

MATERIAL DE APOIO

Google Cloud Skill Boost

Laboratório 08 do Intermediate: Como usar o Gemini no ciclo de vida de desenvolvimento do software

Sumário

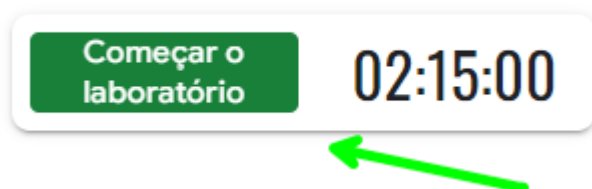
1.	Login no Console do Google Cloud	3
2.	Tarefa 1	5
3.	Tarefa 2	6
4.	Tarefa 3	10
5.	Tarefa 4	17

1. Login no Console do Google Cloud

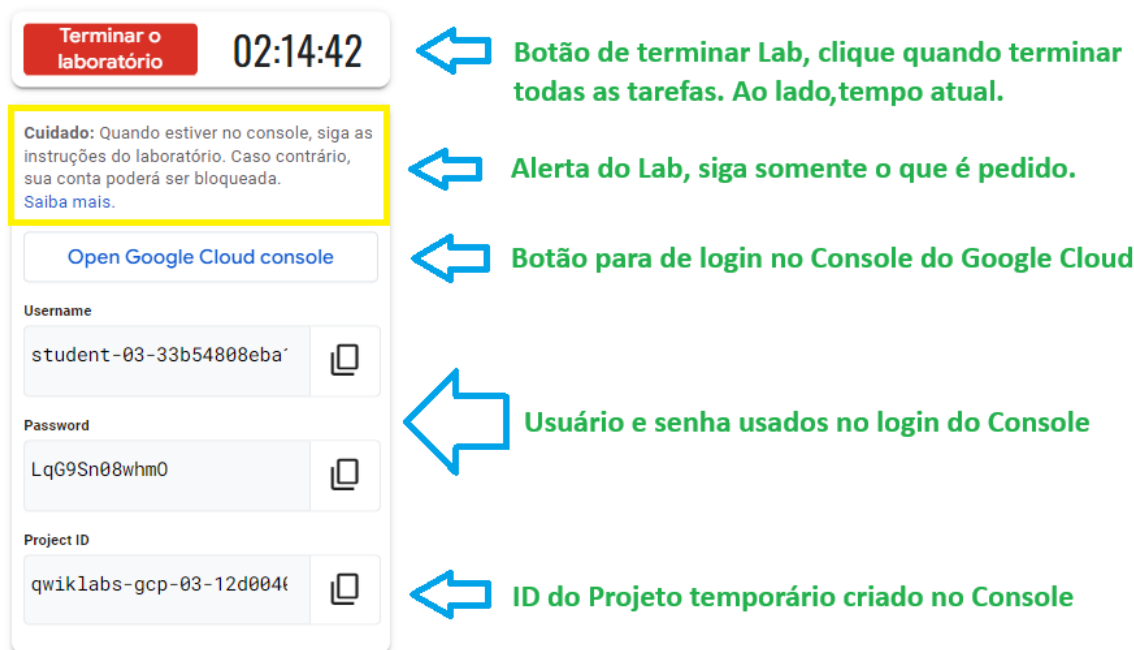
Link para o laboratório:

[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)

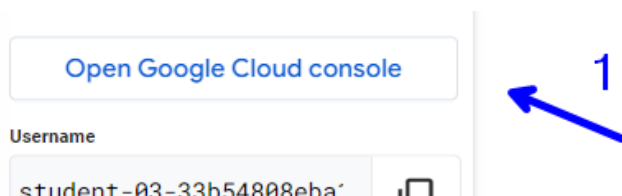
- 1) Primeiro passo é fazer login no Console do Google Cloud, clique no botão verde “Começar o laboratório”:

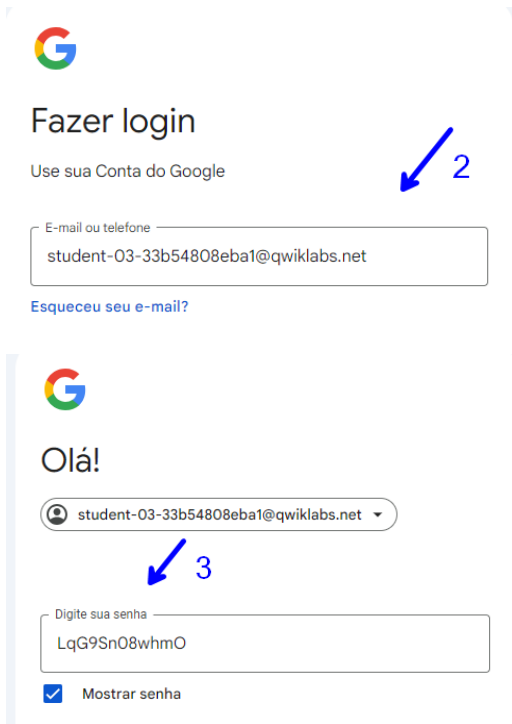


- 2) Após, irá abrir o Painel de login no Console:



- 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](#).

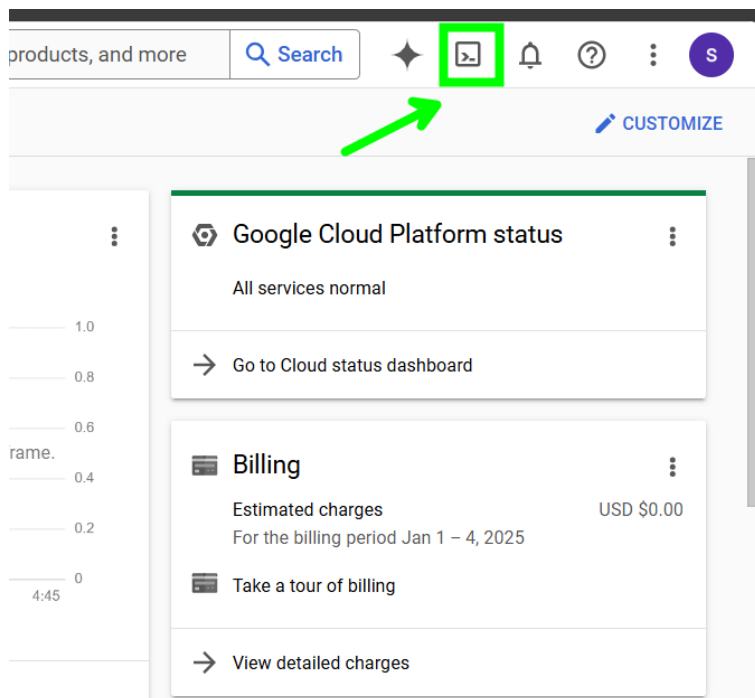
Email updates

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

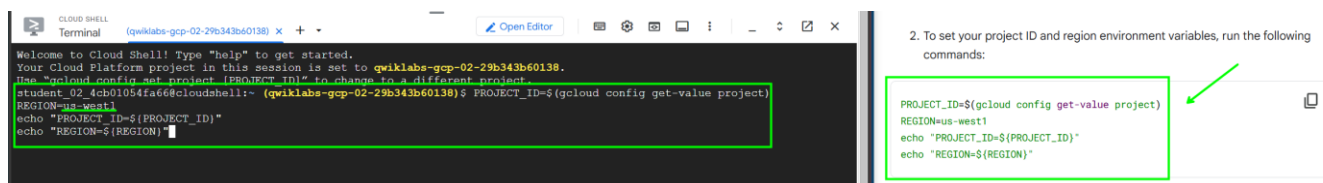
AGREE AND CONTINUE

2. Tarefa 1

1) Habilite o **Cloud Shell** no ícone do canto superior direito :



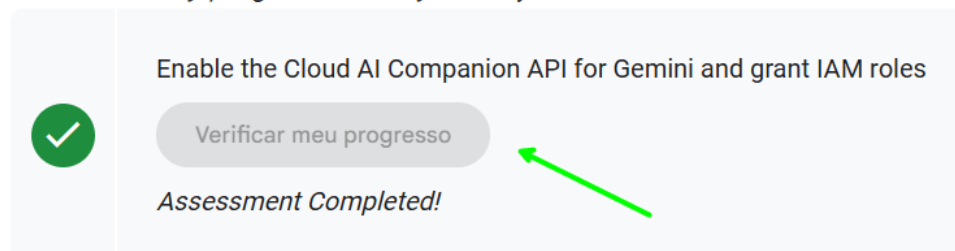
2) Quando terminal abrir, execute **todos os comandos** (Um por vez), indicados na **TASK 1**, do laboratório. Como no exemplo a seguir:



OBS: Para as próximas Tasks, utilize sempre a região que aparece no primeiro comando.

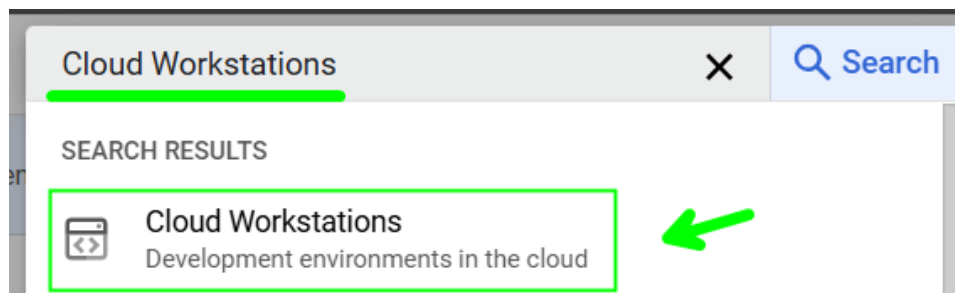
3) Depois, clique no botão do progresso:

Click *Check my progress* to verify the objective.

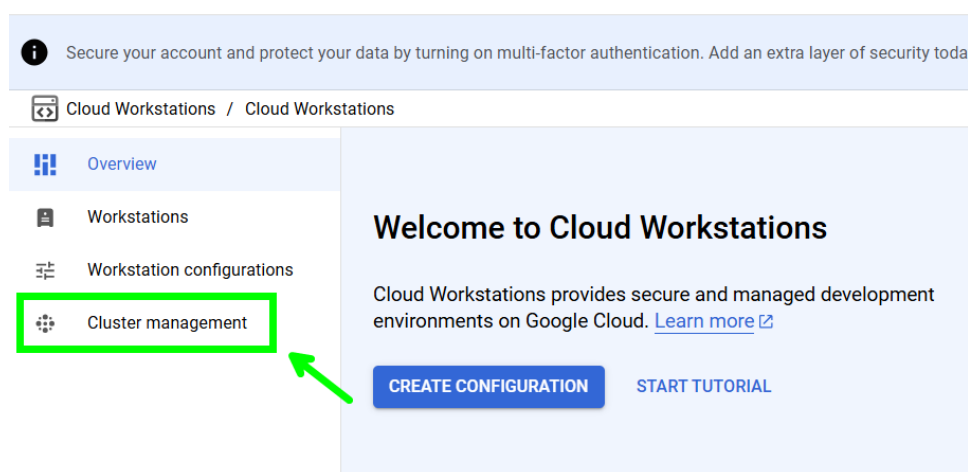


3. Tarefa 2

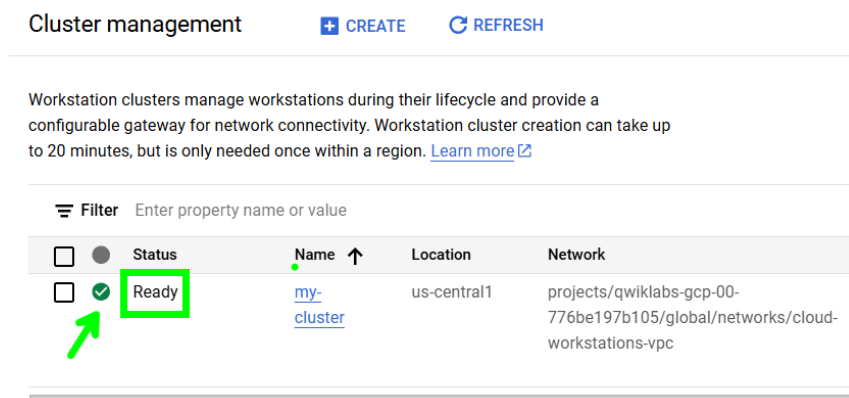
- 1) Pesquise no campo de pesquisa acima da página, por **Cloud Workstations**:



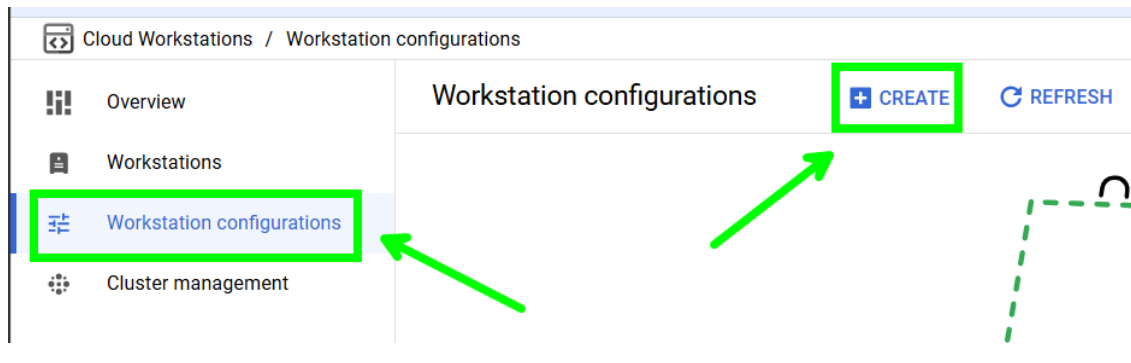
- 2) A seguir, clique em **Cluster Management**:



- 3) Quando Verifique o status do cluster. Se o status do cluster for Reconciling ou Updating, atualize periodicamente e aguarde até que ele mude para Ready antes de seguir para a próxima etapa:



- 4) Após clique em **Workstation configurations**, e depois no botão **CREATE**:



- 5) Na tela seguinte, preencha os as informações com os dados que seu laboratório indica, deixe todos o restante como padrão, e por fim clique em **CREATE** no pé da página:

Propriedade	Valor (digite ou selecione)
Nome	my-config
Cluster de estações de trabalho	my-cluster
Estações de trabalho de início rápido	Desativada

Basic information

i Get AI-powered collaboration with Gemini CodeAssist in Cloud Workstations. [Learn more](#).

Name *
my-config

Workstation cluster *
my-cluster (us-central1)

Quick start workstations ?

☐ Enabled (faster workstation startup)

☒ Disabled (lower cost)

i Disabling this feature will result in lower costs, but workstations can take significantly (several minutes) longer to start.

CREATE CANCEL

- 6) Clique em **REFRESH**, e verifique o Status da configuração que está sendo criada. Se o status for Reconciling ou Updating, atualize periodicamente e aguarde até que ele mude para Ready antes de seguir para a próxima etapa:

Workstation configurations [+ CREATE](#) [REFRESH](#)

Workstation configurations act as templates for workstations. Create a workstation configuration first, and then create workstations. Add creators to a configuration to allow them to create their own workstation. [Learn more](#)

Filter Enter property name or value

<input type="checkbox"/>	Status	Name ↑		Location	Workstation cluster
<input checked="" type="checkbox"/>	Ready	my-config	ADD USERS	us-central1	my-cluster

7) Em seguida, clique em **Workstations**, e depois em **CREATE WORKSTATION**:

Cloud Workstations / My workstations

Overview **Workstations** Workstation configurations Cluster management

Workstations [+ CREATE WORKSTATION](#) [REFRESH](#)

Cloud Workstations provides managed, on-demand, development environments in the cloud. They can be accessed through a browser UI, an SSH terminal, or from your local IDE through an SSH bridge. [Learn more](#)

8) Na tela seguinte, preencha os as informações com os dados que seu laboratório indica, deixe todos o restante como padrão, e por fim clique em **CREATE** no pé da página (Como indicado nas figuras abaixo):

Propriedade	Valor (digite ou selecione)
Nome	my-workstation
Nome de exibição	my-workstation
Configuração	my-config

Cloud Workstations / My workstations / Create workstation

Create workstation

Name *
my-workstation

Display Name
my-workstation

Configuration *
my-config

Environment variables

These environment variables will be passed to the container.

[+ ADD VARIABLE](#)

Labels

Labels are applied to the resource and propagated to the underlying Compute Engine


CREATE


CANCEL


9) Na tela seguinte, clique em **START** para iniciar o workstation:


Workstations [+ CREATE WORKSTATION](#) [REFRESH](#)

My workstations


 **my-workstation**

Status  Stopped


ID my-workstation 




Configuration my-config 

[START](#)

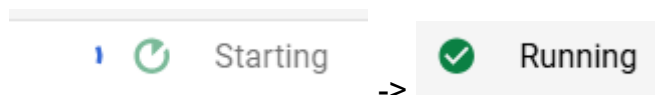


All workstations

 **Filter** Enter property name or value

<input type="checkbox"/>		Status	Name 	Quick action	Location
<input type="checkbox"/>		Stopped	my-workstation	START	us-central1

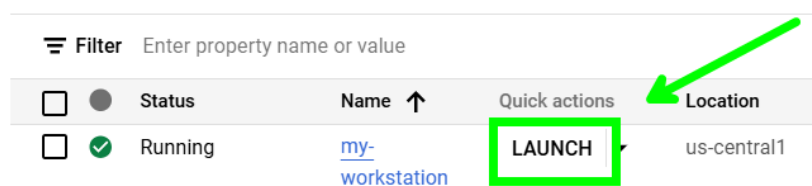
OBS: Durante a inicialização, o status muda para Starting. Espere até que ele mude para Running, o que indica que a estação está pronta para ser usada. Ela pode levar vários minutos para ser inicializada:



10) Siga as orientações do laboratório em **“Iniciar o ambiente de desenvolvimento integrado”**

11) Após clique em **LAUNCH**:

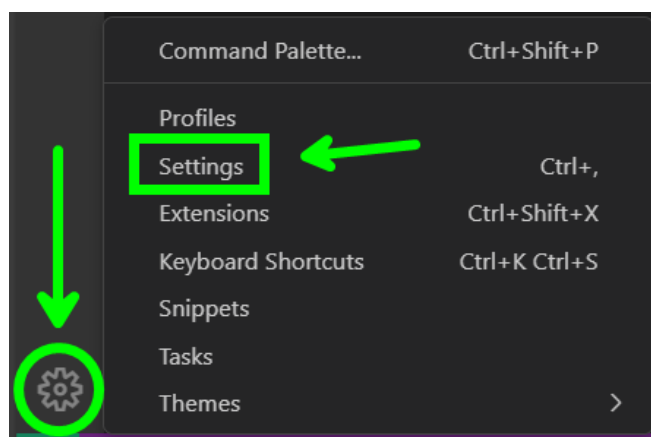
All workstations



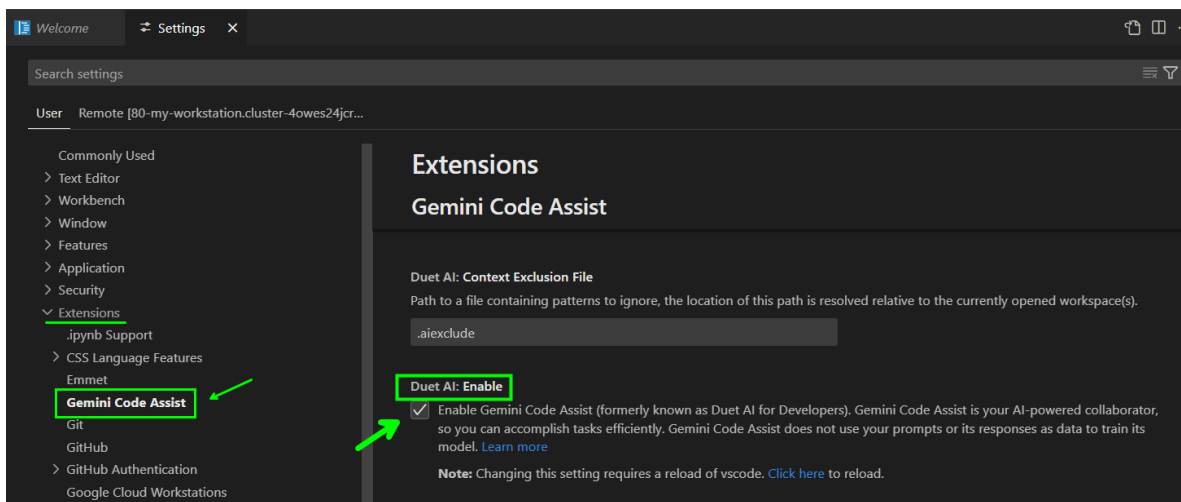
12) Após aberto, clique para ticar a **Task 2** do laboratório.

4. Tarefa 3

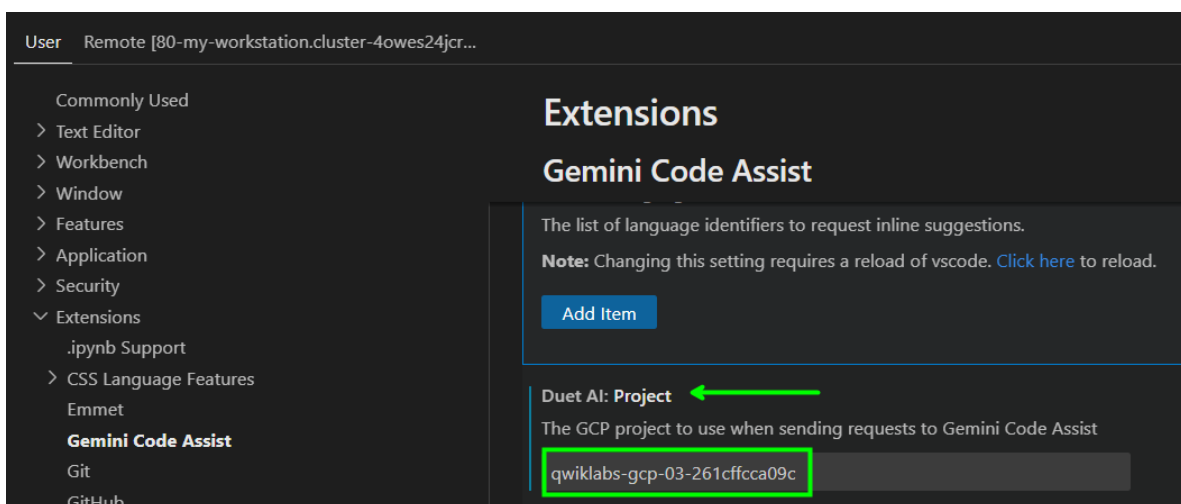
1) Dentro do Workstation, clique na engrenagem que aparece no canto inferior esquerdo, em seguida em **Settings**:



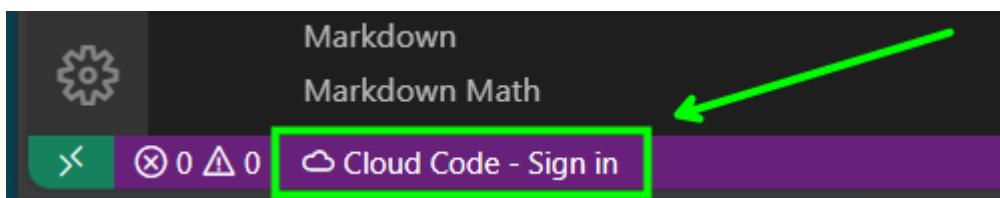
2) Vá em **Extensions > Gemini Code Assist**, e garanta que o **Duet AI** esteja habilitado, através de ticado em “Enable”:



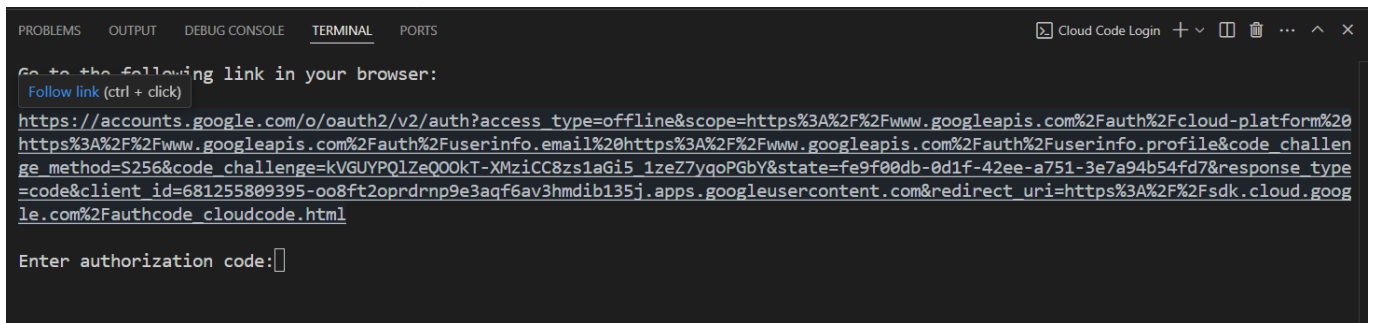
3) Coloque seu **Project ID** em **Duet AI: Project**:



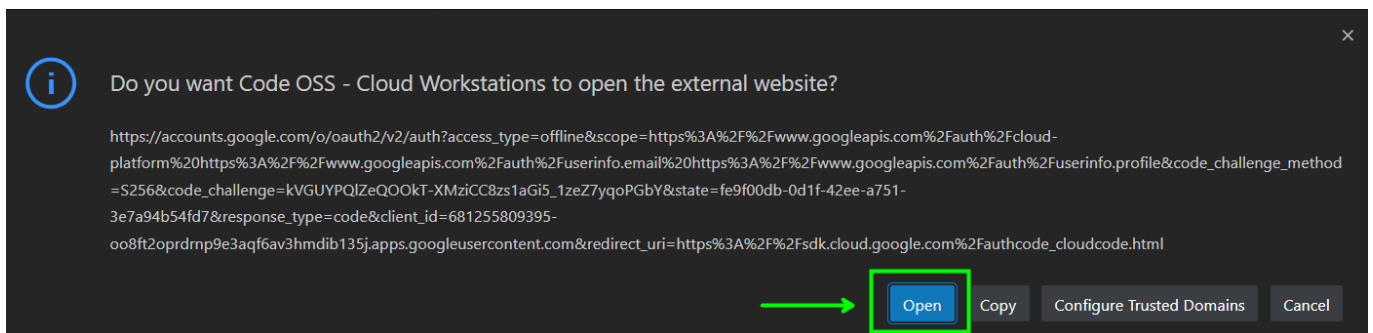
4) No pé da página, no canto inferior esquerdo, embaixo da engrenagem, clique em **Cloud Code – Sign in**:



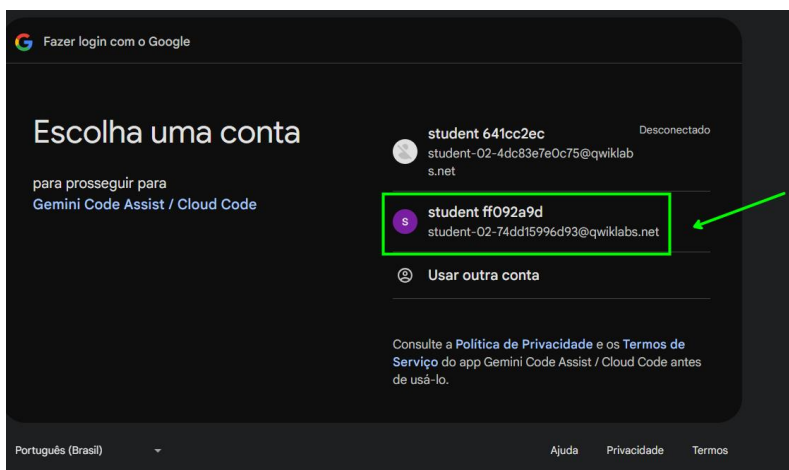
5) Para iniciar o fluxo de login do Google Cloud, pressione **CONTROL** (para Windows e Linux) ou **COMMAND** (para macOS) e clique no URL no console:



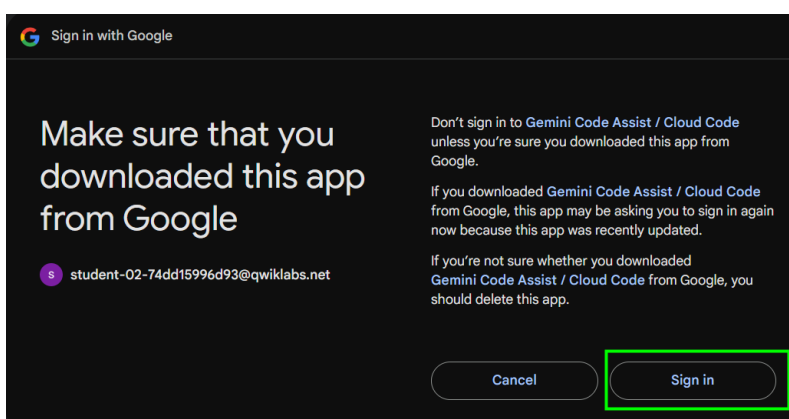
6) Clique em **OPEN** para confirmar a abertura do site externo:



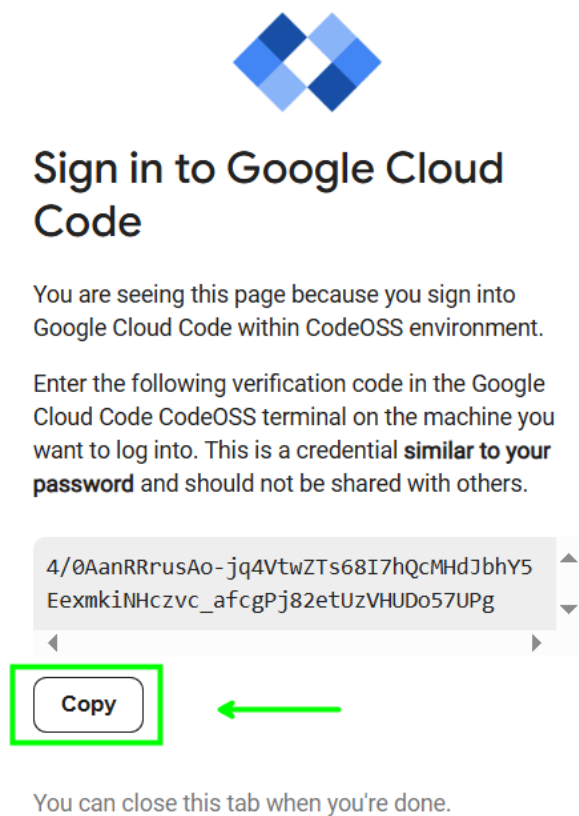
7) Clique no endereço de e-mail de estudante:



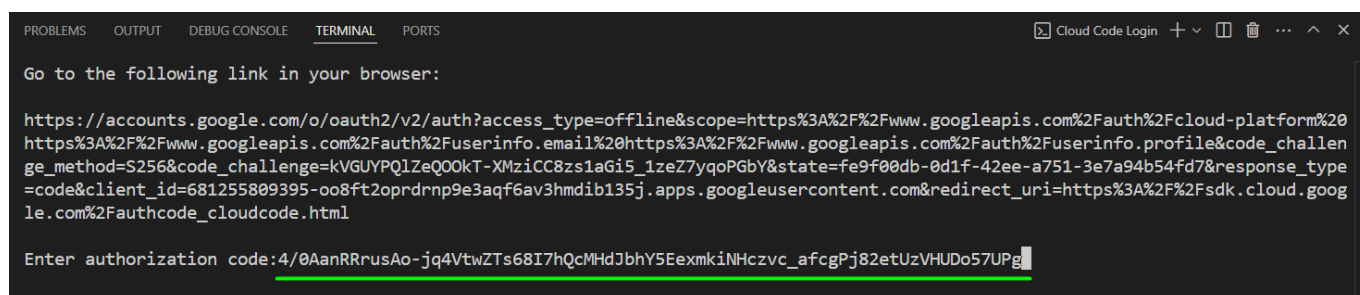
8) Na tela seguinte clique em **SIGN IN**:



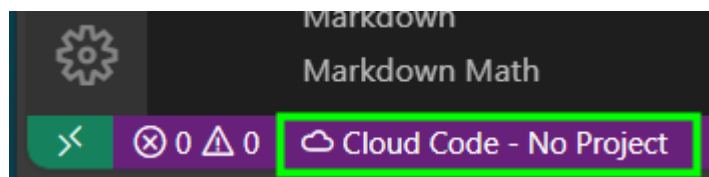
9) Copie o código de autenticação:



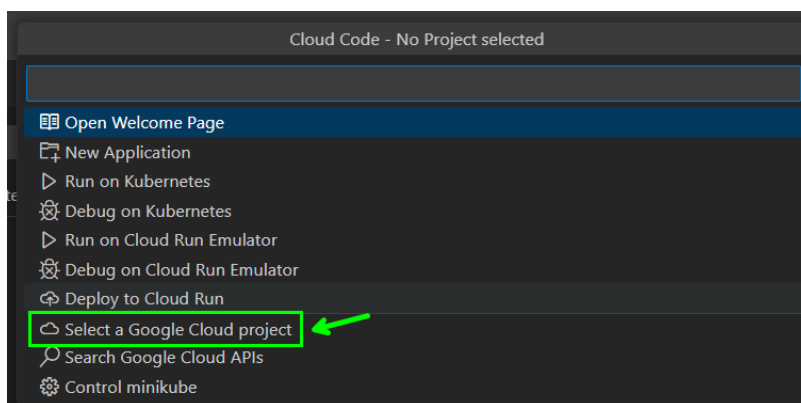
10) E cole no terminal, como indicado na imagem, e em seguida aperte **ENTER**:



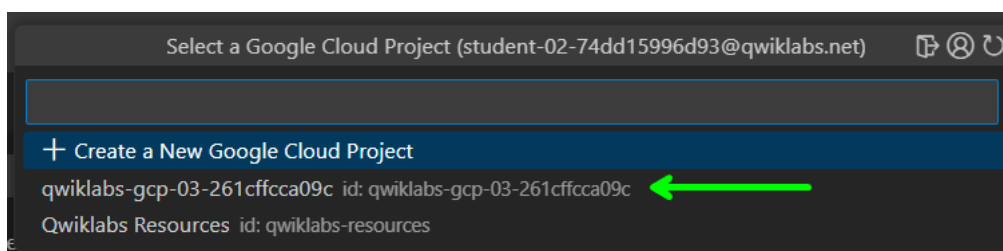
11) Clique novamente no **Cloud Code**, porém agora verá que o botão mudou para **No Project**:



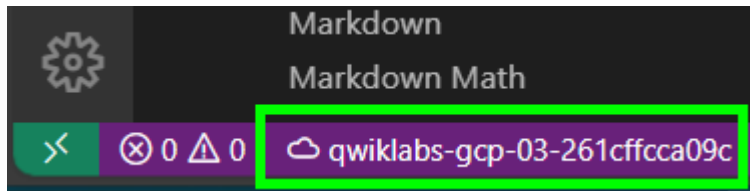
12) Clique em **Select a Google Cloud Project**, como mostra na imagem:



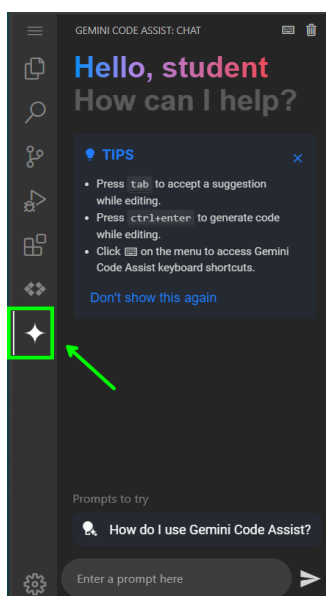
13) Selecione seu Project:



14) Pronto, agora verá uma indicação no pé da página que seu Project foi selecionado, e que você está autenticado:



15) Clique no símbolo do Gemini no menu lateral esquerdo, e verifique que está tudo ok antes de continuar:



OBS: Se algo estiver diferente, pare imediatamente e siga as alternativas sugeridas pela página do laboratório.

