

Fundamentos de DevOps

<http://linkedin.com/in/guijac>

Aula 14 – Monitoramento e Observabilidade em DevOps

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 linkedin.com/in/guijac

Roteiro

- Monitoramento:
 - Definição;
 - Contexto Histórico.
- Observabilidade:
 - Definição;
 - Pilares.
- Monitoramento x Observabilidade;
- Ferramentas:
 - Prometheus;
 - Grafana;
 - AWS Cloud Watch.
- Monitoramento e Observabilidade com Spring Boot;
- Referências Bibliográficas.

Monitoramento: Definição

“ Processo de monitorar o **desempenho**, a **disponibilidade** e a **experiência** do usuário final de uma **aplicação** para garantir que a mesma esteja funcionando corretamente.

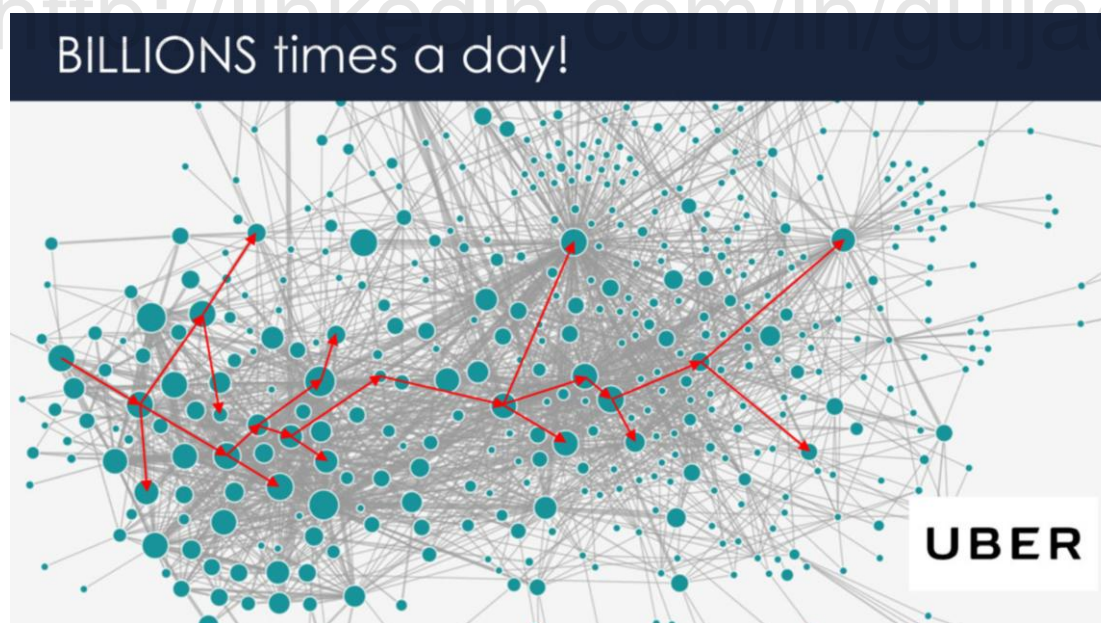
O monitoramento de aplicações inclui uma série de **painéis** para várias métricas relacionadas à **análise de rede**, utilização de **hardware** e **desempenho** de software.

As ferramentas de monitoramento também podem **coletar informações** detalhadas, como sistema operacional, dispositivo, tamanho da tela, localização, tempo gasto no local e produtos adquiridos. ”

VMAWARE (2025)

Monitoramento: Contexto Histórico

- Monitoramento tradicional focava em disponibilidade e métricas básicas, como CPU e memória;
- Em aplicações **monolíticas**, era simples **rastrear** o caminho de uma **requisição**;
- Com **microsserviços** e **contêineres** há um **aumento da complexidade**, tornando o monitoramento isolado insuficiente.



