

# HACKATHON

## MBA - Arquitetura de Soluções

Introdução

Contexto

[Sobre a SpotMusic](#)

Etapas

Fase 1- Explorando o AS IS

Fase 2- Implementando a PoC

A. Controle de Versão

B. Desacoplamento da Aplicação

C. Automatização da Infraestrutura

D. Automatização do Fluxo de Desenvolvimento

Fase 3 - Documento da Arquitetura de Solução

## Introdução

O DevOps evoluiu nos últimos anos, deixando de ser apenas uma abordagem alternativa para se tornar uma das principais estratégias de TI. Empresas, startups e indivíduos adoram usar as práticas de DevOps para simplificar seus fluxos de trabalho existentes e alcançar o máximo de eficiência.

**Este Hackathon convida você a mostrar o poder do DevOps ao mundo e tornar o mundo um lugar melhor ao fazê-lo.**

Em um cenário lúdico, você e seu grupo passarão por uma experiência que terá muito hands on para que possam trazer toda a experiência adquirida no MBA.

Viveremos essa experiência do hackathon sob uma perspectiva de Arquiteto de Soluções então é importante reforçar alguns pontos sobre o perfil desse profissional, que devem servir de referência para trilhar essa maratona:

- ☐ O arquiteto de solução precisa ter a **capacidade de compartilhar e comunicar ideias** com clareza, tanto oralmente como **por escrito**, aos executivos, patrocinadores do negócio e recursos técnicos em uma linguagem clara e concisa, ou seja, na linguagem que cada grupo seja capaz de compreender.
- ☐ **Habilidade** para preparar e **apresentar propostas técnicas** de alta complexidade, assim como uma visão da solução e atuar na evangelização de como o produto endereça as necessidades críticas do usuário.
- ☐ **Apoiar as demonstrações de soluções com provas de conceito**, avaliações, demonstrações ou mesmo auxiliar na instalação de produtos quando necessário.
- ☐ Habilidade para se **envolver tanto tecnicamente** com os usuários **como com gerentes de unidades de negócio** para entender os requisitos dos usuários, mapeá-los e desenhar uma solução.

## Contexto

Você e seu grupo foram recém contratados pelos donos da SpotMusic para realizar uma consultoria que prevê a entrega de 2 serviços:

- **Uma PoC** (Proof of Concept, ou “prova de conceito”) que prevê implementar de forma mais moderna seu principal produto de armazenamento de Playlists Musicais.
- **Um Documento da Arquitetura de Solução** que terá como principal objetivo ser um guia para a estruturação dos times, como também servirá para a construção do roadmap de tecnologia da empresa.

A expectativa na finalização do trabalho é que exista basicamente dois entregáveis:

- **PoC funcionando no Google Cloud** trazendo os conceitos de Cloud Native (Arquitetura de Micro Serviço usando práticas de uma cultura DevOps que prioriza ambientes automatizados).
- **Documento com o desenho da solução e tomadas de decisões do seu grupo.**

### Importante:

O escopo da PoC já foi desenhado e acordado para a implementação. Seu grupo não precisará se preocupar em definir o que será implementado. Vocês serão avaliados referente aos itens implementados com sucesso.

Com relação ao Documento da Arquitetura da Solução, seu grupo é livre para trazer pontos que acreditam ser relevantes para constar no desenho final, principalmente, seguindo os pilares de Arquitetura. Mais detalhes desse entregável serão explicados mais à frente.

### Sobre a SpotMusic

A SpotMusic é uma startup que acabou de receber um Startup Funding (financiamento inicial – ou capital inicial – é o dinheiro necessário para lançar um novo negócio) com o objetivo de garantir que a empresa tenha os recursos necessários para evoluir aspectos importantes da tecnologia visando segurança, escalabilidade e disponibilidade, pois em paralelo a essa ação na área de Tecnologia também está sendo investido na área de marketing para criar a campanha de lançamento do produto.

A empresa basicamente nasceu da ideia de dois alunos de graduação que enxergaram um potencial produto em seu trabalho de conclusão de curso e resolveram mergulhar de cabeça nessa ideia.

Sem muitos conhecimentos sobre arquitetura moderna, porém, usando a facilidade da Cloud, eles criaram e lançaram o produto.

### Etapas

#### Fase 1- Explorando o AS IS

O primeiro desafio do grupo será entender como o ambiente atual está e garantir acesso ao ambiente disponibilizado pela SpotMusic, que basicamente é um clone do ambiente de produção.

