

PROTOTYPING

E QUE COMECEM OS **PREPARATIVOS**

HENRIQUE RUIZ POYATOS NETO



LISTA DE FIGURAS

Figura 1 – Listagem de recebimentos - Protótipo no Figma.....	11
---	----



LISTA DE QUADROS

Quadro 1 – Requisitos funcionais e não funcionais da Fintech	5
Quadro 2 – Requisitos funcionais e não funcionais da Fintech, versão 2	6
Quadro 3 – Requisitos funcionais e não funcionais da Fintech, versão 3	7
Quadro 4 – Requisitos funcionais e não funcionais da Fintech, versão 4	8
Quadro 5 – Requisitos funcionais e não funcionais da Fintech, versão 5	9

EMSE

SUMÁRIO

1 E QUE COMECEM OS PREPARATIVOS!.....	5
1.1 Introdução	5
1.2 Mais funcionalidades para a Fintech	5
1.3 Detalhando as funcionalidades para o sistema Fintech	9
1.3.1 Detalhes sobre dados do usuário	10
1.3.2 Detalhes sobre “Alterar senha do usuário”	10
1.3.3 Detalhes sobre “Exibir <i>dashboard</i> do usuário”	10
1.3.4 Detalhes sobre os recebimentos do usuário	11
1.3.5 Detalhes sobre um valor gasto	12
1.3.6 Detalhes sobre investimentos	12
1.3.7 Detalhes sobre objetivo financeiro	12
1.4 Vamos lá?	13
REFERÊNCIAS.....	14
GLOSSÁRIO	15

1 E QUE COMECEM OS PREPARATIVOS!

1.1 Introdução

Olá, seja bem-vindo(a) ao segundo módulo do projeto Fintech!

Após entenderem como um software e um projeto funcionam (no primeiro módulo), o grande objetivo desta nova fase é prototipar o sistema. Você avançará com a documentação do projeto do sistema por meio dos diagramas de casos de uso. Aprenderemos mais sobre o gerenciamento de tempo e custos e você deverá realizar o protótipo do sistema Fintech, na forma de *wireframes* ou desenhos para definir como as telas do sistema serão!

Vamos começar? Não temos tempo a perder!

1.2 Mais funcionalidades para a Fintech

Seu trabalho levantando os requisitos de sistema da Fintech no módulo 1 foi bem-sucedido!

Até o momento, você tem as seguintes informações:

Requisitos funcionais	Requisitos não funcionais
Alterar dados do usuário.	A aplicação deve ser construída utilizando uma API Rest.
Adicionar valor das receitas do usuário.	A interface deve ser uma aplicação Web que consome a API desenvolvida.
Adicionar valor de um gasto efetuado.	O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, JavaScript, JQuery, Bootstrap, ReactJS e Java.
Adicionar um valor investido.	O sistema deve armazenar as informações em um banco de dados relacional Oracle.
Adicionar um objetivo financeiro.	O sistema deve funcionar adequadamente nas versões atualizadas dos navegadores web Google Chrome, Mozilla Firefox e Microsoft Edge.

Quadro 1 – Requisitos funcionais e não funcionais da Fintech

Fonte: FIAP (2021)

Até então, nada de novo, não é mesmo? Apenas organizamos o que tínhamos. Pois bem, em uma nova rodada com o patrocinador do projeto, você descobriu:

“O usuário precisa ter uma maneira de se cadastrar no sistema. Além disso, não convém que outros usuários enxerguem suas informações, não é mesmo? Sendo assim, ele deve se identificar no sistema em todas as ocasiões que quiser utilizá-lo.”

Ficou acertado entre vocês que o “autenticar-se no sistema” será feito com o endereço de e-mail do usuário e uma senha. Uma necessidade de segurança como essa compreende, entre outras coisas, novas funcionalidades, por exemplo: “Esqueci minha senha”, “Alterar a senha” e “Sair do sistema”.

