

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Plano de Projeto: Integração Recruta.AI & Senior</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap"
rel="stylesheet">
  <!-- Chosen Palette: Warm Neutrals -->
  <!-- Application Structure Plan: A aplicação está estruturada como um painel de projeto
interativo, em vez de um documento linear. Utiliza uma navegação superior para acesso
rápido às seções principais: Visão Geral, Cronograma interativo, Recursos e
Documentação. O núcleo da aplicação é o gráfico de cronograma interativo estilo Gantt, que
representa visualmente as fases do projeto e sua duração em sprints. Clicar em uma fase
no gráfico exibe dinamicamente suas tarefas e entregas detalhadas abaixo. Essa estrutura
foi escolhida para fornecer um resumo visual de alto nível, permitindo que os usuários
explorem os detalhes sob demanda, o que é mais envolvente e menos sobrecarregado do
que uma página de texto longa e estática. Transforma um plano de projeto de uma
experiência de leitura passiva em uma ferramenta de exploração ativa. -->
  <!-- Visualization & Content Choices:
- Cronograma do Projeto: Informação do Relatório -> Fases do projeto e durações em
sprints. Objetivo -> Mostrar mudança/progressão ao longo do tempo. Visualização/Método
-> Gráfico de Barras Horizontais (estilo Gantt). Interação -> Clicar na barra de uma fase
para revelar detalhes. Justificativa -> Um gráfico de Gantt é a forma mais intuitiva de
visualizar um cronograma de projeto. A interação permite a divulgação progressiva de
informações, melhorando a usabilidade. Biblioteca -> Chart.js (Canvas).
- Detalhes da Fase (Tarefas, Entregas): Informação do Relatório -> Detalhamento de
cada fase do projeto. Objetivo -> Organizar e Informar. Visualização/Método -> Cartões
HTML gerados dinamicamente. Interação -> O conteúdo aparece/atualiza com base na
seleção do gráfico. Justificativa -> Os cartões fornecem uma maneira estruturada e
visualmente atraente de apresentar informações detalhadas, e o carregamento dinâmico
mantém a interface limpa. Biblioteca/Método -> Vanilla JS, HTML, Tailwind CSS.
- Recursos & Documentação: Informação do Relatório -> Listas de membros da equipe,
sistemas e links de API. Objetivo -> Informar. Visualização/Método -> Listas e cartões HTML
estilizados. Interação -> Hiperlinks padrão. Justificativa -> Este é um material de referência,
melhor apresentado em um formato claro e de fácil leitura. Biblioteca/Método -> HTML,
Tailwind CSS.
-->
  <!-- CONFIRMATION: NO SVG graphics used. NO Mermaid JS used. -->
  <style>
    body {
      font-family: 'Inter', sans-serif;
      background-color: #f8f7f4;

```

```

        color: #333;
    }
    .nav-link {
        transition: color 0.3s, border-bottom-color 0.3s;
    }
    .nav-link:hover {
        color: #c084fc;
    }
    .nav-link.active {
        color: #a855f7;
        border-bottom: 2px solid #a855f7;
    }
    .phase-card {
        background-color: white;
        border-radius: 0.75rem;
        box-shadow: 0 4px 6px -1px rgb(0 0 0 / 0.1), 0 2px 4px -2px rgb(0 0 0 / 0.1);
        transition: transform 0.3s, box-shadow 0.3s;
    }
    .phase-card:hover {
        transform: translateY(-5px);
        box-shadow: 0 10px 15px -3px rgb(0 0 0 / 0.1), 0 4px 6px -4px rgb(0 0 0 / 0.1);
    }
    .chart-container {
        position: relative;
        width: 100%;
        max-width: 1024px;
        margin-left: auto;
        margin-right: auto;
        height: 300px;
        max-height: 400px;
    }
    @media (min-width: 768px) {
        .chart-container {
            height: 400px;
        }
    }
    .section-title {
        color: #868075;
        font-weight: 600;
        letter-spacing: 0.05em;
    }
}
</style>
</head>
<body class="antialiased">