Ao menos um integrante do grupo já tem acesso ao projeto disponibilizado para vocês, os demais integrantes devem ser cadastrados por um colega que já tenha acesso (IAM).

Pensando numa cloud sustentável que não deixa recursos não usados sendo cobrados, o servidor encontra-se desligado. Iniciar ligando e acessando o mesmo é um bom ponto de partida! :)

**Dica:**

É extremamente relevante que em seu Documento da Fase 3 tenha a descrição de como está o ambiente AS IS.

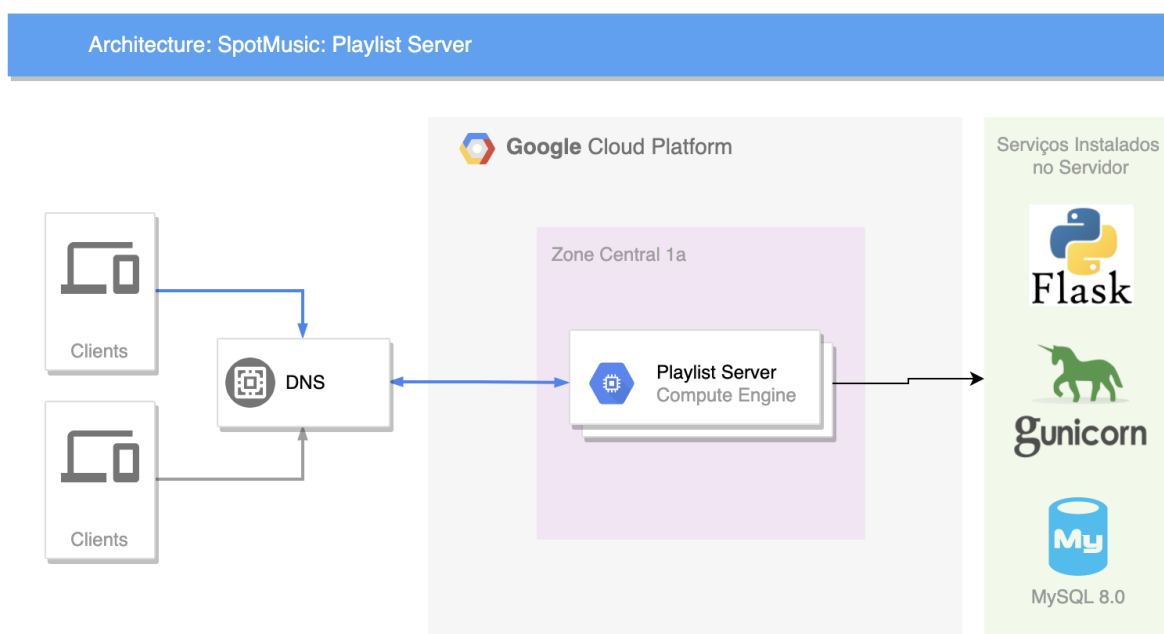
# HACKATHON

## MBA - Arquitetura de Soluções

Foi solicitado à Spot Music que entregasse minimamente uma documentação de como o ambiente está hoje e quais são os processos do ciclo de vida da aplicação atualmente.

### Abaixo o que foi passado pela empresa:

Nossa aplicação foi desenvolvida em Python Flask usando banco de dados MySQL para armazenamento de dados. A aplicação roda 100% em nuvem pública do Google (GCP), usando o serviço de Compute Engine (Máquina Virtual).



Conforme desenho acima, todos os serviços atualmente rodam no mesmo servidor que foi construído de forma manual. Não se preocupe muito com o código da aplicação que roda nesse monolito, pois no Kit Hackathon você já recebeu o código atualizado.

