Universidad ORT Uruguay
Facultad de Ingeniería

Obligatorio 1 de Diseño de Aplicaciones 2

Agustín Juárez - 236487, Agustín Campón - 233006

Tutor: Nicolas Fierro

Repositorio con la solución

2023

Evidencia de la ejecución de las pruebas de la API con Postman

Casos de Prueba

Para los endpoints que no mencionan sobre el header de Authorization, se debe agregar el mismo: headers: { Authorization: <token> }

Un usuario debe poder registrarse

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado

1	Crear usuario con datos válidos y roles	POST /api/users
		
		Body: {
		"firstName": "John",
		"lastName": "Doe",
		"email": "johndoe@example.com",
		“username” : “JohnDoe”,
		"password": "password",
		"roles": [1, 2]
		}
			Status code: 201 (Created)
			Response Body: {
			"id": <new_user_id>,
			"firstName": "John",
			"lastName": "Doe",
			"email": "johndoe@example.com",
			“username” : “JohnDoe”,
			"roles": [1, 2]
			}

2	Crear usuario con datos requeridos faltantes	POST /api/users
		
		Body: {
		"lastName": "Doe",
		"email": "johndoe@example.com",
		“username” : “JohnDoe”,
		"password": "password",
		"roles": [1,2]
		}
			Status code: 400 (Bad Request)
			
			Response body: "The 'firstName' field is required."

3	Crear usuario con email invalido	POST /api/users
		
		Body: {
		"firstName": "John",
		"lastName": "Doe",
		"email": "johndoexample.com",
		“username” : “JohnDoe”,
		"password": "password",
		"roles": [1,2]
		}
			Status code: 400 (Bad Request)
			
			Response body: "The 'email' field is not a valid email address."

4	Crear usuario con email duplicado	POST /api/users Body: { "firstName": "John", "lastName": "Doe",	Status code: 409 (Conflict) Response body: "A user with the username 'JohnDoe' already exists.”

		"email": "johndoe@example.com", “username” : “JohnDoe”, "password": "password", "roles": [1,2] }	
5	Crear usuario con rol inexistente	POST /api/users Body: { "firstName": "John", "lastName": "Doe", "email": "johndoe@example.com", “username” : “JohnDoe”, "password": "password", "roles": [1,100] }	Status code: 404 (Not Found) Response body: "Could not find specified Role 100"
6	Crear usuario sin rol	POST /api/users Body: { "firstName": "John", "lastName": "Doe", "email": "johndoe@example.com", “username” : “JohnDoe”, "password": "password", "roles": [] }	Status code: 400 (Bad Request) Response body: "User should have at least one role"

Un usuario debe poder modificar su perfil

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado
7	Modificar usuario correctamente	PUT /api/users/{id} Donde {id} es el ID del usuario autenticado Body: { "firstName": "John", "lastName": "Doe", "email": "johndoe@example.com", "roles": [1, 2] }	Status code: 200 (OK) Response body: { "id": {id}, "firstName": "John", "lastName": "Doe", "email": "johndoe@example.com", "roles": [1, 2] }
8	Modificar usuario con datos faltantes.	PUT /api/users/{id} Body: { "roles": [1, 2]	Status code: 400 (Bad Request) Response body: "Invalid input: missing or invalid fields" (o

		}	mensaje similar)

9	Modificar usuario con rol no existente.	PUT /api/users/{id}
		
		Body: {
		"roles": [99],
		"username": "testuser",
		"password": "testpassword",
		"email": "testuser@test.com"
		}
			Status code: 404(Not Found)
			
			Response body: "Could not find specified Role 99"

Un usuario debe poder loguearse en el sistema

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado

10	Credenciales de Login válidas.	POST /api/sessions
		
		Body: {
		"email": "admin@hotmail.com",
		"password": "cleancode2023"
		}
			Status code: 200 (OK)
			
			Response body:
			{
			"token": <new_token>
			}

11	Credenciales de Login inválidas con usuario no existente en el sistema.	POST /api/sessions
		
		Body: {
		"email": "user1@gmail.com",
		"password": "12345678"
		}
			Status code: 400 (Bad Request)
			
