**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 Huamantinco, 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. 3**](#_heading=h.ah7m4q98u6uh)

[POST /login 3](#_heading=h.17ii2sbqs0d4)

[POST /register 3](#_heading=h.hr1itikvyje7)

[GET /verify/[email-token] 3](#_heading=h.gchutgmxq1q2)

[**2. Comentarios y reviews 4**](#_heading=h.uracfhnhuvdg)

[GET /professors/[professor]/comments/[course]/ 4](#_heading=h.6d2wd8er6v59)

[POST /professors/[professor]/comments/[course]/ 4](#_heading=h.3v6mrhb5gu40)

[**3. Professors 5**](#_heading=h.lx4e7xdu6mj7)

[GET /professors 5](#_heading=h.cehh6goc8tk)

[GET /professors/[id] 5](#_heading=h.39zz6ryoa1s2)

[**4. Forums 6**](#_heading=h.c819hujsgi3g)

[GET /forum/posts/ 6](#_heading=h.8y7w5vmwy18x)

[GET /forum/posts/[id]/ 6](#_heading=h.op8w191rnvak)

[POST /forum/posts/ 6](#_heading=h.pmoinewndlrw)

[GET /forum/posts/[id]/comments/ 7](#_heading=h.zcj4zbn6zr3b)

[POST /forum/posts/[id]/comments/ 7](#_heading=h.qpyhtpm1knae)

**Documento de Documentación de API**

### Tecnologías

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

### Repositorio

El desarrollo del backend para PEP se hizo

<https://github.com/delusional-team/DELU/tree/0088488c5a118951bc54083b38e9fb0aee522ec3/desarrollo/PEP/fuentes/server>

### 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.

### 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",

“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.