



**Certified Tech
Developer**

The Ultimate Degree

Backend I

Objetivos de aprendizagem

Implementar um backend implica em conhecer uma grande quantidade de tecnologias, geralmente não é suficiente conhecer apenas linguagens de programação. No decorrer desta disciplina, o aluno aprenderá os frameworks mais utilizados através de um conjunto de tecnologias para o desenvolvimento de aplicações web, desde o acesso aos dados, passando pela implementação do padrão MVC e a exposição dos serviços API REST.

Metodologia de Ensino-Aprendizagem

A Digital House possui como modelo de ensino momentos de aprendizagem síncronos e assíncronos com foco no vínculo da teoria com a prática, mediante uma aprendizagem ativa e colaborativa.

Nossa proposta inclui aulas ao vivo com seu grupo de alunos e docentes, as quais poderá participar de onde estiver. Também possuímos um campus virtual, nela você encontrará as aulas virtuais, vídeos, apresentações e recursos interativos para realizar no seu próprio ritmo antes da aula ao vivo.

Ao longo da sua experiência de aprendizagem na Digital House você conseguirá desenvolver habilidades técnicas e pessoais, como trabalho em equipe, criatividade, responsabilidade, compromisso, comunicação efetiva e autonomia.

Na Digital House utilizamos a metodologia “Sala de Aula Invertida”. E o que isso quer dizer? Cada semana vamos precisar que você se prepare para as aulas mediante a leitura de texto, assistindo vídeos, realizando tarefas, entre outros materiais. Desta forma, quando ocorrer o

encontro ao vivo, você estará pronto para uma abordagem do tema mais proveitosa e profunda.

Utilizamos atividades e estratégias baseadas em uma metodologia participativa e ativa para colocar em prática, uma vez que só é possível saber se efetivamente aprenderam enquanto realizam uma atividade prática. Por esse motivo, organizamos as aulas para que em todas tenham atividades práticas com as diferentes ferramentas, linguagens e competências que compõem a formação de um programador. Concebendo que a sala de aula é um espaço para a prática.

Um dos principais focos da nossa metodologia de aprendizagem é a aprendizagem prática. Por este motivo, durante a matéria haverá muitos exercícios de diversos tipos e complexidades que permitirão que você se desenvolva e garanta a assimilação dos ensinamentos. Desta forma, será possível um processo de aprendizagem mais significativo e efetivo, assimilação do conteúdo mais eficaz e duradoura, relacionar o aprendizado com a realidade dos desenvolvedores web, fomentar a autonomia e o autoconhecimento, melhorar a análise, a relação e compreensão dos conceitos. Todas essas ferramentas te ajudarão a exercitar uma grande variedade de competências.

A aprendizagem entre pares é um dos elementos centrais de nossa metodologia, por isso, em cada aula propomos uma mesa de trabalho junto aos seus companheiros de sala. Ao longo das aulas, os grupos irão variar os integrantes para potencializar a cooperação. Essa proposta visa que o aluno siga sua trajetória acadêmica como parte de uma equipe, de forma a potencializar a trajetória de cada aluno. A distribuição em grupos de trabalho fomenta a diversidade e aproveitamento do potencial de cada integrante para melhorar o rendimento da equipe.

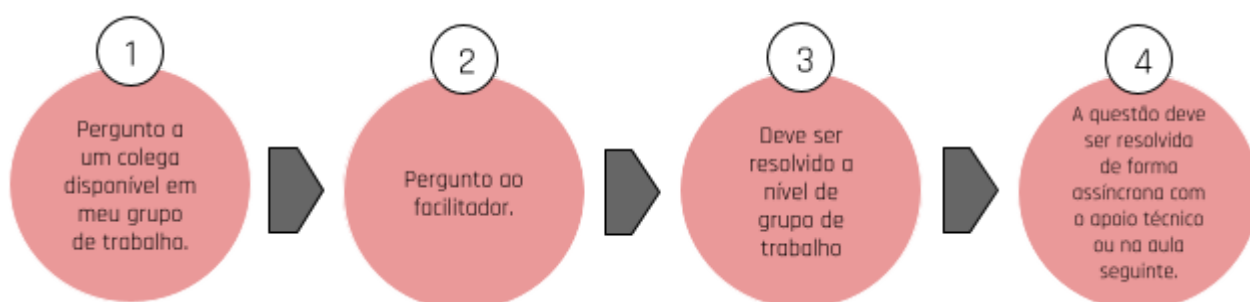
A explicação recíproca por parte dos alunos como prática diária facilita o aprendizado dos demais colegas, assim como potencializa a consolidação do conhecimento por parte de quem explica. Responsabilidade, autonomia e proatividade são promovidas, tudo no âmbito da cooperação. Isso leva a redefinir a experiência de aprendizagem e vinculá-la a emoções positivas.

