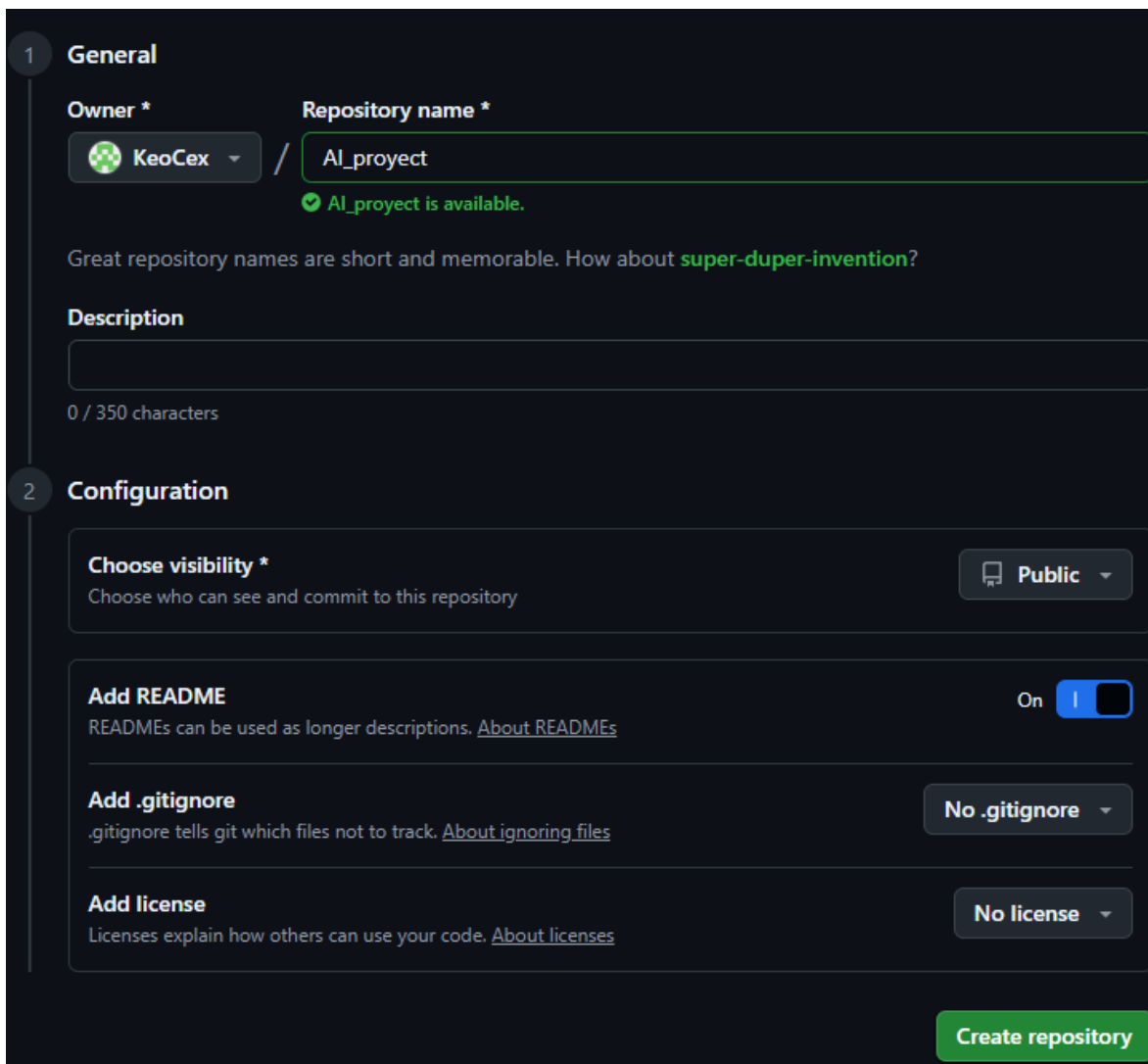


Actividad formativa: GitHub Copilot

Una vez iniciada la sesión, creamos un repositorio en GitHub, asignamos como nombre `AI_project`, repositorio publico, habilitamos el README y creamos el repositorio



The screenshot shows the GitHub repository creation page with a dark theme. It is divided into two sections: '1 General' and '2 Configuration'.

1 General

- Owner ***: A dropdown menu showing 'KeoCex' with a green checkmark icon.
- Repository name ***: A text input field containing 'AI_proyect'. Below the field, a green checkmark and the text 'AI_proyect is available.' are displayed.
- Description**: A text input field with a character count '0 / 350 characters' below it.