<header class="bg-white/80 backdrop-blur-lg sticky top-0 z-50 shadow-sm">
  <nav class="container mx-auto px-6 py-4 flex justify-between items-center">
    <h1 class="text-2xl font-bold text-purple-600">Integração Recruta.AI + Senior</h1>

```

```
<div class="hidden md:flex space-x-8">
  <a href="#overview" class="nav-link text-gray-600 font-medium pb-1">Visão
Geral</a>
  <a href="#timeline" class="nav-link text-gray-600 font-medium
pb-1">Cronograma</a>
  <a href="#resources" class="nav-link text-gray-600 font-medium
pb-1">Recursos</a>
  <a href="#docs" class="nav-link text-gray-600 font-medium
pb-1">Documentação</a>
</div>
</nav>
</header>
```

```
<main class="container mx-auto px-6 py-12">
```

```
<section id="overview" class="mb-20 text-center">
  <h2 class="text-4xl font-bold text-gray-800 mb-4">Visão Geral do Projeto</h2>
  <p class="max-w-3xl mx-auto text-lg text-gray-600">
    O objetivo principal deste projeto é integrar as plataformas Recruta.AI e Senior
    Sistemas para automatizar o processo de admissão de novos colaboradores, reduzindo a
    duplicação de dados e agilizando as contratações. A automação visa otimizar o fluxo de
    trabalho do RH, garantindo maior eficiência e precisão das informações.
  </p>
</section>
```

```
<section id="timeline" class="mb-20">
  <div class="text-center mb-12">
    <h2 class="text-4xl font-bold text-gray-800 mb-2">Cronograma Interativo</h2>
    <p class="text-lg text-gray-600">Clique em uma fase no gráfico para ver seus
detalhes.</p>
  </div>
  <div class="bg-white p-4 sm:p-6 rounded-xl shadow-lg">
    <div class="chart-container">
      <canvas id="timelineChart"></canvas>
    </div>
  </div>

  <div id="phase-details" class="mt-12"></div>
</section>
```

```
<section id="resources" class="mb-20">
  <div class="text-center mb-12">
    <h2 class="text-4xl font-bold text-gray-800 mb-2">Recursos Necessários</h2>
    <p class="text-lg text-gray-600">Equipe, sistemas e documentação essenciais
para o sucesso do projeto.</p>
  </div>
  <div class="grid md:grid-cols-3 gap-8">
    <div class="phase-card p-6">
```

```

<h3 class="text-xl font-bold text-purple-600 mb-3">👤 Equipe</h3>
<ul class="space-y-2 text-gray-700 list-disc list-inside">
  <li>Gerente de Projeto</li>
  <li>Especialista de RH/R&S</li>
  <li>Analista de TI</li>
  <li>Suporte técnico da Senior</li>
  <li>Suporte técnico da Recruta.AI</li>
</ul>
</div>
<div class="phase-card p-6">
  <h3 class="text-xl font-bold text-purple-600 mb-3">💻 Sistemas</h3>
  <ul class="space-y-2 text-gray-700 list-disc list-inside">
    <li>Acesso à plataforma Recruta.AI</li>
    <li>Acesso ao sistema Senior</li>
    <li>Permissões de API em ambos</li>
  </ul>
</div>
<div class="phase-card p-6">
  <h3 class="text-xl font-bold text-purple-600 mb-3">📖 Documentação</h3>
  <ul class="space-y-2 text-gray-700 list-disc list-inside">
    <li>Manuais de API da Senior</li>
    <li>Documentação de integração da Recruta.AI</li>
  </ul>
</div>
</div>
</section>

<section id="docs">
  <div class="text-center mb-12">
    <h2 class="text-4xl font-bold text-gray-800 mb-2">Documentação Técnica</h2>
    <p class="text-lg text-gray-600">Links úteis para a equipe de desenvolvimento e
    TI.</p>
  </div>
  <div class="bg-white rounded-xl shadow-lg p-8 space-y-6">
    <div>
      <h3 class="text-xl font-semibold text-gray-800">Documentação de APIs da
      Senior</h3>
      <p class="text-gray-600 mt-1">Recursos essenciais para a equipe técnica
      iniciar o processo de integração, detalhando serviços, fluxos de autenticação e dados
      necessários.</p>
      <div class="mt-4 grid sm:grid-cols-1 md:grid-cols-3 gap-4">
        <a href="https://documentacao.senior.com.br/" target="_blank" class="block
        p-4 bg-purple-50 hover:bg-purple-100 rounded-lg transition-colors">
          <p class="font-semibold text-purple-700">Portal de Documentação
          Senior</p>
          <p class="text-sm text-purple-600">Documentação geral.</p>
        </a>
      </div>
    </div>
  </div>
</section>

```

