

Auditoria Visual – Estúdio Corpo & Alma

1. Tipografia

A interface utiliza **duas famílias de fonte** integradas via Google Fonts: **Playfair Display** (serifada) e **Lato** (sem serifa) ¹. Esta dupla tipográfica confere um ar elegante e claro ao sistema. Para manter a consistência:

- **Fontes e papéis:** Use a fonte serifada Playfair Display apenas em títulos principais e destaques elegantes (ex: títulos de tela em `<h1>`), enquanto a fonte Lato (sans-serif) deve ser usada em textos de interface, labels e conteúdo geral ¹ ². Por exemplo, as telas atuais já seguem este padrão: os títulos “Sua Agenda de Janeiro” e “Novo Cliente” usam a fonte serifada, ao passo que labels e parágrafos (como “Olá, Janaína” ou textos de ajuda) usam a fonte sans-serif ³ ⁴.
- **Hierarquia de pesos:** Padronize os pesos das fontes para indicar hierarquia. O sistema atual utiliza muitos textos em **extrabold** (Lato 900) para pequenos títulos e labels em caixa-alta, garantindo legibilidade mesmo em tamanhos reduzidos ⁵ ⁶. Recomenda-se manter esse uso de **Lato 900** para labels importantes e cabeçalhos de seção (ex: labels de formulário em **caps lock**, como “Nome Completo” e “Procedimento” ⁷ ⁸). Já os títulos principais em Playfair podem permanecer em peso regular ou semibold (400–600) para um aspecto sofisticado – note que o `<h1>` “Novo Agendamento” aparece em Playfair regular, contrastando com um label pequeno em Lato extrabold acima dele ⁹ ¹⁰. Defina no guia de estilo quais elementos usam **Lato extrabold** (ex: pequenos títulos, etiquetas) e quais usam **Playfair** (títulos de tela, seções especiais) para que todos os desenvolvedores mantenham esse padrão.
- **Escala de tamanhos:** Crie uma escala fixa de tamanhos tipográficos. Atualmente, o design emprega desde fontes bem pequenas (10px a 12px para legendas e ícones) até títulos de ~24px. Por exemplo, labels em caixa-alta estão em torno de **11px** (classe `text-[11px]`) ou **12px** (tailwind `text-xs`) ¹¹, textos de corpo em **14px** (tailwind `text-sm`), e títulos principais em **20-24px** (ex: `text-2xl ~24px`) ⁴. Essa graduação deve ser documentada: por exemplo, **Títulos** (H1) ~24px/Playfair; **Subtítulos** (H2/H3) ~16-18px/Lato bold ou Playfair; **Texto base** 14px/Lato regular; **Notas/legendas** 11-12px/Lato extrabold ou medium. Com uma escala definida, todas as telas usarão tamanhos consistentes. Observação: evitar tamanhos fora da escala (o uso de `text-[11px]` customizado poderia ser substituído por `text-xs` se possível, ou documentado como exceção para melhorar precisão).
- **Estilos consistentes:** Garanta consistência nos estilos auxiliares. O uso frequente de **caixa-alta com tracking expandido** para pequenas legendas (ex: “ONLINE”, “CADASTRO”) traz uniformidade ⁴. Documente que textos de apoio e seções devem usar **uppercase** e espaçamento entre letras (**tracking**) padronizado (no código atual, varia entre `tracking-wide` e valores específicos como `tracking-[0.24em]` ⁴ – ideal unificar). Da mesma forma, manter **line-height** adequado (no Tailwind já se usa `leading-tight` em títulos para compactar e dar um visual moderno ⁴). Cada nível hierárquico (título de tela, seção, label, corpo) deve ter estilo próprio definido no guia para fácil aplicação.

Em suma, a tipografia já transmite **clareza e sofisticação**, misturando títulos elegantes com textos limpos. A padronização deve reforçar o uso correto de cada fonte em seu contexto, tamanhos coerentes e pesos adequados, assegurando que um mesmo tipo de texto (por exemplo, todos os rótulos de campos obrigatórios) tenha sempre a mesma aparência em todas as telas do sistema.

2. Paleta de Cores

A paleta de cores do Estúdio Corpo & Alma é consistente e bem definida nas telas atuais. O Tailwind configura cores customizadas que refletem a identidade visual do produto ¹² ¹³. Para padronização, recomendamos formalizar os seguintes tons e seus usos previstos:

- **Cor primária - “Verde Estúdio”:** Um verde acinzentado (#6A806C) que é a cor de marca principal ¹². Atualmente ele é usado em textos de destaque e ícones principais (ex: texto “Olá, Janaína” e ícones ativos) e em botões primários (fundo ou texto) ¹⁴ ¹⁵. Deve continuar sendo a cor de realce principal para elementos interativos e estados ativos. Inclui variantes:
- **Verde Escuro (hover/ativo):** Um verde mais escuro (#4E5F50) definido como `studio.dark` ¹², utilizado em hovers de botões verdes (ex: no botão “Agendar” ao passar o cursor, muda de `bg-studio-green` para `bg-studio-dark` ¹⁶). Padronize esse tom para *hover/active* de botões primários e talvez títulos em destaque se precisar de contraste extra.
- **Verde Claro (background):** Um verde muito claro quase cinza (#F3F6F4) definido como `studio.light` ¹². Atualmente serve de fundo de componentes e botões em estado normal (ex: fundo do botão de voltar/circular e do botão “Hoje” em estado normal é esse verde claro ¹⁷ ¹⁸). Use-o como **cor de fundo neutra** para chips selecionáveis, ícones de ação padrão, caixas em destaque suave etc., garantindo que haja consistência em todos os lugares onde um fundo neutro for necessário.
- **Cor de destaque/secundária - “Acento”:** Há uma cor acento marrom-claro/dourada (#D4A373) configurada como `studio.accent` ¹². Nas telas, ela aparece em pequenos detalhes (por exemplo, no ícone de folha indicando status “Online” em combinação com verde ¹⁴). Esta cor deve ser usada com parcimônia para **destaques secundários** que complementam o verde (por exemplo, ícones ilustrativos, indicadores especiais ou talvez detalhes do logo). Documente seu uso para que seja consistente – p.ex., todos os ícones decorativos ou gráficos podem usar o acento para manter a paleta equilibrada.
- **Neutros - “Papel” e tons de cinza:** O fundo principal do app é um off-white apelidado de “**papel**” (#FAF9F6) ¹⁹, garantindo uma base suave e aconchegante em vez de branco puro. Deve ser o **fundo padrão** de todas as telas e contêineres de scroll ²⁰. Combine-o com cinzas para textos e bordas:
 - **Texto principal:** um cinza bem escuro (#2C3333) configurado como `text.main` ¹⁹, usado nos textos padrão (parágrafos, títulos em Lato) sobre o fundo claro. Mantém boa legibilidade sem ser preto duro.
 - **Texto secundário:** um cinza médio (#868E96) configurado como `text.muted` ¹⁹, usado para textos de apoio, placeholders e informações menos importantes (ex: legendas “Última visita: 15 Jan” usam esse tom ²¹ ²²). Padronizar para todos os *placeholders*, descrições e textos desabilitados para criar uniformidade.
 - **Linhos e bordas:** um cinza claríssimo semitransparente (`line: rgba(44, 51, 51, 0.06)`) é usado para divisas sutis e bordas neutras ²³. Vemos esse tom, por exemplo, na borda de campos de input e divisórias de listas (há uso de `border-gray-200` e `border-line` similares nas telas) ²⁴ ²⁵. Consolidar em documentação qual cor usar para **borders gerais** – idealmente uma única referência (`line` ou um gray 200) para todas as divisórias, garantindo que todas as telas tenham a *mesma intensidade de linhas* delimitadoras.
- **Cores de feedback (semânticas):** O sistema apresenta cores dedicadas a status/feedback:
 - **Sucesso/OK:** Verde vivo (#16A34A) definido como `ok` ²⁶ ¹³, utilizado para indicar etapas concluídas e confirmações (ex: nos “dots” de etapas concluídas do fluxo de atendimento ²⁷, e possivelmente em mensagens de sucesso). Aplicar essa cor para realçar estados **positivos** (confirmações, concluído) de forma consistente – por exemplo, o dot de etapa concluída fica verde e o texto associado também adota este verde escuro ²⁷.

