

CARRERA DE INGENIERÍA EN SOFTWARE



METODOLOGIA AGIL DE DESARROLLO

Autores:

Muñiz Rivas Leopoldo Miquel

Tema:

Agno

Curso:

5 - B

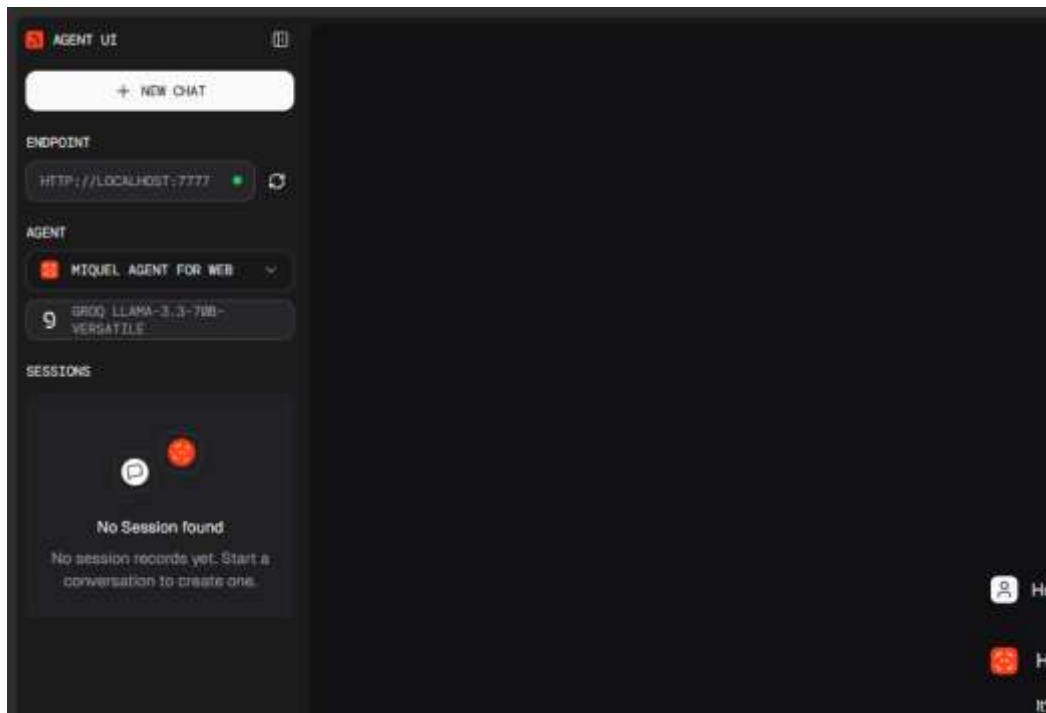
Periodo 2025(1)

