

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUCAS BAUER DE SOUZA

Atualização do sistema SPT-Sars-Cov-2

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Marcelo Soares Pimenta

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^ª. Patricia Pranke

Pró-Reitora de Graduação: Prof^ª. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^ª. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Me faltam palavras para descrever o quão gratificante é para mim o momento do fim de minha graduação. A minha graduação foi um grande desafio em minha vida, onde vivi diversos momentos marcantes, que sem dúvida me fizeram crescer não apenas no meio profissional, mas também me proporcionaram um grande desenvolvimento pessoal.

A escolha do tema para meu trabalho de conclusão do curso de Engenharia de Computação teve um grande peso pessoal. Eu e minha família vivemos de perto o que uma pandemia pode causar, mas eu sempre tive uma certeza muito forte dentro de mim de que tanto meu pai quanto minha mãe estariam comigo nesse momento. Aos meus amados pais, Alessandro Quadros de Lima e Denise Pizzá Bauer de Lima, o meu agradecimento mais sincero do fundo do meu coração, pois foram a base e o incentivo necessário para essa conquista. Ao meu irmão João Pedro Bauer de Lima, por sempre estar ao meu lado e que me acompanhou de perto não só nessa caminhada, mas como em todas em minha vida. Agradeço também a minha amada avó, Eva Eunice Pizzá, por ser sempre o nosso suporte quando precisamos, e que nessa jornada não foi diferente.

Ao professor Marcelo Soares Pimenta, por aceitar em ser o meu orientador deste trabalho. Meu muito obrigado pela mentoria durante o desenvolvimento e pela confiança ao fornecer o contato pessoal, onde foi o nosso principal meio de comunicação. Agradeço também ao professor João Cesar Netto, por me fornecer os acessos e contatos necessários para esse desenvolvimento, além das sugestões de melhora de meu trabalho por meio de sua avaliação da banca, à professora Renata, por também fazer parte da minha banca e fornecer observações que permitissem o aperfeiçoamento de meu trabalho e ao professor Marcelo Lammers, pela oportunidade em elaborar novas funcionalidades ao SPT-Sars-Cov-2.

Por fim, um agradecimento também a Carine Bertagnolli Bathaglini, pela sua tanta prestatividade e paciência comigo e que foi essencial para me familiarizar com a visão prática do sistema, através de reuniões e inúmeras mensagens trocadas.

RESUMO

A pandemia que o mundo enfrentou devido a propagação abrupta da contaminação em grande escala das populações de diversos países devido ao vírus SARS-CoV-2, o novo coronavírus, causou a necessidade de mudanças bruscas nas mais diversas áreas da sociedade e nela em si. Para auxiliar a lidar com esta pandemia, em um projeto da UFRGS com o Governo do Estado do RS, foi criado um sistema - denominado "SPTSars-Cov-2" - cujo objetivo é coletar e disponibilizar dados relacionados ao gerenciamento dos exames de detecção da Covid-19 no estado do RS. Criado através de desenvolvimento incremental, ou seja, o software vai sendo entregue em "pedaços" ou "incrementos", o SPTSars-Cov-2 continua operando e sendo usado ainda no presente ano, apresentando as mais diversas funcionalidades a fim de atender às necessidades apresentadas pela pandemia até então. O sistema ainda permite a adição de novas funcionalidades, não necessariamente relacionadas aos testes RT-PCR (*Reverse Transcription - Polymerase Chain Reaction*) de detecção do vírus, mas sim de outros tipos de testes ou ainda de aplicabilidades que envolvam consulta e filtragem para uma melhor visão de sua base de dados, sendo este o foco do presente trabalho: especificar as novas funcionalidades desenvolvidas desejadas e requisitadas pelos seus usuários a fim de proporcionar uma melhor experiência de acordo com as suas necessidades.

Palavras-chave: Pandemia, Covid-19, Desenvolvimento Incremental, Experiência de Usuário, RT-PCR.

System SPT-Sars-Cov-2 update

ABSTRACT

The pandemic that the world faced due to the abrupt spread of the large-scale contamination of populations in different countries due to the SARS-CoV-2 virus, the new coronavirus, has caused the need for sudden changes in the most diverse areas of society and in it itself. To help deal with this pandemic, in a project of UFRGS with the Government of the State of RS, a system was created - called "SPTSars-Cov-2" - whose objective is to collect and make available data related to the management of the detection exams of the virus Covid-19 in the state of RS. Created through incremental development, that is, the software is being delivered in "pieces" or "increments", the SPTSars-Cov-2 continues to operate and is still used in the present year, presenting the most diverse functionalities in order to meet the needs presented by the pandemic so far. The system also allows the addition of new functionalities, not necessarily related to the RT-PCR (*Reverse Transcription - Polymerase Chain Reaction*) virus detection tests, but also other types of tests or even applicability involving consultation and filtering for a better view of your database, which is the focus of the present work: specify the new developed functionalities desired and requested by its users in order to provide a better experience according to their needs.

Keywords: Pandemic, Covid-19, Incremental Development, User Exeperience, RT-PCR.

LISTA DE FIGURAS

Figura 1.1	Curva de óbitos do estado do Rio Grande do Sul.....	11
Figura 2.1	Página inicial do SPT-Sars-Cov-2	14
Figura 2.2	Consulta de laudos cadastrados no sistema	15
Figura 2.3	Módulo de gráficos do sistema	16
Figura 2.4	Módulo de estatísticas	16
Figura 2.5	Controle de recursos pelo usuário administrador	17
Figura 3.1	Desenvolvimento Incremental	19
Figura 3.2	Ciclos de incrementos.....	20
Figura 3.3	Camadas para aplicações Web.....	24
Figura 3.4	Camadas tratadas como funcionalidade e informação.....	26
Figura 3.5	Modelo MVC.....	29
Figura 3.6	Objeto BO.....	30
Figura 3.7	Classe ConsultaPersonalizadaINT.....	31
Figura 4.1	Distribuição das tecnologias utilizadas por camada	35
Figura 5.1	Opção "Pacientes Repetidos"do novo módulo	37
Figura 5.2	Início do fluxo para a funcionalidade de pacientes repetidos	39
Figura 5.3	Calendário clicável em cada campo de input	40
Figura 5.4	Dados dos pacientes repetidos	41
Figura 6.1	Opção "Consulta de tabelas"do novo módulo	43
Figura 6.2	Apresentação da funcionalidade de consulta personalizada ao usuário	44
Figura 6.3	Início do fluxo.....	45
Figura 6.4	Apresentação dos novos componentes	46
Figura 6.5	Apresentação dos dados contidos na tabela selecionada	46
Figura 6.6	Apresentação dos novos botões para caminhos alternativos no fluxo	47
Figura 6.7	Tabela personalizável com dados resultantes da consulta	48
Figura 6.8	Planilha exportada com dados resultantes da consulta	49

LISTA DE TABELAS

Tabela 2.1 Componentes da arquitetura do sistema	18
Tabela 5.1 Arquivos adicionados ou modificados para o primeiro incremento	38
Tabela 6.1 Arquivos adicionados ou modificados para o segundo incremento.....	50

LISTA DE ABREVIATURAS E SIGLAS

BD	<i>Banco de Dados</i>
BO	<i>Business Object</i>
CSS	<i>Cascading Style Sheets</i>
DAO	<i>Data Access Object</i>
DTO	<i>Data Transfer Object</i>
FAURGS	Fundação de Apoio da Universidade Federal do Rio Grande do Sul
HTML	<i>HyperText Markup Language</i>
ICBS	Instituto de Ciências Básicas da Saúde
INF	Instituto de Informatica
INT	Interface
MVC	<i>Model-View-Controller</i>
PHP	<i>Hypertext Preprocessor</i>
RS	Rio Grande do Sul
RT-PCR	<i>Reverse Transcription - Polymerase Chain Reaction</i>
SPTSars-Cov-2	Sistema de Procedimentos de Testes da Covid-19
TCC	Trabalho de Conclusão de Curso
UFRGS	Universidade Federal do Rio Grande do Sul

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Objetivos	11
1.2 Metodologia	12
1.3 Estrutura do trabalho	13
2 O SISTEMA DE PROCEDIMENTOS DE TESTES DA COVID-19	14
2.1 Visão Geral	14
2.2 Principais funcionalidades	14
2.2.1 Consulta de laudos	15
2.2.2 Gráficos	15
2.2.3 Planilhas	16
2.2.4 Recursos	17
2.3 Arquitetura do sistema	17
2.4 Pendências	17
3 FUNDAMENTAÇÃO TEÓRICA	19
3.1 Desenvolvimento Incremental	19
3.2 Experiência de Usuário	23
3.3 Padrões de Projeto	28
3.3.1 Padrão MVC	28
3.3.2 DTO e DAO	29
3.3.3 Singleton	30
3.4 Programação orientada a objetos	31
4 TECNOLOGIAS UTILIZADAS	33
5 PRIMEIRO INCREMENTO	36
5.1 O processo de geração de laudo	36
5.2 Acessando a funcionalidade	37
5.3 Arquivos adicionados ou modificados	38
5.4 Fluxo de execução	39
6 SEGUNDO INCREMENTO	42
6.1 Acessando a funcionalidade	42
6.2 Arquivos adicionados ou modificados	43
6.3 Fluxo de execução	43
7 CONCLUSÃO	51
REFERÊNCIAS	52

1 INTRODUÇÃO

É inegável que a pandemia do Covid-19 certamente causou grande impacto na vida das pessoas do mundo todo, sejam elas de diferentes áreas profissionais ou de diferentes classes sociais. Como consequência desta mudança brusca, diversas modificações na forma que interagimos foram inevitáveis para que pudéssemos nos adaptar a essa nova realidade. Praticamente todas as atividades que poderiam ser adaptadas para um modo remoto tiveram de ser adaptadas para todos, seja em sua vida profissional ou pessoal, a fim de sua necessária segurança física.

Há mais de uma teoria para o início de contaminação do vírus, mas sabe-se que o início desenfreado de alastramento da contaminação tem origem na cidade de Wuhan, na China, em dezembro de 2019, que celeremente se espalhou no continente (Instituto Butantan, 2022). A partir daí, demorou apenas alguns meses para a doença fazer-se presente no Brasil, tendo o seu primeiro caso oficializado em 26 de fevereiro do ano de 2020 (Governo Federal, 2020).

Com o primeiro caso reconhecidamente presente no país, foi um momento de grande desafio para a população: adaptar-se a nova realidade frente a uma pandemia. Das mais diversas áreas profissionais que estavam presentes na sociedade, pode-se dizer que duas delas foram extremamente exigidas: a da tecnologia da informação e a da saúde.

No caso da saúde, os casos precisavam obrigatoriamente ser monitorados e com urgência, no objetivo de amenizar a contaminação do vírus, tornando necessária a alta demanda de exames em massa. Em virtude disso, o Instituto de Ciências Básicas da Saúde (ICBS), da Universidade Federal do Rio Grande do Sul (UFRGS), juntamente com o Governo do Estado, assinou um termo de cooperação, através da UFRGS e da Fundação de Apoio da Universidade Federal do Rio Grande do Sul (FAURGS), para a realização da análise de exames de detecção da Covid-19 (Instituto de Ciências Básicas da Saúde, 2020). No caso da tecnologia da informação, uma área cada vez mais presente na vida das pessoas, perante este desafio, tornou-se da mesma forma cada vez mais necessária vistas as consequências que esse novo cenário em nosso país trazia, pois, como mencionado anteriormente, praticamente todas as nossas atividades rotineiras ficaram atreladas ao modo remoto.

É seguindo esse contexto que surge a necessidade de um sistema para gerenciar o armazenamento, registro e gerenciamento desses exames, estes representando os resultados dos testes RT-PCR. Vista a grande quantidade de dados para esse processo, surge

o Sistema de Procedimentos de Testes da Covid-19 (SPTSars-Cov-2), resultado de uma parceria firmada entre o ICBS e o Instituto de Informática da UFRGS, cuja primeira versão foi finalizada em 2021, um ano em que um grande número de casos de óbitos foi registrado devido a superlotação das unidades de saúde, como mostra a Figura 1.1.

Além do registro, armazenamento e gerenciamento dos resultados de amostras de testes RT-PCR, o Sistema de Procedimentos de Testes da Covid-19 também apresenta uma série de funcionalidades que vão de encontro ao acompanhamento do andamento da pandemia, isto é, uma visão em escalas e gráficos para reconhecimento em quantidade e distribuição por área em mapas da situação desses testes. É possível a construção de gráficos personalizados, visualizar mapas de acordo com os dados armazenados, entre outras funcionalidades com capacidade de personalização assim como as novas funcionalidades adicionadas neste trabalho. Em virtude disso, o uso do sistema SPTSars-Cov-2 tem sido fundamental para o monitoramento da contaminação da população.

