<responseCode>1</responseCode>\n<authCode>11Y2W3</authCode>\n<avsResultCode>Y</avsResultCode>\n<cvvResultCode>P</c
vvResultCode>\n<cvvResultCode>2</cvvResultCode>\n<transId>60023243745</transId>\n<refTransID
/>\n<transHash>5D6185B6D05E637BEE23A2A3054CABAD</transHash>\n<testRequest>0</testRequest>\n<accountNumber>XXXX0027</accountNumber>\n<accountType>Visa</account
Type>\n<messages>\n<message>\n<code>1</code>\n<description>This transaction has been
approved.</description>\n</message>\n</messages>\n<userFields>\n<userField>\n<name>x_currency_code</name>\n<value>EUR</value>\n</userField>\n</userFields>\n<transHashSha2
/>\n</transactionResponse>\n</createTransactionResponse>\n"
read 999 bytes
Conn close
..

Finished in 3.424736s, 0.5840 runs/s, 7.0078 assertions/s.

2 runs, 24 assertions, 0 failures, 0 errors, 0 skips

```

Y ahora el test de integración de facturación:

```

ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/checkout_test.rb
Run options: --seed 5050

# Running:

opening connection to apitest.authorize.net:443...
opened
starting SSL for apitest.authorize.net:443...
SSL established
< - "POST /xml/v1/request.api HTTP/1.1\r\nContent-Type: text/xml\r\nAccept-Encoding: gzip;q=1.0,deflate;q=0.6,identity;q=0.3\r\nAccept: */*\r\nUser-Agent:
Ruby\r\nConnection: close\r\nHost: apitest.authorize.net\r\nContent-Length: 1409\r\n\r\n"

```

```

< ?xml version="1.0" encoding="UTF-8"?><createTransactionRequest
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd"><merchantAuthentication><name>637vLCZkhM</name><transactionKey>3Cwe2ZnF7478up6</transactionKey>
</merchantAuthentication><refid>980190964</refid><transactionRequest><transactionType>authCaptureTransaction</transactionType><amount>100.00</amount>
<payment><creditCard><cardNumber>4007000000027</cardNumber><expirationDate>03/2018</expirationDate><cardCode>000</cardCode></creditCard></pay
ment><order><invoiceNumber>980190964</invoiceNumber><description>Compras en
UCASport</description></order><customer><email>email@email.com</email></customer><billTo><firstName>Firstname</firstName><lastName>Lastname</las
tName><address>Address</address><city>City</city><state>n/a</state><zip>00000</zip><country>COUNTRY</country><phoneNumber>666112233</phoneNumb
er></billTo><shipTo><firstName>Firstname</firstName><lastName>Lastname</lastName><address>Address</address><city>City</city><state></state><zip>00000</zi
p><country>COUNTRY</country></shipTo><customerIP>127.0.0.1</customerIP><cardholderAuthentication><authenticationIndicator></cardholderAuthenticatio
nValue></cardholderAuthentication><transactionSettings><userFields><userField><name>x_currency_code</name><value>EUR</value></userField></userF
ields></transactionRequest></createTransactionRequest></>
-> "HTTP/1.1 200 OK\r\n"
-> "Cache-Control: private\r\n"
-> "Content-Type: application/xml; charset=utf-8\r\n"
-> "Server: Microsoft-IIS/7.5\r\n"
-> "Access-Control-Allow-Origin: \r\n"
-> "Access-Control-Allow-Methods: PUT, OPTIONS, POST, GET\r\n"
-> "Access-Control-Allow-Headers: x-requested-with, cache-control, content-type, origin, method, SOAPAction\r\n"
-> "X-Cnection: close\r\n"
-> "X-Powered-By: ASP.NET\r\n"
-> "Content-Length: 999\r\n"
-> "Date: Fri, 05 May 2017 09:50:00 GMT\r\n"
-> "Connection: close\r\n"
-> "\r\n"
reading 999 bytes...
-> "\xEF\xBB\xBF?xml version="1.0" encoding="utf-8"?><createTransactionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="AnetApi/xml/v1/schema/AnetApiSchema.xsd"><refid>980190964</refid><messages><resultCode>Ok</resultCode><message><code>i00001</code><text>Successful.</
text></message></messages><transactionResponse><responseCode>1</responseCode><authCode>GRL5QL</authCode><avsResultCode>Y</avsResultCode><cvvResultCode>P</cv
vResultCode><cvvResultCode>2</cvvResultCode><transId>60023243775</transId><refTransID
/><transHash>4C064E90652F0BBE999309B887EB9589</transHash><testRequest>0</testRequest><accountNumber>XXXX0027</accountNumber><accountType>Visa</accountTy
pe><messages><message><code>1</code><description>This transaction has been
approved.</description></message></messages><userFields><userField><name>x_currency_code</name><value>EUR</value></userField></userFields><transHashSha2
56</transHashSha256></transactionResponse></createTransactionResponse></>
read 999 bytes
Conn close
.

Finished in 2.985924s, 0.6698 runs/s, 3.6840 assertions/s.