Fonte: [Would You Like Some Tracing With Your Monitoring? - Yuri Shkuro, Uber Technologies](#)

Observabilidade: Definição

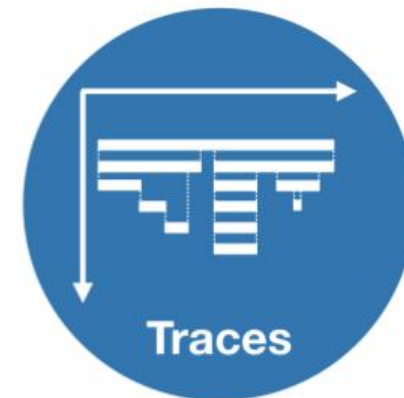
“ *Prática que permite que as equipes de DevOps, incluindo desenvolvedores e operadores de software, identifiquem problemas onde e **quando** eles **acontecem** em **sistemas distribuídos**.*

*Permite **agregar métricas, rastreios (traces), logs e eventos** de um sistema distribuído e correlacioná-los em vários componentes e serviços de aplicações.* ”

VMAWARE (2025)

Observabilidade: Pilares

- **Logs:** registros detalhados de **eventos** que permitem rastrear o histórico de execução;
- **Métricas:** **valores numéricos** em **séries temporais** que mostram desempenho e saúde do sistema;
- **Traces:** rastreiam o **caminho** de **requisições** entre serviços, revelando gargalos e dependências.



Fonte: [What's Next for Observability | Grafana Labs](#)

Monitoramento x Observabilidade

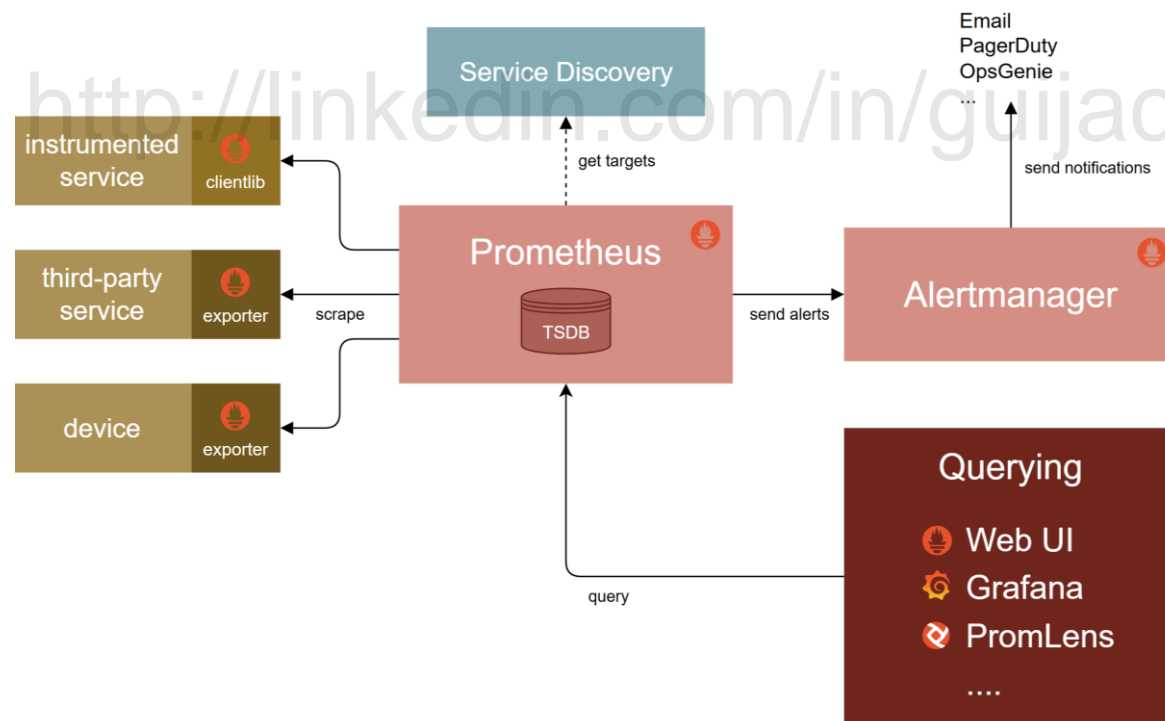
	Monitoramento	Observabilidade
O que é?	Medir métricas predefinidas para acompanhar a integridade do sistema.	Coletar métricas, logs e traces para entender o comportamento interno de sistemas distribuídos.
Foco	Coletar dados para identificar efeitos anômalos no sistema.	Investigar a causa raiz dos efeitos anômalos do sistema.
Sistemas envolvidos	Normalmente se preocupa com sistemas autônomos .	Normalmente se preocupa com sistemas distribuídos .
Rastreabilidade	Limitado às bordas do sistema.	Disponível onde os sinais são emitidos em diferentes arquiteturas de sistema.

