

# Valderi Leithardt, Dr.

IPW (Class fourteen notes)

# Summary

- Review of previous class \*
- Deploy
- Exercises
- Work 2, 3 and 4

# Script Cliente *(Review of previous class)*

## Conceito front-end

- ❖ Para que um site seja iniciado, é preciso contar com processamentos em dois lados: no navegador (cliente) e no servidor, que hospeda o site. O primeiro compreende, naturalmente, toda a parte visual: menus, imagens, vídeos, links, estrutura do site, composição dos elementos e tudo o que se refere à usabilidade.
- ❖ No desenvolvimento web, chamamos de front-end a parte responsável por aspectos visíveis e interativos de uma página web, da parte de quem o acessa (cliente). Para quem está acessando uma página, é tudo aquilo que ela vê e manipula — como recursos gráficos, elementos de uma interface e até mesmo parte da lógica de programação do site.
- ❖ Essa área trata das escolhas mais básicas, como a definição de onde o menu deverá ser posicionado, até questões mais complexas, como a validação de dados. Para essas funções, o front envolve o uso de linguagens como HTML, CSS e JavaScript — linguagem mais usada (67,7%) em projetos web, segundo [Stack Overflow](#) 2020 e [2023](#).

# Script Cliente *(Review of previous class)*

## Conceito Back-end

- ❖ A parte mais robusta e complexa da lógica de programação de um site vai ser definida pelo que chamamos de back-end. Trata-se das funções base de uma aplicação web, a infraestrutura por trás do que está visível para utilizadores. Por isso, envolve também mais abstração.
- ❖ O back-end trata da lógica do servidor. Em uma estrutura web, o servidor é quem responde às requisições de clientes e oferece o que foi solicitado. Nesse viés, auxilia a gestão da conexão com base de dados, a questão da segurança e conexão com outras aplicações, por meio de APIs (um código em comum que possibilita a comunicação entre dois sistemas).
- ❖ **Full stack** é o profissional que trabalha front end e back end. Esse perfil consegue dominar conceitos e tecnologias das duas áreas, portanto, tem capacidade de desenvolver bem uma interface gráfica e uma lógica de interação e estabelecer comunicação com APIs, base de dados e outros elementos necessários para uma aplicação web.

# Deploy

- ❖ O verbo deploy, em inglês, quer dizer implantar. Em programação, seu sentido está relacionado à sua tradução literal: fazer um deploy, em termos práticos, significa colocar no ar alguma aplicação que teve seu desenvolvimento concluído.
- ❖ Com base nas fases do ciclo de vida de desenvolvimento de software ([SDLC – Software Development Life Cycle](#)), temos a seguinte ordem:
  - ✓ Planejamento de como será determinada função do seu software;
  - ✓ Posteriormente a criação da interface dessa funcionalidade para que entre para o desenvolvimento com todos os testes necessários;
  - ✓ Quando estiver pronto, esse código pode ser disponibilizado para que seja disponibilizado ao cliente (aqui entra o deploy);
  - ✓ Por fim, o sistema entrará em manutenção, formando um ciclo sem fim com o planejamento de novas funcionalidades.

# Deploy

três possíveis formas de se fazer um deploy:

- **Deploy manual via FTP** (Protocolo de Transferência de Ficheiro), mas precisa de interferência humana para acontecer. Entre as maiores **desvantagens** é o tempo e recursos gastos com uma atividade que poderia ser feita rapidamente, otimizando a produtividade da equipa;
- **Parcialmente automatizado:** Atualizar um repositório GIT, por exemplo, é fazer um deploy parcialmente automatizado, apesar de alguns comandos humanos serem necessitados, o push do branch master que ocorre através de um pequeno hook e atualiza o servidor é totalmente automático. Outra vantagem deste deploy é o controle de versões do GIT;
- **Completamente automatizado:** É possível garantir mais segurança, qualidade e eficiência nas atualizações e melhorias em sites, aplicativos ou softwares de grande utilização, como redes sociais e até mesmo serviços Google. Existem hoje, no mercado, diversas ferramentas voltadas à automatização total do processo de deploy, entre as mais populares estão Jenkins, GitLab, Azure Pipelines e Circle CI. Esta categoria de deploy, além de transmitir automaticamente as suas atualizações para o servidor, também realiza aquilo que chamamos de integração contínua. O conceito de CI, ou [Continuous Integration](#), é, em resumo, a prática de unir diversas alterações de código em um repositório central, automaticamente, com o objetivo de fazer testagens e execuções antes do deploy final.