2 runs, 11 assertions, 0 failures, 0 errors, 0 skips

```

5.7.7. Dificultades encontradas

Hemos experimentado algunos problemas con la plataforma Authorize.net, por ser algo totalmente nuevo para nosotros, y entender el funcionamiento de la pasarela de pago, con la ayuda del profesor hemos podido ir solventando dichos problemas.

5.7.8. Objetivos alcanzados

Se ha conseguido con éxito la facturación y la gestión de los pedidos por parte del administrador.

5.8. Sprint 8 (Autenticación)

En esta octava iteración hemos implementado la autenticación del administrador, siendo el único capaz de administrar los artículos, fabricantes, proveedores y pedidos. Para ello se desarrollarán dos CRUDs. El primero gestiona el acceso identificado al administrador de una tienda on-line, mientras que el segundo permite registrar a un administrador, consultar y modificar sus datos.

5.8.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de Migración	0	60	0	0	0	0	0
Modelo	0	30	30	0	0	0	0
Vista y Controlador	0	20	20	30	20	0	0
Test	0	20	0	0	0	0	0
Total	0	130	50	30	20	0	0

Tabla 5.8: Sprint 8 backlog (minutos)

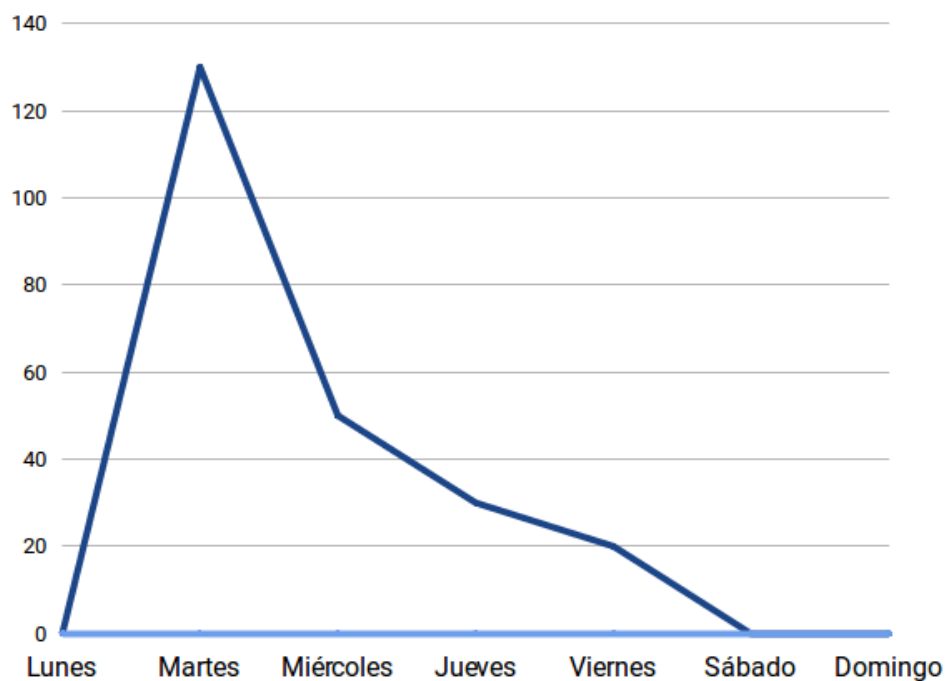


Figura 5.8: Diagrama burndown

5.8.2. Gemas utilizadas

En este sprint hemos necesitado añadir las gemas:

- authlogic, versión 3.4.6: gestiona el acceso identificado mediante contraseña.

5.8.3. Ficheros de migración

Vamos a introducir el comando que permite crear las plantillas del modelo, del test del modelo y del fichero de migración de User y UserSession:

```
ine@gii ~/projects/eshop6 $ rails generate model User
ine@gii ~/projects/eshop6 $ rails generate model UserSession
```

Editamos pues los ficheros de migración:

Fichero: db/migrate/20170509090901_create_users.rb

```
class CreateUsers < ActiveRecord::Migration
  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
      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
```

Fichero: db/migrate/20170509090942_create_user_sessions.rb

```
class CreateUserSessions < ActiveRecord::Migration
  def change
    create_table :user_sessions do |t|
      t.timestamps null: false
    end
  end
end
```

Para que se creen convenientemente las tablas en la base de datos:

```
ine@gii ~/projects/eshop6 $ rake db:migrate
```

5.8.4. Modelos ORM

Los ficheros correspondiente al modelo quedarían de la siguiente forma:

Fichero: app/models/user.rb

```
class User < ActiveRecord::Base
  acts_as_authentic do |a|
    a.validate_login_field = true
    a.validate_password_field = true
    a.require_password_confirmation = true
    a.logged_in_timeout = 5.minutes # default is 10.minutes
  end

  validates_presence_of :name, :message => "No has introducido tu nombre."
  validates_presence_of :login, :message => "No has introducido el usuario."
  validates_presence_of :email, :message => "No has introducido el e-mail."
  validates_presence_of :password, :message => "No has introducido la contraseña."
  validates_presence_of :password_confirmation, :message => "No has confirmado la contraseña."
  validates_length_of :name, :in => 3..225
  validates_uniqueness_of :name, :login, :email
end
```

Fichero: app/models/user_session.rb

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

5.8.5. Vistas y controladores

Para obtener ficheros de controladores de User, UserSession y admin/Authenticated, ejecutar:

```
ine@gii ~/projects/eshop6 $ rails generate controller user new create edit update show
ine@gii ~/projects/eshop6 $ rails generate controller user_sessions new create destroy
ine@gii ~/projects/eshop6 $ rails generate controller admin/authenticated
```

Editamos ambos controladores convenientemente:

Fichero: app/controllers/user_controller.rb

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

  def new
    @user = User.new
    @page_title = 'Crear nueva cuenta'
    if current_user
      flash[:notice] = 'Sólo una cuenta puede ser creada.'
      redirect_to :controller => 'about', :action => 'index'
    else
      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] = "La cuenta #{@user.name} ha sido creada correctamente. Usuario identificado."
      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] = "La cuenta #{@user.name} ha sido correctamente actualizada."
      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
```

Y también...

