# Levantamento de Requisitos

Dentre os métodos de levantamento de requisitos disponíveis, os quais são, JAD, PIECES e entrevista, optamos pelo último, pois nosso cliente sabe como quer seu software.

A metodologia JAD é utilizada para desenvolver softwares que já estão funcionando bem e PIECES para quando quer resolver problemas no software do cliente, logo não se encaixam bem no nosso projeto. Pelo fato de não existir nenhum sistema, essas duas metodologias serão descartadas.

Na entrevista, participaram todos os nossos membros da equipe: Diego Nunes, Adriano Vitoriano, Luiz Felipe e Marlon Rodrigues, com a finalidade de levantar questões e adquirir o máximo de respostas a fim de entender melhor as necessidades do nosso cliente.

O relatório detalhado encontra-se no apêndice 1.

# Levantamento de risco

Após levantado os requisitos, fizemos uma reunião para detalhar o projeto, onde levantamos os riscos que tomaríamos durante todo o desenvolvimento e elaborando um plano de contingência de como lidar com eles conforme abaixo:

**Requisitos:** Entrevista Pobre e Levantamentos de Requisitos Pouco Claros

**Impacto:** Alto

**Justificativa:** Requisitos mal definidos podem resultar em um produto final que não atende às expectativas do cliente, prejudicando a imagem profissional da equipe e causando retrabalho.

**Como Lidar:**

* Documentar detalhadamente todas as informações obtidas durante as entrevistas.
* Realizar reuniões periódicas com o cliente para revisão e alinhamento contínuo dos requisitos.
* Utilizar técnicas de elicitação de requisitos, como workshops e sessões de brainstorming, para garantir que todas as necessidades do cliente sejam compreendidas e registradas.

**Conhecimento Técnico:** Desconhecimento das Tecnologias, Ideias e Conceitos Novos

**Impacto:** Alto

**Justificativa:** A falta de conhecimento técnico pode aumentar a complexidade do desenvolvimento, limitar as funcionalidades do software e introduzir bugs, causando insatisfação do cliente.

**Como Lidar:**

* Realizar pesquisas e estudos sobre as tecnologias necessárias antes e durante o desenvolvimento.
* Buscar apoio dos professores e mentores para orientação técnica.
* Participar de comunidades de programadores e fóruns online para resolver dúvidas e compartilhar conhecimentos.
* Investir em cursos e treinamentos específicos, se necessário.

**Financeiro:** Necessidade de Compra de Equipamentos ou Softwares Especializados Caros

**Impacto:** Baixo

**Justificativa:** O projeto não exija uma infraestrutura robusta e a instituição já possui uma infraestrutura bem estruturada e funcional.

**Como Lidar:**

* Utilizar softwares gratuitos e de código aberto sempre que possível.
* Solicitar apoio institucional para a aquisição de recursos, caso necessário.
* Planejar o orçamento do projeto antecipadamente para identificar e mitigar possíveis necessidades financeiras.

**Equipe:** Falta de Consenso entre Ideias e Visões

**Impacto:** Alto

**Justificativa:** A desunião na equipe pode paralisar o desenvolvimento, causar conflitos e até sabotagem, prejudicando o progresso do projeto.

**Como Lidar:**

* Estabelecer divisões claras de tarefas e responsabilidades desde o início.
* Promover um ambiente de respeito e colaboração, onde todas as ideias sejam ouvidas e discutidas.
* Realizar reuniões frequentes para alinhamento de visão e resolução de conflitos.
* Implementar métodos de gerenciamento de projetos, como Scrum ou Kanban, para melhorar a comunicação e a coordenação dentro da equipe.
* Designar um líder de projeto que mediará conflitos e facilitará a tomada de decisões.

**Tempo:** Atrasos no Cronograma

**Impacto:** Alto

**Justificativa:** Atrasos podem comprometer a entrega do projeto, causando insatisfação do cliente e pressão adicional sobre a equipe.

**Como Lidar:**

* Estabelecer um cronograma realista com margens de segurança para imprevistos.
* Monitorar o progresso regularmente e ajustar o cronograma conforme necessário.
* Dividir o projeto em etapas menores com entregas parciais para manter o ritmo de desenvolvimento.
* Alocar recursos extras em caso de atrasos significativos.

