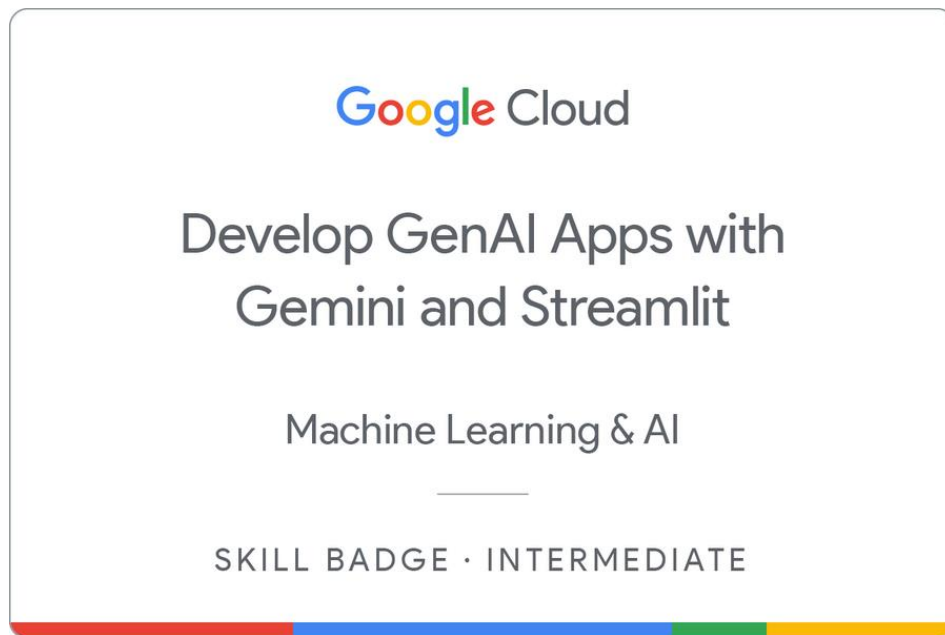


MATERIAL DE APOIO

Google Cloud Skill Boost

Laboratório 13 do Intermediate: Develop GenAI Apps with Gemini and Streamlit: laboratório com desafio



Sumário

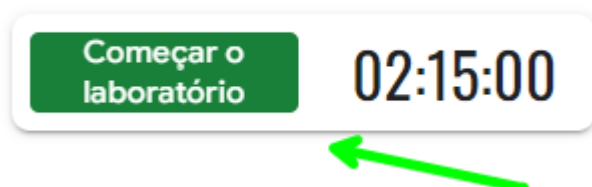
1.	Login no Console do Google Cloud	3
2.	Tarefa 1	5
3.	Tarefa 2	9
4.	Tarefa 3	12
5.	Tarefa 4	14
6.	Tarefa 5	15

1. Login no Console do Google Cloud

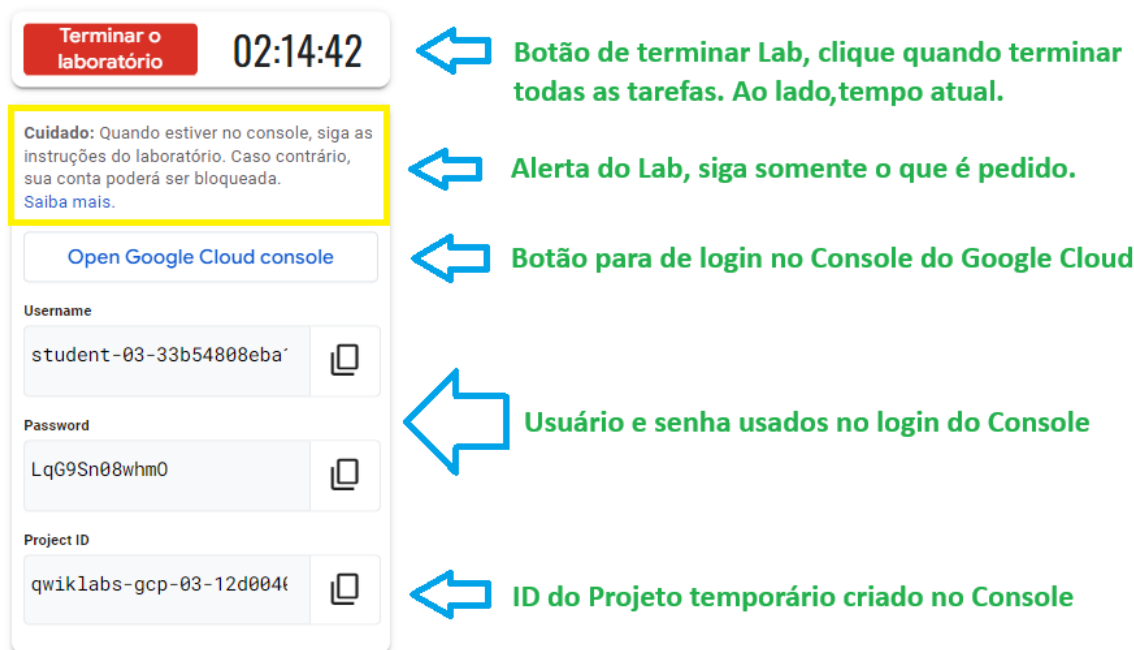
Link para o laboratório:

[https://www.cloudskillsboost.google/paths/236/course_templates/978/labs/488168?locale=pt_B
R](https://www.cloudskillsboost.google/paths/236/course_templates/978/labs/488168?locale=pt_BR)

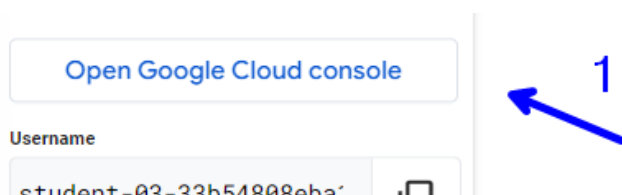
- 1) Primeiro passo é fazer login no Console do Google Cloud, clique no botão verde “Começar o laboratório”:

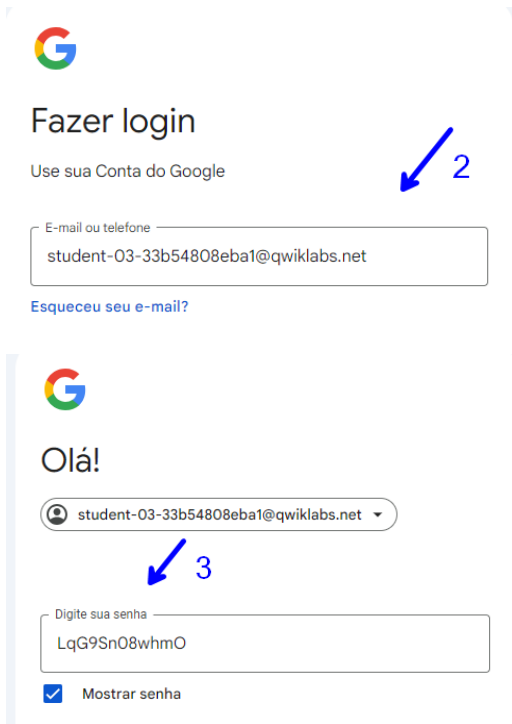


- 2) Após, irá abrir o Painel de login no Console:



- 3) Clique no botão “Open Google Cloud Console” (1) com o botão direito do mouse e escolha para abrir em uma aba Anônima/InPrivate, depois preencha o usuário (2) e senha (3) na página de login:





Fazer login

Use sua Conta do Google

E-mail ou telefone
student-03-33b54808eba1@qwiklabs.net

[Esqueceu seu e-mail?](#)

Olá!

student-03-33b54808eba1@qwiklabs.net

Digite sua senha
LqG9Sn08whmO

☒ Mostrar senha

4) Aceite todos os termos e condições do Google Cloud (1-2):

em myaccount.google.com.

Seu uso dos Serviços do Google com esta conta também é regido por políticas internas da sua organização.

Entendi

Google Cloud

Welcome student fe2a879d!

Create and manage your Google Cloud instances, disks, networks, and other resources in one place.



student fe2a879d

student-03-33b54808eba1@qwiklabs.net

[SWITCH ACCOUNT](#)

Country

Brazil

Terms of Service

☒ I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#).

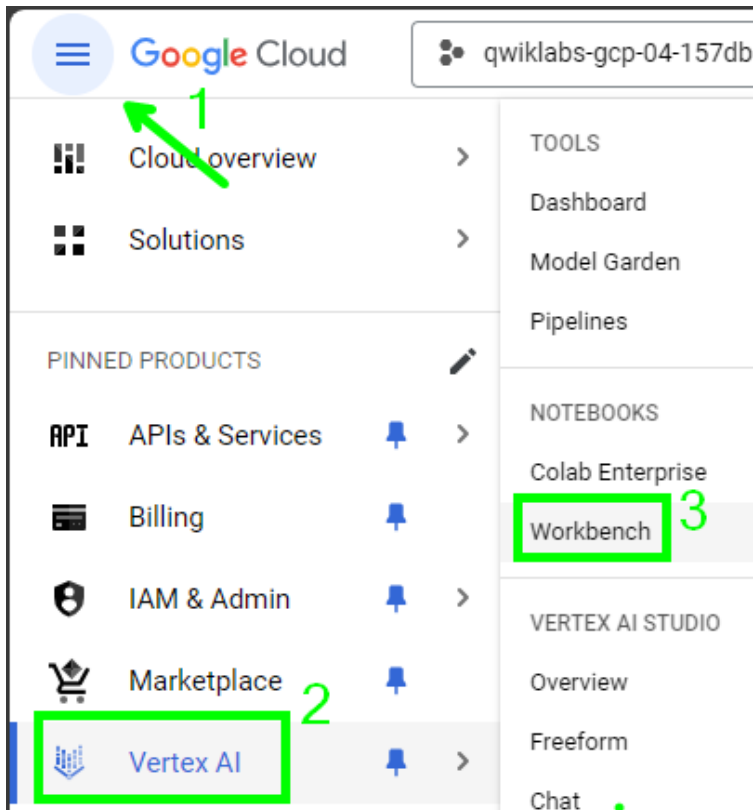
Email updates

☐ I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners.