			Response body: "Invalid credentials"

12	Contraseña faltante.	POST /api/sessions
		
		Body: {
		"email": "user@gmail.com"
		}
			Status code: 400 (Bad Request)
			
			Response body: {
			…,
			"errors": {
			"Password": [
			"The Password field is required."
			]
			}
			
			}

Un usuario debe poder desloguearse del sistema

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado

13	Desloguearse con el token correcto.	DELETE /api/sessions
		
		headers: {
		“Authorization” : “asdasd12213”
		}
			Status code: 204 (No Content)

14	Desloguearse con el token incorrecto.	DELETE /api/sessions
		
		headers: {
		“Authorization” : “”
		}
			Status code: 401 (Unauthorized)
			
			Response body: {
			"message": "Authorization header is missing"
			}

Un usuario debe poder crear, modificar o eliminar artículos

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado

15	Crear artículo con datos válidos.	POST /api/articles
		
		Body: {
		"title": "My first public article",
		"Private": false,
		"content": "My first content",
		"image": "https://example.com/article.jpg",
		"template": 0,
		"createdAt": "2023-04-15T12:00:00Z"
		}
			Status code: 201 (Created)
			
			Response body: {
			"id": <new_article_id>,
			"authorId": <usuario_logueado_id>,
			"title": "My first public article",
			"private": false,
			"comments": null,
			"content": "My first content",
			"image": "https://example.com/article.jpg",
			"template": 0,
			"createdAt": "2023-05-02T19:11:40.8301144-03:00",
			"updatedAt": null,
			"deletedAt": null
			}

16	Crear artículo con datos faltantes.	POST /api/articles Body: { "title": “”, "Private": false, "content": "My first content", "image": "https://example.com/article.jpg", "template": 0, "createdAt":	Status code: 400 (Bad Request) Response body: “Title or content empty”

		"2023-04-15T12:00:00Z" }	

17	Crear artículo con datos inválidos; template.	POST /api/articles
		
		Body: {
		"title": "My first article",
		"Private": false,
		"content": "My first content",
		"image": "https://example.com/article.jpg",
		"template": 10,
		"createdAt": "2023-04-15T12:00:00Z"
		}
			Status code: 400 (Bad Request)
			
			Response body: “Invalid template”

18	Crear artículo sin estar autenticado.	POST /api/articles
		
		headers: {
		“Authorization” : “”
		}
			Status code: 401 (Unauthorized)
			
			Response body: {
			"message": "Authorization header is missing"
			}

19	Modificar artículo con datos válidos.	PUT /api/articles/{id}
		
		Body: {
		"title": "My first public article",
		authorId: 1,
		"Private": false,
		"content": "My first content",
		"image": "https://example.com/article.jpg",
		"template": 0,
		"createdAt": "2023-04-15T12:00:00Z"
		}
			Status code: 200 (OK)
			
			Response body: {
			"id": {id},
			"authorId": 1,
			"title": "My first public article",
			"private": false,
			"comments": null,
			"content": "My first content",
			"image": "https://example.com/article.jpg",
			"template": 0,
			"createdAt": "2023-05-02T19:11:40.8301144-03:00",
			"updatedAt": null,
			"deletedAt": null
			}

20	Modificar artículo con datos faltantes.	PUT /api/articles/{id}
		
		Body: {
		"title": “”,
		"Private": false,
		"content": "My first content",
		"image": "https://example.com/article.jpg",
		"template": 0,
		"createdAt": "2023-04-15T12:00:00Z"
		}
			Status code: 400 (Bad Request)
			
			Response body: “Title or content empty”

21	Modificar artículo con datos inválidos; template.	PUT /api/articles/{id}
		
		Body: {
		"title": "My first article",
		"Private": false,
		"content": "My first content",
		"image": "https://example.com/article.jpg",
		"template": 10,
		"createdAt": "2023-04-15T12:00:00Z"
		}
			Status code: 400 (Bad Request)
			
			Response body: “Invalid template”

22	Modificar artículo sin estar autenticado.	PUT /api/articles/{id}
		
		headers: {
		“Authorization” : “”
		}
			Status code: 401 (Unauthorized)
			
