# **Documento de Análisis del Proyecto Web “Nikola Tesla”**

## **1. Información General del Proyecto**

### **1.1. Nombre del proyecto:**

Plataforma Web Nikola Tesla

### **1.2. Descripción general:**

El proyecto “Nikola Tesla” consiste en el desarrollo de una plataforma web de cursos en línea enfocada en la enseñanza de Programación en Python, Inteligencia Artificial (IA) y Ciencia de Datos con Python.

El objetivo es ofrecer un espacio moderno, dinámico e intuitivo donde los usuarios puedan registrarse, explorar los cursos disponibles e inscribirse para acceder al contenido educativo.

### **1.3. Justificación:**

En la actualidad, el aprendizaje en línea de tecnologías emergentes como Python e Inteligencia Artificial se encuentra en auge. La plataforma “Nikola Tesla” busca posicionarse como un entorno accesible y atractivo para estudiantes y profesionales que deseen adquirir competencias tecnológicas de forma remota y flexible.

## **2. Objetivos del Proyecto**

### **2.1 Objetivo General**

Diseñar y desarrollar una plataforma web educativa que permita la gestión de cursos en línea de programación, IA y ciencia de datos, utilizando Django (Python) como base del backend y un framework moderno para el frontend.

### **2.2 Objetivos Específicos**

* Implementar un sistema de registro, autenticación y gestión de usuarios.
* Desarrollar un módulo de administración de cursos (creación, edición y eliminación).
* Crear una interfaz amigable, moderna y responsive para la navegación de cursos.
* Establecer una API REST para la comunicación entre frontend y backend.
* Preparar la infraestructura para futuras funciones, como pasarela de pagos y emisión de certificados.

## **3. Alcance del Proyecto (Fase Inicial – MVP)**

Durante la primera fase se desarrollará un Producto Mínimo Viable (MVP) que incluirá:

### **3.1. Front End**

* Página de inicio con presentación de la organización y los cursos.
* Catálogo de cursos con búsqueda y filtrado básico.
* Página de detalle del curso (información, duración, requisitos).
* Registro e inicio de sesión de usuarios.
* Panel del usuario con visualización de cursos inscritos.

### **3.2. Back End (Django)**

* Modelos para usuarios, cursos y matrículas.
* Sistema de autenticación y permisos con Django Auth + JWT.
* API REST (usando Django REST Framework).
* Panel administrativo nativo de Django para gestión de contenido.
* Base de datos relacional (PostgreSQL o MySQL).

## **4. Análisis de Requerimientos**

### **4.1 Requerimientos Funcionales**

|  |  |
| --- | --- |
| Código | Descripción |
| RF01 | El sistema debe permitir el **registro de nuevos usuarios**. |
| RF02 | El sistema debe permitir el **inicio de sesión** mediante credenciales. |
| RF03 | El sistema debe mostrar un **catálogo de cursos disponibles**. |
| RF04 | El sistema debe mostrar el **detalle de cada curso** (temario, duración, descripción). |
| RF05 | El usuario podrá **inscribirse a cursos** disponibles (modo prueba). |
| RF06 | El usuario podrá **consultar sus cursos inscritos** desde su panel. |
| RF07 | El administrador podrá **gestionar cursos** (crear, editar, eliminar). |
| RF08 | El sistema deberá permitir **cerrar sesión** de forma segura. |

### **4.2 Requerimientos No Funcionales**

|  |  |
| --- | --- |
| Categoría | Descripción |
| Rendimiento | Las consultas a la API deben responder en menos de 2 segundos promedio. |
| Seguridad | Uso de contraseñas encriptadas, autenticación JWT y control de permisos. |
| Usabilidad | Diseño responsivo, interfaz limpia y navegación intuitiva. |
| Escalabilidad | Estructura modular para agregar nuevos cursos y módulos. |
| Mantenibilidad | Separación lógica entre frontend, backend y base de datos. |
| Disponibilidad | Arquitectura preparada para despliegue en servicios en la nube. |

## **5. Roles y Responsabilidades**

|  |  |  |
| --- | --- | --- |
| Rol | Integrantes | Funciones principales |
| Analistas del proyecto | 2 | Elaboración del análisis, documentación técnica, modelado funcional, diseño de requerimientos y coordinación del equipo. |
| Desarrolladores Back End (Django + Python) | 2 | Diseño de modelos, creación de API REST, configuración de base de datos, seguridad y lógica del negocio. |
| Desarrolladores Front End | 2 | Diseño y desarrollo de la interfaz de usuario, consumo de API, maquetación y pruebas de interfaz. |

## **6. Arquitectura Propuesta**

### **6.1 Tipo de Arquitectura**

Arquitectura Cliente–Servidor con comunicación mediante API REST.