Fonte: Adaptado de [Observabilidade x monitoramento — Diferença entre processos baseados em dados — AWS](#)

- Monitoramento mostra o **que** está acontecendo por meio de métricas predefinidas.
- Observabilidade revela o **porquê**, usando dados diversos para investigar falhas.

Ferramentas: Prometheus

- Ferramenta open-source para coleta de **métricas** em tempo real;
- Armazena dados em **séries temporais**, permitindo consultas flexíveis em PromQL;
- Muito utilizado para configurar **alertas** automáticos em sistemas distribuídos.



Fonte: [Introduction to Prometheus | Prometheus Trainings by PromLabs](#)

Ferramentas: Grafana

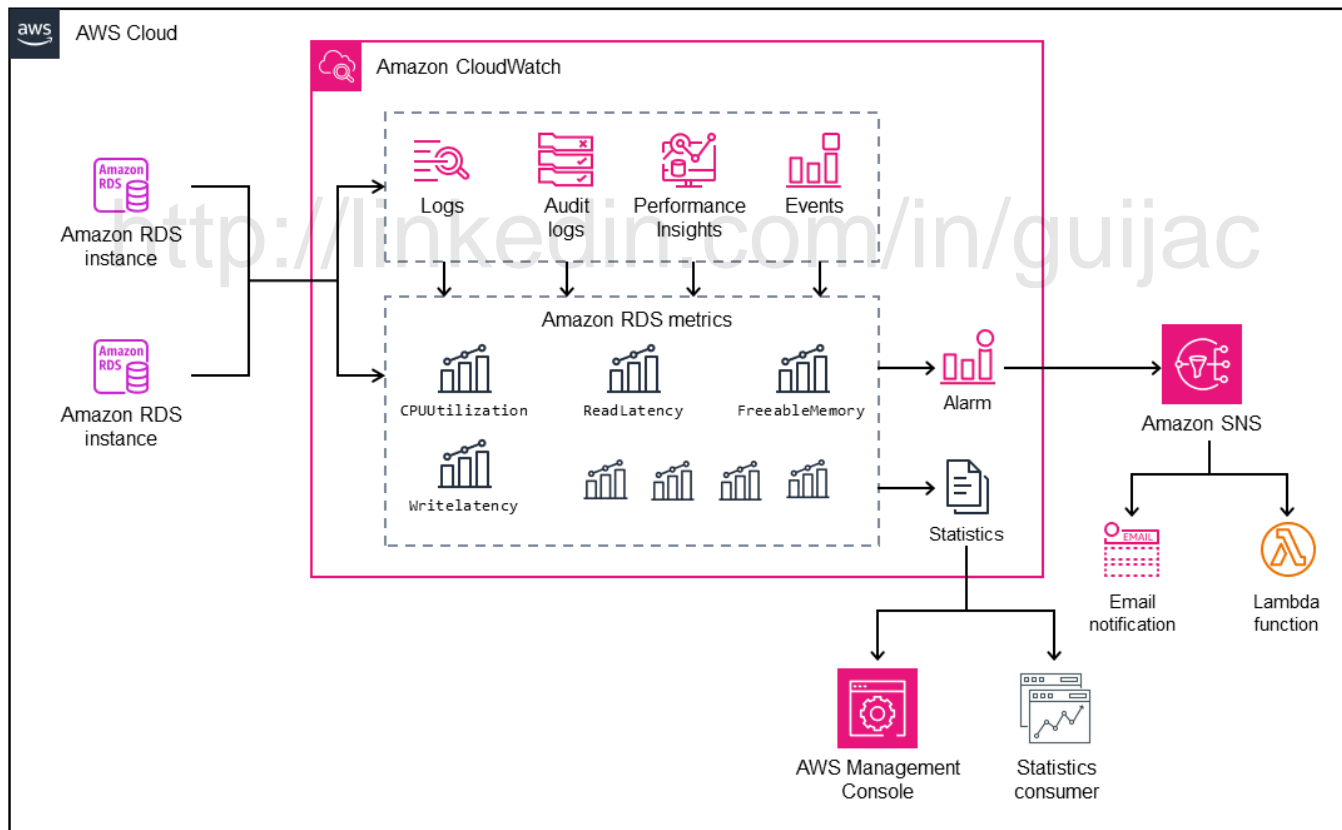
- Plataforma de **visualização de dados** que cria *dashboards* interativos;
- Integra-se a diversas fontes, incluindo Prometheus, para **exibir métricas**.



