

Radar Inteligentes

Radares integrados ao do netservice.

- [Radar Inteligente de Incidentes](#)

Radar Inteligente de Incidentes

📄 Objetivo

Garantir **visibilidade, antecipação de riscos e direcionamento de ação** sobre incidentes, de forma simples, contínua e baseada em dados.

📄 O Radar permite identificar rapidamente:

- Incidentes críticos
 - Gargalos operacionais
 - Falta de atuação
 - Problemas recorrentes (ITIL)
 - Necessidade de priorização
-

📄 O que é o Radar Inteligente

O **Radar Inteligente de Incidentes** é um modelo estruturado de acompanhamento que utiliza:

- Base unificada de incidentes (NetService)
- Análise automatizada (GPT)
- Dashboard executivo
- Direcionamento semanal

📄 Não é apenas um relatório — é um **mecanismo de gestão operacional e tomada de decisão**.

⚙️ Como funciona

1. Extração da base unificada de incidentes
2. Validação automática dos dados (Modo Auditor)
3. Análise inteligente (GPT)
4. Geração do Radar (visão executiva)

☐☐ Fonte de Dados

A base do Radar é:

☐☐ Radar Inteligente — Base Unificada de Incidentes

Essa base contém:

- ICs abertos e fechados
- Datas (abertura, encerramento, movimentação)
- Classificação automática (Crítico, Atenção, Controlado)
- Tempo por departamento
- Status atual
- Título e descrição (para análise de problema ITIL)

☐☐ Princípio fundamental: CSV gerado a partir dessa base é a única fonte de verdade.

☐☐ Validação de Dados (Modo Auditor)

Antes de qualquer análise, os dados passam por validação automática:

- Conferência de totais
- Consistência de datas
- Verificação de campos obrigatórios
- Identificação de inconsistências

Classificação da Qualidade

- ☐☐ Aprovado → Pode ser apresentado
- ☐☐ Aprovado com ressalvas → Atenção
- ☐☐ Reprovado → NÃO deve ser utilizado

☐☐ Se reprovado, o Radar não deve ser gerado.

☐☐ Estrutura do Radar

O Radar é composto por:

1. Resumo Executivo

Visão clara do cenário atual (máx. 5 linhas)

2. Situação Atual

- Total de ICs
 - ICs críticos
 - ICs em atenção
 - ICs sem movimentação
 - ICs muito antigos
-

3. Análise por Time

- Concentração de incidentes
 - Times com maior risco
 - Pontos de atenção
-

4. Gargalos por Departamento

- Atendimento
- Análise
- Infraestrutura e Banco de Dados
- Outros

Identificação de onde o tempo está sendo consumido

5. Evolução

- Volume resolvido
 - Tempo médio
 - Tendência (melhora ou piora)
-

6. Recorrência e Padrões

- Identificação de repetição de incidentes
 - Agrupamento inicial por similaridade
-

7. Problema ITIL (CRÍTICO)

Validação real de problemas com base em:

- Título
- **Descrição do incidente (obrigatório)**

Regra:

Título parecido NÃO define problema

A descrição deve confirmar:

- ✓ mesmo erro
- ✓ mesmo comportamento
- ✓ mesma falha

Classificação:

- ✓ Mesmo problema provável
- ⚠ Semelhança parcial
- Incidentes diferentes

Apenas problemas confirmados devem ser considerados

8. Direcionamento da Semana

Ações práticas para atuação:

- Priorizar ICs críticos
 - Atuar ICs sem movimentação
 - Reduzir gargalos
 - Avaliar abertura de Problema ITIL
-

Dashboard

O Radar gera um dashboard executivo contendo:

- KPIs principais
- ICs por Time
- Classificação
- Gargalos
- Ranking

Regras:

- Baseado 100% nos dados validados
 - Sem ajustes manuais
 - Sem estimativas
-

📄 Histórico

O Radar mantém evolução ao longo do tempo:

Data | ICs Abertos | Críticos | Atenção | Sem Movimentação | Muito Antigos | Tempo Atendimento | Tempo Análise | Tempo Infra | Resolvidos | Tempo Médio

