Plan de Pruebas

Proyecto: Desarrollo de Test QA

Versión: 1.0

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Guillermo Echagüe | Creación del documento | Mar 2021 |
|  |  |  |  |

Contenido

[1 Introducción 1](#_Toc66307987)

[1.1 Alcance 1](#_Toc66307988)

[2 Contexto de las Pruebas 1](#_Toc66307989)

[2.1 Proyecto / Subprocesos de Prueba 1](#_Toc66307990)

[2.2 Elementos de Prueba 1](#_Toc66307991)

[2.3 Alcance de la Prueba 2](#_Toc66307992)

[2.4 Suposiciones y Restricciones 2](#_Toc66307993)

[3 Estrategia de Prueba 3](#_Toc66307994)

[3.1 Subprocesos de prueba 3](#_Toc66307995)

[3.2 Entregables de Prueba 3](#_Toc66307996)

[3.3 Técnicas de diseño de Prueba 3](#_Toc66307997)

[3.4 Criterio de Finalización y Prueba 3](#_Toc66307998)

[3.5 Requisitos del entorno de Pruebas 4](#_Toc66307999)

[3.5.1 Ambiente de pruebas 4](#_Toc66308000)

[3.5.2 Herramientas de Pruebas 4](#_Toc66308001)

[4 Actividades y Estimados de Prueba 4](#_Toc66308002)

[5 Desarrollo de Pruebas 5](#_Toc66308003)

[5.1 Desarrollo de Pruebas Unitarias y de Integración 5](#_Toc66308004)

[5.2 Desarrollo de Pruebas End to End 6](#_Toc66308005)

[6 Conclusiones 16](#_Toc66308006)

[7 Repositorio de Pruebas 16](#_Toc66308007)

[8 Anexos 16](#_Toc66308008)

[8.1 RSpec 16](#_Toc66308009)

# 1 Introducción

## 1.1 Alcance

El propósito de este documento es proporcionar la información y el marco requerido para planificar y desarrollar las actividades del proceso de pruebas para el cargo de Testing QA.

# 2 Contexto de las Pruebas

## 2.1 Proyecto / Subprocesos de Prueba

El desarrollo de esta actividad tiene como intensión lograr realizar las Requisitos principales he Historias de usuario descritas en las solicitaciones.

A continuación se muestran los módulos que se relacionan con las funcionalidades:

## 2.2 Elementos de Prueba

**Requisitos principales**

1. El usuario debe tener un usuario que será su email con formato válido de email
2. El usuario debe tener una contraseña para validar su ingreso
3. Los artículos deben tener un título, cuerpo, estado (público, privado y archivado) y quedar asociados al usuario que los escriba.
4. Los comentarios deben tener un cuerpo, estado (público, privado y archivado) y quedar asociados al usuario que los escriba.
5. Los usuarios pueden ver todos los artículos creados por su usuario.
6. Los usuarios solo pueden ver artículos públicos de otros usuarios.
7. Los usuarios pueden ver todos los comentarios asociados a un artículo creados por su usuario.
8. Los usuarios solo pueden ver los comentarios públicos asociados a un artículo creados por otro usuario.
9. Los usuarios pueden eliminar solo sus artículos.
10. Los usuarios pueden eliminar solo sus comentarios.
11. Los usuarios solo pueden crear o eliminar *Api Keys.* Al crearlas, éstas quedarán asociadas a su usuario.
12. Los usuarios pueden acceder a la API utilizando una Api Key válida, en caso de no tenerla recibirá un error de autenticación.
13. Los recursos disponibles en la API solo permiten extraer información (GET) y no agregar información al sistema.

**Historias de usuario**

1. Como usuario no registrado, quiero poder registrarme con mi email válido y contraseña.
2. Como usuario registrado, quiero poder acceder a la aplicación con mi email válido y contraseña.
3. Como usuario registrado, quiero poder cerrar sesión.
4. Como usuario quiero ver el listado de artículos disponibles.
5. Como usuario quiero crear un artículo.
6. Como usuario quiero eliminar un artículo.
7. Como usuario quiero editar un artículo.
8. Como usuario quiero ver el detalle de un artículo.
9. Como usuario quiero ver los comentarios asociados al mostrar un artículo.
10. Como usuario quiero eliminar mis comentarios.
11. Como usuario quiero ver el listado de Api Keys.
12. Como usuario quiero crear Api Keys.
13. Como usuario quiero eliminar Api Keys.
14. Como usuario quiero acceder a la API del sistema usando una Api Key y ver el listado de usuarios.
15. Como usuario quiero acceder a la API del sistema usando una Api Key y ver el listado de artículos.
16. Como usuario quiero acceder a la API del sistema usando una Api Key y ver el listado de artículos y sus comentarios.

## 2.3 Alcance de la Prueba

El sistema compuesto por los módulos mencionados en el acápite 2.2

Los factores de la calidad no funcionales como el rendimiento, la seguridad informática y la usabilidad no se probarán en este proyecto de prueba, estas pruebas no están contempladas en esta oportunidad.

## 2.4 Suposiciones y Restricciones

Suposiciones:

* El ambiente de pruebas será un ambiente de desarrollo y producción de un servicio web de forma que puedan cumplir los requisitos principales y las historias de usuario.

Restricciones:

* Para el desarrollo de esta actividad se tiene un tiempo desde la entrega del proyecto el día Martes 09-03-2021 hasta el día Miércoles 10-03-2021.

# 3 Estrategia de Prueba

El enfoque del desarrollo de las pruebas es poder lograr la verificación de los requisitos principales e historias de usuario, detectando nuevas historias recomendadas para obtener una mejor experiencia de usuario.

## 3.1 Subprocesos de prueba

Las pruebas para el desafío propuesto incluirán los siguientes subprocesos de prueba:

* Pruebas unitarias.
* Pruebas de integración.
* Pruebas End-to-End.

## 3.2 Entregables de Prueba

Para cada subproceso de pruebas se debe generar la siguiente documentación:

* Plan de pruebas del subproceso de prueba;
* Especificación de Casos de Pruebas
* Informes de Estado de las Pruebas.
* Informe de Finalización del subproceso de prueba.

## 3.3 Técnicas de diseño de Prueba

En este acápite se identifican las técnicas que se utilizarán para el diseño de las pruebas.

* Pruebas de Casos de Uso
* Particiones de Equivalencia y Análisis de Valores Límites.
* Pruebas Exploratorias

## 3.4 Criterio de Finalización y Prueba

Las pruebas deben alcanzar una cobertura de requisitos del 80% y todos los procedimientos de pruebas deben ejecutarse sin fallas de gravedad 1 (alta).

## 3.5 Requisitos del entorno de Pruebas

### 3.5.1 Ambiente de pruebas

|  |  |
| --- | --- |
| Navegadores | Chrome Versión 89.0.4389.82 (Build oficial) (64 bits) |
| Sistemas Operativos | Windows 10 (64 bits) |

### 3.5.2 Herramientas de Pruebas

|  |  |
| --- | --- |
| **Herramienta** | **Función** |
| rspec | herramienta de prueba de lenguaje específica del dominio de la computadora escrita en el lenguaje de programación Ruby para probar código Ruby |
| Selenium Web Driver | API para automatizar sistemas Web |
| Ruby on Rails | Framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby |
| Python | Python es un lenguaje de programación interpretado |

# 4 Actividades y Estimados de Prueba

Las pruebas se dividirán en las siguientes actividades principales:

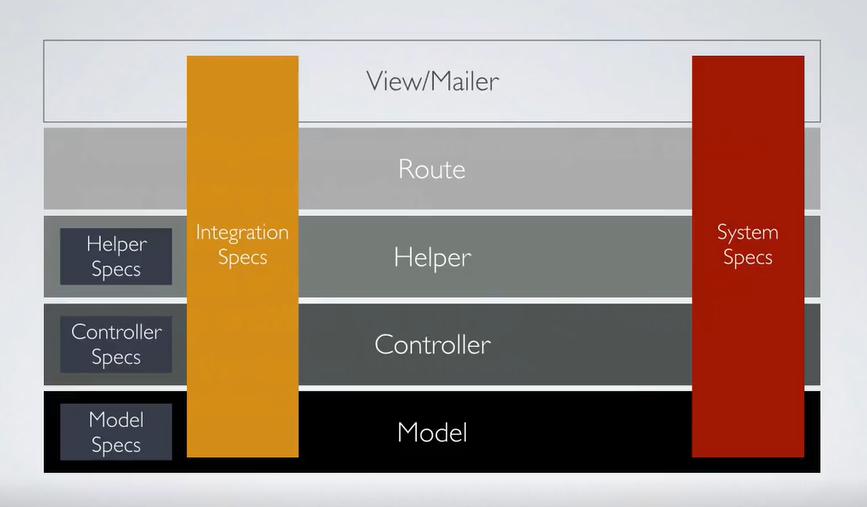
1. Definición de una estructura general de las pruebas por conjuntos de casos de uso a probar.
2. Especificación detallada de los casos de pruebas.
3. Establecimiento del entorno de pruebas.
4. Primer ciclo de ejecución de las pruebas.
5. Informe de finalización de las pruebas.

# 5 Desarrollo de Pruebas

## 5.1 Desarrollo de Pruebas Unitarias y de Integración

Las pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software funcionan juntos correctamente probándolos en grupo.

La siguiente imagen, muestra la estructura de las pruebas unitarias y de integración de un proyecto desarrollado en Ruby on Rails.



Dentro del escenario de plan de pruebas se encuentra:

Planes en la Base de Datos

Article:

* validate presence of required fields
* validate relations
* when user is not present
* when resource is found
* when resource is not found
* when resource is not owned

Comment:

* when comment is not present
* when user is not present

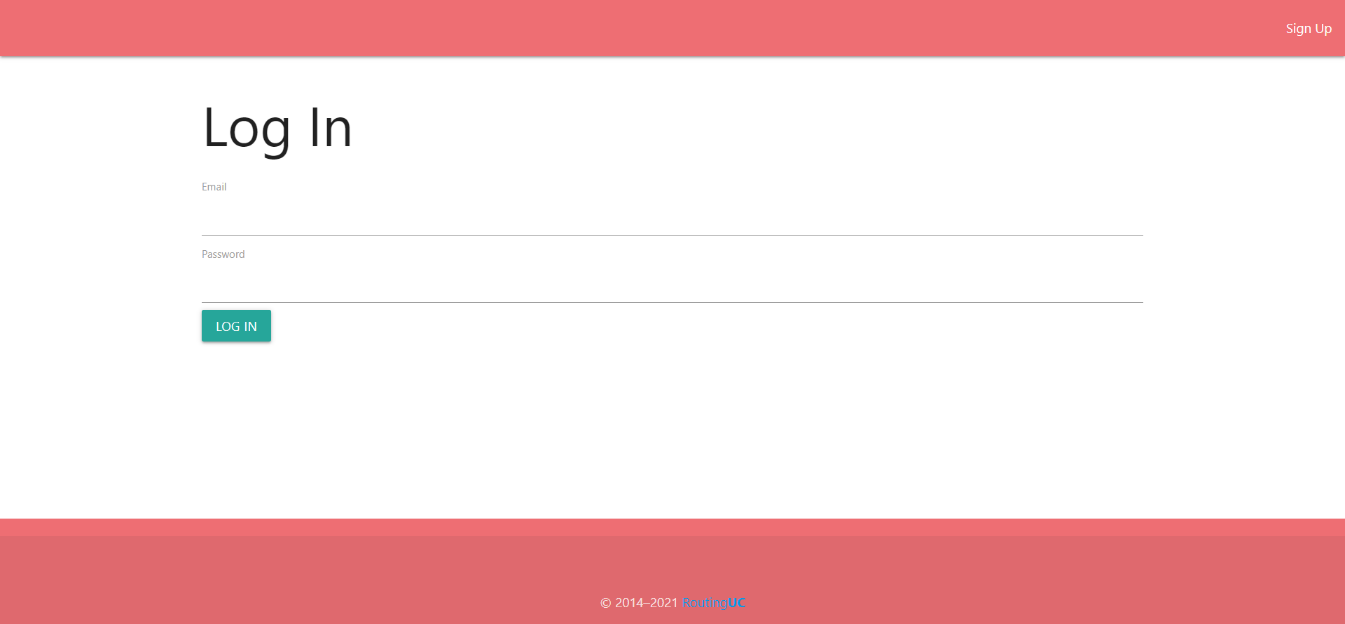
User:

