

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática
E.P. de Ingeniería de Software



Documentación del backend de DELU

Integrantes

Calle Huamantínco, Luis Eduardo	22200255
Calongos Jara, Leonid	22200102
Flores Cóngora, Paolo Luis	22200232
Matthew Alexandre, Pariona Molina	22200235
Moreno Zevallos Eva Lucía	20200277
Luján Vila Frank José	12200058

Curso: Gestión de la Configuración del Software

Docente: Wong Portillo, Lenis Rossi

ÍNDICE

1. Autenticación y registro de usuarios.

POST /login

POST /register

GET /verify/[email-token]

2. Comentarios y reviews

GET /professors/[professor]/comments/[course]/

POST /professors/[professor]/comments/[course]/

3. Professors

GET /professors

GET /professors/[id]

4. Forums

GET /forum/posts/

GET /forum/posts/[id]/

POST /forum/posts/

GET /forum/posts/[id]/comments/

POST /forum/posts/[id]/comments/

Documento de Documentación de API

1. Tecnologías

- Rust: Como lenguaje de programación de backend
- Rocket: Como framework web
- sqlx: Como driver para la comunicación con Postgres

2. Repositorio

El desarrollo del backend para PEP se hizo

[https://github.com/delusional-team/DELU/tree/](https://github.com/delusional-team/DELU/tree/0088488c5a118951bc54083b38e9fb0aee522ec3/desarrollo/PEP/fuentes/server)

[0088488c5a118951bc54083b38e9fb0aee522ec3/desarrollo/PEP/fuentes/server](https://github.com/delusional-team/DELU/tree/0088488c5a118951bc54083b38e9fb0aee522ec3/desarrollo/PEP/fuentes/server)

3. Documentación de endpoints

POST /login

- **Descripción:** Login de un usuario (estudiante) existente.

Request Body:

```
{  
  "email": "string",  
  "password": "string"  
}
```

- **Response:**

- **200 OK:** Usuario logueado exitosamente. Retorna también un JSON Web Token para futuras autenticaciones.

```
{  
  "token": "jwt"  
}
```

- **401 Unauthorized:** Credenciales inválidas

POST /register

- **Descripción:** Registra un nuevo estudiante y envía un correo de confirmación. El correo debe ser de San Marcos (@unmsm.edu.pe) para poder validarse como estudiante fisiano. El resto de datos como nombres, apellidos, base, grado, etc., son sacados del correo electrónico.

Request Body:

```
{  
  "student_code": "code",  
}
```

```
"email": "string",  
"password": "string"  
}
```

- **Response:**

- **201 Created:** Cuenta creada exitosamente. Correo de confirmación enviado.
- **400 Bad Request:** Credenciales inválidas.

GET /verify/[email-token]

- **Descripción:** Verifica a un usuario con solo ser clickeado, y lo redirige a la página principal.

- **Response:**

- **200 OK:** Verificado exitosamente.
 - **404 Not found:** Inválido o no disponible.
-

4. Comentarios y reviews

GET /professors/[professor]/comments/[course]/

- **Descripción:** Retrieve all comments related to a specific professor or course.

- **Query Parameters:**

- **professor_id:** ID del profesor (required).
- **course_id:** ID del curso que dictó (required).

- **Request Headers:**

- **Authorization:** Bearer <token>

- **Response:**

- **200 OK:** List of comments/reviews.

```
[  
  {  
    "-": "string",  
    "profile_image": "url",  
    "text": "string",  
    "rating": 5,  
    "datetime": "2024-09-16T11:12:01-5:00"  
  }  
]
```

- **404 Not Found:** No comments found.

POST /professors/[professor]/comments/[course]/

- **Descripción:** Crea un nuevo comentario en un curso que ha dictado un profesor.
- **Query Parameters:**
 - `professor_id`: ID del profesor (required).
 - `course_id`: ID del curso que dictó (required).
- **Request Headers:**
 - `Authorization: Bearer <token>`

Request Body:

```
{  
  "professor_id": 123,  
  "course_id": 123,  
  "text": "string",  
  "rating": "number" // Rating should be between 1 and 5.  
}
```

- **Response:**
 - `201 Created`: Comentario creado exitosamente.
 - `401 Unauthorized`: No autorizado.
 - `404 Not Found`: El curso no existe para este profesor.
-

3. Professors

GET /professors

- **Descripción:** Lista de todos los profesores en paginación.
- **Response:**
 - `200 OK`: List of professors.

```
[  
  {  
    "id": "string",  
    "name": "string",  
    "grade": 6.9,  
    "department": "string"  
  }  
]
```

GET /professors/[id]

- **Descripción:** Información detallada de un profesor específico.
- **Path Parameters:**

- **id**: ID del profesor.
- **Response:**
 - **200 OK**: Professor details.

```
{  
  "id": "string",  
  "name": "string",  
  "department": "string",  
  "courses": [{...}, {...}],  
  "grade": 6.9,  
  "total_reviews": "number"  
}
```

- **404 Not Found**: Professor not found.
-

4. Forums

GET /forum/posts/

- **Descripción**: Devuelta todos los posts activos, en paginación.
- **Response:**
 - **200 OK**: List of forums.

```
[  
  {  
    "id": 123,  
    "title": "string",  
    "upvotes": 123,  
    "content": "string",  
    "user": {...}  
  }  
]
```

GET /forum/posts/[id]/

- **Descripción**: Devuelve el detalle de un post. Nótese que no retorna los comentarios, pues son aparte.
- **Response:**
 - **200 OK**: List of forums.

```
{  
  "id": "string",  
  "title": "string",  
  "content": "string",  
  "user": {...}
```

```

        "upvotes": 123,
        "content": "string",
        "user": {...}
    }

```

POST /forum/posts/

- **Descripción:** Crea un nuevo formulario de discusiones. Only authenticated users can create forums.
- **Request Headers:**
 - **Authorization:** Bearer <token>

Request Body:

```

{
  "title": "string",
  "content": "string"
}

```

- **Response:**
 - **201 Created:** Post creado exitosamente
 - **401 Unauthorized:** No autorizado.

GET /forum/posts/[id]/comments/

- **Descripción:** Obtiene, recursivamente, todos los comentarios de un post.
- **Query Parameters:**
 - **post_id:** ID del post (required).
- **Response:**
 - **201 Created:** Comentario creado exitosamente.

```

[
  {
    "id": 123,
    "user": {...},
    "text": "string",
    "upvotes": 5,
  }
]

```

- **401 Unauthorized:** No autorizado.
- **404 Not Found:** El curso no existe para este profesor.

POST /forum/posts/[id]/comments/

- **Descripción:** Crea un nuevo comentario en un post de foro.
- **Query Parameters:**
 - `post_id`: ID del post (required).
- **Request Headers:**
 - `Authorization: Bearer <token>`

Request Body:

```
{  
  "text": "string",  
}
```

- **Response:**
 - `201 Created`: Comentario creado exitosamente.
 - `401 Unauthorized`: No autorizado.
 - `404 Not Found`: El post no existe.