- 16) Aperte **CTRL + J** para abrir o **Terminal**, ou selecione **Terminal > Novo Terminal**. Execute o comando a seguir (Troque **Project** em negrito pelo Project ID de seu laboratório):

```
gcloud config set project PROJECT-ID
```

- 17) Depois execute esse comando, clique Y para aceitar (Siga as orientações do laboratório se precisar se autenticar novamente):

```
gcloud auth configure-docker
```

- 18) Após, execute esse comando (Siga as orientações do laboratório se precisar se autenticar novamente):

```
gcloud storage cp -r gs://cloud-training/OCBL435/cymbal-superstore .
```

- 19) Após, execute os seguintes comandos em ordem (Caso haja erro, copie o comando fornecido pelo laboratório):

```
cd ~/cymbal-superstore/backend
```

```
docker build --platform linux/amd64 -t gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-inventory-api .
```

```
docker push gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-inventory-api
```

```
gcloud run deploy inventory --image=gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-inventory-api --port=8000 --region=us-central1 --set-env-vars=PROJECT_ID=qwiklabs-gcp-03-261cffcca09c --allow-unauthenticated
```

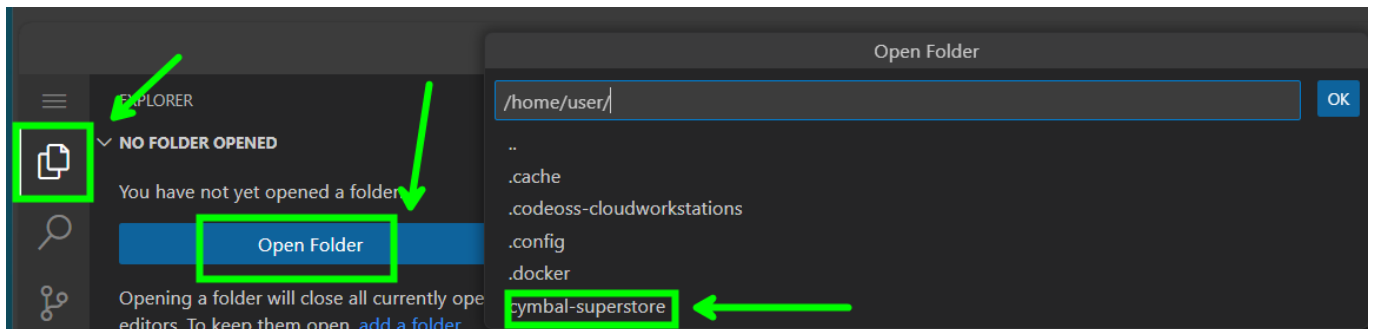
- 20) Copie o valor do URL de serviço exibido na saída do comando `gcloud run deploy`:

```

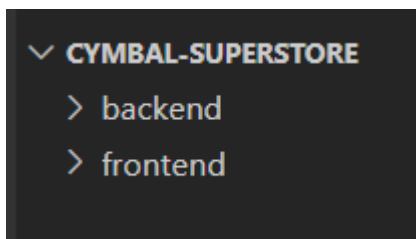
user@my-workstation:~/cymbal-superstore/backend$ gcloud run deploy inventory --image=gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-in
ventory-api --port=8000 --region=us-central1 --set-env-vars=PROJECT_ID=qwiklabs-gcp-03-261cffcca09c --allow-unauthenticated
Deploying container to Cloud Run service [inventory] in project [qwiklabs-gcp-03-261cffcca09c] region [us-central1]
✓ Deploying new service... Done.
✓ Creating Revision...
✓ Routing traffic...
✓ Setting IAM Policy...
Done.
Service [inventory] revision [inventory-00001-f21] has been deployed and is serving 100 percent of traffic.
Service URL: https://inventory-375676006189.us-central1.run.app
user@my-workstation:~/cymbal-superstore/backend$

```

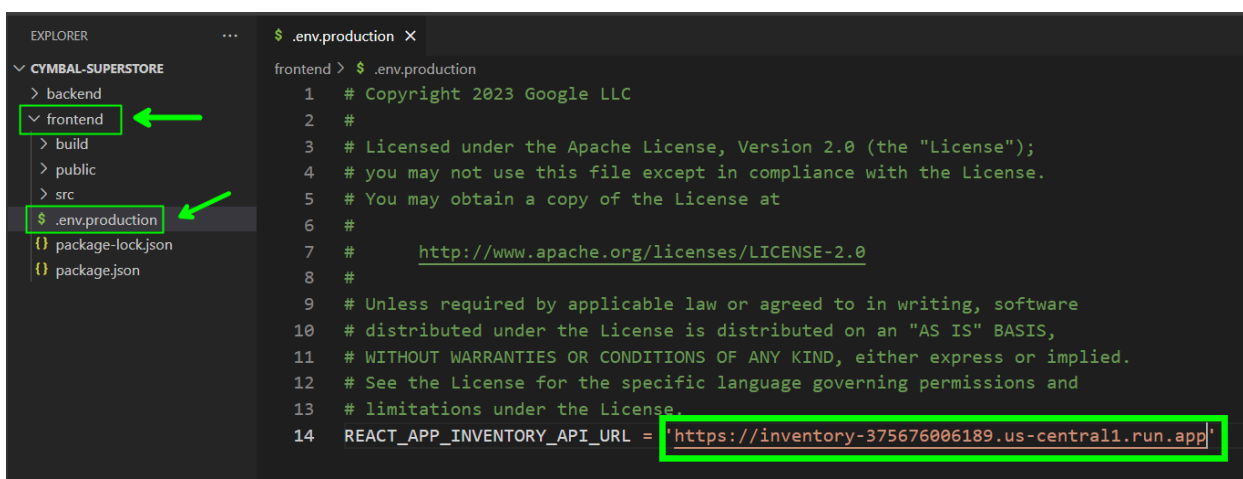
21) Clique em **Explorer** no menu lateral esquerdo, no botão azul **OPEN FOLDER**, e depois na pasta **cymbal-supestore**. E clique em **OK**:



22) Agora verá as duas pastas no **Explorer**, dessa forma podendo abrir qualquer arquivo a partir dali:



23) Abra o arquivo **frontend/.env.production**. No arquivo, substitua o valor de **REACT_APP_INVENTORY_API_URL** colando o valor do URL do endpoint do serviço de back-end do Cloud Run que você copiou antes:



24) Aperte **CTRL+S** para salvar, ou no caminho **File > Save**.

25) Volte ao **Terminal**, e execute os seguintes comandos em ordem (Troque o Project em negrito pelo seu project, ou copie o comando completo da página do laboratório):

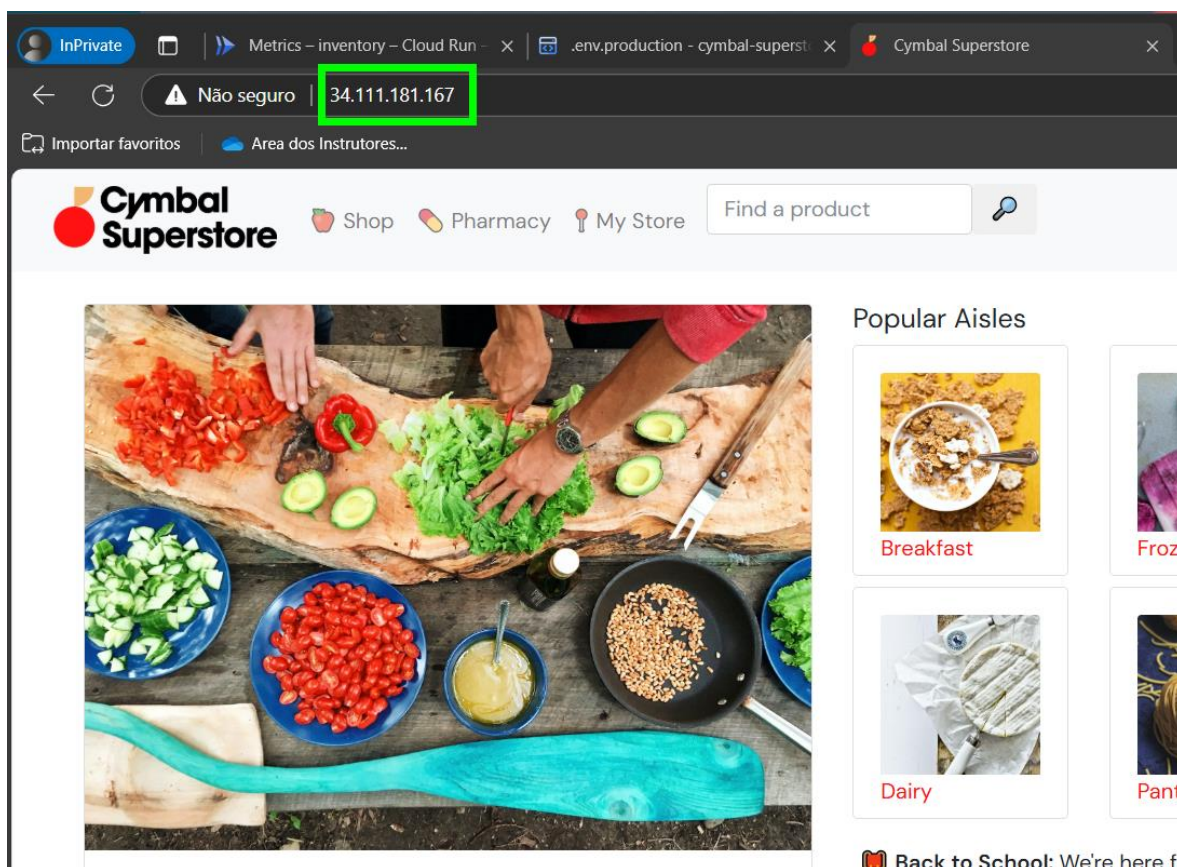
```
cd ~/cymbal-superstore/frontend
```

```
npm install && npm run build
```

```
gcloud storage cp -r build/* gs://PROJECT-ID-cymbal-frontend
```

26) Cole esse endereço IP em uma aba de seu navegador e verifique que está tudo ok:

<http://34.111.181.167>

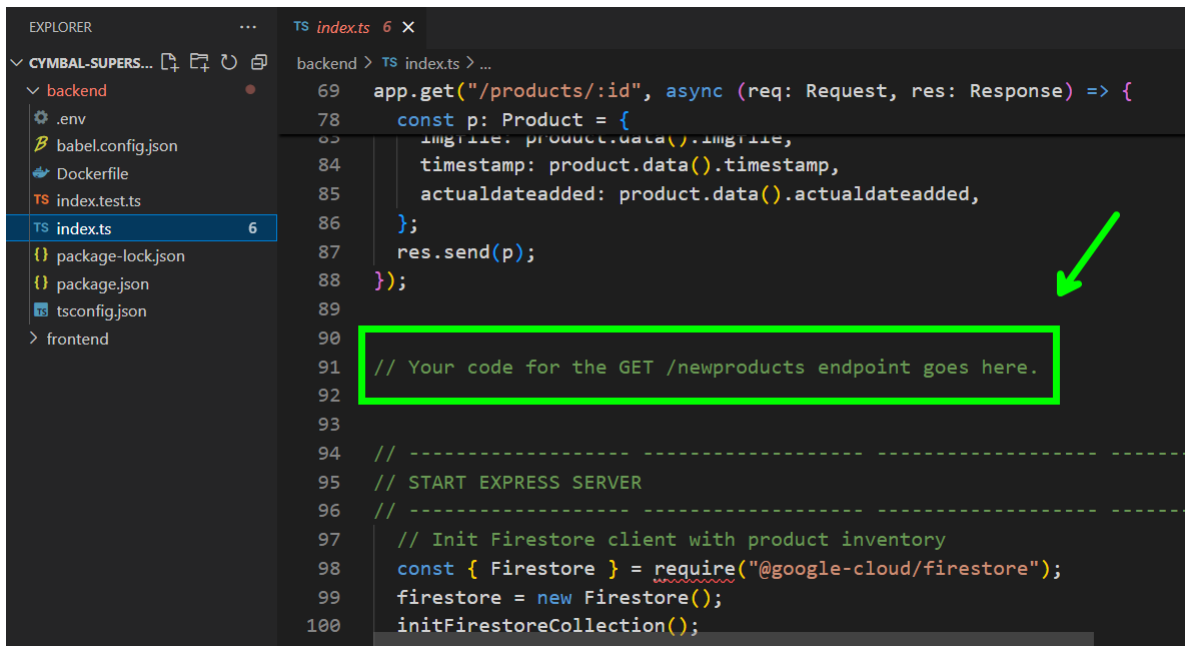


27) Tique a **Task 3** na página do laboratório.

5. Tarefa 4

1) Nas próximas etapas, iremos interagir com o Gemini para preencher a parte do código faltante do endpoint no backend. Abra o arquivo **backend/index.ts**.