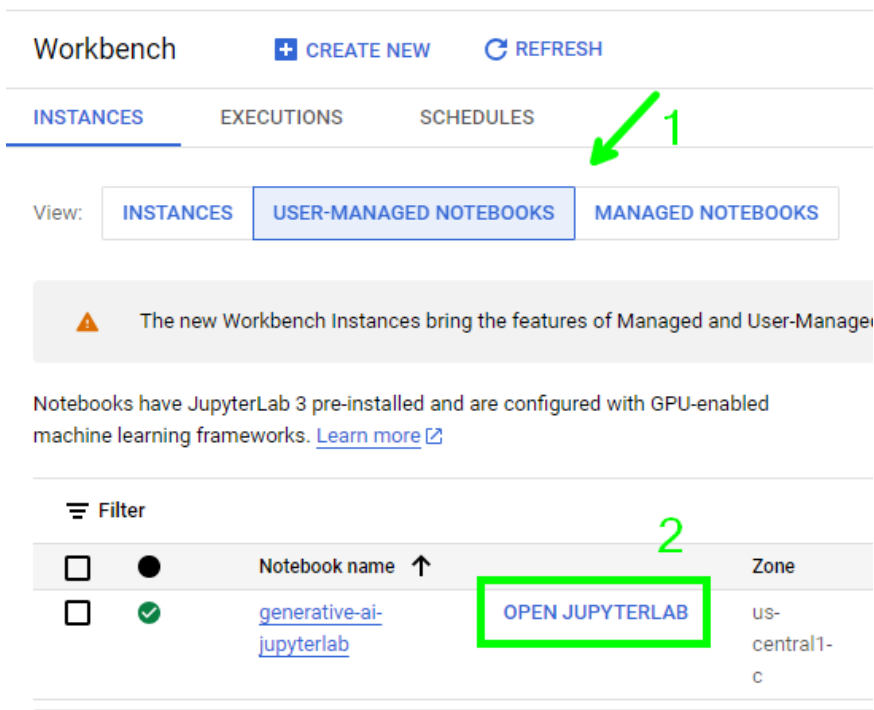
AGREE AND CONTINUE

2. Tarefa 1

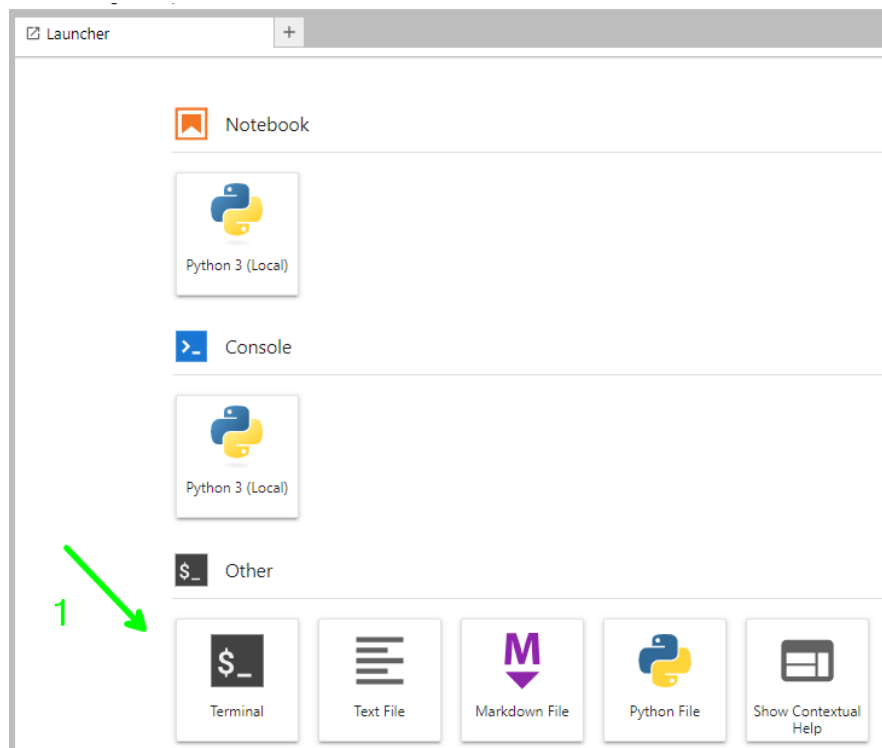
1) Depois que entrar no Console, clique no Menu (1) no canto superior esquerdo, procure o recurso “Vertex AI” (2), e clique em “Workbench” (3):



2) Em **Workbench**, clique na aba do menu **USER-MANAGED NOTEBOOKS** (1), e depois em **OPEN JUPYTERLAB** (2).



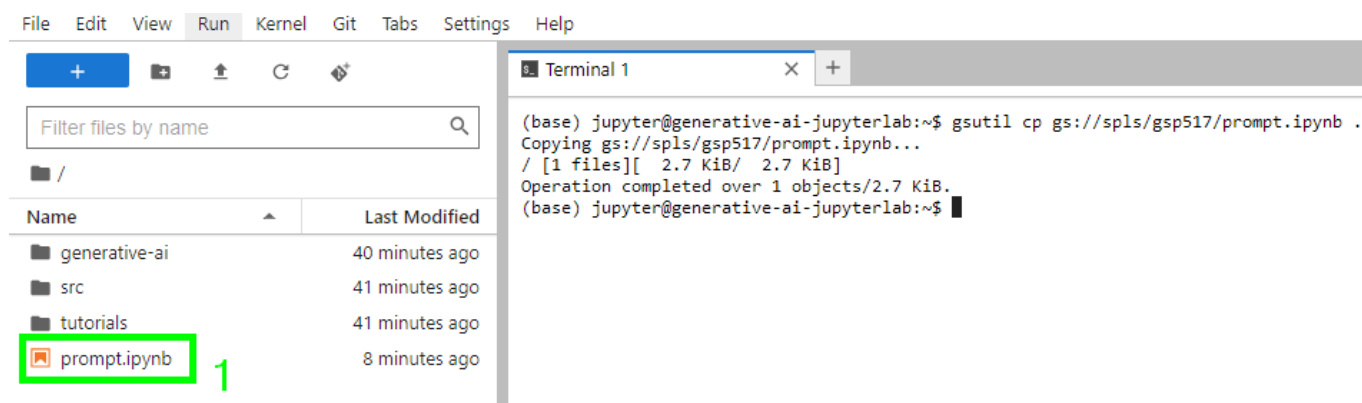
3) Clique em **Terminal (1)**:



4) No **Terminal**, execute o seguinte comando:

`gsutil cp gs://spl/s/gsp517/prompt.ipynb .`

5) Dê dois cliques no arquivo **prompt.ipynb** no explorador de arquivos à esquerda do terminal que foi baixado no passo anterior:



6) Na terceira célula, escreva o **PROJECT_ID (1)** e **LOCATION (2)** nos valores de variáveis correspondentes (Dentro das aspas duplas). Esses valores podem ser consultados no **Painel**

do Laboratório, como visto no **Item 1.2** desse Material de Apoio:

```
[ ]: ! pip3 install --upgrade --user google-cloud-aiplatform

[ ]: import IPython

app = IPython.Application.instance()
app.kernel.do_shutdown(True)

[ ]: # Task 1.3
# Modify these variables to include your project ID and region.
PROJECT_ID = "quiklabs-gcp-04-157db23d0309"
LOCATION = "us-central1"

[ ]: import os

os.environ["PROJECT_ID"] = PROJECT_ID
os.environ["LOCATION"] = LOCATION
os.environ["API_ENDPOINT"] = f"{LOCATION}-aiplatform.googleapis.com"

[ ]: %bash

# Task 1.4
# Use cURL to test a prompt with the API, by modifying prompt with the prompt from the Lab guide.

MODEL_ID="gemini-pro"

curl -X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
https://$API_ENDPOINT/v1/projects/$PROJECT_ID/locations/$LOCATION/publishers/google/models/$MODEL_ID:streamGenerateContent \
-d '{
  "contents": {
    "role": "USER",
    "parts": { "text": "Why is the sky blue?" }
  }
}'
```

7) Na célula 5, troque o prompt dentro das aspas duplas “Why is the sky blue?” (Procure com Ctrl+F se precisar) por (Copie somente o prompt, sem a seta):

→ I am a Chef. I need to create Japanese recipes for customers who want low sodium meals. However, I do not want to include recipes that use ingredients associated with a peanuts food allergy. I have ahi tuna, fresh ginger, and edamame in my kitchen and other ingredients. The customer wine preference is red. Please provide some for meal recommendations. For each recommendation include preparation instructions, time to prepare and the recipe title at the beginning of the response. Then include the wine paring for each recommendation. At the end of the recommendation provide the calories associated with the meal and the nutritional facts.

```
[ ]: ! pip3 install --upgrade --user google-cloud-aiplatform

[ ]: import IPython

app = IPython.Application.instance()
app.kernel.do_shutdown(True)

[ ]: # Task 1.3
# Modify these variables to include your project ID and region.
PROJECT_ID = "quiklabs-gcp-04-157db23d0309"
LOCATION = "us-central1"

[ ]: import os

os.environ["PROJECT_ID"] = PROJECT_ID
os.environ["LOCATION"] = LOCATION
os.environ["API_ENDPOINT"] = f"{LOCATION}-aiplatform.googleapis.com"

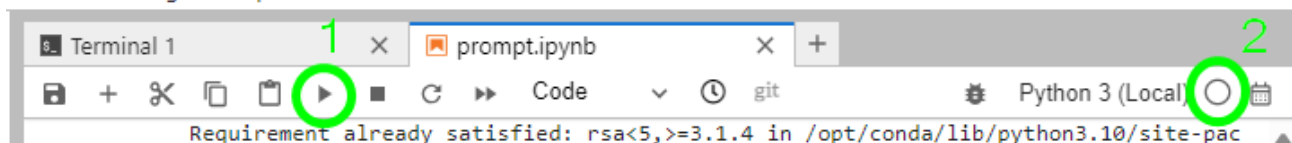
[ ]: %bash

# Task 1.4
# Use cURL to test a prompt with the API, by modifying prompt with the prompt from the Lab guide.

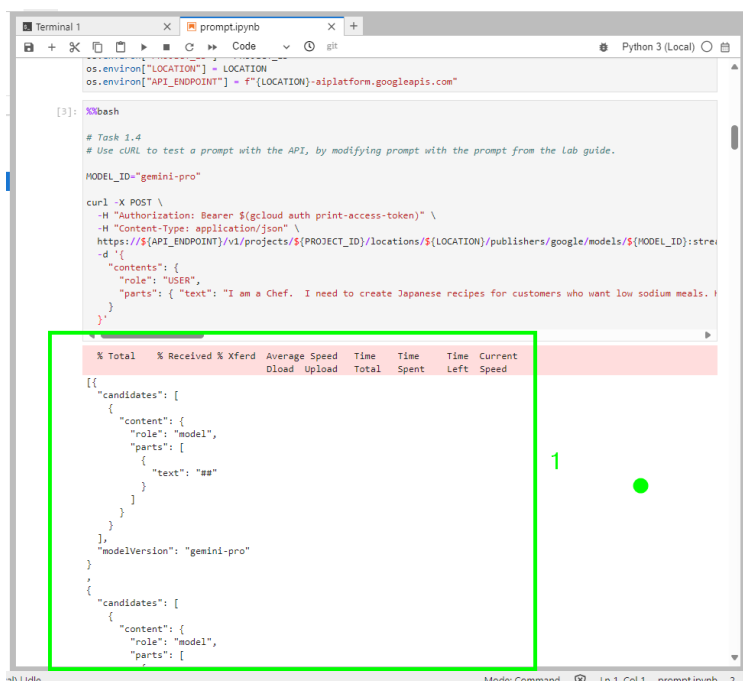
MODEL_ID="gemini-pro"

curl -X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
https://$API_ENDPOINT/v1/projects/$PROJECT_ID/locations/$LOCATION/publishers/google/models/$MODEL_ID:streamGenerateContent \
-d '{
  "contents": {
    "role": "USER",
    "parts": { "text": "I am a Chef. I need to create Japanese recipes for customers who want low sodium meals. However, I do not want to incl
```

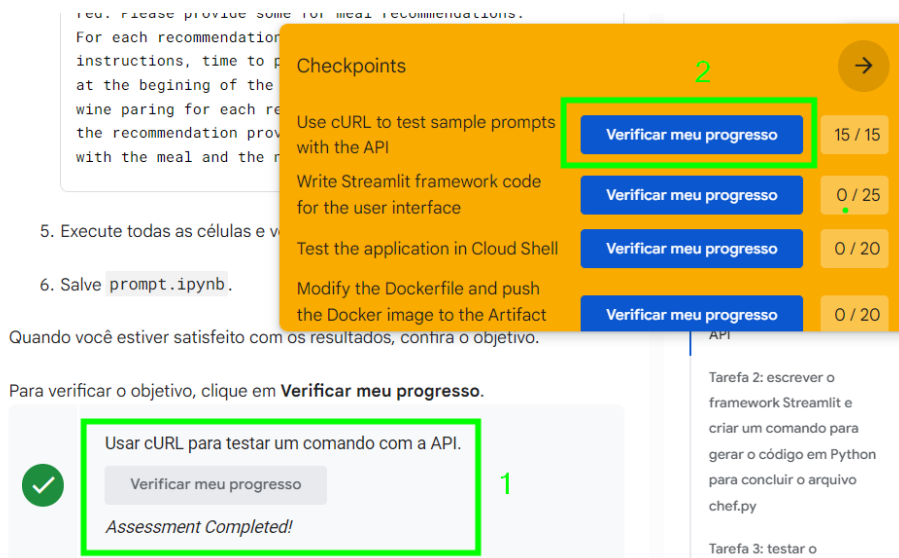
- 8) Execute a célula uma a uma apertando o símbolo de ► no menu superior (1), e só siga para a próxima se o Kernel estiver **Ocioso (Idle)**, indicado quando o círculo estiver branco (2):



