# Engenheiro de Software-2021-1

Índice [[ocultar](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#)]

* [1 Pesquisa Tecnológica](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#Pesquisa_Tecnol.C3.B3gica)
* [2 Evolução na profissão](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#Evolu.C3.A7.C3.A3o_na_profiss.C3.A3o)
* [3 Descrição da atividade atual](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#Descri.C3.A7.C3.A3o_da_atividade_atual)
* [4 Requisitos mínimos para exercer a profissão](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#Requisitos_m.C3.ADnimos_para_exercer_a_profiss.C3.A3o)
* [5 Ferramentas usadas no dia-a-dia](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#Ferramentas_usadas_no_dia-a-dia)
* [6 Tecnologias envolvidas](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#Tecnologias_envolvidas)
* [7 Ética profissional](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#.C3.89tica_profissional)
* [8 Dúvidas](http://www.sourceinnovation.com.br/wiki/Engenheiro_de_Software-2021-1#D.C3.BAvidas)

# Pesquisa Tecnológica[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=1) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=1)]

* Levando em conta as ferramentas utilizadas pelo Marco, pesquise e escreva sobre o que evoluiu? Quais aplicações e vantagens dessa mudança, ou seja, compare uma ferramenta atual com uma que ele usava e explique o que melhorou nelas?
  + Exemplo: Supondo que ele citou uma ferramenta X para usar como Gestão de projetos e agora existe uma ferramenta Y que faz o mesmo, mostre as melhorias da ferramenta Y.

**Para desenvolvimento no sistema Mac: Objective-C e Swift**

Objective-C: Consiste-se em uma linguagem de programação orientada criada por Brad Cox juntamente com a equipe da sua empresa (StepStone Corporation) no início da década de 80, que, posteriormente, em 1988 foi licenciada pela NeXT, de Steve Jobs, vindo a se tornar a linguagem utilizada para o desenvolvimento do sistema operacional NeXTstep, sistema esse responsável por alavancar a popularidade da empresa NeXT, concedendo-a uma posição de provedor de workstations bastante popular durante essa época. Em anos seguintes, a empresa estendeu o compilador GCC para que pudesse suportar a Objectice-C, resultando no desenvolvimento das bibliotecas AppKit e Foundation Kit pela NeXT, nas quais a interface de usuário NeXTSTEP e o Interface Builder foram baseados. Embora as estações de trabalho NeXT não tenham causado um grande impacto no mercado, as ferramentas foram amplamente elogiadas na indústria. Depois de adquirir a NeXT em 1996, a Apple Computer usou o OpenStep em seu então novo sistema operacional, Mac OS X. Isso incluía Objective-C, a ferramenta de desenvolvedor baseada em Objective-C da NeXT, Project Builder, e sua ferramenta de design de interface, Interface Builder, ambos agora mesclados em um aplicativo, Xcode. Atualmente, a Objective-C é utilizada como uma linguagem de programação do MacOS X, que é baseado no NeXTstep. A versão da Apple do ambiente NeXTStep/GNUStep com adições é denominada Cocoa.

Swift: A linguagem de programação Swift foi lançada em 2014 pela empresa de tecnologia Apple com o objetivo de desenvolver aplicações para o sistema OS X e iOS. De lá pra cá essa linguagem está cada vez mais popular e de acordo com o ranking de linguagens mais usadas do sítio ‘TIOBE’, ela está em uma posição à frente da linguagem Objective-C, linguagem a qual era a mais usada no desenvolvimento dos sistemas da Apple e que perdeu o posto para a Swift. A premissa inicial para a criação da linguagem seria a falta de compatibilidade em projetos futuros com o uso da linguagem Objective-C, que foi construída tendo a linguagem C como base. Além disso, algo que gastaria várias linhas de código com Objective-C poderia ter esse código reduzido com a Swift. Logo, um dos principais objetivos para a criação da linguagem foi trazer mais simplicidade e objetividade no código a fim de atrair mais desenvolvedores, tanto programadores iniciantes quanto experientes. Outro fator é que Swift possui uma certa compatibilidade com a linguagem Objective-C, para que os desenvolvedores não sintam uma mudança muito acentuada se desejarem migrar para a linguagem Swift. Atualmente, a linguagem passou a ser totalmente open-source(código aberto) e está disponível para download no próprio domínio do GitHub. Além disso, ela se tornou disponível para ser desenvolvida no Ubuntu, uma das distribuições Linux.