**Jurídico:** Conformidade e Legalidade

**Impacto:** Baixo

**Justificativa:** Não há preocupações jurídicas significativas, pois a aplicação é particular e interna à pastelaria.

**Como Lidar:**

* Garantir que o desenvolvimento do software esteja em conformidade com as regulamentações locais.
* Manter documentação adequada do projeto para referência futura.
* Consultar um especialista jurídico se necessário para questões específicas.

# Cronograma Project

Para formalizar o planejado, estruturamos um cronograma utilizando o software Project Libre:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nome** | **Duração** | **Início** | **Fim** | **Antecessores** |
| 1 | **Projeto Integrador** | **113 dias?** | **05/08/24** | **25/11/24** |  |
| 2 | Definir Grupos | 23 dias | 05/08/24 | 27/08/24 |  |
| 3 | Definir Cliente | 2 dias | 28/08/24 | 29/08/24 | 2 |
| 4 | **Engenharia de Software** | **24 dias?** | **30/08/24** | **22/09/24** | **3** |
| 5 | **Engenharia de Requisitos** | **15,8 dias** | **30/08/24** | **14/09/24** | **3** |
| 6 | Reunião com o Cliente | 15 dias | 30/08/24 | 13/09/24 | 3 |
| 7 | Relatório da Reunião | 0,1 dias | 14/09/24 | 14/09/24 | 6 |
| 8 | Análise de requisitos | 0,7 dias | 14/09/24 | 14/09/24 | 7 |
| 9 | **Análise de Risco** | **0,2 dias** | **14/09/24** | **14/09/24** | **5** |
| 10 | Levantamento de Risco | 0,1 dias | 14/09/24 | 14/09/24 | 5;8 |
| 11 | Plano de Contingência | 0,1 dias | 14/09/24 | 14/09/24 | 10 |
| 12 | **Planejamento do Projeto** | **8 dias?** | **15/09/24** | **22/09/24** | **5;9** |
| 13 | Cronograma do Projeto | 0,3 dias | 15/09/24 | 15/09/24 | 8 |
| 14 | PDCA | 0,3 dias | 15/09/24 | 15/09/24 | 13 |
| 15 | Diagramas UML | 8 dias | 15/09/24 | 22/09/24 | 5 |
| 16 | Function Point | 1 dia? | 15/09/24 | 15/09/24 | 5 |
| 17 | **Planejamento das Telas** | **6 dias?** | **15/09/24** | **20/09/24** | **5;9** |
| 18 | Esboço | 1 dia? | 15/09/24 | 15/09/24 | 5 |
| 19 | Alinhamento com Cliente | 4 dias | 16/09/24 | 19/09/24 | 18 |
| 20 | Ajustes | 1 dia | 20/09/24 | 20/09/24 | 19 |
| 21 | **Codificação** | **65 dias** | **21/09/24** | **24/11/24** | **5;17** |
| 22 | **Banco de Dados** | **2 dias** | **21/09/24** | **22/09/24** | **5** |
| 23 | Modelagem Conceitual | 0,5 dias | 21/09/24 | 21/09/24 | 5 |
| 24 | Modelagem Lógica | 0,5 dias | 21/09/24 | 21/09/24 | 23 |
| 25 | Criação do Banco de dados | 1 dia | 22/09/24 | 22/09/24 | 23;24 |
| 26 | **Funcionalidades** | **63 dias** | **23/09/24** | **17/11/24** | **17;22** |
| 27 | Cadastro Online no Sistema | 7 dias | 23/09/24 | 29/09/24 | 17;22 |
| 28 | Cadastro de Cursos | 7 dias | 30/09/24 | 06/10/24 | 27 |
| 29 | Cadastro de Pré-Requisitos | 7 dias | 07/10/24 | 13/10/24 | 28 |
| 30 | Cadastro online para o curso | 7 dias | 14/10/24 | 20/10/24 | 29 |
| 31 | Tela de perfil | 7 dias | 21/10/24 | 27/10/24 | 30 |
| 32 | Enviar mensagens / e-mails | 7 dias | 28/10/24 | 03/11/24 | 31 |
| 33 | Gerar Turmas | 7 dias | 04/11/24 | 10/11/24 | 32 |
| 34 | Relatórios | 7 dias | 11/11/24 | 17/11/24 | 33 |
| 35 | Testes | 7 dias | 18/11/24 | 24/11/24 | 34 |
| 36 | Apresentação | 2 dias | 25/11/24 | 26/11/24 | 26 |

