



igti

# RELATÓRIO

---

## PROJETO APLICADO



Instituto de Gestão e Tecnologia da Informação  
Relatório do Projeto Aplicado

# App para controle de feedbacks de módulos do IGTI

Raphael Gomide  
Orientador(a): Raphael Gomide

06/10/2021



RAPHAEL GOMIDE

INSTITUTO DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO

RELATÓRIO DO PROJETO APLICADO

# App para controle de feedbacks de módulos do IGTI

Relatório de Projeto Aplicado desenvolvido  
para fins de conclusão do curso de MBA em  
"Desenvolvimento Full Stack".

Orientador(a): Raphael Gomide

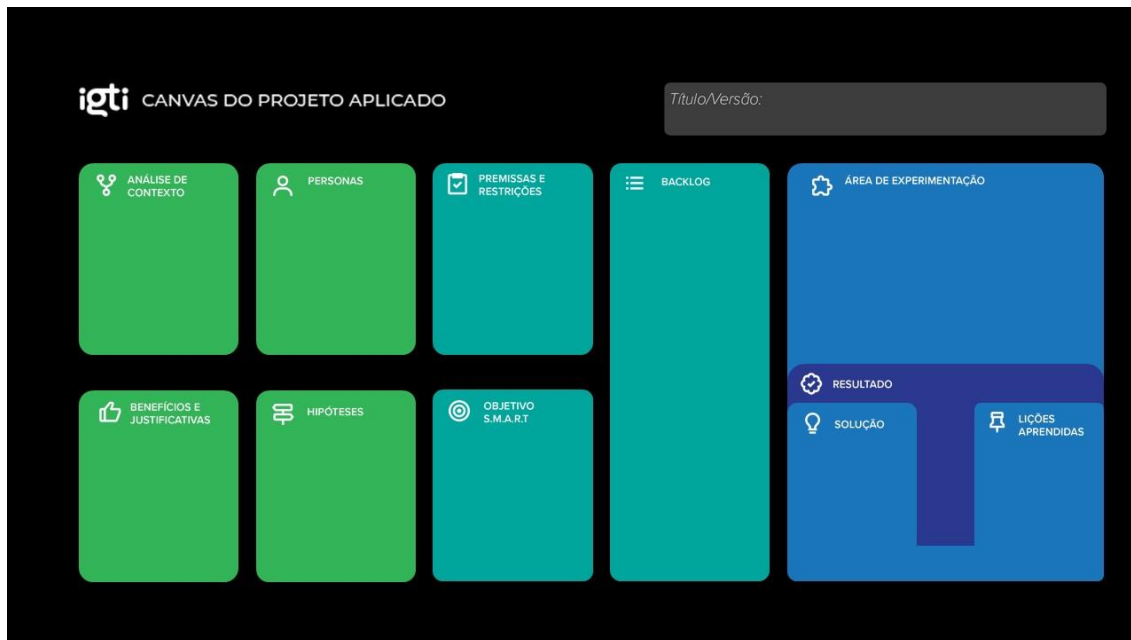
**Belo Horizonte**  
**06/10/2021**

## Sumário

|  |    |
|--|----|
| 1. CANVAS do Projeto Aplicado              | 5  |
| 1.1 Desafio                                | 6  |
| 1.1.1 Análise de Contexto                  | 6  |
| 1.1.2 Personas                             | 9  |
| 1.1.3 Benefícios e Justificativas          | 10 |
| 1.1.4 Hipóteses                            | 11 |
| 1.2 Solução                                | 13 |
| 1.2.1 Objetivo SMART                       | 13 |
| 1.2.2 Premissas e Restrições               | 13 |
| 1.2.3 Backlog de Produto                   | 14 |
| 2. Área de Experimentação                  | 15 |
| 2.1 Sprint 1                               | 15 |
| 2.1.1 Solução                              | 15 |
| • Evidência do planejamento:               | 15 |
| • Evidência da execução de cada requisito: | 16 |
| • Evidência da solução:                    | 20 |
| 2.1.2 Lições aprendidas                    | 23 |
| 2.2 Sprint 2                               | 24 |
| 2.2.1 Solução                              | 24 |
| • Evidência do planejamento:               | 24 |
| • Evidência da execução de cada requisito: | 24 |
| • Evidência da solução:                    | 26 |
| 2.2.2 Lições aprendidas                    | 28 |
| 2.3 Sprint 3                               | 30 |
| 2.3.1 Solução                              | 30 |
| • Evidência do planejamento:               | 30 |
| • Evidência da execução de cada requisito: | 30 |
| • Evidência da solução:                    | 32 |
| 2.3.2 Lições aprendidas                    | 33 |
| 3. Considerações Finais                    | 34 |
| 3.1 Resultados                             | 34 |
| 3.2 Contribuições                          | 35 |
| 3.3 Próximos passos                        | 36 |

# 1. CANVAS do Projeto Aplicado

Figura conceitual, que representa todas as etapas do Projeto Aplicado.



## 1.1 Desafio

### 1.1.1 Análise de Contexto

O [IGTI](#) (Instituto de Gestão e Tecnologia da Informação) é uma das referências em ensino à distância de tecnologia no Brasil e trabalha com diversos cursos e, portanto, diversas professoras/professores. O IGTI utiliza atualmente um sistema de [código aberto](#), o [Canvas LMS](#), para gerenciar seus cursos, turmas, corpo discente e corpo docente.