### **6.2 Componentes Principales**

* Front End: React o Vue.js (SPA).
* Back End: Django + Django REST Framework.
* Base de Datos: PostgreSQL (opcionalmente MySQL).
* Autenticación: JWT (JSON Web Token).
* Servidor de despliegue: Render, Railway o AWS EC2.
* Control de versiones: GitHub.

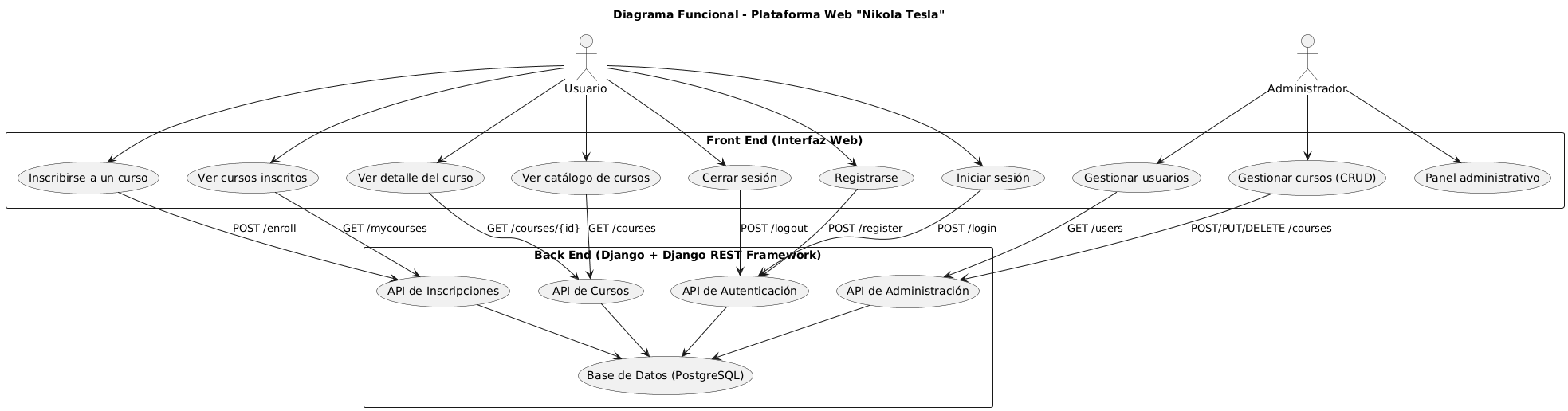
## **7. Herramientas y Tecnologías**

|  |  |
| --- | --- |
| Categoría | Herramienta / Tecnología |
| Lenguaje principal | Python |
| Framework backend | Django + Django REST Framework |
| Framework frontend | React o Vue.js |
| Base de datos | PostgreSQL |
| Control de versiones | Git + GitHub |
| Entorno de desarrollo | Visual Studio Code |
| Despliegue | Render / Vercel |
| Diseño UI/UX | Figma / Canva |

## **8. Entregables Iniciales**

* Documento de análisis del proyecto (este documento).
* Diagrama funcional del sistema (PlantUML).
* Modelo de datos preliminar (diagrama entidad-relación).
* Plan de trabajo inicial (cronograma por roles).
* Configuración de repositorios Git para cada equipo.

## **9. Diagrama funcional**



### **9.1. Explicación del Diagrama**

**Actores principales:**

* Usuario: Interactúa con la interfaz del front end para registrarse, iniciar sesión, navegar por cursos e inscribirse.
* Administrador: Gestiona los cursos y los usuarios desde el panel administrativo.

**Front End:**

Representa las pantallas y componentes que interactúan directamente con el usuario.

**Back End (Django):**

Contiene las API REST que procesan las solicitudes y gestionan los datos en la base de datos PostgreSQL.

**Comunicación:**

Cada caso de uso del front end se comunica con una API REST específica (autenticación, cursos, inscripciones o administración).

