

Informe de Desarrollo de la Aplicación Constellation

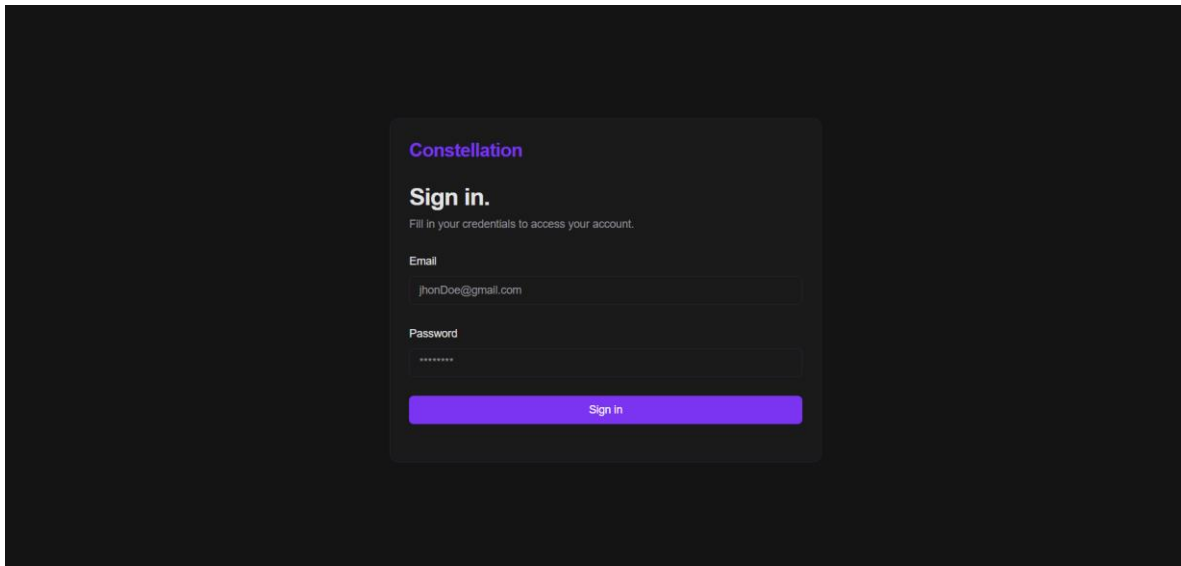
Descripción del Proyecto

Constellation es una aplicación web de gestión académica, desarrollada en **Next.js** y desplegada en **Vercel**. Está diseñada para optimizar la administración de actividades, cursos, rúbricas, grupos y usuarios dentro de un entorno educativo. Este proyecto se creó en el contexto del curso de **Computación en Internet 3**, aplicando una metodología ágil que permitió responder eficazmente a los requisitos cambiantes y específicos del sector educativo, asegurando un desarrollo flexible y centrado en el usuario.

Funcionalidades Implementadas

1. Autenticación y Autorización:

- Página de Inicio de Sesión:



La página de inicio de sesión permite a los usuarios autenticarse mediante un formulario intuitivo que solicita correo electrónico y contraseña. La autenticación se realiza usando un token **JWT (JSON**

Web Token) que se guarda en las cookies del navegador para autorizar futuras solicitudes.

Estos usuarios deben de estar previamente ya registrados en nuestra base de datos, para así cuando se inicie sesión según que rol posea, tiene ciertos roles.

