

# Informação Completa da APP Norbidel

Gerado em: 05/05/2026, 12:08:25

Este PDF consolida a informação disponível sobre a APP Norbidel: objetivo funcional, stack técnica, sprint, backlog, base de dados, prioridades, autenticação/utilizadores/clientes e checklists de caldeiras a gás/biomassa.

## Resumo executivo

A APP Norbidel é uma aplicação interna para apoiar a administração e controlo operacional da empresa, com foco inicial no MVP web responsivo.

Prioridades atuais confirmadas:

1. Auth real: login, sessão, proteção de rotas e distinção por perfil.
2. Seed inicial: perfis base, permissões base e utilizador admin inicial.
3. CRUD real de utilizadores: criar, editar, ativar/desativar e associar perfil.
4. CRUD real de clientes: criar, editar, listar e pesquisar.
5. Regras mínimas de permissões: técnico vs admin, escritório vs terreno.

O que não é prioridade agora: app mobile nativa, design refinado, módulos completos de RH, contratos/obras/faturação. O módulo funcional seguinte recomendado é pedidos/entradas.

## Documento Mestre Funcional MVP

Fonte: Documento\_Mestre\_Funcional\_MVP\_Norbidel.md

## Documento Mestre Funcional do MVP

### Sistema Interno de Gestão Operacional, Norbidel

Versão: v3

Data: Abril 2026

---

#### 1. Objetivo do sistema

Criar uma app interna para a Norbidel que organize os processos, reduza esquecimentos, centralize informação, simplifique o trabalho diário e ligue todos os setores num fluxo contínuo, controlado e auditável.

A app deve servir para:

- centralizar a relação com o cliente
- registar e acompanhar processos operacionais
- apoiar engenharia, administração, armazém, faturação e técnicos
- planejar trabalho
- apoiar execução no terreno
- garantir auditoria
- garantir preparação correta da faturação
- reduzir perdas de informação entre setores

Regras base:

- a app é para uso interno da empresa
- a empresa trabalha para indústria
- a app integra com software externo de faturação, não o substitui

---

#### 2. Problemas que o sistema tem de resolver

Os principais problemas atuais identificados são:

- esquecimentos
- processos informais
- dependência excessiva de Excel e lógica distribuída por ficheiros
- pedidos recebidos por telefone sem registo consistente

- trabalho feito mas não faturado
- material usado mas não registado
- folhas de ponto mal preenchidas
- falta de ligação entre setores
- processos que ficam parados sem dono claro
- dificuldade em saber rapidamente o estado real de cada processo
- dependência de conhecimento tácito de poucas pessoas

---

### 3. Objetivo operacional do MVP

O MVP deve garantir que a empresa consegue:

- saber o que entrou
- criar logo o processo correto quando o caso é claro
- não perder contactos/pedidos quando o caso ainda não está claro
- saber o que está por fazer
- saber o que está atrasado
- saber o que está a ser executado
- saber o que foi executado
- saber o que falta auditar
- saber o que falta faturar
- saber o que está por expedir
- saber que material foi usado
- saber quem tratou cada etapa
- preparar corretamente a informação para o software externo de faturação

---

### 4. Princípios base do sistema

1. Nada importante avança sem registo
2. Todo processo tem estado
3. Todo processo tem responsável
4. Todo bloqueio importante deve ser visível
5. Processos incompletos não avançam
6. Só admin pode desbloquear exceções
7. Tudo importante deve ficar rastreado
8. A app deve ser simples para usar, sobretudo para técnicos
9. Os estados não devem ser escolhidos livremente, devem resultar de ações reais
10. O sistema deve privilegiar criação direta do tipo de processo correto
11. Só quando o caso não for claro deve existir registo preliminar para posterior tratamento

---

### 5. Tipos principais de processo

Os tipos principais do sistema são:

- orçamento
- contrato
- obra
- assistência
- venda comercial

### 5.1 Natureza de cada processo

- **Orçamento / proposta**: comercial
- **Contrato**: comercial e técnico
- **Obra**: técnica
- **Assistência**: técnica
- **Venda comercial**: comercial

### 5.2 Diferença entre venda comercial e contrato

#### Venda comercial

- venda de material, equipamento ou artigo
- pode existir com ou sem orçamento prévio
- não inclui mão de obra

- pode incluir portes

#### #### Contrato

- é uma intervenção previamente orçamentada
- vem sempre de um orçamento aceite
- tem componente comercial e técnica

---

## 6. Entrada dos processos no sistema

### ### 6.1 Regra principal

O contacto/pedido do cliente é a origem do negócio, mas não tem de existir sempre como entidade autónoma no sistema.

Na prática, quando alguém com contexto suficiente recebe o pedido do cliente, deve criar logo no sistema o tipo de processo correto:

- orçamento
- obra
- assistência
- venda comercial

Ou seja, o sistema deve privilegiar **entrada direta no processo final**, sem obrigar a uma etapa intermédia de pedido.

### ### 6.2 Regra excepcional

Se quem recebe o contacto:

- não tiver qualificação suficiente
- ou não conseguir decidir o tipo de processo

então deve existir um registo preliminar simples, para posterior tratamento por engenharia ou administração.

Esse registo preliminar é excepcional, não o fluxo principal.

### ### 6.3 Regra de decisão

A decisão sobre que tipo de processo criar é:

- humana
- feita por engenharia ou administração
- baseada sobretudo no tipo de trabalho

Essa decisão pode acontecer:

- logo na entrada, quando o caso é claro
- ou mais tarde, se tiver havido registo preliminar

### ### 6.4 Fluxos principais

#### #### Fluxo A, com orçamento

**\*\* Orçamento !' Contrato \*\***

Usado quando o trabalho precisa de proposta prévia.

#### #### Fluxo B, sem orçamento prévio

**\*\*Obra\*\***

ou

**\*\*Assistência\*\***

Usado quando não é necessário orçamento antes.

#### #### Fluxo C, comercial

**\*\*Venda comercial\*\***

ou

**\*\* Orçamento !' Venda comercial \*\***

### ### 6.5 Trabalho extra após contrato

Se houver trabalho extra depois de um contrato:

- deve ser criada uma **obra** para esses trabalhos extra

---

## 7. Perfis de utilizador

### ### 7.1 Admin

- permissão total
- pode tudo

- pode desbloquear exceções
- vê informação de gestão e controlo

### ### 7.2 Administração

- cria processos
- pode classificar casos quando aplicável
- acompanha circuitos administrativos
- acompanha processos por tratar, por expedir, por faturar

### ### 7.3 Engenharia

- cria processos
- classifica o tipo de processo quando necessário
- acompanha execução
- lança ou corrige informação operacional quando aplicável
- audita
- envia para faturação

### ### 7.4 Compras / Armazém

- valida disponibilidade
- separa material para expedição
- confirma rotura
- trata pedidos ligados a material
- reencaminha para compras quando o armazém não consegue satisfazer

### ### 7.5 Faturação / RH

- acompanha processos em faturação
- recebe informação preparada para faturar
- acompanha alertas administrativos e documentais

### ### 7.6 Técnicos

- usam tablet da empresa
- veem processos técnicos atribuídos
- abrem e fecham sessões
- registam horas, km, material, fotos, checklist, assinatura e observações
- não precisam de ver preços nem custos
- trabalham com interface simples e rápida

### ### 7.7 Modelo de permissões

No MVP, as permissões devem ser **\*\*simples por perfil\*\***.

---

## 8. CRM e ficha do cliente

A app deve ter um CRM forte, não apenas uma lista de clientes.

### ### 8.1 Dados principais do cliente

Cada cliente deve ter:

- designação / nome
- NIF
- morada
- email
- telefone
- contactos
- equipamentos associados
- anexos
- notas
- histórico de processos

### ### 8.2 Regra de moradas

No MVP, **\*\*uma morada basta\*\***.

Essa morada serve como referência base e pode ser usada como local por defeito dos processos.

### ### 8.3 Timeline única do cliente

A ficha do cliente deve ter uma timeline cronológica única com:

- eventos automáticos do sistema
- notas manuais da equipa
- orçamentos

- contratos
- obras
- assistências
- vendas comerciais
- anexos e documentos

#### ### 8.4 Regras adicionais

- o código de identificação já existente na base de dados atual deve ser aproveitado
- o sistema deve suportar anexos de vários tipos
- a morada do cliente deve poder pré-preencher o local do processo

---

## 9. Dados mínimos à criação de um processo

Devem existir logo à criação:

- cliente
- contacto
- descrição
- tipo de trabalho
- equipamento, se existir
- urgência, quando aplicável
- referência interna
- anexos
- nome de quem fez o pedido
- email ou contacto associado

#### ### Regras adicionais

- o local não precisa de ser pedido sempre, porque por defeito é a morada do cliente
- o processo pode não ter equipamento associado
- se houver equipamento, deve ser possível seleccionar existente ou criar novo
- no MVP, no equipamento basta a designação

---

## 10. Orçamentos, propostas e contratos

#### ### 10.1 Dados importantes no orçamento

No orçamento/proposta é importante guardar:

- cliente
- data
- referência
- descrição
- preço
- validade
- condições de pagamento
- garantia
- exclusões
- prazo

#### ### 10.2 Condições comerciais

As condições comerciais devem ser um misto de:

- modelos frequentes
- edição manual caso a caso

#### ### 10.3 Estados de orçamento / proposta / contrato

- rascunho
- proposta
- contrato
- arquivado

#### ### 10.4 Regras de transição

- até ser enviado ao cliente é rascunho
- depois de enviado passa a proposta
- se aceite passa a contrato
- se rejeitado ou expirar a validade passa a arquivado

### ### 10.5 Regra de contrato

O contrato:

- não acrescenta dados obrigatórios novos em relação ao orçamento
- representa a aceitação da proposta
- fica fechado após conversão a contrato

---

## **11. Obras e assistências**

Obras e assistências são processos técnicos com lógica operacional semelhante.

### ### 11.1 Dados operacionais principais

Devem registrar:

- horas
- km
- material realmente usado

### ### 11.2 Ligação a equipamento

- o processo pode ficar ligado a um equipamento
- essa ligação não é obrigatória

### ### 11.3 Quem cria

Obras e assistências podem ser criadas por:

- administração
- engenharia

### ### 11.4 Estados de obra e assistência

- aberto
- em curso
- em auditoria
- em faturação
- concluído
- arquivado

### ### 11.5 Regras de transição

- primeira sessão iniciada passa o processo a em curso
- quando técnico ou engenharia dá por concluído, passa a em auditoria
- depois da auditoria, passa a em faturação
- depois de faturado, passa a concluído
- depois segue para arquivo

---

## **12. Venda comercial**

Venda comercial é um processo distinto.

### ### 12.1 Definição

Venda comercial é:

- venda de material, equipamento ou artigo
- com ou sem orçamento prévio
- sem mão de obra
- podendo incluir portes

### ### 12.2 Dados principais

Na venda comercial interessa guardar:

- artigo, material ou equipamento
- quantidade
- valor
- portes
- prazo
- observações

### ### 12.3 Estados de venda comercial

- pedido
- venda comercial
- separação para expedição

- expedido
- em faturação
- arquivado

### ### 12.4 Regras do armazém

Na venda comercial, o armazém:

- separa
- confirma disponibilidade
- confirma rotura, quando aplicável

### ### 12.5 Stock no MVP

No MVP:

- basta validar stock disponível
- não é obrigatório reservar stock formalmente

---

## **13. Sessões de trabalho**

O modelo de sessões é central em obra e assistência.