## **10. Diseño Conceptual del Modelo de Datos**

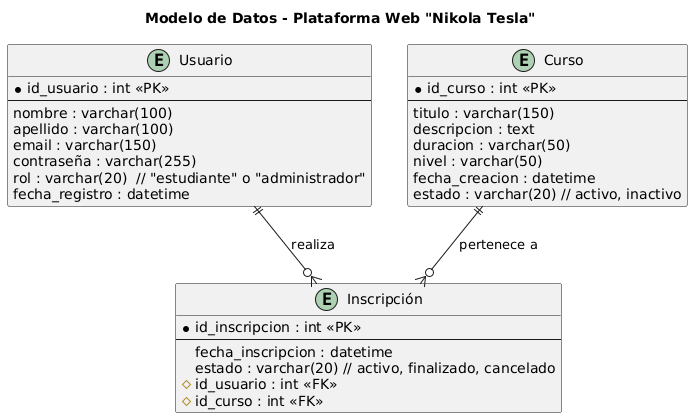
La base de datos debe permitir gestionar:

* Usuarios del sistema (alumnos y administradores)
* Cursos (nombre, descripción, duración, etc.)
* Inscripciones (relación entre usuarios y cursos)

### **10.1. Entidades Principales**

|  |  |  |
| --- | --- | --- |
| Entidad | Descripción | Atributos Clave |
| Usuario | Representa a los estudiantes o administradores registrados. | id\_usuario, nombre, email, contraseña, rol |
| Curso | Contiene la información de cada curso disponible. | id\_curso, título, descripción, duración, nivel |
| Inscripción | Relación entre un usuario y un curso al que se inscribe. | id\_inscripcion, fecha\_inscripcion, estado |

### **10.2. Relaciones**



**Usuario:**

Representa tanto a los estudiantes como a los administradores (el campo rol define el tipo).

Las contraseñas se almacenan encriptadas.

**Curso:**

Contiene los datos principales del curso y puede ser administrado por el administrador del sistema.

**Inscripción:**

Registra cada vez que un usuario se inscribe a un curso. Puede incluir el estado (activo, finalizado, cancelado).

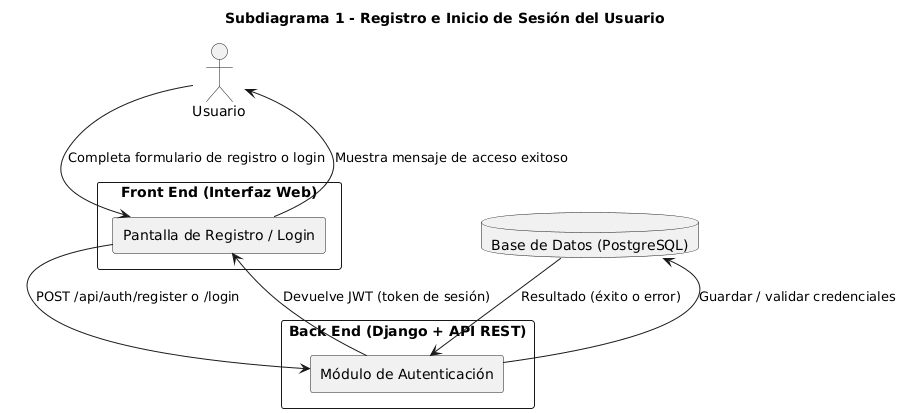
## **11. Descripción del flujo principal**

1. El usuario accede al portal y visualiza la página principal.
2. Si no tiene cuenta, se registra; si ya tiene, inicia sesión.
3. El sistema valida las credenciales a través de la API de autenticación (Django).
4. Una vez autenticado, el usuario puede:

* Ver el catálogo de cursos.
* Consultar el detalle de un curso.
* Inscribirse a un curso.
* Revisar sus cursos inscritos desde su panel.

1. El backend procesa las solicitudes y se comunica con la base de datos PostgreSQL.
2. El sistema responde al frontend con los datos solicitados.