**Para desenvolvimento Android: Java e Kotlin**

Java: Linguagem de programação criada pela Sun Microsystems em 1995 com o objetivo de facilitar a vida do desenvolvedor permitindo que o mesmo execute o que foi escrito uma vez através de qualquer dispositivo. Possui características que o destacam como sua simplicidade, sua ampla biblioteca, segurança, execução simultânea de várias tarefas e etc. Sua popularidade vem principalmente por usar o modelo de programação orientado a objetos. Porém, a linguagem Java também ganhou muita popularidade no desenvolvimento Android já que a mesma possibilita um uso que independe da própria plataforma, ou seja, seus códigos podem ser executados em qualquer tipo de hardware, o que facilita a escrita e adaptação para o sistema Android. É utilizada até hoje como uma das principais linguagens, sendo a 2° no ranking do site ‘TIOBE’, mantendo sua eficiência apesar da pouca mudança na última década.

Kotlin: Essa linguagem de programação foi criada pela empresa de desenvolvimento JetBrains, foi anunciada em 2011 mas sua primeira versão ocorreu apenas em 2016. A ferramenta é executada através da Máquina Virtual do Java(JVM), com o objetivo de ser usada para substituir a linguagem Java no desenvolvimento de aplicativos para o sistema Android. Kotlin conta com as vantagens de ser uma linguagem mais intuitiva e “amigável” com o programador em comparação a linguagem Java, mas ela ainda mantém certas semelhanças com Java, para que as empresas façam uma migração não tão radical entre as duas linguagens. Além disso, a linguagem Kotlin permite escrever menos código para determinadas ações, pode-se usar bibliotecas de Java e possui uma ótima documentação oficial feita pela própria empresa desenvolvedora da linguagem, a JetBrains.

**IDE’s de desenvolvimento: Xcode, Eclipse e Android Studio**

X-code: É um ambiente de desenvolvimento integrado(IDE) da empresa Apple com o objetivo de desenvolver aplicativos para os produtos da empresa (iPad, iPhone, Apple Watch e Mac, estando atualmente, na sua versão 12.4. Possuindo suporte para o código-fonte para as linguagens de programação C, C ++, Objective-C, Objective-C ++, Java, AppleScript, Python, Ruby, ResEdit (Rez) e Swift, com uma variedade de modelos de programação, incluindo, mas não se limitando a Cocoa, Carbon e Java. Não só isso, o X-code inclui a ferramenta GUI Instruments, que é executada em uma estrutura de rastreamento dinâmico, DTrace, criada pela Sun Microsystems e lançada como parte do OpenSolaris.

Eclipse: É um ambiente de desenvolvimento integrado (IDE) usado na programação de computadores, fruto de um consórcio de vários administradores com o propósito de desenvolver um software de código aberto, originando assim a Eclipse Foundation em janeiro de 2004. Possuindo uma ampla gama de ferramentas e um sistema de plug-in extensível para customizar o ambiente, essa IDE é escrita principalmente em Java e seu uso principal é desenvolver aplicações dessa linguagem, porém, pode ser usada para desenvolver aplicativos em outras linguagens de programação, por meio de plug-ins, incluindo Ada, ABAP, C, C ++, C #, Clojure, COBOL, D , Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (incluindo o framework Ruby on Rails), Rust, Scala e Scheme. Não só isso, ela ainda pode ser usada para desenvolver documentos com LaTeX e pacotes para o software Mathematica.

Android Studio: Essa ferramenta é um ambiente de desenvolvimento integrado(IDE) para aplicativos do sistema Android lançado em 2013 pela empresa Google. A IDE foi criada para simplificar algumas tarefas do desenvolvedor e trazer um ambiente mais amigável no desenvolvimento de apps. O Android Studio possui suporte para a correção de erros, é compatível com aplicativos de outras versões do sistema Android, além de contar com uma visualização gráfica mais intuitiva e ser disponibilizado gratuitamente pela Google. Desde o seu lançamento em 2013 até os dias de hoje, a ferramenta vem sendo utilizada pelos desenvolvedores da Google por meio de versões todos os anos, por exemplo a última versão principal lançada do Android Studio foi a versão 4.1 em Outubro de 2020. Essas versões servem para tratar correções de erros e trazer algumas melhorias que deixem o ambiente mais agradável no desenvolvimento.

**Gerenciamento de tarefa: Microsoft Project, Trello e Asana**

MS Project: É o software mais aceito mundialmente para o gerenciamento de projetos devido a sua versatilidade e facilidade de uso. Ele atua em diferentes etapas da gestão, porém onde ele mais se destaca é na etapa de planejamento e controle. Através de seu uso pode-se idealizar e programar atividades e até controlar prazos para as mesmas, traçar metas ou custos e muitas outras possibilidades. Tudo isso dá ao usuário um vasto campo de possibilidades para fazer a gestão de suas ambições.