Fichero: app/controllers/user_sessions_controller.rb

```
class UserSessionsController < ApplicationController
  before_filter :require_no_user, :only => [:new, :create]

  def new
    @user_session = UserSession.new
    @page_title = 'Inicio de sesión'
  end

  def create
    @user_session = UserSession.new(params[:user_session])
    @user_session.remember_me = false
    @page_title = 'Inicio de sesión'
    if @user_session.save
      flash[:notice] = "Sesión iniciada correctamente."
      redirect_back_or_default :controller => '/admin/supplier', :action => :index
    else
      render :action => :new
    end
  end

  def destroy
    if current_user_session
      current_user_session.destroy
      flash[:notice] = "Sesión cerrada correctamente."
    end
    redirect_to :controller => :catalog, :action => :index
  end

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

Y el último controlador:

Fichero: app/controllers/admin/authenticated_controller.rb

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

Como las entidades administradas por el administrador se deben hacer previa autenticación de este deberemos modificar la primera línea de los controladores de artículo, fabricante, proveedor y pedido:

Fichero: app/controllers/admin/article_controller.rb

```
class Admin::ArticleController < Admin::AuthenticatedController
  ...
  ...
end
```


Fichero: app/controllers/admin/manufacture_controller.rb

```
class Admin::ManufacturerController < Admin::AuthenticatedController
...
end
```

Fichero: app/controllers/admin/supplier_controller.rb

```
class Admin::SupplierController < Admin::AuthenticatedController
...
end
```

Fichero: app/controllers/admin/order_controller.rb

```
class Admin::OrderController < Admin::AuthenticatedController
...
end
```

Seguidamente vamos a mostrar los ficheros correspondientes a las vistas, pero primero debemos modificar application.html.erb

Fichero: app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
<head>
  <title><%= @page_title || 'UCASport' %></title>
  <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track' => true %>
  <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
  <%= csrf_meta_tags %>
</head>
<body>
  <div id="header">
    <h1 id="logo">UCASport</h1>
    <h2 id="slogan">Expertos en zapatillas deportivas</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">
    <ul>
      <li><a href="/admin/supplier">Proveedores</a> &nbsp;&nbsp;&nbsp;</li>
      <li><a href="/admin/manufacture">Fabricantes</a> &nbsp;&nbsp;&nbsp;</li>
      <li><a href="/admin/article">Artículos</a> &nbsp;&nbsp;&nbsp;</li>
    </ul>
  </div>
</body>
</html>
```

```

    <li><a href="/admin/order">Pedidos</a>&nbsp;&nbsp;&nbsp;</li>
    <li><a href="/">Catálogo</a>&nbsp;&nbsp;&nbsp;</li>
    <li><a href="/about">Sobre UCASport</a>&nbsp;&nbsp;&nbsp;</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; 2015-2017 UCASport
</div>
</body>
</html>

```

Y ahora sí, las vistas:

Fichero: app/views/user/_form.html.erb

```

<% if @user.errors.any? %>
  <div id="errorExplanation">
    <h2><%= pluralize(@user.errors.count, "error") %> prohíben crear esta cuenta:</h2>
    <ul>
      <% @user.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
    </ul>
  </div>
<% end %>

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

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

<div class="field">
  <p><label for="user_email">E-mail</label><br/>
  <%= text_field 'user', 'email' %></p>
</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>

```

```
</div>
```

Fichero: app/views/user/new.html.erb

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

Fichero: app/views/user/edit.html.erb

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

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

Fichero: app/views/user/show.html.erb

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

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

  <dt>E-mail</dt>
  <dd><%= @user.email %></dd>

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

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

  <% if @user.last_login_at %>
  <dt>Último inicio de sesión</dt>
  <dd><%= @user.last_login_at %></dd>
  <% end %>

  <dt>Inicio de sesión actual</dt>
  <dd><%= @user.current_login_at %></dd>

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

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

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

Fichero: app/views/user_sessions/_form.html.erb

```
<% if @user_session.errors.any? %>
```

```

<div id="errorExplanation">
<h2><%= pluralize(@user_session.errors.count, "error") %> prohíben que este usuario se identifique:</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>

```

Fichero: app/views/user_sessions/new.html.erb

```

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

```

5.8.6. Test

Se han realizado las siguientes pruebas de integración:

```

ine@gii ~/projects/eshop6 $ rails generate integration_test authentication
ine@gii ~/projects/eshop6 $ rails generate integration_test User

```

Editamos los ficheros:

Fichero: 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_sucesfully("george", "cheetah")
  end

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

```

```

    harry.fails_login("harry", "micky")
  end

  private

  module BrowsingTestDSL
    include ERB::Util
    attr_writer :name

    def tries_to_go_to_admin
      get '/admin/article/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/article/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', :user_session => { :login => login, :password => password }
    end

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

```

Fichero: test/integration/user_test.rb

```

require 'test_helper'

class UserTest < ActionDispatch::IntegrationTest

  def setup
  end

  test "user_account" do
    george = new_session_as(:george)
    user_account = george.create_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 BrowsingTestDSL
  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', parameters
    assert_response :redirect
    follow_redirect!
    assert_response :success
    assert_template 'user/show'

    assert_equal User.last.name, 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], "La cuenta #{user_name} ha sido creada correctamente. Usuario identificado."
    assert_select 'div#notice', "La cuenta #{user_name} ha sido creada correctamente. Usuario identificado."

    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)
    user = User.find_by_id(user_account.id)
    get "/user/edit?id=#{user.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', parameters
  end
end

```

```

assert_response :redirect
follow_redirect!
assert_response :success
assert_template 'user/show'
user_name = parameters[:user][:name]

assert_equal User.last.name, 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], "La cuenta #{user_name} ha sido correctamente actualizada."
assert_select 'div#notice', "La cuenta #{user_name} ha sido correctamente actualizada."
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
end

```

Los ejecutamos:

```

ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/authentication_test.rb
Run options: --seed 23838

# Running:

..

Finished in 2.447998s, 0.8170 runs/s, 0.0000 assertions/s.

2 runs, 0 assertions, 0 failures, 0 errors, 0 skips

ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/user_test.rb
Run options: --seed 58711

# Running:

.

Finished in 1.285825s, 0.7777 runs/s, 0.0000 assertions/s.