### **11.1. Diagrama 1 - Registro e Inicio de Sesión del Usuario**

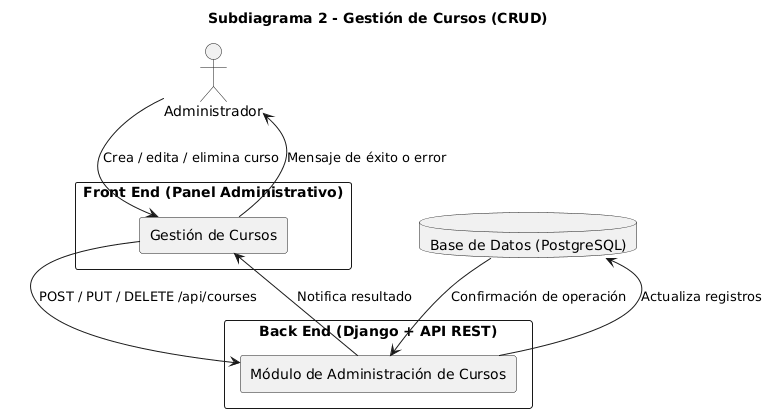


### **11.2. Diagrama 2 - Exploración y Detalle de Cursos**

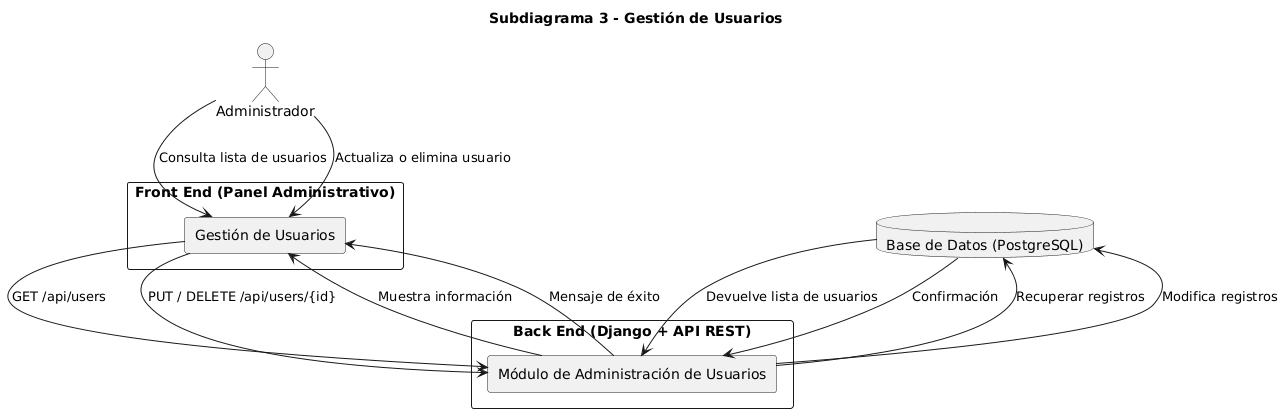
### **11.3. Diagrama 3 - Inscripción y Panel del Usuario**

### **11.4. Diagrama 4 - Administrador al Sistema**

### **11.5. Diagrama 5 - Gestión de Cursos (CRUD)**



### **11.6. Diagrama 6 - Gestión de Usuarios**



}}}}}

## **12. Cronograma de trabajo**

**Duración total:** 13/10/2025 → 21/11/2025 (6 semanas calendario)  
**Metodología sugerida:** Enfoque iterativo con entregables semanales (similar a Scrum simplificado).

