1. Criar E-mail Google ou usar 1 particular (o @portalsesisp.org.br da erro)  
   <<[SeuEmail@gmail.com](mailto:SeuEmail@gmail.com)>>
2. Cadastrar no ClassRoom com o <<[SeuEmail@gmail.com](https://github.com/paulobrandaodev/paulobrandaodev)>>:  
   <https://classroom.google.com/>
3. Solicitar para entrar na turma do ClassRoom para:   
   **Daniel Felipe Dos Santos Silva -** [**daniel.fsilva@sp.senai.br**](https://projetosintegrados.senai.info/)
   1. Link para entrar na turma:[https://classroom.google.com/c/NTM2NjY2OTI5MTk4?cjc=sofjgnl](https://github.com/?cjc=sofjgnl)
4. Cadastrar no CloudSkillsBoost com <<[SeuEmail@gmail.com](https://cloud.google.com/certification)>>:  
   <https://www.cloudskillsboost.google>

***-- Sempre abrir o CloudSkillsBoost na guia anônima (CTRL+SHIFT+N).***

IMPOTANTE: usar o tradutor do CloudSkillsBoost  (não use o tradutor do navegador)

1. Solicitar Créditos para testes no laboratório GoogleCloud para:  
   **Daniel Felipe Dos Santos Silva -** [**daniel.fsilva@sp.senai.br**](mailto:daniel.fsilva@sp.senai.br)
   1. Como liberar créditos para o laboratório CloudSkillsBoost:[https://witty-cornet-caa.notion.site/Libera-o-de-cr-ditos-para-aluno-39d1ba3ca202469cb5158d39449ac8f5](https://classroom.google.com/c/NTM2NjY2OTI5MTk4)
2. Ler a Documentação:
   1. Máquinas Virtuais no Google Cloud Platform  
      [https://cloud.google.com/compute/docs/machine-types?hl=pt-br](mailto:SeuEmail@gmail.com?hl=pt-br)
   2. Armazenamento de Arquivos com Cloud Storage:  
      <https://cloud.google.com/storage/docs/introduction>
3. Realizar os Laboratórios na sequência (Dia1): (**IMPORTANTE: Usar a guia anônima)**
   1. Conhecer o QuickLabs da Skills Boost (Laboratório de estudos):  
      [https://www.cloudskillsboost.google/focuses/2794?locale=pt\_BR&parent=catalog](https://git-scm.com/downloads?locale=pt_BR&parent=catalog)
   2. Crie uma máquina virtual com o Console do Cloud (Navegação Intuitiva):  
      <https://www.cloudskillsboost.google/focuses/3563?catalog_rank=%7B%22rank%22%3A1%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=17544002>
   3. Crie uma máquina virtual com a Gcloud(linha de comando Cloud Shell).

<https://www.cloudskillsboost.google/focuses/560?catalog_rank=%7B%22rank%22%3A1%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=17544021>  
  
*Obs: Quando executar o Cloud Shell insira esse comando gcloud auth list para aparecer uma janela para clicar em autorizar.*

* 1. Responder o Formulario Abaixo (verificar informações na documentação no item 6 acima):  
     [https://forms.gle/xfBfMsdBt5CQZzrF9](https://www.cloudskillsboost.google/course_templates/154)