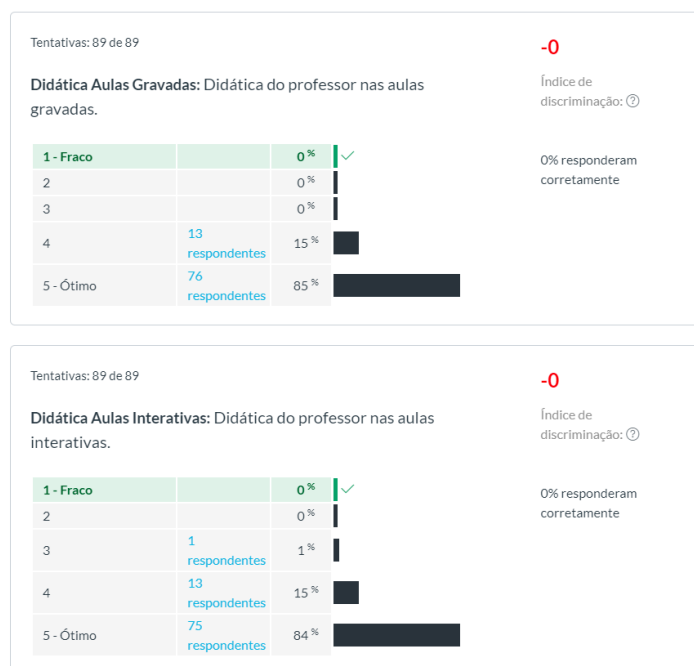
Um dos módulos do Canvas é o sistema de **feedbacks**, que é *alimentado* pelas alunas/alunos a cada oferta de disciplina e/ou módulo de cursos. Esses feedbacks fazem parte dos principais indicadores que afetam diretamente no desempenho das professoras/professores. Em resumo, o feedback consiste em algumas questões onde as alunas/alunos, de forma anônima, dão nota de 1 a 5 em diversos critérios tais como:

- Didática nas aulas gravadas
- Didática nas aulas interativas
- Qualidade da Apostila
- Relevância do Desafio
- Relevância do Trabalho Prático

Além disso, os alunos podem descrever, através de texto livre, sobre dois outros aspectos importantes:

- Pontos positivos da disciplina/módulo
- Pontos de melhoria para as próximas ofertas da disciplina/módulo

Em linhas gerais, a professora/professor tem acesso ao feedback ao final da disciplina/módulo e o Canvas possui uma interface visual para que seja possível visualizar o feedback e analisar o desempenho, conforme imagem abaixo:



Interface do sistema Canvas para visualização de feedbacks por parte das professoras/professores

Entretanto, percebe-se que o sistema de feedbacks provido por padrão pelo Canvas possui os seguintes problemas:

- Não há uma ligação entre as ofertas, ou seja, a professora/professor não consegue, via sistema, realizar um comparativo de sua evolução na disciplina/módulo.
- Poucas opções de exportação dos dados. Atualmente, é possível exportar os dados apenas no formato [CSV](#) (*Comma-Separated Values*).
- Período limitado para obtenção do arquivo CSV: o arquivo fica disponível para download somente durante 30 dias.
- Apesar do formato CSV ser compatível tanto com editores de texto simples como *software* de planilhas como o Microsoft Excel e o LibreOffice Calc, o arquivo possui uma difícil interpretação dos dados, conforme imagem abaixo:

| 1  | section          | section_id   | submitted               | attempt | 388208: Autoavaliação: Seu grau de dedicação e envolvimento na disciplina. | 388209: Didática: Didática do professor nas aulas gravadas e aulas interativas. |
|----|------------------|--------------|-------------------------|---------|--|---|
| 2  | Turma Extra      | 19657        | 2021-08-03 02:19:27 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 3  | DFS211A          | 19548        | 2021-08-03 02:11:42 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 4  | Turma Extra      | 19657        | 2021-08-03 01:44:07 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 5  | DFS202A          | 19577        | 2021-08-03 01:02:17 UTC | 1       | 5 - Ótimo  | 2.0 5 - Ótimo   |
| 6  | DFS202A          | 19577        | 2021-08-03 00:59:22 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 7  | DFS205A, DFS202A | 19549, 19577 | 2021-08-03 00:57:49 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 8  | DFS202A, DFS205A | 19577, 19549 | 2021-08-03 00:57:29 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 9  | DFE211A          | 19547        | 2021-08-03 00:55:18 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 10 | DFS211A          | 19548        | 2021-08-03 00:54:18 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 11 | DFS211A          | 19548        | 2021-08-03 00:52:38 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 12 | DFS202A, DFS205A | 19577, 19549 | 2021-08-02 23:56:44 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 13 | DFS211A          | 19548        | 2021-08-02 23:32:35 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 14 | DFS211A          | 19548        | 2021-08-02 23:31:30 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 15 | DFS211A          | 19548        | 2021-08-02 23:11:08 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 16 | DFE211A          | 19547        | 2021-08-02 23:08:43 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 17 | DFS211A          | 19548        | 2021-08-02 23:02:32 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 18 | DFS211A          | 19548        | 2021-08-02 22:47:14 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 19 | DFS211A          | 19548        | 2021-08-02 22:18:09 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 20 | DFS202A          | 19577        | 2021-08-02 22:16:22 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 21 | DFS211A          | 19548        | 2021-08-02 22:13:39 UTC | 1       | 5 - Ótimo  | 4.0 5 - Ótimo   |
| 22 | DFS211A          | 19548        | 2021-08-02 21:44:17 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 23 | DFS202A, DFS205A | 19577, 19549 | 2021-08-02 21:19:26 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 24 | DFS202A, DFS205A | 19577, 19549 | 2021-08-02 21:15:41 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 25 | DFS202A, DFS205A | 19577, 19549 | 2021-08-02 21:09:04 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 26 | Turma Extra      | 19657        | 2021-08-02 21:04:16 UTC | 1       | 5 - Ótimo  | 2.0 5 - Ótimo   |
| 27 | DFS202A          | 19577        | 2021-08-02 20:56:26 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 28 | DFS211A          | 19548        | 2021-08-02 20:48:32 UTC | 1       | 5 - Ótimo  | 2.0 5 - Ótimo   |
| 29 | DFS211A          | 19548        | 2021-08-02 20:25:13 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 30 | DFS202A, DFS205A | 19577, 19549 | 2021-08-02 19:21:47 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 31 | DFS202A, DFS205A | 19577, 19549 | 2021-08-02 19:13:18 UTC | 1       | 5 - Ótimo  | 0.0 5 - Ótimo   |
| 32 | DFS202A          | 19577        | 2021-08-02 18:45:05 UTC | 1       | 5 - Ótimo  | 3.0 5 - Ótimo   |
| 33 | Turma Extra      | 19657        | 2021-08-02 15:57:51 UTC | 1       | 5 - Ótimo  | 2.0 5 - Ótimo   |

Trecho de um arquivo CSV de feedback exportado pelo Canvas e aberto no LibreOffice Calc

Após as análises acima, é possível identificar a **dor** da professora/professor: como utilizar o feedback das alunas/alunos para evoluir a didática em médio/longo prazos a partir do comparativo desses feedbacks agrupados por módulos/disciplinas? Além disso, como ter um controle dos feedbacks textuais das alunas/alunos para melhoria contínua?

