**1.ORIENTAÇÃO A OBJETOS**

Inovações tecnológicas surgidas na área de Informática têm criado uma necessidade de utilização e manipulação de informações que antigamente não eram utilizadas. Os tipos de dados complexos, como os objetos, passaram a ser manipulados através das linguagens de programação, que passaram a receber a conotação de Linguagem de Programação Orientada a Objetos.

A programação estruturada a objeto, em de tratado principalmente na manutenção de sistemas possuindo assim taxas de reutilização.

O objetivo principal da orientação a objetos e modelar o mundo real assim possibilitando uma maior taxa de manutenção, e isso se torna possível pois se utilizando uma linguagem de programação orientada a objetos consegue-se obter-se um desenvolvimento mais rápido, vendo-se que o desenvolvimento ocorre em blocos de códigos correspondentes aos objetos referentes ao seu acoplamento. A reutilização também se torna garantida de uma maneabilidade melhor do programa pois os testes referentes aos componentes já foram previamente executados.

O surgimento da programação a objeto se tem pela década de 60, baseada na Teoria dos Tipos da Álgebra mais apenas na década de 90 ela começou a ser amplamente utilizada computacionalmente. Seu principal fundamento desdém de representar o mundo real e da forma de interagir com os objetos.

O objeto algo que se possibilita a ser tocado e perceber, mas mesmos que tocamos um objeto ele e diferente dos outros que estão em uso de outro. É esta a característica principal dos objetos, eles são únicos e somente há um único objeto no mundo.

Agora vamos pensar no mundo dos smartphone. Pensamos em uma fábrica de celulares. Cada celular que sai dessa fábrica é único, e possui uma identificação incorporada ao próprio aparelho que é denominado ESN (Eletronic Serial Number), ou seja, Número Serial Eletrônico. Sendo assim cada smartphone é único, mesmo que tenham o modelo em comum e a cor, da mesma marca, mesma categoria possuindo assim cada um a suas próprias características. E assim são os objetos.

As classes são as montadoras que, é quem detém a “receita” de como montar, quem contém os moldes dos celulares, é uma abstração dos celulares, na verdade a classe contém os moldes de celulares e não p celular propriamente dito. No caso de orientação a objetos, as classes são as montadoras dos objetos, as classes são vistas como agrupamento de objetos que cotem a mesmas características e os mesmos comportamentos.

Os objetos ou as classes possuem características que determinam quem eles realmente são, pensemos em um bolo de chocolate, onde a receita possui características que a determina, que são os ingredientes. Abstraímos um pouco este bolo, e imaginemos a receita como sendo a Classe, e o bolo pronto como sendo o objeto. Os ingredientes são parte da receita, assim são as características de uma receita e poderemos denominar eles de atributos da Classe Receita. E o modo de preparo do bolo? O modo de preparo do bolo é o método que vamos utilizar para elaborar o bolo. Assim, o modo de preparo é o método. E o bolo pronto, assado, cheirando, como já dissemos anteriormente ele é o objeto, é único, pois se você fizer este mesmo bolo e seguir a mesma receita, provavelmente o seu bolo será melhor que o meu, pois ficará mais solto na assadeira, tem uma cor melhor, mais amarela, gosto saboroso. E estas características do bolo pertencem somente ao meu ou ao seu bolo, assim são atributos do objeto, também denominado de atributos de instância (CAMERA, 2002)

Através desses exemplos podemos observar os fundamentos da orientação a objeto. Os objetos podem se comunicar entre si tendo assim uma troca de mensagem. A troca de mensagem pode ser abstraída como a aplicação de métodos em determinados objetos. Por exemplo, quando um objeto A deseja que o objeto B execute um dos seus métodos, o objeto A envia uma mensagem ao objeto B, como segue na Figura 1.

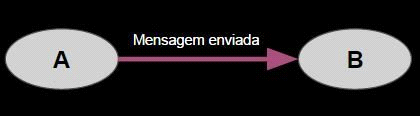


Figura 1. Troca de mensagens entre os objetos.