			Response body: {
			"message": "Authorization header is missing"
			}

23	Eliminar artículo estando autenticado con el autor del mismo.	DELETE /api/articles/{id}
		
		headers: {
		“Authorization” : <article_author_token>
		}
			Status code: 204 (No Content)

24	Eliminar artículo estando autenticado con otro usuario no autor del artículo.	DELETE /api/articles/{id}
		
		headers: {
		“Authorization” : <other_author_token>
		}
			Status code: 401 (Unauthorized)
			
			Response body: {
			"message": "You are not authorized to perform this action"
			}

25	Eliminar artículo no existente.	DELETE /api/articles/{id}	Status code: 404 (Not Found) Response body: “Could not find specified article {id}”

Un usuario debe poder comentar artículos y responder comentarios de sus propios artículos

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado
26	Comentar el artículo estando autenticado con el autor del mismo.	POST /api/comments Request Body: {	Status code: 201 (Created) Response Body:

		"articleId": {id},
		"content": "nuevo comentario para el articulo 1"
		}
			{
			"id": <new_comment_id>,
			"authorId": 1,
			"replyId": null,
			"content": "nuevo comment para article 1",
			"articleId": {id},
			"createdAt": "2023-05-02T20:08:13.6649634-03:00",
			"updatedAt": null,
			"deletedAt": null
			}

27	Comentar un artículo no existente	POST /api/comments
		
		Request Body: {
		"articleId":{id},
		"content": "nuevo comentario para el articulo 1"
		}
			Status code: 404 (Not Found)
			
			Response body: “Could not find specified article {id}”

28	Responder comentario de un artículo estando autenticado con el autor del mismo	POST /api/comments/{id}/reply
		
		Request Body: {
		"content": "Mi respuesta al comentario {id}"
		}
			Status code: 201 (Created)
			
			Response body: {
			"id": <new_reply_id>,
			"authorId": 1,
			"replyId": null,
			"content": "Mi respuesta al comment {id}",
			"articleId": 1,
			"createdAt": "2023-05-02T20:17:46.2175431-03:00",
			"updatedAt": null,
			"deletedAt": null
			}

29	Responder comentario no existente de un artículo	POST /api/comments/{id}/reply
		
		Request Body: {
		"content": "Mi respuesta al comentario{id}"
		}
			Status code: 404 (Not Found)
			
			Response body: “Could not find specified comment {id}”

Un usuario debe tener la posibilidad de buscar un artículo a partir de un texto

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado

30	Buscar un artículo a partir de un texto	GET api/articles Query Params: q={texto}	Status code: 200 (OK) Response Body: [{ "id": 2, "authorId": 1, "title": "Mi articulo {texto}", "private": false, "comments": [], "content": "Mi contenido", "image": "https://example.com/article.jpg", "template": 0, "createdAt": "2023-05-02T19:11:40.8301144", "updatedAt": null, "deletedAt": null },]

<u>Un usuario Administrador debe tener la posibilidad de crear, modificar y eliminar usuarios</u>

ID	Nombre / Descripción	Datos / Pasos de prueba	Resultado esperado
31	Igual al caso de uso 1.	-	-
32	Igual al caso de uso 2.	-	-
33	Igual al caso de uso 3.	-	-
34	Igual al caso de uso 4.	-	-
35	Igual al caso de uso 5.	-	-
36	Igual al caso de uso 6.	-	-
37	Igual al caso de uso 7.	-	-
38	Igual al caso de uso 8.	-	-
39	Igual al caso de uso 9.	-	-
40	Igual al caso de uso 10.	-	-
41	Igual al caso de uso 11.	-	-
42	Igual al caso de uso 12.	-	-
43	Eliminar usuario estando autenticado como	DELETE /api/users/{id}	Status code: 204 (No Content)

	Administrador		
44	Eliminar usuario no existente estando autenticado como Administrador	DELETE /api/users/{id_no_existente}	Status code: 404 (Not Found) Response body: “Could not find specified user {id_no_existente}”

Video con evidencias de los casos de prueba mencionados anteriormente

Link