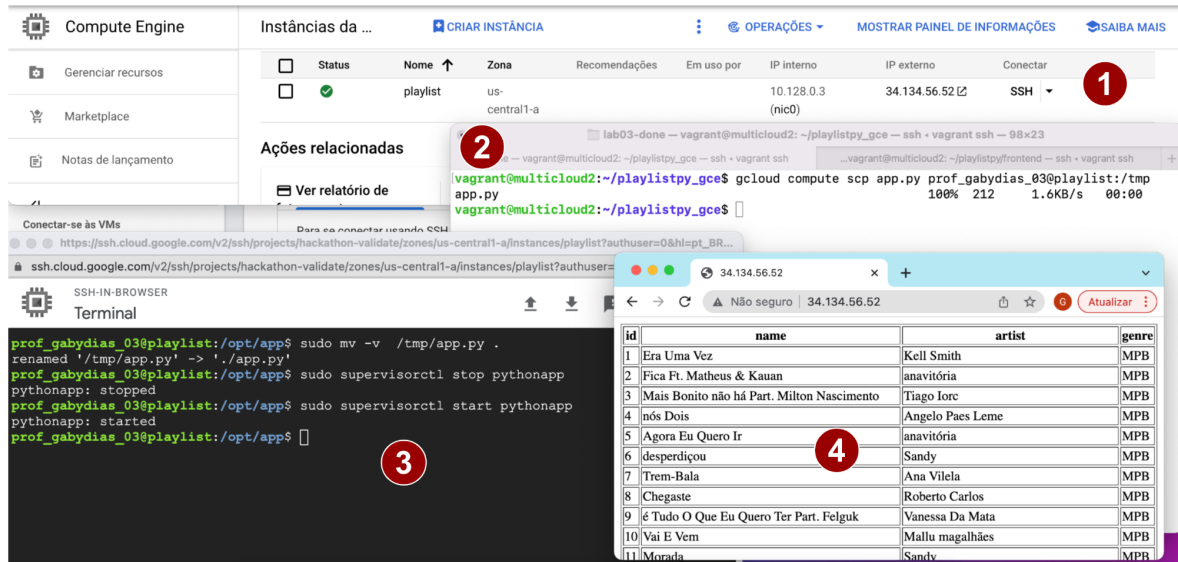
**Escalabilidade** - Quando temos necessidade de escalar o ambiente fazemos isso trocando o tipo da máquina entregando mais memória e CPU.

**Segurança** - Não temos nenhum controle de segurança adicional fora o Firewall nativo do Google que libera apenas HTTP e SSH para esse servidor.

# HACKATHON

## MBA - Arquitetura de Soluções

**Ciclo de Desenvolvimento** - Para fazer deploy de uma nova versão da aplicação conforme figura abaixo realizamos os seguintes passos:



The screenshot illustrates the deployment process in Google Cloud. It shows the 'Compute Engine' interface with a VM named 'playlist' in the 'us-central1-a' zone. The public IP is 34.134.56.52. A terminal window is open, showing the execution of commands to copy the application files and restart the service. A table of songs is also visible in the background.

id	name	artist	genre
1	Era Uma Vez	Kell Smith	MPB
2	Fica Ft. Matheus & Kauan	anavitória	MPB
3	Mais Bonito não há Part. Milton Nascimento	Tiago Iorc	MPB
4	nós Dois	Angelo Paes Leme	MPB
5	Agora Eu Quero Ir	anavitória	MPB
6	desperdiçou	Sandy	MPB
7	Trem-Bala	Ana Vilela	MPB
8	Chegaste	Roberto Carlos	MPB
9	é Tudo O Que Eu Quero Ter Part. Felguk	Vanessa Da Mata	MPB
10	Vai E Vem	Mallu magalhães	MPB
11	Morada	Sandy	MPB

1. Abrir o painel do Google Cloud em Compute Engine para abrir um terminal SSH do servidor Playlist (Janela do Item 03) e pegar o IP público que está cadastrado na VM.

2. Abrir o terminal na máquina de desenvolvimento para realizar uma cópia remota da nova versão da aplicação através do comando "gcp compute scp".

3. No terminal do Servidor Playlist sobrescrever os arquivos enviados dentro da pasta `/opt/app` que é onde fica a aplicação. Após realizar essa etapa, fazer o restart da aplicação através do comando:

```
$ sudo supervisorctl restart pythonapp
pythonapp: stopped
pythonapp: started
```

#### 4. Testar a aplicação no Browser.

**Dica:**

Ao ser questionado sobre controle de versão do código, monitoramento e controles de segurança, a SpotMusic respondeu que está no planejamento evoluir nesses pilares e esse foi um dos motivos da contratação da consultoria, pois, de fato, até hoje eles ficaram bem focados em deixar a aplicação funcionando.

## Fase 2- Implementando a PoC

O segundo desafio do grupo será fazer a implementação da PoC que já irá trazer vários ganhos para a SpotMusic e servirá de base para criar o Documento da Arquitetura da Solução.