2 Configuration

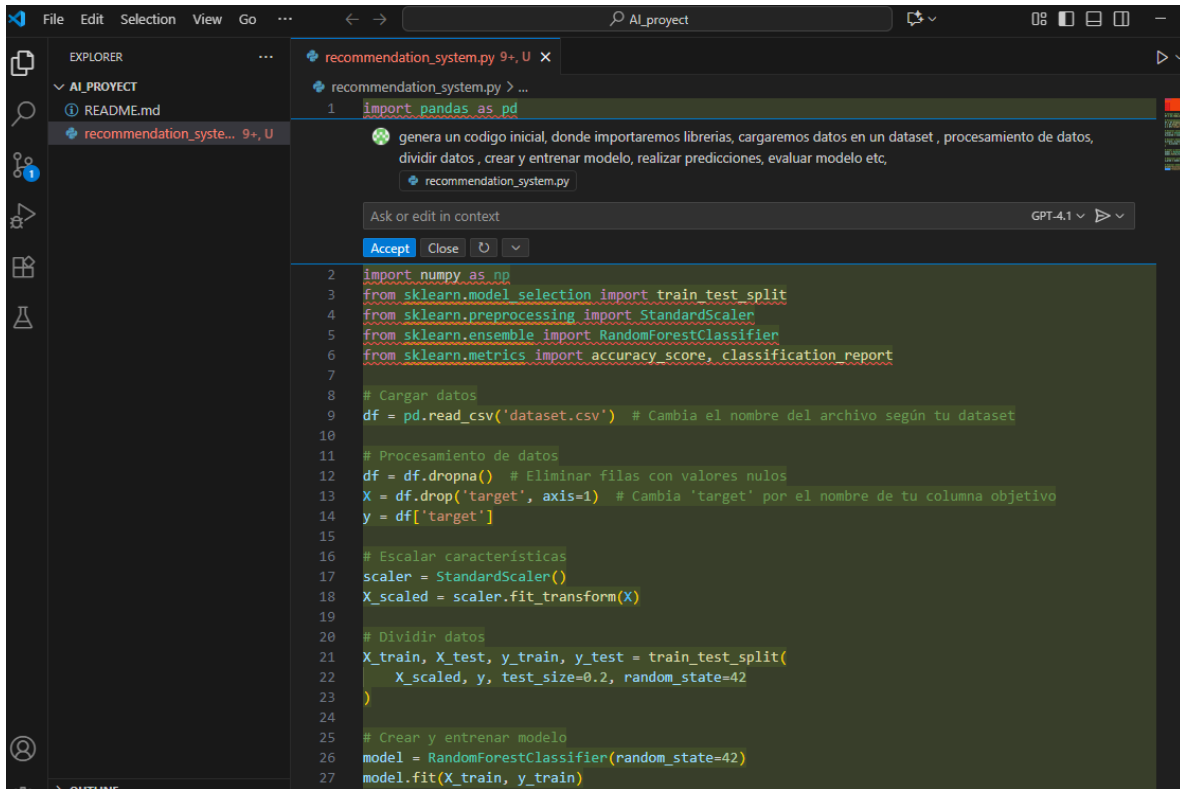
- Choose visibility ***: A dropdown menu showing 'Public' with a lock icon.
- Add README**: A toggle switch labeled 'On' is turned on. Below it, text says 'READMEs can be used as longer descriptions. [About READMEs](#)'.
- Add .gitignore**: A dropdown menu showing 'No .gitignore'. Below it, text says '.gitignore tells git which files not to track. [About ignoring files](#)'.
- Add license**: A dropdown menu showing 'No license'. Below it, text says 'Licenses explain how others can use your code. [About licenses](#)'.

A green button labeled 'Create repository' is located at the bottom right of the configuration section.

Clonamos el repositorio de GitHub en VisualStudio Code

https://github.com/KeoCex/AI_project

creamos un nuevo archivo con nombre `recommendation_system.py`, y solicitamos a la IA que nos genere un código inicial



```
File Edit Selection View Go ... AI_proyect
EXPLORER
  AI_PROYECT
    README.md
    recommendation_syste... 9+, U
recommendation_system.py 9+, U
  1 import pandas as pd

  genera un código inicial, donde importaremos librerías, cargaremos datos en un dataset, procesamiento de datos,
  dividir datos, crear y entrenar modelo, realizar predicciones, evaluar modelo etc.
  recommendation_system.py
  Ask or edit in context GPT-4.1
  Accept Close
  2 import numpy as np
  3 from sklearn.model_selection import train_test_split
  4 from sklearn.preprocessing import StandardScaler
  5 from sklearn.ensemble import RandomForestClassifier
  6 from sklearn.metrics import accuracy_score, classification_report
  7
  8 # Cargar datos
  9 df = pd.read_csv('dataset.csv') # Cambia el nombre del archivo según tu dataset
  10
  11 # Procesamiento de datos
  12 df = df.dropna() # Eliminar filas con valores nulos
  13 X = df.drop('target', axis=1) # Cambia 'target' por el nombre de tu columna objetivo
  14 y = df['target']
  15
  16 # Escalar características
  17 scaler = StandardScaler()
  18 X_scaled = scaler.fit_transform(X)
  19
  20 # Dividir datos
  21 X_train, X_test, y_train, y_test = train_test_split(
  22     X_scaled, y, test_size=0.2, random_state=42
  23 )
  24
  25 # Crear y entrenar modelo
  26 model = RandomForestClassifier(random_state=42)
  27 model.fit(X_train, y_train)
```