



Especialização em Back End I

Fundamentação

Desde o surgimento de arquiteturas projetadas na nuvem e diante da necessidade de aplicações mais robustas, escaláveis, com alta disponibilidade e baixo desperdício de processamento, surgiram as arquiteturas de software baseadas em microsserviços.

Trata-se, então, de uma abordagem arquitetônica e organizacional para o desenvolvimento de software, na qual ele é composto de pequenos serviços independentes que se comunicam por meio de protocolos bem definidos. Os proprietários destes serviços são pequenas equipes independentes que promovem este modelo arquitetônico, uma solução tecnológica à proposta dada por metodologias ágeis para o processo de desenvolvimento de software.

Por isso, diante dos desafios que as empresas apresentam — pela necessidade de implantar novos negócios e conseqüentemente a necessidade de evoluir rapidamente o software que os suporta em pequenos incrementos — é imprescindível conhecer em detalhes os conceitos e componentes de software que regem este modelo arquitetônico.

Objetivos de aprendizagem

- Adquirir as bases e desenvolver a capacidade de programar aplicações de software a partir da perspectiva de uma arquitetura baseada em microsserviços.
- Compreender e analisar os diferentes desafios enfrentados pelas equipes de trabalho de hoje ao desenvolver softwares em nuvem robustos, escalonáveis e altamente disponíveis.
- Os conceitos serão aplicados na linguagem de programação Java, tomando como referência o framework de microsserviços desenvolvido pela Netflix, Spring Cloud, um dos mais utilizados para desenvolvimentos em empresas de TI atualmente.



Metodologia de ensino-aprendizagem

Na Digital House, propomos um modelo educacional que inclui ambientes de aprendizagem síncronos e assíncronos com uma abordagem que vincula teoria e prática, por meio da aprendizagem ativa e colaborativa.

Nossa proposta inclui aulas ao vivo com seu grupo de alunos e professores, nas quais você pode participar de onde estiver. Além disso, contamos com um campus virtual customizado, onde você encontrará aulas virtuais, com atividades, vídeos, apresentações e recursos interativos, para fazer no seu ritmo antes de cada aula presencial.

Ao longo do seu aprendizado na Digital House você poderá desenvolver hard skills e soft skills, como trabalho em equipe, criatividade, responsabilidade, comprometimento, comunicação eficaz e autonomia.

Na Digital House usamos a metodologia “sala de aula invertida”. O que isso significa? A cada semana vamos pedir a vocês que se preparem para a próxima, lendo textos, assistindo vídeos, fazendo atividades, entre outros recursos. Dessa forma, ao chegar à reunião ao vivo, você poderá abordar o assunto e aproveitar ao máximo essa instância.

Utilizamos atividades e estratégias baseadas em métodos participativos e ativos para colocar você em movimento, pois você só sabe o que está fazendo. Por isso, organizamos as aulas para que possa realmente trabalhá-las e colocar em prática as diferentes ferramentas, linguagens e competências que constituem a formação de um programador. Em outras palavras, pensamos na sala de aula como um espaço de trabalho.

Uma das questões centrais da nossa metodologia de ensino é aprender fazendo. Por isso, ao longo do curso os exercícios estarão muito presentes, ou seja, a prática de atividades de vários tipos e níveis de complexidade que lhe permitirão consolidar a aprendizagem e verificar se a assimilou corretamente. Desta forma, a incorporação dos conteúdos é alcançada de forma mais significativa e profunda, e a assimilação do conhecimento torna-se mais eficaz e duradoura. Relacionar o que foi aprendido com a realidade dos desenvolvedores web, promover autonomia e autoconhecimento, melhorar a análise, relacionamento e compreensão de conceitos auxilia no exercício de múltiplas competências.

A aprendizagem entre pares é um dos elementos centrais da nossa metodologia, por este motivo, a cada aula sugerimos que trabalhe em mesa de trabalho com os seus colegas - ao



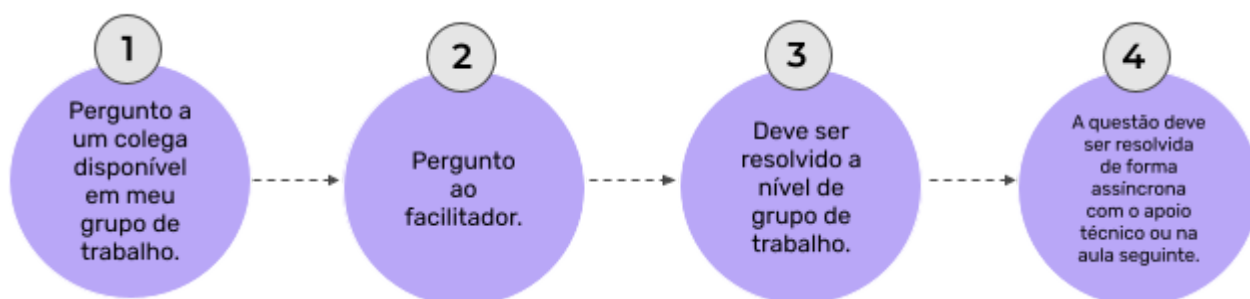
longo do curso iremos trocar a composição dos grupos para realçar a cooperação. O que se propõe é uma mudança de perspectiva sobre o curso em questão, o aluno não é mais considerado individualmente, mas como parte de uma equipe que resulta da soma das potencialidades de cada um. A distribuição em grupos de trabalho incentiva a diversidade e o aproveitamento do potencial de cada membro para melhorar o desempenho da equipe.

