

JUSTIFICAR O PYTHON E AS FERRAMENTAS (FRAMEWORK) PARA O DESENVOLVIMENTO RAD

CONTEXTUALIZAÇÃO

Uma das principais características da RAD é o desenvolvimento de protótipos, de modo que desenvolvedores e usuários possam interagir em um sistema funcional:

Dessa forma, os usuários podem usar o software, ainda que tenha limitações, e fazer comentários e sugestões.

Os comentários auxiliam os desenvolvedores a tomar decisões que maximizem a satisfação dos usuários.

E com a validação do protótipo, os desenvolvedores montam um plano de trabalho para atender aos requisitos definidos pelos usuários.

Nesse sentido, fazer uso de ferramentas de desenvolvimento rápido é fundamental para que haja boa tradução entre ideias e representações visuais e funcionais.

A utilização de ferramentas que acelerem o desenvolvimento auxilia na redução do tempo de lançamento no mercado. No ambiente competitivo do mercado de software, trata-se de uma característica muito importante. Frameworks facilitam a codificação e padronizam o uso de recursos; desse modo, a “reusabilidade” de componentes é um dos benefícios que são obtidos e que podem ajudar no desenvolvimento de novos módulos do sistema.

A seguir, serão apresentadas características da linguagem Python, porém já pode ser adiantado que a sintaxe dela é bem mais simples do que muitas outras linguagens do mercado, como o Java, por exemplo. Portanto, escolher o Python como linguagem de desenvolvimento de um projeto pode reduzir consideravelmente o tempo de produção. A ideia é simples: supondo que o número de linhas que um desenvolvedor pode produzir em determinado intervalo é limitado, escolher uma linguagem de sintaxe mais simples ajuda a aumentar a produtividade.



Que outra razão você apresenta para a escolha da linguagem Python?

PYTHON

Python é uma linguagem de programação muito usada para diversos fins. Ela possui muitos pacotes para diferentes tipos de aplicações, bibliotecas de GUI, além de ser uma linguagem de programação “enxuta”, no sentido de que sua sintaxe é mais simples do que muitas outras linguagens.

Dadas essas características, entre outras que serão tratadas um pouco mais adiante neste texto, o Python é uma linguagem adequada para a aplicação da metodologia RAD.

Outro ponto que fortalece a escolha do Python para projetos RAD é que muitos frameworks são baseados nela com o objetivo de acelerar o desenvolvimento e garantir desempenho e extensibilidade do código.

Linguagem Python É uma linguagem de tipo dinâmico, interpretada e de alto nível. Nela, é possível desenvolver programas estruturados, orientados a objetos e, até mesmo, aplicar conceitos de linguagens funcionais. Possui uma sintaxe que a diferencia de outras linguagens de programação, como Java, C++ e JavaScript.

Para fazer o desenvolvimento de um sistema em Python, é necessário utilizar um ambiente de desenvolvimento, e um dos mais usados é o Spyder (Spyder, 2020), mas existem muitos outros.

Saiba mais

A instalação dos pacotes no Python também é bem simples, de modo geral. Normalmente, basta digitar: “pip install nome-do-pacote”. Em algumas situações, pode-se ter um pouco mais de trabalho por causa da compatibilidade de versões. Essa questão de configurações de versões, inclusive, é um problema que abrange os projetos de desenvolvimento de modo geral e, sem o apoio de uma equipe focada em questões que incluam esse tipo de problema, perde-se bastante tempo no projeto.

Entre os motivos para escolher o Python em projetos RAD, estão:

Portabilidade

Pode-se trabalhar com Python em diversos sistemas operacionais.

Licença de código aberto

As distribuições do Python são gratuitas e de código aberto sem restrições do uso da licença.

Integração com outros sistemas

Já existem muitos pacotes desenvolvidos para Python que abrangem aplicações de bancos de dados, Web, interfaces gráficas, ciências de dados, entre muitos outros.

Desenvolvimento rápido

Por possuir uma sintaxe mais simples do que muitas outras linguagens de programação e, além disso, ter diversos pacotes e frameworks, o Python é uma linguagem muito apropriada para projetos RAD, em que a velocidade do desenvolvimento é considerada um fator fundamental.

Otimização de desempenho

O Python possui tratamentos específicos para manipulação de listas e de dados de grande porte que otimizam o desempenho do sistema, inclusive para aplicações de big data.

Linguagem interpretada

Torna mais simples fazer o uso interativo com o programa. No caso de desenvolvimento de protótipos, trata-se de uma característica prática.

Outra vantagem de usar o Python é que a comunidade de usuários é ampla, formada por pessoas com bastante conhecimento e prestativas. Uma das grandes preocupações de qualquer projeto é sobre o suporte que será fornecido ao longo do desenvolvimento; no caso do Python, a comunidade é ativa, ou seja, as pessoas realmente se esforçam para responder a perguntas de diversos níveis de conhecimento. O nível de suporte técnico disponível, gratuitamente, é considerável.

