

Dashboard Migration Guide

FASE B.2 - Consolidação de Dashboards



Overview

Este guia explica como migrar dashboards existentes para usar os novos componentes base (`ExecutiveDashboardBase` e `AnalyticsDashboardBase`).



Objetivos da Migração

1. **Reduzir duplicação:** Consolidar ~14.700 linhas de código duplicado
 2. **Padronizar interface:** Garantir consistência visual entre dashboards
 3. **Facilitar manutenção:** Atualizações em um único lugar
 4. **Melhorar performance:** Lazy loading e otimizações automáticas
 5. **Type safety:** 100% TypeScript com validação de tipos
-



Estrutura dos Componentes Base

Componentes Disponíveis

```
// Base Components
import {
  ExecutiveDashboardBase,
  AnalyticsDashboardBase
} from "@components/dashboard-base";

// Widgets
import {
  KPICard,
  ChartWidget,
  MetricIndicator,
  TableWidget,
  FilterPanel
} from "@components/dashboard-base";

// Hooks
import {
  useDashboardData,
  useDashboardFilters,
  useDashboardExport
} from "@components/dashboard-base";

// Types
import type {
  ExecutiveDashboardConfig,
  AnalyticsDashboardConfig,
  WidgetConfig,
  KPIConfig,
  ChartConfig
} from "@components/dashboard-base";
```

Processo de Migração

Passo 1: Criar Arquivo de Configuração

```
// src/components/dashboard/configs/my-dashboard.config.ts
import { ExecutiveDashboardConfig } from "@types/dashboard-config";
import { Ship, TrendingUp, Users } from "lucide-react";

export const myDashboardConfig: ExecutiveDashboardConfig = {
  id: "my-dashboard",
  title: "Meu Dashboard",
  description: "Descrição do dashboard",

  layout: {
    type: "grid",
    columns: 12,
    gap: 16,
  },

  widgets: [
    {
      id: "kpi-1",
      type: "kpi",
      colspan: 3,
      config: {
        id: "revenue",
        title: "Receita",
        value: "R$ 125.000",
        change: 8.7,
        icon: TrendingUp,
      },
    },
    // ... mais widgets
  ],
};
```

Passo 2: Usar o Componente Base

```
// src/components/dashboard/my-dashboard.tsx
import { ExecutiveDashboardBase } from "@components/dashboard-base";
import { myDashboardConfig } from "../configs/my-dashboard.config";

export const MyDashboard = () => {
  return <ExecutiveDashboardBase config={myDashboardConfig} />;
};
```

Passo 3: Marcar Dashboard Antigo como Deprecated

```
/**
 * @deprecated Use MyDashboard with ExecutiveDashboardBase instead
 * Will be removed in v3.0.0
 */
export const OldDashboard = () => {
  // ... código antigo
};
```



Tipos de Widgets Suportados

1. KPI Card

```
{
  id: "kpi-example",
  type: "kpi",
  colspan: 3,
  config: {
    id: "metric-id",
    title: "Título do KPI",
    value: "1.234",
    change: 12.5,
    trend: "up",
    icon: TrendingUp,
    prefix: "R$ ",
    suffix: "%",
    target: 1500,
    description: "vs mês anterior",
  },
}
```

2. Chart Widget

```
{
  id: "chart-example",
  type: "chart",
  colspan: 8,
  config: {
    id: "revenue-chart",
    type: "line", // "line" | "bar" | "area" | "pie" | "donut"
    title: "Tendência de Receita",
    data: [
      { month: "Jan", revenue: 42000 },
      { month: "Fev", revenue: 48000 },
    ],
    dataKeys: ["revenue"],
    xAxisKey: "month",
    colors: ["#3b82f6"],
    height: 300,
  },
}
```

3. Metric Indicator

```
{
  id: "metric-example",
  type: "metric",
  colspan: 4,
  config: {
    id: "efficiency",
    label: "Eficiência",
    value: 87.5,
    target: 90,
    unit: "%",
    format: "percentage",
  },
}
```