### ### 13.1 Regras principais

- as sessões são abertas e fechadas diretamente no sistema
- não existe estado de rascunho da sessão
- pode haver várias sessões no mesmo processo
- uma sessão pode ter vários técnicos

### ### 13.2 Quem lança sessões

- técnico
- engenharia

### ### 13.3 Dados da sessão

Cada sessão deve poder guardar:

- data
- técnico ou técnicos
- hora de início
- hora de fim
- duração
- km
- observações
- material usado

### ### 13.4 Regras adicionais

- os km são por sessão
- o material tem origem na sessão e consolida no processo
- o material executado significa material realmente usado

---

## **14. Material, armazém e stock**

### ### 14.1 Registo de material

O registo de material deve aceitar:

- catálogo
- texto livre

### ### 14.2 Material urgente

Fluxo de material urgente:

- técnico pede
- vai para armazém
- se armazém não satisfizer, reencaminha para compras

### ### 14.3 Stock

No MVP:

- existe stock central
- o stock cobre materiais e equipamentos
- deve ser possível validar disponibilidade

### ### 14.4 Número de série

O sistema deve suportar número de série quando existir, sobretudo em equipamentos relevantes.

---

## 15. Módulo técnico

O módulo técnico é uma área crítica do MVP.

### ### 15.1 Requisitos de interface

- uso em tablet da empresa
- interface simples
- cards grandes e visuais
- rapidez de utilização
- preparado para contexto real de campo

### ### 15.2 O técnico deve conseguir

- ver processos do dia
- ver processos técnicos abertos
- abrir detalhe do processo
- abrir e fechar sessão
- registar horas
- registar km
- registar material
- tirar fotos
- preencher checklist
- recolher assinatura
- escrever observações

### ### 15.3 Regra de horas

- preferência por registo automático com iniciar e terminar
- correções manuais existem, mas como exceção controlada

### ### 15.4 Histórico visível para o técnico

No detalhe do processo técnico, o técnico deve ver:

- cliente
- local
- equipamento, se existir
- histórico anterior relevante
- recomendações preventivas destacadas

---

## 16. Checklist, fotos e assinatura

### ### 16.1 Checklist

A checklist entra no MVP.

Regras:

- checklist por tipo de equipamento, quando aplicável
- respostas simples tipo sim ou não
- usada na execução e auditoria
- não é obrigatória em todos os cenários

### ### 16.2 Fotos

As fotos entram no MVP.

Regra:

- mínimo de 1 foto por processo técnico, por defeito

### ### 16.3 Assinatura

A assinatura do responsável do cliente entra no MVP.

Regra:

- obrigatória por defeito
- pode existir exceção controlada administrativamente

---

## 17. Auditoria

A auditoria é uma etapa obrigatória dos processos técnicos.

### ### 17.1 Quem audita

- quem abriu o processo

### ### 17.2 O que é validado

Na auditoria valida-se:

- horas
- km
- material
- coerência técnica
- pronto para faturar
- fotos, quando aplicável
- assinatura, quando aplicável
- checklist, quando aplicável

### ### 17.3 Resultado

- se houver problema, o processo deve voltar para correção
- se estiver correto, segue para faturação

---

## 18. Faturação

A app não substitui o software externo de faturação.

### ### 18.1 Função da app neste ponto

A app deve:

- preparar a faturação
- enviar a ordem para o software externo de faturação
- dar visibilidade do que está pronto a faturar

### ### 18.2 Informação a enviar

Deve ser enviada pelo menos:

- cliente
- nossa referência
- preços e valores
- detalhe do que está a ser faturado

Idealmente deve incluir:

- ordem
- resumo
- linhas detalhadas
- materiais
- horas
- km
- outros elementos faturáveis

---

## 19. Dashboard, listas e pesquisa

### ### 19.1 Dashboard

Na dashboard, o mais importante é ver:

- o que está atrasado
- o que está por auditar
- o que está por faturar
- o que está por expedir
- o que está por decidir

### ### 19.2 Listas específicas por tipo

Devem existir listas e filtros específicos para:

- orçamentos
- contratos
- obras

- assistências
- vendas comerciais

### ### 19.3 Distinção visual

É importante distinguir visualmente:

- obra
- assistência
- venda comercial
- proposta / contrato

Faz sentido reaproveitar a lógica atual de capas com cores diferentes.

### ### 19.4 Pesquisa global

A pesquisa global deve suportar pesquisa por:

- cliente
- NIF
- referência
- equipamento
- texto livre

---

## 20. Anexos e notas internas

### ### 20.1 Anexos

Os anexos devem existir em todos os tipos de processo:

- orçamento
- contrato
- obra
- assistência
- venda comercial
- registo preliminar, quando existir

### ### 20.2 Notas internas

Devem existir notas internas livres por processo.

Regras:

- as notas podem ser editadas
- a edição deve ficar registada

---

## 21. Alertas

Os alertas no MVP devem aparecer dentro da app.

Alertas principais:

- propostas a expirar
- processos parados
- por auditar
- por faturar
- por expedir
- stock baixo
- casos pendentes de decisão, quando existirem

---

## 22. Regras críticas de bloqueio

### ### 22.1 Um processo técnico não avança se faltar

- horas
- km
- material obrigatório
- assinatura, quando obrigatória
- foto mínima obrigatória
- checklist obrigatória, quando aplicável
- campos obrigatórios

### ### 22.2 Exceção

- só admin pode desbloquear
- deve ficar registado quem desbloqueou, quando e porquê

---

## 23. Módulos principais do MVP

1. Dashboard
2. CRM / Clientes
3. Orçamentos
4. Contratos
5. Obras
6. Assistências
7. Vendas comerciais
8. Planeamento
9. Técnico / Sessões
10. Materiais / Armazém
11. Auditoria
12. Faturação
13. Administração

### ### Nota

Pode existir um mecanismo de registo preliminar / rascunho para casos indefinidos, mas isso não é o fluxo principal.

---

## 24. Lista resumida de ecrãs críticos do MVP

- Login
- Dashboard
- Lista de clientes
- Ficha do cliente
- Criar orçamento
- Ficha de orçamento
- Lista de contratos
- Ficha de contrato
- Lista de obras
- Ficha de obra
- Lista de assistências
- Ficha de assistência
- Lista de vendas comerciais
- Ficha de venda comercial
- Planeamento diário
- Planeamento semanal
- Home do técnico
- Detalhe do processo técnico
- Registo de sessão
- Registo de km
- Registo de material
- Fotos
- Checklist
- Assinatura
- Lista por auditar
- Ficha de auditoria
- Lista por faturar
- Gestão de utilizadores

---

## 25. Modelo de dados principal, resumo

As principais entidades do sistema são:

- User
- Cliente

- ContactoCliente
- TimelineCliente
- Equipamento
- Orcamento
- OrcamentoLinha
- Contrato
- Obra
- Assistencia
- VendaComercial
- SessaoTrabalho
- RegistoKM
- ProcessoMaterial
- FotoProcesso
- AssinaturaProcesso
- ChecklistModelo
- ChecklistResposta
- Material
- PedidoMaterial
- PedidoMaterialItem
- Fornecedor
- AuditoriaProcesso
- PreparacaoFaturacao
- AnexoProcesso
- NotaInternaProcesso
- Alerta
- BloqueioProcesso
- ExcecaoAdmin
- LogSistema
- RegistoPreliminar, quando aplicável

Notas adicionais sobre dados:

- deve existir suporte para códigos já existentes na base de dados atual
- o sistema deve suportar anexos de vários tipos
- o stock deve cobrir materiais e equipamentos
- o número de série deve ser tratado como dado relevante quando existir

---

## 26. Decisões fechadas até agora

- software é só para uso interno
- a operação é industrial
- existem 5 tipos principais de processo: orçamento, contrato, obra, assistência e venda comercial
- o pedido do cliente é origem do negócio, mas não é etapa obrigatória no sistema
- o normal é criar diretamente o tipo de processo correto
- só em casos indefinidos deve existir registo preliminar
- venda comercial não inclui mão de obra
- contrato vem sempre de orçamento aceite
- contrato é comercial e técnico
- obra e assistência são técnicas
- obra é uma intervenção contabilizada a posteriori
- trabalho extra ao contrato gera obra autónoma
- processos podem existir sem equipamento associado
- na ficha do equipamento, no MVP, basta designação
- uma morada de cliente basta no MVP
- a morada do cliente pode pré-preencher o local do processo
- condições comerciais são estruturadas e editáveis
- estados devem resultar de ações reais e não de seleção livre
- obras e assistências usam sessões
- sessão pode ter vários técnicos
- pode haver várias sessões por processo
- km são por sessão
- material executado é material realmente usado

- material nasce na sessão e consolida no processo
- auditoria é feita por quem abriu o processo
- auditoria valida horas, km, material e coerência técnica
- o sistema prepara e envia informação para faturação externa
- venda comercial tem fluxo próprio com expedição
- stock disponível deve ser validado no MVP
- anexos existem em todos os tipos de processo
- notas internas podem ser editadas com rasto
- pesquisa global é obrigatória
- listas por tipo são obrigatórias
- distinção visual por tipo é importante
- permissões simples por perfil no MVP

---

## 27. Próximos passos recomendados

1. Separar o que é MVP do que fica para fase 2
2. Atualizar a estrutura base de dados com esta versão funcional
3. Produzir backlog detalhado por módulo
4. Desenhar wireframes e fluxos por perfil
5. Mapear estados e transições por processo
6. Definir integração concreta com o software de faturação

## Estrutura Base de Dados MVP

Fonte: Estrutura\_Base\_Dados\_MVP\_Norbidel.md

## Estrutura Base de Dados do MVP

### Sistema Interno de Gestão Operacional, Norbidel

Versão: v1

Data: Abril 2026

---

### 1. Objetivo deste documento

Este documento traduz o Documento Mestre Funcional do MVP para uma estrutura lógica de base de dados.

O objetivo nesta fase não é fechar ainda todos os tipos SQL finais, índices e constraints técnicas de implementação, mas sim definir:

- entidades principais
- relações entre entidades
- campos essenciais
- regras de negócio com impacto direto na estrutura dos dados
- base suficiente para começar modelação técnica e desenvolvimento

---

### 2. Princípios de modelação

1. Toda a informação operacional importante deve ficar rastreada
2. Os estados dos processos devem ser explícitos
3. O histórico deve ser preservado
4. O sistema deve suportar auditoria interna
5. O modelo deve ser simples o suficiente para arrancar rápido, mas preparado para crescer
6. O MVP privilegia clareza operacional sobre sofisticação excessiva

---

### 3. Grandes blocos do modelo

O modelo pode ser dividido em 10 áreas:

1. Utilizadores e permissões
2. RH e colaboradores
3. CRM e clientes
4. Pedidos e anexos
5. Orçamentos e contratos
6. Intervenções e planeamento
7. Execução técnica
8. Materiais, stock e fornecedores
9. Auditoria e faturação
10. Alertas, bloqueios e rastreabilidade

---

## 4. Entidades principais

### ### 4.1 User

Representa cada utilizador interno do sistema.

Campos principais:

- id
- nome
- email
- telefone
- username
- password\_hash
- perfil\_id
- ativo
- ultimo\_login\_at
- created\_at
- updated\_at

Observações:

- todos os utilizadores são internos
- técnicos, admin, faturação, engenharia e armazém usam a mesma base de utilizadores

### ### 4.2 Perfil

Define o papel principal do utilizador.

Campos principais:

- id
- nome
- codigo
- descricao
- ativo

Valores esperados no MVP:

- ADMIN
- FATURACAO\_RH
- ENGENHARIA\_PROJETOS
- COMPRAS\_ARMAZEM
- TECNICO

### ### 4.3 Permissao

Catálogo de permissões específicas.

Campos principais:

- id
- codigo
- nome
- descricao

### ### 4.4 PerfilPermissao

Tabela de ligação entre perfis e permissões.

Campos principais:

- id
- perfil\_id
- permissao\_id

### ### 4.5 Colaborador

Representa o colaborador enquanto entidade de RH.

Campos principais:

- id
- user\_id, nullable
- nome
- telefone
- email
- morada
- nif
- numero\_identificacao
- validade\_identificacao
- numero\_seg\_social
- data\_nascimento
- nacionalidade
- funcao
- data\_entrada
- data\_saida, nullable
- tipo\_contrato
- estado
- salario\_base
- valor\_hora\_extra
- suplementos, nullable
- observacoes
- created\_at
- updated\_at

Regras:

- nem todo o colaborador precisa de ser utilizador ativo da app
- no MVP a organização pode assentar apenas na função, sem equipas formais
- a função é texto livre
- o estado deve suportar pelo menos ativo, baixa\_medica, licenca e ferias
- deve existir estrutura para total mensal de horas extra, mesmo sem detalhe operacional completo

### ### 4.6 ColaboradorDocumento

Documentos associados ao colaborador.

