

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\main.py

```
# main.py

from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from core.configs import settings
from utils.scheduler import tarefa_diaria, scheduler
import uvicorn
from contextlib import asynccontextmanager
from api.v1.endpoints.categoria import router as categoria_router
from api.v1.endpoints.setor import router as setor_router
from api.v1.endpoints.usuario import router as usuario_router
from api.v1.endpoints.item import router as item_router
from api.v1.endpoints.retirada import router as retirada_router
from api.v1.endpoints.relatorios import router as relatorio_router
from api.v1.endpoints.alerta import router as alerta_router
from fastapi.staticfiles import StaticFiles
from frontend.routes.home import router as frontend_router
import mimetypes

# Importar o roteador WebSocket e o manager
from utils.websocket_endpoints import websocket_router, manager

# Forçar o tipo MIME para arquivos .js. Isso deve ser feito ANTES de StaticFiles ser montado.
mimetypes.add_type('application/javascript', '.js')
mimetypes.add_type('application/javascript', '.mjs') # Para módulos ES6 com extensão .mjs

@asynccontextmanager
async def lifespan(app: FastAPI):
    # Executado antes do app iniciar
    try:

        scheduler.add_job(tarefa_diaria, 'cron', hour=6) #todo dia às 6h
        scheduler.start()
        print("Scheduler iniciado com sucesso via lifespan.")
    except Exception as e:
        print("Erro ao iniciar scheduler via lifespan:", e)

    yield # app roda

    # Executado quando o app estiver encerrando
    scheduler.shutdown()
    print("Scheduler finalizado.")

app = FastAPI(
    title="Sistema de Gerenciamento de Almoxarifado",
    description="API para gerenciar o estoque e retirada de materiais",
    version="1.0.0",
    lifespan=lifespan
)

app.add_middleware(
    CORSMiddleware,
    allow_origins=["http://localhost:8082"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

# Incluindo os endpoints - Back-end
```

```

app.include_router(setor_router, prefix=settings.API_STR, tags=['Gerenciamento de Setores'])
app.include_router(categoria_router, prefix=settings.API_STR, tags=['Gerenciamento de Categorias'])
app.include_router(usuario_router, prefix=settings.API_STR, tags=['Gerenciamento de Usuários'])
app.include_router(item_router, prefix=settings.API_STR, tags=['Gerenciamento de Itens'])
app.include_router(retirada_router, prefix=settings.API_STR, tags=['Gerenciamento de Retiradas'])
app.include_router(relatorio_router, prefix=settings.API_STR, tags=['Geração de Relatórios de Itens'])
app.include_router(alerta_router, prefix=settings.API_STR, tags=['Gerenciamento de Alertas'])

#Incluir o roteador WebSocket
app.include_router(websocket_router, prefix=settings.API_STR) # Prefixo para o WebSocket

# Montar pasta de arquivos estáticos
app.mount("/static", StaticFiles(directory="frontend/static"), name="static")

# Incluindo os endpoints - Front-End
app.include_router(frontend_router)

if __name__ == "__main__":
    uvicorn.run("main:app", host="0.0.0.0", port=8082, reload=True)

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\routes\home.py

```

#frontend\routes\home.py

from fastapi import APIRouter, Request, Cookie
from fastapi.responses import HTMLResponse, RedirectResponse
from fastapi.templating import Jinja2Templates
from jwt import decode, PyJWTError
from core.configs import settings

router = APIRouter()
templates = Jinja2Templates(directory="frontend/templates")

@router.get("/", response_class=HTMLResponse)
async def home(request: Request, access_token: str | None = Cookie(None)):
    print("access_token recebido:", access_token)

    if access_token:
        try:
            payload = decode(
                access_token,
                settings.JWT_SECRET,
                algorithms=[settings.ALGORITHM]
            )
            tipo = payload.get("tipo_usuario")
            if tipo == 1:
                return RedirectResponse("/dashboardServidor", status_code=303)
            elif tipo == 2:
                return RedirectResponse("/dashboardAlmoxarifado", status_code=303)
            elif tipo == 3:
                return RedirectResponse("/dashboardDirecao", status_code=303)
        except PyJWTError:
            pass # Cookie inválido ou expirado

    return templates.TemplateResponse("index.html", {"request": request})

@router.get("/dashboardAlmoxarifado", response_class=HTMLResponse)
async def dashboard_almoxarifado(request: Request):
    return templates.TemplateResponse("dashboardAlmoxarifado.html", {"request": request})

```

```
@router.get("/dashboardServidor", response_class=HTMLResponse)
async def dashboard_servidor(request: Request):
    return templates.TemplateResponse("dashboardServidor.html", {"request": request})

@router.get("/dashboardDirecao", response_class=HTMLResponse)
async def dashboard_direcao(request: Request):
    return templates.TemplateResponse("dashboardDirecao.html", {"request": request})
```

```
@charset "UTF-8";/*!
 * Bootstrap v5.3.5 (https://getbootstrap.com/)
 * Copyright 2011-2025 The Bootstrap Authors
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 */:root,[data-bs-theme=light]{--bs-blue:#0d6efd;--bs-indigo:#6610f2;--bs-purple:#6f42c1;--bs-pink:#d63384;
/*# sourceMappingURL=bootstrap.min.css.map */
```

```

/* frontend\static\css\custom.css */

/* === Header superior === */

.custom-header {
    background-color: #0052cc;
    color: #fff;
}

.custom-header header-logo {
    height: 40px;
}

.custom-header .header-title {
    font-weight: bold;
    font-size: 1rem;
}

/* === Navbar === */

.custom-navbar {
    background-color: #0052cc;
}

/* Dropdown menus */

.navbar-nav .dropdown-menu {
    border-radius: 0.25rem;
    box-shadow: 0 0.5rem 1rem rgba(0,0,0,0.1);
}

/* Itens do dropdown */

.navbar-nav .dropdown-item { /* Corrigido de .navbar-nav.dropdown-item */
    padding: 0.5rem 1rem;
    font-size: 0.95rem;
}

```

```

}

.navbar-nav .dropdown-item:hover { /* Corrigido de .navbar-nav.dropdown-item:hover */
  background-color: #f8f9fa;
}

/* Links do navbar */

.nav-link {
  padding: 0.5rem 1rem;
}

.nav-link:hover, .nav-link:focus {
  background-color: rgba(255,255,255,0.1);
  border-radius: 0.25rem;
}

/* Ícones */

.navbar-nav .bi {
  font-size: 1.3rem;
  margin: 0 0.3rem;
  cursor: pointer;
}

/* Ajuste responsivo: evitar que o ícone de dropdown quebre linha */

.nav-item.dropdown {
  white-space: nowrap;
}

/* Estilos para o contêiner de toasts (alvo a classe Bootstrap diretamente) */
.toast-container {
  top: 1rem !important; /* Ajusta a distância do topo */
  right: 1rem !important; /* Ajusta a distância da direita */
  max-width: 350px !important; /* Define uma largura máxima para o contêiner de toasts */
  width: 90% !important; /* Garante que em telas menores, ele ocupe 90% da largura */
  padding: 0 !important; /* Remove padding duplicado se já houver no p-3 */
  margin: 0 !important;
}

.toast {
  width: 100% !important;
  word-wrap: break-word !important;
  box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.15) !important;
  margin-bottom: 0.5rem !important; /* Espaçamento entre toasts empilhados */
}

/* Ajusta o corpo do toast para não ter padding excessivo que o corte */
.toast-body {
  padding-right: 1.5rem !important; /* Garante espaço para o botão de fechar */
}

/* Estilos para os itens do dashboard de relatórios */

.report-item-style {
  background-color: #f8f9fa; /* Um tom claro para o fundo */
  border-radius: 0.5rem; /* Bordas arredondadas */
  margin-bottom: 0.5rem; /* Espaçamento entre os itens */
  transition: all 0.2s ease-in-out; /* Transição suave para hover */
  border: 1px solid #e9ecef; /* Uma borda sutil */
}

.report-item-style:hover {

```

```

        background-color: #e2e6ea; /* Um tom um pouco mais escuro ao passar o mouse */
        transform: translateY(-2px); /* Efeito de "levantar" */
        box-shadow: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075); /* Sombra sutil */
        border-color: #dee2e6;
    }

    .report-item-style i {
        margin-left: 1rem; /* Espaçamento entre o texto e o ícone */
    }

    /* Estilos para o sino de notificação */
    .notification-dot {
        width: 12px;
        height: 12px;
        border-radius: 50%;
        background-color: red;
        display: block;
        position: absolute;

        /* Posicionamento base, será modificado pela animação */
        top: 6px;
        right: 12px;

        z-index: 1070;
        animation: pulse 1.5s infinite; /* animação */
    }

    /* Animação de pulso */
    @keyframes pulse {
        0% {
            opacity: 1;
        }
        50% {
            opacity: 0.6;
        }
        100% {
            opacity: 1;
        }
    }

    .nav-link.position-relative {
        position: relative;
        padding-right: 15px; /*para dar espaço à bolinha no lado direito */
        padding-top: 5px; /* para dar espaço à bolinha no lado superior */
    }

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\css\dashboards.css

```

/* static/css/dashboards.css */

/* Cores principais */
:root {
    --blue-top: #0052cc;
    --blue-nav: #003166;
}

/* Barra superior */
.top-bar {
    background-color: var(--blue-top);
    color: #fff;

```

```

padding: 0.5rem 1rem;
}

.top-bar .logo {
font-size: 1.25rem;
font-weight: 600;
text-align: center;
color: white;
}

/* Navbar secundária */
.navbar-custom {
background-color: var(--blue-nav);
}

.navbar-custom .nav-link,
.navbar-custom .navbar-brand,
.navbar-custom .btn-link {
color: #fff;
}

.navbar-custom .nav-link:hover,
.navbar-custom .btn-link:hover {
color: #ddd;
}

/* Ajuste para centralizar os menus dropdown */
.navbar-nav {
display: flex;
justify-content: center;
width: 100%;
}

.nav-item.dropdown {
flex: 0 1 auto; /* Não cresce, pode encolher, largura automática */
white-space: nowrap; /* Impede que o texto quebre em várias linhas */
padding: 0 10px; /* Espaçamento entre os itens */
}

.dropdown-menu {
min-width: 12rem;
}

/* Estilos para os cards de acesso rápido */
.quick-access-card {
padding: 1.5rem;
background-color: white;
border-radius: 0.375rem;
box-shadow: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075);
text-align: center;
transition: transform 0.2s;
height: 100%;
}

.quick-access-card:hover {
transform: translateY(-5px);
}

.quick-access-card i {
font-size: 2rem;
color: #0d6efd;
}

.quick-access-card p {

```

```

    margin-top: 0.75rem;
    margin-bottom: 0;
}

/* Ajuste para mobile */
@media (max-width: 992px) {
    .navbar-nav {
        flex-direction: column;
    }

    .nav-item.dropdown {
        padding: 0;
        text-align: left;
    }
}

.modal-content.rounded-4 {
    border-radius: 2rem;
}

.modal-content .modal-header,
.modal-content .modal-footer {
    border: none;
}

/* Campos e labels alinhados e espaçados */
.modal-body .form-label {
    font-weight: 500;
}

.modal-body .form-control {
    border-radius: 0.5rem;
}

/* Botões maiores e com sombra */
.modal-footer .btn {
    padding: 0.75rem 1.5rem;
    border-radius: 1rem;
    box-shadow: 0 4px 6px rgba(0,0,0,0.1);
}

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\css\login.css

```

/* Cabeçalho azul customizado */
.custom-header {
    background-color: #1659bf;
    color: white;
}

/* Logo */
.logo-img {
    max-height: 100px;
}

/* Botão de login com cor customizada */
.custom-login-btn {
    background-color: #1659bf;
    border-color: #1659bf;
}

/* Botão hover */

```

```

.custom-login-btn:hover {
  background-color: #104ba3;
  border-color: #104ba3;
}

/* Card opcional: padding e largura máxima */
.login-card {
  width: 100%;
  max-width: 400px;
}

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\css\style.css

```

/* frontend\static\css\style.css */

/* Reset simples */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  background-color: #f7f9fc;
  display: flex;
  flex-direction: column;
}

header {
  background-color: #0052cc;
  padding: 1rem;
  text-align: center;
  color: white;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}

main {
  flex: 1;
  display: flex;
  justify-content: center;
  align-items: center;
}

.login-card {
  background-color: white;
  padding: 2rem;
  border-radius: 2rem;
  box-shadow: 0 8px 16px rgba(0,0,0,0.1);
  width: 320px;
  text-align: center;
  position: relative;
}

.avatar-placeholder {
  width: 100px;
  height: 100px;
  background-color: #e0e0e0;
  border-radius: 1rem;
  margin: 0 auto 1.5rem;
}

```



```

}

.login-card label {
  display: block;
  text-align: left;
  margin-top: 0.5rem;
  font-weight: bold;
}

.login-card input {
  width: 100%;
  padding: 0.5rem;
  margin-top: 0.25rem;
  border: 1px solid #ccc;
  border-radius: 1rem;
}

#recover-link {
  display: inline-block;
  margin: 1rem 0;
  font-size: 0.9rem;
  color: #0052cc;
  text-decoration: none;
}

#login-btn {
  background-color: #0052cc;
  color: white;
  border: none;
  padding: 0.75rem 2rem;
  border-radius: 1rem;
  font-size: 1rem;
  cursor: pointer;
  margin-top: 1rem;
}

#login-btn:hover {
  opacity: 0.9;
}

.list-group-item-action {
  cursor: pointer;
  transition: background-color .2s ease;
}

.list-group-item-action:hover {
  background-color: #f1f5f9;
}

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\alertasModule.js

```

// frontend/static/js/alertasModule.js

import { apiService } from './apiService.js';
import { uiService } from './uiService.js';
import { showAlert, formatDateTime, setNewAlertsFlag, updateNotificationBellUI } from './utils.js';

class AlertasModule {
  constructor() {
    this.currentPage = 1;
    this.pageSize = 10;
  }
}

```

```

this.currentSearchTerm = null;
this.currentTipoAlerta = null;
this.currentTotalPages = 1;

// Armazenar as referências das funções bound uma única vez
this._boundHandlePaginationClick = this._handlePaginationClick.bind(this);
this._boundHandlePageSizeChange = this._handlePageSizeChange.bind(this);
this._boundHandleTableActionClick = this._handleTableActionClick.bind(this);
this._boundHandleSearchAlert = this._handleSearchAlert.bind(this);
this._boundHandleClearAlertSearch = this._handleClearAlertSearch.bind(this);
}

init() {
  // Nada de especial aqui, os bindings são feitos após a renderização
}

async renderAlertsPage(){
  uiService.showLoading();
  try {
    const params = {
      page: this.currentPage,
      size: this.pageSize
    };

    if (this.currentTipoAlerta !== null && this.currentTipoAlerta !== undefined && this.currentT
      params.tipo_alerta = this.currentTipoAlerta;
    }

    if (this.currentSearchTerm !== null && this.currentSearchTerm !== undefined && this.currentS
      params.search_term = this.currentSearchTerm;
    }

    const data = await apiService.get('/alertas/paginated', params);
    this.currentTotalPages = data.total_pages;
    const alerts = data.items;

    const tableHeaders = ['Tipo', 'Mensagem', 'Item ID', 'Data do Alerta', 'Ações'];

    const searchAndFilterHtml = `
      <div id="alertas-search-bar" class="row mb-3">
        <div class="col-md-4 mb-2">
          <input type="text" id="alert-search-term" class="form-control"
            placeholder="Buscar por mensagem ou ID do item" value="${this.currentSearchT
          </div>
        <div class="col-md-4 mb-2">
          <select id="alert-type-filter" class="form-select">
            <option value="">Todos os Tipos</option>
            <option value="1" ${this.currentTipoAlerta === 1 ? 'selected' : ''}>Estoque
            <option value="2" ${this.currentTipoAlerta === 2 ? 'selected' : ''}>Validade
          </select>
        </div>
        <div class="col-md-4 d-flex mb-2">
          <button id="btn-search-alert" class="btn btn-primary me-2">Buscar</button>
          <button id="btn-clear-alert-search" class="btn btn-secondary">Limpar</button>
        </div>
      </div>`;

    const tableHtml = uiService.renderTable(tableHeaders, alerts, {
      noRecordsMessage: "Nenhum alerta encontrado.",
      rowMapper: (alerta) => {
        const tipoAlertaText = this._getTipoAlertaText(alerta.tipo_alerta);
        return [
          tipoAlertaText,

```

```

        alerta.mensagem_alerta,
        alerta.item_id,
        formatDateTime(alerta.data_alerta)
    ];
    },
    actionsHtml: (alerta) => `
        <button class="btn btn-sm btn-info me-1 btn-ver-item" data-item-id="${alerta.item_id}">
            <i class="bi bi-eye"></i> Item
        </button>
        ${!alerta.ignorar_novos ? `
        <button class="btn btn-sm btn-warning btn-ignorar-alerta" data-id="${alerta.alerta_id}">
            <i class="bi bi-x-circle"></i> Ignorar
        </button>` : `

```

```

        btnSearchAlert.removeEventListener('click', this._boundHandleSearchAlert);
    }
    if (btnClearAlertSearch) {
        btnClearAlertSearch.removeEventListener('click', this._boundHandleClearAlertSearch);
    }

    // Add new listeners
    if (alertsPaginationNav) {
        alertsPaginationNav.addEventListener('click', this._boundHandlePaginationClick);
    }
    if (pageSizeSelect) {
        pageSizeSelect.addEventListener('change', this._boundHandlePageSizeChange);
    }
    if (btnSearchAlert) {
        btnSearchAlert.addEventListener('click', this._boundHandleSearchAlert);
    }
    if (btnClearAlertSearch) {
        btnClearAlertSearch.addEventListener('click', this._boundHandleClearAlertSearch);
    }
}

_handleSearchAlert() {
    this.currentSearchTerm = document.getElementById('alert-search-term').value.trim();
    const selectedType = document.getElementById('alert-type-filter').value;
    this.currentTipoAlerta = selectedType ? parseInt(selectedType) : null;
    this.currentPage = 1;
    this.renderAlertsPage();
}

_handleClearAlertSearch() {
    document.getElementById('alert-search-term').value = '';
    document.getElementById('alert-type-filter').value = '';
    this.currentSearchTerm = '';
    this.currentTipoAlerta = null;
    this.currentPage = 1;
    this.renderAlertsPage();
}

_handlePaginationClick(e) {
    e.preventDefault();

    // Specific for alerts pagination links (e.g., data-page="alerts-1")
    const clickedPageLink = e.target.closest('a[data-page^="alerts-"]');
    if (clickedPageLink) {
        const pageValue = clickedPageLink.dataset.page.split('-')[1]; // Extract the number part
        const newPage = parseInt(pageValue);
        if (!isNaN(newPage) && newPage !== this.currentPage) {
            this.currentPage = newPage;
            this.renderAlertsPage();
        }
    }
    return;
}

// Specific for alerts action buttons (e.g., data-action="alerts-prev")
const clickedActionButton = e.target.closest('a[data-action^="alerts-"]');
if (clickedActionButton) {
    const action = clickedActionButton.dataset.action;
    let newPage = this.currentPage;

    if (action === 'alerts-prev') {
        if (newPage > 1) newPage--;
    } else if (action === 'alerts-next') {
        if (newPage < this.currentTotalPages) newPage++;
    }
}

```

```

    }

    if (newPage !== this.currentPage) {
      this.currentPage = newPage;
      this.renderAlertsPage();
    }
    return;
  }
}

_handlePageSizeChange(e) {
  this.pageSize = parseInt(e.target.value);
  this.currentPage = 1;
  this.renderAlertsPage();
}

_bindTableActions() {
  const mainContent = document.getElementById('main-content');
  if (mainContent) {
    // Usa a referência da função bound armazenada
    mainContent.removeEventListener('click', this._boundHandleTableActionClick);
    mainContent.addEventListener('click', this._boundHandleTableActionClick);
  }
}

async _handleTableActionClick(e) {
  const ignoreButton = e.target.closest('.btn-ignorar-alerta');
  if (ignoreButton) {
    const alertId = parseInt(ignoreButton.dataset.id);
    if (confirm('Tem certeza que deseja ignorar futuros alertas para este item/motivo?')) {
      await this.ignoreAlert(alertId);
    }
    return;
  }

  const viewItemButton = e.target.closest('.btn-ver-item');
  if (viewItemButton) {
    const itemId = parseInt(viewItemButton.dataset.itemId);
    uiService.showLoading();
    try {
      const itemDetails = await apiService.getItemById(itemId);
      uiService.fillModalDetalhesItem(itemDetails);
      uiService.getModalInstance('modalDetalheItem').show();
    } catch (error) {
      console.error("Erro ao carregar detalhes do item:", error);
      showAlert("Não foi possível carregar os detalhes do item.", "danger");
    } finally {
      uiService.hideLoading();
    }
    return; // Add return to prevent further propagation if a button is clicked
  }
}

async ignoreAlert(alertId) {
  uiService.showLoading();
  try {
    await apiService.patch(`/alertas/ignorar/${alertId}`);
    showAlert('Alerta ignorado com sucesso. Futuros alertas para este item/motivo não serão gera
    this.renderAlertsPage();
  } catch (error) {
    console.error('Erro ao ignorar alerta', error);
    showAlert(error.message || 'Erro ao ignorar o alerta.', 'danger');
  } finally {

```

```

        uiService.hideLoading();
    }
}

_getTipoAlertaText(tipoAlertaValue) {
    switch (tipoAlertaValue) {
        case 1:
            return '<span class="badge bg-danger"><i class="bi bi-box-fill"></i> Estoque Baixo</span>';
        case 2:
            return '<span class="badge bg-warning text-dark"><i class="bi bi-calendar-x"></i> Validade Baixa</span>';
        default:
            return 'Desconhecido';
    }
}

export const alertasModule = new AlertasModule();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\apiService.js

```

// frontend/static/js/apiService.js

class ApiService {

    constructor(baseUrl = '/api/almoxarifado') {
        this.baseUrl = baseUrl;
        this.token = localStorage.getItem('token');
        if (!this.token) {
            console.warn("Token de autenticação não encontrado no localStorage.");
        }
    }

    _getHeaders() {
        return {
            'Content-Type': 'application/json',
            'Authorization': `Bearer ${this.token}`
        };
    }

    async _fetch(endpoint, options = {}){
        const url = `${this.baseUrl}${endpoint}`;
        try {
            const response = await fetch(url, {
                ...options,
                headers: {
                    ...this._getHeaders(),
                    ...options.headers
                }
            });
            if (!response.ok) {
                const errorData = await response.json().catch(() => ({ detail: 'Erro desconhecido na resposta' }));
                throw new Error(errorData.detail || `Erro na API: ${response.status} ${response.statusText}`);
            }
            // Se a resposta for 204 No Content, response.json() falhará.
            // Verifique o status antes de tentar parsear como JSON.
            if (response.status === 204) {
                return {}; // Retorna um objeto vazio para 204 No Content
            }
            return response.json();
        } catch (error) {
            console.error('Erro ao fazer fetch:', error);
            return null;
        }
    }
}

