

COMPONENTE DE INICIO DE SESIÓN CON OAUTH 2.0 - CREA VI

Autores:

Daniel David Doria Diaz, Kissy Carolina Manjarrez Murillo, Jabib Roberto Manzur Lemos,
Daniel Alejandro Márquez Araujo, Jesús Ballesta Charrasquiel

Tutor: *Alexander Toscano Ricardo.*



CREA VI

MOODLE

Ingresar con CREA VI

Correo

example@correo.com

Contraseña

[Ver contraseña](#)

Iniciar Sesión

[Olvidé mi contraseña](#)

[Crear Usuario](#)

Inicia sesión o regístrate en cuestión de segundos

Usa tu correo electrónico u otro servicio para continuar



Continuar con Google



Continuar con Facebook



Continuar con correo electróni...

[Continuar de otra forma](#)

Al continuar, estás aceptando los [términos y condiciones de uso](#). Consulta nuestra [política de privacidad](#).



Regístrate con tu correo de trabajo

Contenido

COMPONENTE DE INICIO DE SESIÓN CON OAUTH 2.0 - CREA VI	1
Breve reseña	4
OAUTH 2.0	4
INICIO DE SESIÓN CON OAUTH 2.0	6
PROPIEDADES QUE INDICA EL PROVEEDOR OAUTH 2.0	8
Design Thinking	10
Introducción.....	12
Propósito del Documento	13
Alcance del Proyecto	15
Definiciones y Acrónimos	17
Descripción General	19
Objetivos del Sistema	19
Funcionalidades Generales	19
Usuarios del Sistema	20
Resquitos Funcionales	21
Requisitos no Funcionales	22
Requisitos de Desempeño.....	22
Casos de Uso	23
Diagrama de caso de usos.....	23
Diagramas de Flujo de Casos de Uso	24
Descripción detallada de cada caso de uso	27
Descripción detallada de cada caso de uso	28
Descripción detallada de cada caso de uso	30
Descripción detallada de cada caso de uso	32
Descripción detallada de cada caso de uso	34
Descripción detallada de cada caso de uso	36
FUNCIONALIDADES Y DATOS	39
Modelando Componente de inicio de sesión con OAuth 2.0 Creavi.....	40
Cardinalidad.....	41
Anexos	49

Breve reseña

Nuestra propuesta es realizar un componente que permita brindar al administrador agregar diferentes maneras de iniciar sesión con cuentas alternas como lo son Google, Facebook, X, Instagram.

OAUTH 2.0

En el modelo tradicional de autenticación cliente-servidor, el cliente solicita un recurso de acceso restringido (recurso protegido) en el servidor autenticándose con el servidor utilizando el nombre del propietario del recurso. cartas credenciales. Para proporcionar acceso a aplicaciones de terceros a recursos restringidos, el propietario del recurso comparte sus credenciales con el tercero. Esto crea varios problemas y limitaciones:

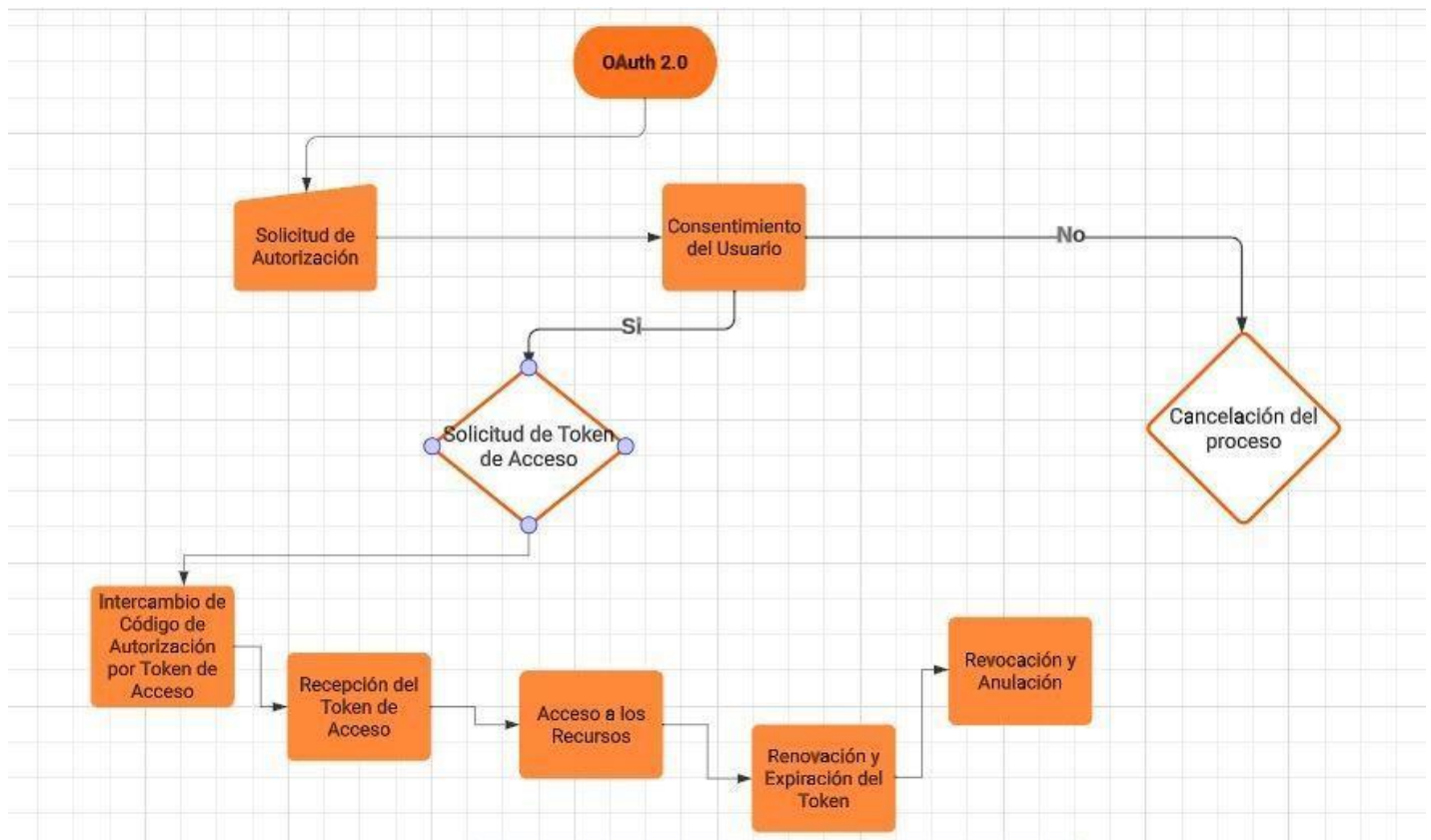
- Se requieren aplicaciones de terceros para almacenar el recurso. credenciales del propietario para uso futuro, generalmente una contraseña en Borrar texto.
- Se requiere que los servidores admiten la autenticación de contraseña, a pesar de las debilidades de seguridad inherentes a las contraseñas.
- Las aplicaciones de terceros obtienen un acceso demasiado amplio al recurso. recursos protegidos del propietario, dejando a los propietarios de recursos sin ninguna capacidad de restringir la duración o el acceso a un subconjunto limitado de recursos.
- Los propietarios de recursos no pueden revocar el acceso a un tercero individual sin revocar el acceso a todos los terceros,

OAuth aborda estos problemas introduciendo una capa de autorización y separar el rol del cliente del del recurso dueño. En OAuth, el cliente solicita acceso a los recursos controlados por el propietario del recurso y alojado por el servidor de recursos, y es emitido un conjunto de credenciales diferente a las del recurso dueño. En lugar de utilizar las credenciales del propietario del recurso para acceder a sitios protegidos, el cliente obtiene un token de acceso, una cadena que denota un alcance específico, duración y otros atributos de acceso. Los tokens de acceso son emitidos a clientes de terceros por un servidor de autorización con la aprobación del propietario del recurso.

El cliente utiliza el token de acceso para acceder a los recursos protegidos alojados en el servidor de recursos. Por ejemplo, un usuario final (propietario del recurso) puede otorgar una impresión acceso del servicio (cliente) a sus fotos protegidas almacenadas en un servicio compartido

(servidor de recursos), sin compartir su nombre de usuario y contraseña con el servicio de impresión. En cambio, ella autentica directamente con un servidor de confianza del servicio para compartir fotos (servidor de autorización), que emite la delegación del servicio de impresión credenciales específicas (token de acceso).

Este es el diagrama de flujo de OAuth 2.0



INICIO DE SESIÓN CON OAUTH 2.0

Generalmente implica permitir que los usuarios se autenticuen en la aplicación utilizando sus credenciales de un proveedor de autenticación externo, como Google, Facebook, GitHub, o cualquier otro proveedor de OAuth 2.0. A continuación, se mostrará el diagrama de flujo al implementar el inicio de sesión con OAuth 2.0 en una aplicación web:

Solicitud de Autorización

El proceso comienza cuando un usuario desea dar permiso a una aplicación para acceder a sus datos en un servidor de recursos (por ejemplo, una cuenta de Google, Facebook o una API web).

La aplicación redirige al usuario a un servidor de autorización con una solicitud de autorización. Esta solicitud incluye información como el alcance de los permisos necesarios y la URL de redirección de la aplicación.

Consentimiento del Usuario

El servidor de autorización muestra una página al usuario para que otorgue o niegue su consentimiento a la aplicación.

El usuario autentica su identidad en el servidor de autorización (por ejemplo, inicia sesión en su cuenta de Google) y decide si otorgar acceso a la aplicación.

Solicitud de Token de Acceso

Si el usuario otorga su consentimiento, el servidor de autorización genera un código de autorización y lo envía a la aplicación a través de la URL de redirección.

Intercambio de Código de Autorización por Token de Acceso

La aplicación recibe el código de autorización y, junto con su identificador secreto, hace una solicitud al servidor de autorización para intercambiar el código por un token de acceso.

Recepción del Token de Acceso

El servidor de autorización valida la solicitud de intercambio y, si es válida, emite un token de acceso a la aplicación.

Acceso a los Recursos

La aplicación utiliza el token de acceso para realizar solicitudes a un servidor de recursos en nombre del usuario.

El servidor de recursos valida el token de acceso y, si es válido, proporciona acceso a los recursos solicitados.

Renovación y Expiración del Token

Los tokens de acceso tienen un período de validez limitado. Para mantener el acceso a los recursos, la aplicación puede renovar el token si es necesario.

Revocación y Anulación

El usuario o la aplicación pueden revocar el acceso en cualquier momento, lo que invalida el token de acceso.

PROPIEDADES QUE INDICA EL PROVEEDOR OAUTH 2.0

Las propiedades pueden variar de un proveedor OAuth 2.0 a otro y la implementación específica puede depender de las políticas y características de seguridad del proveedor en cuestión.

- **Client ID (ID de Cliente):**

Es un identificador único para la aplicación cliente registrada en el proveedor de OAuth 2.0. Se utiliza para identificar la aplicación en las solicitudes de autorización.

- **Client Secret (Secreto de Cliente):**

Se refiere a un secreto compartido entre la aplicación cliente y el proveedor OAuth 2.0 que se utiliza para autenticar la aplicación.

- **Endpoint de Autorización:**

Es la URL a la que se redirige al usuario para que otorgue su autorización a la aplicación cliente. Aquí es donde se inicia el flujo de autorización.