```

        <a href="https://dev.senior.com.br/apis_publicas/" target="_blank"
class="block p-4 bg-purple-50 hover:bg-purple-100 rounded-lg transition-colors">
        <p class="font-semibold text-purple-700">Portal de APIs Senior X</p>
        <p class="text-sm text-purple-600">Portal específico para APIs.</p>
        </a>
        <a
href="https://documentacao.senior.com.br/seniorxplatform/manual-do-usuario/hcm/admissao
-digital/integracoes/integracao-via-api.htm" target="_blank" class="block p-4 bg-purple-50
hover:bg-purple-100 rounded-lg transition-colors">
        <p class="font-semibold text-purple-700">API de Admissão Digital</p>
        <p class="text-sm text-purple-600">Detalhes da API específica.</p>
        </a>
    </div>
</div>
</div>
</section>
</main>

```

```

<footer class="bg-gray-800 text-white mt-20">
    <div class="container mx-auto px-6 py-4 text-center">
        <p>&copy; 2025 Painel de Projeto Interativo. Todos os direitos reservados.</p>
    </div>
</footer>

```

```

<script>
    const projectData = [
        {
            phase: 'Fase 1: Planejamento e Descoberta',
            sprints: [1, 2],
            objective: 'Alinhar expectativas, mapear processos e definir o escopo do projeto.',
            details: [
                { title: 'Reunião de Kick-off', content: 'Participantes: Líderes de RH, TI, R&S, e representantes da Senior e Recruta.AI. Pauta: Apresentação do projeto, definição de papéis e responsabilidades, e alinhamento do cronograma.' },
                { title: 'Mapeamento de Processos', content: 'Documentar o processo atual de Recrutamento e Seleção e Admissão. Identificar os campos de dados que precisam ser transferidos (ex: nome, CPF, etc.).' },
                { title: 'Análise de Viabilidade Técnica', content: 'A equipe de TI valida os pré-requisitos, como permissões de API no Senior e acesso à plataforma Recruta.AI.' }
            ],
            deliverable: 'Documento de mapeamento de campos e validação técnica.'
        },
        {
            phase: 'Fase 2: Configuração e Desenvolvimento',
            sprints: [3, 5],
            objective: 'Configurar as plataformas e habilitar a integração.',
            details: [

```

```
    { title: 'Configuração no Recruta.AI', content: 'Acessar o painel de integrações e inserir as credenciais de acesso ao sistema Senior.' },
```

```
    { title: 'Configuração no Senior', content: 'A equipe de TI, com o suporte da Senior, habilita as APIs necessárias no ambiente da empresa e garante que o módulo de Admissão Digital esteja ativo.' },
```

```
    { title: 'Desenvolvimento (Opcional)', content: 'Se houver campos customizados, a equipe de desenvolvimento cria as regras de negócio para a transferência de dados via API.' }
```

```
  ],  
  deliverable: 'Integração ativada e pronta para testes.'
```

```
},
```

```
{
```

```
  phase: 'Fase 3: Testes e Validação',
```

```
  sprints: [6, 7],
```

```
  objective: 'Garantir que a integração funcione perfeitamente e sem erros.',
```

```
  details: [
```

```
    { title: 'Testes de Integração', content: 'Realizar testes de ponta a ponta em um ambiente de homologação, simulando um processo de seleção completo.' },
```

```
    { title: 'Teste de Massa', content: 'Realizar a transferência de dados de múltiplos candidatos para garantir que o processo seja escalável e sem perda de informação.' },
```

```
    { title: 'Treinamento da Equipe', content: 'Treinar a equipe de R&S e o Departamento Pessoal sobre o novo fluxo.' }  
  ],
```