O trabalho cooperativo permite estabelecer relações responsáveis e duradouras, aumenta a motivação e promove um bom desenvolvimento cognitivo e social. A cooperação surge diante da dúvida. Se você tiver alguma dúvida, pode perguntar a alguém do seu grupo que esteja disponível. Se a dúvida persistir, o facilitador será convocado. Se não resolverem, o facilitador pedirá a todos que parem para cooperar em equipe na resolução do conflito que levantou a

dúvida. É assim que todos os membros da mesa vão debater em busca da solução. Caso ainda não consigam resolver, deverão anotar a questão para que seja abordada de forma assíncrona pelo assistente 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 da aula, os alunos reconhecerão um dos membros do grupo com quem compartilharam aquele dia. O critério para esse reconhecimento é a cooperação.

Cada grupo terá um facilitador que será escolhido dentre as premiações e gerando um sistema de rodízio onde qualquer pessoa poderá desempenhar esse papel. O facilitador não é uma figura estática, mas cumpre um papel dinâmico e versátil. O facilitador é um aluno que mobiliza a realização dos objetivos comuns da equipe colocando em jogo a cooperação. É ele quem partilha o seu potencial com a mesa a favor do resto da equipa e, por isso, promove a cooperação.

Informações da matéria

- Modalidade 100% à distância
- Número total de semanas: 9
- Número de reuniões semanais síncronas: 6
- Aulas virtuais no nosso campus de Playground: 54
- Número de aulas presenciais: 54

Requisitos e correlatividades

Para cursar a disciplina de Back End I, você precisa ter sido aprovado nas disciplinas de: Programação Orientada a Objetos e Banco de Dados. Da mesma forma, para cursar as disciplinas de especialização de Back End, é necessário ter concluído o Back End I.

Modalidade de trabalho

Nossa proposta educacional é especialmente pensada para esta modalidade 100% a distância, por meio da aprendizagem ativa e colaborativa seguindo nosso pilar de “aprender fazendo”.

Os ambientes de aprendizagem são síncronos e assíncronos, com uma abordagem que une teoria e prática, para que ambas estejam presentes em todos os momentos.

Temos um Campus Virtual próprio no qual encontraremos atividades, vídeos, apresentações e recursos interativos com instâncias de trabalho individual e em equipe para aprofundar cada um dos conceitos.

Além disso, realizaremos encontros online e ao vivo com o grupo de alunos e professores, nos quais poderemos participar 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. A partir disso, é gerado um conhecimento de natureza retroalimentadora, ou seja, tem uma função de compreender o processo de aprendizagem, pois permite conhecer os processos de ensino e aprendizagem. Tem também uma função de melhoria contínua pois permite-nos saber onde estamos no processo, validar se continuamos no caminho planejado ou, se necessitamos, de tomar novas decisões para cumprir os objetivos propostos.

Por fim, a avaliação desempenha um papel importante em termos de promoção do desenvolvimento de competências valiosas.

Nosso objetivo é fugir da avaliação tradicional, onde muitas vezes é um momento difícil, chato



e tenso. Para isso, vamos usar a gamificação, uma técnica onde são aplicados elementos do jogo para que o conteúdo seja mais atrativo e você se sinta mais motivado e imerso no processo e use o conteúdo de aprendizagem como desafios que você realmente quer superar e aprender com o erro.

Por sua vez, para registrar esta formação, utilizamos um conjunto de instrumentos, para os quais é imprescindível utilizar a maior variedade e técnicas de análise possíveis.

Critérios de aprovação

- 90% de presença em aulas síncronas
- Conclusão de, pelo menos, 80% das atividades no campus virtual.
- Obter uma pontuação de 7 ou mais na avaliação final.
- Obter uma pontuação de 7 ou mais na nota final da disciplina.

Requisitos e correlatividades

Quais disciplinas o aluno deve ter cursado anteriormente?

- *Programação Orientada a Objetos*
- *Banco de dados*

Que matérias ele estuda a seguir?

- Backend II

Módulo 1: Design Patterns

Neste módulo, serão abordados os padrões de design (design patterns) mais usados na camada de backend das aplicações. Há uma grande variedade de bibliotecas e frameworks que os implementam e conhecê-los é crucial para entender e fazer o uso ideal da stack de tecnologias necessárias para acelerar o desenvolvimento.