Campos principais:

- id
- colaborador\_id
- tipo\_documento
- titulo
- nome\_ficheiro
- mime\_type
- caminho\_armazenamento
- data\_emissao
- data\_validade
- obrigatorio
- observacoes
- uploaded\_by\_user\_id
- uploaded\_at

Tipos esperados:

- contrato
- certificado\_formacao
- carta\_conducao
- habilitacao
- medicina\_trabalho
- seguranca
- identificacao\_civil
- nif
- seguranca\_social
- outro

Regras:

- por defeito, todos os tipos podem ser obrigatórios exceto carta\_conducao
- carta\_conducao é opcional

### ### 4.7 ColaboradorEquipamento

Regista equipamento atribuído ao colaborador.

Campos principais:

- id
- colaborador\_id
- tipo\_equipamento
- descricao
- data\_entrega
- data\_devolucao
- observacoes
- created\_at
- updated\_at

Tipos esperados:

- telemovel
- tablet
- epi
- ferramenta
- viatura
- outro

Regras:

- no MVP basta controlo por tipo de equipamento atribuído
- viatura deve poder migrar mais tarde para módulo próprio

### ### 4.8 AlertaRH

Alertas administrativos ligados a RH.

Campos principais:

- id
- colaborador\_id
- colaborador\_documento\_id, nullable
- tipo\_alerta
- titulo
- descricao
- data\_referencia
- severidade
- resolvido
- resolvido\_por\_user\_id, nullable
- resolvido\_at, nullable
- created\_at

Tipos esperados:

- contrato\_a\_terminar
- certificado\_a\_expirar
- documento\_em\_falta
- medicina\_trabalho\_a\_expirar
- validacao\_horas\_extra

Regras:

- alertas RH devem ser visíveis para perfis admin e faturacao\_rh
- alertas de medicina do trabalho devem disparar com 15 dias de antecedência

---

## 5. CRM e clientes

### ### 5.1 Cliente

Representa a entidade cliente.

Campos principais:

- id
- codigo\_interno

- designacao\_social
- nif
- morada
- codigo\_postal
- localidade
- pais
- email
- telefone
- setor\_atividade
- observacoes
- ativo
- created\_at
- updated\_at

Regras:

- deve suportar aproveitamento de códigos já existentes na base de dados atual
- o foco é cliente industrial

### ### 5.2 ContactoCliente

Contactos associados ao cliente.

Campos principais:

- id
- cliente\_id
- nome
- cargo
- telefone
- telemovel
- email
- principal
- ativo
- observacoes

### ### 5.3 ClienteDocumento

Documentos associados ao cliente.

Campos principais:

- id
- cliente\_id
- nome\_ficheiro
- tipo\_documento
- mime\_type
- caminho\_armazenamento
- descricao
- uploaded\_by\_user\_id
- uploaded\_at

Observações:

- deve suportar documentos técnicos, projetos, desenhos e anexos diversos

### ### 5.4 TimelineCliente

Histórico cronológico do cliente.

Campos principais:

- id
- cliente\_id
- tipo\_evento
- titulo
- descricao
- data\_evento
- origem\_tipo
- origem\_id
- criado\_por\_user\_id
- created\_at

Objetivo:

- centralizar histórico comercial e operacional num único feed cronológico

### ### 5.5 Equipamento

Equipamentos associados ao cliente.

Campos principais:

- id
- cliente\_id
- codigo\_interno
- numero\_serie
- designacao
- marca
- modelo
- tipo\_equipamento
- local\_instalacao
- data\_instalacao
- ativo
- observacoes
- created\_at
- updated\_at

Regras:

- o número de série deve existir quando aplicável
- uma intervenção pode existir sem equipamento

---

## 6. Pedidos e entradas

Esta área modela a fila central de pedidos da empresa. Inclui pedidos formais criados por Engenharia/Admin, rascunhos criados por Faturação/RH, anexos, estados finos de pendente, SLA e log de auditoria de transições.

### ### 6.1 Pedido

Representa qualquer entrada formal no sistema, depois de registada e visível na fila de triagem.

Campos principais:

- id
- numero\_pedido
- cliente\_id
- contacto\_cliente\_id
- origem
- descricao
- urgencia
- estado
- sub\_estado
- tipo\_resultado
- resultado\_ref\_tipo
- resultado\_ref\_id
- motivo\_arquivamento
- rascunho\_origem\_id
- recebido\_por\_user\_id
- criado\_por\_user\_id
- triado\_por\_user\_id
- data\_entrada
- data\_triagem
- data\_limite\_sla
- sla\_excedido
- alerta\_sla\_enviado\_em
- observacoes
- created\_at
- updated\_at

Valores sugeridos:

- origem: telefone, email, presencial, interno, outro
- urgencia: alta, normal, baixa
- estado: pendente\_triagem, a\_aguardar\_info\_cliente, a\_aguardar\_decisao\_interna, triado, arquivado
- sub\_estado: opcional, texto livre controlado por enum interno para detalhar motivos (por exemplo: aguarda\_assinatura,

- aguarda\_desenho, aguarda\_aprovacao\_comercial)
- tipo\_resultado: intervencao, orcamento, arquivado
- resultado\_ref\_tipo: intervencao, orcamento, nenhum

#### Observações:

- um Pedido só pode transitar para estado `triado` preenchendo `tipo\_resultado`, `resultado\_ref\_tipo` e `resultado\_ref\_id` (ou `motivo\_arquivamento` se o resultado for arquivado)
- `data\_limite\_sla` é calculado a partir de `data\_entrada` + limite configurado para a urgência
- `sla\_excedido` e `alerta\_sla\_enviado\_em` suportam a sinalização visual da fila e os alertas automáticos
- `rascunho\_origem\_id` liga um Pedido formal ao rascunho que lhe deu origem (quando aplicável)
- apenas users com perfil ADMIN ou ENGENHARIA\_PROJETOS podem criar um Pedido directamente
- o tratamento de alguns cenários depende do tipo de pedido; esta diferenciação fica capturada pelo par (`tipo\_resultado`, `sub\_estado`) e por validações aplicacionais

### ### 6.2 PedidoRascunho

Representa um pedido em rascunho criado por Faturação/RH quando não há Engenharia ou Admin disponível para registar pedido formal.

#### Campos principais:

- id
- numero\_rascunho
- cliente\_id
- contacto\_cliente\_id
- contacto\_texto\_livre
- origem
- urgencia\_percebida
- descricao\_curta
- estado
- motivo\_rejeicao
- pedido\_id
- criado\_por\_user\_id
- validado\_por\_user\_id
- data\_entrada
- data\_validacao
- data\_limite\_sla
- sla\_excedido
- alerta\_sla\_enviado\_em
- observacoes
- created\_at
- updated\_at

#### Valores sugeridos:

- origem: telefone, email, presencial, interno, outro
- urgencia\_percebida: alta, normal, baixa
- estado: pendente\_validacao, validado, rejeitado

#### Observações:

- `cliente\_id` e `contacto\_cliente\_id` são opcionais; se o contacto ainda não existe na base, usar `contacto\_texto\_livre` (nome + telefone + empresa em texto corrido) para não bloquear a criação
- `pedido\_id` fica preenchido quando o rascunho é validado e dá origem a um Pedido formal
- apenas users com perfil FATURACAO\_RH, ENGENHARIA\_PROJETOS ou ADMIN podem criar rascunhos
- apenas users com perfil ENGENHARIA\_PROJETOS ou ADMIN podem validar ou rejeitar rascunhos
- a validação de um rascunho cria um Pedido novo com `rascunho\_origem\_id` apontando para este rascunho
- rejeição exige `motivo\_rejeicao` e notifica automaticamente o criador
- o SLA do rascunho segue os mesmos limites do Pedido (alta 4h, normal 24h, baixa 72h)

### ### 6.3 AnexoPedido

Ficheiros associados a um Pedido ou a um PedidoRascunho.

#### Campos principais:

- id
- pedido\_id
- pedido\_rascunho\_id
- nome\_ficheiro
- tipo\_documento
- mime\_type

- caminho\_armazenamento
- uploaded\_by\_user\_id
- uploaded\_at

Observações:

- exactamente um de `pedido\_id` ou `pedido\_rascunho\_id` deve estar preenchido
- quando um rascunho com anexos é validado e origina um Pedido, os anexos passam a referenciar o Pedido novo e o rascunho deixa de ser a referência principal (mas o histórico do rascunho fica preservado através de PedidoEventoLog)

### ### 6.4 PedidoEventoLog

Log de auditoria de todas as transições de estado e eventos relevantes em Pedidos e PedidoRascunhos. Suporta o requisito de auditoria interna e análise posterior.

Campos principais:

- id
- pedido\_id
- pedido\_rascunho\_id
- tipo\_evento
- estado\_anterior
- estado\_novo
- detalhe
- user\_id
- created\_at

Valores sugeridos para tipo\_evento:

- rascunho\_criado
- rascunho\_validado
- rascunho\_rejeitado
- pedido\_criado
- pedido\_atualizado
- estado\_alterado
- triagem\_concluida
- pedido\_arquivado
- sla\_excedido
- alerta\_enviado

Observações:

- exactamente um de `pedido\_id` ou `pedido\_rascunho\_id` deve estar preenchido
- esta tabela serve como base para futuras métricas operacionais (tempos médios de triagem, taxa de arquivamento, etc.)

### ### 6.5 PedidoSlaConfig

Configuração de SLA por urgência, separada em tabela para permitir afinação futura sem mudar código.

Campos principais:

- id
- urgencia
- limite\_horas
- ativo
- updated\_at
- updated\_by\_user\_id

Valores iniciais sugeridos:

- alta !' 4 horas
- normal !' 24 horas
- baixa !' 72 horas

Observações:

- ambos Pedido e PedidoRascunho usam esta configuração; não existe SLA separado para rascunhos
- alterações a esta tabela só podem ser feitas por ADMIN e ficam auditadas em log global
- `data\_limite\_sla` do Pedido e do PedidoRascunho é calculado no momento de criação e fica persistido, para manter estabilidade mesmo que a tabela seja alterada no futuro

---

## 7. Orçamentos e contratos

### ### 7.1 Orcamento

Representa proposta comercial.

Campos principais:

- id
- numero\_orcamento
- cliente\_id
- pedido\_id
- estado
- data\_criacao
- data\_envio
- data\_resposta
- criado\_por\_user\_id
- observacoes
- created\_at
- updated\_at

Estados esperados:

- em\_preparacao
- enviado
- pendente
- aceite
- recusado

Regra:

- orçamento aceite converte sempre em contrato

### ### 7.2 OrcamentoLinha

Linhas do orçamento.

Campos principais:

- id
- orcamento\_id
- descricao
- quantidade
- unidade
- observacoes
- ordem

### ### 7.3 Contrato

Representa o contrato criado a partir do orçamento aceite.

Campos principais:

- id
- numero\_contrato
- cliente\_id
- orcamento\_id
- tipo\_contrato
- condicao\_pagamento\_id
- estado
- data\_inicio
- data\_fim
- observacoes
- created\_at
- updated\_at

Regras:

- todo contrato nasce de orçamento aceite
- todo contrato gera sempre uma intervenção

### ### 7.4 CondicaoPagamento

Modelo de faseamento de pagamento.

Campos principais:

- id
- codigo
- nome
- descricao
- ativo

Valores iniciais:

- 100\_FIM
- 50\_50
- 30\_30\_40

---

## 8. Intervenções e planeamento

### ### 8.1 Intervencao

Entidade central operacional.

