



E-book Módulo 1 - Conceitos Fundamentais para um Desenvolvedor de Software Low-Code

- ≡ O que é Low-Code?
- ≡ Exemplos de Empresas Usando Low-Code
- ≡ Vantagens e Desvantagens Low-Code
- ≡ O que é Preciso para ser um Desenvolvedor Low-Code?
- ≡ Algoritmo e Lógica de Programação
- ≡ Variável
- ≡ Função e Ação
- ≡ Encerramento

O que é Low-Code?

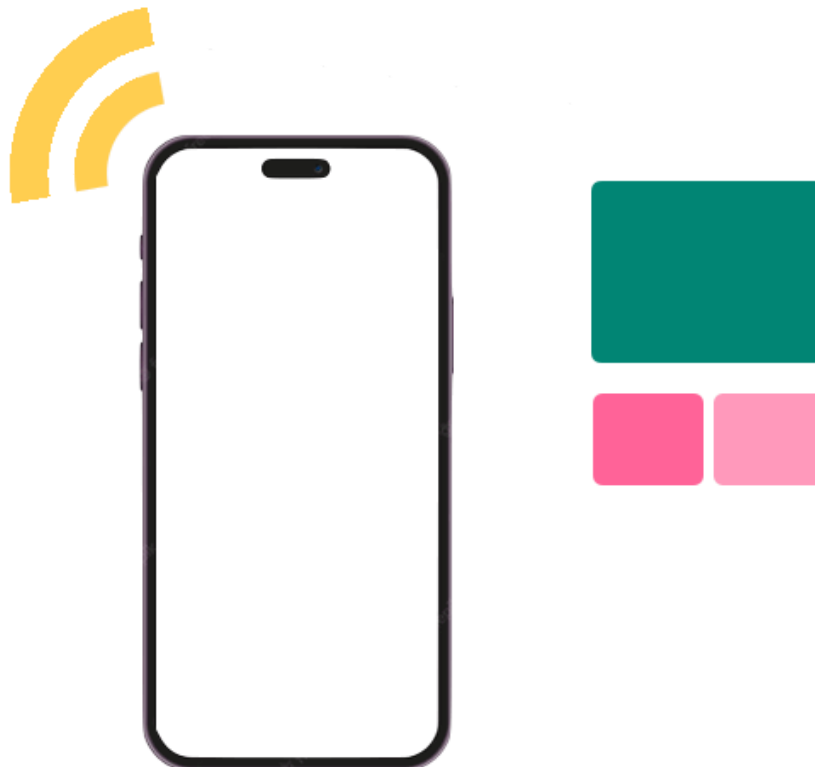
O que é Low-Code?

Low-Code é uma abordagem de desenvolvimento de softwares (p.ex. aplicativos para smartphones e páginas web) que usa plataformas/ferramentas que exigem nenhuma ou pouca programação manual por parte dos desenvolvedores.



Com essa abordagem o foco é tornar mais simples e rápido a criação de softwares, permitindo que pessoas com menos experiência em programação possam criar aplicativos funcionais.

Plataformas Low-Code facilitam esse desenvolvimento, possibilitando que a criação desses softwares sejam principalmente a partir da manipulação de interfaces visuais, que permitem realizar ações de "arrastar e soltar", assim como o uso de componentes pré-configurados.



Visualize a ideia do que é o “arrastar e soltar”.

Exemplos de Empresas Usando Low-Code

Exemplos de Empresas Usando Low-Code

1

Itaú

O Itaú do Uruguai usou uma plataforma Low-Code para transformar seu aplicativo nativo, o Itaú Pagos, em um super app, ou seja, aplicativo que agrega diferentes aplicações em um só.

