**📋 Módulos do Sistema**

**1. Cadastro de Usuários e Controle de Acesso**

* **Campos:** Nome, E-mail, Senha, Papel (Admin, P.O., Usuário Padrão, Visualizador)
* **Funcionalidades:**
  + Login com autenticação.
  + Controle de permissões: quem pode visualizar, editar, aprovar, etc.

**2. Cadastro de Projetos**

* **Campos Obrigatórios na Criação:**
  + Código do Projeto (PRJ-00000)
  + Nome/Descrição
  + Módulo/Sistema relacionado
  + Criticidade (Alta, Média, Baixa)
  + Data estimada pelo P.O. (Início/Fim)
* **Campos que podem ser atualizados ao longo do tempo:**
  + Data planejada pelo Time (Início/Fim)
  + Data realizada (Início/Fim)
  + Status (Planejado, Em Andamento, Concluído, Cancelado)
  + Observações e Impedimentos

**2. Cadastro de Atividades**

* **Campos Obrigatórios na Criação:**
  + Código do Projeto (ATV-00000)
  + Nome/Descrição
  + Responsável (usuário)
  + Tipo Atividade
  + Data estimada (Início/Fim)
* **Campos que podem ser atualizados ao longo do tempo:**
  + Nome/Descrição
  + Responsável (usuário)
  + Tipo Atividade
  + Data estimada (Início/Fim)

**3. Relatórios Gerenciais**

**A. Diagrama de Gantt (como o da imagem enviada)**

* Comparativo visual entre:
  + Planejado P.O.
  + Planejado Time
  + Execução Real
* Diferenciação de cores por tipo de dado
* Tooltip com detalhes do projeto ao passar o mouse

**B. Relatórios Analíticos**

* **Atrasos:** Projetos com maior diferença entre planejamento e execução
* **Confiabilidade:** Diferença entre datas do P.O. e do Time (quem está estimando melhor?)
* **Distribuição por Criticidade**
* **Projetos por Módulo**
* **Tempo Médio de Execução por Criticidade**
* **Projetos em andamento / concluídos / cancelados**

**C. Painel de Indicadores (Dashboard)**

* Total de projetos ativos
* % de entregas no prazo
* Projetos por responsável
* Gráfico de burn-down (opcional)

**⚙️ Funcionalidades Complementares**

* Filtro por datas, criticidade, módulo, status e responsável
* Exportar relatórios para PDF/Excel
* Histórico de alterações nos projetos
* Comentários / anotações por projeto (tipo “log de andamento”)
* Notificações de prazos próximos ou atrasados

**🛠️ Sugestão de Stack Técnica**

* **Backend:** C# (.NET 8 ou 9)
* **Frontend:** WinForms (para desktop - Inicial) , React (Fase04 – liberação Frontend)
* **Banco de Dados:** SQL Server, PostgreSQL ou SQLite
* **Autenticação:** JWT, IdentityServer, ou Identity padrão do ASP.NET
* **Lib para Gantt:** Google Charts, Highcharts Gantt, Syncfusion, ou DHTMLX Gantt (se for web)

**🗺️ Planejamento Inicial para Desenvolvimento**

1. **Fase 1: MVP (mínimo produto viável)**
   * Cadastro de projeto com datas P.O.
   * Tela de visualização básica com filtro
   * Diagrama de Gantt estático ou semi-estático
2. **Fase 2: Funcionalidades Avançadas**
   * Inserção das datas do Time e execução real
   * Relatórios analíticos
   * Extração .csv
   * Criação de usuários e login
   * Permissões
3. **Fase 3: Polimento**
   * Dashboard
   * Exportações e notificações
   * Logs e histórico de alterações
4. **Fase 4: Liberação Web**
   * Criação telas React
     1. Login
     2. Visualizações
     3. Diagramas
     4. Inserção/alteração dados
     5. Relatórios
     6. Dashboard
     7. Extrações

## 🖼️ 2. Telas (Protótipo estilo WinForms)

Em vez de HTML puro, vou descrever o layout com base em **controles do Windows Forms** para facilitar sua implementação:

### 📋 Tela: Login

* TextBox para Email
* TextBox para Senha
* Button "Entrar"
* Label de erro
* Link: Esqueceu a senha

### 📋 Tela: Nova Senha

* TextBox para Email
* TextBox para Código Gerado (será gerado um código na tela de usuário e enviado pelo ADMIN)
* TextBox para Nova Senha
* Button "Entrar"
* Label de erro

### 🏠 Tela: Dashboard Principal

* MenuStrip com opções: **Projetos**, **Usuários**, **Relatórios**
* ListView com projetos recentes
* Indicadores principais em Labels ou GroupBox:
  + Total de projetos
  + Projetos atrasados
  + Projetos em andamento

