

MULTIMONEY - Sistema de Gestión de Proyectos Multinacional

Estructura del Proyecto

```
src/
├── components/
│   ├── MultimoneySystem.jsx ✓ # Componente principal del sistema
│   └── common/
│       └── TicketViewer.jsx ✓ # Componente para visualizar tickets
├── modules/
│   ├── DashboardModule.jsx ✓ # Módulo de dashboard con KPIs
│   ├── EpicasModule.jsx ✓ # Módulo de análisis de épicas
│   ├── TimelineModule.jsx # Módulo de vista temporal (por implementar)
│   ├── RisksModule.jsx # Módulo de análisis de riesgos (por implementar)
│   └── FinalizadosModule.jsx # Módulo de proyectos finalizados (por implementar)
├── utils/
│   ├── constants.js ✓ # Constantes del sistema
│   ├── dataProcessing.js ✓ # Funciones de procesamiento de datos
│   └── scoringUtils.js ✓ # Utilidades de scoring y análisis
└── styles/
    └── multimoney.css ✓ # Estilos específicos de MULTIMONEY
```

Instalación y Configuración

1. Dependencias Requeridas

```
bash

npm install react lucide-react papaparse
```

2. Configuración de Tailwind CSS

Asegúrate de tener Tailwind CSS configurado en tu proyecto. Si no lo tienes:

```
bash

npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

3. Importar en tu aplicación principal

```
javascript

// App.js
import React from 'react';
import MultimoneySystem from './src/components/MultimoneySystem.jsx';
import './src/styles/multimoney.css';

function App() {
  return (
    <div className="App">
      <MultimoneySystem />
    </div>
  );
}

export default App;
```

Módulos Implementados

Dashboard Module

- Ubicación: `src/modules/DashboardModule.jsx`
- Características:
 - 6 KPIs principales
 - Matriz regional de proyectos
 - Distribución por estados
 - Top 8 épicas por volumen
 - Métricas de eficiencia y riesgo

Épicas Module

- Ubicación: `src/modules/EpicasModule.jsx`
- Características:
 - Análisis de salud con fórmula weighted (60% completados + 25% en proceso + 15% otros)
 - Drill-down clickeable por categorías de estado
 - Modal de detalles con información completa
 - Indicadores de salud (Excelente, Buena, Regular, Crítica)

Módulos Pendientes

Timeline Module

```
javascript

// src/modules/TimelineModule.jsx
import React from 'react';
import { Clock } from 'lucide-react';
import { filterData } from '../utils/dataProcessing.js';

const TimelineModule = ({ data, filters }) => {
  // Implementar vista cronológica de 9 meses
  // Mostrar métricas por período
  // Análisis de velocidad y eficiencia

  return (
    <div className="space-y-8">
      { /* Tu implementación aquí */ }
    </div>
  );
};

export default TimelineModule;
```

Risks Module

```
javascript
```

```
// src/modules/RisksModule.jsx
import React from 'react';
import { AlertTriangle } from 'lucide-react';
import { analyzeEpicRisks, analyzeCountryRisks } from '../utils/scoringUtils.js';

const RisksModule = ({ data, filters }) => {
  // Excluir proyectos finalizados del análisis
  const activeData = data.filter(item =>
    item.state !== 'Finalizado' && item.state !== 'Cancelado'
  );

  // Implementar análisis de riesgo
  // Proyectos bloqueados, sin actualizar, alta prioridad

  return (
    <div className="space-y-8">
      /* Tu implementación aquí */
    </div>
  );
};

export default RisksModule;
```

Finalizados Module

```
javascript

// src/modules/FinalizadosModule.jsx
import React from 'react';
import { CheckCircle } from 'lucide-react';

const FinalizadosModule = ({ data, filters }) => {
  // Filtrar solo proyectos finalizados
  const completedData = data.filter(item => item.state === 'Finalizado');

  // Implementar análisis de proyectos completados
  // Tiempo de completado, tendencias, análisis por épica/país

  return (
    <div className="space-y-8">
      /* Tu implementación aquí */
    </div>
  );
};

export default FinalizadosModule;
```



Estructura de Datos

Formato CSV Esperado

csv

Clave del Ticket,Resumen,Estado,País,Épica,Persona Asignada,Responsable Dev,Fecha de Creación,Fe
MM-001,Implementar login,Finalizado,Guatemala,Autenticación,Juan Pérez,Ana García,01/01/2024,15/0
MM-002,API de usuarios,En Proceso,Costa Rica,Backend Core,María López,Carlos Ruiz,02/01/2024,20

Objeto de Datos Procesado

javascript