Fonte: <https://lh3.googleusercontent.com/proxy/76qpztcZl_hnJYBEcXSQmbOVtYpItVIsN54HrVxPiP7iyKskYmCZxDb955CrPAljUI_0XnggVPliiVrJqiDz348XN-4AnZx288cOhM7nT2SLffIUhs5Cz26vf9I17UxExC67hBo0vIvASEziSRuwbLNMV0ufGnvw3QsNgn1LoeJ06xdNpMElPqF1UA>

Outro conceito da programação orientada a objeto é o encapsulamento de informações, através do encapsulamento se pode ocultar informações irrelevantes para os outros objetos que interagem com este objeto. O encapsulamento permite que os atributos de uma determinada classe somente sejam modificados utilizando métodos que interajam com o mesmo. Assim, as modificações através dos métodos garantem que não há manipulações indevidas aos atributos. Sempre que for possível garantir que os atributos sejam encapsulados e a manipulação deles ocorra através de métodos, o faça.

A herança é mais uma característica do mundo real incorporado no mundo orientado a objetos. É um mecanismo que define variáveis e métodos comuns a todos os objetos de um certo tipo. No mundo real há os nossos pais, nossos avós e nossos filhos e netos. Onde alguma pessoa sempre lhe diz que você se parece mais com o seu pai ou com a sua mãe, e seus filhos e netos sempre herdam algum comportamento do pai ou da mãe ou ainda dos avós paternos ou maternos...enfim, a herança é “uma transmissão dos caracteres físicos ou morais aos descendentes” (FERREIRA, 1993)

Uma classe também pode ter subclasses e superclasses. A subclasses não são limitadas aos estados e comportamentos definidos pela superclasse, elas podem adicionar variáveis e métodos ao seu escopo. Entre estas características da herança, a manipulação das classes que herdam métodos e atributos garantido assim uma coesão das informações.

Outro conceito adicional na orientação a objeto utilizada no Java é o polimorfismo, onde pode-se definir duas abordagens como a sobreposição(override) e sobrecarga(overload). A sobreposição ocorre quando duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que possuem a mesma identificação(assinatura). Enquanto que a sobrecarga dos métodos ocorre quando em uma mesma classe de métodos possuem os mesmos nomes, mas assinaturas diferentes, um exemplo disso são os métodos construtores do Java, que serão abordados mais adiante neste livro.

Podemos criar uma estrutura mais próxima da manipulação de classes e objetos dentro da linguagem Java, como podemos te uma classe ANIMAL que possui atributos(características) tais como: Tipo, Raça, Idade, Sexo. Esta classe ANIMAL também possui métodos (ações, comportamentos) tais como: Andar, Correr, Morder, Comer.

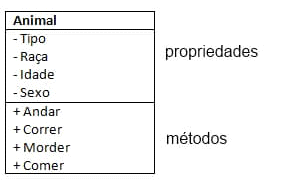


Figura 2. Classe ANIMAL

Fonte: <https://www.luis.blog.br/userfiles/image/classe-propriedades-e-metodos.jpg>

**2. INTRODUÇÃO A JAVA**

O Java foi desenvolvido como linguagem de programação orientada a objetos pela empresa Sun Microsystems, desenvolvida para criar tanto aplicativos como programas para linguagem desktop, para aplicações comerciais e softwares robustos completos e independentes. Além disso, caracteriza-se por se assemelhar com a linguagem de programação C++, eliminando as características consideradas complexas, das quais ponteiros e herança múltipla.

**2.1 Histórico**

Em 1991, um pequeno grupo de funcionários daSun Microsystems incluindo James Gosling mudou-se para a San Hill Road, uma empresa filial. O grupo estava iniciando um projeto denominado Green Project, que consistia na criação de tecnologias modernas de software para empresas eletrônicas de consumo, como dispositivos de controle remoto das TV a cabo. O grupo percebeu que não poderia ficar preso apenas as plataformas, pois os clientes não estavam interessados no tipo de processador que estavam utilizando e fazer uma versão do projeto para cada tipo de sistema seria inviável. Desenvolveram o sistema operacional GreenOS, com a linguagem de programação Oak. Basearam-se no inventor do Pascal, através da linguagem USCD Pascal.