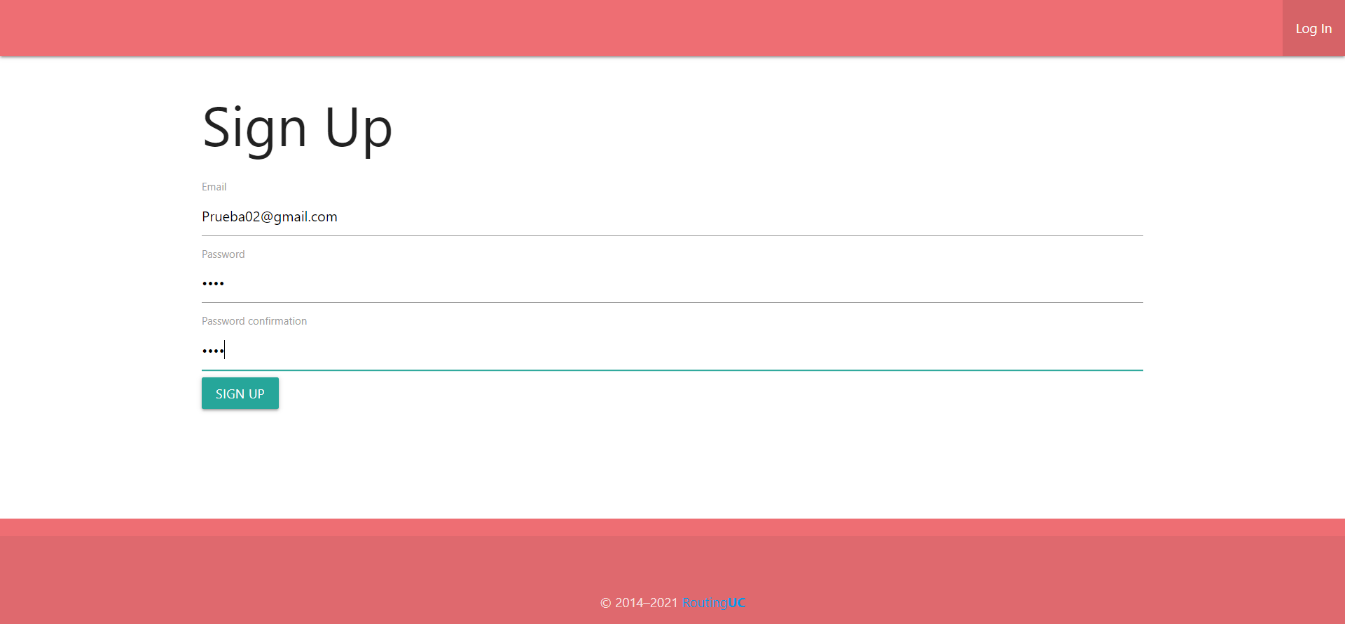
* when email is not present

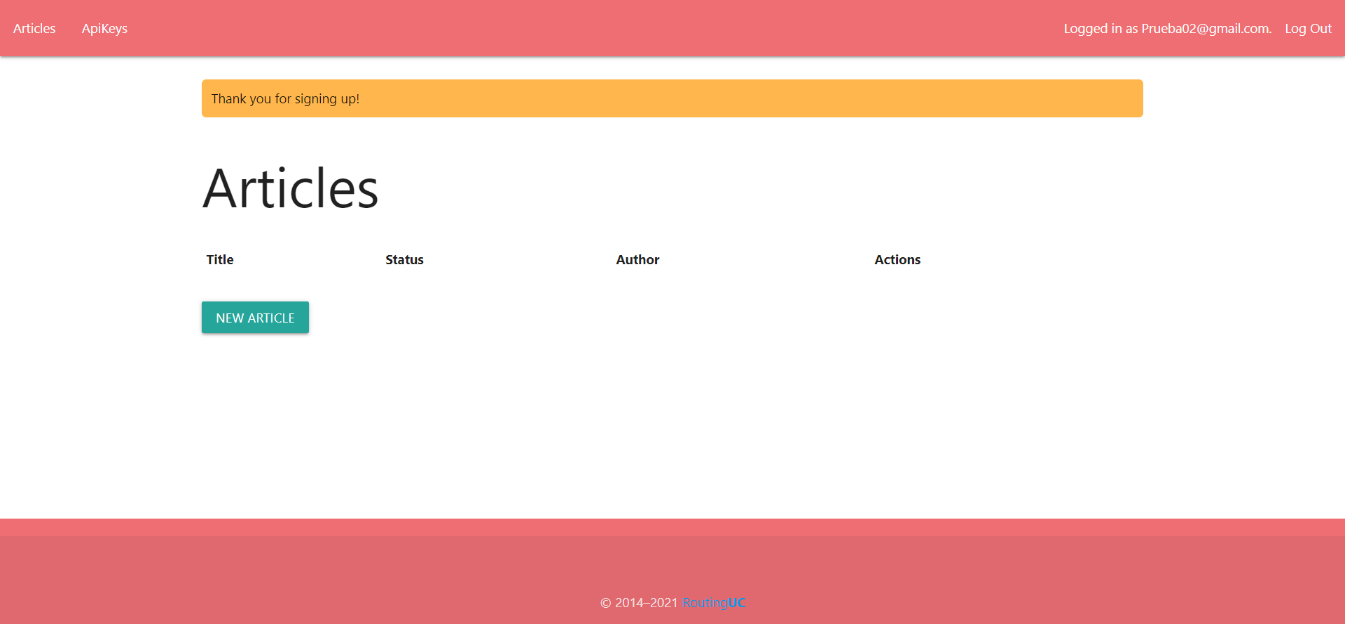
## 5.2 Desarrollo de Pruebas End to End

Las pruebas End to End (E2E) permiten definir los casos de prueba en lenguaje casi “natural” y cercano al usuario. Permite disponer de un set de pruebas incremental de manera ágil y robusta, junto con integrarse en un ciclo de integración continua para implementar pruebas automáticas. Como se menciona en el punto 3.5 se utiliza Selenium implementado en Python para el desarrollo de esta sección.