Nossa lista de requisitos ganha novas funcionalidades:

Requisitos funcionais	Requisitos não funcionais
Cadastrar usuário no sistema.	A aplicação deve ser construída utilizando uma API Rest.
Alterar dados do usuário.	A interface deve ser uma aplicação Web que consome a API desenvolvida.
Alterar senha do usuário.	O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, JavaScript, JQuery, Bootstrap, ReactJS e Java.
Relembrar senha do usuário.	O sistema deve armazenar as informações em um banco de dados relacional Oracle.
Identificar-se no sistema.	O sistema deve funcionar adequadamente nas versões atualizadas dos navegadores web Google Chrome, Mozilla Firefox e Microsoft Edge.
Sair do sistema.	
Adicionar valor das receitas do usuário.	
Adicionar valor de um gasto efetuado.	
Adicionar um valor investido.	
Adicionar um objetivo financeiro.	

Quadro 2 – Requisitos funcionais e não funcionais da Fintech, versão 2
Fonte: FIAP (2021)

Sobre adicionar informações no sistema, também foi dito:

“Além de adicionar receitas, gastos efetuados, investimentos e objetivos financeiros, o usuário precisa ter a possibilidade de alterar essas informações. E se ele as insere errado por alguma razão? Além disso, ele precisa ter a opção de removê-las do sistema por alguma razão.”

Mensagem recebida:

Requisitos funcionais	Requisitos não funcionais
Cadastrar usuário no sistema.	A aplicação deve ser construída utilizando uma API Rest.
Alterar dados do usuário.	A interface deve ser uma aplicação Web que consome a API desenvolvida.
Alterar senha do usuário.	O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, JavaScript, JQuery, Bootstrap, ReactJS e Java.
Relembrar senha do usuário.	O sistema deve armazenar as informações em um banco de dados relacional Oracle.
Identificar-se no sistema.	O sistema deve funcionar adequadamente nas versões atualizadas dos navegadores web Google Chrome, Mozilla Firefox e Microsoft Edge.
Sair do sistema.	
Adicionar valor das receitas do usuário.	
Alterar valor das receitas do usuário.	
Remover valor das receitas do usuário.	
Adicionar valor de um gasto efetuado.	
Alterar valor de um gasto efetuado.	
Remover valor de um gasto efetuado.	
Adicionar um valor investido.	
Alterar um valor investido.	
Remover um valor investido.	
Adicionar um objetivo financeiro.	
Alterar um objetivo financeiro.	
Remover um objetivo financeiro.	

Quadro 3 – Requisitos funcionais e não funcionais da Fintech, versão 3

Fonte: FIAP (2021)

“Precisamos de uma forma para exibir no sistema os dados desse usuário, todas as receitas recebidas, todos os gastos efetuados e os investimentos realizados...”

Sim, é verdade:

Requisitos funcionais	Requisitos não funcionais
Cadastrar usuário no sistema.	A aplicação deve ser construída utilizando uma API Rest.
Exibir dados do usuário.	A interface deve ser uma aplicação Web que consome a API desenvolvida.
Alterar dados do usuário.	O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, JavaScript, JQuery, Bootstrap, ReactJS e Java.
Alterar senha do usuário.	O sistema deve armazenar as informações em um banco de dados relacional Oracle.
Relembrar senha do usuário.	O sistema deve funcionar adequadamente nas versões atualizadas dos navegadores web Google Chrome, Mozilla Firefox e Microsoft Edge.
Identificar-se no sistema.	
Sair do sistema.	
Exibir lista valores das receitas do usuário.	
Adicionar valor das receitas do usuário.	
Alterar valor das receitas do usuário.	
Remover valor das receitas do usuário.	
Listar os valores gastos pelo usuário.	
Adicionar valor de um gasto efetuado.	
Alterar valor de um gasto efetuado.	
Remover valor de um gasto efetuado.	
Listar os valores investidos pelo usuário.	
Adicionar um valor investido.	
Alterar um valor investido.	
Remover um valor investido.	
Listar os objetivos financeiros.	
Adicionar um objetivo financeiro.	
Alterar um objetivo financeiro.	
Remover um objetivo financeiro.	

