**Full Stack**

Pilha cheia de coisas

Aulas: <https://glysns.gitbook.io/java-full-stack>

Em tradução literal, **Full Stack** quer dizer ***Pilha Cheia***, mas qual a relação deste nova categoria de profissionais **Desenvolvedor Full Stack** na área de tecnologia?

Até pouco tempo atrás as empresas classificavam os profissionais de tecnologia em:

* **Programador Front-End**
* **Programador Back-End**
* **Analista de Banco de Dados**
* **Analista de Infraestrutura (DevOps)**

Homem sentado na cadeira

Descrição gerada automaticamente com confiança média

Com o tempo as empresas perceberam uma enorme dificuldade em gerenciar estes times individualmente com a proposta de construção de produto como um software.

O maior dilema para as empresas com times distribuídos por habilidades, é o fato da dificuldade de compreensão dos requisitos e a proposta de funcionalidades do projeto.

É por esta razão que hoje o mercado de TI tem exigido e recrutado profissionais em desenvolvimento que sejam capazes de dominar as quatro habilidades citadas acima, sendo denominado como um **Desenvolvedor** **Full Stack**.

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

**Back-end**

Back-end representa a parte ***por trás*** de toda interação de usuários em um software, seja um site, um sistema desktop, uma aplicação web e até mesmo os aplicativos.

Também demoninado como **Server Side** (do lado do servidor), a camada de back-end é responsável por disponibilizar os recursos abaixo:

* Regra de negócio
* Acesso a dados
* Segurança
* Integração via APIs

Para nos tornarmos um programador back-end, precisamos conhecer uma ou algumas das linguagens de programação e bancos de dados disponíveis e utilizados pelas empresas.

**Linguagens de Programação**

* JavaScript
* Java
* C#
* PHP
* Kotlin
* Scala
* Perl
* C++
* Python
* Ruby

Conheça mais sobre as características de cada linguagem pelo link [ewally](https://www.ewally.com.br/blog/ajudando-sua-empresa/backend/)​

**Front-end**

Front-end está diretamente relacionado ao desenvolvimento dos recursos disponíveis para que haja a interação dos usuários em nosso software ou aplicativo. Todos os componentes gráficos e compartamentos que acontecem devido ao uso do aplicação é de desenvolvida por um profissional denominado de **Dev Front End**.

Um profissional Dev Front End deve pensar em requisitos de suma importância para o desenvolvimento de qualquer aplicação que necessite de interação com o usuário.

* Aparência
* Usabilidade
* Objetivo
* Segurança

Diferentemente da camada de back-end, o front-end é composto por uma linguagem de programação a Java Script e tecnologias complementares que são conhecidas como: Hypertextos, Gerador de estilos, Bibliotecas ou Frameworks nos quais vamos listar abaixo:

Tecnologia

Definição

HTML

Linguagem de marcação de hypertexto

CSS

Linguagem de estilização de conteúdo HTML

Bootstrap

Biblioteca de estilos CSS pré-definidos criado pelo Twitter

JQuery

Biblioteca Javascript muito utilizada antes da era das aplicações SPA.

NodeJS\*

Plataforma capaz de executar códigos JavaScript fora de um navegador web.

VueJS

Framework JavaScript de código-aberto, focado no desenvolvimento de interfaces de usuário e aplicativos de página única.

React

O React é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário em páginas web. É mantido pelo Facebook, Instagram, outras empresas e uma comunidade de desenvolvedores individuais.

Angular

Angular é uma plataforma de aplicações web de código-fonte aberto e front-end baseado em **TypeScript** liderado pela Equipe Angular do Google e por uma comunidade de indivíduos e corporações

TypeScript

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de **JavaScript** e ***adiciona tipagem estática opcional*** à linguagem.

**Desenvolvedor Web Full Stack**

​

Um desenvolvedor web full stack é uma pessoa que pode desenvolver software **cliente** e **servidor** .

Além de dominar HTML e CSS, ele também sabe:

* Programe em um **navegador** (como usar JavaScript, jQuery, Angular ou Vue)
* Programar um **servidor** (como usar PHP, Java, Python ou Node)
* Programar um **banco** de dados (como usar SQL, SQLite ou MongoDB)

Para se tornar um excelente **Desenvolvedor Web Full Stack** você jamais deverá esquecer que as empresas também buscam e valorizam profissionais especialistas capazes de dominar com excelência algum ou alguns dos requisitos essenciais no desenvolvimento de softwares.