📄 Utilizado para:

- Acompanhar evolução
 - Medir eficiência
 - Apresentação para diretoria
-

📄 Papéis e Responsabilidades

Scrum Master

- Gerar o Radar
 - Garantir visibilidade
 - Identificar riscos
 - Direcionar ações
-

Product Owner / Supervisor

- Priorizar ICs
 - Atuar sobre os incidentes
 - Garantir resposta e andamento
-

Times

- Executar correções
 - Resolver incidentes
 - Apoiar análise
-

⚠️ Regras do Processo

- ✓ Não é relatório → é direcionamento
- ✓ Não é cobrança → é gestão de risco
- ✓ Não é volume → é ação baseada em dados
- ✓ Não é opinião → é evidência

☐ Frequência

- Semanal (obrigatório)
- Pontual (quando necessário)

☐☐ Princípio

Dados → Validação → Análise → Insight → Ação

☐☐ Resultado Esperado

Após algumas semanas de uso:

- Redução de ICs críticos
- Redução de ICs sem movimentação
- Maior atuação dos times
- Identificação de problemas estruturais
- Melhor tomada de decisão

☐☐ Conclusão

O Radar Inteligente de Incidentes transforma dados operacionais em:

- ✓ Visão clara
- ✓ Decisão rápida
- ✓ Ação direcionada
- ✓ Gestão estruturada

☐ Tornando-se um **instrumento essencial de gestão da operação**.

- Após conectar o banco do MySQL do Netservice via ODBC no Codex.
- Utilize a Query abaixo para ele consultar os dados necessários e extrairmos as informações via CSV.
- Continua o processo através do GPT Radar Inteligente de Incidentes

Query - Base Unificada de Incidentes

```
SELECT
  t.TarefaID AS IC,
  t.Titulo AS Titulo,
  t.Descricao AS DescricaoIncidente,

  e.NomeEmpresa AS Cliente,
  s.NomeSistema AS TimeSistema,
  m.NomeModulo AS Produto,
  a.NomeAtividade AS Servico,

  u.NomeUsuario AS ResponsavelAtual,
  COALESCE(st.NomeSituacao, 'Sem status') AS StatusAtual,

  t.DataCriacao AS DataAbertura,
  t.DataEncerramento AS DataEncerramento,

  CASE
    WHEN t.DataEncerramento IS NULL THEN 'Aberto'
    ELSE 'Fechado'
  END AS SituacaoIC,

  DATEDIFF(
    COALESCE(t.DataEncerramento, CURDATE()),
    t.DataCriacao
  ) AS DiasDesdeAbertura,

  CASE
    WHEN t.DataEncerramento IS NOT NULL THEN
      DATEDIFF(t.DataEncerramento, t.DataCriacao)
    ELSE NULL
  END AS DiasParaEncerrar,
```

Im.UltimaMovimentacao AS UltimaMovimentacao,

DATEDIFF(

COALESCE(t.DataEncerramento, CURDATE()),

COALESCE(Im.UltimaMovimentacao, t.DataCriacao)

) AS DiasSemMovimentacao,

CASE

WHEN t.DataEncerramento IS NULL

AND DATEDIFF(CURDATE(), t.DataCriacao) >= 15 THEN 'Crítico'

WHEN t.DataEncerramento IS NULL

AND DATEDIFF(CURDATE(), t.DataCriacao) BETWEEN 12 AND 14 THEN 'Atenção'

WHEN t.DataEncerramento IS NULL THEN 'Controlado'

ELSE 'Encerrado'

END AS ClassificacaoRadar,

CASE

WHEN t.DataEncerramento IS NULL

AND DATEDIFF(

CURDATE(),

COALESCE(Im.UltimaMovimentacao, t.DataCriacao)

) >= 5 THEN 'Sim'

ELSE 'Não'

END AS FlagSemMovimentacao,

CASE

WHEN t.DataEncerramento IS NULL

AND DATEDIFF(CURDATE(), t.DataCriacao) >= 30 THEN 'Sim'