Para melhor *ilustrar* essa **dor**, seguem duas ferramentas importantes: Matriz CSD e Observação POEMS:



Matriz CSD





## Observação POEMS

### 1.1.2 Personas

O foco do problema são as professoras, professores e também as coordenadoras e coordenadores de curso do IGTI, que utilizam o Canvas mas não possuem, atualmente, uma forma prática para controlar os feedbacks recebidos de suas disciplinas/módulos. Para uma melhor ilustração, será apresentado um Mapa de Empatia.



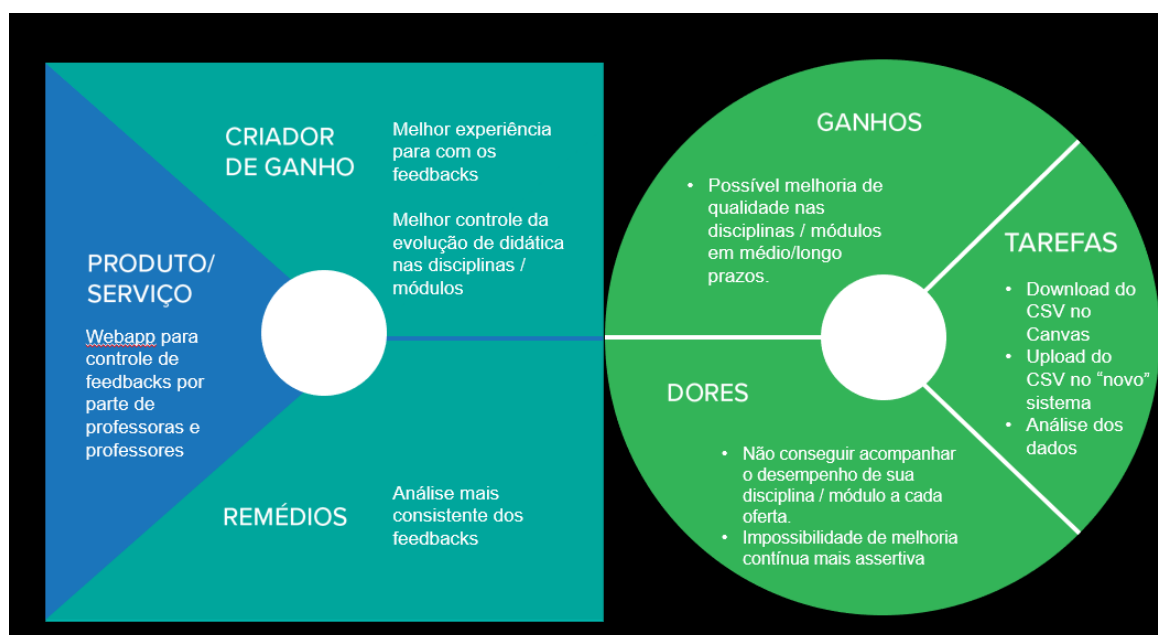
Mapa de Empatia

### 1.1.3 Benefícios e Justificativas

Para uma análise mais assertiva sobre os Benefícios e Justificativas deste Projeto Aplicado, serão utilizadas as ferramentas **Blueprint** e **Canvas de Proposta de Valor**, conforme imagens abaixo:

| Ações dos clientes          | Obter feedback   | Analisar o feedback   |
|-----------------------------|--|---|
| <b>Objetivos</b>            | Verificar como foi o andamento da disciplina / módulo                                      | Melhoria contínua no ensino   |
| <b>Atividades</b>           | Realizar o download do arquivo CSV referente ao feedback                                   | Cadastrar o arquivo CSV no "novo" sistema   |
| <b>Questões</b>             | Não se aplica  | Meu feedback foi satisfatório?  |
| <b>Barreiras</b>            | A professora / professor tem somente 30 dias para realizar o download do arquivo CSV       | Permissão de acesso ao "novo" sistema   |
| <b>Funcionalidades</b>      | Não se aplica  | Cadastro do CSV, análise qualitativa do feedback, comparativo de feedbacks com ofertas anteriores |
| <b>Interação</b>            | Acessar o sistema Canvas e realizar o download do arquivo CSV da disciplina / módulo       | Acessar o "novo" sistema e fazer o upload do arquivo CSV obtido através do Canvas                 |
| <b>Onde ocorre</b>          | Qualquer local com acesso à Internet   | Qualquer local com acesso à Internet  |
| <b>Tarefas aparentes</b>    | Não se aplica  | Estudar o formato CSV, estudar infra-estrutura  |
| <b>Tarefas escondidas</b>   | Não se aplica  | Implementação do Sistema com Node.js, React e MongoDB.  |
| <b>Processos de suporte</b> | Reforçar às professoras e professores sobre a importância de se obter o feedback do Canvas | Comunicação às professoras e professores sobre o novo sistema                                     |
| <b>Saída desejável</b>      | Hábito de obtenção do feedback ao final de cada oferta                                     | Hábito de cadastrar o feedback no "novo" sistema e posterior análise para melhoria contínua       |

Blueprint



Canvas da Proposta de Valor

#### 1.1.4 Hipóteses

Seguem algumas **hipóteses** com respeito ao problema dos feedbacks:

- Nem todas as professoras e professores acessam ou conhecem o caminho para acessar o seu feedback.
- Nem todas as professoras e professores se lembram do último feedback para realizar uma comparação com o feedback atual.
- Nem todas as professoras e professores conseguem medir se estão em evolução com suas disciplinas/módulos.
- O processo atual de visualização dos feedbacks no sistema Canvas não é muito intuitivo.