- 2) No arquivo de código `index.ts`, role até a linha 91, onde há o comentário do marcador de posição para o endpoint `/newproducts`:

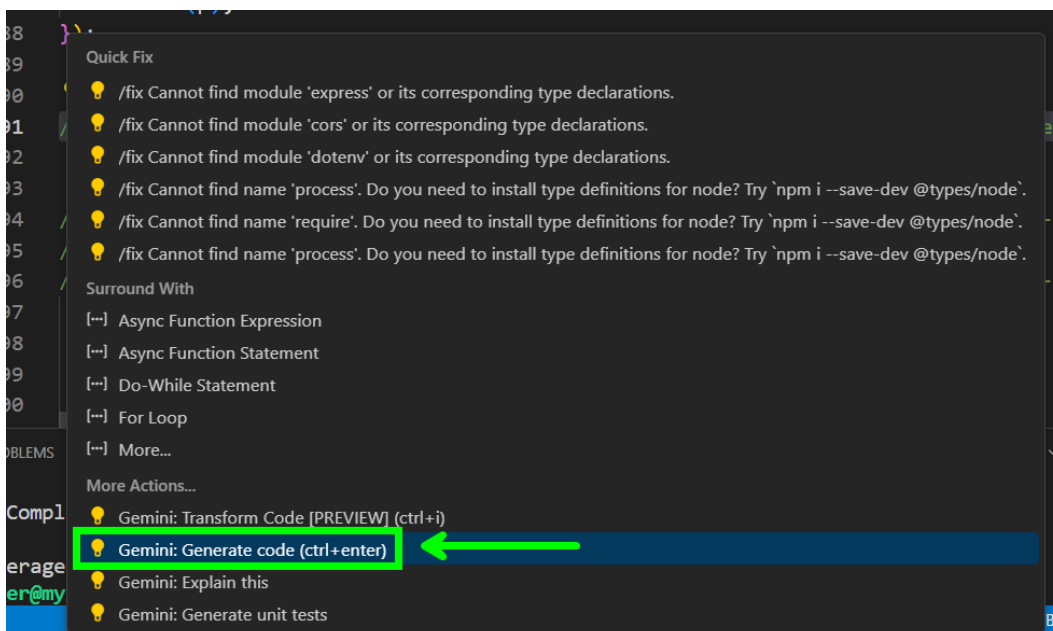


```

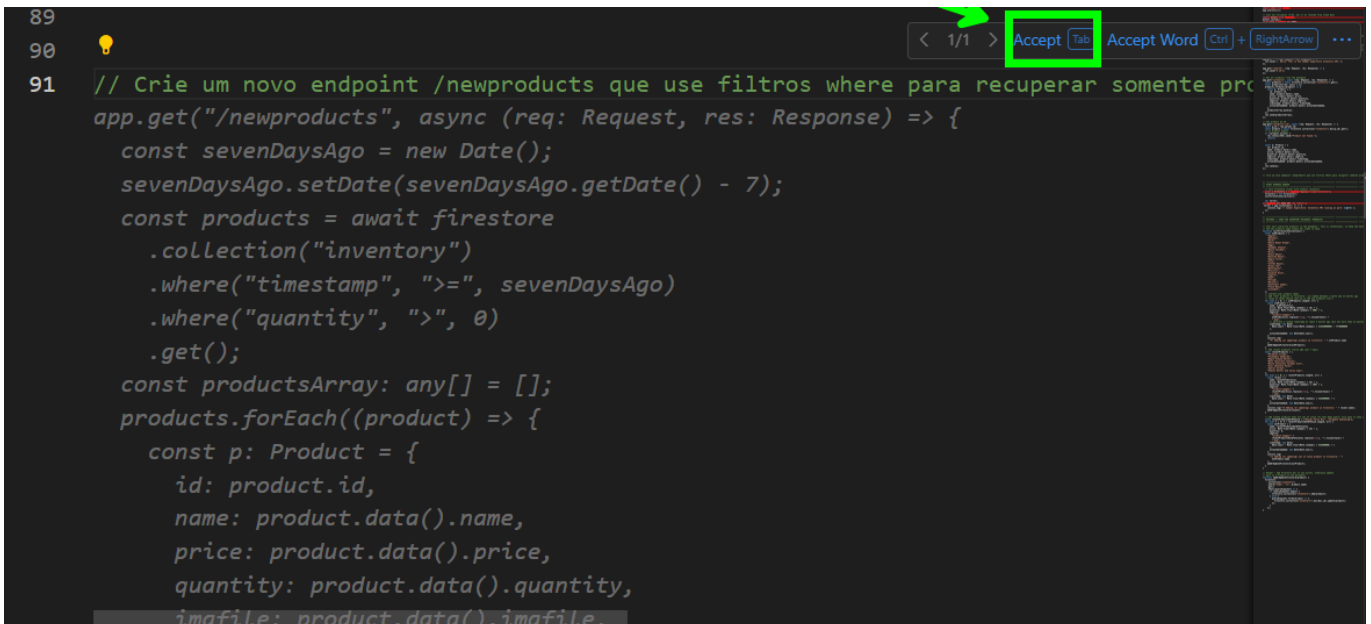
69 app.get("/products/:id", async (req: Request, res: Response) => {
70   const p: Product = {
71     imgurl: product.data().imgurl,
72     timestamp: product.data().timestamp,
73     actualdateadded: product.data().actualdateadded,
74   };
75   res.send(p);
76 };
77
78 // Your code for the GET /newproducts endpoint goes here.
79
80 // -----
81 // START EXPRESS SERVER
82 // -----
83
84 // Init Firestore client with product inventory
85 const { Firestore } = require("@google-cloud/firestore");
86 firestore = new Firestore();
87 initFirestoreCollection();

```

- 3) Substitua o comentário do marcador de posição pelo seguinte comando para o Gemini:
- // Crie um novo endpoint `/newproducts` que use filtros `where` para recuperar somente produtos que foram adicionados nos últimos sete dias e que estão em estoque.
- 4) Em seguida, selecione toda a linha desse novo comentário, clique na lâmpada que aparece ao lado 💡. E em seguida em **Gemini: Generate code**:



- 5) Passe o curso por cima da sugestão gerada e clique em **Accept**, ou clique em **Tab** no seu teclado para aceitar o código gerado:

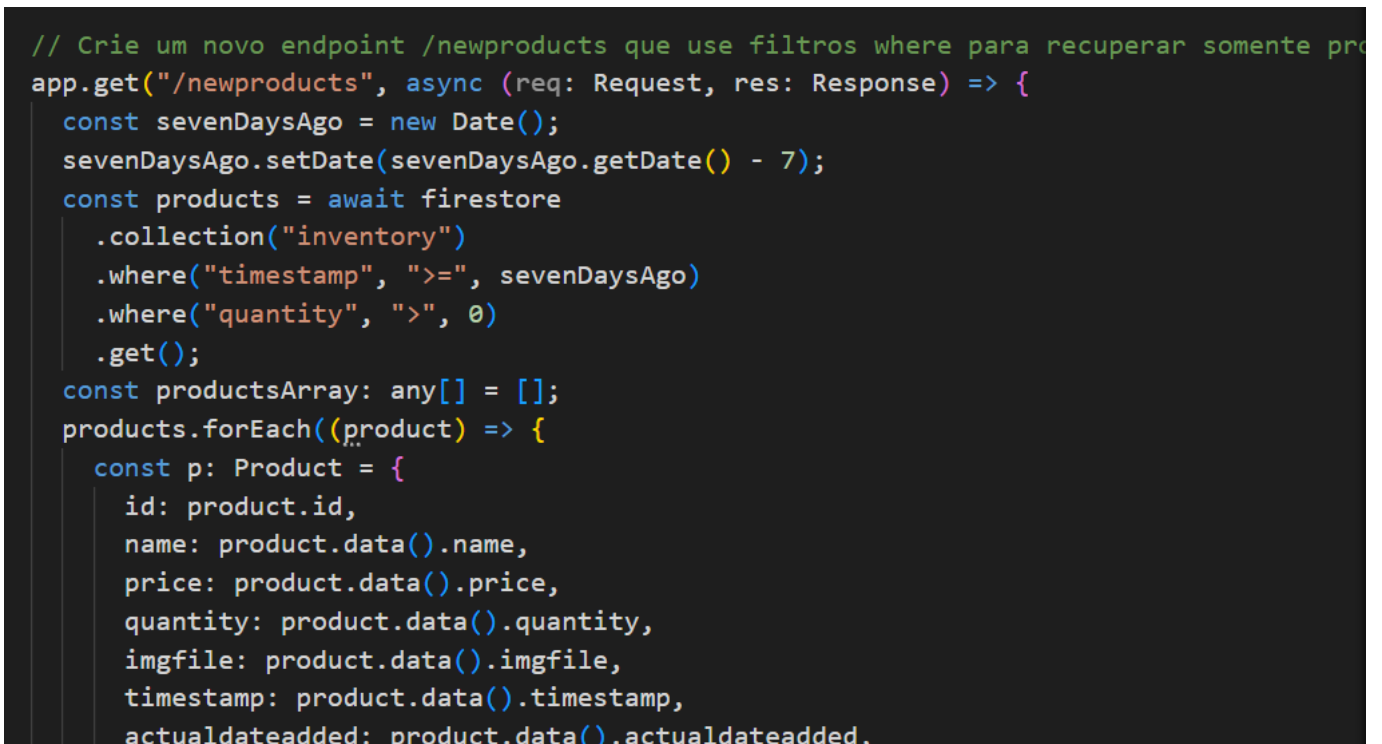


```

89
90
91 // Crie um novo endpoint /newproducts que use filtros where para recuperar somente pro
app.get("/newproducts", async (req: Request, res: Response) => {
  const sevenDaysAgo = new Date();
  sevenDaysAgo.setDate(sevenDaysAgo.getDate() - 7);
  const products = await firestore
    .collection("inventory")
    .where("timestamp", ">=", sevenDaysAgo)
    .where("quantity", ">", 0)
    .get();
  const productsArray: any[] = [];
  products.forEach((product) => {
    const p: Product = {
      id: product.id,
      name: product.data().name,
      price: product.data().price,
      quantity: product.data().quantity,
      imgfile: product.data().imgfile,

```

Ficando dessa forma:



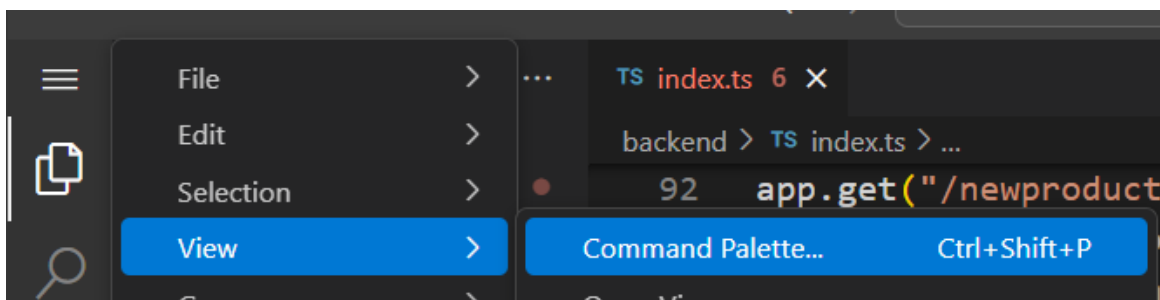
```

// Crie um novo endpoint /newproducts que use filtros where para recuperar somente pro
app.get("/newproducts", async (req: Request, res: Response) => {
  const sevenDaysAgo = new Date();
  sevenDaysAgo.setDate(sevenDaysAgo.getDate() - 7);
  const products = await firestore
    .collection("inventory")
    .where("timestamp", ">=", sevenDaysAgo)
    .where("quantity", ">", 0)
    .get();
  const productsArray: any[] = [];
  products.forEach((product) => {
    const p: Product = {
      id: product.id,
      name: product.data().name,
      price: product.data().price,
      quantity: product.data().quantity,
      imgfile: product.data().imgfile,
      timestamp: product.data().timestamp,
      actualdateadded: product.data().actualdateadded,

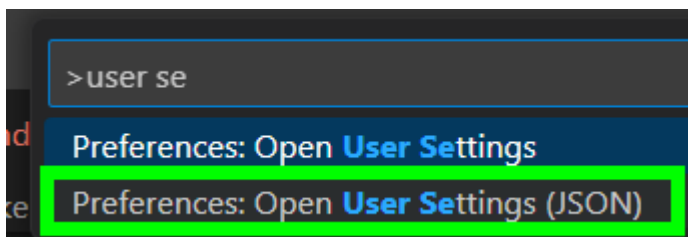
```

- 6) Se o código gerado não for parecido com o exemplo da etapa anterior, substitua o código em index.ts pelo código do laboratório (Garanta que ele esteja todo identado corretamente e sem erros).

