**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA**

**PAULA SOUZA**

**ETEC PROF. BASILIDES DE GODOY**

**M-TEC Desenvolvimento de Sistemas**

**João Paulo Amaral de Araujo**

**João Vitor dos Santos Nunes**

**Regular Expressions e Data Annotations**

**São Paulo**

**2025**

**João Paulo Amaral de Araujo**

**João Vitor dos Santos Nunes**

**Regular Expressions e Data Annotations**

Trabalho solicitado pela Etec Prof. Basilides de Godoy no 2ºM-tec Desenvolvimento de Sistemas como requisito parcial para aprovação no componente curricular de Desenvolvimento de Sistemas

Prof. Orientador: Nivia Maria Domingues

**São Paulo**

**2025**

*Dedico este trabalho à minha família, pelo apoio constante em minha jornada.*

**AGRADECIMENTOS**

A todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

*"O verdadeiro conhecimento está em sabermos que nada sabemos."*

*Sócrates*

**RESUMO**

Este trabalho tem como objetivo apresentar de forma clara os conceitos de Regular Expressions (expressões regulares) e Data Annotations (anotações de dados), abordando suas aplicações, principais formatos utilizados, linguagens de programação compatíveis, diferenças entre frameworks e possíveis implicações no uso com bancos de dados. Inicialmente, são discutidas as expressões regulares, seguidas pelas anotações de dados, conforme a organização exigida. A pesquisa visa esclarecer as funcionalidades dessas ferramentas no desenvolvimento de software.

Palavras-chave: Regular Expression, Data Annotations, Frameworks.

**ABSTRACT**

This paper aims to clearly present the concepts of Regular Expressions and Data Annotations, addressing their applications, commonly used formats, supported programming languages, differences among frameworks, and possible implications when used with databases. Regular Expressions are discussed first, followed by Data Annotations, according to the required structure. The research seeks to clarify the functionalities of these tools in software development.

Keywords: Regular Expressions, Data Annotations, Frameworks.

**SUMÁRIO**

**1 INTRODUÇÃO....................................................................................................7**

**2 REGULAR EXPRESSION E DATA ANNOTATION...........................................8**

**2.1 O que é Regular Expression?........................................................................8**

**2.1.2 Como é utilizado?.......................................................................................9**

**2.1.2 Os mais utilizados.......................................................................................9**

**2.2 O que é Data annotation................................................................................9**

**2.2.1 Como é utilizado?......................................................................................10**

**2.2.2 Os mais utilizados......................................................................................10**

**2.3 Diferença entre ambos.................................................................................10**

**2.3.1 E framework?..............................................................................................xx**

**2.4 Quais linguagens os utilizam?....................................................................11**

**2.5 Aplicação no banco de dados......................................................................xx**

**3 CONCLUSÃO....................................................................................................12**

**REFERÊNCIAS....................................................................................................13**

**1 INTRODUÇÃO**

O desenvolvimento de sistemas modernos exige mecanismos robustos para garantir a integridade, segurança e organização dos dados. Entre as ferramentas mais relevantes nesse contexto, destacam-se as Regular Expressions e as Data Annotations. As Regular Expressions oferecem uma forma poderosa e flexível de identificar e manipular padrões em cadeias de texto, sendo muito utilizadas em linguagens como JavaScript, Python e C#. Já as Data Annotations, permitem a configuração direta de regras e validações sobre classes e propriedades, especialmente em ambientes .NET. Mais pra frente, exploraremos mais a fundo essas duas abordagens, analisando suas estruturas, formas de utilização, principais aplicações, diferenças entre linguagens e frameworks, além de analisar sua utilização com bancos de dados.

**2 REGULAR EXPRESSION E DATA ANNOTATION**

**2.1 O que é Regular Expression?**

**2.1 Como é utilizado?**

**2.2 Tipos de Framework**

**2.2.1 Front-end**

Angular: Desenvolvido inicialmente por Miško Hevery e Adam Abrons, adquirido pela Google em 2009. Consiste em um framework de código aberto (open source) que teve sua primeira versão disponibilizada ao público em 2012, mantendo-o atualizado e implementando novas funcionalidades a cada atualização.

Bootstrap: Inicialmente desenvolvido por Mark Otto e Jacob Thornton no Twitter, O Bootstrap é um framework de código aberto (open source) que teve sua primeira versão disponibilizada ao público em 19 de agosto de 2011. Desde então, tem sido continuamente atualizado e novas funcionalidades são implementadas a cada versão.

**2.2.2 Back-end**

Django: Desenvolvido inicialmente por Adrian Holovaty e Simon Willison no Lawrence Journal-World, foi lançado como um projeto de código aberto (open source) em junho de 2007. A Django Software Foundation (DSF) foi formada posteriormente para promover, apoiar e avançar o framework, mantendo-o atualizado com novas funcionalidades e melhorias a cada lançamento.