Aula 1: Testes unitários JUnit

- Introdução a matéria



- Provas unitárias com JUnit
- Introdução: Teste Unitário vs Teste de Integração
- Benefícios e princípio FIRST
- Arquitetura JUnit
- Configuração da biblioteca
- Anotações, asserções
- Teste parametrizado
- Suite, Suite de Suites
- Exemplo em Java

Aula 2: Template Method pattern

- Template Method pattern
- Apresentação em UML
- Exemplo
- Implementação típica em Java

Aula 3: Integradora I

Aula 4: Chain of Responsibility pattern

- Chain of Responsibility pattern
- Apresentação em UML
- Exemplo
- Implementação típica em Java

Aula 5: Proxy pattern

- Proxy pattern
- Apresentação em UML
- Exemplo
- Implementação típica em Java

Aula 6: Integradora II

Aula 7: Flyweight pattern

- Flyweight pattern
- Apresentação em UML
- Exemplo
- Implementação típica em Java

Aula 8: Facade pattern (Fachada)

- Facade pattern (Fachada)
- Apresentação em UML
- Exemplo
- Implementação típica em Java

Aula 9: Integradora III

Módulo 2: Testes Unitários, Logging e Acesso a Dados

Estudaremos os três principais recursos que todo back-end deve possuir. Estes consistem em obter uma boa rastreabilidade no registro para detecção rápida de erros (solução de problemas), teste dos softwares que estamos desenvolvendo para garantir a qualidade nos estágios iniciais e acesso a bancos de dados relacionais por meio de boas práticas.

Aula 10: Logging (Trace e Debug)

Apresentar a importância de ter um framework de logging. Conceito de rastreabilidade e troubleshooting (solução de problemas). Uso de log4j, configurações e opções.

- Introdução ao logging (vantagens, desvantagens, avaliação de frameworks de logging)



- Arquitetura Log4j
- Instalação Log4j
- Criar arquivo de configuração Log4j
- Level Object, Logger
- Exemplo em Java
- Appender Objects (ConsoleAppender, FileAppender, RollingFileAppender, DailyRollingFileAppender, AsyncAppender)

Aula 11: Acesso a Banco de Dados

JDBC. Statement. Criação de Conexões. Resultset. Inserções, exclusões, atualizações e consultas. Gerenciamento de transações.

- H2 Database
- Arquitetura JDBC
- Conexão a um banco de dados
- Statement
- Insert e delete

Aula 12: Integradora IV

Aula 13: Consultas e transações no banco de dados

- Consultar dados (execute, ResultSet)
- Modificar dados (executeUpdate)
- Utilização de PreparedStatement
- Transações (commit, auto commit, Rollback)
- Invocar StoredProcedures
- Conectar no banco de dados com JDBCAppender

Aula 14: Padrão DAO (Data Access Object)

Camadas para acesso a dados com JDBC. Importância de ter uma camada de acesso a dados. Uso do Template Method pattern no padrão DAO. Reuso. Introdução ao conceito ORM.

- DAO
- Apresentação em UML
- Implementação típica em Java
 - Classes de acesso a dados
 - Utilização de Factory e Template Method
 - Classe de Serviço
 - Geração de tipos de exceção em camadas
- Teste unitário de cada camada
- Introdução ao conceito de ORM

Aula 15: Integradora V

Aula 16: Oficina Coding

Aula 17: Avaliação 1

Aula 18: Maven

Introdução à gestão de projetos em Java, seu mecanismo e repositório, gestão de dependências, informação de projeto e plugins.

- Introdução ao Maven. O que é?
- Instalação do Maven
- Modelo conceitual de um projeto em Maven
- POM (Project Object Model)
- Super POM
- Dependências

- Ciclo de vida

Aula 19: Serialização de Objetos e I/O de Arquivos

Processamento de fluxos (streams) e uso adequado deles. Serializar e desserializar objetos. Distinguir os leitores e escritores dos fluxos, e selecionar apropriadamente entre eles. Gerenciamento de I/O de arquivos.

- Hierarquia de classes de fluxo (InputStream, OutputStream).
- Java NIO, gerenciamento de arquivos, pastas e recursos.
- Criação, escrita, leitura, exclusão, cópia e movimentação de arquivos.
- Serialização de objetos.
- Gerenciamento de pastas.

Aula 20: Integradora VI

Módulo 3: Framework para o desenvolvimento ágil de aplicações

Neste módulo o aluno irá adquirir os conhecimentos necessários para trabalhar com um dos frameworks mais utilizados nas empresas que facilita e acelera o desenvolvimento de aplicações de negócio. Você terá um conjunto de ferramentas para tornar o acesso aos dados mais simples, rápido, seguro e elegante por meio de um ORM, construção de APIs e visualizações web por meio do uso do padrão MVC. Durante o módulo, o aluno desenvolverá um trabalho integrador para colocar em prática o uso correto do framework.