7) Em seguida, clique em **Viem > Command Palette**



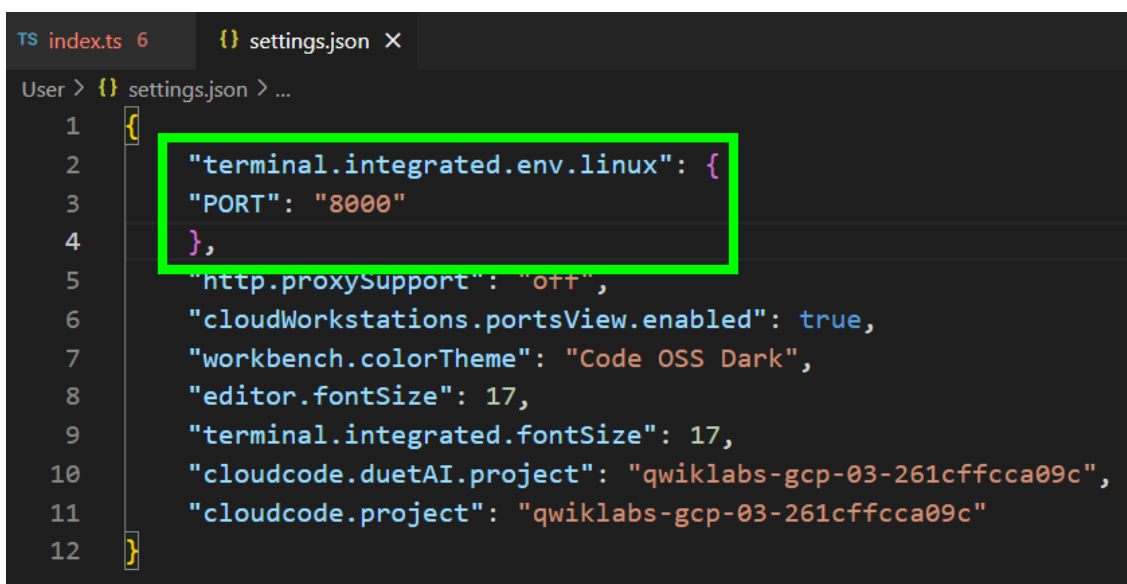
8) Comece a digitar **user settings json**, e selecione o indicado na imagem:



9) Adicione o objeto json com os seguintes parâmetros:

```
"terminal.integrated.env.linux": {
  "PORT": "8000"
},
```

Ficando da seguinte forma, depois salve o arquivo:



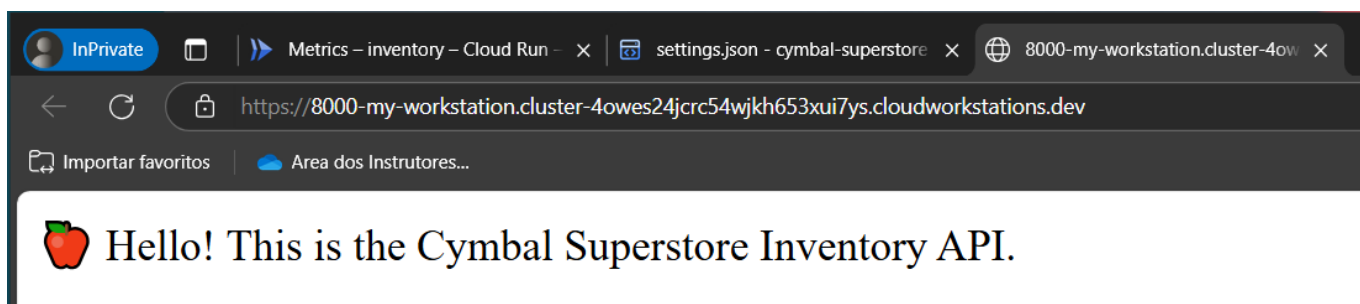
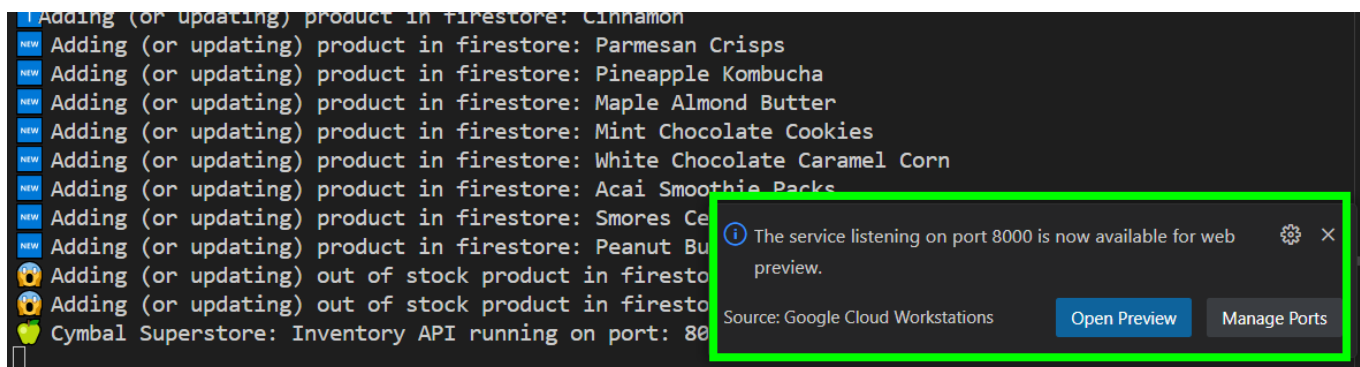
10) Execute os seguintes comandos em ordem no **Terminal** (Se logue como anteriormente):

```
gcloud auth application-default login
```

```
cd ~/cymbal-superstore/backend
```

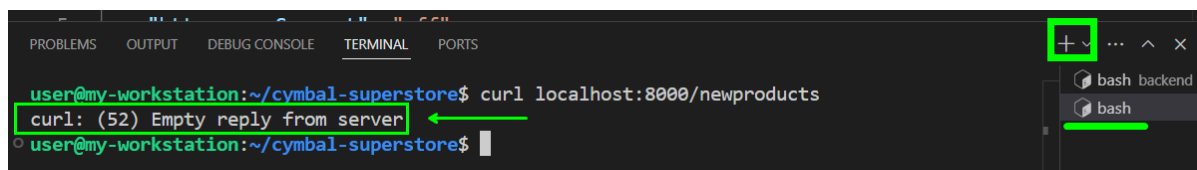
```
npm run start
```

- 11) Quando tiver uma confirmação que o serviço está escutando, clique no botão azul **OPEN PREVIEW** e veja que está ok:



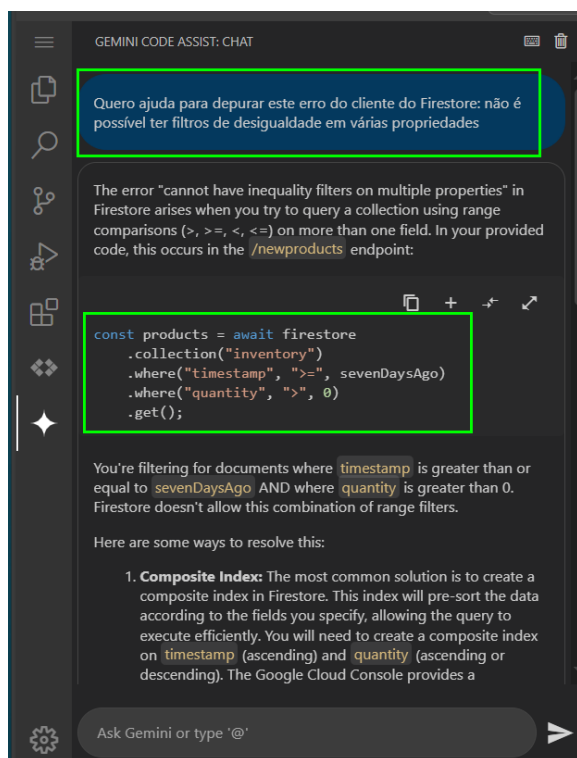
- 12) Na barra de menus do **Terminal**, clique em + para abrir um novo terminal. Execute o comando e veja se gera o erro indicado no laboratório:

```
curl localhost:8000/newproducts
```



- 13) Frete ao erro, peça agora ajuda ao Gemini para a correção, clique no símbolo do Gemini no menu lateral esquerdo, e envie esse prompt (Ficando como indica a figura):

Quero ajuda para depurar este erro do cliente do Firestore: não é possível ter filtros de desigualdade em várias propriedades



- 14) Para se ter apenas um filtro de igualdade, no arquivo backend/index.ts, para remover o filtro de desigualdade quantity, exclua a seguinte linha do código do gerenciador /newproducts no arquivo **backend/index.ts**:

```
.where("quantity", ">", 0)
```

- 15) Depois, como indicado na página do laboratório, dê um tab em todo bloco do **CONST**, e coloque tudo dentro da nova linha de **IF**, ficando como mostra na figura (Salve o arquivo depois):

The screenshot shows the backend/index.ts file. The code is as follows:


```
const products = await firestore
  .collection("inventory")
  .where("timestamp", ">=", sevenDaysAgo)
  .get();
const productsArray: any[] = [];
products.forEach((product) => {
  if (product.data().quantity > 0) {
    const p: Product = {
      id: product.id,
      name: product.data().name,
      price: product.data().price,
      quantity: product.data().quantity,
      imgfile: product.data().imgfile,
      timestamp: product.data().timestamp,
      actualdateadded: product.data().actualdateadded,
    };
    productsArray.push(p);
  });
});
res.send(productsArray);
```

 A green box highlights the inner block of code (the Product object creation and push) and a green arrow points to the if statement condition, indicating where to place a tab to wrap the entire block in an if statement.

16) Execute os comandos no **Terminal** nessa ordem:

```
cd ~/cymbal-superstore/backend
```

```
npm run start
```

```
curl localhost:8000/newproducts
```

17) Deve vir agora a matrix de produtos:

```
user@my-workstation:~/cymbal-superstore$ curl localhost:8000/newproducts
[{"id":"GKTm9UAgELiMjk8SWz5","name":"Peanut Butter and Jelly Cups","price":7,"quantity":22,"imgfile":"product-images/peanutbutterandjellycups.png","timestamp":{"_seconds":1735609063,"_nanoseconds":684000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":360000000}},{"id":"AE9c9Mz2pzAV3NHjfiCk","name":"White Chocolate Caramel Corn","price":10,"quantity":13,"imgfile":"product-images/whitechocolatecaramelcorn.png","timestamp":{"_seconds":1735727206,"_nanoseconds":757000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":360000000}},{"id":"jbhmczSnQUWZE86eeGi7","name":"Maple Almond Butter","price":2,"quantity":98,"imgfile":"product-images/maplealmondbutter.png","timestamp":{"_seconds":1735766551,"_nanoseconds":743000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":350000000}},{"id":"LnI7qThI2MZDu0DpcJG","name":"Acai Smoothie Packs","price":5,"quantity":33,"imgfile":"product-images/acaismoothiepacks.png","timestamp":{"_seconds":1735780240,"_nanoseconds":191000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":360000000}},{"id":"CwCzXjc7fuJhHaQrsyt","name":"Mint Chocolate Cookies","price":3,"quantity":22,"imgfile":"product-images/mintchocolatecookies.png","timestamp":{"_seconds":1735796622,"_nanoseconds":175000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":350000000}},{"id":"WkghaRBjKcCFMaPDJkc2","name":"Parmesan Crisps","price":9,"quantity":41,"imgfile":"product-images/parmesancrisps.png","timestamp":{"_seconds":1735844865,"_nanoseconds":296000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":340000000}},{"id":"rPn927QsMhZImAXZqRaC","name":"Smores Cereal","price":6,"quantity":98,"imgfile":"product-images/smoescereal.png","timestamp":{"_seconds":1735982020,"_nanoseconds":946000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":360000000}},{"id":"wV3nrDPhogViEafkVQs","name":"Pineapple Kombucha","price":4,"quantity":11,"imgfile":"product-images/pineapplekombucha.png","timestamp":{"_seconds":1736088713,"_nanoseconds":509000000},"actualdateadded":{"_seconds":1736097858,"_nanoseconds":350000000}}]user@my-workstation:~/cymbal-superstore$
```