A explicação recíproca como eixo do trabalho diário não só facilita a aprendizagem entre pares, mas, sobretudo, potencializa a consolidação do conhecimento por parte de quem explica. Promovem-se a responsabilidade, a autonomia, a pró-atividade, tudo no quadro da cooperação. O que leva a ressignificar a experiência de aprendizagem e que está ligada a emoções positivas.

O trabalho cooperativo permite o estabelecimento de relações responsáveis e duradouras, aumenta a motivação e o comprometimento, além de promover o bom desenvolvimento cognitivo e social. A cooperação surge diante da dúvida. Se um aluno tem uma pergunta, ele pergunta a um membro disponível do grupo designado. Se a dúvida persistir, o facilitador é convocado. Caso não o resolvam, o facilitador pedirá a todos que parem e cooperem como uma equipe na resolução da questão levantada. É assim que todos os membros da mesa vão debater, procurando a solução. Se ainda assim não conseguirem resolver, irão anotar a dúvida que será abordada de forma assíncrona pelo suporte técnico ou de forma síncrona na próxima aula pelo professor.

O trabalho começa junto ao docente, perante a dúvida:

COOPERAÇÃO



Todos os dias, ao final do encontro, os alunos reconhecerão um dos integrantes do grupo com quem compartilharam aquele dia. O critério para esse reconhecimento é a cooperação.



Cada grupo terá um facilitador, que será escolhido com base nos reconhecimentos, e desenvolverão um sistema de rotação onde qualquer pessoa pode assumir esse papel. O facilitador não é uma figura estática, mas cumpre um papel dinâmico e versátil: é um estudante que mobiliza a realização dos objetivos comuns da equipe, pondo em ação a cooperação. É ele quem partilha com a mesa o seu potencial a favor do resto da equipe e que, por isso, promove a cooperação.

Informações sobre a matéria

- Modalidade 100% à distancia.
- Número total de semanas: 9.
- Número de aulas virtuais no Playground: 27.
- Número total de aulas ao vivo: 27.

Requisitos e correlações

Para começar esta primeira parte de Especialização em Back End, é necessário ser aprovado no Projeto Integrador 1. Por sua vez, ser aprovado nessa disciplina é um requisito para se iniciar o Projeto Integrador 2.

Modalidade de trabalho

Nossa proposta educacional é pensada especialmente para a modalidade 100% à distância, através da aprendizagem ativa e colaborativa sob o nosso lema “aprender fazendo”. É por isso que os ambientes de aprendizagem são síncronos e assíncronos, com uma abordagem que vincula teoria e prática, para que ambas estejam presentes o tempo todo.

Temos um campus virtual próprio onde encontraremos atividades, vídeos, apresentações e recursos interativos com modelos de trabalho individual e em equipe para aprofundar cada um dos conceitos.

Além disso, realizaremos reuniões online e ao vivo com o grupo de alunos e professores, às quais podemos nos associar de onde estivermos por meio de uma plataforma de videoconferência com nossa câmera e microfone para gerar uma experiência próxima.



Metodologia de avaliação

A avaliação formativa é um processo contínuo que gera informações sobre a formação de nossos alunos e de nós como educadores. Isso gera um conhecimento do tipo feedback, ou seja, tem uma função de conhecimento, pois nos permite conhecer os processos de ensino e aprendizagem. Também tem uma função de melhoria contínua, porque nos permite saber onde estamos no processo, para validar se continuamos no caminho planejado ou se precisamos tomar novas decisões para cumprir os objetivos propostos.

Finalmente, a avaliação desempenha um papel importante em termos de promoção do desenvolvimento de competências valiosas. Nosso objetivo é nos diferenciar da avaliação tradicional, que muitas vezes é um momento difícil, chato e tenso. Para isso, vamos utilizar a gamificação, que é uma técnica onde são aplicados elementos de jogo para tornar o conteúdo mais atraente, para que, assim, os participantes se sintam motivados e imersos no processo, utilizem os conteúdos de aprendizagem como desafios que realmente desejam superar e aprendam com o erro.

Por sua vez, para registrar tal formação, é utilizado um conjunto de instrumentos, para os quais é essencial utilizar a maior variedade possível, e técnicas de análise.

Critérios de aprovação

- Realizar as atividades do Playground (80% de conclusão).
- Presença em encontros síncronos (90% de presença).
- Obter uma pontuação de 7 ou mais na avaliação final.
- Obter uma pontuação de 7 ou mais na nota final da matéria.



Conteúdos

Módulo 1: Introdução a microsserviços

Neste módulo iremos abordar os conceitos e características que definem uma arquitetura de software baseada em microsserviços. Sua diferença com as arquiteturas mais tradicionais e as vantagens sobre estas em um contexto de alta demanda, escalabilidade e mudanças de software.