Figura 1.1 – Curva de óbitos do estado do Rio Grande do Sul

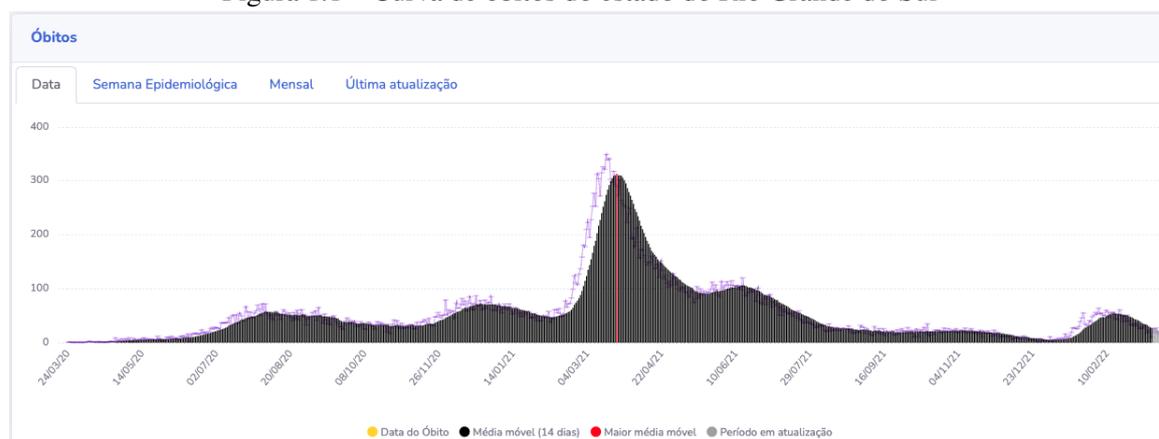


Gráfico apresentando a curva de óbitos do estado do Rio Grande do Sul desde, aproximadamente, o início da pandemia. Tendo o seu ápice entre o início do mês de março e final de abril do ano de 2021

Fonte: Secretaria da Saúde do Estado do Rio Grande do Sul (2022).

1.1 Objetivos

Este trabalho de conclusão de curso tem como principal finalidade o desenvolvimento de novas funcionalidades requisitadas pelos usuários do Sistema de Procedimentos de Testes da Covid-19, o SPT-Sars-Cov-2, de acordo com a conjuntura corrente do sistema e pendências geradas a partir da atual situação da pandemia buscando uma adaptação do sistema à mesma.

Sendo assim, foram estabelecidas as seguintes metas:

- desenvolver uma funcionalidade capaz de detectar pacientes que tiveram resultado positivo em seus laudos mais de uma vez em um determinado intervalo de tempo, sendo este intervalo personalizado pelo usuário;
- desenvolver uma funcionalidade capaz de tornar visível ao usuário o que está armazenado na base de dados do sistema, sem que os mesmos sejam alterados;
- em ambas as funcionalidades desenvolvidas, as mesmas procurarão conceder uma filtragem sob os dados coletados dinamicamente de modo a permitir uma personalização desses dados de acordo com o desejo do usuário;
- as funcionalidades serão inspiradas em comportamentos semelhantes a algumas já presentes no sistema, visando padronizar a experiência de usuário já consolidada do mesmo.

1.2 Metodologia

Para a elaboração deste trabalho, foi considerado o modelo de processo de desenvolvimento incremental. Além de ser um modelo que pode permitir uma maior flexibilidade com novas funcionalidades adicionadas a um sistema já consolidado sob um certo ponto de vista, o mesmo já foi adotado desde os incrementos iniciais desenvolvidos do sistema, ou seja, dar-se-á o andamento de um processo já robusto sob a perspectiva da adição de novas funcionalidades ao sistema em questão.

Considerando o termo incremento equivalente a cada nova funcionalidade agregada ao sistema ao longo do tempo, o processo consiste em especificação das funcionalidades desejadas, implementação do software e validação do mesmo, ou ainda, aperfeiçoamento, modificações, atualização, entre outras possibilidades após o processo de validação, caso seja necessário.

Além do processo de desenvolvimento incremental, alguns conceitos de experiência de usuário também foram implementados. Visto que as novas funcionalidades são interativas e personalizáveis aos usuários que utilizem-nas. Ainda assim, esses conceitos também buscam padronizar e não impactar a experiência já proporcionada aos seus usuários em todos os aspectos.

Todos os conceitos abordados para a aplicação da metodologia em questão serão

melhormente detalhados nos próximos capítulos, afim de uma boa compreensão do pro-
veito que procurou-se tirar desses conceitos em uma convergência com as tecnologias
utilizadas e recursos presentes no sistema no que diz respeito ao desenvolvimento das
novas funcionalidades para o mesmo.

1.3 Estrutura do trabalho

O texto estrutura-se em 7 capítulos, o capítulo 2 consiste em uma visão geral do sistema e uma breve descrição das suas principais funcionalidades atualmente, além de apresentar como a sua arquitetura funciona, com figuras para ilustrar cada tópico. Ainda assim, descrever as pendências apresentadas.

O capítulo 3 apresenta a fundamentação teórica que serviu como base de inspira-
ção para o desenvolvimento da atualização do sistema, ou seja, as novas funcionalidades
incluídas. Buscando uma descrição que possa introduzir esses conhecimentos na sua re-
lação com essas funcionalidades. Abordando conceitos de desenvolvimento incremental,
experiência de usuário, padrões de projeto e programação orientada a objetos.

No capítulo 4, busca-se esclarecer as tecnologias utilizadas no sistema. Desta-
cando como essas tecnologias são usadas nas diferentes camadas do mesmo, desde sua a
interface, consistindo na camada de *front-end*, passando pela sua manipulação de dados,
camada de *back-end*, até atingir a consulta de onde os mesmos estão armazenados, ou
seja, o seu banco de dados.

Sucessivamente, os capítulos 5 e 6 visam apresentar as funcionalidades que foram
adicionadas ao sistema na considerada atualização mencionada neste trabalho, de modo a
buscar a compreensão das mesmas, e de que assim seja possível relacionar alguns pontos
dos seus desenvolvimentos com os conhecimentos apresentados anteriormente.

Por fim, o capítulo 7 destaca quais conclusões puderam-se deduzir a partir do
desenvolvimento das novas funcionalidades criadas. Relacionando as mesmas com os
objetivos mencionados, destacando pontos positivos e dificuldades encontradas durante a
construção deste trabalho. Além disso, também destaca pontos futuros para o Sistema de
Procedimentos de Testes da Covid-19.

2 O SISTEMA DE PROCEDIMENTOS DE TESTES DA COVID-19

2.1 Visão Geral

O SPT-Sars-Cov-2 consiste em um sistema capaz de armazenar, monitorar e gerar dados estatísticos relacionados a laudos resultantes de amostras de testes RT-PCR implementados em pacientes. Apresenta também outras funções como, por exemplo, diferenciar usuários por tipo. Tendo suas funcionalidades separadas por grupos de acordo com o menu lateral esquerdo, conforme mostra a Figura 2.1.

Figura 2.1 – Página inicial do SPT-Sars-Cov-2

The screenshot shows the dashboard of the SPT-Sars-Cov-2 system. The top left corner displays 'COVID - 19' and a hamburger menu icon. The top right corner shows the user name 'Lucas Bauer' with a dropdown arrow. The main content area is titled 'Dashboard' and includes two buttons: 'Sugestões' and 'Relatar problema'. On the left, there is a dark teal navigation menu with the following sections:

- NAVEGAÇÃO**
 - Página Inicial
 - Editar meu perfil
 - Cadastrar meus horários
 - Novos Procedimentos
- RECURSOS EXTRAS**
 - últimos 3 meses
 - envio de email do feriado (ativar)
 - envio de email dos novos procedimentos
- RECEPÇÃO**
 - Cadastrar amostra (apenas UFRGS)
 - Cadastrar amostra
 - Ver amostras
 - Ver amostras dos coletadores
 - Minhas amostras
 - Cadastrar paciente

The main content area features a 'Avisos' section with a 'Novo aviso' button. Below this is a table with five rows of notifications:

<input type="checkbox"/>	Confiram os novos procedimentos aqui	19/11/2021 - 23:46:43	Não resolvido	
<input type="checkbox"/>	Confiram os novos procedimentos aqui	19/11/2021 - 23:33:30	Não resolvido	
<input type="checkbox"/>	Confiram os novos procedimentos aqui Novos procedimentos	19/11/2021 - 23:31:58	Não resolvido	
<input type="checkbox"/>	[Trabalho] 20/04/2021 no turno da Manhã	07/05/2021 - 23:36:39	Não resolvido	
<input type="checkbox"/>	[Sobreaviso] 24/04/2021 no turno da Tarde	07/05/2021 - 23:36:39	Não resolvido	

Fonte: Elaborado pelo autor (2022)

2.2 Principais funcionalidades

Esta seção visa evidenciar as principais funcionalidades do sistema, destacando-se as descritas nas próximas subseções, sendo que a sua maioria apresenta alguma forma de personalização possível ao usuário.

2.2.1 Consulta de laudos

Toda criação de laudo inicia-se por meio da seleção do usuário para amostras que ainda não possuam nenhum laudo associado. A partir daí, são informados os dados da amostra para que o mesmo seja registrado no sistema. Uma vez registrado, sendo o paciente do laudo em questão do perfil UFRGS ou Equipe de Voluntários, automaticamente será enviado um e-mail a esse paciente para informar qual o resultado desse laudo, fornecendo também um link para download do documento enviado. Além disso, o módulo de laudos do sistema também permite a consulta dos mesmos. Fornecendo a possibilidade de vários filtros para consulta, conforme mostra a Figura 2.2

Figura 2.2 – Consulta de laudos cadastrados no sistema

The screenshot shows the 'Listar laudos' (List Reports) interface. It features a sidebar with navigation options like 'LAUDOS', 'DIAGNÓSTICO', and 'ESTATÍSTICAS'. The main area contains search filters for patient name, sample code, date, patient profiles, results, and status. Below the filters is a table with two rows of data.

Cod laudo	Cod amostra	Resultado	Geração	Liberação	Responsável	Reenviar email laudo
68284	P204202	DETECTADO	18/02/2022 - 23:24:52	18/02/2022 - 23:24:46	86251791004	
68283	P230203	NAO_DETECTADO	18/02/2022 - 23:24:20	18/02/2022 - 23:24:24	86251791004	

Fonte: Elaborado pelo autor (2022)

2.2.2 Gráficos

O módulo de gráficos consiste principalmente em fornecer visualmente ferramentas que possibilitem que se tenha o conhecimento em quantidade dos dados cadastrados no sistema ao longo do tempo. Uma série de gráficos é apresentada e, além disso, ainda dispõe da funcionalidade que permite ao usuário construir o seu próprio gráfico e é capaz de relacionar mapas com os dados do sistema. módulos este apresentado na Figura 2.3

Figura 2.3 – Módulo de gráficos do sistema

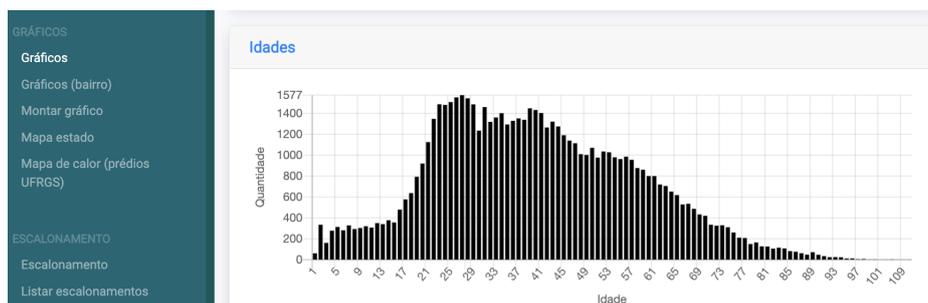


Gráfico relacionando quantidade de amostras com idade dos pacientes.

Fonte: Elaborado pelo autor (2022)

2.2.3 Planilhas

Conforme mencionado anteriormente, uma das principais funções do Sistema de Procedimentos de Testes da Covid-19 é o acompanhamento de dados estatísticos relacionados aos laudos e amostras armazenados no sistema. Sendo assim, o sistema apresenta um módulo apenas para a produção dessas estatísticas com a geração de planilhas de acordo com filtros personalizados pelo usuário, conforme mostra a Figura 2.4

Figura 2.4 – Módulo de estatísticas

Fonte: Elaborado pelo autor (2022)

2.2.4 Recursos

O sistema diferencia os usuários de acordo com o seu tipo de perfil. Isso é feito porque o sistema faz um controle de acesso aos seus usuários por meio de recursos. Nem todo usuário pode ter todos os acessos possíveis ao sistema. Sendo assim, o sistema apresenta o módulo "Relacionamento perfil de usuário e seu recurso". Onde o usuário administrador é capaz de personalizar quais recursos ficarão disponíveis para quais usuários, sendo que cada recurso representa uma funcionalidade do sistema. Funcionalidade esta apresentada conforme a Figura 2.5.