18) Execute os comandos novamente (Sempre substitua os os valores referentes ao seu laboratório em negrito):

```
cd ~/cymbal-superstore/backend
```

```
docker build --platform linux/amd64 -t gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-inventory-api .
```

```
docker push gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-inventory-api
```

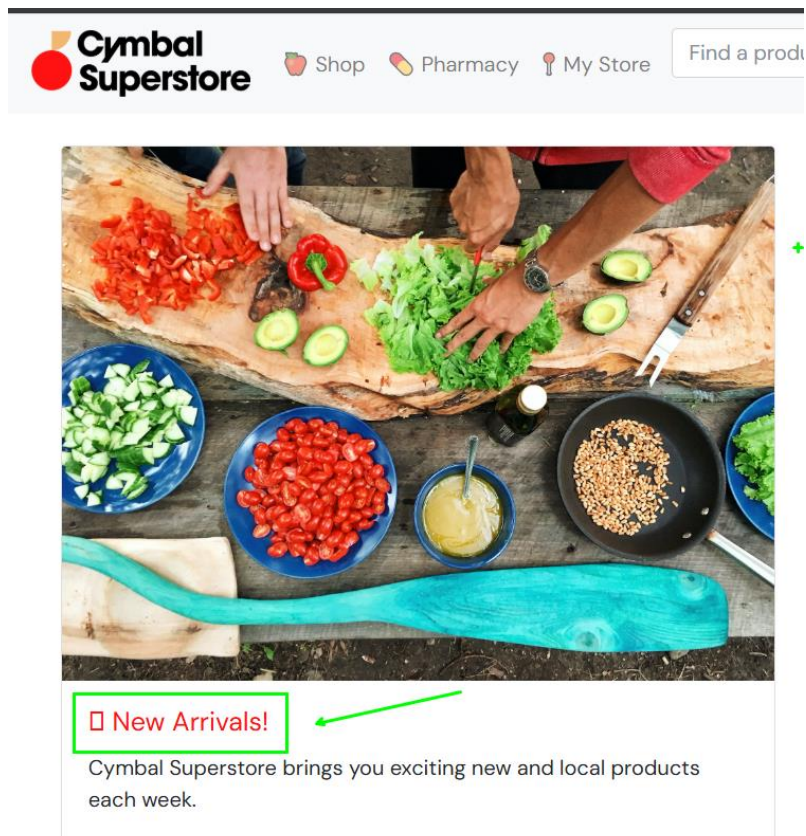
```
gcloud run deploy inventory --image=gcr.io/qwiklabs-gcp-03-261cffcca09c/cymbal-inventory-api --port=8000 --region=us-central1 --set-env-vars=PROJECT_ID=qwiklabs-gcp-03-261cffcca09c --allow-unauthenticated
```

19) Coloque o endereço IP em uma nova aba de seu navegador:

<http://34.111.181.167>

20) Clique em Novidades.

- 21) A página deve exibir produtos retornados na resposta do endpoint `/newproducts`, que você implementou na tarefa anterior.



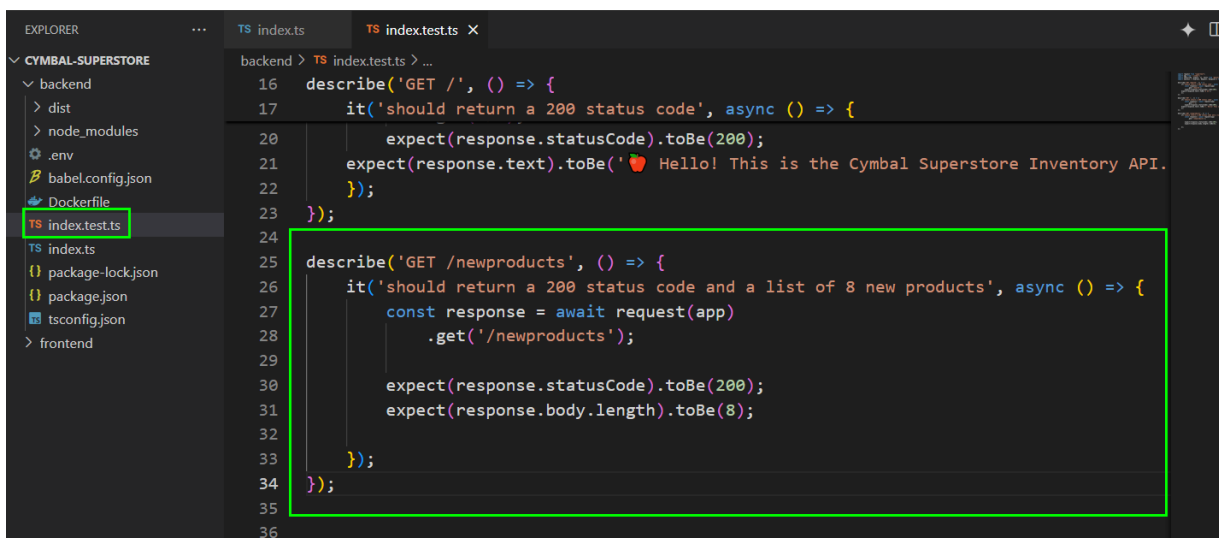
- 22) Agora, depois que a página de novos produtos está sendo preenchida, iremos pedir ao Gemini para que ele desenvolva um teste para a API. Coloque o seguinte comando no chat do Gemini:

Quero ajuda para escrever um teste Express.js usando o Jest, em TypeScript, para o gerenciador GET `/newproducts` em `index.ts`. O teste vai verificar se o código de resposta é 200 e se o tamanho da lista de novos produtos é 8.

Copie somente o bloco **DESCRIBE** como indicado na imagem:



23) Adicione esse bloco de código ao fim do arquivo **backend/index.test.ts**, como mostrado na figura:



24) Execute esse comando no **Terminal**:

```
cd ~/cymbal-superstore/backend
```

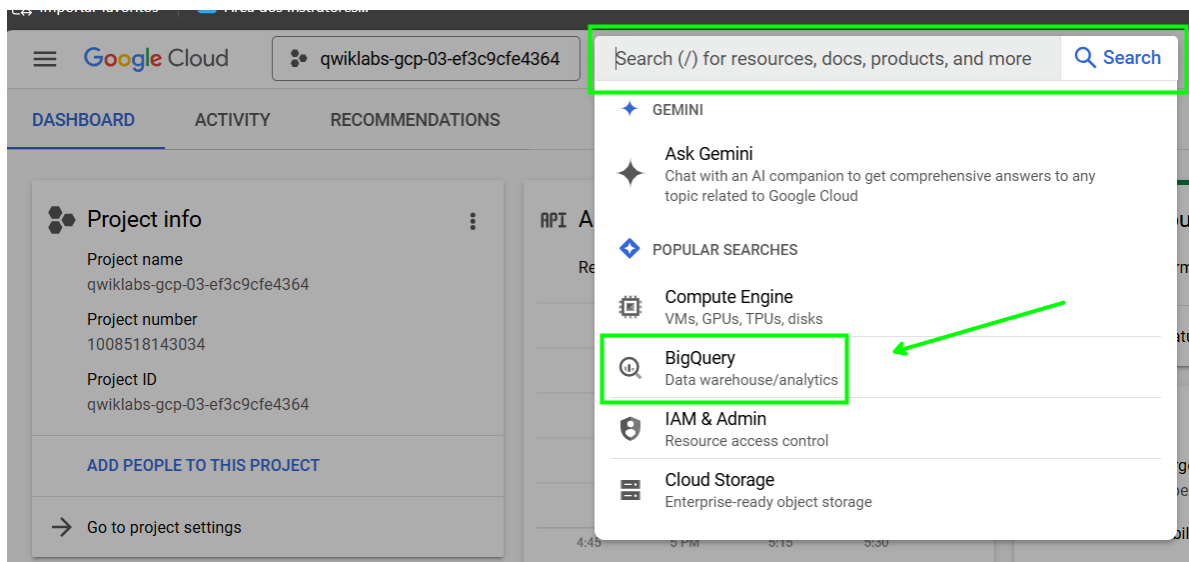
```
npm run test
```

```

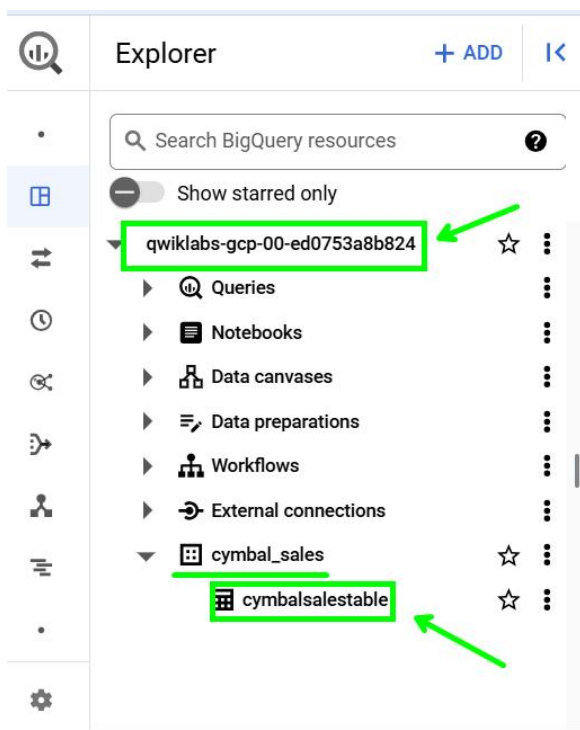
Test Suites: 2 passed, 2 total
Tests:      6 passed, 6 total
Snapshots:  0 total
Time:       9.426 s
Ran all test suites.
user@my-workstation:~/cymbal-superstore/backend$

```

25) Voltando na página inicial. No campo de pesquisa, localizado acima na página inicial, procure na busca popular o serviço **BigQuery** e clique nele:




26) No painel **Explorador** do BigQuery, expanda seu **PROJETO** e o conjunto de dados **cymbal_sales** e selecione a tabela **cymbalsalestable**:

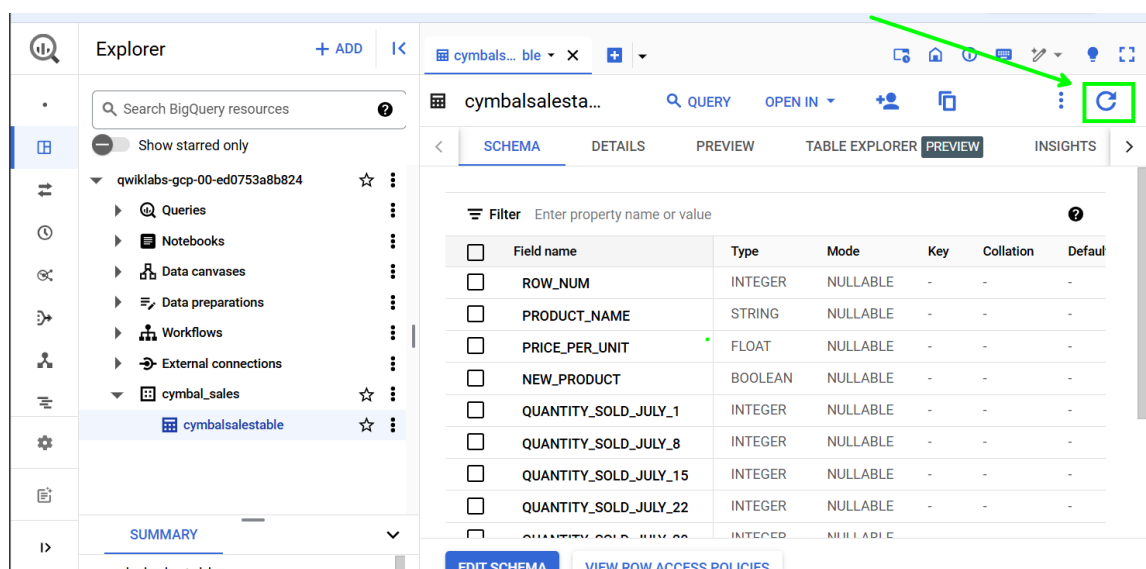


27) Interaja com o Gemini como o laboratório orienta.