Gleyson Sampaio

**Introdução**

Interessada em dispositivos eletrônicos inteligentes, a Sun Microsystems financiou uma pesquisa interna com o codinome Green em 1991. O projeto resultou no desenvolvimento de uma linguagem baseada em C e C++ que seu criador, James Gosling, chamou de Oak (carvalho) em homenagem a uma árvore que dava para a janela do seu escritório na Sun.

Hoje a linguagem Java é mantida pela [Oracle Inc.](https://www.oracle.com/java/technologies/downloads/) e já esta na sua versão 18 do JDK.

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança média

# Orientada a Objetos

Em programação orientada da objetos, um objeto é uma abtração do mundo real composto por um identificador (nome), atributos (propriedades) e métodos (comportamentos) definidos em uma classe.

Em Java, uma classe é descrita em um arquivo com a extensão .java. Abaixo iremos explorar algumas expressões utilizadas na programação orientada a objetos e uma ilustração de estrutura de uma classe Student e a criação de objetos a partir desta classe.

* **Instanciar** (new)**:** É o ato de criar um objeto a partir de estrutura definida em uma classe.
* **Comportamentos** (behavior)**:** Também conhecido como **ações** ou **métodos**, é toda parte comportamental que um objeto dispõe.
* **Características** (states)**:** Também conhecido como **atributos** ou **propriedades**, é toda informação que representa o estado do objeto.
* **Identificador** (identity): Propósito existencial aos objetos que serão criados.
* **Classe** (class): A estrutura e ou a representação que direciona a criação dos objetos de mesmo tipo.

Diagrama

Descrição gerada automaticamente

Para ilustrar as etapas de desenvolvimento orientada a objetos em Java, iremos reproduzir a imagem acima em forma de código para explicar que primeiro criamos a estrutura correspondente para assim podermos cria-los com as características e possibilidade de realização de ações (comportamentos) como se fosse no "mundo real".

1

// Criando a classe Student

2

// Com todas as características e compartamentos aplicados

3

​

4

public class Student {

5

String name;

6

int age;

7

Color color;

8

Sex sex;

9

​

10

void eating(Food food){

11

//NOSSO CÓDIGO AQUI

12

}

13

void drinking(Eat eat){

14

//NOSSO CÓDIGO AQUI

15

}

16

void running(){

17

//NOSSO CÓDIGO AQUI

18

}

19

}

20

​

Copied!

1

// Criando objetos a partir da classe Student

2

​

3

public class School {

4

public static void main(String[] args) throws Exception {

5

Student student1 = new Student();

6

student1.name= "John";

7

student1.age= 12;

8

student1.color= Color.FAIR;

9

student1.sex= Sex.MALE;

10

​

11

Student student2 = new Student();

12

student2.name= "Sophia";

13

student2.age= 10;

14

student2.color= Color.FAIR;

15

student2.sex= Sex.FEMALE;

16

​

17

Student student3 = new Student();

18

student3.name= "Lily";

19

student3.age= 11;

20

student3.color= Color.DARK;

21

student3.sex= Sex.FEMALE;

22

}

23

}

24

​

25

​

Copied!

O que queremos ilustrar aqui é que em linguagem orientada a objetos, nós desenvolvedores focamos na estrutura da classe e é a aplicação que fica encarregada da criação dos objetos através de interação dos usuários na aplicação.

​

**Tecnologias**

Quando falamos em programar em Java devemos dominar além dos fundamentos da linguagem, um conjunto de ferramentas e frameworks exclusivos. Abaixo iremos listar as tecnologias que iremos utilizar, mas claro, cientes que além de disponibilidade de outras ferramentas, existem também recursos similares e com a mesma proposta.

* **Java JDK**: Kit de Ferramentas para desenvolvimento na linguagem.
* **Eclipse IDE**: Programa para edição, depuração compilação, testes e execução de programas Java
* **Hibernate**: Framework baseado no conceito de ORM para a persitência de dados em um banco relacional.
* **Spring Framework:** Plataforma para desenvolvimento de aplicações Java baseado nos padrões de projeto de inversão de controle e injeção de dependências.
* **Postgres**: Banco de dados relacional utilizado em projetos de grande porte
* **Swagger**: Ferramenta de documentação de APIs e Webservices
* **Github**: Repositório e gerenciador de versionamento de código fonte
* **Maven**: Sistema para gestão de dependência e distribuição de uma aplicação
* **REST**: Arquitetura para transferência de dados representativos como JSON
* **Heroku**: Plataforma para hospedagem de aplicações com fácil integração com Github

Logotipo, nome da empresa

Descrição gerada automaticamente

# Arquitetura

O projeto Banckline será desenvolvida na arquitetura do Spring Framework + Springboot estruturado no padrão MVC para prover uma API Rest para movimentações financeiras simples.