```

```

    } catch (error) {
      console.error(`Falha na requisição para ${url}`, error);
      throw error;
    }
  }

  async get (endpoint, params = {}) {
    const queryString = new URLSearchParams (params).toString();
    const urlWithParams = `${endpoint}${queryString ? `?${queryString}` : ''}`;
    return await this._fetch(urlWithParams, { method: 'GET' });
  }

  async post(endpoint, data) {
    return this._fetch(endpoint, {
      method: 'POST',
      body: JSON.stringify(data)
    });
  }

  async put(endpoint, data) {
    return this._fetch(endpoint, {
      method: 'PUT',
      body: JSON.stringify(data)
    });
  }

  async patch(endpoint, data = {}) {
    return this._fetch(endpoint, {
      method: 'PATCH',
      body: JSON.stringify(data)
    });
  }
}

// Métodos específicos (mantidos do seu código original)

async getUsuarioById(id) {
  try {
    const user = await this.get(`/usuarios/${id}`);
    return user.nome_usuario;
  } catch (e) {
    return `#${id}`;
  }
}

async getSetorById(id) {
  try {
    const setor = await this.get(`/setores/${id}`);
    return setor.nome_setor;
  } catch (e) {
    return `#${id}`;
  }
}

async fetchAllRetiradas (page, pageSize, filters) {
  const params = { page, page_size: pageSize };
  const queryParamsForApi = {};

  if (filters.status !== null && filters.status !== undefined && filters.status !== '') {
    queryParamsForApi.status = filters.status;
  }

  if (filters.solicitante) {
    queryParamsForApi.solicitante = filters.solicitante;
  }
}

```

```

    if (filters.start_date) {
        queryParamsForApi.start_date = filters.start_date;
    }
    if (filters.end_date) {
        queryParamsForApi.end_date = filters.end_date;
    }

    const hasActiveFilters = Object.keys(queryParamsForApi).length > 0;
    const endpoint = hasActiveFilters ? '/retiradas/search' : '/retiradas/paginated';

    const responseData = await this.get(endpoint, { ...params, ...queryParamsForApi });

    return {
        current_page: responseData.page,
        total_pages: responseData.pages,
        total_items: responseData.total,
        items: responseData.items
    };
}

async fetchRetiradasPendentes (page, pageSize) {
    const responseData = await this.get(`/retiradas/pendentes/paginated`, { page, page_size: pageSize });
    return {
        current_page: responseData.page,
        total_pages: responseData.pages,
        total_items: responseData.total,
        items: responseData.items
    };
}

async updateRetiradaStatus(id, status, detail) {
    return this.put(`/retiradas/${id}`, { status, detalhe_status: detail });
}

async fetchAllItens() {
    return this.get('/itens');
}

async fetchAllSetores() {
    return this.get('/setores');
}

async solicitarRetirada (data) {
    return this.post('/retiradas/', data);
}

async searchItems(nome = null, categoria = null, page = 1, size = 10) {
    const params = { page, size };
    if (nome) {
        params.nome = nome;
    }
    if (categoria) {
        params.categoria = categoria;
    }
    return this.get('/itens/buscar', params);
}

async getItemById(itemId) {
    return this.get(`/itens/${itemId}`);
}

// NOVO: Obter contagem de alertas não visualizados

```



```

    async getUnviewedAlertsCount() {
      try {
        const response = await this.get('/alertas/unviewed-count');
        return response.count;
      } catch (error) {
        console.error('Erro ao buscar contagem de alertas não visualizados:', error);
        return 0;
      }
    }

    // Marcar todos os alertas como visualizados
    async markAllAlertsAsViewed() {
      try {
        await this.patch('/alertas/mark-viewed');
      } catch (error) {
        console.error('Erro ao marcar alertas como visualizados:', error);
      }
    }
  }

  export const apiService = new ApiService();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\bootstrap.bundle.min.js

```

/*!
 * Bootstrap v5.3.5 (https://getbootstrap.com/)
 * Copyright 2011-2025 The Bootstrap Authors (https://github.com/twbs/bootstrap/graphs/contributors)
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 */
!function(t,e){"object"==typeof exports&&"undefined"!=typeof module?module.exports=e():"function"==typeof define?define(["module"],function(){return e(module)}):"undefined"!=typeof window?window.bootstrap=e():t.bootstrap=e()}(this,(function(){
  //# sourceMappingURL=bootstrap.bundle.min.js.map

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\cadastrar-categorias.js

```

// frontend/static/js/cadastrar-categorias.js

document.addEventListener('DOMContentLoaded', () => {
  const modalEl = document.getElementById('modalCadastrarCategoria');
  const modal = new bootstrap.Modal(modalEl);
  const form = document.getElementById('form-cadastrar-categoria');

  // Abrir
  document.getElementById('btn-open-cadastrar-categoria')?.addEventListener('click', e => {
    e.preventDefault(); form.reset(); modal.show();
  });
  document.getElementById('btn-cancelar-categoria')?.addEventListener('click', () => {
    modal.hide(); document.querySelectorAll('.modal-backdrop').forEach(el=>el.remove());
  });

  // Salvar cadastro
  document.getElementById('btn-salvar-categoria').onclick = async () => {
    if(!form.checkValidity()) return form.reportValidity();
    const data = { nome_categoria: form.nome_categoria.value, descricao_categoria: form.descricao_categoria.value };
    const token = localStorage.getItem('token');
    const res = await fetch('/api/almoxarifado/categorias/', {
      method: 'POST', headers: { 'Content-Type': 'application/json', 'Authorization': `Bearer ${token}` }, body: JSON.stringify(data)
    });
    if(res.ok) {
      alert('Categoria cadastrada com sucesso!');
      form.reset();
    } else {
      alert('Erro ao cadastrar categoria. Verifique os dados e tente novamente.');
    }
  };

```

```

});
if(!res.ok) return alert('Erro ao cadastrar categoria');

form.reset(); modal.hide(); document.querySelectorAll('.modal-backdrop').forEach(el=>el.remove());
renderizarCategorias();

// alerta de sucesso
const alertBox = document.getElementById('alert-categoria');
alertBox.innerHTML = `<div class="alert alert-success alert-dismissible fade show" role="alert">Cate
setTimeout(()=>{ const a = alertBox.querySelector('.alert'); if(a) bootstrap.Alert.getOrCreateInstan
});
});

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxa rifado\_ets\frontend\static\js\cadastrar-item.js

```

//frontend\static\js\cadastrar-item.js

document.addEventListener('DOMContentLoaded', () => {
  const modalEl = document.getElementById('modalCadastrarItem');
  const modal = new bootstrap.Modal(modalEl);

  async function preencherCategoriasNoCadastro() {
    const token = localStorage.getItem('token');
    const resp = await fetch('/api/almoxarifado/categorias', {
      headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
    });
    if (!resp.ok) {
      console.error('[Cadastro] Erro ao carregar categorias:', resp.status);
      return;
    }
    const cats = await resp.json();

    // Busca o <select> dentro do próprio modal
    const sel = modalEl.querySelector('select[name="categoria_id"]');
    if (!sel) return console.error('[Cadastro] select[name="categoria_id"] não encontrado!');

    sel.innerHTML = '<option value="" disabled selected>Selecione...</option>';
    cats.forEach(c => {
      const o = document.createElement('option');
      o.value = c.categoria_id;
      o.textContent = `${c.categoria_id} - ${c.nome_original.toUpperCase()}`;
      sel.append(o);
    });
  }

  // Use shown.bs.modal para garantir que o modal já está completamente aberto
  modalEl.addEventListener('shown.bs.modal', preencherCategoriasNoCadastro);

  // resto do código continua igual...
  document.querySelectorAll('#btn-open-cadastrar-item').forEach(btn => {
    btn.addEventListener('click', e => {
      e.preventDefault();
      modal.show();
    });
  });
});

async function preencherCategoriasNoCadastro() {
  const token = localStorage.getItem('token');
  const resp = await fetch('/api/almoxarifado/categorias', {

```

```

    headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
  });
  if (!resp.ok) return console.error('Falha ao carregar categorias');
  const cats = await resp.json();
  const sel = document.getElementById('categoria_id');
  sel.innerHTML = '<option value="" disabled selected>Selecione...</option>';
  cats.forEach(c => {
    const o = document.createElement('option');
    o.value = c.categoria_id;
    o.textContent = `${c.categoria_id} - ${c.nome_categoria.toUpperCase()}`;
    sel.append(o);
  });
}
modalEl.addEventListener('show.bs.modal', preencherCategoriasNoCadastro);

const form = document.getElementById('form-cadastrar-item');
const btnSalvar = document.getElementById('btn-salvar-item');

// 1) Cancelar
document.getElementById('btn-cancelar-item').addEventListener('click', () => {
  modal.hide();
});

// 2) Ao esconder o modal, limpa backdrop e classe no body
modalEl.addEventListener('hidden.bs.modal', () => {
  // remove sombra e bloqueio de clique
  document.body.classList.remove('modal-open');
  document.querySelectorAll('.modal-backdrop').forEach(el => el.remove());
});

btnSalvar.addEventListener('click', async () => {
  if (!form.checkValidity()) {
    form.reportValidity();
    return;
  }

  // monta objeto e filtra valores vazios
  const formData = new FormData(form);
  const data = {};
  for (const [key, value] of formData.entries()) {
    if (value !== '') {
      // converte numéricos
      if (['quantidade_item', 'quantidade_minima_item', 'categoria_id'].includes(key)) {
        data[key] = Number(value);
      } else {
        data[key] = value;
      }
    }
  }

  // formata data_entrada_item se existir
  if (data.data_entrada_item) {
    data.data_entrada_item = new Date(data.data_entrada_item).toISOString();
  }

  // data_validade_item já vem em yyyy-mm-dd e o Pydantic aceita

  try {
    const token = localStorage.getItem('token');
    const resp = await fetch('/api/almoxarifado/itens/', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`
      },
    },

```

```

        body: JSON.stringify(data)
    });

    if (resp.status === 201) {
        alert('✔ Item cadastrado com sucesso!');
        form.reset();
        modal.hide();
        // atualiza listagem se aberta
        if (typeof renderizarListaItens === 'function') renderizarListaItens();
    } else {
        // exibe erros vindo da API
        const err = await resp.json();
        if (Array.isArray(err.detail)) {
            // Pydantic retorna lista de erros
            const msgs = err.detail.map(e => `${e.loc.join('.')}: ${e.msg}`);
            alert('Erros:\n' + msgs.join('\n'));
        } else {
            alert(err.detail || 'Falha ao cadastrar item.');
```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\dataService.js

```

// frontend/static/js/dataService.js
import { apiService } from '../apiService.js';
import estadoGlobal from '../estadoGlobal.js';

class DataService {
    async buildLookupMaps(retiradas) {
        const userIds = [...new Set(retiradas.map(r => r.usuario_id).filter(id => id != null))];
        // Coleta IDs de usuários que autorizaram também
        const adminIds = [...new Set(retiradas.map(r => r.autorizado_por).filter(id => id != null))];
        const allUserIds = [...new Set([...userIds, ...adminIds])]; // Combina e remove duplicatas

        const setorIds = [...new Set(retiradas.map(r => r.setor_id).filter(id => id != null))];

        const [usuariosArr, setoresArr] = await Promise.all([
            Promise.all(allUserIds.map(id => apiService.getUsuarioById(id).then(name => [id, name]))),
            Promise.all(setorIds.map(id => apiService.getSetorById(id).then(name => [id, name]))
        ]));

        return {
            usuarioMap: Object.fromEntries(usuariosArr), // Este mapa agora contém todos os nomes de usu
            setorMap: Object.fromEntries(setoresArr)
        };
    }

    async getProcessedRetiradas(fetchFunction, page, pageSize, filters = {}) {
        const data = await fetchFunction(page, pageSize, filters);
        const { usuarioMap, setorMap } = await this.buildLookupMaps(data.items);

        const processedItems = data.items.map(r => ({
            ...r,
            usuario_nome: r.solicitado_localmente_por || usuarioMap[r.usuario_id] || 'N/A',

```

```

        setor_nome: setorMap[r.setor_id] || 'N/A',
        autorizado_por_nome: r.autorizado_por ? usuarioMap[r.autorizado_por] : 'N/A' // Adiciona o nome
    }));

    return {
        ...data,
        items: processedItems
    };
}

export const dataService = new DataService();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\estadoGlobal.js

```

// frontend\static\js\estadoGlobal.js
const estadoGlobal = {
    allRetiradas: [],
    pendentesRetiradas: [],
    currentHistoricoPage: 1,
    totalHistoricoPages: 1,
    currentHistoricoPageSize: 10,
    currentHistoricoFilters: {},
    currentPendentesPage: 1,
    totalPendentesPages: 1,
    currentPendentesPageSize: 10,

    PAGE_SIZE_OPTIONS: [5, 10, 25, 50, 100],

    statusMap: { // converter de INT para STRING (para exibição)
        1: 'PENDENTE',
        2: 'AUTORIZADA',
        3: 'CONCLUÍDA', // Verifique se está presente
        4: 'NEGADA'
    },
    statusMapUpdate: { // converter de STRING para INT (para envio à API)
        PENDENTE: 1,
        AUTORIZADA: 2,
        CONCLUIDA: 3, // Verifique se está presente
        NEGADA: 4
    },

    // Métodos para atualizar o estado de forma controlada
    setHistoricoPagination(currentPage, totalPages, pageSize, filters) {
        this.currentHistoricoPage = currentPage;
        this.totalHistoricoPages = totalPages;
        this.currentHistoricoPageSize = pageSize;
        this.currentHistoricoFilters = { ...filters };
    },

    setPendentesPagination(currentPage, totalPages, pageSize) {
        this.currentPendentesPage = currentPage;
        this.totalPendentesPages = totalPages;
        this.currentPendentesPageSize = pageSize;
    },

    setAllRetiradas(data) {
        this.allRetiradas = data;
    },
    setPendentesRetiradas(data) {

```

```

        this.pendientesRetiradas = data;
    }
};

export default estadoGlobal;

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\listar-categorias.js

```

// frontend/static/js/listar-categorias.js

// Estado de paginação e busca
let currentPageCat = 1;
let pageSizeCat = 10;
let searchCat = '';

// APIs
async function carregarCategoriasPag(page = currentPageCat, size = pageSizeCat) {
    const token = localStorage.getItem('token');
    const resp = await fetch(
        `/api/almoxarifado/categorias/paginated?page=${page}&size=${size}`, {
            headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
        }
    );
    if (!resp.ok) throw new Error(resp.status);
    return resp.json();
}

async function buscarCategorias(nome, page = currentPageCat, size = pageSizeCat) {
    const token = localStorage.getItem('token');
    const params = new URLSearchParams({ page, size });
    if (nome) params.append('nome', nome);
    const resp = await fetch(`/api/almoxarifado/categorias/buscar?${params}`, {
        headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
    });
    if (!resp.ok) throw new Error(resp.status);
    return resp.json();
}

// Renderização
async function renderizarCategorias() {
    const main = document.getElementById('main-content');
    const data = searchCat
        ? await buscarCategorias(searchCat, currentPageCat, pageSizeCat)
        : await carregarCategoriasPag(currentPageCat, pageSizeCat);

    main.innerHTML = `
        <div class="row mb-3" id="search-bar">
            <div class="col-md-4">
                <input type="text" id="search-categoria-nome" class="form-control" placeholder="Buscar por nome">
            </div>

            <div class="col-md-4 d-flex">
                <button id="btn-search-cat" class="btn btn-primary me-2">Buscar</button>
                <button id="btn-clear-search-cat" class="btn btn-secondary">Limpar</button>
            </div>
        </div>
        <h3 class="mb-3">Lista de Categorias</h3>
        <div class="table-responsive">
            <table class="table table-bordered table-striped">
                <thead class="table-secondary text-center">

```

```

        <tr>
          <th>ID</th>
          <th>Nome</th>
          <th>Descrição</th>
          <th>Ações</th>
        </tr>
      </thead>
      <tbody>
        ${data.items.map((c,i) => `
          <tr>
            <td class="text-center">${c.categoria_id}</td>
            <td>${c.nome_original}</td>
            <td>${c.descricao_categoria || '-'}</td>
            <td class="text-center">
              <button class="btn btn-sm btn-primary btn-editar-cat me-1" data-id="${c.categoria_id}">Editar</button>
              <button class="btn btn-sm btn-danger btn-deletar-cat" data-id="${c.categoria_id}">Deletar</button>
            </td>
          </tr>`}).join('')}
      </tbody>
    </table>
  </div>
  <nav>
    <ul class="pagination justify-content-center">
      <li class="page-item ${currentPageCat === 1 ? 'disabled' : ''}"><a class="page-link" href="#">1</a>
      ${(() => {
        const pages = [];
        const start = Math.max(1, currentPageCat - 2);
        const end = Math.min(data.total_pages, currentPageCat + 2);
        for (let p = start; p <= end; p++) {
          pages.push(`<li class="page-item ${p === currentPageCat ? 'active' : ''}"><a class="page-link" href="#">${p}</a>`);
        }
        return pages.join('');
      })()}
      <li class="page-item ${currentPageCat === data.total_pages ? 'disabled' : ''}"><a class="page-link" href="#">...</a>
    </ul>
  </nav>
  <div class="d-flex justify-content-center my-2">
    <label class="me-2">Categorias por página:</label>
    <select id="page-size-cat" class="form-select w-auto">
      ${[5,10,25,50,100].map(opt => `<option value="${opt}" ${opt === pageSizeCat ? 'selected' : ''}>${opt}</option>`)}
    </select>
  </div>
`;
bindCategoriaActions();
bindPaginationCat(data.total_pages);
}

// Ações
function bindCategoriaActions() {
  const main = document.getElementById('main-content');

  document.getElementById('btn-search-cat').onclick = () => {
    searchCat = document.getElementById('search-categoria-nome').value.trim();
    currentPageCat = 1;
    renderizarCategorias();
  };

  document.getElementById('btn-clear-search-cat').onclick = () => {
    searchCat = '';
    currentPageCat = 1;
    renderizarCategorias();
  };

  main.querySelectorAll('.btn-editar-cat').forEach(btn => {

```

```

    btn.onclick = async () => {
      const id = btn.dataset.id;
      const token = localStorage.getItem('token');
      const res = await fetch(`/api/almoxarifado/categorias/${id}`, {
        headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
      });
      if(!res.ok) return alert('Erro ao carregar categoria');
      const cat = await res.json();

      // popula modal de edição
      document.getElementById('edit-nome_categoria').value = cat.nome_original;
      document.getElementById('edit-descricao_categoria').value = cat.descricao_categoria || '';
      const saveBtn = document.getElementById('btn-salvar-editar-categoria');
      saveBtn.dataset.id = id;

      // exibe modal
      new bootstrap.Modal(document.getElementById('modalEditarCategoria')).show();
    };
  });

  main.querySelectorAll('.btn-deletar-cat').forEach(btn => {
    btn.onclick = async () => {
      if (!confirm('Excluir categoria?')) return;

      const id = btn.dataset.id;
      const token = localStorage.getItem('token');
      const resp = await fetch(
        `/api/almoxarifado/categorias/${id}`,
        { method: 'DELETE', headers: { 'Authorization': `Bearer ${token}` } }
      );

      if (!resp.ok) {
        // tenta ler o detalhe retornado pelo FastAPI
        let err;
        try {
          const body = await resp.json();
          err = body.detail || resp.statusText;
        } catch {
          err = resp.statusText;
        }
        return alert(`Erro ao excluir categoria:\n${err}`);
      }

      // só re-renderiza se deu sucesso
      renderizarCategorias();
    };
  });
}

// Salvar edição
document.getElementById('btn-salvar-editar-categoria').onclick = async e => {
  const id = e.currentTarget.dataset.id;
  if(!id) return alert('ID não definido');
  const form = document.getElementById('form-editar-categoria');
  if(!form.checkValidity()) return form.reportValidity();

  const data = {
    nome_categoria: form.nome_categoria.value,
    descricao_categoria: form.descricao_categoria.value || undefined
  };
  const token = localStorage.getItem('token');
  const res = await fetch(`/api/almoxarifado/categorias/${id}`, {
    method: 'PUT',

```



```

        headers: { 'Content-Type': 'application/json', 'Authorization': `Bearer ${token}` },
        body: JSON.stringify(data)
    });
    if(!res.ok) return alert('Erro ao salvar categoria');

    bootstrap.Modal.getInstance(document.getElementById('modalEditarCategoria')).hide();
    renderizarCategorias();
};

// Paginação
function bindPaginationCat(totalPages) {
    const main = document.getElementById('main-content');
    main.querySelector('[data-action="prev-cat"]').onclick = e => {
        e.preventDefault(); if(currentPageCat>1){ currentPageCat--; renderizarCategorias(); }
    };
    main.querySelector('[data-action="next-cat"]').onclick = e => {
        e.preventDefault(); if(currentPageCat<totalPages){ currentPageCat++; renderizarCategorias(); }
    };
    main.querySelectorAll('[data-page-cat]').forEach(el => el.onclick = e => {
        e.preventDefault(); currentPageCat = +el.dataset.pageCat; renderizarCategorias();
    });
    document.getElementById('page-size-cat').onchange = e => {
        pageSizeCat = +e.target.value; currentPageCat = 1; renderizarCategorias();
    };
}

// Inicialização
document.getElementById('listar-categoria-link')?.addEventListener('click', e =>{ e.preventDefault(); cu
document.getElementById('listar-categoria-link-quick')?.addEventListener('click', e =>{ e.preventDefault

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\listar-itens.js

```

// frontend/static/js/listar-itens.js

// Estado de paginação
let currentPage = 1;
let pageSize = 10;
const pageSizeOptions = [5, 10, 25, 50, 100];

// guarda estado da última busca
let searchNome = '';
let searchCategoria = '';

// 1) API de categorias
async function carregarCategorias() {
    const token = localStorage.getItem('token');
    const resp = await fetch(`/api/almoxarifado/categorias`, {
        headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
    });

    if (resp.status === 401) {
        window.location = '/';
        return [];
    }

    if (!resp.ok) throw new Error(`HTTP ${resp.status}`);

    return resp.json();
}

// 2) API de itens

