Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

**Instituto de Gestão e Tecnologia da Informação**

**Relatório do Projeto Aplicado**

App de Gerenciamento para Psicólogos

Marcelo José Aragão Ramos

Orientador(a): Professor Bruno Augusto Teixeira

**** 03/06/2022

**MARCELO JOSÉ ARAGÃO RAMOS**

**INSTITUTO DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO**RELATÓRIO DO PROJETO APLICADO

App de Gerenciamento para Psicólogos

Relatório de Projeto Aplicado desenvolvido para fins de conclusão do curso MBA em Desenvolvimento Full Stack.  
  
Orientador (a): Bruno Augusto Teixeira

**Fortaleza  
09/06/2022**

**Sumário**

[1. CANVAS do Projeto Aplicado 4](#_Toc101281510)

[1.1 Desafio 5](#_Toc101281511)

[1.1.1 Análise de Contexto 5](#_Toc101281512)

[1.1.2 Personas 7](#_Toc101281513)

[1.1.3 Benefícios e Justificativas 8](#_Toc101281514)

[1.1.4 Hipóteses 9](#_Toc101281515)

[1.2 Solução 10](#_Toc101281516)

[1.2.1 Objetivo SMART 10](#_Toc101281517)

[1.2.2 Premissas e Restrições 10](#_Toc101281518)

[1.2.3 Backlog de Produto 11](#_Toc101281519)

[2. Área de Experimentação 12](#_Toc101281520)

[2.1 Sprint 1 12](#_Toc101281521)

[2.1.1 Solução 12](#_Toc101281522)

[● Evidência do planejamento: 12](#_Toc101281523)

[● Evidência da execução de cada requisito: 13](#_Toc101281524)

[● Evidência dos resultados: 17](#_Toc101281525)

[2.1.2 Experiências vivenciadas 18](#_Toc101281526)

[2.2 Sprint 2 20](#_Toc101281527)

[2.2.1 Solução 20](#_Toc101281528)

[● Evidência do planejamento: 20](#_Toc101281529)

[● Evidência da execução de cada requisito: 20](#_Toc101281530)

[● Evidência dos resultados: 20](#_Toc101281531)

[2.2.2 Experiências vivenciadas 23](#_Toc101281532)

[2.3 Sprint 3 24](#_Toc101281533)

[2.3.1 Solução 24](#_Toc101281534)

[● Evidência do planejamento: 24](#_Toc101281535)

[● Evidência da execução de cada requisito: 24](#_Toc101281536)

[● Evidência dos resultados: 27](#_Toc101281537)

[2.3.2 Experiências vivenciadas 28](#_Toc101281538)

[3. Considerações Finais 29](#_Toc101281539)

[3.1 Resultados 29](#_Toc101281540)

[3.2 Contribuições 29](#_Toc101281541)

[3.3 Próximos passos 29](#_Toc101281542)

## 1. CANVAS do Projeto Aplicado

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



## Desafio

### 1.1.1 Análise de Contexto

Com a chegada do COVID-19 e a população tendo de ficar mais tempo em suas casas por conta do lockdown mundial, acabou trazendo à tona muitos problemas psicológicos para diversas pessoas, como ansiedades, depressões dentre outros.

A figura do psicólogo nessa fase é de essencial necessidade para todos que precisam da ajuda. Porém muitos desses profissionais acabaram também tendo que trabalhar de suas próprias casas em formato de home-office, fazendo com que as organizações de sessões e controle financeiro ficassem a cargo deles.

Muitos acabavam fazendo o uso de aplicativos ou de planilhas, porém sem integração e sem unificação nenhuma. Exemplo de controle de sessão é o que demonstra a Figura 1, onde tem o aplicativo do Google Calendar.

