FullCycle 3.0

Evolua no mundo do Desenvolvimento.

Esteja entre os melhores.

Aprenda a desenvolver aplicações de grande porte e tenha um dos perfis mais desejados e bem pagos do mercado

Sobre o Curso Full Cycle

O Curso Full Cycle é uma formação completa para fazer com que pessoas desenvolvedoras sejam capazes de trabalhar em projetos expressivos sendo capazes de desenvolver aplicações de grande porte utilizando de boas práticas de desenvolvimento.

A pessoa desenvolvedora terá contato com as tecnologias mais modernas do mercardo que são fortemente utilizadas em grandes corporações, além de aprender de forma profunda os conceitos e técnicas mais atuais de arquitetura de software.

O curso Full Cycle vai muito além de uma pós-graduação ou especialização, porém, por apenas uma fração do preço.

De Fullstack para Full Cycle

Atualmente, muitos desenvolvedores estão completamente empenhados para serem considerados Fullstack, contudo, grande parte deles não possuem os conhecimentos necessários para participar do início ao fim (dos commits iniciais ao go live) de um projeto que realmente traga grandes impactos na vida das pessoas.

O foco desse treinamento é apresentar o novo mundo que todos os desenvolvedores terão de fazer parte. Aonde eles deixam de ser puramente criadores de código e tornam-se parte essencial do ciclo completo de entrega da solução.

Este treinamento terá foco em 3 pilares

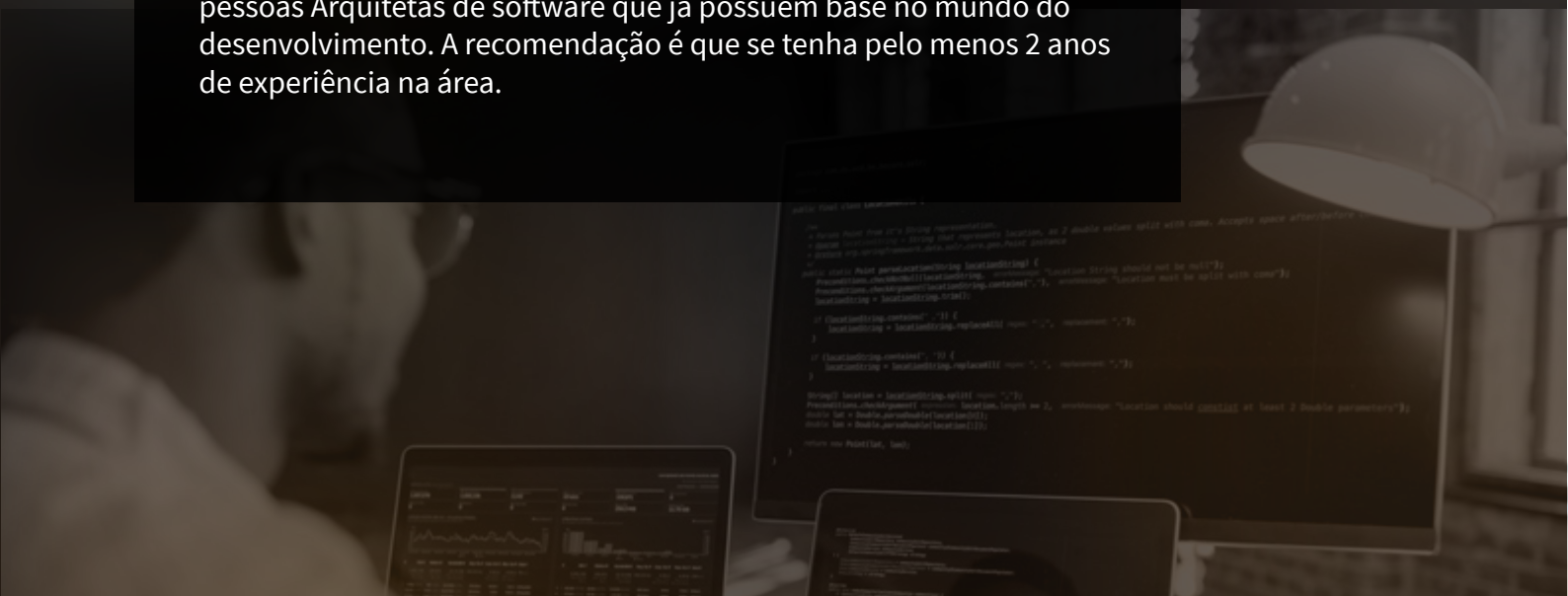
- Arquitetura
- Desenvolvimento
- DevOps