- **Endpoint de Token:**

La URL a la que la aplicación cliente envía una solicitud de token de acceso después de que se haya otorgado la autorización. El proveedor devuelve el token de acceso en esta URL.

- **Alcance (Scope):**

El alcance especifica qué recursos o acciones el cliente tiene permiso para acceder en nombre del usuario. Los alcances pueden variar y son definidos por el proveedor.

- **Duración del Token (Token Expiry):**

El proveedor puede especificar cuánto tiempo es válido un token de acceso antes de que expire. Esto se conoce como el tiempo de vida del token.

- **URL de Usuario:**

La URL a la que la aplicación cliente puede acceder para obtener información adicional sobre el usuario autenticado.

- **Información sobre el Usuario:**

El proveedor puede proporcionar información sobre el usuario autenticado, como su nombre, dirección de correo electrónico, identificación única, etc.

- **Actualizaciones y Revocación de Tokens:**

Los proveedores OAuth 2.0 pueden proporcionar mecanismos para actualizar o revocar los tokens de acceso, lo que permite al usuario o al proveedor revocar los permisos otorgados a la aplicación cliente en cualquier momento.

- **Información de error:**

En caso de errores durante el proceso de autorización o autenticación, el proveedor puede devolver información detallada sobre el error, como mensajes de error y códigos de estado.

- **Políticas de Privacidad y Términos de Uso:**

El proveedor puede proporcionar enlaces a sus políticas de privacidad y términos de uso, que deben ser aceptados por el usuario durante el proceso de autorización.

Design Thinking

Componente de inicio de sesión con OAuth 2.0

Empatizar:

Se realizó una exhaustiva investigación, logrando empatizar con los usuarios y administradores. Se realizaron entrevistas con los usuarios finales y los administradores del sistema para comprender sus necesidades, desafíos y expectativas en relación con la autenticación. Se logró identificar una problemática. La cual, a los usuarios se les complicaba crear una cuenta de Creavi por la cantidad de información que el formulario pide. Por otra parte, los administradores no cuentan con una herramienta que les permite agregar nuevas maneras de iniciar sesión.

Definir:

Definición del Problema: Los usuarios encuentran complicado crear una cuenta de Creavi debido a la cantidad de información que el formulario solicita. Además, los administradores carecen de una herramienta que les permite agregar nuevas formas de inicio de sesión.

Nuestro componente fortalecerá la seguridad de la autenticación y autorización de usuarios para proteger sus datos y cuentas. Implementar OAuth 2.0 siguiendo las mejores prácticas de seguridad, incluyendo el uso de tokens seguros. Proteger contra ataques comunes, como suplantación de identidad y ataques de redirección abierta.

Facilidad en su uso proporcionando a los usuarios una experiencia de autenticación sencilla y sin trabas. Ofrecer opciones de autenticación flexibles, como autenticación social o de un solo clic. Facilitar la integración del componente de autenticación con aplicaciones y sistemas ya existentes.

Idear:

Se realizaron sesiones de lluvia de ideas. Las cuales, brindaron soluciones innovadoras para crear y diseñar el componente. Posteriormente se diseñó un prototipo con las soluciones más prometedoras y fueron aprobadas por los usuarios.

Como resultado se obtuvo un prototipo. El cual, permitirá solucionar la problemática. Este prototipo le va a permitir a los administradores poder agregar y quitar botones de inicio de sesión con solo llenar 5 recuadros. Los cuáles, serán las propiedades necesarias para agregar el un nuevo inicio de sesión.

Validar

Se hicieron pruebas de usabilidad y seguridad con usuarios reales. Se observaron las diferentes interacciones con el prototipo, logrando así recopilar información para poder ir ajustando el prototipo. Los ajustes fueron mínimos como la ubicación del menú de inicio de sesión.

Implementar

El componente se desarrollará teniendo en cuenta lo anteriormente expuesto, manejando los mejores estándares de seguridad de OAuth 2.0 implementando con personas reales las cuales darán su experiencia de usuario.

Introducción

En el entorno actual de la tecnología, la autenticación y autorización seguras son fundamentales para proteger los datos y garantizar que las aplicaciones y servicios en línea sean accesibles sólo para las partes autorizadas. OAuth 2.0, un protocolo de autorización ampliamente adoptado, proporciona una solución robusta para permitir que las aplicaciones accedan a recursos en nombre de los usuarios sin comprometer sus credenciales.

Propósito del Documento

En el contexto actual de la tecnología, la seguridad de la autenticación y la autorización es de suma importancia para salvaguardar los datos y asegurar que las aplicaciones y servicios en línea solo están al alcance de quienes tienen permiso. El protocolo ampliamente adoptado de OAuth 2.0 ha demostrado ser una solución sólida que permite a las aplicaciones acceder a recursos en nombre de los usuarios sin poner en riesgo sus credenciales.

Este documento tiene como objetivo proporcionar una descripción minuciosa de la implementación de un componente de autenticación basado en OAuth 2.0. Su propósito principal es servir como guía detallada para desarrolladores y administradores, ayudándoles a comprender los aspectos técnicos relacionados con la configuración, integración y uso de este componente en aplicaciones y sistemas. La documentación abarca información sobre la configuración de clientes OAuth, los flujos de autorización, la gestión de tokens de acceso, medidas de seguridad, buenas prácticas y ejemplos de código, todo ello con el fin de facilitar la correcta implementación del componente de autenticación OAuth 2.0.

En el trasfondo de esta documentación, se exploran los desafíos actuales que enfrentan los usuarios al intentar acceder a diversas plataformas en línea, y se ilustra cómo nuestra solución aborda eficazmente estos desafíos. Además, se hace hincapié en la importancia de proporcionar a los usuarios opciones de inicio de sesión que se ajusten a sus preferencias y hábitos, contribuyendo así a una experiencia más agradable y fluida.

El componente que estamos a punto de desarrollar marca un hito significativo en la autenticación en línea, reflejando nuestro compromiso con la mejora continua de la experiencia del usuario en nuestra plataforma. La implementación de este componente, basado en el protocolo OAuth 2.0, es un paso crucial para garantizar la seguridad de las aplicaciones y sistemas que utilizan esta tecnología.

Etapa 1: Diseño de la Aplicación y Análisis de Requisitos

En esta etapa, el propósito de la documentación es proporcionar una guía detallada para los desarrolladores y administradores que están en la fase de diseño de la aplicación y el análisis de requisitos. Esto se logra al resaltar la importancia de la autenticación y autorización seguras en el entorno tecnológico actual. Además, se identifican los desafíos comunes que enfrentan los usuarios al acceder a aplicaciones en línea y se introduce la solución propuesta. Sienta las bases para la comprensión del problema y la importancia de la solución. La autenticación OAuth 2.0 se integrará en la aplicación, qué aspectos técnicos deben considerarse y cómo se configurarán los clientes OAuth para garantizar la seguridad. Esta etapa se enfoca en establecer una base sólida para la implementación segura de la autenticación, asegurando que se cumplan todos los requisitos previos antes de pasar a la siguiente fase.

Etapa 2: Persistencia de Datos con Backend – Servidor

En esta etapa, se guiará a través de la implementación del componente de autenticación en el backend del servidor. Se abordan aspectos específicos, como la configuración de clientes OAuth, flujos de autorización, tokens de acceso, medidas de seguridad y mejores prácticas. El objetivo es proporcionar a los desarrolladores y administradores los conocimientos y recursos necesarios para integrar y utilizar el componente de autenticación de manera efectiva en aplicaciones y sistemas. Se centra en aspectos técnicos y prácticos de programar. Proporciona información sobre cómo se deben gestionar los flujos de autorización, tokens de acceso y las medidas de seguridad necesarias para proteger los datos. Garantizar que el servidor tenga la capacidad de autenticar de manera segura a los usuarios y proteger sus credenciales. Se centra en la implementación técnica del componente en el backend, asegurando su robustez y seguridad.

Etapa 3: Desarrollo Frontend – Cliente

En la última etapa, orientar en la integración del componente de autenticación OAuth 2.0 en el frontend de la aplicación o cliente. Se aborda la importancia de ofrecer opciones de inicio de sesión que se adapten a las preferencias de los usuarios, lo que contribuirá a una experiencia más agradable y fluida. Proporciona ejemplos de código, mejores prácticas y directrices para garantizar que los usuarios puedan iniciar sesión de manera segura y eficaz. Además, se aborda la experiencia del usuario, ofreciendo opciones de inicio de sesión que se adapten a sus preferencias y hábitos. Se enfoca en mejorar la experiencia del usuario al mismo tiempo que se garantiza la seguridad en la autenticación en línea.

Alcance del Proyecto

El desarrollo del Componente de Inicio de Sesión de Creavi con OAuth 2.0 tiene como objetivo principal permitir a los usuarios de la plataforma Creavi realizar el proceso de inicio de sesión utilizando múltiples proveedores de autenticación directa que operan bajo el estándar del modelo tradicional de autenticación cliente-servidor OAuth 2.0 como por ejemplo Facebook, Instagram, Twitter, etc. Además, se debe permitir a los futuros desarrolladores con el rol de administrador de Creavi la capacidad de agregar uno o más proveedores adicionales que utilicen el mismo modelo de autenticación OAuth 2.0. Este componente se diseñará teniendo en cuenta que su implementación no se limitará exclusivamente a Creavi, sino que será aplicable a cualquier plataforma que no utilice un modelo de autenticación cliente-servidor.

Objetivos:

- ❖ Facilitar a los usuarios de Creavi la posibilidad de autenticarse utilizando diversos proveedores de autenticación directa que sigan el estándar OAuth 2.0.
- ❖ Permitir a los administradores de Creavi la incorporación de nuevos proveedores de autenticación OAuth 2.0.

Los principales entregables incluyen:

- ❖ Componente de Inicio de Sesión que permite a los usuarios autenticarse con éxito utilizando múltiples proveedores de autenticación basados en el estándar OAuth 2.0.
- ❖ Módulo de administración que permita a los administradores gestionar los proveedores de autenticación OAuth 2.0 disponibles, incluyendo su habilitación o deshabilitación
- ❖ Documentación detallada que describa cómo configurar, administrar y mantener los proveedores de autenticación.

Requisitos y Recursos:

- ❖ Acceso a los servidores y sistemas donde se implementará el componente.
- ❖ Tiempo para el desarrollo y pruebas del proyecto.

Se identificarán los problemas que podrían surgir durante el proyecto y se desarrollará un plan para solucionarlos.

Funcionalidades Futuras:

❖ "Olvidar Contraseña":

Implementar una funcionalidad que permita a los usuarios restablecer su contraseña en caso de que la olviden o necesite cambiarla por razones de seguridad. Los usuarios podrán acceder a esta funcionalidad desde la página de inicio de sesión. Deberá incluir un proceso de verificación de identidad para asegurar la seguridad de la solicitud de restablecimiento de contraseña. Una vez verificada la identidad, los usuarios recibirán un enlace de restablecimiento de contraseña en su correo electrónico registrado.

Definiciones y Acrónimos

OAuth 2.0: Es un estándar abierto para la autorización de APIs, que nos permite compartir información entre sitios sin tener que compartir la identidad.

API: Interfaz de Programación de Aplicaciones (Application Programming Interface).

DBMS: Sistema de Gestión de Bases de Datos (Database Management System).

SQL: Lenguaje de Consulta Estructurada (Structured Query Language).

HTTP: Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol).

REST: Transferencia de Estado Representacional (Representational State Transfer).

JSON: Notación de Objetos de JavaScript (JavaScript Object Notation).