## **13. Modelado de requerimientos**

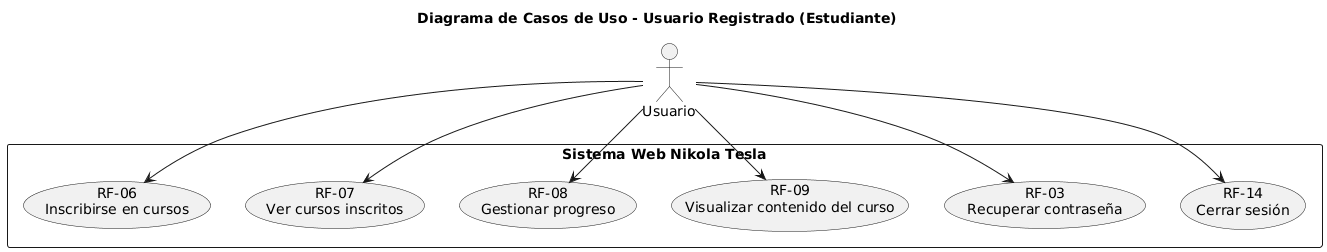
### **13.1. Requerimientos Funcionales del Sistema Web “Nikola Tesla”**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Código | Nombre del Requerimiento | Descripción | Entradas | Proceso / Regla de Negocio | Salidas / Resultado Esperado |
| RF-01 | Registro de usuario | Permite crear una cuenta nueva en el sistema ingresando datos personales (nombre, correo, contraseña). | Datos del usuario: nombre, correo, contraseña. | Validar que el correo no esté registrado. Encriptar contraseña antes de guardar. | Cuenta creada correctamente y mensaje de confirmación. |
| RF-02 | Autenticación de usuario | Permite iniciar sesión usando correo y contraseña. | Correo y contraseña. | Verificar credenciales. Generar token JWT para sesión activa. | Acceso al sistema y redirección al panel correspondiente. |
| RF-03 | Recuperación de contraseña | Permite solicitar una nueva contraseña mediante el correo asociado. | Correo electrónico registrado. | Validar existencia del usuario. Enviar enlace o código temporal de recuperación. | Correo de recuperación enviado. |
| RF-04 | Visualización de cursos disponibles | Muestra todos los cursos activos con su información básica. | Solicitud de vista de cursos. | Consultar base de datos y filtrar cursos activos. | Listado de cursos con nombre, duración, nivel y precio. |
| RF-05 | Búsqueda y filtrado de cursos | Permite buscar cursos por nombre, categoría o nivel. | Término de búsqueda y filtros. | Aplicar filtros dinámicos sobre los registros de cursos. | Resultados de búsqueda actualizados según criterios. |
| RF-06 | Inscripción a cursos | Permite al usuario inscribirse en un curso disponible. | ID de usuario e ID de curso. | Verificar si el usuario ya está inscrito. Registrar nueva inscripción. | Confirmación de inscripción exitosa. |
| RF-07 | Visualización de cursos inscritos | Muestra los cursos en los que el usuario está actualmente inscrito. | ID de usuario autenticado. | Consultar inscripciones activas del usuario. | Lista de cursos inscritos con estado de avance. |
| RF-08 | Gestión de progreso en cursos | Muestra el progreso del usuario dentro de cada curso. | ID de usuario, ID de curso. | Calcular porcentaje de progreso según lecciones completadas. | Porcentaje de avance mostrado en la interfaz. |
| RF-09 | Visualización de contenido del curso | Permite acceder al contenido educativo del curso (videos, materiales, cuestionarios). | ID de curso y usuario autenticado. | Verificar acceso (usuario inscrito). Mostrar contenido asociado. | Contenido mostrado correctamente. |
| RF-10 | Gestión de usuarios (CRUD) | Permite al administrador crear, editar y eliminar usuarios. | Datos de usuario (nombre, correo, rol). | Validar permisos del administrador. Ejecutar operaciones CRUD. | Usuario creado, editado o eliminado exitosamente. |
| RF-11 | Gestión de cursos (CRUD) | Permite al administrador gestionar los cursos ofrecidos. | Datos del curso (nombre, descripción, nivel, precio, estado). | Validar campos y permisos. Ejecutar operaciones CRUD. | Curso creado, editado o eliminado exitosamente. |
| RF-12 | Gestión de inscripciones | Permite al administrador visualizar y controlar las inscripciones de usuarios. | ID de usuario o curso. | Consultar registros de inscripciones y su estado. | Reporte o vista de inscripciones actualizadas. |
| RF-13 | Panel de administración | Permite visualizar un panel con estadísticas y accesos rápidos. | Rol = administrador. | Mostrar métricas de usuarios, cursos y progreso general. | Panel administrativo mostrado correctamente. |
| RF-14 | Cierre de sesión | Permite cerrar sesión de forma segura y finalizar el token de autenticación. | Token JWT del usuario. | Invalidar token y cerrar sesión activa. | Sesión cerrada y redirección a página de inicio. |

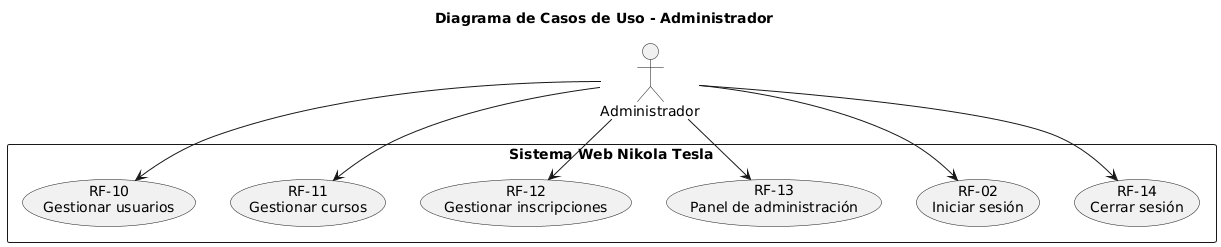
#### **13.1.1. Diagrama de Casos de Uso – Visitante**

