

Grupo: Mínimo 4 e máximo 6 alunos

- 1x Scrum Master
 - o Elaborar a estrutura analítica do projeto em no máximo uma página A4, resumida e no máximo 2 slides na apresentação
 - o Elaborar cronograma (Gráfico de Gantt), pode usar Excel, Google Sheets, Project ou semelhantes
 - o Atribuir tarefas e definir prazos
 - Agenda no Google Calendar com membros
 - o Projeto no Trello com membros
 - Pools: TODO, PLANNED, DOING, WAITING, DONE, MEETINGS, CANCEL, REFERENCES
 - o Elaborar planilha de custos
 - Exportar em PDF
 - o Elaborar apresentação em PDF
 - Seções: Capa, Equipe, Agenda (sumário), Estrutura Analítica do Projeto - EAP, Atribuição de Tarefas, Custos e investimentos, Cronograma com Gantt, Resumo do Trello, Fluxo de trabalho no repositório, Casos de Uso, Protótipos (mínimo 5 slides), Extratos do código (mínimo de 5 slides), Demonstração de uso (mínimo de 10 slides com telas), Lições aprendidas, Agradecimentos, Referências
- 1x Gerente de configuração
 - o Criação do repositório git
 - Nome: IonicNews
 - Público, contendo todos os membros
 - Arquivos base
 - .gitignore (projeto node)
 - CONTRIBUTING.md (Seções: Descrição do projeto, Como baixar o repositório (lista de comandos), Pré-requisitos, Como contribuir (lista de comandos e pull requests)
 - LICENSE (Creative Commons)
 - README.md (Deve ter ao menos 5 imagens, seções: Descrição do projeto, Como baixar o repositório, Pré-requisitos, Contribuindo, Autores, Licença)
 - TODO.md (Opcional, pois contém o que falta fazer)
 - o Estrutura de pastas básicas para o repositório, deve obedecer a seguinte grafia, e caso não tenha arquivo em alguma das pastas mencionadas abaixo, deixe nela um arquivo vazio chamado ".gitkeep":
 - ionicnewsapi
 - Estrutura de pastas do projeto API, caso implemente
 - ionicnewsapp
 - Estrutura de pastas do projeto Ionic

- ionicnewsdocs
 - Models
 - o classes
 - Diagramas de classes, pode usar qualquer aplicação, como o Astah, StarUML ou semelhantes, deve salvar o arquivo nesta pasta
 - o database
 - Recomendo utilizar o MySQL Workbench e criar diagramas. Deve salvar o arquivo nesta pasta
 - o mindmap
 - Recomendo utilizar o FreeMind ou FreePlane, deve salvar o arquivo nesta pasta
 - o Mockup
 - Recomendo utilizar o Evolus Pencil, Canva ou qualquer ferramenta grátis, deve salvar o arquivo nesta pasta
 - Sql
 - o Eventuais arquivos sql que sejam necessários ao projeto
 - UseCase
 - o Arquivos de caso de uso, edite em qualquer ferramenta e salve aqui ao menos o arquivo em pdf.
- o Definição do fluxo de trabalho no repositório, desenvolver em um Branch e depois fazer o merge para o main, pode ser um Branch por semana ou por desenvolvedor, fica à sua escolha. Pode fazer um fluxograma identificando o fluxo de trabalho e exportar em png ou pdf, esse fluxo deve constar no README.md
- o Definição do tutorial para fluxo de trabalho, lista de comandos para todas as ações git que forem utilizadas durante o desenvolvimento do projeto. Exportar em pdf e referenciar o tutorial nos arquivos README.md e CONTRIBUTING.md
- o Definição de ferramentas, bibliotecas, frameworks e dispositivos para desenvolver o projeto, comandos de instalação e configuração. Exportar em pdf e referenciar nos arquivos README.md e CONTRIBUTING.md
- 2x Documentador
 - o Elaboração dos casos de uso, de forma simples e deverá ser exportado em pdf. Um modelo interessante é utilizar tabela no Word para colocar cada seção em uma linha. Pode contar com as seções abaixo, busque também por UseCase Template.
 - ID: UC001
 - Título: Realizar Login no Sistema
 - Ator principal: Usuário