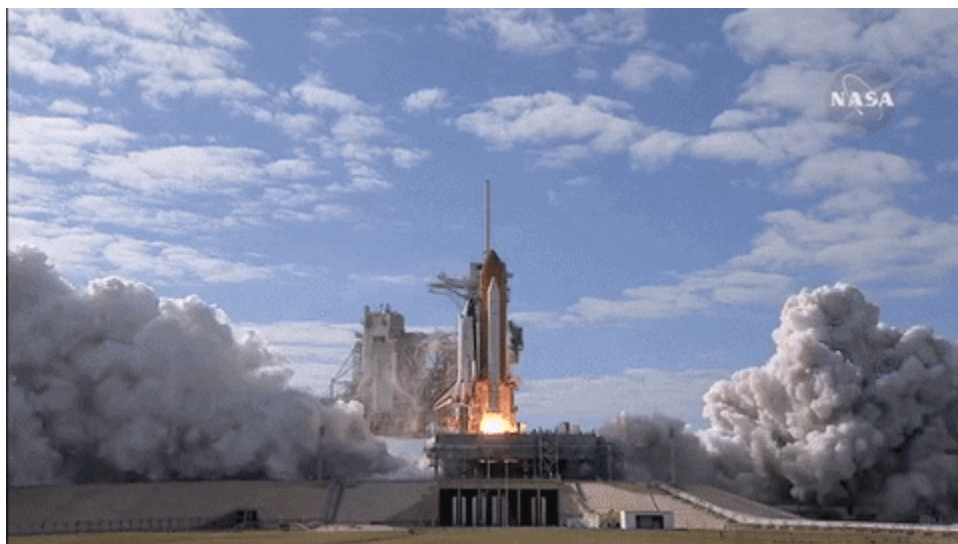
A mudança do app nativo para um super app aconteceu em poucas semanas, devido o uso da plataforma Low-Code, permitindo uma experiência unificada de fornecedores e clientes do Itaú e que reunisse diversos serviços e oportunidades de negócios em uma única aplicação.

Fonte: Baguete Diário, Setembro de 2023.

NASA

A agência espacial NASA tem adotado o uso de uma plataforma Low-Code, líder de mercado, para transformar sua abordagem digital! Seu uso tem permitido revitalizar sua presença na web, por exemplo, a partir da remodelagem do seu site principal e sua plataforma de streaming.

Fonte: Artigo intitulado "Why NASA Chose WordPress for Revamping Its Flagship Website".



Petrobras

Adotou uma plataforma Low-Code desde 2021, e tem procurado fazer com que aplicações existentes passem a ser criadas e evoluídas a partir desse tipo de plataforma. Espera-se que sejam contratados diversos profissionais especializados nessa tecnologia e em outras áreas complementares.

Na área pública, outros órgãos também tem adotado o uso de Low-Code, como por exemplo, o Serpro e a Anatel.

Fonte: InforChannel, Agosto de 2022.

Vantagens e Desvantagens Low-Code

Vantagens e Desvantagens Low-Code

Clique nos itens em destaque.



- > 1. *Aceleração no desenvolvimento*
- > 2. *Redução de custos*
- > 3. *Facilidade de uso*
- > 4. *Flexibilidade*
- > 5. *Facilidade em integração*
- > 6. *Foco na Inovação*
- > 7. *Manutenção simplificada*



- > 1. *Limitação na Personalização*
- > 2. *Limitação para integração*
- > 3. *Dependência do fornecedor*



Vantagens

- 1. *Aceleração no desenvolvimento*
- 2. *Redução de custos*
- 3. *Facilidade de uso*
- 4. *Flexibilidade*
- 5. *Facilidade em integração*
- 6. *Foco na Inovação*
- 7. *Manutenção simplificada*



Desvantagens

- 1. *Limitação na Personalização*
- 2. *Limitação para integração*
- 3. *Dependência do fornecedor*

1. Aceleração no desenvolvimento

Low-Code permite a criação mais rápida de aplicações ao reduzir a quantidade de código manual que precisa ser escrito. Isso é muito útil, por exemplo, para projetos com prazos apertados.



