

1) Lanchonete (Mega Lanches)

RF

- **RF1:** Exibir cardápio com múltiplos itens, cada um contendo imagem, nome e preço.
- **RF2:** Permitir adicionar item ao pedido através de botão “Adicionar”.
- **RF3:** Manter carrinho com itens adicionados, acumulando quantidade por item quando o mesmo é adicionado mais de uma vez.
- **RF4:** Exibir total do pedido (soma de subtotais).
- **RF5:** Persistir carrinho no localStorage (recupera estado ao recarregar a página).
- **RF6:** Finalizar Pedido limpa o carrinho e atualiza a interface.
- **RF7:** Badge de quantidade no ícone do carrinho refletindo o número total de itens.
- **RF8:** Abrir carrinho em offcanvas com listagem dos itens.
- **RF9:** Banner/carrossel rotativo com intervalo configurado.
- **RF10:** Geração dos cards do cardápio dinamicamente a partir de uma lista (quando essa opção está ativada).
- **RF11:** Feedback de adição (ex.: alerta simples) após adicionar item.

RNF

- **RNF1:** Layout responsivo via Bootstrap (grid e componentes).
- **RNF2:** Imagens do banner configuradas para não distorcer (ex.: `object-fit: cover`) e suporte a `srcset`.
- **RNF3:** Uso de delegação de eventos para evitar listeners duplicados.
- **RNF4:** Organização de arquivos front-end separando **HTML/CSS/JS**.

RN

- **RN1:** Acúmulo por item — adicionar um item já existente incrementa a quantidade em vez de criar nova linha no carrinho.
 - **RN2:** Total do pedido é a soma de preço × quantidade de cada item.
 - **RN3:** Limpar carrinho ao finalizar pedido remove dados do localStorage e reseta contadores/visual.
 - **RN4:** Badge deve refletir o total de unidades no carrinho (não a quantidade de tipos de produto).
-

2) Biblioteca Virtual (Leitura Livre)

RF

- **RF1:** Exibir catálogo (Home) com livros contendo capa, título e autor.
- **RF2:** Adicionar à Estante um livro do catálogo.
- **RF3:** Renderizar Estante (página dedicada) listando apenas os livros adicionados.
- **RF4:** Persistir estante no localStorage e restaurar ao recarregar.
- **RF5:** Remover livro da estante.
- **RF6:** Marcar “Lido” por livro (checkbox) e persistir esse estado.
- **RF7:** Contadores: exibir total de livros na estante e total de lidos.
- **RF8:** Filtro por gênero:
 - Home: filtra o catálogo pelo gênero selecionado.
 - Estante: filtra apenas os livros salvos cujo gênero corresponda.

RNF

- **RNF1:** Bootstrap para responsividade, cards e grid.

- **RNF2:** Map/Join para renderização em lote; DocumentFragment não é obrigatório no estado atual.
- **RNF3:** ES Modules com `<script type="module">` e execução via servidor local (ex.: Live Server), evitando erro de import.
- **RNF4:** Acessibilidade mínima: `alt` nas capas; IDs únicos para checkboxes; foco visual mantido pelos componentes Bootstrap.
- **RNF5:** Separação de responsabilidades:
 - `home.js` não depende de elementos exclusivos da estante e vice-versa (guards por `getElementById`).

RN

- **RN1:** Unicidade na Estante — um livro não pode ser adicionado duas vezes; ao tentar, mantém uma única entrada.
 - **RN2:** Estado “Lido” é por livro na estante e persistente entre sessões.
 - **RN3:** Remoção: ao remover um livro da estante, o item é excluído do storage e a UI é atualizada imediatamente.
 - **RN4:** Contadores: o total exibido reflete exatamente o tamanho da estante; o total de lidos conta somente os itens com `lido = true`.
-

3) Organizador de Tarefas (Agendoo)

RF

- **RF1.** Adicionar tarefa com título e data via formulário; só cria se ambos estiverem preenchidos e limpa o formulário após salvar.
- **RF2.** Listar tarefas com: checkbox de conclusão, título e data em badge e botão excluir.
- **RF3.** Concluir/Desconcluir tarefa pelo checkbox, refletindo imediatamente na interface e no armazenamento.
- **RF4.** Excluir tarefa pela ação do botão de lixeira, atualizando a lista e persistência.
- **RF5.** Filtrar tarefas por status (**todas / pendentes / concluídas**).

- **RF6.** Ordenar tarefas por data ou título.
- **RF7.** Persistir tarefas em localStorage (carrega na inicialização e salva a cada alteração).

RNF

- **RNF1.** UI responsiva com Bootstrap (grid, formulários, badges e utilitários).
- **RNF2.** Persistência local por JSON em localStorage sob a chave do app.
- **RNF3.** Tratamento de data “timezone-safe” (constrói `Date` a partir de ano/mês/dia para evitar deslocamento de fuso).
- **RNF4.** Tema claro/escuro aplicado via classes no `<body>` e ajustes de cores em títulos, inputs, rodapé, itens e ícones.
- **RNF5.** Placeholder e sombras compatíveis com tema escuro (CSS dedicado).

RN

- **RN1.** Obrigatoriedade de dados: uma tarefa só é criada se título (trimado) e data forem fornecidos.
- **RN2.** Conclusão altera apresentação: tarefa concluída fica opaca, com título tachado e badge “success”; tarefa não concluída usa badge claro no tema ativo.
- **RN3.** Ordenação:
 - Data em ordem crescente por comparação de `Date`; Título via `localeCompare`.
 - Observação: na inicialização, o item de exemplo presente no HTML é removido do DOM antes do uso (higienização da lista).