1 runs, 0 assertions, 0 failures, 0 errors, 0 skips

```

5.8.7. Dificultades encontradas

Hemos experimentado algunos problemas hasta que nos dimos cuenta que tuvimos que hacer alguna ligera modificación en la primera línea de los controladores que están dentro del directorio admin.

5.8.8. Objetivos alcanzados

Hemos conseguido que ahora solo el usuario administrador pueda administrar los artículos, proveedores y fabricantes, así como los pedidos. Que se cree el usuario admin cuando no existe ninguno y se pida su autenticación en las secciones requeridas.

5.9. Sprint 9 (RSS y AJAX)

Vamos a añadir RSS (Really Simple Syndication) y AJAX (Asynchronous JavaScript And XML) a nuestra tienda. RSS es un sistema para obtener información de actualizaciones sobre una web, y AJAX es una técnica de desarrollo web para crear aplicaciones interactivas, queremos que se hagan las operaciones del carrito de la compra de forma asíncrona sin necesidad de actualizar la web, solo la información plasmada en ella.

5.9.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Vista y Controlador	0	50	60	50	0	0	0
Integración RSS	0	50	30	30	0	0	0
Integración AJAX	0	50	30	30	0	0	0
Tests	0	20	20	0	0	0	0
Total	0	170	140	110	0	0	0

Tabla 5.9: Sprint 9 backlog (minutos)

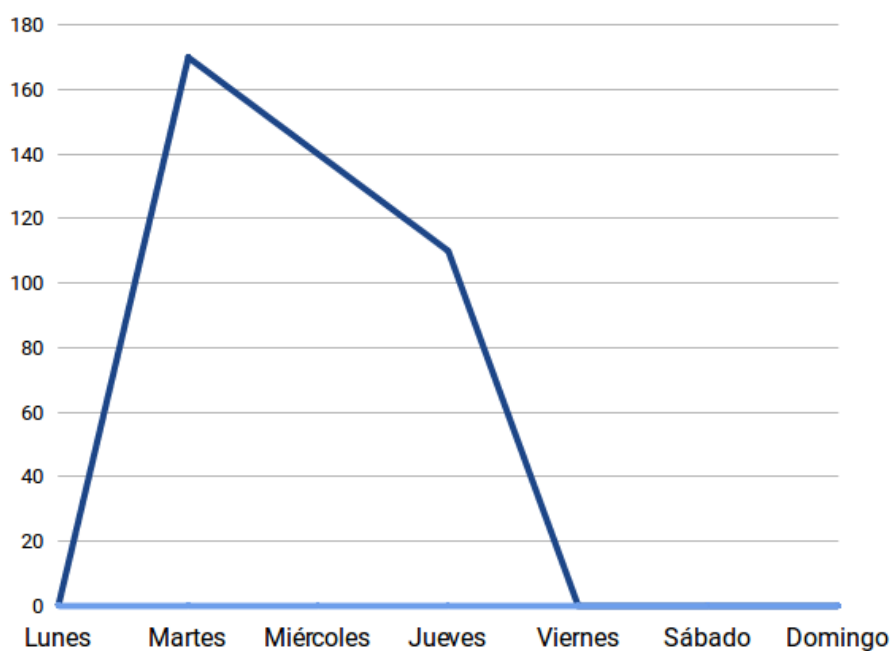


Figura 5.9: Diagrama burndown

5.9.2. Gemas utilizadas

En este sprint hemos necesitado añadir las gemas:

- 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 es necesario añadir código. Revisar que las operaciones en `cart.rb` están correctamente diseñadas.

5.9.5. Vistas y controladores

Modificamos los siguientes archivos para añadir RSS al Catálogo:

Fichero: `app/controllers/catalog_controller.rb`

```
class CatalogController < ApplicationController
  before_filter :initialize_cart, :except => :show
  before_filter :require_no_user

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

  def index
    @articles = Article.order("articles.id desc").includes(:suppliers, :manufacturer).paginate(:page => params[:page],
    :per_page => 5)
    @page_title = 'Catálogo'
  end

  def latest
    @articles = Article.latest 5
    @page_title = 'Últimos artículos'
  end

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

Fichero: 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 "Eshop6: Zapatillas deportivas"
    xml.link url_for :action => 'index', :only_path => false
    xml.language "es-es"
    xml.ttl "60"

    for article in @articles do
      xml.item do
        xml.title article.title
        xml.description "#{article.title} by #{article.manufacturer.name}"
        xml.link url_for :action => "show", :id => article, :only_path => false
        xml.guid url_for :action => "show", :id => article, :only_path => false
        xml.pubDate article.created_at.to_s :long
        xml.manufacturer article.manufacturer.name
      end
    end
  end
end
```

Modificamos los siguientes archivos para añadir AJAX al Carrito de la compra:

Fichero: app/controllers/cart_controller.rb

```
class CartController < ApplicationController
  before_filter :initialize_cart

  def add
    @article = Article.find params[:id]
    @page_title = 'Añadir artículo'
    if request.xhr?
      @item = @cart.add params[:id]
      flash.now[:cart_notice] = "Artículo <em>#{@item.article.title}</em> añadido satisfactoriamente al carrito."
      render :controller => 'cart', :action => 'add_with_ajax'
    elsif request.post?
      @item = @cart.add params[:id]
      flash[:cart_notice] = "Artículo <em>#{@item.article.title}</em> añadido satisfactoriamente al carrito."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'add', :template => 'cart/add'
    end
  end

  def remove
    @article = Article.find params[:id]
    @page_title = 'Eliminar artículo'
    if request.xhr?
      @item = @cart.remove params[:id]
      flash.now[:cart_notice] = "Artículo <em>#{@item.article.title}</em> eliminado correctamente del carrito."
      render :controller => 'cart', :action => 'remove_with_ajax'
    elsif request.post?
      @item = @cart.remove params[:id]
      flash[:cart_notice] = "Artículo <em>#{@item.article.title}</em> eliminado correctamente del carrito."
      redirect_to :controller => 'catalog'
    else
      render :controller => 'cart', :action => 'remove', :template => 'cart/remove'
    end
  end
end
```

```

def clear
  @page_title = 'Vaciar el carrito'
  if request.xhr?
    @cart.cart_items.destroy_all
    flash.now[:cart_notice] = "Carrito vaciado"
    render :controller => 'cart', :action => 'clear_with_ajax'
  elsif request.post?
    @cart.cart_items.destroy_all
    flash[:cart_notice] = "Carrito vaciado"
    redirect_to :controller => 'catalog'
  else
    render :controller => 'cart', :action => 'clear'
  end
end
end

```

Fichero: app/views/catalog/_articles.html.erb

