



MUST
UNIVERSITY
FLORIDA - USA

LOW CODE - LCD680 - 2.2

```
    f = remove_last_coma(f);  
    return i.push(this.app_settings.__ROOT__  
    endsWith(c, "/") && (i[o] = remove_  
    makeCALL = function(a) {  
        this.  
        !((this).data("page_type"));  
        replace(r.app_settings.__ROOT__  
        a.split("/");  
        function(a, r) {  
            this.  
            set_history_state(a),  
            contexted(i, "loader")  
            set(i.pages[r].blank_render, funct  
            contexted(i, "loader")  
            set(i.  
            set(i.app_settings.api_url,  
            set(i.app_settings.api_url, a
```

Metodologias para implantação de Low Code





Metodologias para implantação de *Low Code*

Conteúdo organizado por **Renato Cividini Matthiesen** em 2022 dos artigos “**Análise da abordagem LOW-CODE como facilitador da transformação digital em indústrias**”, “**Desenvolvimento de aplicação móvel para submissão/revisão de despesas recorrendo a metodologias ágeis de desenvolvimento e à plataforma low-code OutSystems**” e “**Desenvolvimento rápido de aplicações – comparação de soluções em Outsystems e Mendix**”.

Objetivos de Aprendizagem

- Verificar a importância dos aplicativos para os negócios, Metodologias para implantação de *Low Code*, planejar e desenvolver aplicativos de *Low Code*, desenvolver sites de negócios *Low Code*, Estudo de caso *Low Code*.

Introdução

Os avanços das tecnologias da informação levaram o mundo empresarial a depender intensamente do uso de sistemas de software. Enquanto no final do século XX os sistemas nas empresas eram idealizados para realização de trabalhos localizados em ambientes físicos, dentro de prédios e indústrias localizadas geograficamente em algum local, ainda que já interconectados à internet, o novo milênio trouxe a mobilidade amparada pela disseminação das redes de computadores e acesso à internet junto a computação móvel. Este fato, fez com que os sistemas de software passassem a ser idealizados para operações em formato de aplicações, ou aplicativos, que operam em quaisquer plataformas, como *desktops*, *notebooks*, *tablets*, *netbooks*, *smartphones*, *smartwatches* ou dispositivos de Internet das Coisas, conforme relata Rogers (2021).

Uma nova forma de desenvolvimento de sistemas de software já se consolida como necessária no contexto de uso intenso e quase que ubíquo de sistemas de softwares, aplicações e aplicativos em grande parte dos aparelhos eletrônicos da atualidade.

Neste momento, o Low Code visa então diminuir ou eliminar a quantidade de código escrito de forma manual por programadores de sistemas, conforme nos ensina Karmali (2019). Isto leva o desenvolvimento de programas, permitindo que profissionais além de especialistas em tecnologia da informação possam desenvolver estas aplicações, principalmente considerando sistemas mais simples e tradicionais para gestão empresarial.

O Low Code e o No Code representam tecnologias que surgiram como alternativas ao formato tradicional de desenvolvimento de aplicações de software através de programação via uso de linguagens de programação escritas em formato de algoritmos. O Low Code foi a primeira destas tecnologias a começarem a se popularizar, trazendo componentes (blocos ou módulos) já programados para serem reutilizados em outras aplicações.

Ambas as tecnologias citadas auxiliaram a consolidação do conceito de desenvolvimento de sistemas de software no formato de aplicações ou aplicativos para plataformas móveis (*tablets* e *smartphones*, por exemplo), que intensificaram ainda mais a necessidade de programação de sistemas, já que podemos observar

que vivenciamos um mundo intensamente servido por sistemas de software, conforme reflexão de Rogers (2021).

Como uma abordagem tecnológica relativamente nova, metodologias, plataformas, ferramentas ainda estão em pleno desenvolvimento, ou mesmo em início de desenvolvimento e deverão ser apresentadas em artigos, revistas especializadas, livros dentre outros veículos de disseminação de ciência. Temos aqui uma interessante oportunidade de investimento e negócios.