Campos principais:

- id
- numero\_intervencao
- cliente\_id
- equipamento\_id
- pedido\_id
- contrato\_id
- tipo\_intervencao\_id
- prioridade
- estado
- responsavel\_user\_id
- titulo
- descricao
- data\_prevista
- periodo\_previsto
- data\_inicio\_prevista
- data\_fim\_prevista
- data\_conclusao
- origem
- criada\_por\_user\_id
- observacoes
- created\_at
- updated\_at

Valores sugeridos:

- periodo\_previsto: manha, tarde, dia\_completo

Estados sugeridos para MVP:

- criada
- planeada
- em\_execucao
- pausada
- concluida
- aguarda\_auditoria
- devolvida\_correcao
- por\_faturar
- faturada
- cancelada

Regras:

- pode existir sem equipamento
- pode estar ligada a contrato quando aplicável
- pode nascer de pedido ou de contrato

### ### 8.2 TipoIntervencao

Catálogo de tipos de intervenção.

Campos principais:

- id
- codigo
- nome
- descricao
- cor\_hex
- ativo

Tipos base esperados:

- contrato
- obra
- assistencia

Observações:

- já existem cores para diferentes intervenções, por isso a sinalização adicional deve ser leve

### ### 8.3 IntervencaoTecnico

Técnicos associados a uma intervenção.

Campos principais:

- id
- intervencao\_id
- user\_id
- papel
- tecnico\_principal
- created\_at

Observações:

- o MVP suporta vários técnicos na mesma intervenção
- deve existir um técnico principal para a sessão principal

### ### 8.4 PlaneamentoIntervencao

Reserva o agendamento operacional.

Campos principais:

- id
- intervencao\_id
- data\_planeada
- periodo
- observacoes
- planeado\_por\_user\_id
- created\_at
- updated\_at

Observações:

- pode ser incorporado em Intervencao numa fase inicial, mas ter tabela própria dá mais flexibilidade futura

---

## 9. Execução técnica

### ### 9.1 SessaoTrabalho

Sessão real de execução em campo.

Campos principais:

- id
- intervencao\_id
- tecnico\_responsavel\_user\_id
- started\_at
- ended\_at
- estado
- observacoes\_inicio
- observacoes\_fim
- created\_at
- updated\_at

Estados sugeridos:

- iniciada
- pausada
- terminada

Regra:

- se a sessão parar sem conclusão da intervenção, isso conta como pausa

### ### 9.2 SessaoTrabalhoTecnico

Técnicos participantes numa sessão.

Campos principais:

- id
- sessao\_trabalho\_id
- user\_id
- papel
- presente

### ### 9.3 RegistoHoras

Registo consolidado de horas por intervenção ou sessão.

Campos principais:

- id
- intervencao\_id
- sessao\_trabalho\_id
- user\_id
- inicio
- fim
- minutos\_trabalhados
- origem\_registo
- editado\_manual
- motivo\_edicao
- validado\_em\_auditoria
- created\_at
- updated\_at

Regra:

- o normal é nascer automaticamente da sessão
- edição manual existe, mas como exceção controlada

### ### 9.4 RegistoKM

Registo de quilómetros.

Campos principais:

- id
- intervencao\_id
- sessao\_trabalho\_id
- km\_ida
- km\_volta
- km\_total
- observacoes
- created\_at
- updated\_at

Regra:

- registado por deslocação da equipa, não por técnico individual

### ### 9.5 IntervencaoObservacao

Observações livres da execução.

Campos principais:

- id
- intervencao\_id
- sessao\_trabalho\_id
- user\_id
- texto
- visivel\_na\_auditoria
- created\_at

### ### 9.6 FotoIntervencao

Fotos tiradas em campo.

Campos principais:

- id
- intervencao\_id
- sessao\_trabalho\_id
- caminho\_armazenamento
- legenda
- captured\_by\_user\_id

- captured\_at

Regra:

- mínimo de 1 foto por intervenção

### ### 9.7 AssinaturaIntervencao

Assinatura recolhida no local.

Campos principais:

- id  
- intervencao\_id  
- sessao\_trabalho\_id  
- nome\_responsavel\_cliente  
- cargo\_responsavel\_cliente  
- assinatura\_path  
- assinada\_at  
- recolhida\_por\_user\_id  
- excecao\_autorizada  
- excecao\_autorizada\_por\_user\_id  
- motivo\_excecao

Regra:

- assinatura é exigida por defeito  
- exceções devem ficar rastreadas

### ### 9.8 ChecklistModelo

Modelo de checklist.

Campos principais:

- id  
- tipo Equipamento  
- nome  
- descricao  
- ativo  
- created\_at  
- updated\_at

### ### 9.9 ChecklistItem

Perguntas de cada checklist.

Campos principais:

- id  
- checklist\_modelo\_id  
- pergunta  
- ordem  
- ativo

### ### 9.10 ChecklistResposta

Respostas da checklist numa intervenção.

Campos principais:

- id  
- intervencao\_id  
- sessao\_trabalho\_id  
- checklist\_item\_id  
- resposta  
- observacoes  
- respondido\_por\_user\_id  
- responded\_at

Valores esperados:

- sim  
- nao  
- na

Observações:

- o modelo é sim/não, mas pode ser útil prever na no desenho técnico  
- a obrigatoriedade pode depender do fluxo e do tipo de intervenção/equipamento

---

## 10. Materiais, stock e fornecedores

### ### 10.1 Material

Catálogo de materiais e equipamentos com impacto no stock.

Campos principais:

- id
- codigo
- designacao
- tipo\_item
- unidade
- rastreio\_numero\_serie
- ativo
- observacoes
- created\_at
- updated\_at

Valores sugeridos para tipo\_item:

- material
- equipamento

### ### 10.2 StockLocal

Locais de stock.

Campos principais:

- id
- codigo
- nome
- tipo\_local
- ativo

Valores iniciais:

- CENTRAL

Observações:

- o MVP pode arrancar só com stock central

### ### 10.3 StockItem

Quantidade em stock por item e local.

Campos principais:

- id
- stock\_local\_id
- material\_id
- quantidade\_atual
- quantidade\_reservada
- stock\_minimo
- updated\_at

### ### 10.4 MovimentoStock

Histórico de movimentos de stock.

Campos principais:

- id
- material\_id
- stock\_local\_id
- tipo\_movimento
- quantidade
- numero\_serie
- origem\_tipo
- origem\_id
- observacoes
- executado\_por\_user\_id
- created\_at

Valores sugeridos:

- entrada
- saída
- ajuste
- reserva
- devolucao

### ### 10.5 IntervencaoMaterial

Material usado na intervenção.

Campos principais:

- id
- intervencao\_id
- sessao\_trabalho\_id
- material\_id
- descricao\_livre
- quantidade
- unidade
- numero\_serie
- origem\_registro
- created\_by\_user\_id
- created\_at
- updated\_at

Regras:

- deve aceitar catálogo e texto livre
- quando houver item rastreável, o número de série deve poder ser guardado

### ### 10.6 PedidoMaterial

Pedido urgente de material vindo do técnico.

Campos principais:

- id
- intervencao\_id
- pedido\_por\_user\_id
- estado
- destino\_setor
- observacoes
- created\_at
- updated\_at

Estados sugeridos:

- pedido
- em\_armazem
- reencaminhado\_compras
- satisfeito
- cancelado

### ### 10.7 PedidoMaterialItem

Linhas do pedido de material.

Campos principais:

- id
- pedido\_material\_id
- material\_id
- descricao\_livre
- quantidade
- unidade
- observacoes

### ### 10.8 Fornecedor

Catálogo de fornecedores.

Campos principais:

- id
- nome
- nif
- email
- telefone

- morada
- ativo
- observacoes

### ### 10.9 IntervencaoFornecedor

Ligação entre intervenção e fornecedores.

Campos principais:

- id
- intervencao\_id
- fornecedor\_id
- tipo\_relacao
- observacoes

Observações:

- uma intervenção pode ter vários fornecedores

---

## 11. Auditoria e faturação

### ### 11.1 AuditoriaIntervencao

Auditoria formal feita por engenharia.

Campos principais:

- id
- intervencao\_id
- auditado\_por\_user\_id
- data\_auditoria
- estado
- horas\_validadas
- km\_validados
- material\_validado
- fotos\_validadas
- checklist\_validada
- assinatura\_validada
- observacoes
- devolvida\_para\_correcao
- created\_at
- updated\_at

Estados sugeridos:

- em\_analise
- aprovada
- devolvida

Regra:

- quem abre a intervenção deve ser quem a audita

### ### 11.2 AuditoriaInconformidade

Lista de falhas encontradas na auditoria.

Campos principais:

- id
- auditoria\_intervencao\_id
- tipo\_inconformidade
- descricao
- resolvida
- resolvida\_at

### ### 11.3 PreparacaoFaturacao

Registo da passagem da intervenção para faturação.

Campos principais:

- id
- intervencao\_id
- contrato\_id
- preparado\_por\_user\_id

- estado
- data\_preparacao
- observacoes
- created\_at
- updated\_at

Estados sugeridos:

- por\_faturar
- em\_tratamento
- faturado

Observações:

- no MVP o foco é workflow e contexto, não cálculo financeiro detalhado

---

## 12. Alertas, bloqueios e rastreabilidade

### ### 12.1 Alerta

Alertas internos do sistema.

Campos principais:

- id
- tipo\_alerta
- entidade\_tipo
- entidade\_id
- titulo
- descricao
- severidade
- estado
- user\_destino\_id
- created\_at
- lido\_at
- resolvido\_at

Tipos esperados:

- intervencao\_atrasada
- aguarda\_auditoria
- aguarda\_faturacao
- orcamento\_pendente
- stock\_baixo
- processo\_parado

### ### 12.2 BloqueioProcesso

Registo de bloqueios funcionais.

Campos principais:

- id
- entidade\_tipo
- entidade\_id
- motivo\_bloqueio
- detalhe
- ativo
- created\_at
- resolvido\_at

### ### 12.3 ExcecaoAdmin

Registo de desbloqueios administrativos.

Campos principais:

- id
- entidade\_tipo
- entidade\_id
- bloqueio\_processo\_id
- autorizado\_por\_user\_id
- motivo
- created\_at

Regra:

- toda exceção deve ficar rastreada com quem, quando e porquê

### ### 12.4 LogSistema

Log transversal de eventos importantes.

Campos principais:

- id
- entidade\_tipo
- entidade\_id
- acao
- descricao
- user\_id
- metadata\_json
- created\_at

Objetivo:

- suportar auditoria, histórico e timeline automática

---

## 13. Relações principais entre entidades

### Relações nucleares

- Perfil 1:N User
- Perfil N:N Permissao via PerfilPermissao
- Cliente 1:N ContactoCliente
- Cliente 1:N ClienteDocumento
- Cliente 1:N TimelineCliente
- Cliente 1:N Equipamento
- Cliente 1:N Pedido
- Cliente 1:N PedidoRascunho
- PedidoRascunho 1:0..1 Pedido (relacao de origem via `rascunho\_origem\_id`)
- Pedido 1:N AnexoPedido
- PedidoRascunho 1:N AnexoPedido
- Pedido 1:N PedidoEventoLog
- PedidoRascunho 1:N PedidoEventoLog
- PedidoSlaConfig 1:N Pedido
- PedidoSlaConfig 1:N PedidoRascunho
- Pedido 1:0..1 Orcamento
- Orcamento 1:N OrcamentoLinha
- Orcamento 1:0..1 Contrato
- Contrato N:1 CondicaoPagamento
- Contrato 1:N Intervencao ou 1:1 no fluxo principal inicial
- Cliente 1:N Intervencao
- Equipamento 1:N Intervencao
- Intervencao N:N User via IntervencaoTecnico
- Intervencao 1:N SessaoTrabalho
- SessaoTrabalho N:N User via SessaoTrabalhoTecnico
- SessaoTrabalho 1:N RegistoHoras
- Intervencao 1:N RegistoKM
- Intervencao 1:N IntervencaoMaterial
- Intervencao 1:N FotoIntervencao
- Intervencao 1:0..1 AssinaturaIntervencao
- ChecklistModelo 1:N ChecklistItem
- Intervencao 1:N ChecklistResposta
- Material 1:N StockItem
- Material 1:N MovimentoStock
- Intervencao 1:N PedidoMaterial
- PedidoMaterial 1:N PedidoMaterialItem
- Intervencao N:N Fornecedor via IntervencaoFornecedor
- Intervencao 1:N AuditoriaIntervencao
- AuditoriaIntervencao 1:N AuditoriaInconformidade
- Intervencao 1:N PreparacaoFaturacao