Trello: É uma ferramenta de gerenciamento de projetos e tarefas gratuita criada pela Fog Creek Software em 2011, a aplicação opera tanto em desktop quanto em mobile. Ela é extremamente versátil e também é extremamente fácil de utilizar, oferece uma interface simples e intuitiva com vários recursos capazes de organizar tarefas ou grandes projetos. Seu recurso charme para ajudar o usuário é a divisão de tudo em pequenas etiquetas, que podem ser coloridas de cores específicas e podendo ter checklists ou até mesmo arquivos multimídia cujo upload é feito pelos próprios usuários. Esse recurso permite que várias pessoas possam saber o que fazer ou até mesmo quem está fazendo, assim sendo útil para planejar estudos, viagens e atividades de trabalho. Embora seja gratuita, a ferramenta disponibiliza um plano de assinatura premium com recursos e capacidades ainda mais refinados para ajudar na organização.

Asana: A ferramenta Asana é um dos gerenciadores de tarefas mais populares nos últimos anos. O Asana é uma plataforma com versões web, Android e IOS, que pode ser usada de forma gratuita por 30 dias, caso a pessoa queira continuar utilizando o serviço, há planos que são cobrados anualmente. Além disso, podem ser criadas equipes que possam visualizar uma mesma página para acompanhar o que deve ser feito, seja no trabalho ou escola. A plataforma serve para administrar as tarefas e afazeres de uma forma organizada e de uma maneira que fique bem visível para o usuário o que deve ser feito. Por exemplo, uma equipe pode criar tópicos como “Tarefas para fazer”, “Em execução”, “Já foi feito”, para poder sempre orientar os colegas sobre a situação atual de um projeto, trabalho em equipe ou ações similares. E também conta com um chat para a equipe ter uma forma de se comunicar mais rapidamente e de forma eficiente. Além disso, a ferramenta auxilia na definição de horários para dar uma noção de tempo para os integrantes de uma equipe, definindo prazos de entrega de cada parte de um projeto para cumprir com a data do prazo final, a fim de saber se a equipe está adiantada, no prazo ideal ou até mesmo atrasada com determinada tarefa.

# Evolução na profissão[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=2) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=2)]

* Como foi a evolução (acadêmica e profissional) até chegar na sua posição atual?
* Quais foram as principais formações (certificação, especialização, curso, etc) extra-curriculares?

O Marco começou como suporte de internet 3G na Algar Telecom. Cursou sistemas de informação na UFU, e na graduação entrou em uma startup de tecnologia como estagiário fazendo aplicativos para o sistema Android. No meio da graduação fez parte do PET e com isso fez iniciação científica dentro da universidade. Entrou em uma empresa em que foi desenvolvedor web e linguagem Java por 1 ano. Depois conseguiu preencher uma vaga para ser desenvolvedor mobile, e foi ganhando experiência de acordo com o tempo. Começou a ser analista de desenvolvimento mobile e aprendeu a desenvolver para o sistema IOS.

# Descrição da atividade atual[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=3) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=3)]

* O que efetivamente faz o profissional na sua condição?
* O que entrega como resultado?

Um engenheiro de software realiza a viabilidade de um projeto, ou seja, analisa se irá compensar fazer alguma modificação ou implementação no programa. Quando o projeto é muito grande, realiza pré-projetos, com a finalidade de fazer passo a passo e ir acompanhando melhor o andamento do projeto. Ter um sincronismo com a equipe de trabalho em relação aos requisitos passados e o prazo que vai ser gasto, por isso, o engenheiro de software fica ligado quanto as habilidades e fraquezas dos integrantes da equipe, distribuindo esses integrantes nas posições em que eles tenham uma produtividade maior.

# Requisitos mínimos para exercer a profissão[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=4) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=4)]

* Cite alguns soft skills necessários para quem tem intenção de seguir essa carreira?
* Sugira alguns hard skills.

Alguns soft skills que foram citados são ter foco e dedicação para alcançar os objetivos. Já os hard skills são ter conhecimento de linguagens de programação, familiaridade com IDE's e saber usar as ferramentas de gereciamento de tarefas, como por exemplo, os programas Trello e Microsoft Project.

# Ferramentas usadas no dia-a-dia[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=5) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=5)]

* Cite algumas ferramentas, tecnologias ou metodologias das quais depende para sucesso nas atividades.

As ferramentas usadas no dia-a-dia são divididas em linguagens, ambientes de desenvolvimento(IDE) e programas de desenvolvimento de tarefa. Nas linguagens podemos citar a linguagem Swift para o desenvolvimento IOS e Java para o desenvolvimento Android. Em relação as IDE's são usadas o Android Studio para Android e Xcode para o sistema IOS. Já em relação ao desenvolvimento de tarefa, são usados os programas Trello, Asana e Microsoft Project.

