

Make Rusty AI

Arquitecturas de IA en Rust

Jorge D. Ortiz Fuentes

Canonical Examples

2026-03-17

A Canonical Examples production

Intro

Agenda

- ★ LLMs
- ★ Chat REPL
- ★ RAG
- ★ Agentes con herramientas
- ★ Multi-agente

Objetivos

- ★ Aprender los fundamentos de las aplicaciones con IA
- ★ 100% con Rust (Nada de Python, circulen)

Fuera de alcance

- ★ Fundamentos de Rust
- ★ Estado del arte de la IA

LLMs 



Di *tu* palabra cuando cuente 3 (1,2,3)

Yo estoy estudiando ...

Me encanta escribir código en Rust
porque es ...

Hay muchas cosas que se pueden
echar al cocido, pero nunca pongas ...

**iCreemos nuestra bola
de cristal!**



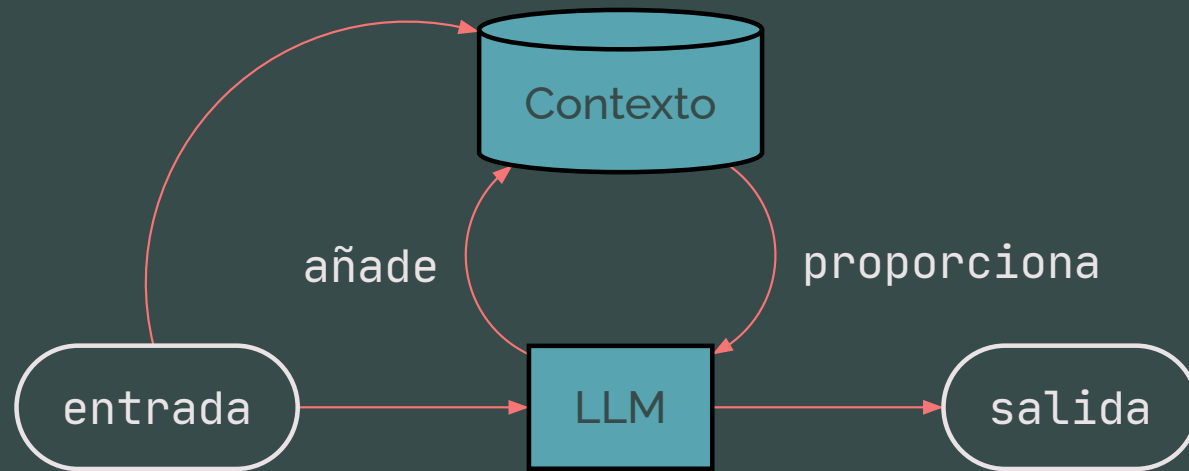
iA programar!





Chat REPL

Chat REPL



Contexto

```
let mut messages = vec![
    ChatMessage::user().content("Hello. Call me Jorge.").build(),
    ChatMessage::assistant()
        .content("Hello, Jorge! How can I help you today.")
        .build(),
];

loop {
    match readline {
        Ok(line) => {
            messages.push(ChatMessage::user().content(line).build());
            chat_interaction(&llm, &mut messages).await;
        },
        _ => { } // Other cases
    }
}
```

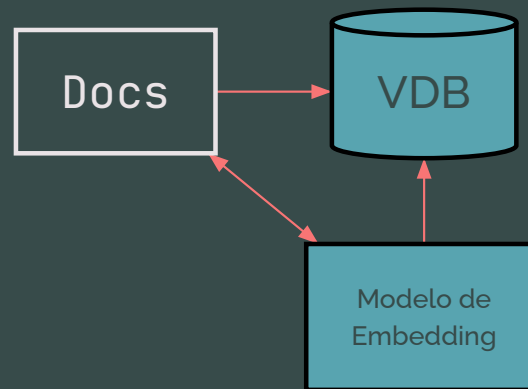
A man with a shaved head, wearing a dark blue suit, white shirt, and dark tie, is seated in a futuristic, high-tech chair. He is looking directly at the camera with a serious expression. His hands are resting on the armrests of the chair, which have some control panels. The background is a dark, industrial-looking space with glowing blue and white lights, suggesting a futuristic or high-tech environment. The overall tone is serious and professional.