Seguem algumas **ideias** para resolver as hipóteses:

11. Alteração no sistema Canvas para comportar um histórico de feedbacks.
12. Criar um sistema à parte focado nos feedbacks.

Seguem alguns **critérios** referentes às ideias, com base na matriz BASICO:

- Benefícios
- Abrangência
- Satisfação
- Investimentos
- Cliente
- Operacionalidade

A partir das hipóteses e ideias acima, foi montada a seguinte **Matriz de Priorização de Ideias**:

| Ideias | Critérios de comparação |   |   |   |   |   | Somatório |
|--------|-------------------------|---|---|---|---|---|-----------|
|        | B                       | A | S | I | C | O |           |
| I1     | 4                       | 5 | 5 | 1 | 2 | 1 | 18        |
| I2     | 4                       | 4 | 4 | 3 | 5 | 3 | 23        |

Matriz de priorização de ideias

## 1.2 Solução

### 1.2.1 Objetivo SMART

Criar, em até 2 meses, desenvolvendo por cerca de 2 horas/dia, um MVP de uma aplicação web para controlar os feedbacks das professoras e professores, agrupados por módulo.

### 1.2.2 Premissas e Restrições

Para este projeto, foram identificadas as seguintes premissas:

- Serão utilizadas cerca de 2 horas diárias para o desenvolvimento do MVP.
- O MVP não vai englobar a funcionalidade de login, considerando que não é o foco principal do MVP.

Para este projeto, foram identificadas as seguintes restrições:

- O sistema será desenvolvido, a princípio, somente para a plataforma web.
- O sistema a ser desenvolvido **não** deve interferir no Canvas, que é o sistema acadêmico oficial do IGTI.
- O MVP do sistema deve ser desenvolvido durante o Projeto Aplicado, cuja duração gira em torno de 2 meses.

Com base nas premissas e restrições acima, foram identificados, conforme a **Matriz de Riscos** abaixo, os seguintes riscos:

| Risco identificado                 | Impacto potencial     | Ações preventivas        | Ações corretivas           |
|------------------------------------|-----------------------|--------------------------|----------------------------|
| Mudanças nos critérios do feedback | Sistema inconsistente | Parametrização de regras | Alteração no código-fonte  |
| Descontinuidade do sistema Canvas  | Sistema inútil        | Parametrização de regras | Reimplementação do sistema |

Matriz de riscos

### 1.2.3 Backlog de Produto

Segue o backlog do MVP proposto com base nas análises feitas durante as etapas anteriores:



Backlog do produto

## 2. Área de Experimentação

### 2.1 Sprint 1

O foco da Sprint 1 foi na definição de tecnologias a serem desenvolvidas e implementação de algumas funcionalidades mais básicas para o MVP.

Considerando o meu tempo de dedicação disponível para o projeto, preferi me manter na minha "zona de conforto" e utilizar somente tecnologias às quais tenho mais experiência. Sendo assim, as tecnologias escolhidas para a implementação do MVP são:

- [React.js](#) para o Front End
- [Node.js](#) para o Back End

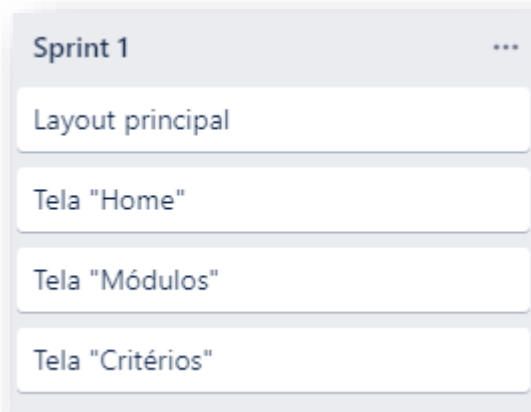
Considerando que esta Sprint tem foco maior no Front End da aplicação, foram também definidas as tecnologias mais específicas para tal. Sendo assim, a aplicação Front End será implementada com base na ferramenta [Create React App](#), muito utilizada para a construção de *Single Page Applications* com o React. Além disso, serão utilizadas as seguintes tecnologias auxiliares:

- Estilização de CSS com [Tailwind CSS](#)
- Roteamento com [react-router-dom](#)
- Ícones com [react-icons](#)
- Indicadores de carregamento de dados com [react-spinners](#)

#### 2.1.1 Solução

- Evidência do planejamento:

Para a Sprint 1, foram alocadas as seguintes tarefas:



Tarefas alocadas na Sprint 1

- Evidência da execução de cada requisito:

**Tarefa *Layout principal*:**

Utilizando a biblioteca **react-router-dom**, foi criada uma estrutura simples de navegação de forma dinâmica, contendo todas as páginas a serem acessadas. A imagem abaixo demonstra um *array* contendo todas as rotas da aplicação, cujo acesso será visualizado no layout principal:



```
export const APP_ROUTES = [
  {
    id: idServiceNewId(),
    path: '/',
    description: 'Home',
    icon: <HomeIcon />,
    component: <HomePage />,
  },
  {
    id: idServiceNewId(),
    path: '/modules',
    description: 'Módulos',
    icon: <NewIcon />,
    component: <ModulesPage />,
  },
  {
    id: idServiceNewId(),
    path: '/criteria',
    description: 'Critérios',
    icon: <CheckIcon />,
    component: <CriteriaPage />,
  },
  {
    id: idServiceNewId(),
    path: '/upload',
    description: 'Upload',
    icon: <UploadIcon />,
    component: <ImportCsvPage />,
  },
  {
    id: idServiceNewId(),
    path: '/reports',
    description: 'Relatórios',
    icon: <ReportIcon />,
    component: <ReportsPage />,
  },
];
```