- **Aviso/Alerta:** Amarelo-alaranjado (#F59E0B) configurado como `warn`¹³, usado para indicar atenção (ex: flag “Plantão” bloqueado aparece com fundo amarelo claro e texto amarelo escuro²⁸). Deve padronizar qualquer situação de **aviso** ou estado pendente. Atualmente, clientes marcados com “Atenção” também usam um tom semelhante (nas listas, clientes alertas aparecem com iniciais em laranja claro e tag “ATENÇÃO” provavelmente nesse tom)²⁹. Padronize que *alertas/atenção* usem este laranja nas tags e ícones correspondentes.
- **Erro/Perigo:** Vermelho intenso (#DC2626) definido como `danger`³⁰³¹, empregado para erros ou ações destrutivas. Vemos seu uso nos horários atrasados (horário em vermelho e indicador de atraso) e em tags de bloqueio (ex: “Bloqueado” em vermelho)³²³³. Deve ser a cor única para **erros, bloqueios ou avisos críticos**, usada em textos, ícones de alerta e fundos de pill vermelhas.
- **Informativo/Destaque específico:** **Domicílio (roxo)** – uma cor roxa média (#A855F7) está mapeada como `dom`³⁴ e representa atendimentos em domicílio. É usada para destacar elementos relacionados a atendimentos em casa (ex: borda lateral e ícone de “carro” em agendamentos domiciliares estão em roxo³⁵³⁶, e na legenda do calendário a bolinha roxa indica “domicílio”³⁷). No design system, trate essa cor como parte da paleta **informativa** ou de categoria (não exatamente neutra, nem semântica padrão, mas um identificador de tipo de atendimento). Documente seu uso restrito: tudo que indicar “em domicílio” deve usar o roxo padrão e, quando necessário, variações claras/escuras dele (como `purple-50` e `purple-800` do Tailwind que já aparecem nos pills e fundos relacionados³⁸³⁹).
- **Equilíbrio e combinação:** Aplique sempre as cores de forma consistente. Por exemplo, as **pílulas de status** já seguem um padrão coerente: fundo claro da cor temática e texto no tom forte da mesma cor (vide pill “Confirmado” com fundo verde claro e texto verde escuro⁴⁰, pill “Domicílio” com fundo roxo claríssimo e texto roxo forte⁴¹, pill “Bloqueado” em vermelho claro etc.⁴²). Esse padrão deve ser seguido para qualquer novo novo selo/pill de status. Da mesma forma, **ícones em círculos** seguem o padrão de cor do contexto: ícone de prédio (atendimento no estúdio) tem círculo de fundo verde claro e ícone verde⁴³, ícone de carro (domicílio) tem fundo roxo claro e ícone roxo⁴⁴. Essas **simetrias visuais** facilitam o entendimento e precisam estar na documentação para que novos componentes usem as mesmas convenções.

Em resumo, a paleta atual mostra **tons suaves e sofisticados** – verde acinzentado como carro-chefe, off-white como base, e acentos sutis. A auditoria confirma que os tons são consistentes entre as telas; o próximo passo é consolidar isso num guia de cores, definindo nome, valor hex e exemplos de uso para cada cor (principal, secundária, neutras, feedbacks), evitando variações não planejadas. Isso garantirá que “verde estúdio”, “papel”, “acento”, etc. sejam sempre os mesmos em todas as implementações e mantidos no espírito visual pretendido.

3. Espaçamento e Layout

O sistema utiliza **princípios de espaçamento uniformes** (muito influenciados pelas classes do Tailwind) que devem ser padronizados em um guia para garantir que todos os elementos mantenham proporções e respiro consistentes:

- **Grid e estrutura base:** Todas as telas estão centradas em um “frame” móvel de largura máxima **414px** (largura típica de celular grande)²⁰, seguindo a abordagem *mobile-first*. O container `#mobile-frame` define esse limite e aplica `display:flex` coluna, permitindo que header, conteúdo e footer se empilhem verticalmente de forma consistente²⁰. Documente que **qualquer tela nova** deve usar essa mesma estrutura – um contêiner central responsivo com largura máxima 414px e preenchimento lateral padrão – para que nenhuma tela fique com largura diferente ou conteúdo espremido/estourando horizontalmente.

- **Margens e padding consistentes:** As telas apresentam *paddings* internos regulares, geralmente múltiplos de 4px (sistema **4px grid** do Tailwind). Por exemplo, os headers têm padding horizontal de 1.25rem (20px, classe `px-5`) ou 1.5rem (24px, `px-6`) dependendo da tela ⁴⁵ ⁴⁶, e padding vertical como `pt-8 pb-4` em alguns casos ⁴⁷. Recomenda-se unificar esses valores para que todos os headers tenham o mesmo espaçamento interno – sugere-se `px-6` (24px) e `pt-8 pb-4` como padrão, alinhado às telas mais novas ⁴⁷. Da mesma forma, os conteúdos das páginas usam espaçamentos como `p-4` ou `p-6` de acordo com a necessidade. **Defina uma escala fixa** de espaçamentos (8px, 12px, 16px, 24px etc.) e aplique-a uniformemente: por exemplo, margens entre seções de formulário sempre 24px (`mb-6`), espaçamento interno de cards 16px (`p-4`), distância entre título e subtítulo 4px (`mb-1`), etc. A inspeção do HTML mostra repetidamente esses valores, indicando uma adesão parcial a esse grid – padronizar formalmente evitará quaisquer discrepâncias. A classe `space-y-*` do Tailwind é muito usada para espaçar verticalmente itens em formulários (ex: `space-y-4` entre campos ⁷ ⁴⁸) – decidir que `space-y-4` (16px) é o intervalo padrão entre campos de formulário e listar isso na documentação ajudará a manter todos os formulários coerentes.
- **Tamanhos de componentes:** Estabeleça tamanhos padrão para elementos de UI comuns. Atualmente, botões circulares icônicos (como o voltar "<" ou o botão "+") têm diâmetro de **40px** (classe `w-10 h-10`) em geral ¹⁷ ⁴⁹, enquanto alguns botões menores (como setas do calendário mensal) usam **36px** (`w-9 h-9`) ⁵⁰. Para consistência, defina que botões de ícone pequenos = 40px, e use 36px apenas onde o espaço é muito limitado. **Cards e containers** apresentam cantos arredondados grandes (geralmente 16px ou mais) e sombras suaves – por exemplo, cartões de agendamento e listas têm `rounded-2xl` (16px raio) ou `rounded-3xl` (~24px) ⁵¹ ⁴⁶. Observou-se uma leve variação: a header da Agenda tem cantos arredondados 2xl ⁴⁵ enquanto em outras telas usam 3xl ⁴⁶. Seria melhor unificar: recomenda-se **bordas arredondadas 3xl (~24px)** em headers e cartões principais para um visual consistente e moderno (os protótipos mais recentes parecem optar por 3xl). Itens menores (botões, inputs) usam `rounded-full` (totais para círculos) ou `rounded-xl/2xl` conforme o caso – isso pode ser padronizado no guia (p.ex.: botões e inputs com 16px de borda arredondada para um estilo pill consistente, já visto nos campos e botões atuais ⁵² ⁵³).
Consistência de layout entre telas: Verificou-se que todas as telas usam layouts parecidos – um header sticky no topo, conteúdo principal rolável, e às vezes um footer nav fixo. **Seções equivalentes devem ter tratamento equivalente.** Por exemplo, as telas de formulário ("Novo Agendamento" e "Novo Cliente") ambas apresentam o conteúdo dentro de cartões brancos com cantos arredondados grandes, separados por seções claras com títulos pequenos em cinza ⁵⁴ ¹¹. Isso sugere um layout padrão para formulários: cada seção dentro de um card com padding interno 20px, título de seção no topo. O guia de layout deve esclarecer isso (ex: "Use cards brancos (`bg-white shadow-soft rounded-3xl`) para agrupar blocos de formulário ou lista em todas as telas. Dentro de cada card, mantenha 16px de espaçamento interno e, se necessário, títulos de seção sticky no topo"). Nas listas de clientes, segue-se padrão similar: agrupa-se itens por letra dentro de um cartão branco com borda arredondada e sombra leve ⁵⁵ ⁵⁶. Assim, embora os contextos variem, a *linguagem de design* é a mesma – e deve continuar sendo, para que o usuário sinta unidade visual navegando entre telas.
- **Espaçamento externo e alinhamento:** A maioria dos elementos está **bem alinhada e espaçada**, usando flex e grid do Tailwind para distribuir conteúdo. Por exemplo, títulos e botões de ação nos headers usam `justify-between` para ocupar todo o espaço lateral de forma equilibrada ⁴⁶ ⁵⁷. Listas e formulários têm padding lateral consistente de 24px (`px-6`) para alinhamento uniforme das bordas do conteúdo ⁵⁷ ⁵⁸. Esse alinhamento deve ser mantido – um mesmo recuo lateral em todas as telas faz com que cartões e texto se alinhem verticalmente quando o usuário troca de tela. Certifique-se de que não haja telas com "sangria" diferente. No código atual isso já foi observado e deve ser reforçado.

- **Utilização de espaços em branco:** O design mantém uma “respiração” elegante – não é excessivamente denso. Os cartões possuem margem inferior (ex: `mb-4` entre agendamentos do dia ⁵⁹ ⁶⁰, ou `mb-3`/`mb-4` em títulos) que evita junções coladas. Continue seguindo a regra de **nunca colar componentes sem uma margem mínima**. Defina no guia: margem padrão abaixo de títulos, entre inputs, entre botões, etc. (muitos já no código como `0.5rem`, `1rem`, `1.5rem` dependendo do caso). Essa regularidade de espaçamento contribui para a organização visual.
- **Sombras e elevação:** A padronização de sombras é parte do layout. Foi definido no Tailwind uma sombra suave chamada `shadow-soft` (`0 4px 20px -2px rgba(106,128,108,0.15)`) presente nos cards e headers ⁶¹. Em alguns locais, sombras maiores (`shadow-float` ou `shadow-xl`) são usadas para elementos acima de outros (ex: botão FAB e modais usam sombras mais intensas para destacar na frente ⁶² ¹⁵). Inclua no design system quais níveis de elevação existem: *suave* para elementos no fluxo normal (cards, header), *elevado* para elementos flutuantes (FAB, menus expansíveis), etc., assegurando que todos os desenvolvedores usem as classes corretas em vez de estilos arbitrários. No código, por exemplo, `shadow-float` é usado para o menu flutuante e modal ⁶² ⁶³ – mantenha essa consistência em todos os contextos semelhantes (qualquer *popup/overlay* deve usar a mesma sombra *float*).