Esse treinamento é para você que quer

- Ter mais confiança como pessoa desenvolvedora sabendo que está programando do “jeito certo”
- Evoluir exponencialmente em sua carreira
- Ser capaz de trabalhar em projetos expressivos e de grande porte
- Estar um passo à frente do que há de mais novo no mundo do desenvolvimento
- Participar de reuniões de grandes projetos sem receio de colaborar de forma ativa
- Estar entre os melhores

Para quem o curso é destinado?

Para pessoas desenvolvedoras Juniors, Plenas, Seniors e também para pessoas Arquitetas de software que já possuem base no mundo do desenvolvimento. A recomendação é que se tenha pelo menos 2 anos de experiência na área.



Projeto Prático

Para tornar o aprendizado prático, o treinamento terá como objetivo desenvolver uma aplicação similar a da Netflix, onde o usuário final terá a possibilidade de se registrar, contratar os serviços de streaming de vídeos, navegar pelo catálogo, bem como reproduzir os vídeos.




Escolha a tecnologia que você irá desenvolver

Além da linguagem Go, que é utilizada para ensinar diversos conceitos, você poderá escolher entre as linguagens abaixo para desenvolver o projeto prático:

.Net • Java • PHP • Python • TypeScript • *GO







* Os exemplos conceituais do curso são realizados em Go

* O aluno poderá escolher a linguagem que desenvolverá o projeto prático dentre as citadas acima, com exceção de Go

* O microsserviço de processamento de vídeos do projeto prático será desenvolvido em Go independente da linguagem escolhida

Arquitetura de software

Fundamentos de arquitetura de software

- Tipos de Arquitetura
- Papel do Arquiteto de Software
- Por que aprender arquitetura de software
- Arquitetura vs Design
- Pilares da arquitetura de software
- Requisitos arquiteturais
- Características arquiteturais
- Estilos arquiteturais
- Performance
- Escalabilidade

Comunicação entre sistemas

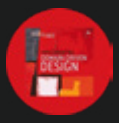
- Comunicação síncrona vs assíncrona
- REST
- gRPC
- GraphQL
- Filas com RabbitMQ
- Apache Kafka

Domain Driven Design

- Entendendo DDD
- Linguagem Ubíqua
- Domínio e subdomínios
- Contextos delimitados
- Mapas de contextos
- Design patterns

SOLID

- Fundamentos do SOLID
- Single Responsibility
- Open/Closed
- Liskov substitution
- Interface segregation
- Dependency inversion
- SOLID na prática



DDD: Modelagem Tática e Patterns

- Modelagem estratégica vs tática
- Entidades vs Agregados
- Value Objects
- Domain Services
- Application Services
- Domain Events
- Factories
- Respositories

Arquitetura Hexagonal

- Fundamentos
- Motivações
- Evoluções
- Principais camadas
- Direcionamento único
- Dependency Inversion

Event Storming na Prática

- O que é Event Storming
- Vantagens de se utilizar ES
- Entendimento de atores
- Levantamento de eventos
- Transformando eventos em ações
- Cronologia
- Definição dos fluxos e Agregados
- Descobrindo e definindo Contextos

Clean Architecture

- O que é Clean architecture
- Histórico dos tipos arquiteturais
- DDD vs Clean Architecture
- Entidades vs Enterprise Business
- Roles
- Desenvolvimento orientado a Use
- Cases
- DTOs como Input e Output
- Presenters vs DTO
- Adaptadores e camadas externas
- Gateways

Sistemas monolíticos

- O que são sistemas monolíticos
- Monolíticos estão ultrapassados?
- Porque sistemas monolíticos falham
- Como criar um sistema monolítico escalável
- 12 Fatores
- Como criar um sistema monolítico modular
- Sistemas monolíticos vs DDD