- 9) A saída deve ser uma resposta em formato JSON (1), se der tudo ok, salve o arquivo com CTRL+S:

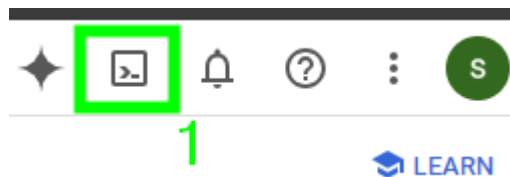


- 10) Tique o botão de progresso (1), e primeira tarefa na aba amarela expansível no canto superior direito (2):

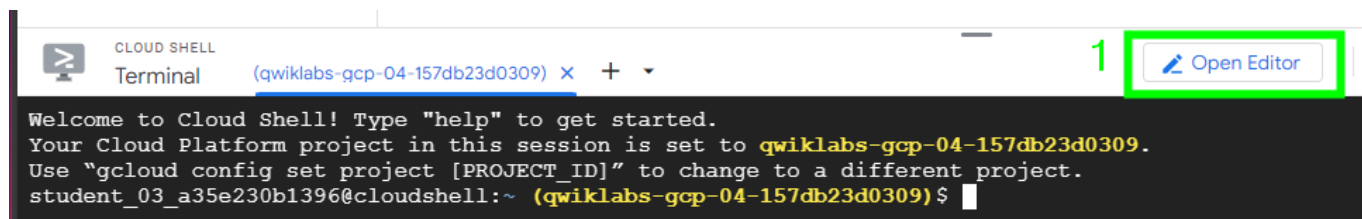


3. Tarefa 2

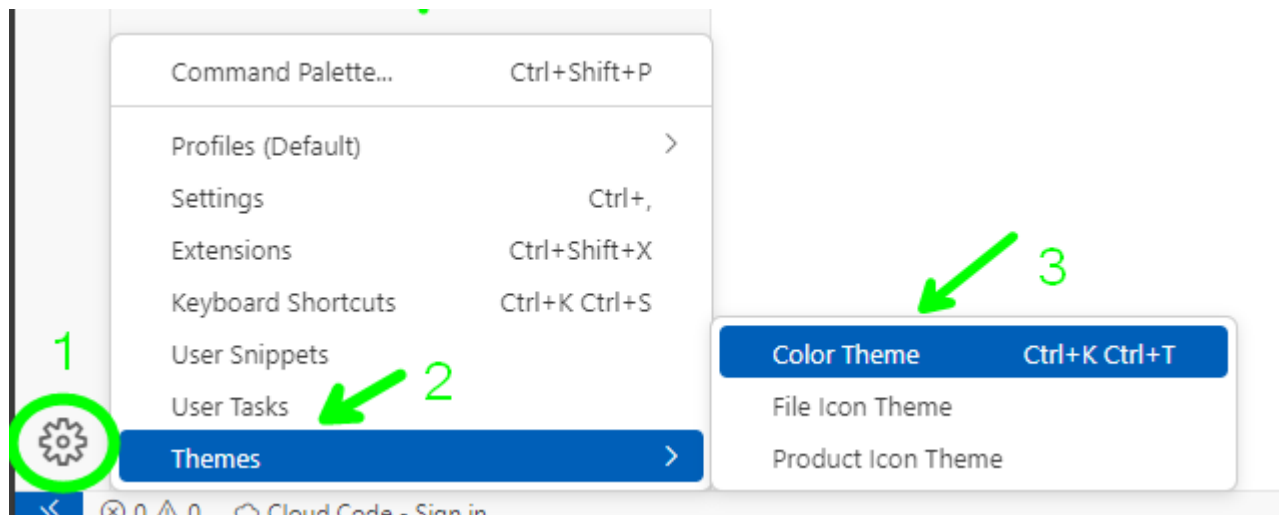
1) Volte para o **Console do GCP** (Onde estava na página do Workbench), no canto superior direito clique em **Cloud Shell** (Ao lado do botão de perfil do estudante) (1), e aperte Continue depois que abrir o terminal:



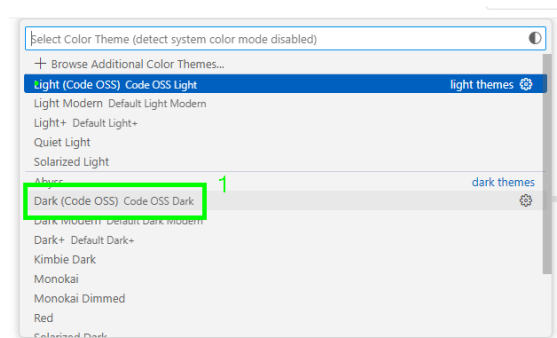
2) Depois que o **Terminal** carregar, clique em **Open Editor** (1):