### 📁 Tela: Cadastro de Projeto

* TextBox: Código Projeto
* TextBox: Nome
* ComboBox: Equipe
* ComboBox: Módulo
* ComboBox: Criticidade
* DateTimePicker: Data início / fim (P.O.)
* DateTimePicker: Data início / fim (Time) – desabilitado inicialmente
* DateTimePicker: Data início / fim (Real) – desabilitado inicialmente
* ComboBox: Status
* ComboBox: Responsável
* RichTextBox: Observações
* Button: Salvar / Atualizar

### 📁 Tela: Cadastro de Tarefas

* TextBox: Código Tarefa
* TextBox: Nome
* ComboBox: Projeto (pode ser digitado e deve ser consultado no banco se existe)
* ComboBox: Equipe
* DateTimePicker: Data início / fim
* ComboBox: Status
* ComboBox: Responsável
* RichTextBox: Observações
* Button: Salvar / Atualizar

### 📃 Tela: Relatório de Projetos

* DataGridView com filtros por:
  + Equipe, Criticidade, módulo, período
* Botões: Gerar Gantt, Exportar PDF / HTML
* Button: Abrir projeto selecionado

### 📊 Tela: Diagrama de Gantt

* Pode ser implementado usando:
  + Biblioteca Gantt personalizada WinForms
  + Ou exportando para HTML e carregando via WebBrowser
  + Alternativamente: criar gráfico de barras com Chart

### 👤 Tela: Gerenciar Usuários (somente Admin)

* DataGridView com usuários
* Button: Novo usuário
* Form: Nome, Email, Papel, Lista de Equipes que o usuário pertence, Tipo Usuário (Admin, P.O., Usuário Padrão), Ativar/Desativar, Código Gerar Nova Senha

### 👤 Tela: Gerenciar Equipes (somente Admin)

* DataGridView com usuários por equipe
* Button: Nova Equipe
* Form: Nome, Ativar/Desativar, Add Usuários

### 👤 Processo Permissão: Ações de Usuários

* Admin
  + Acesso full ao sistema
  + Cadastrar Usuários
  + Visualizar e Cadastrar projeto para todos os times
* P.O.
  + Acesso a todos módulos
  + Visualizar e cadastrar projetos para os times que está associado.
  + Visualizar os relatórios da equipe
* Usuário Padrão:
  + Acesso nos módulos de consulta
  + Visualizar Projetos da mesma equipe

### Requisitos não funcionais

Esses são os "qualidades técnicas" que o sistema precisa ter.  
Por exemplo:

 **Tempo de Comunicação e Processamento**

* API deve responder em até 60 segundos
* Atualização de dados dashboard em tempo real ("refresh" em 5 minutos)
* Tolerância a falhas em 5 segundos

**Usabilidade**

* Sistema acessível para diferentes dispositivos
* Interface intuitiva

**Segurança**

Aqui vai todo conteúdo referente aos controles de acesso e trafego:

* A autenticação será feita via login e senha do usuário
* Autenticação e autorização
  + Através do Login e Senha deve ser retornado um JWT com os acessos do usuário no sistema
* O armazenar as informações sensíveis,
  + a senha deve ser gravada criptografada no banco
  + Proteção contra SQL Injection, XSS
* A comunicação entre o Back-end e o front-end da aplicação ocorrerá via API.
* Uso de SSL/TLS
* Logs de acesso

### Tecnologias e Frameworks Adotados

|  |  |  |
| --- | --- | --- |
| Item | Tecnologia | Observação |
| Linguagem Backend | C# .NET 9 | Aplicação ficará no servidor |
| Framework Frontend Fase 01 | Windows Forms | On-Premises |
| Framework Frontend Fase 02 | React 18 | SPA responsiva |
| Banco de Dados | SQL Server 2019 | Servidor Backend |
| Hospedagem | Servidor Windows |  |
| Autenticação | Login e Senha |  |
| Mensageria | - | - |

**Documentação Técnica**

**Padrões de Desenvolvimento**

* Clean Architecture ou arquitetura em camadas
* Repository Pattern para acesso a dados
* Unit of Work para transações
* Dependency Injection

**Testes**

Testes unitários para lógica de negócio

## 🎯 **Roadmap Atualizado**

**Fase 1: MVP Core**

* Autenticação básica
* CRUD de projetos
* Visualização simples

**Fase 2: Funcionalidades Essenciais**

* Gantt chart
* Relatórios básicos
* Controle de permissões

**Fase 3: Funcionalidades Avançadas**

* Dashboard completo
* Análise preditiva simples
* Exportações

**Fase 4: Polimento e Web**

* Interface React
* Integrações externas
* Otimizações de performance