- **Middleware de Protección de Rutas:**

Se implementó un middleware que valida el token de autenticación antes de conceder acceso a las rutas protegidas. Si el usuario no está autenticado, el middleware lo redirige automáticamente a la página de inicio de sesión.

```
const protectedRoutes = ['/dashboard'];

export default function middleware(request: NextRequest) {
  const currentUser = request.cookies.get("token")?.value;

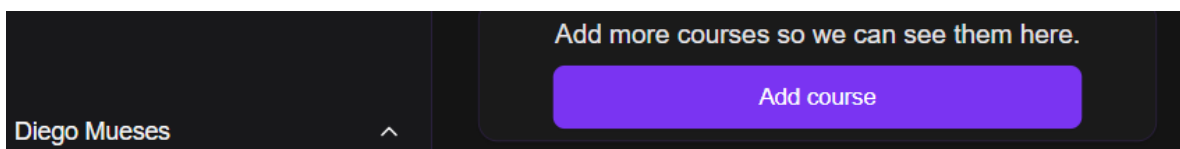
  if (protectedRoutes.includes(request.nextUrl.pathname) && !currentUser) {
    const response = NextResponse.redirect(new URL('/signin', request.url));
    return response;
  }

  return NextResponse.next();
}
```

r

- **Roles y Permisos:** Los usuarios se clasifican por roles (por ejemplo, profesor, estudiante) que determinan los permisos y el acceso a diferentes funcionalidades. Este sistema de roles garantiza que solo los usuarios autorizados puedan realizar acciones específicas, como crear cursos o actividades.

Profesor - Privilegios



Courses

Create New Course

Course Name

Description

Create Course

Estudiante – Privilegios



2. Gestión de Cursos y Actividades:

- Página de Cursos:

Constellation

Dashboard

Courses

Courses

4

Activities

2

Teams

2

Last courses

Discretas 78

Empresas, poder y nación

Algebra

Last activities

#	Name	Description
1	Taller next	Este es el taller next que deben entregar el 07/11/2024
2	Actividad prueba	Actividad prueba

A list of your recent activities.

La página de cursos permite a los usuarios visualizar y gestionar sus cursos, además de ver las ultimas actividades.

Los profesores pueden crear nuevos cursos, especificando nombre y descripción, y asignar generar grupos si es necesario.

Algoritmos y Programacion III

Upload studentsGenerate teams

7 Students

Teacher: Diego

Este curso finaliza la secuencia del bloque de Algoritmos y Programación brindando al estudiante una visión de cómo resolver problemas de una naturaleza radicalmente distinta a los que ha estudiado en los cursos anteriores. Estos son problemas para los que se han planteado soluciones de diversos tipos, inspirados en la forma en la que un humano los resolvería, que implican adoptar modos de razonamiento particulares, y que históricamente se han clasificado dentro de las técnicas de lo que actualmente se conoce como Inteligencia Artificial (IA). A lo largo del curso, los estudiantes explorarán diversas áreas, desde conceptos básicos hasta aplicaciones avanzadas, con énfasis en algoritmos y técnicas utilizadas en la resolución de problemas complejos en la Inteligencia Artificial.

Activities

New Activity

#	Name	Description
1	Tarea Integradora	Se debe de hacer un juego de 2 personas
2	RUBRICA DE ACTIVIDAD	DESCRIPCION DE PRUEBA DE ACTIVIDAD
3	Tarea Integradora de APO I	Realizar tarea integradora en scala
4	Seguimiento 476	Realizar Todo

Por otro lado, tenemos un apartado de subir los estudiantes mediante un template que nos genere aplicaciones como intu para añadir y genere los equipo

localhost:3000/courses/310a684d-8586-410e-a903-a0e23966226e

Algoritmos y Programacion III

Upload studentsGenerate teams

7 Students

Teacher: Diego

Este curso finaliza la secuencia del bloque de Algoritmos y Programación brindando al estudiante una visión de cómo resolver problemas de una naturaleza radicalmente distinta a los que ha estudiado en los cursos anteriores. Estos son problemas para los que se han planteado soluciones de diversos tipos, inspirados en la forma en la que un humano los resolvería, que implican adoptar modos de razonamiento particulares, y que históricamente se han clasificado dentro de las técnicas de lo que actualmente se conoce como Inteligencia Artificial (IA). A lo largo del curso, los estudiantes explorarán diversas áreas, desde conceptos básicos hasta aplicaciones avanzadas, con énfasis en algoritmos y técnicas utilizadas en la resolución de problemas complejos en la Inteligencia Artificial.