28) Abra o **Cloud Shell**, e execute o seguinte comando (Troque os dados em negrito pelo que aparece em seu laboratório, ou copie o comando completo passado para você):

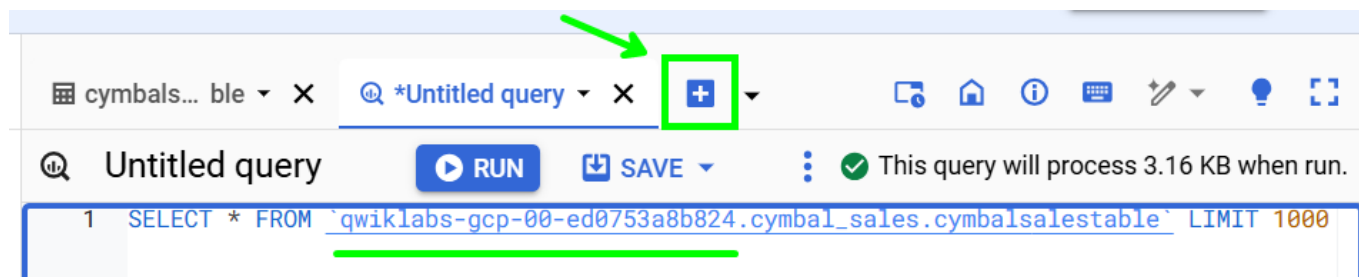
```
bq load --source_format=CSV --autodetect cymbal_sales.cymbalsalestable gs://qwiklabs-gcp-03-25359bd5059c-cymbal-frontend/sales_bq_rawdata.csv
```

29) De volta na tabela, clique em Atualizar :



30) Clique no sinal de + para abrir outro **Editor**, e coloque o comando passado pelo laboratório (Troque o negrito pelas informações dadas a você nessa sessão):

```
SELECT * FROM `qwiklabs-gcp-00-ed0753a8b824.cymbal_sales.cymbalsalestable` LIMIT 1000
```

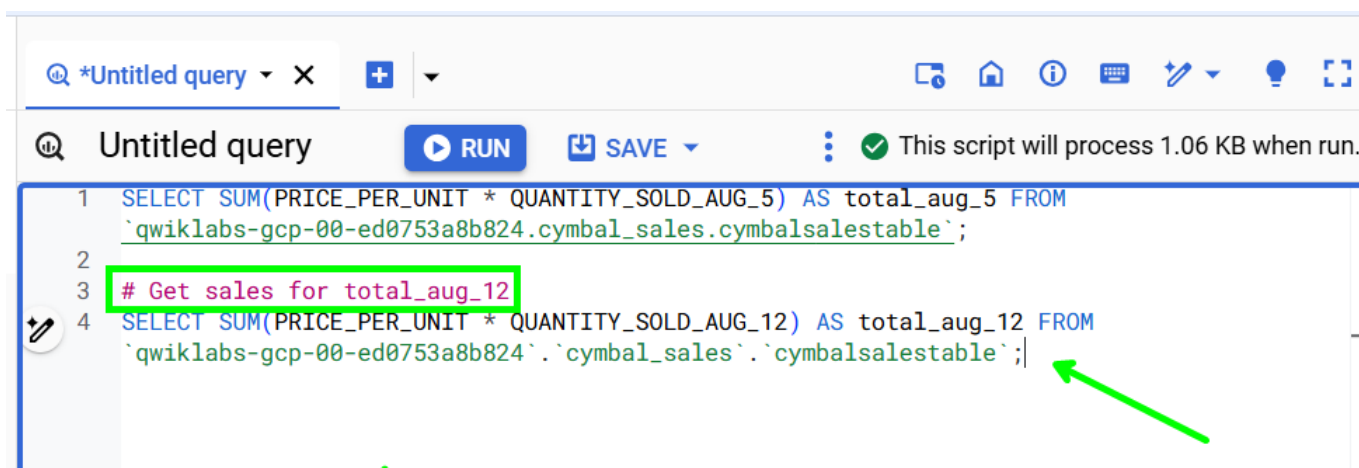


31) Limpe o **Editor**, e coloque a nova consulta sugerida pelo laboratório. Depois coloque o comentário em algumas linhas abaixo, e tecla **Enter** ao final da linha. O Gemini irá gerar uma

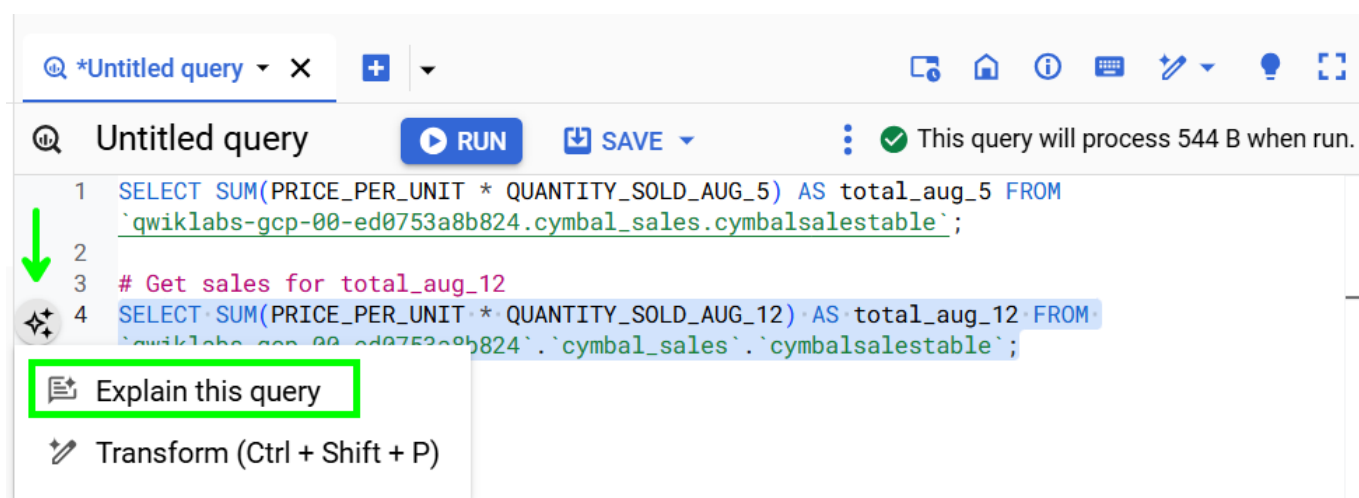
sugestão de código, passe o mouse por cima e clique em **Accept**, ou pressione **Tab** para aceitar o código.

```
SELECT SUM(PRICE_PER_UNIT * QUANTITY_SOLD_AUG_5) AS total_aug_5 FROM
`qwiklabs-gcp-00-ed0753a8b824.cymbal_sales.cymbalsalestable`;
```

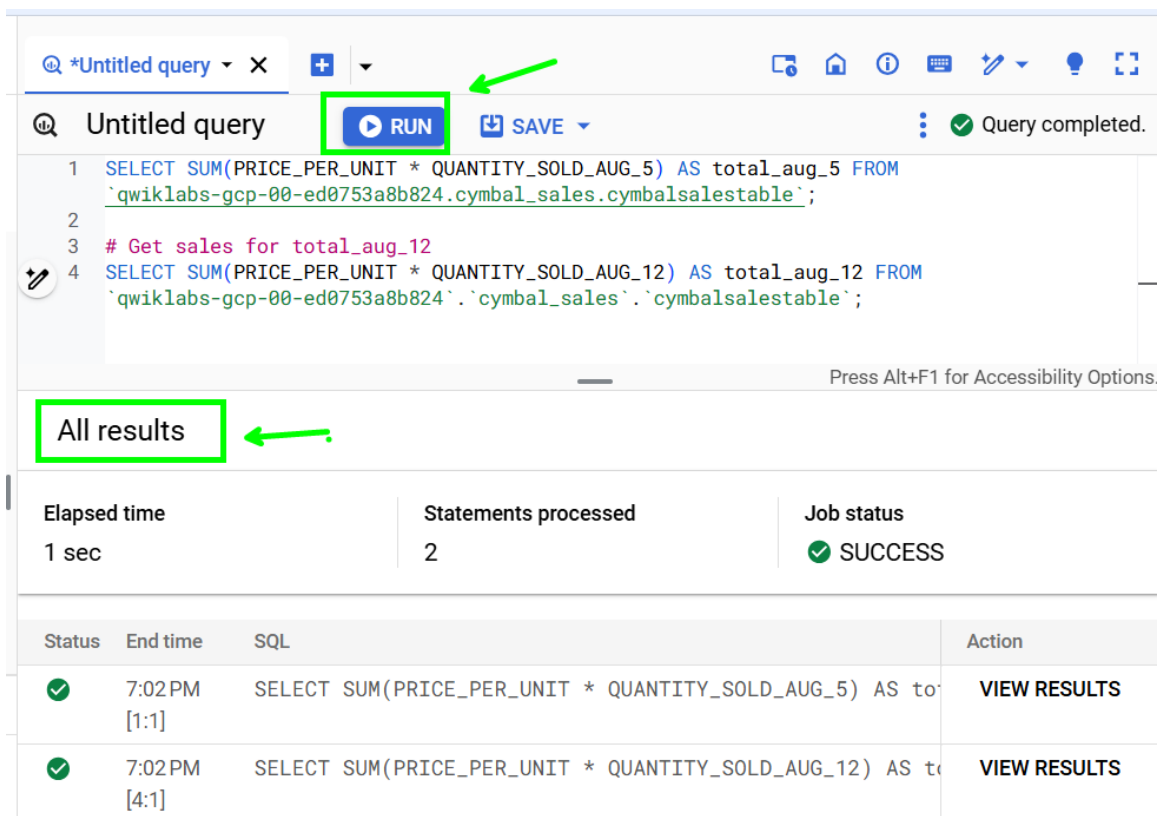
Get sales for total_aug_12



32) Em seguida, selecione a consulta, clique com o botão direito do mouse e selecione **Explain this query**. Também é possível clicar no lápis mágico do Gemini que aparece na margem esquerda:



33) Após a explicação aparecer no chat do Gemini, execute o código clicando no botão azul **RUN**:



Query completed.

```

1 SELECT SUM(PRICE_PER_UNIT * QUANTITY_SOLD_AUG_5) AS total_aug_5 FROM
   `qwiklabs-gcp-00-ed0753a8b824.cymbal_sales.cymbalsalestable`;
2
3 # Get sales for total_aug_12
4 SELECT SUM(PRICE_PER_UNIT * QUANTITY_SOLD_AUG_12) AS total_aug_12 FROM
   `qwiklabs-gcp-00-ed0753a8b824`.`cymbal_sales`.`cymbalsalestable`;

```

Press Alt+F1 for Accessibility Options.

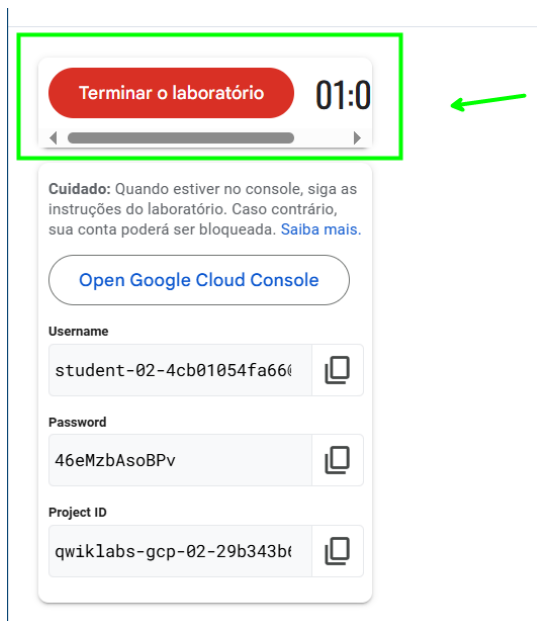
All results

Elapsed time	Statements processed	Job status
1 sec	2	✓ SUCCESS

Status	End time	SQL	Action
✓	7:02 PM [1:1]	SELECT SUM(PRICE_PER_UNIT * QUANTITY_SOLD_AUG_5) AS to	VIEW RESULTS
✓	7:02 PM [4:1]	SELECT SUM(PRICE_PER_UNIT * QUANTITY_SOLD_AUG_12) AS to	VIEW RESULTS

34) Tique a última Task do laboratório, garanta que todas as tarefas foram ticadas no quadrado amarelo flutuante localizado ao lado esquerdo da página:

35) Finalize o laboratório no botão vermelho do painel:



Terminar o laboratório 01:00

Cuidado: Quando estiver no console, siga as instruções do laboratório. Caso contrário, sua conta poderá ser bloqueada. [Saiba mais.](#)

[Open Google Cloud Console](#)

Username
student-02-4cb01054fa66

Password
46eMzbAsoBPv

Project ID
qwiklabs-gcp-02-29b343bt