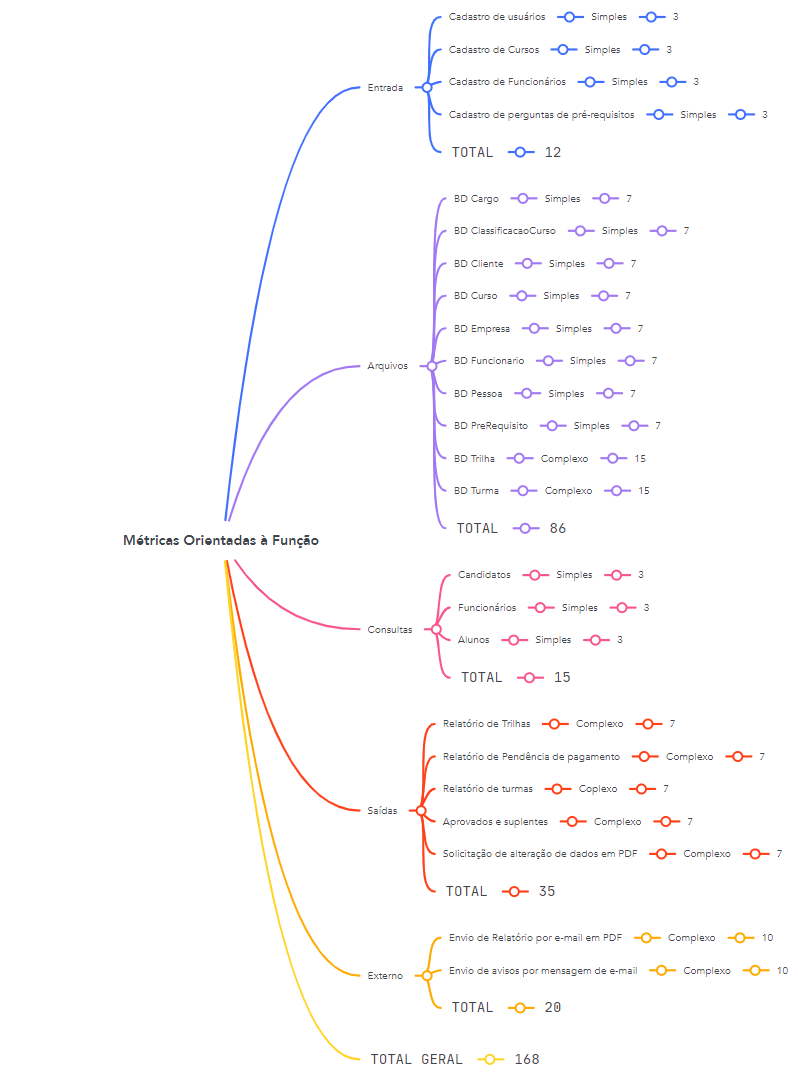
# PDCA

Após termos levantado os requisitos e o escopo do projeto, formalizamos as etapas num cronograma e com o auxílio do ciclo PDCA