Figura 2.5 – Controle de recursos pelo usuário administrador

PERFIL	RECURSOS	
Gestão	1. principal 2. cadastrar_amostra 3. listar_amostra 4. editar_amostra 5. editar_paciente 6. listar_cadastroAmostra 7. buscar_amostra 8. minhas_amostras 9. cadastrar_meus_horarios 10. editar_meu_perfil	
Recepção	1. principal 2. cadastrar_amostra 3. listar_amostra 4. editar_amostra 5. editar_paciente	

Fonte: Elaborado pelo autor (2022)

2.3 Arquitetura do sistema

A arquitetura do Sistema de Procedimentos de Testes da Covid-19 está dividida em 3 componentes principais, descritos de acordo com a Tabela 2.1.

2.4 Pendências

Representantes do ICBS destacaram duas pendências do sistema para novas funcionalidades. A primeira delas seria com relação a pacientes repetidos. Com o avanço

Tabela 2.1 – Componentes da arquitetura do sistema

Componente	Descrição
Controlador	Arquivo responsável pelo controle do que será fornecido ao usuário visualmente de acordo com o recurso que o mesmo possui. Além disso, detecta quais ações o usuário faz em tela.
Ações	Diretório responsável por cada arquivo do tipo <i>action</i> associado a funcionalidade do sistema. Cada arquivo <i>action</i> faz a comunicação entre as classes necessárias para manipular a interface e os dados no fluxo de execução para a comunicação com o controlador.
Classes	Diretório de diretórios, onde em cada diretório interno estão contidas as classes definidas e necessárias para a execução de cada funcionalidade. Cada funcionalidade tem, pelo menos, a sua classe BD, que se comunica com o banco de dados realizando as consultas, a sua classe INT, que monta os elementos da interface, e a classe RN, que faz a comunicação entre a classe BD e o arquivo <i>action</i> correspondente.

Fonte: Elaborado pelo autor (2022).

da pandemia, dado um certo tempo, alguns pacientes podem ter testado positivo mais de uma vez nas amostras do sistema, na qual o mesmo ainda não apresentava um filtro para essa análise.

A segunda delas está relacionada ao banco de dados. O usuário administrador desejava que alguns usuários pudessem visualizar certos dados, mas os mesmos não podiam alterá-los.

São essas as pendências combatidas pelas novas funcionalidades incluídas por meio desta atualização, que serão mais detalhadamente especificadas na sequência.

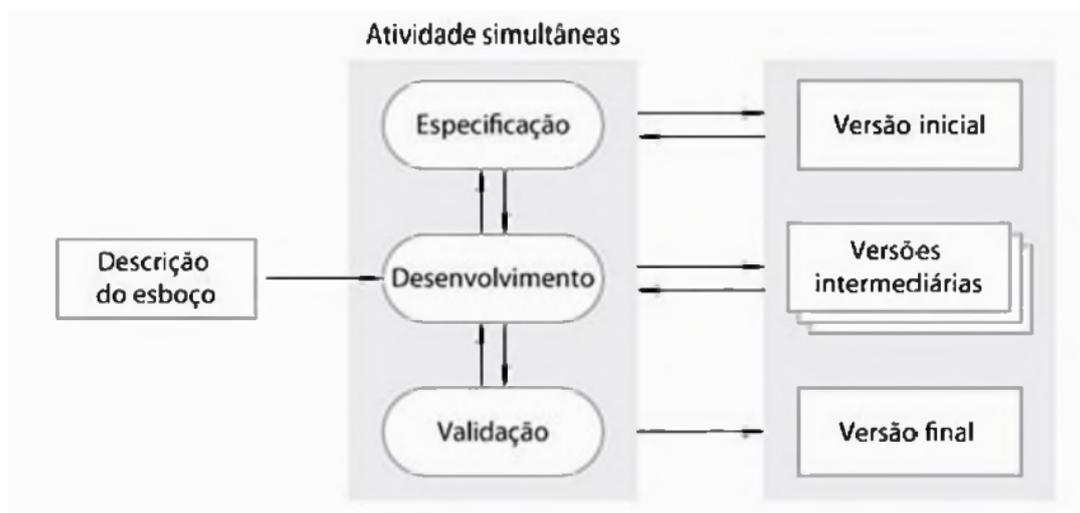
3 FUNDAMENTAÇÃO TEÓRICA

Capítulo que detalha conceitos teóricos abordados para o desenvolvimento das funcionalidades elaboradas na atualização do Sistema de Procedimentos de Testes da Covid-19.

3.1 Desenvolvimento Incremental

O modelo ou desenvolvimento incremental consiste em um processo de produção de um software onde a principal ideia constitui-se basicamente em efetuar uma versão inicial do projeto, mostrá-la aos seus usuários, e, dependendo do seu *feedback*, chegar a conclusão se a funcionalidade ou sistema desenvolvido estão adequados ou não a sua versão final esperada (SOMMERVILLE, 2011). Caso não estejam, o processo se repete e assim sucessivamente até que se apresente uma versão satisfatória do sistema ou funcionalidade em questão, conforme ilustrado na Figura 3.1

Figura 3.1 – Desenvolvimento Incremental



Fonte: SOMMERVILLE (2011, p. 22).

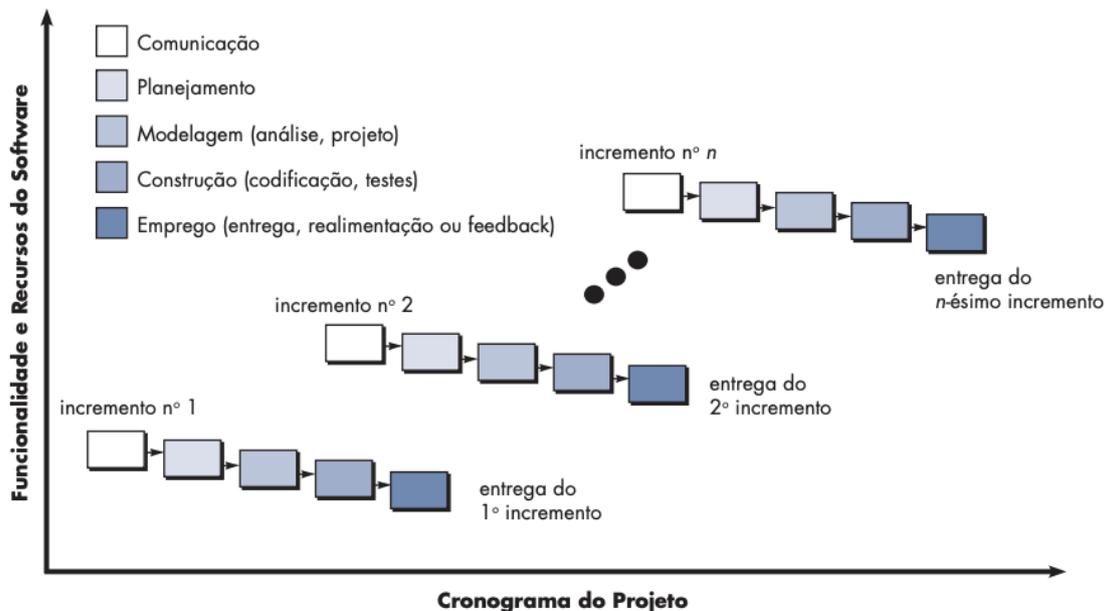
Neste modelo, denomina-se "incremento" uma nova funcionalidade no sistema ou uma série de funcionalidades que em conjunto formam o mesmo a fim de atender às necessidades do cliente ou seus usuários.

Frequentemente comparado e ao mesmo tempo semelhante ao modelo cascata em alguns aspectos, onde o desenvolvimento do produto (seja funcionalidade ou sistema como um todo) ocorre e evolui de forma sequencial em suas fases até a sua versão entre-

gável, o modelo incremental por sua vez prega a ideia de que vários ciclos simultâneos podem ocorrer para a adição de novas funcionalidades para um sistema. O modelo cascata consiste em um único ciclo, onde caso mudanças sejam necessárias na fase de desenvolvimento de alguma funcionalidade, torna-se dificilmente alcançável o processo de implementá-las e voltar a estágios anteriores desse desenvolvimento (SOMMERVILLE, 2011).

Entretanto, quando o modelo incremental é abordado, podem existir vários ciclos para uma entrega, onde, no caso de mudanças tornarem-se necessárias durante a fase de desenvolvimento, um novo ciclo é feito para adaptá-las ao produto entregável e assim continuamente até uma versão onde as necessidades do cliente são atendidas de uma forma pertinente, ou seja, apropriado à finalidade a que se destina. Isto ainda quando pensa-se em uma única funcionalidade, pois o mesmo processo pode ocorrer em paralelo. Caso o projeto já exista, alguns incrementos podem ocorrer simultaneamente se tratando de funcionalidades novas e independentes entre si, e assim sucessivamente para cada nova funcionalidade requerida do projeto pelos seus usuários, processo ilustrado conforme mostra a Figura 3.2.

Figura 3.2 – Ciclos de incrementos



Fonte: PRESSMAN (2011, p. 61).

Segundo Sommerville (2011, p.22), o desenvolvimento incremental apresenta algumas vantagens quando comparado ao cascata, são elas:

1. **Custo:** Quando comparado ao modelo cascata, a quantidade de análise e documentação a ser refeita é muito menor do que o necessário, pois mudanças já são previamente abordadas e esperadas e incrementos são voltados para a ideia de pedaços de um projeto. Com isso, o custo de ajustar as mudanças nos requisitos do cliente é reduzido.
2. **Feedback:** Torna-se mais acessível o *feedback* dos clientes sobre o que já foi desenvolvido. Uma vez que o desenvolvimento está dividido em incrementos, "os clientes podem fazer comentários sobre as demonstrações do software e ver o quanto foi implementado. Os clientes têm dificuldade em avaliar a evolução por meio de documentos de projeto de software" Sommerville (2011, p.22).
3. **Rapidez em entrega:** A implementação e entrega rápida de um software útil ao cliente torna-se possível mesmo se toda a funcionalidade não for incluída, já que ele é entregue em partes. Os clientes podem interagir com essas partes antes de um modelo em processo cascata finalizado, por exemplo.

Um dos pontos fortes do desenvolvimento incremental é considerar modificações do produto essenciais já na fase de planejamento, a fim de serem adaptadas de uma melhor forma às necessidades do cliente e à entrega de recursos e funcionalidades adicionais, ou seja, mudanças já são previamente esperadas e concebidas, onde o processo se repete após a liberação de cada incremento na busca da produção do produto completo para impactar o desenvolvimento da menor forma possível (PRESSMAN; MAXIM, 2016).

Considerando o final de cada incremento uma "entrega incremental" e baseando-se nos seus aspectos teóricos, Sommerville ainda destaca vantagens da aplicação dessas entregas incrementais (Sommerville, p.32), sendo elas:

- Através dos incrementos iniciais de um produto, os clientes ganham experiência de usuário sobre o mesmo. Com isso, podem já prever os requisitos de próximos incrementos.
- Uma entrega incremental permite a utilização imediata do sistema, ou seja, não é necessário o produto estar inteiramente desenvolvido ou criado para o cliente utilizá-lo. A ideia se relaciona com o fato de atender requisitos mais críticos.
- Conforme mencionado anteriormente, as entregas incrementais já estão sujeitas a mudanças de acordo com o seu planejamento inicial.

- A prioridade dos serviços entregues vai estar relacionada com a quantidade de testes recebida. Sommerville afirma: "Quanto maior a prioridade dos serviços entregues e, em seguida, incrementos integrados, os serviços mais importantes recebem a maioria dos testes. Isso significa que a probabilidade de os clientes encontrarem falhas de software nas partes mais importantes do sistema é menor".

Sendo assim, para uma boa compreensão a respeito do processo de desenvolvimento de software incremental, muitas vezes faz-se necessária a sua comparação ao modelo cascata, por ser o paradigma mais antigo da engenharia de software (PRESSMAN; MAXIM, 2016). Conclui-se que os dois modelos são semelhantes, mas diferentes. Em outras palavras, o modelo incremental relaciona os elementos do modelo cascata, que são aplicados de forma repetida e incrementalmente. Sendo o principal objetivo identificar os requisitos do sistema a partir do usuário e trabalhar com incrementos para obter o produto final (SOMMERVILLE, 2011).

O SPTSars-Cov-2 foi desenvolvido incrementalmente, visto que novas funcionalidades foram sendo adicionadas ao longo do tempo, de acordo com as necessidades e sugestões de seus usuários (BATHAGLINI, 2021). Sendo assim, a atualização do sistema, principal objetivo deste trabalho, não será diferente. A implementação de novas funcionalidades irá de acordo com as sugestões de melhora baseadas na versão atual desse sistema, preservando seus padrões para a criação de novas aplicabilidades a fim de desenvolver novos incrementos em função da familiarização do cliente.