Os desafios neste momento focam na popularização e maior desenvolvimento de plataformas de Low Code e No Code, de forma que a cultura do desenvolvimento de sistemas e aplicações de software possam ser melhor recebidas por profissionais que necessariamente não são especializados em tecnologias e sistemas de informação conforme observa Castelo Branco (2022).

Saiba Mais

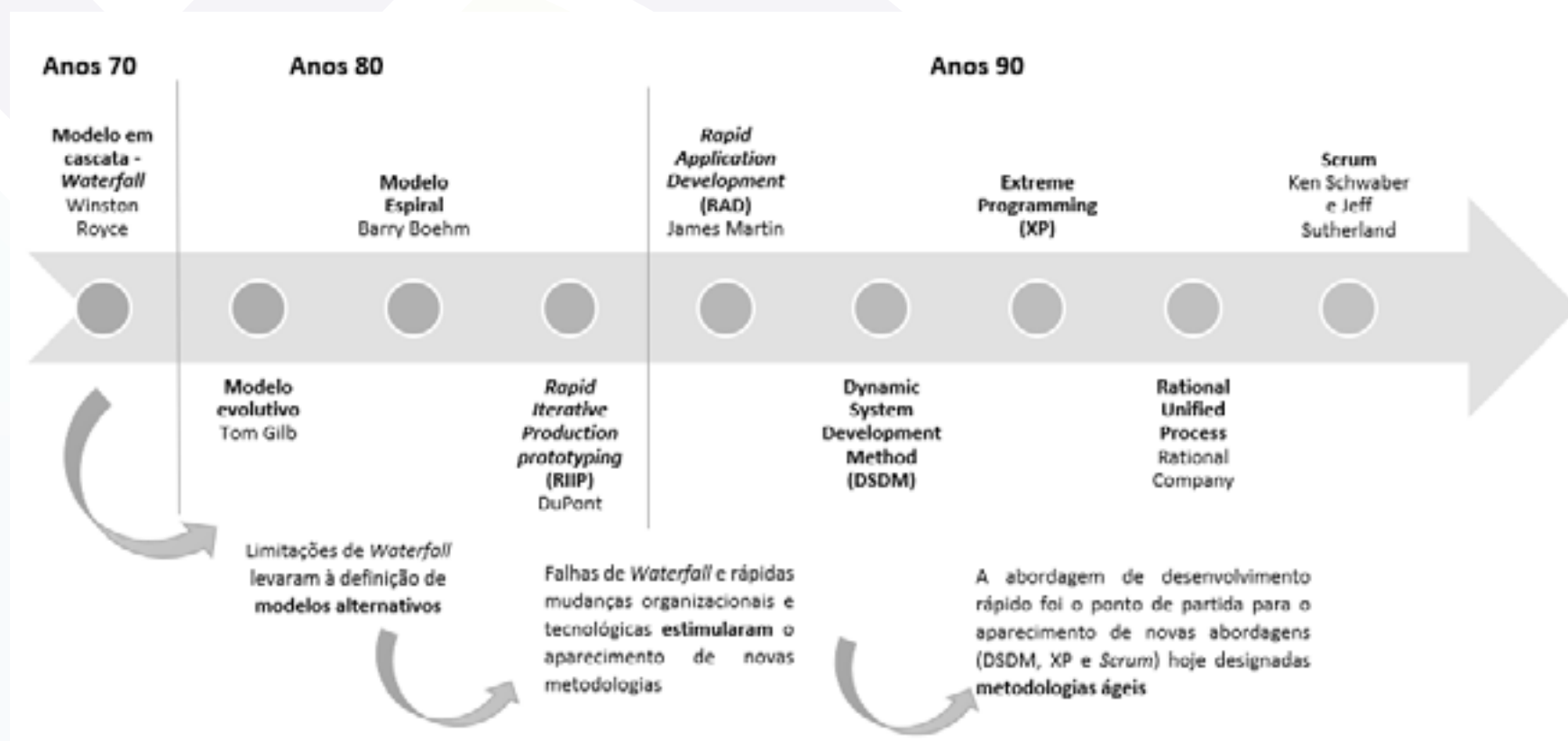
Leia o artigo: O que são as metodologias low-code e no-code? O artigo foi escrito por Dácio Castelo Branco em 2022 e está disponível no Canaltech.

