Ferramentas de teste e desenvolvimento de software

https://lh3.googleusercontent.com/a-/AOh14GhuFlRlwc-65WpchRSTyqvEZhLcVj25_N2TiTPIjQ=s40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Texugo <lucas103barreto@gmail.com>** | | Thu, Apr 28, 11:57 AM |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | to me  https://mail.google.com/mail/u/0/images/cleardot.gif | | | |

Os testes garantem que o software passe por ajustes e manutenções enquanto ainda é possível detectar falhas. Caso os erros apareçam quando o software já estiver rodando, grande parte do trabalho terá de ser refeito, o que atrasa a plena implantação do novo sistema e pode trazer prejuízos pela suspensão de processos.

Além disso, existem ferramentas de teste de software e técnicas de automatização que podem ser utilizadas para facilitar estas etapas de verificação.

O processo de realização dos testes é otimizado com softwares voltados para otimizar a gestão dos recursos da empresa, detectar e corrigir problemas de performance e de integração com outros sistemas da empresa.

Para isto, algumas das ferramentas de teste de software que podem ser utilizadas são:

1. Appium

Esta ferramenta de teste de software é open source e pode auxiliar em praticamente todas as linguagens de programação. Também pode operar testes em aplicações nativas, híbridas ou até mesmo multiplataforma.

Outra vantagem dela é a automatização de testes simultâneos para os sistemas iOS e Android. Ainda, pode ser conectada à nuvem ou a um servidor local.

2. Robotium

Este possui testes focados em sistemas Android e tem código aberto. Possui alta performance e é voltado para testes de interface gráfica. Além de ser compatível com Gradle, Ant e Maven, auxilia na escrita de automação de testes, de delays e de timings.

3. Selenium

É a ferramenta de teste mais conhecida do mercado por ser open source. É versátil e pode ser utilizada em aplicações voltadas para web nas plataformas e browsers Linux, Mac e Windows.

O que é linguagem cliente e linguagem servidor

https://lh3.googleusercontent.com/a-/AOh14GhuFlRlwc-65WpchRSTyqvEZhLcVj25_N2TiTPIjQ=s40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Texugo <lucas103barreto@gmail.com>** | | Thu, Apr 28, 11:50 AM |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | to me  https://mail.google.com/mail/u/0/images/cleardot.gif | | | |

Existem dois tipos de linguagens para desenvolvermos para internet: as linguagens client-side (lado cliente) e as linguagens server-side (lado servidor). Estes nomes podem ser meio diferentes em outros lugares como por exemplo, front-end, se referindo às linguagens client-side e back-end se referindo às linguagens server-side , mas no contexto , são as mesmas coisas.

As linguagens server-side são linguagens que o SERVIDOR entende. Isso quer dizer que vai escrever um código onde o servidor vai processá-lo e então vai mandar para o navegador a resposta.

As linguagens client-side são linguagens onde apenas o NAVEGADOR vai entender. Quem vai processar essa linguagem não é o servidor, mas o browser.

Exemplo: se criarmos um script em linguagem back-end (PHP, Asp, Java, Ruby, etc) que apenas calcula a soma de 2 + 2, será o SERVIDOR (ou back, o server) que calculará este resultado. Se fizermos esse cálculo em alguma linguagem front-end, como o JavaScript, quem calculará essa conta é o BROWSER do usuário. Por isso o termo client ou front.

Assim, os profissionais que trabalham na interface do usuário, são chamados de Desenvolvedores front-end, e aqueles que trabalham no core da aplicação, fazendo uma programação que somente o servidor irá entender são chamados de Desenvolvedores back-end.

Exemplos de linguagens:

Essas linguagens são linguagens client-side.

HTML (xHTML, HTML4, HTML5...)

CSS (CSS2.1, CSS3...)

