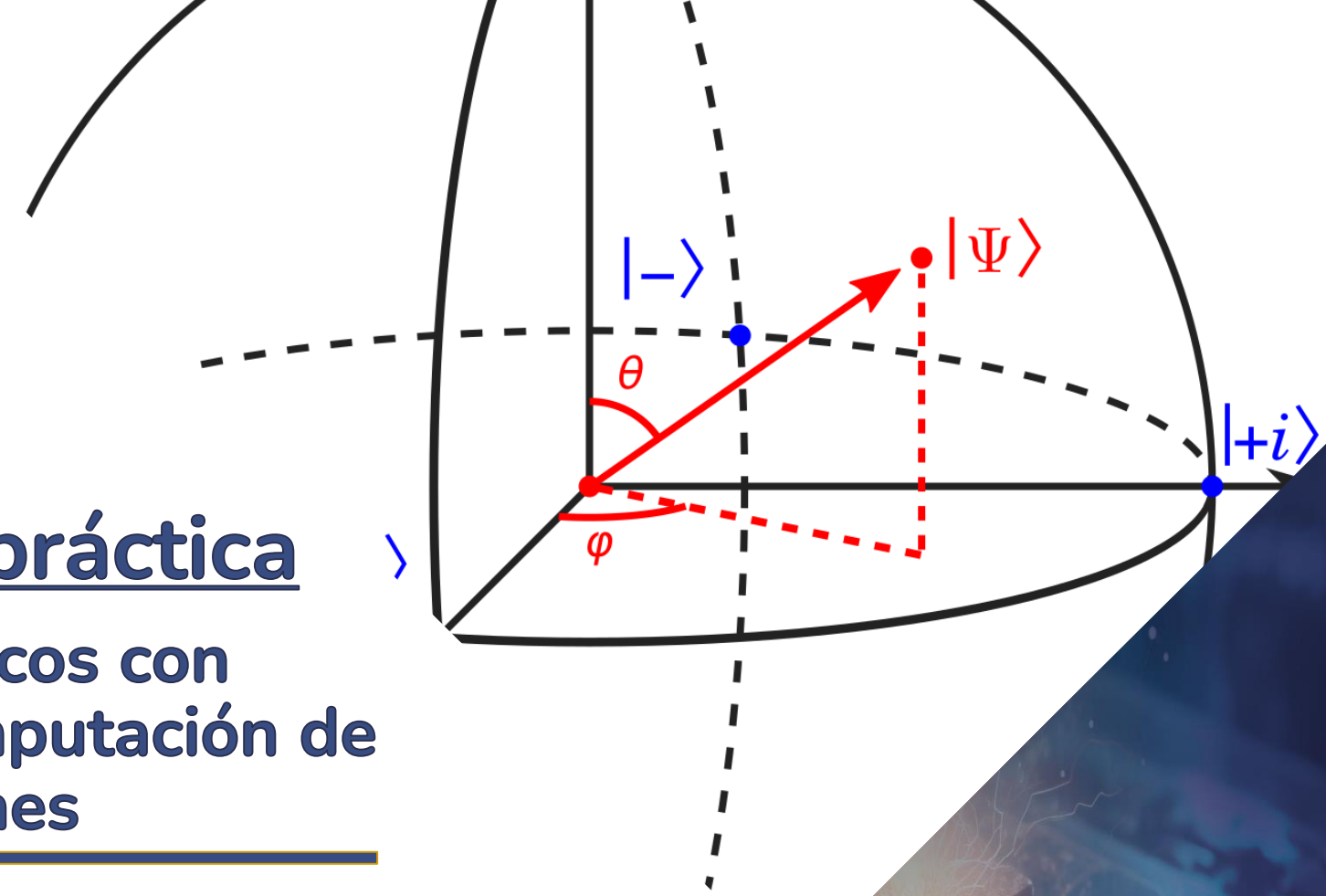


Simulación práctica

Circuitos cuánticos con técnicas de computación de altas prestaciones

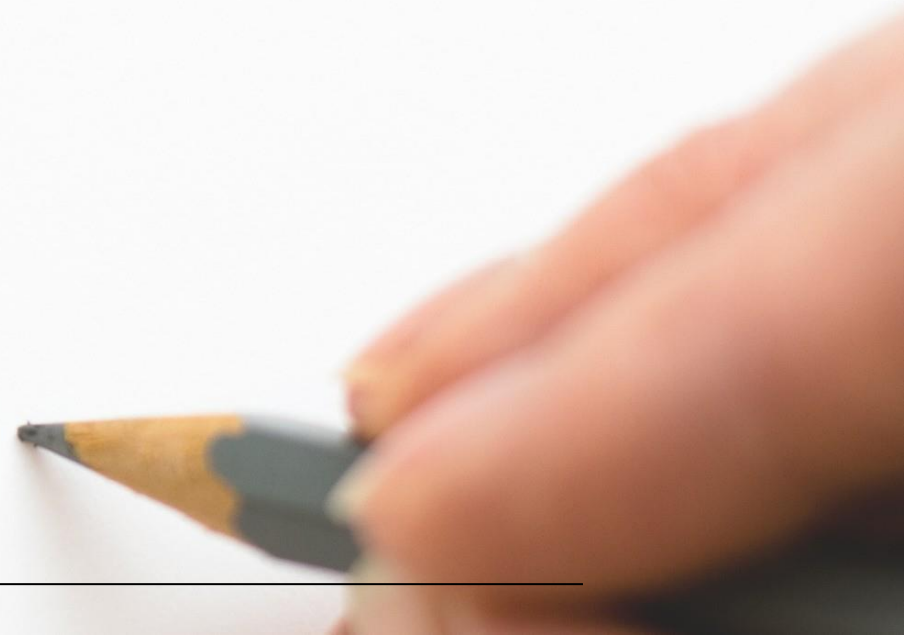
Manuel González - ITCL
Erik Skibinsky - ITCL

28-Noviembre-2023

competitividad empresarial ice | Junta de Castilla y León
Centr@Tec Servicios Avanzados de Innovación para Pymes
ITCL CENTRO TECNOLÓGICO
SCAYLE SUPERCOMPUTACIÓN CASTILLA Y LEÓN

Índice

- Simuladores de circuitos cuánticos
  - Visión computacional
- Gestión de recursos en simuladores cuánticos
  - Primera vía
  - Segunda vía
- Arquitectura de nuestro servidor
  - Front-End
  - Back-End
- Demo práctica
  - Simplicidad
  - Múltiples usuarios
  - Problemas de resolución cuántica



