Sumário

[Equipe/Professores do Bootcamp 2](#_Toc131502723)

[INTRODUÇÃO DESENVOLVIMENTO MODERNO DE SOFTWARE 3](#_Toc131502724)

[SISTEMAS DESKTOP X SISTEMAS WEB 3](#_Toc131502725)

[Vantagens do sistema desktop: 3](#_Toc131502726)

[Desvantagens do sistema desktop: 3](#_Toc131502727)

[Vantagens do sistema web: 3](#_Toc131502728)

[Desvantagens do sistema web: 3](#_Toc131502729)

[UX/UI DESIGNER 3](#_Toc131502730)

[UX DESIGNER 3](#_Toc131502731)

[UI DESIGNER 3](#_Toc131502732)

[MODELO CLIENTE-SERVIDOR 3](#_Toc131502733)

[FRAMEWORKS 3](#_Toc131502734)

[QUALITY ASSURANCE (QA) 3](#_Toc131502735)

[TRABALHANDO EM EQUIPES ÁGEIS 3](#_Toc131502736)

[MANIFESTO ÁGIL 3](#_Toc131502737)

[Valores 3](#_Toc131502738)

[Os 12 Princípios 3](#_Toc131502739)

[TIME ÁGIL 3](#_Toc131502740)

[INTRODUÇÃO À PROGRAMAÇÃO E PENSAMENTO COMPUTACIONAL 3](#_Toc131502741)

[Os 4 pilares do Pensamento Computacional 3](#_Toc131502742)

[1. Decomposição 3](#_Toc131502743)

[2. Reconhecimento de padrões 3](#_Toc131502744)

[3. Abstração 3](#_Toc131502745)

[4. Algoritmos: 4](#_Toc131502746)

[TECNICAS DE PROGRAMAÇÃO 4](#_Toc131502747)

# Equipe/Professores do Bootcamp

Leonardo Ropleato

Iglá Generoso

Juliana Mascarenhas @in/juliana-mascarenhas-ds

Venilton Falvo Jr @falvojr

Leonardo Buta @leonardo-buta

***Bootcamp Microsoft Dynamics 365***

# INTRODUÇÃO DESENVOLVIMENTO MODERNO DE SOFTWARE

## SISTEMAS DESKTOP X SISTEMAS WEB

Sistemas desktop são programas que são executados diretamente em um computador local. Eles geralmente têm uma interface gráfica do usuário (GUI) e são instalados no disco rígido do computador. Os sistemas desktop geralmente têm acesso direto a recursos locais, como a unidade de disco, impressoras e dispositivos USB. Eles podem oferecer desempenho mais rápido e melhor usabilidade do que os sistemas web, pois toda a aplicação é executada localmente.

Por outro lado, os sistemas web são aplicativos que são executados em um servidor e acessados através de um navegador web. Eles são executados em uma infraestrutura de nuvem, o que significa que não há necessidade de instalar nenhum software no computador local. Isso faz com que os sistemas web sejam mais acessíveis e portáteis do que os sistemas desktop, pois podem ser acessados de qualquer dispositivo com um navegador web e conexão à internet.

Vamos comparar as principais vantagens e desvantagens do sistema web e desktop:

### Vantagens do sistema desktop:

* Desempenho: o sistema desktop é executado diretamente no computador local, o que pode permitir um desempenho mais rápido em comparação com os sistemas web.
* Acesso a recursos locais: os sistemas desktop geralmente têm acesso direto a recursos locais, como a unidade de disco, impressoras e dispositivos USB. Isso pode tornar mais fácil trabalhar com arquivos e dispositivos locais.
* Controle total: os sistemas desktop oferecem controle total sobre a aplicação e seus dados. Isso pode ser especialmente importante para aplicações que lidam com dados sensíveis ou confidenciais.
* Maior usabilidade: a interface de usuário dos sistemas desktop pode ser mais rica e mais interativa do que a dos sistemas web.

### Desvantagens do sistema desktop:

* Portabilidade: os sistemas desktop são geralmente instalados em um único computador e não são tão facilmente portáteis como os sistemas web.
* Custo: o desenvolvimento de sistemas desktop pode ser mais caro do que os sistemas web, já que geralmente envolvem licenças de software e a necessidade de desenvolvimento em várias plataformas.
* Manutenção: os sistemas desktop precisam ser atualizados e mantidos localmente em cada computador onde estão instalados.
* Segurança: os sistemas desktop podem ser mais vulneráveis a ataques de segurança e malware do que os sistemas web, pois não são executados em um ambiente controlado como a nuvem.