Vantagens

1. *Aceleração no desenvolvimento*



2. *Redução de custos*

3. *Facilidade de uso*

4. *Flexibilidade*

5. *Facilidade em integração*

6. *Foco na Inovação*

7. *Manutenção simplificada*



Desvantagens

1. *Limitação na Personalização*

2. *Limitação para integração*

3. *Dependência do fornecedor*

2. Redução de custos

O tempo para desenvolver algum software (p.ex. sistemas web e aplicativo) é menor, como consequência, custos associados tendem a reduzir, como por exemplo, mão de obra.



1. *Aceleração no desenvolvimento*

2. *Redução de custos*



3. *Facilidade de uso*

4. *Flexibilidade*

5. *Facilidade em integração*

6. *Foco na Inovação*

7. *Manutenção simplificada*



1. *Limitação na Personalização*

2. *Limitação para integração*

3. *Dependência do fornecedor*

3. Facilidade de uso

Os recursos visuais de arrastar e soltar componentes para ir montando a solução tornam as plataformas Low-Code acessíveis a usuários com pouca ou nenhuma experiência anterior em programação. Dessa forma, mais pessoas podem contribuir com o desenvolvimento.



Vantagens

1. Aceleração no desenvolvimento

2. Redução de custos

3. Facilidade de uso



4. Flexibilidade

5. Facilidade em integração

6. Foco na Inovação

7. Manutenção simplificada



Desvantagens

1. Limitação na Personalização

2. Limitação para integração

3. Dependência do fornecedor

4. Flexibilidade

Apesar de fornecer facilidades visuais para a construção de soluções, caso haja a necessidade de inserir códigos de programação, as plataformas permitem. Dessa forma, se houver a necessidade, algum desenvolvedor mais experiente pode colaborar.



5. Facilidade em integração

Muitas plataformas Low-Code oferecem integrações fáceis com outros sistemas, permitem armazenar dados usando diferentes tecnologias, entre outras integrações. Essa vantagem traz agilidade no desenvolvimento de soluções.



6. Foco na inovação

Reduzindo os desafios técnicos que são lidados pelas plataformas Low-Code, empresas e desenvolvedores podem dar mais atenção em como trazer inovação para seu negócio e para suas soluções.



7. Manutenção simplificada

As aplicações desenvolvidas em Low-Code tendem a ser mais fáceis de realizar mudanças. Com isso, traz uma tranquilidade adicional tanto para a equipe envolvida.



Vantagens

1. *Aceleração no desenvolvimento*

2. *Redução de custos*

3. *Facilidade de uso*

4. *Flexibilidade*

5. *Facilidade em integração*

6. *Foco na Inovação*

7. *Manutenção simplificada*



Desvantagens

1. *Limitação na Personalização*

2. *Limitação para integração*

3. *Dependência do fornecedor*

1. Limitação da personalização

As plataformas Low-Code podem fornecer funcionalidades e componentes pré-fabricados, sendo que alguns deles nem sempre poderão ser personalizados. Assim, o desenvolvedor pode ficar limitado para realizar certas mudanças específicas necessárias.



Vantagens

1. *Aceleração no desenvolvimento*

2. *Redução de custos*

3. *Facilidade de uso*

4. *Flexibilidade*

5. *Facilidade em integração*

6. *Foco na Inovação*

7. *Manutenção simplificada*



Desvantagens

1. *Limitação na Personalização*



2. *Limitação para integração*

3. *Dependência do fornecedor*

2. Limitação para integração

Integrar aplicativos desenvolvidos a partir de plataformas Low-Code com sistemas mais antigos, chamados sistemas legados, ou com outras ferramentas podem ter limitações, o que pode dificultar a integração com o ecossistema tecnológico existente de alguma empresa que esteja desejando usar Low-Code.