ELSE 'Não'

END AS FlagMuitoAntigo,

CASE

WHEN st.NomeSituacao IS NULL OR st.NomeSituacao = '' THEN 'Sim'

ELSE 'Não'

END AS FlagStatusAusente,

COALESCE(td.DiasAtendimento, 0) AS DiasAtendimento,

```
COALESCE(td.DiasAnalise, 0) AS DiasAnalise,  
COALESCE(td.DiasInfraestruturaBancoDados, 0) AS DiasInfraestruturaBancoDados,  
COALESCE(td.DiasOutrosDepartamentos, 0) AS DiasOutrosDepartamentos
```

```
FROM netservice.tarefas t
```

```
LEFT JOIN netservice.empresas e  
  ON e.EmpresaID = t.EmpresaID
```

```
LEFT JOIN netservice.modulos m  
  ON m.ModuloID = t.ModuloID
```

```
LEFT JOIN netservice.sistemas s  
  ON s.SistemaID = m.SistemaID
```

```
LEFT JOIN netservice.atividades a  
  ON a.AtividadeID = t.AtividadeID
```

```
LEFT JOIN netservice.usuarios u  
  ON u.UsuarioID = t.ResponsavelID
```

```
LEFT JOIN netservice.situacoestarefas st  
  ON st.SituacaoTarefaID = t.SituacaoTarefaID
```

```
LEFT JOIN (  
  SELECT  
    TarefaID,  
    MAX(DataAtribuicao) AS UltimaMovimentacao  
  FROM netservice.andamentostarefas  
  GROUP BY TarefaID  
) Im  
  ON Im.TarefaID = t.TarefaID
```

```
LEFT JOIN (  
  SELECT  
    mov.TarefaID,  
  
    SUM(CASE  
      WHEN dep.DepartamentoID = 3 THEN mov.DiasNoDepartamento  
      ELSE 0  
    END) AS DiasAtendimento,
```

```
SUM(CASE
  WHEN dep.DepartamentoID = 1 THEN mov.DiasNoDepartamento
  ELSE 0
END) AS DiasAnalise,
```

```
SUM(CASE
  WHEN dep.DepartamentoID = 6 THEN mov.DiasNoDepartamento
  ELSE 0
END) AS DiasInfraestruturaBancoDados,
```

```
SUM(CASE
  WHEN dep.DepartamentoID NOT IN (1, 3, 6)
  OR dep.DepartamentoID IS NULL
  THEN mov.DiasNoDepartamento
  ELSE 0
END) AS DiasOutrosDepartamentos
```

```
FROM (
  SELECT
    atf.AndamentoTarefaID,
    atf.TarefaID,
    atf.DestinoID,

    DATEDIFF(
      COALESCE((
        SELECT MIN(atf2.DataAtribuicao)
        FROM netservice.andamentostarefas atf2
        WHERE atf2.TarefaID = atf.TarefaID
        AND (
          atf2.DataAtribuicao > atf.DataAtribuicao
          OR (
            atf2.DataAtribuicao = atf.DataAtribuicao
            AND atf2.AndamentoTarefaID > atf.AndamentoTarefaID
          )
        )
      ), COALESCE(tf.DataEncerramento, NOW())),
    atf.DataAtribuicao
  ) AS DiasNoDepartamento
```

```
FROM netservice.andamentostarefas atf
```

```
INNER JOIN netservice.tarefas tf
  ON tf.TarefaID = atf.TarefaID
  AND tf.Tipo = 4
) mov

LEFT JOIN netservice.usuarios ud
  ON ud.UsuarioID = mov.DestinoID

LEFT JOIN netservice.departamentos dep
  ON dep.DepartamentoID = ud.DepartamentoID

GROUP BY mov.TarefaID
) td
  ON td.TarefaID = t.TarefaID

WHERE
  t.Tipo = 4

ORDER BY
  SituacaoI ASC,
  DiasDesdeAbertura DESC,
  DiasSemMovimentacao DESC;
```

OBSERVAÇÃO: GERAR AQUIVO COM NOME: Base Unificada de Incidentes