|  |  |  |
| --- | --- | --- |
| **Código** | | |
| **Plan** | Iniciar Desenvolvimento do software | **21/set** |
| **O que precisa para fazer o software** | | |
| **Do** | Finalizar o levantamento de requisitos com o cliente | **14/set** |
| Finalizar o planejamento e modelagem do projeto | **16/set** |
| Fechar valor e projeto com cliente | **19/set** |
| **Check** | Finalizar e revisar o planejamento e modelagem | **20/set** |
| **Act** | Se o planejamento e modelagem do projeto ainda não estiver pronto: Fazer horas extras, incluindo finais de semanas e feriados afim de manter o prazo |  |
|  |  |  |
| **Teste** | | |
| **Plan** | Iniciar o teste de software | **18/nov** |
| **O que precisa para testar o software?** | | |
| **Do** | Todos os módulos estruturados e funcionais | **17/nov** |
| Finalizar integração da API com a interface | **17/nov** |
| **Check** | Verificar o andamento do desenvolvimento do software | **10/nov** |
| **Act** | Caso o software ainda não tenha sido finalizado: Fazer horas extras, incluindo finais de semanas e feriados afim de manter o prazo. |  |
|  |  |  |
| **Implementação** | | |
| **Plan** | Implementar o software | **26/nov** |
| **O que precisa para implementar o software?** | | |
| **Do** | Software operacional e testado | **17/nov** |
| Testes finalizados com êxito | **24/nov** |
| **Check** | Apresentação e aprovação do cliente | **24/nov** |
| **Act** | Caso o cliente não aprove a implementação do software, teremos que fazer uma nova reunião para alinhar uma possível mudança ou melhoria, reiniciando o projeto. | **N/A** |

# Function point

Depois de definido a estrutura do nosso site, utilizamos a métrica do Function Point para levantar todos os recursos necessários para podermos deixar o software operante:



O sistema comunica dados com outros sistemas?

O sistema usa processamento distribuído?

Existem requisitos específicos de desempenho (tempo de resposta ou taxa de transferência) para o sistema?

O sistema é projetado para ser operado em configurações de hardware específicas?

O sistema processa um grande volume de transações?

O sistema inclui funções de entrada de dados on-line?

O sistema requer uma interface de usuário final eficiente (facilidade de uso)?

O sistema inclui funções de atualização de dados on-line?

O sistema realiza um processamento complexo?

O sistema foi projetado para ser reutilizável em outras aplicações?

O sistema é projetado para facilitar a instalação?

O sistema é projetado para facilitar a operação (facilidade de manutenção)?

O sistema será usado em múltiplos locais?

O sistema é projetado para facilitar futuras modificações e melhorias?

**FP** = (Contagem Total **X**[0,65+0,01**X**Soma(FI)])\*VFP **=** ... **X** [0,65**+**0,01**X**...] = **\* [valor justo] = R$ 00,00**

Por fim, o Function Point avalia nosso software com o preço de **R$ 00,00.**

# Solicitação de manutenção

Na aula de manutenção de software aprendemos sobre a importância da formalização das solicitações de manutenção no software. Existem 3 tipos de manutenção: Correção: Para efetuar reparos, Adaptação: resultando de mudanças no ambiente no qual o software está inserido e Aperfeiçoamento: quando deseja-se satisfazer as crescentes necessidades do usuário. No apêndice 2 encontra-se um modelo solicitação de manutenção incremental utilizando nosso software como exemplo.

# Apêndice 1: Relatório da Entrevista

**Indaiatuba, 17 de setembro de 2024**

**Relatório 01:** Sistema de Gestão de Cursos SENAI

**Entrevistado:** Fábio Cabrero (Coordenador de cursos FIC)

**Entrevistadores:** Diego, Adriano, Leonardo, Luiz, Marlon

## Necessidades:

# Descrição Geral:

**Cadastro de Alunos:** Implementação de um sistema de cadastro de alunos independente do sistema central, permitindo o registro e a gestão das informações pessoais dos alunos.

**Gestão de Cursos FIC:** Cadastro e gerenciamento dos cursos de Formação Inicial e Continuada (FIC), sendo cursos abertos ao público, para empresas e inclusão de cursos especiais obrigatórios como NRs.

**Cadastro e Gestão de Trilhas:** Definição e gestão das trilhas de aprendizado, indicando pré-requisitos entre cursos.

**Cadastro Online e Inscrição:** Sistema para inscrição online em cursos, validação de pré-requisitos e gerenciamento de pagamentos.

**Geração de PDF e Perfil do Aluno:** Geração de documentos de inscrição e visualização do perfil do aluno.

**Comunicação e Avisos:** Sistema de notificações para apressar matrículas e informar status de inscrição.

**Geração e Gestão de Turmas:** Criação de turmas baseadas no número de inscritos e atribuição de professores.

**Relatórios e Exportação:** Geração de relatórios e análise detalhada.

**Acesso e Perfis:** Gerenciamento de perfis e permissões para diferentes tipos de usuários (gerente, secretaria, professor, aluno).

