

# Aplicaciones LLM

LlamaIndex: create-llama

# create-llama

- Allows you to create a full-functional RAG application with a few commands in the terminal.
- Easy to deploy to Vercel and have a free public app hosted there.

# Steps to create a RAG app and deploy it to Vercel

- Decide: local development or github codespace?
- Create the RAG app
- Deploy the app to Vercel

# Decide: local development or github codespace

- If you do it locally, you will need to have several packages installed (node, etc) and then load the app to a github repo.
- If you do it in github codespaces, you will use the terminal there, the necessary packages will be pre-installed already, the github repo will be created automatically and the Vercel deployment will be fast and easy.
  - This is our recommendation.

# Creating a github codespace

- Go to your github account
- Create a new repository
- Open in in a codespace
- Now you can use the terminal from the codespace

# Enter this commands in the codespace terminal

- `npx create-llama@latest`
  - downloads the package
- enter the name of the app
- select template to use:
  - chat without streaming
  - chat with streaming (default)
- select the framework to use:
  - NextJS (default: for Vercel)
  - Express
  - FastAPI
- select (the chat) UI
  - just html (default)
  - shadcn
- select the chat engine:
  - SimpleChatEngine (default)
  - ContextChatEngine: for RAG
- provide OpenAI API key
  - include `.env` in `.gitignore` later
- select if you like to use ESLint

# Upload your RAG private documents to /data

- Remove the sample file in my-app/data
- Upload your private documents in my-app/data
- `cd my-ap`
- `npm run generate`
- It splits your private documents in chunks and convert them in embeddings using OpenAIEmbedding

# Analyze file structure of the app

- Typical NextJS App structure
- In app-name/app/api/engine/route.js
  - here is the chat route



# Run the app

- `npm run dev`
- See the link to the local server
- If you need to stop it, CTRL+C

# Load changes in github repo

- `cd ..`
- `git status`
- `git add my-app`
- `git status`
- `git commit -m "llamaindex app created using create-llama"`
- `git status`
- `git push`

# Deploy to Vercel

- **Go to vercel.com**
- Sign up with Github
- Give access to import the app's github repo
- **Import the repo into Vercel**
- Root directory: click edit and select the my-app folder
- Environment variables: enter the OpenAI API Key
- **Click on the deploy button**
- It will take some time to complete the deployment
- **If you click on the app screen, it will open the app on Vercel**
- Confirm the app works OK
- Check the Vercel dashboard
- Now in the page of the github repo there is a link to the app on Vercel
- If you want to delete the project from Vercel: dashboard>settings>delete project