Rotas da aplicação

Já a imagem abaixo ilustra os componentes referentes ao layout principal:

```
export default function App() {
  return (
    <>
      <BrowserRouter>
        <Header title="Feedback IGTI">
          <Routes />
        </Header>

        <Main>
          <Pages />
        </Main>
      </BrowserRouter>
    </>
  );
}
```

Layout principal da aplicação

Tarefa Tela "Home":

Foi criado um componente React cujo conteúdo explica o funcionamento da aplicação, conforme imagem abaixo:

```
export default function HomePage() {
  return (
    <div className="flex flex-col space-y-2 p-4">
      <h3 className="text-center font-semibold mb-6">
        Seja bem-vinda(o) ao sistema de feedbacks!
      </h3>

      <ul style={{ listStyleType: 'square' }}>
        <li>
          Acesse <strong>"Módulos"</strong> para cadastrar as suas
          disciplinas/módulos dos cursos.
        </li>
        <li>
          Acesse <strong>"Critérios"</strong> para cadastrar os critérios que
          impactam em seu feedback.
        </li>
      </ul>
    </div>
  );
}
```

Componente React referente à tela "Home"

#### Tarefa Tela "Módulos":

Foi criado um componente React cujo conteúdo permite a exibição, criação, edição e exclusão de registros referentes aos módulos (disciplinas) que a professora ou professor lecionam. A imagem abaixo ilustra a renderização do componente:

```

<SimpleCrudContainer
  itemDescription="módulo"
  itemDescriptionPlural="módulo(s)"
  onNewItem={handleModuleCreate}
>
  {modules.map(item => {
    return (
      <SimpleCrudItem
        key={item.id}
        onConfirmEdit={handleModuleEdit}
        onDelete={handleModuleDelete}
      >
        {item}
      </SimpleCrudItem>
    );
  })}
</SimpleCrudContainer>

```

Componente React referente à tela "Módulos"

#### Tarefa Tela "Critérios":

Foi criado um componente React cujo conteúdo permite a exibição, criação, edição e exclusão de registros referentes aos critérios de avaliação dos feedbacks. Como a estrutura de dados tanto de "Critérios" quanto de "Módulos" é muito semelhante, foi possível reaproveitar partes do código-fonte, conforme imagem abaixo:

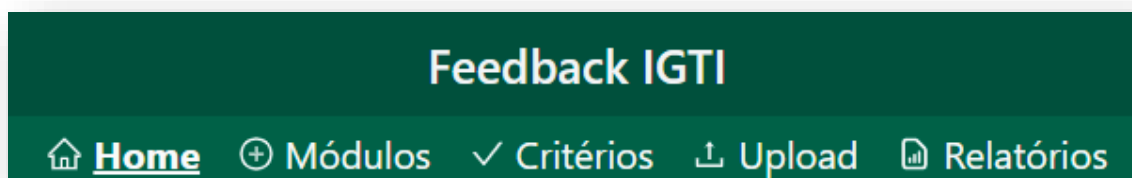
```
<SimpleCrudContainer
  · itemDescription="critério"
  · itemDescriptionPlural="critério(s)"
  · onNewItem={handleCriteriumCreate}
>
  · {criteria.map(item => {
  ·   · return (
  ·     · <SimpleCrudItem
  ·       · key={item.id}
  ·       · onConfirmEdit={handleCriteriumEdit}
  ·       · onDelete={handleCriteriumDelete}
  ·     · >
  ·       · {item}
  ·     · </SimpleCrudItem>
  ·   · );
  · })}
</SimpleCrudContainer>
```

Componente React referente à tela "Critérios", muito semelhante à tela de "Módulos".

- Evidência da solução:

Tarefa *Layout principal*:

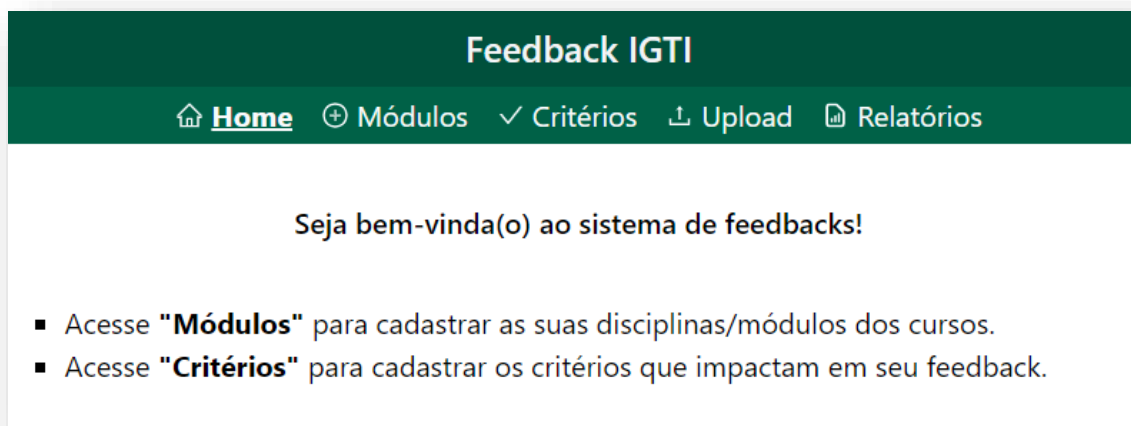
A renderização do layout principal a partir do código-fonte é feita conforme imagem abaixo:



Renderização do layout principal do App

### Tarefa Tela "Home":

A renderização da página *Home* a partir do código-fonte é feita conforme imagem abaixo:



Renderização da página *Home*

### Tarefa Tela "Módulos":

A renderização da página *Módulos* a partir do código-fonte é feita conforme imagem abaixo:

Feedback IGTI

[Home](#)
[Módulos](#)
[Critérios](#)
[Upload](#)
[Relatórios](#)

2 módulo(s) encontrado(s)

Novo módulo:

Módulo(s) cadastrado(s):

|                         |                                     |
|-------------------------|-------------------------------------|
| React I                 | <a href="#">✎</a> <a href="#">✖</a> |
| Desenvolvimento Reativo | <a href="#">✎</a> <a href="#">✖</a> |

Renderização da página *Módulos*

### Tarefa Tela "Critérios":

A renderização da página *Critérios* a partir do código-fonte é feita conforme imagem abaixo:

Feedback IGTI

[Home](#)
[Módulos](#)
[Critérios](#)
[Upload](#)
[Relatórios](#)

5 critério(s) encontrado(s)

Novo critério:

Critério(s) cadastrado(s):

|                   |                                     |
|-------------------|-------------------------------------|
| Aulas gravadas    | <a href="#">✎</a> <a href="#">✖</a> |
| Aulas interativas | <a href="#">✎</a> <a href="#">✖</a> |
| Apostila          | <a href="#">✎</a> <a href="#">✖</a> |
| Trabalho Prático  | <a href="#">✎</a> <a href="#">✖</a> |
| Desafio           | <a href="#">✎</a> <a href="#">✖</a> |

Renderização da página *Critérios*

### 2.1.2 Lições aprendidas

Durante a Sprint 1, foram aprendidas as seguintes *lições*:

- É, de fato, muito comum a constante reorganização do backlog do produto à medida em que estudamos mais sobre o Desafio e Solução.
- Em um MVP, acredito ser melhor começar a implementação pelo Front End, considerando que o foco é o produto e não necessariamente o código-fonte. Não adianta implementar o Back End se não der tempo de implementar o Front End e, com isso, demonstrar somente a exibição de *rotas* no *pitch*.
- O React é uma excelente ferramenta para o reaproveitamento de código. Foi possível utilizar desse recurso para implementar duas das tarefas desta Sprint - a tela *Módulos* e a tela *Critérios*.

## 2.2 Sprint 2

O foco da Sprint 2 foi no cadastro dos feedbacks, que é feito através do *upload* do arquivo CSV e da consequente associação dos dados a uma disciplina/módulo. Além disso, a professora/professor pode realizar a associação das perguntas do feedback a determinado critério de avaliação. É importante frisar que tanto a disciplina/módulo quanto os critérios de avaliação devem estar previamente cadastrados (isso foi abordado na Sprint 1).

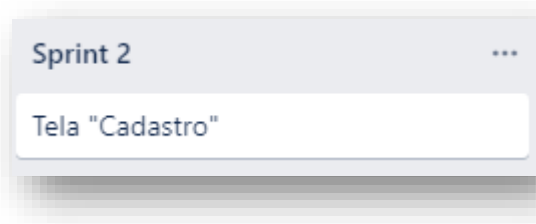
Considerando o meu tempo de dedicação disponível para o projeto, continuei em minha "zona de conforto" e utilizei somente tecnologias às quais tenho mais experiência. Sendo assim, as tecnologias escolhidas para a implementação da Sprint 2 foram:

- Upload e processamento de arquivos CSV com o [react-papaparse](#).
- Utilização de abas com o [react-tabs](#).
- Utilização de mensagens amigáveis ao usuário com o [react-toastify](#).
- Operações com datas com a biblioteca [date-fns](#).
- Operações com objetos com a biblioteca [lodash](#).

### 2.2.1 Solução

- Evidência do planejamento:

Para a Sprint 2, foram alocadas as seguintes tarefas:



Tarefas alocadas na Sprint 2

- Evidência da execução de cada requisito:

Tela "Cadastro":



Foi criado um novo componente do tipo "Page" para comportar os componentes visuais e de interação. A imagem abaixo demonstra, de forma resumida, o código-fonte do *layout* JSX desta tela.

```
<>
  <ToastContainer
    autoClose={2000} ...
    rtl={false}
  />

  <Tabs>
    <TabList>
      <Tab>Upload de CSV</Tab>
      <Tab>Listagem de feedbacks ({backendFeedback.length})</Tab>
    </TabList>

    <TabPanel>
      <div className="flex flex-col items-center justify-center">
        <CSVReader ...

        {loadingCsv && <CenteredLoading />}
        {isSavingFeedback && <CenteredLoading />}

        {currentFeedback.questions.length > 0 && !isSavingFeedback && (
          <div className="container">...
        )}
      </div>
    </TabPanel>

    <TabPanel>
      <FeedbackList onDelete={handleDeleteFeedback}>
        {backendFeedback}
      </FeedbackList>
    </TabPanel>
  </Tabs>
</>
```

Trecho de código-fonte da tela "Cadastro"

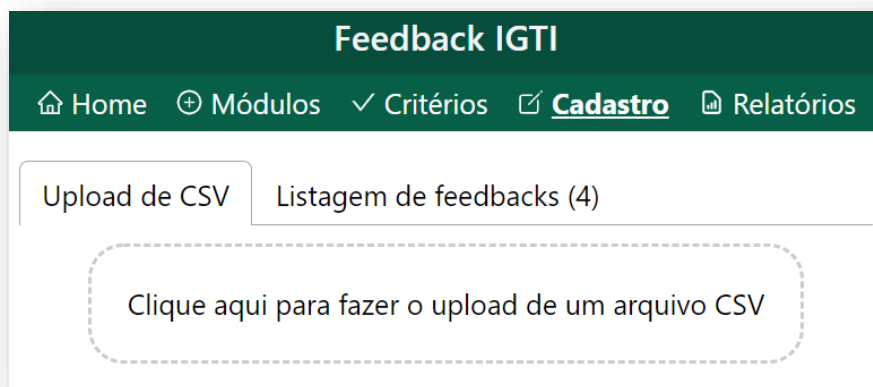
Em resumo:

- **ToastContainer:** refere-se ao componente para mensagens amigáveis
- **Tabs, TabList e TabPanel:** referem-se aos componentes de abas para uma divisão mais lógica e visual do conteúdo. O conteúdo foi basicamente dividido em "Upload do arquivo CSV" e "Listagem dos feedbacks".
- Foram feitas várias implementações de componentes, como por exemplo o **FeedbackList**.

