

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

Beatriz Harumi Fukumizu Oliveira	SP3014207
Bruna Rodrigues de Oliveira	SP3031853
Camila Lopez Franqueira de França	SP302427X
Fernando de Sousa Silva	SP3017061
Gabriel Pinheiro Brants Gonçalves	SP3013456
Lucas Ferreira do Nascimento	SP3015751

NewGen

São Paulo - SP - Brasil

2021

**IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo**

Beatriz Harumi Fukumizu Oliveira	SP3014207
Bruna Rodrigues de Oliveira	SP3031853
Camila Lopez Franqueira de França	SP302427X
Fernando de Sousa Silva	SP3017061
Gabriel Pinheiro Brants Gonçalves	SP3013456
Lucas Ferreira do Nascimento	SP3015751

NewGen

Documento da aplicação apresentado no curso
de Tecnologia em Análise e Desenvolvimento
de Sistemas do Instituto Federal de São Paulo,
da disciplina de PI1A5.

Professor: Ivan Francolin Martinez

Professor: José Braz de Araújo

IFSP - Instituto Federal de Educação, Ciência e Tecnologia
Câmpus São Paulo

Tecnologia em Análise e Desenvolvimento de Sistemas

PI1A5 - Projeto Integrado I

São Paulo - SP - Brasil

2021

Resumo

O uso de espaços diferenciados para trabalho tem crescido nos últimos anos, abandonando o conceito de escritórios fechados dedicados a apenas uma empresa. Entre as novas tendências deste movimento, surge o coworking, um espaço compartilhado que simula um escritório tradicional, com foco no *networking* entre profissionais. O objetivo deste projeto é criar um sistema que permita centralizar o gerenciamento de um coworking, integrando a gestão backoffice em um website *White Label SaaS*. Para o gerenciamento deste projeto, é utilizada a metodologia Scrum, aproveitando-se de métodos ágeis para as entregas parciais do sistema. O sistema de gerenciamento foi desenvolvido em ambiente web, utilizando-se a linguagem de programação Python e o *framework* bootstrap. Este trabalho apresenta o produto mínimo viável necessário para a execução de uma reserva e aluguel, que será complementado através das entregas futuras das outras funcionalidades desta aplicação.

Palavras-chaves: Sistema de Gestão. Coworking. Locação.

Abstract

The usage of alternative spaces for working has been growing in the last years, substituting the concept of whole offices dedicated exclusively to only one business. Among the new tendencies of this movement, we have the Coworking, a shared space that simulates a traditional office, with a focus on networking among professionals. The objective of this project is to create a system that allows the centralization of all projects in a coworking, linking the back office management in a **White Label SaaS** website. For the management of this project, the Scrum methodology is used, using the agile methods for the partial deliveries of the system. The management system was developed for websites, utilizing the programming language Python and the bootstrap framework. This work represents the minimum viable product needed to execute a sign up, a log-in, a reservation and a renting, which will be complemented through the future deliveries of other functionalities of this system.

Key-words: Management System. Coworking. Renting.

Lista de ilustrações

Figura 1 – Curva evolutiva de adoção de um produto	19
Figura 2 – Crescimento de empresas que disponibilizam espaços de coworking dentro do Brasil ao longo dos anos	19
Figura 3 – Distribuição das empresas de coworking pelos estados e cidades brasileiras no ano de 2018	20
Figura 4 – Rendimento das empresas clientes de coworking no Brasil em 2018 . . .	21
Figura 5 – Informações dos clientes de coworking no Brasil no ano de 2018 acerca da pergunta “Como considera a sua relação com os administradores do espaço?”	21
Figura 6 – <u>Informações dos Avaliação de serviços essenciais no coworking para os clientes de coworking no Brasil no ano de 2018</u>	23
Figura 7 – <u>Código Quick Response (QR) da aplicação</u>	39
Figura 8 – <u>Código QR do repositório Subversion (SVN)</u>	40
Figura 9 – <u>Código QR do repositório no GitHub</u>	40
Figura 10 – <u>Código QR do Blog da equipe</u>	40
Figura 11 – <u>Código QR do canal do Youtube</u>	41
Figura 12 – <u>Contribuição ao SVN em total de linhas</u>	42
Figura 13 – <u>Contribuição ao SVN mensalmente</u>	43
Figura 14 – <u>Dispersão de commits</u>	44
Figura 15 – <u>Quantidade de linhas de código</u>	45
Figura 16 – <u>Tipos de atividades dos autores</u>	45
Figura 17 – <u>Commits por horário</u>	46
Figura 18 – <u>Commits por dia da semana</u>	47
Figura 19 – <u>Modelo MTV no Django</u>	49
Figura 20 – Servidor para processo de requisições de usuário.	50
Figura 21 – Esquematização de fluxo para comunicação entre ambientes.	51
Figura 22 – <u>Exemplo de Identação em Código Simples</u>	53
Figura 23 – <u>Exemplo de Identação em Array</u>	54
Figura 24 – <u>Exemplo de Tamanho de Linhas adequado</u>	54
Figura 25 – <u>Exemplo de Imports Simples</u>	55
Figura 26 – <u>Exemplo de Imports Múltiplos</u>	55
Figura 27 – <u>Exemplo de Settings</u>	56
Figura 28 – <u>Exemplo de Tools</u>	56
Figura 29 – <u>Exemplo de Generic</u>	57
Figura 30 – <u>Exemplo de Elements</u>	57