## Spring Framework

Framework open source desenvolvido para a plataforma Java baseado nos padrões de projetos inversão de controle e injeção de dependência.

Sua estrutura é composta por módulos afins de reduzir a complexidade no desenvolvimento aplicações simples ou corporativa.

Diagrama

Descrição gerada automaticamente

​

### Spring versus JavaEE

Diagrama

Descrição gerada automaticamente

​

Olhando um pouco a história, há muito, mas muito tempo atrás, o Java EE era realmente muito complicado e nem era necessário entrar numa discussão, usar o Spring era um caminho mais simples e mais fácil de evoluir. Aí chegou a versão 5 do Java EE e a discussão voltou a ficar um pouco mais quente.

### O Spring Framework é estruturado em dois pilares

**Inversion of Control ou IoC**, trata-se do redirecionamento do fluxo de execução de um código retirando parcialmente o controle sobre ele e delegando-o para um container. O principal propósito é minimizar o acoplamento do código.

**Injeçã o de dependência** é um padrão de desenvolvimento com a finalidade de manter baixo o nível de acoplamento entre módulos de um sistema.

Para desfrutrar dos recursos oferecidos pelo Spring Framework, precisamos compreender três conceitos essenciais.

* **Beans**: Objeto que é instanciado (criado), montado e gerenciado por um container através do princípio da inversão de controle.
* **Scopes**: Controle da existência de nossos objetos **Components** da aplicação em uma esfera de utilização: ***Singleton, Prototype, Reques, Session, Global***
* **Autowired:** Um recurso em forma deanotação (indicação) onde deverá ocorrer uma injeção automática de dependência.

## Springboot

Enquanto que o Spring Framework é baseado no padrão de injeção de dependências, o Springboot foca na configuração automática.

Diagrama

Descrição gerada automaticamente

Dado que a maior parte das configurações necessárias para o início de um projeto são sempre as mesmas, por que não iniciar um projeto com todas estas configurações já definidas?

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

### Starters

Os starters são dependências que agrupam outras dependências com um propósito em comum. Dessa forma, somente uma configuração é realizada no seu gerenciador de dependências

Aplicativo

Descrição gerada automaticamente com confiança média

### Alguns Starters

**spring-boot-starter-\***

* **data-jpa**: Integração ao banco de dados via JPA - Hibernate.
* **data-mongodb**: Interação com banco de dados MongoDB.
* **web**: Inclusão do container Tomcat para aplicações REST.
* **web-services**: Webservices baseados na arquitetura SOAP.
* **batch**: Implementação de JOBs de processos.
* **test**: Disponibilização de recursos para testes unitários como JUnit
* **openfeign**: Client HTTP baseado em interfaces
* **actuator**: Gerenciamento de monitoramento da aplicação.

**Configuração**

# Java JDK

A primeira etapa é instalarmos o JDK versão 8 ou superior, o Eclipse IDE para testarmos nosso primeiro programa de exibição de boas vindas.

​

### Instalando o JDK no Windows

Algumas IDEs como o [VSCode](https://code.visualstudio.com/docs/java/java-tutorial) já consegue baixar e instalar o JDK e realizar a configuração de forma fácil.

* Busque no Google por Java [JDK xxx](https://www.oracle.com/br/java/technologies/javase/javase8-archive-downloads.html) (versão desejada)
* Selecione a opção **.exe** de acordo com o seu sistema operacional. No meu exemplo eu baixei a versão: [jdk-8u202-windows-x64.exe](https://www.oracle.com/br/java/technologies/javase/javase8-archive-downloads.html#license-lightbox)​
* Role um pouco baixa e visualizará a lista de opções de download de acordo com o seu sistema operacional.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

* Depois você precisará realizar um login no site da [Oracle](https://login.oracle.com/mysso/signon.jsp).