```

```

async function carregarListaltens(page = currentPage, size = pageSize) {
  const token = localStorage.getItem('token');
  const resp = await fetch(
    `/api/almoxarifado/itens/paginated?page=${page}&size=${size}`, {
      headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
    });

  if (resp.status === 401) {
    window.location = '/';
    return { items: [], total_pages: 0 };
  }

  if (!resp.ok) throw new Error(`HTTP ${resp.status}`);

  const data = await resp.json();

  // Ordena itens alfabeticamente por nome
  data.items.sort((a, b) => a.nome_item_original.localeCompare(b.nome_item_original));
  return data;
}

async function buscarItens(nome, categoria, page = currentPage, size = pageSize) {
  const token = localStorage.getItem('token');
  const params = new URLSearchParams({ page, size });

  if (nome) params.append('nome', nome);
  if (categoria) params.append('categoria', categoria);

  const resp = await fetch(`/api/almoxarifado/itens/buscar?${params}`, {
    headers: { 'Authorization': `Bearer ${token}`, 'Accept': 'application/json' }
  });

  if (!resp.ok) throw new Error(`HTTP ${resp.status}`);

  const data = await resp.json();

  // Ordena itens da busca
  data.items.sort((a, b) => a.nome_item_original.localeCompare(b.nome_item_original));
  return data;
}

//3) Templates HTML
function criarSearchBar() {
  return `
    <div class="row mb-3" id="search-bar">
      <div class="col-md-4">
        <input type="text" id="search-nome" class="form-control"
          placeholder="Buscar por nome" value="${searchNome}">
      </div>
      <div class="col-md-4">
        <input type="text" id="search-categoria" class="form-control"
          placeholder="Buscar por categoria" value="${searchCategoria}">
      </div>
      <div class="col-md-4 d-flex">
        <button id="btn-search" class="btn btn-primary me-2">Buscar</button>
        <button id="btn-clear-search" class="btn btn-secondary">Limpar</button>
      </div>
    </div>`;
}

function criarTabelaltens(itens, categoryMap) {
  let html = `
    <h3 class="mb-3">Lista de Itens</h3>
  `;

```

```

<div class="table-responsive">
  <table class="table table-bordered table-striped">
    <thead class="table-secondary text-center">
      <tr>
        <th>ID</th><th>Nome</th><th>Descrição</th><th>Unidade de Medida</th><th>Quantidade</th>
        <th>Validade</th><th>Entrada</th><th>Marca</th><th>Categoria</th><th>Ações</th>
      </tr>
    </thead>
    <tbody>`;

items.forEach(item => {
  const cat = categoryMap[item.categoria_id];
  const label = cat
    ? `${item.categoria_id} ${cat.nome_original.toUpperCase()}`
    : item.categoria_id;

  html += `
    <tr>
      <td>${item.item_id}</td>
      <td>${item.nome_item_original}</td>
      <td>${item.descricao_item || '-'}</td>
      <td>${item.unidade_medida_item}</td>
      <td class="text-center">${item.quantidade_item}</td>
      <td class="text-center">${item.data_validade_item
        ? new Date(item.data_validade_item).toLocaleDateString()
        : '-'}</td>
      <td class="text-center">${item.data_entrada_item
        ? new Date(item.data_entrada_item).toLocaleDateString()
        : '-'}</td>
      <td>${item.marca_item || '-'}</td>
      <td class="text-center">${label}</td>
      <td class="text-center">
        <button class="btn btn-sm btn-primary me-1 btn-editar" data-id="${item.item_id}">
          <i class="bi bi-pencil-square"></i> Editar
        </button>
        <button class="btn btn-sm btn-danger btn-deletar" data-id="${item.item_id}">
          <i class="bi bi-trash"></i> Deletar
        </button>
      </td>
    </tr>`;
});

html += `</tbody></table></div>`;
return html;
}

function criarControlesPaginacao(totalPages) {
  // Note: The 'type' parameter is now passed from uiService.renderPagination
  // So, data-page will be like "itens-1", "itens-2", etc.
  let html = `<nav aria-label="Page navigation" id="itens-pagination-nav"><ul class="pagination justify-content-end">
    <li class="page-item ${currentPage === 1 ? 'disabled' : ''}>
      <a class="page-link" href="#" data-action="itens-prev">Anterior</a>
    </li>`;

  const start = Math.max(1, currentPage - 2);
  const end = Math.min(totalPages, currentPage + 2);

  for (let p = start; p <= end; p++) {
    html += `
      <li class="page-item ${p === currentPage ? 'active' : ''}>
        <a class="page-link" href="#" data-page="itens-${p}">${p}</a>
      </li>`;
  }
}

```

```

html += `<li class="page-item ${currentPage === totalPages ? 'disabled' : ''}>
  <a class="page-link" href="#" data-action="itens-next">Próximo</a>
</li>`;
html += `</ul></nav>
  <div class="d-flex justify-content-center my-2">
    <label class="me-2">Itens por página: </label>
    <select id="page-size-select" class="form-select w-auto">
      ${pageSizeOptions.map(opt => `<option value="${opt}" ${opt === pageSize ? 'selected' : ''}>${opt}</option>`)}
    </select>
  </div>`;
return html;
}

//4) Carrega e renderiza tudo
async function renderizarListaltens() {
  try {
    // se tiver filtro, chama buscarItens, senão lista geral
    const data = (searchNome || searchCategoria)
      ? await buscarItens(searchNome, searchCategoria, currentPage, pageSize)
      : await carregarListaltens(currentPage, pageSize);

    const categorias = await carregarCategorias();
    const categoryMap = {};
    categorias.forEach(c => categoryMap[c.categoria_id] = c);

    const main = document.getElementById('main-content');
    if (main) { // Ensure main content element exists
      main.innerHTML = `
        ${criarSearchBar()}
        ${criarTabelaltens(data.items, categoryMap)}
        ${criarControlesPaginacao(data.total_pages)}
      `;

      bindSearch();
      bindPagination(data.total_pages);
      bindRowActions(categorias);
    } else {
      console.error("Elemento 'main-content' não encontrado em listar-itens.js");
    }
  } catch (err) {
    console.error(err);
    const main = document.getElementById('main-content');
    if (main) {
      main.innerHTML = `<div class="alert alert-warning">Erro: ${err.message}</div>`;
    }
  }
}

//5) Bindings de busca
function bindSearch() {
  document.getElementById('btn-search')?.addEventListener('click', e => {
    e.preventDefault();
    const nome = document.getElementById('search-nome').value.trim();
    const cat = document.getElementById('search-categoria').value.trim();

    // atualiza estado de busca
    searchNome = nome;
    searchCategoria = cat;
    currentPage = 1;

    // re-renderiza via mesma função
    renderizarListaltens();
  });
}

```

```

    });

    document.getElementById('btn-clear-search')?.addEventListener('click', e => {
        e.preventDefault();

        // limpa filtros
        searchNome = '';
        searchCategoria = '';
        currentPage = 1;
        pageSize = 10; // Reset page size as well

        renderizarListaltens();
    });
}

//6) Bindings de paginação
function bindPagination(totalPages) {
    const itensPaginationNav = document.getElementById('itens-pagination-nav');
    const pageSizeSelect = document.getElementById('page-size-select');

    if (!itensPaginationNav) {
        console.warn("Elemento 'itens-pagination-nav' não encontrado para bindPagination.");
        return;
    }

    // Remove previous listeners to prevent duplicates
    itensPaginationNav.removeEventListener('click', handlePaginationClick);
    if (pageSizeSelect) {
        pageSizeSelect.removeEventListener('change', handlePageSizeChange);
    }

    // Add new listeners (using delegation for page links)
    itensPaginationNav.addEventListener('click', handlePaginationClick);
    if (pageSizeSelect) {
        pageSizeSelect.addEventListener('change', handlePageSizeChange);
    }

    function handlePaginationClick(e) {
        e.preventDefault();
        const clickedPageLink = e.target.closest('a[data-page^="itens-"]'); // Specific for itens pagination
        const clickedActionButton = e.target.closest('a[data-action^="itens-"]'); // Specific for itens pagination

        if (clickedPageLink) {
            const pageValue = clickedPageLink.dataset.page.split('-')[1]; // Extract page number (e.g., 1)
            const newPage = parseInt(pageValue);
            if (!isNaN(newPage) && newPage !== currentPage) {
                currentPage = newPage;
                renderizarListaltens();
            }
            return;
        }

        if (clickedActionButton) {
            const action = clickedActionButton.dataset.action;
            if (action === 'itens-prev') {
                if (currentPage > 1) {
                    currentPage--;
                    renderizarListaltens();
                }
            } else if (action === 'itens-next') {
                if (currentPage < totalPages) {
                    currentPage++;
                    renderizarListaltens();
                }
            }
        }
    }
}

```

```

        }
      }
      return;
    }
  }

function handlePageSizeChange(e) {
  pageSize = parseInt(e.target.value);
  currentPage = 1;
  renderizarListaltens();
}

//7) Bindings de ações nas linhas
function bindRowActions(categorias) {
  // deletar
  document.querySelectorAll('.btn-deletar').forEach(btn => {
    btn.onclick = async () => {
      if (!confirm('Excluir este item?')) return;
      const id = btn.dataset.id;
      const token = localStorage.getItem('token');

      try {
        const resp = await fetch(`/api/almoxarifado/itens/${id}`, {
          method: 'DELETE',
          headers: { 'Authorization': `Bearer ${token}` }
        });

        if (!resp.ok) {
          const error = await resp.json();
          // Extraí IDs de retiradas da mensagem de erro (se existirem)
          const matches = error.detail?.match(/retirada_item\\.DETAIL: Chave \\(item_id\\)=\\(\\d+\\)/);
          if (matches && matches[1]) {
            throw new Error(`Item vinculado à retirada ID: ${matches[1]}. Não pode ser excluído`);
          }
          throw new Error(error.detail || 'Erro ao excluir item');
        }
        renderizarListaltens();
      } catch (err) {
        console.error(err);
        // Mensagem personalizada para erros de chave estranha
        if (err.message.includes('retirada_item')) {
          alert('Item vinculado a uma ou mais retiradas.\\n\\nPrimeiro exclua as retiradas relacionadas');
        } else {
          alert(err.message);
        }
      }
    };
  });

  // editar
  document.querySelectorAll('.btn-editar').forEach(btn => {
    btn.onclick = async e => {
      e.preventDefault();
      const id = btn.dataset.id;
      const token = localStorage.getItem('token');

      //1) busca item
      const respItem = await fetch(`/api/almoxarifado/itens/${id}`, {
        headers: { 'Authorization': `Bearer ${token}` }
      });

      if (!respItem.ok) return alert('Erro ao carregar item');
    };
  });
}

```

```

const item = await respItem.json();

// 2) popula categorias no select de edição
const sel = document.getElementById('edit-categoria_id');
sel.innerHTML = '<option value="" disabled>Carregando...</option>';
categorias.forEach(c => {
    const o = document.createElement('option');
    o.value = c.categoria_id;
    o.textContent = `${c.categoria_id} ${c.nome_categoria.toUpperCase()}`;
    sel.append(o);
});
sel.value = item.categoria_id;

// 3) preenche demais campos
const form = document.getElementById('form-editar-item');
form.nome_item.value = item.nome_item_original;
form.unidade_medida_item.value = item.unidade_medida_item;
form.descricao_item.value = item.descricao_item;
form.quantidade_item.value = item.quantidade_item;
form.quantidade_minima_item.value = item.quantidade_minima_item || '';
form.data_validade_item.value = item.data_validade_item?.split('T')[0] || '';
form.data_entrada_item.value = item.data_entrada_item?.slice(0, 16) || '';
form.marca_item.value = item.marca_item || '';

// 4) guarda id no botão salvar
const saveBtn = document.getElementById('btn-salvar-editar-item');
saveBtn.dataset.id = id;

//5) exibe modal
new bootstrap.Modal(
    document.getElementById('modalEditarItem')
).show();
    };
});
}

//8) salvar edição
document.getElementById('btn-salvar-editar-item')?.addEventListener('click', async e => {
    const id = e.currentTarget.dataset.id;
    if (!id) return alert('ID de edição não definido');

    const form = document.getElementById('form-editar-item');
    if (!form.checkValidity()) {
        form.reportValidity();
        return;
    }

    const data = {
        nome_item: form.nome_item.value.trim(), // Adjusted to correct field
        unidade_medida_item: form.unidade_medida_item.value.trim(),
        descricao_item: form.descricao_item.value.trim(),
        quantidade_item: Number(form.quantidade_item.value),
        categoria_id: Number(form.categoria_id.value),
    };

    // Campos opcionais
    if (form.quantidade_minima_item.value.trim())
        data.quantidade_minima_item = Number(form.quantidade_minima_item.value);
    if (form.data_validade_item.value.trim())
        data.data_validade_item = form.data_validade_item.value;
    if (form.data_entrada_item.value.trim())
        data.data_entrada_item = form.data_entrada_item.value;
    if (form.marca_item.value.trim())

```

```

        data.marca_item = form.marca_item.value.trim();

    try {
        const token = localStorage.getItem('token');
        const resp = await fetch(`/api/almoxarifado/itens/${id}`, {
            method: 'PUT',
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${token}`
            },
            body: JSON.stringify(data)
        });

        if (!resp.ok) {
            const err = await resp.json();
            throw new Error(err.detail || `Erro HTTP ${resp.status}`);
        }

        bootstrap.Modal.getInstance(document.getElementById('modalEditarItem')).hide();
        renderizarListaltens();
    } catch (err) {
        console.error(err);
        alert('Erro ao salvar: ' + err.message);
    }
});

// 9) inicialização
const linkListar = document.getElementById('listar-item-link');
if (linkListar) {
    linkListar.addEventListener('click', e => {
        e.preventDefault();
        currentPage = 1;
        pageSize = 10;
        renderizarListaltens();
    });
}

// Export the function to be accessible globally
window.renderizarListaltens = renderizarListaltens;

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\login.js

```

//frontend\static\js\login.js

window.addEventListener('DOMContentLoaded', () => {
    const form = document.getElementById('login-form');
    if (!form) return;

    form.addEventListener('submit', async (e) => {
        e.preventDefault();

        const username = document.getElementById('username').value;
        const password = document.getElementById('password').value;
        const body = new URLSearchParams({ username, password });

        try {
            const resp = await fetch('api/almoxarifado/usuarios/token', {
                method: 'POST',
                headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
                credentials: 'include',
            });

```



```

        body: body.toString()
    });

    if (!resp.ok) {
        const err = await resp.json();
        alert(err.detail || 'Falha ao autenticar');
        return;
    }

    const { access_token } = await resp.json();

    // ■■■■ AQUI: grava o JWT no localStorage para uso futuro ■■■■
    localStorage.setItem('token', access_token);

    const payload = JSON.parse(atob(access_token.split('.')[1]));
    const tipo = payload.tipo_usuario;

    if (tipo === 1)        window.location.href = '/dashboardServidor';
    else if (tipo === 2)   window.location.href = '/dashboardAlmoxarifado';
    else if (tipo === 3)   window.location.href = '/dashboardDirecao';
    else                   alert('Tipo de usuário desconhecido.');
```

```

    } catch (error) {
        alert('Erro de conexão com o servidor.');
```

```

    }
    });
});

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\logout.js

```

//frontend\static\js\logout.js

window.addEventListener('DOMContentLoaded', () => {
    const logoutLink = document.getElementById('logout-link');
    if (!logoutLink) return;

    logoutLink.addEventListener('click', async (e) => {
        e.preventDefault();

        // Tenta chamar o endpoint (opcional)
        try {
            await fetch('/api/almoxarifado/usuarios/logout', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                credentials: 'include'
            });
        } catch (err) {
            console.warn('Logout no servidor falhou, mas continua no client', err);
        }

        // Limpa o token e volta para a tela de login
        localStorage.removeItem('token');
        window.location.href = '/'; // ou '/index.html'
    });
});

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\main.js

```
// frontend/static/js/main.js

import { retiradasModule } from './retiradasModule.js';
import { solicitarRetiradaModule } from './solicitar-retirada.js';
import { selecionarItemModule } from './selecionar-item-module.js';
import { reportsModule } from './reportsModule.js';
import { alertasModule } from './alertasModule.js';
import { apiService } from './apiService.js';
import { setNewAlertsFlag, getNewAlertsFlag, updateNotificationBellUI, showAlert, setNewWithdrawalRequest } from './utils.js';

const NOTIFICATION_SOUND_PATH = '/static/audio/notificacao01.mp3';
const NOTIFICATION_SOUND_PATH_RETIRADA = '/static/audio/notificacao02.mp3';

// Função auxiliar para re-inicializar dropdowns do Bootstrap
export function reinitializeBootstrapDropdowns() {
  const dropdownToggleList = [].slice.call(document.querySelectorAll('[data-bs-toggle="dropdown"]'));
  dropdownToggleList.map(function (dropdownToggleEl) {
    const dropdownInstance = bootstrap.Dropdown.getInstance(dropdownToggleEl);
    if (!dropdownInstance) {
      return new bootstrap.Dropdown(dropdownToggleEl);
    }
    return dropdownInstance;
  });
}

const mainContent = document.getElementById('main-content');
let defaultHTML = mainContent ? mainContent.innerHTML : ''; // Store initial content

document.addEventListener('DOMContentLoaded', () => {
  const homeButton = document.getElementById('home-button');

  if (homeButton && mainContent) {
    homeButton.addEventListener('click', e => {
      e.preventDefault();
      mainContent.innerHTML = defaultHTML;
      bindQuickAccessLinks();
      bindLogoutLink();
      checkAlertsNotification(); // (Manter para a verificação inicial ao carregar a página)
      reinitializeBootstrapDropdowns();
    });
  }

  function bindQuickAccessLinks() {
    const modalCadastrarItemEl = document.getElementById('modalCadastrarItem');
    if (modalCadastrarItemEl) {
      const modalItem = new bootstrap.Modal(modalCadastrarItemEl);
      document.querySelectorAll('#btn-open-cadastrar-item').forEach(btn => {
        btn.onclick = e => {
          e.preventDefault();
          modalItem.show();
        };
      });
    }
  }

  document.getElementById('listar-item-link')?.addEventListener('click', e => {
    e.preventDefault();
    if (typeof window.renderizarListaltens === 'function') window.renderizarListaltens();
    else console.warn('Função global 'renderizarListaltens' não encontrada. Verifique se listar-');
  });
});
```

```

    });

    document.getElementById('listar-item-link-quick')?.addEventListener('click', e => {
        e.preventDefault();
        if (typeof window.renderizarListaltens === 'function') window.renderizarListaltens();
        else console.warn("Função global 'renderizarListaltens' não encontrada.");
    });

    const modalCadastrarCategoriaEl = document.getElementById('modalCadastrarCategoria');
    if (modalCadastrarCategoriaEl) {
        const modalCat = new bootstrap.Modal(modalCadastrarCategoriaEl);
        document.getElementById('btn-open-cadastrar-categoria')?.addEventListener('click', e => {
            e.preventDefault();
            modalCat.show();
        });
    }

    document.getElementById('listar-categoria-link')?.addEventListener('click', e => {
        e.preventDefault();
        if (typeof window.renderizarCategorias === 'function') window.renderizarCategorias();
        else console.warn("Função global renderizarCategorias não encontrada. Verifique se listar-ca");
    });

    document.getElementById('listar-categoria-link-quick')?.addEventListener('click', e => {
        e.preventDefault();
        if (typeof window.renderizarCategorias === 'function') window.renderizarCategorias();
        else console.warn("Função global 'renderizarCategorias' não encontrada.");
    });

    document.getElementById('listar-retiradas-pendentes-quick')?.addEventListener('click', e => {
        e.preventDefault();
        retiradasModule.renderPendentesRetiradas();
    });

    // Modal de relatórios
    document.getElementById('open-reports-dashboard')?.addEventListener('click', e => {
        e.preventDefault();
        reportsModule.modalReportsDashboard.show();
    });

    // NOVO: Listener para o link "Listar Alertas" do menu de navegação principal
    document.getElementById('open-alertas-modal')?.addEventListener('click', e => {
        e.preventDefault();
        alertasModule.renderAlertsPage();
    });

    // Clique no sino de notificação - AGORA ABRE UM DROPDOWN
    const alertBell = document.getElementById('alert-notification-bell');
    if (alertBell) {
        // O sino agora é um dropdown toggle, seu comportamento é gerido pelo Bootstrap
        // Os itens do dropdown terão seus próprios listeners
    }

    // NOVO: Listeners para os itens do dropdown de notificação
    document.getElementById('new-alerts-menu-item')?.addEventListener('click', e => {
        e.preventDefault();
        setNewAlertsFlag(false); // Limpa o flag de alertas
        updateNotificationBellUI(); // Atualiza a UI do sino
        alertasModule.renderAlertsPage(); // Redireciona para a página de alertas
        bootstrap.Dropdown.getInstance(document.getElementById('alert-notification-bell'))?.hide();
    });

    document.getElementById('new-withdrawal-requests-menu-item')?.addEventListener('click', e => {

```

```

        e.preventDefault();
        setNewWithdrawalRequestsFlag(false); // Limpa o flag de solicitações de retirada
        updateNotificationBellUI(); // Atualiza a UI do sino
        retiradasModule.renderPendentesRetiradas(); // Redireciona para a página de retiradas pendentes
        bootstrap.Dropdown.getInstance(document.getElementById('alert-notification-bell'))?.hide();
    });

    // NOVO: Listener para o link "Ver todas as notificações"
    document.getElementById('open-all-notifications-link')?.addEventListener('click', e => {
        e.preventDefault();
        // Limpa ambos os flags ao clicar em "Ver todas"
        setNewAlertsFlag(false);
        setNewWithdrawalRequestsFlag(false);
        updateNotificationBellUI();
        alertasModule.renderAlertsPage(); // Por padrão, leva para a página de alertas
        bootstrap.Dropdown.getInstance(document.getElementById('alert-notification-bell'))?.hide();
    });
}

function bindLogoutLink() {
    document.getElementById('logout-link')?.addEventListener('click', e => {
        e.preventDefault();
        if (typeof window.logout === 'function') window.logout();
        else console.warn("Função global 'logout' não encontrada. Verifique se logout.js está carregada.");
    });
}

// Função para verificar alertas e solicitações de retirada (agora chamada inicialmente e via WebSocket)
async function checkAlertsNotification() {
    try {
        const alertsCount = await apiService.getUnviewedAlertsCount();
        // Futuramente, você pode ter um endpoint para contar novas solicitações de retirada
        // Por enquanto, o flag será definido apenas pelo WebSocket
        // const withdrawalRequestsCount = await apiService.getUnviewedWithdrawalRequestsCount();

        if (alertsCount > 0) {
            setNewAlertsFlag(true);
        } else {
            setNewAlertsFlag(false);
        }
        // O setNewWithdrawalRequestsFlag será chamado pelo WebSocket ao receber a notificação
        updateNotificationBellUI(); // Garante que a UI do sino seja atualizada com base em ambos os counts
    } catch (error) {
        console.error('checkAlertsNotification: Erro ao verificar notificações:', error);
        setNewAlertsFlag(false);
        setNewWithdrawalRequestsFlag(false);
    }
}

// Chamada inicial de binding
bindQuickAccessLinks();
bindLogoutLink();

// Lógica de Retiradas (modular) listeners que estão fora do main-content
document.getElementById('listar-retiradas-link')?.addEventListener('click', e => {
    e.preventDefault();
    retiradasModule.renderHistoricoRetiradas();
});

document.getElementById('listar-retiradas-pendentes-link')?.addEventListener('click', e => {
    e.preventDefault();
    retiradasModule.renderPendentesRetiradas();
});

```

```

const btnConfirmarAutorizar = document.getElementById('btn-confirmar-autorizar-retirada');
if (btnConfirmarAutorizar) {
    btnConfirmarAutorizar.addEventListener('click', retiradasModule._handleAuthorizeDeny.bind(retiradasModule));
}

const btnConfirmarNegar = document.getElementById('btn-confirmar-negar-retirada');
if (btnConfirmarNegar) {
    btnConfirmarNegar.addEventListener('click', retiradasModule._handleAuthorizeDeny.bind(retiradasModule));
}