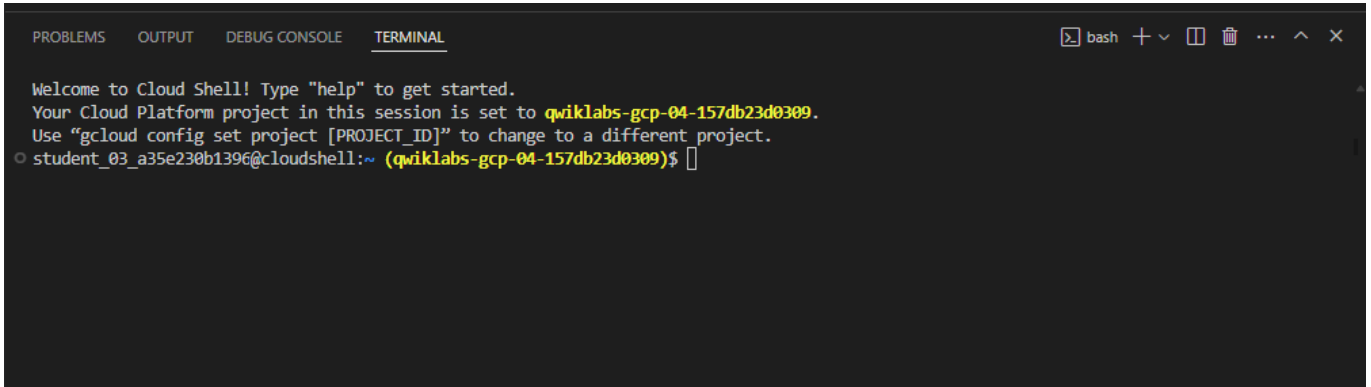
3) Assim que o Editor carregar, clique na engrenagem no canto inferior esquerdo (1), depois em **Themes** (2), e em seguida em **Color Themes** (3):



4) Clique em **Dark (Code OSS)** (1):



5) Aperte CTRL+J para abrir o Terminal (Confirme que está no projeto correto):



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-04-157db23d0309.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
student_03_a35e230b1396@cloudshell:~ (qwiklabs-gcp-04-157db23d0309)$

```

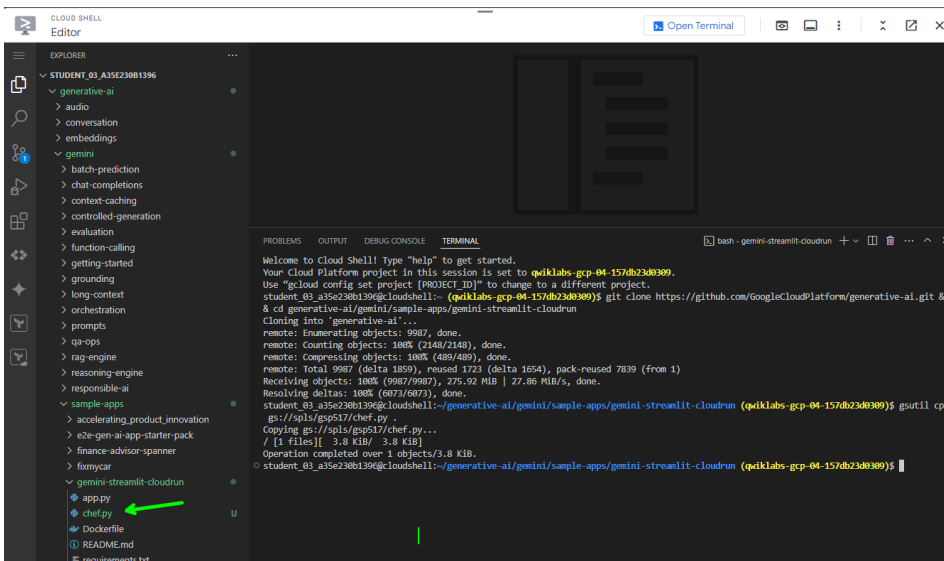
6) Execute o seguinte comando (mantenha-se nesse diretório até o final), clique em **Authorize** se precisar:

```
git clone https://github.com/GoogleCloudPlatform/generative-ai.git
```

```
cd generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun
```

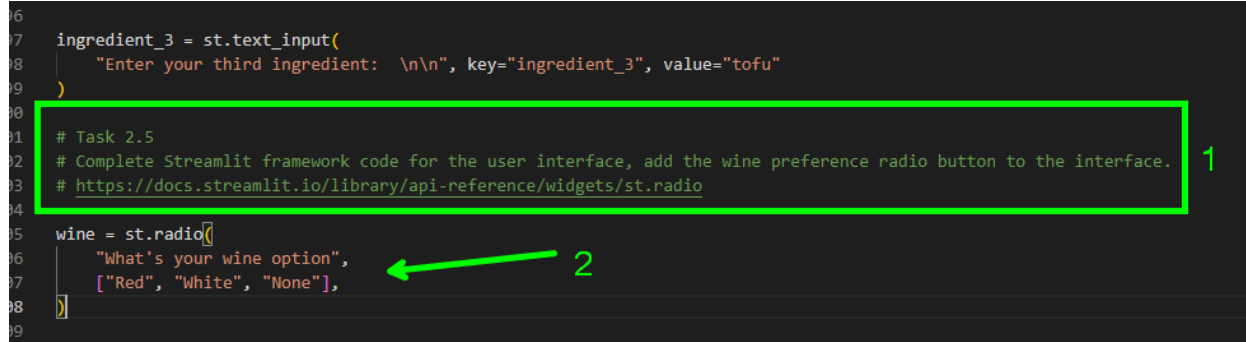
```
gsutil cp gs://spl/spl517/chef.py .
```

7) À esquerda, no explorador de arquivos, verá que o arquivo chef.py foi adicionado ao diretório atual, clique nele:



8) Abaixo da parte comentada como **Task 2.5** (1), coloque o código a seguir (2) como mostrado na figura:

```
wine = st.radio(
    "What's your wine option",
    ["Red", "White", "None"],
)
```