JWT: Token de Web JSON (JSON Web Token).

CRUD: Crear, Leer, Actualizar y Borrar (Create, Read, Update, Delete).

ORM: Mapeo Objeto-Relacional (Object-Relational Mapping).

MVC: Modelo-Vista-Controlador (Model-View-Controller).

API RESTful: API que sigue los principios de REST.

CI/CD: Integración Continua / Entrega Continua (Continuous Integration / Continuous Delivery).

SaaS: Software como Servicio (Software as a Service).

SSL/TLS: Capa de sockets seguros/Seguridad de la Capa de Transporte (Secure Sockets Layer/Transport Layer Security).

HTML: Lenguaje de Marcado de Hipertexto (Hypertext Markup Language).

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets).

JS: JavaScript.

DOM: Modelo de Objeto del Documento (Document Object Model).

UI: Interfaz de Usuario (User Interface).

UX: Experiencia del Usuario (User Experience).

SPA: Aplicación de Página Única (Single Page Application).

AJAX: Asíncrono JavaScript y XML (Asynchronous JavaScript and XML).

CMS: Sistema de Gestión de Contenido (Content Management System).

CDN: Red de Distribución de Contenido (Content Delivery Network).

SEO: Optimización de Motores de Búsqueda (Search Engine Optimization).

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment).

CLI: Interfaz de Línea de Comandos (Command Line Interface).

PWA: Aplicación Web Progresiva (Progressive Web App).

Descripción General

Objetivos del Sistema

El objetivo del sistema es proporcionar a los administradores gestionar los inicios de sesión de los usuarios de manera eficiente y efectiva dentro de CREA VI que permita a los usuarios poder iniciar sesión garantizando que dichos inicios de sesión sean seguros y protegidos contra amenazas comunes, como la suplantación de identidad. Por otro lado, este componente acortará el tiempo en el que un usuario se demora llenando formularios para realizar su registro, posteriormente, su inicio de sesión.

Funcionalidades Generales

- ❖ **Iniciar sesión con OAuth 2.0:** Puede gestionar sesiones de usuario, permitiendo a los usuarios mantenerse autenticados durante un período de tiempo determinado.
- ❖ **Editar perfil OAuth 2.0:** Los usuarios pueden personalizar sus perfiles, agregar información adicional y configurar preferencias.
- ❖ **Agregar nuevo proveedor OAuth 2.0:** Permite a los usuarios administradores agregar un nuevo proveedor de inicio de sesión OAuth 2.0.
- ❖ **Editar proveedor OAuth 2.0:** Permite a los usuarios administradores editar los proveedores de inicio de sesión OAuth 2.0.
- ❖ **Eliminar Proveedor OAuth 2.0:** El administrador podrá eliminar los proveedores y editar los requerimientos que componen a cualquier proveedor.
- ❖ **Listar proveedores OAuth 2.0:** El administrador podrá ver la lista de los proveedores.

Usuarios del Sistema

Funcionalidades	Administrador	Sistema	Usuario
Iniciar sesión con OAuth 2.0			
Editar perfil OAuth 2.0			
Agregar nuevo proveedor OAuth 2.0			
Editar proveedor OAuth 2.0			
Eliminar Proveedor OAuth 2.0			
Listar proveedores OAuth 2.0			

Restricciones

Bloqueo temporal de la cuenta: Después de un número específico de intentos de inicio de sesión fallidos, la cuenta de usuario se bloquea temporalmente. Durante este período de bloqueo, el usuario no podrá intentar iniciar sesión. Este enfoque disuade a los atacantes, ya que sus intentos serán infructuosos después de un cierto número de intentos fallidos.

Resquicitos Funcionales

Iniciar sesión con OAuth 2.0:

- El sistema se encargará de gestionar todo el inicio de sesión.

Editar perfil OAuth 2.0:

- Se crearán los perfiles una vez los usuarios inicien sesión y tengan acceso a la plataforma.

Agregar nuevo proveedor OAuth 2.0:

- El administrador podrá anexar nuevos proveedores de inicio de sesión que trabajen con OAuth 2.0

Editar proveedor OAuth 2.0:

- El administrador podrá agregar y editar los requerimientos que componen a cualquier proveedor.

Eliminar Proveedor OAuth 2.0:

- El administrador podrá eliminar los proveedores y editar los requerimientos que componen a cualquier proveedor.

Listar proveedores OAuth 2.0:

- El administrador podrá ver la lista de los proveedores.