Este modelo foi escolhido para a criação do sistema SPTSars-Cov-2 devido ao tamanho da equipe responsável pelo seu desenvolvimento (BATHAGLINI, 2021) e também pelo nível crítico em que seus incrementos iniciais foram requisitados, devido aos mesmos serem desenvolvidos em uma época onde a pandemia se encontrava em um cenário catastrófico no Rio Grande do Sul. Como a equipe é composta por um número pequeno de pessoas, não é viável desenvolver um sistema completo e entregá-lo ao cliente em um prazo factível (PRESSMAN, 2006). Usando incrementos, cada funcionalidade pode ser desenvolvida pouco a pouco e analisada com os usuários a fim de confirmar se o sistema atende às suas necessidades a partir de cada uma delas.

3.2 Experiência de Usuário

Quando pensa-se no tema de experiência de usuário, significa a relação de uma pessoa (no caso, usuário) com um produto. Sendo este produto um serviço, um software, um objeto físico, entre inúmeras possibilidades. Toda vez que usa-se um produto, o mesmo proporciona uma experiência. Normalmente dá-se mais atenção a "o que" um produto faz, com relação a funcionalidade capaz de realizar, do que "como" ele faz. Fator que pode diferenciar um produto que teve sucesso ou falha e que traz a noção da importância deste tema (GARRETT, 2011).

Existem diversos casos para exemplificação onde a experiência de usuário age para concretizar um maior entendimento do tema, pode-se pensar no exemplo em que James Garret considera como o dia de "má sorte". Neste exemplo, uma pessoa acorda com a luz do sol vindo da janela e percebe que o seu despertador não tocou, pois estava com horário errado. Já atrasada, a pessoa liga sua cafeteira para se vestir paralelamente e, quando vai pegar o seu café, percebe que a máquina não funcionou e não tem tempo para verificar a causa em virtude do atraso. Ao sair de casa, percebe que é necessário abastecer o seu carro e ao chegar em um posto de combustível, percebe que a máquina não aceita o seu cartão de crédito. Sendo assim, é necessário ir até o caixa para pagamento em dinheiro, onde uma fila já está formada por outras pessoas e com isso o seu atraso aumentará. Ao sair do posto, percebe um acidente de trânsito, sendo inevitável uma rota alternativa para o trabalho. Chegando em sua mesa, apesar de todos os esforços, a pessoa chega atrasada, sem o seu café e provavelmente incomodada por todas essas causas.

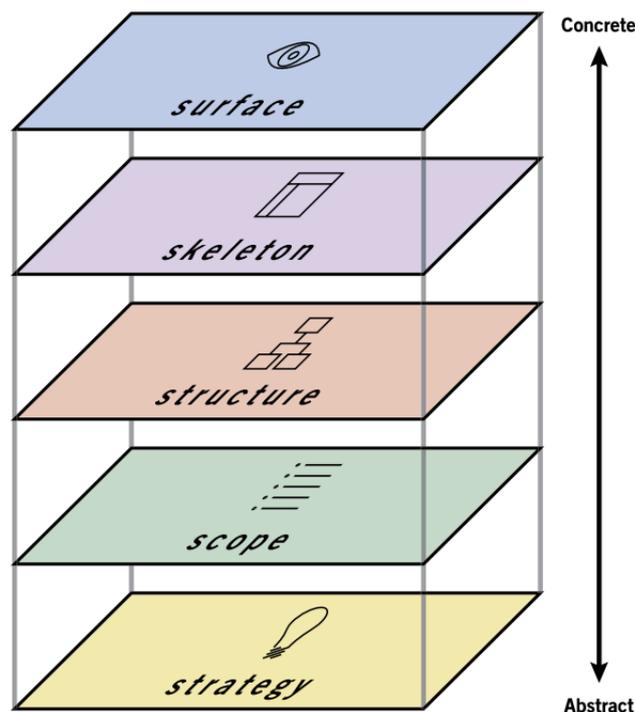
Entretanto, essa de série acontecimentos poderiam ter sido evitadas. O acidente aconteceu por causa de o motorista estar mexendo no rádio ao invés de olhar para frente, fazendo isso porque era impossível de verificar o volume do rádio sem focar a sua visão no mesmo. No posto de combustível, a máquina teria aceitado o cartão se ele fosse passado ao lado contrário do que a pessoa passou, mas não havia nada na máquina indicando isso. A fila de pagamento no posto de combustível demorou mais que o esperado porque o sistema de pagamento era complexo e confuso, não dando espaço para erros ao caixa. No caso da máquina de café, o botão não foi apertado completamente para ligá-la e, além disso, a máquina não fornece nenhum sinal explicitamente de que foi ligada, como por exemplo uma luz, um barulho ou visor que notifique o ligamento da mesma. No caso do relógio, causador inicial de todo atraso, ele não estava com horário errado. O que aconteceu foi que o gato da pessoa acabou pisando e apertando seu botão de *reset*, algo

que poderia ter sido evitado por uma configuração de botões diferente.

É nesse aspecto que a experiência de usuário é analisada. Por mais que existam inúmeros produtos e serviços disponíveis a todos atualmente, são os pequenos detalhes que sempre contam para uma boa experiência (GARRETT, 2011). Sendo assim, a experiência de usuário é indispensável para todos eles. Entretanto, sendo o presente trabalho referente a uma aplicação Web, será abordada a experiência de usuário para este tipo de produto.

Segundo Garret, a experiência de usuário para aplicações web pode ser construída a partir de 5 camadas fundamentais (GARRETT, 2011), sendo elas ilustradas conforme a Figura 3.3. A ordem de leitura e abordagem é de baixo para cima, as camadas mais abaixo sustentam as de cima por serem mais abstratas.

Figura 3.3 – Camadas para aplicações Web



Fonte: GARRET (2011, p. 22).

- **Estratégia (Strategy):** Se refere ao ponto inicial de desenvolvimento. Onde faz-se a pergunta do porquê está se desenvolvendo algo, o ponto de partida, e é fundamentada a criação, redesign ou melhoria de projeto na busca da criação de uma forma de interação (interface);

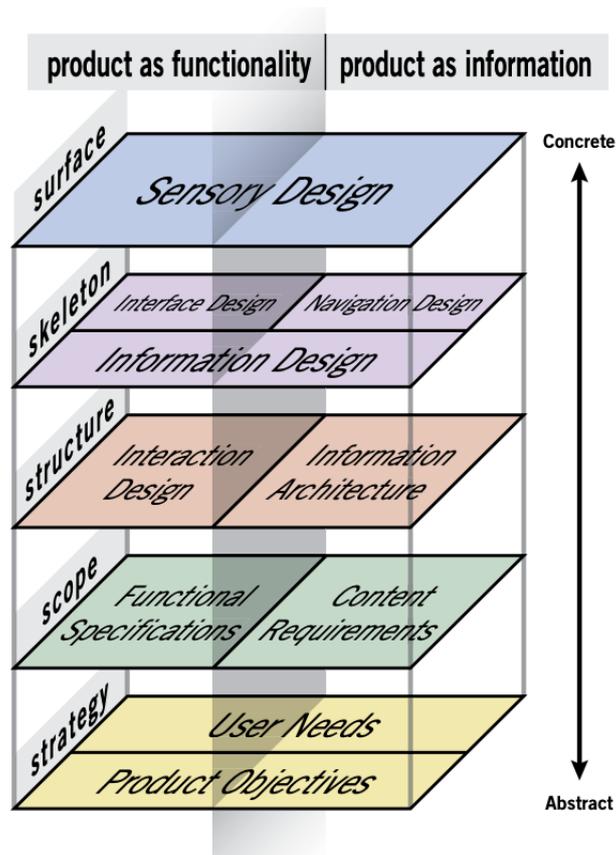
- **Escopo (*Scope*):** Camada que traz a abordagem sobre as limitações de um projeto. Quais as tecnologias que serão utilizadas para o seu desenvolvimento, o seu público alvo, entre outros pontos a considerar;
- **Estrutura (*Structure*):** Define a distribuição de dados e informações que o produto irá fornecer, a hierarquia e estrutura de tarefas que ele será capaz de realizar. Por exemplo considerar uma página em que o usuário se encontra, como ele chegou ali, quais caminhos pode navegar a partir do ponto em que está e quais informações acessar;
- **Esqueleto (*Skeleton*):** Localização em que os elementos da tela serão distribuídos, sendo eles botões, textos, imagens, etc. para uma máxima eficiência dinâmica para o usuário a utilizar a aplicação;
- **Superfície (*Surface*):** Camada que define a parte visual da aplicação, onde o usuário será capaz de interagir com os elementos apresentados a ele e visualizar a mesma. O ponto final dessa abordagem.

O entendimento e compreensão dessas camadas faz-se necessário para a próxima abordagem. Garret menciona que, com o crescimento e evolução das aplicações Web, as mesmas foram tendo uma maior capacidade de executar funcionalidades complexas e robustas. Desempenhando não apenas a coleta de informação mas sim a manipulação dela. Consequentemente, esse avanço teve impacto direto na experiência de usuário e em como projetá-la. Em virtude disso, essa projeção pode apresentar uma certa dualidade quando executada. Sendo essa dualidade representada por produtos orientados a **funcionalidade** ou produtos orientados a **informação**. Quando trata-se de funcionalidade, refere-se a tarefas, os passos envolvidos em um processo e o que as pessoas pensam ao completa-los. Por outro lado, quando trata-se de informação, refere-se a qual tipo de informação o produto apresenta ao usuário e o que ela representa para ele. Essa dualidade para as camadas está representada de acordo com a Figura 3.4.

- **Estratégia (*Strategy*):** As duas visões, tanto funcionalidade quanto informação, buscam atender as necessidades do usuário;
- **Escopo (*Scope*):** No que diz respeito a funcionalidade, corresponde a especificações funcionais do produto. Para informação, os elementos de conteúdo necessários e úteis ao usuário;

- **Estrutura (*Structure*):** Na visão de funcionalidade, o foco é na projeção de interação, ou seja, como a aplicação irá responder às ações do usuário. Na visão de informação, como o conteúdo estará distribuído em tela para um bom entendimento do usuário;
- **Esqueleto (*Skeleton*):** Para a visão de funcionalidade, aborda-se a projeção da interface, onde os elementos interativos irão proporcionar os usuários da aplicação interagir de uma forma fácil com as funcionalidades do sistema. Para a visão de informação, os elementos em tela que permitem o usuário navegar pela arquitetura de informação;
- **Superfície (*Surface*):** Para ambas as visões, a camada trata de aspectos sensoriais do usuário ao utilizar a aplicação, o momento em que ele percebe as funcionalidades e comunicação de seus elementos.

Figura 3.4 – Camadas tratadas como funcionalidade e informação



Fonte: GARRET (2011, p. 29).

Esses conceitos foram abordados e levados em consideração para a construção,

projeção e desenvolvimento das novas funcionalidades incluídas no Sistema de Procedimentos de Testes da Covid-19. Uma das vantagens que pode-se levar em consideração é que, por se tratar de um sistema que já está em uso por um tempo perdurável, os seus usuários já tem uma certa experiência em seu uso. Sendo assim, essas novas funcionalidades buscam manter alguns aspectos estruturais relacionados a disposição de funcionalidades e de informações apresentadas pela aplicação sem mudanças bruscas, de modo a não prejudicar esse padrão de usabilidade já proporcionado. Não significa também que essas novas funcionalidades não direcionem o usuário na aplicação, mas o usuário já entende alguns pontos do sistema e como ele funciona.

Todas as abordagens descritas fornecem uma ideia abstrata e conceitual para o desenvolvimento das novas funcionalidades relacionadas a experiência de usuário. Entretanto, por essas funcionalidades se tratarem de algo perceptível visualmente ao usuário, a última camada, de superfície, tem destaque. Em virtude disso, alguns princípios de *design* foram aplicados nessa camada (ROGERS; SHARP; PREECE, 2013). São eles:

- **Visibilidade:** O layout apresentado ao usuário deve mostrar as suas funções o mais visível possível, assim os seus usuários saberão o que fazer. As funcionalidades novas adicionadas ao SPT-Sars-Cov-2 buscam orientar os seus usuários no seu fluxo de execução com elementos da tela;
- **Feedback:** Se relaciona a ideia de visibilidade. Todas as ações do usuário devem ter um retorno de informação, se algo ocorreu bem ou não;
- **Restrições:** Consiste em limitar a interação do usuário. Uma das funcionalidades adicionadas só mostra elementos em tela que dependem do momento no fluxo de execução, só é necessário mostrá-las após ações prévias do usuário;
- **Mapeamento:** Conexão entre controles disponíveis ao usuário e os seus efeitos. Por exemplo, na tabela em que são mostrados os dados resultantes das novas funcionalidades (que será mostrada em capítulos posteriores) todas as colunas apresentam setas para cima ou para baixo, dependendo da ordenação desejada;
- **Consistência:** Princípio de *design* chave para as novas funcionalidades. "Projetar interfaces de modo que tenham operações semelhantes e que utilizem elementos semelhantes para a realização de tarefas similares"(ROGERS; SHARP; PREECE, 2013, p.45), ou seja, as novas funcionalidades aplicadas não realizam equivalente-

mente as mesmas tarefas, mas em alguns aspectos são semelhantes às já existentes.