```
{
  ticketKey: 'MM-001',
  summary: 'Implementar login',
  state: 'Finalizado',
  country: 'Guatemala',
  epic: 'Autenticación',
  assignee: 'Juan Pérez',
  devResponsible: 'Ana García',
  createdAt: Date,
  updatedAt: Date,
  businessPriority: 8,
  techPriority: 7,
  sizing: 3,
  businessArea: 'Frontend',
  originalRow: { /* datos originales del CSV */ }
}
```

Sistema de Filtros

Filtros Globales

- **País/Región:** Filtro por países de COUNTRIES array
- **Épica:** Filtro por épicas disponibles
- **Finalizados desde:** Filtro por fecha para proyectos completados

Filtros Contextuales

- **Timeline desde:** Solo disponible en módulo Timeline
- Otros filtros específicos por módulo

Utilidades Disponibles

Data Processing (`utils/dataProcessing.js`)

- `processCSVData(csvData)`: Procesa datos del CSV
- `filterData(data, filters)`: Aplica filtros a los datos
- `calculateMetrics(data)`: Calcula métricas básicas
- `parseDate(dateStr)`: Convierte strings a fechas

Scoring Utils (`utils/scoringUtils.js`)

- `calculateEpicHealth(epicData)`: Calcula health score de épicas
- `calculateRiskMetrics(data)`: Analiza riesgos en proyectos activos
- `analyzeEpicRisks(data)`: Análisis de riesgo por épica
- `getHealthColor(score)`: Obtiene color según health score

Constants (`utils/constants.js`)

- `PROJECT_STATES`: Estados de proyecto con colores y prioridades
- `COUNTRIES`: Lista de países/regiones
- `SCORING_WEIGHTS`: Pesos para cálculos de scoring
- `RISK_THRESHOLDS`: Umbrales para análisis de riesgo

Cómo Agregar Nuevos Módulos

1. Crear el Módulo

```
javascript
```

```
// src/modules/MiNuevoModule.jsx
import React from 'react';
import { Icon } from 'lucide-react';
import { filterData } from '../utils/dataProcessing.js';

const MiNuevoModule = ({ data, filters }) => {
  const filteredData = filterData(data, filters);

  return (
    <div className="space-y-8">
      <h2 className="text-2xl font-bold">Mi Nuevo Módulo</h2>
      { /* Implementación */ }
    </div>
  );
};

export default MiNuevoModule;
```

2. Registrar en Sistema Principal

```
javascript

// En MultimoneySystem.jsx
import MiNuevoModule from '../modules/MiNuevoModule.jsx';

// Agregar al array de módulos
const modules = [
  { id: 'dashboard', name: 'Dashboard', icon: BarChart3, color: 'blue' },
  { id: 'epicas', name: 'Épicas', icon: Target, color: 'purple' },
  { id: 'mi-nuevo', name: 'Mi Nuevo', icon: Icon, color: 'indigo' }, // ← Nuevo
  // ...otros módulos
];

// Agregar renderizado condicional
{currentModule === 'mi-nuevo' && (
  <MiNuevoModule data={processedData} filters={filters} />
)}
```

Funcionalidades Principales

✅ Sistema de Carga

- Drag & Drop para archivos CSV
- Carga desde Google Drive con URLs configurables
- Validación automática de estructura
- Pantalla de carga con progreso paso a paso

✅ Procesamiento de Datos

- Parsing inteligente de fechas múltiples formatos
- Categorización automática de estados
- Cálculos en tiempo real de métricas
- Validación de estructura de archivos

✅ Sistema de Filtros

- Filtros globales compartidos entre módulos
- Filtros contextuales específicos por módulo
- Contadores dinámicos de elementos filtrados
- Botones de limpieza para cada filtro

✅ Exportación

- Exportación de datos filtrados a CSV
- Preservación de datos originales

🎨 Personalización de Estilos

Colores de Estado

```
css

/* En multimoney.css */
.text-status-completed { color: #10b981; }
.text-status-in-progress { color: #f59e0b; }
.text-status-blocked { color: #ef4444; }
```

Efectos Visuales

- Gradientes personalizados
- Animaciones suaves
- Efectos de hover
- Glassmorphism para modales

🔗 URLs de Configuración

Google Drive

```
javascript

// En constants.js
export const GOOGLE_DRIVE_URLS = [
  'https://drive.google.com/uc?id=TU_FILE_ID',
  'https://docs.google.com/spreadsheets/d/TU_SHEET_ID/export?format=csv'
];
```

Jira Integration

```
javascript

export const JIRA_BASE_URL = 'https://tu-instancia.atlassian.net/browse/';
```

🐛 Resolución de Problemas

Error: Módulo no encontrado

- Verificar rutas de importación
- Asegurar que los archivos estén en las ubicaciones correctas

CSV no se procesa

- Verificar que el CSV tenga los campos requeridos
- Comprobar formato de fechas
- Revisar encoding del archivo (UTF-8 recomendado)

Estilos no se aplican

- Importar `multimoney.css` en el componente principal
- Verificar configuración de Tailwind CSS

📞 Soporte

Para soporte técnico o preguntas sobre implementación, consultar:

- Documentación de constantes en `utils/constants.js`
- Ejemplos de uso en módulos existentes
- Funciones de utilidad en `utils/`

Desarrollado por el equipo MULTIMONEY 🚀