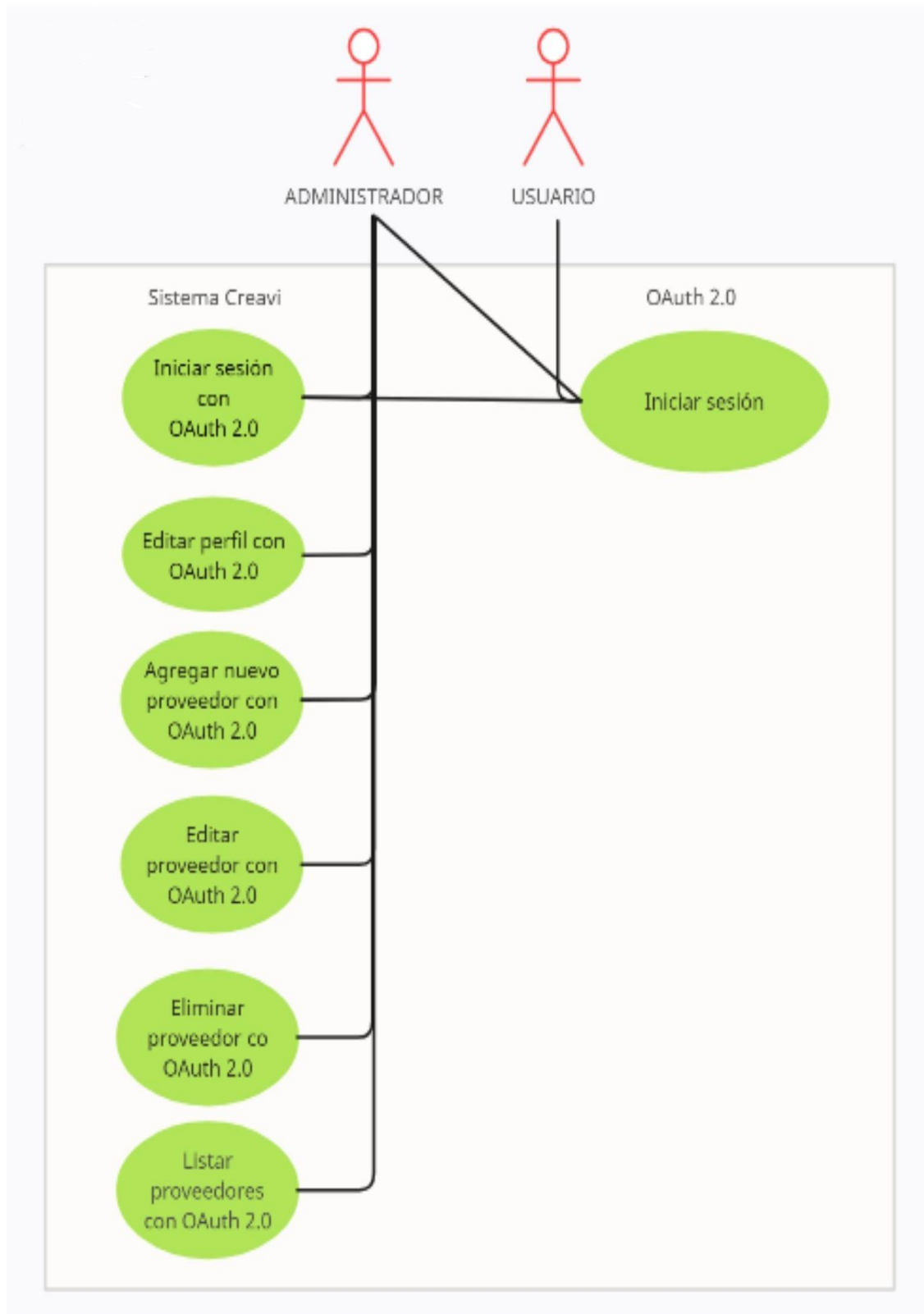
Requisitos no Funcionales

Requisitos de Desempeño

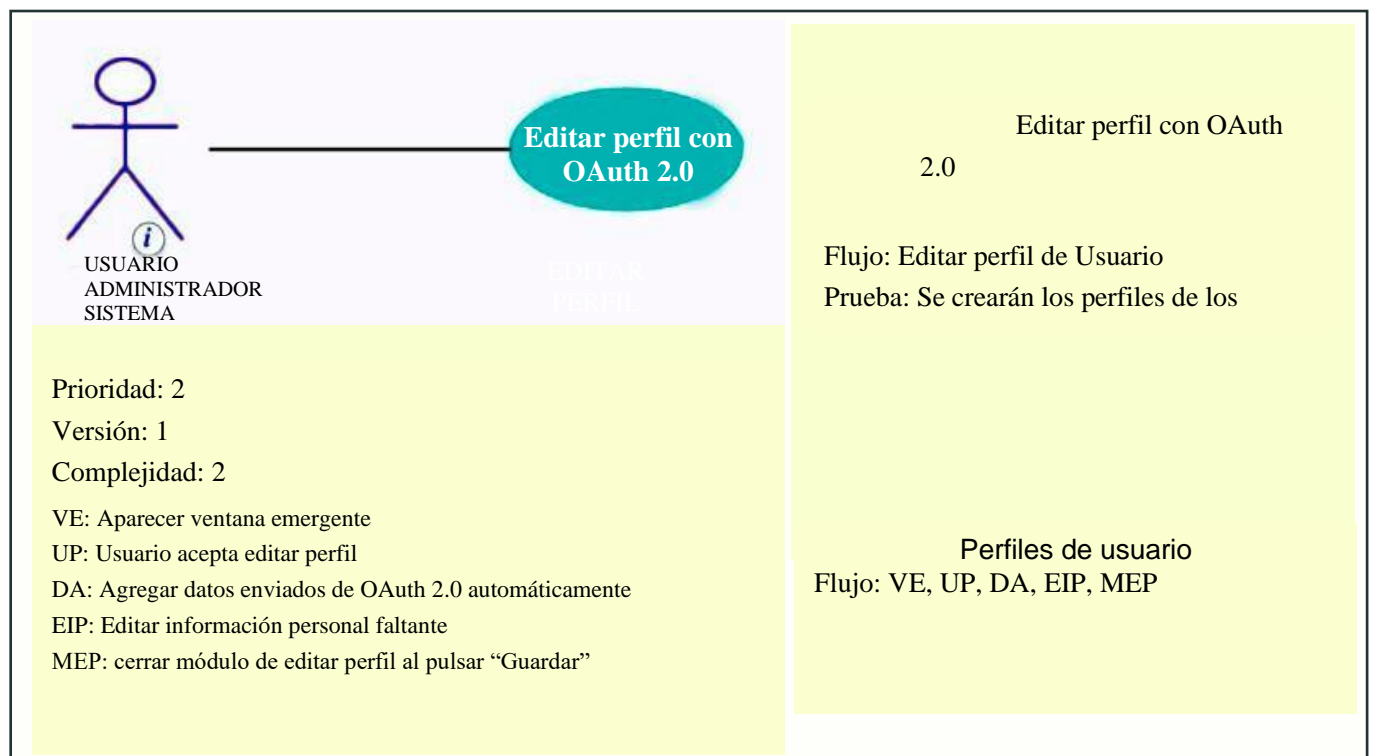
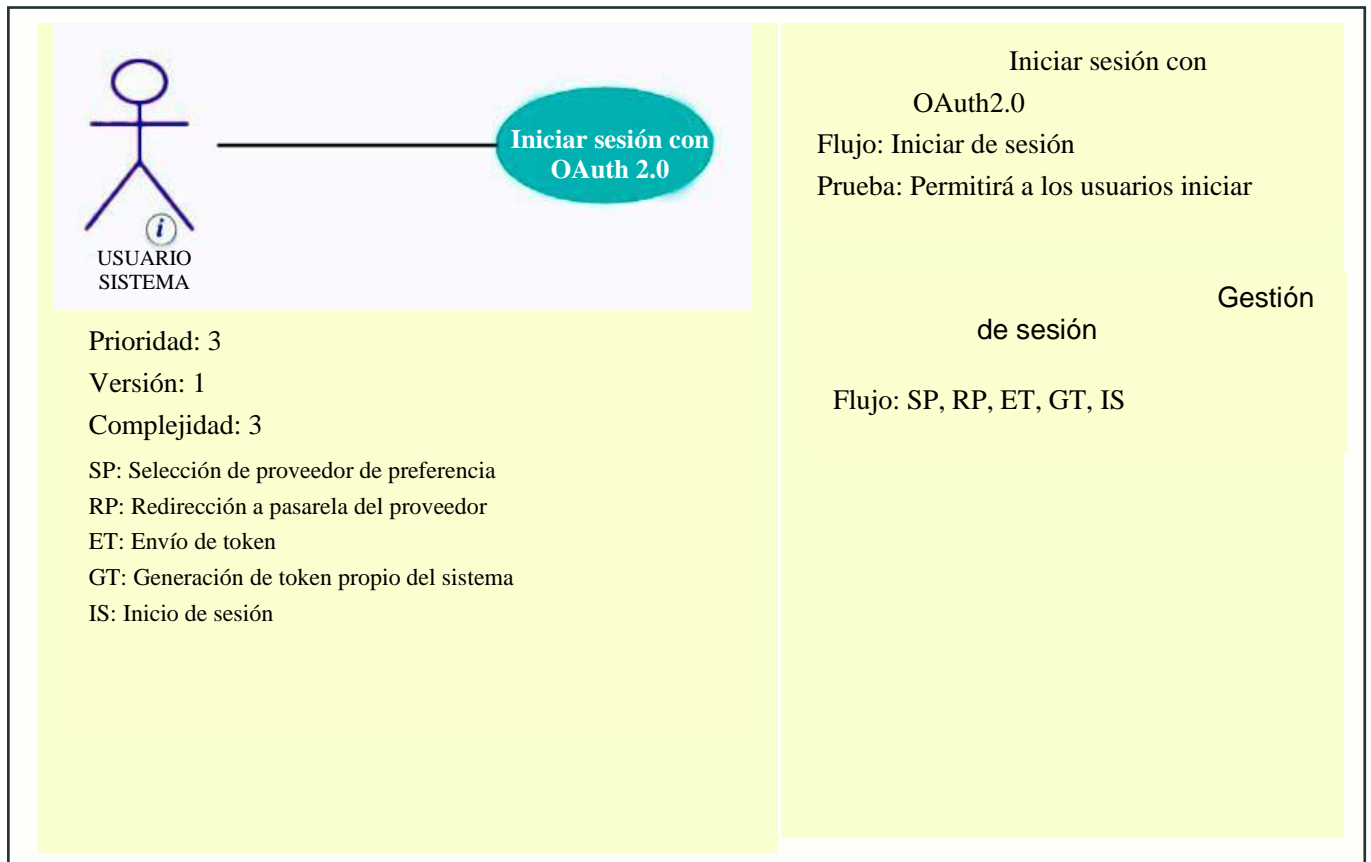
- **Tiempo de respuesta:** El sistema debe tomar la emisión de tokens de acceso o la autorización de recursos a través de OAuth 2.0. Por ejemplo, establecer un límite de 1 segundo para la respuesta del servidor.
- **Disponibilidad:** Debe contar con disponibilidad requerida para el sistema que implementa OAuth 2.0. Por ejemplo, podrías exigir que el sistema esté disponible el 99.9% del tiempo.
- **Seguridad y rendimiento:** Asegurarse que la implementación de OAuth 2.0 no degrade la seguridad del sistema en situaciones de alto rendimiento. Esto incluye protección contra ataques de fuerza bruta en las solicitudes de tokens de acceso.

Casos de Uso

Diagrama de caso de usos



Diagramas de Flujo de Casos de Uso





ADMINISTRADOR
SISTEMA

Agregar nuevo
proveedor
con OAuth 2.0

Prioridad: 5

Versión: 1

Complejidad: 5

O: Opciones

AP: Agregar nuevos proveedores

MP: Desplegar módulo de requerimientos para proveedores

GP: Guardar proveedor

RL: Redirigir a “listar proveedores”

Agregar nuevo proveedor
con OAuth 2.0

Flujo: Agregar nuevo proveedor OAuth 2.0

Prueba: el administrador podrá agregar
nuevos inicios de sesión con OAuth 2.0



ADMINISTRADOR
SISTEMA

Editar proveedor
con OAuth 2.0

Prioridad: 4

Versión: 1

Complejidad: 4

O: Opciones

EP: Editar proveedor

CR: Cambiar requerimientos necesarios para el funcionamiento del
proveedor

GC: Guardar cambios

Editar proveedores

Flujo: Editar proveedores

Prueba: El administrador podrá editar los
parametros de los proveedores

Editar proveedores
de OAuth 2.0

Flujo: O, EP, CR, GC



ADMINISTRADOR
SISTEMA

Eliminar proveedor
con OAuth 2.0

Prioridad: 4

Versión: 1

Complejidad: 4

A: opciones.

DP: Eliminar proveedor.

SPE: Seleccionar proveedor que se desee eliminar.

OE: Pulsar la opción Eliminar.

EP: El proveedor se elimina de la lista y del inicio de sesión.

Eliminar proveedor con
OAuth 2.0

Flujo: Eliminar proveedor OAuth 2.0.

Prueba: Permitirá a un administrador.

Eliminar proveedor
de OAuth 2.0

Flujo: A, DP, SPE, OE, EP



ADMINISTRADOR
SISTEMA

Listar Proveedores
con OAuth 2.0

Prioridad: 4

Versión: 1

Complejidad: 4

BD: Dirigir a base de datos

SPE: Seleccionar proveedores existentes

RD: Retornar los datos

MP: Mostrar en pantalla de inicio de sesión

Listar proveedores con
OAuth 2.0

Flujo: Listar proveedores

Prueba: El administrador podrá listar todo

Listar proveedores de
OAuth 2.0

Flujo: BD, SPE, RD, MP

CASO No. 1 Iniciar sesión con OAuth 2.0

ID:	CU-1	
Nombre	Iniciar sesión con OAuth2.0	
Actores	Sistema y Usuario	
Objetivo	Este caso debe permitir al usuario poder iniciar sesión con OAuth 2.0	
Urgencia	2	
Esfuerzo	2	
Pre-condiciones	<ul style="list-style-type: none"> - El usuario debe haberse logueado con algún proveedor OAuth 2.0 del sistema, necesariamente el mismo que seleccione para iniciar sesión en la plataforma. 	
Flujo Normal	Sistema	Usuario
		Inicio de sesión usando un proveedor OAuth 2.0 disponible para usar en la plataforma.
	Entrega de token de acceso a la plataforma.	
	Recepción, Verificación y generación de tokens y cookies de sesión propio de la plataforma	
	Inicio de sesión satisfactoriamente	
Flujo alternativo 1		Registrarse directamente en Creavi
Post-condiciones	El usuario inició sesión correctamente con el proveedor selección.	
Excepciones	El usuario no tiene cuenta en ninguna de las otras plataformas que permiten iniciar sesión.	

CASO No. 2 Editar perfil OAuth 2.0

ID:	CU-2	
Nombre	Editar perfil	
Actores	Sistema y Usuario	
Objetivo	Este caso debe crear los perfiles de los usuarios una vez inician sesión con OAuth 2.0	
Urgencia	2	
Esfuerzo	2	
Pre-condiciones	<ul style="list-style-type: none"> - El sistema debe haber verificado de forma correcta la información del usuario en el sistema. - Que exista un logueo previo en alguno de los proveedores de OAuth 2.0 disponibles, necesariamente el que el usuario eligió para iniciar sesión en la plataforma. 	
Flujo Normal	Sistema	Usuario
	Preparación del perfil en la plataforma	
	Despliega las opciones de guardar perfil o editar perfil	
		El usuario da clic en Editar perfil
	Despliega los campos para editar la información del usuario.	
		Rellena y edita los campos.
		Guardar.
	Guardar cambios.	
Flujo alternativo 1	Preparación del perfil en la plataforma	
	Despliega las opciones de guardar perfil o editar perfil	

		Clic en guardar perfil
	Guarda la información existente	
Post-condiciones	Se creó el perfil correctamente.	

CASO No. 3 Agregar nuevo proveedor OAuth 2.0

ID:	CU-3	
Nombre	Agregar nuevo proveedor OAuth 2.0	
Actores	Administrador y Sistema	
Objetivo	En este caso el administrador podrá agregar nuevos inicios de sesión OAuth 2.0.	
Urgencia	5	
Esfuerzo	5	
Pre-condiciones	<ul style="list-style-type: none"> - El administrador ha iniciado sesión en el sistema. - Es importante asegurarse de que el proveedor OAuth 2.0, que se está agregando no sea un duplicado de uno que ya esté registrado en el sistema. 	
Flujo Normal	Administrador	Sistema
	El administrador inicia sesión en el sistema.	
	Selecciona administración de proveedores OAuth 2.0	
		Despliega opciones de administración de proveedores OAuth 2.0
	Selecciona la opción Agregar nuevo proveedor OAuth 2.0	
		despliega los campos necesarios para proveedor OAuth 2.0
	Guarda los campos del nuevo proveedor OAuth 2.0	

	Confirma la adición del proveedor OAuth 2.0 en el inicio de sesión.	
		Registra al nuevo proveedor OAuth 2.0
Flujo alternativo 1	Cancela la opción de nuevo proveedor OAuth 2.0	
		No afectará ni se guardará la información.
Post-condiciones		El nuevo proveedor OAuth 2.0 se agrega correctamente al apartado de inicio de sesión.
Excepciones	En caso de fallo del sistema durante el proceso, se muestra un mensaje de error y se notifica al administrador.	

CASO No. 4 Editar proveedores OAuth 2.0

ID:	CU-4	
Nombre	Editar proveedores OAuth 2.0	
Actores	Administrador y Sistema	
Objetivo	En este caso el administrador podrá editar los proveedores OAuth 2.0	
Urgencia	4	
Esfuerzo	4	
Pre-condiciones	<ul style="list-style-type: none"> - El administrador debió haber iniciado sesión. - Existe al menos un proveedor OAuth 2.0 registrado en el sistema. 	
Flujo Normal	Administrador	Sistema
	El administrador inicia sesión en el sistema.	
	Navega a la sección de administración de proveedores OAuth 2.0	
		Se despliega la sección de administración de proveedores OAuth 2.0
	Selecciona el proveedor OAuth 2.0 que desea editar de la lista existente.	
		Se despliegan los campos del proveedor OAuth 2.0
	Modifica los campos necesarios para el funcionamiento del proveedor OAuth 2.0	
	Pulsar para guardar	
		Confirma la actualización

 Flujo alternativo 1	Si el administrador decide cancelar la operación en cualquier punto.	
		No afectará ni se guardará la información
Post-condiciones		El proveedor OAuth 2.0 se editó correctamente en la plataforma.
		Debe aparecer en el inicio de sesión.
Excepciones	Si hay un fallo en la conexión, se muestra un mensaje de error y se notifica al administrador..	