3.3 Padrões de Projeto

Os padrões de projeto consistem em um conjunto de práticas que tem como objetivo principal aumentar o fator de reusabilidade de um software (Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, 2006). Sendo assim, o desenvolvimento do sistema SPTSars-Cov-2 foi baseado conforme os padrões descritos nas próximas subseções.

3.3.1 Padrão MVC

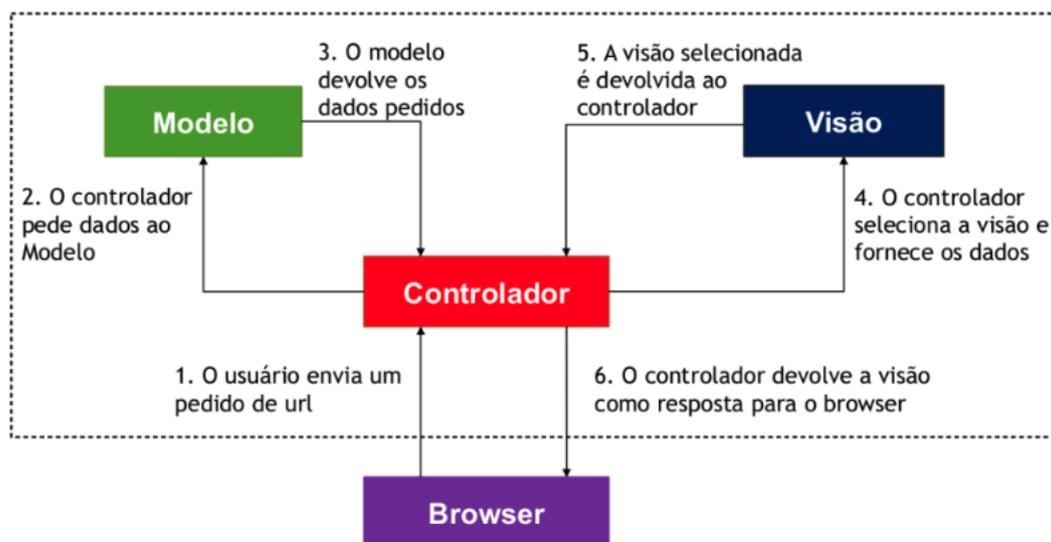
Conforme o nome indica, é formado por 3 componentes:

- **Modelo:** representa a camada da aplicação que organiza os dados;
- **Visão:** representa a interface do usuário;
- **Controlador:** responsável pela integração entre Modelo e Visão.

No caso do Sistema de Procedimentos de Testes da Covid-19, uma possível analogia seria pensar a seguinte sequência:

1. usuário faz alguma ação na tela (camada de visão) representada pela classe **formulario.php**, disparando uma requisição ao controlador, representado pela classe **controlador.php**;
2. o controlador **controlador.php** aciona a camada de modelo com base na ação do usuário;
3. a camada de modelo por sua vez aplica as regras de negócios para fazer uma filtragem no banco de dados do sistema e responde ao controlador;
4. ao receber a resposta, o controlador fornece esses dados recebidos para a camada de visão;

Figura 3.5 – Modelo MVC



Visão ilustrativa e simplificada de como funciona o modelo MVC.

Fonte: LOPES (2015).

5. a camada de visão processa as modificações que serão necessárias e devolve o HTML modificado para o controlador;
6. o controlador devolve o html modificado recebido em tela para o usuário.

3.3.2 DTO e DAO

Os dois são descritos em uma única seção em função de estarem sendo usados de forma integrada no sistema. No que diz respeito ao padrão DTO (*Data Transfer object*), consiste em objetos que tem como função principal transferir dados de uma parte para outra na aplicação, não sendo elaborados e são relativamente simples em sua estrutura. São usados normalmente vinculados a camadas que fazem a comunicação com o banco de dados, em função disso visam ocultar a camada de persistência das demais classes (ALUR; CRUPI; MALKS, 2003). No SPTSars-Cov-2, ele é usado na comunicação entre as camadas de visão e modelo.

Quando falamos sobre o padrão DAO (*Data Access Object*), corresponde a ideia de que a camada de persistência deve estar deslocada em relação as outras partes da aplicação. Sendo assim, qualquer acesso aos dados, seja consulta, alteração ou remoção fica restrito a essa camada (ALUR; CRUPI; MALKS, 2003).

Usando um padrão juntamente com outro, tem-se origem ao objeto BO (*Business*

Object) do sistema, para fins de coesão do código e aumentar a reusabilidade do mesmo. Esse objeto permite separarmos as regras de negócios da lógica da persistência. Objeto implementado de acordo com a ideia apresentada na Figura 3.6.

Figura 3.6 – Objeto BO

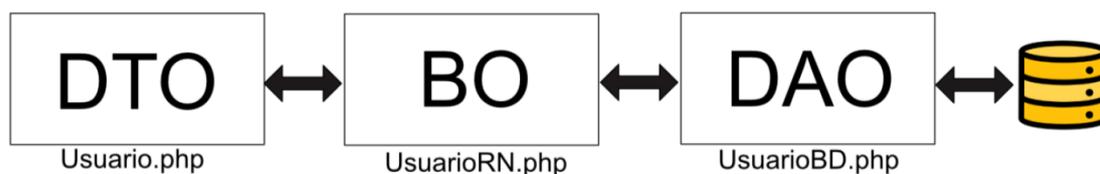


Ilustração do conceito de objeto BO aplicada no SPT-Sars-Cov-2.

Fonte: BATHAGLINI (2021, p.23)

3.3.3 Singleton

O padrão Singleton consiste em fornecer um ponto global de acesso para uma classe e assegurar que a mesma tenha somente uma instância (Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, 2006). Isso significa que a classe gerencia a sua instância e não permite que uma outra classe crie uma instância dela. Esse padrão é utilizado na classe Pagina no código fonte do SPTSars-Cov-2, como mostra a Figura 3.7

A classe ConsultaPersonalizadaINT tem o atributo privado e estático *instance* e a função *getInstance*. Quando esta função é chamada, é feita uma consulta no valor desse atributo. Quando tem valor nulo, ou seja, ainda não foi criada a instância, a mesma é criada nesse momento. Caso contrário, é utilizada a instância já criada anteriormente.

As principais justificativas para utilização desse padrão de projeto para essa classe foram com relação a comunicação com o banco de dados, pois é feita apenas uma conexão para todo o sistema, então não há necessidade de outras instâncias da classe. Além disso, as páginas que compõem as interfaces do sistema são também montadas seguindo esse padrão, pois "todas elas apresentam uma consistência com um menu superior, um menu lateral esquerdo e com o corpo da página. Desse modo, uma única instância monta todas as partes que compõem as interfaces visuais"(BATHAGLINI, 2021).

Figura 3.7 – Classe ConsultaPersonalizadaINT

```
class ConsultaPersonalizadaINT
{
    private static $instance;
    public $tabela;

    public static function getInstance()
    {
        if (self::$instance == null) {
            self::$instance = new ConsultaPersonalizadaINT();
        }
        return self::$instance;
    }
}
```

Utilização do padrão Singleton para a classe ConsultaPersonalizadaINT.

Fonte: Elaborado pelo autor (2022).

3.4 Programação orientada a objetos

A programação orientada a objetos é o padrão de desenvolvimento que trabalha com o conceito de classes e objetos que compõem as partes de um sistema. Uma classe representa a definição, documentação e a especificação das características de um objeto. O objeto por sua vez representa uma instância derivada de uma classe, é criado conforme o que foi descrito na mesma (CARVALHO, 2016).

Fazendo uma analogia ao mundo físico, pode-se dizer que a classe seria a planta de uma casa, descrevendo a sua parte estrutural, planejamento e modelo a ser seguido para que uma casa seja construída e o objeto seria a casa em si, algo concreto e do mundo físico, onde nela está presente, de fato, o que foi definido na classe.

A estrutura do sistema SPTSars-Cov-2 se baseia em pilares desse paradigma, são eles:

- **Abstração:** consiste em considerar apenas o necessário no que diz respeito ao que a classe irá definir, ou seja, focando apenas no que há de importante no que um objeto irá possuir ao ser criado, ignorando-se detalhes desnecessários e levando a atenção para a principal função de um objeto no sistema;
- **Encapsulamento:** nem tudo que um objeto possui necessariamente precisa estar

visível ao mundo externo, garantindo segurança na comunicação com outros;

- **Herança:** conceito essencial para reuso de código. Significa que objetos herdam características de outros. Quando isso acontece, a reescrita de código torna-se desnecessária.

4 TECNOLOGIAS UTILIZADAS

Qualquer que seja a atualização de um sistema ou adição de novas funcionalidades, seja ela pequena ou grande, sempre deve-se revisar e ter conhecimento das tecnologias usadas pelo mesmo, por precaução, visto que foram analisadas as possibilidades de impacto tanto interno quanto externo. "A preocupação da produção de um software de qualidade não se restringe apenas a atingir os objetivos (ou requisitos) esperados pelos usuários, mas também em obter um ciclo de vida de produção de software que apresente as características desejáveis em qualquer processo de desenvolvimento de software, como boa manutenibilidade, alta reusabilidade e baixo acoplamento"(BALTHAZAR, 2007).

A atualização do projeto não modificou quaisquer tecnologias já utilizadas para o desenvolvimento da versão atual do SPTSars-Cov-2, visando não impactar o seu comportamento já consolidado e compartilhando do pensamento da citação do parágrafo anterior. Portanto, as novas funcionalidades desenvolvidas foram fruto da manipulação dos recursos já existentes no sistema. São elas:

- **Linguagem de programação PHP:** orientada a objetos, a linguagem PHP apresenta uma boa compatibilidade com diversos sistemas operacionais, bancos de dados e também pode ser utilizada para desenvolver um sistema no padrão MVC (DALL’OGLIO, 2016). Usada para desenvolver a camada de backend desse sistema;
- **MariaDB:** é um dos mais populares bancos de dados relacionais de código aberto. Dos mesmos criadores do MySQL, prega pelo bom desempenho e estabilidade (MariaDB Foundation, 2022);
- **HTML (*HyperText Markup Language*):** Sua finalidade é a de determinar a estrutura de uma página web. É formada por uma coleção de elementos, chamados de hipertextos ou tags, que indicam ao navegador que tipo de estrutura deve aparecer em tela (Beatriz Moura e Matheus Henrique, 2021);
- **CSS (*Cascading Style Sheets*):** é um mecanismo para a estilização de páginas web (The World Wide Web Consortium (W3C), 2022);
- **JavaScript:** linguagem de programação para desenvolver interfaces dinâmicas e interativas com usuários, através da manipulação de HTML (Daniela Rocha Silva e

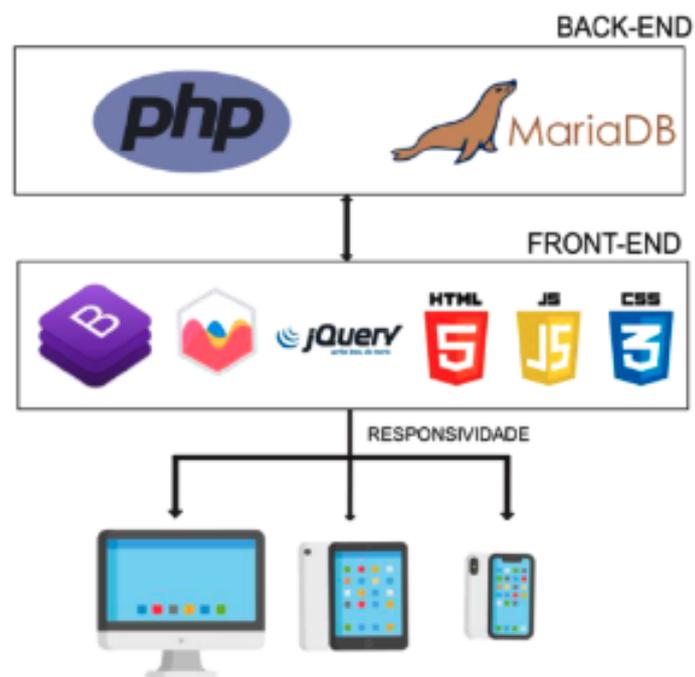
Luis Felipe Bentin Sobral, 2017). Foi usada para desenvolver a camada de front-end do sistema;

- **Bootstrap:** ferramenta para desenvolvimento front-end de aplicações web com HTML, CSS e JavaScript. Permite a utilização de componentes já pré-construídos para interfaces (BOOTSTRAP, 2022);
- **jQuery:** é um framework para JavaScript, tem códigos prontos para facilitar a manipulação do HTML (JQUERY, 2022);
- **Chart.js:** biblioteca JavaScript para a criação de diversos gráficos dinâmicos em tela (CHART.JS, 2022).