// Inicializar os módulos
solicitarRetiradaModule.init();
selecionarItemModule.init();
reportsModule.init();
alertasModule.init();

// Iniciar a verificação inicial de alertas (ao carregar a página)
checkAlertsNotification();

// Conectar ao WebSocket para receber alertas em tempo real
let ws; // Variável para a instância do WebSocket

function connectAlertsWebSocket() {
    // Usa wss:// para HTTPS e ws:// para HTTP
    const protocol = window.location.protocol === 'https:' ? 'wss:' : 'ws:';
    // URL do WebSocket para corresponder ao endpoint do backend
    const wsUrl = `${protocol}://${window.location.host}/api/almoxarifado/ws/alerts`;

    ws = new WebSocket(wsUrl);

    ws.onopen = (event) => {
        console.log("WebSocket para notificações conectado:", event);
    };

    ws.onmessage = (event) => {
        const message = JSON.parse(event.data);
        console.log("Mensagem WebSocket recebida:", message);

        if (message.type === "new_alert") {
            setNewAlertsFlag(true);
            showAlert("Novo alerta: " + message.message, "info", 5000);
            // Tocar o som de notificação
            try {
                const audio = new Audio(NOTIFICATION_SOUND_PATH);
                audio.play().catch(e => console.error("Erro ao tocar som de notificação:", e));
            } catch (e) {
                console.error("Não foi possível criar objeto de áudio para notificação:", e);
            }
        } else if (message.type === "new_withdrawal_request") {
            setNewWithdrawalRequestsFlag(true);
            showAlert("Nova solicitação de retirada: " + message.message, "primary", 5000);
            // Tocar o som de notificação
            try {
                const audio = new Audio(NOTIFICATION_SOUND_PATH_RETIRADA);
                audio.play().catch(e => console.error("Erro ao tocar som de notificação:", e));
            } catch (e) {
                console.error("Não foi possível criar objeto de áudio para notificação:", e);
            }
        }
        updateNotificationBellUI(); // Garante que o sino seja atualizado
    };

    ws.onclose = (event) => {

```

```

        console.warn("WebSocket para notificações desconectado:", event.code, event.reason);
        setTimeout(connectAlertsWebSocket, 5000); // Tenta reconectar a cada 5 segundos
    };

    ws.onerror = (error) => {
        console.error("Erro no WebSocket para notificações:", error);
        ws.close();
    };
}

// Chama a função para conectar ao WebSocket quando o DOM estiver pronto
connectAlertsWebSocket();

// Re-inicializa os dropdowns do Bootstrap uma vez que o DOM está pronto e todos os scripts foram car
reinitializeBootstrapDropdowns();
});

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\reportsModule.js

```

// frontend/static/js/reportsModule.js

import { apiService } from './apiService.js';
import { uiService } from './uiService.js';
import { showAlert } from './utils.js';
import { selecionarItemModule } from './selecionar-item-module.js';

class ReportsModule {
    constructor() {
        // References to main report dashboard modal
        this.modalReportsDashboard = uiService.getModalInstance('modalReportsDashboard');

        // References for Quantidade de Itens report modal
        this.modalReportQuantidadeItens = uiService.getModalInstance('modalReportQuantidadeItens');
        this.formReportQuantidadeItens = document.getElementById('form-report-quantidade-itens');
        this.btnGerarReportQuantidadeItens = document.getElementById('btnGerarReportQuantidadeItens');
        this.filterCategoryReportSelect = document.getElementById('filterCategoryReport');

        // Novas referências para seleção de produto
        this.selectedProductNameReportInput = document.getElementById('selectedProductNameReport');
        this.selectedProductIdReportInput = document.getElementById('selectedProductIdReport'); // Campo
        this.btnOpenSelectProductReportModal = document.getElementById('btnOpenSelectProductReportModal');
        this.btnClearSelectedProductReport = document.getElementById('btnClearSelectedProductReport');

        // References for Entrada de Itens report modal
        this.modalReportEntradaItens = uiService.getModalInstance('modalReportEntradaItens');
        this.formReportEntradaItens = document.getElementById('form-report-entrada-itens');
        this.btnGerarReportEntradaItens = document.getElementById('btnGerarReportEntradaItens');

        // References for Retiradas por Setor report modal
        this.modalReportRetiradasSetor = uiService.getModalInstance('modalReportRetiradasSetor');
        this.formReportRetiradasSetor = document.getElementById('form-report-retiradas-setor');
        this.selectSetor = document.getElementById('selectSetor');
        this.btnGerarReportRetiradasSetor = document.getElementById('btnGerarReportRetiradasSetor');

        // References for Retiradas por Usuário report modal
        this.modalReportRetiradasUsuario = uiService.getModalInstance('modalReportRetiradasUsuario');
        this.formReportRetiradasUsuario = document.getElementById('form-report-retiradas-usuario');
        this.selectUsuario = document.getElementById('selectUsuario');
        this.btnGerarReportRetiradasUsuario = document.getElementById('btnGerarReportRetiradasUsuario');
    }
}

```

```

init() {
    this._bindEvents();
}

_bindEvents() {
    // --- Event listeners for generating reports ---
    if (this.btnGerarReportQuantidadeItens) {
        this.btnGerarReportQuantidadeItens.addEventListener('click', () => this._handleGerarReportQua
    }

    if (this.btnGerarReportEntradaItens) {
        this.btnGerarReportEntradaItens.addEventListener('click', () => this._handleGerarReportEntra
    }

    if (this.btnGerarReportRetiradasSetor) {
        this.btnGerarReportRetiradasSetor.addEventListener('click', () => this._handleGerarReportRet
    }

    if (this.btnGerarReportRetiradasUsuario) {
        this.btnGerarReportRetiradasUsuario.addEventListener('click', () => this._handleGerarReportR
    }

    // --- Event listeners for modal show/hide to handle dynamic content and resets ---
    if (this.modalReportRetiradasSetor) {
        this.modalReportRetiradasSetor._element.addEventListener('show.bs.modal', () => this._popula
        this.modalReportRetiradasSetor._element.addEventListener('hidden.bs.modal', () => this.formR
    }
    if (this.modalReportRetiradasUsuario) {
        this.modalReportRetiradasUsuario._element.addEventListener('show.bs.modal', () => this._popu
        this.modalReportRetiradasUsuario._element.addEventListener('hidden.bs.modal', () => this.for
    }
    if (this.modalReportQuantidadeItens) {
        this.modalReportQuantidadeItens._element.addEventListener('show.bs.modal', () => this._popul
        this.modalReportQuantidadeItens._element.addEventListener('hidden.bs.modal', () => {
            this.formReportQuantidadeItens.reset();
            this._clearSelectedProductFilter();
        });
    }

    // Eventos para o novo botão de seleção de produto e limpeza
    if (this.btnOpenSelectProductReportModal) {
        this.btnOpenSelectProductReportModal.addEventListener('click', () => {
            this.modalReportQuantidadeItens.hide();
            // Opcional: Adicione um listener temporário para reabrir o modal
            // quando o modal de seleção for escondido, independente de seleção.
            const reOpenReportModal = () => {
                this.modalReportQuantidadeItens.show();
                // Remove o listener temporário para evitar que ele reabra o modal em outros con
                selecionarItemModule.modalSelecionarItem._element.removeEventListener('hidden.bs
            };
            selecionarItemModule.modalSelecionarItem._element.addEventListener('hidden.bs.modal'

            setTimeout(() => {
                selecionarItemModule.openModal('itemSelectedForReportFilter', 10);
            }, 300);
        });
    }
    if (this.btnClearSelectedProductReport) {
        this.btnClearSelectedProductReport.addEventListener('click', () => this._clearSelectedPr
    }
}
if (this.modalReportEntradaItens) {
    this.modalReportEntradaItens._element.addEventListener('hidden.bs.modal', () => this.formRep
}

```

```

// Listener para o evento customizado disparado pelo selecionarItemModule (quando um item É sele
document.addEventListener('itemSelectedForReportFilter', (event) => {
    this._handleItemSelectedForReportFilter(event.detail.item);
    // O modal de seleção já se fechou ao despachar o evento.
    // O listener 'hidden.bs.modal' adicionado acima já cuidará de reabrir o modal de relatório.
});
}

async _populateSetores() {
    this.selectSetor.innerHTML = '<option value="" disabled selected>Carregando setores...</option>';
    uiService.showLoading();
    try {
        const setores = await apiService.fetchAllSetores();
        this.selectSetor.innerHTML = '<option value="" disabled selected>Selecione um setor</option>';
        setores.forEach(setor => {
            const option = document.createElement('option');
            option.value = setor.setor_id;
            option.textContent = setor.nome_setor;
            this.selectSetor.appendChild(option);
        });
    } catch (error) {
        console.error('Erro ao carregar setores:', error);
        showAlert('Erro ao carregar setores para o relatório.', 'danger');
        this.selectSetor.innerHTML = '<option value="" disabled selected>Erro ao carregar</option>';
    } finally {
        uiService.hideLoading();
    }
}

async _populateUsuarios() {
    this.selectUsuario.innerHTML = '<option value="" disabled selected>Carregando usuários...</option>';
    uiService.showLoading();
    try {
        const usuarios = await apiService.get('/usuarios');
        this.selectUsuario.innerHTML = '<option value="" disabled selected>Selecione um usuário</option>';
        usuarios.forEach(usuario => {
            const option = document.createElement('option');
            option.value = usuario.usuario_id;
            option.textContent = usuario.nome_usuario;
            this.selectUsuario.appendChild(option);
        });
    } catch (error) {
        console.error('Erro ao carregar usuários:', error);
        showAlert('Erro ao carregar usuários para o relatório.', 'danger');
        this.selectUsuario.innerHTML = '<option value="" disabled selected>Erro ao carregar</option>';
    } finally {
        uiService.hideLoading();
    }
}

async _populateCategoriesForReport() {
    this.filterCategoryReportSelect.innerHTML = '<option value="">Todas as Categorias</option>';
    uiService.showLoading();
    try {
        const categorias = await apiService.get('/categorias');
        categorias.forEach(cat => {
            const option = document.createElement('option');
            option.value = cat.categoria_id;
            option.textContent = cat.nome_original;
            this.filterCategoryReportSelect.appendChild(option);
        });
    } catch (error) {
        console.error('Erro ao carregar categorias para o relatório:', error);
    }
}

```



```

        showAlert('Erro ao carregar categorias.', 'danger');
    } finally {
        uiService.hideLoading();
    }
}

_clearSelectedProductFilter() {
    this.selectedProductNameReportInput.value = '';
    this.selectedProductIdReportInput.value = '';
}

_handleItemSelectedForReportFilter(selectedItem) {
    if (selectedItem) {
        this.selectedProductNameReportInput.value = selectedItem.nome_item_original;
        this.selectedProductIdReportInput.value = selectedItem.item_id;
        showAlert('Produto "${selectedItem.nome_item_original}" selecionado para filtro.', 'info');
    } else {
        this._clearSelectedProductFilter();
        showAlert('Nenhum produto selecionado.', 'warning');
    }
}

async _handleGerarReportQuantidadeItens() {
    if (!this.formReportQuantidadeItens.checkValidity()) {
        this.formReportQuantidadeItens.reportValidity();
        return;
    }

    const filtro_categoria = this.filterCategoryReportSelect.value;
    const filtro_produto_id = this.selectedProductIdReportInput.value;
    const formato = document.getElementById('formatoQuantidadeItens').value;

    const params = new URLSearchParams({ formato });
    if (filtro_categoria) params.append('filtro_categoria', filtro_categoria);
    if (filtro_produto_id) {
        params.append('filtro_produto', this.selectedProductNameReportInput.value); // Envia o nome,
    }

    uiService.showLoading();
    try {
        const token = localStorage.getItem('token');
        const response = await fetch(`/api/almoxarifado/relatorios/quantidade-itens/?${params.toString()}`, {
            method: 'GET',
            headers: {
                'Authorization': `Bearer ${token}`
            }
        });

        if (!response.ok) {
            const errorData = await response.json().catch(() => ({ detail: 'Erro desconhecido ao gerar relatório' }));
            throw new Error(errorData.detail || `Erro HTTP: ${response.status}`);
        }

        const blob = await response.blob();
        const contentDisposition = response.headers.get('content-disposition');
        let filename = `relatorio_quantidade_itens.${formato}`;

        if (contentDisposition) {
            const filenameMatch = contentDisposition.match(/filename*?=(?:UTF-8'')?([^\s;]+)/i);
            if (filenameMatch && filenameMatch[1]) {
                filename = decodeURIComponent(filenameMatch[1].replace(/"/g, ''));
            } else {
                const simpleSplit = contentDisposition.split('filename=')[1];
            }
        }
    }
}

```

```

        if (simpleSplit) {
            filename = simpleSplit.replace(/^"|"$/g, '');
        }
    }

    this._downloadFile(blob, filename);

    showAlert('Relatório de quantidade de itens gerado com sucesso!', 'success');
    this.modalReportQuantidadeItens.hide();
} catch (error) {
    console.error('Erro ao gerar relatório de quantidade de itens:', error);
    showAlert(error.message || 'Ocorreu um erro ao gerar o relatório de quantidade de itens.', 'error');
} finally {
    uiService.hideLoading();
}
}

async _handleGerarReportEntradaItens() {
    if (!this.formReportEntradaItens.checkValidity()) {
        this.formReportEntradaItens.reportValidity();
        return;
    }

    const data_inicio = document.getElementById('dataInicioEntrada').value;
    const data_fim = document.getElementById('dataFimEntrada').value;
    const formato = document.getElementById('formatoEntradaItens').value;

    if (!data_inicio || !data_fim) {
        showAlert('Por favor, preencha as datas inicial e final.', 'warning');
        return;
    }

    const params = new URLSearchParams({ data_inicio, data_fim, formato });

    uiService.showLoading();
    try {
        const token = localStorage.getItem('token');
        const response = await fetch(`/api/almoxarifado/relatorios/entrada-itens/?${params.toString()}`, {
            method: 'GET',
            headers: {
                'Authorization': `Bearer ${token}`
            }
        });

        if (!response.ok) {
            const errorData = await response.json().catch(() => ({ detail: 'Erro desconhecido ao gerar relatório' }));
            throw new Error(errorData.detail || `Erro HTTP: ${response.status}`);
        }

        const blob = await response.blob();
        const contentDisposition = response.headers.get('content-disposition');
        let filename = `relatorio_entrada_itens.${formato}`;

        if (contentDisposition) {
            const filenameMatch = contentDisposition.match(/filename*=?(?:UTF-8'')?(^[^;]+)/i);
            if (filenameMatch && filenameMatch[1]) {
                filename = decodeURIComponent(filenameMatch[1].replace(/^"|"$/g, ''));
            } else {
                const simpleSplit = contentDisposition.split('filename=')[1];
                if (simpleSplit) {
                    filename = simpleSplit.replace(/^"|"$/g, '');
                }
            }
        }
    }
}

```

```

    }
  }

  this._downloadFile(blob, filename);

  showAlert('Relatório de entrada de itens gerado com sucesso!', 'success');
  this.modalReportEntradaItens.hide();
} catch (error) {
  console.error('Erro ao gerar relatório de entrada de itens:', error);
  showAlert(error.message || 'Ocorreu um erro ao gerar o relatório de entrada de itens.', 'danger');
} finally {
  uiService.hideLoading();
}
}

async _handleGerarReportRetiradasSetor() {
  if (!this.formReportRetiradasSetor.checkValidity()) {
    this.formReportRetiradasSetor.reportValidity();
    return;
  }

  const setor_id = this.selectSetor.value;
  const data_inicio = document.getElementById('dataInicioRetiradaSetor').value;
  const data_fim = document.getElementById('dataFimRetiradaSetor').value;
  const formato = document.getElementById('formatoRetiradasSetor').value;

  if (!setor_id || !data_inicio || !data_fim) {
    showAlert('Por favor, preencha todos os campos obrigatórios.', 'warning');
    return;
  }

  const params = new URLSearchParams({ setor_id, data_inicio, data_fim, formato });

  uiService.showLoading();
  try {
    const token = localStorage.getItem('token');
    const response = await fetch(`/api/almoxarifado/relatorios/retiradas-setor/?${params.toString()}`, {
      method: 'GET',
      headers: {
        'Authorization': `Bearer ${token}`
      }
    });

    if (!response.ok) {
      const errorData = await response.json().catch(() => ({ detail: 'Erro desconhecido ao gerar relatório' }));
      throw new Error(errorData.detail || `Erro HTTP: ${response.status}`);
    }

    const blob = await response.blob();
    const contentDisposition = response.headers.get('content-disposition');
    let filename = `relatorio_retiradas_setor.${formato}`;

    if (contentDisposition) {
      const filenameMatch = contentDisposition.match(/filename\*?=(?:UTF-8'')?([^\s;]+)/i);
      if (filenameMatch && filenameMatch[1]) {
        filename = decodeURIComponent(filenameMatch[1].replace(/"/g, ''));
      } else {
        const simpleSplit = contentDisposition.split('filename=')[1];
        if (simpleSplit) {
          filename = simpleSplit.replace(/"/g, '');
        }
      }
    }
  }
}

```

```

        this._downloadFile(blob, filename);

        showAlert('Relatório de retiradas por setor gerado com sucesso!', 'success');
        this.modalReportRetiradasSetor.hide();
    } catch (error) {
        console.error('Erro ao gerar relatório de retiradas por setor:', error);
        showAlert(error.message || 'Ocorreu um erro ao gerar o relatório de retiradas por setor.', 'error');
    } finally {
        uiService.hideLoading();
    }
}

async _handleGerarReportRetiradasUsuario() {
    if (!this.formReportRetiradasUsuario.checkValidity()) {
        this.formReportRetiradasUsuario.reportValidity();
        return;
    }

    const usuario_id = this.selectUsuario.value;
    const data_inicio = document.getElementById('dataInicioRetiradaUsuario').value;
    const data_fim = document.getElementById('dataFimRetiradaUsuario').value;
    const formato = document.getElementById('formatoRetiradasUsuario').value;

    if (!usuario_id || !data_inicio || !data_fim) {
        showAlert('Por favor, preencha todos os campos obrigatórios.', 'warning');
        return;
    }

    const params = new URLSearchParams({ usuario_id, data_inicio, data_fim, formato });

    uiService.showLoading();
    try {
        const token = localStorage.getItem('token');
        const response = await fetch(`/api/almoxarifado/relatorios/retiradas-usuario/?${params.toString()}`, {
            method: 'GET',
            headers: {
                'Authorization': `Bearer ${token}`
            }
        });

        if (!response.ok) {
            const errorData = await response.json().catch(() => ({ detail: 'Erro desconhecido ao gerar relatório de retiradas por usuário' }));
            throw new Error(errorData.detail || `Erro HTTP: ${response.status}`);
        }

        const blob = await response.blob();
        const contentDisposition = response.headers.get('content-disposition');
        let filename = `relatorio_retiradas_usuario.${formato}`;

        if (contentDisposition) {
            const filenameMatch = contentDisposition.match(/filename\*?=(?:UTF-8'')?([^\s;]+)/i);
            if (filenameMatch && filenameMatch[1]) {
                filename = decodeURIComponent(filenameMatch[1].replace(/"/g, ''));
            } else {
                const simpleSplit = contentDisposition.split('filename=')[1];
                if (simpleSplit) {
                    filename = simpleSplit.replace(/"/g, '');
                }
            }
        }

        this._downloadFile(blob, filename);
        showAlert('Relatório de retiradas por usuário gerado com sucesso!', 'success');
    } catch (error) {
        console.error('Erro ao gerar relatório de retiradas por usuário:', error);
        showAlert(error.message || 'Ocorreu um erro ao gerar o relatório de retiradas por usuário.', 'error');
    } finally {
        uiService.hideLoading();
    }
}

```

```

        this.modalReportRetiradasUsuario.hide();
    } catch (error) {
        console.error('Erro ao gerar relatório de retiradas por usuário:', error);
        showAlert(error.message || 'Ocorreu um erro ao gerar o relatório de retiradas por usuário.',
    } finally {
        uiService.hideLoading();
    }
}

_downloadFile(blob, filename) {
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.href = url;
    a.download = filename;
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
    window.URL.revokeObjectURL(url);
}

}

export const reportsModule = new ReportsModule();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\retiradasModule.js

```

// frontend/static/js/retiradasModule.js

import { apiService } from './apiService.js';
import { dataService } from './dataService.js';
import { uiService } from './uiService.js';
import { showAlert, getStatusText, getStatusValue } from './utils.js';
import estadoGlobal from './estadoGlobal.js';

class RetiradasModule {
    constructor() {
        this.historicoPageSizeSelectId = 'historicoPageSize';
        this.pendentesPageSizeSelectId = 'pendentesPageSize';

        // Referências aos elementos do modal de conclusão (NOVO)
        this.modalConcluirRetirada = uiService.getModalInstance('modalConcluirRetirada');
        this.concluirRetiradaIdInput = document.getElementById('concluirRetiradaId');
        this.concluirRetiradaDisplayId = document.getElementById('concluirRetiradaDisplayId');
        this.concluirDetalheStatusInput = document.getElementById('concluirDetalheStatus');
        this.btnConfirmarConcluirRetirada = document.getElementById('btn-confirmar-concluir-retirada');

        // Bindings para eventos que são removidos e adicionados novamente
        this._boundHandleHistoricoPaginationClick = this._handleHistoricoPaginationClick.bind(this);
        this._boundHandleHistoricoPageSizeChange = this._handleHistoricoPageSizeChange.bind(this);
        this._boundHandleHistoricoFilterSubmit = this._handleHistoricoFilterSubmit.bind(this);
        this._boundHandleHistoricoClearFilters = this._handleHistoricoClearFilters.bind(this);
    }

    async renderHistoricoRetiradas(page = 1, filters = estadoGlobal.currentHistoricoFilters, pageSize =
        uiService.showLoading();
        try {
            const data = await dataService.getProcessedRetiradas(apiService.fetchAllRetiradas.bind(apiSe
                estadoGlobal.setHistoricoPagination(data.current_page, data.total_pages, pageSize, filters);
                estadoGlobal.setAllRetiradas(data.items); // Isso atualiza as retiradas para o histórico

            const filterFormHtml = this._getHistoricoFilterFormHtml(filters);

```

```

const tableHeaders = ['ID', 'Usuário', 'Data', 'Status', 'Ações'];

const tableHtml = uiService.renderTable(tableHeaders, estadoGlobal.allRetiradas, {
  noRecordsMessage: "Nenhum histórico de retirada encontrado.",
  rowMapper: (r) => [
    r.retirada_id,
    r.usuario_nome,
    new Date(r.data_solicitacao).toLocaleDateString('pt-BR'),
    getStatusText(r.status)
  ],
  actionsHtml: (r) => {
    let actions = `
      <button class="btn btn-sm btn-primary btn-detalhes-retirada" data-id="${r.retirada_id}">
        <i class="bi bi-eye"></i> Detalhes
      </button>
    `;
    // Adicionar botão de "Concluir" SOMENTE se o status for AUTORIZADA
    if (r.status === estadoGlobal.statusMapUpdate.AUTORIZADA) {
      actions += `
        <button class="btn btn-sm btn-success btn-concluir-retirada-trigger ms-2" data-id="${r.retirada_id}">
          <i class="bi bi-check-circle"></i> Concluir Retirada
        </button>
      `;
    }
    return actions;
  }
});

const paginationHtml = uiService.renderPagination(
  estadoGlobal.currentHistoricoPage,
  estadoGlobal.totalHistoricoPages,
  'historico',
  this.historicoPageSizeSelectId,
  estadoGlobal.currentHistoricoPageSize
);

uiService.renderPage('Histórico de Retiradas', `
  <div class="card mb-4">
    <div class="card-header">Filtros de Busca</div>
    <div class="card-body">${filterFormHtml}</div>
  </div>
  ${tableHtml}
  ${paginationHtml}
`);

this._bindHistoricoEvents();
this._bindCommonRetiradaActions(); // Mantenha esta chamada
this._bindConcluirRetiradaEvents(); // NOVO: Bind para o modal de conclusão