CASO No. 5 Eliminar Proveedor OAuth 2.0

ID:	CU-5	
Nombre	Eliminar proveedor OAuth 2.0	
Actores	Administrador y Sistema	
Objetivo	En este caso el administrador podrá eliminar proveedores de OAuth 2.0 que no se quieran mostrar en el inicio de sesión.	
Urgencia	3	
Esfuerzo	3	
Pre-condiciones	<ul style="list-style-type: none"> - El administrador ha iniciado sesión en el sistema. - Existe al menos un proveedor OAuth 2.0 registrado en el sistema. 	
Flujo Normal	Administrador	Sistema
	El administrador inicia sesión en el sistema.	
	Navega a la sección de administración de proveedores OAuth 2.0.	
		Despliega la administración de proveedores OAuth 2.0
	Selecciona el icono Eliminar que aparece al lado del nombre de cada proveedor OAuth 2.0	
		Despliega una advertencia "Estás seguro de eliminar este proveedor, recuerda que no hay marcha atrás"
	Clic en la opción "Aceptar"	
		Elimina el proveedor OAuth 2.0 seleccionado del apartado inicio de sesión.

Flujo alternativo 1	Si el administrador decide cancelar la operación en cualquier momento.	
		No afectará ni se guardará la información.
Post-condiciones		El proveedor OAuth 2.0 desaparece del apartado de inicio de sesión.

CASO No. 6 Listar proveedores OAuth 2.0

ID:	CU-6	
Nombre	Listar proveedores OAuth 2.0	
Actores	Administrador y sistema	
Objetivo	En este caso el administrador podrá listar todos los proveedores OAuth 2.0 disponibles.	
Urgencia	3	
Esfuerzo	3	
Pre-condiciones	<ul style="list-style-type: none"> - El administrador debió haber iniciado sesión. - Deben existir al menos dos proveedores OAuth 2.0 agregados. 	
Flujo Normal	Administrador	Sistema
	El administrador inicia sesión en el sistema.	
	Navega en la sección de administración de proveedores OAuth 2.0	
		Se despliega la sección de administración de proveedores OAuth 2.0
		Se muestra un icono de un “ojo” en la parte superior derecha del título “Administración de proveedores OAuth 2.0”
	Clic en el icono “ojo”	
		Se muestran los proveedores OAuth 2.0 con bordes marcados y flotantes.
	Clic en regresar	

Propiedades de Requerimiento

Desde el análisis de los requisitos, funcionalidades y el proceso de design thinking, se ha desarrollado una matriz de prioridades para los requisitos. Esta matriz utiliza una escala con valores específicos para interpretar y asignar importancia a cada requisito.

La escala se utiliza como un marco de referencia para evaluar y clasificar la prioridad de cada requisito en función de su relevancia y contribución al éxito general del proyecto.

Eje de Urgencia:

- Obligatoria (5)
- Alta (4)
- Moderada (3)
- Menor (2)
- Baja (1)

Eje de Esfuerzo:

- Muy alto (5)
- Alto (4)
- Medio (3)
- Bajo (2)
- Muy bajo (1)

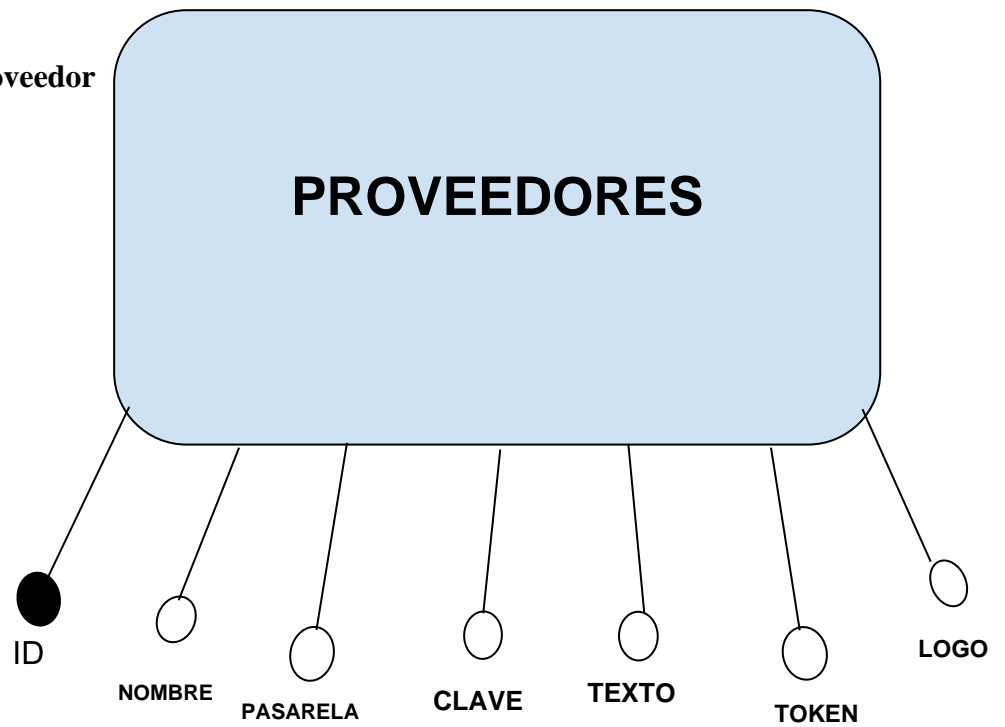
	Urgencia					
Esfuerzo		1- Baja	2- Menor	3- Moderada	4- Alta	5- Obligatoria
	5-Muy alto	5	10	15	20	25
						CU-3
	4-Alto	4	8	12	16	20
					CU-4	
	3-Medio	3	6	9	12	15
				CU-5 CU-6		
	2-Bajo	2	4	6	8	10
			CU-1 CU-2			
	1-Muy bajo	1	2	3	4	5

FUNCIONALIDADES Y DATOS

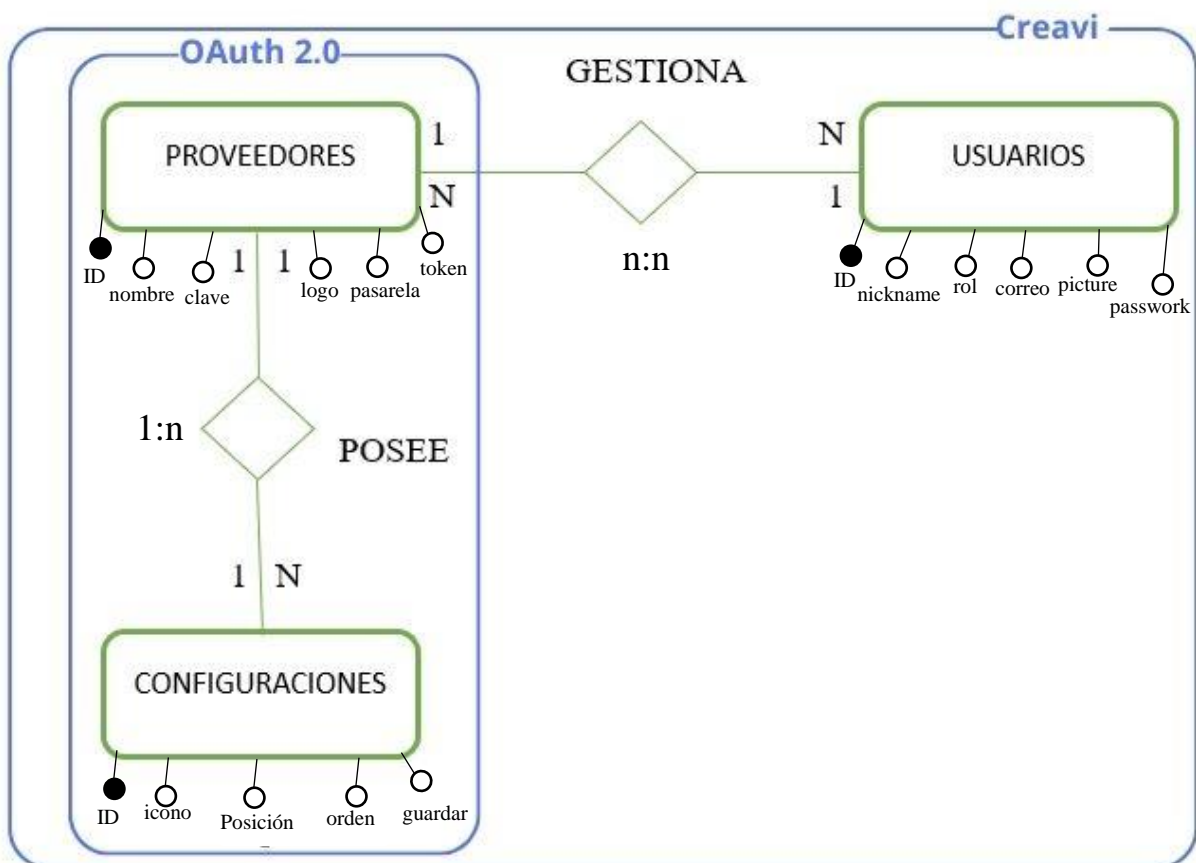
Iniciar sesión con OAuth 2.0	Proveedor	[Google, Facebook...]
	Token	[IKIL...]
	Pasarela	[www.googleOAuth2 Inicio.com]
	Información Adicional	"Soy Joven Soñador de Córdoba..."
Agregar nuevo proveedor OAuth 2.0	Nombre del Proveedor	Facebook
	Clave API	"*****"
	Secreto	"*****"
Eliminar Proveedor OAuth 2.0	ID del proveedor a eliminar:	"122438520"
Editar perfil OAuth 2.0	Logo	"HpYs. Png"
	Usuario	"Pepito Pérez"
	Correo Electrónico	"pp.oauth@correo.com"
	Contraseña	"spy#hdjs"
	Nombres	"Pepito Andres"
	Apellidos	"Perez Doria"
	Foto de Perfil	"7x47xJ.Jpg"