- entidades várias 1:N Alerta
- entidades várias 1:N BloqueioProcesso
- entidades várias 1:N ExcecaoAdmin
- entidades várias 1:N LogSistema

---

## 14. Campos de estado que devem ser catálogos ou enums

É recomendável tratar como enum ou tabela controlada:

- perfil.codigo
- pedido.origem
- pedido.urgencia
- pedido.estado
- pedido.sub\_estado
- pedido.tipo\_resultado
- pedido\_rascunho.origem
- pedido\_rascunho.urgencia\_percebida
- pedido\_rascunho.estado
- pedido\_evento\_log.tipo\_evento
- pedido\_sla\_config.urgencia
- orcamento.estado
- contrato.estado
- contrato.tipo\_contrato
- intervencao.prioridade
- intervencao.estado
- intervencao.periodo\_previsto
- tipo\_intervencao.codigo
- sessao\_trabalho.estado
- registo\_horas.origem\_registro
- material.tipo\_item
- pedido\_material.estado
- auditoria\_intervencao.estado
- alerta.tipo\_alerta
- alerta.severidade
- alerta.estado

---

## 15. Regras estruturais importantes

1. Uma intervenção pode existir sem equipamento
2. Um orçamento aceite gera sempre contrato
3. Um contrato gera sempre intervenção
4. Uma intervenção pode ter vários técnicos
5. A sessão principal pertence a um técnico responsável
6. O registo de horas deve nascer preferencialmente da sessão
7. O registo de km é por deslocação da equipa
8. O material deve aceitar catálogo e texto livre
9. O stock cobre materiais e equipamentos
10. O número de série deve ser guardado quando existir
11. A assinatura é exigida por defeito, mas exceções têm de ficar registadas
12. A checklist existe no MVP, mas a obrigatoriedade pode depender do fluxo
13. A auditoria pode devolver a intervenção para correção
14. Só admin pode desbloquear exceções
15. Tudo o que altera estado relevante deve gerar log

---

## 16. Tabelas mínimas para arrancar desenvolvimento

Se for preciso começar rápido, o núcleo mínimo é:

- User
- Perfil
- Cliente

- ContactoCliente
- Equipamento
- Pedido
- PedidoRascunho
- PedidoEventoLog
- PedidoSlaConfig
- AnexoPedido
- Orcamento
- OrcamentoLinha
- Contrato
- CondicaoPagamento
- Intervencao
- TipoIntervencao
- IntervencaoTecnico
- SessaoTrabalho
- SessaoTrabalhoTecnico
- RegistoHoras
- RegistoKM
- IntervencaoMaterial
- FotoIntervencao
- AssinaturalIntervencao
- ChecklistModelo
- ChecklistItem
- ChecklistResposta
- Material
- StockItem
- MovimentoStock
- PedidoMaterial
- PedidoMaterialItem
- Fornecedor
- IntervencaoFornecedor
- AuditorialIntervencao
- PreparacaoFaturacao
- Alerta
- BloqueioProcesso
- ExcecaoAdmin
- LogSistema

---

## 17. Próximos passos recomendados

1. Converter este modelo lógico em esquema técnico real
2. Definir tipos de dados, chaves, índices e constraints
3. Fechar enums e catálogos oficiais
4. Desenhar diagrama entidade-relação
5. Partir o sistema em backlog técnico por módulo
6. Definir API e fluxos transacionais críticos

## Backlog Técnico MVP

Fonte: Backlog\_Tecnico\_MVP\_Norbidel.md

## Backlog Técnico do MVP

### Sistema Interno de Gestão Operacional, Norbidel

Versão: v1

Data: Abril 2026

---

## 1. Objetivo deste documento

Este documento transforma a visão funcional e a estrutura de dados do MVP num backlog técnico executável.

O objetivo é:

- dividir o produto em módulos
- organizar o desenvolvimento por épicos e tarefas
- identificar prioridades
- criar base para estimativas e planeamento

---

## 2. Lógica de prioridade

### Prioridade P0

Sem isto o MVP não funciona operacionalmente.

### Prioridade P1

Muito importante para o fluxo principal, mas pode entrar logo após o núcleo.

### Prioridade P2

Importante, mas pode entrar depois sem bloquear o arranque do MVP.

---

## 3. Ordem recomendada de implementação

1. Base técnica e autenticação
2. Utilizadores e permissões
3. RH e colaboradores
4. CRM / Clientes
5. Pedidos / Entradas
6. Orçamentos
7. Contratos
8. Intervenções
9. Planeamento
10. Módulo técnico
11. Materiais / Stock / Pedidos urgentes
12. Auditoria
13. Faturação / Por faturar
14. Dashboard e alertas
15. Administração e rastreabilidade

---

## 4. Épico A, Base técnica do sistema

### Objetivo

Criar a fundação técnica para a app.

### Prioridade

P0

### Tarefas

- criar estrutura inicial do projeto
- configurar base de dados
- configurar ORM ou camada de acesso a dados
- criar sistema base de migrações
- configurar autenticação
- configurar gestão de sessão
- configurar controlo de permissões base
- configurar storage para anexos, fotos e assinaturas
- configurar logs de aplicação
- configurar ambiente dev, staging e produção
- definir convenções de erros, validações e respostas da API

### Entregáveis

- projeto inicial funcional

- login técnico operacional
- base de dados ligada
- sistema pronto para crescer por módulos

---

## 5. Épico B, Utilizadores e permissões

### ### Objetivo

Garantir acesso controlado por perfil.

### ### Prioridade

P0

### ### Tarefas

- criar tabela e modelo de perfis
- criar tabela e modelo de utilizadores
- criar tabela de permissões
- criar associação perfil-permissão
- implementar CRUD de utilizadores
- implementar ativação e desativação de utilizadores
- implementar associação de perfil ao utilizador
- bloquear ações por perfil
- garantir que técnicos não veem preços nem custos
- criar ecrã de gestão de utilizadores

### ### Critérios práticos

- admin tem acesso total
- faturação vê o necessário para faturar
- engenharia cria, planeia, audita e corrige
- armazém gere materiais e pedidos
- técnico vê e executa trabalho de campo

---

## 6. Épico C, RH e colaboradores

### ### Objetivo

Criar a base administrativa de recursos humanos do MVP, com foco em cadastro, documentação, alertas e equipamento atribuído.

### ### Prioridade

P1

### ### Tarefas

- criar entidade Colaborador
- criar entidade ColaboradorDocumento
- criar entidade ColaboradorEquipamento
- criar entidade AlertaRH
- criar estrutura para total mensal de horas extra por colaborador
- permitir associação opcional entre utilizador da app e colaborador
- implementar CRUD de colaboradores
- permitir função em texto livre
- implementar campos salariais base, valor de hora extra e suplementos
- implementar upload e gestão documental do colaborador
- suportar datas de validade e documentos obrigatórios
- tratar carta de condução como documento opcional
- implementar registo de equipamento entregue ao colaborador
- implementar alertas internos para contratos e documentos a expirar
- configurar alerta de medicina do trabalho com antecedência de 15 dias
- restringir visibilidade do módulo RH por perfil
- garantir que técnicos não têm acesso ao módulo RH
- limitar visibilidade de alertas RH a admin e faturação/RH
- preparar separação futura do módulo de viaturas
- criar ecrã de ficha de colaborador
- criar lista de colaboradores com filtro por função
- criar lista de alertas RH

### ### Critérios práticos

- a app não substitui a câmara de ponto no MVP
- férias e faltas ficam fora desta fase como fluxos administrativos, embora o estado férias exista na ficha do colaborador
- o módulo RH é administrativo, documental e de controlo
- aprovação futura fica no perfil admin
- deve ser possível saber que documentos e equipamentos estão atribuídos a cada colaborador

---

## 7. Épico D, CRM / Clientes

### ### Objetivo

Criar base forte de clientes e histórico.

### ### Prioridade

P0

### ### Tarefas

- criar entidade Cliente
- criar entidade ContactoCliente
- criar entidade ClienteDocumento
- criar entidade TimelineCliente
- criar entidade Equipamento
- implementar CRUD de clientes
- implementar CRUD de contactos
- implementar upload e listagem de documentos do cliente
- implementar registo de equipamentos associados ao cliente
- implementar timeline cronológica única do cliente
- permitir reutilização de código interno existente
- criar lista de clientes com pesquisa e filtros
- criar ficha completa do cliente

### ### Critérios práticos

- ficha do cliente tem visão comercial e operacional
- histórico do cliente fica centralizado
- documentos técnicos podem ser anexados

---

## 8. Épico E, Pedidos / Entradas

### ### Objetivo

Garantir entrada formal de todos os pedidos.

### ### Prioridade

P0

### ### Tarefas

- criar entidade Pedido
- criar entidade AnexoPedido
- implementar formulário de criação de pedido
- suportar origem do pedido
- suportar urgência
- suportar anexos no pedido
- implementar lista de pedidos
- implementar filtro por estado, urgência, cliente e data
- implementar triagem do pedido
- permitir decidir se pedido vira intervenção, orçamento ou análise
- registar quem recebeu e quem triou
- gerar entrada na timeline do cliente

### ### Critérios práticos

- pedidos vindos por telefone também têm de ser registados
- anexos podem ser projetos, desenhos ou outros documentos

---

## 9. Épico F, Orçamentos

### ### Objetivo

Gerir o ciclo de orçamentação.

### ### Prioridade

P0

### ### Tarefas

- criar entidade Orcamento
- criar entidade OrcamentoLinha
- implementar criação de orçamento a partir de pedido
- implementar edição de orçamento
- implementar estados do orçamento
- implementar lista de orçamentos
- implementar detalhe do orçamento
- permitir marcar orçamento como enviado
- permitir marcar orçamento como aceite
- permitir marcar orçamento como recusado
- registar datas de envio e resposta
- gerar eventos na timeline do cliente

### ### Critérios práticos

- orçamento pendente e recusado ficam registados
- orçamento aceite deve permitir conversão em contrato

---

## 10. Épico G, Contratos

### ### Objetivo

Gerir contratos e ligação ao fluxo operacional.

### ### Prioridade

P0

### ### Tarefas

- criar entidade Contrato
- criar entidade CondicaoPagamento
- implementar criação de contrato a partir de orçamento aceite
- suportar condições 100% fim, 50/50 e 30/30/40
- criar lista de contratos
- criar ficha de contrato
- ligar contrato ao cliente
- ligar contrato ao orçamento de origem
- garantir criação de intervenção associada
- permitir consulta de fases/condições relevantes para faturação

### ### Critérios práticos

- contrato é separado de orçamento
- contrato aceite/originado deve dar lugar a intervenção

---

## 11. Épico H, Intervenções

### ### Objetivo

Criar o núcleo operacional da app.

### ### Prioridade

P0

### ### Tarefas

- criar entidade Intervencao
- criar entidade TipoIntervencao
- criar entidade IntervencaoTecnico
- implementar criação manual de intervenção
- implementar criação de intervenção a partir de pedido

- implementar criação de intervenção a partir de contrato
- suportar associação opcional a equipamento
- suportar associação a contrato quando aplicável
- suportar tipo de intervenção
- suportar prioridade
- suportar responsável
- suportar estado
- criar lista de intervenções
- criar filtros por estado, prioridade, cliente, técnico e data
- criar ficha da intervenção
- mostrar histórico da intervenção
- mostrar informação do cliente e equipamento
- mostrar recomendações preventivas relevantes
- gerar eventos na timeline do cliente

#### ### Critérios práticos

- intervenção pode existir sem equipamento
- tipos base: contratos, obras, assistências
- cores por tipo já devem estar previstas

---

## 12. Épico I, Planeamento

#### ### Objetivo

Planear o trabalho diário e semanal.