```

<dl id = 'articles'>
  <% for article in @articles %>
    <dt>
      <%= link_to article.title, :action => 'show', :id => article %>
      <%= link_to '+', :controller => 'cart', :action => 'add', :id => article, :remote => true %>
    </dt>
    <% for supplier in article.suppliers %>
      <dd><%= supplier.last_name %>, <%= supplier.first_name %></dd>
    <% end %>
    <dd><%= sprintf("Precio: %.2f €, Talla: %d", article.price, article.size) %> </dd>
    <dd><small>Fabricante: <%= article.manufacturer.name %></small></dd>
  <% end %>
</dl>

```

Fichero: app/views/cart/add_with_ajax.js.erb

```

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

```

Fichero: 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);

```

Fichero: app/views/cart/remove_with_ajax.js.erb

```

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

```

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

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

<h3>Tu carrito de la Compra</h3>
<p>
  <strong>
    <% unless controller.controller_name == 'checkout' %>
      <%= link_to 'Proceder al pago', :controller => 'checkout' %>
    <% end %>
  </strong>
</p>
<ul>
  <% for item in @cart.cart_items %>
    <li id="cart_item_<%= item.article.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', :remote => true %></b>
  </p>
<% end %>
```

Fichero: app/views/cart/_item.html.erb

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

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

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

Fichero: app/assets/javascripts/application.js

```
// This is a manifest file that'll be compiled into application.js, which will include all the files
// listed below.
//
// Any JavaScript/Coffee file within this directory, lib/assets/javascripts, vendor/assets/javascripts,
// or any plugin's vendor/assets/javascripts directory can be referenced here using a relative path.
//
// It's not advisable to add code directly here, but if you do, it'll appear at the bottom of the
// compiled file.
//
// Read Sprockets README (https://github.com/rails/sprockets#sprockets-directives) for details
// about supported directives.
//
//= require jquery
//= require jquery_ujs
//= require jquery.ui.effect-fade
//= require jquery.ui.effect-highlight
//= require turbolinks
//= require_tree .
```

5.9.6. Tests

Tras las pertinentes modificaciones los tests quedarán de la siguiente forma:

Fichero: test/integration/browsing_and_searching_test.rb

```
require 'test_helper'

class BrowsingAndSearchingTest < ActionDispatch::IntegrationTest
  fixtures :manufacturers, :suppliers, :articles, :articles_suppliers

  test "browse" do
    jill = new_session_as :jill
    jill.index
    jill.second_page
    jill.article_details 'Articulo_9'
    jill.latest_articles
    jill.reads_rss
  end

  module BrowsingTestDSL
    include ERB::Util
    attr_writer :name

    def index
      get '/catalog/index'
      assert_response :success
      assert_select 'dl#articles' do
        assert_select 'dt', :count => 5
      end
      assert_select 'dt' do
        assert_select 'a', 'Articulo_10'
      end
      check_article_links
    end

    def second_page
      get '/catalog/index?page=2'
      assert_response :success
      assert_template 'catalog/index'
      assert_equal Article.find_by_title('Articulo_1'),
        assigns(:articles).last
      check_article_links
    end

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

    def latest_articles
      get '/catalog/latest'
      assert_response :success
      assert_template 'catalog/latest'
      assert_select 'dl#articles' do
        assert_select 'dt', :count => 5
      end
    end
  end
end
```

```

end
@articles = Article.latest(5)
@articles.each do |a|
  assert_select 'dt' do
    assert_select 'a', a.title
  end
end
end

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

  @articles = Article.latest(5)
  @articles.each do |article|
    assert_select 'item' do
      assert_select 'title', article.title
    end
  end
end
end

```

...y también:

Fichero: test/controllers/cart_controller_test.rb

```

require 'test_helper'

class CartControllerTest < ActionController::TestCase
  fixtures :suppliers, :manufacturers, :articles

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

```

```

test "remove" do
  post :add, :id => 4
  assert_equal [Article.find(4)], Cart.find(@request.session[:cart_id]).articles

  post :remove, :id => 4
  assert_equal [], Cart.find(@request.session[:cart_id]).articles
end

test "clear" do
  post :add, :id => 4
  assert_equal [Article.find(4)], Cart.find(@request.session[:cart_id]).articles

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

test "add_xhr" do
  assert_difference(CartItem, :count) do
    xhr :post, :add, :id => 5
  end

  assert_response :success
  assert_equal 1, Cart.find(@request.session[:cart_id]).cart_items.size
end
end

```

Pasamos a ejecutar ambos:

```

ine@gii ~/projects/eshop6 $ rake test TEST=test/integration/browsing_and_searching_test.rb
Run options: --seed 3707

# Running:

.

Finished in 1.364453s, 0.7329 runs/s, 0.0000 assertions/s.

1 runs, 0 assertions, 0 failures, 0 errors, 0 skips

ine@gii ~/projects/eshop6 $ rake test TEST=test/controllers/cart_controller_test.rb
Run options: --seed 5929

# Running:

....

Finished in 1.071695s, 3.7324 runs/s, 13.9965 assertions/s.