JavaScript

Essas são linguagens server-side, são as linguagens com que os programadores back-end trabalham:

PHP

ASP

Java

Ruby

O que é IDEs e os principais mais usados (javascript)

https://lh3.googleusercontent.com/a-/AOh14GhuFlRlwc-65WpchRSTyqvEZhLcVj25_N2TiTPIjQ=s40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Texugo <lucas103barreto@gmail.com>** | | Thu, Apr 28, 11:44 AM |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | to me  https://mail.google.com/mail/u/0/images/cleardot.gif | | | |

Um IDE (ou Ambiente de Desenvolvimento Integrado) é um software que te permite codar com muitos recursos extras além de simplesmente digitar suas linhas de código. Com IDEs, você pode editar, modificar, debugar, executar e implantar seu código. Os IDEs vêm com muitos recursos extras, como o preenchimento automático de texto, por exemplo, dando mais flexibilidade para o usuário.

Com a ajuda de IDEs, você pode aumentar a sua produtividade. Existem muitos IDEs na Internet e, felizmente, muitos deles são de código aberto.

Lista abaixo dos mais usados:

1.Eclipse

Empresa: IBM (autora original)/Eclipse Foundation

Plataformas: Windows, Linux, MacOS, Solaris

Código aberto: sim

Ano de desenvolvimento: 2001

Linguagens: inicialmente foi construído apenas para Java, mas hoje trabalha muito bem com plugins para JavaScript, C, C++, PHP, Python, Kotlin, e mais

Vantagens:

Ótimo gerenciamento de projetos (Aplication Lifecycle Management);

Quase todos os pacotes suportam integração com Git;

Destaque de sintaxe editável;

Depuração de alto nível (debugging);

Bom preenchimento automático;

Programação em várias linguagens como Java, JavaScript, PHP, C, C++, C#, Ruby, Phyton, Haskel, Cobol, e muitas outras;

Ambiente flexível por ser modular;

Capacidade de integrar JUnit;

Depuração remota (ao usar JVM);

Declaração de variáveis, classes e métodos;

Desenvolvimento de uma versão completa do projeto principal .net sem sair do IDE;

Projetos construídos com o MS test e xUnit podem ser executados diretamente no IDE.

É o IDE gratuito mais usado atualmente

Desvantagens:

Alguns iniciantes podem se assustar com a quantidade de possibilidades;

Alguns plugins nem sempre vão funcionar muito bem, então escolha os mais bem estabelecidos na comunidade;

Muitas das alterações precisam de um reboot para funcionar.

2. Visual Studio Code

Empresa: Microsoft

Plataformas: Windows, Linux, MacOS

Código aberto: sim

Ano de desenvolvimento: 2015

Linguagens: mais de 30 linguagens de programação, como Java, JavaScript, C#, C++, PHP, SQL, R, Python, TypeScript, JSON, XML etc.

Vantagens:

[ASP.NET](http://asp.net/) 5 e Node.js, além de uma ótima integração com WSL

Bom depurador

Suporte para um terminal dentro da janela

Destaque de sintaxe

Trabalha diretamente com o Github

Bom preenchimento automático (IntelliSense)

Vem praticamento zerado ao baixar, o que o torna leve

Trabalho remoto colaborativo

Controle de versões

Desvantagens:

Minimalista, pode não ser adequado para projetos muito grandes;

Para ter suporte a uma linguagem não suportada nativamente, é necessário procurar extensões na store e configurar manualmente;

As ferramentas de debugar poderiam ser melhores;

Interface pode assustar iniciantes.

3. NetBeans

Empresa: Oracle (adquirido em 2010)

Plataformas: Windows, Linux, macOS, BSD, Solaris

Código aberto: sim

Ano de desenvolvimento: 2000

Linguagens: inicialmente foi construído apenas para Java, mas hoje trabalha muito bem com JavaScript, PHP, Python, HTML5, CSS3 e mais

Vantagens:

Destaque de sintaxe

Ótima refatoração

Busca automática por erros

Interface intuitiva (função arrastar e soltar)

Bibliotecas dinâmicas e estáticas

Capacidade de desenvolvimento remoto

Suporte Qt

Suporta vários compiladores, incluindo CLang / LLVM, Cygwin, GNU, MinGW e Oracle Solaris Studio

Rastreamento de problemas integrado com Jira e Bugzilla

Possibilidade de pesquisar tarefas, salvar pesquisas, atualizar e resolver tarefas em seu repositório de registros.