Resumindo, a malha de espaçamento e layout está **bem pensada e relativamente consistente**. Pequenas divergências (p.ex. cantos `2xl` vs `3xl`, paddings de header levemente diferentes) devem ser ajustadas para uniformidade total. Ao criar um guia de “Layout e Espaçamento”, baseie-se nas regras já seguidas nessas telas, garantindo que novos desenvolvimentos usem os mesmos **valores de margem, padding, largura e proporções**. Isso manterá a experiência coesa e visualmente harmônica em todo o sistema.

4. Botões e Ícones

Os botões e ícones no sistema apresentam um estilo coeso, mesclando a paleta de cores da marca com formatos consistentes. A auditoria foca em garantir que cada tipo de botão e uso de ícone siga uma lógica única:

- **Estilos de Botão (primário, secundário, etc.):** Identificam-se pelo menos três estilos de botão:
 - **Botão Primário:** geralmente de **ação principal** (ex: botão `submit` “Agendar” ou “Salvar”). Caracteriza-se por **fundo verde estúdio e texto branco**, largura total quando apropriado, canto arredondado `2xl` e fonte `extrabold` ¹⁶. Exemplo: o botão de agendar ocupa toda largura (`w-full`) com `bg-studio-green text-white font-extrabold rounded-2xl` ¹⁶. Em hover, fica verde mais escuro (`studio.dark`) ¹⁶. Este estilo deve ser padrão para qualquer ação primária em qualquer tela (confirmar, continuar, salvar). Documente-o como **Primary Button** e enfatize consistência: mesmo cor, fontes e comportamento em todas instâncias.
 - **Botão Secundário:** usado para ações de menor ênfase ou opções alternativas. O sistema costuma implementar este como **fundo claro com texto verde**, ou borda verde com fundo neutro. Por exemplo, o botão “Hoje” na Agenda é um secundário – tem fundo verde claro (`bg-studio-light`) e texto verde, e ao interagir inverte para fundo verde e texto branco ¹⁸. Outro exemplo: o botão “Fechar” no modal é secundário – fundo `studio.light` e texto verde, contrastando com o primário “Editar” verde sólido ⁶⁴. Recomenda-se padronizar: botões secundários terão **fundo neutro (papel ou verde claro)** com **texto na cor primária** e talvez um **borda fina** da cor primária ou transparente. No código alguns têm borda transparente que colore no hover (ex: botões de horário livre têm borda cinza que fica verde no hover ⁶⁵). Escolha uma apresentação única (bordas sim ou não) para o guia. O importante é que em todas as telas o usuário identifique: botões verdes sólidos = ação principal; botões claros contornados = ação secundária.

- **Botão “Texto” ou Terciário:** são links ou ações de mínima ênfase, exibidos apenas como texto colorido. Exemplo: o botão “Buscar” no cabeçalho da Agenda é apenas texto cinza que fica verde no hover, sem fundo ou borda ⁶⁶. É usado para ações como links de navegação ou filtros (menos destaque). Deve ser consistente: texto no estado normal em cinza médio (`text-gray-400`) ou verde, com indicador de hover (sublinhado ou cor verde) para indicar clicável ⁶⁶. No design system, classifique esses como **Text Buttons** e use-os parcimoniosamente – geralmente em lugares onde um botão cheio seria visualmente pesado (como dentro de títulos ou em barra de ferramentas).
- **Botões de Ação Flutuante (FAB e menu):** O sistema possui um **botão FAB circular** principal – o “+” no canto inferior, verde, circular e de destaque máximo ¹⁵ – e um menu de ações flutuantes que se expande a partir dele ⁶² ¹⁵. Esses seguem um design especial: o FAB é um **círculo verde** de 56px aprox. com ícone branco grande, com leve efeito de hover/active scale ¹⁵. Quando expande, os botões do menu flutuante aparecem como **pílulas brancas com ícone colorido**: por exemplo, “Bloquear Plantão” aparece como um botão alongado branco com texto cinza e um círculo vermelho contendo o ícone ⁶². Ao hover, o círculo icônico inverte (fica vermelho cheio com ícone branco) ⁶⁷. Isso é muito elegante e deve ser mantido: defina o padrão de **botão flutuante** (geralmente único, verde, bottom-right) e **ações rápidas** (estilo pílula com ícone à direita, cores conforme ação). O importante é que se outras telas tiverem ações flutuantes, reutilizem o mesmo componente visual.
- **Chips e Toggles:** Além dos botões convencionais, há **chips clicáveis** (filtros “Todos, VIP, Atenção...” na lista de clientes) e **toggles** (botões de seleção de opção, como “No Estúdio” vs “Em Domicílio”). Os chips funcionam como botões de filtro: têm formato de pílula com texto extrabold 11px, podendo estar ativos (fundo verde se selecionado) ou inativos (fundo papel, borda fina e texto cinza) ⁶⁸ ⁶⁹. Já os toggles (como a escolha de local) usam um estilo próprio definido no CSS: estado ativo com fundo branco e borda colorida (verde ou roxo) e estado inativo com fundo neutro (`#F3F4F6`) e texto cinza ⁷⁰ ⁷¹. Padronize esses componentes como variações de botão: **Chip** – pequeno, pode ser selecionado/deselecionado, usar classe `.active` para fundo colorido. **Toggle** – um botão de opção mutuamente exclusiva, com classes `.active-studio`, `.active-home`, etc., conforme o selecionado ⁷⁰. No HTML eles adicionam/removem classes para indicar seleção ⁷² ⁷³. O design system deve documentar como estilizar toggles e chips, e incentivar reutilizar esses padrões (por ex., se surgir outro conjunto de botão binário, usar o mesmo estilo de toggle, não inventar um novo).
- **Uso de Ícones:** Os ícones são providos pela biblioteca Lucide (importada em todas as telas ⁷⁴). Eles têm estilo linear e coerente. As regras para uso de ícones incluem:
 - **Tamanho consistente relativo ao texto:** Observamos que os ícones são dimensionados proporcionalmente ao texto ao lado deles. Exemplos: ícones em labels pequenas usam classe `w-3 h-3` (~12px) quando ao lado de texto 11px ⁷⁵ ⁷⁶; ícones em botões regulares usam `w-5` ou `w-6` (~20-24px) para harmonizar com textos 14px ¹⁷ ⁷⁷; ícone do FAB é maior `w-7` (~28px) já que não há texto e é ação principal ⁷⁸. Essa lógica deve ser registrada: defina tamanhos de ícone padrões – **pequeno** (12px) para texto pequeno, **médio** (16px) para texto normal, **grande** (24px) para botões sem texto ou destaque especial.
 - **Cor e contexto:** Ícones geralmente seguem a cor do elemento textual ou contexto em que estão. Se um botão tem texto verde, o ícone também é verde; se um selo de status é roxo, ícone roxo; se está dentro de um pill colorido, provavelmente ícone branco (como no caso do ícone “hospital” dentro de círculo vermelho no FAB menu) ⁶⁷. **Nunca usar cores aleatórias nos ícones** – sempre herdar ou especificar conforme o padrão do componente. Por exemplo, ícones do menu inferior são cinza quando inativos e verde quando ativo, acompanhando a cor do texto do rótulo ⁷⁹. Garantir que isso ocorra em todos os lugares (já é feito no código via classes utilitárias).
 - **Bordas e fundos de ícone:** Alguns ícones aparecem dentro de formas geométricas (círculos) com fundo colorido, para representar um *badge* visual (ex: ícones de “building” e “car” dentro de

círculos coloridos nos cards de agendamento ⁸⁰ ⁴⁴, ou as letras-iniciais dos clientes em círculos coloridos na lista ⁸¹ ⁸²). O design system deve ter um componente **Avatar/Ícone Circular**, definindo tamanho (padrão 40px, classe `w-10 h-10` já usada ⁸¹), tipografia (inicial do nome em font-serif bold, como "AC" no avatar do cliente, enquanto ícones lucide centralizados seguem o mesmo sizing), e esquemas de cor: p.ex., *avatar padrão cliente* usa fundo verde claro se sem tag especial, ou usa cor distinta se VIP (no exemplo, VIP Amanda tem fundo roxo claro, possivelmente indicando tag VIP) ⁸². Esse mesmo estilo de círculo com icon/letra é reutilizado em vários contextos – padronizar evitará recriações inconsistentes.