EDA - Event Driven Architecture

- O que é EDA?
- Vantagens e desvantagens
- Princípios do EDA
- Events e Event Generator
- Brokers
- Mediator
- Mediator vs Broker
- Classificação de eventos
- Event sourcing

Arquitetura baseada em microsserviços

- O que são microsserviços
- Sistemas monolíticos vs Microsserviços
- Vantagens e desvantagens
- 9 Características
- Comunicação síncrona vs assíncrona
- Resiliência com microsserviços
- Coeorografia vs Orquestração
- Saga Pattern
- Principais patterns
- Trace distribuído

API Gateway

- O que é API Gateway
- Principais conceitos
- Principais soluções
- Projeto Kong e Konga
- Roteamento
- Plugins
- Autenticação
- Rate Limit
- Stateful vs Stateless
- Kong no Kubernetes

RabbitMQ

- Fundamentos de Mensageria
- Conceitos básicos do RabbitMQ
- Exchanges
- Filas
- Dinâmica de funcionamento
- Garantias de entrega e recebimento

Apache Kafka

- Principais conceitos
- Conceitos básicos a prática
- Desenvolvendo aplicação
- Kafka Connect na prática
- Serviços gerenciados

Autenticação e Keycloak

- Introdução ao OAuth 2 e OpenID Connect
- Iniciando com Keycloak
- Geração de Access Token e IDToken
- Atributos de usuário
- Roles e Grupos
- Gerenciamento de temas

Service Discovery com Consul

- Entendendo Service Discovery
- Visão geral do Consul
- Service Registry
- Health check
- Tipos de agentes
- Criando Cluster
- Criptografia
- User Interface

Arquitetura do projeto prático - Codeflix

- Decisões arquiteturais
- Microsserviços do Projeto
- Dinâmica dos microsserviços
- Diagrama C4
- Autenticação
- Docker, Kubernetes e Cloud Providers

Desenvolvimento de Microsserviços

Java

- Arquitetura limpa
- Spring Boot e seu ecossistema
- Mensageria com RabbitMQ
- API Rest
- Testes automatizados
- Autenticação via Keycloak
- Integração com GCP Storage
- Integração com Object Storage
- Docker e docker-compose

.Net 6

- Arquitetura limpa
- API REST
- Asp.Net MVC Core
- Ambiente e aplicação Containerizados
- Integração com RabbitMQ
- Autenticação com Keycloack
- Autenticação e validação de Token JWT e Roles
- Integração com GCP Cloud Storage
- Principais libs do ecossistema .Net
- DDD e Clean Architecture

Typescript com Nest.js

- Arquitetura limpa
- Nodejs utilizando o Framework Nest.js
- DDD e Clean Arquitect
- Boas práticas do Nest.js
- Conceitos de Desenvolvimento de Microsserviços
- API REST
- Testes de Unidade, Integração e E2E utilizando JEST
- Mensageria com RabbitMQ
- Integração com GPC Cloud Storage
- Integração Continua
- Boas práticas para desenvolvimento de ambiente de desenvolvimento e produção

Processamento de vídeos com Golang

- Introdução a conceitos de computação
- Memória, Threads e Goroutines
- Trabalhando com testes
- Conversão e fragmentação de vídeos
- Upload Manager
- Integração com RabbitMQ

Python com Django

- Arquitetura limpa
- Área administrativa
- Integração com RabbitMQ
- Autenticação com Keycloak
- Django ORM
- Testes automatizados

PHP

- Arquitetura limpa
- Configuração do ambiente com Docker
- Desenvolvimento de APIs REST
- Testes automatizados
- Integração RabbitMQ
- Autenticação com Keycloak
- Armazenamento na Google Cloud Storage

Frontend com React.js

- TypeScript
- React Hooks
- Material UI
- Context API
- Redux e Redux Saga
- Uploads paralelos de imagens e vídeos

DevOps

Docker do básico ao avançado

- Instalação
- Gerenciamento básico de containers
- Volumes
- Networks
- Docker-compose
- Build de imagens
- Trabalhando com templates
- Otimizando imagens

