

ft_transcendence

Documento de apoio: visão dos módulos escolhidos e explicação do que é cada um.

Visão geral do projeto

O objetivo principal é captar e qualificar leads por campanhas de e-mail e levar o usuário a uma landing page com formulário (core do negócio). A plataforma também terá área logada para clientes finais (acesso a produtos adquiridos e chat) e área administrativa para o cliente (dashboards e gestão de campanhas).

Resumo dos módulos e pontuação

Categoria	Módulo	Tipo	Pontos
IV.1 Web	Framework de frontend + backend	Major	2
IV.3 Usuários	RBAC (permissões avançadas)	Major	2
IV.3 Usuários	Sistema de organizações	Major	2
V.7 DevOps	Monitoramento com Prometheus + Grafana	Major	2
IV.8 Dados e Análises	Dashboard avançado de análises	Major	2
IV.1 Web	ORM	Minor	1
IV.1 Web	Design System customizado	Minor	1
IV.1 Web	Busca avançada (filtros, ordenação, paginação)	Minor	1
IV.3 Usuários	OAuth 2.0 (autenticação remota)	Minor	1
IV.8 Dados e Análises	Conformidade GDPR	Minor	1
V.1 Web	Sistema de notificações	Minor	1
V.7 DevOps	Health check + status + backups	Minor	1
TOTAL:			17

Detalhamento por módulo

A seguir, cada módulo selecionado é descrito com definição, intenção e aplicação direta no projeto.

1. Framework de frontend + backend (Major - 2 pontos)

Categoria: IV.1 Web

Usar um framework no frontend e um no backend (ou um framework full-stack usando recursos de frontend e backend). O framework define estrutura, roteamento, ciclo de vida, organização do código e padrões de desenvolvimento.

- Frontend: páginas públicas, formulário, área logada, dashboards.
- Backend: autenticação, RBAC, organizações, APIs, integrações e persistência.
- Critério de avaliação: não basta 'apenas usar'; deve estar integrado ao fluxo principal do produto.

2. RBAC (permissões avançadas) (Major - 2 pontos)

Categoria: IV.3 Usuários

RBAC (Role-Based Access Control) é um modelo de autorização onde permissões são atribuídas a papéis (roles), e usuários recebem papéis. Define com clareza 'quem pode fazer o quê' no sistema.

- Papéis típicos: admin, consultor, cliente_final.
- Aplicar no backend (proteção real de endpoints) e refletir no frontend (UI condicional).
- Evitar 'if espalhado' e garantir auditoria de ações sensíveis.

3. Sistema de organizações (Major - 2 pontos)

Categoria: IV.3 Usuários

Organizações estruturam o produto como multi-tenant: usuários e dados pertencem a uma organização (por exemplo, uma corretora). Permite isolar campanhas, contatos e métricas por organização, reduzindo risco de vazamento de dados.

- CRUD de organizações e gestão de membros.
- Isolamento de dados por organization_id em consultas e políticas de acesso.
- Base para escalar o produto para múltiplos clientes no futuro.

4. Monitoramento com Prometheus + Grafana (Major - 2 pontos)

Categoria: IV.7 DevOps

Monitoramento por métricas (não confundir com logs). Prometheus coleta métricas de serviços e infraestrutura; Grafana exibe dashboards e alertas. O foco é saúde operacional e detecção precoce de incidentes.

- Métricas úteis: latência, taxa de erro, throughput, uso de CPU/memória, conexões ativas, jobs assíncronos.
- Dashboards no Grafana para API, banco e serviços críticos.
- Avaliação: demonstrar métricas reais, não apenas containers ligados.

5. Dashboard avançado de análises (Major - 2 pontos)

Categoria: IV.8 Dados e Análises

Dashboard analítico com gráficos interativos, filtros por período, filtros customizáveis e exportação. Transforma eventos de campanha (sent/open/click/bounce/unsubscribe) em indicadores úteis para tomada de decisão.

- Gráficos: séries temporais (linha), comparativos (barra), distribuição (pizza ou equivalente).
- Filtros: data, campanha, segmento, canal; paginação em listagens associadas.
- Exportação: ao menos CSV e, se possível, PDF.

6. ORM (Minor - 1 ponto)

Categoria: IV.1 Web

ORM (Object-Relational Mapping) mapeia tabelas e relacionamentos para modelos no código, padronizando queries, migrações e validações.