Modelando Componente de inicio de sesión con OAuth 2.0 Creavi

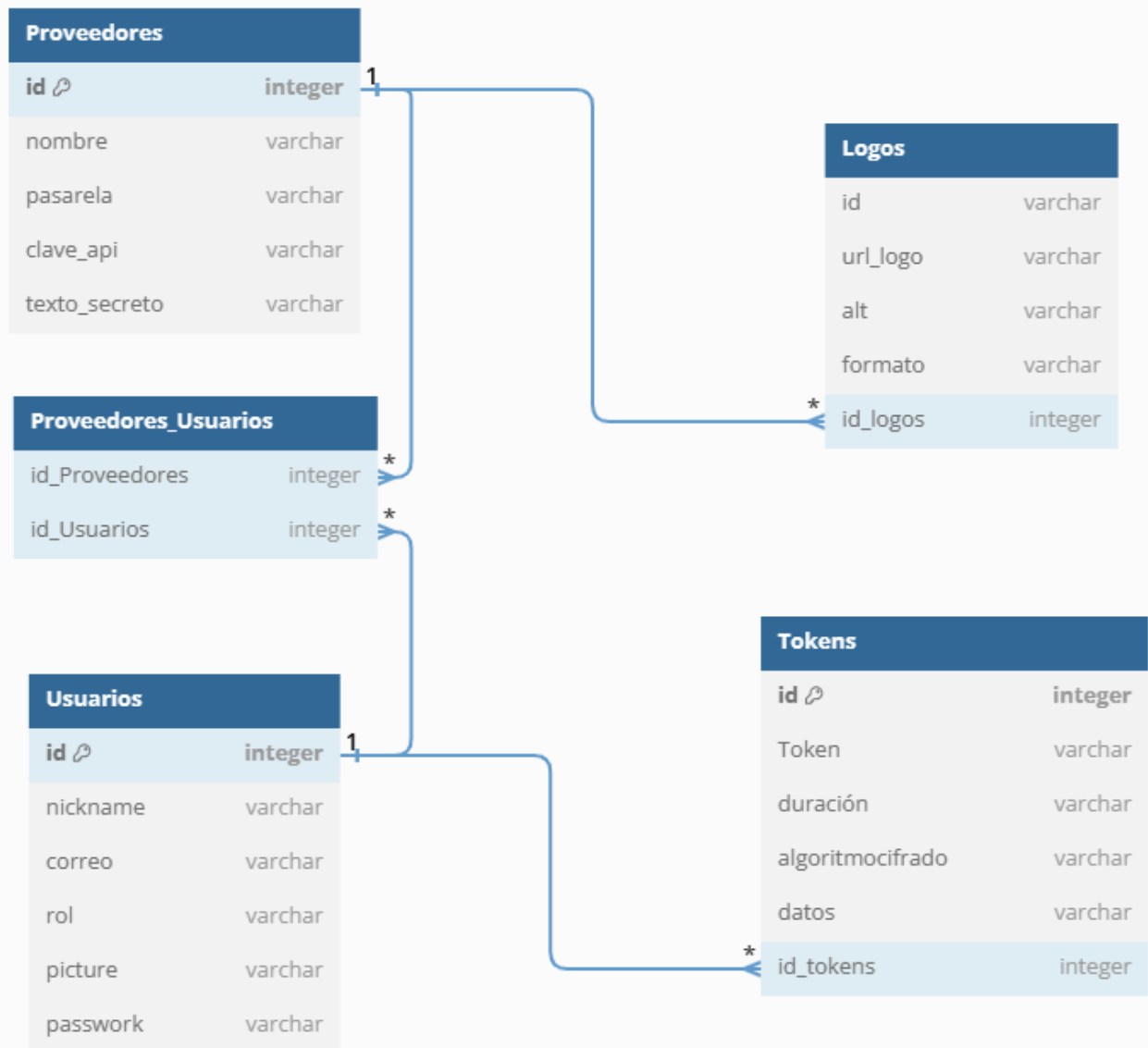
- Proveedor
- Token
- Pasarela
- Nombre del proveedor
- Clave API
- ID
- Logo
- Nombre



Cardinalidad



Modelo Relacional



Modelo no relacional

Proveedores: {

Id: object.id,
Nombre: string,
Pasarela: string,
Clave_api: string,
Texto_secreto: string,
Id_usuario: [object.id]
}

Tokens: {

Id: object.id,
Token: string,
Duración: string,
Algoritmo_cifrado: string,
Datos: string
}

Usuarios: {

Id: object.id,
Nickname: string,
Correo: string,
Rol: string
Picture: string,
Passwork: string,
Id_proveedores: [object.id]
}

Logos: {

id: object.id,
Url_logo: string,
Alt: string,
Formato: string
id_logos: object.id
}


COLECCIÓN LOGOS

CRUD LOGOS, POSTMAN





[https://github.com/Grupo-Investigacion-](https://github.com/Grupo-Investigacion-Bimadino/pads_oauth2/blob/Conection/documents/postman/Logos.CRUD_jabibmanzu.json)

[Bimadino/pads_oauth2/blob/Conection/documents/postman/Logos.CRUD_jabibmanzu.json](https://github.com/Grupo-Investigacion-Bimadino/pads_oauth2/blob/Conection/documents/postman/Logos.CRUD_jabibmanzu.json)

[pads_oauth2](#) / [documentos](#) / [cartero](#) / Logos.CRUD_jabibmanzu.json


 jabibmanz Logotipos_CRUD 1dd9a07 · hace 3 semanas [Historia](#)

Código Culpa 134 líneas (134 loc) · 2,4 KB





Crudo    

```
1 {
2   "información": {
3     "_cartero_id": "a4a0be00-d5cd-4af7-962d-1544246ce654",
4     "nombre": "Logotipos",
5     "esquema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
6     "_exportador_id": "34922052"
7   },
8   "artículo": [
9     {
10      "nombre": "Logotipos",
11      "pedido": {
12        "método": "CONSEGUIR",
13        "encabezamiento": [],
14        "URL": {
15          "crudo": "http://localhost:3000/logotipos",
16          "protocolo": "http",
17          "anfitrión": [
18            "servidor local"
19          ]
20        }
21      }
22    ]
23  }
```

[pads_oauth2/src/logos/Schemas/logos.ts](#) at main · [Grupo-Investigacion-Bimadino/pads_oauth2](#) ([github.com](#))

 jabibmanz esquemas 1f542f8 · ayer [Historia](#)

Código Culpa 25 líneas (17 loc) · 410 Bytes

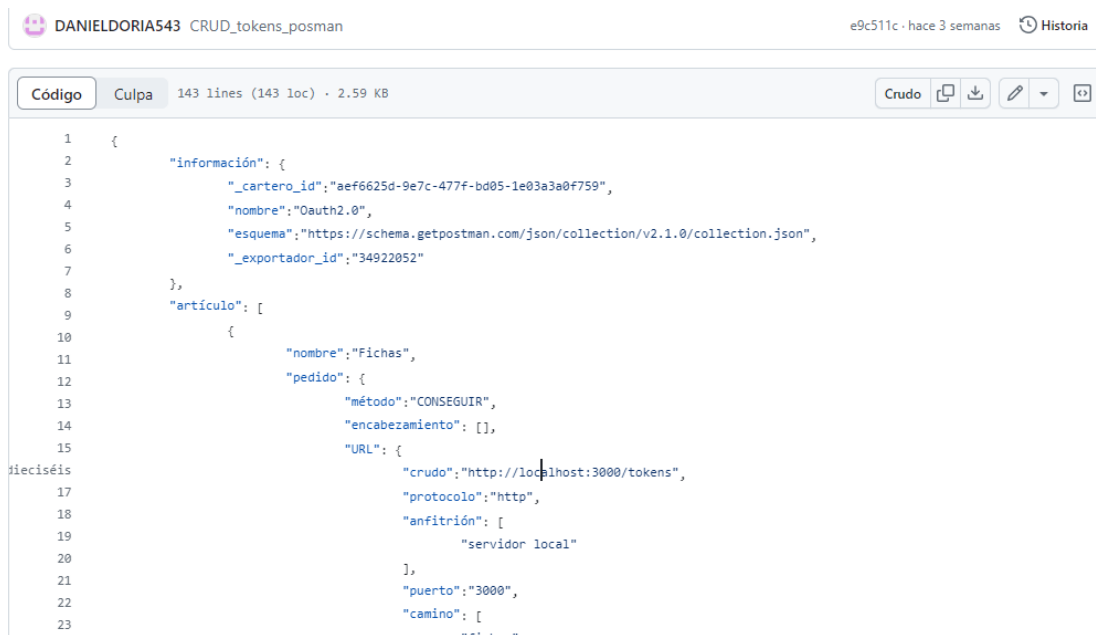
Crudo    

```
1  import { Apuntalar, Esquema, fábrica de esquemas } de '@nestjs/mangosta';
2  import { Documento } de 'mangosta';
3  import * como mangosta de 'mangosta';
4  import { marca de tiempo } de 'rxjs';
5
6  @Esquema({
7    marcas de tiempo: verdadero, })
8
9
10 export clase logotipos se extiende Documento {
11   @Apuntalar()
12   identificación: Cadena;
13
14   @Apuntalar()
15   logotipo_url: Cadena;
16
17   @Apuntalar()
18   formato: Cadena;
19
20   @Apuntalar()
21   id_logotipos: Cadena;
22
23 }
24
```

COLECCIÓN TOKENS

CRUD TOKENS, POSTMAN

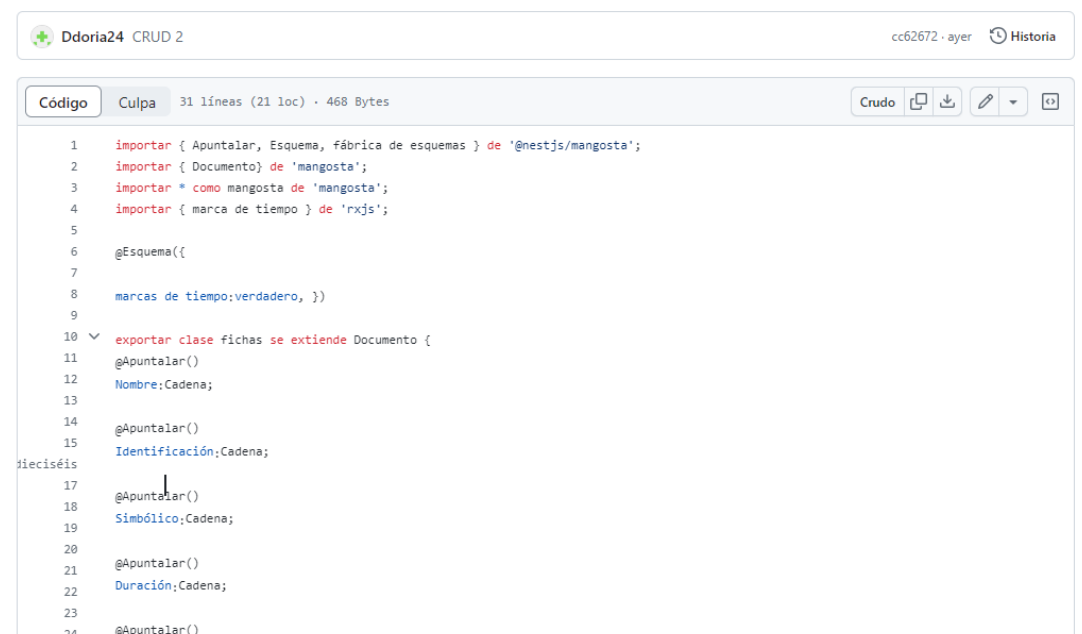
[pads_oauth2/documents/postman/Oauth2.0.CRUD_tokens_daniel_doria.json at Conection · Grupo-Investigacion-Bimadino/pads_oauth2 \(github.com\)](https://github.com/Grupo-Investigacion-Bimadino/pads_oauth2/blob/main/src/tokens/Schemas/tokens.ts)