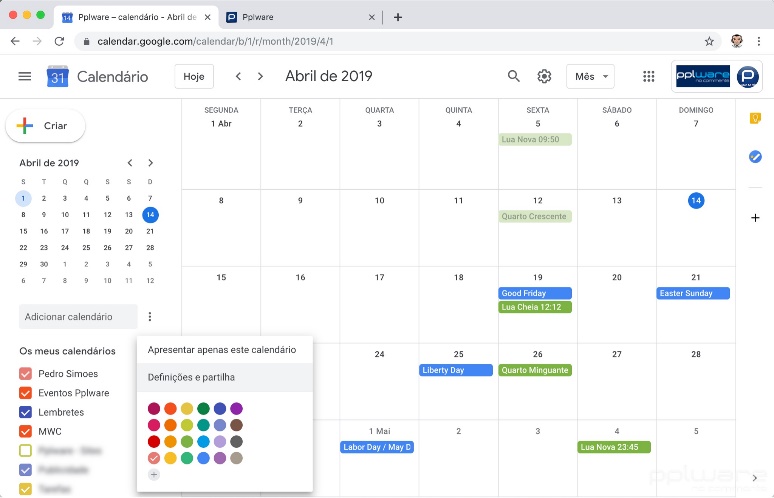


Figura - Aplicativo Google Calendar

Tabela, Excel

Descrição gerada automaticamenteExemplo de controle financeiro ficava a cargo de planilhas em excel, conforme a Figura 2.

Figura - Planilha do Excel

Analisando o formato de trabalho e controle desses profissionais, percebe-se uma grande necessidade de controle e gerenciamento de suas tarefas.

Com essas informações bem analisadas e verificadas, é possível comprovar a dor e a necessidade desses psicólogos, como: onde posso verificar as sessões que terei naquele dia? Qual foi o meu recebimento financeiro dentro de um período específico? Dentro das sessões do mês, quais os pacientes já efetuaram pagamentos?

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamenteAs ilustrações abaixo irão ajudar no melhor no entendimento dessas dores: Na figura 3 podemos ver a matriz CSD. Onde podemos destacar suas certezas, suposições e dúvidas perante o sistema.

Figura - Matriz CSD

Com a matriz CSD exposta acima, pode-se verificar a real necessidade passada por esses profissionais.

Entrando em imersão profunda e se colocando no lugar deles, é montando o POEMS:

* Pessoas
* Objetos
* Ambiente
* Mensagem
* Serviços

Como demonstrado na figura 4, que mostra a imersão profunda relevante ao sistema.

Interface gráfica do usuário

Descrição gerada automaticamente

Figura - Observação POEMS

### 1.1.2 Personas

Os grandes interessados pela resolução desse sistema são os psicólogos, principalmente os autônomos que não fazem uso de secretárias precisam realizar suas atividades de gerenciamento de sessões e financeiros sozinhos.

Verônica é uma psicóloga com idade entre 41-59 anos. Sua responsabilidade profissional é de tratar pacientes com problemas emocionais, e tem em seu objetivo tratá-los através da resposta emocional do paciente. O seu grande desafio é não ter uma ferramenta que possa auxiliar no seu gerenciamento de sessões/recebíveis.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Figura - Persona

A figura 6 ilustra a persona principal dentro desse aplicativo.

Uma imagem contendo Site

Descrição gerada automaticamente

Figura - Mapa de Empatia

### 1.1.3 Benefícios e Justificativas

Tabela

Descrição gerada automaticamenteNesta seção pode-se observar através da tabela 1 que corresponde a ferramenta de Blueprint, o mapeamento e a listagem das ações do cliente, para resolver o problema em análise.

Tabela - Blueprint

Tela de celular com aplicativo aberto

Descrição gerada automaticamente com confiança médiaA ferramenta ilustrada na figura 7 mostra o Canvas Proposta de Valor, onde podemos entender o cliente e verificar como o produto irá se adequar a suas necessidades.

Figura - Canvas da Proposta de Valor

### 1.1.4 Hipóteses

Está relacionado abaixo algumas hipóteses com respeito ao problema da falta do gerenciamento por parte dos psicólogos.

