Módulo1A #TDW #MCTW

MINIPROJETO M1A

CONTEXTO DO DESAFIO

Com a evolução da Web e o crescente número de aplicações que através dela têm vindo a ser disponibilizadas, tem-se vindo a observar em toda a vertente de desenvolvimento técnico subjacente à mesma um aumento exponencial de complexidade, que se observa transversalmente a todas as facetas associadas a esse desenvolvimento. Nunca foi tão complicado desenvolver uma aplicação para a Web como o é hoje, embora por outro lado também nunca tenha sido tão fácil gerir essa complexidade.

Associado a essa temática—mais concreta na construção de soluções que permitam colocar uma aplicação nas mãos do utilizador final em poucos minutos—este desafio passa, em primeiro lugar, pelo desenvolvimento de uma *pipeline* de integração contínua para um projeto já existente e, em seguida, da sua replicação em qualquer outra plataforma de *CI/CD* à escolha. Alguns exemplos de outras plataformas:

- GitLab
- CircleCI
- Semaphore CI
- Travis CI
- etc...

DINÂMICA GERAL

Sendo que o objetivo do desafio passa pelo desenvolvimento de uma *pipeline* de integração contínua e não de código aplicacional, fornece-se à partida uma base de código que pode ser utilizado para este propósito:

- https://github.com/deca-ua/mp1-template-david

Não é obrigatória a sua utilização, mas qualquer código adicional que seja desenvolvido não será o fator principal de avaliação.

O projeto fornecido inclui todas as dependências necessárias à realização do desafio, e consiste numa aplicação desenvolvida em Next.js que, quando compilada, vai buscar conteúdos a um espaço de Contentful e renderizar as páginas estáticas correspondentes a esses conteúdos (resultando essencialmente em páginas HTML de um *master / detail* simples).

Requisitos

- Integração de ferramentas de validação de qualidade de código (e.g. ESLint, Prettier, e Jest) que corram tanto localmente como remotamente;
- Integração com o Contentful (ou qualquer outro CMS) para obtenção de dados a renderizar;
- Integração com o Netlify ou outro à sua escolha para efetuar o *deployment* do projeto;
- Desenvolvimento de uma *pipeline* em GitHub que cumpra os seguintes requisitos:
    - Tenha várias etapas que corram em sequência para instalar dependências, validar a qualidade do código e fazer *deploy* do projeto;
    - Apenas corra o *deploy* quando a pipeline for executada na *branch* default;
    - Possa ser executada:
        - Nos vários *commits* que sejam feitos no repositório;
        - Na *branch default*, sempre que os conteúdos do CMS forem atualizados (i.e. através de um *webhook*);
        - Na *branch default*, às 00:00 de todos os dias da semana.
            - Depois de validar que este *trigger* funciona, tem atenção para o desligar para não exceder os limites do workspace do GitHub.
- Replicação dessa mesma *pipeline* em qualquer outra plataforma de CI/CD.

Implementação

Para o efeito do desenvolvimento e entrega do desafio, deves fazer um *fork* do repositório base do projeto ou criar um novo e atribuir-lhe o seguinte nome(pode usar o repo que o docente lhe forneceu, garantindo que é privado):

tdw-mp1-[primeironome]-[ultimonome]

Todo o trabalho deve ser desenvolvido nesse *branch*, e qualquer outro repositório utilizado (e.g. para fazer uso do GitLab) deve ser partilhado com os docentes.

A integração do Contentful pode ser feita através de uma conta criada ou da conta do docente. Para o Netlify pode usar as seguintes credenciais:

https://app.netlify.com/

user: jpmca@ua.pt

pass: 6Bv*J3@Dym3oNL%FkBg

Sobre a replicação da *pipeline* de GitHub numa outra plataforma de CI/CD - é expectável que não haja paridade completa entre *features*, e tendo isso em conta a adaptação de uma para a outra não precisa de ser 1:1.

Do ponto de vista técnico, serão valorizados aspetos como:

- A opção por soluções desenvolvidas pelo estudante, em detrimento de soluções copiadas de páginas Web ou do ChatGPT;
- Utilização apropriada do git (i.e., utilização de commits para adição incremental de funcionalidades, ao invés de um *big bang* commit no final do projeto);
- Utilização apropriada de *merge requests* para integrar código na *branch* principal do repositório;
- Implementação de limitações ao *merge* de código quando a *pipeline* estiver a falhar;
- Implementação de limitações ao *deploy* quando algum dos paços da *pipeline* estiver a falhar;
- Utilização segura de credenciais secretas do projeto;
- Otimização do tempo de execução da *pipeline*;
- Implementação de *features* específicas da plataforma de CI/CD escolhida.

REGRAS DE AVALIAÇÃO

As regras de avaliação são as seguintes:

- O trabalho é considerado entregue até às 23:59h do dia 22 de outubro, assumindo a existência de um repositório de GitHub com o nome mencionado na página anterior;
- Qualquer URL ou credencial adicionais utilizados para a concretização do desafio devem ser partilhados com os docentes aquando da entrega – via relatório;
- O trabalho será apresentado no dia 23 de outubro;
- A apresentação deve ser realizada a partir da versão publicada no dia 22 de outubro;
- A avaliação terá em conta os seguintes parâmetros:
    - 40% - Resultado apresentado do desafio desenvolvido no GitHub;
    - 30% - Resultado apresentado do desafio desenvolvido noutra plataforma de CI/CD;
    - 10% - Apresentação e defesa pública;
    - 20% - Relatório.

Os resultados do trabalho devem ser descritos num relatório a entregar até às 23:59 do dia 22 de outubro com o máximo de 1500 palavras. Sugere-se que sejam abordados os seguintes tópicos:

- Identificação d@ estudante e URL do(s) repositório(s) do projeto
- Apresentação do desafio e da plataforma de CI/CD escolhida
- Implementação técnica
- Diferenças observadas entre plataformas
- Principais desafios e obstáculos não ultrapassados
- Principais aprendizagens
- Conclusões/Reflexão crítica

Bom trabalho!

David Oliveira, Carlos Santos