



## 2ª Avaliação

O *Astronomy Picture Of the Day*, ou simplesmente APOD, é um dos serviços abertos mais populares da NASA (*National Aeronautics and Space Administration*), a famosa agência espacial dos Estados Unidos. Trata-se de uma API de acesso público (*Open API*), que estrutura e disponibiliza imagens espaciais e seus metadados para que possam ser utilizados por outros aplicativos.



Como base nesta API, desenvolva um aplicativo em Flutter que implemente os seguintes requisitos e funcionalidades:

### REQUISITOS E FUNCIONALIDADES:

**Cada equipe é livre para definir o *layout* do aplicativo, o qual será considerado no momento da avaliação. Interfaces mais bem elaboradas, com melhor *design* e funcionalidades serão mais bem avaliadas.**

1. A tela principal do aplicativo deverá exibir um conjunto N imagens disponibilizadas pela API.
2. O número de imagens exibidas na tela principal ( $1 \leq N \leq 20$ ) é um parâmetro do sistema que poderá ser alterado pelo usuário durante a execução do aplicativo por meio do acesso a uma página de configurações.
3. Para cada imagem apresentada na tela principal deverão ser exibidos seu título e data.
4. Deverá haver uma opção visual para atualização das imagens na tela.
5. Adicionalmente, a tela principal também deverá implementar o recurso “*pull-to-refresh*” para selecionar um novo conjunto de imagens.
6. Ao tocar (*tap*) em uma imagem, o usuário deverá ser conduzido para uma nova tela de detalhe que apresente os seguintes metadados da imagem: imagem, título, data, explicação e copyright.
7. A imagem na tela de detalhes deverá ser expandida até o limite da resolução horizontal da tela, mantendo o aspecto original da imagem (sem distorção).

*Desafio: tente implementar suporte para a rotação da tela em disposição retrato e paisagem (este requisito é opcional e não será avaliado).*

8. Caso alguns dos campos (metadados) da imagem não sejam disponibilizados pela API, o aplicativo deve lidar adequadamente com este fato.
9. Dados longos devem ser processados adequadamente para que possam ser rolados na tela.  
*Atenção para como a rolagem é implementada, para que o usuário não precise de muitas iterações para visualizar todos os campos (metadados) da imagem.*
10. O tempo de espera entre uma requisição e o recebimento de dados da API deverá ser sinalizado pela exibição de um *spinner* na tela.
11. Todos os possíveis erros de comunicação com a API devem ser previstos pelo aplicativo e notificados adequadamente para o usuário por meio de mensagens não técnicas.
12. O aplicativo deverá possuir uma opção para exibir os dados dos membros da equipe (nova tela, janela, snackbar, etc).
13. A codificação do projeto deverá observar os princípios de estilo do Effective Dart (<https://dart.dev/guides/language/effective-dart/style>)
14. O projeto deverá ser disponibilizado em um repositório com acesso público do GitHub, com “*main*” como seu *branch* principal. Qualquer outro *branch* não será considerado.

## DOCUMENTAÇÃO DA API:

1. A partir do navegador, acesse [api.nasa.gov](https://api.nasa.gov)
2. Selecione a opção “Browse API”, depois selecione o primeiro item da lista
3. Para acessar a API é necessário informar uma chave de aplicativo (APP\_KEY), que pode ser gerada no menu principal do site [api.nasa.gov](https://api.nasa.gov)

## COMPOSIÇÃO DA EQUIPE:

1. A implementação poderá ser realizada em equipes de ATÉ 3 (três) membros;
2. Os autores de cada implementação poderão ser questionados sobre o código implementado, com o objetivo de comprovar a participação de cada membro na execução do projeto;
3. A comprovação de não participação no projeto por algum membro da equipe implicará em:
  - Sua exclusão da equipe, e atribuição de nota zero;
  - Penalidade de 30% na nota da equipe.

## INTRUÇÕES PARA REMESSA DO PROJETO:

1. Ao finalizar o projeto, remeter o *link* do repositório GitHub para o e-mail: [eyder@phb.uespi.br](mailto:eyder@phb.uespi.br) com o seguinte assunto: “PROG4 – AVAL2”, juntamente com os nomes dos membros da equipe no corpo do e-mail;
2. Prazo de entrega: 30/06/2022;
3. O projeto deverá ser compatível com a versão 3.0.0 (ou superior) do Flutter.

## CRITÉRIOS DE AVALIAÇÃO:

- Cada requisito/funcionalidade será avaliado individualmente e receberá uma das seguintes notas:

Nota	Grau de atendimento ao requisito/funcionalidade
0	Não atende
1 a 4	Atende parcialmente
5	Atende completamente

- A nota final será determinada pela média ponderada das notas de cada requisito/funcionalidade.