Figura 31 – Exemplo de Object.	57
Figura 32 – Exemplo de Components.	58
Figura 33 – Exemplo de Trumps.	58
Figura 34 – Exemplo de um Design Pattern Adapter	59
Figura 35 – Exemplo de um Design Pattern Strategy	60
Figura 36 – Esquematização do Amazon Web Services CodePipeline	61
Figura 37 – Esquema de Sprint na Ferramenta Jira	62
Figura 38 – MER - Modelo Entidade-Relacionamento	98
Figura 39 – DER - Diagrama Entidade-Relacionamento	99

Lista de quadros

Quadro 1 – Comparação de Funcionalidades Entre Concorrentes	29
Quadro 2 – Escopo do projeto.	48
Quadro 3 – Relação de requisitos funcionais	67
Quadro 4 – Continuação da relação de requisitos funcionais	68
Quadro 5 – Relação de requisitos não funcionais	69
Quadro 6 – Listagem de regras de negócios	71
Quadro 7 – <u>Descrição dos dados da tabela 'tipoEspaco'</u>	100
Quadro 8 – <u>Descrição dos dados da tabela 'espaco'</u>	100
Quadro 9 – <u>Descrição dos dados da tabela 'equipamentos'</u>	101
Quadro 10 – <u>Descrição dos dados da tabela 'pagamento'</u>	101
Quadro 11 – <u>Descrição dos dados da tabela 'funcionario'</u>	102
Quadro 12 – <u>Descrição dos dados da tabela 'cliente'</u>	103
Quadro 13 – <u>Descrição dos dados da tabela 'pacoteHoras'</u>	103
Quadro 14 – <u>Descrição dos dados da tabela 'reserva'</u>	104
Quadro 15 – <u>Descrição dos dados da tabela 'equipamentoReserva'</u>	104
Quadro 16 – <u>Descrição dos dados da tabela 'convidados'</u>	105
Quadro 17 – <u>Descrição dos dados da tabela 'advertencia'</u>	105

Listas de abreviaturas e siglas

AWS	<i>Amazon Web Services</i> - Serviços Web da Amazon - Citado em 49 , 50
DER	Diagrama Entidade-Relacionamento - Citado em 52 , 78
HTTPS	<i>Hypertext Transfer Protocol</i> - Protocolo de Transferência de Hipertexto - Citado em 69 , 74
LGPD	Lei Geral de Proteção de Dados - Citado em 73
MER	Modelo Entidade-Relacionamento - Citado em 52 , 78
MVP	<i>Minimum Viable Product</i> - Mínimo Produto Viável - Citado em 27 , 47 , 48
POC	<i>Proof of Concept</i> - Prova de Conceito - Citado em 47 , 48
QR	<i>Quick Response</i> - Resposta Rápida - Citado em 4 , 39 , 40 , 41 , 68
SaaS	<i>Software as a Service</i> - Software como serviço - Citado em 11 , 17 , 26 , 75 , 78 , 143
SVN	Subversion - Citado em 4 , 39 , 40