Activities

New Activity

#	Name	Description
1	Tarea Integradora	Se debe de hacer un juego de 2 personas
2	RUBRICA DE ACTIVIDAD	DESCRIPCION DE PRUEBA DE ACTIVIDAD
3	Tarea Integradora de APO I	Realizar tarea integradora en scala

Upload students

Upload a CSV file with the students to be added to the course

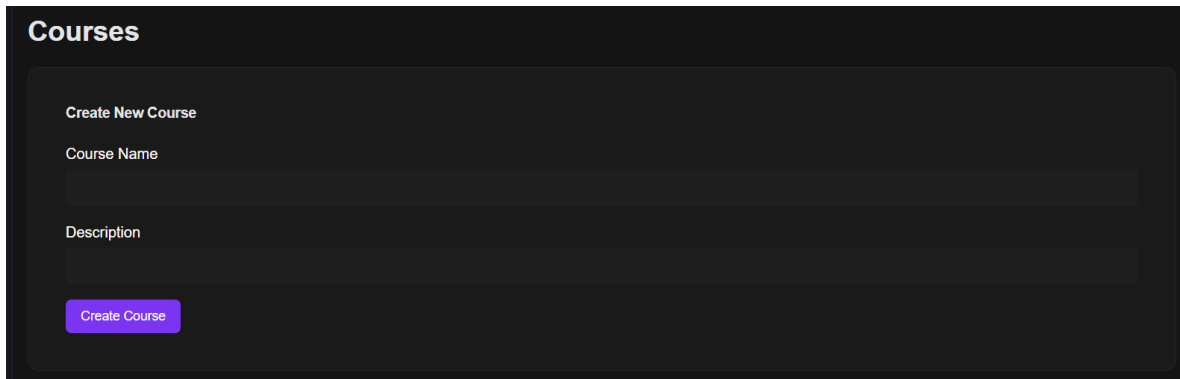


Drag & drop any file here
or [browse file](#) from device

Submit

- Formulario para Crear Cursos: El componente `CreateCourseForm` facilita la creación de cursos a través de un formulario controlado que solicita el nombre y la descripción del curso. Tras su creación, el curso se almacena en la base de datos y se refleja en la lista de cursos

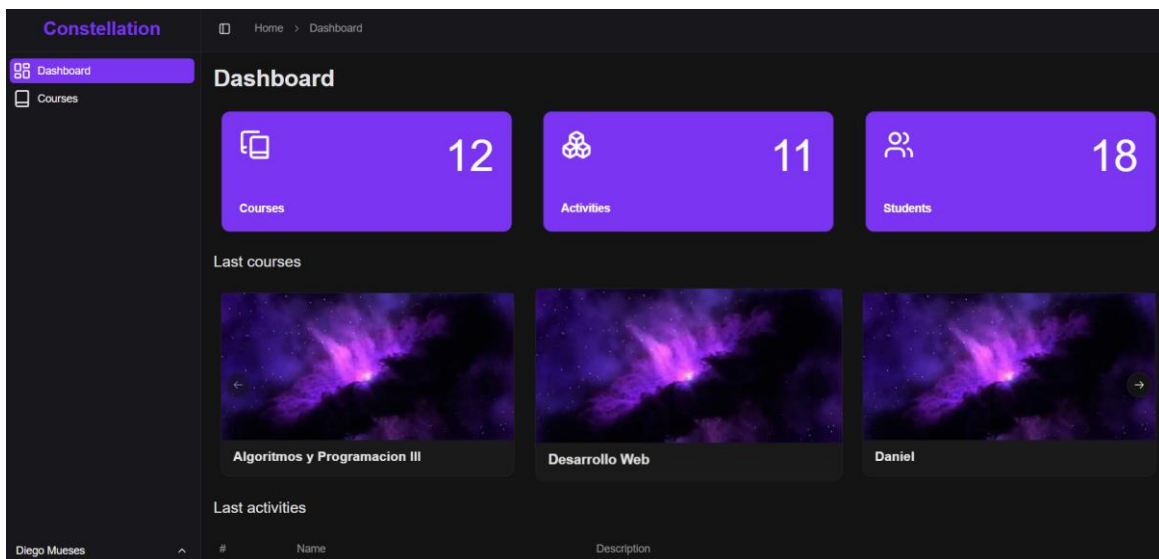
mostrada en la interfaz del usuario cabe aclarar que solo lo pueden generar las personas con rol profesor.



