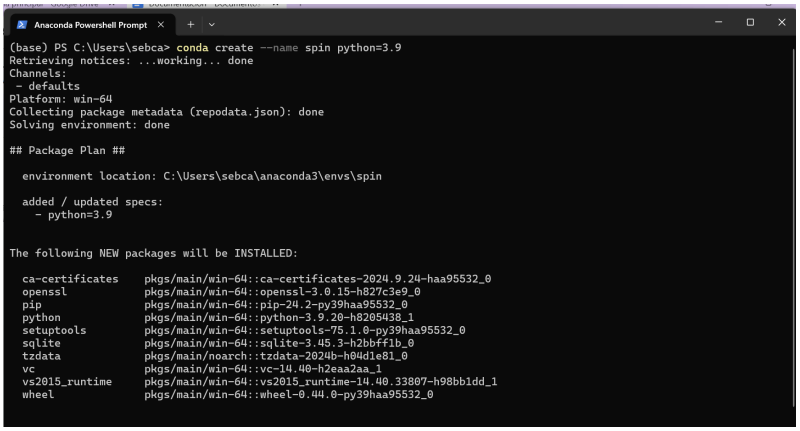
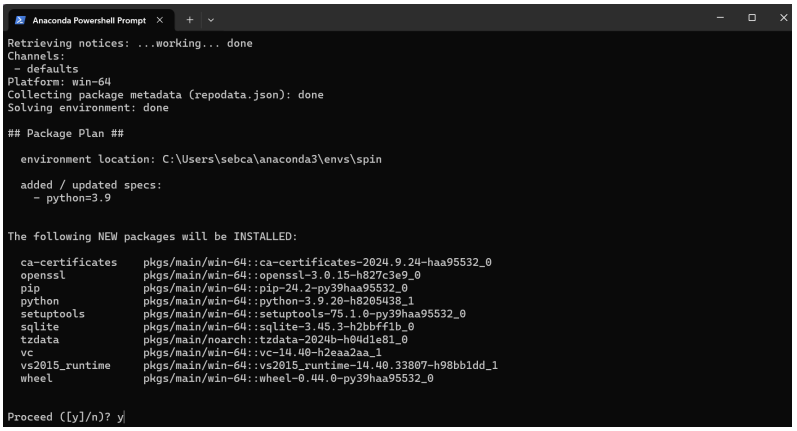
Documentación Conexión QC

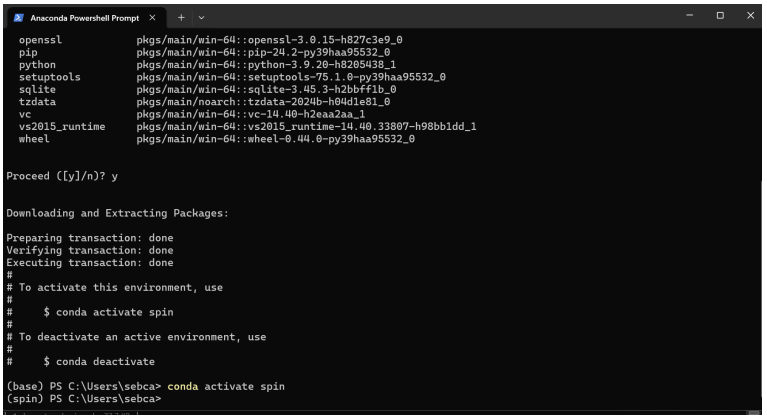
- Instalar Anaconda (Si es en Windows, ejecutar todos los comando en Anaconda Powershell Prompt)

- Crear nuevo entorno de trabajo
  conda create - -name nombre_entorno python=3.9