```

06
07 ingredient_3 = st.text_input(
08     "Enter your third ingredient: \n\n", key="ingredient_3", value="tofu"
09 )
10
11 # Task 2.5
12 # Complete Streamlit framework code for the user interface, add the wine preference radio button to the interface. 1
13 # https://docs.streamlit.io/library/api-reference/widgets/st.radio
14
15 wine = st.radio(
16     "What's your wine option",
17     ["Red", "White", "None"],
18 )
19

```

9) Abaixo da parte comentada **Task 2.6** (1), inclua o prompt a seguir (2) (depois dessas alterações salve o arquivo com CTRL+S):

→ prompt = f"""I am a Chef. I need to create {cuisine} \n
 recipes for customers who want {dietary_preference} meals. \n
 However, don't include recipes that use ingredients with the customer's {allergy} allergy. \n
 I have {ingredient_1}, \n
 {ingredient_2}, \n
 and {ingredient_3} \n
 in my kitchen and other ingredients. \n
 The customer's wine preference is {wine} \n
 Please provide some for meal recommendations.
 For each recommendation include preparation instructions,
 time to prepare
 and the recipe title at the beginning of the response.
 Then include the wine paring for each recommendation.
 At the end of the recommendation provide the calories associated with the meal
 and the nutritional facts.

"""

```

# Task 2.6
# Modify this prompt with the custom chef prompt.
prompt = f"""I am a Chef. I need to create {cuisine} \n
recipes for customers who want {dietary_preference} meals. \n
However, don't include recipes that use ingredients with the customer's {allergy} allergy. \n
I have {ingredient_1}, \n
{ingredient_2}, \n
and {ingredient_3} \n
in my kitchen and other ingredients. \n
The customer's wine preference is {wine} \n
Please provide some for meal recommendations.
For each recommendation include preparation instructions,
time to prepare
and the recipe title at the beginning of the response.
Then include the wine paring for each recommendation.
At the end of the recommendation provide the calories associated with the meal
and the nutritional facts.
"""

```

10) Execute o seguinte comando no **Terminal**, troque a parte do Project ID para o que o que foi provisionado para você:

DICA: Na página do laboratório, o comando aparece pronto para ser copiado.

→ `gcloud storage cp chef.py gs://qwiklabs-gcp-04-157db23d0309-generative-ai/`

11) Aperte o botão de verificar progresso (1) e o botão de progresso na aba amarela (2):

8. Salve o arquivo `chef.py`.

Quando o resultado do código do `chef.py` for satisfatório, faça upload do `chef.py` para o bucket `gcp-04-157db23d0309-generative-ai` no Cloud Shell. No console do Cloud, clique em **Verificar meu progresso** na aba amarela.

`gcloud storage cp chef.py gs://157db23d0309-generative-ai/`

Checkpoints	
Use cURL to test sample prompts with the API	15 / 15
Write Streamlit framework code for the user interface	25 / 25
Test the application in Cloud Shell	0 / 20
Modify the Dockerfile and push the Docker image to the Artifact Registry	0 / 20

Para verificar o objetivo, clique em **Verificar meu progresso**.

Escrever o framework Streamlit e criar um comando para gerar o código em Python para concluir o arquivo `chef.py`

Verificar meu progresso (1)

Assessment Completed!

Tarefa 2: escrever o framework Streamlit e criar um comando para gerar o código em Python para concluir o arquivo `chef.py`

Tarefa 3: testar o

4. Tarefa 3

1) Verifique se você ainda está neste caminho: `generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun`, e execute o seguinte comando no **Terminal**:

→ `python3 -m venv gemini-streamlit`

`source gemini-streamlit/bin/activate`

`pip install -r requirements.txt`

2) Ainda no **Terminal**, execute o seguinte comando (Não copie a seta) com os valores de variáveis dados a você no **Painel do Laboratório** dentro das aspas simples (depois dessas alterações salve o arquivo com CTRL+S):

GCP_PROJECT='Project ID' #Troque pelo valor que aparece para você

GCP_REGION='Region' #Troque pelo valor que aparece para você

3) Execute o seguinte comando:

```
streamlit run chef.py \
  --browser.serverAddress=localhost \
  --server.enableCORS=false \
  --server.enableXsrfProtection=false \
  --server.port 8080
```

4) Clique na URL da saída do comando **(1)** com CTRL+Botão Esquerdo do Mouse:

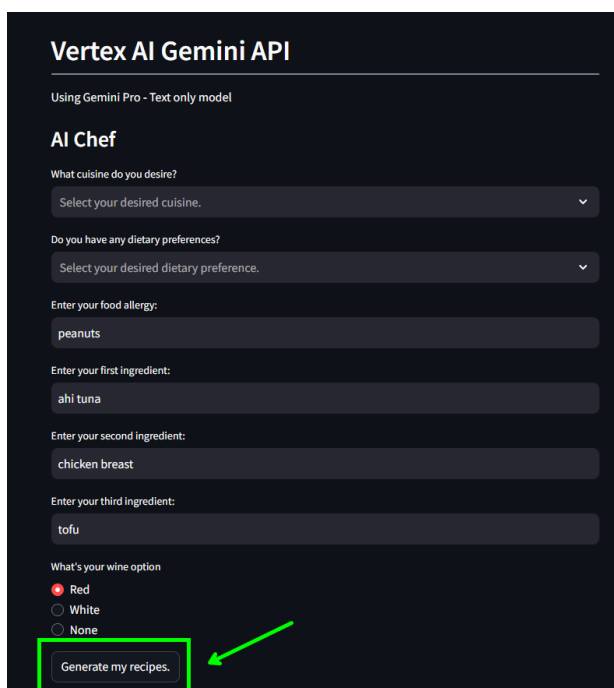


```
(gemini-streamlit) student_03_a35e230b1396@cloudshell:~/generative-ai/gemini/sample-apps/gemini-streamlit-cloudrun (quiklabs-gcp-04-157db23d0309)$ streamlit run ch
ef.py \
  --browser.serverAddress=localhost \
  --server.enableCORS=false \
  --server.enableXsrfProtection=false \
  --server.port 8080

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.