**Abaixo você terá o escopo da PoC que seu grupo precisará implementar:**

**Dica:**

No seu Documento da Arquitetura de Solução **será importante trazer o racional das decisões tomadas na PoC**, como também todos os itens que ainda possam estar faltando para garantir uma arquitetura de referência de qualidade.

**A. Não existe controle de versionamento de código.** A aplicação não possui um repositório de código para fazer o controle de versão. No escopo da PoC está previsto migrar os códigos da aplicação para um repositório do GitHub.

**B. Não existe segregação de camadas.** A aplicação (frontend e backend) e a camada de dados (banco de dados mysql) rodam no mesmo servidor.

- a) Está previsto um desacoplamento de funções na camada da aplicação. A expectativa é que a aplicação rode em **dois containers** onde o frontend foi reescrito em React e o backend continuou sendo em Python/Flask.
- b) A camada de banco de dados hoje roda em um MySQL na VM do Servidor Playlist e precisa ser migrado para um **serviço gerenciado** do Google Cloud chamado **Cloud SQL**.

# HACKATHON

## MBA - Arquitetura de Soluções

**C. Não existe IaC (Infra as Code) no ambiente.** Atualmente é usado apenas o serviço Google Compute Engine (VM) e o mesmo foi criado de forma manual usando a console web. Para a PoC serão usados basicamente 3 serviços onde a expectativa é subir via automação usando Terraform.

**Cloud Run:** Desenvolva e implante aplicativos em contêineres altamente escalonáveis em uma plataforma totalmente gerenciada e sem servidor. A aplicação terá 2 contêineres: um será o backend em python que precisará se conectar com o banco de dados e o outro será o frontend em react que fará chamadas para o backend.

### Importante:

O Cloud Run é bem parecido com o Google AppEngine que foi usado durante a Fase 5, porém, ele é mais indicado quando você já tem a aplicação rodando em containers docker. Por isso, iremos usá-lo na PoC.

No caso do Cloud Run, ele não precisa ser provisionado pelo Terraform, basta fazer deploy via pipeline da aplicação.

**Artifactory Registry:** A nova geração do Container Registry. Armazene, gerencie e proteja os artefatos do seu build. Teremos **dois repositórios** responsáveis por armazenar as imagens docker do backend e do frontend.

**Cloud SQL:** Serviço de banco de dados relacional totalmente gerenciado para MySQL, PostgreSQL e SQL Server. Será migrado o banco de dados gerenciado pela SpotMusic dentro da VM para esse serviço gerenciado.

**D. Não existe fluxo de deploy automatizado.** Nós sabemos a importância de automatizar o fluxo de desenvolvimento através de pipelines Devops, portanto, no escopo da PoC, serão entregues 3 pipelines:

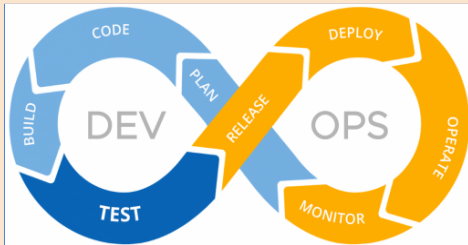
- ☐ Pipeline em GitHub Actions do Backend que fará deploy no Cloud Run.
- ☐ Pipeline em GitHub Actions do Frontend que fará deploy no Cloud Run.
- ☐ Pipeline em Terraform Cloud que fará o provisionamento da Infraestrutura de serviços que serão usados na PoC.



# HACKATHON

## MBA - Arquitetura de Soluções

### Importante:



Na fase 5 nós tivemos a oportunidade de criar uma pipeline DevOps. Para o HackaThon seu grupo poderá usar os conhecimentos obtidos nessa fase para estruturar a pipeline. Decisões de quais integrações e etapas são responsabilidade do grupo. Devido a limitação de tempo do Hackathon se preocupe num primeiro momento em garantir o deploy da aplicação funcionando corretamente. Se sobrar tempo, o grupo poderá trazer outros steps na pipeline que darão uma pontuação adicional.

### Dica:

Ao término da PoC é **importante construir o desenho TO BE de como ficou a estrutura**. Isso faz parte de um entregável do Documento da Arquitetura de Soluções.

## A. Controle de Versão

No Kit do Hackathon foi disponibilizado para seu grupo um arquivo zipado com o código da aplicação já desacoplado entre frontend e backend.