* Psicólogos não conseguem ter o controle de suas sessões e os valores de recebimento por parte dos pacientes.
* Os psicólogos não sabem se a sessão ou o pagamento foi realizada ou não.

Com as hipóteses evidenciadas acima, segue algumas ideias para resolvê-las:

I1. Criação de um sistema de gerenciamento para psicólogos.

I2. Alterar a planilha para que possa ser informado todas as necessidades dos psicólogos.

Tela de celular

Descrição gerada automaticamente com confiança médiaCom as ideias acimas expostas foi montado uma Matriz de Priorização de Ideias, conforme é possível analisar na tabela 2:

Tabela - Matriz de priorização de ideias

A utilização para a construção dessa matriz foi a base da matriz BASICO:

* **B**enefícios
* **A**brangências
* **S**atisfação
* **I**nvestimentos
* **C**liente
* **O**peracionalidade

## 1.2 Solução

### 1.2.1 Objetivo SMART

Desenvolver um aplicativo em até 2 meses, com cerca de 2 horas por dia, um MVP para gerenciar as sessões dos psicólogos.

### 1.2.2 Premissas e Restrições

As premissas vistas para este projeto são:

* Utilização de 2 horas por dia para o desenvolvimento do mesmo.
* Não será desenvolvido em primeiro momento a funcionalidade de login.

As restrições encontradas para o mesmo são:

* O sistema será desenvolvido somente para o ambiente Web.
* O MVP deverá ser desenvolvido durante o Projeto Aplicado, que tem a duração de 2 meses.

Interface gráfica do usuário, Texto

Descrição gerada automaticamenteCom as informações das premissas e restrições do projeto, foi levantado alguns riscos conforme tabela 3 – Matriz de riscos.

Tabela - Matriz de riscos

### 1.2.3 Backlog de Produto

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamenteSegue abaixo a figura 8 onde pode ser visto a proposta de backlog para o desenvolvimento do MVP:

Figura - Backlog do produto

A informação do backlog pode ser visto e tratado também no link: https://trello.com/b/k8yt77JU/projeto-aplicado

# 2. Área de Experimentação

## 2.1 Sprint 1

As principais funcionalidades implementadas dentro da Sprint 1 foram as mais básicas para consistir no MVP.

As tecnologias utilizadas para a construção do sistema do projeto aplicado foram:

* Front End: React.js
* Back End: Node.js

O foco da Sprint foi desenvolver tanto o front end quanto o back end, ou seja, seguir o fluxo total das telas desde a visualização até a sua inserção nas tabelas correspondentes. Foram utilizadas várias tecnologias que serviu para auxiliar no desenvolvimento, como descrito abaixo:

* Front End
  + Tailwind-Css – Estilização das páginas;
  + React-Table – Criação de tabelas;
  + React-Router-Dom – Roteamento das páginas;
  + Fontawesome – Inserção de ícones nas páginas;
  + Material-ui – Utilização de componentes nas páginas;
* Back End
  + Sequelize – Auxilia na conexão com o banco de dados
  + Express – Roteamento das api’s

### 2.1.1 Solução

#### Evidência do planejamento:

Para a Sprint 1 foram desenvolvidas as seguintes telas, conforme visto na figura 9 que mostra a tela do trello:

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Figura - Visão da Sprint 1 no Trello

#### Evidência da execução de cada requisito:

**Tarefa Tela Home**

Texto

Descrição gerada automaticamenteUtilizando a biblioteca react-router-dom, foi criado a rota para a tela de Home, como podemos ver na figura 10.

Figura - Function App

Texto

Descrição gerada automaticamenteJá na figura 11 pode ser observado o componente de onde ficará todos os links para acessar cada tela específica.

Figura - GerenciamentoPsicoScreen

Texto

Descrição gerada automaticamenteNa figura 12 podemos ver o componente que levará para a tela inicial, a tela HomePage.

Figura - HomePage

**Tarefa Tela Manter Pacientes**

Utilizando o express no lado do back end pode ser observado na figura 13 as rotas utilizadas dentro da aplicação.