Figura 3. The Green Team. Da esquerda para a direita, Al Frazier, Joe Palrang, Mike Sheridan, Ed Frank, Don Jackson, Faye Baxter, Patrick Naughton, Chris Warth, James Gosling, Bob Weisblatt, David Lavallee e Jon Payne. Ausentes: Cindy Long, Chuck Clanton, Sheueling Chang e Craig Forrest.

Fonte: <https://duke.dev.java.net/green/GreenBBQ.jpg>.

Em 1993, surgiu a possibilidade para o grupo Green, agora incorporado como FirstPerson a Time-Warner, uma empresa que estava solicitando propostas de sistemas operacionais de decodificadores e tecnologias de vídeo sob demanda. Na mesma época em que o NCSA lançou o MOSAIC 1.0, o primeiro navegador gráfico para Web. A FirstPerson apostou nos testes de TV da Time-Warner, mas esta empresa preferiu a tecnologia disponibilizada pela Silicon Graphics.

Logo após o fracasso, a FirstPerson se dissolveu-se e metade do pessoal foi trabalhar para a Sun Interactive como servidores digitais de vídeo. Mais, entretanto o restante da equipe continuou os projetos de trabalho da Sun. Apostando na Web, observando que todos os projetos estavam voltados para WWW, então uma ideia surgiu desenvolver um browser com independência de plataforma, que foi o HotJava.

A equipe de desenvolvimento ingeria muito café enquanto trabalhavam, várias xicaras de café foram consumidas ate que o projeto estivesse pronto. Finalmente na data de maio de 1995 , a Sun anunciou um ambiente denominado Java (homenagem às xícaras de café) que obteve sucesso graças a incorporação deste ambiente aos navegadores (browsers) populares como o Netscape Navigator e padrões tridimensionais como o VRML (Virtual Reality Modeling Language – Linguagem de Modelagem para Realidade Virtual).



Figura4. Ícone Java

Fonte: <https://s2.glbimg.com/eSjj-63oF-0MxDTOmH-SRR-D_DQ=/695x393/s2.glbimg.com/xHftWyMcmpeuwcf7uNsi6WDrdnY=/695x0/s.glbimg.com/po/tt2/f/original/2014/07/15/java-logo.jpg>

Em 2008 a Oracle Corporation adquire a empresa responsável pela linguagem Java, a Sun Microsystems, por US$ 7,4 bilhões, com o objetivo de levar o Java e outros produtos da Sun ao dispor dos consumidores.

**2.2 Web x Aplicativos**

Programas escritos em Java, podem ser Applets, Aplicativos ou ainda Servlets. Os aplicativos são programas que necessitam de um interpretador instalado na máquina. Enquanto que Applets são programas carregados juntamente com páginas HTML. O interpretador, no caso das Applets, é o próprio browser. Não necessita instalação, basta que o browser usado ofereça suporte a Java. Já no caso dos Servlets, são programas desenvolvidos em Java que são interpretados pelo Servidor Web. Os servlets são utilizados na geração dinâmica de páginas HTML. Atualmente, são muito utilizados na combinação com JSP (Java Server Pages) para a utilização do MVC (Model View Controller).

**2.3 Java Developement Kit - JDK**

O JDK é um kit de desenvolvimento Java fornecido livremente pela Sun. Constitui de um conjunto de programas que engloba compilador, interpretador e utilitários. A primeira versão deste Kit foi a 1.0. Atualmente, o JDK está na versão 1.4.x, tendo sido lançado recentemente a versão beta 1.5.x, também denominada Tiger. O JDK é separado em 3 edições: o Java 2 Standard Edition (J2SDK), o Java 2 Enterpise Edition (J2EE) e o Java 2 Micro Edition (J2ME). Cada uma engloba um conjunto de pacotes diferentes fornecendo aos usuarios uma forma organizada e diferenciada para desenvolver aplicações. Ou seja, os usuários que desejem desenvolver aplicações para Palm Tops, celulares, dispositivos pequenos, deve utilizar o J2ME para desenvolver as suas aplicações.

Os principais componentes do kit de desenvolvimento são:

javac ( compilador )

java ( interpretador )

appletviewer ( visualizador de applets )

javadoc ( gerador de documentação )

jar ( programa de compactação )

**3. NetBeans**

O NetBeans é um ambiente de desenvolvimento integrado (IDE) para a linguagem Java desenvolvida pela empresa Sun Microsystems.