Quadro 4 – Requisitos funcionais e não funcionais da Fintech, versão 4

Fonte: FIAP (2021)

A propósito, temos um jargão técnico muito utilizado para esse conjunto tão comum de funcionalidades (cadastrar, exibir dados, alterar e remover). Costumamos chamar de **CRUD**, que é um acrônimo em inglês para **Create**, **Retrieve**, **Update** and **Delete**, as quatro operações citadas, respectivamente.

Por fim (será realmente o fim? Com um cliente, nunca se sabe!), o patrocinador solicitou uma tela inicial depois de o usuário se autenticar, que exibe algumas informações úteis. Entre outras sumarizadas, temos: saldo mensal do usuário, a evolução dos investimentos e a porcentagem de progresso até o objetivo financeiro. É comum chamar esse tipo de funcionalidade de *dashboard* (painel de controle).

Nossa lista de requisitos fica desta maneira:

Requisitos funcionais	Requisitos não funcionais
Cadastrar usuário no sistema.	A aplicação deve ser construída utilizando uma API Rest.
Exibir dados do usuário.	A interface deve ser uma aplicação Web que consome a API desenvolvida.
Alterar dados do usuário.	O sistema deve ser desenvolvido utilizando as linguagens HTML, CSS, JavaScript, JQuery, Bootstrap, ReactJS e Java.
Alterar senha do usuário.	O sistema deve armazenar as informações em um banco de dados relacional Oracle.
Relembrar senha do usuário.	O sistema deve funcionar adequadamente nas versões atualizadas dos navegadores web Google Chrome, Mozilla Firefox e Microsoft Edge.
Identificar-se no sistema.	
Exibir Dashboard do sistema.	
Sair do sistema.	
Exibir lista valores das receitas do usuário.	
Adicionar valor das receitas do usuário.	
Alterar valor das receitas do usuário.	
Remover valor das receitas do usuário.	
Listar os valores gastos pelo usuário.	
Adicionar valor de um gasto efetuado.	
Alterar valor de um gasto efetuado.	
Remover valor de um gasto efetuado.	
Listar os valores investidos pelo usuário.	
Adicionar um valor investido.	
Alterar um valor investido.	
Remover um valor investido.	
Listar os objetivos financeiros.	
Adicionar um objetivo financeiro.	
Alterar um objetivo financeiro.	
Remover um objetivo financeiro.	

Quadro 5 – Requisitos funcionais e não funcionais da Fintech, versão 5

Fonte: FIAP (2021)

Sim, há muito a fazer! Mas, bem... nunca lhe dissemos que seria fácil, não é mesmo?

1.3 Detalhando as funcionalidades para o sistema Fintech

Conforme dissemos, durante este módulo do sistema Fintech, você terá que descrever o funcionamento do sistema por meio dos casos de uso. Terá ainda que

desenhar as telas do sistema (em um caráter de protótipo) e começar a organizar as tarefas, seja em um cronograma ou no quadro de tarefas. Faz-se necessário, portanto, saber um pouco mais sobre os requisitos aqui apresentados.

1.3.1 Detalhes sobre dados do usuário

As informações solicitadas para cadastrar o usuário, as quais serão exibidas (em uma verdadeira página de perfil) e poderão ser alteradas, devem ser as seguintes:

- Nome completo.
- Data de nascimento.
- Gênero (masculino/feminino).
- E-mail de contato.
- Senha.

1.3.2 Detalhes sobre “Alterar senha do usuário”

Como de praxe, quando o usuário deseja alterar sua senha, o sistema deverá solicitar a antiga senha e a nova, duas vezes. A senha só será alterada caso o usuário acerte a senha antiga.

Tanto para o cadastro inicial quanto para a alteração, a senha do usuário deverá ter no mínimo seis caracteres, combinando letras e números.

