



SÃO
PAULO
TECH
SCHOOL

Técnicas de programação WEB

Aula 1

Apresentação da Disciplina

Diego Brito

`diego.lima@sptech.school`

Tópicos da Aula

- Apresentação do professor
- Regras
- Intervalo
- Avaliações
- Conteúdo da disciplina
- Mercado de trabalho
- Revisão uso de tipos
- Introdução a programação WEB

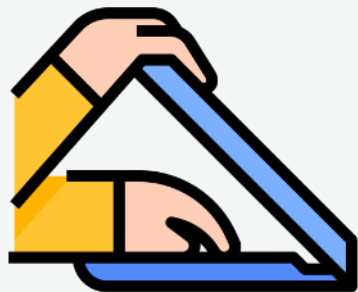


Diego Brito

- 29 anos, casado
- **Formação**
 - Análise e Desenvolvimento de sistemas (SPTECH);
 - Pós-graduado em Tecnologias para Aplicações Web (Anhanguera)
 - Mestrando Administração de Organizações (FEA-RP/USP).
- **Experiência profissional e projeto que atuou**
 - Consultor Oracle:
 - Movice
 - Chama (App do gás)
 - WebMotors
 - Ebanx
 - Latam
 - Matrix
 - Supervisor na **SPTECH Solutions**
 - Professor Titular na **SPTech**;

Regras Gerais

Regras básicas da sala de aula



1. **Notebooks Fechados:** Aguarde a liberação do professor;
2. Celulares em modo **silencioso e guardado**, para não tirar sua atenção
 - Se, caso haja uma situação urgente e você precisar **atender ao celular**, peça licença para sair da sala e atenda fora da aula.



3. **Proibido usar Fones de ouvido:** São liberados apenas com autorização do professor.

4. **Foco total no aprendizado**, pois nosso tempo em sala de aula é precioso.

- Venham sempre com o **conteúdo da aula passada em mente** e as atividades realizadas.
- Tenham caderno e caneta;
- **Evitem faltas e procure ir além** daquilo que lhe foi proposto.
- **Capricho, apresentação e profundidade** no assunto serão observados.



- **“frequentar as aulas** e demais atividades curriculares aplicando a **máxima diligência no seu aproveitamento**” (Direitos e deveres dos membros do corpo discente - Manual do aluno, p. 31)

Regras básicas da sala de aula



As aulas podem e devem ser divertidas! Mas:

- **Devemos respeitar uns aos outros** – cuidado com as brincadeiras.
 - “observar e cumprir o regime escolar e disciplinar e comportar-se, dentro e fora da Faculdade, **de acordo com princípios éticos condizentes**” (Direitos e deveres dos membros do corpo discente – Manual do aluno, p. 31)

Boas práticas no Projeto

COMPROMISSO



COM VOCÊ:
ARRISQUE, NÃO
TENHA MEDO DE
ERRAR



COM OS
PROFESSORES:
ORGANIZE A **ROTINA**
PARA OS ESTUDOS

COM OS COLEGAS:
PARTICIPAÇÃO
ATIVA E PRESENTE



COM O PROJETO:
RESPEITO E
FLEXIBILIDADE


Respeito

Boas práticas no Projeto

Reações **defensivas** não levam
ao envolvimento verdadeiro!

Transforme cada problema e
cada dificuldade em uma
OPORTUNIDADE de aprendizado
e crescimento.

EVITE:

- Justificativas e Desculpas
- Transferir a culpa
- Se conformar com o que sabe
- Se comparar com o outro

Dica: **Como ter sucesso** (Maiores índices de aprovações)