Vantagens

1. Aceleração no desenvolvimento

2. Redução de custos

3. Facilidade de uso

4. Flexibilidade

5. Facilidade em integração

6. Foco na Inovação

7. Manutenção simplificada



Desvantagens

1. Limitação na Personalização

2. Limitação para integração

3. Dependência do fornecedor



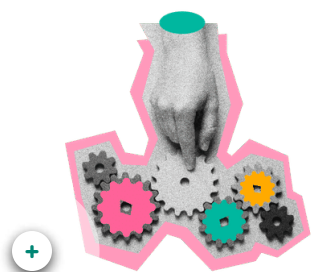
3. Dependência do fornecedor

Uma empresa, por exemplo, ao optar pelo uso de uma plataforma Low-Code, pode ficar vinculada ao fornecedor dessa plataforma. Mudar para outra plataforma pode ser um desafio.

O que é Preciso para ser um Desenvolvedor Low-Code?

O que é Preciso para ser um Desenvolvedor Low-Code?

Clique nos itens para entender o que é necessário para se tornar um desenvolvedor Low-Code.



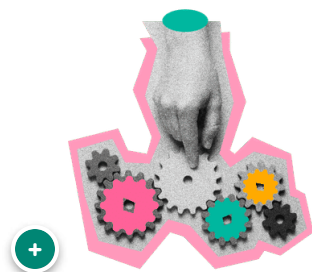
*Etapas para desenvolver
soluções tecnológicas*



*Conceitos relacionados
a programação*



*Plataforma
Low Code*



*Etapas para desenvolver
soluções tecnológicas*



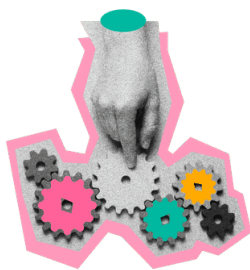
*Conceitos relacionados
a programação*



*Plataforma
Low Code*

Etapas para desenvolver soluções tecnológicas

Entender quais etapas principais estão envolvidas na construção de qualquer software é algo muito importante. Apesar da abordagem Low-Code facilitar a criação de softwares, ela está inserida em um conjunto de passos que devem ser realizados. Esses passos caracterizam um processo de desenvolvimento. Entender o que é esse processo e o que deve ser considerado para criar qualquer software, permite potencializar o uso de Low-Code.



*Etapas para desenvolver
soluções tecnológicas*



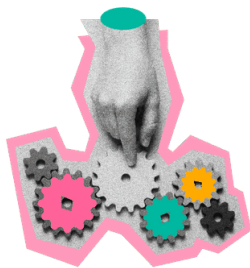
*Conceitos relacionados
a programação*



*Plataforma
Low Code*

Conceitos relacionados à programação

Mesmo Low-Code reduzindo o uso de código para criar soluções, conhecer conceitos fundamentais relacionados a programação são úteis na comunicação da equipe, planejamento de atividades e uso da plataforma Low-Code.



*Etapas para desenvolver
soluções tecnológicas*



*Conceitos relacionados
a programação*



*Plataforma
Low Code*

Plataforma Low-Code

Atualmente existem diversas plataformas Low-Code. Cada uma delas tem suas particularidades trazendo facilidades para certas situações. Assim, conhecer bem a plataforma que irá usar, é algo importante para permitir o desenvolvimento de soluções usando o seu potencial. Durante o curso, você terá a oportunidade de iniciar o uso prático de uma plataforma conhecida e muito usada em mercado.

Algoritmo e Lógica de Programação

Algoritmo e Lógica de Programação

O que é um Algoritmo?

Um algoritmo é uma sequência finita de instruções bem definidas e não ambíguas, cada uma das quais pode ser executada em um período de tempo finito e com uma quantidade de esforço finita.