Sua missão será criar um repositório para cada micro serviço. Esse repositório precisará ter ao menos 2 integrantes do grupo, pois a **branch main deverá ser bloqueada** para receber push e todas as versões do código precisarão ser feitas através de branch, solicitando um PR que se aprovado poderá ser feito merge para master.

### Obrigatório:

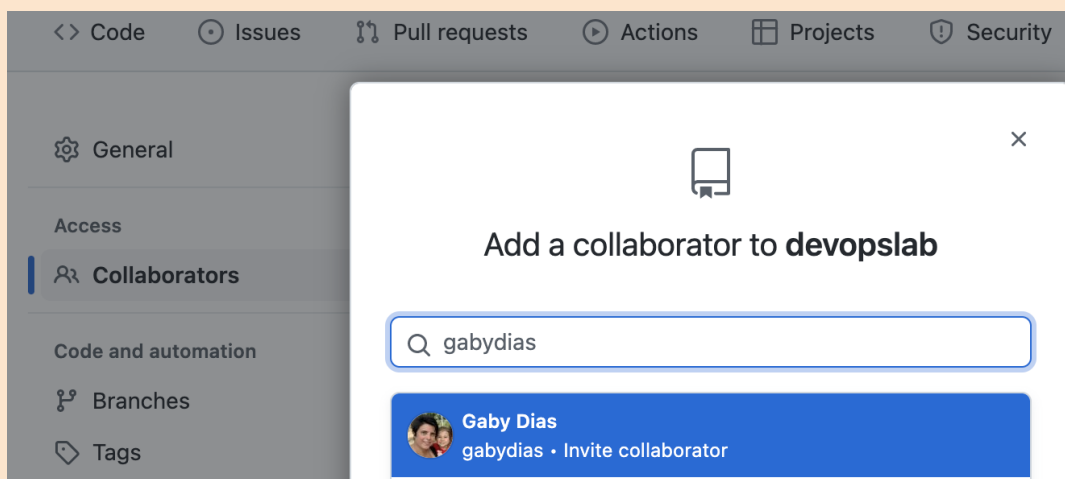
A pessoa que fez o commit e push **NÃO** pode ser a mesma pessoa que fará a aprovação do PR! Isso é uma boa prática de desenvolvimento, garantindo code review e 2 pessoas envolvidas na tomada de decisão de uma mudança em produção.

# HACKATHON

## MBA - Arquitetura de Soluções

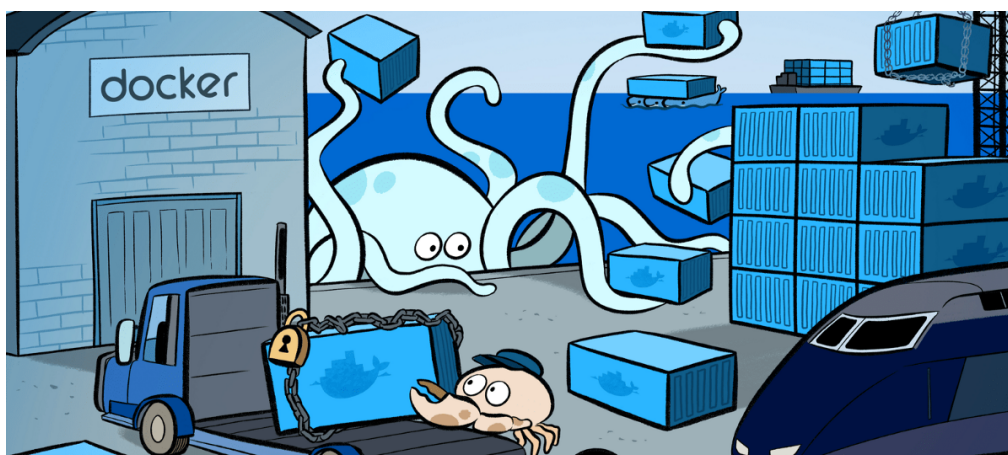
### Importante:

Para que seu grupo possa ser pontuado pelas entregas é **OBRIGATÓRIO** colocar o usuário **gabydias** e o usuário do seu mentor caso ele forneça em todos os repos criados para o hackathon.



## B. Desacoplamento da Aplicação

Agora que já temos os repositórios do back e do front, o próximo passo é **colocar esses micro serviços em containers** e para isso o grupo precisará construir a imagem Docker de cada um deles.



