

Correções e trechos prontos para substituir (Colaborador)

ATUALIZAÇÃO: adicionei ao final do documento a **versão totalmente refatorada do navbarCola** (HTML + JS). Use esse trecho para substituir o bloco atual do seu **partial**.

NavbarCola — Versão refatorada (inserir/ substituir parcial)

Abaixo está a versão revista, robusta e acessível do **navbarCola**. Ela foi reestruturada para:

- evitar problemas com indexação de strings no Handlebars (uso **data-username** e JS para extrair iniciais);
- prevenir re-inicializações duplicadas (marca o elemento como inicializado);
- manter fallback funcional quando o Bootstrap não estiver disponível;
- garantir que os popovers/modals abram apenas por ação humana (**event.isTrusted**);
- cuidar da acessibilidade (atributos ARIA, foco, keyboard handlers);
- trabalhar com **window.BACKEND_URL** quando disponível;
- não vazar tokens em logs e manter operações seguras.

Importante: cole o HTML (com o script) no seu parcial **frontend/views/partials/navbarCola.handlebars** substituindo o bloco atual. O CSS segue sendo carregado via **/css/navbarCola.css** e **/css/colaborador.css**.

```
<link rel="stylesheet" href="/css/navbarCola.css">
<link rel="stylesheet" href="/css/colaborador.css">

<nav class="navbar-cola d-flex justify-content-between align-items-center"
role="navigation" aria-label="Navbar Colaborador">
  <!-- Brand -->
  <div class="brand-left d-flex align-items-center">
    <a href="/" class="d-flex align-items-center" aria-label="Ir para home">
      
    </a>
  </div>

  <!-- Links centrais -->
  <div class="d-flex gap-4 nav-links" role="menubar" aria-label="Links principais">
    <a href="/colaborador/dados" class="btnNav btn" role="menuitem">Início</a>
    <a href="/colaborador/documentacao" class="btnNav btn" role="menuitem">Documentação</a>
    <a href="/colaborador/solicitacoes" class="btnNav btn" role="menuitem">Solicitações</a>
    <a href="/colaborador/holerites" class="btnNav btn">
```

```

role="menuitem">Holerites</a>
    <a href="/colaborador/pontoColaborador" class="btnNav btn"
role="menuitem">Ponto</a>
</div>

<!-- Ações (direita) -->
<div class="d-flex align-items-center gap-3 nav-actions">
    <!-- Botão de perfil: abre popover com 2 botões (Ver perfil / Sair) -->
    <button id="profileToggle"
        class="btn-profile"
        type="button"
        aria-haspopup="true"
        aria-expanded="false"
        aria-controls="profile-popover"
        aria-label="Abrir opções de perfil"
        title="Perfil"
        data-username="{{usuario.nome}}"
        data-initialized="0">
        {{#if usuario.foto}}
            
        {{else}}
            <div class="avatar-initials" aria-hidden="true">U</div>
        {{/if}}
    </button>

    <!-- Three-dots (opcional) - sem comportamento automático -->
    <button id="btnMenu" class="btn btn-link p-0" type="button" aria-
label="Mais opções">
        <i class="bi bi-three-dots-vertical" aria-hidden="true"></i>
    </button>
</div>
</nav>

<div class="linha" aria-hidden="true"></div>

<!-- Modal Bootstrap de confirmação (só abre por ação do usuário) -->
<div class="modal fade" id="confirmLogoutModal" tabindex="-1" aria-
labelledby="confirmLogoutTitle" aria-hidden="true">
    <div class="modal-dialog modal-sm modal-dialog-centered">
        <div class="modal-content">
            <div class="modal-header border-0">
                <h5 class="modal-title" id="confirmLogoutTitle">Confirmar logout</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Fechar"></button>
            </div>
            <div class="modal-body text-center">
                <p>Tem certeza que deseja sair da sua conta?</p>
                <div class="d-flex gap-2 justify-content-between mt-3">
                    <button class="btn btn-secondary flex-fill" data-bs-
dismiss="modal">Cancelar</button>