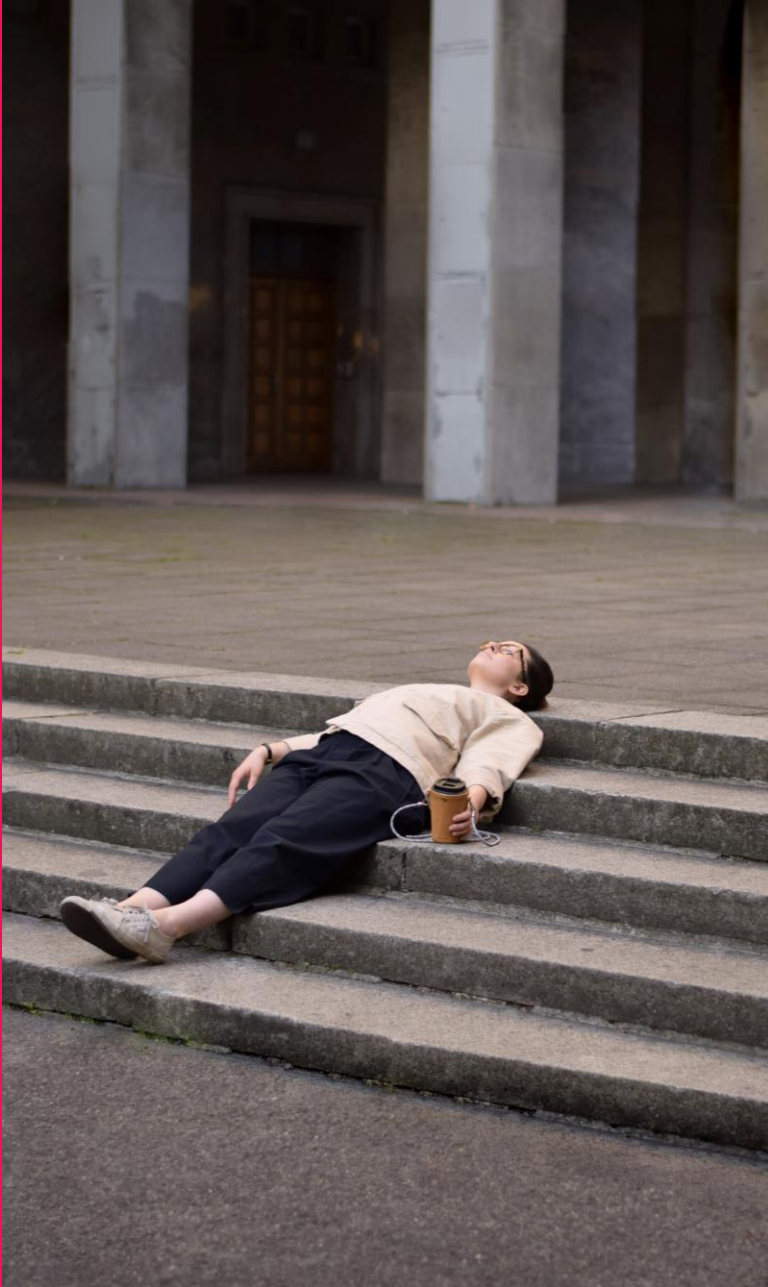
Comprometimento

- Não ter faltas e atrasos. Estar presente (*Não fazer 2 coisas ao mesmo tempo*)
- Fazer o combinado cumprindo os prazos

Atitudes Esperadas:

- **Profissionalismo**: Entender que não é mais ensino médio (*Atitude, comportamento, etc.*)
- **Não estar aqui só pelo** estágio ou pelo diploma
- Não ficar escondido: precisa **experimentar**
- **Trabalhar** em grupo e **participar** na aula
- **Não ser superficial** ou “achar que sabe”
- **Não se enganar** utilizando de “cola”
- Assumir a responsabilidade: Não colocar a culpa em outra coisa. **Não se vitimizar.**

Regras de intervalo



Intervalo

Horário: 19:00 – 19:30

Atenção: Atrasados deverão aguardar autorização para entrar na sala.

Break

> 10 minutos, definidos pelo professor.

Obs: Permanecer no andar, casos específicos me procurar.

Avaliações

Avaliações da Disciplina

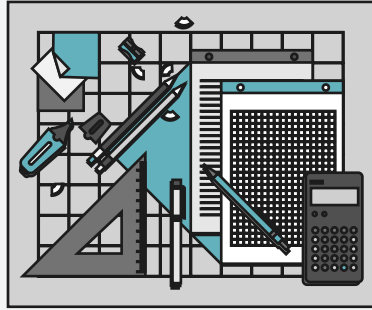
Avaliação Continuada

- 3 continuadas, sendo 1 por sprint
- Conjunto de atividades que formam a nota de uma continuada
- Representa **40%** da nota final, ou seja, cada nota de continuada vale no máximo 1,33