- Objetivo: Permitir que o usuário entre no sistema utilizando credenciais válidas.
- Pré-condição: O usuário deve possuir uma conta no sistema.
- Pós-condição: O usuário terá acesso às funcionalidades restritas do sistema.
- Diagrama: Um fluxograma ou diagrama ou mapa mental simples
- Fluxo principal (quando tudo dá certo)
- Fluxo opcional (e.g.: Esqueci minha senha)
- Fluxo alternativo (quando algo dá errado)
- Tratamento de exceções (quando ocorre falhas que não dependem do sistema, ou quando o usuário erra muito a senha)
- o (Opcional) Elaboração dos casos de teste
- o Elaboração dos protótipos, pode utilizar canva, Pencil, Draw.io ou assemelhados. Exportar em png ou pdf
- o Elaboração dos diagramas UML. Exportar em png ou pdf.
- o (Opcional) Elaboração dos diagramas ER. Fazer pelo MySQL Workbench e exportar em png ou pdf. Manter no git
- 2x Desenvolvedor
 - o Codificação da solução

--

Descrição do Projeto: Desenvolvimento de Aplicativo Móvel com Ionic e Consumo de APIs REST

Nome do aplicativo: ionicnewsapp

Título: App de Notícias Personalizadas com Ionic

Objetivo:

Desenvolver um aplicativo móvel para a plataforma Android, utilizando o framework Ionic, que consuma dados de APIs REST externas. O objetivo é criar uma aplicação de notícias personalizadas, onde o usuário pode buscar e visualizar artigos de diversas categorias (tecnologia, esportes, entretenimento etc.), provenientes de uma API de notícias pública.

Funcionalidades Principais:

1. Autenticação de Usuário:

- o Implementação de um sistema de login e cadastro simples, utilizando serviços de autenticação com Firebase ou JSON Web Token (JWT), para que os usuários possam criar contas e personalizar sua experiência no app.

2. Integração com API de Notícias:

- o Consumo de uma API REST externa, como a NewsAPI (<https://newsapi.org/>), para buscar notícias em tempo

real com base em categorias selecionadas pelos usuários.

3. Interface de Usuário Responsiva:

- o Utilização de componentes do Ionic para criar uma interface de usuário limpa e responsiva, adaptada para diferentes tamanhos de tela.
- o Navegação entre telas (categorias de notícias, detalhes da notícia, tela de perfil etc.) utilizando o roteamento nativo do Ionic.

4. Filtros Personalizados:

- o O usuário poderá selecionar categorias de interesse (ex: esportes, tecnologia, negócios) e definir preferências para a exibição de notícias com base nesses filtros.

5. Detalhamento de Notícias:

- o Exibição de uma lista de notícias com título, imagem e uma breve descrição. Ao selecionar uma notícia, o usuário pode visualizar o conteúdo completo em uma nova tela.

6. Favoritos:

- o O usuário poderá salvar suas notícias favoritas para consultar mais tarde. Essa funcionalidade pode ser implementada armazenando os dados no armazenamento local do dispositivo ou em uma base de dados externa (Firebase, por exemplo).

7. Notificações Push:

- o Implementação de notificações push para alertar os usuários sobre novas notícias nas categorias de interesse.

8. Cache e Performance:

- o Utilizar o armazenamento local (Local Storage ou SQLite) para armazenar os dados das notícias e reduzir a necessidade de requisições constantes à API, melhorando a performance do aplicativo.

Requisitos Técnicos:

- **Framework:** Ionic
- **Linguagem:** TypeScript
- **Backend:** Firebase ou qualquer outro serviço de backend para autenticação e armazenamento de dados (opcional)
- **API de Notícias:** NewsAPI ou qualquer outra API pública de notícias

- **Notificações:** Capacitor Push Notifications
- **Armazenamento Local:** Local Storage ou SQLite

Critérios de Avaliação:

1. **Funcionalidade:** O aplicativo deve ser capaz de consumir a API externa e exibir as informações de maneira correta e eficiente.
2. **Interface:** A interface do usuário deve ser intuitiva e responsiva.
3. **Usabilidade:** O aplicativo deve permitir uma navegação fluida e fácil compreensão por parte dos usuários.
4. **Integração de API:** As requisições REST devem ser implementadas corretamente, com tratamento de erros e respostas da API.
5. **Autenticação e Segurança:** Se a autenticação for implementada, ela deve ser segura e eficiente.
6. **Armazenamento Local e Offline:** O cache de dados locais deve funcionar corretamente para oferecer uma boa experiência offline.

Prazo:

O desenvolvimento deverá ser concluído ao longo do semestre, com entregas parciais para acompanhamento e ajustes. Ao final, o projeto será apresentado com uma demonstração funcional e a entrega do código-fonte para avaliação.