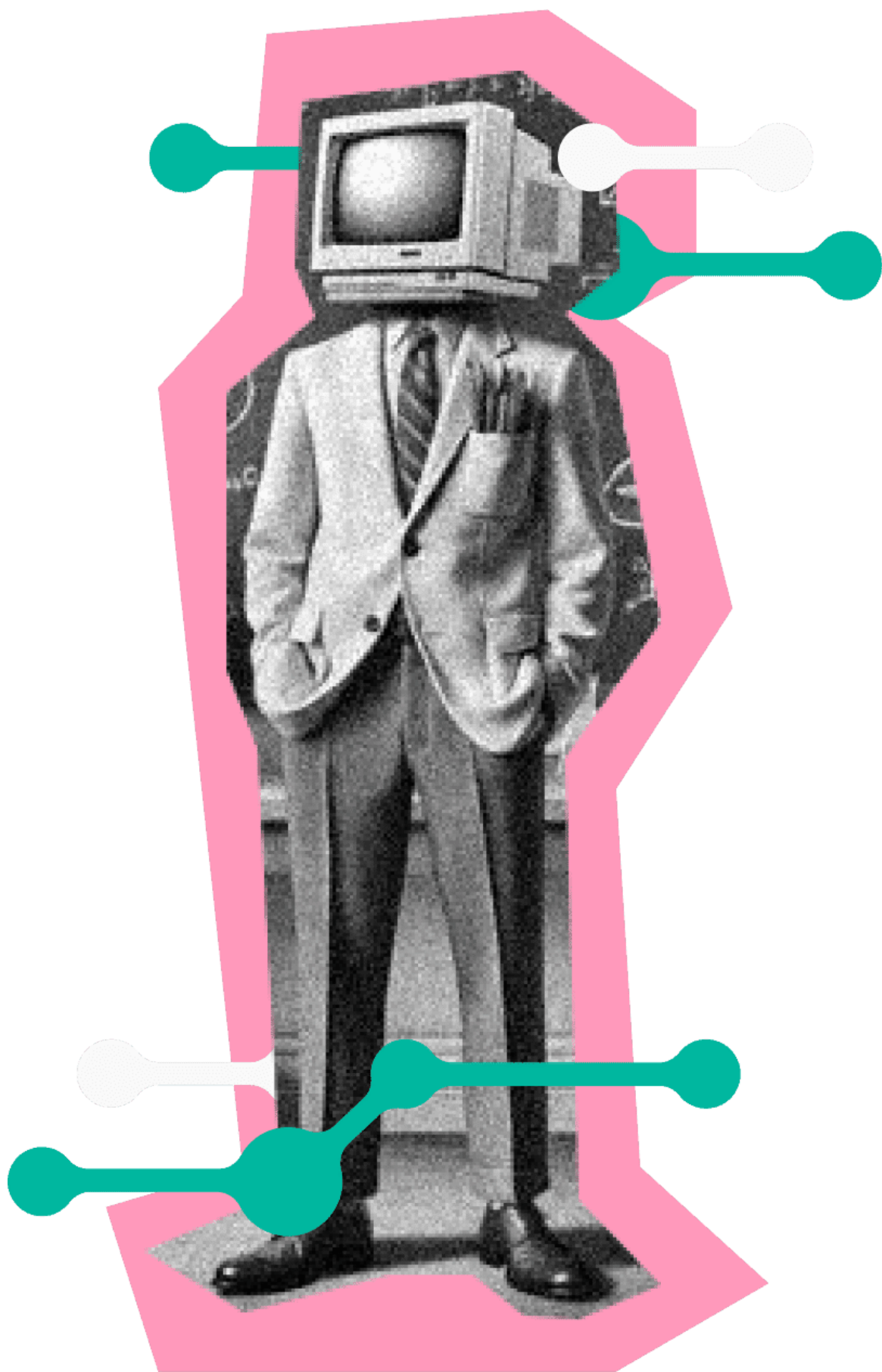
Em termos mais simples, um algoritmo é um conjunto de passos para resolver um problema ou realizar uma tarefa. Os algoritmos são a base de tudo o que é feito em programação de computadores, servindo como a receita que diz ao computador o que fazer com os dados fornecidos.