Link: <<https://bit.ly/3T2BwWb>>. Acesso em: 22 set. 2022.

Métodos e metodologias de desenvolvimento de software

O desenvolvimento de aplicações de software, ou especificamente de sistemas de software vem evoluindo e acompanhando a própria evolução da tecnologia da informação, ou computadores. Veja a figura 2 com a evolução cronológica das abordagens de desenvolvimento de sistemas apresentada por Pinto (2021).

Figura 2 – Cronologia da evolução de métodos e metodologias de desenvolvimento



Fonte: Pinto (2021, p. 6)

Na década de 1970 o modelo concebido para desenvolver sistemas de software era o modelo em cascata ou *Waterfall*, chamado de modelo clássico por Pressman e Maxim (2021) relaciona-se a um modelo de ciclo de vida linear, onde as etapas de desenvolvimento são dependentes de suas antecessoras. Neste modelo, os requisitos e escopo precisam ser bem definidos no início do projeto. No modelo espiral dos anos 1980, há introdução de um modelo evolutivo e do RUP (*Rapid Iterative Prototyping*), formalizada pela DuPont, tendo sido definidas técnicas que coletivamente vieram ser conhecidas mais tarde como RAD (*Rapid Application Development*), reflete Pinto (2021). Mais adiante, metodologias que depois deram origem ao manifesto ágil e o aparecimento das chamadas metodologias ágeis foram desenvolvidas como o DSDM (*Dynamic System Development Method*) e a XP (*eXtreme Programming*), já nos anos 1990 quando em um momento de intenso desenvolvimento de computadores pessoais, internet e plataformas de software baseadas em interfaces gráficas. Também importante, foi o desenvolvimento do RUP (*Rational Unified Process*), que reuniu os elementos de todos os modelos de processo genéricos (cascata, incremental, integração/configuração) e apoia a prototipação e a entrega incremental. Este modelo tem as seguintes fases: *Concepção*: para estabelecer o *business case*; *Elaboração*: para

desenvolver os requisitos e a arquitetura do software; *Construção*: em que o software é codificado e *Transição*: em que o sistema é implantado.

No final dos anos 1990 surge o Scrum, como uma nova proposta de desenvolvimento de sistemas de software e gestão de projetos, que mais recentemente foi incorporada também na vida corporativa para gestão de processos. O *Scrum* é a metodologia ágil mais utilizada na atualidade, com origem no ano de 1993. De acordo com Sutherland e Sutherland (2019) tem como foco o andamento do projeto de software através de controles realizados por um subconjunto de atividades realizadas em formato de rituais, chamados de *Sprints*. Diversas outras metodologias de desenvolvimento de software foram desenvolvidas e muitas outras ainda deverão surgir, considerando a intensa necessidade de evolução tecnológica da atualidade. Veremos um pouco mais sobre esta metodologia futuramente, considerando que faz base para os modelos que abarcam o Low Code como abordagem de desenvolvimento.

Nesta aula, vamos observar a proposta evolutiva do RAD (Rapid Application Development) desenvolvida por Alves e Alcalá (2022) após um estudo sobre o novo paradigma de desenvolvimento de aplicações com a abordagem Low Code.

Proposta de metodologia para implantação de Low Code



Após realizarem um estudo sobre o Low Code como facilitador da transformação digital nas indústrias, Alves e Alcalá (2022) apresentaram como contribuição de sua pesquisa uma proposta de implantação do Low Code em empresas. A metodologia proposta pelos autores é apresentada na figura 3, com uma sistematização sustentada pela ausência de referências bibliográficas consolidadas, devido à natureza de vanguarda da tecnologia que até então pode ser considerada como nova.