- Evidência da solução:

Tela "Cadastro":

A renderização da tela "Cadastro" a partir do código-fonte será demonstrada nas diversas imagens abaixo:



Renderização inicial da tela "Cadastro"

Módulo a ser associado:  
React I

Data da oferta do módulo:  
12/09/2021

Salvar Cancelar

Notas

| Didática Aulas Gravadas: Didática do professor nas aulas gravadas |        | Didática Aulas Interativas: Didática do professor nas aulas interativas |        |
|---|--------|---|--------|
| [0] 1,25% (1)   |        | [0] 1,25% (1)   |        |
| [1] 0% (0)  | 97,50% | [1] 0% (0)  | 93,75% |
| [2] 0% (0)  | 0%     | [2] 0% (0)  | 0%     |
| [3] 1,25% (1)   |        | [3] 5% (4)  |        |
| [4] 16,25% (13)   |        | [4] 15% (12)  |        |
| [5] 81,25% (65)   |        | [5] 78,75% (63)   |        |

Crítério: Aulas gravadas

Crítério: Aulas interativas

Renderização após o upload do arquivo CSV

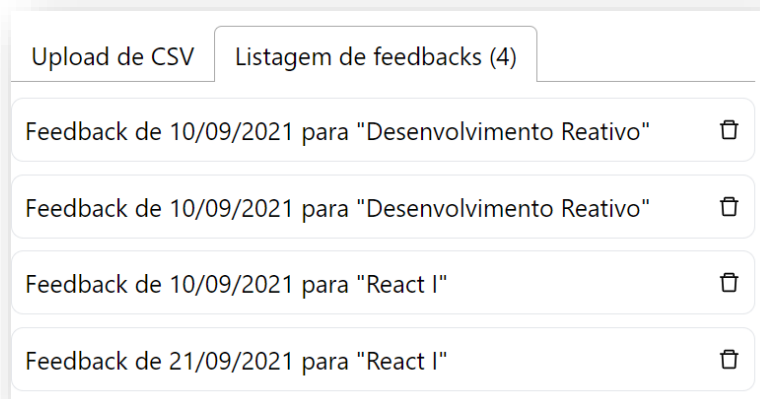
Feedback cadastrado com sucesso!

Home Módulos

Upload de CSV Listagem de feedbacks (5)

Aguarde...

Mensagem amigável após o cadastro



Listagem de feedbacks cadastrados, com possibilidade de exclusão

### 2.2.2 Lições aprendidas

Apesar de conter somente uma tarefa, esta Sprint foi a mais desafiadora até o momento. Destaco as seguintes lições aprendidas:

- Para a página, foi necessária a criação de diversas variáveis de estado. Este é um bom argumento para a utilização de **useReducer** ao invés de **useState**. Além disso, seria uma ótima oportunidade para a criação de um **hook customizado**. Entretanto, devido ao tempo limitado, optou-se por não fazer isso neste momento.
- Foi utilizado o recurso **Promise.all** para execução "paralela" de requisições HTTP. Este recurso é útil principalmente para requisições independentes. Além disso, a utilização de **Promise.all** pode levar a uma melhor performance da aplicação.
- Foi utilizado também o hook **useRef** para referenciar o componente de upload de CSV (**CSVReader**). Assim foi possível realizar a remoção do arquivo CSV após o usuário confirmar o cadastro, de forma imperativa, mantendo o fluxo correto da aplicação e evitando uso desnecessário de memória.
- Foi utilizado o método **cloneDeep** da biblioteca **lodash** para clonar o objeto inicial que representa o feedback. Isso foi necessário porque o objeto em

si contém arrays, e isso faz com que uma cópia mais comum (com o spread operator `{...}`) não seja suficiente. Isso é conhecido como *shallow copy*.

## 2.3 Sprint 3

O foco da Sprint 3 foi o relatório que é gerado a partir dos feedbacks cadastrados.

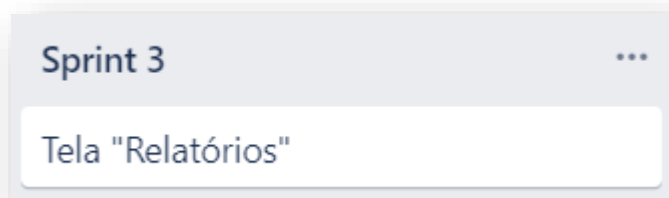
Considerando o meu tempo de dedicação disponível para o projeto, continuei em minha "zona de conforto" e utilizei somente tecnologias às quais tenho mais experiência. Sendo assim, as tecnologias escolhidas para a implementação da Sprint 3 foram:

- Exibição e processamento de gráficos com o [react-google-charts](https://react-google-charts.com/).

### 2.3.1 Solução

- Evidência do planejamento:

Para a Sprint 3, foram alocadas as seguintes tarefas:



Tarefas da Sprint 3

- Evidência da execução de cada requisito:

**Tela "Relatórios":**

Foi criado um novo componente do tipo *"Page"* para comportar os componentes visuais e de interação. A imagem abaixo demonstra, de forma resumida, o código-fonte do *layout JSX* desta tela.

```
<>
  <Select
    labelDescription="Escolha um dos módulos:"
    selectValue={selectedModule}
    onSelectChange={handleModuleChange}
  >
    {modules}
  </Select>

  <div className="flex flex-row items-center justify-center">
    <BarChart
      title={`Feedback de ${moduleDescription}`}
      verticalLabel="Data da oferta"
      horizontalLabel="%"
      labels={['% de aprovação', '% de rejeição']}
    >
      {filteredReports}
    </BarChart>
  </div>
</>
```

Código JSX referente à tela "Relatórios"

Foi criado também o componente "BarChart" que abstrai um dos gráficos do React Google Charts. A imagem abaixo demonstra, de forma resumida, o código-fonte do *layout* JSX deste componente.