O que é Lógica de Programação?

Lógica de programação, por outro lado, é a técnica de **encadear pensamentos** para alcançar um determinado **objetivo**.

Estes pensamentos podem ser um conjunto de instruções que serão traduzidas em uma linguagem de programação e então executadas por um computador.

A lógica de programação é fundamental para o desenvolvimento de algoritmos, pois é através dela que se define a melhor sequência de passos (instruções) para resolver um problema.



Onde as coisas se fundem?



A **lógica de programação** é usada para **estruturar e desenvolver os algoritmos** que, por sua vez, são implementados em uma linguagem de programação específica para criar programas de computador.

Ela ajuda os desenvolvedores a pensar de maneira estruturada e sequencial, permitindo-lhes decompor problemas complexos em partes menores e mais gerenciáveis, que podem ser resolvidas com algoritmos.



Variável



Variável

Uma variável, em programação, pode ser definida como um espaço de armazenamento nomeado usado para guardar valores que podem ser modificados durante a execução de um programa.

Ela funciona como uma etiqueta que você coloca em uma caixa para saber o que tem dentro: essa "caixa" pode conter diferentes tipos de dados, como números, textos, ou

mesmo valores mais complexos.



A natureza "variável" do conteúdo é o que dá nome a essa estrutura, pois o valor dentro dela pode variar conforme necessário ao longo do tempo.

As variáveis são fundamentais porque permitem que os programas armazenem e manipulem dados de maneira dinâmica.



Ao usar variáveis, os desenvolvedores têm a flexibilidade para realizar cálculos, tomar decisões lógicas, e gerenciar informações de maneira eficiente dentro de seus programas.

Cada variável é definida com um tipo específico que determina a natureza dos dados que ela pode conter, garantindo que as operações realizadas com esses dados sejam apropriadas e seguras.

Função e Ação

Função e Ação

Uma função em programação é uma sequência de instruções agrupadas que realizam uma tarefa específica, podendo ser reutilizadas várias vezes dentro de um programa.

As funções são úteis para dividir programas complexos em componentes menores e mais gerenciáveis, facilitando tanto a escrita quanto a manutenção do código. Geralmente, uma função recebe dados de entrada (conhecidos como parâmetros), executa operações com esses dados e retorna um resultado.

O uso de **funções** também promove a **modularidade** e a **reutilização de código**, permitindo que os desenvolvedores **evitem redundâncias** e façam modificações em um único lugar, que automaticamente se aplicam onde a função é chamada.

Além disso, as funções podem ajudar a definir interfaces claras entre diferentes partes do código, o que é essencial para trabalhar em projetos grandes ou em equipe. Isso torna as funções fundamentais para qualquer linguagem de programação e essenciais para a criação de software eficiente e fácil de entender.

Ação no OutSystems

Conceito existente no OutSystems, uma ação é um bloco de lógica que você define para realizar uma operação específica em sua aplicação. Essas ações são bastante semelhantes às funções em linguagens de programação tradicionais, mas se diferem por não retornar valores.

Visualmente, as ações no OutSystems são representadas no Service Studio como blocos ou fluxos de lógica que você pode arrastar e conectar em um diagrama de fluxo. Isso facilita a visualização e o entendimento do fluxo de dados e do controle dentro da aplicação.



Em resumo

Enquanto as funções em linguagens de programação são blocos de código designados para tarefas específicas, as ações no OutSystems estendem esse conceito para integrar mais diretamente com a interface de usuário e outras partes da aplicação, de maneira visual, facilitando assim o desenvolvimento rápido de aplicações com menos código tradicional.

Encerramento

Clique na seta abaixo para fazer o download deste E-book.



File Attachment Block

No file added





PARABÉNS

Você completou este Módulo