## Funcionamento:

**RN =** Regras de negócio

**RF =** Requisitos funcionais

**RNF =** Requisitos não funcionais

**RIU =** Requisitos da interface com o usuário

**MSG =** Mensagens do sistema

|  |  |
| --- | --- |
| **RN01** | Funcionário pode cadastrar candidatos |
| **RN02** | Funcionário muda o status de pagamento de pendente para pago manualmente |
| **RN03** | O gerente deve ter todas as permissões que os funcionários tem |
| **RN04** | Informações para cadastro online: nome, nascimento, CPF, Escolaridade, e-mail, telefone (fixo e celular), endereço, PCD, possui dificuldade |
| **RN05** | No primeiro login o candidato deve alterar a senha de e-mail, funcionários não precisam alterar |
| **RN06** | Toda criação e alteração de trilha, cursos e turmas podem ser feitos pelos funcionários e devem ser aprovados pelo gerente. |
| **RN07** | Os candidatos podem ser aceitos nos cursos mesmo que não atendam os pré-requisitos. |
| **RN08** | **Gerente:** Ver turmas, perfis dos alunos e perfis dos funcionários. (visualização, cadastro e alteração) **Secretaria:** Ver todas as turmas, perfis dos alunos. (visualização, cadastro e alteração) **Professor:** Ver as turmas abertas e ministradas por ele. (visualização) **Aluno:** Ver o próprio perfil (visualização), cursos, menos cursos FIC empresa |
| **RN09** | Os dados do candidato podem ser alterados por ele mesmo. Os dados do aluno podem ser alterados pela secretaria apenas, através da entrega de uma ficha de solicitação de alteração. |
|  |  |
| **RF01** | O sistema deve validar se o usuário possui cadastro, se sim direciona para login |
| **RF02** | O sistema deve permitir o cadastro online |
| **RF03** | O sistema deve oferecer vários tipos de pagamento |
| **RF04** | O sistema deve validar o e-mail do candidato, funcionários não precisam de validação |
| **RF05** | O sistema deve gerar uma senha provisória para o candidato |
| **RF06** | O sistema deve permitir linkar cursos com as trilhas. Turmas com os cursos, alunos e professores. |
| **RF07** | O sistema deve permitir cadastro de trilhas, cursos, turmas e pessoas |
| **RF08** | O sistema deve permitir escolher a forma de resposta (dissertativa, alternativa) das perguntas que os candidatos devem responder, por quem estiver criando |
| **RF09** | O sistema deve mandar e-mail com os dados de pagamento, bem como mostrar em tela ao finalizar a inscrição |
| **RF10** | O sistema deve permitir o candidato entrar na lista de espera de abertura de uma turma de um curso. |
| **RF11** | O sistema deve permitir assinatura digital dos candidatos |
| **RF12** | O sistema deve permitir a remoção de um candidato com o pagamento pendente, mesmo aprovado |
| **RF13** | O sistema deve gerar PDF da inscrição do aluno no curso e ficha de solicitação de alteração dos dados do perfil |
| **RF14** | O sistema deve permitir exportar os dados em Excel |
|  |  |
| **RNF01** | O sistema deve ter as cores e estilo dos sistemas do SENAI |
|  |  |
| **RIU01** | O botão de "inscreva-se" deve mudar para "declarar interesse no curso" quando não tiver turma aberta ou com previsão de abertura |
| **RIU02** | Todos podem ver o perfil dos alunos, exceto outros alunos que podem apenas o seu próprio |
| **RIU03** | O candidato deve poder ver os dados do curso antes de se inscrever: nome, período, área, descrição (com upload de fotos e vídeos), presencial, curso livre,  programação (tudo que vai aprender), requisitos, trilha, perfil profissional, carga horária, preço, data de início |
| **RIU04** | O candidato deve poder filtrar os cursos por: nome, período, área, presencial, curso livre, requisitos, trilha, carga horária, preço, data de início |
|  |  |
| **MSG01** | E-mail de confirmação de e-mail ao se cadastrar |
| **MSG02** | E-mail de aviso de previsão de abertura de turma para um curso |
| **MSG03** | E-mail de aviso de aproximação da data limite de inscrição |
| **MSG04** | E-mail de aviso de dados alterados |
| **MSG05** | E-mail de aviso de aprovação ou suplência |
| **MSG06** | E-mail de chamada para matrícula |
| **MSG07** | E-mail de confirmação de pagamento, ou pendência de pagamento |
| **MSG08** | E-mail de confirmação de matrícula |
| **MSG09** | Mensagem de salvamento de dados |
| **MSG10** | Mensagem de alteração de dados |
| **MSG11** | Mensagem ativação de notificações de abertura de turmas para um curso |