1.3.3 Detalhes sobre “Exibir *dashboard* do usuário”

Em reuniões com as partes interessadas, ficou determinado que a tela inicial de *dashboard* (logo após a autenticação com sucesso do usuário) terá as seguintes informações:

- Nome do usuário.
- Valor total dos recebimentos do usuário do mês corrente.
- Valor total dos gastos do usuário do mês corrente.

- Último valor gasto com a descrição e data em que isso aconteceu.
- Valor total dos investimentos do usuário.
- Saldo do mês corrente (valor total dos recebimentos menos o valor total dos gastos).

1.3.4 Detalhes sobre os recebimentos do usuário

Ao inserir ou alterar essa informação, o usuário deverá informar o valor (em reais), a data em que o valor foi recebido e uma descrição. O sistema pode trazer a data atual previamente preenchida para a comodidade do usuário, embora possa alterá-la e registrar um valor de receita retroativo.

A partir da lista de recebimentos (que deve ser classificada dos valores com datas mais recentes para as datas mais antigas), têm-se as possibilidades de cadastrar, alterar e remover a informação. Cada informação deverá vir acompanhada de seu próprio botão/hyperlink de alteração e remoção, como você pode ver na Figura “Listagem de recebimentos – Protótipo no Figma”:



Valor	Data	Descrição	
R\$ 2500,00	05/01/2022	Salário	 Editar  Excluir
R\$ 500,00	15/01/2022	Freelance	 Editar  Excluir
R\$ 2500,00	05/02/2022	Salário	 Editar  Excluir
R\$ 100,00	25/02/2022	Dividendos	 Editar  Excluir

Figura 1 – Listagem de recebimentos – Protótipo no Figma
Fonte: Elaborado pelo autor (2021)

Esse padrão de navegação **CRUD** é válido para qualquer informação do sistema.

1.3.5 Detalhes sobre um valor gasto

Sobre um valor gasto, haverá uma listagem similar à de receitas do usuário, que administrará as seguintes informações:

- Categoria: valor predefinido (aluguel, alimentação, academia, transporte, educação etc.).
- Descrição: uma breve descrição opcional do que foi gasto.
- Valor: valor em reais do gasto efetuado.
- Data e horário de quando o gasto foi realizado.

1.3.6 Detalhes sobre investimentos

Sobre os investimentos do usuário, haverá uma listagem similar à de receitas, que terá as seguintes informações:

- Tipo: valor predefinido (CDB, Tesouro Direto, Poupança, FIs, Ações etc.).
- Nome da aplicação financeira.
- Nome do banco ou da corretora de valores, valor predefinido.
- Valor da aplicação.
- Data de quando o investimento foi realizado.
- Data do vencimento do investimento, se aplicável.

1.3.7 Detalhes sobre objetivo financeiro

Os objetivos financeiros também deverão ser listados, sendo necessárias as seguintes informações:

- Nome do objetivo: um valor para nomear o objetivo (por exemplo: casa própria, próxima viagem, carro novo etc.).

- Descrição: uma breve descrição opcional do objetivo financeiro.
- Valor: quantidade de dinheiro necessária para atingir o objetivo financeiro.
- Data em que deseja atingir o objetivo financeiro.

1.4 Vamos lá?

Você está preparado para mais um módulo de nosso projeto Fintech?

Temos muito a fazer!

FINTECH

REFERÊNCIAS

INVESTIDOR.GOV. **Fundo de Investimento Imobiliário**. 2021. Disponível em: <https://www.investidor.gov.br/menu/Menu_Investidor/valores_mobiliarios/fundos_imobiliarios.html>. Acesso em: 24 fev. 2021.

EXEMPLO

GLOSSÁRIO

FIIs	Fundos imobiliários, são fundos de investimento destinados à aplicação em empreendimentos imobiliários.
CDB	Certificado de depósito bancário. É um título de renda fixa, representativo de depósitos a prazo, utilizado pelos bancos comerciais como mecanismo de captação de recursos.

EXEMPLO