- Entidades típicas: User, Organization, Campaign, Contact, Message, Event, Unsubscribe.

- Benefícios: consistência, migrações versionadas, menos SQL duplicado.

- Cuidado: evitar N+1, escolher índices e controlar eager/lazy loading.

7. Design System customizado (Minor - 1 ponto)

Categoria: IV.1 Web

Design system é um conjunto de componentes reutilizáveis e padrões visuais (cores, tipografia, ícones) que garantem consistência e velocidade de desenvolvimento.

- Requisito: pelo menos 10 componentes reutilizáveis (ex.: Button, Input, Select, Modal, Table, Badge, Card, Toast, Pagination, Loader).

- Documentar uso e variações (tamanhos/estados/feedback).

- Ganhos: UI consistente, menos retrabalho e bugs de interface.

8. Busca avançada (filtros, ordenação, paginação) (Minor - 1 ponto)

Categoria: IV.1 Web

Busca avançada permite explorar dados de forma eficiente em listas grandes, com filtros combináveis, ordenação e paginação.

- Aplicações: contatos, campanhas, leads e eventos.

- Boas práticas: filtros no backend, paginação por cursor quando necessário, índices em colunas filtradas.

- UX: manter estado dos filtros e permitir reset rápido.

9. OAuth 2.0 (autenticação remota) (Minor - 1 ponto)

Categoria: IV.3 Usuários

OAuth 2.0 permite login via provedores externos (Google, GitHub, 42 etc.), reduzindo atrito e evitando armazenamento de senha local para esses usuários.

- Fluxo: redirect para provedor, callback, troca por token, criação/associação de conta local.

- Cuidado: validar state/nonce, controlar sessão e tratar usuários já existentes.

- Complementa o login tradicional; não substitui autorização (RBAC).

10. Conformidade GDPR (Minor - 1 ponto)

Categoria: IV.8 Dados e Análises

Conjunto de funcionalidades para transparência e controle de dados pessoais: exportar dados do usuário, solicitar remoção e confirmar operações.

- Exportar dados em formato legível (ex.: JSON/CSV).

- Deleção com confirmação e registro/auditoria do pedido.

- Cuidado: definir deleção vs anonimização para manter integridade analítica.

11. Sistema de notificações (Minor - 1 ponto)

Categoria: IV.1 Web

Notificações in-app informam o usuário sobre ações importantes (criação, atualização, deleção) e eventos relevantes (ex.: novo lead, falha de envio).

- Pode ser combinado com tempo real (WebSockets) quando necessário.

- Implementar ao menos: toast imediato + central de notificações persistente.

- Registrar status (lida/não lida) e permitir ação rápida quando aplicável.

12. Health check + status + backups (Minor - 1 ponto)

Categoria: IV.7 DevOps

Conjunto operacional mínimo: endpoints de saúde, página de status e procedimentos de backup/recuperação.

- Health/ready: verificar dependências (banco, serviços essenciais).
- Status page: visão simples do estado dos componentes.
- Backups: rotina automatizada e procedimento documentado de restore (DR).

Ordem sugerida de implementação

1 Base: framework, ORM e modelo de dados (User/Organization/Campaign/Event).

2 Segurança: autenticação, OAuth (opcional no começo), RBAC e isolamento por organização.

3 UX: design system e páginas principais (landing, área logada, admin).

4 Dados: busca avançada nas listas e dashboard analítico (gráficos, filtros e exportação).

5 Operação: notificações, health/status/backups e, por último, Prometheus + Grafana com métricas reais.

Roteiro curto de demo (8 passos)

1 Login (local ou OAuth) como admin.

2 Criar/selecionar uma organização e convidar um usuário.

3 Validar RBAC: acessar uma página proibida com role diferente (bloqueio no backend).

4 Criar campanha e importar/selecionar contatos (se aplicável ao seu MVP).

5 Listar campanhas/contatos com busca avançada (filtros/ordenação/paginação).

6 Abrir dashboard analítico, aplicar filtro de período e exportar CSV/PDF.

7 Executar GDPR: exportar dados do usuário e solicitar deleção/anonimização.

8 Mostrar status/health e dashboards do Grafana com métricas reais.