Para construir a imagem Docker vocês já devem ter percebido que os códigos não vieram com o Dockerfile que precisará ser construído pelo seu grupo.

### Dica:

Na fase 5 nós "dockerizamos" uma aplicação Python pra docker, será que a resposta já está pronta? ;)

Como será que faz para "dockerizar" uma aplicação React? Felizmente o Google Cloud nos mostra:

<https://cloud.google.com/community/tutorials/deploy-react-nginx-cloud-run>

Nós sabemos que existem vários caminhos para se chegar a um determinado resultado e o seu grupo poderá decidir como fará a construção do Dockerfile. Um ponto super importante que o grupo precisa levar em consideração é que o **frontend** faz uma chamada no backend e isso está basicamente declarado no arquivo App.js.

```
function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    fetch("$BACKEND_URL")
      .then((res) => res.json())
      .then((result) => setData(result))
      .catch((err) => console.log("error"));
  }, []);
```

Considerando que `$BACKEND_URL` é uma variável de ambiente, sua pipeline precisará lidar com essa substituição ou até mesmo mudar a forma como a aplicação está desenvolvida hoje que pode ser um caminho mais longo e não recomendado.

Dentro do Kit temos um arquivo docker-compose.yml que está funcional se as imagens forem "buildadas" corretamente conforme você pode ver na imagem:

# HACKATHON

## MBA - Arquitetura de Soluções

```

vagrant@workspace6aso:/vagrant/hacka$ ls
Playlist.sql  devopslab-infra  frontend
devopslab     docker-compose.yml  mysql-apt-config_0.8.22-1_all.deb
vagrant@workspace6aso:/vagrant/hacka$ docker-compose up -d
Creating hacka_frontend_1 ... done
Creating hacka_db_1      ... done
Creating hacka_backend_1 ... done
vagrant@workspace6aso:/vagrant/hacka$ docker-compose ps

```

Name	Command	State	Ports
hacka_backend_1	/bin/sh -c exec unicorn - ...	Up	0.0.0.0:5000->8080/tcp, :5000->8080/tcp
hacka_db_1	docker-entrypoint.sh --def ...	Up	0.0.0.0:3306->3306/tcp, :3306->3306/tcp, 33060/tcp
hacka_frontend_1	/docker-entrypoint.sh /bin ...	Up	80/tcp, 0.0.0.0:3000->8080/tcp, :3000->8080/tcp

```

vagrant@workspace6aso:/vagrant/hacka$
vagrant@workspace6aso:/vagrant/hacka$ curl 127.0.0.1:5000
[{"artist":"Kell Smith","genre":"MPB","song_id":1,"title":"Era Uma Vez"}, {"artist":"anavit\u00f3ria","genre":"MPB","song_id":2,"title":"Fica Ft. Matheus & Kauan"}, {"artist":"Tiago Iorc","genre":"MPB","song_id":3,"title":"Mais Bonito n\u00e3o h\u00e1 Part. Milton Nascimento"}, {"artist":"Angelo Paes Leme","genre"

```

Ele pode servir para testar localmente antes de se aventurar no Google Cloud, ou pode confundir o grupo. É importante seu grupo decidir se faz sentido uma engenharia reversa ou irão seguir outra estratégia.

### Dica:

Antes de partir para a automação do provisionamento do Banco de Dados talvez faça sentido seu grupo subir o banco manual para garantir que a aplicação está rodando e na sequência refazer o banco de forma automatizada que é o requisito do próximo item.

Algumas literaturas podem fazer vocês chegarem mais rápido à solução:

[Connect from Cloud Run | Cloud SQL for MySQL](#)

<https://cloud.google.com/sql/docs/mysql/connect-instance-cloud-shell?hl=pt-br>

### C. Automação da Infraestrutura

Chegou a hora de criar a infraestrutura do Google de forma automatizada com Terraform. Serão basicamente dois serviços criados via terraform:

- **Cloud SQL** - Precisa garantir a criação da Instância do Banco de Dados, a Base de Dados e o Usuário que será usado pela aplicação.

[Google Provider Configuration Reference | Guides | hashicorp/google | Terraform Registry](#)

- **Artifact Registry** - Precisa garantir a criação do repositório que irá armazenar as imagens Docker.