- **Estados visuais (hover/ativo/desabilitado):** Apesar de em dispositivos móveis não haver "hover" real, o protótipo considera mudanças de estilo no hover (úteis para desktop e para feedback de toque). Todos os botões têm transições suaves (transition) e mudanças de cor ou escala em interação:
 - Botões primários escurecem no hover (verde escuro) e reduzem levemente no active (ex.: `active:scale-95` no FAB ¹⁵ ou `active:scale-[0.99]` nos botões de agendamento ⁵⁹).
 - Botões secundários normalmente invertendo cor de fundo/texto no hover (ex: círculo de voltar fica verde cheio no hover ¹⁷, chips de filtro provavelmente ficam com cor ativa se clicados, etc.).
 - Itens de lista usam highlight de fundo cinza claro no hover e uma leve escala no active para simular o clique ⁵⁶ ⁸³.

A regra geral é: **feedback visual em todas interações de clique**. Deve-se padronizar o uso de `active:scale` (redução ~1-2%) para todos os elementos clicáveis principais – isso já aparece consistentemente (lista de clientes, itens da agenda, botões grandes, todos usam) ⁵⁹ ⁸⁴. Igualmente, usar `transition-all` ou `transition-colors` para suavizar mudanças de cor. O guia deve apontar: "Ao tocar, botões e itens sofrem leve redução de tamanho (escala ~0.98) e/ou mudança de cor de fundo para indicar ação; este comportamento deve ser aplicado uniformemente a todos botões e componentes interativos.". Também padronize estados *disabled*: nos arquivos não vimos muitos exemplos, mas seguindo a paleta, um botão desabilitado deve usar texto e bordas em cinza claro e fundo neutro, sinalizando indisponibilidade (pode-se usar `opacity-50` como foi aplicado no menu inferior para itens inativos ⁸⁵).

No geral, **há muita coerência nos botões e ícones atuais** – a padronização formal vai principalmente documentar essas boas práticas já aplicadas. Assim, cada desenvolvedor saberá, por exemplo, que um novo botão de ação principal em qualquer tela deve ser idêntico aos já existentes (verde cheio, texto branco, canto arredondado, extrabold, etc.), e que qualquer ícone a ser adicionado deve preferencialmente vir da biblioteca Lucide e seguir os tamanhos/cor padrão do sistema.

5. Títulos e Hierarquia Visual

Garantir que os títulos e seções sigam uma hierarquia consistente é crucial para orientação do usuário. Analisando as telas fornecidas, identificam-se níveis hierárquicos bem demarcados – precisamos padronizá-los para que toda nova tela obedeça à mesma estrutura:

- **Título principal (H1):** Cada tela possui um título principal que identifica a tela atual, geralmente situado no header. Ele é estilizado com fonte **Playfair Display**, tamanho ~2XL (aprox. 22–24px) e cor cinza escuro ⁴. Exemplos: "Sua Agenda de Janeiro" (tela Agenda), "Novo Agendamento", "Meus Clientes" ⁸⁶, "Novo Cliente" ⁴. Este H1 costuma vir acompanhado de um pequeno texto auxiliar logo acima ou abaixo:
- Em telas de seção/gestão, aparece um **label contextual acima do H1** em fonte sans-serif extrabold uppercase, 11px, cor verde ou cinza, indicando a seção do sistema. Ex: "Gestão" acima de "Meus Clientes" ⁸⁶, "Cadastro" acima de "Novo Cliente" ⁴. Isso ajuda a categorizar a tela (ex.: telas de cliente estão sob gestão/cadastro).

- Em telas mais pessoais, o H1 pode vir precedido de uma **saudação ou informação de estado**. Ex: na Agenda, acima do "Sua Agenda de...", há "Olá, Janaína" e um indicador "Online" ⁸⁷. Ambos estão em Lato uppercase pequeno – "Olá, Janaína" em verde extrabold 12px e "Online" em cinza com ícone verde ⁸⁷. Essa composição difere um pouco do caso anterior, mas cumpre papel similar de fornecer contexto ao título principal.

Padronize o uso do H1 e seu elemento auxiliar: no guia, defina que cada tela deve ter exatamente um título principal, em Playfair 2xl, cor texto principal. Se for uma tela de módulo, incluir um label de contexto em Lato extrabold uppercase 11px acima do H1 (cor verde estúdio ou cinza – recomenda-se talvez usar sempre verde para manter destaque da marca, como "Gestão" e "Cadastro" aparecem em verde nos exemplos recentes ⁸⁶ ⁴). Se for tela inicial/pessoal, pode-se usar uma saudação personalizada no mesmo estilo (uppercase extrabold pequena). O importante é manter: **label contextual sem serifa, uppercase pequeno, seguido de H1 serifado maior** – assim todas as telas terão um cabeçalho visualmente similar. - **Subtítulos e descrições (H2/H3 ou parágrafos introdutórios):** Algumas telas complementam o título principal com um pequeno texto descriptivo logo abaixo, em fonte sans-serif pequena e cor neutra. Ex: "Preencha os detalhes do atendimento." logo abaixo de "Novo Agendamento" ¹⁰; ou no modal de detalhes da agenda: título "Detalhes" seguido de um parágrafo "Toque em editar para ajustar o agendamento." ⁸⁸. Essas descrições contextualizam a função da tela. Padrão: fonte Lato **text-xs** (~12px) ou **text-sm** (14px) em cor **text-gray-400** (cinza muted) ¹⁰ ⁸⁸. No design system, determinar que sempre que necessário um subtítulo/descriptivo abaixo do título principal, ele deve usar esse estilo (tamanho pequeno, cor secundária) e margem pequena no topo (**mt-1** pelo que se vê no código ¹⁰). - **Títulos de seção (H2/H3 contextuais):** Dentro das telas, especialmente as de formulário ou lista longa, há subdivisões com seus próprios cabeçalhos: - Nos formulários ("Novo Cliente"), blocos de campos como "Dados básicos", "Endereço", "Saúde & Cuidados" etc. são precedidos por um rótulo sticky no topo do grupo ¹¹ ⁸⁹. O estilo desses rótulos: **<h3>** em **Lato extrabold uppercase 11px**, cor principal verde para a letra indicativa ou seção geral (no exemplo do índice alfabético de clientes, a letra "A", "B" sticky está em verde ⁹⁰ ⁵⁵; mas nos formulários, parece que os títulos de seção estão em cinza uppercase – "Dados básicos" etc. aparecem em cinza muted extrabold ¹¹). Há uma pequena inconsistência: na lista de clientes, optou-se por destacar a letra da seção com verde, enquanto nos formulários os títulos de seção estão em cinza. Seria bom unificar a cor desses **subcabeçalhos** – recomendação: usar verde estúdio para títulos de seção também, destacando-os ligeiramente do texto comum, já que representam marcos importantes na rolagem (como foi feito no índice alfabético dos clientes ⁹⁰). Mantenha as demais propriedades: tipografia sans-serif extrabold uppercase, tamanho bem pequeno (11px). Esses cabeçalhos de seção geralmente estão dentro de elementos sticky semi-transparentes para ficarem fixos no topo ao rolar (usam **backdrop blur** e **bg-paper/90** no CSS ⁹¹). Documentar isso no guia: "**Títulos de Seção:** use fonte sans-serif uppercase 10-11px extrabold; cor de preferência verde principal (ou cinza escuro, mas idealmente escolher um padrão); colocar dentro de um container com fundo semi-transparente para sobrepor o conteúdo ao rolar (aplicável a listas longas e formulários extensos)". - Nos fluxos multi-etapas (como o atendimento), os títulos de seção aparecem numerados. Ex: "1 Cliente", "2 O que e Onde?" no formulário de agendamento interno ⁹². Aqui o padrão é similar: número dentro de círculo verde seguido de título uppercase extrabold cinza 12px ⁵⁴. Esse componente "etapa do formulário" pode ser tratado no design system como um tipo de título de seção também, combinando ícone/ número e texto pequeno. Padrão já consistente nos arquivos: circulo 24px verde com número em branco, texto da etapa em Lato extrabold uppercase cinza ao lado ⁵⁴. Deve-se repetir esse estilo para qualquer wizard ou passo-a-passo no app. - **Destaque de conteúdo (H4/H5):** Dentro de cards ou itens, às vezes há títulos menores que destacam nomes ou informações principais daquele item. Por exemplo, nos cards de agendamento, o nome da cliente "Ana Clara" é renderizado como **<h3>** porém estilizado em Lato extrabold 16px (text-lg) ⁹³; nas linhas da lista de clientes, o nome do cliente é **<h4>** extrabold 14px (text-sm) ⁹⁴ ⁹⁵. São títulos de itens, não globais, mas ainda assim devem ser padronizados: **nomes e títulos de itens** (como nome do cliente, nome do serviço, título de um card)

usam fonte sans-serif **bold/extrabold**, em tamanho levemente maior que o texto normal ao redor, e cor de texto principal. Já informações secundárias dentro do item (tipo de serviço, “Última visita: 15 Jan”) ficam em fonte normal/semibold menor e cor muted²¹²². Isso já é consistente nas telas – formalize no guia que, por exemplo, **título de item/lista = Lato 14px extrabold, cor #2C3333, e subtexto de item = Lato 11px semibold, cor #868E96.** - **Consistência sem sobrecarga:** A hierarquia deve evitar estilos conflitantes. Felizmente, as telas atuais já seguem uma lógica: não há uso de muitas cores diferentes nos títulos (predominantemente verde ou cinza escuro) nem fontes aleatórias. Para manter isso, restrinja a paleta tipográfica dos títulos a apenas as cores de texto definidas (verde estúdio para destaques, cinza 800 para títulos normais, cinza 400 para subtítulos). Assim, um desenvolvedor não usará um azul ou outro tom que não exista na paleta. Da mesma forma, evite sublinhados, itálicos ou efeitos extras em títulos – nada disso é usado atualmente e manter o design *clean* ajuda na sofisticação visual.