} catch (error) {
  console.error("Erro ao renderizar histórico de retiradas:", error);
  showAlert(error.message || 'Ocorreu um erro ao carregar o histórico de retiradas.', 'danger');
} finally {
  uiService.hideLoading();
}

}

async renderPendentesRetiradas(page = 1, pageSize = estadoGlobal.currentPendentesPageSize) {
  uiService.showLoading();
  try {
    const data = await dataService.getProcessedRetiradas(apiService.fetchRetiradasPendentes.bind(
      estadoGlobal.setPendentesPagination(data.current_page, data.total_pages, pageSize);
      estadoGlobal.setPendentesRetiradas(data.items); // Isso atualiza as retiradas pendentes
    ));
  }
}

```

```

const tableHeaders = ['ID', 'Usuário', 'Setor', 'Data', 'Ações'];

const tableHtml = uiService.renderTable(tableHeaders, estadoGlobal.pendentesRetiradas, {
  noRecordsMessage: "Nenhuma retirada pendente encontrada.",
  rowMapper: (r) => [
    r.retirada_id,
    r.usuario_nome,
    r.setor_nome,
    new Date(r.data_solicitacao).toLocaleDateString('pt-BR')
  ],
  actionsHtml: (r) => `
    <button class="btn btn-sm btn-success btn-autorizar-retirada-trigger" data-id="${r.retirada_id}">
      <i class="bi bi-check-circle"></i> Autorizar/Negar
    </button>
    <button class="btn btn-sm btn-info btn-detalhes-retirada" data-id="${r.retirada_id}">
      <i class="bi bi-eye"></i> Ver detalhes
    </button>
  `;
});

const paginationHtml = uiService.renderPagination(
  estadoGlobal.currentPendentesPage,
  estadoGlobal.totalPendentesPages,
  'pendentes',
  this.pendentesPageSizeSelectId,
  estadoGlobal.currentPendentesPageSize
);

uiService.renderPage('Retiradas Pendentes', `
  ${tableHtml}
  ${paginationHtml}
`);

this._bindPendentesEvents();
this._bindCommonRetiradaActions();
} catch (error) {
  console.error("Erro ao renderizar retiradas pendentes:", error);
  showAlert(error.message || 'Ocorreu um erro ao carregar as retiradas pendentes.', 'danger');
} finally {
  uiService.hideLoading();
}
}

// NOVO: Lógica para o modal de Concluir Retirada
_bindConcluirRetiradaEvents() {
  // Delegate event for "Concluir Retirada" buttons
  document.querySelectorAll('.btn-concluir-retirada-trigger').forEach(btn => {
    btn.onclick = () => {
      const id = parseInt(btn.dataset.id);
      // Encontra a retirada na lista de todas as retiradas
      const retirada = estadoGlobal.allRetiradas.find(r => r.retirada_id === id);
      if (retirada) {
        uiService.fillModalConcluir(retirada); // Preenche o novo modal
        this.modalConcluirRetirada.show(); // Abre o novo modal
      } else {
        showAlert('Retirada não encontrada para conclusão.', 'warning');
      }
    };
  });
}

// Evento para o botão de confirmação dentro do modal de conclusão
if (this.btnConfirmarConcluirRetirada) {
  this.btnConfirmarConcluirRetirada.onclick = () => {

```

```

        const retiradaId = parseInt(this.concluirRetiradaIdInput.value);
        const detalhe = this.concluirDetalheStatusInput.value.trim();
        this._handleConcluirRetirada(retiradaId, detalhe);
    };
}

_getHistoricoFilterFormHtml(currentFilters) {
    return `
        <form id="form-filter-historico">
            <div class="d-flex flex-wrap align-items-end gap-3 mb-3">
                <div class="flex-grow-1">
                    <label for="filterStatus" class="form-label">Status</label>
                    <select class="form-select" id="filterStatus">
                        <option value="">Todos</option>
                        <option value="PENDENTE" ${currentFilters.status === estadoGlobal.statusMapU}>PENDENTE</option>
                        <option value="AUTORIZADA" ${currentFilters.status === estadoGlobal.statusMa}>AUTORIZADA</option>
                        <option value="CONCLUIDA" ${currentFilters.status === estadoGlobal.statusMap}>CONCLUIDA</option>
                        <option value="NEGADA" ${currentFilters.status === estadoGlobal.statusMapUpd}>NEGADA</option>
                    </select>
                </div>
                <div class="flex-grow-1">
                    <label for="filterSolicitante" class="form-label">Solicitante</label>
                    <input type="text" class="form-control" id="filterSolicitante" value="${currentF}>
                </div>
                <div class="flex-grow-1">
                    <label for="filterStartDate" class="form-label">Data Inicial</label>
                    <input type="date" class="form-control" id="filterStartDate" value="${currentFil}>
                </div>
                <div class="flex-grow-1">
                    <label for="filterEndDate" class="form-label">Data Final</label>
                    <input type="date" class="form-control" id="filterEndDate" value="${currentFilt}>
                </div>
                <div class="d-flex align-items-end gap-2">
                    <button type="submit" class="btn btn-primary" id="btn-search-historico">Buscar</button>
                    <button type="button" class="btn btn-secondary" id="btn-clear-filters">Limpar Fi
                </div>
            </div>
        </form>
    `;
}

_bindHistoricoEvents() {
    // Get the specific pagination nav element and the form
    const historicoPaginationNav = document.getElementById('historico-pagination-nav');
    const formFilter = document.getElementById('form-filter-historico');
    const pageSizeSelect = document.getElementById(this.historicoPageSizeSelectId);
    const btnClearFilters = document.getElementById('btn-clear-filters');

    // Remove previous listeners to avoid duplicates
    if (historicoPaginationNav) {
        historicoPaginationNav.removeEventListener('click', this._boundHandleHistoricoPaginationClick);
    }
    if (pageSizeSelect) {
        pageSizeSelect.removeEventListener('change', this._boundHandleHistoricoPageSizeChange);
    }
    if (formFilter) {
        formFilter.removeEventListener('submit', this._boundHandleHistoricoFilterSubmit);
    }
    if (btnClearFilters) {
        btnClearFilters.removeEventListener('click', this._boundHandleHistoricoClearFilters);
    }
    // Add new listeners

```



```

    if (historicoPaginationNav) {
        historicoPaginationNav.addEventListener('click', this._boundHandleHistoricoPaginationClick);
    }
    if (pageSizeSelect) {
        pageSizeSelect.addEventListener('change', this._boundHandleHistoricoPageSizeChange);
    }
    if (formFilter) {
        formFilter.addEventListener('submit', this._boundHandleHistoricoFilterSubmit);
    }
    if (btnClearFilters) {
        btnClearFilters.addEventListener('click', this._boundHandleHistoricoClearFilters);
    }
}

async _handleHistoricoFilterSubmit(e) {
    e.preventDefault();
    console.log("Filtro de histórico submetido!"); // Debug
    const selectedStatusString = document.getElementById('filterStatus').value;
    const statusInt = selectedStatusString ? estadoGlobal.statusMapUpdate[selectedStatusString] : null;
    const filters = {
        status: statusInt,
        solicitante: document.getElementById('filterSolicitante').value.trim(),
        start_date: document.getElementById('filterStartDate').value,
        end_date: document.getElementById('filterEndDate').value,
    };
    console.log("Filtros aplicados:", filters); // Debug
    this.renderHistoricoRetiradas(1, filters, estadoGlobal.currentHistoricoPageSize);
}

_handleHistoricoClearFilters() {
    console.log("Limpar filtros de histórico clicado!"); // Debug
    document.getElementById('filterStatus').value = '';
    document.getElementById('filterSolicitante').value = '';
    document.getElementById('filterStartDate').value = '';
    document.getElementById('filterEndDate').value = '';
    this.renderHistoricoRetiradas(1, {}, estadoGlobal.currentHistoricoPageSize);
}

_handleHistoricoPaginationClick(e) {
    e.preventDefault();
    // Look for links with data-page starting with 'historico-'
    const clickedPageLink = e.target.closest('a[data-page^="historico-"]');
    const clickedActionButton = e.target.closest('a[data-action^="historico-"]');

    if (clickedPageLink) {
        // Extract the page number from the data-page attribute (e.g., "historico-4" -> "4")
        const pageValue = clickedPageLink.dataset.page.split('-')[1];
        const newPage = parseInt(pageValue);
        if (!isNaN(newPage) && newPage !== estadoGlobal.currentHistoricoPage) {
            this.renderHistoricoRetiradas(newPage, estadoGlobal.currentHistoricoFilters, estadoGlobal.currentHistoricoPageSize);
        }
        return;
    }

    if (clickedActionButton) {
        const action = clickedActionButton.dataset.action;
        let newPage = estadoGlobal.currentHistoricoPage;

        if (action === 'historico-prev') {
            if (newPage > 1) newPage--;
        } else if (action === 'historico-next') {
            if (newPage < estadoGlobal.totalHistoricoPages) newPage++;
        }
    }
}

```

```

        if (newPage !== estadoGlobal.currentHistoricoPage) {
            this.renderHistoricoRetiradas(newPage, estadoGlobal.currentHistoricoFilters, estadoGlobal.currentHistoricoPageSize);
        }
        return;
    }
}

_handleHistoricoPageSizeChange(e) {
    if (e.target.id === this.historicoPageSizeSelectId) {
        const newPageSize = parseInt(e.target.value);
        this.renderHistoricoRetiradas(1, estadoGlobal.currentHistoricoFilters, newPageSize);
    }
}

_bindPendientesEvents() {
    const pendientesPaginationNav = document.getElementById('pendientes-pagination-nav');
    const pageSizeSelect = document.getElementById(this.pendientesPageSizeSelectId);

    if (!pendientesPaginationNav) return;

    // Remove previous listeners to prevent duplicates
    pendientesPaginationNav.removeEventListener('click', this._boundHandlePendientesPaginationClick);
    if (pageSizeSelect) {
        pageSizeSelect.removeEventListener('change', this._boundHandlePendientesPageSizeChange); // A
    }

    // Add new listeners
    pendientesPaginationNav.addEventListener('click', this._boundHandlePendientesPaginationClick);
    if (pageSizeSelect) {
        pageSizeSelect.addEventListener('change', this._boundHandlePendientesPageSizeChange);
    }
}

// New bound handler for pendientes pagination clicks
_boundHandlePendientesPaginationClick = (e) => {
    e.preventDefault();
    const clickedPageLink = e.target.closest('a[data-page^="pendientes-"]');
    const clickedActionButton = e.target.closest('a[data-action^="pendientes-"]');

    if (clickedPageLink) {
        const pageValue = clickedPageLink.dataset.page.split('-')[1];
        const newPage = parseInt(pageValue);
        if (!isNaN(newPage) && newPage !== estadoGlobal.currentPendientesPage) {
            this.renderPendientesRetiradas(newPage, estadoGlobal.currentPendientesPageSize);
        }
        return;
    }

    if (clickedActionButton) {
        const action = clickedActionButton.dataset.action;
        let newPage = estadoGlobal.currentPendientesPage;

        if (action === 'pendientes-prev') {
            if (newPage > 1) newPage--;
        } else if (action === 'pendientes-next') {
            if (newPage < estadoGlobal.totalPendientesPages) newPage++;
        }

        if (newPage !== estadoGlobal.currentPendientesPage) {
            this.renderPendientesRetiradas(newPage, estadoGlobal.currentPendientesPageSize);
        }
        return;
    }
}

```

```

}

// New bound handler for pendentes page size changes
_boundHandlePendentesPageSizeChange = (e) => {
  if (e.target.id === this.pendentesPageSizeSelectId) {
    const newPageSize = parseInt(e.target.value);
    this.renderPendentesRetiradas(1, newPageSize);
  }
}

_bindCommonRetiradaActions() {
  // Remove previous listeners to avoid duplicates for common actions
  document.querySelectorAll('.btn-detahes-retirada').forEach(btn => {
    btn.removeEventListener('click', this._boundHandleDetahesRetiradaClick);
  });
  document.querySelectorAll('.btn-autorizar-retirada-trigger').forEach(btn => {
    btn.removeEventListener('click', this._boundHandleAutorizarRetiradaClick);
  });

  // Add new listeners for common actions
  document.querySelectorAll('.btn-detahes-retirada').forEach(btn => {
    btn.addEventListener('click', this._boundHandleDetahesRetiradaClick);
  });

  document.querySelectorAll('.btn-autorizar-retirada-trigger').forEach(btn => {
    btn.addEventListener('click', this._boundHandleAutorizarRetiradaClick);
  });
}

// Bound handler for detalhes retirada button
_boundHandleDetahesRetiradaClick = (e) => {
  const id = parseInt(e.currentTarget.dataset.id);
  const retirada = [...estadoGlobal.allRetiradas, ...estadoGlobal.pendentesRetiradas].find(r => r.id === id);
  if (retirada) {
    uiService.fillModalDetahes(retirada);
    uiService.getModalInstance('modalVerDetahesRetirada').show();
  } else {
    showAlert('Detahes da retirada não encontrados.', 'warning');
  }
}

// Bound handler for autorizar retirada button
_boundHandleAutorizarRetiradaClick = (e) => {
  const id = parseInt(e.currentTarget.dataset.id);
  const retirada = estadoGlobal.pendentesRetiradas.find(x => x.retirada_id === id);
  if (retirada) {
    uiService.fillModalAutorizar(retirada);
    uiService.getModalInstance('modalAutorizarRetirada').show();
  } else {
    showAlert('Retirada pendente não encontrada para autorização.', 'warning');
  }
}

async _handleAuthorizeDeny(action, e) {
  const id = parseInt(e.currentTarget.dataset.id);
  const detalheInput = document.getElementById('autorizarDetalheStatus');
  const detalhe = detalheInput.value.trim();
  const statusValue = getStatusValue(action);

  if (action === 'NEGADA' && !detalhe) {
    showAlert('O detalhe do status (justificativa da negação) é obrigatório ao negar.', 'warning');
    detalheInput.focus();
  }
}

```

```

        return;
    }

    try {
        await apiService.updateRetiradaStatus(id, statusValue, detalhe);
        showAlert(`Retirada ${action === 'AUTORIZADA' ? 'autorizada' : 'negada'} com sucesso!`, 'success');
        uiService.getModalInstance('modalAutorizarRetirada').hide();
        this.renderPendentesRetiradas(estadoGlobal.currentPendentesPage, estadoGlobal.currentPendentesPage);
    } catch (error) {
        showAlert(error.message || `Erro ao ${action === 'AUTORIZADA' ? 'autorizar' : 'negar'} retirada`, 'danger');
    }
}

// NOVO: Função para lidar com a conclusão de uma retirada
async _handleConcluirRetirada(retiradaId, detalhe) {
    uiService.showLoading();
    try {
        await apiService.updateRetiradaStatus(retiradaId, estadoGlobal.statusMapUpdate.CONCLUIDA, detalhe);
        showAlert('Retirada marcada como "Concluída" com sucesso! Estoque decrementado.', 'success');
        this.modalConcluirRetirada.hide(); // Fecha o modal de conclusão

        // Recarrega a tela de histórico para refletir a mudança de status
        this.renderHistoricoRetiradas(estadoGlobal.currentHistoricoPage, estadoGlobal.currentHistoricoPage);
    } catch (error) {
        console.error("Erro ao concluir retirada", error);
        showAlert(error.message || 'Erro ao concluir a retirada. Verifique o estoque!', 'danger');
    } finally {
        uiService.hideLoading();
    }
}

export const retiradasModule = new RetiradasModule();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\selecionar-item-module.js

```

// frontend/static/js/selecionar-item-module.js

import { apiService } from './apiService.js';
import { uiService } from './uiService.js';
import { showAlert } from './utils.js';

class SelecionarItemModule {
    constructor() {
        this.modalSelecionarItem = uiService.getModalInstance('modalSelecionarItemRetirada');

        this.searchForm = document.getElementById('form-search-item-retirada');
        this.searchItemNameInput = document.getElementById('searchItemName');
        this.btnSearchItem = document.getElementById('btn-search-item');
        this.btnClearItemSearch = document.getElementById('btn-clear-item-search');

        this.itensListContainer = document.getElementById('itens-list-container');
        this.itensPaginationContainer = document.getElementById('itens-pagination-container');

        this.currentPage = 1;
        this.pageSize = 10; // Default page size
        this.currentSearchTerm = '';

        this.callerEventName = 'itemSelectedForRetirada';
        this.currentTotalPages = 1; // Novo atributo para armazenar o total de páginas
    }
}

```

```

}

init() {
  this._bindEvents();
}

_bindEvents() {
  this.searchForm.addEventListener('submit', (e) => {
    e.preventDefault();
    this.currentSearchTerm = this.searchItemNameInput.value.trim();
    this.currentPage = 1;
    this.loadItems();
  });

  this.btnClearItemSearch.addEventListener('click', () => {
    this.searchItemNameInput.value = '';
    this.currentSearchTerm = '';
    this.currentPage = 1;
    this.loadItems();
  });

  this.itensPaginationContainer.addEventListener('click', (e) => {
    e.preventDefault(); // Previne o comportamento padrão do link para todos os cliques de paginação

    // Lógica para links de página individuais (números)
    // Agora, procurando por data-page, consistente com listar-itens.js e a alteração no uiService
    const clickedPageLink = e.target.closest('a[data-page]');
    if (clickedPageLink) {
      const pageValue = clickedPageLink.dataset.page; // Lendo data-page
      const parsedPage = parseInt(pageValue);

      if (isNaN(parsedPage)) {
        console.error('ERRO: parseInt retornou NaN para o valor de pageValue:', pageValue);
        showAlert('Erro na paginação: o número da página não é válido.', 'danger');
        return;
      }

      this.currentPage = parsedPage;
      this.loadItems();
      return;
    }

    // Lógica para botões "Anterior" ou "Próximo" (mantém data-action="${type}-prev/next")
    const clickedActionButton = e.target.closest('a[data-action]');
    if (clickedActionButton) {
      const action = clickedActionButton.dataset.action;
      let newPage = this.currentPage;

      const totalPages = this.currentTotalPages;

      if (action.includes('prev')) { // Ainda verificará se action.includes('itemSearch-prev')
        if (newPage > 1) newPage--;
      } else if (action.includes('next')) { // Ainda verificará se action.includes('itemSearch-next')
        if (newPage < totalPages) newPage++;
      }

      if (newPage !== this.currentPage) {
        this.currentPage = newPage;
        this.loadItems();
      }
    }
  });

  this.itensPaginationContainer.addEventListener('change', (e) => {

```

```

        if (e.target.id === 'itemSearchPageSize') {
            this.pageSize = parseInt(e.target.value);
            this.currentPage = 1; // Reseta para a primeira página
            this.loadItems();
        }
    });

    this.itensListContainer.addEventListener('click', (e) => {
        if (e.target.classList.contains('btn-selecionar-item')) {
            const itemId = parseInt(e.target.dataset.id);
            this.selectItem(itemId);
        }
    });

    document.getElementById('modalSelecionarItemRetirada').addEventListener('hidden.bs.modal', () => {
        this.searchItemNameInput.value = '';
        this.currentSearchTerm = '';
        this.currentPage = 1;
        this.itensListContainer.innerHTML = '';
        this.itensPaginationContainer.innerHTML = '';
        this.callerEventName = 'itemSelectedForRetirada'; // Reset to default
        this.currentTotalPages = 1; // Reset total pages on modal hide
    });
}

async openModal(eventName = 'itemSelectedForRetirada', pageSize = this.pageSize) {
    this.callerEventName = eventName;
    this.pageSize = pageSize;

    this.currentPage = 1;
    this.currentSearchTerm = '';
    this.searchItemNameInput.value = '';

    await this.loadItems();
    this.modalSelecionarItem.show();
}

async loadItems() {
    uiService.showLoading();
    try {
        const data = await apiService.searchItems(this.currentSearchTerm, null, this.currentPage, this.pageSize);

        this.currentTotalPages = data.total_pages;

        const tableHeaders = ['Nome', 'Ações'];

        this.itensListContainer.innerHTML = uiService.renderTable(tableHeaders, data.items, {
            noRecordsMessage: "Nenhum item encontrado.",
            rowMapper: (item) => [
                item.nome_item_original
            ],
            actionsHtml: (item) => `
                <button class="btn btn-sm btn-primary btn-selecionar-item" data-id="${item.item_id}">
                    Selecionar
                </button>
            `
        });

        this.itensPaginationContainer.innerHTML = uiService.renderPagination(
            data.page,
            data.total_pages,
            'itemSearch', // Este 'type' ainda será usado para data-action="itemSearch-prev/next"
            'itemSearchPageSize',
        );
    }
}

```

```

        this.pageSize
    );

    } catch (error) {
        console.error("Erro ao carregar itens para seleção:", error);
        showAlert(error.message || 'Erro ao carregar itens disponíveis.', 'danger');
        this.itensListContainer.innerHTML = '<p class="text-danger">Não foi possível carregar os ite
        this.itensPaginationContainer.innerHTML = '';
    } finally {
        uiService.hideLoading();
    }
}

async selectItem(itemId) {
    uiService.showLoading();
    try {
        const item = await apiService.getItemById(itemId);

        const event = new CustomEvent(this.callerEventName, {
            detail: { item: item }
        });
        document.dispatchEvent(event);

        this.modalSelecionarItem.hide();
        showAlert(`Item "${item.nome_item_original}" selecionado com sucesso!`, 'success');
    } catch (error) {
        console.error("Erro ao selecionar item:", error);
        showAlert(error.message || 'Erro ao selecionar o item.', 'danger');
    } finally {
        uiService.hideLoading();
    }
}

}

export const selecionarItemModule = new SelecionarItemModule();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxa rifado\_ets\frontend\static\js\solicitar-retirada.js