#### 

#### **13.1.2. Diagrama de Casos de Uso - Usuario (Estudiante)**



#### **13.1.3. Diagrama de Casos de Uso – Administrador**



#### **13.1.4. Descripción Detallada de Casos de Uso**

##### ***CU-01: Registro de Usuario (RF-01)***

|  |  |
| --- | --- |
| Elemento | Descripción |
| Nombre del Caso de Uso: | Registro de Usuario |
| Código: | CU-01 |
| Actor Principal: | Visitante |
| Objetivo: | Permitir que un visitante cree una cuenta ingresando sus datos personales y un correo válido. |
| Precondiciones: | El visitante no debe tener una cuenta registrada con el mismo correo. |
| Flujo Principal: | 1. El visitante accede al formulario de registro.  2. Ingresa sus datos (nombre, correo, contraseña).  3. El sistema valida los campos.  4. Se guarda el usuario en la base de datos con la contraseña encriptada.  5. El sistema muestra un mensaje de registro exitoso. |
| Flujo Alterno: | 4ª.Si el correo ya está registrado, el sistema muestra un mensaje de error. |
| Postcondiciones: | La cuenta del usuario queda registrada en el sistema y puede iniciar sesión. |

##### ***CU-02: Autenticación de Usuario (RF-02)***

|  |  |
| --- | --- |
| Elemento | Descripción |
| Nombre del Caso de Uso: | Autenticación de Usuario |
| Código: | CU-02 |
| Actor Principal: | Usuario / Administrador |
| Objetivo: | Permitir que un usuario o administrador inicie sesión con su correo y contraseña. |
| Precondiciones: | El usuario debe estar previamente registrado. |
| Flujo Principal: | 1. El usuario ingresa su correo y contraseña.  2. El sistema valida las credenciales en la base de datos.  3. Si son válidas, genera un token JWT.  4. El sistema redirige al panel correspondiente (usuario o admin). |
| Flujo Alterno: | 2ª.Si las credenciales son incorrectas, se muestra un mensaje de error. |
| Postcondiciones: | El usuario queda autenticado y con sesión activa mediante token. |

##### ***CU-06: Inscripción a Cursos (RF-06)***

|  |  |
| --- | --- |
| Elemento | Descripción |
| Nombre del Caso de Uso: | Inscripción a Cursos |
| Código: | CU-06 |
| Actor Principal: | Usuario |
| Objetivo: | Permitir que el usuario se inscriba en un curso disponible. |
| Precondiciones: | El usuario debe estar autenticado. |
| Flujo Principal: | 1. El usuario selecciona un curso del catálogo.  2. Hace clic en “Inscribirse”.  3. El sistema verifica si ya está inscrito.  4. Si no lo está, registra la inscripción.  5. Muestra mensaje de confirmación. |
| Flujo Alterno: | 3ª.Si ya está inscrito, el sistema informa que no puede repetir la inscripción. |
| Postcondiciones: | La inscripción queda registrada en la base de datos. |

##### ***CU-09: Visualización de Contenido del Curso (RF-09)***

|  |  |
| --- | --- |
| Elemento | Descripción |
| Nombre del Caso de Uso: | Visualización de Contenido del Curso |
| Código: | CU-09 |
| Actor Principal: | Usuario |
| Objetivo: | Permitir al usuario acceder al material del curso en el que está inscrito. |
| Precondiciones: | El usuario debe estar autenticado y tener inscripción activa en el curso. |
| Flujo Principal: | 1. El usuario accede a “Mis Cursos”.  2. Selecciona el curso.  3. El sistema verifica inscripción.  4. Muestra el contenido (videos, documentos, cuestionarios). |
| Flujo Alterno: | 3ª.Si no está inscrito, se muestra un mensaje de acceso denegado. |
| Postcondiciones: | El contenido se carga correctamente en pantalla. |