Sendo assim, as tecnologias descritas estão distribuídas de acordo com a representação gráfica das camadas da aplicação mostrada na Figura 4.1.

O modo como o Sistema de Procedimento de Testes da Covid-19 foi desenvolvido, permitiu essa prevenção de impacto por meio das novas funcionalidades desenvolvidas e, além disso, não foram necessárias a adição de novas tecnologias. Essas funcionalidades são capazes de exercerem as suas aplicabilidades de forma isolada. Sendo assim, caso sejam necessários futuros refinamentos sob as mesmas, não há necessidade de maiores mudanças no que diz respeito aos aspectos internos e arquiteturais desse sistema.

Figura 4.1 – Distribuição das tecnologias utilizadas por camada



Fonte: BATHAGLINI (2021, p.26)

5 PRIMEIRO INCREMENTO

A primeira funcionalidade nova desenvolvida para o Sistema de Procedimento de Testes da Covid-19 consiste em uma filtragem na base de dados para pacientes que testaram positivo em um certo intervalo de tempo. Sendo este intervalo de tempo personalizado pelo usuário.

Essa nova funcionalidade veio em consequência da fase em que a pandemia se encontra desde o seu início, no caso especificamente na segunda metade do ano de 2022. O número de pacientes repetidos, ou seja, que tiveram laudo com resultado como detectado ou positivo pode ter aumentado.

Eventualmente, caso seja feita uma busca de um modo mais incisivo sobre os laudos e relacionando-os com os pacientes registrados na base de dados do sistema, há a possibilidade de que pacientes repetidos sejam encontrados. Entretanto, o sistema ainda não apresentava uma funcionalidade exclusiva para isso, ainda mais com a particularidade de uma filtragem temporal.

5.1 O processo de geração de laudo

Todo laudo, para ser gerado, precisa de um diagnóstico prévio da amostra. O desenvolvimento da geração de diagnósticos relacionado às amostras armazenadas no Sistema de Procedimentos de Testes da Covid-19 não será descrito detalhadamente, por estar fora do escopo do trabalho. Esse processo consiste em um dos incrementos iniciais do sistema, sendo este incremento especificamente o incremento 4, denominado "Diagnóstico"(BATHAGLINI, 2021).

Entretanto, faz-se necessário um breve levantamento sobre o que seria considerado um laudo válido, e o processo da sua geração, pois são a base de consulta para o primeiro incremento desenvolvido neste trabalho. Laudos válidos consistem em laudos que apresentam um resultado finalizado do teste RT-PCR, ou seja, positivo ou negativo, resultante por meio do seu diagnóstico, e ainda que receberam uma confirmação do usuário administrador.

Para a geração de um laudo são necessários: o paciente envolvido na coleta, a amostra da mesma e o resultado do diagnóstico, dados que são possíveis de serem consultados no sistema. Cada laudo consiste em um documento formalizado sobre esses dados, fornecendo um detalhamento das informações de cada um dos mesmos. Como informa-

ções pessoais do paciente, data de exame, quais *kits* de extração de coleta foram usados, entre outras.

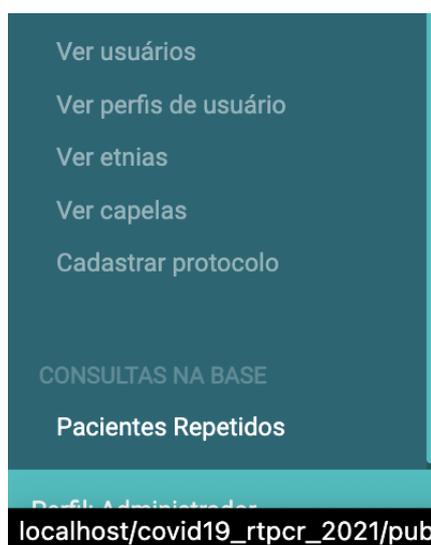
Um laudo pode ser gerado por usuários do tipo **Laudo**. No momento em que um laudo é gerado, esse momento corresponde a sua **data hora geração**. Entretanto, cada alteração em laudos por meio desses usuários necessita da autorização do usuário administrador, resultando na **data hora liberação** do laudo. Com essa informação por meio do usuário administrador, o laudo apresenta-se na **situação** de "Concluído", do contrário, "Pendente".

Portanto, um laudo válido e de resultado positivo, deve apresentar uma **data hora liberação**, uma **situação** como "Concluído" e **resultado** como "Detectado".

5.2 Acessando a funcionalidade

Para ter acesso a essa nova funcionalidade, o usuário precisa ter associado ao seu perfil o recurso de "pacientes_repetidos". Fazendo isso, conseqüentemente, o mesmo perceberá que terá acesso ao novo módulo no menu de funcionalidades do sistema, determinado como "CONSULTAS NA BASE". Com o recurso já associado ao seu perfil, o usuário acessa essa funcionalidade ao selecionar a opção "Pacientes Repetidos". Acesso este que é apresentado ao usuário conforme a Figura 5.1.

Figura 5.1 – Opção "Pacientes Repetidos" do novo módulo



Fonte: Elaborado pelo autor (2022).

5.3 Arquivos adicionados ou modificados

Os arquivos adicionados ou modificados para o desenvolvimento dessa consulta estão listados conforme a tabela 5.1.

Tabela 5.1 – Arquivos adicionados ou modificados para o primeiro incremento

Arquivo	Finalidade
pacientesRepetidos.php	Arquivo novo adicionado ao sistema do tipo <i>action</i> , responsável por manipular as instâncias de classes para a consulta a ser executada, além de relacionar os elementos estruturais da tela com os dados, nas quais os mesmos sendo fruto dessa manipulação de instâncias necessária.
controlador.php	O controlador foi alterado para que fosse necessária a validação do novo recurso adicionado, "pacientes_repetidos", fazendo uma ligação com o novo arquivo <i>action</i> mencionado.
Pagina.php	Sendo a classe responsável por renderizar o menu lateral esquerdo de funcionalidades do sistema, foi adicionada a nova opção de "Pacientes Repetidos" para o novo módulo criado "CONSULTAS NA BASE".
pacientesRepetidosINT.php	Novo arquivo adicionado ao projeto na qual representa a classe que tem por objetivo montar os elementos estruturais da interface a ser apresentada.
pacientesRepetidosRN.php	Arquivo novo que tem como função principal ser o meio de comunicação entre a classe pacientesRepetidosBD.php e o arquivo <i>action</i> pacientesRepetidos.php por meio de uma instância da classe BD que esse arquivo dispõe. Além disso, esse arquivo separa a classe do arquivo mencionados inclusive para isolar a consulta ao banco de dados da interface.
pacientesRepetidosBD.php	Classe adicionada ao sistema responsável por realizar a consulta ao banco de dados do mesmo, se comunicando apenas com a classe pacientesRepetidosRN.php .

Fonte: Elaborado pelo autor (2022).

5.4 Fluxo de execução

No momento em que o usuário acessa a funcionalidade de "Pacientes Repetidos", é apresentada a tela de acordo com a Figura 5.2, consistindo em dois campos para input de data. O primeiro para a data de início da pesquisa, e o segundo para a data final, respectivamente. Sendo assim, a consulta será feita para pacientes repetido, ou seja, que apresentaram laudos positivos e válidos desde a data do primeiro input até a data do segundo input.

Figura 5.2 – Início do fluxo para a funcionalidade de pacientes repetidos

Consultas Sugestões Relatar problema

Este menu consiste em consultar os pacientes que testaram positivo mais de uma vez em um determinado intervalo de tempo.

Informe a data de início:

Informe a data de fim:

Consultar

Fonte: Elaborado pelo autor (2022).

Além de já indicar o formato de data que deve ser inserido, no caso dd/mm/aaaa, cada campo de input ainda apresenta um calendário clicável onde pode-se inserir uma data específica manualmente, como mostra a Figura 5.3.

Ao selecionar as datas desejadas, o usuário deve clicar no botão de rótulo "Consultar". Assim então, dá-se início ao fluxo de execução da consulta. O arquivo *action pacientesRepetidos.php* detecta a ação em tela e notifica a sua instância do tipo **PacientesRepetidosRN.php** informando a data de início e data de fim da consulta. Dentro da instância **PacientesRepetidosRN**, a instância do tipo **PacientesRepetidosBD** é acionada, fazendo a consulta ao BD conforme as datas enviadas já definidas pelo usuário.

Para realizar a consulta, é necessário antes fazer a filtragem de laudos válidos considerando as datas inseridas. Sendo assim, na tabela "tab_laudo", são considerados os laudos que tem situação como concluída, resultado como positivo, que a sua data de liberação seja no mínimo maior ou igual que a data de início informada pelo usuário e no máximo menor ou igual que a sua data de fim informada pelo mesmo. Respeitando todas essas condições, são filtradas todas as amostras que estão relacionadas com esses

Figura 5.3 – Calendário clicável em cada campo de input

forme a data de inicio:

forme a data de fim:

The image shows a web form with two input fields for dates. The first field is labeled 'forme a data de inicio:' and contains the placeholder 'dd/mm/aaaa'. The second field is labeled 'forme a data de fim:' and contains 'dd'. Below these fields is a dark green button labeled 'Consultar'. A calendar dropdown is open for the first field, showing the month of September 2022. The calendar grid has columns for days of the week (S, T, Q, Q, S, S, D) and rows for dates. The date 28 is highlighted in blue. At the bottom of the calendar, there are two links: 'Limpar' and 'Hoje'.

Fonte: Elaborado pelo autor (2022).

laudos, pois a tabela de laudos apresenta uma chave *idAmostra_fk*, chave primária para cada amostra da tabela "tb_amostra".

Com as amostras filtradas a partir dos laudos válidos, é feita a segunda filtragem, na tabela de amostras "tb_amostra". Cada amostra está relacionada com um paciente, através da chave primária *idPaciente_fk*, chave primária para a tabela "tb_paciente". Essa consulta retorna todos os pacientes relacionados com essas amostras.

Tendo os pacientes filtrados, é feita uma validação nesse conjunto retornado para verificar quais estão repetidos. Sendo assim, esses são os pacientes repetidos válidos, pois apresentam as condições necessárias.

O último passo não se trata necessariamente de uma filtragem, mas sim de uma consulta na tabela de pacientes novamente, para o início da construção da tabela dinâmica que será apresentada ao usuário com os dados resultantes dessas filtragens mencionadas. Essa consulta consiste em verificar quais são os tipos de dados disponíveis presentes na tabela de pacientes, onde cada um deles corresponderá a uma coluna dessa tabela dinâmica. Consulta esta realizada da mesma forma que a interação dos arquivos descrita anteriormente.

O resultado dessa consulta é enviado para a instância contida no arquivo *action* do tipo **PacientesRepetidosINT.php**, que usa tanto esse resultado quanto a resposta das filtragens executadas para a montagem da tabela dinâmica, que é apresentada ao usuário conforme a Figura 5.4. Onde são fornecidos ao usuário os dados dos pacientes repetidos disponíveis na base de dados, conforme estão determinados na tabela "tb_pacientes".

Figura 5.4 – Dados dos pacientes repetidos

Mostrando registros por página

Pesquisar

IDPACIENTE ↑	IDSEXO_FK ↑	NOME ↑	NOMEMAE ↑	CPF ↑	RG ↑	CEP ↑
72120	2	LUCAS TESTE	MÃE DE LUCAS TESTE	012345		9176000
72124	4	TESTE PACIENTE XXXX				

Mostrando página 1 de 1

Anterior Próxima

Dados meramente ilustrativos resultantes de testes da funcionalidade.
 Fonte: Elaborado pelo autor (2022).

Realçando-se a tabela exibida ao usuário no final do fluxo, a mesma consiste em uma tabela totalmente personalizável. Apresenta quatro fortes recursos para aprimorar a experiência de usuário. Sendo o primeiro relacionado a ordenamento dos dados, uma vez que permite que o usuário ordene os dados conforme a coluna a sua escolha por ordem crescente ou decrescente. O segundo está relacionado com o primeiro, e destaca a questão da visibilidade ao usuário, cada coluna apresenta ícones indicando qual a coluna foi escolhida pelo usuário para ordenação além de indicar se a ordenação está no momento em ordem crescente ou decrescente. O terceiro destaca-se pela quantidade de dados personalizável ao usuário, o mesmo pode escolher quantos registros (linhas de tabela) máximos deseja visualizar por página da tabela com a opção do canto superior esquerdo. O quarto vai de encontro com filtragem de dados na tabela, sendo possível o usuário pesquisar por um dado específico em todos da tabela com a caixa de pesquisa do canto superior direito.

