

[Começar o laboratório](#)

01:00:00

Crie PDFs com Go e Cloud Run

1 hora

Gratuito

**GSP762****Google Cloud Self-Paced Labs**

GSP762

-/100

Visão geral

Objetivos

Configuração

Arquitetura

Use o Googleapis

Faça o download do código-fonte

Crie um microsserviço de fatura

Crie o serviço pdf-conversion

Crie uma conta de serviço

Teste o serviço do Cloud Run

Gatilho do Cloud Storage

Teste a notificação do Cloud Storage

Parabéns!

Terminar o laboratório

Visão geral

Neste laboratório, você criará um app da Web no Cloud Run, um serviço sem servidor, para converter automaticamente em PDF os arquivos armazenados no Google Drive e colocá-los em pastas separadas.

Objetivos

- Converter um aplicativo Go em um contêiner
- Aprender a criar contêineres com o Google Cloud Build
- Criar um serviço do Cloud Run que converte arquivos em PDF na nuvem
- Entender como criar contas de serviço e adicionar permissões
- Usar o processamento de eventos com o Google Cloud Storage

Pré-requisitos

Para este laboratório de **nível fundamental**, é preciso ter familiaridade com o Console do Cloud e o Cloud Shell.

Você também precisa ter noções básicas de edição de arquivos. Escolha um editor de texto (como nano, vi etc.) ou inicie o editor de código do Cloud Shell, disponível na barra superior:



Configuração

Antes de clicar no botão Start Lab

Leia estas instruções. Os laboratórios são cronometrados e não podem ser pausados. O timer é iniciado quando você clica em **Começar o laboratório** e mostra por quanto tempo os recursos do Google Cloud ficarão disponíveis.

Este laboratório prático do Qwiklabs permite que você realize as atividades em um ambiente real de nuvem, não em uma simulação ou demonstração. Você receberá novas credenciais temporárias para fazer login e acessar o Google Cloud durante o laboratório.

O que é necessário

Para fazer este laboratório, você precisa ter:

- acesso a um navegador de Internet padrão (recomendamos o Chrome);
- tempo para concluir as atividades.

Observação: não use seu projeto ou sua conta do Google Cloud neste laboratório.

Observação: se estiver usando um dispositivo Chrome OS, abra uma janela anônima para executar o laboratório.

Como iniciar seu laboratório e fazer login no console do Google Cloud

1. Clique no botão **Começar o laboratório**. Se for preciso pagar, você verá um pop-up para selecionar a forma de pagamento. No painel **Detalhes do laboratório** à esquerda, você verá o seguinte:

- O botão **Abrir Console do Cloud**
- Tempo restante
- As credenciais temporárias que você vai usar neste laboratório
- Outras informações se forem necessárias

2. Clique em **Abrir Console do Google**. O laboratório ativa recursos e depois abre outra guia com a página **Fazer login**.

Dica: coloque as guias em janelas separadas lado a lado.

Observação: se aparecer a caixa de diálogo **Escolher uma conta**, clique em **Usar outra conta**.

3. Caso seja preciso, copie o **Nome de usuário** no painel **Detalhes do laboratório** e cole esse nome na caixa de diálogo **Fazer login**. Clique em **Avançar**.

4. Copie a **Senha** no painel **Detalhes do laboratório** e a cole na caixa de diálogo **Olá**. Clique em **Avançar**.

Importante: você precisa usar as credenciais do painel à esquerda. Não use suas credenciais do Google Cloud Ensina.

Observação: se você usar sua própria conta do Google Cloud neste laboratório, é possível que receba cobranças adicionais.

5. Acesse as próximas páginas:

- Aceite os Termos e Condições.
- Não adicione opções de recuperação nem autenticação de dois fatores (porque essa é uma conta temporária).
- Não se inscreva em testes gratuitos.

Depois de alguns instantes, o console do GCP vai ser aberto nesta guia.

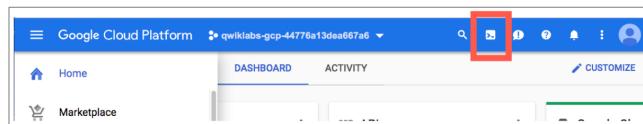
Observação: para ver uma lista dos produtos e serviços do Google Cloud, clique no **Menu de navegação** no canto superior esquerdo.