The screenshot shows a 'Courses' form with a dark background. At the top, it says 'Create New Course'. Below this, there are two input fields: 'Course Name' and 'Description'. At the bottom of the form, there is a blue button labeled 'Create Course'.

-Búsqueda y Filtro de Cursos: Los usuarios pueden buscar y filtrar cursos en tiempo real mediante un campo de búsqueda. Esto mejora la accesibilidad de los contenidos, facilitando la localización de cursos específicos.

3. Dashboard:



- **Página Principal de Usuario:** Tras iniciar sesión, el usuario accede al Dashboard, que presenta un resumen de su actividad

reciente en la plataforma, incluyendo los cursos y actividades en las que participa.

- **Componentes del Dashboard:**

- **CardDashboard:** Presenta tarjetas con información relevante, como cursos y actividades recientes.

- **DashboardCarousel:** Muestra los cursos recientes en un formato de carrusel interactivo, facilitando el acceso a los contenidos más importantes.

- **DashboardTable:** Lista las actividades recientes en una tabla, ayudando al usuario a organizar sus próximas tareas.

4. Gestión de Actividades:

- **Formulario de Creación de Actividades:**

The screenshot shows a web interface for a course titled "Algoritmos y Programacion III". A modal window titled "Activity" is open, allowing a user to create a new activity. The modal contains a "Name" field with the placeholder "Activity name" and a "Description" field with the placeholder "Describe here...". A "Submit" button is at the bottom of the modal. In the background, the course page is visible, showing a list of activities in a table.

#	Name	Description
1	Tarea Integradora	Se debe de hacer un juego de 2 personas
2	RUBRICA DE ACTIVIDAD	DESCRIPCION DE PRUEBA DE ACTIVIDAD
3	Tarea Integradora de APO I	Realizar tarea integradora en scala
4	Examen final	Realizar Todo

Los usuarios con permisos pueden crear actividades dentro de cursos específicos a través del **`ActivityForm`**, que recoge el nombre y la descripción de la actividad y los envía al backend para su almacenamiento.

Implementación de Autenticación, Autorización y Gestión del Estado

1. Autenticación y Almacenamiento de Tokens:

- **Proceso de Inicio de Sesión:** La autenticación comienza cuando el usuario ingresa sus credenciales. Si son correctas, se genera un token JWT que se guarda en el almacenamiento local del navegador o en las cookies. Este token se envía con cada solicitud al backend para verificar la identidad del usuario.

```
export function useLogin() {  
  const onSubmit = async (values: LoginFormValues) => {  
    try {  
      setIsSubmitting(true);  
      setError(null);  
      const response = await login(values);  
  
      if (response.success) {  
        if (response.data) Cookies.set("token", response.data.token);  
        router.push("/dashboard");  
      } else {  
        setError(  
          response.error?.message ??  
          "An unexpected error occurred. Please try again later."  
        );  
      }  
    } catch (error) {  
      setError("An unexpected error occurred. Please try again later." + error);  
    } finally {  
      setIsSubmitting(false);  
    }  
  }  
}
```