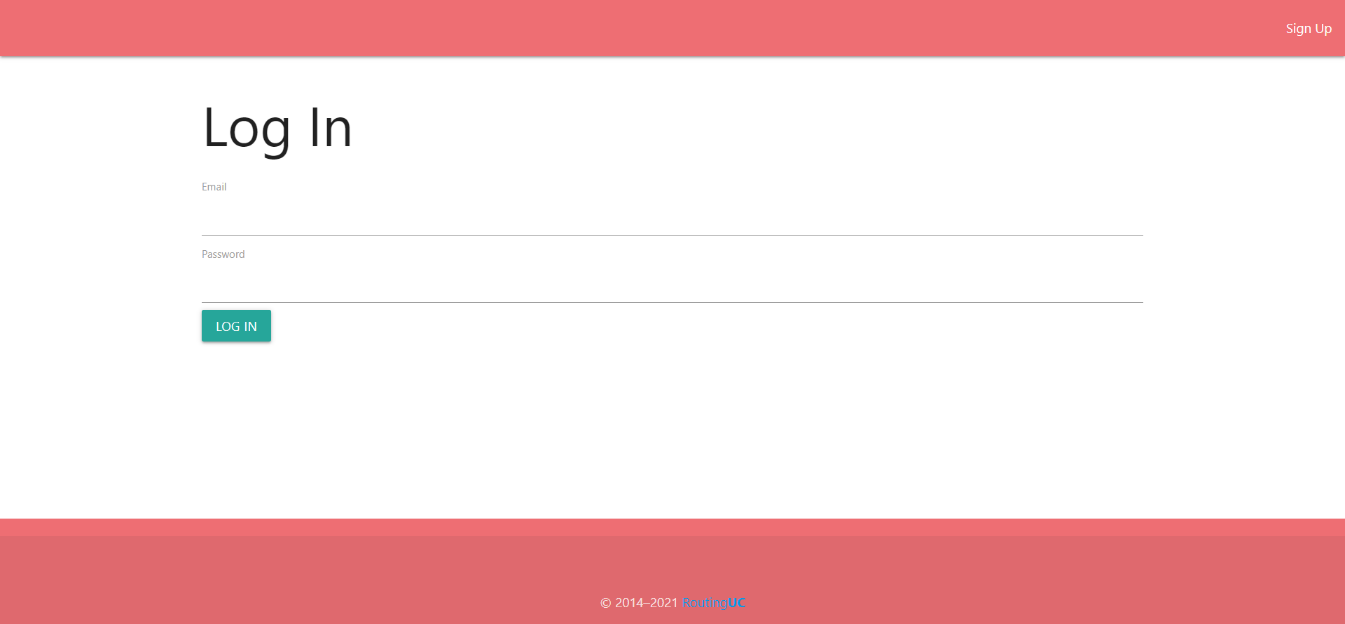
**SIGN UP de nuevo usuario**

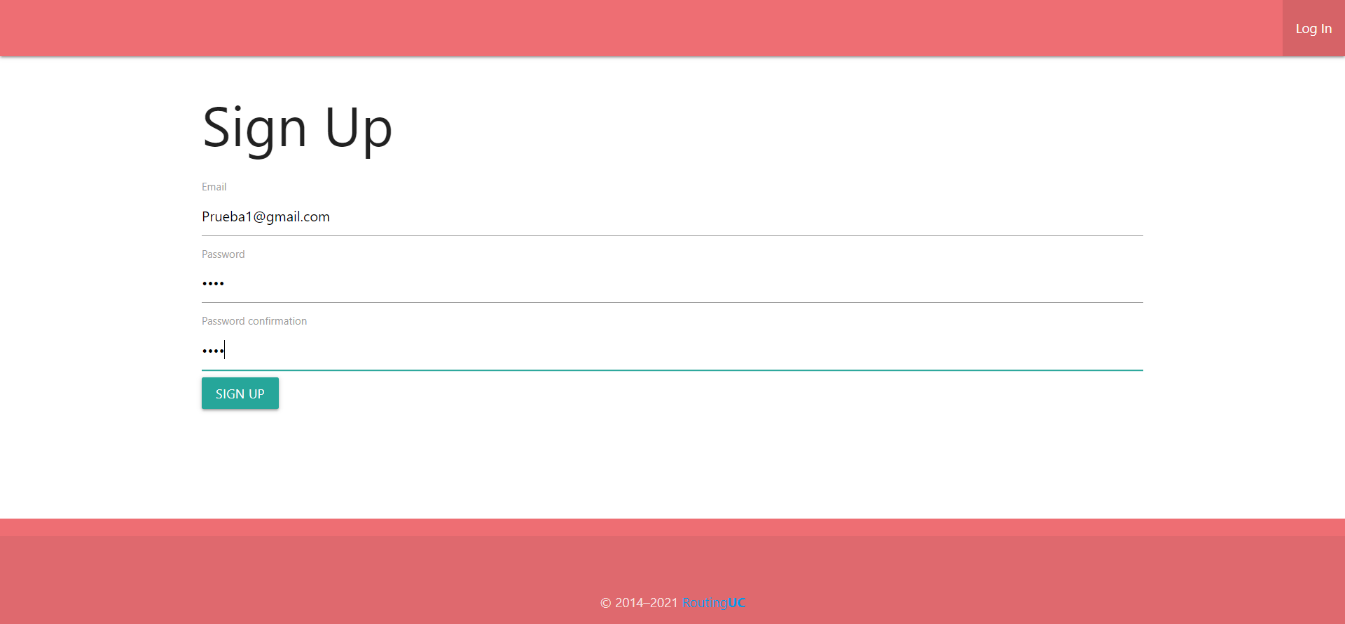


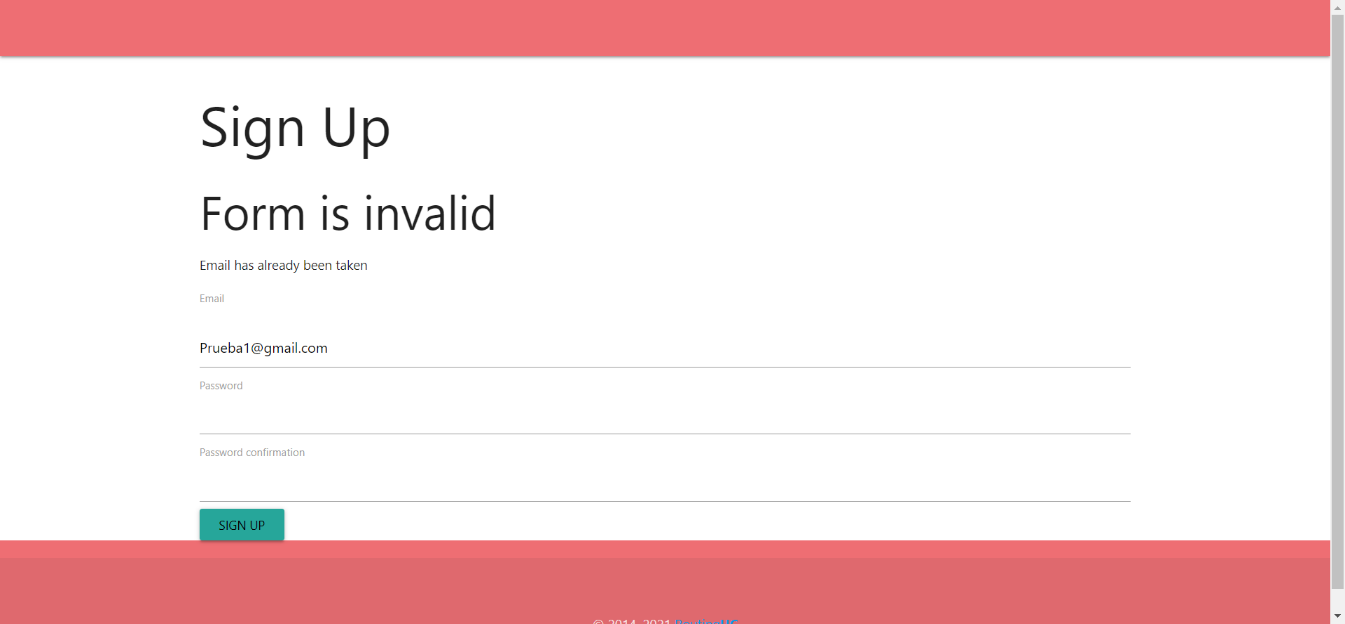




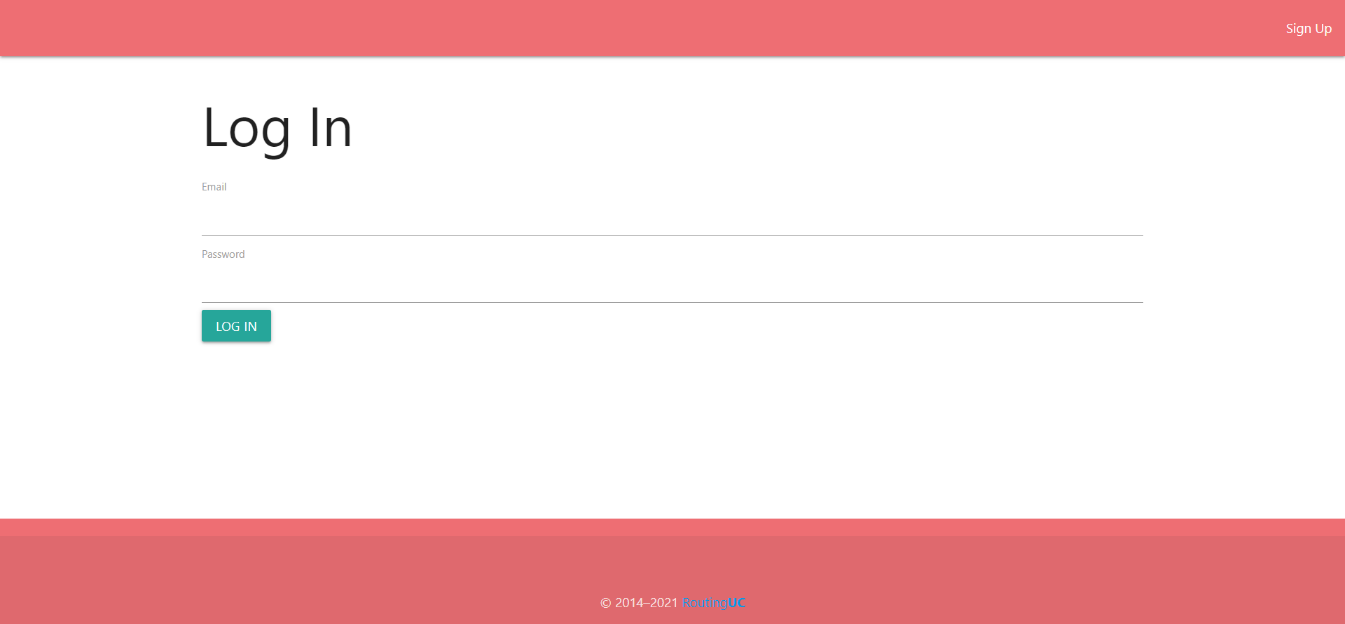
**Sign Up de usuario ya registrado**

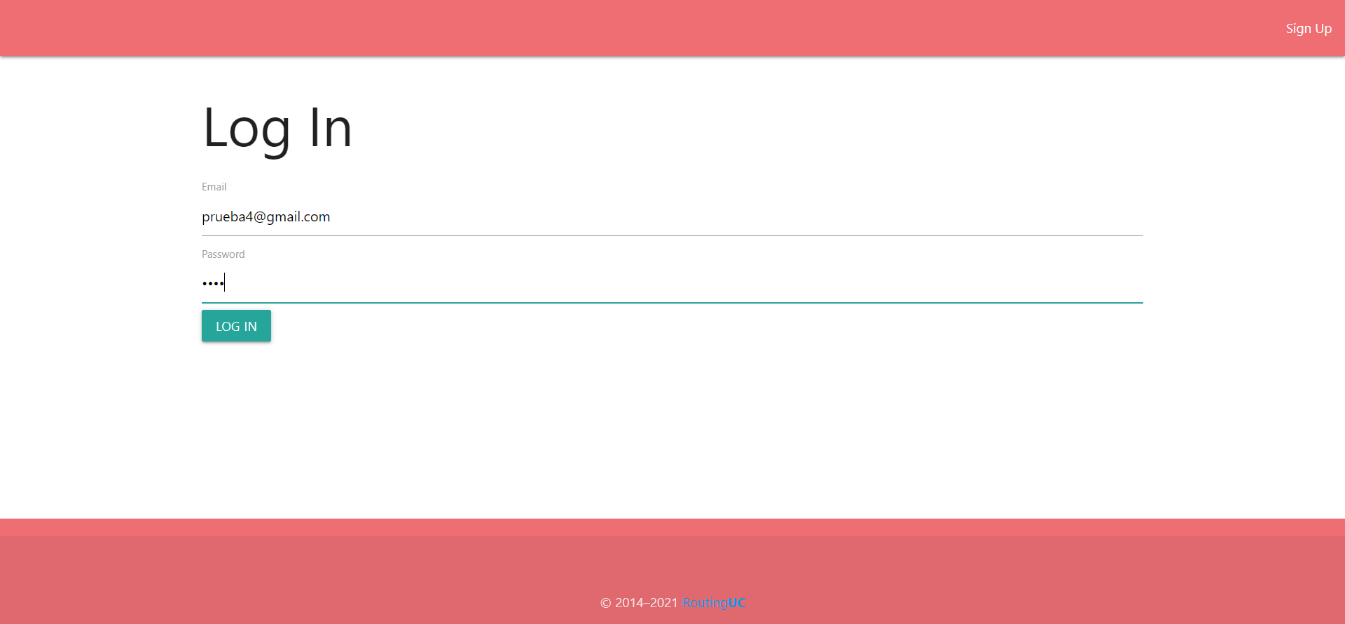


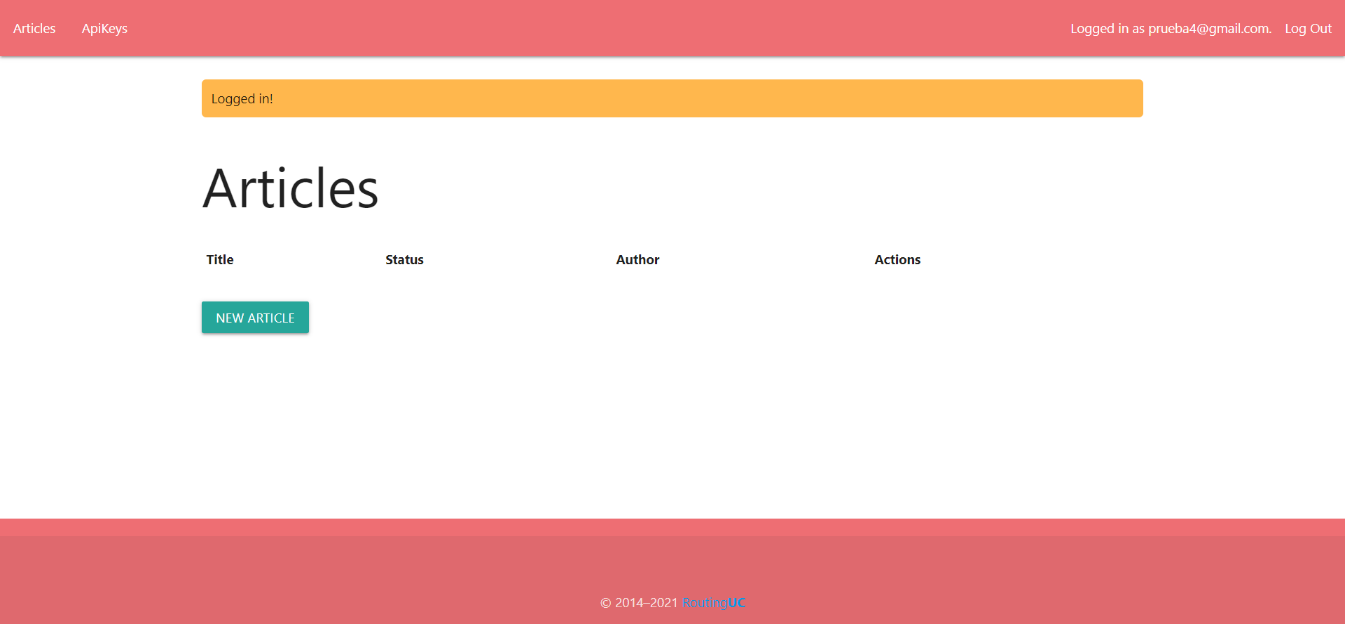




**API KEY**

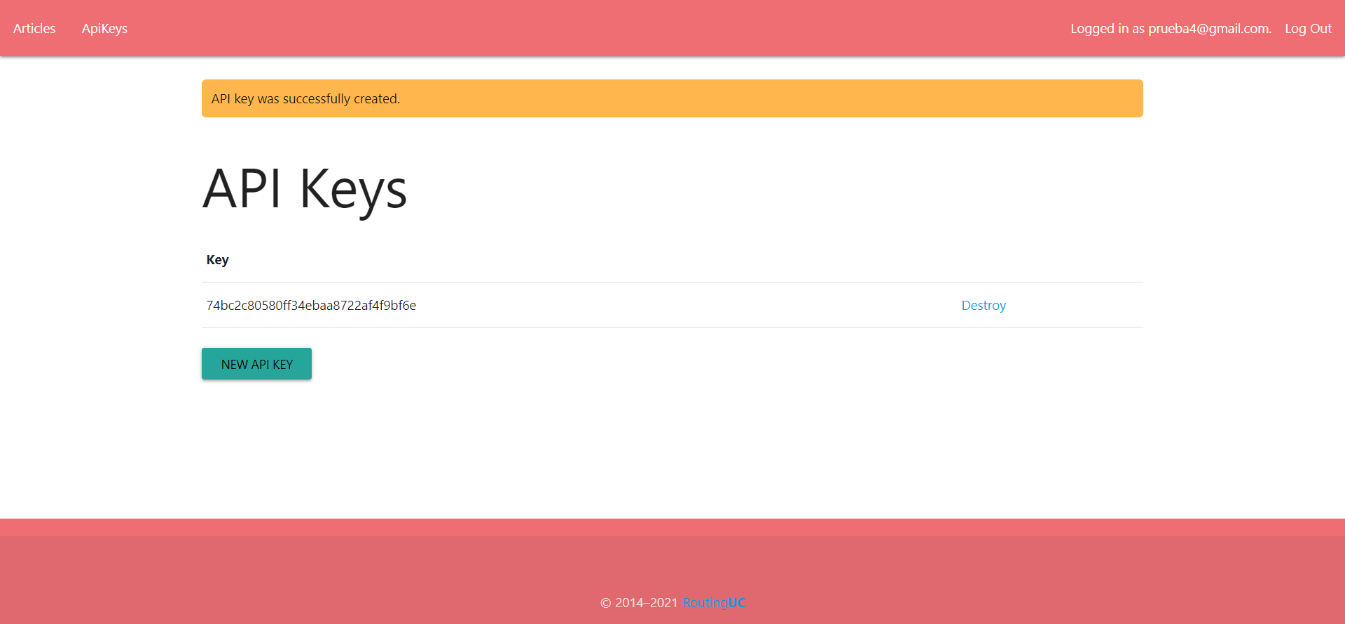




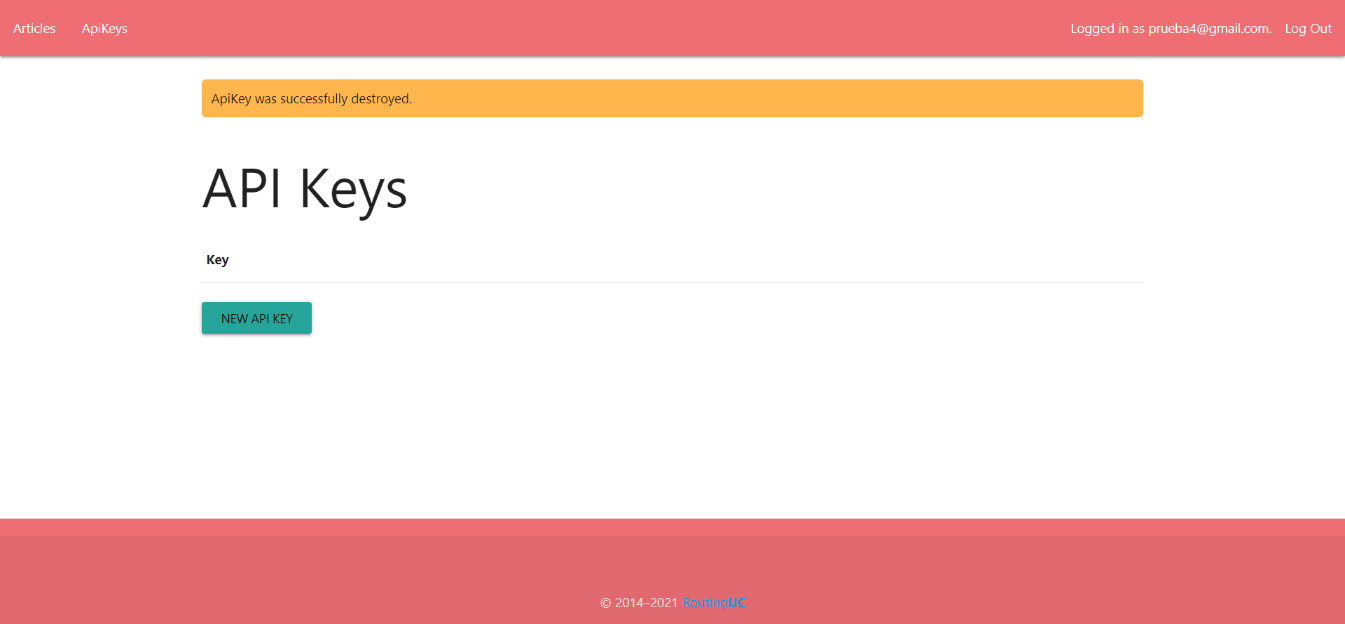




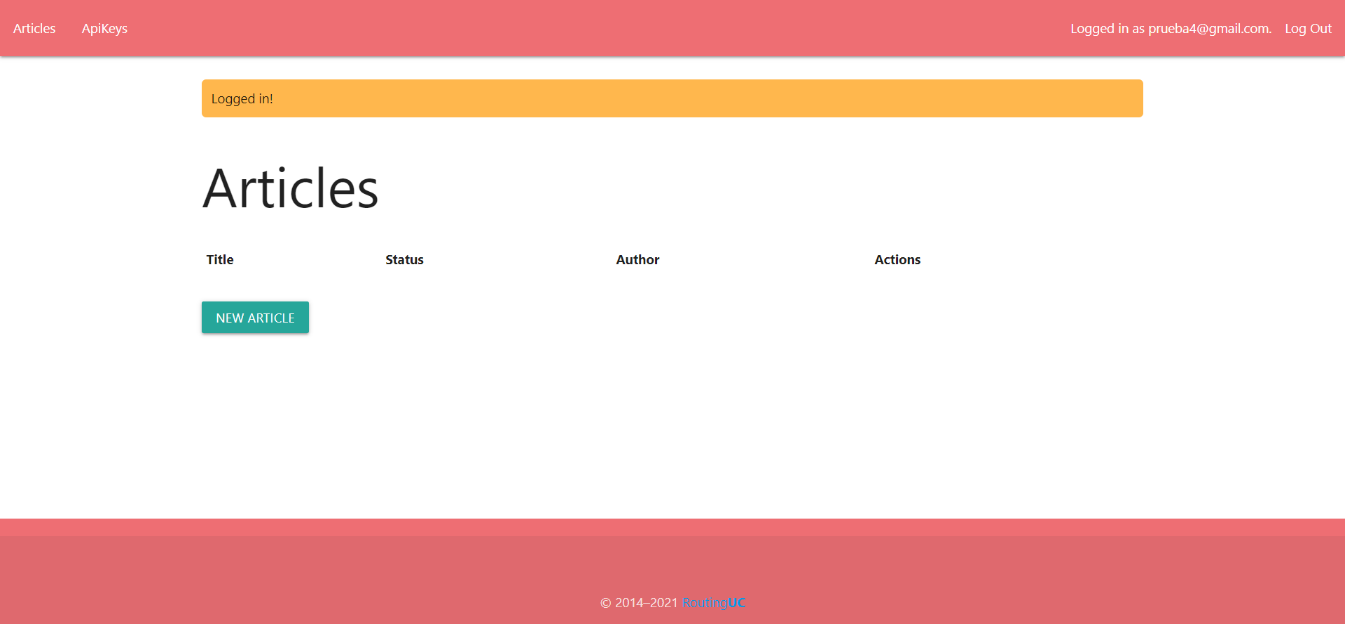
**Creación API KEY**

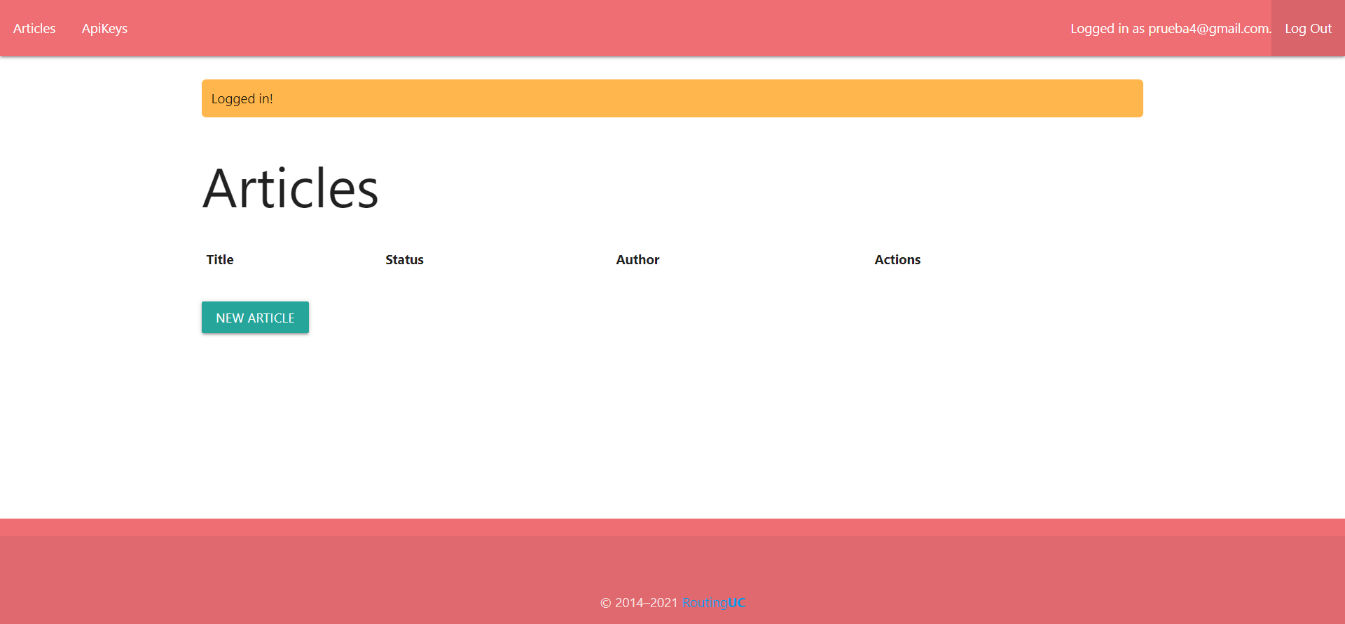