# Deploy

três possíveis estratégias para deploy:

## Blue-Green

- A estratégia tem característica de que todos os testes de uma aplicação são feitos em um ambiente de produção espelhado, chamado de mirror. Com esta estratégia, o blue representa o ambiente antigo, enquanto o green representa o ambiente atualizado. Para implementar o deploy nestes moldes, a aplicação é espelhada em produção, mas os utilizadores não têm acesso enquanto não estiver totalmente pronta. Quando o deploy é finalizado, um load balancer direciona o tráfego para o novo ambiente, enquanto o antigo é excluído.

## Canary

- Os testes da nova versão são feitos na prática pelos seus utilizadores finais. Esta é a mais complexa estratégia de deploy porque consiste em liberar a nova versão de um software, site ou aplicativo apenas para uma parte dos clientes que o utilizam. Em seguida, são feitas análises de seus comportamentos para a identificação de erros e falhas. Com o Canary, a nova versão também coexiste com a antiga, com a diferença de que aqui as duas versões estão em um ambiente de produção acessível aos utilizadores finais.

## Rolling

- Esta estratégia consiste em fazer com que duas versões de uma mesma aplicação coexistam enquanto o deploy é executado. Na prática, este deploy é realizado substituindo totalmente o código fonte de uma aplicação por um novo código, que já contém as alterações a serem implementadas. Este deploy é feito de modo gradual e a versão antiga só sairá do ar quando a nova estiver totalmente pronta.

# Deploy

❖ Segundo [AWS](#) (2023) os modelos de SDLC mais usados são:

→ Cascata, Iterativo, Espiral, Ágil.

❖ De acordo com [ServiceNow](#) (2023), Ciclo de vida de desenvolvimento de software é um processo de planejamento, criação, teste e implantação de sistemas de informação para hardware e software.