Práticas avançadas com Github

- Regras importante para os branches
- CODEOWNERS
- Configuração do processo de Code Review
- Geração de Tags e Releases
- Bumb versioning
- Assinatura de commits
- Semantical versioning
- Conventional Commits

Gitflow

- Entendendo Gitflow
- Gitflow e Pull Requests
- Trabalhando com Releases
- Hotfix

Integração contínua

- Introdução
- Trabalhando com Github Actions
- Gerenciamento de Secrets
- Geração automática de versão
- Integração com diferente Dockerhub
- Deploy no Kubernetes

Qualidade de código com SonarQube

- Visão geral
- Instalando SonarQube e sonar-scanner
- Configuração de projetos
- Exclusão e inclusões de pastas
- Trabalhando com SonarCloud
- SonarCloud no processo de CI

Kubernetes

- Introdução ao mundo Kubernetes
- Configurando Kubernetes local com Kind
- Pods, ReplicaSets e Deployments
- Secrets e ConfigMaps
- Gerenciamento de namespaces
- Gerenciamento de recursos computacionais
- Horizontal Pod Autoscaler
- Kubernetes Lens
- Instalação de pacotes com Helm

Observabilidade

- Prometheus e Grafana
- Elastic Stack
- Observabilidade com Kiali

API Gateway com Kong e Kubernetes

- Kong como ingress
- Principais plugins
- Autenticação básica
- Autenticação via introspecção

Service Mesh com Istio

- Conceitos básicos sobre Service Mesh
- Instalando Istio
- Virtual Service e Ingress Gateway
- Regras para Load Balancer
- Timeouts e Retries
- Cirtcuite Breaker
- Geração de certificados SSL

OpenTelemetry

- Conceitos básicos
- Formatos de collector
- Formatos de instrumentação
- Trabalhando com Logs, métricas e tracing
- Padrões abertos vs vendors

IaC com Terraform

- Principais conceitos
- Variáveis, Outputs e Datasources
- Recursos na prática
- Provisionando Cluster Kubernetes
- Criação de módulos
- States remoto

Ansible

- Entendendo o mundo Ansible
- Inventário, módulos e argumentos
- Rodando Ansible com Docker
- Rodando Ansible na AWS
- Playbooks
- Ansible-galaxy