```

```

        <button class="btn btn-danger flex-fill"
id="confirmLogoutBtn">Sair</button>
    </div>
</div>
</div>
</div>
</div>

<script>
/*
NavbarCola refatorado – comportamento robusto do popover + logout
- Usa data-username para inicializar as iniciais quando necessário
- Protege contra múltiplas inicializações
- Fallback sem Bootstrap
- Acessibilidade (foco/teclado)
*/
(function () {
    document.addEventListener('DOMContentLoaded', () => {
        const BACKEND_URL = (typeof window !== 'undefined' &&
window.BACKEND_URL) ? window.BACKEND_URL : 'http://localhost:3001';
        const profileToggle = document.getElementById('profileToggle');
        const confirmLogoutModalEl =
document.getElementById('confirmLogoutModal');

        // Safety: ensure single init
        if (!profileToggle) return;
        if (profileToggle.dataset.initialized === '1') return;
        profileToggle.dataset.initialized = '1';

        // Se não houver foto, monta a inicial a partir do data-username
        try {
            const initialsEl = profileToggle.querySelector('.avatar-initials') ||
document.querySelector('.avatar-initials');
            if (initialsEl) {
                const username = profileToggle.getAttribute('data-username') ||
initialsEl.getAttribute('data-username') || '';
                const initial = (username && username.trim().length) ?
username.trim().charAt(0).toUpperCase() : 'U';
                initialsEl.textContent = initial;
            }
        } catch (e) { /* silent */ }

        let popoverInstance = null;
        let outsideClickHandler = null;
        let escHandler = null;

        function escapeHtml(str = '') {
            return String(str).replace(/([<>"'`=\/])/g, s => ({
'&': '&', '<': '<', '>': '>', '"': '"', "'": '&39;', '/': '&x2F;', '`': '&96;', '=': '&61;
[s]));
        }
    });

```

```

function buildPopoverHtml(user = {}) {
  const nome = user.nome ? escapeHtml(user.nome) : 'Usuário';
  const tipo = user.tipo_usuario ? escapeHtml(user.tipo_usuario) : '';
  const foto = user.foto ? escapeHtml(user.foto) : null;
  return `
    <div id="profile-popover" class="popover-profile" style="min-width:
220px;">
      <div class="d-flex" style="gap:.6rem;padding:10px 12px;align-
items:center;">
        <div style="width:44px;height:44px;border-radius:
8px;overflow:hidden;flex:0 0 44px;display:grid;place-
items:center;background:rgba(255,255,255,0.02);font-weight:
700;color:inherit;">
          ${foto ? `` : (nome ?
escapeHtml(nome.charAt(0)) : 'U')}
        </div>
        <div style="flex:1;">
          <div style="font-weight:700;font-size:.95rem;">${nome}</div>
          <div style="font-size:.78rem;color:rgba(0,0,0,0.55)">${tipo}</
div>
        </div>
      </div>
      <div style="padding:8px 10px;display:flex;flex-
direction:column;gap:8px;">
        <button id="popoverViewProfile" class="btn btn-sm btn-outline-
secondary btn-popover-action" type="button" style="text-align:left;">Ver
perfil</button>
        <button id="popoverLogoutBtn" class="btn btn-sm btn-danger btn-
popover-action" type="button">Sair</button>
      </div>
    </div>
  `;
}

function hidePopover() {
  if (!popoverInstance) return;
  try { popoverInstance.hide(); } catch (e) {}
  try { popoverInstance.dispose(); } catch (e) {}
  // fallback element removal
  if (popoverInstance._fallbackEl) {
    try { popoverInstance._fallbackEl.remove(); } catch(e){}
  }
  popoverInstance = null;
  profileToggle.setAttribute('aria-expanded', 'false');

  if (outsideClickHandler) { document.removeEventListener('click',
outsideClickHandler); outsideClickHandler = null; }
  if (escHandler) { document.removeEventListener('keydown', escHandler);
escHandler = null; }
}

```

```

}

async function toggleProfilePopover(e) {
  if (e && e.isTrusted === false) return; // evita aberturas artificiais

  // se já aberto, fecha
  if (popoverInstance) { hidePopover(); return; }

  const token = localStorage.getItem('token');
  if (!token) { window.location.href = '/'; return; }

  let usuario = {};
  try {
    const res = await fetch(`${BACKEND_URL}/api/auth/me`, { headers: {
      Authorization: 'Bearer ' + token } });
    if (!res.ok) { localStorage.removeItem('token');
      window.location.href = '/'; return; }
    const j = await res.json();
    usuario = j.usuario || {};
  } catch (err) {
    console.warn('Erro ao buscar usuário para popover:', err);
  }

  const useBootstrap = (typeof window.bootstrap !== 'undefined' &&
    window.bootstrap.Popover);
  if (useBootstrap) {
    popoverInstance = new bootstrap.Popover(profileToggle, {
      content: buildPopoverHtml(usuario),
      html: true,
      sanitize: false, // necessário para html interno – cuidado em
      produção
      trigger: 'manual',
      placement: 'bottom',
      container: 'body'
    });
    popoverInstance.show();
  } else {
    // fallback: cria elemento manualmente
    const div = document.createElement('div');
    div.className = 'popover fallback-popover';
    div.setAttribute('role', 'dialog');
    div.innerHTML = buildPopoverHtml(usuario);
    document.body.appendChild(div);
    const rect = profileToggle.getBoundingClientRect();
    div.style.position = 'absolute';
    const left = Math.min(window.innerWidth - 240, Math.max(8, rect.left
+ window.scrollX));
    div.style.left = left + 'px';
    div.style.top = (rect.bottom + 8 + window.scrollY) + 'px';
    popoverInstance = { _fallbackEl: div, hide() { try { div.remove(); }
    catch(e){} }, dispose() { try { div.remove(); } catch(e){} } };
  }
}