Sendo assim, esse detalhamento consiste no fluxo de execução do primeiro incremento desenvolvido. A etapa de maior complexidade do seu desenvolvimento foi com relação aos requisitos das filtragens que devem ser realizadas para a obtenção de dados válidos no objetivo de apresentação dos mesmos ao usuário em interface. Essa filtragem de certo modo poder ser classificada como minuciosamente específica, visto que relacionou tabelas diferentes com tipos de dados diferentes, a fim de produzir a tabela com os dados resultantes.

6 SEGUNDO INCREMENTO

A segunda funcionalidade desenvolvida para o sistema consiste em uma consulta personalizada ao banco de dados do mesmo. Essa consulta permite ao usuário selecionar qual a tabela que deseja consultar e ainda filtrar por meio de colunas para que se apresentem em concordância com os dados que o mesmo pretende visualizar disponíveis na base de dados.

Essa funcionalidade foi inspirada na ideia de que o usuário administrador pretendia conceder a outros usuários a visualização de como o banco de dados do sistema está estruturado e o que o mesmo apresenta em seu armazenamento.

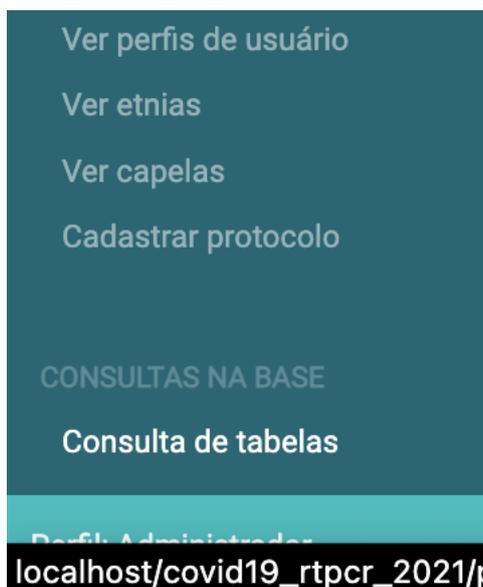
Por ser executado por meio da ferramenta *XAMPP*, consistindo em um ambiente para projetos PHP (Apache Friends, 2022), o ambiente em que o código do sistema é executado já fornece um recurso para uma visualização e acesso ao banco de dados do projeto. Entretanto, esse recurso permite a alteração, remoção e inclusão de novos dados ou tabelas.

Para evitar qualquer tipo de manipulação de dados da base, foi incluída essa funcionalidade com o intuito de proporcionar ao usuário uma visão mais clara e ampla sobre com quais dados esse sistema interage, sem a possibilidade de alteração desses dados. Destaca-se não apenas o impedimento de alteração, mas também o fato de que a funcionalidade é capaz de detectar dados corrompidos, tabelas não utilizadas, entre outras possibilidades de pendências da base.

6.1 Acessando a funcionalidade

Para ter acesso a essa nova funcionalidade, o usuário precisa ter associado ao seu perfil o recurso de "consulta_personalizada". Fazendo isso, conseqüentemente, o mesmo perceberá que terá acesso ao novo módulo no menu de funcionalidades do sistema, determinado como "CONSULTAS NA BASE". Uma vez que já tem o recurso associado ao seu perfil, o usuário acessa essa funcionalidade ao selecionar a opção "Consulta de tabelas". Acesso este apresenta-se conforme a Figura 6.1.

Figura 6.1 – Opção "Consulta de tabelas" do novo módulo



Fonte: Elaborado pelo autor (2022).

6.2 Arquivos adicionados ou modificados

Para o desenvolvimento dessa consulta, foram adicionados ou modificados os arquivos conforme a distribuição da tabela 6.1.

6.3 Fluxo de execução

Ao selecionar essa nova opção, o usuário já visualiza o início do fluxo para a consulta. A interface consiste em uma descrição do principal objetivo da funcionalidade a ser utilizada, além de apresentar ao usuário um componente *dropdown* que, ao ser selecionado, apresenta todas as tabelas disponíveis na base de dados do sistema, sendo resultado de uma consulta ao mesmo pela camada de *backend*, com a finalidade da escolha de qual tabela será consultada. Interface inicialmente apresentada ao usuário de acordo com a Figura 6.2 em conformidade com a Figura 6.3.

Em seguida, ao selecionar a tabela para consulta dentre as opções apresentadas no componente *dropdown*, o usuário, intuitivamente, deve clicar no botão "Selecionar tabela", onde dá-se início de quais os dados a tabela apresenta para consulta na base.

Uma vez que o botão é clicado, automaticamente, o controlador **controlador.php** notifica o arquivo *action* **consultaPersonalizada.php** que por sua vez notifica a sua ins-

Figura 6.2 – Apresentação da funcionalidade de consulta personalizada ao usuário

Consultas Sugestões Relatar pr

Este menu consiste em consultar os dados armazenados na base de dados do sistema.

Tabelas disponíveis da base dados:

Escolha a tabela

Selecionar tabela

Fonte: Elaborado pelo autor (2022).

tância da classe **ConsultaPersonalizadaRN.php** que uma consulta de tabela será realizada, tabela esta selecionada pelo usuário. Com isso, este último arquivo solicita a consulta ao banco de dados por meio de sua instância da classe **ConsultaPersonalizadaBD.php**, que realiza a consulta em questão e retorna o resultado nesse fluxo até o arquivo *action* novamente.

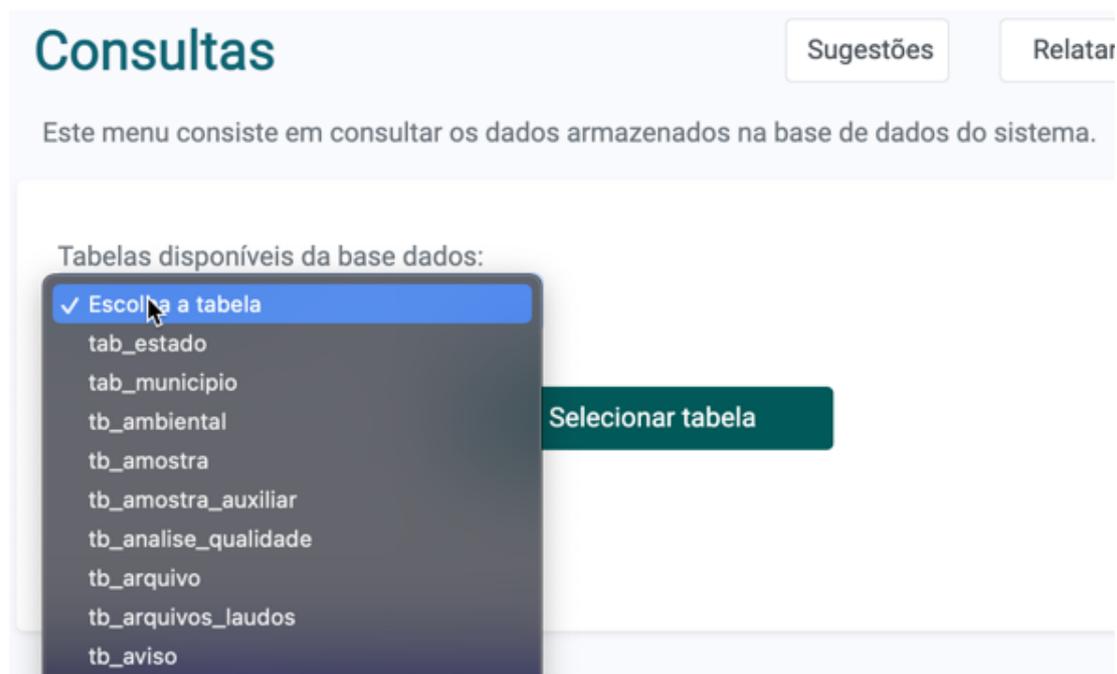
A consulta de tabelas disponíveis na base de dados para apresentação no primeiro *dropdown* funciona de forma semelhante ao fluxo descrito no parágrafo anterior, entretanto, mesma é ativada instantaneamente no momento em que o usuário acessa essa funcionalidade, pois já devem ser fornecidas as tabelas para consulta antes de qualquer ação na interface.

Realizada a consulta dos dados disponíveis na tabela e os mesmos estando disponíveis no nível do arquivo *action*, esse arquivo envia esses dados resultantes para sua instância da classe **ConsultaPersonalizadaINT.php**, onde os novos elementos da tela são montados com esses dados. Resultando em mais um componente *dropdown* acompanhado de um novo botão.

Sendo assim, o resultado dessa consulta após a ação do usuário deve apresentar quais os dados que podem ser consultados na tabela selecionada que ficarão disponíveis nesse segundo componente *dropdown*. Dados estes que serão a primeira coluna da tabela dinâmica que será apresentada no fim desse fluxo de execução da funcionalidade. Estes novos componentes, apresentados após o clique do botão "Selecionar tabela", apresentam-se inicialmente de acordo com a Figura 6.4.

Ao acessar o novo componente *dropdown*, já são apresentados os dados que estão

Figura 6.3 – Início do fluxo



Fonte: Elaborado pelo autor (2022).

presentes na tabela selecionada anteriormente, conforme mostra a Figura 6.5. Os dados disponíveis na tabela "tab_estado" são apresentados nessa fase do fluxo de execução.

E assim o fluxo segue a sua mesma concepção, até que se chegue na coluna de número 3. No momento em que a terceira coluna é selecionada, ou seja, o botão de "Selecionar coluna 3" é clicado, novos componentes são apresentados em tela, além de novos botões para possibilidades alternativas no fluxo de execução. Botões estes exibidos em tela ao usuário conforme mostra a Figura 6.6.

Destacando-se o passo principal dentre os outros da funcionalidade descrita, esse é o momento em que o arquivo *action consultaPersonalizada.php* detecta que a terceira coluna foi selecionada.

Sendo assim, o mesmo consulta os dados de cada coluna selecionada na tabela selecionada por meio de sua instância da classe **ConsultaPersonalizadaRN**, através do processo descrito anteriormente, para cada uma das três colunas, onde cada coluna corresponde a um conjunto de dados do mesmo tipo presente na tabela selecionada.

Feitas as consultas retratadas, o arquivo *action consultaPersonalizada.php* inicia a construção da tabela dinâmica por meio da sua instância da classe **ConsultaPersonalizadaINT.php** com os dados resultantes das consultas de cada coluna, representando o final da consulta personalizada pelo usuário, principal finalidade dessa funcionalidade

Figura 6.4 – Apresentação dos novos componentes

Tabelas disponíveis da base dados:

tab_estado

Selecionar tabela

Dados disponíveis armazenados nesta tabela:

Escolha a coluna

Selecionar coluna 1

Fonte: Elaborado pelo autor (2022).

Figura 6.5 – Apresentação dos dados contidos na tabela selecionada

Tabelas disponíveis da base dados:

tab_estado

Selecionar tabela

Dados disponíveis armazenados nesta tabela:

✓ Escolha a coluna

cod_estado

sigla

nome

Selecionar coluna 1

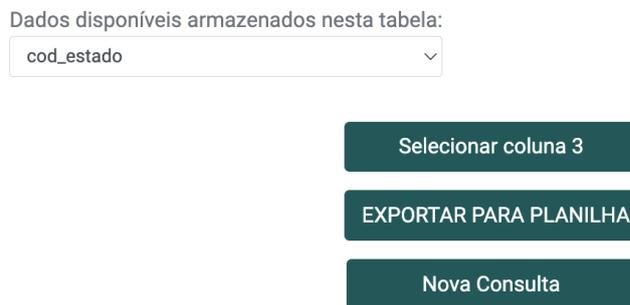
Fonte: Elaborado pelo autor (2022).

descrita.

A tabela dinâmica mencionada consiste em uma tabela equivalente a demonstrada na funcionalidade do capítulo anterior, visando aprimorar a experiência de usuário também para esta funcionalidade. Por meio da possibilidade de ordenar cada coluna na ordem em que deseja com ícones indicando a escolha, filtrar algum dado específico na caixa de pesquisa e selecionar a quantidade de dados visíveis na página em que a tabela se encontra. A tabela construída com os dados desta funcionalidade apresenta-se em interface para o usuário conforme a Figura 6.7.

Destacando-se também outra aplicabilidade da consulta personalizada desenvolvida, um dos novos botões que apresentam-se em tela após a seleção da terceira coluna da tabela dinâmica resultante dos dados é o botão com o rótulo "EXPORTAR PLANILHA",

Figura 6.6 – Apresentação dos novos botões para caminhos alternativos no fluxo



Fonte: Elaborado pelo autor (2022).

apresentado na Figura 6.6.