Fonte: [Set up and observe a Spring Boot application with Grafana Cloud, Prometheus, and OpenTelemetry | Grafana Labs](#)

Ferramentas: CloudWatch

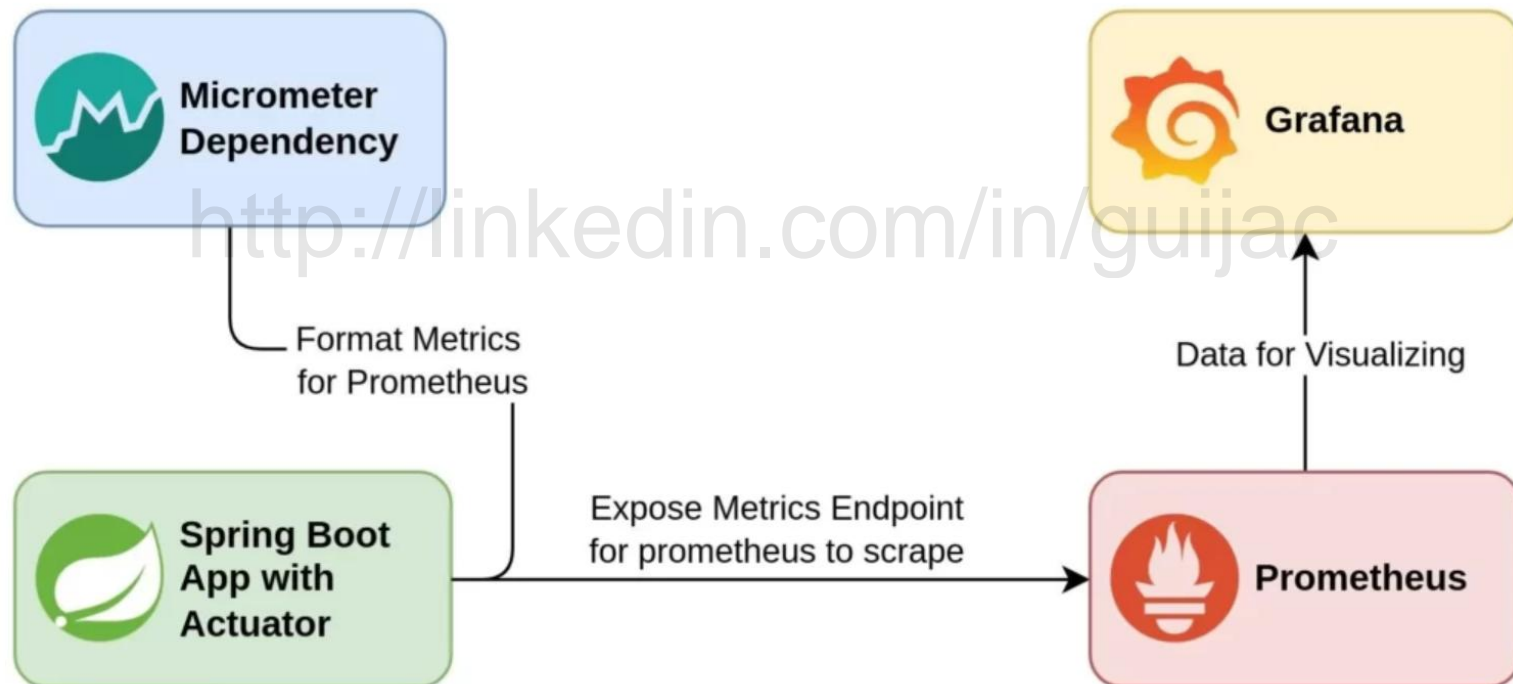
- Serviço **nativo** da **AWS** para monitoramento de recursos e aplicações;
- Coleta métricas, armazena logs e permite configurar alarmes;
- Possui uma **gestão centralizada**, no AWS Console, permitindo uso de *dashboards*.