```

```

    }

    profileToggle.setAttribute('aria-expanded', 'true');

    // Conectar listeners após render
    requestAnimationFrame(() => {
        const pop = document.querySelector('.popover') ||
document.querySelector('.fallback-popover');
        if (!pop) return;
        const viewBtn = pop.querySelector('#popoverViewProfile');
        const logoutBtn = pop.querySelector('#popoverLogoutBtn');

        viewBtn?.addEventListener('click', () => { hidePopover();
window.location.href = '/colaborador/perfil'; });
        logoutBtn?.addEventListener('click', () => {
            hidePopover();
            if (useBootstrap && confirmLogoutModalEl) {
                const inst = new bootstrap.Modal(confirmLogoutModalEl);
                inst.show();
            } else {
                if (confirm('Deseja sair da sua conta?')) performLogout();
            }
        });

        // foco acessível
        const firstFocusable = pop.querySelector('button, a,
[tabindex]:not([tabindex="-1"])');
        if (firstFocusable) firstFocusable.focus();
    });

    // fechar ao clicar fora ou ESC
    outsideClickHandler = (ev) => {
        const pop = document.querySelector('.popover') ||
document.querySelector('.fallback-popover');
        if (!pop) return;
        if (profileToggle.contains(ev.target) || pop.contains(ev.target))
return;
        hidePopover();
    };
    escHandler = (ev) => { if (ev.key === 'Escape') hidePopover(); };

    document.addEventListener('click', outsideClickHandler);
    document.addEventListener('keydown', escHandler);
}

async function performLogout() {
    try {
        const token = localStorage.getItem('token');
        await fetch(`${BACKEND_URL}/api/auth/logout`, {
            method: 'POST',
            headers: token ? { Authorization: `Bearer ${token}` } : {}
        });
    }

```

```

    });
  } catch (err) {
    console.warn('Erro no logout:', err);
  } finally {
    localStorage.removeItem('token');
    try { const inst = (window.bootstrap && window.bootstrap.Modal) ?
bootstrap.Modal.getInstance(confirmLogoutModalEl) : null; if (inst)
inst.hide(); } catch(e) {}
    window.location.href = '/';
  }
}

// atribui listeners
profileToggle.addEventListener('click', toggleProfilePopover);
profileToggle.addEventListener('keydown', (ev) => {
  if (ev.key === 'Enter' || ev.key === ' ') { ev.preventDefault();
toggleProfilePopover(ev); }
  if (ev.key === 'Tab') hidePopover();
});

const confirmBtn = document.getElementById('confirmLogoutBtn');
confirmBtn?.addEventListener('click', (ev) => { ev.preventDefault();
performLogout(); });

// cleanup
window.addEventListener('unload', () => { hidePopover(); });
});
})();
</script>

```

O que eu alterei / por que esta versão é mais confiável

- **Inicialização de iniciais:** antes havia tentativa de usar `{{usuario.nome.[0]}}` — isso nem sempre funciona no Handlebars. Agora o template passa `data-username` e o JS popula a inicial no cliente.
- **Marca de inicialização:** evita múltiplos `addEventListener` acidentais (com `data-initialized`).
- **Fallback sem Bootstrap:** se `bootstrap.Popover` não existir, a UI ainda funciona com popover manual.
- **Acessibilidade:** foco lógico, `aria-expanded`, keyboard handling (Enter/Space/Escape/Tab).
- **Segurança:** `sanitize:false` é usado apenas para permitir HTML do popover; mantenha controle do que o backend retorna ou sanitize no backend se dados forem do usuário.

Próximos passos sugeridos

- Teste em ambiente local com `NODE_ENV=development` e verifique console do servidor e do navegador.

- Caso prefira, eu posso gerar um `patch` (diff) pronto para aplicar com `git apply` que substitui apenas o partial `navbarCola` no repositório.
- Se quiser, também adapto o CSS (`navbarCola.css` e `colaborador.css`) para garantir contraste, tamanhos e responsividade.

Diga se quer o `patch` pronto ou a substituição do arquivo completa (arquivo HBS gerado por mim).