## Relato Detalhado

O SENAI expressou a necessidade de um sistema que deve possibilitar o cadastro e gerenciamento independente dos alunos, cursos, e trilhas de aprendizado, para a administração dos cursos de Formação Inicial e Continuada (FIC).

Os alunos devem poder se inscrever online, com validação de pré-requisitos e pagamentos, e o sistema deve fornecer relatórios detalhados sobre inscrições, pagamentos e turmas formadas. Além disso, o sistema deve facilitar a comunicação com os alunos, enviando notificações sobre status de inscrição e pagamento, e permitir a geração de documentos como PDFs de inscrições.

A gestão de turmas e a atribuição de professores devem ser realizadas de forma automatizada, com base no número de alunos inscritos e na disponibilidade dos professores. O sistema deve ter uma interface amigável e garantir a segurança e privacidade dos dados dos usuários.

O SENAI também enfatizou a importância de relatórios detalhados e a capacidade de exportá-los em formatos compatíveis, além de uma interface que siga a identidade visual da instituição, com cores definidas a serem confirmadas.

Este relatório reflete as principais necessidades e requisitos identificados durante a entrevista e servirá como base para o desenvolvimento do sistema proposto.

1. **Apêndice 2: Solicitação de Manutenção Incremental**

**Projeto:** Sistema Registro de Alunos (SRP) **Nº:**1

**Solicitante:** Sérgio Clauss **Data:** 07/10/2024

**Tipo alteração:** Incremental

**Alteração:** Solicito que incluamos uma versão do sistema desktop além do aplicativo celular e web. Pois para o administrador do sistema fica mais prático para usabilidade e com visão mais abrangente dos dados na tela.

**Analisado por:** Marlon Furlan **Data:**07/10/2024

**Componentes afetados:**

* **Interface do Usuário (UI):**
  + Design e experiência do usuário precisam ser adaptados para o ambiente desktop.
  + Diferenças de resolução, interação (mouse vs. toque) e layout.
* **Segurança:**
  + Implementação de medidas de segurança apropriadas para a aplicação desktop.
  + Autenticação e autorização compatíveis.
* **Backend/Serviços Web:**
  + APIs existentes devem ser reutilizadas ou adaptadas para a nova aplicação.
  + Sincronização e comunicação entre as plataformas.

**Componentes associados:**

* **Performance:**
  + Otimização para rodar eficientemente em diferentes hardwares de desktop.
* **Banco de Dados:**
  + Acesso e gerenciamento de dados devem ser consistentes entre as plataformas.
  + Sincronização offline (se necessário).
* **Segurança:**
  + Implementação de medidas de segurança apropriadas para a aplicação desktop.
  + Autenticação e autorização compatíveis.

**Avaliação da alteração:** Prezado Clauss, obrigado pela colaboração ativa no projeto. Mas de acordo com a entrevista realizada, no dia 17/09/2024, com o Sr. Fábio, ficou definido a necessidade de um sistema para web. Após, uma análise cuidadosa de sua sugestão e uma reunião com a equipe de desenvolvimento, conclui-se que no momento esta alteração é inviável pois não vem ao encontro com a necessidade administrativa, bem como uma alteração neste momento ocasionaria um atraso na entrega do projeto, que está programado para o dia 25/11/2024. Mas, considerando a importância da solicitação e a relevância da alteração para o projeto, tal alteração será estudada com mais atenção após a entrega do sistema.

**Prioridade:** Baixa

**Data decisão:** 07/10/2024