4. Table Widget

```
{
  id: "table-example",
  type: "table",
  colspan: 12,
  config: {
    title: "Performance da Frota",
    columns: [
      { id: "name", label: "Nome", width: "40%" },
      { id: "value", label: "Valor", width: "30%", align: "center" },
    ],
    data: [
      { name: "Item 1", value: "100" },
      { name: "Item 2", value: "200" },
    ],
  },
}
```

Configuração Avançada

Tabs

```
tabs: [  
  {  
    id: "overview",  
    label: "Visão Geral",  
    icon: Activity,  
    widgets: [  
      // Widgets específicos para esta tab  
    ],  
  },  
  {  
    id: "details",  
    label: "Detalhes",  
    icon: BarChart3,  
    widgets: [  
      // Widgets específicos para esta tab  
    ],  
  },  
]
```

Filtros

```
filters: [  
  {  
    id: "period",  
    type: "select",  
    label: "Período",  
    options: [  
      { value: "today", label: "Hoje" },  
      { value: "week", label: "Esta Semana" },  
    ],  
    defaultValue: "week",  
  },  
  {  
    id: "search",  
    type: "search",  
    label: "Buscar",  
    placeholder: "Digite para buscar...",  
  },  
]
```

Data Source (Supabase)

```
dataSource: {  
  type: "supabase",  
  endpoint: "fleet_metrics",  
  refreshInterval: 60000, // 1 minuto  
  transform: (data) => {  
    // Transformar dados antes de exibir  
    return data.map(item => ({  
      ...item,  
      formatted: `R$ ${item.value}`,  
    }));  
  },  
}
```

Ações Customizadas

```
actions: [  
  {  
    id: "export",  
    label: "Exportar",  
    icon: Download,  
    variant: "outline",  
    onClick: async () => {  
      // Lógica de exportação  
    },  
  },  
]
```



Analytics Dashboard Specific Features

Time Range Selector

```
timeRanges: ["7d", "30d", "90d", "1y"],  
defaultTimeRange: "30d",
```

Export Options

```
exportFormats: ["csv", "json", "pdf", "excel"],
```

Real-time Updates

```
realtimeEnabled: true,  
dataSource: {  
  type: "realtime",  
  endpoint: "metrics",  
}
```

Hooks Customizados

useDashboardData

```
const { data, isLoading, error, refresh } = useDashboardData({
  dataSource: {
    type: "supabase",
    endpoint: "metrics",
  },
  autoRefresh: true,
});
```

useDashboardFilters

```
const { filterValues, setFilter, resetFilters } = useDashboardFilters({
  filters: myFilters,
  onFilterChange: (values) => {
    // Handle filter changes
  },
});
```

useDashboardExport

```
const { isExporting, exportData } = useDashboardExport();

const handleExport = () => {
  exportData({
    data: myData,
    format: "csv",
    filename: "dashboard-export",
  });
};
```

Checklist de Migração

- ☐ Criar arquivo de configuração
 - ☐ Testar todos os widgets
 - ☐ Verificar filtros funcionando
 - ☐ Validar data source
 - ☐ Testar exportação (se aplicável)
 - ☐ Marcar dashboard antigo como deprecated
 - ☐ Adicionar warning de deprecation
 - ☐ Executar build sem erros
 - ☐ Testar no navegador
 - ☐ Atualizar documentação
-



Exemplos Completos

Veja os arquivos de exemplo:

- `src/components/dashboard-base/configs/executive-dashboard-example.config.ts`
 - `src/components/dashboard-base/configs/analytics-dashboard-example.config.ts`
 - `src/components/dashboard/executive-dashboard-new.tsx`
 - `src/components/analytics/analytics-dashboard-new.tsx`
-



Breaking Changes

Nenhuma breaking change nesta versão. Dashboards antigos continuam funcionando com warnings de deprecation.



Suporte

Para dúvidas ou problemas, consulte:

- `CHANGELOG_FASE_B2_DASHBOARDS.md`
- `src/types/dashboard-config.ts` (tipos completos)
- Exemplos de configuração em `src/components/dashboard-base/configs/`