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

* Após realizar o download, executar o instalador para instalar o Java no Windows. Este processo instalará tanto o **JDK** quanto a **JRE**.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Evite mudar o diretório de instalação

* Em seu explorer deve ter algo mais ou menos assim:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* Precisamos agora validar se a instalação também já configurou nossa variável de ambiente para poder executar o Java pelo Prompt de comando ou PowerShell do Windows.
* Abre o Prompt de comando e execute o comando java -version.

Texto

Descrição gerada automaticamente

Não iremos configurar as variáveis de ambiente pois não é um pré-requisito para desenvolver em Java considerando que iremos usufruir das mais poderosas IDEs para aumentar nossa produtividade em codificar na linguagem.

# Eclipse IDE

Vamos baixar umas das IDEs mais utilizada para projetos Java.

* Acesse o site para baixar o [Eclipse for Java EE Developers](https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-java-ee-developers).
* Identifique a versão do seu sistema operacional.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* Na sua pasta download deverá os executáveis do Java e do Eclipse, basta iniciar a instalação

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

* A opção recomendada é **Eclipse IDE for Enterprise Java and Web Developers**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Esta etapa é opcional, crie o diretório **C:\dev\tools** para extrair o eclipse.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

A instalação dura em média de 3 a 5 minutos

Um ícone deve ser criado na sua área de tabalho, caso contrário basta acessar a pasta onde extraiu (**C:\dev\tools\eclipse)** o eclipse e executar.

Uma imagem contendo Logotipo

Descrição gerada automaticamente

* **Workspace**: O eclipse costuma organizar nossos projetos por área de desenvolvimento denominados de Worspaces. Você também pode organizar seus projetos por finalidade exemplo: estudo, freela, projetos e etc.

No exemplo abaixo foi criando uma workspace para estudos em **C:\dev\ws\estudos**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Bem-vindo ao **Eclipse IDE**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Na criação de projeto use a jre do java, escolha “Use default JRE”, caso apareça outra JRE clique em Configure JRE e escolha a sua versão

Nas propriedades mude a versao de compilação

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# Postgre SQL

**PostgreSQL** é um sistema de banco de dados poderoso e em código aberto para Windows, Mac e Linux. Ele consiste em um processo de servidor que lê e grava os arquivos de banco de dados reais, e um conjunto de programas cliente que se comunicam com o servidor. O mais comumente utilizado é o comando **psql**, que permite ao usuário executar consultas SQL e visualizar os seus resultados.

Link para instalação [PostgreSQL Database](https://www.enterprisedb.com/downloads/postgres-postgresql-downloads).

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente com confiança média

​<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>​

​

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

O processo de instalação é bem simples, sempre ***Next, Next e Next.***

Nesta etapa iremos instalar 3 componentes do PostgreSQL

* O PostgreSQL Database Server
* PgAdmin para interagir sobre o banco de dados
* O Commando Line Tools para executar instrução via terminal

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Hora de definir o usuário e senha, por recomendação defina ambos como **postgres**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Após a instalação concluída, iremos iniciar o Postgres através do **PgAdmin** localizando no menu iniciar.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

## GitHub desktop

Link para download do [GitHub Desktop](https://desktop.github.com/) - <https://desktop.github.com/>​

**Desenvolvimento Mobile**

O ambiente de desenvolvimento é configurado com Android Studio, possibilitando a construção de aplicativos com Kotlin e Java.

**Android Nativo: Ambiente de Desenvolvimento**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Setup: Elementos Centrais**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Baixar e Instalar o Android Studio**

developer.android.com

\*Na instalação, caso esteja usando um aprelho físico, pode desmarcar o android virtual devices

**Criando um projeto**

Em Minimum SDK refere-se a qual plataforma mínima, clicando em “Help me choose” mostra a porcentagem de usuários das plataformas

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

\*Virtural Device Configuration: cria um dispositivo virtual

Estruturas de projetos Android

Gráfico

Descrição gerada automaticamente

\*Intercionalização: poder do app ser executado em vairias línguas.

**Configurando um aparelho físico.**

* Em “Informações de Software” clique varias vezes em numero da compilação para ativar o “Modo Desenvolvedor”
* No menu entre em “Modo Desenvolvedor”
* Clique em “permanecer ativo”
* Ative a “Depuração USB”
* Conecte o USB
* O Android Studio devera reconhecer o celular

Interface gráfica do usuário, Texto

Descrição gerada automaticamente