- **Almacenamiento Seguro de Tokens:** El token JWT se guarda en una cookie de sesión segura, y las rutas que requieren autenticación están protegidas para prevenir el acceso no autorizado.

```

export const useCurrentUser = () => {
  const [user, setCurrentUser] = useState<IToken | null>(null);

  useEffect(() => {
    const token = Cookies.get('token');
    if(token){
      const decodedToken = jwtDecode<IToken>(token);
      setCurrentUser(decodedToken);
    }
  }, []);

  const updateUser = (newToken: string) => {
    const decodedToken = jwtDecode<IToken>(newToken);
    setCurrentUser(decodedToken);
    Cookies.set('token', newToken);
  };

  return { user, updateUser };
};

```

2. Gestión de Roles y Permisos:

- **Implementación de Roles:** El sistema clasifica a los usuarios en diferentes roles, como "profesor" y "estudiante". Los permisos se definen en función del rol, permitiendo el acceso solo a ciertas funcionalidades y rutas. Por ejemplo, solo los profesores pueden crear o gestionar cursos, además de poder gestionar las actividades, rubricas y los criterios de cada rubrica.

- **Middleware de Autorización:** Las rutas protegidas se validan mediante un middleware que verifica el token del usuario y su rol antes de autorizar el acceso. Si el usuario intenta acceder a una funcionalidad sin los permisos adecuados, el sistema le deniega el acceso y lo redirige al sing in.

3. Gestión del Estado:

Skills Survey

What programming languages do you know?

C++	Angular	React	PostgreSQL
Vue	Go	JavaScript	Java
TypeScript	MongoDB	SQLite	Java
JavaScript	MySQL	C#	Express
	Redis	Python	Oracle
.NET	Python	Cassandra	Bootstrap

Next

At what time are you available?

Monday

--:--	🕒	-	--:--	🕒
15	16			
17	18			
19	20	-	--:--	🕒
21	22			
23	24	-	--:--	🕒

Thursday

Gestión de Estado con Redux y Hooks Personalizados

1. **Manejo del Estado Global con Redux:** Ahora, el estado de la aplicación se centraliza utilizando Redux. Esto permite gestionar datos de manera global en toda la aplicación. En particular, se ha creado un `skillsSlice` que maneja el estado de las habilidades (skills) seleccionadas. El store de Redux se configura con este reducer, lo que facilita la manipulación y acceso de los datos de habilidades desde cualquier componente de la aplicación.
2. **Hooks Personalizados con Redux:** Para simplificar el uso de Redux en los componentes, se han creado hooks personalizados `useAppDispatch` y `useAppSelector`, basados en `useDispatch` y `useSelector` de React-Redux. Esto permite que los componentes accedan al store y despachen acciones de manera tipada, haciendo que el código sea más limpio y menos propenso a errores.
3. **Actualización de Datos en Tiempo Real:** Los datos de los cursos, actividades y habilidades se actualizan en tiempo real mediante Redux. Cuando un usuario realiza una acción, como agregar o eliminar una habilidad, estos cambios se reflejan instantáneamente en la interfaz de usuario sin necesidad de recargar la página.
4. **Manejo de Errores:** Los hooks personalizados, como `useLogin` y `useCurrentUser`, junto con Redux, manejan los errores de manera eficiente. Cuando ocurre un error, se muestran mensajes claros que ayudan a los usuarios a entender el problema y tomar las acciones necesarias.

Conclusión

La aplicación **Constellation** implementa de manera eficiente una arquitectura de gestión de usuarios y cursos con autenticación y autorización robustas, que aseguran una experiencia segura y personalizada para cada tipo de usuario. La utilización de un token JWT para autenticación y las protecciones de rutas mediante middleware y gestión de roles aseguran la seguridad y precisión en el control de acceso. La implementación modular y la gestión de estado

permiten que la plataforma sea escalable y fácil de mantener, proporcionando una herramienta integral y adaptable para la gestión educativa.