1. Código de los agentes

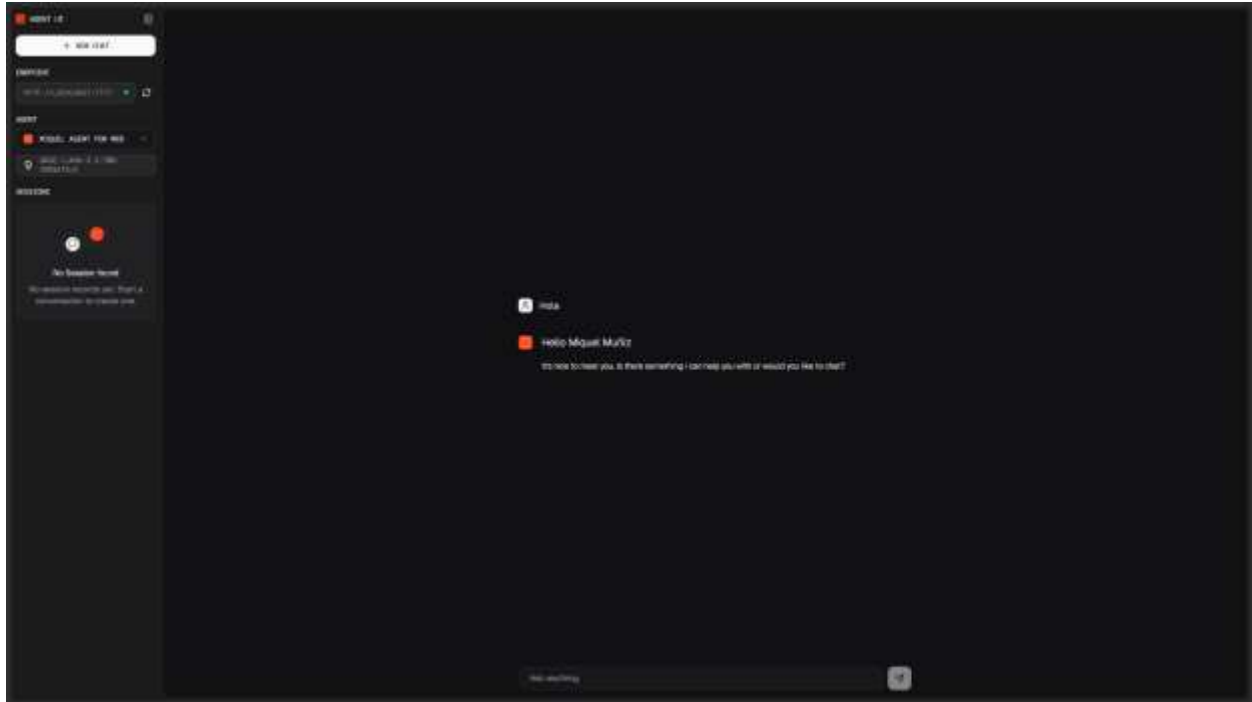
```
from agno.agent import Agent
from agno.models.groq import Groq
from agno.playground import Playground
from agno.storage.sqlite import SqliteStorage
from agno.tools.duckduckgo import DuckDuckGoTools
from agno.tools.yfinance import YFinanceTools
agent_storage: str = "tmp/agents.db"
web_agent = Agent(
    name="Miquel Agent for Web",
    model=Groq(id="llama-3.3-70b-versatile"),
    tools=[DuckDuckGoTools()],
    instructions=["You always referred to the user as Miquel Muñiz"],
    # Store the agent sessions in a sqlite database
    storage=SqliteStorage(table_name="web_agent", db_file=agent_storage),
    # Adds the current date and time to the instructions
    add_datetime_to_instructions=True,
    # Adds the history of the conversation to the messages
    add_history_to_messages=True,
    # Number of history responses to add to the messages
    num_history_responses=5,
    # Adds markdown formatting to the messages
    markdown=True,
)
finance_agent = Agent(
    name="Miquel Agent for Finance",
    model=Groq(id="llama-3.3-70b-versatile"),
    tools=[YFinanceTools(stock_price=True, analyst_recommendations=True,
company_info=True, company_news=True)],
    instructions=["You always referred to the user as Miquel Muñiz"],
    storage=SqliteStorage(table_name="finance_agent", db_file=agent_storage),
    add_datetime_to_instructions=True,
    add_history_to_messages=True,
    num_history_responses=5,
    markdown=True,
)
playground_app = Playground(agents=[web_agent, finance_agent])
app = playground_app.get_app()
if __name__ == "__main__":
    playground_app.serve("main:app", reload=True)
```

2. Agentes

Así es como se observa a los agentes ya funcionando e integrados con la interfaz de usuario (UI). Se puede ver en tiempo real cómo interactúan, responden a eventos, y actualizan visualmente su estado dentro del sistema. Esta visualización facilita el monitoreo, la validación del comportamiento de los agentes y la comprobación de que la integración entre la lógica interna y la capa visual se ha realizado correctamente.



