

# Tutorial 1 – IBM Q Experience Composer

Luís Paulo Santos, Novembro, 2020

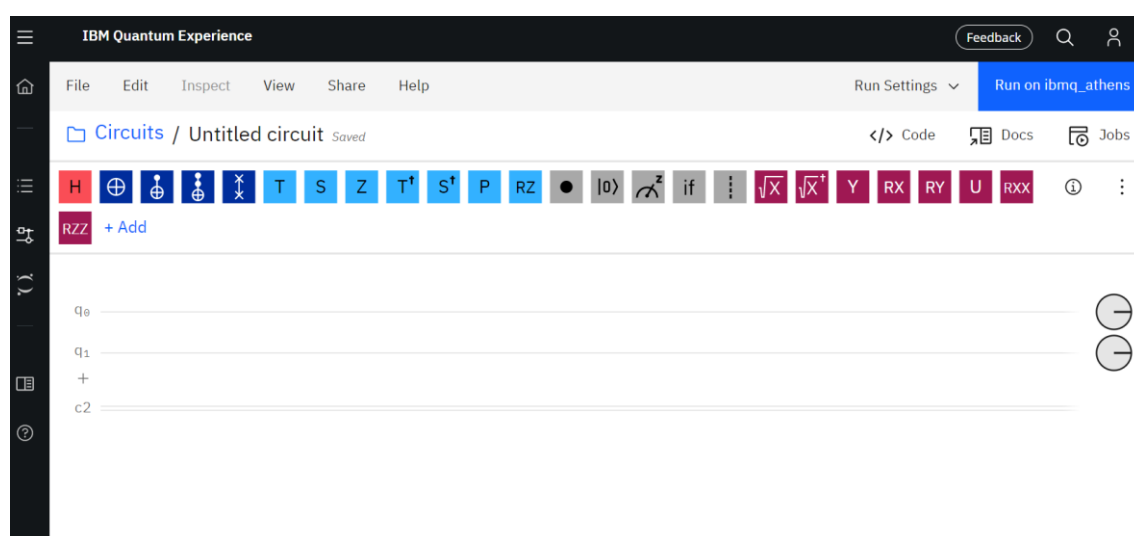
## 1 Registo e Composer

(gravação -- sem som por erro de operação -- desta secção em <https://eu-lti.bbcollab.com/recording/0d89122bc2124b79ba8082c19a572db8> )

Crie a sua conta em <https://quantum-computing.ibm.com/> .

Na barra da esquerda seleccione o COMPOSER

Deve agora ser capaz de visualizar a janela do COMPOSER:

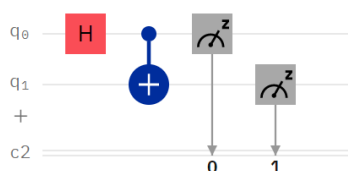


## 2 Bell State

(gravação desta secção em <https://eu-lti.bbcollab.com/recording/6323e1ca2b23454dadbc160bbc98b600> )

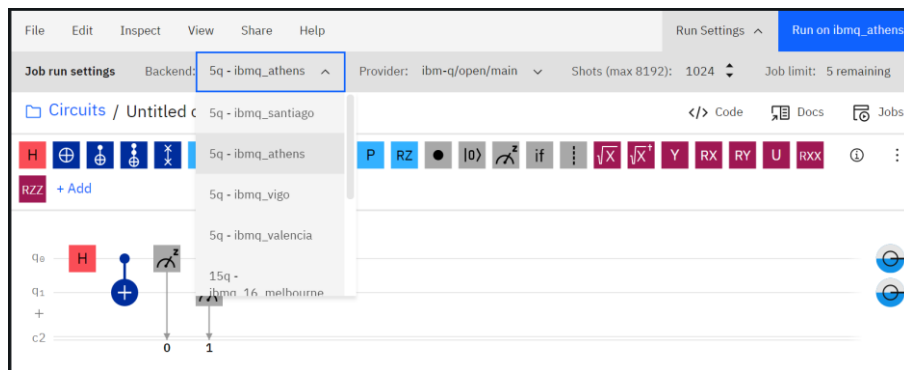
(gravação dos resultados finais da execução do job desta secção em <https://eu-lti.bbcollab.com/recording/b5f1ac8dd68e4bbebe03679642b2deb4> )

Desenhe o circuito que prepara um *Bell state* conforme foi apresentado na aula:



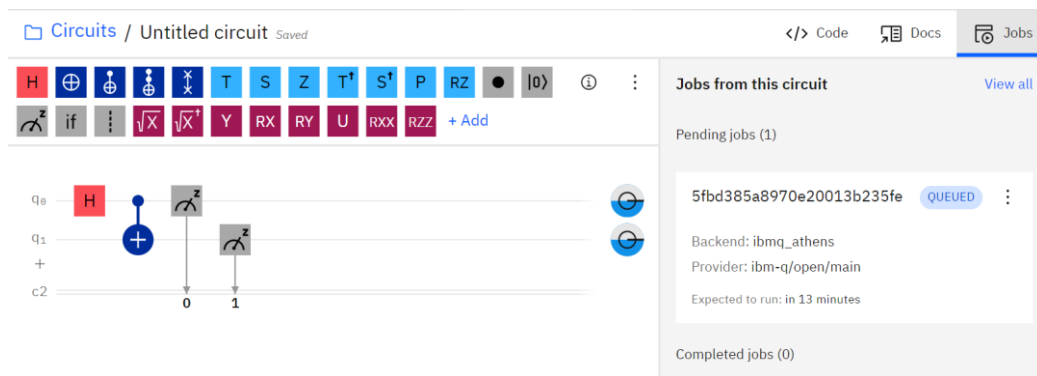
No menu “View” seleccione “Measurement probabilities”. Ajuste a janela de forma a conseguir visualizar o gráfico. Pode consultar este gráfico abaixo na Tabela 1. Comente os resultados obtidos.