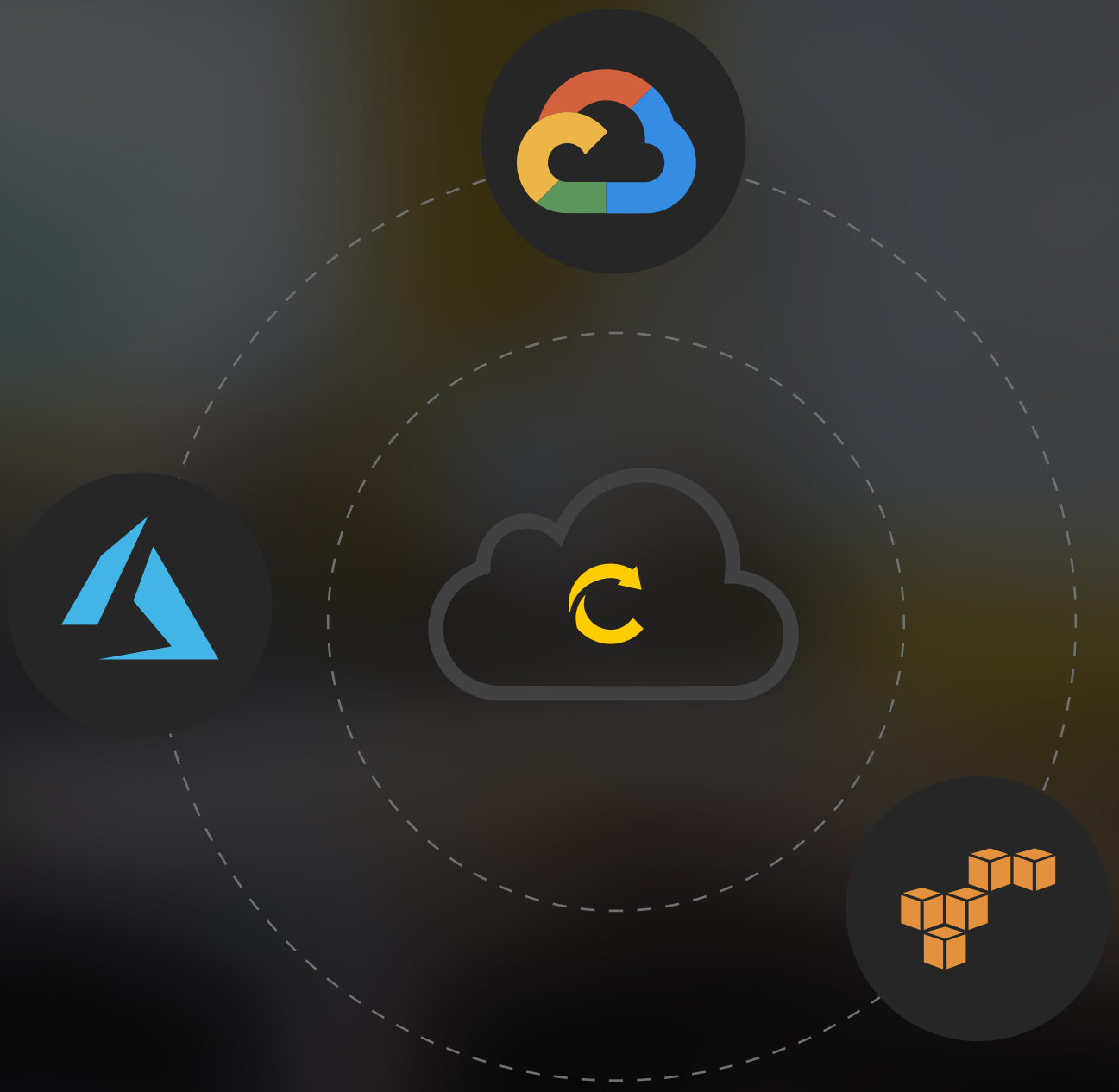
GitOps com Argo CD

- O que é GitOps
- Principais conceitos
- Dinâmica de funcionamento
- Principais ferramentas
- Iniciando com Argo CD
- Introdução ao Kustomize
- Realizando Deploy via Github Actions
- Deploy manual vs automático
- Realizando Rollback

Cloud, provisionamento e processo e deploy

Para que a pessoa desenvolvedora tenha uma experiência realmente prática nos quesitos cloud e deployments, realizaremos todo provisionamento e configuração da infraestrutura de forma automatizada utilizando IaC (infra as code) em conjunto com as ferramentas Terraform e Ansible. O processo de deploy será realizado utilizando GitOps com Argo CD.

Os cloud providers a serem utilizados serão: **AWS, Google Cloud e Azure.**



Suporte 360 graus

A Full Cycle oferece nesse curso um nível extremamente alto de suporte e proximidade entre os alunos e tutores contendo:

Tira dúvidas individual

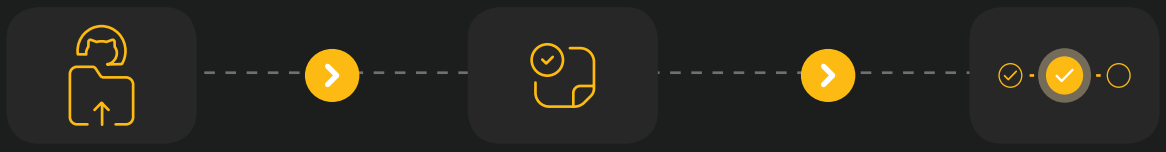
O aluno terá a possibilidade de tirar dúvidas específicas referente ao curso de forma individual com seus tutores.

Fórum

Nesse fórum, alunos e tutores poderão tirar dúvidas e discutir assuntos referentes ao treinamento em questão.