Figura 3 – Metodologia de implementação de abordagem Low Code



Fonte: Alves e Alcalá (2022, p. 15)

Vamos conhecer um pouco melhor as fases dessa metodologia.

- **Análise de processo:** realizado um mapeamento de todas as etapas que constituem um processo é realizado um levantamento dos requisitos e funcionalidades da aplicação. Realização também de análise de viabilidade de digitalização do processo utilizando a tecnologia Low Code.
- **Modelagem de dados:** primeira etapa operacional da abordagem Low Code, onde o usuário cria as entidades (tabelas dos bancos de dados), estabelece os relacionamentos, define as restrições e as dependências, atividades essas geralmente realizadas por meio de recursos de arrastar e soltar componentes, sem codificação.
- **Desenho de interfaces:** etapa destinada para elaboração visual das telas (interfaces) que irão compor o aplicativo, caracterizada pela compatibilidade com a criatividade do profissional que está desenvolvendo a aplicação, o que “permite um aumento da personalização das aplicações e melhor experiência do usuário”, conforme sustentam Alves e Alcalá (2022, p. 16).

- **Menor Produto Viável:** com a conclusão das etapas iniciais, pode-se então liberar uma versão da aplicação no contexto de MVP (*Minimum Viable Product*), ou seja, produto mínimo viável, para experimentação do usuário e contribuição para continuidade de desenvolvimento da aplicação em desenvolvimento.

Saiba Mais

O **MVP (*Minimum Viable Product*)** ou Produto Mínimo Viável é originário de 2001 e evoluiu com o termo de **Lean startup**, que define que as empresas necessitam de processos e ferramentas adaptados às suas necessidades, desenvolvidos com alta velocidade e sem desperdícios. Busca-se a construção de produtos de forma iterativa e evolutiva onde a primeira versão já pode ser utilizada pelo cliente, que interage rapidamente para sua atualização. A figura 4 mostra uma representação do MVP.

Figura 4 – MVP (*Minimum Viable Product*)



Fonte: Pinterest: representando Ries (2019).

- **Implementação de regras de negócio:** definidos o MVP, as regras de negócio definidas na etapa inicial da análise do processo são implementadas utilizando-se funções de sintaxe simples e de fácil compreensão. Os usos de ferramentas visuais de comunicação suportada por dados como o Storytelling podem ser utilizados.
- **Implementação de serviços externos:** integram a aplicação com serviços externos por meio de interface e programação de aplicações, chamada de API (*Application Programming Interface*) de terceiros para integração com diferentes aplicações.
- **Disponibilização:** após a conclusão das etapas de desenvolvimento as aplicações podem ser publicadas e compartilhadas com os usuários finais, considerando que a visualização e a disponibilização da aplicação podem ser realizadas com o menor número de cliques possível.

Saiba Mais

Conforme apresenta Pereira et al (2019, p. 203) **data storytelling pode ser entendida como uma técnica para apresentar uma narrativa ou contar histórias com dados.** Seus principais componentes para a apresentação de dados neste formato são a narrativa, os recursos visuais e naturalmente os dados.

A Narrativa é acoplada aos dados e busca explicar o que está acontecendo no contexto abordado e o significado e a importância de um *insight*. Os recursos visuais aplicados aos dados buscam esclarecer os insights

Figura 5 – Elementos principais do data storytelling



Fonte: Pereira et al (2019, p. 204).

que não poderiam ser visualizados pelo público sem uso de tabelas ou gráficos. Isto nos remete a refletir que a narrativa e o visual misturados buscam envolver o público para concepção e compreensão dos dados, informações e *insights*.

Morais (2020, p. 124) diz que *storytelling* utiliza palavras ou recursos audiovisuais para transmitir uma história, que pode ser contada de improviso ou pode ser polida e trabalhada. Também é muito usado no contexto da aprendizagem, sendo uma importante forma de transmissão de elementos culturais como regras e valores éticos.