Desvantagens:

Precisa de muita memória, então pode ficar lento em algumas máquinas ou projetos grandes;

Pop-ups irritantes.

Controladores de Versão de código

https://lh3.googleusercontent.com/a-/AOh14GhuFlRlwc-65WpchRSTyqvEZhLcVj25_N2TiTPIjQ=s40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Texugo <lucas103barreto@gmail.com>** | | Thu, Apr 28, 11:30 AM |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | to me  https://mail.google.com/mail/u/0/images/cleardot.gif | | | |

Durante um projeto na área de TI, se tem a criação de várias versões de um mesmo sistema, assim como de diversos desenvolvedores atuando em uma mesma tarefa. E ainda, mesmo após a conclusão, podem acontecer atualizações, soluções de erros e criação de novas versões.Tudo isso precisa ser registrado e controlado para garantir que todos os membros do grupo entendam o status de andamento do projeto. Esse processo também evita a perda de dados e facilita a gestão da empresa, contribuindo inclusive para a implementação do "DevOps" na empresa. É exatamente essa a função de uma ferramenta de controle de versão de software.

Entre elas(es) , estão algums exemplos abaixo:

1. CVS

A CVS é uma das ferramentas de controle de software mais antigas no mercado. A primeira versão dela foi desenvolvida em 1968. Essa ferramenta possui como maior desvantagem o fato de ser considerada como uma tecnologia antiga. Porém, ainda é bastante utilizada por equipes de desenvolvedores.

2. Subversion

Subversion é uma ferramenta de controle de versão de software bastante utilizada. Ela é bastante rápida na execução das funcionalidades do sistema e ainda se mostra como uma das mais simples de ser empregada. Isso significa que com um conhecimento básico relacionado ao controle de versão de software é possível executar qualquer comandos na ferramenta. A aprendizagem da equipe também é rápida nesse aspecto.

Um dos problemas do Subversion são as críticas relacionadas a eficácia do software. No passado, essa ferramenta apresentou problemas na hora de executar as principais funções de um controle de versão de software eficiente. Porém, as últimas versões lançadas parecem ter solucionado tudo que foi apontado como erros e desvantagens do programa.

3. TFS

O TFS — sigla para Team Foundation Server — é uma outra ferramenta de controle de versão de software que pode ser utilizada na sua empresa. Ele traz uma série de características interessantes, principalmente se você utiliza uma metodologia agile no setor de TI da sua empresa.

Isso porque ele possibilita a gestão de projetos por meio de "SCRUM ou CMMI". Também permite a utilização de forma centralizada ou distribuída, sendo adequado tanto para equipes que compartilham o mesmo espaço físico, quanto aquelas que trabalham à distância. Outra vantagem apresentada por esse sistema é o fato de não possuir limitações de crescimento e ter integração direta com o Microsoft Office , que ajuda e muito em certas situações.

4. GIT

O GIT é uma das ferramentas de controle de versão de software mais populares. Isso se deve, principalmente, pela popularidade do GitHub, uma plataforma para hospedagem de códigos. Apesar de ela poder ser utilizada em outras ferramentas, o GIT é a que ganhou mais repercussão nesse aspecto.

As principais vantagens dessa ferramenta são o design interno e interface, a eficácia e o desempenho do software. Isso significa que ele é agradável de ser utilizado, consegue atingir todos os objetivos de um bom controle de software e é rápido. Porém, apresenta desvantagens relacionadas à simplicidade de utilização. Apesar de ser a principal ferramenta de controle de versão de software disponível no mercado, possui controles um pouco mais complexos quanto comparado a outros softwares.

Isso significa que os colaboradores precisam entender uma série de conceitos mais profundos, relacionados ao controle de versão de software, para utilizar essa ferramenta corretamente. Apesar disso, é uma ferramenta de controle de versão distribuída, o que significa que é adequado para a utilização em grandes equipes, nas quais os desenvolvedores não estão localizados geograficamente no mesmo local.