Sumário

1	INTRODUÇÃO	11
1.1	Contextualização	11
1.2	Problematização	13
1.2.1	A necessidade da gestão correta	14
1.2.2	Alta demanda e potencial desperdiçado	14
1.3	Objetivos	16
1.4	Solução Proposta	16
1.5	Justificativa	18
1.5.1	Demanda	18
1.5.2	Visão social	21
1.5.3	Vantagens do Sistema NewGen	23
1.6	Análise de concorrência	27
1.6.1	Agendamento Programático	29
1.6.2	Ferramentas de Gerenciamento	29
1.6.3	Locação de Equipamentos	30
1.6.4	Funcionalidades Adiadas	30
2	REVISÃO DE LITERATURA	32
2.1	Coworking	32
2.1.1	Desenvolvimento do Coworking Moderno	32
2.1.2	Benefícios do coworking	33
2.1.3	Porque o coworking	35
2.1.3.1	Ponto de vista do empregador	35
2.1.3.2	Ponto de vista do trabalhador	37
3	DESENVOLVIMENTO DO PROJETO	39
3.1	Links do projeto	39
3.2	Metodologia de Gestão de Projetos	41
3.3	Organização da equipe	41
3.4	Sprints	42
3.5	Métricas do projeto	42
3.5.1	Estatísticas SVN	43
3.6	Definição de Entregas	47
3.7	Tecnologias utilizadas	47
3.8	Arquitetura do sistema	50

3.8.1	Configurações do Servidor EC2	50
3.8.2	Comunicação Entre Ambientes	50
3.8.3	Escalabilidade	51
3.9	Segurança da Informação	
	Ferramentas de testes	52
3.10	Modelagem do Banco de Dados	52
3.11	Coding Convention	52
3.11.1	Back end	52
3.11.2	Front End	56
3.12	Modelagem	57
3.12	Design Patterns	58
3.12.1	Modelagem do Banco de Dados	58
3.13	Integração continua	60
3.14	Product Backlog	61
3.14.1	Histórias de Usuário	61
3.15	Histórias de Usuário	62
3.15.0.1	Épicos	62
3.15.1	Épicos	62
3.15.1.1	Histórias Divididas	63
3.15.2	Histórias Divididas	63
3.16	Definições de Escopo	64
3.15.1	Casos de uso	65
3.16	Análise de requisitos	65
3.16.1	Requisitos Funcionais	65
3.16.2	Requisitos não-funcionais	
	Não Funcionais	68
3.17	Regras de Negócio	69
3.17.1	Definição de Entregas	70
4	VIABILIDADE E MANUTENÇÃO DA APLICAÇÃO	72
3.1	Viabilidade Financeira	72
3.2	Manutenibilidade	
	Segurança da aplicação	
	Informação	73
3.2.1	Coding Convention	73
3.2.0.1	Back end	74
3.3	Escolhas de Descartes	75
3.3.1	White-Label para SaaS	75
3.3.1.1	Front End	76
3.3.2	Alteração da lógica do banco	77
3.3.2.1	Uso do ORM	77
3.3.3	Design Patterns	77
3.3.2.1	Adição de novas colunas	78

3.3.2.2	Estratégia de modelagem	78
3.4	Ferramentas de testes	79
4	CONSIDERAÇÕES FINAIS	80
4.0.1	Integração continua	80
4.1	Tecnologia utilizadas	80
	REFERÊNCIAS	83
	GLOSSÁRIO	85
	APÊNDICES	87
	APÊNDICE A – RELATÓRIOS DAS SPRINTS	88
	APÊNDICE B – MER E DER	97
	APÊNDICE C – DICIONÁRIO DE DADOS	100
	APÊNDICE D – PUBLICAÇÕES DO BLOG	106
	APÊNDICE E – ATAS DE REUNIÃO	113
	APÊNDICE F – ESTUDO INFORMAL SOBRE MIGRAÇÃO DO BANCO	143

1 Introdução

Este ~~documento~~ é um desenho o documento do projeto da aplicação web desenvolvida pela equipe do NewGen com foco em ajudar na administração de empresas de CoWorking. Desenvolvido para a disciplina de Projeto Integrado sob supervisão dos orientadores Ivan Francolin Martinez e José Braz de Araujo, o sistema tem a intenção de facilitar aluguéis, gestão de estoques, criar ferramentas de promoção de vendas, capturar o feedback dos clientes finais do coworking, entre outras funcionalidades como será abordado em outros tópicos mais detalhadamente.