##### ***CU-11: Gestión de Cursos (RF-11)***

|  |  |
| --- | --- |
|  |  |
| Elemento | **Descripción** |
| Nombre del Caso de Uso: | Gestión de Cursos |
| Código: | CU-11 |
| Actor Principal: | Administrador |
| Objetivo: | Permitir que el administrador cree, edite o elimine cursos. |
| Precondiciones: | El administrador debe estar autenticado. |
| Flujo Principal: | 1. El administrador accede al panel de cursos.  2. Puede crear, editar o eliminar un curso.  3. El sistema valida los datos ingresados.  4. Se actualiza la información en la base de datos.  5. Se muestra mensaje de confirmación. |
| Flujo Alterno: | 3ª.Si los datos están incompletos o inválidos, se muestra un error. |
| Postcondiciones: | La información del curso se actualiza en el sistema. |

##### ***CU-13: Panel de Administración (RF-13)***

|  |  |
| --- | --- |
| Elemento | Descripción |
| Nombre del Caso de Uso: | Panel de Administración |
| Código: | CU-13 |
| Actor Principal: | Administrador |
| Objetivo: | Permitir al administrador visualizar métricas del sistema y acceder a gestiones rápidas. |
| Precondiciones: | El administrador debe estar autenticado. |
| Flujo Principal: | 1. El administrador inicia sesión.  2. Accede al panel principal.  3. El sistema muestra métricas generales (usuarios, cursos, inscripciones). |
| Postcondiciones: | El panel muestra datos actualizados del sistema. |

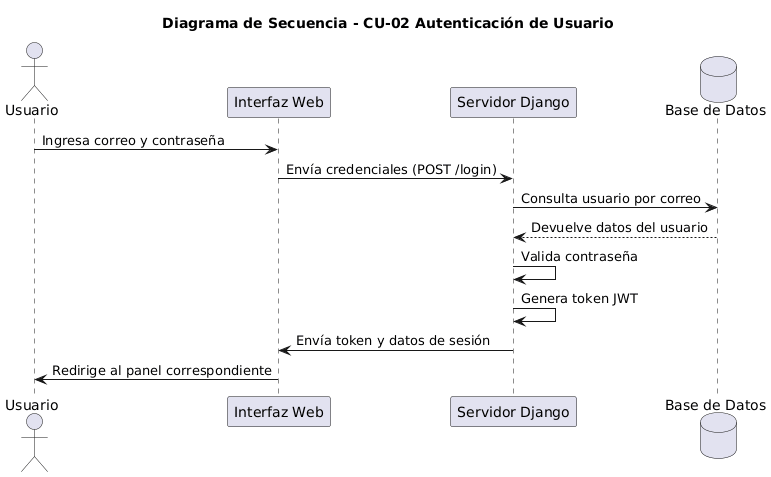
##### ***CU-14: Cierre de Sesión (RF-14)***

|  |  |
| --- | --- |
| Elemento | Descripción |
| Nombre del Caso de Uso: | Cierre de Sesión |
| Código: | CU-14 |
| Actor Principal: | Todos |
| Objetivo: | Permitir que cualquier usuario cierre su sesión de forma segura. |
| Precondiciones: | El usuario debe tener una sesión activa. |
| Flujo Principal: | 1. El usuario selecciona “Cerrar sesión”.  2. El sistema elimina el token de autenticación.  3. Redirige a la página de inicio. |
| Postcondiciones: | La sesión se termina y el acceso queda bloqueado hasta un nuevo login. |