Fonte: [CloudWatch alarmes e painéis - AWS Orientação prescritiva](#)

Monitoramento e Observabilidade com Spring Boot

- **Spring Boot Actuator + Micrometer:**
 - Fornece *endpoints* prontos para monitorar e gerenciar a aplicação;



Fonte: [Run Prometheus and Grafana with Spring boot Actuator](#)

Monitoramento e Observabilidade com Spring Boot

- **Spring Boot Actuator + Micrometer:**
 - São instalados através de dependência em seu pom.xml e configurados através do arquivo application.properties.

```
<!-- Actuator para expor métricas e endpoints -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>

<!-- Micrometer registry para Prometheus -->
<dependency>
  <groupId>io.micrometer</groupId>
  <artifactId>micrometer-registry-prometheus</artifactId>
</dependency>
```

Arquivo pom.xml declarando as dependências Actuator e Micrometer.

```
# Actuator / Micrometer - expor endpoints básicos
management.endpoints.web.exposure.include=health,info,metrics,prometheus

# Health endpoint - mostrar detalhes, apenas para fins didáticos
management.endpoint.health.show-details=always
```

Arquivo application.properties expondo rotas para leitura de métricas.

Monitoramento e Observabilidade com Spring Boot

▪ Spring Boot + Prometheus + Grafana:

```
services:
  spring-app:
    build:
      context: app/.
      dockerfile: Dockerfile
    image: simple-rest-api:1.0
    container_name: spring-app
    ports:
      - "8080:8080"

  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    ports:
      - "9090:9090"
    depends_on:
      - spring-app

  grafana:
    image: grafana/grafana:latest
    container_name: grafana
    ports:
      - "3000:3000"
    environment:
      - GF_SECURITY_ADMIN_USER=admin
      - GF_SECURITY_ADMIN_PASSWORD=admin
    depends_on:
      - prometheus
```

Arquivo docker-compose.yml para uma orquestração de contêineres com uma aplicação Spring

Referências Bibliográficas

AWS. **Amazon CloudWatch Documentation**. Disponível em: <https://docs.aws.amazon.com/cloudwatch>. Acesso em: 22 set 2025.

GRAFANA LABS. **Grafana Documentation**. Disponível em: <https://grafana.com>. Acesso em: 22 set 2025.