1. Compreendendo o uso de APIs (Dia 2)   
   1. Conceito de API;
   2. Request e Response / Pergunta e Resposta
   3. Requisições e respostas padronizadas / Transmissão de pedidos através do protocolo HTTP / Uso da linguagem JSON para pedidos e respostas
   4. Métodos HTTP mais usados: GET/POST/PUT/DELETE
   5. Exemplos e Ferrametas para utilizar e retornar dados de API:
      1. Usar cmd e digitar Curl <https://UrlDaApiQueVcForUsar>
      2. [https://www.postman.com/](https://witty-cornet-caa.notion.site/Libera-o-de-cr-ditos-para-aluno-39d1ba3ca202469cb5158d39449ac8f5) - cadastre com a conta do google <<[SeuEmail@gmail.com](https://cloud.google.com/compute/docs/machine-types)>>:
      3. VSCode Rest Client / HTTPie / Insomnia / Paw
   6. Atividade com uso de API (Todos os dados de Pokémon que você precisa em um só lugar,  
      facilmente acessível por meio de uma API RESTful moderna.)
      1. Entrar no site para utilizar a API Pokemon:  
         <https://pokeapi.co/>
      2. No campo de pesquisa digite pokemon/200 (ou qualquer número que desejar) para exibir o link com os dados do pokemon com a id 200:  
         Direct link to results: [https://pokeapi.co/api/v2/pokemon/200](https://www.cloudskillsboost.google/focuses/2794)
      3. Copie a URL gerada e vá para o site postman e descobra quais dados retornará dessa API  
         <https://www.postman.com/>
      4. Localize a linha com o código:   
         "official-artwork": {  
         "front\_default": [https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/200.png](https://www.postman.com/)  
         }
      5. Copie a url apresentada e cole no navegador para descobrir a imagem oficial do pokemon numero 200 ou o número que você digitou.  
         [https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/200.png](mailto:SeuEmail@gmail.com)
      6. Formulario Dia 2 – Uso de API  
         https://forms.gle/UxfREidjAQAs5
2. Laboratório: Criação de Servidor de Arquivos (Dia 2)   
   1. Servidor de Arquivos com o Console do Cloud (Navegação Intuitiva):  
      1 - <https://www.cloudskillsboost.google/focuses/1760?parent=catalog>
   2. Servidor de Arquivos com a Gcloud(linha de comando Cloud Shell):  
      2 - [https://www.cloudskillsboost.google/focuses/569?parent=catalog](https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/200.png?parent=catalog)
3. Links dos Quests Google (DIA 2): -- abrir na guia anônima (CTRL+SHIFT+N).  
   1. 1 - Create and Manage Cloud Resources  
      [https://www.cloudskillsboost.google/quests/120](mailto:SeuEmail@gmail.com)
   2. 2 - Perform Foundational Infraestructure Tasks   
      <https://www.cloudskillsboost.google/quests/118>
   3. 3 - Build and Secure Networks   
      [https://www.cloudskillsboost.google/quests/128](https://pokeapi.co/api/v2/pokemon/200)
   4. 4 - Perform Foundational Data, ML and AI Tasks   
      <https://www.cloudskillsboost.google/quests/117>
4. Links dos Cursos Google (DIA 2): -- abrir na guia anônima (CTRL+SHIFT+N).  
   1. 1 - Cloud Computing Fundamentals  
      <https://www.cloudskillsboost.google/course_templates/177>
   2. 2 - Infrastructure in Google Cloud  
      <https://www.cloudskillsboost.google/course_templates/154>
   3. 3 - Networking and Security in Google Cloud  
      <https://www.cloudskillsboost.google/course_templates/155>
   4. 4 - Data, ML, and AI in Google Cloud  
      <https://www.cloudskillsboost.google/course_templates/156>
5. Git - versionamento e GitHub (Dia 3)  
   GitHub plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git