# Tecnologias envolvidas[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=6) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=6)]

* Sugira alguma tecnologia que possa ser pesquisada por esta equipe e que possa servir de base para futura aplicação nessa profissão.

# Ética profissional[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=7) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=7)]

* Pode dar algum exemplo já vivido por você ou por alguma outra pessoa sobre aspectos éticos na profissão, seja positivo ou negativo?

[[1]](https://youtu.be/KlPThJ6FDCs)= Upload da apresentação =

* Tem apresentação? Faça o upload aqui.

Prazo: 25/04

# Dúvidas[[editar](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&veaction=edit&vesection=8) | [editar código-fonte](http://www.sourceinnovation.com.br/w/index.php?title=Engenheiro_de_Software-2021-1&action=edit&section=8)]

01. Por quê se interessou mais pela área mobile? O que ela traz de diferente das outras que chamou sua atenção?

- A área mobile me cativou pelas possibilidades de projetos que poderiam ser realizados utilizando essa plataforma que, desde suas versões mais iniciais (Android 2.3) durante o período em que eu estava na graduação, já apresentavam um grande potencial de inovação.

02. Quais as linguagens que mais estão sendo usadas no mercado?

-Em relação ao desenvolvimento mobile, as linguagens que estão sendo mais usadas nesse mercado são Objective-C e Swift para desenvolvimento em sistema IOS, e em relação ao desenvolvimento Android, as linguagens Java e Kotlin.

03. Atualmente, as tecnologias usadas por um engenheiro de software mudaram muito de uns anos pra cá?

-Não houveram mudanças muito drásticas nesse meio tempo. O que aconteceu foi um crescimento das linguagens de programação Swift e Kotlin na área de desenvolvimento mobile.

04. Com qual frequência você precisa estudar outras tecnologias?

-Como engenheiro de software, o aprendizado é algo constante na minha carreira, devido não só as variedades de sistemas que são encontrados na área mobile, como também pela renovação constante das tecnologias embarcadas nesses sistemas, que demandam que o profissional esteja sempre em processo de aprendizado para melhor executar o seu trabalho.

05. Quais atividades o diferem do gestor de projetos?

-Dentre as características que diferem o engenheiro de software e o gestor de projetos, podemos citar o caráter mais técnico do engenheiro de software, por estar mais ligado a parte de planejamento e execução dos projetos, acarretando assim com uma maior necessidade de estabelecer uma boa sintonia com o resto dos profissionais envolvidos, facilitando a detecção de eventuais dificuldades que a sua equipe esteja enfrentando, possibilitando o melhor remanejamento de recursos e profissionais para solucionar esses tipos de imprevistos.

06. O que vc acha de frameworks mobile híbridos?

-O desenvolvimento de aplicativos mobile de forma híbrida é útil caso se queira criar um aplicativo que seja um protótipo, e caso se tenha um curto prazo no projeto. Caso esteja procurando por um aplicativo com performance, é recomendável fazer um aplicativo nativo. \*Opinião do grupo

07. Você consideraria sua rotina agitada? Você geralmente se depara com questões de segurança?

-Bastante agitada, principalmente quando se trata de trabalhar em conjunto com o resto da equipe para se alcançar o resultado esperado para um determinado projeto com um prazo de entrega estipulado. E sobre questões de segurança, consequentemente nós lidamos temos que nos atentar com elas, devido a sua extrema importância para o bom funcionamento do software.

08. O que ele precisou estudar e focar no começo da carreira para se tornar um engenheiro de software?

-De acordo com o Marco, foram vários fatores que influenciaram para que ele se torna-se um engenheiro de Software, dentre eles, podemos citar: A busca por conhecer novas tecnologias e áreas de atuação profissional para que assim, pudesse identificar em que área ele melhor se encaixaria, após isso, teve uma experiência como analista de sistemas, que agregou muito para a sua carreira e logo após isso, se especializou na área mobile, tornando-se um engenheiro de software.

09. A área de desenvolvimento é um bom caminho para descobrir em qual mercado me interesso mais?

-Para saber em qual área você se interessa, você deve obter o máximo de experiências possíveis em diferentes áreas para poder comparar e realmente saber se aquilo que você acha que gostaria de fazer é algo que te deixa feliz em fazer. A área de desenvolvimento é um setor dentro de vários outros disponíveis, e é interessante saber como funciona essa área, quais são as tecnologias usadas, como é a rotina de quem trabalha na área, entre outros fatores.

10. Qual a maior dificuldade no dia a dia do engenheiro de software?