Retrieval Augmented Generation

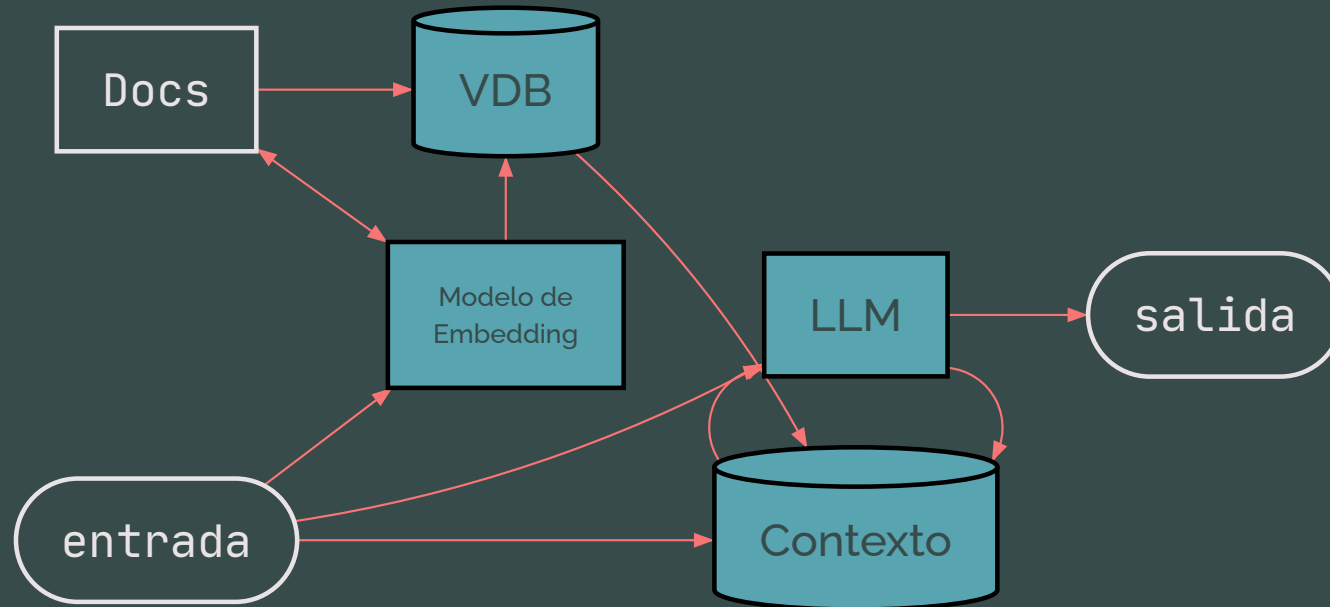
Retrieval Augmented Generation

- ★ Cuándo: Información nueva o no publica
- ★ Cómo:
 1. Usa búsqueda vectorial con el *prompt* para recuperar información relacionada
 2. Usa la información junto con el *prompt* para generar la respuesta

RAG



RAG



RAG

```
let emb_client = client.embedding_model("nomic-embed-text");

let docs = fetch_notes();
let embeddings = EmbeddingsBuilder::new(emb_client.clone())
    .documents(docs)?
    .build()
    .await?;
let vector_store = InMemoryVectorStore::from_documents(embeddings);
let index = vector_store.index(emb_client);

let agent = client
    .agent("qwen3-coder:30b")
    .preamble(AGENT_RULES)
    .dynamic_context(1, index)
    .build();

let response = agent.prompt(line).await?;
```