Avaliação Integrada

- 1 prova múltipla escolha
- 1 prova dissertativa
- Realizadas no mesmo dia no final do semestre
- Questões interdisciplinares
- Representa **60%** da nota final

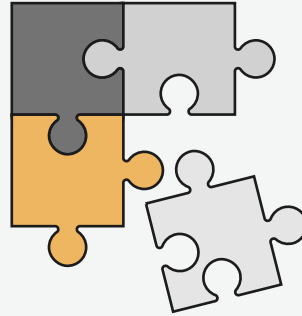
Formato Continuadas



Atividades semanais

Entre 2 a 4 **entregas individuais** (atividades)

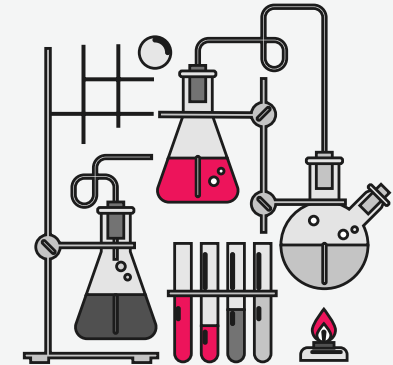
As entregas representam **30%**
Da nota final da continuada.



Integradinha (Moodle)

Prova feita no Moodle com
questões interdisciplinares

A integradinha representa **30%**
da nota final da continuada.



Avaliação Prática

Provas práticas devem ser
entregues via Moodle

A prova prática representa **40%**
Da nota final da continuada.

Conheça Xampson

- Esse foi o desempenho do Xampson na Sprint 1:
- Entregou **somente um** de **dois** projetos das atividades semanais devido a má gestão do tempo.
- Correu atrás do “preju” e tirou **8,5 na integradinha**.
- Caiu na zona de conforto e não praticou o suficiente, obteve 6 na prova prática.

Portanto a nota final de Xampson é...



Nota final [Simulação]

	Atividades Semanais (2 atividades)	Integradinha (Moodle)	Avaliação Prática
Peso	30%	30%	40%
Atividade 1	10	8,5	6
Atividade 2	0	--	--
Atividade 3	--	--	--
Atividade 4	--	--	--
Nota final Atividade	1,5	2,55	2,4
Nota final continuada 1		6,45	

Dicas :

Acreditamos que a nota é reflexo da jornada, portanto:

- São poucas aulas antes de cada prova;
- Tá com dúvida? Pergunte!
- Participe e contribua para uma aula mais rica e com mais exemplos;
- Não deixa para última hora;
- Peça ajuda o quanto antes;
- Colabore com as aulas;
- **Tenha foco e aproveite a jornada!**



Proposta

WEB

Criação de **REST API** (Backend) com **Spring Boot**.

Tecnologias:

- Java
- Spring
- IntelliJ

Bootcamp

Criação de **SPA** (Frontend) com **React**.

Tecnologias:

- WEB (JS, CSS e HTML)
- React
- VSCode

Objetivos

Integração total entre o **frontend** e o **backend**, incluindo integrações com **outras plataformas**.

Conteúdo da Disciplina



Conteúdo Programático

- Conceitos básicos de REST
- Conceitos básicos de Spring Boot
- REST Controllers
- ORM com JPA
- Repositories
- Consumo de outras API's (Client Rest integrado)
- Testes automatizados com JUnit

Mercado de trabalho



Mercado de trabalho

LinkedIn:

- **Spring Boot** – 2.084 vagas (Brasil, 28/01/2023);
- **React** – 3.559 vagas (Brasil, 28/01/2023);

Indeed:

- **Spring Boot** – 881 vagas (Brasil, 28/01/2023);
- **React** – 2.648 vagas (Brasil, 28/01/2023);

**Nos episódios anteriores
em LP...**

Tipos de dados em Java

Diferença entre primitivos e classes Wrapper:

Os tipos **primitivos** em Java são tipos básicos, como int, double, char, etc. Eles não são objetos e não têm métodos.

Já as **classes wrappers** em Java são classes que envolvem os tipos primitivos e os tornam objetos. Por exemplo, o tipo primitivo int é envolvido pela classe Integer. As classes wrappers possuem métodos que permitem trabalhar com os valores de forma mais conveniente.

Resumindo: tipos primitivos são básicos e não são objetos; classes wrappers são objetos que envolvem tipos primitivos.