ASP.NET Core: É um framework back-end de código aberto desenvolvido pela Microsoft, amplamente utilizado para a construção de aplicações web modernas. Ele é uma evolução do ASP.NET tradicional, reescrito com foco em alto desempenho, modularidade e portabilidade. Uma das principais características do ASP.NET Core é sua capacidade multiplataforma, o que significa que aplicações desenvolvidas com ele podem ser executadas em diferentes sistemas operacionais, como Windows, Linux e macOS. Esse framework utiliza principalmente a linguagem C#, mas também oferece suporte a outras linguagens. Seu motor de execução é baseado no .NET Core, que permite uma performance otimizada e leveza na execução das aplicações. Além disso, o ASP.NET Core oferece uma estrutura altamente configurável, que facilita a integração com bancos de dados, autenticação, autorização e segurança da aplicação.

**2.2.3 Mobile**

React Native: Desenvolvido pelo Facebook (agora Meta), é um framework de código aberto (open source) para a construção de aplicações móveis utilizando JavaScript e React. Foi lançado inicialmente em março de 2015. O React Native permite que desenvolvedores criem aplicativos para plataformas iOS e Android a partir de uma única base de código, e continua a ser mantido e atualizado pela Meta e pela comunidade, com novas funcionalidades e melhorias sendo introduzidas regularmente.

Flutter: Desenvolvido pelo Google, é um framework de código aberto (open source) para a criação de interfaces de usuário compiladas nativamente para aplicações mobile, web e desktop a partir de um único código-fonte. Sua primeira versão alfa foi lançada em maio de 2017. Desde então, o Flutter tem passado por atualizações constantes, com a adição de novos recursos e melhorias significativas em cada lançamento, impulsionando seu crescimento e adoção pela comunidade de desenvolvedores.

**2.2.4 Outros**

Embora o Visual Studio não seja um framework, ele é uma ferramenta essencial para o desenvolvimento de aplicações com frameworks como o ASP.NET. Visual Studio é uma IDE (Ambiente de Desenvolvimento Integrado) desenvolvida pela Microsoft, muito utilizada no desenvolvimento de aplicações com as tecnologias .NET.

**2.3 Linguagem de programação, Biblioteca e Framework**

Enquanto a linguagem de programação é a forma como nos comunicamos com o computador para resolver problemas e desenvolver soluções, ela pode ser vista como o conjunto de instruções básicas disponíveis para construir qualquer aplicação, desde programas simples até sistemas complexos. É como se fossem as peças mais fundamentais de LEGO — blocos pequenos, porém extremamente versáteis, com os quais é possível montar praticamente qualquer coisa, desde que se tenha paciência, conhecimento e criatividade suficientes.

As bibliotecas, por sua vez, representam um passo adiante em praticidade e produtividade. Elas são um conjunto de funções, classes e recursos já prontos, criados para realizar tarefas específicas, como manipulação de arquivos, comunicação com a internet, geração de gráficos, entre muitas outras. Utilizar uma biblioteca significa reaproveitar código que já foi testado e validado por outros desenvolvedores, economizando tempo e esforço. Seguindo a analogia do LEGO, as bibliotecas seriam como conjuntos de peças pré-montadas ou módulos com formatos úteis e otimizados, como rodas, portas ou motores, que você pode simplesmente encaixar no seu projeto conforme a necessidade.

Já um framework é uma estrutura mais completa e abrangente, que oferece não apenas ferramentas prontas como as bibliotecas, mas também uma forma padronizada de organizar e desenvolver o seu projeto. Ele fornece um esqueleto de aplicação, definindo fluxos de trabalho, estrutura de arquivos, e até mesmo como certas partes do sistema devem interagir. Usar um framework é como receber uma plataforma ou base montada com espaços definidos para encaixar as peças do LEGO — você ainda tem liberdade para personalizar e construir, mas dentro de uma estrutura que guia e acelera o desenvolvimento.

**3 CONCLUSÃO**

Frameworks são ferramentas que facilitam o desenvolvimento de sistemas, oferecendo uma estrutura pronta que ajuda a organizar o código e acelerar o trabalho. Diferente das linguagens de programação, que são a base para criar qualquer programa, e das bibliotecas, que são conjuntos de funções específicas para serem usadas quando necessário, os frameworks impõem uma estrutura e um fluxo que o desenvolvedor deve seguir.

Eles são recomendados especialmente para projetos maiores ou que precisam ser entregues rapidamente, pois trazem várias funcionalidades prontas e boas práticas de desenvolvimento. Já em projetos pequenos ou simples, pode ser melhor usar apenas bibliotecas ou a própria linguagem.

Na hora de escolher um framework, é importante considerar a linguagem do projeto, o conhecimento da equipe, a comunidade de suporte e se ele atende às necessidades específicas, como desenvolvimento web, mobile ou outras áreas.

REFERÊNCIAS

**GLOSSÁRIO**