Concluindo, a hierarquia visual está **bem estruturada**: títulos de tela chamam atenção na medida certa, seções e itens têm cabeçalhos menores bem definidos. O trabalho de padronização é compilar essas práticas de forma clara – listar cada nível (H1, H2, H3, etc., ou simplesmente “Título de Tela”, “Título de Seção”, “Título de Card/Item”, “Texto de apoio”) com seus estilos de fonte, tamanho, cor e uso previsto. Seguindo isso, qualquer nova informação inserida na interface encontrará seu **lugar visual correto** sem destoar das telas existentes.

6. Responsividade e Mobile-First

Todas as telas fornecidas foram concebidas para mobile, seguindo o princípio *mobile-first*, o que é adequado dado o foco em produção mobile. A auditoria verifica que:

- **Largura máxima e escalabilidade:** Como citado, o container central tem `max-width: 414px` e `width: 100%`²⁰, de modo que em telas menores (ex.: 360px de largura) o layout simplesmente ocupará 100% daquela largura, encolhendo proporcionalmente os elementos. Isso indica responsividade básica já embutida – não há valores fixos maiores que possam quebrar em telas menores. Por exemplo, o uso de classes utilitárias (flex, grid, w-full) garante que listas, inputs e botões ocupem o espaço disponível fluidamente.
- Confirme que nenhum elemento exige largura fixa maior que 100% do container.** Pela inspeção, itens como inputs e botões usam `w-full` ou porcentagens, o que é bom⁵². Apenas componentes como o *slider* de dias têm largura min-content, mas rolável (implementado via `overflow-x`)⁹⁶ – isso é intencional e tratável no mobile via scroll.
- **Breakpoints:** Não se notam breakpoints específicos de desktop (classes `md:` por ex.) nas HTMLs, reforçando que o design alvo é exclusivamente mobile por ora. Documente isso – o design system pode anotar que a interface é otimizada para até 414px e que versões desktop ou tablet terão diretrizes futuras, se for o caso. Entretanto, já podemos recomendar **progressive enhancement**: por exemplo, se a tela for mais larga que 414px (um tablet pequeno), talvez centralizar o container e usar cor de fundo em volta (como já fazem colocando o mobile-frame centralizado em fundo `#222` escuro⁹⁷). Isso está mais relacionado a implementação, mas vale mencionar como boa prática.
- **Fluidez ao redimensionar:** Dentro do container móvel, os elementos usam `flexbox` e `grid` para acomodação adaptativa. Listas de filtros e chips podem rolar horizontalmente se excedem a largura (há `overflow-x-auto` e classes de scrollbar oculto em diversos lugares⁶⁸⁶⁹). Isso é bom para fluidos horizontais (ex: a barra de chips de clientes pode ter muitos itens, e em telas menores ou maiores se ajusta via scroll). Os textos usam `truncate` quando preciso para não quebrar linhas (ex.: nome do cliente longo é cortado com reticências para caber na linha⁹⁴). **Esses padrões evitam quebra de layout** e devem ser mantidos: instruir no guia que, em listas ou menus horizontais, use scroll horizontal com snap se aplicável; em textos de uma linha cujo container é variável (nomes, títulos em cards), usar `text-truncate` para impedir overflow e manter design limpo.
- **Mobile UX - safe area e gestures:** As telas novas demonstram preocupação com *safe areas* do iOS: uso de classes `safe-top` e `safe-bottom` para adicionar padding condicional se houver notch

ou barra inferior [98](#) [99](#). Todos os headers fixos deveriam incluir `safe-top` (os novos já incluem [98](#)) - padronize isso para evitar que conteúdo fique coberto por entalhes/câmera. O mesmo para footers fixos com `safe-bottom`. Adicionalmente, nota-se uso de gestures de swipe: a Agenda permite swipe horizontal nos dias, a tela de clientes implementa transição de tela por slide (detalhes entram pela direita) [100](#) [101](#). Embora essas interações sejam mais comportamentais, vale citar como parte do design mobile-first: **transições baseadas em gesto ou swipe devem ser uniformes** (ex.: se usar swipe horizontal para trocar sub-seção em uma tela, adote padrão semelhante em outras – a Agenda e o Atendimento usam *scroll horizontal snapping* em algumas visões [96](#) [102](#), o que é ótimo). Documente no guia de navegação/behavior que gestures padrão de mobile (swipe entre abas, arrastar lista) são considerados e manterão consistência (por exemplo, sempre que houver abas de igual hierarquia, usar swipe horizontal para navegar, em vez de por exemplo em outra tela usar um comportamento distinto).

- **Desempenho em mobile:** Visualmente, tudo se mantém leve – cores planas, sombras simples e poucas imagens. Isso é favorável. **Atenção** apenas ao uso de *backdrop-filter: blur* (observado em alguns elementos como toasts e header expandido [103](#) [104](#)) – em mobile mais antigo, isso pode ter custo de performance. Mas se for parte do estilo (p.ex. efeito translúcido no rodapé/headers), assegure-se de aplicá-lo consistentemente e teste em dispositivos reais. Está consistente nos locais usados (painel do cabeçalho, barra de estágio, toasts), então é mais uma nota de implementação.

- **Verificação de elementos fora da viewport:** Todos os elementos essenciais ficam dentro de 414px. Ocasionalmente, valores absolutos precisam ser checados – ex.: na lista de clientes, o índice alfabético posicionado `right: 6px` [105](#). Em telas muito estreitas (<320px), 6px from right ainda deixa visível; está ok. O botão flutuante está `right: 1.5rem` (24px) do canto [106](#), também ok. Portanto, não há indicativos de algo quebrando em ecrãs menores. O guia deve incentivar a continuidade disso: usar sempre medidas relativas (%) ou rem) e evitar pixel fixo grande, garantindo adaptabilidade.

Em suma, a interface é **claramente projetada para mobile**, e a consistência aqui significa: todas as telas seguindo o mesmo container, mesmas regras de scroll e responsividade. Para um **produto pronto para produção**, reforce essas boas práticas e teste cenários como: diferentes densidades de tela (ver se fontes muito pequenas continuam legíveis em telas HDPI – possivelmente ajustar se necessário, ex.: 10px pode ficar muito minúsculo). Mas estruturalmente, a base mobile-first está sólida e padronizada.

7. Barra Inferior e Cabeçalhos

Elementos de navegação persistentes, como o menu inferior (navbar) e os cabeçalhos fixos, precisam ser uniformes para que o usuário sempre saiba como navegar e onde encontrar ações principais:

- **Cabeçalhos (header) sticky:** Todas as telas analisadas possuem um header fixo no topo (`sticky top-0`) com fundo branco, sombra suave e cantos arredondados no bottom [45](#) [47](#). Essa construção deve ser idêntica em todas as telas principais:
- **Altura e padding:** Como mencionado, padronize alturas. Parece que a altura varia conforme o conteúdo (por exemplo, Agenda tem header um pouco mais “baixinho” – pt-5 pb-3 – enquanto outras têm pt-8 pb-4). Seria ideal unificar, mas se precisar variar (e.g. tela Agenda inclui mais elementos no header), mantenha pelo menos o *estilo visual* igual: sempre fundo branco, sombra (`shadow-soft`), borda arredondada inferior, e o conteúdo alinhado pelos paddings laterais.
- **Conteúdo do header:** Consiste geralmente em um **título principal** e possivelmente botões de ação (como o botão de adicionar novo cliente no header de “Meus Clientes” [49](#) ou botão de voltar em formulários [46](#)). A posição desses elementos deve seguir padrão: botões de navegar (voltar) à esquerda, título ao centro (ou alinhado à esquerda se houver mais espaço, mas nos exemplos está centralizado quando possível [4](#)) e botões de ação (como adicionar) à direita. Na Agenda, como é tela principal, não há botão voltar – em vez disso há saudação à esquerda e ações à direita [87](#) [107](#). Ainda assim, a estrutura é *esquerda: info do usuário, centro: título, direita: ações*:

ações rápidas. No design system, defina templates de header: **Header principal** (ex.: Agenda, inclui saudação/estado + título grande + ações como busca), **Header de lista** (ex.: Meus Clientes, inclui talvez só título e botão adicionar), **Header de formulário** (com botão voltar + título + talvez nome da marca/status) ⁴⁶. Apesar das variações, todos compartilham estilos base, então enfatize: "Todos os cabeçalhos terão fundo branco, ocuparão a largura total do frame, serão fixos no topo com sombra. Elementos dentro devem respeitar margens padrão e alinhamentos (justificados entre si conforme necessidade)". Isso garante a *uniformidade visual e comportamental* – por exemplo, todos eles devem ocultar parte do conteúdo atrás de si ao rolar (já que são sticky), então reservar espaço (safe-top) como já foi considerado.