```

// frontend/static/js/solicitar-retirada.js
import { apiService } from './apiService.js';
import { uiService } from './uiService.js';
import { showAlert } from './utils.js';
import { selecionarItemModule } from './selecionar-item-module.js';

class SolicitarRetiradaModule {
    constructor() {
        this.modalSolicitarRetirada = uiService.getModalInstance('modalSolicitarRetirada');
        this.form = document.getElementById('form-solicitar-retirada');
        this.selectSetor = document.getElementById('solicitar_setor_id');
        this.inputSolicitadoLocalmentePor = document.getElementById('solicitado_localmente_por');
        this.inputJustificativa = document.getElementById('solicitar_justificativa');

        this.btnAbrirSelecionarItemModal = document.getElementById('btn-abrir-selecionar-item-modal');
        this.inputQuantidadeAddItem = document.getElementById('quantidade_add_item');
        this.btnAdicionarItemRetirada = document.getElementById('btn-adicionar-item-retirada');

        this.itensParaRetiradaContainer = document.getElementById('itens-para-retirada-container');
        this.btnSalvarSolicitacaoRetirada = document.getElementById('btn-salvar-solicitacao-retirada');
        this.noItemsMessage = document.getElementById('no-items-message');
        // NOVAS REFERÊNCIAS PARA FEEDBACK VISUAL
    }
}

```

```

        this.selectedItemDisplay = document.getElementById('selected-item-display'); //
        this.selectedItemName = document.getElementById('selected-item-name'); //
        // Linha removida: this.selectedItemStock = document.getElementById('selected-item-stock');

        this.itensSelecionados = [];
        this.todosItensDisponiveis = [];
        this.currentItemToAddToCart = null;
    }

    init() {
        this._bindEvents();
    }

    _bindEvents() {
        document.getElementById('btn-open-solicitar-retirada')?.addEventListener('click', e => {
            e.preventDefault();
            this.openModal();
        });

        this.btnAbrirSelecionarItemModal?.addEventListener('click', () => {
            this.openSelecionarItemModal();
        });

        this.btnSalvarSolicitacaoRetirada.addEventListener('click', () => this._enviarSolicitacao());

        this.btnAddicionarItemRetirada?.addEventListener('click', () => this._adicionarItemParaRetirada());

        document.getElementById('modalSolicitarRetirada').addEventListener('hidden.bs.modal', () => this

        document.addEventListener('itemSelectedForRetirada', (event) => {
            this.handleItemSelected(event.detail.item);
        });
    }

    async openModal() {
        uiService.showLoading();
        try {
            const setoresData = await apiService.fetchAllSetores();

            this.selectSetor.innerHTML = '<option value="" disabled selected>Selecione um setor</option>';
            setoresData.forEach(setor => {
                const option = document.createElement('option');
                option.value = setor.setor_id;
                option.textContent = setor.nome_setor;
                this.selectSetor.appendChild(option);
            });

            this.itensSelecionados = [];
            this._renderItensParaRetirada();
            this._clearSelectedItemDisplay(); //

            uiService.hideLoading();
            this.modalSolicitarRetirada.show();

        } catch (error) {
            uiService.hideLoading();
            console.error("Erro ao carregar dados iniciais para solicitação:", error);
            showAlert('Erro ao carregar setores. Tente novamente.', 'danger');
            this.modalSolicitarRetirada.hide();
        }
    }

    openSelecionarItemModal() {

```



```

        selecionarItemModule.openModal();
    }

    handleItemSelected(selectedItem) {
        this.currentItemToAddToCart = selectedItem;
        this.inputQuantidadeAddItem.value = 1;

        // ATUALIZA O FEEDBACK VISUAL
        this.selectedItemName.textContent = selectedItem.nome_item_original; //
        // Linha removida: this.selectedItemStock.textContent = selectedItem.quantidade_item;
        this.selectedItemDisplay.style.display = 'block'; //
        this.btnAdicionarItemRetirada.disabled = false; //

        showAlert(`Item selecionado: ${selectedItem.nome_item_original}. Agora, informe a quantidade.`);
    }

    _adicionarItemParaRetirada() {
        if (!this.currentItemToAddToCart) {
            showAlert('Nenhum item selecionado para adicionar.', 'warning');
            return;
        }

        const quantidade = parseInt(this.inputQuantidadeAddItem.value);

        if (isNaN(quantidade) || quantidade <= 0) {
            showAlert('Por favor, informe uma quantidade válida.', 'warning');
            return;
        }

        const itemExistente = this.itensSelecionados.find(item => item.item_id === this.currentItemToAddToCart.item_id);

        const quantidadeDisponivel = this.currentItemToAddToCart.quantidade_item;
        const quantidadeTotalSolicitada = quantidade + (itemExistente ? itemExistente.quantidade_retirada : 0);

        if (quantidadeTotalSolicitada > quantidadeDisponivel) {
            showAlert(`Quantidade solicitada para ${this.currentItemToAddToCart.nome_item_original} excede a disponivel.`);
            return;
        }

        if (itemExistente) {
            itemExistente.quantidade_retirada += quantidade;
        } else {
            this.itensSelecionados.push({
                item_id: this.currentItemToAddToCart.item_id,
                nome_item: this.currentItemToAddToCart.nome_item_original,
                quantidade_retirada: quantidade
            });
        }

        this._renderItensParaRetirada();
        this._clearSelectedItemDisplay(); //
        this.currentItemToAddToCart = null;
        this.inputQuantidadeAddItem.value = 1;
    }

    _renderItensParaRetirada() {
        this.itensParaRetiradaContainer.innerHTML = '';

        if (this.itensSelecionados.length === 0) {
            this.noItemsMessage.style.display = 'block';
            this.itensParaRetiradaContainer.appendChild(this.noItemsMessage);
            return;
        } else {

```

```

        this.noItemsMessage.style.display = 'none';
    }

    this.itensSelecionados.forEach(item => {
        const itemElement = document.createElement('div');
        itemElement.className = 'col-12 d-flex justify-content-between align-items-center bg-light p-2';
        itemElement.innerHTML = `
            <span>${item.nome_item} - Quantidade: ${item.quantidade_retirada}</span>
            <button type="button" class="btn btn-sm btn-danger btn-remove-item" data-item-id="${item.item_id}">
                <i class="bi bi-trash"></i> Remover
            </button>
        `;
        this.itensParaRetiradaContainer.appendChild(itemElement);
    });

    this.itensParaRetiradaContainer.querySelectorAll('.btn-remove-item').forEach(btn => {
        btn.addEventListener('click', (e) => this._removeItemParaRetirada(parseInt(e.currentTarget.id)));
    });
}

_removeItemParaRetirada(itemId) {
    this.itensSelecionados = this.itensSelecionados.filter(item => item.item_id !== itemId);
    this._renderItensParaRetirada();
}

async _enviarSolicitacao() {
    const setorId = parseInt(this.selectSetor.value);
    const justificativa = this.inputJustificativa.value.trim();
    const solicitadoLocalmentePor = this.inputSolicitadoLocalmentePor.value.trim();

    if (isNaN(setorId)) {
        showAlert('Por favor, selecione um setor.', 'warning');
        return;
    }

    if (this.itensSelecionados.length === 0) {
        showAlert('Por favor, adicione pelo menos um item para solicitar a retirada.', 'warning');
        return;
    }

    uiService.showLoading();

    try {
        const retiradaData = {
            setor_id: setorId,
            justificativa: justificativa || null,
            solicitado_localmente_por: solicitadoLocalmentePor || null,
            itens: this.itensSelecionados.map(item => ({
                item_id: item.item_id,
                quantidade_retirada: item.quantidade_retirada
            }))
        };

        await apiService.solicitarRetirada(retiradaData);
        showAlert('Solicitação de retirada enviada com sucesso!', 'success');
        this.modalSolicitarRetirada.hide();
        this._resetForm();
    } catch (error) {
        console.error("Erro ao enviar solicitação de retirada:", error);
        showAlert(error.message || 'Erro ao enviar a solicitação de retirada.', 'danger');
    } finally {
        uiService.hideLoading();
    }
}

```

```

    }

    _resetForm() {
        this.form.reset();
        this.itensSelecionados = [];
        this._renderItensParaRetirada();
        this.noItemsMessage.style.display = 'block';
        this.inputQuantidadeAddItem.value = 1;
        this.currentItemToAddToCart = null;
        this._clearSelectedItemDisplay(); //
    }

    _clearSelectedItemDisplay() {
        this.selectedItemName.textContent = ''; //
        // Linha removida: this.selectedItemStock.textContent = '';
        this.selectedItemDisplay.style.display = 'none'; //
        this.btnAdicionarItemRetirada.disabled = true; //
    }
}

export const solicitarRetiradaModule = new SolicitarRetiradaModule();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\static\js\uiService.js

```

// frontend/static/js/uiService.js

import { formatDate, formatDateTime, getStatusText, showAlert } from './utils.js';
import estadoGlobal from './estadoGlobal.js';

class UiService {
    constructor() {
        this.mainContent = document.getElementById('main-content');
        this.loadingSpinner = document.getElementById('loading-spinner');
    }

    renderPage(title, contentHtml) {
        if (this.mainContent) {
            this.mainContent.innerHTML = `

### >${title}</h3>${contentHtml}`; } else { console.error("Elemento 'main-content' não encontrado."); } } renderTable(headers, rows, options = {}) { const { tableId, noRecordsMessage = 'Nenhum registro encontrado.', rowMapper, actionsHtml = () => '' } = options; const tableContent = rows.length > 0 ? rows.map(item => ` <tr> ${rowMapper(item).map(cell => `<td>${cell}</td>`).join('')} <td>${actionsHtml(item)}</td> </tr> `).join('') : `<tr><td colspan="${headers.length + 1}" class="text-center">${noRecordsMessage}</td></tr>`; return ` <div class="table-responsive"> <table class="table table-bordered table-striped ${tableId ? `id="${tableId}"` : ''}"> <thead class="table-secondary"> <tr> ${headers.map(header => `<th>${header}</th>`).join('')} </tr> </thead> </table> </div> `; } }


```

```

        </thead>
        <tbody>
            ${tableContent}
        </tbody>
    </table>
</div>
`;
}

renderPagination(currentPage, totalPages, type, pageSizeSelectId, currentPageSize) {
    let pageLinks = '';
    let startPage = Math.max(1, currentPage - 2);
    let endPage = Math.min(totalPages, currentPage + 2);

    if (totalPages <= 5) {
        startPage = 1;
        endPage = totalPages;
    } else {
        if (currentPage <= 3) {
            startPage = 1;
            endPage = 5;
        } else if (currentPage + 2 > totalPages) {
            startPage = totalPages - 4;
            endPage = totalPages;
        }
    }

    if (startPage > 1) {
        pageLinks += `<li class="page-item"><a class="page-link" href="#" data-page="${type}-1">1</a>`;
        if (startPage > 2) {
            pageLinks += `<li class="page-item disabled"><span class="page-link">...</span></li>`;
        }
    }

    for (let i = startPage; i <= endPage; i++) {
        pageLinks += `
            <li class="page-item ${i === currentPage ? 'active' : ''}">
                <a class="page-link" href="#" data-page="${type}-${i}">${i}</a>
            </li>`;
    }

    if (endPage < totalPages) {
        if (endPage < totalPages - 1) {
            pageLinks += `<li class="page-item disabled"><span class="page-link">...</span></li>`;
        }
        pageLinks += `<li class="page-item"><a class="page-link" href="#" data-page="${type}-${totalPages}">${totalPages}</a>`;
    }

    const pageSizeSelectOptions = estadoGlobal.PAGE_SIZE_OPTIONS.map(size =>
        `<option value="${size}" ${size === currentPageSize ? 'selected' : ''}>${size}</option>`
    ).join('');

    return `
        <nav aria-label="Page navigation" id="${type}-pagination-nav">
            <ul class="pagination justify-content-center">
                <li class="page-item ${currentPage === 1 ? 'disabled' : ''}">
                    <a class="page-link" href="#" data-action="${type}-prev">Anterior</a>
                </li>
                ${pageLinks}
                <li class="page-item ${currentPage === totalPages || totalPages === 0 ? 'disabled' : ''}">
                    <a class="page-link" href="#" data-action="${type}-next">Próximo</a>
                </li>
            </ul>
        </nav>
    `;
}

```

```

        </nav>
        <div class="d-flex justify-content-center my-2">
            <label class="me-2 align-self-center">Itens por página: </label>
            <select class="form-select w-auto" id="{pageSizeSelectId}" data-action="{type}-pagesiz
                {pageSizeSelectOptions}
            </select>
        </div>
    `;
}

renderPageSizeSelect(id, currentPageSize) {
    const options = [5, 10, 25, 50, 100].map(size =>
        `<option value="{size}" ${currentPageSize === size ? 'selected' : ''}>${size}</option>`
    ).join('');

    return `
        <div class="d-flex justify-content-center my-2">
            <label class="me-2 align-self-center">Itens por página: </label>
            <select class="form-select w-auto" id="{id}">
                {options}
            </select>
        </div>
    `;
}

renderItemList(containerId, items) {
    const cont = document.getElementById(containerId);
    if (!cont) {
        console.error(`Container com ID '{containerId}' não encontrado.`);
        return;
    }

    cont.innerHTML = ''; // Limpa o conteúdo existente
    items.forEach(i => {
        const btn = document.createElement('button');
        btn.type = 'button';
        btn.className = 'list-group-item list-group-item-action';
        btn.textContent = `${i.item.nome_item_original} Quantidade: ${i.quantidade_retirada}`; // Aj
        btn.onclick = () => this.openItemDetail(i.item, i.quantidade_retirada); // Passa a quantidad
        cont.appendChild(btn);
    });
}

// openItemDetail é chamado a partir de renderItemList, que já passa a quantidadeRetirada
openItemDetail(item, qtdRetirada) {
    document.getElementById('itemNome').textContent = item.nome_item_original; // Ajustado para nome
    document.getElementById('itemEstoque').textContent = item.quantidade_item;

    // Condicionalmente exibe/oculta o campo "Solicitado"
    const itemQtdRetiradaElement = document.getElementById('itemQtdRetirada');
    const itemQtdRetiradaLi = itemQtdRetiradaElement ? itemQtdRetiradaElement.closest('li') : null;

    if (qtdRetirada !== undefined && qtdRetirada !== null) {
        itemQtdRetiradaElement.textContent = qtdRetirada;
        if (itemQtdRetiradaLi) itemQtdRetiradaLi.style.display = ''; // Mostra o li
    } else {
        if (itemQtdRetiradaLi) itemQtdRetiradaLi.style.display = 'none'; // Oculta o li
    }

    document.getElementById('itemEstoqueMin').textContent = item.quantidade_minima_item || 0;
    document.getElementById('itemValidade').textContent = item.data_validade_item ? formatDate(item.
    this.getModalInstance('modalDetalheItem').show();
}

```

```

fillModalDetalhes(retirada) {
    document.getElementById('detalheRetiradaId').value = retirada.retirada_id;
    document.getElementById('detalheStatus').value = getStatusText(retirada.status);
    document.getElementById('detalheSetor').value = retirada.setor_nome || '-';
    document.getElementById('detalheUsuario').value = retirada.usuario_nome || '-';
    document.getElementById('detalheSolicitadoPor').value = retirada.solicitado_localmente_por || '-';
    document.getElementById('detalheAutorizadoPor').value = retirada.autorizado_por_nome || '-';
    document.getElementById('detalheData').value = formatDateTime(retirada.data_solicitacao);
    document.getElementById('detalheJustificativa').value = retirada.justificativa || '-';
    document.getElementById('detalheStatusDesc').value = retirada.detalhe_status || '-';

    this.renderItemList('detalheItens', retirada.itens);
}

// fillModalDetalhesItem é chamado a partir de alertasModule, que não passa quantidadeRetirada
fillModalDetalhesItem(item) {
    if (!item) {
        console.error("Item inválido para preencher detalhes do modal.");
        return;
    }

    document.getElementById('itemNome').textContent = item.nome_item_original || 'N/A';
    document.getElementById('itemEstoque').textContent = item.quantidade_item || 0;
    document.getElementById('itemEstoqueMin').textContent = item.quantidade_minima_item || 0;
    document.getElementById('itemValidade').textContent = item.data_validade_item ? formatDate(item.data_validade_item) : '';

    // Oculta o campo "Solicitado" explicitamente quando chamado de alertas
    const itemQtdRetiradaElement = document.getElementById('itemQtdRetirada');
    const itemQtdRetiradaLi = itemQtdRetiradaElement ? itemQtdRetiradaElement.closest('li') : null;
    if (itemQtdRetiradaLi) itemQtdRetiradaLi.style.display = 'none';
}

fillModalAutorizar(retirada) {
    document.getElementById('autorizarRetiradaId').value = retirada.retirada_id;
    document.getElementById('autorizarSetor').value = retirada.setor_nome || '-';
    document.getElementById('autorizarUsuario').value = retirada.usuario_nome;
    document.getElementById('autorizarJustificativa').value = retirada.justificativa || '-';
    document.getElementById('autorizarData').value = formatDateTime(retirada.data_solicitacao);
    document.getElementById('autorizarDetalheStatus').value = '';

    this.renderItemList('autorizarItens', retirada.itens);

    const btnConfirmarAutorizar = document.getElementById('btn-confirmar-autorizar-retirada');
    if (btnConfirmarAutorizar) {
        btnConfirmarAutorizar.dataset.id = retirada.retirada_id;
    } else {
        console.error("Elemento 'btn-confirmar-autorizar-retirada' não encontrado em fillModalAutorizar.");
    }

    const btnConfirmarNegar = document.getElementById('btn-confirmar-negar-retirada');
    if (btnConfirmarNegar) {
        btnConfirmarNegar.dataset.id = retirada.retirada_id;
    } else {
        console.error("Elemento 'btn-confirmar-negar-retirada' não encontrado em fillModalAutorizar.");
    }
}

// NOVO: Função para preencher o modal de concluir retirada
fillModalConcluir(retirada) {
    document.getElementById('concluirRetiradaId').value = retirada.retirada_id;
    document.getElementById('concluirRetiradaDisplayId').textContent = retirada.retirada_id;
    document.getElementById('concluirDetalheStatus').value = retirada.detalhe_status || ''; // Preenche
}

```

```

getModalInstance(id) {
  const modalElement = document.getElementById(id);
  if (!modalElement) {
    console.error(`Elemento do modal com ID '${id}' não encontrado para getModalInstance. Verifi
    return null;
  }
  return bootstrap.Modal.getInstance(modalElement) || new bootstrap.Modal(modalElement);
}

showLoading() {
  if (this.loadingSpinner) {
    this.loadingSpinner.style.display = 'block';
  }
}

hideLoading() {
  if (this.loadingSpinner) {
    this.loadingSpinner.style.display = 'none';
  }
}
}

export const uiService = new UiService();

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxa rifado\_ets\frontend\static\js\utils.js

```

// frontend/static/js/utils.js

import estadoGlobal from './estadoGlobal.js';

export function showAlert(message, type = 'success', duration = 5000) {
  const toastContainer = document.getElementById('toast-container');
  if (!toastContainer) {
    console.error("Elemento 'toast-container' não encontrado para exibir alerta.");
    return;
  }

  const toastElement = document.createElement('div');
  toastElement.className = `toast align-items-center text-white bg-${type} border-0 fade show`;
  toastElement.setAttribute('role', 'alert');
  toastElement.setAttribute('aria-live', 'assertive');
  toastElement.setAttribute('aria-atomic', 'true');
  toastElement.innerHTML = `
    <div class="d-flex">
      <div class="toast-body">
        ${message}
      </div>
      <button type="button" class="btn-close btn-close-white me-2 m-auto" data-bs-dismiss="toast"
    </div>`;

  toastContainer.appendChild(toastElement);

  const toast = new bootstrap.Toast(toastElement, {
    autohide: true,
    delay: duration
  });
  toast.show();

  toastElement.addEventListener('hidden.bs.toast', () => {
    toastElement.remove();
  });
}

```

```

    });
}

export function formatDate(dateString) {
    if (!dateString) return "N/A";
    const date = new Date(dateString);
    return date.toLocaleDateString('pt-BR');
}

export function formatDateTime(dateString) {
    if (!dateString) return "N/A";
    const date = new Date(dateString);
    return date.toLocaleString('pt-BR');
}

export function getStatusText(statusCode) {
    return estadoGlobal.statusMap[statusCode] || 'Desconhecido';
}

export function getStatusValue(statusName) {
    return estadoGlobal.statusMapUpdate[statusName];
}

const NEW_ALERTS_FLAG_KEY = 'hasNewAlerts';
const NEW_WITHDRAWAL_REQUESTS_FLAG_KEY = 'hasNewWithdrawalRequests';

export function setNewAlertsFlag(hasNewAlerts) {
    if (hasNewAlerts) {
        localStorage.setItem(NEW_ALERTS_FLAG_KEY, 'true');
    } else {
        localStorage.removeItem(NEW_ALERTS_FLAG_KEY);
    }
    updateNotificationBellUI(); // Atualiza a UI imediatamente
}

export function getNewAlertsFlag() {
    return localStorage.getItem(NEW_ALERTS_FLAG_KEY) === 'true';
}

export function setNewWithdrawalRequestsFlag(hasNewRequests) {
    if (hasNewRequests) {
        localStorage.setItem(NEW_WITHDRAWAL_REQUESTS_FLAG_KEY, 'true');
    } else {
        localStorage.removeItem(NEW_WITHDRAWAL_REQUESTS_FLAG_KEY);
    }
    updateNotificationBellUI(); // Atualiza a UI imediatamente
}

export function getNewWithdrawalRequestsFlag() {
    return localStorage.getItem(NEW_WITHDRAWAL_REQUESTS_FLAG_KEY) === 'true';
}

export function updateNotificationBellUI() {
    const notificationDot = document.getElementById('notification-dot');
    const newAlerts = getNewAlertsFlag();
    const newWithdrawalRequests = getNewWithdrawalRequestsFlag();

    if (notificationDot) {
        if (newAlerts || newWithdrawalRequests) {
            notificationDot.style.display = 'block';
            notificationDot.classList.add('animate_animated', 'animate_pulse');
        } else {
            notificationDot.style.display = 'none';
        }
    }
}

```



```

        notificationDot.classList.remove('animate_animated', 'animate_pulse');
    }
} else {
    console.warn('updateNotificationBellUI: Elemento #notification-dot não encontrado.');
```

```

    }
}

const newAlertsMenuItem = document.getElementById('new-alerts-menu-item');
const newWithdrawalRequestsMenuItem = document.getElementById('new-withdrawal-requests-menu-item');
const noNotificationsMenuItem = document.getElementById('no-notifications-menu-item'); // NOVO: Refe
```

```

if (newAlertsMenuItem) {
    newAlertsMenuItem.style.display = newAlerts ? 'block' : 'none';
}
if (newWithdrawalRequestsMenuItem) {
    newWithdrawalRequestsMenuItem.style.display = newWithdrawalRequests ? 'block' : 'none';
}
}
```

```

// NOVO: Lógica para exibir/ocultar "Sem notificações"
if (noNotificationsMenuItem) {
    if (!newAlerts && !newWithdrawalRequests) {
        noNotificationsMenuItem.style.display = 'block'; // Exibe "Sem notificações"
    } else {
        noNotificationsMenuItem.style.display = 'none'; // Oculta "Sem notificações"
    }
}
}
}
```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxa rifado\_ets\frontend\static\js\validar-acesso.js

