CardapiOn.IFPI

História 1 - Interface de exibição do cardápio

1) Fixar o escopo

História de usuário: Como aluno, quero visualizar o cardápio semanal do Restaurante Universitário para planejar minhas refeições de forma prática e organizada.

Extração:

- **Ator:** estudante
- **Ação:** visualizar cardápio
- **Benefício:** planejar refeições

Objetivo: permitir com que os alunos visualizem o cardápio de uma forma clara e objetiva.

2) Mapear requisitos visuais a partir dos critérios de aceitação

Critérios de Aceitação:

- O cardápio deve ser exibido de forma clara e separada por dias da semana.
- Deve mostrar as opções de almoço e jantar.
- O layout deve ser simples e fácil de entender.
- Em dias sem cardápio cadastrado, deve aparecer uma mensagem de ‘cardápio indisponível’

3) Derivar fluxos de interface a partir dos cenários

Cenário 1 – Cardápio disponível (sucesso)

- Abrir tela → ver cardápio da semana atual

Cenário 2 – Navegação entre semanas

- Clicar “Próxima semana” → cardápio mostra semana seguinte se disponível
- Clicar “Semana anterior” → cardápio do passado aparece

Cenário 3 – Dia sem cardápio disponível

- Abrir tela → cardápio de um dia aparece como “indisponível”

Cenário 4 – Erro no carregamento

- Mostrar mensagem “não foi possível carregar cardápio” com botão de tentar novamente

Cenário 5 – Cardápio ainda não registrado

- Mostrar ,no lugar da tabela, a mensagem de ‘cardápio ainda não registrado’.

4) Definir a anatomia da tela (layout)

- **Título:** 'Cardápio semanal'
- tabela de 5 colunas com informações sobre do almoço e jantar de cada dia da semana
- **botões:** selecionar semana
- mensagem de indisponibilidade do sistema

5) Preparar estados da UI

- **estado normal**
- **estado com algum dia sem aula**
- **estado de carregamento do sistema**
- **estado de indisponibilidade do cardápio**
- **estado de cardápios anteriores**

História 2 - Interface de login do usuário

1) Fixar o escopo

História de usuário: Como aluno, quero ter acesso ao sistema usando minhas credenciais institucionais, para entrar de forma segura e personalizada.

Extração:

- **Ator:** estudante
- **Ação:** acessar sistema com credenciais institucionais
- **Benefício:** privacidade e segurança

Objetivo: Permitir com que o aluno faça login com suas credenciais em uma interface simples e objetiva.

2) Mapear requisitos visuais a partir dos critérios de aceitação

Critérios de Aceitação:

- permitir inserir usuário e senha.
- validar as credenciais do usuário
- acessar o sistema se as credenciais estiverem corretas
- oferecer opção de recuperar senha
- manter sistema ativo caso o usuário marque a opção ‘lembre de mim’

3) Derivar fluxos de interface a partir dos cenários

Cenário 1 – Login validado (sucesso)

- abrir tela → preencher credenciais corretas → clicar em entrar→ acessar o sistema

Cenário 2 – Login falho

- abrir tela → preencher credenciais erradas → clicar em entrar→ aparecer mensagem de login falho

Cenário 3 – Campos vazias

- Abrir tela → clicar em entrar → aparecer mensagem de preencher os campos antes de entrar

Cenário 4 – Esqueci a senha

- Abrir tela → clicar em esqueci a senha → enviar um email de confirmação de identidade

4) Definir a anatomia da tela (layout)

- **Título:** 'Login/Cadastro'.
- **campos:** usuário, senha.
- **botões:** entrar, esqueci a senha.
- **mensagens:** indisponibilidade do sistema, carregando sistema, login falho.

5) Preparar estados da UI

- **estado normal**
- **estado preenchido**
- **estado de sucesso no login**
- **estado de erro no login**
- **estado de carregamento do site**

História 3 – Interface de exibição de avisos

1) Fixar o escopo

História de usuário: Como administrador ou funcionário, quero enviar avisos aos alunos, para informar sobre mudanças no cardápio e horários do restaurante aos alunos.

Extração:

- **Ator:** Administrador ou funcionário

- **Ação:** enviar avisos
- **Benefício:** manter alunos informados sobre mudanças no cardápio

Objetivo: Permitir que o administrador/funcionário crie, edite e envie avisos aos alunos do restaurante universitário.

2) Mapear requisitos visuais a partir dos critérios de aceitação

Critérios de Aceitação:

- Permitir escrever o conteúdo do aviso → Campo editor simples.
- Permitir definir título do aviso → Campo “Título do aviso”.
- Permitir escolher tipo do aviso → Ex.: Cardápio, Horário, Manutenção, Funcionamento.
- Enviar aviso para todos os alunos → Botão “Enviar Aviso”.
- Exibir mensagem de sucesso → mensagem: “Aviso enviado com sucesso”.
- Exibir erros de campo obrigatório → título ou conteúdo vazio → mensagem: “Preencha os campos obrigatórios”.
- Permitir visualizar avisos já enviados → Lista de histórico de avisos.
- Permitir editar ou excluir aviso anterior → Ícones de edição e exclusão.

3) Derivar fluxos de interface a partir dos cenários

Cenário 1 – Envio bem-sucedido

- Administrador acessa tela “Enviar Aviso”→ Preenche título, tipo e conteúdo → Clica em Enviar → Sistema valida→ Mostra mensagem de sucesso → Aviso aparece na lista de avisos enviados

Cenário 2 – Campos obrigatórios vazios

- Usuário tenta enviar aviso sem preencher algo obrigatório → Sistema bloqueia ação → Exibe mensagem:

“Título é obrigatório”

“conteúdo obrigatório”

Cenário 3 – Edição de aviso

- Usuário clica no ícone “Editar” de um aviso antigo → Abre tela de edição com campos preenchidos → Usuário atualiza o texto → Clica Salvar alterações → Exibe mensagem “Aviso atualizado com sucesso”

Cenário 4 – Exclusão de aviso

- Usuário clica no ícone de excluir → Sistema exibe modal de confirmação → Ao confirmar, o aviso é removido → mensagem: “Aviso excluído com sucesso”

4) Definir a anatomia da tela (layout)

- **Título:** “Avisos”
- **Campos:** título do aviso, conteúdo do aviso
- **Seletor:** “categoria: cardápio, horário, mudança emergencial, outro“
- **Botões:** enviar aviso, cancelar envio, editar aviso**,** excluir aviso
- **Mensagens:** aviso enviado com sucesso, envio cancelado com sucesso, aviso excluído

5) Preparar estados da UI