#### **13.1.5. Diagrama de secuencia**

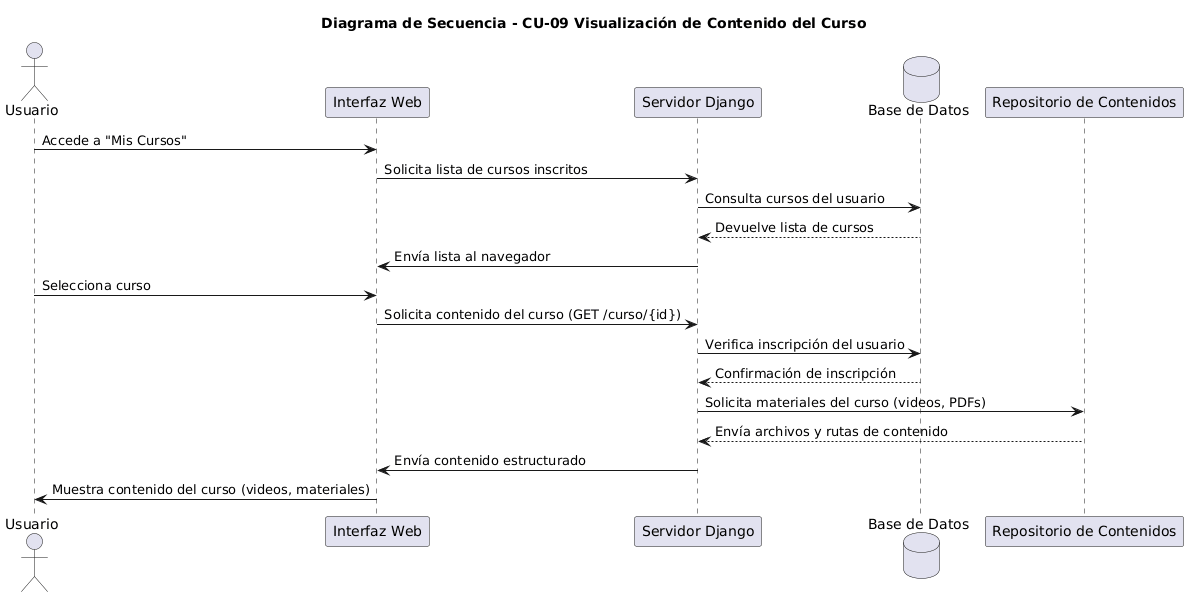
##### ***CU-01: Registro de Usuario***

##### ***CU-02: Autenticación de Usuario***

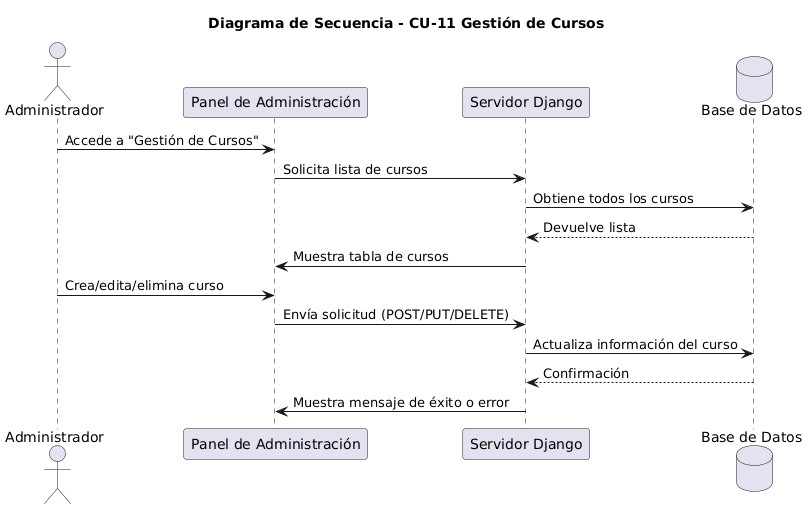


##### ***CU-06: Inscripción a Curso***

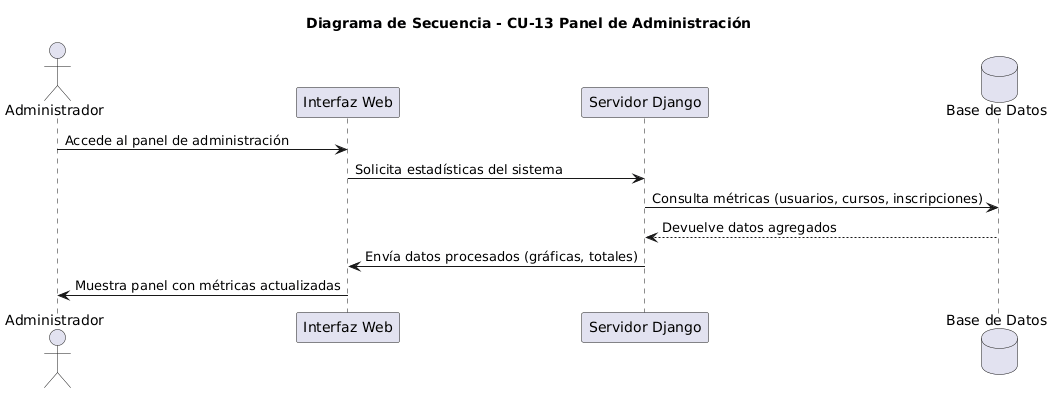
##### ***CU-09: Visualización de Contenido del Curso***



##### ***CU-11: Gestión de Cursos (Administrador)***



##### ***CU-13: Panel de Administración***



##### ***CU-14: Cierre de Sesión***