```

//frontend\static\js\validar-acesso.js

document.addEventListener('DOMContentLoaded', function() {

    document.getElementById('form-login').addEventListener('submit', async function (event) {
        event.preventDefault();

        const form = event.target;
        const formData = new FormData(form);
        const data = new URLSearchParams(formData);

        try {
            const response = await fetch('/api/almoxarifado/usuarios/token', {
                method: 'POST',
                headers: {
                    'Accept': 'application/json'
                },
                body: data,
                credentials: 'include' // permite que o cookie seja salvo
            });

            if (response.ok) {
                // Depois de salvar o cookie com sucesso, redireciona manualmente
                window.location.href = "/";
            } else {
                const result = await response.json();
                alert(result.detail || "Erro ao fazer login");
            }
        } catch (err) {
            console.error("Erro no login:", err);
            alert("Erro de rede ao tentar fazer login.");
        }
    });
});
```

```

    }
  });
});

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\dashboardAlmoxarifado.html

```

<!-- frontend\templates\dashboardAlmoxarifado.html -->
{% include 'partials/head.html' %}
{% include 'partials/topbar.html' %}
{% include 'partials/navbar.html' %}
{% include 'partials/quick_access.html' %}
{% include 'partials/global_overlays.html' %}

<!-- Inclui todos os modais -->
{% include 'partials/modals/modal_cadastrar_item.html' %}
{% include 'partials/modals/modal_editar_item.html' %}
{% include 'partials/modals/modal_cadastrar_categoria.html' %}
{% include 'partials/modals/modal_editar_categoria.html' %}
{% include 'partials/modals/modal_ver_detalhes_retirada.html' %}
{% include 'partials/modals/modal_autorizar_retirada.html' %}
{% include 'partials/modals/modal_item_detalhes.html' %}
{% include 'partials/modals/modal_solicitar_retirada.html' %}
{% include 'partials/modals/modal_selecionar_item_retirada.html' %}
{% include 'partials/modals/modal_concluir_retirada.html' %}
{% include 'partials/modals/modal_reports_dashboard.html' %}
{% include 'partials/modals/modal_report_quantidade_itens.html' %}
{% include 'partials/modals/modal_report_entrada_itens.html' %}
{% include 'partials/modals/modal_report_retiradas_setor.html' %}
{% include 'partials/modals/modal_report_retiradas_usuario.html' %}

<!-- Scripts e inicialização -->
{% include 'partials/scripts.html' %}

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\dashboardDirecao.html

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>SiCA - Sistema de Controle de Almoxarifado</title>

  <!-- Bootstrap CSS offline -->
  <link href="/static/css/bootstrap.min.css" rel="stylesheet">
  <link href="/static/css/custom.css" rel="stylesheet">
</head>
<body class="bg-light">
  <script src="/static/js/validar-acesso.js"></script>

  <!-- Barra superior -->
  <div class="d-flex justify-content-between align-items-center px-3 py-2 custom-header">
    
    <!-- <h5 class="m-0 text-white fw-bold text-center flex-grow-1">SISTEMA DE CONTROLE DE ALMOXARIFADO
  <div class="custom-header text-center py-0 fw-bold">
    SISTEMA DE CONTROLE DE ALMOXARIFADO - CPT

```

```

    </div>
    
</div>

<!-- Barra de navegação -->
<nav class="navbar navbar-expand-lg" style="background-color: #00448d;">
  <div class="container-fluid justify-content-between">
    <div class="navbar-collapse w-100">
      <ul class="navbar-nav d-flex justify-content-between w-100 py-0">

        <!-- Itens -->
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle text-white" href="#" id="itensDropdown" role="button" data-
            Itens
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Cadastrar item</a></li>
            <li><a class="dropdown-item" href="#">Listar item</a></li>
            <li><a class="dropdown-item" href="#">Editar item</a></li>
            <li><a class="dropdown-item" href="#">Excluir item</a></li>
          </ul>
        </li>

        <!-- Categorias -->
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle text-white" href="#" id="categoriasDropdown" role="button" data-
            Categorias
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Cadastrar categoria</a></li>
            <li><a class="dropdown-item" href="#">Listar categorias</a></li>
            <li><a class="dropdown-item" href="#">Editar categorias</a></li>
            <li><a class="dropdown-item" href="#">Excluir categorias</a></li>
          </ul>
        </li>

        <!-- Retiradas -->
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle text-white" href="#" id="retiradasDropdown" role="button" data-
            Retiradas
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Registrar retirada</a></li>
            <li><a class="dropdown-item" href="#">Visualizar solicitações pendentes</a></li>
            <li><a class="dropdown-item" href="#">Visualizar retiradas</a></li>
          </ul>
        </li>

        <!-- Relatórios -->
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle text-white" href="#" id="relatoriosDropdown" role="button" data-
            Relatórios
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Por categoria</a></li>
            <li><a class="dropdown-item" href="#">Por produto</a></li>
            <li><a class="dropdown-item" href="#">Por setor</a></li>
            <li><a class="dropdown-item" href="#">Por servidor</a></li>
            <li><a class="dropdown-item" href="#">Entradas</a></li>
          </ul>
        </li>

        <!-- Alertas -->

```

```

        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle text-white" href="#" id="alertasDropdown" role="button" d
            Alertas
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Baixo estoque</a></li>
            <li><a class="dropdown-item" href="#">Validade próxima</a></li>
          </ul>
        </li>

      </ul>
    </div>
  </div>
</nav>

<!-- Conteúdo da página -->
<div class="container mt-5">
  <h2>Bem-vindo ao SiCA</h2>
  <p>Escolha uma opção no menu acima para começar.</p>
</div>

<!-- Bootstrap JS offline -->
<script src="/static/js/bootstrap.bundle.min.js"></script>

</body>
</html>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxa rifado\_ets\frontend\templates\dashboardServidor.html

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>SiCA - Sistema de Controle de Almoxarifado</title>

  <!-- Bootstrap CSS offline -->
  <link href="/static/css/bootstrap.min.css" rel="stylesheet">
  <link href="/static/css/custom.css" rel="stylesheet">
</head>
<body class="bg-light">
  <script src="/static/js/validar-acesso.js"></script>

  <!-- Barra superior -->
  <div class="d-flex justify-content-between align-items-center px-3 py-2 custom-header">
    
    <!-- <h5 class="m-0 text-white fw-bold text-center flex-grow-1">SISTEMA DE CONTROLE DE ALMOXARIFADO
  <div class="custom-header text-center py-0 fw-bold">
    SISTEMA DE CONTROLE DE ALMOXARIFADO - CPT
  </div>
  
</div>

  <!-- Barra de navegação -->
  <nav class="navbar navbar-expand-lg" style="background-color: #00448d;">
    <div class="container-fluid">
      <div class="navbar-collapse">
        <ul class="navbar-nav mx-auto">
          <!-- Itens -->

```

```

<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle text-white" href="#" id="itensDropdown" role="button" data-
    Itens
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Cadastrar item</a></li>
    <li><a class="dropdown-item" href="#">Listar item</a></li>
    <li><a class="dropdown-item" href="#">Editar item</a></li>
    <li><a class="dropdown-item" href="#">Excluir item</a></li>
  </ul>
</li>

<!-- Categorias -->
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle text-white" href="#" id="categoriasDropdown" role="button" data-
    Categorias
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Cadastrar categoria</a></li>
    <li><a class="dropdown-item" href="#">Listar categorias</a></li>
    <li><a class="dropdown-item" href="#">Editar categorias</a></li>
    <li><a class="dropdown-item" href="#">Excluir categorias</a></li>
  </ul>
</li>

<!-- Retiradas -->
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle text-white" href="#" id="retiradasDropdown" role="button" data-
    Retiradas
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Registrar retirada</a></li>
    <li><a class="dropdown-item" href="#">Visualizar solicitações pendentes</a></li>
    <li><a class="dropdown-item" href="#">Visualizar retiradas</a></li>
  </ul>
</li>

<!-- Relatórios -->
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle text-white" href="#" id="relatoriosDropdown" role="button" data-
    Relatórios
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Por categoria</a></li>
    <li><a class="dropdown-item" href="#">Por produto</a></li>
    <li><a class="dropdown-item" href="#">Por setor</a></li>
    <li><a class="dropdown-item" href="#">Por servidor</a></li>
    <li><a class="dropdown-item" href="#">Entradas</a></li>
  </ul>
</li>

<!-- Alertas -->
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle text-white" href="#" id="alertasDropdown" role="button" data-
    Alertas
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Baixo estoque</a></li>
    <li><a class="dropdown-item" href="#">Validade próxima</a></li>
  </ul>
</li>
</ul>

```

```

        </div>
    </div>
</nav>

<!-- Conteúdo da página -->
<div class="container mt-5">
    <h2>Bem-vindo ao SiCA</h2>
    <p>Escolha uma opção no menu acima para começar.</p>
</div>

<!-- Bootstrap JS offline -->
<script src="/static/js/bootstrap.bundle.min.js"></script>

</body>
</html>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\index.html

```

<!-- frontend\templates\index.html -->

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Login - Sistema Almoxarifado</title>
    <link rel="stylesheet" href="/static/css/style.css" />
</head>
<body>
    <header>
        <h2>Sistema de Controle de Almoxarifado - CPT</h2>
    </header>
    <main>
        <div class="login-card">
            <div class="avatar-placeholder"></div>
            <form id="login-form">
                <label for="username">Usuário:</label>
                <input type="text" id="username" name="username" required />

                <label for="password">Senha:</label>
                <input type="password" id="password" name="password" required />

                <a href="#" id="recover-link">Cadastrar/Recuperação de conta</a>

                <button type="submit" id="login-btn">Entrar</button>
            </form>
        </div>
    </main>

    <script src="/static/js/login.js"></script>
    <script src="/static/js/validar-acesso.js"></script> </body>

</body>
</html>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\global\_overlays.html

```
<!-- frontend\templates\partials\global_overlays.html -->

<div id="loading-spinner" style="
  display: none;
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 1060; /* Acima de modais do Bootstrap (z-index padrão do modal é 1050) */
  background-color: rgba(255, 255, 255, 0.7); /* Fundo semi-transparente */
  border-radius: 10px;
  padding: 20px;
  backdrop-filter: blur(2px); /* Adiciona um leve desfoque ao fundo */
  -webkit-backdrop-filter: blur(2px); /* Para compatibilidade com navegadores WebKit */
">
  <div class="spinner-border text-primary" role="status" style="width: 3rem; height: 3rem;">
    <span class="visually-hidden">Carregando...</span>
  </div>
  <p class="mt-2 text-primary">Carregando...</p>
</div>

<div id="toast-container" class="toast-container position-fixed top-0 end-0 p-3" style="z-index: 1070;">
  </div>
```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\head.html

```
<!-- frontend\templates\partials\head.html -->
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Sistema de Controle de Almoxarifado</title>

  <link href="/static/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css" rel="stylesheet">
  <link rel="stylesheet" href="/static/css/dashboards.css" />
  <link rel="stylesheet" href="/static/css/custom.css"> </head>
<body>
  <script src="/static/js/validar-acesso.js"></script>
```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\navbar.html

```
<nav class="navbar navbar-expand-lg navbar-dark navbar-custom">
  <div class="container-fluid">
    <button class="btn btn-outline-light me-3" id="home-button"><i class="bi bi-house-fill"></i></button>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarMenu"
      aria-controls="navbarMenu" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarMenu">
```

```

<ul class="navbar-nav me-auto mb-2 mb-lg-0">
    {% include 'partials/nav_items.html' %}
</ul>

<div class="d-flex align-items-center">
    <div class="dropdown">
        <a class="nav-link position-relative me-3" href="#" id="alert-notification-bell" role="button">
            <i class="bi bi-bell-fill fs-5"></i>
            <span id="notification-dot" class="notification-dot position-absolute">
                <span class="visually-hidden">Novas notificações</span>
            </span>
        </a>
        <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="alert-notification-bell">
            <li><span class="dropdown-item text-muted" id="no-notifications-menu-item" style="display: block; padding: 5px 10px 5px 15px;">Nenhuma notificação</span>
            <li><a class="dropdown-item" href="#" id="new-withdrawal-requests-menu-item" style="display: block; padding: 5px 10px 5px 15px;">Novas solicitações de saque</a>
            <li><a class="dropdown-item" href="#" id="new-alerts-menu-item" style="display: block; padding: 5px 10px 5px 15px;">Novas alertas</a>
            <li><hr class="dropdown-divider"></li>
        </ul>
    </div>

    <div class="dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="userMenu" data-bs-toggle="dropdown" aria-expanded="false">
            <i class="bi bi-person-circle"></i>
        </a>
        <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="userMenu">
            <li><a class="dropdown-item" href="#" id="logout-link">Sair</a></li>
        </ul>
    </div>
</div>
</div>
</div>
</nav>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\nav\_items.html

```

<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="itensDropdown" data-bs-toggle="dropdown">ITENS</a>
    <ul class="dropdown-menu" aria-labelledby="itensDropdown">
        <li><a class="dropdown-item" id="btn-open-cadastrar-item" href="#">Cadastrar Item</a></li>
        <li><a class="dropdown-item" id="listar-item-link" href="#">Listar Itens</a></li>
    </ul>
</li>

<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="categoriasDropdown" data-bs-toggle="dropdown">CATEGORIAS</a>
    <ul class="dropdown-menu" aria-labelledby="categoriasDropdown">
        <li><a class="dropdown-item" id="btn-open-cadastrar-categoria" href="#">Cadastrar Categoria</a></li>
        <li><a class="dropdown-item" id="listar-categoria-link-quick" href="#">Listar Categoria</a></li>
    </ul>
</li>

<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="retiradasDropdown" data-bs-toggle="dropdown">RETIRADAS</a>
    <ul class="dropdown-menu" aria-labelledby="retiradasDropdown">
        <li><a class="dropdown-item" id="btn-open-solicitar-retirada" href="#">Solicitar Retirada</a></li>
        <li><a class="dropdown-item" id="listar-retiradas-link" href="#">Histórico de Retiradas</a></li>
        <li><a class="dropdown-item" id="listar-retiradas-pendentes-link" href="#">Solicitações Pendentes</a></li>
    </ul>
</li>

```



```

</li>

<li class="nav-item">
  <a class="nav-link" href="#" id="open-reports-dashboard"
    data-bs-toggle="modal" data-bs-target="#modalReportsDashboard">
    RELATÓRIOS
  </a>
</li>

<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="alertasDropdown" data-bs-toggle="dropdown">ALERTAS</a>
  <ul class="dropdown-menu" aria-labelledby="alertasDropdown">
    <li><a class="dropdown-item" href="#" id="open-alertas-modal">Listar Alertas</a></li>
  </ul>
</li>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\quick\_access.html

```

<!-- frontend\templates\partials\quick_access.html -->
<section class="container py-4" id="main-content">
  <h3 class="mb-3">Acesso Rápido</h3>
  <div class="row g-3">
    {% include 'partials/quick_cards.html' %}
  </div>
</section>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\quick\_cards.html

```

<!-- frontend\templates\partials\quick_cards.html -->
<div class="col-6 col-md-3">
  <a href="#" id="btn-open-cadastrar-item" class="text-decoration-none">
    <div class="quick-access-card">
      <i class="bi bi-plus-square"></i>
      <p>Cadastrar Item</p>
    </div>
  </a>
</div>
<div class="col-6 col-md-3">
  <a href="#" id="listar-item-link-quick" class="text-decoration-none">
    <div class="quick-access-card">
      <i class="bi bi-card-list"></i>
      <p>Listar Itens</p>
    </div>
  </a>
</div>
<div class="col-6 col-md-3">
  <a href="#" id="listar-retiradas-pendentes-quick" class="text-decoration-none">
    <div class="quick-access-card">
      <i class="bi bi-card-checklist"></i>
      <p>Retiradas Pendentes</p>
    </div>
  </a>
</div>
<div class="col-6 col-md-3">
  <a href="#" id="open-reports-dashboard" data-bs-toggle="modal" data-bs-target="#modalReportsDashboard">
    <div class="quick-access-card">

```

```

        <i class="bi bi-bar-chart-line"></i>
        <p>Relatórios</p>
    </div>
</a>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\scripts.html

```

<script src="/static/js/bootstrap.bundle.min.js"></script>
<script src="/static/js/logout.js"></script>
<script src="/static/js/listar-itens.js"></script>
<script src="/static/js/listar-categorias.js"></script>
<script src="/static/js/cadastrar-item.js"></script>
<script src="/static/js/cadastrar-categorias.js"></script>

<script type="module" src="/static/js/estadoGlobal.js"></script>
<script type="module" src="/static/js/Utils.js"></script>
<script type="module" src="/static/js/apiService.js"></script>
<script type="module" src="/static/js/dataService.js"></script>
<script type="module" src="/static/js/uiService.js"></script>
<script type="module" src="/static/js/retiradasModule.js"></script>
<script type="module" src="/static/js/solicitar-retirada.js"></script>
<script type="module" src="/static/js/selecionar-item-module.js"></script>
<script type="module" src="/static/js/reportsModule.js"></script>
<script type="module" src="/static/js/main.js"></script>

</body>
</html>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\topbar.html

```

<!-- frontend\templates\partials\topbar.html -->
<header class="top-bar d-flex justify-content-between align-items-center">
    <div class="logo">Sistema de Controle de Almoxarifado</div>
    <div class="d-flex align-items-center">
        <span class="me-2">Centro Profissional e Tecnológico</span>
        
    </div>
</header>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_autorizar\_retirada.html

```

<div class="modal fade" id="modalAutorizarRetirada" tabindex="-1" aria-labelledby="modalAutorizarRetiradaLabel">
    <div class="modal-dialog modal-lg modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalAutorizarRetiradaLabel">Autorizar Retirada</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>
            <div class="modal-body">
                <form class="row g-3">

```

```



```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_cadastrar\_categoria.html

```

<div class="modal fade" id="modalCadastrarCategoria" tabindex="-1" aria-labelledby="modalCadastrarCategoriaLabel">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content rounded-4 shadow">
      <div class="modal-header border-0">
        <h5 class="modal-title" id="modalCadastrarCategoriaLabel">Cadastrar Categoria</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
      </div>
      <div class="modal-body">
        <form id="form-cadastrar-categoria">
          <div class="mb-3">
            <label for="nome_categoria" class="form-label">Nome*</label>
            <input type="text" class="form-control" id="nome_categoria" name="nome_categoria" required>
          </div>
          <div class="mb-3">
            <label for="descricao_categoria" class="form-label">Descrição</label>
            <textarea class="form-control" id="descricao_categoria" name="descricao_categoria" rows="2">
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="modal-footer border-0 justify-content-center">
            <button class="btn btn-secondary me-2" id="btn-cancelar-categoria">Cancelar</button>
            <button class="btn btn-primary" id="btn-salvar-categoria">Salvar</button>
        </div>
    </div>
</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_cadastrar\_item.html

```

<div class="modal fade" id="modalCadastrarItem" tabindex="-1" aria-labelledby="modalCadastrarItemLabel"
    <div class="modal-dialog modal-lg modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalCadastrarItemLabel">Cadastrar Novo Item</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>
            <div class="modal-body">
                <form id="form-cadastrar-item">
                    <div class="row g-3">
                        <div class="col-md-6">
                            <label for="nome_item" class="form-label">Nome do Item*</label>
                            <input type="text" class="form-control" id="nome_item" name="nome_item" required>
                        </div>
                        <div class="col-md-6">
                            <label for="unidade_medida_item" class="form-label">Unidade de Medida*</label>
                            <input type="text" class="form-control" id="unidade_medida_item" name="unidade_medida_item" required>
                        </div>
                        <div class="col-md-6">
                            <label for="descricao_item" class="form-label">Descrição*</label>
                            <textarea class="form-control" id="descricao_item" name="descricao_item" rows="2" required>
                        </div>
                        <div class="col-md-3">
                            <label for="quantidade_item" class="form-label">Quantidade*</label>
                            <input type="number" class="form-control" id="quantidade_item" name="quantidade_item" min="1">
                        </div>
                        <div class="col-md-3">
                            <label for="quantidade_minima_item" class="form-label">Quantidade Mínima</label>
                            <input type="number" class="form-control" id="quantidade_minima_item" name="quantidade_minima_item">
                        </div>
                        <div class="col-md-4">
                            <label for="data_validade_item" class="form-label">Data de Validade</label>
                            <input type="date" class="form-control" id="data_validade_item" name="data_validade_item">
                        </div>
                        <div class="col-md-4">
                            <label for="data_entrada_item" class="form-label">Data de Entrada</label>
                            <input type="datetime-local" class="form-control" id="data_entrada_item" name="data_entrada_item">
                        </div>
                        <div class="col-md-4">
                            <label for="marca_item" class="form-label">Marca</label>
                            <input type="text" class="form-control" id="marca_item" name="marca_item">
                        </div>
                        <div class="col-md-6">
                            <label for="categoria_id" class="form-label">Categoria*</label>
                            <select class="form-select" id="categoria_id" name="categoria_id" required>
                                <option value="" disabled selected></option>
                                <option value="1">Categoria 1</option>
                                <option value="2">Categoria 2</option>
                                <option value="3">Categoria 3</option>
                                <option value="4">Categoria 4</option>
                                <option value="5">Categoria 5</option>
                                <option value="6">Categoria 6</option>
                                <option value="7">Categoria 7</option>
                                <option value="8">Categoria 8</option>
                                <option value="9">Categoria 9</option>
                                <option value="10">Categoria 10</option>
                                <option value="11">Categoria 11</option>
                                <option value="12">Categoria 12</option>
                                <option value="13">Categoria 13</option>
                                <option value="14">Categoria 14</option>
                                <option value="15">Categoria 15</option>
                                <option value="16">Categoria 16</option>
                                <option value="17">Categoria 17</option>
                                <option value="18">Categoria 18</option>
                                <option value="19">Categoria 19</option>
                                <option value="20">Categoria 20</option>
                                <option value="21">Categoria 21</option>
                                <option value="22">Categoria 22</option>
                                <option value="23">Categoria 23</option>
                                <option value="24">Categoria 24</option>
                                <option value="25">Categoria 25</option>
                                <option value="26">Categoria 26</option>
                                <option value="27">Categoria 27</option>
                                <option value="28">Categoria 28</option>
                                <option value="29">Categoria 29</option>
                                <option value="30">Categoria 30</option>
                                <option value="31">Categoria 31</option>
                                <option value="32">Categoria 32</option>
                                <option value="33">Categoria 33</option>
                                <option value="34">Categoria 34</option>
                                <option value="35">Categoria 35</option>
                                <option value="36">Categoria 36</option>
                                <option value="37">Categoria 37</option>
                                <option value="38">Categoria 38</option>
                                <option value="39">Categoria 39</option>
                                <option value="40">Categoria 40</option>
                                <option value="41">Categoria 41</option>
                                <option value="42">Categoria 42</option>
                                <option value="43">Categoria 43</option>
                                <option value="44">Categoria 44</option>
                                <option value="45">Categoria 45</option>
                                <option value="46">Categoria 46</option>
                                <option value="47">Categoria 47</option>
                                <option value="48">Categoria 48</option>
                                <option value="49">Categoria 49</option>
                                <option value="50">Categoria 50</option>
                                <option value="51">Categoria 51</option>
                                <option value="52">Categoria 52</option>
                                <option value="53">Categoria 53</option>
                                <option value="54">Categoria 54</option>
                                <option value="55">Categoria 55</option>
                                <option value="56">Categoria 56</option>
                                <option value="57">Categoria 57</option>
                                <option value="58">Categoria 58</option>
                                <option value="59">Categoria 59</option>
                                <option value="60">Categoria 60</option>
                                <option value="61">Categoria 61</option>
                                <option value="62">Categoria 62</option>
                                <option value="63">Categoria 63</option>
                                <option value="64">Categoria 64</option>
                                <option value="65">Categoria 65</option>
                                <option value="66">Categoria 66</option>
                                <option value="67">Categoria 67</option>
                                <option value="68">Categoria 68</option>
                                <option value="69">Categoria 69</option>
                                <option value="70">Categoria 70</option>
                                <option value="71">Categoria 71</option>
                                <option value="72">Categoria 72</option>
                                <option value="73">Categoria 73</option>
                                <option value="74">Categoria 74</option>
                                <option value="75">Categoria 75</option>
                                <option value="76">Categoria 76</option>
                                <option value="77">Categoria 77</option>
                                <option value="78">Categoria 78</option>
                                <option value="79">Categoria 79</option>
                                <option value="80">Categoria 80</option>
                                <option value="81">Categoria 81</option>
                                <option value="82">Categoria 82</option>
                                <option value="83">Categoria 83</option>
                                <option value="84">Categoria 84</option>
                                <option value="85">Categoria 85</option>
                                <option value="86">Categoria 86</option>
                                <option value="87">Categoria 87</option>
                                <option value="88">Categoria 88</option>
                                <option value="89">Categoria 89</option>
                                <option value="90">Categoria 90</option>
                                <option value="91">Categoria 91</option>
                                <option value="92">Categoria 92</option>
                                <option value="93">Categoria 93</option>
                                <option value="94">Categoria 94</option>
                                <option value="95">Categoria 95</option>
                                <option value="96">Categoria 96</option>
                                <option value="97">Categoria 97</option>
                                <option value="98">Categoria 98</option>
                                <option value="99">Categoria 99</option>
                                <option value="100">Categoria 100</option>
                            </select>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