Saiba mais

Para obter mais detalhes sobre a linguagem Python, sugere-se visitar sua página eletrônica (Python 3.8.5 documentation, 2020).

FERRAMENTAS (FRAMEWORK) PARA O DESENVOLVIMENTO RAD

A linguagem Python em si – com seus diversos pacotes disponíveis para instalação gratuita – é uma escolha muito apropriada para desenvolver projetos RAD, mas, além disso, também possui diversos frameworks que auxiliam no ganho de velocidade do tempo de entrega, além de padronizar o desenvolvimento dos sistemas. Com uma interface gráfica obtida rapidamente, o usuário pode interagir com o desenvolvedor com mais qualidade, ou seja, facilita a comunicação entre as partes interessadas, além de manter a motivação no projeto.

Observe abaixo os tipos de frameworks suportados pelo **Python**.

Frameworks (Python)

Full-Stack:

Fornecem uma solução completa de todos os requisitos para o desenvolvedor, desde a geração e validação de formulários até a disponibilização de modelos layouts.

Microframework:

São frameworks que oferecem uma quantidade mínima de serviços. Normalmente, são usados para o recebimento de solicitações, roteamento, despacho e o retorno de uma resposta HTTP.

Framework assíncronos:

Trata-se de um tipo de microframework que permite lidar com grande conjunto de conexões simultâneas. A estrutura assíncrona é usada para gerenciar operações de longa duração, como transformações de dados.

Ao longo dos últimos anos, o Python tem atraído mais desenvolvedores devido à facilidade de aprendizado e diversidade de aplicações que abrange. A combinação de Python e RAD permite desenvolver rapidamente protótipos que abrangem diversos tipos de aplicações, além do fato de que a aplicação poderá operar em múltiplas plataformas. A seguir, serão tratados alguns exemplos de frameworks em Python para desenvolvimento de GUIs e de aplicações para Web.

FRAMEWORKS GUI PARA PYTHON

A Interface Gráfica do Usuário (GUI) é um aspecto essencial para que possa haver interação com o sistema; portanto, o desenvolvimento de recursos interativos, ainda nos protótipos, possibilita que o usuário tenha uma percepção mais clara de como o projeto está progredindo. Existem diversos frameworks para Python que tratam desses tipos de recursos, tais como botões, ícones, campos de texto, gráficos, e assim por diante.



Tkinter

É uma biblioteca que já está embutida na instalação padrão do Python que permite desenvolver interfaces gráficas (TCL DEVELOPER XCHANGE, 2020).

PyQt

É uma biblioteca de componentes gráficos do Python. Ela não é instalada com o Python, além disso tem diferentes tipos de licença (PYQT, 2020).

PySide

É um software livre que executa nos sistemas operacionais Windows, Mac e Linux (PYSIDE, 2020).

Kivy

É um framework de código aberto, em que é possível desenvolver aplicativos “multi-touch” para dispositivos móveis e computadores (KIVY, 2020).

wxPython

É um kit de ferramentas de código aberto que possui suporte para as plataformas Windows de 32 bits, Unix e Mac OS X (WXPYTHON).

Cada um desses frameworks possui particularidades que podem torná-los mais adequados para determinados projetos. Em especial, o framework Tkinter é o framework GUI mais usado do Python.

TKINTER

O framework GUI mais usado do Python, possui componentes que podem ser organizados para facilitar a interatividade com os usuários. O Tkinter possui três classes para gerenciar a organização dos componentes. São elas:

- Método pack (): organiza os componentes em blocos antes de posicioná-los no componente pai.
- Método grid (): faz a configuração dos componentes em forma de tabela.
- Método place (): o posicionamento do componente é feito em um local específico.

Um dos motivos para o Tkinter ser muito usado é que ele já faz parte da biblioteca padrão do Python e é utilizado pelo framework Web Django – que é o framework Web mais usado – para fazer páginas Web. Portanto, não há necessidade de instalá-lo, e, por ser usado pelo Django, torna mais fácil encontrar documentação e exemplos de como usá-lo. Entre os principais componentes do Tkinter, estão:

- Button: trata-se de um componente botão. Para adicioná-lo no sistema, basta escrever `w=Button(master,option=value)`.
- Canvas: é usado para aplicar design e layouts complexos. Para adicioná-lo no sistema, basta escrever `w=Canvas(master, option=value)`.
- CheckButton: é aplicado para selecionar uma opção. Para adicioná-lo no sistema, basta escrever `w=CheckButton(master, option=value)`.
- Entry: usado para entrar texto. Para adicioná-lo no sistema, basta escrever `w=Entry(master, option=value)`.
- Frame: é usado para agrupar e organizar componentes. Para adicioná-lo no sistema, basta escrever `w=Frame(master, option=value)`.
- Label: exibe uma caixa onde se pode inserir texto. Para adicioná-lo no sistema, basta escrever `w=Label(Master, option=value)`.
- MenuButton: é usado para criar “menus de topo”. Para adicioná-lo no sistema, basta escrever `w=MenuButton(Master, option=value)`.
- Menu: cria menu. Para adicioná-lo no sistema, basta escrever `w=Menu(master, option=value)`.
- Message: usado para exibir múltiplas linhas sem texto editável. Para adicioná-lo no sistema, basta escrever `w=Message(master, option=value)`.