Simuladores de circuitos cuánticos

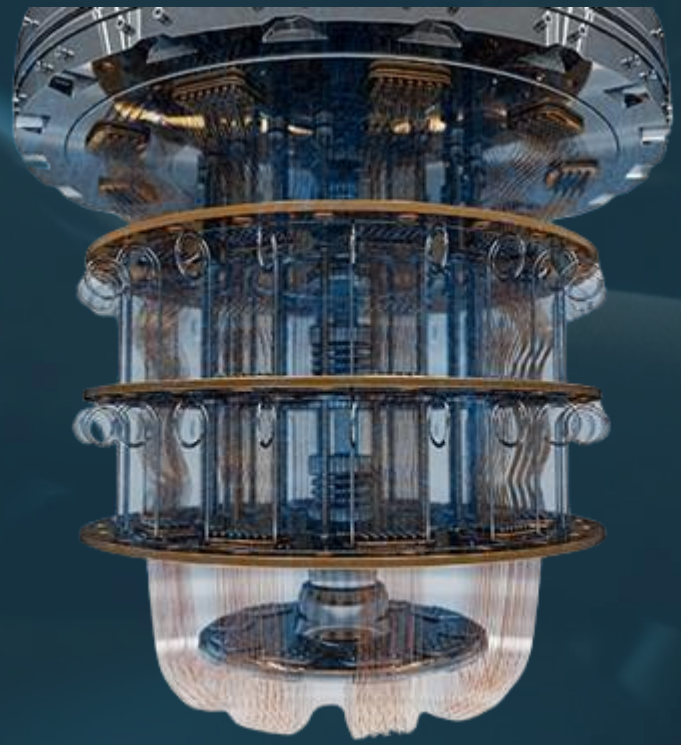
CENTRO TECNOLÓGICO

Simuladores de circuitos cuánticos

ITCL
CENTRO TECNOLÓGICO

Simuladores de circuitos cuánticos

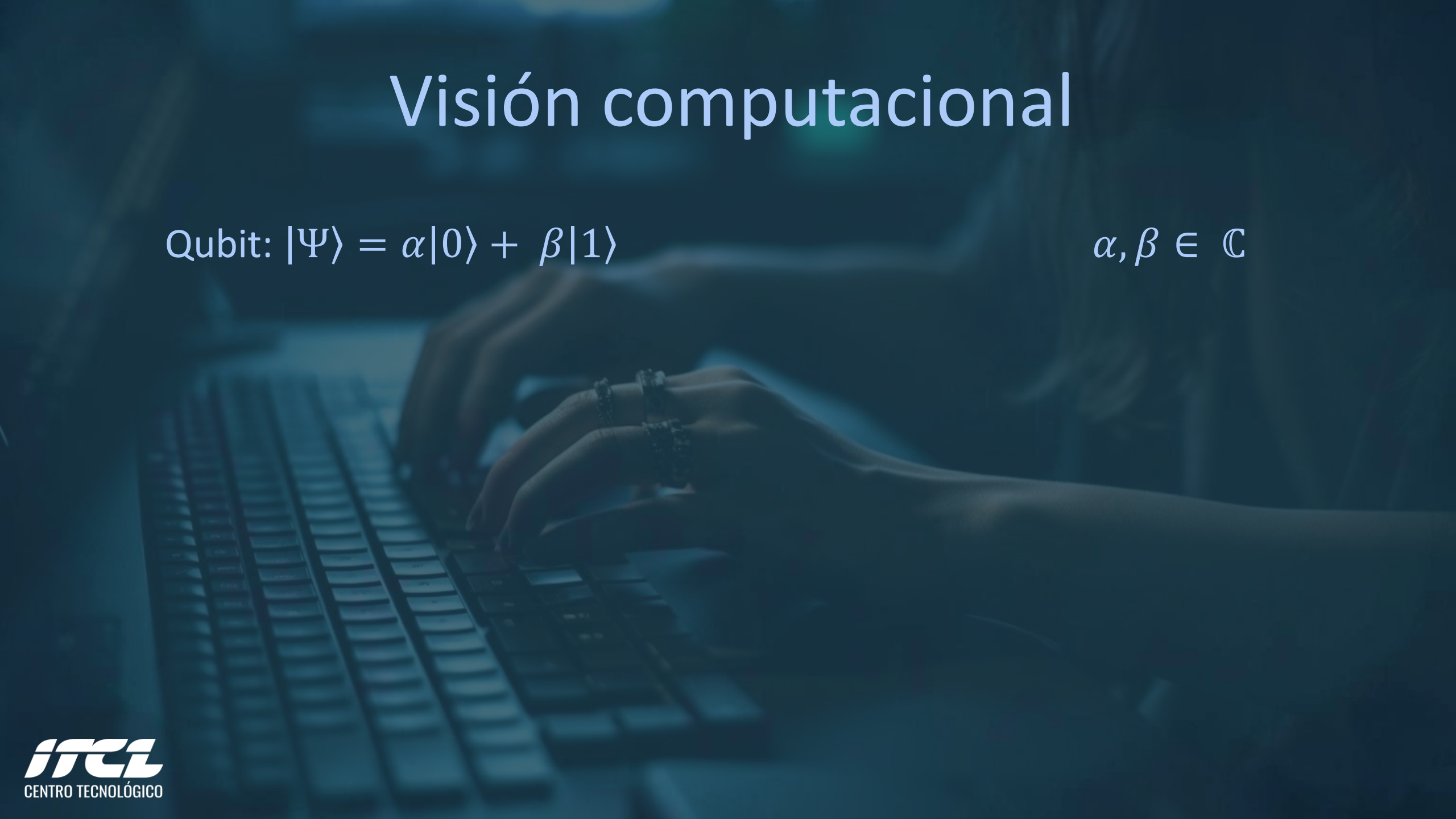
No confundir con ordenadores cuánticos



CENTRO TECNOLÓGICO

Visión computacional

Qubit: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ $\alpha, \beta \in \mathbb{C}$



CENTRO TECNOLÓGICO

Visión computacional

Qubit: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ $\alpha, \beta \in \mathbb{C}$

Puertas lógicas: $X = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $a, b, c, d \in \mathbb{C}$

CENTRO TECNOLÓGICO

Visión computacional

Qubit: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ $\alpha, \beta \in \mathbb{C}$

Puertas lógicas: $X = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $a, b, c, d \in \mathbb{C}$

Circuito cuántico: $X|\Psi\rangle = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

CENTRO TECNOLÓGICO

Visión computacional

Vectores tienen un tamaño exponencialmente creciente con el número de qubits.