#### ### Prioridade

P0

#### ### Tarefas

- criar lógica de planeamento diário
- criar lógica de planeamento semanal
- suportar atribuição de técnicos
- suportar período manhã, tarde ou dia completo
- criar ecrã de planeamento diário
- criar ecrã de planeamento semanal
- destacar o que é para hoje
- destacar o que está atrasado
- permitir editar planeamento
- refletir cores por tipo de intervenção
- aplicar sinalização visual leve adicional quando útil

#### ### Critérios práticos

- o planeamento tem de ser simples e operacional
- organização manhã/tarde deve existir

---

## 13. Épico J, Módulo técnico

#### ### Objetivo

Dar aos técnicos uma área simples e rápida para executar intervenções.

#### ### Prioridade

P0

#### ### Tarefas

- criar home do técnico
- mostrar intervenções do dia
- mostrar intervenções abertas
- criar detalhe técnico da intervenção
- mostrar cliente, local, equipamento e histórico relevante
- implementar ação iniciar trabalho
- implementar ação parar trabalho
- implementar ação concluir trabalho
- criar sessão principal por técnico responsável

- permitir associar restantes técnicos à sessão
- refletir pausa quando a sessão é parada sem conclusão
- criar cards grandes e interface simplificada
- otimizar para uso em tablet
- permitir escrever observações

#### ### Critérios práticos

- UX deve ser muito simples
- fluxo principal deve ser rápido em contexto real de campo

---

## 14. Épico K, Registo de horas

#### ### Objetivo

Registar tempo de trabalho de forma fiável.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade SessaoTrabalho
- criar entidade SessaoTrabalhoTecnico
- criar entidade RegistoHoras
- gerar horas automaticamente a partir da sessão
- calcular duração da sessão
- permitir edição manual controlada
- exigir motivo de edição manual
- mostrar histórico de alterações
- expor dados para auditoria

#### ### Critérios práticos

- padrão é automático
- edição existe como exceção

---

## 15. Épico L, Registo de KM

#### ### Objetivo

Registar deslocação da equipa.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade RegistoKM
- permitir registar km de ida
- permitir registar km de volta
- calcular km total
- associar registo à intervenção ou sessão
- expor dados para auditoria

#### ### Critérios práticos

- o km é da equipa, não individual

---

## 16. Épico M, Material e consumo em intervenção

#### ### Objetivo

Registar material usado nas intervenções.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade Material
- criar entidade IntervencaoMaterial
- permitir escolha por catálogo

- permitir texto livre
- permitir quantidade e unidade
- permitir registo de número de série quando aplicável
- permitir edição controlada
- expor dados para auditoria

#### ### Critérios práticos

- sistema tem de suportar materiais e equipamentos
- rastreio por série entra logo no MVP

---

## 17. Épico N, Fotos

#### ### Objetivo

Registrar evidência fotográfica.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade FotoIntervencao
- permitir captura/upload de foto
- associar foto à intervenção e sessão
- mostrar galeria na intervenção
- validar mínimo obrigatório de 1 foto
- expor fotos para auditoria

#### ### Critérios práticos

- foto mínima obrigatória por intervenção

---

## 18. Épico O, Checklist

#### ### Objetivo

Executar checklist técnica por tipo de equipamento.

#### ### Prioridade

P1

#### ### Tarefas

- criar entidade ChecklistModelo
- criar entidade ChecklistItem
- criar entidade ChecklistResposta
- permitir definir modelos por tipo de equipamento
- carregar checklist adequada na intervenção
- suportar resposta sim/não
- permitir observações por resposta
- expor checklist para auditoria
- implementar regra de obrigatoriedade por fluxo quando aplicável

#### ### Critérios práticos

- checklist existe no MVP
- a obrigatoriedade não é universal em todos os casos

---

## 19. Épico P, Assinatura

#### ### Objetivo

Recolher assinatura do responsável do cliente.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade AssinaturaIntervencao
- implementar captura de assinatura
- guardar nome do responsável do cliente

- guardar data e utilizador que recolheu
- bloquear conclusão quando assinatura for obrigatória e faltar
- permitir exceção administrativa rastreada

#### ### Critérios práticos

- assinatura é exigida por defeito
- exceções têm de ficar justificadas

---

## 20. Épico Q, Materiais urgentes, armazém e compras

#### ### Objetivo

Suportar pedidos urgentes de material a partir do terreno.

#### ### Prioridade

P1

#### ### Tarefas

- criar entidade PedidoMaterial
- criar entidade PedidoMaterialItem
- criar fluxo pedido ! armazém
- criar fluxo armazém ! compras quando não satisfaz
- implementar estados do pedido
- criar lista de pedidos urgentes
- criar detalhe do pedido urgente
- permitir resposta do armazém
- permitir reencaminhamento para compras

#### ### Critérios práticos

- o fluxo tem de refletir a realidade operacional atual

---

## 21. Épico R, Stock e fornecedores

#### ### Objetivo

Dar controlo mínimo de stock ao MVP.

#### ### Prioridade

P1

#### ### Tarefas

- criar entidade StockLocal
- criar entidade StockItem
- criar entidade MovimentoStock
- criar entidade Fornecedor
- criar entidade IntervencaoFornecedor
- suportar stock central
- suportar stock mínimo
- gerar alertas de stock baixo
- registar entradas e saídas
- registar ajustes
- permitir ligar intervenção a um ou mais fornecedores

#### ### Critérios práticos

- stock inclui materiais e equipamentos
- vários fornecedores podem existir por intervenção

---

## 22. Épico S, Auditoria

#### ### Objetivo

Validar tecnicamente o trabalho executado antes da faturação.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade AuditorialIntervencao
- criar entidade AuditorialInconformidade
- criar lista de intervenções por auditar
- criar ficha de auditoria
- mostrar horas, km, material, fotos, checklist e assinatura
- permitir aprovar auditoria
- permitir devolver para correção
- obrigar registo de motivo de devolução
- alterar estado da intervenção conforme resultado
- garantir que quem abre a intervenção é quem a audita

#### ### Critérios práticos

- auditoria é ponto crítico do fluxo
- devolução tem de voltar ao técnico para correção

---

## 23. Épico T, Faturação / Por faturar

#### ### Objetivo

Entregar à faturação tudo o que está pronto após auditoria.

#### ### Prioridade

P0

#### ### Tarefas

- criar entidade PreparacaoFaturacao
- criar lista por faturar
- mostrar contexto da intervenção
- mostrar contrato e condição de pagamento quando existir
- permitir marcar item como em tratamento
- permitir marcar como faturado
- gerar eventos na timeline do cliente

#### ### Critérios práticos

- o MVP não precisa de motor financeiro detalhado
- precisa de workflow claro até ao software certificado

---

## 24. Épico U, Dashboard

#### ### Objetivo

Dar visão operacional rápida do estado da empresa.

#### ### Prioridade

P1

#### ### Tarefas

- criar dashboard principal
- mostrar intervenções em aberto
- mostrar atrasos
- mostrar o que é para hoje
- mostrar o que está parado
- mostrar aguarda auditoria
- mostrar aguarda faturação
- mostrar orçamentos pendentes
- mostrar stock baixo
- criar indicadores principais

#### ### Critérios práticos

- dashboard deve ser simples e útil
- foco em ação, não em excesso de gráficos

---

## 25. Épico V, Alertas

#### ### Objetivo

Gerar visibilidade de situações críticas dentro da app.

### Prioridade

P1

### Tarefas

- criar entidade Alerta
- gerar alerta de intervenção atrasada
- gerar alerta de aguarda auditoria
- gerar alerta de aguarda faturação
- gerar alerta de orçamento pendente
- gerar alerta de stock baixo
- gerar alerta de processo parado
- mostrar alertas em dashboard e áreas relevantes
- permitir marcar alerta como lido ou resolvido

---

## 26. Épico W, Bloqueios e exceções administrativas

### Objetivo

Garantir que o processo não avança sem os elementos obrigatórios.

### Prioridade

P0

### Tarefas

- criar entidade BloqueioProcesso
- criar entidade ExcecaoAdmin
- implementar regras de bloqueio por ausência de dados obrigatórios
- bloquear avanço se faltar horas
- bloquear avanço se faltar km
- bloquear avanço se faltar material obrigatório quando aplicável
- bloquear avanço se faltar assinatura obrigatória
- bloquear avanço se faltar foto mínima
- bloquear avanço se faltar checklist obrigatória quando aplicável
- permitir desbloqueio só por admin
- registar motivo, utilizador e data da exceção

---

## 27. Épico X, Logs e rastreabilidade

### Objetivo

Garantir histórico auditável das ações importantes.

### Prioridade

P0

### Tarefas

- criar entidade LogSistema
- registar criação de pedido
- registar triagem
- registar criação e mudança de estado de orçamento
- registar criação e mudança de estado de contrato
- registar criação e mudança de estado de intervenção
- registar início, pausa e fim de sessão
- registar edições manuais críticas
- registar auditoria
- registar desbloqueios administrativos
- registar preparação para faturação

---

## 28. Épico Y, Offline básico

### Objetivo

Permitir operação mínima em contexto sem internet.

### Prioridade

P2

### Tarefas

- identificar ecrãs críticos para offline
- definir dados mínimos a sincronizar para o técnico
- suportar cache local das intervenções do dia
- suportar registo local temporário de horas, fotos, checklist e assinatura
- criar mecanismo de sincronização quando a ligação voltar
- tratar conflitos básicos

### Critérios práticos

- offline entra no MVP como versão básica, não completa

---

## 29. Épico Z, Administração funcional

### Objetivo

Dar controlo operacional ao admin.

### Prioridade

P1

### Tarefas

- gerir utilizadores
- gerir perfis e permissões
- gerir tipos de intervenção
- gerir condições de pagamento
- gerir catálogos de materiais
- gerir modelos de checklist
- gerir parâmetros base do sistema

---

## 30. Backlog resumido por ordem de execução

### Fase 1, núcleo operacional obrigatório

- Épico A, Base técnica
- Épico B, Utilizadores e permissões
- Épico C, CRM / Clientes
- Épico D, Pedidos / Entradas
- Épico E, Orçamentos
- Épico F, Contratos
- Épico G, Intervenções
- Épico H, Planeamento
- Épico I, Módulo técnico
- Épico J, Registo de horas
- Épico K, Registo de KM
- Épico L, Material em intervenção
- Épico M, Fotos
- Épico O, Assinatura
- Épico R, Auditoria
- Épico S, Faturação / Por faturar
- Épico V, Bloqueios e exceções
- Épico W, Logs e rastreabilidade

### Fase 2, reforço operacional

- Épico N, Checklist
- Épico P, Pedidos urgentes de material
- Épico Q, Stock e fornecedores
- Épico T, Dashboard
- Épico U, Alertas
- Épico Y, Administração funcional

### Fase 3, robustez adicional

- Épico X, Offline básico

---

## 31. Próximos passos recomendados

1. partir cada épico em user stories técnicas menores
2. estimar esforço por módulo
3. decidir stack técnica final
4. definir sprint 1 com os épicos A, B e C
5. desenhar wireframes dos ecrãs mais críticos em paralelo

## Stack Técnica MVP

Fonte: Stack\_Tecnica\_MVP\_Norbidel.md

## Stack Técnica MVP Norbidel

### Sistema Interno de Gestão Operacional

Versão: v1

Data: Abril 2026

---

### 1. Objetivo

Definir a stack técnica recomendada para o MVP da app interna da Norbidel, de forma simples, moderna, escalável e rápida de implementar.