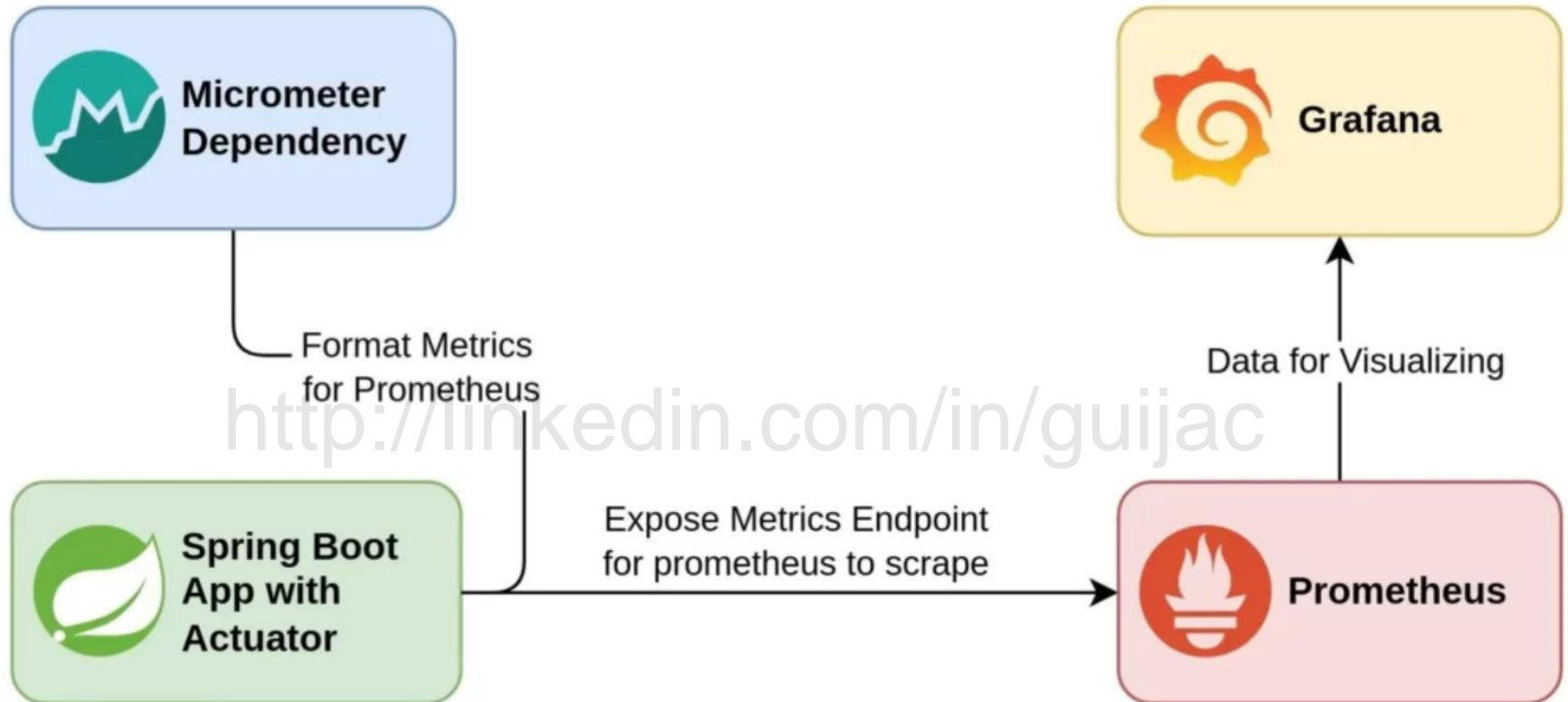
GOOGLE. **Site Reliability Engineering**. 1. ed. Sebastopol: O'Reilly Media, 2016.

HUMBLE, Jez; KIM, Gene; WILLIS, John; DEBOIS, Patrick. **Manual do DevOps**. 1. ed. Portland: IT Revolution Press, 2016.

PROMETHEUS. **Prometheus Documentation**. Disponível em: <https://prometheus.io>. Acesso em: 22 set 2025.

VMWARE. **Application Monitoring**. Disponível em <https://www.vmware.com/topics/application-monitoring>. Acesso em 22 set 2025.

Por hoje (de teoria!) é só!



Fonte: [Run Prometheus and Grafana with Spring boot Actuator](#)

Prof. Esp. Guilherme Jorge Aragão da Cruz

✉ guilherme.jacruz@sp.senac.br

in linkedin.com/in/guijac

Licença

- Este conteúdo está licenciado sob a Licença Creative Commons Atribuição-NãoComercial-Compartilha Igual 4.0 Internacional (CC BY-NC-SA 4.0).
- Todos os direitos autorais sobre este conteúdo pertencem ao autor, e este material não pode ser usado comercialmente sem autorização expressa.
- Material elaborado no contexto da disciplina **Fundamentos de DevOps do Centro Universitário Senac**, para fins educacionais.
- As referências a marcas, produtos e tecnologias têm caráter exclusivamente educacional, não havendo vínculo institucional ou comercial com as organizações citadas.
- Para ver o texto completo da licença, acesse <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>.

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 linkedin.com/in/guijac