Correção de fases do projeto

Conforme o aluno evolui no desenvolvimento do projeto, será necessário que o mesmo faça o upload de seu código desenvolvido em um repositório GIT para que os tutores façam a análise e sugestões de correção e melhorias



O aluno envia o código para o tutor via GitHub

O tutor corrige e valida a próxima fase do projeto

Depois do feedback do tutor, a próxima fase é liberada

Com quem você irá aprender

A Full Cycle sempre contará com tutores extremamente qualificados e especialistas nas mais diversas áreas.

Wesley Willians

Fundador da School of Net e Full Cycle.

Formado em Tecnologia e Mídias Digitais pela PUC-SP, MBA pelo Ibmec-RJ, realizou alguns cursos na Sloan School of Management no MIT e atualmente é mestrando na área de Design Instrucional e Tecnologias de Educação à Distância na Universidad del Turabo.

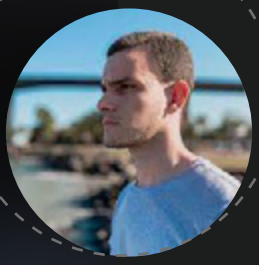
Programador poliglota e atualmente é um grande amante da área de Arquitetura de Software e DevOps.

Luiz Carlos Diniz

Desenvolvedor completamente apaixonado por tecnologia. Especialista em PHP e seus principais frameworks como: ZF, Symfony e Laravel.

Possui grande domínio em tecnologias voltadas para Python, Frontend e Javascript. É tutor na School of Net e Full Cycle.

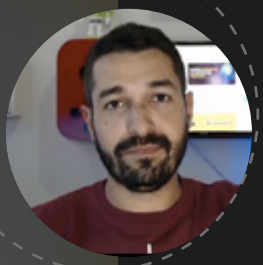
Leonan Luppi



Apaixonado por tecnologia e desenvolvimento, graduado em sistemas de informação e pos graduado em engenharia de software, business intelligence e marketing.

Especialista em Java e Node.js e seus principais frameworks como Spring e Nest.js/Express.js. Tem domínio de diversas tecnologias, é tutor na School of net e Full Cycle. Nas horas vagas curte voar de paraglider e jogar videogame.

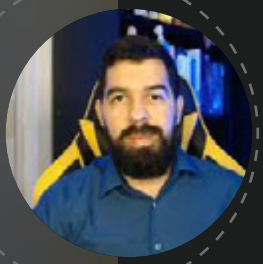
Carlos Ferreira



Desenvolvedor que curte demais o PHP, especialmente usando o seu principal framework, o Laravel.

Também trabalha com as tecnologias: Dart/Flutter, JavaScript e vários de seus frameworks) e tem uma quedinha pelo mundo DevOps.

Wilson Neto



Engenheiro de software fullstack, apaixonado por todo o mundo da tecnologia, especialista em C# e .Net, atuando também com cloud com foco em Azure, trabalhando tamnbém com React.js e Next.js no frontend.

Atualmente atuando como engenheiro de software senior na XP Inc., como tutor de .Net no curso Fullcycle e sempre ativo na comunidade. Se aprofundando cada dia mais em melhores práticas de desenvolvimento, arquitetura e na cultura DevOps.

Sobre a Full Cycle

A Full Cycle foi fundada a partir de um projeto que visa ajudar pessoas desenvolvedoras a terem mais contato com tecnologias, metodologias e práticas de desenvolvimento utilizadas em grandes corporações.

Através de muito conteúdo gratuito e de alta qualidade técnica disponibilizado no canal Full Cycle do Youtube, a Full Cycle começou contribuindo com a comunidade promovendo imersões gratuitas, apresentando na prática tecnologias disruptivas que são utilizadas fortemente no mercado, bem como trazendo estudos de casos de empresas relevantes no mundo do desenvolvimento como: Microsoft, Oracle, Mercado Livre, Magazine Luiza, Itau, Stone, Americanas s.a, PicPay, entre outras.

Atualmente a Full Cycle fornece cursos, mentorias para pessoas desenvolvedoras, bem como uma plataforma para empresas capacitarem seus times de desenvolvimento.

Nossa Missão

A missão da Full Cycle é ajudar pessoas desenvolvedoras e empresas a atingirem seu máximo potencial tecnológico para que sejam capazes de criarem aplicações e sistemas de grande porte que agreguem valor para toda sociedade

fullcycle.com.br