



UFC

UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BEATRIZ FERNANDES BEZERRA

LUCAS PINHEIRO DA COSTA

LUIZA ESTHER MARTINS PESSOA

MARIA LUIZA FELIPE CAROLINO

**RELATÓRIO: PRIMEIRA AVALIAÇÃO DE TÉCNICAS DE PROGRAMAÇÃO -
GERAÇÃO DE DISTRIBUIÇÕES ESTATÍSTICAS EM JAVA**

Fortaleza

2024

Introdução

O presente relatório descreve o desenvolvimento de um projeto em Java de geração de distribuições estatísticas, realizado no âmbito da disciplina de técnicas de programação pelos discentes Beatriz Fernandes Bezerra, Lucas Pinheiro da Costa, Luiza Esther Martins Pessoa e Maria Luiza Felipe Carolino.

O repositório, hospedado no GitHub [neste link](#), inclui os seguintes componentes principais:

- `src/`: Contém o código-fonte.
- `.gitignore`: Exclui arquivos desnecessários do controle de versão.
- `README.md`: Explica o projeto, suas funcionalidades e como usá-lo

Organização da Equipe

A equipe foi estruturada para garantir uma distribuição eficiente das tarefas. Lucas Pinheiro desenvolveu a classe Normal, enquanto Luiza Esther Martins Pessoa cuidou da classe Poisson. Maria Luiza Felipe Carolino implementou a classe Exponencial e gerenciou as exceções do projeto. Beatriz Fernandes foi responsável pela implementação da classe Uniforme e pela organização da estrutura de pastas do projeto, seguindo os padrões para projetos Java.

Além das implementações, Luiza Esther elaborou a documentação técnica, detalhando as funcionalidades e o uso das classes. Beatriz e Luiza Esther também compilaram o relatório final e produziram o vídeo demonstrativo. Lucas adicionou a pasta 'target' com os executáveis, facilitando a execução do software.

O trabalho foi conduzido de forma colaborativa, com revisões de código realizadas no GitHub para assegurar a consistência e a correção dos códigos implementados. Todos os membros da equipe se dedicaram a pesquisar as fórmulas e tecnologias empregadas, garantindo uma base sólida para o desenvolvimento do projeto.

Desenvolvimento

Como requisitos técnicos obrigatórios presentes nas especificações da avaliação, o projeto deve abordar conceitos de Programação Orientada a Objetos, como Classe, Herança, Polimorfismo e o uso de Generics. Portanto, nesta seção, vamos abordar como realizamos essas implementações e outros recursos técnicos não obrigatórios mas que foram adicionados a fim de tornar o código mais coeso. Implementamos o conceito de Classe e Herança ao criar os seguintes arquivos presentes no [repositório](#):

- `Distribuicao.java`: Classe abstrata que serve como base para outras classes que representam diferentes distribuições estatísticas. Ela contém quatro métodos abstratos:
 - `gerarMedia()`: Calcula e retorna a média dos valores da distribuição.
 - `gerarVariancia()`: Calcula e retorna a variância dos valores da distribuição.
 - `gerarDesvioPadrao()`: Calcula e retorna o desvio padrão dos valores da distribuição.
 - `gerarValores(int n)`: Gera e retorna uma lista de valores da distribuição com base no parâmetro `n`, que especifica a quantidade de valores a serem gerados.
 - `getValores()`: Retornar a lista de valores que a distribuição usa internamente, para geração de interface.
- `DistribuicaoNomeDaDistribuicao.java`: Classes concretas que herdam os métodos de `Distribuicao.java`.

Usamos Polimorfismo e Generics ao implementar os métodos abstratos em classes concretas, como, por exemplo: A classe `DistribuicaoNormal.java` é uma subclasse de `Distribuicao.java` e fornece implementações concretas para todos os métodos abstratos definidos na classe base. Este é um exemplo de polimorfismo em tempo de execução (ou polimorfismo dinâmico), onde a subclasse pode fornecer uma implementação específica para os métodos que são definidos na superclasse.

Resumidamente, o polimorfismo no código permite que métodos da classe `Distribuicao` sejam chamados de maneira uniforme, mesmo que as implementações concretas sejam fornecidas por subclasses específicas, promovendo flexibilidade e extensibilidade no código, permitindo que diferentes distribuições sejam tratadas de maneira consistente, mas com comportamentos específicos conforme implementado nas subclasses.

A classe `estatisticas.java` usa generics para ser aplicável a qualquer tipo de distribuição que herde da classe abstrata. Isso proporciona flexibilidade e segurança de tipo, permitindo que a classe trabalhe com diversas distribuições sem precisar alterar seu código interno.

Módulos Utilizados:

- **Array List:** Usado para armazenar os valores que são gerados para calcular média, variância, desvio padrão e os determinados pontos dos gráficos.
- **Random:** Usado para gerar valores aleatórios a partir de um intervalo pré-estabelecido.
- **JFreeChart e Swing:** JFreeChart utilizado para gerar os gráficos e o Swing foi utilizado para criar a janela de visualização dos dados

Output e Demonstração

A saída do programa imprime, obrigatoriamente, cada valor de cada distribuição em uma lista de números separados por uma vírgula. Além disso, como recurso adicional, a equipe optou por adicionar um trecho na `Main.java` que é responsável por gerar e retornar uma lista de determinados valores, criando gráficos que representam o comportamento de cada distribuição estatística.

→ Acesse aqui o link do youtube para assistir a demonstração da aplicação