```
  deliverable: 'Relatório de testes com aprovação das equipes de RH e TI.'
```

```
},
```

```
{
```

```
  phase: 'Fase 4: Go-Live e Pós-Implementação',
```

```
  sprints: [8, 9],
```

```
  objective: 'Colocar a integração em produção e monitorar o desempenho.',
```

```
  details: [
```

```
    { title: 'Go-Live (Entrada em Produção)', content: 'Habilitar a integração no ambiente de produção e comunicar o novo processo a toda a equipe de RH.' },
```

```
    { title: 'Monitoramento e Suporte', content: 'A equipe de TI monitora o fluxo de dados para identificar e corrigir possíveis erros. Estabelecer canal de comunicação com suporte técnico.' },
```

```
    { title: 'Feedback e Melhoria Contínua', content: 'Coletar feedback da equipe sobre o novo fluxo e propor melhorias e otimizações.' }  
  ],
```

```
  deliverable: 'Processo de integração em produção e monitorado.'
```

```
}
```

```
];
```

```
const phaseDetailsContainer = document.getElementById('phase-details');
```

```
let timelineChartInstance = null;
```

```
let activePhaseIndex = 0;
```

```
async function generateSummary(index) {
```

```
  const data = projectData[index];
```

```

const summaryContainer = document.getElementById('summary-container');
const summaryButton = document.getElementById('summary-button');
const originalButtonText = summaryButton.innerHTML;

summaryButton.disabled = true;
summaryButton.innerHTML = `<svg class="animate-spin -ml-1 mr-3 h-5 w-5
text-white" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"><circle
class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor"
stroke-width="4"></circle><path class="opacity-75" fill="currentColor" d="M4 12a8 8 0
018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824
3 7.938l3-2.647z"></path></svg> Gerando...`;
summaryContainer.innerHTML = "";

const prompt = `Resuma o seguinte plano de projeto de uma forma concisa e
amigável, focando nos pontos chave. As informações são: Objetivo: "${data.objective}".
Detalhes: ${data.details.map(d => `${d.title}: ${d.content}`).join(' ')}. Entrega:
"${data.deliverable}":`;

const payload = {
  contents: [{ parts: [{ text: prompt }] }],
  systemInstruction: {
    parts: [{ text: "Você é um especialista em gerenciamento de projetos. Forneça
um resumo objetivo e fácil de entender." }]
  },
  tools: [{ "google_search": {} } ]
};
const apiKey = "";
const apiUrl =
`https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash-preview-05-20:
generateContent?key=${apiKey}`;

try {
  const response = await fetch(apiUrl, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(payload)
  });

  if (!response.ok) {
    throw new Error(`Erro na API: ${response.status} ${response.statusText}`);
  }

  const result = await response.json();
  const summary = result.candidates[0].content.parts[0].text;

  summaryContainer.innerHTML = `
    <div class="bg-gray-100 p-6 rounded-lg mt-6 border-l-4 border-gray-400">
      <h4 class="font-bold text-gray-800 text-lg mb-2">Resumo da Fase</h4>

```

```

        <p class="text-gray-700">${summary}</p>
    </div>
    `;
} catch (error) {
    summaryContainer.innerHTML = `<div class="bg-red-100 border-l-4
border-red-500 text-red-700 p-4 mt-6 rounded" role="alert"><p>Erro ao gerar resumo. Por
favor, tente novamente.</p></div>`;
} finally {
    summaryButton.disabled = false;
    summaryButton.innerHTML = originalButtonText;
}
}