Intuitivamente, esse botão permite ao usuário um meio diferente de visualizar os dados resultantes da consulta realizada. Ao clicar no botão, é processado um *download* no navegador em que o usuário se encontra, consistindo em um arquivo na forma de planilha composta por esses dados, equivalentemente ao que pode-se ver na tabela dinâmica. O arquivo é nomeado automaticamente no formato "emitido_em_{data}{horário}", onde {data} corresponde a data de exportação da planilha no formato aaaa-mm-dd e {horário} corresponde ao horário de exportação da planilha no formato hh_mm_ss, sendo exportação da planilha o momento em que o botão é clicado. A planilha apresenta-se conforme a Figura 6.8.

Um ponto a destacar-se da funcionalidade de exportar planilha no fluxo descrito até então é de que ela vai de encontro a funcionalidades já presentes no sistema que apresentem uma grande quantidade de dados em sua maioria. Afirmando-se isso e, considerando que algumas tabelas da base de dados do sistema são muito extensas, procurou-se manter um comportamento semelhante com relação a exportação de planilhas para apresentação de um grande número de dados.

O outro botão apresentado ao escolher-se a terceira coluna, é o de rótulo "Nova Consulta". Esse botão tem por finalidade iniciar uma nova consulta personalizada, de modo que o usuário volte a um estado do fluxo em conformidade com o que apresenta-se na Figura 6.2. Fornecendo a possibilidade de personalizar a consulta de uma nova tabela, ou ainda da mesma tabela consultada anteriormente mas com uma ordenação diferente de colunas, caso seja necessário.

Portanto, dá-se assim a descrição e detalhamento do fluxo de execução do segundo incremento adicionado ao sistema. A maior complexidade desse fluxo no desenvolvi-

Figura 6.7 – Tabela personalizável com dados resultantes da consulta

Mostrando registros por página

Pesquisar

NOME 	SIGLA 	COD_ESTADO 
Acre	AC	12
Alagoas	AL	27
Amapá	AP	16
Amazonas	AM	13
Bahia	BA	29
Ceará	CE	23
Distrito Federal	DF	53
Espírito Santo	ES	32
Goiás	GO	52

Mostrando página 1 de 3

Anterior Próxima

Fonte: Elaborado pelo autor (2022).

mento deste incremento foi com relação ao desenvolvimento de uma tabela dinâmica sem saber quais serão os dados de entrada. Até então, o sistema ainda não apresentava uma funcionalidade que realizasse uma consulta no seu BD onde nem a tabela a ser consultada era definida, nem qual dado da mesma seria filtrado, fazendo-se necessário um novo algoritmo capaz de detectar esses pontos em tempo de execução e assim realizar uma consulta e filtragem sobre eles.

Figura 6.8 – Planilha exportada com dados resultantes da consulta

	A	B	C	D	E
1	sigla	cod_estado	nome		
2	RO	11	Rondônia		
3	AC	12	Acre		
4	AM	13	Amazonas		
5	RR	14	Roraima		
6	PA	15	Pará		
7	AP	16	Amapá		
8	TO	17	Tocantins		
9	MA	21	Maranhão		
10	PI	22	Piauí		
11	CE	23	Ceará		
12	RN	24	Rio Grande do Norte		
13	PB	25	Paraíba		
14	PE	26	Pernambuco		
15	AL	27	Alagoas		

emitido_em_2022-08-15 20_17_05

Planilha visualizada com o software *Microsoft Excel*.

Fonte: Elaborado pelo autor (2022).

Tabela 6.1 – Arquivos adicionados ou modificados para o segundo incremento

Arquivo	Finalidade
consultaPersonalizada.php	Arquivo novo adicionado ao sistema do tipo <i>action</i> , responsável por manipular as instâncias de classes para a consulta a ser executada, além de relacionar os elementos estruturais da tela com os dados, nas quais os mesmos sendo fruto dessa manipulação de instâncias necessária.
consultaPersonalizadaINT.php	Novo arquivo adicionado ao projeto na qual representa a classe que tem como principal finalidade montar os elementos estruturais da interface a ser apresentada.
consultaPersonalizadaRN.php	Arquivo novo que tem por objetivo ser o meio de comunicação entre a classe consultaPersonalizadaBD.php e o arquivo <i>action</i> consultaPersonalizada.php por meio de uma instância da classe BD que esse arquivo dispõe. Além disso, esse arquivo separa a classe do arquivo mencionados inclusive para isolar a consulta ao banco de dados da interface.
consultaPersonalizadaBD.php	Classe adicionada ao sistema responsável por realizar a consulta ao banco de dados do mesmo, se comunicando apenas com a classe consultaPersonalizadaRN.php .
controlador.php	O controlador foi alterado para que fosse necessária a validação do novo recurso adicionado, "consulta_personalizada", fazendo uma ligação com o novo arquivo <i>action</i> mencionado.
Pagina.php	Sendo a classe responsável por renderizar o menu lateral esquerdo de funcionalidades do sistema, foi adicionada a nova opção de "Consulta de tabelas" para o novo módulo criado "CONSULTAS NA BASE".

Fonte: Elaborado pelo autor (2022).

7 CONCLUSÃO

Dado o fato de que o SPT-Sars-Cov-2 é um sistema que ainda está em uso e que, muito provavelmente, ainda será utilizado a longo prazo, pode-se afirmar que o mesmo apresenta uma possibilidade consideravelmente alta de passar ainda por várias atualizações, e não somente limitar-se-á a atualização descrita no presente TCC.

Ao considerar-se os objetivos estipulados na atualização do sistema detalhada neste trabalho, pode-se assegurar que: (i) o sistema agora é capaz de filtrar de uma forma otimizada por meio de uma funcionalidade nova quais pacientes testaram positivo para os testes RT-PCR em um determinado intervalo de tempo; (ii) o usuário administrador, por meio também de uma nova funcionalidade, a partir de agora, consegue fornecer a outros usuários uma consulta personalizada ao banco de dados sem que os mesmos façam qualquer tipo de alteração na base, mantendo a integridade desses dados e fornecendo uma visão de como essa base está estruturada para futuras mudanças no sistema; (iii) a atualização incluiu funcionalidades que permitem uma filtragem personalizável sobre os dados resultantes; (iv) as novas funcionalidades apresentam alguns aspectos dinâmicos inspirados nas funcionalidades já presentes no sistema, por meio de permitir alguma forma de personalização do usuário, além de padrões estéticos e estruturais.

No que diz respeito ao desenvolvimento desta atualização, feito de maneira individual, a falta de conhecimento da principal linguagem do projeto, no caso PHP e ainda não ter interagido em um sistema baseado em sua maioria no modelo MVC com essa linguagem de programação foram as principais dificuldades enfrentadas pelo autor no início desse desenvolvimento.

Pensando no futuro do sistema, pode-se afirmar que o SPT-Sars-Cov-2 apresenta um grande potencial para outros tipos de testes futuros do ICBS. Além disso, em virtude da maneira como foi desenvolvida a sua arquitetura, o sistema permite que novas funcionalidades sejam incluídas sem comprometer as já presentes no mesmo. Isso está diretamente relacionado ao fato de que as funcionalidades são controladas por recursos, onde o controlador do sistema de certa forma modulariza cada uma delas isoladamente. Sendo assim, o SPT-Sars-Cov-2 poderia ser um sistema não apenas relacionado a testes da Covid-19, mas sim um sistema geral de testes.

REFERÊNCIAS

ALUR, D.; CRUPI, J.; MALKS, D. **Core J2EE Patterns: Best Practices and Design Strategies**. Prentice Hall PTR, 2003. (Core Series). ISBN 9780131422469. Disponível em: <<https://books.google.com.br/books?id=1dx34EMVyi8C>>.

Apache Friends. **O que é o XAMPP?** 2022. Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em 15 set. 2022.

BALTHAZAR, G. da R. **Visão Geral da Qualidade de Software**. 2007. Monografia (Bacharel em Sistemas de Informação), FMG (Faculdade Metodista Granbery), Juiz de Fora, Brasil.

BATHAGLINI, C. B. **SPTSars-Cov-2: Sistema de Procedimentos de Testes da Covid-19**. 2021. Monografia (Bacharel em Ciência da Computação), UFRGS (Universidade Federal do Rio Grande do Sul), Porto Alegre, Brasil.

Beatriz Moura e Matheus Henrique. **O que é o HTML e suas tags? Parte 1: estrutura básica**. 2021. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-html-suas-tags-parte-1-estrutura-basica>>. Acesso em 20 ago. 2022.

BOOTSTRAP. **Build fast, responsive sites with Bootstrap**. 2022. Disponível em: <<http://getbootstrap.com/>>. Acesso em 20 ago. 2022.

CARVALHO, T. **Orientação a Objetos: Aprenda seus conceitos e suas aplicabilidades de forma efetiva**. Casa do Código, 2016. ISBN 9788555192142. Disponível em: <<https://books.google.com.br/books?id=xzUKDQAAQBAJ>>.

CHART.JS. **Chart.js**. 2022. Disponível em <<https://www.chartjs.org/docs/latest/>>. Acesso em 20 ago. 2022.

DALL’OGLIO, P. **PHP Programando com Orientação a Objetos - 4ª edição**. Novatec Editora, 2016. ISBN 9788575226919. Disponível em: <<https://books.google.com.br/books?id=lfxmDwAAQBAJ>>.

Daniela Rocha Silva e Luis Felipe Bentin Sobral. **Um Estudo em Larga Escala sobre a Estrutura do Código-fonte de Pacotes JavaScript**. 2017. Monografia (Bacharel em Sistemas de Informação), UNIRIO (Universidade Federal do estado do Rio de Janeiro), Rio de Janeiro, Brasil.

Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides. **Padrões de Projetos: Soluções Reutilizáveis de Software Orientados a Objetos**. Bookman, 2006. ISBN 9788573076103. Disponível em: <<https://books.google.com.br/books?id=1evwwAEACAAJ>>.

GARRETT, J. **The Elements of User Experience: User-centered Design for the Web and Beyond**. New Riders, 2011. (Always Learning). ISBN 9780321683687. Disponível em: <<https://books.google.com.br/books?id=3W3SmAEACAAJ>>.

Governo Federal. **Primeiro caso de Covid-19 no Brasil permanece sendo o de 26 de fevereiro**. 2020. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/noticias/2020/julho/primeiro-caso-de-covid-19-no-brasil-permanece-sendo-o-de-26-de-fevereiro>>. Acesso em 27 jun. 2022.

Instituto Butantan. **Como surgiu o novo coronavírus?** 2022. Disponível em: <<https://butantan.gov.br/covid/butantan-tira-duvida/tira-duvida-noticias/como-surgiu-o-novo-coronavirus-conheca-as-teorias-mais-aceitas-sobre-sua-origem>>. Acesso em 27 jun. 2022.

Instituto de Ciências Básicas da Saúde. **UFRGS e Governo do Estado assinam termo de cooperação para realização de testes de coronavírus.** 2020. Disponível em: <<https://www.ufrgs.br/icbscovid/2020/04/09/ufrgs-e-governo-do-estado-assinam-termo-de-cooperacao-para-realizacao-de-testes-de-coronavirus>>. Acesso em 27 jun. 2022.

JQUERY. **What is jQuery?** 2022. Disponível em: <<https://jquery.com/>>. Acesso em 20 ago. 2022.

LOPES, B. **Classes de Domínio: Entendendo o modelo MVC.** 2015. Disponível em: <<https://vaidegrails.wordpress.com/2015/06/19/classes-de-dominio-entendendo-o-modulo-mvc/>>.

MariaDB Foundation. **MariaDB Server: The open source relational database.** 2022. Disponível em: <<https://mariadb.org/>>. Acesso em 20 ago. 2022.

PRESSMAN, R.; MAXIM, B. **Engenharia De Software: UMA ABORDAGEM PROFISSIONAL.** MCGRAW HILL - ARTMED, 2016. ISBN 9788580555332. Disponível em: <<https://books.google.com.br/books?id=smNFvgAACAAJ>>.

ROGERS, Y.; SHARP, H.; PREECE, J. **Design de Interação - 3ed.** Bookman Editora, 2013. ISBN 9788582600085. Disponível em: <https://books.google.com.br/books?id=d_s4AgAAQBAJ>.

Secretaria da Saúde do Estado do Rio Grande do Sul. **Painel Coronavírus RS.** 2022. Disponível em: <<https://ti.saude.rs.gov.br/covid19/>>. Acesso em 27 jun. 2022.

SOMMERVILLE, I. **Engenharia de software.** Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>.

The World Wide Web Consortium (W3C). **Cascading Style Sheets home page.** 2022. Disponível em: <<https://getbootstrap.com/>>. Acesso em 20 ago. 2022.