The screenshot shows a Postman collection editor for a collection named "CRUD_tokens_posman" by user "DANIELDORIA543". The collection ID is "e9c511c" and it was last updated "hace 3 semanas". The editor displays a JSON schema for a token collection. The schema is as follows:

```
1  {
2    "información": {
3      "_cartero_id": "aef6625d-9e7c-477f-bd05-1e03a3a0f759",
4      "nombre": "Oauth2.0",
5      "esquema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
6      "_exportador_id": "34922052"
7    },
8    "artículo": [
9      {
10       "nombre": "Fichas",
11       "pedido": {
12         "método": "CONSEGUIR",
13         "encabezamiento": [],
14         "URL": {
15           "crudo": "http://localhost:3000/tokens",
16           "protocolo": "http",
17           "anfitrión": [
18             "servidor local"
19           ],
20           "puerto": "3000",
21           "camino": [
22             "tokens"
23           ]
24         }
25       }
26     ]
27   }
```

https://github.com/Grupo-Investigacion-Bimadino/pads_oauth2/blob/main/src/tokens/Schemas/tokens.ts

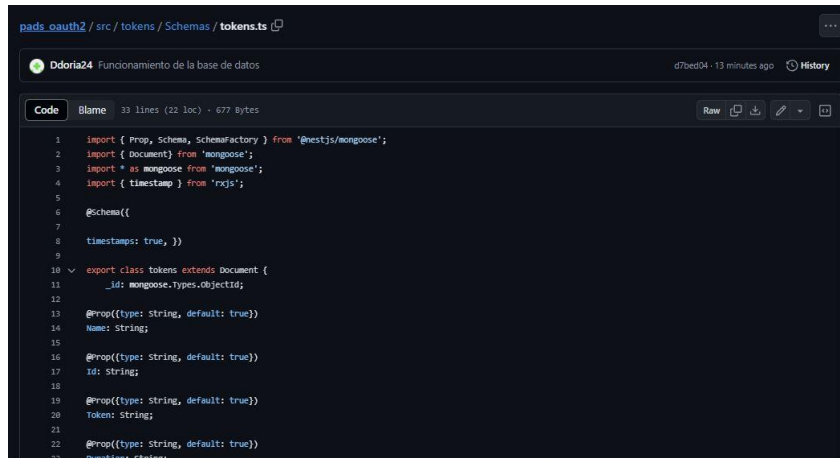


The screenshot shows a TypeScript file named "tokens.ts" by user "Ddoria24". The file ID is "cc62672" and it was last updated "ayer". The file contains the following code:

```
1  import { Apuntalar, Esquema, fábrica de esquemas } de '@nestjs/mangosta';
2  import { Documento } de 'mangosta';
3  import * como mangosta de 'mangosta';
4  import { marca de tiempo } de 'rxjs';
5
6  @Esquema({
7    marcas de tiempo: verdadero, })
8
9  exportar clase fichas se extiende Documento {
10   @Apuntalar()
11   Nombre: Cadena;
12
13   @Apuntalar()
14   Identificación: Cadena;
15
16   @Apuntalar()
17   Símbólico: Cadena;
18
19   @Apuntalar()
20   Duración: Cadena;
21
22   @Apuntalar()
23
24 }
```

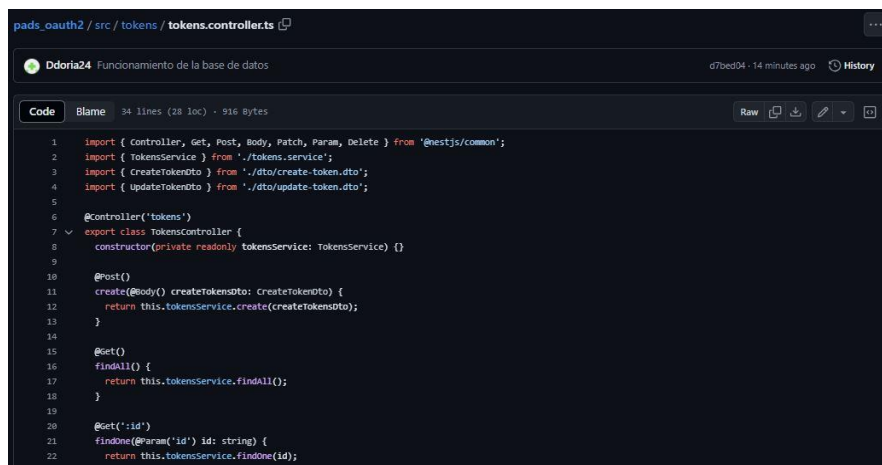
Conexión a la base de Datos Mongo Compass “Tokens” En funcionamiento

[pads_oauth2/src/tokens/Schemas/tokens.ts at main · Grupo-Investigacion-Bimadino/pads_oauth2 \(github.com\)](#)



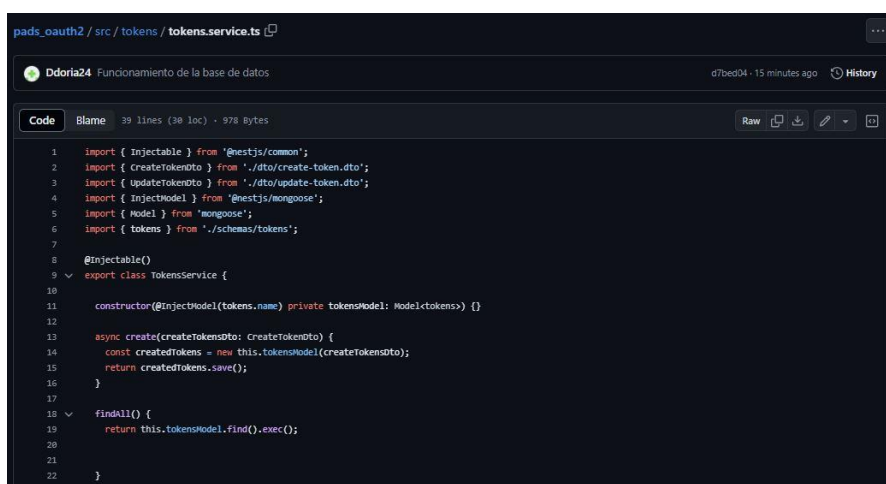
The screenshot shows a code editor with the file path `pads_oauth2 / src / tokens / Schemas / tokens.ts`. The code defines a Mongoose schema for tokens. It imports `prop`, `Schema`, and `SchemaFactory` from `@nestjs/mongoose`, `Document` from `'mongoose'`, `mongoose` from `'mongoose'`, and `timestamp` from `'rxjs'`. The schema is defined with `@Schema({ timestamps: true, })`. The `tokens` class extends `Document` and has the following properties: `_id` (type: `mongoose.Types.ObjectId`), `name` (type: `String`, default: `true`), `id` (type: `String`, default: `true`), `token` (type: `String`, default: `true`), and `duration` (type: `String`, default: `true`).

[pads_oauth2/src/tokens/tokens.controller.ts at main · Grupo-Investigacion-Bimadino/pads_oauth2 \(github.com\)](#)



The screenshot shows a code editor with the file path `pads_oauth2 / src / tokens / tokens.controller.ts`. The code defines a `TokensController` class. It imports `controller`, `get`, `post`, `body`, `patch`, `param`, and `delete` from `@nestjs/common`, `TokensService` from `./tokens.service`, `CreateTokenDto` from `./dto/create-token.dto`, and `UpdateTokenDto` from `./dto/update-token.dto`. The controller has three methods: `create` (POST), `findAll` (GET), and `findOne` (GET). The `create` method calls `this.tokensService.create(createTokenDto)`. The `findAll` method calls `this.tokensService.findAll()`. The `findOne` method calls `this.tokensService.findOne(id)`.

[pads_oauth2/src/tokens/tokens.service.ts at main · Grupo-Investigacion-Bimadino/pads_oauth2 \(github.com\)](#)



The screenshot shows a code editor with the file path `pads_oauth2 / src / tokens / tokens.service.ts`. The code defines a `TokensService` class. It imports `Injectable` from `@nestjs/common`, `CreateTokenDto` from `./dto/create-token.dto`, `UpdateTokenDto` from `./dto/update-token.dto`, `InjectModel` from `@nestjs/mongoose`, `Model` from `'mongoose'`, and `tokens` from `./schemas/tokens`. The service has two methods: `create` and `findAll`. The `create` method calls `this.tokensModel.create(createTokenDto)`. The `findAll` method calls `this.tokensModel.find().exec()`.

COLECCIÓN PROVIDER

CRUD PROVIDER, POSTMAN

<https://github.com/Grupo-Investigacion->

[Bimadino/pads_oauth2/blob/main/documents/postman/Oauth2.0.postman_collection_provider.json](https://bimadino/pads_oauth2/blob/main/documents/postman/Oauth2.0.postman_collection_provider.json)

Kmanjarrez Documentación_Postman_provider

3d4e3a6 · yesterday History

Code Blame 121 lines (121 loc) · 2.42 KB

Raw Copy Download Edit View

```
"info": {
  "_postman_id": "dbdab2ed-3660-47f6-a6d1-0bea16c8b003",
  "name": "OAuth2.0",
  "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
  "_exporter_id": "35081855"
},
"item": [
  {
    "name": "provider",
    "request": {
      "method": "POST",
      "header": [],
      "body": {
        "mode": "raw",
        "raw": "{\\r\\n\\\"id\\\": \\\"1\\\",\\r\\n\\\"name\\\": \\\"Google\\\",\\r\\n\\\"runway\\\": \\\"www.google.com\\\",\\r\\n\\\"clue_api\\\": \\\"123456\\\",\\r\\n\\\"options\\\": {\\r\\n\\\"raw\\\": {\\r\\n\\\"language\\\": \"json\"}}}",
        "options": {
          "raw": {
            "language": "json"
          }
        }
      }
    },
    "response": []
  }
]
```

<https://github.com/Grupo-Investigacion->

[Bimadino/pads_oauth2/blob/main/src/provider/schemas/provider.ts](#)

Kissy Manjarrez

Actualización 18-06-24

0662afd · ayer

Historia

Código

Culpa

31 lines (21 loc) · 471 Bytes

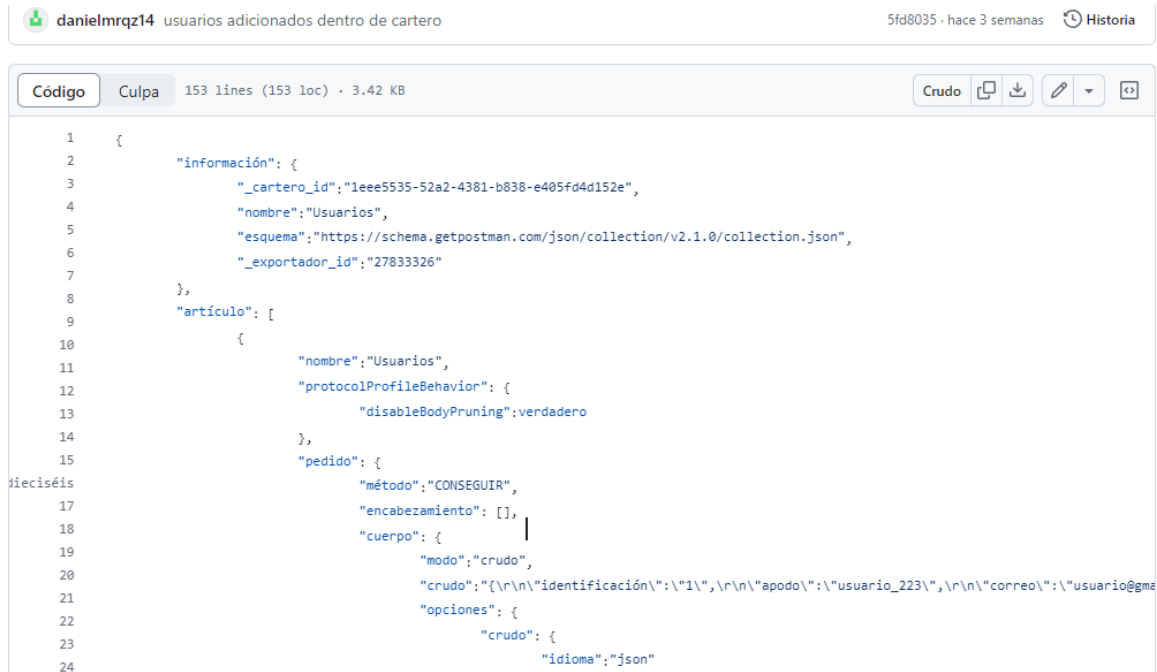
Crudo

```
1  importar { Apuntalar, Esquema, fábrica de esquemas } de '@nestjs/mangosta';
2  importar { Documento } de 'mangosta';
3  importar * como mangosta de 'mangosta';
4  importar { marca de tiempo } de 'rxjs';
5
6  @Esquema({
7
8    marcas de tiempo:verdadero, })
9
10  exportar clase proveedor se extiende Documento {
11    @Apuntalar()
12    Nombre:Cadena;
13
14    @Apuntalar()
15    Identificación:Cadena;
16
17    @Apuntalar()
18    pista:Cadena;
19
20    @Apuntalar()
21    pista_api:Cadena;
22
23    @Apuntalar()
```