Ele se trata de um ambiente de desenvolvimento integrado gratuito e de código aberto para os desenvolvedores. A IDE pode ser executada em diversos tipos de sistemas operacionais, como o Windows, Linux, Solaris e MacOS possui as ferramentas necessárias para criar aplicativos para desktops profissionais, empresariais, Web e móveis multiplataformas.

**2.1 Histórico**

O planejamento dessa IDE foi em 1996 por dois estudantes tchecos na Universidade de Charles, na Praga no período em que a linguagem de programação Java ainda não era popular. O primeiro nome para o projeto era incialmente Xelfi, em alusão ao Delphi pois o planejamento era que ela funcionasse semelhante aos IDE que naquela época se tornava mais atrativo pelas suas ferramentas visuais e por serem mais fáceis de se utilizarem.

1999 o projeto avia evoluído para uma IDE proprietário, que possuía o nome NetBeans DeveloperX2 a ideia do nome veio da reutilização dos componentes que era a base do Java. Nesta época a empresa Sun Microsystems desistiu do seu projeto de IDE Java Workshop e procuravam novas iniciativas de projeto e implementaram NetBeans DeveloperX2 a sua linha de softwares.

Durante alguns meses a Sun mudou o nome do projeto para Forte for Java e o manteu por um longo tempo como software proprietário, mas em julho de 2000 a Sum disponibilizou o código fonte do IDE NetBeans tornando-a uma plataforma OpenSource.

**2.2 Aplicações**

O NetBeans possui suporte suficiente para trabalhar em aplicações mais complexas fazendo utilização de janelas.

Aplicação no design view isso e possível graças ao editor WYSlWYG (what you see is what you get) conhecido como Design View um modo fácil de construir interfaces, no NetBeans.

Escolhendo a opção JFrame Form você recebera assistência comum pequena janela nela haverá uma paleta com alguns elementos já configurados logo de início.

No início da linguagem Java a uma biblioteca nomeada como AWT (Abtract Window Toolkit —caixa de ferramentas de janelas abstratas) para programação GUI avançada, fazendo assim deste o princípio parte do JDK (Java Development Kit). Esta biblioteca é utilizada em aplicações mais simples, mais infelizmente não atendia os requisitos para a criação de uma interface mais complexa e rica com elementos gráficos.

O Swing e o kit de ferramentas GUI que a Sun Microsystems desenvolveu para habilitar o empreendimento de se desenvolver em Java. O Swing pode ser usado para desenvolver amplas aplicações de Java com uma ampla carga de componentes poderosos que podem modificar estes componentes para controlar seu aparecimento e comportamento.

Possuímos diversos itens para a criação de diversos tipos de menu como o jPopupMenul para a criação de menu flutuantes e para criação de itens de menu o jMenuItem criando assim também caixas de seleção também é possível fazer agrupação de itens do menu. Os itens do menu são muito semelhantes aos botões. Na verdade, a classe estende a classe AbstractButton. Portanto assim como nos botões, os menus tem a possibilidade de ter ícones e rótulos.

Uma caixa de combinação fornece uma lista de itens entre os quais o usuário pode escolher. A classe JComboBox de Swing pode ser usada para criar esse componente de interface com o usuário. Você terá um campo que representará os Estados, em UF, que ao invés de ser uma caixa de texto comum, será uma caixa de combinação, podendo você assim, selecionar ao invés de digitar.

Os jLabel são rótulos que fornecem instruções ou informações de texto sobre uma GUI ela pode conter ícones ou não. Ela é criada a parti da classe jLabel um rotulo que é usado diversas vezes para identificar o objetivo de outro determinado componente.

O componente jTextField é um campo de texto em branco que permite ser alterado pelo usuário. É representado pela classe jTextField o qual pode ser manipular uma linha de entrada como um capo de texto.

Um botão não é algo novo NetBeans ele é chamado de jButton. É um componente usado pelo usuário para clicar e acionar uma ação especifica. Um programa Java pode trabalhar com vários tipos de botão.