**Destroy API KEY**

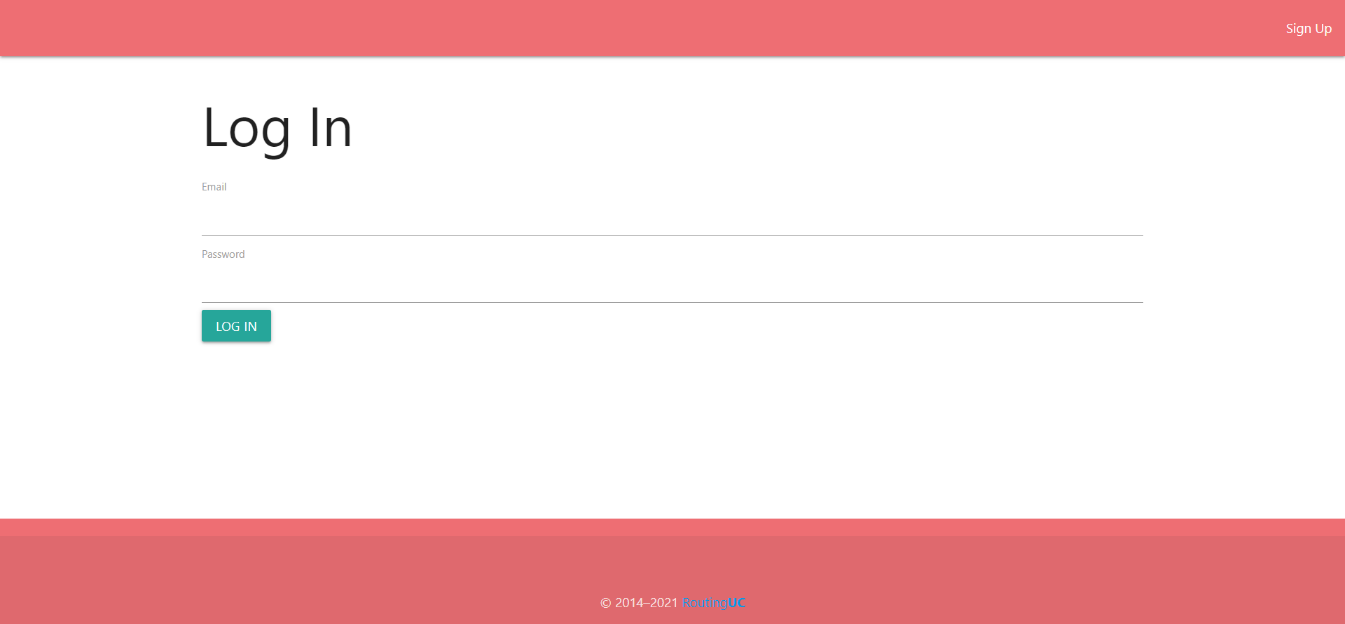


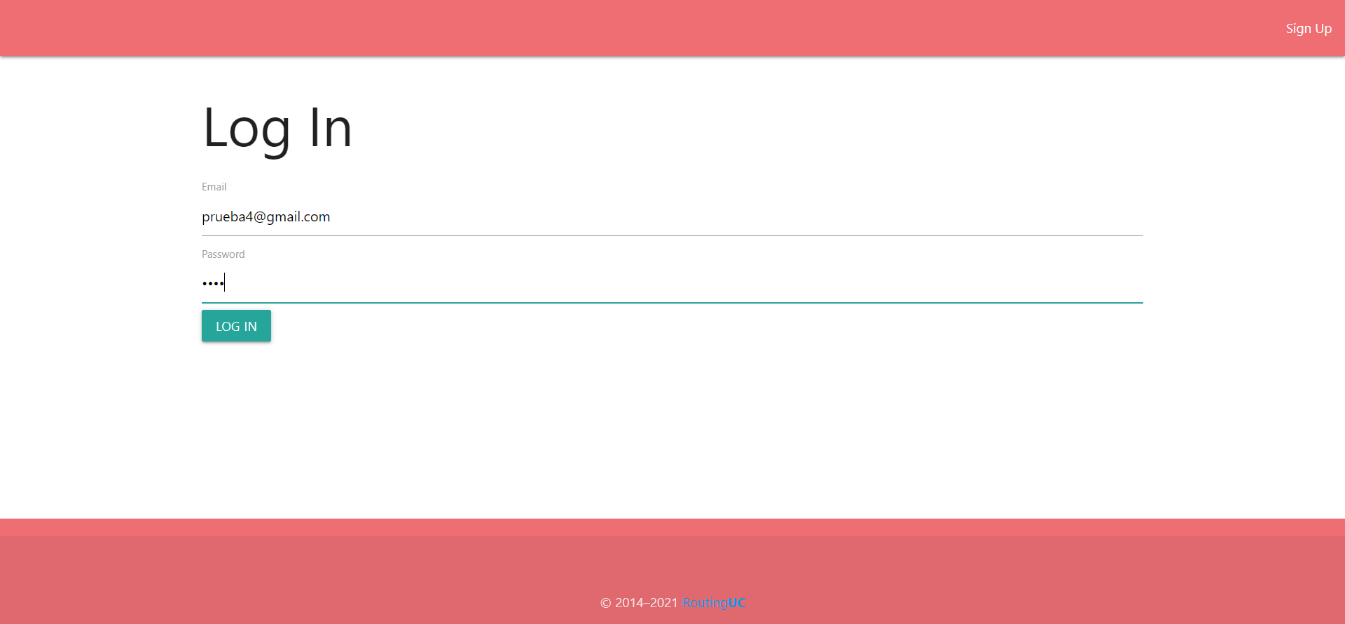
**Log Out**

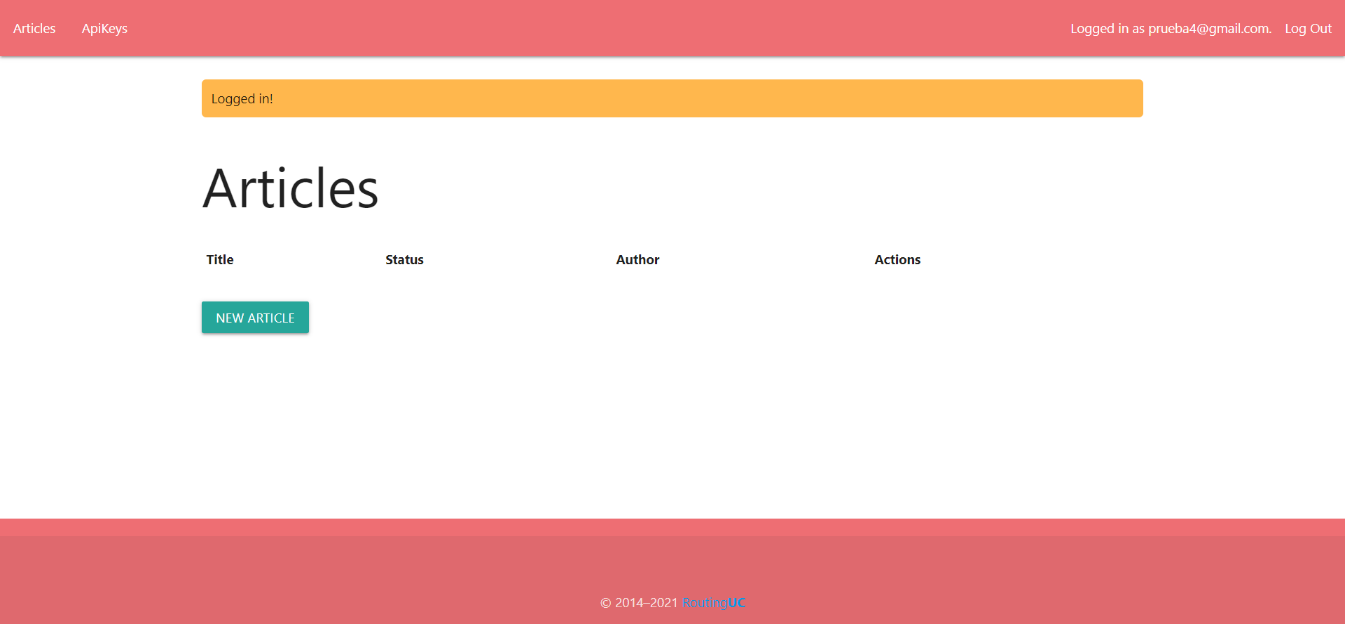




**Sección Artículos**



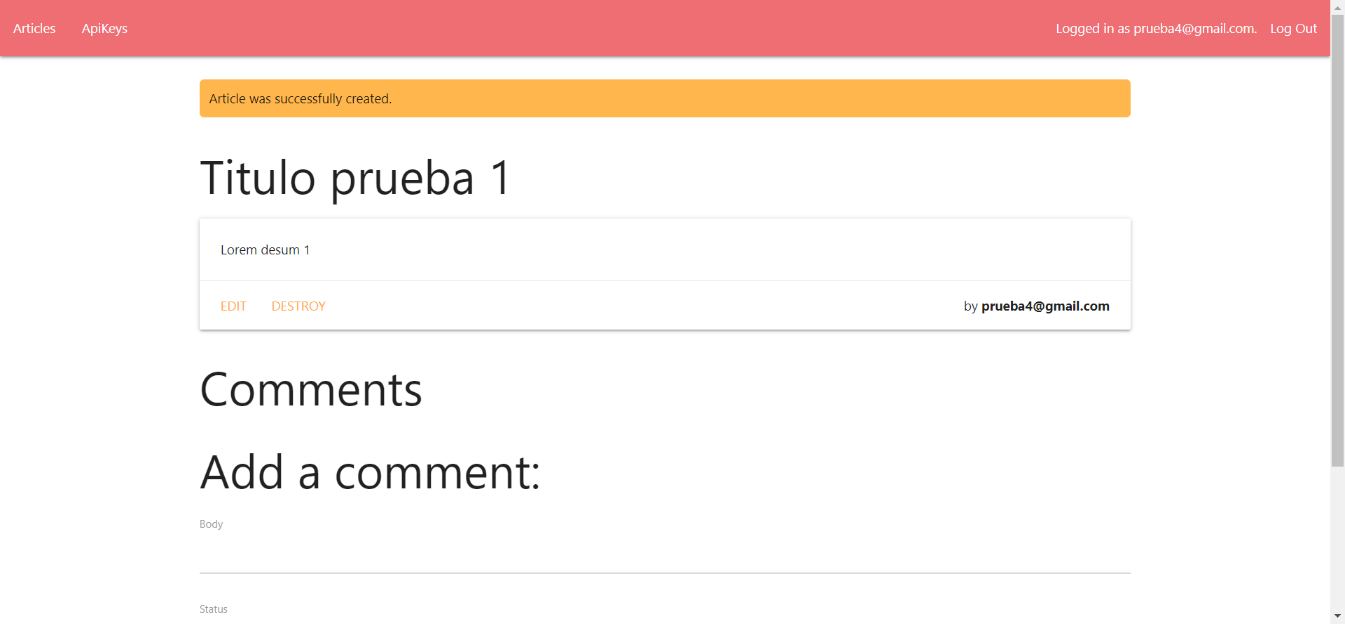




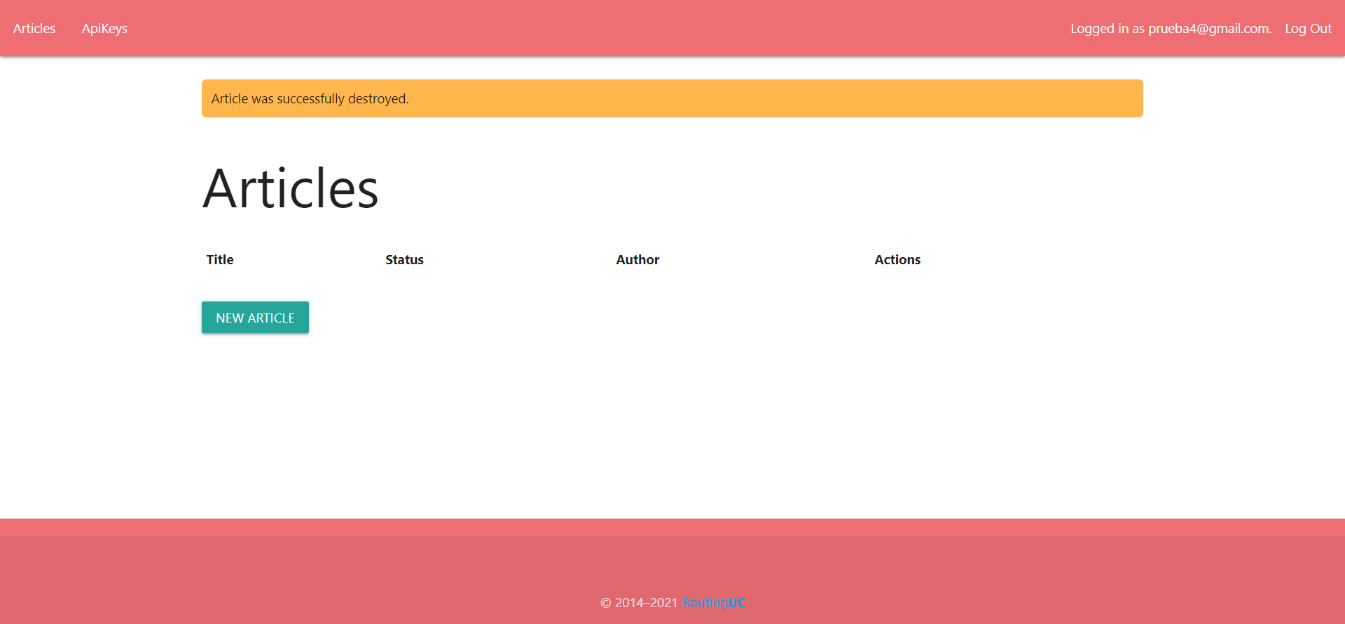
**Crear Nuevo Articulo**







**Destruir Articulo Creado**



# 6 Conclusiones

Dentro del cambio del plan de requisitos, se necesita la opción de recuperación de la clave, junto con la validación del correo como medida de seguridad.

# 7 Repositorio de Pruebas

https://github.com/GuillermoEchague/TestQA