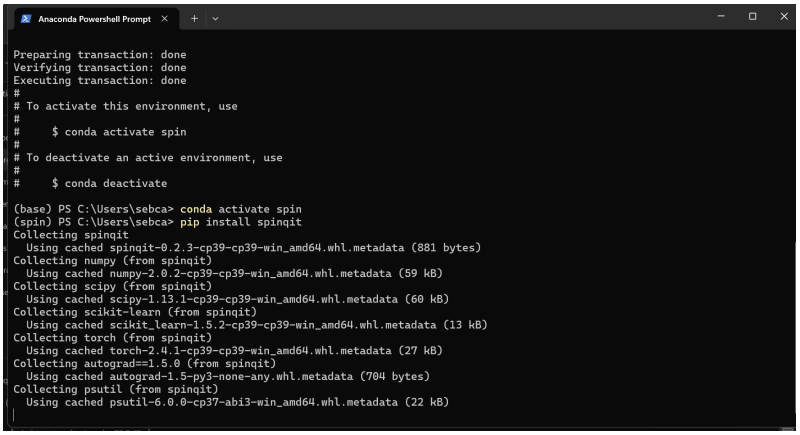


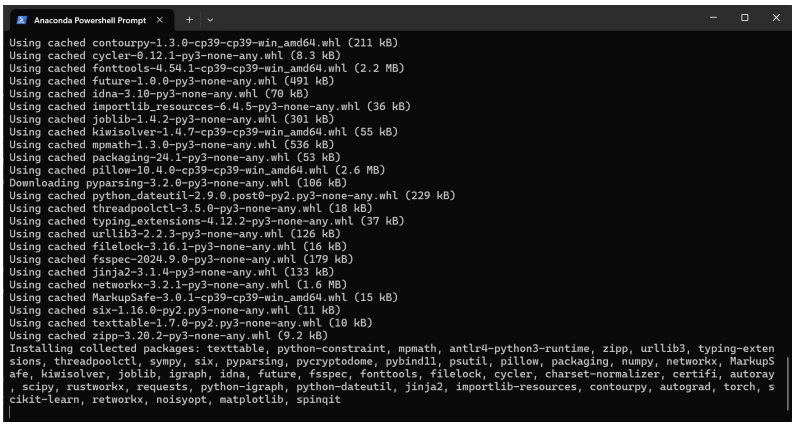


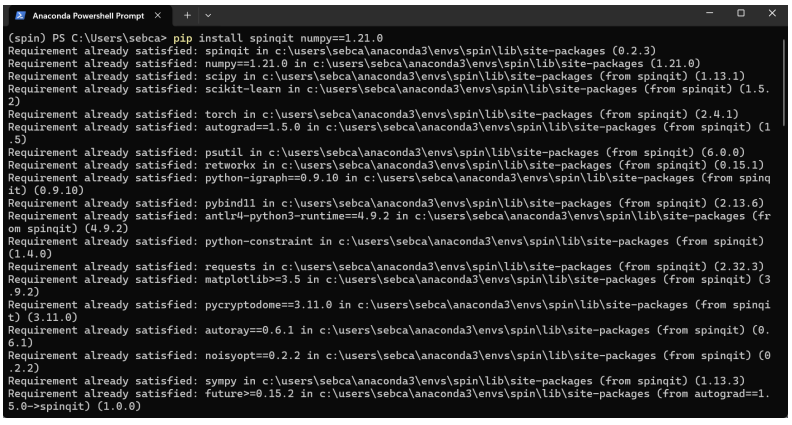
- Activar entorno creado
  conda activate nombre_entorno



- Instalar bibliotecas necesarias
  pip install spinqit
  pip install spinqit numpy==1.21.0







- Crear programa en python o jupyter notebook el cual utilice el entorno creado anteriormente

- Código de ejemplo:

```
from spinqit import get_nmr, get_compiler, Circuit, NMRConfig
from spinqit import H, CX, Rx
```

```
engine = get_nmr()
comp = get_compiler("native")

circ = Circuit()

q = circ.allocateQubits(2)
circ << (H, q[0])
circ << (H, q[1])


exe = comp.compile(circ, 0)
config = NMRConfig()
config.configure_shots(1024)
config.configure_ip("172.27.52.229")
config.configure_port(8989)
config.configure_account("SpinQ001","123456")#usuario y
contraseña
config.configure_task("Test-JUT","Test-JUT") #nombre de la tarea

result = engine.execute(exe, config)
print(result.probabilities)
```

Explicación del código:

- Se importan funciones y clases de la librería **spinqit** para crear y ejecutar circuitos cuánticos en una simulación basada en NMR (resonancia magnética nuclear):

```
from spinqit import get_nmr, get_compiler, Circuit, NMRConfig
from spinqit import H, CX, Rx
```

- Configuración del motor de simulación y compilador:

```
engine = get_nmr()
comp = get_compiler("native")
```

- Creación del circuito:

```
circ = Circuit()
q = circ.allocateQubits(2) #Debido a que el computador cuántico
trabaja con dos qubits reales, solo podemos agregar esta cantidad
al trabajar con get_nmr
circ << (H, q[0])
```

```
circ << (H, q[1])
```

- Compilación del circuito:

```
exe = comp.compile(circ, 0)
```

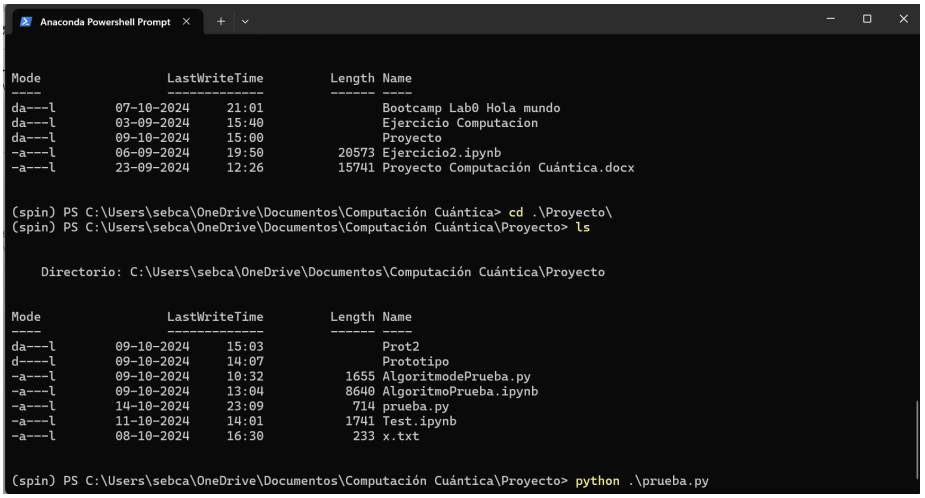
- Configuraciones necesarias para conectarse al computador cuántico:

```
config = NMRConfig()
config.configure_shots(1024)
config.configure_ip("172.27.52.229") #IP del Computador cuantico
config.configure_port(8989) #puerto al cual escucha el computador
cuántico
config.configure_account("SpinQ001","123456")        #usuario    y
contraseña
config.configure_task("Test-JUT","Test-JUT") #tarea
```

- Ejecución y resultados:

```
result = engine.execute(exe, config)
print(result.probabilities)
```

En caso de haber creado archivo *.py*, ejecutar el comando ***python .nombre_app.py***



Esto fue probado tanto en el sistema operativo de Windows como en Ubuntu