4 runs, 15 assertions, 0 failures, 0 errors, 0 skips

```

5.9.7. Dificultades encontradas

Tuvimos dificultades con los métodos *remove* y *clear* del carrito, pero fue simplemente que se nos olvidó escribir el correspondiente `:remote => true`.

5.9.8. Objetivos alcanzados

Podemos consultar el RSS de los últimos artículos (hemos utilizado [Liferea](#)) y funciona perfectamente la implementación en AJAX del Carrito de la Compra junto a sus animaciones, esto hace que nuestra tienda sea más atractiva a la vista del usuario.

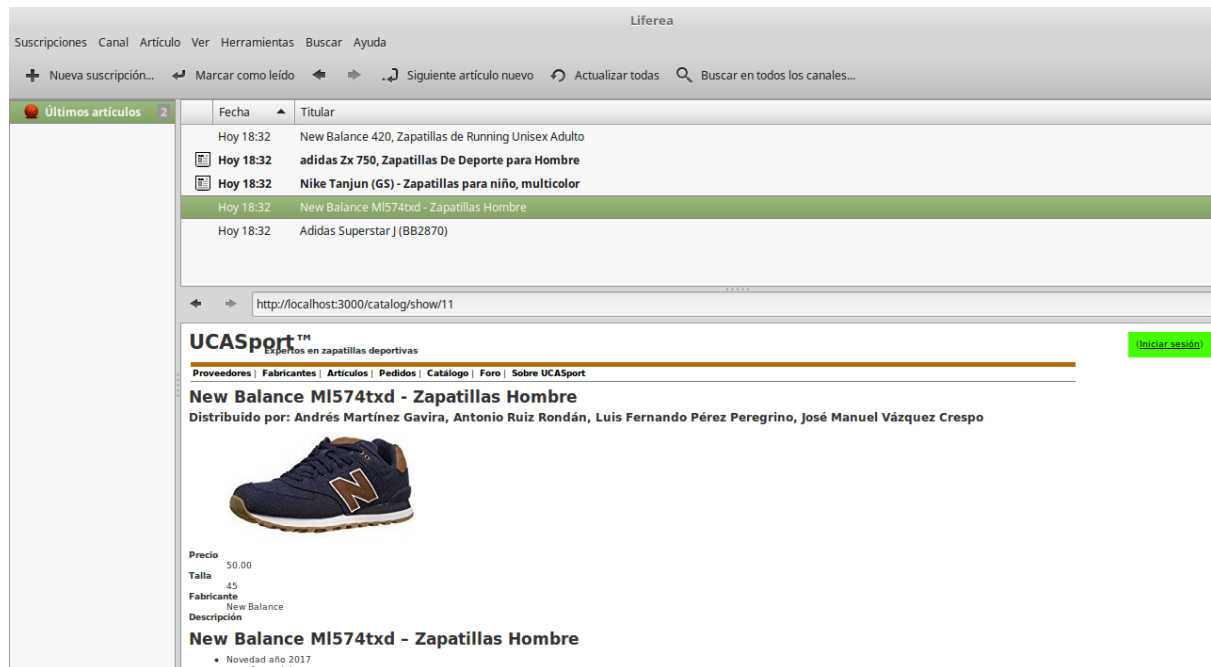


Figura 5.9. Consulta de RSS utilizando Liferea

5.10. Sprint 10 (Foro de discusión y motor de búsqueda)

Vamos a crear el foro. También añadiremos una búsqueda de artículos en el catálogo de nuestra web.

5.10.1. Sprint backlog y diagrama burndown

Tareas	L	M	X	J	V	S	D
Ficheros de migración	0	80	0	0	0	0	0
Vistas y controladores	0	100	0	0	90	0	50
Modelos	0	40	0	0	50	0	0
Tests	0	0	0	0	0	0	40
Total	0	220	0	0	140	0	90

Tabla 5.10: Sprint 10 backlog (minutos)

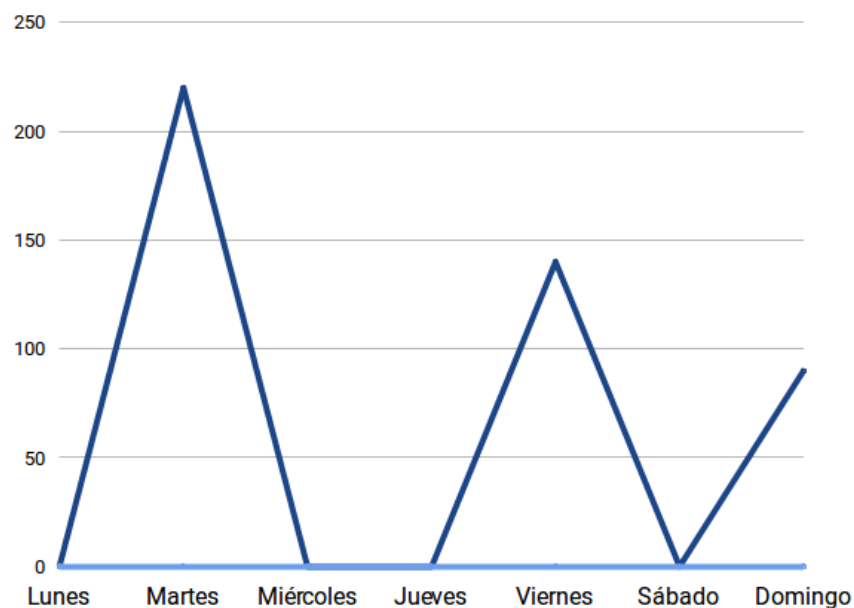


Figura 5.10: Diagrama burndown

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 fichero de migración la estructura del foro y sus mensajes:

Fichero: db/migrate/20170523090915_create_forum_posts.rb

```
class CreateForumPosts < ActiveRecord::Migration
  def change
    create_table :forum_posts do |t|
      t.string :name, :limit => 50, :null => false
      t.string :subject, :limit => 255, :null => false
      t.text :body
      t.integer :root_id, :null => false, :default => 0
      t.integer :parent_id, :null => false, :default => 0
      t.integer :depth, :null => false, :default => 0
      t.timestamps
    end
  end
end
```

5.10.4. Modelos ORM

Luego hemos generado el modelo ForumPost:

Fichero: app/models/forum_post.rb