- **Barra de título expandida:** Uma tela (Atendimento) tem um header mais complexo com expansão/collapse (mostrando info extra quando aberto) ¹⁰⁸. Embora seja um caso especial, se outros módulos tiverem cabeçalhos expansíveis, devem seguir o mesmo mecanismo – fundo com gradiente, transição suave ao fechar ^{108 109}, etc. O guia pode referenciar isso como um padrão de "Cabeçalho Expansível" para casos onde cabe muito conteúdo no topo.
- **Barra de navegação inferior:** O app conta com um menu fixo inferior com ícones e labels para navegar entre as seções principais (Agenda, Clientes, Caixa, Menu) ¹¹⁰. Auditando esse componente:
 - O estilo é consistente: fundo branco, borda superior fina cinza claro, ícones Lucide de 24px, labels de 10px. O item **ativo** está em verde com texto extrabold, itens inativos em cinza com texto médio ^{79 111}. Esse padrão deve ser **imutável** – não usar cores diferentes em telas diferentes. Ou seja, quando o usuário navega até "Clientes", o ícone de usuários deve ficar verde e label extrabold, e "Agenda" virar cinza, etc., seguindo exatamente o mesmo esquema invertido.
 - **Tamanhos e toque:** Cada botão da barra ocupa ~25% da largura (são 4 itens dividindo igualmente, cada com ~96px dentro de 414px). A área de toque vertical inclui o ícone + texto e padding vertical (py-3) ¹¹⁰, totalizando cerca de 48px de altura – isso está bom para acessibilidade de toque. Apenas documente para manter sempre no mínimo essa altura e espaçamento (evitando adicionar itens demais ou reduzir a tocabilidade).
 - **Ícones uniformes:** Todos os ícones são de mesma família e tamanho. Não alterá-los por outros estilos. Se novos itens de menu forem adicionados, usar ícone Lucide e nome curto similar (2-3 sílabas) para caber em 10px font.
 - **Comportamento fixo:** A barra fica fixed bottom (com safe-bottom se necessário). O guia deve instruir a sempre incluir o `<nav>` com essas classes para todas as telas principais. Observamos que nas telas multi-etapa internas (como formulários "Novo Agendamento" ou telas de detalhe abertas sobre a lista) possivelmente ocultam a barra de navegação principal – o arquivo de Lista de Clientes, por exemplo, mostra duas "screens" (lista e detalhes) e não inclui o `<nav>` explícito (pode ser que a nav esteja somente na tela principal e ao abrir detalhes ela continua por baixo) ^{112 57}. Isso indica: **a barra inferior pertence às telas raiz de navegação, mas não às telas de diálogo ou push internas.** Considere documentar essa regra: tela de detalhe que abre em cima (como um modal tela cheia ou pseudo-nova tela) não duplica a barra; a nav inferior permanece apenas nas seções principais. Isso evita barras duplicadas e confusão de navegação.
 - **Elementos persistentes coerentes:** Tanto o header quanto a navbar têm design consistente de fundo branco semi-transparente quando sobrepõe conteúdo. Percebe-se uso de `background-filter: blur(10px)` e opacidade ~0.92 em alguns desses elementos (nav do atendimento e toast, por ex.) ^{113 104}. Seria interessante aplicar isso de forma uniforme: se todos os headers e footers tivessem uma leve translucidez, manteria um estilo sofisticado (similar ao iOS). Contudo, atualmente a barra inferior da Agenda parece opaca normal ¹¹⁰, enquanto a barra de etapas do atendimento usa translucidez e blur ¹¹³. Talvez separar: navbar principal opaca (foco na navegação clara), componentes flutuantes ou secundários podem ter blur. De qualquer forma, esclareça essa diferença no guia para não causar disparidade acidental (ex: dois footers com comportamentos visuais diferentes sem motivo).

- **Indicadores e detalhes em barras:** O header Agenda tem um recurso particular – um **toggle de visualização Dia/Semana/Mês** embutido logo abaixo do título ¹¹⁴. Esse controle aparece como uma barra segmentada dentro do header, com botões formatados (um ativo em branco com texto verde, os outros em texto cinza) ¹¹⁴. Embora seja um componente específico, se outros módulos tiverem controles similares no header, deve-se seguir o mesmo estilo (fundo papel, botões com fonte extrabold 12px, canto arredondado). O design system poderia chamá-lo de **Segmented Control** e documentar estilo e comportamento (similar ao iOS segmented control, visualmente).

No geral, a uniformidade dos **containers de navegação** já está bem encaminhada. A padronização formal garantirá que: - Todo desenvolvedor use o template de header fornecido, sem alterar cores ou sombras. - A barra inferior seja reutilizada exatamente igual em todas as telas de seção principal. - Qualquer variação (header com aba, header expansível) seja implementada com os mesmos padrões base (mesmas cores, fontes) para não parecer outro aplicativo.

8. Interações e Comportamentos Padronizados

Por fim, além da aparência estática, é importante que as **interações do usuário** – rolagem, transições de tela, animações e respostas a cliques – sigam padrões consistentes em todas as telas, proporcionando uma experiência coesa. Observações e recomendações:

- **Rolagem de conteúdo:** Todas as telas usam scroll vertical padrão para conteúdo principal, com headers fixos. Isso já está consistente. Elementos específicos aproveitam scroll horizontal quando faz sentido (ex.: calendário de dias deslizando, lista de chips horizontal, pager de etapas). Esse uso está apropriado e **uniforme dentro de cada contexto** – por exemplo, a *Agenda Dia* implementa um carrossel horizontal com snap para dias ⁹⁶, e o Atendimento tem um pager horizontal para etapas ¹⁰². Ambos usam `scroll-snap` e comportamentos nativos similares, o que é bom. Padronize diretrizes de rolagem: *quando quebrar conteúdo em páginas ou seções relacionadas, considere swipe horizontal com snap (ex.: etapas do atendimento, visualizações dia/semana/mês da agenda)*. Deixe claro no design system que esse é um padrão oficial, evitando que outra equipe tente, por exemplo, usar um carrossel diferente ou sem snap. Além disso, inclua a classe utilitária `.no-scrollbar` (já presente no CSS compartilhado) para esconder scrollbars quando aplicável ¹¹⁵ em todas as implementações que suportem (apenas cosmeticamente nas plataformas móveis modernas).
- **Transições de telas e modais:** O aplicativo implementa transições animadas sutis ao navegar entre “telas” internas: na lista de clientes, ao abrir detalhes do cliente, a nova tela vem da direita (`transform: translateX(0)` de `translateX(10%)`) com animação ease 260ms ¹¹⁶ ¹¹⁷. Isso dá uma sensação de continuidade. Da mesma forma, modais (como o detalhamento do agendamento) são inseridos centralizados com leve fade/scale (aparecem com `scale(1)` de 0, e backdrop escurece) – embora não tenhamos visto a definição exata, o código HTML sugere que o modal tem classes `.hidden` trocadas via JS para animar, e um backdrop semi-transparente é exibido ¹¹⁸. **Padrão:** *transições suaves, rápidas (cerca de 250ms), com easing consistente*. No código observa-se `transition: transform 260ms ease, opacity 260ms ease` para telas deslizantes ¹¹⁹ e keyframes fadeIn 0.25s ease para certas aparições ¹²⁰. Recomenda-se definir um **timing padrão** (ex.: **250ms, easing = ease-in-out**) para todas as animações de entrada/saída de componentes, e usar sempre esse valor para não ter umas muito rápidas e outras muito lentas. Documente também os tipos de transição: *slide in from right* para avançar em hierarquia (ex.: detalhes sobre lista), *slide in from bottom* talvez para modais de baixo (não visto aqui, mas se tiver), *fade in* para conteúdos que aparecem dentro da mesma tela (como a animação de fade nos cards da agenda ao mudar de aba). Consolidar isso evita que

desenvolvedores implementem uma tela com animação abrupta ou nenhuma animação quebrando a continuidade esperada.