Número de Qubits	Tamaño en memoria (hasta)
1	32 B
2	64 B
4	256 B
8	4 kB
16	1 MB
32	64 GB
64	256 EB (~200M de TB)

Visión computacional

Aplicar puertas lógicas, en su versión más simple, implica entonces multiplicar matrices

CENTRO TECNOLÓGICO

Visión computacional

Uso de GPU

- Gran memoria
- Rápida multiplicación de matrices

CENTRO TECNOLÓGICO

Visión computacional

Uso de GPU

- Gran memoria
- Rápida multiplicación de matrices

CENTRO TECNOLÓGICO

Gestión de recursos en simuladores cuánticos

Problema:

- Gestionar un número N de nodos de cálculo acelerado para un número U de usuarios

- Típicamente: **U > N**

CENTRO TECNOLÓGICO

Gestión de recursos en simuladores cuánticos

Primera vía:

- **Asignar** a cada usuario un subconjunto de recursos.

- Se realizan **particiones** de recursos y memoria.

Gestión de recursos en simuladores cuánticos

Segunda vía:

- **Separar** los nodos de cálculo del proceso de usuario

CENTRO TECNOLÓGICO

Gestión de recursos en simuladores cuánticos

En el ITCL hemos decidido adoptar la segunda vía:

- **Pros:**
  - Número arbitrariamente grande de usuarios para los mismos recursos.
  - Los usuarios pueden acceder a **nodos sin particionar**, con mucha memoria y potencia de cálculo
  - Aprovechamiento de los nodos de cómputo.
- **Contras:**
  - Se forman colas.
  - Gestión más compleja.

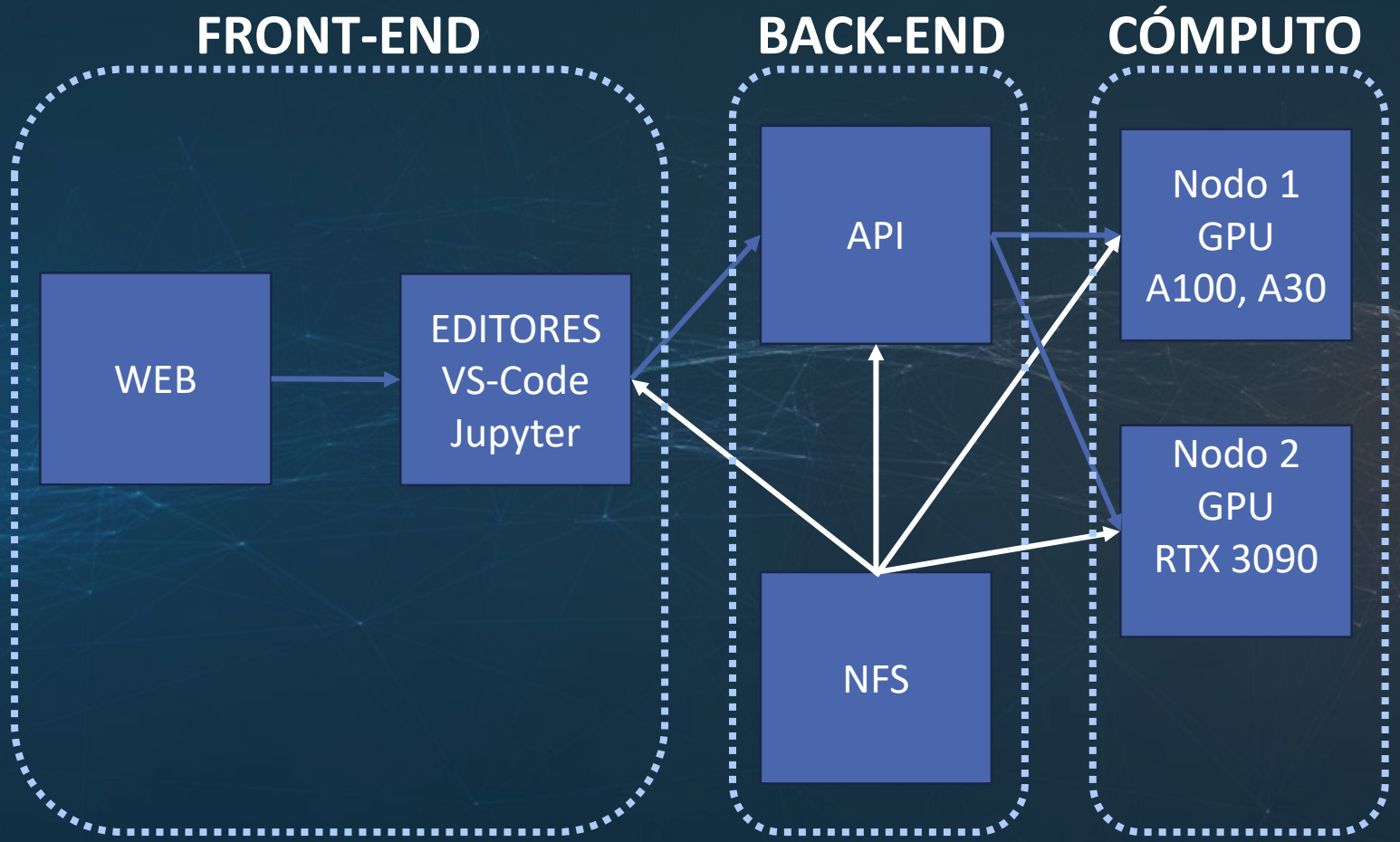
Gestión de recursos en simuladores cuánticos