```
class ForumPost < ActiveRecord::Base
  validates_length_of :name, :within => 2..50, :message => "El nombre debe tener una longitud entre 2 y 50 caracteres"

  validates_length_of :subject, :within => 5..255, :message => "El tema debe tener una longitud entre 5 y 255 caracteres"

  validates_length_of :body, :within => 5..5000, :message => "La descripción debe tener una longitud entre 5 y 5000 caracteres"
end
```

5.10.5. Vistas y Controladores

A continuación Ficheros del controlador y las vistas del foro:

Fichero: app/controllers/forum_controller.rb

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

  def post
    @post = ForumPost.new(:parent_id => 0, :root_id => 0, :depth => 0)
    @page_title = "Postear en el foro."
  end

  def create
    @post = ForumPost.new(post_params)
  end
end
```

```

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 = "Postear en el foro."
  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.name}."
  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} ha sido eliminado correctamente."
  redirect_to :action => 'index'
end

def show
  @post = ForumPost.find(params[:id])
  @posts = ForumPost.where(root_id: @post.root_id)
  @page_title = "#{@post.name}"
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, :created_at, :depth)
end
end

```

Fichero: app/views/forum/_form.html.erb

```

<% if @post.errors.any? %>
  <div id="errorExplanation">
    <h2><%= pluralize(@post.errors.count, "error") %> prohibió este post de ser creado.</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' %>
</div>

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

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

```

Fichero: 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 aún. <%= link_to 'Sé el primero en crear un post', :action => 'post' %></p>
<% end %>

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

```

Fichero: app/views/forum/post.html.erb

```

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

<%= link_to 'Volver', :action => 'index' %>

```

Fichero: app/views/forum/show.html.erb

```

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

```

```

<dd><%= h @post.body %></dd>
</dl>

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

```

...y creamos el siguiente helper:

Fichero: app/helpers/forum_helper.rb

```

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: "¿Seguro que quieres borrar el post: \"#{post.subject}\"?" }
      margin_left = post.depth * 20
      content << %(
<tr><td><div style="margin-left:#{margin_left}px">
  #{url} escrito por #{h post.name} &middot; #{post.created_at.strftime("%H:%M:%S - %d/%M/%Y")}
</div></td>
content << %(<td><div>#{button}</div></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 ficheros que tenemos que modificar para tener un motor de búsqueda en el catálogo de nuestra web:

Añadimos al controlador de catálogo:

Fichero: app/controllers/catalog_controller.rb

```

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

```

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

Fichero: `app/views/catalog/index.html.erb`

```
...
...
<p><%= link_to 'Últimos artículos', :action => 'latest' %></p>
<p><%= link_to 'RSS', :action => 'rss' %></p>
<p><%= link_to 'Buscar', :action => 'search' %></p>
```

Por último creamos las siguientes vistas:

Fichero: `app/views/catalog/_search_box.html.erb`

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

Fichero: `app/views/catalog/search.html.erb`

```
<p>
  Introduzca el nombre (o parte del nombre) del artículo que quiere buscar.
</p>

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

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

Fichero: `test/integration/forum_test.rb`

```
require '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 => 'Eshop11 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
  include ERB::Util
  attr_writer :name

  def post_to_forum(parameters)
    get '/forum/post'
    assert_response :success
    assert_template 'forum/post'
    post '/forum/create', 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 'h1', 'Foro'
    assert_select 'a', 'Nuevo post'
  end

  def view_post(post)
    get "/forum/show/?id=#{post.id}"
    assert_response :success
    assert_template 'forum/show'
    assert_select 'h1', :content => "&#39;#{post.subject}&#39;"
  end

  def reply_to_post(post, parameters)
    get "/forum/reply/?id=#{post.id}"
    assert_response :success
    assert_select 'h1', :content => "Responder a &#39;#{post.subject}&#39;"
    post '/forum/create', :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 'a', :content => parameters[:post][:subject]
  end
end

```



```

def delete_post(post)
  post "/forum/destroy/?id=#{post.id}"
  assert_response :redirect
  follow_redirect!
  assert_response :success
  assert_template 'forum/index'
  assert_select 'div', { :id => "notice" },
    :content => "El post #{post.subject} ha sido eliminado correctamente."
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

En esta iteración hemos encontrado dificultades para que funcionaran algunos métodos y también tuvimos alguna dificultad para definir el método en el helper del foro (`display_like_a_tree`). Tuvimos algún problema con el test de integración que pudimos solventar.

5.10.8. Objetivos alcanzados

Se han conseguido completar todas las funciones del foro, así como un motor de búsqueda en el catálogo de artículos.

5.11. Sprint 11 (Borrado de órdenes)

Vamos a dar la posibilidad al administrador del sitio de poder borrar pedidos.

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
Vistas y controladores	0	30	0	0	0	0	0
Modelos	0	0	0	0	0	0	0
Tests	0	0	0	0	0	0	0
Total	0	30	0	0	0	0	0

Tabla 5.11: Sprint 11 backlog (minutos)