# 8 Anexos

## 8.1 RSpec

Para instalar la gema rspect se debe agregar en el archivo Gemfile.rb del proyecto desarrollado en RoR de la siguiente forma:

**group :development, :test do**

**# Call 'byebug' anywhere in the code to stop execution and get a debugger console**

**gem 'byebug', platforms: [:mri, :mingw, :x64\_mingw]**

**gem 'rspec-rails', '~> 3.5'**

**end**

**group :test do**

**gem 'factory\_bot\_rails'**

**gem 'shoulda-matchers'**

**gem 'faker'**

**gem 'database\_cleaner'**

**end**

Posteriormente se debe ejecutar el siguiente comando para actualizar las gemas del proyecto:

**$ bundle install**

Esta gema al tener un comportamiento especial debe ser instalada de forma individual

**$ rails generate rspec:install**

Para ejecutar la sección de test se ejecuta por medio del siguiente código:

**$ bundle exec rspec**

Las siguientes lineas de codigo deben ser agregados en el archivo creado anteriormente rails\_helpers.rb para instalar gemas como shoulda-matchers.

**# Line 8 Add additional requies below ...**

**Shoulda::Matchers.configure do |config|**

**config.integrate do |with|**

**with.test\_framework :rspec**

**# Keep as many of these lines as are necessary:**

**with.library :active\_record**

**with.library :active\_model**

**with.library :action\_controller**

**with.library :rails**

**end**

**end**

Dentro de RSpec.configure do |config|, agregar

**config.include FactoryBot::Syntax::Methods**

**config.before(:suite) do**

**DatabaseCleaner.strategy = :transaction**

**DatabaseCleaner.clean\_with(:truncation)**

**end**

**config.around(:each) do |example|**

**DatabaseCleaner.cleaning do**

**example.run**

**end**

**end**