Em Resumo

Nesta aula, tivemos a oportunidade de refletir sobre o Low Code e o No Code representarem tecnologias que surgiram como alternativas ao formato tradicional de desenvolvimento de aplicações de software através de programação via uso de linguagens de programação escritas em formato de algoritmos. O Low Code foi a primeira destas tecnologias a começarem a se popularizar, trazendo componentes (blocos ou módulos) já programados para serem reutilizados em outras aplicações. Vimos que diversas metodologias vêm sendo desenvolvidas desde a década de 1970 e que mais recentemente, as metodologias preconizam o desenvolvimento ágil de sistemas, como o Scrum e o eXtreme Programming, assim como outras o seu desenvolvimento rápido, como a RAD (*Rapid Iterative Prototyping*). A aula trouxe também uma nova proposta desenvolvida com um estudo sobre aplicação de plataformas Low Code para desenvolvimento de aplicações, baseada nas fases de: análise de processo, modelagem de dados, desenho de interfaces, Menor Produto Viável, implementação de regras de negócio, implementação de serviços externos e disponibilização.

Na ponta da língua



Referências Bibliográficas

ALVES, F.; ALCALÁ, S. G. S. (2022). Análise da abordagem *LOW-CODE* como facilitador da transformação digital em indústrias. [Revista e-TECH: Tecnologias para competitividade industrial, V. 15, n. 2]. **E-TECH Tecnologias para competitividade Industrial**. Florianópolis, BR. DOI: <https://doi.org/10.18624/etech.v15i1.1186>.

DÁCIO, Castelo Branco. (2022). O que são as metodologias low-code e no-code? [Documento Web]. **Canaltech**. Recuperado de: <https://canaltech.com.br/mercado/o-que-sao-as-metodologias-low-code-e-no-code-205938/>. Acesso em: 17 set. 2022.

KARMALI, S. (2019). Desenvolvimento de aplicação móvel para submissão/revisão de despesas recorrendo a metodologias ágeis de desenvolvimento e à plataforma *low-code* OutSystems. [Departamento de Ciências e Tecnologias da Informação da ISCTE]. **Instituto Universitário de Lisboa**. Lisboa, PT.

MORAIS, Felipe. (2020). **Transformação digital**. São Paulo: Saraiva Educação.

PEREIRA, Mariana A. et al. (2019). **Framework de Big data**. Porto Alegre: SAGAH.

PINTO, I. M. M. M. (2021). Desenvolvimento rápido de aplicações – comparação de soluções em Outsystems e Mendiz. [Instituto Superior de Engenharia do Produto]. **Politécnica do Porto**. Porto, PT.

PRESSMAN, Roger, S.; MAXIM, Bruce, R. (2021). **Engenharia de Software**: uma abordagem profissional, 9. Ed. Porto Alegre: AMGH.

RIES, Eric. (2019). **A startup enxuta**. Rio de Janeiro: Sextante.

ROGERS, David, L. (2021). **Transformação digital**: repensando seu negócio para a era digital. 1. ed. 4. Reimp. São Paulo: Autêntica Business.

SUTHERLAND, Jeff; SUTHERLAND, J. J. (2019). **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. Rio de Janeiro: Sextante.





ARTIGOS DE REFERÊNCIA:

Análise da abordagem LOW-CODE como facilitador da transformação digital em indústrias.

[Revista e-TECH: Tecnologias para competitividade industrial, V. 15, n. 2]. E-TECH Tecnologias para competitividade Industrial. Florianópolis, BR - ALVES, F.; ALCALÁ, S. G. S. (2022). DOI: <https://bit.ly/3S1UNpC>

Desenvolvimento de aplicação móvel para submissão/revisão de despesas recorrendo a metodologias ágeis de desenvolvimento e à plataforma low-code OutSystems.

[Departamento de Ciências e Tecnologias da Informação da ISCTE]. Instituto Universitário de Lisboa. Lisboa, PT - KARMALI, S. (2019).

Desenvolvimento rápido de aplicações – comparação de soluções m Outsystems e Mendiz.

[Instituto Superior de Engenharia do Produto]. Politécnica do Porto. Porto, PT. PINTO, I. M. M. M. (2021).



MUST
UNIVERSITY
FLORIDA - USA