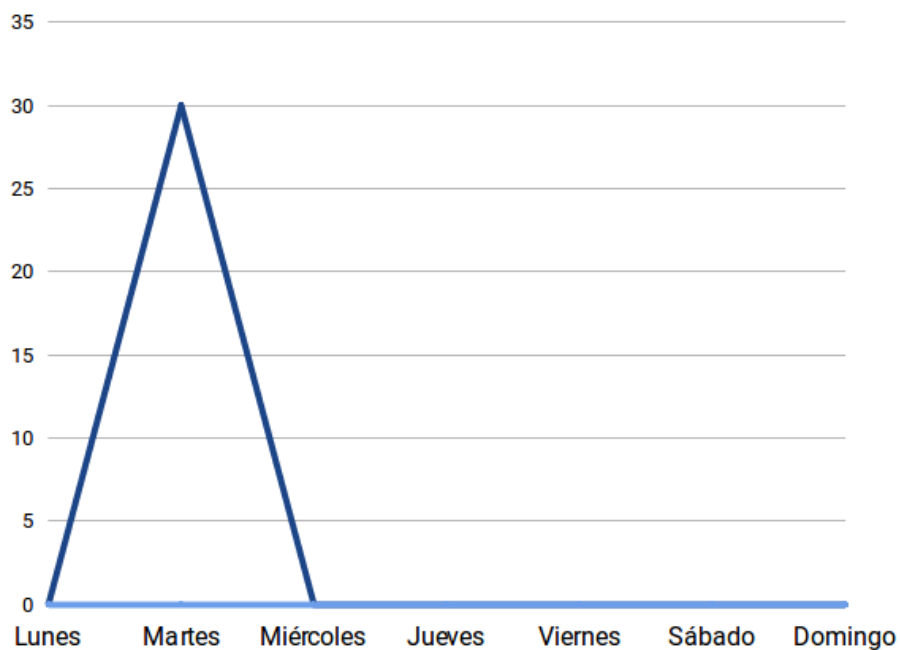


Figura 5.11: Diagrama burndown

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`

Fichero: `app/controllers/admin/order_controller.rb`

```
class Admin::OrderController < Admin::AuthenticatedController
  def close
    order = Order.find(params[:id])
    order.close
    flash[:notice] = "La orden ##{order.id} ha sido cerrada."
    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?
      @status = ''
      conditions = nil
    else
      conditions = "status = '#{@status}'"
    end
    @orders = Order.where(conditions).paginate(:page => params[:page], :per_page => 10)

    if @status == 'open'
      @status = 'abiertos'
    elsif @status == 'processed'
      @status = 'procesados'
    elsif @status == 'closed'
      @status = 'cerrados'
    elsif @status == 'failed'
      @status = 'fallidos'
    end

    @page_title = "Mostrando pedidos #{@status}"
  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:

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

```
<% if @orders == [] %>
  <h2><%= "No hay pedidos con estado: '#{@status}'." %></h2>
<% else %>
  <table>
    <tr>
      <th>ID</th>
      <th>Estado</th>
      <th>Precio total</th>
      <th>Cantidad</th> <!-- amount -->
      <th>Creado el</th>
      <th>Actualizado el</th>
    </tr>
    <% for order in @orders %>
      <tr>
        <td align="center"><%= order.id %></td>
        <td align="center"><%= order.status[0].capitalize + order.status[1..order.status.length-1] %></td>
        <td align="center"><%= order.total %></td>
        <td align="center"><%= order.amount %></td> <!-- amount -->
        <td align="center"><%= order.created_at.strftime("%d-%m-%Y %I:%M") %></td>
        <td align="center"><%= order.updated_at.strftime("%d-%m-%Y %I:%M") %></td>
        <td><%= link_to 'Mostrar', :action => 'show', :id => order %></td>
        <!-- AÑADIR PARA EL BORRADO DE ÓRDENES -->
        <td>
          <%= button_to 'Borrar', { :action => 'destroy', :id => order },
            data: { confirm: "¿Realmente quiere borrar la orden ##{order.id}?" } %>
        </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' %>
```

5.11.6. Test

No se han desarrollado tests en esta iteración.

5.11.7. Dificultades encontradas

No se han encontrado dificultades reseñables en esta iteración.

5.11.8. Objetivos alcanzados

Hemos conseguido que el administrador del sitio pueda borrar las órdenes que él desee.

6. Equipo

A continuación detallamos la función asumida por cada miembro del equipo de trabajo, junto con una descripción de su contribución al proyecto de desarrollo. También especificaremos el entorno tecnológico utilizado en este proyecto.

6.1. Componentes

- **Martínez Gavira, Andrés**

- Coordinador
- Analista-diseñador
- Contribución al proyecto:
 - Colaboración en desarrollo de la Base de Datos (BD) y las migraciones.
 - Colaboración en desarrollo de los modelos, controladores y vistas.
 - Planificación del trabajo en grupo.
 - División de las tareas de cada miembro del grupo.

- **Ruiz Rondán, Antonio**

- Administrador de base de datos
- Desarrollador
- Contribución al proyecto:
 - Elaboración del informe de desarrollo.
 - Diseño de la estructura de la BD.
 - Desarrollo de las migraciones.

- **Perez Peregrino, Luis Fernando**

- Ingeniero de pruebas
- Analista-diseñador
- Contribución al proyecto:
 - Diseño y desarrollo de pruebas.
 - Colaboración en el diseño de la BD y las migraciones.
 - Colaboración en desarrollo de los modelos, controladores y vistas.

- **Vázquez Crespo, José Manuel**

- Desarrollador
- Analista-diseñador
- Contribución al proyecto:
 - Diseño y desarrollo de los modelos, controladores y vistas.
 - Colaboración en el diseño de la BD.

- **Rodríguez Cárdenas, Antonio José**
 - Desarrollador
 - Analista-diseñador
 - Contribución al proyecto:
 - Diseño y desarrollo de los modelos, controladores y vistas.
 - Colaboración en el diseño de la BD.

6.2. Entorno tecnológico

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

- Portátil Toshiba Satellite con las siguientes especificaciones técnicas:
 - Procesador: Intel Core i7
 - Memoria: 6 GB RAM
 - SO: Linux Mint 18
- Portátil HP con las siguientes especificaciones técnicas:
 - Procesador: Intel Core i5
 - Memoria: 8 GB RAM
 - SO: Ubuntu 16
- Portátil Acer con las siguientes especificaciones técnicas:
 - Procesador: Intel Core i3
 - Memoria: 4 GB RAM
 - SO: Ubuntu 14.04 LTS
- Portátil Asus con las siguientes especificaciones técnicas:
 - Procesador: Intel Core i5
 - Memoria: 6 GB RAM
 - SO: Ubuntu 16

El software que se ha utilizado para el desarrollo ha sido:

- Geany como editor de texto/entornos de desarrollo.
- Ruby on Rails como framework.
- Google Documents como sistema de composición de textos para la creación de este documento.

7. Conclusiones

En este proyecto, además de familiarizarnos con Ruby y Ruby on Rails, hemos aprendido a desarrollar de forma rápida y efectiva un sistema de comercio electrónico. Ésto nos parece muy útil e interesante de cara a posibles futuros proyectos. Por este motivo, pensamos ampliar nuestros conocimientos en los frameworks usados en el proyecto.

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.