COLECCIÓN USER

CRUD USER, POSTMAN

https://github.com/Grupo-Investigacion-Bimadino/pads_oauth2/blob/Conection/documents/postman/Users.json

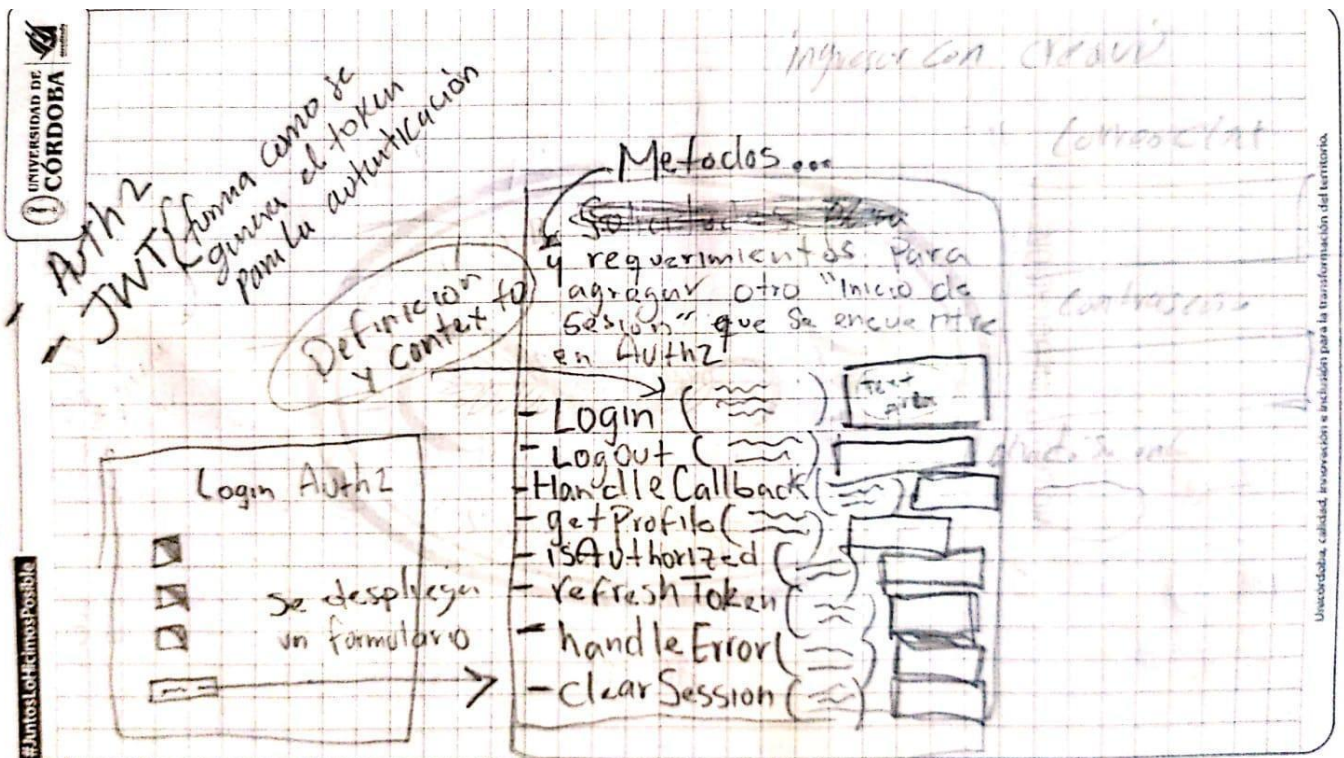
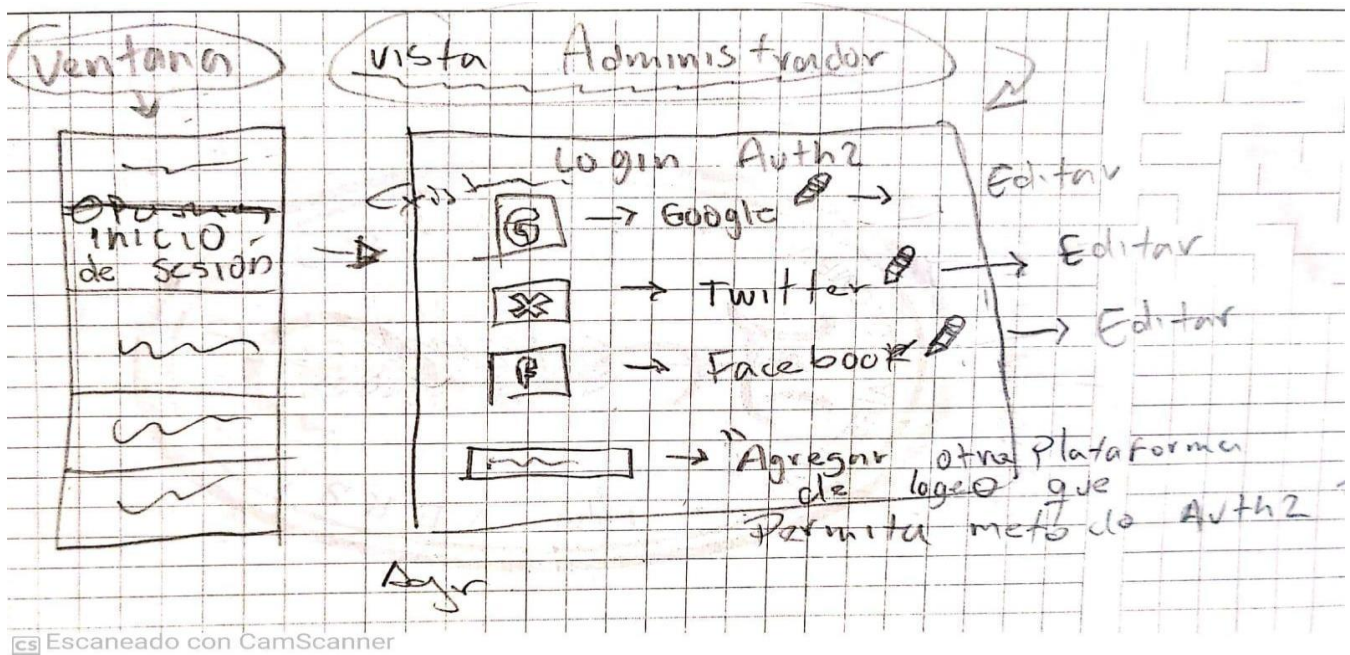


The screenshot shows the Postman interface for a collection named 'usuarios adicionados dentro de cartero' by user 'danielmrqz14'. The collection ID is '5fd8035' and it was created 'hace 3 semanas'. The editor is in 'Código' (Code) view, showing a JSON document with 153 lines (153 loc) and a size of 3.42 KB. The JSON structure is as follows:

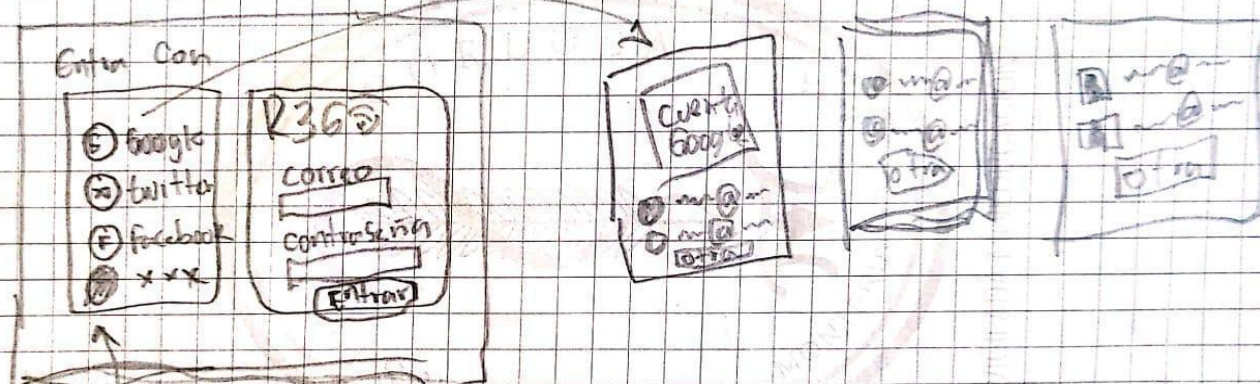
```
1  {
2    "información": {
3      "_cartero_id": "1eee5535-52a2-4381-b838-e405fd4d152e",
4      "nombre": "Usuarios",
5      "esquema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
6      "_exportador_id": "27833326"
7    },
8    "artículo": [
9      {
10        "nombre": "Usuarios",
11        "protocolProfileBehavior": {
12          "disableBodyPruning": verdadero
13        },
14        "pedido": {
15          "método": "CONSEGUIR",
16          "encabezamiento": [],
17          "cuerpo": {
18            "modo": "crudo",
19            "crudo": "{\n  \"identificación\": \"1\", \n  \"apodo\": \"usuario_223\", \n  \"correo\": \"usuario@gmail.com\", \n  \"password\": \"123456789\" \n}",
20            "opciones": {
21              "crudo": {
22                "idioma": "json"
23              }
24            }
25          }
26        }
27      }
28    ]
29  }
```


Anexos

Prototipar



vista de usuario



La que el
Adminis fuder
agregue

BIBLIOGRAFÍA

- OAuth.net. (2021). Understanding OAuth 2.0. <https://oauth.net/2/>
- Bizagi 11.2.3 BPM Suite User Guide - Digital Business Platform. (s/f). Bizagi.com. Recuperado el 23 de octubre de 2023, de https://help.bizagi.com/bpm-suite/es/11.2.3/index.html?cloud_auth_oauth.htm
- Hardt, D. (2012). The OAuth 2.0 authorization framework (D. Hardt, Ed.). RFC Editor.
- ¿Qué es OAuth 2.0 y para qué sirve? (s/f). Auth0. Recuperado el 23 de octubre de 2023, de <https://auth0.com/es/intro-to-iam/what-is-oauth-2>