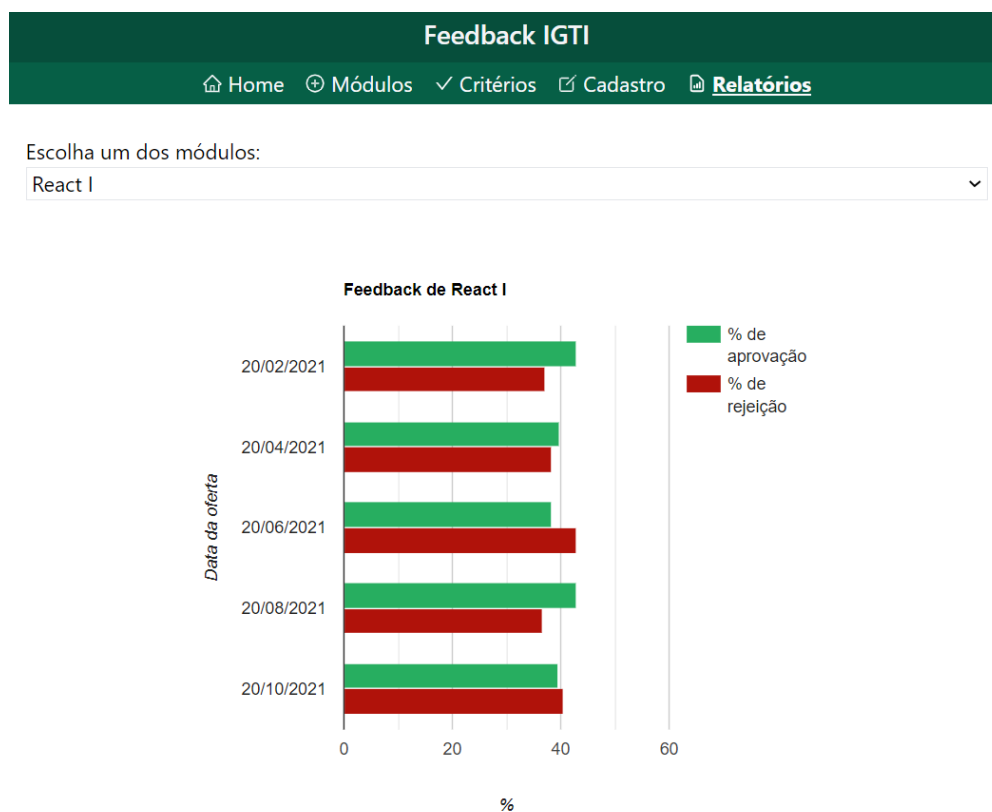
```
<Chart
  width={'500px'}
  height={'500px'}
  chartType="BarChart"
  loader=<div>{loadingMessage}</div>
  data={chartData}
  options={{
    title,
    chartArea: { width: '50%' },
    colors: ['#27ae60', '#b0120a'],
    hAxis: {
      title: horizontalLabel,
      minValue: 0,
    },
    vAxis: {
      title: verticalLabel,
    },
  }}
  // For tests
  rootProps={{ 'data-testid': '1' }}
/>
```

Código JSX referente ao componente "BarChart"

- Evidência da solução:

Tela "Cadastro":

Segue uma imagem referente à renderização da tela "Relatórios".



Renderização da tela "Relatórios"



### 2.3.2 Lições aprendidas

Imaginei que esta Sprint seria a mais complexa. Entretanto, considero que a Sprint 2 foi, de fato, a mais desafiadora. Destaco as seguintes lições aprendidas:

- É importante e interessante a técnica de abstração de componentes. Neste caso, abstraí o componente "Chart" de react-google-charts para um componente BarChart com algumas *props* pré-definidas. Isso pode facilitar o trabalho do desenvolvedor.
- Acredito que o app atendeu as expectativas para um MVP. Entretanto, ainda há muito o que ser feito no app quanto a funcionalidades. Essas tarefas serão descritas na próxima seção deste documento.

## 3. Considerações Finais

### 3.1 Resultados

Principais resultados alcançados pelo Projeto Aplicado:

- Implementação de um MVP utilizando tecnologia moderna (Node.js + React).
- "Dor" sanada parcialmente pelo MVP.
- Muitas possibilidades para a evolução do app.

Pontos positivos:

- Muito aprendizado.
- Aumento de conhecimento sobre as tecnologias envolvidas.
- Sensação de "dever cumprido".

Pontos negativos:

- Não foi possível implementar tudo o que foi planejado na ideia inicial.

Dificuldades enfrentadas:

- Adequar-se ao tempo disponível para a implementação do MVP.

Lições aprendidas:

- O estudo é constante e "eterno".
- Há muito mais para ser desenvolvido do que o que planejamos.

## 3.2 Contribuições

Destaco as seguintes contribuições:

- O MVP pode ser considerado "melhor" que o Canvas em se tratando da funcionalidade "feedback".
- As professoras e professores já podem fazer uso do MVP para avaliarem os seus respectivos módulos. Entretanto, por enquanto isso é possível somente de forma "individualizada", ou seja, cada professora/professor deve efetuar a instalação do app localmente em seu próprio computador.
- O MVP permite que novas funcionalidades sejam implementadas mais rapidamente.

### 3.3 Próximos passos

Ainda há muito o que ser feito, como por exemplo:

- Implementação de um backend mais "robusto".
- Integração com Bancos de Dados.
- Implementação de autenticação.
- Implantação do app em nuvem.
- Funcionalidade para controle dos "pontos positivos" e "pontos a melhorar".
- Integração com a API do Canvas.