Aula 1: O que é um microsserviço?

- Boas-vindas à disciplina
- O que é a arquitetura de software?
- O que é um microsserviço?
- Monolito vs. microsserviços
- Vantagens de uma arquitetura orientada a microsserviços

Aula 2: Introdução a microsserviços

- Desafios dos microsserviços
- Padrões de design
 - Service discovery
 - Edge server
 - Central configuration
 - Centralized log analysis
 - Distributed tracing
 - Circuit breaker
 - Reactive microservices
 - Centralized monitoring and alarms



Aula 3: Integradora I

Módulo 2: Framework Spring Cloud

Estudaremos detalhadamente os diferentes componentes do framework Spring Cloud, utilizado pela Netflix para a construção de uma arquitetura orientada a serviços, com base em suas características, configurações, desenvolvimentos e a própria implementação.

Aula 4: Eureka Server

- Arquitetura Eureka Server
- Registro e descobrimento de microserviços: que problemas ele vem solucionar?
- Configuração de Eureka Server
- Spring boot actuator

Aula 5: Configuração em sistemas distribuídos

- Introdução a spring cloud config server
- Configuração de microserviços
- Conexão de spring cloud config server a Git
- Debugging com spring cloud config server

Aula 6: Integradora II

Aula 7: Invocações REST declarativas e balanceamento de carga

- Introdução a Feign
- Feign REST client para invocação de serviços
- Balanceamento de carga com Eureka, Feign e Spring Cloud Loadbalancer



Aula 8: API Gateway

- Configuração API Gateway
- Configuração de Zuul API Gateway
- Discovery Locator
- Routing
- Logging Filter

Aula 9: Integradora III

Aula 10: Oficina de código

Aula 11: Prática pré-avaliação

Aula 12: Avaliação I

Aula 13: API Gateway - Segurança

- Segurança com OAuth

Aula 14: Circuit Breaker pattern

- O que é a tolerância a falhas?
- Resilience4j
- Novas tentativas (Retry & Fallback)

Aula 15: Integradora IV

Aula 16: Distributed Tracing

- O que é distributed tracing?
- Implementação de Spring Cloud Sleuth



- Zipkin
- Zipkin dashboard

Módulo 3: Mensagens assíncronas

Vamos nos apresentar às mensagens assíncronas, que são amplamente utilizadas em aplicativos de negócios de alto desempenho e segurança. Aprenderemos como implementar uma fila de mensagens em aplicativos que precisam garantir que nenhuma delas se perca.

Aula 17: Comunicação assíncrona

- Comunicação síncrona vs. assíncrona
- RabbitMQ
- Spring Cloud Stream
- Implementação de Zipkin com RabbitMQ

Aula 18: Integradora VI

Aula 19: Log Aggregation

- Padrão “Log Aggregation”. O que é? Qual problema soluciona? Como?
- Boas práticas
- Implementação com ELK

Módulo 4: Implementação sobre Docker

Neste último módulo nos dedicaremos a abordar o problema da infraestrutura mais adequada para implementar os componentes do Spring Cloud.

Aula 20: Docker e microsserviços - Parte I

- Arquitetura Docker com microsserviços
- Instalação de Docker



- Deploy de uma aplicação em Spring Boot
- Conceitos de Docker
 - Registry
 - Repository
 - Tag
 - Image
 - Container

Aula 21: Integradora VII

Aula 22: Docker e microsserviços - Parte II

- Zipkin e RabbitMQ com Docker
- Eureka com Docker
- Spring Cloud API Gateway com Docker

Aula 23: Prática pré-avaliação

Aula 24: Avaliação II

Aula 25: Aula especial I

Aula 26: Aula especial II

Aula 27: Encerramento

Material de referência

- Larsson, M. (2021). *Microservices with Spring Boot and Spring Cloud - Second Edition: Build resilient and scalable microservices using Spring Cloud, Istio, and Kubernetes*. Packt Publishing.
(<http://library.lol/main/2C1CEAC5CF6E2070C9823C97265A3AFC>)



- Mińkowski, P. (2018). *Mastering Spring Cloud*. Packt Publishing.
(<http://library.lol/main/062FC41CE4504CE94E1DD4CDA1AAA767>)
- Sharma, S. (2017). *Mastering microservices with Java 9 : build domain-driven microservice-based applications with Spring, Spring Cloud, and Angular*. Packt Publishing. (<http://library.lol/main/E5F71FCCE5BDD3BF72CE62084DADA426>)
- Karanam, R. R. (2017). *Mastering Spring 5.0: Master reactive programming, microservices, Cloud Native applications, and more*. Packt Publishing.
(<http://library.lol/main/C54624B796FBC87FE2F5D8770CF1BE8E>)
- Long, J. y Bastani, K. (2017). *Cloud Native Java Designing Resilient Systems with Spring Boot, Spring Cloud, and Cloud Foundry*. O' Reilly.
(<http://library.lol/main/EEC3E15D8687C0784E938F2B20B5D1BD>)