

Práctica 1.3: Proyecto colaborativo inicial con seguimiento en Jira y desarrollo con GitHub, Django y React



Objetivo

Aprender a colaborar en equipo utilizando herramientas profesionales como Jira para el seguimiento de tareas y GitHub para el control de versiones, desarrollando un proyecto con Django (Backend) y React (Frontend). Este proyecto integrará un sistema de login moderno y funcional, con validación real de usuarios.

- A partir de esta actividad, deberán buscar un compañero de trabajo o de escuela para realizar las prácticas.



PASOS

Paso 1: Crear un proyecto en Jira y organizar tareas.

- **Crea un nuevo proyecto:**
 - ➔ Tipo: "Scrum Software project" o "Kanban" (usa Scrum si es posible).
 - ➔ Nombre: Proyecto Login Frontend-Backend.
- **Crea dos miembros en el equipo:**
 - ➔ Tú serás uno.
 - ➔ El otro será tu compañero (invítalo por correo).
- **Crea tareas (llamadas "issues") para cada rol dependiendo de las habilidades que tenga cada uno:**

- Tareas para el Frontend:

- ➔ Diseñar la interfaz del login con React.
- ➔ Agregar animaciones modernas con framer-motion.
- ➔ Conectar formulario al backend usando fetch.
- ➔ Mostrar mensajes de error si el login falla.

- Tareas Para el Backend:

- ➔ Crear el proyecto Django.
- ➔ Configurar vista de login (API REST).
- ➔ Validar credenciales contra la base de datos.
- ➔ Restringir acceso a la vista protegida si no hay login.

- Invita a tu compañero de trabajo o escuela al proyecto para que pueda ver y tomar sus tareas.

Paso 2: Crear un repositorio en GitHub y agregar al compañero.

- ➔ Nombre: `login-proyecto`
- ➔ Elige: `Público`
- ➔ Marca: "Add a README file"
- ➔ Ve a `Settings -> Collaborators` y agrega el usuario GitHub de tu compañero.
- ➔ Cada uno debe realizar la clonación del proyecto dentro de VS Code.

Paso 3: Crear el Backend.

- Este paso debe ser realizado por el compañero que le fue asignado Backend.
- Crear proyecto:

```
django-admin startproject backend  
cd backend
```

- Crear app para login:

`python manage.py startapp loginapp`

- Configurar archivos importantes
- Activar la app en `settings.py`

```
# backend/settings.py
INSTALLED_APPS = [
    ...
    'corsheaders',
    'loginapp',
    'rest_framework',
]
```

- Instalar CORS (permite conexión entre React y Django):
`pip install django-cors-headers django-rest-framework`
- Agrega en `settings.py`:

```
MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    ...
]

CORS_ALLOW_ALL_ORIGINS = True
```

- Crear vista de login (API)
- Código en `loginapp/views.py`:

```
from django.contrib.auth import authenticate
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
import json

@csrf_exempt
def login_view(request):
    if request.method == 'POST':
        data = json.loads(request.body)
        username = data.get('username')
        password = data.get('password')

        user = authenticate(username=username, password=password)
        if user is not None:
            return JsonResponse({'status': 'ok'})
        else:
            return JsonResponse({'status': 'fail'}, status=401)
```

- **Configurar las rutas**

- En loginapp/urls.py (crear si no existe):

```
from django.urls import path
from .views import login_view

urlpatterns = [
    path('login/', login_view),
]
```

- En backend/urls.py:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('loginapp.urls')),
]
```

- Crear superusuario para probar login en la terminal de VS Code:

```
python manage.py migrate
python manage.py createsuperuser
```

- Correr el servidor:

```
python manage.py runserver
```

- **Carga cambios (Push)**

- Seleccionamos todos los archivos archivo que modificamos con el comando:

- **git add.**

- Agregamos commit para comentar los cambios realizados:

- **git commit -m "Terminé Backend"**

- Subimos nuestros cambios al repositorio

- **git push**

- Una vez realizado esto, ingresamos a git hub para verificar que los cambios se hayan aplicado correctamente.

- Descargar cambios (Pull)
- `git pull origin main`
- Esto asegurará que tienes la versión más actual del proyecto y cargar los archivos de Frontend.

Paso 4: Crear el Frontend con React.

- Crear proyecto:

```
npx create-react-app frontend  
cd frontend
```

- Instalar Bootstrap y animaciones:

```
npm install bootstrap framer-motion
```

- Importar estilos en src/index.js:

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

- Instalar CORS (permite conexión entre React y Django):

```
pip install django-cors-headers djangorestframework
```

- Crear formulario de login con animaciones

- Código completo de src/App.js:

```
import React, { useState } from 'react';  
import { motion } from 'framer-motion';  
  
function App() {  
  const [username, setUsername] = useState('');  
  const [password, setPassword] = useState('');  
  const [message, setMessage] = useState('');  
  
  const handleLogin = async () => {  
    const res = await fetch('http://localhost:8000/login/', {  
      method: 'POST',  
      headers: { 'Content-Type': 'application/json' },  
      body: JSON.stringify({ username, password })  
    });  
  });
```

```

    if (res.ok) {
      setMessage('✅ Login exitoso');
    } else {
      setMessage('❌ Usuario o contraseña incorrectos');
    }
  };

  return (
    <div className="container mt-5">
      <motion.div
        className="card p-4"
        initial={{ opacity: 0, y: -20 }}
        animate={{ opacity: 1, y: 0 }}
        transition={{ duration: 1 }}
      >
        <h3 className="mb-4 text-center">Iniciar Sesión</h3>
        <input
          className="form-control mb-3"
          placeholder="Usuario"
          value={username}
          onChange={(e) => setUsername(e.target.value)}
        />
        <input
          className="form-control mb-3"
          type="password"
          placeholder="Contraseña"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
        />
        <button className="btn btn-primary w-100" onClick={handleLogin}>
          Entrar
        </button>
        {message} && <div className="alert alert-info mt-3">{message}</div>
      </motion.div>
    </div>
  );
}

export default App;

```

- **Carga cambios (Push)**
- Seleccionamos todos los archivos archivo que modificamos con el comando:

- `git add.`
- Agregamos commit para comentar los cambios realizados:
- `git commit -m "Terminé Frontend"`
- Subimos nuestros cambios al repositorio
- `git push`
- Una vez realizado esto, ingresamos a git hub para verificar que los cambios se hayan aplicado correctamente.
- Esto asegurará que tienes la versión más actual del proyecto y cargar los archivos de Backend.
- **Probar la aplicación**
- Asegúrate de que Django esté corriendo:
- `python manage.py runserver`
- En otra terminal, ejecuta React:
- `npm start`
- En el navegador, abre <http://localhost:3000>, llena el formulario y prueba tu superusuario.