### Vantagens do sistema web:

* Acessibilidade: os sistemas web podem ser acessados de qualquer lugar, a qualquer momento, desde que haja uma conexão à internet. Isso faz com que os sistemas web sejam altamente portáteis e acessíveis.
* Custos: os custos de desenvolvimento e implantação de sistemas web são geralmente menores do que os sistemas desktop, uma vez que não há necessidade de licenças de software ou instalação local.
* Manutenção: os sistemas web são mais fáceis de manter, já que as atualizações podem ser feitas em um único local no servidor e imediatamente disponibilizadas para todos os usuários.
* Colaboração: os sistemas web podem ser facilmente compartilhados e colaborados em equipe, permitindo que múltiplos usuários trabalhem juntos em tempo real.

### Desvantagens do sistema web:

* Desempenho: os sistemas web podem ser mais lentos do que os sistemas desktop, especialmente quando há uma conexão de internet lenta.
* Segurança: como os dados são armazenados em servidores remotos, os sistemas web podem ser vulneráveis a ataques de segurança. Isso significa que a segurança precisa ser cuidadosamente gerenciada e monitorada.
* Dependência de conexão de internet: o acesso a um sistema web requer uma conexão à internet confiável, o que pode ser um problema em áreas onde a conexão é instável ou inexistente.
* Interface de usuário limitada: a interface de usuário dos sistemas web é geralmente menos rica e menos interativa do que a dos sistemas desktop.

Em resumo, enquanto os sistemas desktop podem oferecer melhor desempenho, acesso a recursos locais e controle total, eles podem ser menos acessíveis e portáteis do que os sistemas web, além de serem mais caros e difíceis de manter. Por outro lado, os sistemas web são altamente acessíveis e portáteis, fáceis de manter e oferecem recursos de colaboração, mas podem ser mais lentos e menos seguros do que os sistemas desktop, além de exigir uma conexão confiável com a internet.

Em última análise, a escolha entre um sistema desktop e um sistema web dependerá das necessidades específicas do usuário e da aplicação em questão. É importante considerar cuidadosamente as vantagens e desvantagens de cada opção antes de tomar uma decisão.

## UX/UI DESIGNER

UX/UI designer é um profissional que se concentra na criação de interfaces de usuário atraentes, intuitivas e eficazes. UX (User Experience) e UI (User Interface) são duas áreas distintas do design que trabalham juntas para fornecer uma experiência geral de alta qualidade para o usuário.

## UX DESIGNER

O designer de UX é responsável por criar uma experiência de usuário positiva, garantindo que os usuários possam navegar e interagir com o produto de forma clara e fácil. O designer de UX trabalha em colaboração com outros membros da equipe, como gerentes de produto, desenvolvedores e designers de UI, para garantir que a experiência do usuário seja sempre priorizada.

## UI DESIGNER

O designer de UI é responsável pela aparência visual do produto, criando layouts e elementos de design que são atraentes, coerentes e eficazes. O designer de UI trabalha em colaboração com o designer de UX para criar interfaces de usuário que sejam atraentes, fáceis de usar e que transmitam a mensagem certa.

Em resumo, o designer de UX/UI é um profissional altamente especializado que trabalha na criação de interfaces de usuário eficazes e atraentes para produtos digitais, com o objetivo de garantir que os usuários tenham uma experiência positiva e satisfatória ao usar o produto.

## MODELO CLIENTE-SERVIDOR

O modelo cliente-servidor é um modelo de computação distribuída que divide o processamento de uma aplicação entre um servidor e um ou vários clientes. O servidor fornece um serviço centralizado, geralmente um banco de dados ou um aplicativo, e os clientes interagem com o servidor para acessar e manipular esses dados ou recursos.

O cliente envia solicitações para o servidor, que processa e retorna as informações ou respostas solicitadas. O servidor é responsável por gerenciar e armazenar os dados, enquanto o cliente é responsável por apresentar as informações ao usuário e fornecer uma interface de usuário amigável.

Esse modelo tem muitas vantagens, como a possibilidade de ter muitos clientes acessando e processando informações em tempo real, a facilidade de manutenção e atualização do sistema centralizado e a escalabilidade para suportar muitos usuários. No entanto, também apresenta algumas desvantagens, como a necessidade de uma rede confiável para conectar clientes e servidor e a possibilidade de sobrecarga do servidor em caso de muitos clientes acessando o sistema ao mesmo tempo.

## FRAMEWORKS

Frameworks são conjuntos de bibliotecas, componentes e ferramentas pré-construídos que fornecem uma estrutura para o desenvolvimento de aplicativos ou sistemas. Eles fornecem um conjunto de padrões, convenções e diretrizes para o desenvolvimento, permitindo que os desenvolvedores construam aplicativos de maneira mais rápida e eficiente, reutilizando códigos e recursos já existentes.