---

### 2. Recomendação principal

### Frontend + Backend

- **Next.js**
- **TypeScript**
- **Node.js**

### Base de dados

- **PostgreSQL**

### ORM

- **Prisma**

### UI

- **Tailwind CSS**
- **shadcn/ui**

### Autenticação

- **Auth.js** (ou implementação equivalente com sessões internas e controlo por perfis)

### Storage de ficheiros

- **S3-compatible storage**
- exemplo: Cloudflare R2
- alternativa: Supabase Storage

### Deploy

- **Vercel** para app
- **PostgreSQL gerido** ou servidor dedicado para base de dados
- opção prática: **Railway / Render / VPS** conforme custo e controlo pretendido

---

### 3. Porque esta stack é a mais adequada

A app da Norbidel precisa de:

- muitos formulários e registos internos
- estados de processo bem definidos

- perfis e permissões
- anexos, documentos, fotos e assinaturas
- histórico e auditoria
- utilização em escritório e no terreno

Esta stack permite:

- desenvolvimento rápido
- boa estrutura para crescer por módulos
- forte compatibilidade com PostgreSQL
- boa experiência web em desktop e mobile
- manutenção relativamente simples
- facilidade em contratar desenvolvimento no futuro

---

## 4. Forma recomendada do produto no MVP

### Web app interna responsiva

O MVP deve ser uma **\*\*web app responsiva\*\***, não uma app mobile nativa de início.

Isto permite:

- uso em computador por administração, engenharia, faturação e armazém
- uso em telemóvel por técnicos no terreno
- menor custo inicial
- entrega mais rápida
- menos complexidade técnica na fase inicial

Mais tarde pode evoluir para:

- PWA mais robusta
- app mobile nativa, se fizer sentido

---

## 5. Estrutura técnica recomendada

### App

- painel administrativo
- área operacional
- área técnica mobile-friendly

### API

- rotas server-side no ecossistema Next.js
- validação centralizada
- controlo por permissões
- logs e auditoria por ação relevante

### Base de dados

- PostgreSQL com migrações controladas
- modelo alinhado com:
  - utilizadores e perfis
  - clientes
  - pedidos
  - orçamentos
  - contratos
  - intervenções
  - planeamento
  - materiais
  - auditoria
  - faturação

### Ficheiros

- documentos do cliente
- documentos de colaborador
- anexos de processo
- fotos técnicas
- assinaturas

---

## 6. Segurança e controlo

Desde o MVP, a app deve prever:

- login obrigatório
- perfis por função
- permissões por módulo/ação
- técnicos sem acesso a preços/custos
- rastreabilidade de ações críticas
- histórico de alterações relevantes

---

## 7. O que não é recomendado nesta fase

### Flutter / app nativa logo no início

Não recomendado para a fase MVP porque aumenta tempo, custo e complexidade.

### No-code / low-code como base principal

Não recomendado para este caso devido à complexidade operacional, auditoria, permissões e crescimento futuro.

### Arquitetura demasiado pesada logo de raiz

Exemplo: microserviços, múltiplos backends separados, infraestrutura enterprise prematura.

Para o MVP, o ideal é:

- arquitetura simples
- modularidade lógica
- capacidade de escalar depois

---

## 8. Conclusão

A stack recomendada para o MVP da Norbidel é:

- **Next.js**
- **TypeScript**
- **PostgreSQL**
- **Prisma**
- **Tailwind + shadcn/ui**
- **Auth.js**
- **Storage S3-compatible**

A forma inicial do produto deve ser:

- **web app interna responsiva**

Esta abordagem dá o melhor equilíbrio entre:

- velocidade
- custo
- qualidade
- escalabilidade
- simplicidade operacional

---

## 9. Próximo passo recomendado

Depois desta decisão, o próximo passo deve ser:

**definir o Sprint 1 técnico do MVP**

Esse Sprint 1 deve arrancar com:

1. base do projeto
2. autenticação
3. perfis e permissões
4. estrutura inicial da base de dados
5. módulo de utilizadores
6. módulo de clientes

# Sprint 1 Técnico MVP

Fonte: Sprint\_1\_Tecnico\_MVP\_Norbidel.md

## Sprint 1 Técnico MVP Norbidel

### Sistema Interno de Gestão Operacional

Versão: v1

Data: Abril 2026

---

#### 1. Objetivo do Sprint 1

Transformar a decisão funcional e a stack técnica numa base real de produto pronta para começar a crescer com ordem.

O Sprint 1 deve entregar o núcleo técnico mínimo para arrancar o MVP com segurança.

---

#### 2. Objetivos práticos

No final deste sprint deve existir:

- projeto criado e organizado
- base de dados ligada
- autenticação funcional
- sistema de perfis e permissões base
- estrutura inicial de layout e navegação
- módulo de utilizadores funcional
- módulo de clientes em base inicial
- fundação pronta para os módulos seguintes

---

#### 3. Escopo do Sprint 1

### Bloco A - Base do projeto

- criar projeto com **Next.js + TypeScript**
- configurar estrutura base de pastas
- configurar Tailwind CSS
- configurar shadcn/ui
- definir convenções de código
- definir estratégia de environment variables
- criar layout base da aplicação
- criar navegação protegida

### Bloco B - Base de dados e ORM

- configurar **PostgreSQL**
- configurar **Prisma**
- criar primeira schema base
- criar sistema de migrações
- preparar seeds iniciais
- documentar arranque local da base de dados

### Bloco C - Autenticação

- configurar login interno
- criar sessão autenticada
- proteger rotas privadas
- preparar logout
- criar base de recuperação/gestão futura de credenciais

### Bloco D - Perfis e permissões

- criar modelos de:
  - perfil

- permissão
- perfil\_permissão
- utilizador
- definir perfis iniciais:
  - ADMIN
  - FATURACAO\_RH
  - ENGENHARIA\_PROJETOS
  - COMPRAS\_ARMAZEM
  - TECNICO
- bloquear acesso por perfil
- impedir que técnico veja áreas administrativas

#### ### Bloco E - Módulo de utilizadores

- criar listagem de utilizadores
- criar formulário de criação
- editar utilizador
- ativar/desativar utilizador
- associar perfil ao utilizador
- registar timestamps base

#### ### Bloco F - Módulo inicial de clientes

- criar entidade Cliente
- criar listagem de clientes
- criar ficha simples de cliente
- criar formulário de criação/edição
- permitir pesquisa base
- preparar terreno para contactos, documentos e equipamentos

---

## 4. Fora do Sprint 1

Não entram ainda neste sprint:

- RH completo
- pedidos/entradas completos
- orçamentos
- contratos
- intervenções
- planeamento
- stock
- faturação
- auditoria avançada
- notificações complexas
- app mobile nativa

---

## 5. Entregáveis esperados

No fim do sprint deve existir:

#### ### Entregáveis técnicos

- repositório inicial estruturado
- app a correr localmente
- base de dados ligada
- Prisma configurado
- migração inicial criada
- autenticação funcional
- perfis seeded
- permissões base funcionais

#### ### Entregáveis funcionais

- login
- dashboard inicial simples
- gestão de utilizadores
- gestão base de clientes

---

## 6. Modelo de dados mínimo do Sprint 1

Entidades mínimas a implementar já:

- User
- Perfil
- Permissao
- PerfilPermissao
- Cliente

Campos detalhados podem seguir a estrutura definida em:

- `Estrutura\_Base\_Dados\_MVP\_Norbidel.md`

---

## 7. Ordem recomendada dentro do sprint

1. bootstrap do projeto
2. Tailwind + UI base
3. PostgreSQL + Prisma
4. schema inicial
5. autenticação
6. perfis e permissões
7. layout protegido
8. CRUD utilizadores
9. CRUD clientes
10. seed inicial e validação final

---

## 8. Critérios de aceitação

O Sprint 1 está concluído quando:

- um admin consegue fazer login
- um admin consegue criar, editar e desativar utilizadores
- o sistema distingue perfis
- rotas protegidas não são acessíveis sem login
- um utilizador técnico não vê áreas indevidas
- clientes podem ser criados, editados e pesquisados
- a base do projeto fica pronta para o Sprint 2

---

## 9. Riscos a evitar

- tentar fazer módulos a mais neste sprint
- complicar autenticação cedo demais
- criar permissões excessivamente detalhadas antes de validar uso real
- perder tempo com design visual demasiado refinado
- tentar resolver já toda a lógica operacional da empresa

Este sprint deve focar-se em **\*\*fundação sólida e velocidade controlada\*\***.

---

## 10. Recomendação prática

A melhor abordagem agora é:

- fechar este Sprint 1 como referência
- criar logo o projeto base
- começar implementação pela ordem definida

---

## 11. Próximo passo após este documento

Depois deste documento, o passo mais lógico é um destes:

1. **eu gerar a estrutura inicial técnica do projeto**
2. **eu preparar o plano do Sprint 2**

A recomendação certa é:

**começar já a estrutura inicial do projeto.**

## SPEC Bloco 1 - Auth, Seed e Utilizadores

Fonte: SPEC\_Bloco1\_Auth\_Seed\_Utilizadores.md

# Norbidel App - Especificação: Bloco 1 (Auth + Seed + Utilizadores)

## 1. Autenticação (NextAuth.js / Auth.js v5)

### 1.1 Provider

- **Credentials** (email + password)
- Base de dados de sessões: **PostgreSQL** (tabela `accounts`, `sessions`, `users` do NextAuth)

### 1.2 Utilizador Admin Inicial (Seed)

- Email: `admin@norbidel.com` (ou email a definir por Fred)
- Password: criada via seed, alterável pelo admin após primeiro login
- Perfil: **admin**

### 1.3 Proteção de Rotas

- Middleware Next.js (`middleware.ts`) protege todas as rotas `/admin/`
- `Sem sessão válida!` redirect para `/login`
- Rotas públicas: `/login`, `/` (landing), `/api/auth/`

### 1.4 Sessão

- JWT strategy (simples, sem session DB table extra)
- Sessão contém: `userId`, `email`, `name`, `role`
- Tempo de expiração: 7 dias

### 1.5 Logout

- Botão/logout que faz `signOut()` e redirect para `/login`

---

## 2. Seed - Perfis e Permissões Base

### 2.1 Perfis (tabela `profiles`)

ID	Nome	Descrição
1	admin	Acesso total à plataforma
2	escritorio	Acesso a clientes, não áreas técnicas
3	tecnico	Acesso a terreno/operações, sem admin

### 2.2 Permissões Base (tabela `permissions`)

ID	Perfil	Rotas/Ações permitidas
P001	admin	<code>*</code> (tudo)
P002	escritorio	<code>/clientes</code> , <code>/clientes/*</code> , <code>/pedidos</code> , <code>/pedidos/*</code>
P003	tecnico	<code>/pedidos</code> , <code>/pedidos/novo</code> , <code>/pedidos/[id]</code>

### 2.3 seed script

- Script `prisma/seed.ts` que cria:
  - 3 perfis se não existirem
  - Permissões base
  - Admin inicial

---

## 3. Schema Prisma (utilizadores)

```
``prisma
```

```

model User {
  id    String  @id @default(cuid())
  email String  @unique
  password String // hashed bcrypt
  name   String
  active Boolean @default(true)
  profileId Int
  profile Profile @relation(fields: [profileId], references: [id])
  createdAt DateTime @default(now())
  updatedAt DateTime @updatedAt
}

model Profile {
  id    Int    @id @default(autoincrement())
  name  String  @unique
  description String?
  permissions Permission[]
  users  User[]
}

model Permission {
  id    Int    @id @default(autoincrement())
  profileId Int
  resource String // e.g. "clientes", "utilizadores", "admin"
  actions String // e.g. "read,write,delete"
  profile Profile @relation(fields: [profileId], references: [id])
}
...
---
```

## 4. CRUD de Utilizadores

### ### 4.1 Listar Utilizadores

- Endpoint: `GET /api/utilizadores`
- Return: array de users (sem password)
- Filtros: `?ativo=true/false`, `?perfilId=1`
- Apenas **admin** pode ver a lista

### ### 4.2 Criar Utilizador

- Endpoint: `POST /api/utilizadores`
- Body: `{ email, password, name, profileId }`
- Validações: email único, password mínimo 8 chars, perfil existe
- Apenas **admin**

### ### 4.3 Editar Utilizador

- Endpoint: `PATCH /api/utilizadores/[id]`
- Body: `{ name?, profileId?, active? }`
- Não permite editar email ou password aqui (futuro: reset password)
- Apenas **admin**