Leitura sugerida
apostila: introdução a linguagem de programação

Tipos de dados em Java

Pontos de atenção:

Primitivos **não permitem o valor nulo!**

Primitivos quando utilizados como **atributos de instância**, possuem um valor padrão caso não sejam inicializados.

O tipo **Character** ou o primitivo equivalente **char** são utilizados com aspas simples.

Leitura sugerida
apostila: introdução a linguagem de programação

Introdução a programação WEB

Integração de sistemas

A **integração de sistemas** foi um dos maiores desafios com o crescimento da computação.

Antigamente, a integração entre sistemas era feita manualmente, com exportação de dados de um sistema e digitação em outro pelo usuário.

Existia uma profissão chamada **digitador**.

A **chegada da internet** só aumentou o desafio, com o surgimento das linguagens de programação.

Integração de sistemas

As primeiras tentativas de integrar sistemas depois do "Boom" da computação precisavam considerar a plataforma de desenvolvimento, ex: No caso do Java foi criado o **EJB (Enterprise JavaBeans)**.

A complexidade enfrentada na época era a de estabelecer comunicação através da rede com outras aplicações, que haviam sido construídas em plataformas diferentes.

SOAP

SOAP (Simple Object Access Protocol) foi a primeira solução que emplacou no mundo corporativo, para permitir a interoperabilidade entre aplicativos em diferentes plataformas e linguagens de programação.

O SOAP é baseado em **XML** e usa o protocolo HTTP ou SMTP para transportar as mensagens entre aplicativos. Ele foi projetado para fornecer uma forma padronizada de acesso a objetos em diferentes sistemas, independentemente de plataforma ou linguagem de programação.

O SOAP inclui uma camada adicional de segurança e autenticação, o que o torna adequado para aplicativos empresariais que precisam de proteção contra acesso não autorizado.

Apesar de ter sido amplamente utilizado em aplicativos corporativos, o SOAP perdeu popularidade com o crescimento do REST, que é mais simples e flexível. No entanto, o SOAP ainda é amplamente utilizado em aplicativos críticos, como sistemas bancários, de saúde e financeiros, devido à sua segurança e confiabilidade.

REST

REST (Representational State Transfer) é um conjunto de princípios de arquitetura para construir aplicativos da web. Ele foi proposto por Roy Fielding, um dos criadores do protocolo HTTP, em sua tese de doutorado de 2000.

A arquitetura REST é baseada em alguns princípios:

- Interface uniforme: a interface de comunicação entre cliente e servidor é baseada em **URI** (Uniform Resource Identifier) e HTTP.
- Sistema de mensagens: a comunicação entre cliente e servidor é baseada em mensagens HTTP, como **GET, POST, PUT** e **DELETE**.
- Sistema de camadas: o sistema pode ser dividido em camadas, o que permite uma maior flexibilidade e escalabilidade.

A ideia principal do REST é que a arquitetura da web é **baseada em recursos**, que são representados por **URIs** e manipulados através de **verbo HTTP** como GET, POST, PUT, DELETE.

REST é usado para construir aplicativos simples e escaláveis, que são fáceis de manter e expandir.

SOAP X REST

- A principal diferença entre eles é que SOAP é um protocolo **mais antigo e mais pesado**, enquanto REST é mais **moderno e mais leve**.
- SOAP é um protocolo baseado em **XML** que fornece uma estrutura para troca de mensagens, incluindo autenticação, validação, criptografia e outras funcionalidades adicionais. Ele é usado principalmente para aplicativos de grande escala e integração de sistemas.
- REST, por outro lado, é um conjunto de princípios de arquitetura para construir aplicativos da web. Ele usa HTTP para obter recursos e usa o formato JSON para transportar dados. Ele é **mais flexível e escalável** do que o SOAP.
- Outra diferença importante é que SOAP é conhecido por sua segurança, enquanto REST é conhecido por sua escalabilidade e simplicidade.

Exercício

APP

WhatsApp

**Internet
Banking**

**Caixa
Eletrônico**

**Como eu posso verificar meu saldo em uma conta
bancária?**

Telefone

**Agência
Bancária**

**Extrato
Bancário
(email/correio)**

Agradeço
a sua atenção!

Diego Brito

diego.lima@sptech.school

SÃO
PAULO
TECH
SCHOOL