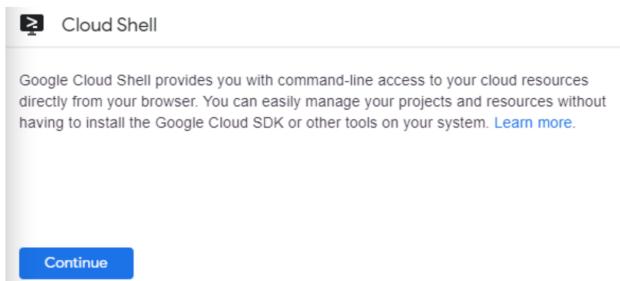
Ative o Google Cloud Shell

O Google Cloud Shell é uma máquina virtual com ferramentas de desenvolvimento. Ele conta com um diretório principal permanente de 5 GB e é executado no Google Cloud. O Google Cloud Shell permite acesso de linha de comando aos seus recursos do GCP.

1. No Console do GCP na barra de ferramentas superior direita, clique no botão **Abrir o Cloud Shell**.



2. Clique em **Continue** (continuar):



Demora alguns minutos para provisionar e conectar-se ao ambiente. Quando você está conectado, você já está autenticado e o projeto é definido como seu **PROJECT_ID**. Por exemplo:

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6)$
```

gcloud é a ferramenta de linha de comando do Google Cloud Platform. Ele vem pré-instalado no Cloud Shell e aceita preenchimento com tabulação.

É possível listar o nome da conta ativa com este comando:

```
gcloud auth list
```

Saída:

```
ACTIVE: *
ACCOUNT: student-01-xxxxxxxxxxxx@qwiklabs.net
To set the active account, run:
  $ gcloud config set account 'ACCOUNT'
```

É possível listar o ID de projeto com este comando:

```
gcloud config list project
```

Saída:

```
[core]
project = <project_ID>
```

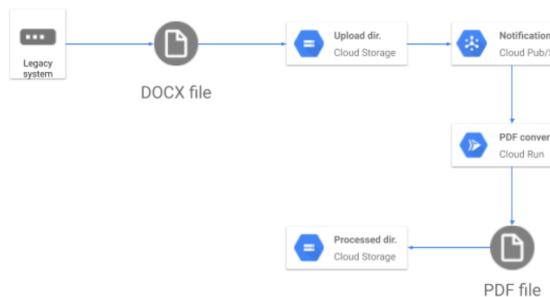
Exemplo de saída:

```
[core]
project = quickstart-gcp-44776a13dea667a6
```

A documentação completa do **gcloud** está disponível na página [Visão geral do gcloud](#) do Google Cloud.

Arquitetura

Neste laboratório, você ajudará a Pet Theory Veterinary a converter automaticamente as faturas dos clientes em formato PDF para facilitar a consulta.



Use o Googleapis

Neste laboratório, você usará as APIs do Google. As seguintes APIs foram ativadas para você:

Nome	API
Cloud Build	cloudbuild.googleapis.com
Cloud Storage	storage-component.googleapis.com
Cloud Run	run.googleapis.com

Faça o download do código-fonte

Para começar, faça o download do código necessário para este laboratório.

- Ative sua conta de laboratório:

```
gcloud auth list --filter=status:ACTIVE --format="value(account)"
```

- Execute este comando para clonar o repositório da Pet Theory:

```
git clone https://github.com/Deleplace/pet-theory.git
```

3. Acesse o diretório correto:

```
cd pet-theory/lab03
```

Crie um microsserviço de fatura

Nesta seção, você criará um aplicativo Go para processar solicitações. Conforme descrito no diagrama da arquitetura, você integrará o Cloud Storage como parte da solução.

Abra o código-fonte `server.go` e edite-o para corresponder ao texto abaixo:

```
package main
import (
    "fmt"
    "io/ioutil"
    "log"
    "net/http"
    "os"
    "os/exec"
    "regexp"
    "strings"
)
func main() {
    http.HandleFunc("/", process)
    port := os.Getenv("PORT")
    if port == "" {
        port = "8080"
        log.Printf("Defaulting to port %s", port)
    }
    log.Printf("Listening on port %s", port)
    err := http.ListenAndServe(fmt.Sprintf(":%s", port), nil)
    log.Fatal(err)
}
func process(w http.ResponseWriter, r *http.Request) {
    log.Println("Serving request")
    if r.Method == "GET" {
        fmt.Fprintln(w, "Ready to process POST requests from Cloud
Storage trigger")
        return
    }
    //
    // Read request body containing GCS object metadata
    //
    gcsInputFile, err1 := readBody(r)
    if err1 != nil {
        log.Printf("Error reading POST data: %v", err1)
        w.WriteHeader(http.StatusBadRequest)
        fmt.Fprintf(w, "Problem with POST data: %v \n", err1)
        return
    }
    //
    // Working directory (concurrency-safe)
    //
    localDir, errDir := ioutil.TempDir("", "")
    if errDir != nil {
        log.Printf("Error creating local temp dir: %v", errDir)
        w.WriteHeader(http.StatusInternalServerError)
        fmt.Fprintf(w, "Could not create a temp directory on
server. \n")
        return
    }
    defer os.RemoveAll(localDir)
    //
    // Download input file from GCS
    //
    localInputFile, err2 := download(gcsInputFile, localDir)
    if err2 != nil {
        log.Printf("Error downloading GCS file [%s] from bucket
[%s]: %v",
            gcsInputFile.Name, gcsInputFile.Bucket, err2)
        w.WriteHeader(http.StatusInternalServerError)
        fmt.Fprintf(w, "Error downloading GCS file [%s] from bucket
[%s]",
            gcsInputFile.Name, gcsInputFile.Bucket)
        return
    }
    //
    // Use LibreOffice to convert local input file to local PDF file.
    //
```

```

localPDFFilePath, err3 := convertToPDF(localInputFile.Name(),
localDir)
if err3 != nil {
    log.Printf("Error converting to PDF: %v", err3)
    w.WriteHeader(http.StatusInternalServerError)
    fmt.Fprintf(w, "Error converting to PDF.")
    return
}
//
// Upload the freshly generated PDF to GCS
//
targetBucket := os.Getenv("PDF_BUCKET")
err4 := upload(localPDFFilePath, targetBucket)
if err4 != nil {
    log.Printf("Error uploading PDF file to bucket [%s]: %v",
targetBucket, err4)
    w.WriteHeader(http.StatusInternalServerError)
    fmt.Fprintf(w, "Error downloading GCS file [%s] from bucket
[%s]",
gcsInputFile.Name, gcsInputFile.Bucket)
    return
}
//
// Delete the original input file from GCS.
//
err5 := deleteGCSFile(gcsInputFile.Bucket, gcsInputFile.Name)
if err5 != nil {
    log.Printf("Error deleting file [%s] from bucket [%s]: %v",
gcsInputFile.Name,
gcsInputFile.Bucket, err5)
    // This is not a blocking error.
    // The PDF was successfully generated and uploaded.
}
log.Println("Successfully produced PDF")
fmt.Println(w, "Successfully produced PDF")
}

func convertToPDF(localFilePath string, localDir string) (resultFilePath
string, err error) {
    log.Printf("Converting [%s] to PDF", localFilePath)
    cmd := exec.Command("libreoffice", "--headless", "--convert-to",
"pdf",
        "--outdir", localDir,
        localFilePath)
    cmd.Stdout, cmd.Stderr = os.Stdout, os.Stderr
    log.Println(cmd)
    err = cmd.Run()
    if err != nil {
        return "", err
    }
    pdfFilePath :=
regexp.MustCompile(`\.\w+$`).ReplaceAllString(localFilePath, ".pdf")
    if !strings.HasSuffix(pdfFilePath, ".pdf") {
        pdfFilePath += ".pdf"
    }
    log.Printf("Converted %s to %s", localFilePath, pdfFilePath)
    return pdfFilePath, nil
}

```

Agora execute este comando para criar o aplicativo:

```
go build -o server
```

As funções chamadas por esse código de nível superior estão nos arquivos de origem:

- server.go
- notification.go
- gcs.go

Depois que o aplicativo for criado, você poderá criar o serviço pdf-conversion.

Crie o serviço pdf-conversion

O serviço de PDF usará o Cloud Run e o Cloud Storage para iniciar um processo toda vez que for feito o upload de um arquivo no armazenamento designado.

Para isso, você usará um padrão comum de notificações de eventos com o Cloud Pub/Sub. Assim, o aplicativo realiza apenas o processamento das informações. O transporte e a transmissão de informações são feitos por outros serviços que mantêm

Adicione o pacote do LibreOffice para usar no seu aplicativo.

A criação do módulo de fatura requer a integração de dois componentes:



Adicione o pacote do LibreOffice para usar no seu aplicativo.

1. Abra o manifesto do `Dockerfile` e atualize o arquivo da seguinte maneira:

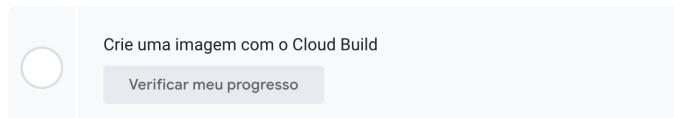
```
FROM debian:buster
RUN apt-get update -y \
    && apt-get install -y libreoffice \
    && apt-get clean
WORKDIR /usr/src/app
COPY server .
CMD [ "./server" ]
```

2. **Salve** o `Dockerfile` atualizado.

3. Inicie a recriação da imagem `pdf-converter` usando o Cloud Build:

```
gcloud builds submit \
--tag gcr.io/$GOOGLE_CLOUD_PROJECT/pdf-converter
```

Clique em **Verificar meu progresso** para conferir se você executou a tarefa acima.



4. Implante o serviço `pdf-converter` atualizado.

OBSERVAÇÃO: é recomendável que o LibreOffice use 2 GB de RAM. Veja a linha com a opção `--memory`.

Execute estes comandos para criar e implantar o contêiner:

```
gcloud run deploy pdf-converter \
--image gcr.io/$GOOGLE_CLOUD_PROJECT/pdf-converter \
--platform managed \
--region us-east1 \
--memory=2Gi \
--no-allow-unauthenticated \
--set-env-vars PDF_BUCKET=$GOOGLE_CLOUD_PROJECT-processed \
--max-instances=3
```

Clique em **Verificar meu progresso** para conferir se você executou a tarefa acima.



O serviço do Cloud Run foi implantado corretamente. No entanto, o aplicativo implantado requer permissões corretas para ser acessado.

Crie uma conta de serviço

A [conta de serviço](#) é um tipo especial de conta com acesso às APIs do Google.

Neste laboratório, você usará uma conta de serviço para acessar o Cloud Run quando um evento do Cloud Storage for processado. O Cloud Storage oferece um conjunto avançado de notificações que podem ser usadas para acionar eventos.

Em seguida, atualize o código para notificar o aplicativo quando for feito o upload de um arquivo.

1. Clique no **Menu de navegação > Armazenamento** e verifique se dois buckets foram criados. Você verá:

- PROJECT_ID-processed
- PROJECT_ID-upload

2. Crie uma notificação do Pub/Sub para indicar que um novo arquivo foi enviado ao bucket de documentos ("uploaded"). As notificações serão marcadas com o tópico "new-doc".

```
gsutil notification create -t new-doc -f json -e OBJECT_FINALIZE  
gs://$GOOGLE_CLOUD_PROJECT-upload
```

3. Crie uma conta de serviço para acionar os serviços do Cloud Run:

```
gcloud iam service-accounts create pubsub-cloud-run-invoker --display-  
name "PubSub Cloud Run Invoker"
```

4. Conceda permissão à conta de serviço para invocar o serviço de conversão em PDF:

```
gcloud run services add-iam-policy-binding pdf-converter \  
--member=serviceAccount:pubsub-cloud-run-  
invoker@$GOOGLE_CLOUD_PROJECT.iam.gserviceaccount.com \  
--role=roles/run.invoker \  
--region us-east1 \  
--platform managed
```

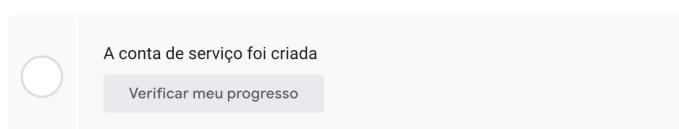
5. Execute este comando para encontrar o número do seu projeto:

```
PROJECT_NUMBER=$(gcloud projects list \  
--format="value(PROJECT_NUMBER)" \  
--filter="$GOOGLE_CLOUD_PROJECT")
```

6. Permita que seu projeto crie tokens de autenticação do Cloud Pub/Sub:

```
gcloud projects add-iam-policy-binding $GOOGLE_CLOUD_PROJECT \  
--member=serviceAccount:service-$PROJECT_NUMBER@gcp-sa-  
pubsub.iam.gserviceaccount.com \  
--role=roles/iam.serviceAccountTokenCreator
```

Clique em **Verificar meu progresso** para conferir se você executou a tarefa acima.



A conta de serviço criada pode ser usada para invocar o serviço do Cloud Run.

Teste o serviço do Cloud Run

Antes de continuar, teste o serviço implantado. Lembre-se de que o serviço requer autenticação. O teste confirmará se apenas pessoas autorizadas podem acessá-lo.

1. Salve o URL do serviço na variável de ambiente `$SERVICE_URL`:

```
SERVICE_URL=$(gcloud run services describe pdf-converter \
--platform managed \
--region us-east1 \
--format "value(status.url)")
```

2. Exiba o SERVICE URL:

```
echo $SERVICE_URL
```

3. Envie uma solicitação GET anônima ao novo serviço:

```
curl -X GET $SERVICE_URL
```

A solicitação GET anônima será respondida com uma mensagem de erro: "Your client does not have permission to get URL". Isso está correto, já que usuários anônimos não têm permissão para chamar o serviço.

4. Agora tente chamar o serviço como um usuário autorizado:

```
curl -X GET -H "Authorization: Bearer $(gcloud auth print-identity-token)" $SERVICE_URL
```

5. A resposta será: "Ready to process POST requests from Cloud Storage trigger"

Parabéns! Você implantou um serviço autenticado do Cloud Run.

Gatilho do Cloud Storage

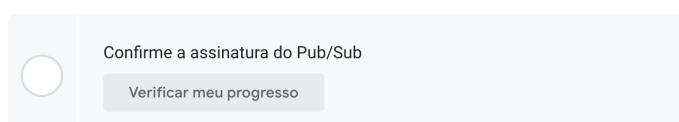
Para ativar uma notificação quando um conteúdo novo for enviado ao Cloud Storage, adicione uma assinatura ao tópico do Pub/Sub atual.

Lembrete: as notificações do Cloud Storage enviarão automaticamente uma mensagem para sua fila de tópicos quando um conteúdo novo for enviado. O uso de notificações permite criar aplicativos que respondem a eventos sem precisar escrever mais código.

Crie uma assinatura do Pub/Sub para executar o conversor em PDF sempre que uma mensagem for publicada no tópico `new-doc`:

```
gcloud pubsub subscriptions create pdf-conv-sub \
--topic new-doc \
--push-endpoint=$SERVICE_URL \
--push-auth-service-account=pubsub-cloud-run-
invoker@$GOOGLE_CLOUD_PROJECT.iam.gserviceaccount.com
```

Clique em *Verificar meu progresso* para conferir se você executou a tarefa acima.



Agora sempre que é feito o upload de um arquivo, a assinatura do serviço do Pub/Sub interage com sua conta. A conta de serviço iniciará o serviço de conversão em PDF do Cloud Run.

Teste a notificação do Cloud Storage

Para testar o serviço do Cloud Run, use os arquivos de exemplo disponíveis.

1. Copie os arquivos de teste no bucket de upload:

```
gsutil -m cp -r gs://splz/gsp762/* gs://$GOOGLE_CLOUD_PROJECT-upload
```

2. No Console do Cloud, observe o bucket de **upload** do Cloud Storage e clique no botão **Atualizar bucket** algumas vezes para confirmar que os arquivos são excluídos, um por um, à medida que são convertidos em PDF. Em seguida, clique em **Armazenamento** e depois no bucket com o nome que termina em "-processed". Ele contém as versões em PDF de todos os arquivos. Abra os arquivos PDF para verificar se eles foram convertidos corretamente.
3. Quando o upload for concluído, use o **Menu de navegação** para acessar o **Cloud Run** e clique no serviço **pdf-converter**.

The screenshot shows the Google Cloud Platform Cloud Run interface. At the top, there are tabs for 'Cloud Run' (selected), 'Services', '+ CREATE SERVICE', and 'MANAGE CUSTOM DOMAINS'. Below this, a message states: 'Each Cloud Run service has a unique endpoint and autoscales deployed containers. [Learn more](#)'. A 'Filter services' dropdown is open, showing 'pdf-converter' selected. The table below lists the service details:

Name	Req/sec	Region	Authentication	Connectivity
pdf-converter	0	us-central1		External

Below the table, a section for the 'pdf-converter' service is shown. It includes the service name, region (us-central1), URL (https://pdf-converter-ufnifnovq-uc.a.run.app), and a 'Logs' button. The 'Logs' tab is selected, showing a list of log messages:

Logs	Showing 6 messages
Default ▾	
Filter	
Converting	
Scanned up to 9/20/20, 8:59 AM. Scanned 1.3 GB.	
2020-09-23T18:33:30.816356Z 2020/09/23 18:33:38 Converting [/tmp/404833929/file-sample_100kB.doc] to PDF	
2020-09-23T18:33:42.391673Z 2020/09/23 18:33:42 Converting [/tmp/147146550/file-sample_100kB.doc] to PDF	
2020-09-23T18:51:36.534653Z 2020/09/23 18:51:36 Converting [/tmp/102174502/file-sample_100kB.doc] to PDF	
2020-09-23T18:51:37.027572Z 2020/09/23 18:51:37 Converting [/tmp/120393293/file-sample_100kB.doc] to PDF	
2020-09-23T18:51:46.569592Z 2020/09/23 18:51:46 Converting [/tmp/449872280/file-sample_100kB.doc] to PDF	
2020-09-23T18:51:47.278092Z 2020/09/23 18:51:47 Converting [/tmp/493053447/file-sample_100kB.doc] to PDF	

4. Selecione a guia **Registros** e adicione um filtro para "Converting" e veja os arquivos convertidos.

5. No **Menu de navegação**, acesse **Armazenamento**, abra a pasta "upload" e confirme se todos os arquivos foram processados.

Excelente trabalho! Você criou um serviço para converter os arquivos enviados ao Cloud Storage em PDF.

Parabéns!

O serviço pdf-converter converte documentos em PDF e os armazena no bucket "processed". Neste laboratório, você aprendeu a:

- Converter um aplicativo Go em um contêiner
- Criar contêineres com o Google Cloud Build
- Criar um serviço do Cloud Run
- Conceder permissões usando uma conta de serviço
- Usar o processamento de eventos do Google Cloud Storage

Termine a Quest

Este laboratório autoguiado é parte da Quest [Workshop do Google Cloud Run sem servidor](#) do Qwiklabs. Uma Quest é uma série de laboratórios relacionados que formam o programa de aprendizado. Concluir esta Quest dá a você o selo acima como reconhecimento pela sua conquista. Você pode publicar os selos e incluir um link para eles no seu currículo on-line ou nas redes sociais. Caso você já tenha feito este laboratório, inscreva-se nesta Quest para ganhar os créditos de conclusão imediatamente. Veja outras [Quests do Qwiklabs](#).

Saiba mais / Próximas etapas

- Contêineres sem servidor: [vídeo da Next 19 no YouTube](#)

Terminar o laboratório

Após terminar seu laboratório, clique em **End Lab**. O Qwiklabs removerá os recursos usados e limpará a conta para você.

Você poderá classificar sua experiência neste laboratório. Basta selecionar o número de estrelas, digitar um comentário e clicar em **Submit**.

O número de estrelas indica o seguinte:

- 1 estrela = muito insatisfeito
- 2 estrelas = insatisfeito
- 3 estrelas = neutro
- 4 estrelas = satisfeito
- 5 estrelas = muito satisfeito

Feche a caixa de diálogo se não quiser enviar feedback.

Para enviar seu feedback, fazer sugestões ou correções, use a guia **Support**.

Manual atualizado em 30 de dezembro de 2020

Laboratório testado em 30 de dezembro de 2020

Copyright 2020 Google LLC. Todos os direitos reservados. Google e o logotipo do Google são marcas registradas da Google LLC. Todos os outros nomes de produtos e empresas podem ser marcas registradas das respectivas empresas a que estão associados.

Continuar a Quest