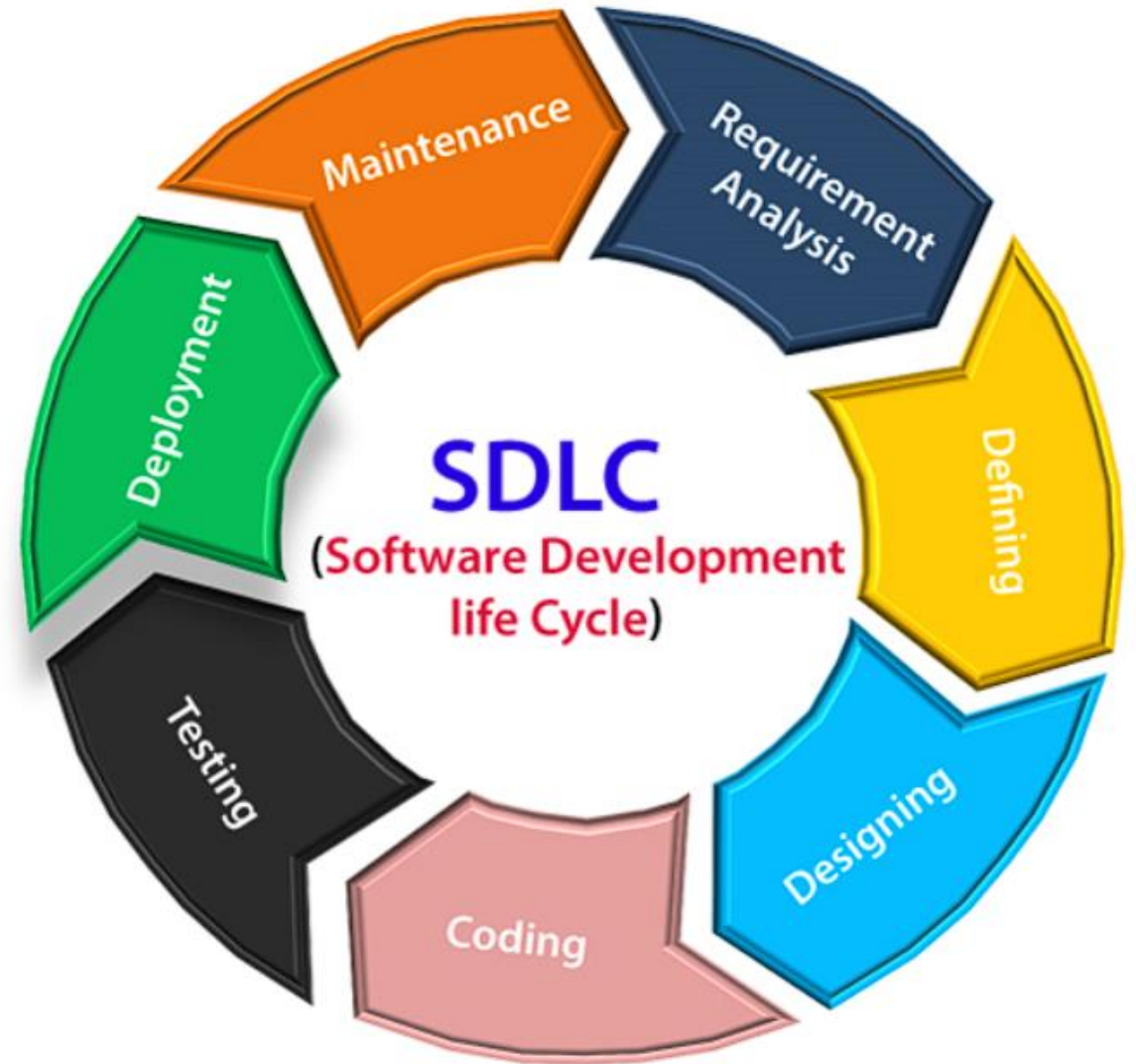


Figura extraída de [Java Point](#) (2023).



# Deploy

- Tutorial: [Implementar uma aplicação Web Node.js no Azure](#)
- Tutorial: [Deploy da sua aplicação para a web usando Express.js e Heroku](#)
- Tutorial: [Implementando um Sítio Estático](#)
- Tutorial: [Fundamentos do Deployment Manager](#) (Google cloud)

# Exercises

❖ Exercícios de capítulos 14 e 15 do livro da disciplina disponível em:

➤ Exercícios do livro capítulo 14, disponível em:

- [https://eloquentjavascript.net/14\\_dom.html](https://eloquentjavascript.net/14_dom.html)

➤ Exercícios do livro capítulo 15, disponível em:

- [https://eloquentjavascript.net/15\\_event.html](https://eloquentjavascript.net/15_event.html)

- 2º, 3º e 4º Trabalhos disponibilizados em:

<https://github.com/isel-leic-ipw/2324i-IPW-LEIC31D/wiki>

**\*\* Concluir exercícios das aulas anteriores.**

# References

- Acessado em 2023 <https://www.w3.org/news/2023/new-w3c-website-deployment/>
- Também foram realizadas adaptações e modificações com base no material disponibilizado por Professor Luís Falcão, acesso online em: <https://github.com/isel-leic-ipw/>
- Aulas gravadas Professor Falcão:
- Aula 14 (30/10): <https://videoconf-colibri.zoom.us/rec/share/uJqbBQ6gKlIQOmm3eB l0h64uX5wh97jwPVU3cGe5lhunB uCaljuijBasP0 p9FW 1m.0uErW8kPXolpwm7q>
- Aula 15 (06/11): <https://videoconf-colibri.zoom.us/rec/share/pD0bViUhr6cNutXipMID3DZjDPwdJevXq IFMv1uN1KVo5jQtIPGYILIQY EetmU5O .r1GBu4nB jZuKiWk>
- Aula 16 (10/11): <https://iselleicipwpi-sdr2356.slack.com/archives/C05R7HWC8PM/p1699713953288629>
- Aula 17 (10/11): <https://videoconf-colibri.zoom.us/rec/share/4pWzSMoPB2YgMV8pwRuGGhyyw-xvhpGv-Bv08nK87PKg9eMN-TRdu8AJZfJwL3Cz.LqMvaQ7R48iac8W1>

\* Todos os links foram acessados em 2023.

# Valderi Leithardt, Dr.

Professor IPW

[valderi.leithardt@isel.pt](mailto:valderi.leithardt@isel.pt)