6. Crie uma conta no GitHub - Cadastrar com o <<[SeuEmail@gmail.com](https://www.cloudskillsboost.google/focuses/569)>>:  
   [https://github.com/](https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/200.png)
7. Download do Git:  
   <https://git-scm.com/downloads>  
   1. Instalar o Git ( NEXT sempre até o final)
8. Verifique se o git foi instalado:  
   abra o cmd, digite: git - - help 🡪enter
9. Configurar git no computador:
   1. No terminal Git Bash digite:  
      git config - -global user.email “[SeuEmail@gmail.com](https://www.cloudskillsboost.google/quests/128)” (e-mail cadastrado no GitHub)  
      git config - -global user.name “[Seu](mailto:SeuEmail@gmail.com) Nome Sobrenome” (ficará atrelado a versao do código que vc alterar)
10. Criar seu primeiro repositório git:
    1. Crie uma pasta para seu projeto, ex: <<Projeto1SesiSenai>>
    2. Abra a pasta e dentro da pasta iremos inicializar nosso repositório GIT
    3. Botão dir / Git Bash Here / (irá abrir a linha de comando do GIT para a pasta)
    4. Digite: git init 🡪 enter
    5. Na pasta, habilite a opção para exibir itens ocultos (a pasta .git ira aparecer)
    6. Na raiz da pasta <<Projeto1SesiSenai>> abra um bloco de notas e escreva uma lista com 5 tens e salve como lista.txt (poderia ser qualquer arquivo, nesse exemplo será .txt)
    7. Digite no console do git: git status
    8. Ira aparecer:
       1. No commits yet (não existe nenhuma versão do código ainda)
       2. Ira aparecer os arquivos que ainda não foram adicionados no controle de versão  
          no nosso caso o lista.txt
    9. Vamos incluir a lista.txt no controle de versão  
       digite: git add “lista.txt” 🡪 enter
    10. Para adicionar vários arquivos ao mesmo tempo  
        digite: git add . (tem que digitar o ponto)
11. Criar a versão utilizando a função COMMIT  
    digite: git commit –m “versão inicial” (dentro das “ ” você escreve a informação referente aquela versão)  
    1. Não irá funcionar se o git não estiver configurado com sua conta do GitHub
12. Como enviar a versão do código para o GitHub:  
    digite: git push 🡪 enter  
    Obs... não ira enviar por não ter o link do projeto no github
13. Criar o projeto no github
    1. New – novo projeto
    2. Digite o nome do repositório – Ex, projeto1
    3. Definir repositório publico ou privado - (no inicio publico por enquanto)
    4. [x] Adicionar um arquivo README
    5. Criar repositório
    6. Pegue o link gerado que esta na url do navegador (ctrl+c)  
       o link contem: github.com/NomeDoSeuUsuarioGitHub/nomedorepositorio
    7. Volte para a linha de comando do GIT
    8. Digite: git remote add origin <https://github.com/NomeDoSeuUsuarioGitHub/nomedorepositorio>
14. Enviar o Código após criar e configurar o repositório do GitHub
    1. Git push - - set-upstream origin master
    2. Faça o login com a conta do github (escolha abrir no navegador e escolha autorizar)
15. Sequência para enviar mais arquivos:
    1. Crie ou faça a alteração no arquivo
    2. Adicione os novos arquivos no repositório ( comando: git add .)
    3. Registre a alteração (comando: git commit -m ”nome da alteração”)
    4. Envie para o repositório no github (comando: git push)
    5. Verifique no github se esta ok
16. Verificar históricos de versão no git e subir versão anterior
    1. Git reflog (exibe as versões do repositório)
    2. Copie o id da versão que queira retornar
    3. Digite o comando: git reset - -hard idDaVersao  
       Ex: git reset - - hard d87e695
    4. O arquivo na pasta local já ficara com a versão escolhida
17. Comando para Trazer a versão online para a maquina local:

Comando: Git pull

1. Clonar repositório que está no GitHub:
   1. Crie uma pasta: NomeProjeto
   2. Abra a pasta / botão dir / Git Bash Here / (irá abrir a linha de comando do GIT para a pasta)
   3. Digite: git clone <https://github.com/paulobrandaodev/landingpageexample.git> (irá clonar a pasta que esta no repositório no GitHub.
2. Github professor Paulo  
   [paulobrandaodev/**paulobrandaodev**](https://forms.gle/xfBfMsdBt5CQZzrF9)
3. Treinamento completo de GitHub:

<https://www.youtube.com/watch?v=kB5e-gTAl_s&ab_channel=DevAprender>

1. Comandos GitHub

aaa

1. **Laboratório:**  
   [https://www.cloudskillsboost.google/focuses/588?parent=catalog](mailto:SeuEmail@gmail.com?parent=catalog)
2. **Não esqueça de responder o formulário abaixo:**  
   [https://forms.gle/71S55e9pzV3U7ukf6](https://www.cloudskillsboost.google/quests/120)
3. Equipe Google cloud Computing Fundations Academy – Adriana Petenguill (Dia 4)
   1. Curso 1 - Fundamentos de nuvem
   2. Curso 2 - Infraestrutura
   3. Curso 3 - Redes e segurança
   4. Curso 4 - Aprendizado de maquina IA ( inteligência artificial)
   5. [support@qwiklabs.com](mailto:support@qwiklabs.com) – em português e resposta em 24h ou chat Ingles
4. Cadastrar os créditos
   1. Abrir na janela anônima – sempre para acessar o cloud skills boost
   2. <https://www.cloudskillsboost.google/catalog_lab/1281?qlcampaign=4p-EDUCR-GCCFSESISP-JUL22-90>
      1. Clique em "entrar" e escreva o e-mail que você cadastrou no formulário.
      2. Clique em "Iniciar o laboratório" e faça "Um tour pelo Qwiklabs e Google Cloud" (certifique-se de passar pelo menos cinco minutos no laboratório para receber os 200 créditos em sua conta).
      3. Quando finalizar o laboratório, clique em "Concluir laboratório".
      4. Caso não receba os créditos envie um e-mail informando que não recebeu os creditos para [support@qwiklabs.com](mailto:support@qwiklabs.com)
      5. Se precisar de suporte envie o código da url da atividade que esta com erro no chat ou email: 4p-EDUCR-GCCFSESISP-JUL22-90
      6. Link para verificar se os créditos estão ok:  
         <https://www.cloudskillsboost.google/my_account/credits>

IMPORTANTE: após os 200 créditos creditarem em sua conta, se inscreva e inicie os cursos e nas quests abaixo:

1. Links dos Quests Google (DIA 4): -- abrir na guia anônima (CTRL+SHIFT+N).  
   1. 1 - Create and Manage Cloud Resources  
      [https://www.cloudskillsboost.google/quests/120](mailto:SeuEmail@gmail.com)
   2. 2 - Perform Foundational Infraestructure Tasks   
      <https://www.cloudskillsboost.google/quests/118>
   3. 3 - Build and Secure Networks   
      [https://www.cloudskillsboost.google/quests/128](https://pokeapi.co/api/v2/pokemon/200)
   4. 4 - Perform Foundational Data, ML and AI Tasks   
      <https://www.cloudskillsboost.google/quests/117>
2. Links dos Cursos Google (DIA 4) : -- abrir na guia anônima (CTRL+SHIFT+N).  
   1. 1 - Cloud Computing Fundamentals  
      <https://www.cloudskillsboost.google/course_templates/177>
   2. 2 - Infrastructure in Google Cloud  
      <https://www.cloudskillsboost.google/course_templates/154>
   3. 3 - Networking and Security in Google Cloud  
      <https://www.cloudskillsboost.google/course_templates/155>
   4. 4 - Data, ML, and AI in Google Cloud  
      <https://www.cloudskillsboost.google/course_templates/156>
3. Cluster – Conceitos e Aplicações:
   1. Combinação de vários computadores para processar o mesmo sistema para potencializar o desempenho  
      Ex de Cluster: https://www.youtube.com/watch?v=qfIyuEPIzvU
      1. Cluster de alta disponibilidade;
      2. Elastic Pool – para necessidade de alta disponibilidade por um período específico (aumenta a qtd de nós para atender a demanda)
4. Container – caixa com processos isolados. Para reutilizar quando necessário.
   1. Docker – software para criação de container (win 10 pro)  
      baixar Docker <https://www.docker.com/get-started/>
   2. Descobrir a versão do Docker> docker –version
   3. Cadastrar no site Docker com <<[SeuEmail@gmail.com](https://cloud.google.com/certification)>>
      1. como utilizar container no Docker – pesquisar vídeo.
5. Kubernetes – gerenciador de containers
   1. Garante os serviços dos containers funcionando... se parar 1 container parar ele gerencia e passa para outro container continuar os processos do que parou. - ele sobe 1 container no lugar do que parar.
   2. Faz balanceamento de carga entre os containers para o serviço ficar estável
   3. Autocorreção
6. Dados   
   engenharia de dados = transformação dos dados  
   Analista de dados = analise  
   1. Estruturado
      1. Banco de dados sql
   2. Semi estruturado
   3. Não estruturado
7. Atividade 04 - Containers, Clusters e Dados  
   - Subir um container do Apache seguindo a documentação oficial do Docker Hub