- Scale: é um componente deslizante que permite selecionar valores de uma escala específica. Para adicioná-lo no sistema, basta escrever `w=Scale(master, option=value)`.

Todos esses componentes estão disponíveis no Tkinter.

Recomendação

Demais frameworks também possuem diversos recursos, portanto a escolha de um framework é uma decisão que deve ser feita no início do projeto.

FRAMEWORKS WEB PARA PYTHON

Um framework Web é um conjunto de pacotes que habilitam os desenvolvedores a desenvolver aplicações para Web, ou serviços, sem ter de implementar excesso de detalhes, como protocolos, soquetes ou gerenciamento de processos/threads. A maioria dos frameworks para Web focam no desenvolvimento de aplicações do lado do servidor. A responsabilidade pelas comunicações e pela infraestrutura ficam sob a responsabilidade do framework, permitindo que o desenvolvedor possa se concentrar na lógica do sistema.



Os frameworks dão suporte para diversas atividades, como interpretar solicitações – obtenção de parâmetros de formulário, gerenciamento de cookies e de sessões –, produzir respostas (apresentação de dados, como o formato JSON, por exemplo) e fazer armazenamento persistente de dados. É característico de aplicações Web que haja vários tipos diferentes de camadas de programação, geralmente empilhadas umas sobre as outras. Nesse sentido, os frameworks facilitam o rápido desenvolvimento de protótipos.

De modo geral, um framework Web é formado por um conjunto de bibliotecas e um arquivo principal, onde é feita, de fato, a programação. A maioria dos frameworks para aplicações Web incluem padrões que devem ter:

Roteamento de URL

Trata de solicitações HTTP recebidas por parte específica do código Python.

Objetos de solicitação e resposta

Agrupar as informações recebidas, ou enviadas, para o navegador de um usuário.

Template Engine

São normalmente usados como um formato intermediário escrito por desenvolvedores para produzir um ou mais formatos de saída, normalmente, HTML, XML ou PDF.

Servidor Web de Desenvolvimento

Executa um servidor HTTP em máquinas de desenvolvimento. Quando os arquivos são atualizados, recarrega automaticamente o código do lado do servidor.

SAIBA MAIS

Existem frameworks Web para Python diversos. Dentre eles, estão:

- Django: é um framework de código aberto que é muito bem-sucedido para criar sites complexos baseados em dados. Possui modelos, bibliotecas e APIs que dão suporte na criação de projetos de desenvolvimento da Web escaláveis (Django, 2020).
- TurboGears: é um framework que usa a arquitetura MVC (Model View Control), que ajuda no rápido desenvolvimento de aplicações Web (Turbogears, 2020).
- Flask: é um framework que suporta o envio de solicitações REST (Flask, 2020).
- Bottle: é um framework distribuído como um módulo de arquivo único, sem dependências além da biblioteca padrão do Python. Suporta o envio de solicitações com suporte a URL, bancos de dados de chave/valor e modelos e um servidor HTTP interno (Bottle, 2020).
- CherryPy: é um framework que fornece as funcionalidades CRUD (Criar, Recuperar, Atualizar e Excluir) (CherryPy, 2020).
- Falcon: é um framework para o desenvolvimento de programas de pequena escala; usa a arquitetura REST e tem disponível vários pacotes complementares para facilitar o desenvolvimento.

O Python possui muitos outros pacotes que servem para operações numéricas, manipulações de listas e impressão de gráficos que são usados, por exemplo, em aplicações de Ciências de Dados, como:

- Numpy: com funções para operações matemáticas e lógicas em matrizes (NumPy, 2020).
- Pandas: aplicada, especialmente, na análise de dados (Pandas, 2020).
- Matplotlib: é uma biblioteca de plotagem que auxilia na criação de gráficos e plotagens 2D, que incluem gráficos de barras, histogramas e muitos outros usando scripts Python.

A abrangência do Python é extensa com frameworks e bibliotecas que podem ser usadas para diversos tipos de aplicações.

A linguagem Python tem características que a tornam uma escolha adequada para aplicação em projetos RAD. Além de ter uma biblioteca e pacotes que cobrem diversos tipos de aplicações, também possui frameworks que são bem documentados e que facilitam a criação de protótipos.