No menu “Run settings” seleccione um *backend* real, por exemplo: `ibmq_athens`



Certifique-se que, tal como na imagem anterior, o número de *shots* é elevado, por exemplo, 1024. Quer isto dizer que o circuito será executado 1024 vezes, para que obtenhamos uma distribuição de probabilidade sobre as medições.

Selecione a opção “Run on `ibmq_athens`”. Deverá visualizar algo semelhante a :



Aguarde calmamente que o seu *job* seja executado.

Assim que o *job* terminar seleccione-o e observe o gráfico com a distribuição das medições sobre os estados possíveis.

*Tabela 1-Resultados para o Bell State*

<p>Measurement Probabilities ▾</p> <p>Measurement probability (%)</p> <p>00 11</p>	<p>Result</p> <p>Histogram</p> <p>Measurement probability (%)</p> <p>00 01 10 11</p> <p>Computational basis states</p>
<p>Distribuição de probabilidade obtida com o simulador (sem ruído)</p>	<p>Medições efectivamente observadas na máquina <code>ibmqx2</code> (note que estes resultados não foram obtidos na máquina <code>ibmq_athens</code> conforme indicado acima - experimente executar nesse backend e compare)</p>

Os resultados obtidos demonstram claramente que a computação é contaminada com ruído. Dado o actual estado de desenvolvimento da tecnologia é necessário recorrer a técnicas de tolerância a erros para obter resultados com alguma fiabilidade.

### 3 Oráculo

(gravação desta secção em <https://eu-lti.bbcollab.com/recording/b88a98b241254af684ac6c338af4707d> )

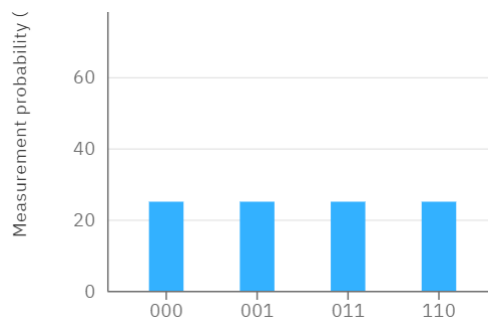
Consegue desenhar um circuito que opera sobre uma sobreposição de 2 qubits e coloque a 1 um 3º qubit apenas para o estado “01” da sobreposição?

Este tipo de circuitos é normalmente designado por **oráculo**, uma vez que marcam os estados considerados relevantes. Neste caso a marcação do estado “01” é conseguida colocando o qubit 2 a 1 apenas para esse estado.

Tente desenhar o circuito você mesmo/a antes de ver a solução na página seguinte (Figura 1). Tome em consideração os seguintes aspectos:

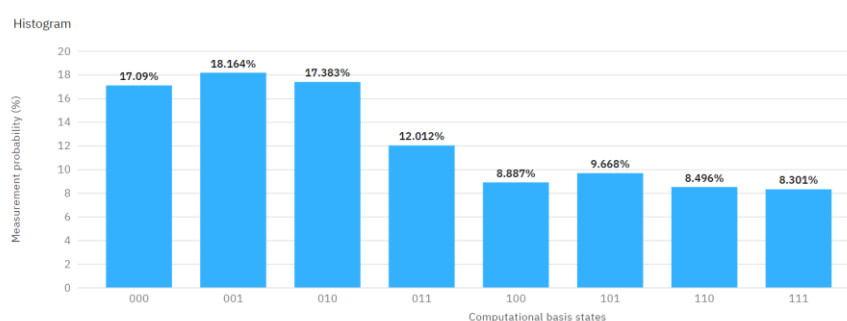
- precisamos de 3 qubits
- os qubits q0 e q1 devem ser colocados numa sobreposição usando duas gates de Hadamard. A sobreposição resultante é uma combinação linear dos 4 estados 00, 01, 10 e 11, cada com peso  $\frac{1}{2}$  (a probabilidade de medição é o quadrado do peso, ou seja,  $\frac{1}{4}$ ).
- Podemos controlar o 3º qubit usando a gate CC-NOT (Toffoli gate) que inverte o 3º qubit se os 2 qubits de controlo forem iguais a 1 (estado 11).

A figura abaixo apresenta, para o caso ideal sem ruído (simulação), a distribuição de probabilidades sobre os 4 estados que podem ser medidos:



Conforme esperado o qubit 2 (o da esquerda, mais significativo) só é colocado a 1 quando  $|q_1q_0\rangle = |10\rangle$ .

A máquina ibmqx2 costuma estar bastante disponível (pouco tempo de espera), mas tratando-se de uma máquina mais antiga tem elevados níveis de ruído. Os resultados de 1024 shots nesta máquina são os seguintes:



O qubit q2 aparece a 1 para todos os estados possíveis de  $|q_1q_0\rangle$ , fruto do ruído nas medições. Experimente executar este programa na máquina ibmq\_athens para verificar a sua robustez.

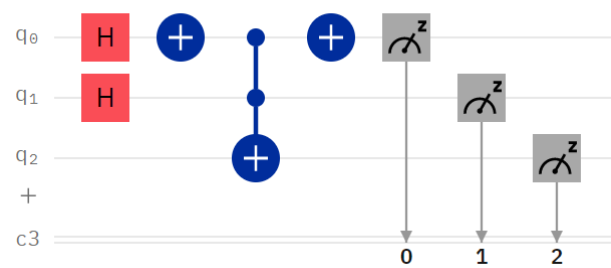


Figura 1- Oráculo:  $q_2$  a 1 quando  $q_1$  e  $q_0$  são 1 0