- **Feedback de cliques e toques:** Já mencionamos no item de botões, mas reforçando: **todas as interações tátteis oferecem feedback imediato**. Isso é crítico em mobile. O sistema usa: highlight de cor no elemento (ex.: hover bg-gray-50 em linhas de lista ⁸⁴), leve redução de escala no momento do clique (`active:scale`), e alterações de ícone/cor para indicar seleção (ex.: toggles “No Estúdio/Em Domicílio” trocam de cor e ícone visivelmente quando selecionados ⁷² ⁷³). Esses comportamentos devem ser padronizados para todos os componentes interativos:
- **Active State:** Como regra, qualquer botão ou item clicável deve ter um estado `:active` definido (no Tailwind, muitos já aplicam `active:scale-[.]`, ou via CSS custom no caso dos stage-dots pulse etc.). Isso garante que mesmo em dispositivos de toque, o usuário sinta o elemento reagindo.
- **Hover State (simulado):** Em mobile não há hover, mas no desktop ou protótipo, foi implementado hover para depurar e para uso eventual em PWA no desktop. Mantenha por consistência: no guia de estilos, descreva o estado hover de cada componente (botão primário escurece, secundário inverte, item de lista fica cinza claro, etc.) – assim, se um dia houver versão web, já está consistente.
- **Disabled State:** inclua também como padronizar visualmente quando botões estiverem desabilitados (provavelmente usando `opacity-50` e removendo sombra, e tornando não clicável). Mesmo que não muito visível no protótipo, é uma boa prática documentar.
- **Consistência em componentes interativos similares:** Caso de uso: os **stage dots** (etapas pequenas no atendimento) são uma forma de indicar progresso. Eles usam classes `.dot.running` (etapa atual, verde escuro) e `.dot.done` (etapas concluídas, verde sucesso) ¹²¹ . Se outro fluxo usar indicadores, deve reutilizar esse estilo (talvez mudando ícone se necessário, mas mantendo o visual de pílulas com bolinha dentro e texto uppercase 10px). Portanto, defina *Components de Estado* como esses para reaproveitamento. Outro exemplo: **toast** de notificação aparece no design (no Novo Cliente definiram um container `#toast` com fundo branco translúcido e blur) ¹²² ¹²³ . Se toasts forem usados em outras telas, todos devem ter aparência idêntica (mesma posição central em bottom, mesmo estilo de fundo e texto). O styleguide deve ter seção “Notificações/Toast” cobrindo isso.
- **Animações de ícones e microinterações:** Notei no FAB o ícone de “+” gira na abertura/fechamento (tem `transition-transform duration-300` no ícone ¹²⁴ , provavelmente para rodar 45 graus e virar “x” quando o menu abre). Microinterações assim enriquecem a experiência. Deve-se manter coerência: se um ícone animar de uma forma em um lugar (ex.: o + vira x no FAB), outro lugar que use um conceito semelhante deve animar do mesmo jeito para manter familiaridade. Documente essas microinteractions, talvez com descrições ou GIFs no guia, para que a equipe de desenvolvimento as implemente consistentemente.
- **Performance e suavidade:** As interações padronizadas devem ser concebidas pensando em performance mobile – evitar animações travadas ou muito longas. Como dito, todas são ~0.25s e usam transform/opacidade (ótimo, são propriedades baratas). Continue assim. Recomenda-se evitar animações de layout pesado (p.ex. não animar height de listas grandes sem necessidade) – no atual, só se anima height no acordeão de endereço, e com transition 0.4s ¹²⁵ , o que é ok. Padronize no guia que transições devem usar CSS quando possível e propriedades com aceleração, para garantir responsividade do app.

Em síntese, a padronização de interações visa que o usuário **tenha a mesma resposta do sistema a cada ação**, independentemente de qual tela esteja. O estilo atual já demonstra um cuidado com isso – a tarefa aqui é consolidar todas essas interações previstas em um capítulo do design system, cobrindo: estados de botão, transições de tela/modal, animações de componentes especiais (toggles, FAB, etc.), e

guidelines gerais (tempo de animação, easing, feedback tátil). Assim, além de bonito visualmente, o app se comportará de forma **coerente e intuitiva** em toda a jornada do usuário.

□ Proposta de Documentação Visual – UI/UX Design System

Para apoiar a manutenção desse estilo consistente, sugere-se criar uma documentação (estilo *Design System* do produto) organizada em seções claras. Essa documentação pode ser estruturada em pastas e arquivos (ou páginas, se online), da seguinte forma:

- **1. Tipografia** – *Guia de fontes e textos*: Detalhar as fontes usadas (Playfair Display, Lato) com seus pesos disponíveis ¹, quando usar cada uma, escala de tamanhos e exemplos. Incluir tokens ou classes utilitárias para estilos de texto padronizados (ex.: `.heading1`, `.heading2`, `.label`, `.paragraph`) se houver, ou indicar as classes Tailwind correspondentes). Também abordar espaçamento de texto (tracking) em casos de labels uppercase ⁴. Este documento assegura que todos os títulos, labels e parágrafos novos sigam o padrão estabelecido.
- **2. Paleta de Cores** – *Guia de cores da marca*: Listar cada cor do sistema com amostra e código (hex/RGB). Por exemplo: **Verde Estúdio** (#6A806C) – primária, uso em botões primários e destaque de texto ¹²; **Verde Escuro** (#4E5F50) – hover/active do primário ¹²; **Verde Claro (Studio Light)** (#F3F6F4) – fundos de destaque ¹²; **Acento** (#D4A373) – detalhes e ícones especiais ¹²⁶; **Papel** (#FAF9F6) – fundo geral ¹⁹; Neutros (Text Main #2C3333, Text Muted #868E96, Line rgba(44,51,51,0.06)) ³¹ ¹²⁷; **Sucesso (Ok)** (#16A34A) ¹³, **Aviso (Warn)** (#F59E0B) ¹³, **Erro (Danger)** (#DC2626) ³¹; **Informativo (Domicílio)** (#A855F7) ¹²⁸ etc. Para cada cor, descrever utilizações típicas (ex.: Danger – textos/ícones de erro, fundos de alertas; Muted – textos secundários, ícones inativos). Incluir também se há variações de opacidade usadas (por ex.: preto 40% para backdrop de modal ¹¹⁸, branco 92% para fundo translúcido). Isso servirá de referência rápida para designers e devs, evitando tons fora do padrão.
- **3. Componentes Reutilizáveis** – *Catálogo de componentes UI*: Aqui convém subdividir em categorias de componentes, cada um com especificação visual e exemplos de uso. Sub-pastas/arquivos sugeridos:
 - **Botões** – Exemplificar botões Primários, Secundários, Terciários (Texto), Toggle e FAB. Para cada, mostrar estado normal, hover, ativo, desabilitado, e fornecer código exemplo (classes Tailwind ou estilo CSS). Incluir variantes de tamanho se houver (p. ex., botão ícone pequeno vs botão largo). Citando exemplos reais: botão primário “Salvar” ¹⁶, secundário “Fechar” ¹²⁹, FAB “+” ¹⁵, toggle “No Estúdio/Em Domicílio” ⁷² ⁷³, chip “VIP” ativo/inativo ⁸².
 - **Campos e Inputs** – Mostrar estilos de inputs de texto, selects, textareas, etc. No sistema atual, os inputs têm design arredondado com ícone interno e foco com borda verde clara ¹³⁰ ¹³¹. Documentar: estilo normal (fundo papel, borda cinza clara, texto escuro), placeholder (texto cinza 90% ¹³²), foco (borda verde translúcida, box-shadow leve) ¹³³, campo inválido (borda vermelha, as classes `aria-invalid=true` definidas no CSS) ¹³⁴. Incluir também exemplo de input com ícone prefixado (como campo de busca com ícone de lupa) ⁶ e máscaras como telefone.
 - **Cards e List Items** – Componentes que servem de container clicável: cartão de agendamento, item de lista de cliente. Explicar o layout: padding, sombra, borda lateral colorida se aplica (ex.: card de agendamento confirmado com borda verde, domicílio com borda roxa ¹³⁵ ¹³⁶), uso de

- **truncate** em textos longos, ícones de ação (chevron para indicar naveável ¹³⁷). Também incluir o padrão de separador de itens dentro de um card (linha pontilhada ou sólida) ¹³⁸ ²⁵.
- **Pills e Tags de Status** – Definir a classe `.pill` ¹³⁹ e `.chip` ¹⁴⁰ e quando usar. Exemplos: pill de “Confirmado” ⁴⁰, tag “VIP” em lista de clientes ¹⁴¹. Indicar paleta de cores para cada status (confirmado = verde claro, cancelado = vermelho, etc., conforme houver).
- **Nav Bar e Header** – Componentizar o header e o footer nav para fácil reutilização. Mostrar o header genérico com título, e variações (com botão voltar, com botão ação, com toggle segmentado como na Agenda ¹¹⁴). Para o nav inferior, mostrar item ativo vs inativo, e enfatizar que a ordem e ícones devem ser conforme especificado.
- **Modais e Popups** – Descrever o estilo do modal (fundo branco arredondado, padding, título font-serif 18px, texto de apoio xs ⁸⁸, botões alinhados à direita em row ou distribuídos) e do backdrop semitransparente ¹¹⁸. Incluir também toasts (posição central inferior, fundo blur branco, texto pequeno) ¹²² e o *timer bubble* (círculo flutuante translúcido) se for componente generalizado para usar em outros contextos.
- **Ícones e Avatares** – Especificar que todos ícones vêm do Lucide (lista de alguns usados corrente: calendar, users, etc.) e mostrar os tamanhos padronizados em contexto. Incluir avatar de iniciais (40px círculo com fundo variável) e ícones dentro de círculos (como no card de agendamento).
- **(Outros componentes relevantes)** – Qualquer outro padrão que emerja, como barras de progresso (stage dots do atendimento), índice alfabético (lista de letras rolável na lista de clientes), etc., podem ter subseções aqui.

Cada componente deve ter capturas ou ilustrações (quando possível) e código de exemplo. O foco é permitir a **reutilização**: por exemplo, se for criar um novo módulo de “Funcionários”, pode-se usar os mesmos componentes de Lista de Clientes (avatar + nome + chevron) sem criar estilo novo.