En el ITCL hemos decidido adoptar la segunda vía

- ¿A qué nivel separamos el cómputo del usuario?:
  - **Usuario:** La separación no es transparente, el usuario debe adaptar su código.
  - **Librería:** Transparente al usuario, implementación ad hoc por librería.
  - **Driver:** Transparente para la librería, implementación difícil

Arquitectura de nuestro servidor

En el ITCL hemos decidido separar a nivel librería



CENTRO TECNOLÓGICO

Arquitectura de nuestro servidor

Front-End: Editores

Arquitectura de nuestro servidor

Back-End: API

- Distribuye trabajos a los nodos de cómputo mediante SLURM.
- Implementación ad-hoc para cada librería.
- Se ha realizado para: qiskit 0.4.0

backend_original.py

```
24 from qiskit.circuit import QuantumCircuit, ParameterExpression, Delay
25 from qiskit.compiler import assemble
26 from qiskit.providers import BackendV1 as Backend
27 from qiskit.providers.models import BackendStatus
28 from qiskit.pulse import Schedule, ScheduleBlock
29 from qiskit.qobj import QasmQobj, PulseQobj
30 from qiskit.result import Result
31 from qiskit.utils import deprecate_arguments
32 from ..aererror import AerError
33 from ..jobs import AerJob, AerJobSet, split_qobj
34 from ..noise.noise_model import NoiseModel, QuantumErrorLocation
35 from ..noise.errors.quantum_error import QuantumChannelInstruction
36 from .aer_compiler import compile_circuit
37 from .backend_utils import format_save_type, circuit_optypes
```

Snipped

backend_modificado.py

```
24 from qiskit.circuit import QuantumCircuit, ParameterExpression, Delay
25 from qiskit.compiler import assemble
26 from qiskit.providers import BackendV1 as Backend
27 from qiskit.providers.models import BackendStatus
28 from qiskit.pulse import Schedule, ScheduleBlock
29 from qiskit.qobj import QasmQobj, PulseQobj
30 from qiskit.result import Result
31 from qiskit.utils import deprecate_arguments
32 from ..aererror import AerError
33 from ..jobs import AerJob, AerJobSet, split_qobj
34 from itcl_job.itcljob import LocalJob as AerJob
35 from ..noise.noise_model import NoiseModel, QuantumErrorLocation
36 from ..noise.errors.quantum_error import QuantumChannelInstruction
37 from .aer_compiler import compile_circuit
38 from .backend_utils import format_save_type, circuit_optypes
```

Snipped

CENTRO TECNOLÓGICO

Arquitectura de nuestro servidor

Back-End: API

CENTRO TECNOLÓGICO

Arquitectura de nuestro servidor

Back-End: Discriminador

- Número de qubits
- Tamaño del circuito
- Estado del servidor y los recursos en uso

Se puede extender a otras políticas más complejas.

Demo Práctica

CENTRO TECNOLÓGICO

Manuel González
manuel.gonzalez@itcl.es

Erik Skibinsky
erik.skibinsky@itcl.es