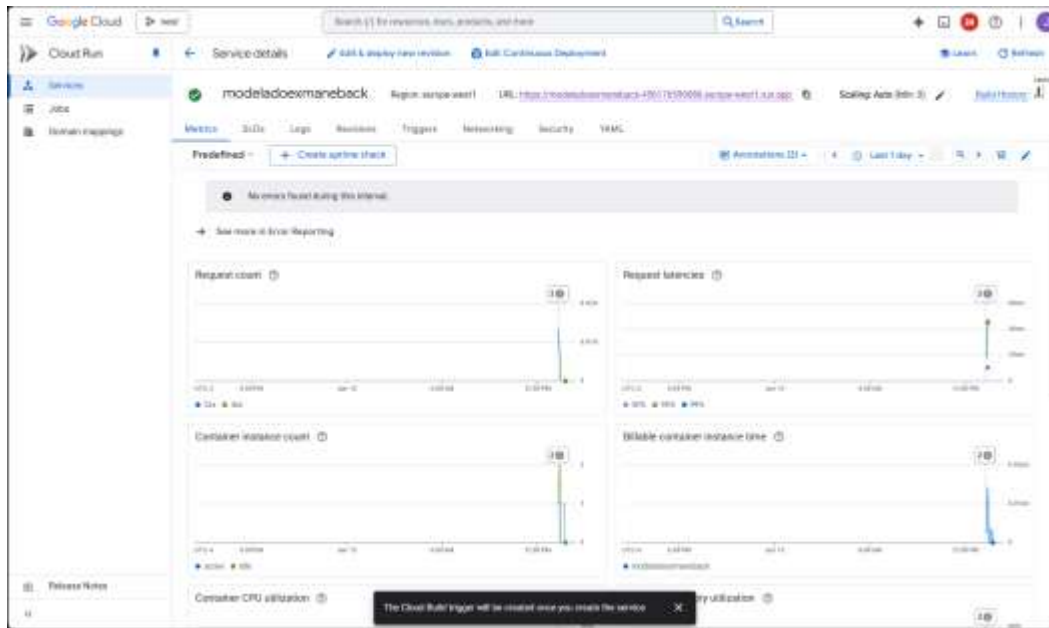
3. Ejecucion de la vista



4. Agentes subido con code run

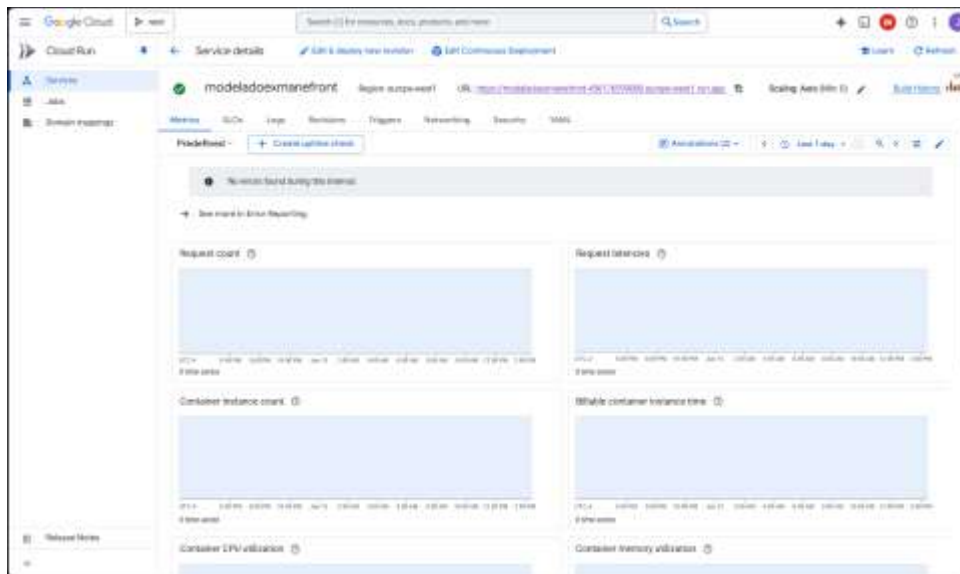
Aquí podemos observar la interfaz de usuario (UI) junto con los agentes desplegados y ejecutándose en Code Run. Esta plataforma permite visualizar cómo los agentes funcionan en tiempo real dentro del entorno de ejecución, mostrando la interacción directa entre la UI y la lógica de los agentes. De esta manera, se facilita el seguimiento de su comportamiento, la detección de posibles errores y la evaluación del rendimiento general del sistema mientras está activo.

Link: <https://modeladoexmaneback-456176599080.europe-west1.run.app>



5. Ui subido en code run

Link : <https://modeladoexmanefront-456176599080.europe-west1.run.app>

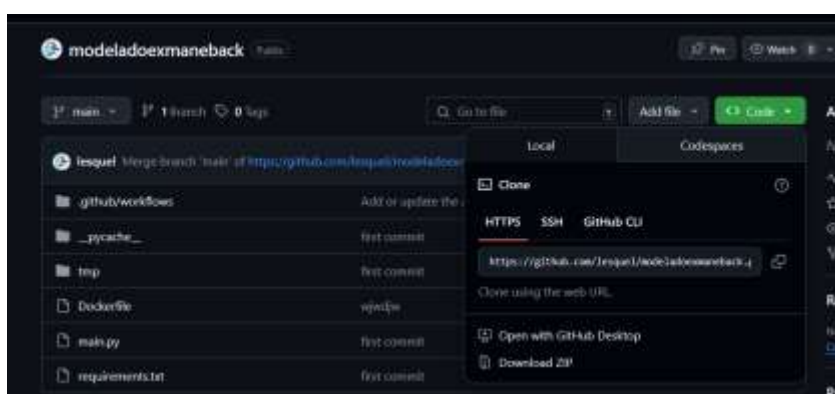


6. Link del repositorio de los agente:

Aquí podemos observar la interfaz de usuario (UI) junto con los agentes y los repositorios. La UI muestra de forma clara y organizada la información y el estado actual de los

agentes, quienes interactúan con los datos almacenados en los repositorios. Estos repositorios actúan como almacenes centrales donde se guardan y gestionan los datos que los agentes utilizan para tomar decisiones y realizar acciones. De esta manera, la integración entre la UI, los agentes y los repositorios permite una gestión eficiente y transparente del flujo de información dentro del sistema.

Link: <https://github.com/lesquel/modeladoexmaneback.git>



7. Link del repositorio en la ui:

Link : <https://github.com/lesquel/modeladoexmanefront.git>