Aula 21 Introdução ao Spring Boot

Introdução e primeiros passos com o framework para o desenvolvimento de aplicações Java

- Arquitetura cliente-servidor
- Protocolo HTTP
- Framework
 - Definição



- Framework vs Biblioteca
- Introdução ao Spring Framework
- Spring vs Spring Boot
- Criar um projeto em Spring Boot
 - Initializr
 - Estrutura do projeto
 - Application.java

Aula 22: Padrão MVC

- Model View Controller
- Objetivo MVC
- Spring MVC
- Spring Boot
- View

Aula 23: Integradora VII

Aula 24: Oficina de coding sobre el trabajo integrador

Construcción por parte de los alumnos del resto de las entidades con la capa de acceso a datos (DAO, Service) con testeos unitarios.

Aula 25: API Rest

- O que é uma API?
- Introdução, abordagem e diferenças com um monolito
- Rest vs SOAP
- API Rest / RESTful
- JSON
- Instalação Postman

Aula 26: Integradora VIII



Aula 27: API Rest II

- Anotações no Controller
- DTO
- ResponseEntity
- Consumir API

Aula 28: Consumir APIs

- Invocar uma API a partir da view
- Invocar uma API a partir do backend

Aula 29: Integradora IX

Aula 30: Injeção de Dependências

- Injeção de dependências
- IoC - Inversion of Control
- Formas de Injetar dependências

Aula 31: ORM

- Arquitetura JPA
- Hibernate
- Annotations
- Configurar Hibernate com Spring Boot

Aula 32: Integradora X

Aula 33: Spring Data - Hibernate

- Relações (1 para 1, 1 para muitos, muitos para muitos)
- JoinColumn

- Cascade
- Fetch Type

Aula 34: HQL

- Características
- Sintaxe
- Localização da consultas em Spring

Aula 35: Integradora XI

Aula 36: Oficina de Coding

Revisão pelo professor da construção da camada de acesso a dados (Repositório), camada de negócios (serviço), APIs (controlador) e a camada de visualização (HTML com JavaScript) de uma entidade e seu mapeamento com a base de dados, mostrando as alterações em relação ao DAO original.

Aula 37: Oficina de Coding

Construção pelos alunos da camada de acesso a dados (Repositório), negócio (serviço), APIs (controlador) e visualização (HTML com JavaScript) de uma entidade e seu mapeamento com o banco de dados.

Aula 38: Bancos de dados não-relacionais: MongoDB

- Introdução ao MongoDB
- MongoDB vs Bancos de dados relacionais
- Spring Data + MongoDB
- Criação de uma base MongoDB
- Acesso ao MongoDB

Aula 39: Integradora XII



Aula 40: Sistemas Distribuídos

- O que é um micro-serviço?
- Padrões de comunicação entre micro-serviços
- Monolito vs Micro-serviços

Aula 41: Tratamento de Exceções

- ControllerAdvice
- ExceptionHandler
- Exception Handling para serviços RESTful

Aula 42: Integradora XIII

Aula 43 Segurança

- Biblioteca starter de Spring Security
- Autenticação e segurança de URL
- Criptografar senhas
- @PreAuthorize e @PostAuthorize

Aula 44: Autenticação de APIs baseadas em tokens JWT

- Json Web Token (JWT)
- Construção de uma API de Geração de Token

Aula 45: Integradora XIV

Aula 46: Testes de Integração

- Teste de integração com Spring Boot
- Testes de integração para controllers com Mock



- Cobertura de Código

Aula 47: Oficina de Coding (TI) sobre o nosso trabalho integrador

Adicionando segurança ao trabalho integrador usando um formulário de login de spring security

Aula 48: Oficina de Coding (TI) e entrega

Aula 49: Documentação

- Swagger OpenApi
- Hateoas

Aula 50: Implantar APIs no Docker

- Teste de Integração Cucumber
- Gherkin
- Maven Archetype
- Docker Compose

Aula 51: Integradora XVI

Aula 52 Oficina Coding (TI)

Aula 53 Estamos chegando ao final

Aula 54: Chegamos ao final!

Classes de negócio, Classes de acesso a dados, Uso de ORM, Classe de serviço, Teste Unitário e Integral, View (templates) e Controllers (MVC), Expor uma API Rest, Preencher uma combinação com API Rest