You can now view your Streamlit app in your browser.
URL: http://localhost:8080 1
```

5) Teste a aplicação clicando no botão **Generate my recipes** na parte debaixo da página:



Vertex AI Gemini API

Using Gemini Pro - Text only model

AI Chef

What cuisine do you desire?
Select your desired cuisine.

Do you have any dietary preferences?
Select your desired dietary preference.

Enter your food allergy:
peanuts

Enter your first ingredient:
ahi tuna

Enter your second ingredient:
chicken breast

Enter your third ingredient:
tofu

What's your wine option
☒ Red
☐ White
☐ None

Generate my recipes.

6) Se estiver tudo correto, tique o progresso do laboratório no corpo (1) e aba amarela (2):

Checkpoints

Checkpoint	Progress
Use cURL to test sample prompts with the API	15 / 15
Write Streamlit framework code for the user interface	25 / 25
Test the application in Cloud Shell	20 / 20

Verificar meu progresso

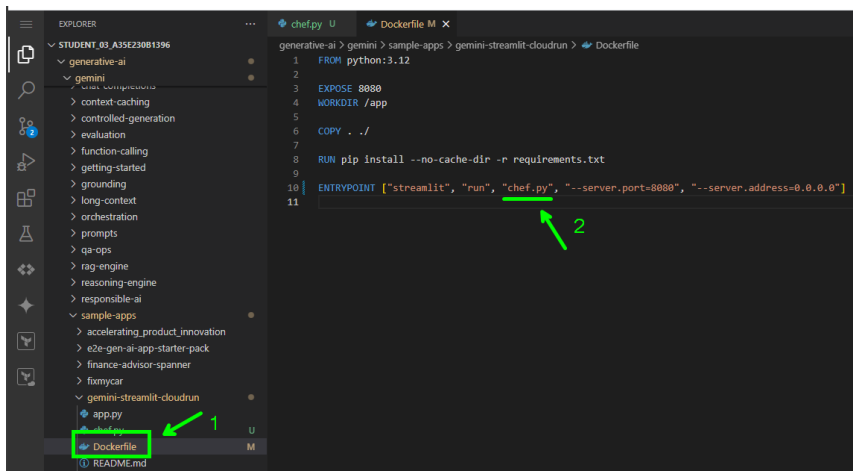
Testar o aplicativo

Verificar meu progresso

Assessment Completed!

5. Tarefa 4

1) No arquivo **Dockerfile** (1), mude o arquivo referenciado em **ENTRYPOINT** para “chef.py” (2) (Salve o arquivo após a mudança):



2) Execute o seguinte comando no **Terminal** (Não copie a seta):

OBS: Clique CTRL+Z para sair da aplicação se ainda estiver rodando

```
AR_REPO='chef-repo'
```

```
SERVICE_NAME='chef-streamlit-app'
```

```
gcloud artifacts repositories create \"$AR_REPO\" --location=\"$GCP_REGION\" --repository-format=Docker
```

```
gcloud builds submit --tag \"$GCP_REGION-docker.pkg.dev/$GCP_PROJECT/$AR_REPO/$SERVICE_NAME\"
```

3) Aguarde a conclusão do comando, se a saída do código for **STATUS: SUCCESS**, tique no botão de progresso do laboratório (1) e aba amarela (2):

Parâmetro	Valor
tag	"\$REGION-docker.pkg.dev/\$GCP_PROJECT/\$AR_REPO/\$SERVICE_NAME"

5. Aguarde a conclusão do comando. Verifique o objetivo após a conclusão.

Para verificar o objetivo, clique em **Verificar meu progresso**.

Modificar o Dockerfile e enviar a imagem Docker para o Artifact Registry.

Verificar meu progresso

Assessment Completed!

API

Tarefa 2: escrever o framework Streamlit e criar um comando para gerar o código em Python para concluir o arquivo chef.py

6. Tarefa 5

1) Ainda no **Terminal** do editor, execute o comando a seguir (Não copie a seta), preencha com Y quando pedir:

```
gcloud run deploy "$SERVICE_NAME" \
  --port=8080 \
  --image="$GCP_REGION-docker.pkg.dev/$GCP_PROJECT/$AR_REPO/$SERVICE_NAME" \
  --allow-unauthenticated \
  --region=$GCP_REGION \
  --platform=managed \
  --project=$GCP_PROJECT \
  --set-env-vars=GCP_PROJECT=$GCP_PROJECT,GCP_REGION=$GCP_REGION
```

2) Depois que a implantação for concluída (aguarde alguns minutos), clique na URL gerada pela saída do comando com CTRL+Botão Esquerdo do Mouse, e em seguida teste novamente a aplicação clicando no botão para gerar receita ao pé da página (Como no passo anterior, porém agora no Cloud Run).

3) Tique o progresso do laboratório e em seguida a aba amarela para a tarefa:

continuar (isso vai levar alguns minutos). Se a saída for afirmativa, selecione Y para continuar.

A implantação vai levar alguns minutos. Depois de receber um URL para o serviço, abra o navegador para visualizar o aplicativo implantado.

2. Teste o aplicativo com o link fornecido.

Depois de testar com sucesso o aplicativo, clique no objetivo.

Para verificar o objetivo, clique em **Verificar meu progresso**.

✓

Implantar e testar o aplicativo no Cloud Run.

Verificar meu progresso

Assessment Completed!

1

Write Streamlit framework code for the user interface	Verificar meu progresso	25 / 25
Test the application in Cloud Shell	Verificar meu progresso	20 / 20
Modify the Dockerfile and push the Docker image to the Artifact Registry	Verificar meu progresso	20 / 20
Deploy the application to Cloud Run and test it	Verificar meu progresso	20 / 20

2

API

Tarefa 2: escrever o framework Streamlit e criar um comando para gerar o código em Python para concluir o arquivo chef.py

Tarefa 3: testar o

4) Se estiver 100/100 na aba amarela, finalize o laboratório.