Texto

Descrição gerada automaticamenteNa figura 14 pode-se verificar como ficou interessante a utilização da biblioteca react-table que ajudou na visualização dos pacientes cadastrados.

Figura - Rotas da API do paciente

Texto

Descrição gerada automaticamenteT**arefa Manter Sessões**

Figura - Tela para mostrar a lista de pacientes

Para a tarefa de manter sessão pode-se ver na figura 15 ver um trecho de código onde é realizado a chamada ao banco de dados para realizar suas devidas tarefas.

Texto

Descrição gerada automaticamenteNa figura 16 pode ser visto o componente CustomTable que também é usado na tela para mostrar as sessões.

Figura - Repositório da sessão

Texto

Descrição gerada automaticamente

Figura - Componente CustomTable

#### Evidência dos resultados:

**Tarefa Tela Home**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamenteNa figura 17 é possível ver a renderização da tela principal do sistema.

Figura - Renderização da Tela Inicial

Para descrever as tabelas que foram utilizadas na estrutura de dados dentro da aplicação foi realizado o seguinte comando SQL, conforme demonstrado na figura 18.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente **Tarefa Manter pacientes**

Figura - Tabelas usadas no banco de dados

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamenteNa figura 19 é evidenciado a tela renderizada da tela de paciente

Figura - Tela de Paciente

**Tarefa Manter Sessão**

Na figura 20 pode ser também verificado a evidência da tela de sessão.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura - Tela de Sessão

### 2.1.2 Experiências vivenciadas

Durante a Sprint 1 foram constatadas algumas experiências que serviu de lição, como:

* As inúmeras bibliotecas disponibilizadas na internet para ajudar e auxiliar na construção de telas foi de grande importância;
* A construção do back end no início do projeto, foi interessante por tratar-se de um assunto que é mais fácil de assimilar e compreender;

## 2.2 Sprint 2

O foco da Sprint 2 foi nos relatórios em que o usuário irá poder tirar do sistema. Para a apresentação e visualização desses relatórios, os pacientes e sessões devem de estar previamente cadastrados, conforme visto e avaliado na Sprint 1.

Para esta Sprint foi utilizada apenas uma tecnologia para a geração dos relatórios em pdf:

* PDFMake

### 2.2.1 Solução

#### Evidência do planejamento:

Conforme evidenciado no backlog do produto a tarefa a ser realizada na Sprint 2 está sendo demonstrada na figura 21:

Tabela

Descrição gerada automaticamente com confiança baixa

Figura – Visão da Sprint 2 no Trello

#### Evidência da execução de cada requisito:

**Tarefa Tela Relatório**

Texto

Descrição gerada automaticamenteA chamada dos relatórios utilizando o PDFMake facilitou muito o trabalho para a sua geração. No exemplo da figura 22 pode-se observar que a aplicação chama apenas o componente do relatório:

Figura - Chamada do relatório de pacientes

A configuração do relatório é realizada através do componente criado para cada um. Na figura 23 é demonstrado a parte inicial da chamada do relatório, nela é possível ser visto a definição do título e dos dados:

Texto

Descrição gerada automaticamente

Figura - Parte Inicial do Componente

Texto

Descrição gerada automaticamenteNa figura 24 é possível observar os detalhes do relatório, onde consegue ser visualizado o nome de cada coluna, como também os dados que serão impressos, já vistos na figura passada:

Figura - Detalhes do Componente do Relatório

Texto

Descrição gerada automaticamenteFinalizando o componente é possível visualizar na figura 25 o rodapé, as definições e a criação do PDF.

