

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

Gabriel de Moraes Ruiz	SP3046893
Grazielli Lima Berti	SP3046966
Gustavo Lourenço de Freitas	SP3049566
João Vitor Silva Bispo	SP3052672
Tayna Rita de França Souza	SP3050173
Vinicius Almeida Soares	SP3046991
Viviane de Santana Queiroz	SP3053601

Relatório Semanal

São Paulo - SP - Brasil

09/05/2022

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

Gabriel de Moraes Ruiz	SP3046893
Grazielli Lima Berti	SP3046966
Gustavo Lourenço de Freitas	SP3049566
João Vitor Silva Bispo	SP3052672
Tayna Rita de França Souza	SP3050173
Vinicius Almeida Soares	SP3046991
Viviane de Santana Queiroz	SP3053601

Relatório Semanal

Relatório Semanal de desenvolvimento do
Projeto Integrado.

Professores: Jose Braz de Araujo
Marcelo Tavares de Santana

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

Tecnologia em Análise e Desenvolvimento de Sistemas

PI1A5 - Projeto Integrado I

São Paulo - SP - Brasil

09/05/2022

Sumário

1	REALIZAÇÃO DA PROVA DE CONCEITO DA ARQUITETURA DA APLICAÇÃO	3
1.1	Front-end:	3
1.2	Back-end:	3

1 Realização da prova de conceito da arquitetura da aplicação

1.1 Front-end:

Relacionado ao FrontEnd iniciamos o nosso setup Mobile com o Expo Bare Workflow um framework que facilita muito o desenvolvimento de Apps, pois é um conjunto de ferramentas e serviços construídos em torno de plataformas nativas em React Native que ajudam a desenvolver, construir, implantar e iterar rapidamente em aplicativos iOS, Android e web a partir da mesma base de código JavaScript/TypeScript. Esse tipo de inicialização foi necessária para ajudar na hora da integração com o backend, dessa forma foi realizado algumas instalações e dependências para iniciar a aplicação. Sendo elas como o TypeScript, ESLint, Babel, Prettier e o Native Base.

O Native Base vale a atenção nesse relatório, pois decidimos utilizá-lo porque disponibiliza rapidamente componentes comuns e reusáveis como os botões, textos e menus que vemos nos apps nativos de forma rápida e fácil, assim facilitando o nosso desenvolvimento.

Na parte de autenticação, decidimos utilizar o OAuth2 que nada mais é que um protocolo de autorização que permite que uma aplicação se autentique em outra. Essa decisão foi tomada conforme os nossos conhecimentos para aplicar o protocolo e uma maneira mais rápida e mais leve de autenticar os nossos usuários dentro do nosso aplicativo. Dessa maneira, quando o usuário solicita uma autenticação ocorre a validação dessa autenticação através do token que foi criado e configurado no CGP da Google que redireciona o usuário através de um deep link e este estará permitido de logar na nossa aplicação. Também foi implementado o modo de edição de perfil do usuário autenticado.

1.2 Back-end:

Foi fundado o projeto NodeJs com o framework NestJs junto às configurações necessárias para a utilização do TypeScript, adotado pelo time para garantir maior consistência entre as tipagens do código e facilitar a identificação dos retornos esperados. Para a análise estática, foram configurados ESLint e Prettier, linters com os quais a equipe está mais familiarizada. Além disso, ainda visando a padronização do projeto, foi escolhido utilizar o pacote NPM chamado Husky, que automatiza scripts de padronização de mensagens de commits semânticos os quais contribuem para a fluidez no desenvolvimento, pois incentivam commits menores e mais simples de serem interpretados, o que gera velocidade e facilidade para a localização, remoção, adição e refatoração de código. Para a execução

da API, imagens docker foram criadas e por fim, foi realizada a configuração de acesso autorizado ao banco de dados MongoDB.

Durante a integração com o Prisma, houve problemas relacionados a rede, mas os quais foram contornados. A equipe também implementou o workflow de CI/CD via Github Actions para o deploy automático no Heroku da API, o que contribui para que somente código validado por linters e testes chegue à produção, permitindo, assim, maior foco no desenvolvimento das funcionalidades. Por fim, a validação da integração entre API e Banco de Dados ocorreu por meio da implementação do CRUD de usuário, o qual poderá se cadastrar na plataforma com sua da Google após o login social com OAuth. O usuário pode editar as informações de nome e biografia presentes em seu perfil. Por fim, para validar a integração com o Cloudinary, foi implementado o upload de imagem de perfil do usuário.