~~Um desenho do projeto tem o objetivo de especificar as necessidades – sejam desde as demandas de mercado para criação da ferramenta, até os requisitos para que o cliente/usuário da ferramenta tenha desfruto com a aplicação, resolvendo problemas reais seus. Além de também ser tratado o detalhamento dos processos que serão rotineiros na vida dos usuários~~ O objetivo da equipe é construir uma aplicação, as metodologias utilizadas para a concretização da ferramenta por parte da equipe, o planejamento das entregas do sistema e alguns outros critérios relevantes ao funcionamento da aplicação web, um Software as a Service (SaaS) que auxilie nosso público alvo (administradores de coworkings) a gerenciar seu ambiente. Em outras palavras, traremos facilitações de agendamento, gerenciamento de estoques, controle de entrada, notificações de reuniões, entre outros. Essas funcionalidades estarão inseridas em uma plataforma web, adquirida após a efetivação do pagamento, através de um sistema de planos. Ou seja, o usuário poderá optar aquele que englobe todas suas necessidades e mais se enquadre em sua realidade. Dentre as opções disponíveis, todas se baseiam na adesão de uma plataforma para o administrador do tal coworking com opção de entrada para clientes, onde a interface muda de acordo com as atividades que este poderá se submeter.

1.1 Contextualização

Com o avanço do poder dos aparatos tecnológicos, o mundo se viu passando por grandes mudanças. Hoje em dia, um smartphone tem muito mais poder computacional que o computador que levou o homem à Lua, por exemplo. Neste sentido, o modo de se trabalhar não foi uma exceção: o século XXI está sendo marcado por uma expansão de modelos de trabalho mais flexíveis, como home-offices, business centers e o coworking, que é tema central da aplicação abordada neste documento.

No âmbito da flexibilidade digital, o coworking surge como um novo método de trabalho que permite tanto compartilhamento de ideias, como uma facilitação na gestão

dos preços. A fim de mensurar o quanto o coworking tem crescido dentro do ambiente de trabalho, de acordo dados de uma revista online de coworking, a DESKMAG (2018), o setor cresceu de 160 estabelecimentos de coworking em 2008, registrados ao redor do mundo, para mais de 19000 no final de 2018, o que é impressionante se levar em conta que a roupagem do coworking moderno surgiu em 2005.

Hoje em dia, o modelo de coworking é visto como um potencial revitalizador de regiões urbanas que estão estagnadas ou em crise econômica, como também um criador de oportunidades para cidades pequenas que desejam crescer (LUO, 2020).

De acordo com Luo (2020), o coworking também é altamente lucrativo para as empresas que oferecem espaços de locação para as outras, além de ser benéfico tanto para empresas ou trabalhadores que decidem adotar um modelo visto como modelo do futuro por diversas razões.

“Em 2020, o mundo foi impactado de todas as formas pela pandemia de Covid-19, causada pelo novo coronavírus. O isolamento social e o exercício do home office causaram mudanças nas estruturas logísticas de muitas empresas. Com a retomada das atividades, o modelo de coworking mostra-se uma excelente opção para profissionais informais, freelancers e empresas, barateando custos e oferecendo facilidades.” (DELTA BC, 2021).

Segundo o COWORKING BRASIL (2021b), o coworking está redefinindo a forma como nós trabalhamos e vivemos. O movimento do open source alinhado com a natureza transformadora das áreas de tecnologia está construindo um futuro sustentável através de um novo equilíbrio entre vida e trabalho.

~~O movimento~~ A organização COWORKING BRASIL (2021a) fala em “coopetição”, um neologismo derivado da mistura da palavra “competição” com “cooperação”, para traçar um paralelo de como os trabalhadores, inspirados pela cultura participativa e de co-criação, junto com um grupo de pessoas e pequenas empresas conectadas, estão remodelando o paradigma do mercado focando em criatividade e inovação das comunidades locais e ao redor do mundo. Desse modo juntando as comunidades para construir uma economia mais humana, interligada e sustentável.

O manifesto Coworking - termo difundido pelo COWORKING BRASIL (2021a) - preza por: colaboração acima da competição; coletivo acima da individualidade; participação acima da observação; “fazer” acima de “dizer”; camaradagem acima da indiferença; ousadia acima da garantia; aprendizagem acima da expertise; orgânico acima de mecânico; gentileza acima de desconfiança.

~~Ainda citando a fonte, foi feito um censo para medir a evolução do coworking no Brasil (COWORKING BRASIL, 2019a). Abaixo segue alguns gráficos imagéticos, retirados~~

~~deste censo, acerca da cultura do coworking brasileira que ilustram de forma clara e intuitiva baseada em dados reais que mostram o paradigma brasileiro do Coworking.~~

A DESKMAG (2013) aponta o Brasil como uma das economias de crescimento mais rápido no cenário internacional e que o coworking brasileiro segue esta tendência de crescimento, apresentando uma rede complexa de ambientes trabalho dentro um mercado dinâmico.