### ### 4.4 Ativar/Desativar

- Via `PATCH /api/utilizadores/[id]` com `{ active: false }`
- Utilizadores desativados não fazem login

### ### 4.5 Eliminar

- Endpoint: `DELETE /api/utilizadores/[id]`
- Soft delete: `active = false` (reversível)
- Apenas **admin**

---

## 5. Páginas UI (Next.js App Router)

...

```
/login                ! LoginPage
```

```
/admin                !' Dashboard (protegido admin)
/admin/usuarios       !' Lista de utilizadores (protegido admin)
/admin/usuarios/novo  !' Criar utilizador (protegido admin)
/admin/usuarios/[id] !' Editar utilizador (protegido admin)
/admin/clientes       !' Lista de clientes (protegido escritorio+admin)
/admin/clientes/[id] !' Detalhe/edit cliente
```

## 6. Fluxo de Protecção por Perfil

Login OK

```
! admin ! /admin/dashboard
! escritorio ! /admin/clientes
! tecnico ! /admin/pedidos
```

Middleware:

```
/admin/*           !' requer sessão + role !== tecnico (opcionalmente)
/admin/usuarios/* !' requer role === admin
```

## 7. Dependências a Instalar

```
``bash
npm install next-auth@beta @auth/prisma-adapter bcryptjs
npm install -D prisma
npx prisma init
```

## 8. Ordem de Implementação

1. `prisma/schema.prisma` - definir modelos User, Profile, Permission
2. `prisma/seed.ts` - criar perfis + admin inicial
3. Configurar NextAuth - `auth.ts` / `[...nextauth]/route.ts`
4. Middleware protecção - `middleware.ts`
5. API Routes - CRUD utilizadores (`/api/usuarios`)
6. UI Pages - login, lista, criar, editar
7. Teste end-to-end - login admin, criar user, login novo user

## 9. Notas

- Passwords guardadas com `bcrypt` (salt rounds: 12)
- A password do admin inicial gerada no seed deve ser alterada no primeiro login
- Todas as APIs devolvem JSON; erros com `{ error: "mensagem" }`
- NÃO guardar passwords em claro - nunca

## Checklists de Caldeiras para APP

Resumo otimizado do Excel recebido com checklists para introduzir na APP.

Decisão de documentação: o PDF mantém apenas a visão consolidada e uma amostra representativa; o detalhe completo deve ficar no ficheiro Excel original/extrato técnico, para evitar duplicação e manter o PDF legível.

Folhas incluídas: 10

Linhas úteis totais no extrato: 246

## Excel - 00\_Fluxo\_Geral

Folha: 00\_Fluxo\_Geral

Conteúdo: FLUXO GERAL PARA APP - CHECKLISTS CALDEIRAS

Linhas úteis no ficheiro original: 9

Registos/campos identificados: 7

Regras/alertas visíveis: Gera recomendações e alertas

Amostra representativa:

- 1 | Escolher equipamento | Gás | Biomassa | Filtra checklists disponíveis | Campo: equipment\_type | Alta
- 2 | Escolher tipo de intervenção | Diagnóstico em funcionamento | Manutenção com equipamento parado | Carrega checklist correta | Campo: intervention\_mode | Alta
- 3 | Preencher identificação | Cliente, local, técnico, data, nº intervenção | Cabeçalho comum a todas as checklists | Usar campos meta comuns | Alta

## Excel - 01\_App\_Schema

Folha: 01\_App\_Schema

Conteúdo: APP SCHEMA - CAMPOS PARA REPLIT / BASE DE DADOS

Linhas úteis no ficheiro original: 83

Registos/campos identificados: 81

Ecrãs/áreas principais: geral, Gás, Biomassa

Campos técnicos detetados: 81

Regras/alertas visíveis: Normal, Se NOK: bloquear conclusão como operacional., Crítico, Alerta

Amostra representativa:

- geral | todos | 0 | Identificação | 1 | intervention\_ref
- geral | todos | 0 | Identificação | 2 | client\_name
- geral | todos | 0 | Identificação | 3 | site\_location

## Excel - 02\_Ecras\_App

Folha: 02\_Ecras\_App

Conteúdo: ECRÃS / FLUXO DE NAVEGAÇÃO

Linhas úteis no ficheiro original: 36

Registos/campos identificados: 34

Ecrãs/áreas principais: Biomassa, Gás

Regras/alertas visíveis: Indícios de acumulação de cinza no corpo de pressão? Escória na grelha dentro do normal diário?, Detetar riscos críticos no tratamento de fumos., Filtro de mangas a funcionar normalmente? Sinais de sobreaquecimento/incêndio no filtro de mangas? Sinais de cúpula/acumulação no ciclone?, Verificar o principal ponto crítico em biomassa.

Amostra representativa:

- Biomassa | Diagnóstico em funcionamento | 1 | Segurança | Garantir que o técnico pode avançar. | Situação segura para diagnóstico visual?
- Biomassa | Diagnóstico em funcionamento | 2 | Combustível | Preencher verificações rápidas deste bloco. | Combustível em uso
- Biomassa | Diagnóstico em funcionamento | 3 | Combustão | Preencher verificações rápidas deste bloco. | Combustão estável? Temperatura fornalha, se disponível

## Excel - 03\_Gas\_Diagnostico

Folha: 03\_Gas\_Diagnostico

Conteúdo: GÁS - DIAGNÓSTICO EM FUNCIONAMENTO

Linhas úteis no ficheiro original: 12

Registos/campos identificados: 10

Ecrãs/áreas principais: Segurança, Funcionamento, Medições, Fumos/ambiente, Conclusão

Campos técnicos detetados: 10

Regras/alertas visíveis: Se NOK: bloquear conclusão como operacional., Se muito alto: alerta de segurança/afinação.

Amostra representativa:

- Segurança | 1 | gas\_diag\_safe | Situação segura para diagnóstico visual? | select | OK|NOK|NA
- Funcionamento | 2 | gas\_diag\_flame\_stable | Chama estável? | select | OK|NOK|NA
- Funcionamento | 3 | gas\_diag\_start\_stop\_ok | Arranque e paragem normais? | select | OK|NOK|NA

## Excel - 04\_Gas\_Manutencao

Folha: 04\_Gas\_Manutencao

Conteúdo: GÁS - MANUTENÇÃO COM EQUIPAMENTO PARADO

Linhas úteis no ficheiro original: 24

Registos/campos identificados: 22

Ecrãs/áreas principais: Segurança, Inspeção, Limpeza, Queimador, Seguranças/controlo, Funcionamento, Gases, Conclusão

Campos técnicos detetados: 22

Regras/alertas visíveis: Se NOK: alerta., gas\_maint\_soot\_abnormal, Fuligem/deposição anormal?, Se muito alto: alerta.

Amostra representativa:

- Segurança | 1 | gas\_maint\_safe | Equipamento seguro para manutenção? | select | OK|NOK|NA
- Segurança | 2 | gas\_maint\_no\_leaks\_before | Sem fugas de gás aparentes antes da intervenção? | select | OK|NOK|NA
- Inspeção | 3 | gas\_maint\_general\_state | Estado geral da caldeira aceitável? | select | OK|NOK|NA

## Excel - 05\_Biomassa\_Diagnostico

Folha: 05\_Biomassa\_Diagnostico

Conteúdo: BIOMASSA - DIAGNÓSTICO EM FUNCIONAMENTO

Linhas úteis no ficheiro original: 19

Registos/campos identificados: 17

Ecrãs/áreas principais: Segurança, Combustível, Combustão, Medições, Alimentação, Cinzas, Filtro/ciclone, Resposta à carga

Campos técnicos detetados: 17

Regras/alertas visíveis: Normalmente estável; registar exceções., Escória na grelha dentro do normal diário?, A manutenção diária é normalmente do operador do cliente., Filtro de mangas a funcionar normalmente?

Amostra representativa:

- Segurança | 1 | bio\_diag\_safe | Situação segura para diagnóstico visual? | select | OK|NOK|NA
- Combustível | 2 | bio\_diag\_fuel | Combustível em uso | select | PKS|Pellets|Estilha|Caroço de azeitona|Casca de amêndoa|Caroço de açaí|Outro
- Combustão | 3 | bio\_diag\_combustion\_stable | Combustão estável? | select | OK|NOK|NA

## Excel - 06\_Biomassa\_Manutencao

Folha: 06\_Biomassa\_Manutencao

Conteúdo: BIOMASSA - MANUTENÇÃO COM EQUIPAMENTO PARADO

Linhas úteis no ficheiro original: 26

Registos/campos identificados: 24

Ecrãs/áreas principais: Segurança, Combustível, Limpeza corpo pressão, Grelha/fornalha, Alimentação, Ar/EGR, Ciclone, Filtro de mangas

Campos técnicos detetados: 24

Regras/alertas visíveis: Baixa|Normal|Elevada|Crítica, Quando bem cozido, normalmente dura 5-10 anos., Se Sim: alerta crítico e recomendar substituição/intervenção., À saída, normalmente cerca de 100-110 °C.

Amostra representativa:

- Segurança | 1 | bio\_maint\_safe | Equipamento seguro para manutenção? | select | OK|NOK|NA
- Combustível | 2 | bio\_maint\_fuel | Combustível principal usado | select | PKS|Pellets|Estilha|Caroço de azeitona|Casca de amêndoa|Caroço de açaí|Outro
- Limpeza corpo pressão | 3 | bio\_maint\_pressure\_body\_clean | Limpeza do corpo de pressão efetuada? | select | OK|NOK|NA

## Excel - 07\_Logica\_Alertas

Folha: 07\_Logica\_Alertas

Conteúdo: LÓGICA SIMPLES DE ALERTAS

Linhas úteis no ficheiro original: 12

Registos/campos identificados: 10

Ecrãs/áreas principais: R001, R002, R003, R004, R005, R006, R007, R008

Regras/alertas visíveis: Campo crítico = NOK, Crítico, Mostrar alerta de segurança/afinação e pedir observação., Alerta

Amostra representativa:

- R001 | Todos | Campo crítico = NOK | Não permitir estado final 'Operacional' sem observação/recomendação. | Crítico
- R002 | Gás | CO elevado | Mostrar alerta de segurança/afinação e pedir observação. | Crítico
- R003 | Gás diagnóstico | Chama instável OU arranque/paragem NOK | Sugerir manutenção/intervenção. | Alerta

## Excel - 08\_Opcoes

Folha: 08\_Opcoes

Conteúdo: OPÇÕES / ENUMS

Linhas úteis no ficheiro original: 11

Registos/campos identificados: 9

Ecrãs/áreas principais: OK\_NOK\_NA, SIM\_NAO, SIM\_NAO\_NA, ESTADO, EQUIPAMENTO, MODO, COMB\_GAS, COMB\_BIOMASSA

Campos técnicos detetados: 5

Regras/alertas visíveis: Baixa|Normal|Alta|Urgente

Amostra representativa:

- OK\_NOK\_NA | OK|NOK|NA | Usar no campo options
- SIM\_NAO | Sim|Não | Usar no campo options
- SIM\_NAO\_NA | Sim|Não|NA | Usar no campo options

## Excel - 09\_Notas\_Tecnicas

Folha: 09\_Notas\_Tecnicas

Conteúdo: NOTAS TÉCNICAS E ASSUNÇÕES

Linhas úteis no ficheiro original: 14

Registos/campos identificados: 12

Ecrãs/áreas principais: Conceito principal, Gás diagnóstico, Gás manutenção, Biomassa diagnóstico, Biomassa manutenção, Biomassa combustíveis, Biomassa temperaturas, Biomassa controlo

Regras/alertas visíveis: Risco crítico: filtro arder.

Amostra representativa:

- Conceito principal | Separar diagnóstico em funcionamento de manutenção com equipamento parado. | Evita checklists densas e campos sem sentido.
- Gás diagnóstico | Pensado para inspeção rápida sem desmontagem. | Equipamento em funcionamento.
- Gás manutenção | Pensado para equipamento parado, com limpeza, testes e análise de combustão. | Base v3 simplificada.