        </div>
    </form>
</div>
<div class="modal-footer border-0 justify-content-center">
    <button type="button" class="btn btn-secondary me-2" id="btn-cancelar-item">Cancelar</button>
    <button type="button" class="btn btn-primary" id="btn-salvar-item">Enviar</button>
</div>
</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_concluir\_retirada.html

```

<div class="modal fade" id="modalConcluirRetirada" tabindex="-1" aria-labelledby="modalConcluirRetiradaLabel">
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalConcluirRetiradaLabel">Concluir Retirada</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>
            <div class="modal-body">
                <form id="form-concluir-retirada">
                    <input type="hidden" id="concluirRetiradaId">
                    <p>Você tem certeza que deseja concluir a retirada ID: <strong id="concluirRetiradaId"></strong></p>
                    <p>Esta ação decrementa o estoque dos itens!</p>
                    <div class="mb-3">
                        <label for="concluirDetalheStatus" class="form-label">Detalhe do Status (Opcional)</label>
                        <textarea class="form-control" id="concluirDetalheStatus" rows="3" placeholder="Detalhe do Status (Opcional)"></textarea>
                    </div>
                </form>
            </div>
            <div class="modal-footer border-0 justify-content-center">
                <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
                <button type="button" class="btn btn-success" id="btn-confirmar-concluir-retirada">Confirmar</button>
            </div>
        </div>
    </div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_editar\_categoria.html

```

<div class="modal fade" id="modalEditarCategoria" tabindex="-1" aria-labelledby="modalEditarCategoriaLabel">
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalEditarCategoriaLabel">Editar Categoria</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body">
                <form id="form-editar-categoria">
                    <div class="mb-3">
                        <label for="edit-nome_categoria" class="form-label">Nome*</label>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

        <input type="text" class="form-control" id="edit-nome_categoria" name="nome_categoria" required="" />
    </div>
    <div class="mb-3">
        <label for="edit-descricao_categoria" class="form-label">Descrição</label>
        <textarea class="form-control" id="edit-descricao_categoria" name="descricao_categoria" rows="3" />
    </div>
</form>
</div>
<div class="modal-footer border-0 justify-content-center">
    <button class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
    <button class="btn btn-primary" id="btn-salvar-editar-categoria">Salvar</button>
</div>
</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_editar\_item.html

```

<!-- frontend/templates/partials/modals/modal_editar_item.html -->

<div class="modal fade" id="modalEditarItem" tabindex="-1" aria-labelledby="modalEditarItemLabel" aria-hidden="true">
    <div class="modal-dialog modal-lg modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">

            <!-- Modal Header -->
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalEditarItemLabel">Editar Item</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>

            <!-- Modal Body -->
            <div class="modal-body">
                <form id="form-editar-item">
                    <div class="row g-3">
                        <!-- Campos de edição -->
                        <div class="col-md-6">
                            <label for="nome_item" class="form-label">Nome do Item*</label>
                            <input type="text" class="form-control" id="nome_item" name="nome_item" required="" />
                        </div>
                        <div class="col-md-6">
                            <label for="unidade_medida_item" class="form-label">Unidade de Medida*</label>
                            <input type="text" class="form-control" id="unidade_medida_item" name="unidade_medida_item" required="" />
                        </div>
                        <div class="col-md-6">
                            <label for="descricao_item" class="form-label">Descrição*</label>
                            <textarea class="form-control" id="descricao_item" name="descricao_item" rows="2" required="" />
                        </div>
                        <div class="col-md-3">
                            <label for="quantidade_item" class="form-label">Quantidade*</label>
                            <input type="number" class="form-control" id="quantidade_item" name="quantidade_item" min="1" />
                        </div>
                        <div class="col-md-3">
                            <label for="quantidade_minima_item" class="form-label">Quantidade Mínima</label>
                            <input type="number" class="form-control" id="quantidade_minima_item" name="quantidade_minima_item" min="1" />
                        </div>
                        <div class="col-md-4">
                            <label for="data_validade_item" class="form-label">Data de Validade</label>
                            <input type="date" class="form-control" id="data_validade_item" name="data_validade_item" />
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
    <div class="col-md-4">
        <label for="data_entrada_item" class="form-label">Data de Entrada</label>
        <input type="datetime-local" class="form-control" id="data_entrada_item" name="data_entrada_item">
    </div>
    <div class="col-md-4">
        <label for="marca_item" class="form-label">Marca</label>
        <input type="text" class="form-control" id="marca_item" name="marca_item">
    </div>
    <div class="col-md-6">
        <label for="edit-categoria_id" class="form-label">Categoria*</label>
        <select class="form-select" id="edit-categoria_id" name="categoria_id" required>
            <option value="" disabled selected></option>
        </select>
    </div>
</div>
</form>
</div>

<!-- Modal Footer -->
<div class="modal-footer border-0 justify-content-center">
    <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
    <button type="button" class="btn btn-primary" id="btn-salvar-editar-item">Salvar</button>
</div>

</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_item\_detalhes.html

```

<!-- frontend\templates\partials\modals\modal_item_detalhes.html -->
<div class="modal fade" id="modalDetalheItem" tabindex="-1" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title">Detalhes do Item</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body">
                <ul class="list-unstyled mb-0">
                    <li><strong>Nome:</strong> <span id="itemNome"></span></li>
                    <li><strong>Em Estoque:</strong> <span id="itemEstoque"></span></li>
                    <li><strong>Solicitado:</strong> <span id="itemQtdRetirada"></span></li>
                    <li><strong>Estoque Mínimo:</strong> <span id="itemEstoqueMin"></span></li>
                    <li><strong>Validade:</strong> <span id="itemValidade"></span></li>
                </ul>
            </div>
            <div class="modal-footer border-0">
                <button class="btn btn-secondary" data-bs-dismiss="modal">Fechar</button>
            </div>
        </div>
    </div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_reports\_dashboard.html

```
<div class="modal fade" id="modalReportsDashboard" tabindex="-1" aria-labelledby="modalReportsDashboardLabel">
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content rounded-4 shadow">
      <div class="modal-header border-0">
        <h5 class="modal-title" id="modalReportsDashboardLabel">Gerar Relatórios</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
      </div>
      <div class="modal-body">
        <div class="list-group list-group-flush"> {# Adicionado list-group-flush para remover border-top
          <button type="button" class="list-group-item list-group-item-action py-3 px-4 d-flex justify-content-between">
            Relatório de Quantidade de Itens
            <i class="bi bi-box-seam text-primary fs-4"></i> {# Ícone de sua escolha #}
          </button>
          <button type="button" class="list-group-item list-group-item-action py-3 px-4 d-flex justify-content-between">
            Relatório de Entrada de Itens
            <i class="bi bi-arrow-down-square text-success fs-4"></i> {# Ícone de sua escolha #}
          </button>
          <button type="button" class="list-group-item list-group-item-action py-3 px-4 d-flex justify-content-between">
            Relatório de Retiradas por Setor
            <i class="bi bi-building text-info fs-4"></i> {# Ícone de sua escolha #}
          </button>
          <button type="button" class="list-group-item list-group-item-action py-3 px-4 d-flex justify-content-between">
            Relatório de Retiradas por Usuário
            <i class="bi bi-person-fill text-warning fs-4"></i> {# Ícone de sua escolha #}
          </button>
        </div>
      </div>
      <div class="modal-footer border-0 justify-content-center">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Fechar</button>
      </div>
    </div>
  </div>
</div>
```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_report\_entrada\_itens.html

```
<div class="modal fade" id="modalReportEntradaItens" tabindex="-1" aria-labelledby="modalReportEntradaItensLabel">
  <div class="modal-dialog modal-md modal-dialog-centered">
    <div class="modal-content rounded-4 shadow">
      <div class="modal-header border-0">
        <h5 class="modal-title" id="modalReportEntradaItensLabel">Relatório de Entrada de Itens</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
      </div>
      <div class="modal-body">
        <form id="form-report-entrada-itens">
          <div class="mb-3">
            <label for="dataInicioEntrada" class="form-label">Data Inicial*</label>
            <input type="date" class="form-control" id="dataInicioEntrada" name="data_inicio">
          </div>
          <div class="mb-3">
            <label for="dataFimEntrada" class="form-label">Data Final*</label>
            <input type="date" class="form-control" id="dataFimEntrada" name="data_fim" required>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
```



```

        </div>
        <div class="mb-3">
            <label for="formatoEntradaItens" class="form-label">Formato do Relatório*</label>
            <select class="form-select" id="formatoEntradaItens" name="formato" required>
                <option value="csv" selected>CSV</option>
                <option value="xlsx">XLSX</option>
            </select>
        </div>
    </form>
</div>
<div class="modal-footer border-0 justify-content-center">
    <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
    <button type="button" class="btn btn-primary" id="btnGerarReportEntradaItens">Gerar Relatório</button>
</div>
</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_report\_quantidade\_itens.html

```

<div class="modal fade" id="modalReportQuantidadeItens" tabindex="-1" aria-labelledby="modalReportQuantidadeItensLabel">
    <div class="modal-dialog modal-md modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalReportQuantidadeItensLabel">Relatório de Quantidade de Itens</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>
            <div class="modal-body">
                <form id="form-report-quantidade-itens">
                    <div class="mb-3">
                        <label for="filterCategoryReport" class="form-label">Filtrar por Categoria (opcional)</label>
                        <select class="form-select" id="filterCategoryReport" name="filtro_categoria">
                            <option value="">Todas as Categorias</option>
                        </select>
                    </div>
                    <div class="mb-3">
                        <label for="selectedProductNameReport" class="form-label">Filtrar por Produto (opcional)</label>
                        <div class="input-group">
                            <input type="text" class="form-control" id="selectedProductNameReport" name="filtro_produto_nome">
                            <button class="btn btn-outline-secondary" type="button" id="btnOpenSelectProduct">
                                <i class="bi bi-search"></i> Selecionar
                            </button>
                            <button class="btn btn-outline-danger" type="button" id="btnClearSelectedProduct">
                                <i class="bi bi-x-lg"></i>
                            </button>
                        </div>
                        <input type="hidden" id="selectedProductIdReport" name="filtro_produto_id">
                    </div>
                    <div class="mb-3">
                        <label for="formatoQuantidadeItens" class="form-label">Formato do Relatório*</label>
                        <select class="form-select" id="formatoQuantidadeItens" name="formato" required>
                            <option value="csv" selected>CSV</option>
                            <option value="xlsx">XLSX</option>
                        </select>
                    </div>
                </form>
            </div>
            <div class="modal-footer border-0 justify-content-center">

```

```

        <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
        <button type="button" class="btn btn-primary" id="btnGerarReportQuantidadeItens">Gerar R
    </div>
</div>
</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_report\_retiradas\_setor.html

```

<div class="modal fade" id="modalReportRetiradasSetor" tabindex="-1" aria-labelledby="modalReportRetiradasSetorLabel">
    <div class="modal-dialog modal-md modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalReportRetiradasSetorLabel">Relatório de Retiradas por Setor</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>
            <div class="modal-body">
                <form id="form-report-retiradas-setor">
                    <div class="mb-3">
                        <label for="selectSetor" class="form-label">Setor*</label>
                        <select class="form-select" id="selectSetor" name="setor_id" required>
                            <option value="" disabled selected>Carregando setores...</option>
                        </select>
                    </div>
                    <div class="mb-3">
                        <label for="dataInicioRetiradaSetor" class="form-label">Data Inicial*</label>
                        <input type="date" class="form-control" id="dataInicioRetiradaSetor" name="data_inicio">
                    </div>
                    <div class="mb-3">
                        <label for="dataFimRetiradaSetor" class="form-label">Data Final*</label>
                        <input type="date" class="form-control" id="dataFimRetiradaSetor" name="data_fim">
                    </div>
                    <div class="mb-3">
                        <label for="formatoRetiradasSetor" class="form-label">Formato do Relatório*</label>
                        <select class="form-select" id="formatoRetiradasSetor" name="formato" required>
                            <option value="csv" selected>CSV</option>
                            <option value="xlsx">XLSX</option>
                        </select>
                    </div>
                </form>
            </div>
            <div class="modal-footer border-0 justify-content-center">
                <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
                <button type="button" class="btn btn-primary" id="btnGerarReportRetiradasSetor">Gerar Relatório</button>
            </div>
        </div>
    </div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_report\_retiradas\_usuario.html

```

<div class="modal fade" id="modalReportRetiradasUsuario" tabindex="-1" aria-labelledby="modalReportRetiradasUsuarioLabel">

```

```

<div class="modal-dialog modal-md modal-dialog-centered">
  <div class="modal-content rounded-4 shadow">
    <div class="modal-header border-0">
      <h5 class="modal-title" id="modalReportRetiradasUsuarioLabel">Relatório de Retiradas por
      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
    </div>
    <div class="modal-body">
      <form id="form-report-retiradas-usuario">
        <div class="mb-3">
          <label for="selectUsuario" class="form-label">Usuário*</label>
          <select class="form-select" id="selectUsuario" name="usuario_id" required>
            <option value="" disabled selected>Carregando usuários...</option>
          </select>
        </div>
        <div class="mb-3">
          <label for="dataInicioRetiradaUsuario" class="form-label">Data Inicial*</label>
          <input type="date" class="form-control" id="dataInicioRetiradaUsuario" name="data_inicial">
        </div>
        <div class="mb-3">
          <label for="dataFimRetiradaUsuario" class="form-label">Data Final*</label>
          <input type="date" class="form-control" id="dataFimRetiradaUsuario" name="data_fim">
        </div>
        <div class="mb-3">
          <label for="formatoRetiradasUsuario" class="form-label">Formato do Relatório*</label>
          <select class="form-select" id="formatoRetiradasUsuario" name="formato" required>
            <option value="csv" selected>CSV</option>
            <option value="xlsx">XLSX</option>
          </select>
        </div>
      </form>
    </div>
    <div class="modal-footer border-0 justify-content-center">
      <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</button>
      <button type="button" class="btn btn-primary" id="btnGerarReportRetiradasUsuario">Gerar</button>
    </div>
  </div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_selecionar\_item\_retirada.html

```

<!-- frontend\templates\partials\modals\modal_selecionar_item_retirada.html -->
<div class="modal fade" id="modalSelecionarItemRetirada" tabindex="-1" aria-labelledby="modalSelecionarItemRetiradaLabel">
  <div class="modal-dialog modal-md modal-dialog-centered">
    <div class="modal-content rounded-4 shadow">
      <div class="modal-header border-0">
        <h5 class="modal-title" id="modalSelecionarItemRetiradaLabel">Selecionar Item para Retirada
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
      </div>
      <div class="modal-body">
        <form id="form-search-item-retirada" class="mb-3">
          <div class="row g-3">
            <div class="col-md-7">
              <label for="searchItemName" class="form-label visually-hidden">Buscar por Nome
              <input type="text" class="form-control" id="searchItemName" placeholder="Buscar por Nome">
            </div>
            <div class="col-md-5 d-flex">
              <button type="submit" class="btn btn-primary flex-grow-1 me-2" id="btn-search-item-retirada">Buscar</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

```

        <button type="button" class="btn btn-secondary" id="btn-clear-item-search">L
    </div>
</form>

<div id="itens-list-container" class="overflow-auto" style="max-height: 300px;">
</div>

<div id="itens-pagination-container" class="mt-3">
</div>
</div>
<div class="modal-footer border-0 justify-content-end">
    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Fechar</button>
</div>
</div>
</div>
</div>

```

## Arquivo: C:\Users\Victor\Desktop\projeto\_almoxarifado\sist\_almoxarifado\_ets\frontend\templates\partials\modals\modal\_solicitar\_retirada.html

```

<div class="modal fade" id="modalSolicitarRetirada" tabindex="-1" aria-labelledby="modalSolicitarRetiradaLabel">
    <div class="modal-dialog modal-lg modal-dialog-centered">
        <div class="modal-content rounded-4 shadow">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="modalSolicitarRetiradaLabel">Solicitar Retirada de Itens</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
            </div>
            <div class="modal-body">
                <form id="form-solicitar-retirada">
                    <div class="row g-3">
                        <div class="col-md-6">
                            <label for="solicitar_setor_id" class="form-label">Setor*</label>
                            <select class="form-select" id="solicitar_setor_id" name="setor_id" required>
                                <option value="" disabled selected>Selecione um setor</option>
                            </select>
                        </div>
                        <div class="col-md-6">
                            <label for="solicitado_localmente_por" class="form-label">Solicitado Localmente por*</label>
                            <input type="text" class="form-control" id="solicitado_localmente_por" name="solicitado_localmente_por">
                            <small class="form-text text-muted">Preencha se a solicitação for feita por alguém</small>
                        </div>
                        <div class="col-12">
                            <label for="solicitar_justificativa" class="form-label">Justificativa da Retirada</label>
                            <textarea class="form-control" id="solicitar_justificativa" name="justificativa"></textarea>
                        </div>
                    </div>
                    <hr class="my-4">
                    <h5>Itens para Retirada</h5>
                    <div id="itens-para-retirada-container" class="row g-3 mb-3">
                        <div class="col-12 text-muted" id="no-items-message">Nenhum item adicionado</div>
                    </div>
                    <div class="row g-3 align-items-end">
                        <div class="col-12" id="selected-item-display" style="display: none;">
                            <p class="mb-1">Item Selecionado: <strong id="selected-item-name"></strong></p>
                        </div>
                        <div class="col-md-6">
                            <label for="quantidade_add_item" class="form-label">Quantidade</label>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```
<input type="number" class="form-control" id="quantidade_add_item" min=
</div>
<div class="col-md-6 d-flex align-items-end">
    <button type="button" class="btn btn-info w-100 me-2" id="btn-abrir-sele
    <button type="button" class="btn btn-success w-100" id="btn-adicionar-it
</div>
</div>

</div>
</form>
</div>
<div class="modal-footer border-0 justify-content-center">
    <button type="button" class="btn btn-secondary me-2" data-bs-dismiss="modal">Cancelar</b
    <button type="button" class="btn btn-primary" id="btn-salvar-solicitacao-retirada">Solic
</div>
</div>
</div>
</div>
```

```
<div class="modal fade" id="modalVerDetalhesRetirada" tabindex="-1" aria-labelledby="modalVerDetalhesRet
<div class="modal-dialog modal-lg modal-dialog-centered">
  <div class="modal-content rounded-4 shadow">
    <div class="modal-header border-0">
      <h5 class="modal-title" id="modalVerDetalhesRetiradaLabel">Detalhes da Retirada</h5>
      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Fechar"></button>
    </div>
    <div class="modal-body">
      <form class="row g-3">
        <div class="col-md-6">
          <label for="detalheRetiradaId" class="form-label">ID da Retirada</label>
          <input type="text" class="form-control" id="detalheRetiradaId" readonly>
        </div>
        <div class="col-md-6">
          <label for="detalheStatus" class="form-label">Status</label>
          <input type="text" class="form-control" id="detalheStatus" readonly>
        </div>
        <div class="col-md-6">
          <label for="detalheSetor" class="form-label">Setor Solicitante</label>
          <input type="text" class="form-control" id="detalheSetor" readonly>
        </div>
        <div class="col-md-6">
          <label for="detalheUsuario" class="form-label">Usuário Solicitante</label>
          <input type="text" class="form-control" id="detalheUsuario" readonly>
        </div>
        <div class="col-12">
          <label for="detalheJustificativa" class="form-label">Justificativa da Solicitação</label>
          <textarea class="form-control" id="detalheJustificativa" rows="2" readonly></textarea>
        </div>
        <div class="col-md-6">
          <label for="detalheSolicitadoPor" class="form-label">Solicitado Localmente Por</label>
          <input type="text" class="form-control" id="detalheSolicitadoPor" readonly>
        </div>
        <div class="col-md-6">
          <label for="detalheAutorizadoPor" class="form-label">Autorizado/Negado Por</label>
          <input type="text" class="form-control" id="detalheAutorizadoPor" readonly>
        </div>
      </form>
    </div>
  </div>
</div>
```

```

<div class="col-12">
  <label for="detalheData" class="form-label">Data da Solicitação</label>
  <input type="text" class="form-control" id="detalheData" readonly>
</div>
<div class="col-12">
  <label for="detalheStatusDesc" class="form-label">Detalhe do Status</label>
  <input type="text" class="form-control" id="detalheStatusDesc" readonly>
</div>
<div class="col-12">
  <label class="form-label">
    Itens Solicitados
    <small class="text-muted">(clique para ver detalhes)</small>
    <i class="bi bi-info-circle" data-bs-toggle="tooltip"
      title="Cada item é clicável para abrir detalhes"></i>
  </label>
  <div id="detalheItens" class="list-group"></div>
</div>
</form>
</div>
<div class="modal-footer border-0 justify-content-center">
  <button class="btn btn-secondary" data-bs-dismiss="modal">Fechar</button>
</div>
</div>
</div>
</div>

```