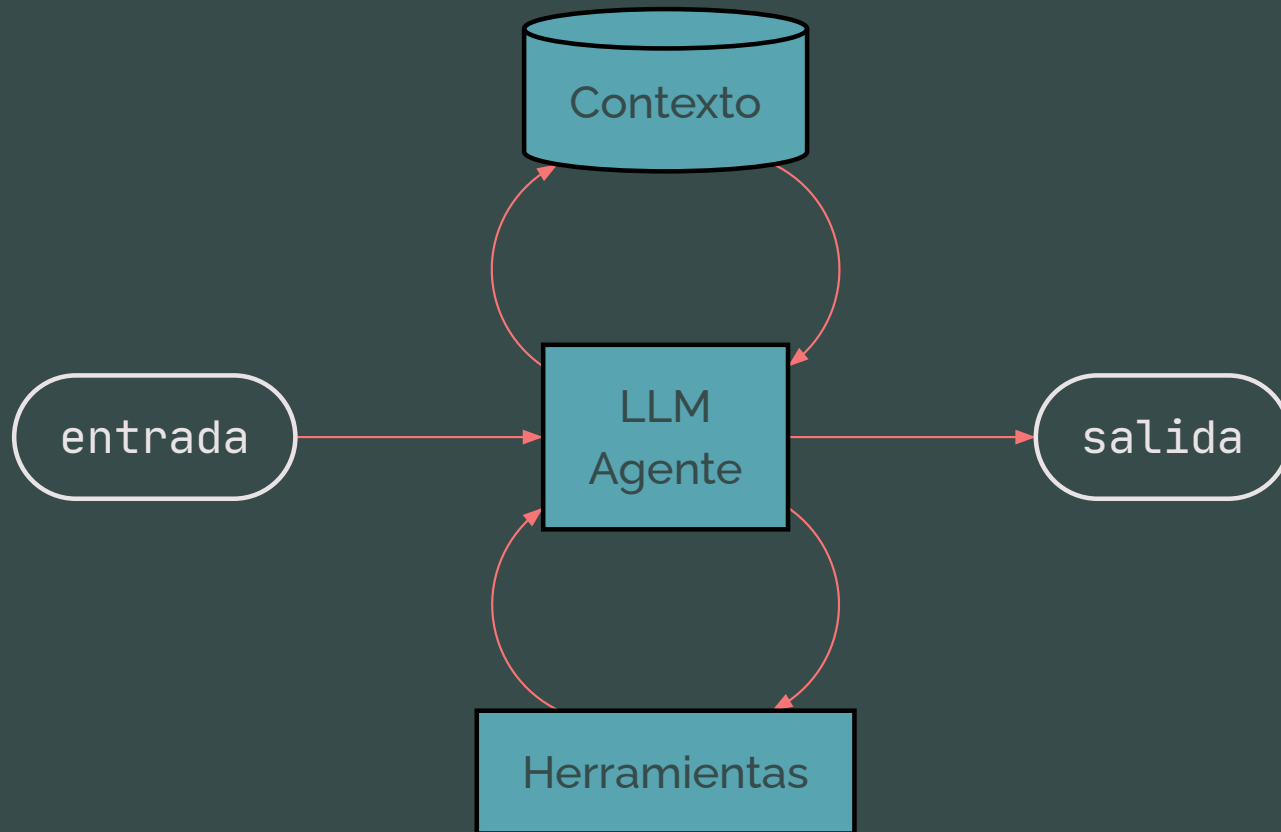


Agents

Agentes

- ★ No solo contestan, hacen cosas en nuestro nombre
 - Herramientas** Registra funciones con su descripción y E/S estructurada
 - MCP** Servicios estructurados que proporcionan herramientas, recursos y prompts

Agente



Agente

```
loop {
  let compl_resp = agent.completion(&input, chat.to_owned())
    .await?.send().await?;
  if let AssistantContent::Text(rsp) = compl_resp.choice.first() {
    let response = rsp.text();
    let next_input: String = match parse_action(response) {
      Ok(act_req) => {
        let result = match act_req.tool_name.as_str() {
          "list_files" => { list_files() }
          "read_file" => {
            act_req.args.get("file_name").map_or(json!
({ "error": "No file_name."}), |f| read_file(f))
          }
          _ => { } // Other tools
        };
        if finish { agent_rsp = result.to_string(); }
        iter += 1;
        if iter ≥ MAX_ITER { bail!("Too long"); }
        result.to_string()
      }
    }
  }
}
```

```
        Err(err) => { } // Deal with error
    };
    chat.push(Message::user(input));
    chat.push(Message::assistant(response));
    input = next_input;
    if finish { return Ok(agent_rsp); }
}
}
```



Multi-agents

Multi-agentes

- ★ Usa varios agentes:
 - Personas / especializaciones diferentes
 - Herramientas diferentes
 - Responsabilidades diferentes
- ★ Orquestración
 - A2A

Conclusiones

- ★ No necesitas ser un experto en ML para escribir aplicaciones de IA
- ★ Arquitecturas. Usa
 - Contexto + Bucle** memoria
 - RAG** Información nueva/adicional
 - MCP/Herramientas** IA hace cosas
 - Multi-agente** Acción combinada de **personas** diferentes

Gracias

¿Preguntas?

@jdortiz

github.com/jdortiz