function displayPhaseDetails(index) {
    activePhaseIndex = index;
    const data = projectData[index];
    phaseDetailsContainer.innerHTML = `
    <div class="bg-white rounded-xl shadow-lg p-8 animate-fade-in">
        <h3 class="text-2xl sm:text-3xl font-bold text-purple-600">${data.phase}</h3>
        <p class="text-gray-500 mt-1 mb-6 text-lg">Sprints:
    ${data.sprints[0]}-${data.sprints[1]}</p>
        <p class="text-lg text-gray-700 mb-8">${data.objective}</p>
        <div class="grid md:grid-cols-1 lg:grid-cols-3 gap-6 mb-8">
            ${data.details.map(detail => `
                <div class="bg-gray-50 p-6 rounded-lg border border-gray-200">
                    <h4 class="font-semibold text-gray-800 text-lg mb-2">${detail.title}</h4>
                    <p class="text-gray-600">${detail.content}</p>
                </div>
            `).join("")}
        </div>
        <div class="bg-purple-50 border-l-4 border-purple-500 p-4 rounded-r-lg">
            <h4 class="font-bold text-purple-800">Entrega Principal:</h4>
            <p class="text-purple-700">${data.deliverable}</p>
        </div>
        <div class="mt-8 flex justify-end">
            <button id="summary-button" onclick="generateSummary(${index})"
class="bg-purple-600 hover:bg-purple-700 text-white font-bold py-2 px-6 rounded-full
transition-colors flex items-center justify-center space-x-2 shadow-lg">
                <span>Gerar Resumo ✨</span>
            </button>
        </div>
        <div id="summary-container"></div>
    </div>
    `;

    updateChartHighlight();

    const keyframes = `

```



```

    @keyframes fade-in {
      0% { opacity: 0; transform: translateY(20px); }
      100% { opacity: 1; transform: translateY(0); }
    }
  `;

  const styleSheet = document.createElement("style");
  styleSheet.innerText = keyframes;
  document.head.appendChild(styleSheet);
  phaseDetailsContainer.querySelector('.animate-fade-in').style.animation = 'fade-in
0.5s ease-out forwards';
}

function updateChartHighlight() {
  if (timelineChartInstance) {
    timelineChartInstance.data.datasets[0].backgroundColor = projectData.map((_, i)
=> i === activePhaseIndex ? 'rgba(168, 85, 247, 1)' : 'rgba(209, 178, 255, 0.7)');
    timelineChartInstance.data.datasets[0].borderColor = projectData.map((_, i) => i
=== activePhaseIndex ? 'rgba(147, 51, 234, 1)' : 'rgba(168, 85, 247, 0.5)');
    timelineChartInstance.update();
  }
}

document.addEventListener('DOMContentLoaded', () => {
  const ctx = document.getElementById('timelineChart').getContext('2d');
  timelineChartInstance = new Chart(ctx, {
    type: 'bar',
    data: {
      labels: projectData.map(p => p.phase),
      datasets: [{
        label: 'Sprints',
        data: projectData.map(p => p.sprints),
        backgroundColor: projectData.map((_, i) => i === activePhaseIndex ?
'rgba(168, 85, 247, 1)' : 'rgba(209, 178, 255, 0.7)'),
        borderColor: projectData.map((_, i) => i === activePhaseIndex ? 'rgba(147,
51, 234, 1)' : 'rgba(168, 85, 247, 0.5)'),
        borderWidth: 2,
        borderSkipped: false,
      }]
    },
    options: {
      indexAxis: 'y',
      responsive: true,
      maintainAspectRatio: false,
      scales: {
        x: {
          beginAtZero: true,
          grid: {
            display: false

```

```

    },
    title: {
      display: true,
      text: 'Sprints do Projeto',
      font: {
        size: 14,
        weight: 'bold'
      }
    }
  },
  y: {
    grid: {
      display: true,
      color: '#e5e7eb'
    }
  }
},
plugins: {
  legend: {
    display: false
  },
  tooltip: {
    callbacks: {
      label: function(context) {
        const item = context.raw;
        return `Sprints: ${item[0]} a ${item[1]}`;
      }
    }
  }
},
onClick: (event, elements) => {
  if (elements.length > 0) {
    const index = elements[0].index;
    displayPhaseDetails(index);
  }
},
onHover: (event, chartElement) => {
  event.native.target.style.cursor = chartElement[0] ? 'pointer' : 'default';
}
}
});

```

```
displayPhaseDetails(0);
```

```

document.querySelectorAll('a[href^="#"]').forEach(anchor => {
  anchor.addEventListener('click', function (e) {
    e.preventDefault();
    document.querySelector(this.getAttribute('href')).scrollIntoView({

```

```
        behavior: 'smooth'
    });
});
});
</script>
</body>
</html>
```