Os frameworks são frequentemente usados em desenvolvimento web, como por exemplo para desenvolver aplicativos em JavaScript ou PHP. Eles fornecem uma série de recursos e ferramentas para criar uma aplicação web, como gerenciamento de sessão, roteamento, validação de formulários e manipulação de banco de dados.

## QUALITY ASSURANCE (QA)

Quality Assurance, ou Garantia de Qualidade, é um processo que envolve o monitoramento e a avaliação da qualidade de um produto ou serviço para garantir que ele atenda aos requisitos e expectativas dos clientes. Isso pode envolver testes, revisões e avaliações contínuas do produto ou serviço para identificar e corrigir quaisquer problemas ou falhas.

O objetivo da Garantia de Qualidade é garantir que um produto ou serviço seja entregue com a mais alta qualidade possível, com foco na prevenção de defeitos em vez de sua correção após a entrega. Isso ajuda a garantir a satisfação do cliente, a manutenção da reputação da empresa e a redução de custos associados à correção de problemas após o lançamento.

Os profissionais de Garantia de Qualidade trabalham em conjunto com outras equipes, como desenvolvedores e gerentes de projetos, para definir padrões e processos de qualidade, e para garantir que esses padrões sejam cumpridos durante todo o ciclo de vida do produto ou serviço.

# TRABALHANDO EM EQUIPES ÁGEIS

## MANIFESTO ÁGIL

O Manifesto Ágil é um documento que descreve os valores e princípios da metodologia ágil de desenvolvimento de software. Foi criado em 2001 - Utah, Estados Unidos, em uma reunião de 17 desenvolvedores de software que compartilhavam a frustração com as práticas tradicionais de desenvolvimento.

O objetivo do Manifesto Ágil é propor uma abordagem mais eficiente e eficaz para o desenvolvimento de software, enfatizando a colaboração, a flexibilidade e a entrega rápida de software funcional. Desde então, a metodologia ágil se tornou amplamente adotada em todo o mundo.

### Valores

Os valores do Manifesto Ágil são um conjunto de quatro declarações que representam as crenças fundamentais da metodologia ágil de desenvolvimento de software. São eles:

* Indivíduos e interações acima de processos e ferramentas: o foco deve estar nas pessoas e na colaboração entre elas, em vez de enfatizar apenas processos e ferramentas.
* Software funcionando acima de documentação abrangente: o objetivo deve ser criar software funcional e entregá-lo ao cliente em vez de se concentrar em documentação extensa.
* Colaboração com o cliente acima de negociação de contratos: a colaboração constante com o cliente é essencial para garantir que o projeto esteja no caminho certo, em vez de depender de contratos inflexíveis.
* Responder a mudanças acima de seguir um plano: a metodologia ágil valoriza a capacidade de se adaptar às mudanças e de ajustar o plano para atender às necessidades em constante evolução do projeto, em vez de se apegar rigidamente a um plano original.

### Os 12 Princípios

Os 12 princípios do Manifesto Ágil são as diretrizes que guiam o desenvolvimento de software ágil. São eles:

1. Satisfação do cliente através da entrega contínua e adiantada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo no final do desenvolvimento.
3. Entregar software funcionando com frequência, na escala de semanas a meses, com preferência aos períodos mais curtos.
4. Pessoas relacionadas ao negócio e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o projeto.
5. Construir projetos em torno de indivíduos motivados.
6. Fornecer um ambiente de suporte e confiança e permitir que os indivíduos realizem seu trabalho.
7. Conversa face a face é o método mais eficiente e eficaz de transmitir informações dentro de uma equipe de desenvolvimento.
8. Software funcional é a medida primária de progresso.
9. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
10. Atenção contínua à excelência técnica e bom design aumentam a agilidade.
11. Simplicidade - a arte de maximizar a quantidade de trabalho não realizado - é essencial.
12. As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis.

O Manifesto Ágil pode ser acessado através do seguinte link: <https://agilemanifesto.org/>

## TIME ÁGIL

Um time ágil possui diversas características que o diferenciam de equipes tradicionais de desenvolvimento de software. Algumas das principais características de um time ágil incluem:

1. Colaboração e comunicação: membros da equipe trabalham juntos em estreita colaboração e se comunicam regularmente para garantir que o projeto esteja seguindo na direção certa e para resolver quaisquer problemas ou desafios que surjam ao longo do caminho.
2. Flexibilidade e adaptabilidade: um time ágil é capaz de se adaptar rapidamente às mudanças e desafios que surgem durante o desenvolvimento de um projeto. O time é flexível o suficiente para mudar de direção rapidamente, caso as necessidades do cliente mudem ou novos desafios surjam.
3. Foco no valor: o time ágil está constantemente focado em fornecer valor ao cliente. Eles priorizam recursos e funcionalidades que fornecem valor significativo ao cliente, em vez de se concentrar em recursos que não são críticos para o sucesso do projeto.
4. Auto-organização: os times ágeis são auto-organizados e auto-gerenciados. Cada membro da equipe tem um papel importante a desempenhar e todos trabalham juntos para alcançar os objetivos do projeto.
5. Transparência: um time ágil é transparente em relação ao seu processo de desenvolvimento, progresso e desafios. Eles trabalham em um ambiente onde todas as informações são compartilhadas livremente para todos os membros da equipe.
6. Entrega contínua: um time ágil trabalha com o objetivo de entregar software funcional de forma contínua e adiantada. Eles trabalham em ciclos curtos, chamados de sprints, onde uma quantidade significativa de trabalho é concluída e entregue ao cliente.

Essas são algumas das principais características de um time ágil. É importante lembrar que cada equipe ágil é única e pode ter suas próprias maneiras de trabalhar, mas todas compartilham essas características centrais.

VUCA(Volátil, Incerto, Complexo, Ambíguo)

# INTRODUÇÃO À PROGRAMAÇÃO E PENSAMENTO COMPUTACIONAL

O Pensamento Computacional (PC) é um conjunto de habilidades mentais que permite aos indivíduos resolver problemas de forma eficiente e eficaz, utilizando conceitos e métodos que são utilizados pelos cientistas da computação. Ele é uma habilidade essencial para o sucesso na era digital e pode ser aplicado em várias áreas, incluindo engenharia, ciência, medicina, negócios e educação.

## Os 4 pilares do Pensamento Computacional

1. Decomposição: é a habilidade de dividir um problema complexo em partes menores e mais gerenciáveis, facilitando a resolução do problema como um todo. Ao decompor um problema, podemos entender melhor cada uma de suas partes e como elas se relacionam umas com as outras.
2. Reconhecimento de padrões: é a habilidade de identificar padrões em dados, informações ou eventos. Isso envolve a capacidade de identificar semelhanças e diferenças entre diferentes conjuntos de dados e identificar padrões que possam ajudar na resolução de problemas.
3. Abstração: é a habilidade de generalizar a informação, reduzindo a complexidade para lidar com o problema em um nível mais alto de abstração. Essa habilidade envolve a capacidade de identificar os aspectos mais importantes de um problema e ignorar os detalhes irrelevantes.
4. Algoritmos: são sequências lógicas de instruções que são executadas para resolver um problema. A habilidade de criar e executar algoritmos é essencial para a resolução de problemas em computação e em outras áreas que requerem soluções baseadas em processos lógicos.

Ao combinar esses quatro pilares, o pensamento computacional pode ajudar indivíduos a resolver problemas de maneira mais eficaz, eficiente e criativa. Ele também ajuda a promover habilidades importantes, como resolução de problemas, pensamento crítico e colaboração.

## TECNICAS DE PROGRAMAÇÃO

As técnicas de programação linear, estruturada e modular são abordagens diferentes para organizar e desenvolver código em projetos de software.

Técnica linear: é uma abordagem antiga e obsoleta que consiste em escrever todo o código de um programa em uma única sequência linear de instruções, sem divisão em funções ou módulos. Esse tipo de programação pode ser difícil de ler, entender e manter, e é especialmente problemático em projetos de grande escala. Portanto, a técnica linear não é mais amplamente utilizada na programação moderna.

Técnica estruturada: é uma abordagem mais organizada para a programação que foi desenvolvida na década de 1960. Ela se concentra em dividir um programa em blocos de código que são fáceis de entender e manter. Isso é feito por meio do uso de estruturas de controle de fluxo, como loops, condicionais e sub-rotinas. A técnica estruturada torna o código mais fácil de ler, entender e manter, o que é especialmente importante em projetos de grande escala.

Técnica modular: é uma extensão da técnica estruturada que se concentra em dividir um programa em módulos independentes e reutilizáveis. Cada módulo é responsável por uma função específica dentro do programa, e esses módulos podem ser facilmente combinados para criar programas maiores e mais complexos. A técnica modular permite que os programadores se concentrem em um módulo de cada vez, o que torna o processo de desenvolvimento mais gerenciável e escalável.

Em resumo, enquanto a técnica linear é obsoleta e impraticável, a técnica estruturada e modular são técnicas modernas e amplamente utilizadas na programação. A técnica estruturada ajuda a tornar o código mais organizado e fácil de entender, enquanto a técnica modular permite que os programadores criem programas mais complexos e escaláveis, dividindo-os em módulos independentes e reutilizáveis.

# INTRODUÇÃO AO .NET

O .NET é uma plataforma de desenvolvimento unificado que permite a construção de sistemas e aplicações.

C# é uma linguagem de programação usada no .NET.