Figura - Detalhes final do componente do Relatório

#### Evidência dos resultados:

**Tarefa Relatório**

Uma imagem contendo Texto

Descrição gerada automaticamenteNa figura 26 é possível observar o relatório dos pacientes:

Figura - Relatório de Pacientes

Tabela

Descrição gerada automaticamenteNa figura 27 já é possível visualizar o relatório de sessões:

Figura - Relatório de Sessões

### 2.2.2 Experiências vivenciadas

Durante a Sprint 2, por mais que tenha sido desenvolvido apenas uma única tarefa a mesma foi a mais árdua. A falta de experiência com a tecnologia para a geração de relatório fez com que tivesse que realizar diversas buscas na internet para encontrar a que fosse mais adaptável ao projeto.

No primeiro momento foi encontrada uma tecnologia que acabou sendo necessário baixar outras bibliotecas para que ela funcionasse, mesmo assim a aplicação apresentou diversos problemas.

Pesquisando mais sobre o assunto foi encontrado uma melhor biblioteca, onde a mesma, tinha exemplos em diversos sites dedicados a tecnologia, bem como em sites de compartilhamento de vídeos.

## 2.3 Sprint 3

O foco da Sprint 3 foi na tela em que o usuário consegue verificar através de um calendário, suas sessões em que foram cadastradas. Nela é possível verificar também se o paciente realizou o pagamento ou não.

Para esta Sprint a tecnologia utilizada para mostrar o calendário foi:

* rsuitejs

### 2.3.1 Solução

#### Evidência do planejamento:

Conforme visto no backlog do produto a tarefa a ser realizada na Sprint 3 está sendo demonstrada na figura 28:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

Figura – Visão da Sprint 3 no Trello

#### 

#### Evidência da execução de cada requisito:

**Tarefa Tela Calendário**

Continuando na utilização de bibliotecas para auxiliar na construção do MVP o rsuitejs foi de grande importância.

Apenas com um simples componente é possível a criação do calendário, como pode ser observado na figura 29:

Figura - Componente Calendário do rsuitejs

A propriedade principal deste componente encontra-se no renderCell, onde nela será inserida cada dia em que a sessão foi cadastrada.

Antes de começar a popular o calendário se faz necessário recuperar todas as sessões estando elas ordenadas pela data da sessão.

A ordenação foi realizada no backend da aplicação, conforme é visto na figura 30.

Texto

Descrição gerada automaticamente

Figura - Função de retorno das sessões ordenada por data

Com as sessões recebidas através da API, é utilizado o filtro, ou seja, a separação das sessões por data.

Na figura 31, é possível verificar a chamada da API e logo em seguida a função em que separa as datas.

Texto

Descrição gerada automaticamenteCom as informações das sessões populadas a função de renderCell começa a ser construída.

Figura - Recuperação das sessões e função para separar

Será possível verificar mais dois componentes utilizados, que também se encontram na biblioteca rsuitejs, são elas Whisper e Popover. As duas irão trabalhar em conjunto para quando no mesmo dia houver mais de uma sessão cadastrada. A figura 32 pode ser demonstrada como fica o código implementado.

Texto

Descrição gerada automaticamente

Figura - Função renderCell

Tela de computador com letras e números em fundo preto

Descrição gerada automaticamenteFinalizando a função renderCell é possível ver mais um componente da rsuitejs que é o Badge, que apenas será mostrado como forma de organização da sessão que será realizada na data. Na figura 33, pode ser visto o retorno da função.

Figura - Retorno da função renderCell

#### Evidência dos resultados:

A figura 34 é evidenciada a tela em que mostra o resultado de um mês específico com as sessões cadastradas:

Calendário

Descrição gerada automaticamente

Figura - Tela de Calendário

### 2.3.2 Experiências vivenciadas

No momento do desenvolvimento, dificuldades com o tempo por conta do trabalho fez com que a sprint ficasse um pouco mais estreita para seu término, mesmo assim o desenvolvimento acabou saindo um pouco da zona de conforto, o aprendizado de mais um componente que ainda não tinha sido utilizado dificultou um pouco mais a finalização da sprint.

# 3. Considerações Finais

## 3.1 Resultados

## 3.2 Contribuições

## 3.3 Próximos passos