- **4. Regras de Layout e Espaçamento** – *Guia de composição*: Explicar as diretrizes de grid e spacing discutidas. Exemplos de conteúdo:
- Grid base de 4px (todos os margin/padding multiplicam 4).
- Container mobile de largura máxima 414px, centralizado com fundo escuro externo para simular device ⁹⁷.
- Uso de flexbox e grid para dividir áreas (e.g. dois botões lado a lado usando `grid-cols-2 gap-2` como no toggle de local ⁷¹ ⁷³, ou colunas no formulário de endereço).
- Padrões de **padding**: ex.: outer padding das telas 24px; padding interno de cards 16px; etc., conforme levantado.
- **Breakpoints**: nota de que o design é mobile-first e atualmente focado em <=414px, sem layout específico de tablet/desktop – mas pode citar comportamento em telas maiores (centralização do conteúdo).
- **Safe areas**: instruções de usar `safe-top` e `safe-bottom` nas views para evitar notch e barra.
- **Altura mínima**: assegurar que a área de scroll possa acomodar barra inferior sem sobrepor conteúdo (margem inferior de ~88px aplicada nos containers scroll, como `pb-24` em muitos `main` ⁹⁶). Sim, notamos várias vezes `pb-24` (96px) no main content, provavelmente para dar espaço para FAB e navbar ⁹⁶ ⁹¹. Isso deve estar no guia – sempre incluir padding-bottom suficiente em conteúdos scrolláveis.
- **Elementos sticky e z-index**: citar que headers são sticky top com z-index alto, footers fixed bottom, section titles sticky com top calculado (ex.: 118px) ¹⁰⁴, e que essas posições talvez precisem ajuste se header height mudar – mas mantendo consistência entre telas.
- **5. Comportamentos e Navegação** – *Guia de UX interativo*: Descrever padrões de fluxo:

- Navegação principal via barra inferior (como funciona, quais seções existem).
- Navegação interna: telas push (ex.: abrir detalhes do cliente) devem animar da direita e ter botão de voltar no header. Definir padrão de animação e presença de botão back.
- Gestos de swipe: em seções horizontais (agenda dia/semana/mês, etapas) e talvez swipe-back para voltar (se suportado no framework).
- Estados de interação: reforçar a presença de feedback (hover/active) e listar os tempos/ animações padronizados (p. ex., "todas transições usam 250ms ease, classes Tailwind transition e duration-200/duration-300").
- Comportamentos de componentes: ex.: toggle de local abre/fecha seção de endereço com transição de altura ¹²⁵ – se houver outros acordeões, seguir o mesmo esquema (classe .address-open/closed por ex.). Documentar a existência de animações CSS custom como essa e a fadeIn keyframe ¹²⁰, para reutilização.
- *Estados especiais*: login off-line/erro? (Não aparece nas telas, mas se tiver, padronizar modal ou indicação).
- *Acessibilidade de navegação*: garantir que elementos possam ser focados (se for web), ordem lógica, etc. – isto pode entrar aqui ou na seção de acessibilidade.
- **6. Acessibilidade e Boas Práticas** – *Recomendações transversais*: Incluir diretrizes para garantir que o app seja usável por todos:
 - **Contraste de cores**: Assegurar que textos tenham contraste mínimo. Expor possíveis ajustes – p.ex., o verde estúdio claro em fundo branco em texto pequeno (ratio ~4.2:1) está no limite ⁹; talvez recomendar usar verde escuro para textos miúdos em vez do verde médio, ou aumentar fonte. Garantir que cinza usado em texto secundário também mantenha contraste com papel (atualmente #868E96 em #FAF9F6 dá ~3.1:1, abaixo do ideal). Sugestão: escurecer um pouco o texto muted (#6c757d, por ex., daria ~4.5:1) ou usar para textos que não são críticos. Essas decisões devem ser registradas para futuras melhorias.
 - **Tamanho mínimo de fonte**: Evitar fontes menores que 12px para textos longos ou essenciais. Se 10-11px for usado (como nas labels uppercase), deve ser em curto rótulo, alto contraste, e idealmente extrabold como foi feito para legibilidade.
 - **Área de toque**: Conforme identificado, garantir que elementos clicáveis tenham pelo menos ~44px de área útil. No design atual, alguns chips de filtro são menores (~30px altura ¹⁴²). A documentação deve chamar atenção para isso – se possível, aumentar padding ou espaçamento nesses chips ou justificar por que abriu exceção (talvez por serem muitos e scrolláveis). Em geral, padronizar 40px como altura mínima de botões e itens de lista (como já ocorre nas listagens de clientes e agendamentos).
 - **Semântica e navegação por leitor de tela**: Instruir o uso correto de tags (ex.: usar <button> para itens clicáveis como feito corretamente nas listas ⁸³, usar <label> associados a inputs ⁶, etc.). Garantir que ícones decorativos tenham aria-hidden ou títulos apropriados. Por exemplo, o ícone de busca dentro do input é decorativo e já está implementado apenas via CSS, o input ainda possui placeholder textual – isso é bom. Continuar assim e documentar.
 - **Feedback de foco**: Se a aplicação web ou híbrida permitir navegação por teclado, seria bom definir estilos de focus visíveis (Tailwind focus:ring já indica, ex.: inputs com ring verde ao focar ⁵²). Garantir que botões também tenham algum outline ou efeito no focus (atualmente talvez não definido, pode adicionar).
 - **Boas práticas gerais**: Incluir tópicos como: manter consistência (não reinventar estilo em cada módulo), testar em aparelhos reais diversos (iOS/Android), otimizar assets (já praticamente não há imagens pesadas, o logo e ícones são leves), e internacionalização se pertinente (ex.: strings em pt-BR já estão, mas se planeja outros idiomas, ver tamanho de textos nos botões etc.).

Estruturar a documentação dessa forma cria um **Design System robusto**. Com pastas dedicadas a cada assunto, fica fácil para a equipe encontrar respostas: por exemplo, se um desenvolvedor tiver dúvida de que cor usar para um texto de aviso, consulta “Paleta de cores”; se for criar um novo formulário, consulta “Componentes -> Campos e Inputs” e “Layout e Espaçamento” para seguir padding padrão; se vai implementar uma nova funcionalidade com etapas, consulta “Comportamentos e Navegação” para usar o mesmo estilo de etapas existente.

Em conclusão, a auditoria confirma que a base visual do *Estúdio Corpo & Alma* é **bem consistente e sofisticada**, necessitando apenas pequenos ajustes e, principalmente, formalização em um guia. Seguindo as recomendações acima, o sistema terá uma aparência unificada em produção mobile, mantendo a **elegância suave** proposta – que equilibra clareza (tipografia legível, espaçamentos), sofisticação (cores sóbrias, serif nos títulos, blur/transparência moderno) e organização (estrutura repetível de layouts), sem poluir com excessos visuais. A documentação proposta garantirá que esse estilo seja **mantido e escalado** no futuro, entregando uma experiência coesa e refinada aos usuários finais. [12](#) [13](#)

[1](#) [26](#) [27](#) [102](#) [103](#) [108](#) [109](#) [113](#) [121](#) Visão da Nova Tela de Atendimento.htm

file:///file_000000002ac4720e8fde6a64eb8bc04e

[2](#) [4](#) [11](#) [13](#) [23](#) [31](#) [47](#) [89](#) [98](#) [104](#) [122](#) [123](#) [127](#) [132](#) [133](#) [134](#) [140](#) Visão da Nova Tela de Novo Cliente.htm

file:///file_000000008fc8720e8bc2364090770764

[3](#) [12](#) [14](#) [15](#) [18](#) [19](#) [20](#) [28](#) [30](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [40](#) [41](#) [42](#) [43](#) [44](#) [45](#) [50](#) [51](#) [59](#) [60](#) [61](#) [62](#) [63](#)

[64](#) [66](#) [67](#) [74](#) [75](#) [77](#) [78](#) [79](#) [80](#) [85](#) [87](#) [88](#) [93](#) [96](#) [97](#) [106](#) [107](#) [110](#) [111](#) [114](#) [115](#) [118](#) [120](#) [124](#) [126](#) [128](#) [129](#) [135](#)

[136](#) [138](#) [139](#) Visão da Nova Tela Agenda 2.htm

file:///file_0000000032e8720e9b5ec8ac22a7aab9

[5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [16](#) [17](#) [24](#) [46](#) [48](#) [52](#) [54](#) [65](#) [70](#) [71](#) [72](#) [73](#) [76](#) [92](#) [125](#) [130](#) [131](#) Visão da Nova Tela de Fomulário Agendamento Interno.htm

file:///file_000000000b10720eb16d55105aed680b

[21](#) [22](#) [25](#) [29](#) [39](#) [49](#) [53](#) [55](#) [56](#) [57](#) [58](#) [68](#) [69](#) [81](#) [82](#) [83](#) [84](#) [86](#) [90](#) [91](#) [94](#) [95](#) [99](#) [100](#) [101](#) [105](#) [112](#) [116](#)

[117](#) [119](#) [137](#) [141](#) [142](#) Visão da Nova Telas de Lista de Clientes e Detalhes de Clientes.htm

file:///file_0000000084f4720e8faadf2d5732f3c1