- Clonar o seguinte repositório:

- <https://github.com/paulobrandaodev/landingpageexample.git>

1. Links Importantes:  
   1. Certificação goolge:  
      [https://cloud.google.com/certification?hl=pt-br](https://github.com/?hl=pt-br)
   2. Formulário para certificado Senai  
      <https://forms.office.com/r/8sHcTWMbWD>
   3. Git Versionamento  
      [https://git-scm.com/downloads](mailto:daniel.fsilva@sp.senai.br)
   4. GitHub – repositório de Arquivos  
      [https://github.com/](https://forms.gle/71S55e9pzV3U7ukf6)
   5. VSCode -   
      <https://code.visualstudio.com/download>
   6. Github professor Paulo  
      [paulobrandaodev/**paulobrandaodev**](https://github.com/paulobrandaodev/paulobrandaodev)
   7. Download do SDK do CLI gcloud  
      <https://cloud.google.com/sdk/docs/install?hl=pt-br>
   8. Projeto Integrado Senai

<https://projetosintegrados.senai.info/>

* 1. Lista de comandos Git

criar versões do código com commit

git commit -m "commit inicial"

como enviar alterações usando git push

git push

git remote add origin (+link do repositório)]

git push

colocar branches

git push --set-upstream origin master (fazer login pela primera vez e atualizar pagina git para verificação)

ciclo de atualização do código

git status

git add .

git status (novamente)

git commit -m "permitir cadastrar novos usuarios" (criar próxima versão, exemplo)

git push (enviar para controle de versão)

como verificar histórico de atualizações

git reflog

como navegar entre versões do seu código

git reset --hard (+id da versão)

como criar branches

git branch

git branch (+nome > "staging")

mudar de branch

git branch (escolha a que deseja mudar)

git checkout (+nome da branch > staging)

como unir o código git merge

git checkout (+nome da brench que irá receber atualizações >master)

git merge (+nome da brench que irá enviar atualizações>staging)

git pull (trazer atualizações do sevidor para a máquina local)

gir merge staging

pull request

git checkout (+nome)

git add . (caso haja atualização no código)

git commit -m""(+nome)

git push

------------------------------------------------------------------

git hub site

compare & pull request

escolher branch para mediar

create pull request

merge pull request

como excluir arquivos de controle de versão

touch .gitignore

git add .

git commit -m""

git push

git init = inicializa um novo repositório

git add . = adiciona os arquivos atuais ao próximo commit

git status = verificar o status atual dos repositórios git

git commit -m "mensagem do commit" = cria um novo commit com uma mensagem

git push = envia as atualizações para a nuvem na branch atualmente ativa

git branch = permite listar e ver qual branch está ativa atualmente

git checkout nome-da-branch = permite mudar para uma nova branch

git checkout -b "nome da branch de origem" "nome da nova branch" = permite mudar e cirar uma nova branch com base em outra

git merge "branch rceber merge" = permite fazer o merge da branch ativa atualmente com outra branch

git pull atualiza a branch atualmente ativa