[google\\_artifact\\_registry\\_repository | Resources | hashicorp/google | Terraform Registry](#)

Temos dentro do Kit do Hackathon um arquivo SQL para alimentar os dados na base de dados. Você pode fazer isso de forma manual ou automatizada. Para o hackathon estamos considerando fazer a inserção de dados na base de forma manual usando a sdk (gcloud) para ganhar tempo. Se seu grupo decidir fazer isso de forma automatizada garantirão pontos extras. :)

Não esqueça de começar as atividades aqui garantindo uma conta de serviço para ser usada nas automações e pipelines.

#### Importante:

A criação do Cloud Run consiste em criar um serviço já passando a imagem do container que você deseja rodar, portanto, **não é necessário** fazer nenhum provisionamento via terraform e sim pela pipeline de cada aplicação. Não podemos esquecer de ativar a API do serviço.

### D. Automação do Fluxo de Desenvolvimento

Para fechar a fase 2 precisamos criar as pipelines que serão responsáveis por fazer o deploy das aplicações (front e back) no Cloud Run.

Devido a limitação de tempo do Hackathon se preocupe em garantir o deploy da aplicação (Steps Build e Deploy) funcionando corretamente. Se sobrar tempo, o grupo poderá trazer outros steps na pipeline que darão uma pontuação adicional.

### Fase 3 - Documento da Arquitetura de Solução

Chegou a hora de criar o Documento da Arquitetura da Solução. Esse documento, conforme falado no início, será utilizado pela SpotMusic em todo o processo de modernização da solução para balizar e fornecer um entendimento completo no projeto.

#### Importante:

A escrita do documento é Free Style, ou seja, seu grupo que irá definir o nível de detalhe do documento até porque temos um tempo restrito para a conclusão do Hackathon. Se atente às dicas a seguir e não deixe de colocar as informações obrigatórias.

#### Dica:

*"If I Had More Time, I Would Have Written a Shorter Letter" - Blaise Pascal.*

Sejam objetivos, não existe uma expectativa de um documento muito extenso, a proposta da construção desse documento é garantir que o grupo teve:

**Primeiro** - O entendimento completo da solução proposta na PoC e quais serão os ganhos para a SpotMusic, uma vez que implementar a solução em produção. **Não deixem de trazer o racional dos ganhos.**

**Segundo** - Toda PoC tem suas limitações, sabemos que ainda existem alguns pontos importantes para serem levados em consideração para uma solução completa, **é importante o grupo trazer isso no Documento da Fase 3 - Arquitetura da Solução.**

Sugestões de Informações que são recomendadas na construção do Documento:

- ☐ **Obrigatório:** A descrição da solução - Seja sucinto e objetivo, mas escreva de tal forma que as pessoas consigam entender sobre o que o documento está tratando.
- ☐ **Obrigatório:** Desenho macro estrutural da solução - Desenhe um diagrama completo que contemple toda a solução proposta, incluindo uma breve descrição de cada componente.

# HACKATHON

## MBA - Arquitetura de Soluções

São esperados 2 desenhos da solução. O desenho do **TO BE da PoC** e o desenho do **TO BE da Solução** que deverá apresentar elementos de evolução do que foi feito em PoC.

- ☐ Componentes que fazem parte da solução - Faça o diagrama de todas as ferramentas a serem utilizadas na solução.
- ☐ Se a solução já existe e trata-se de um upgrade, descreva as novas tecnologias que serão inseridas.
- ☐ Traga elementos de Segurança da informação - A segurança da informação é parte fundamental na vida de um arquiteto de solução e precisa estar bem descrita para que os demais times envolvidos se sintam seguros ao implementar o que está sendo proposto.
- ☐ Traga elementos que garantam um SLA de uptime e downtime - Descreva os SLAs acordados com base no que foi proposto de solução.
- ☐ Abordagem de testes a ser utilizada - Descreva os tipos de teste a serem executados para garantia da qualidade da solução, os testes unitários, a automação de testes, etc.
- ☐ Traga elementos que mostrem o custo da Solução após ser implementada - Estamos considerando custos estimados.
- ☐ **Opcional:** Outras soluções - Sempre há outras soluções que podem ser utilizadas para atender uma necessidade, algumas precisam de muitos ajustes ou adequações e outras se encaixam na necessidade do negócio de forma mais prática ou com poucas modificações. O arquiteto de solução precisa descrever estas possíveis soluções para auxiliar na justificativa de sua escolha.