Além disso, fica claro o papel significante do coworking dentro do segmento da economia brasileira e considerando que este ainda é um mercado que possui grande potencial de expansão, por sua roupagem relativamente nova, é impossível distanciar o coworking como um contribuidor para a revitalização da economia.

“Em tempos de incerteza, as pessoas anseiam por pertencer. Querem juntar coisas que os tiram de casa e da insegurança. Tentamos integrar as melhores partes de um centro comunitário em um espaço de trabalho profissional. Conforme nos aproximamos do novo ano, estamos sendo solicitados a ajudar outras organizações comunitárias a replicar este modelo.

O que temos é um trabalho em andamento, quem sabe se os coworkers pós-Covid vão querer algo diferente e voltar para um espaço de trabalho tradicional e conservador? Meu palpite é que eles não vão!” (Carrick-Davies, 2020 ou 2021 apud COWORKING BRASIL, 2021a) (fonte primária não encontrada).

Em suma, o mercado do coworking está em constante expansão e levará alguns anos ainda para que ele atinja seu potencial máximo, também no mercado internacional, mas principalmente no Brasil. Outras análises e benefícios do coworking são abordados em [Coworking 2.1](#) [Coworking](#) neste documento.

1.2 Problematização

Fica notório o crescimento e o potencial do Coworking como uma ferramenta não só de trabalho, mas como também uma ferramenta que possibilita o crescimento econômico do país, além de construir uma cultura saudável de cooperação e comunicação de conhecimento, podendo também expandir, nesse sentido, o paradigma [eduaciona](#) [brasileiro.](#)

[empresarial brasileiro.](#) No entanto, [aparecem alguns entraves à tal crescimento:](#) [existem alguns problemas a serem analisados.](#)

1.2.1 A necessidade da gestão correta

É necessário cautela quanto ao crescimento do coworking no Brasil: há quem fale em uma possível criação de bolha dentro deste mercado ([GANDINI, 2015](#)). Cabe lembrar que “bolhas” é um termo cunhado dentro da teoria econômica para se referir à expansão anormal de mercados que não conferem valor a si próprios, ou seja, elas expandem de forma anômala em relação ao seu real valor e uma vez que algum mecanismo autorregulador mercadológico agir, é possível que haja o “estouro” dessa bolha, fazendo ela retornar ao seu real valor ou até menor, desferindo grande prejuízo financeiro aos investidores do setor.

Para [Wan \(2018\)](#), a fim de impedir uma bolha, é necessário distinguir os investidores especulativos dos investidores fundamentais, conseguindo pôr em pauta a "demanda real" com base nos fundamentos do próprio mercado. Tal especulação acontece quando existe dinheiro mascarando um valor fictício. Segundo o [COWORKING BRASIL \(2017\)](#), existem vários estabelecimentos que se vendem como coworking apenas pelo marketing, mas que cumprem com os requisitos para atender os benefícios supostos de um estabelecimento de coworking – tal qual como o networking. É nesse sentido que uma bolha do coworking pode surgir; sem o suporte adequado, os clientes do coworking não vão ver vantagem em optar por um estabelecimento desse tipo.

[Costa \(2018\)](#) conta o caso da startup de coworking WeWork que foi um fiasco de abertura, mesmo com uma avaliação inicial de 47 bilhões de dólares. Segundo o autor: “Crescer em escala sem deixar de lado a preocupação com a comunidade do espaço é um desafio e tanto. [...] se o coworking tem alguma escala de negócio, é no suporte à rede, [...] O espaço é só uma desculpa para começar a compartilhar ideias e fomento à inovação”. O problema apontado, em suma, é em como fazer coworking. Urge um parâmetro ou ferramenta que auxilie os investidores desse ramo a guiar o seu negócio da maneira correta. Além disso, saber criar valor no seu próprio negócio é algo que interessa não somente os empresários, como também todos os beneficiados com o nicho criado pelo coworking, pois gerir adequadamente impede especulações e perdas espantosas de capital, atendendo a demanda real.

1.2.2 Alta demanda e potencial desperdiçado

Existe uma alta demanda por estabelecimentos de coworking: segundo dados do [COWORKING BRASIL \(2019b\)](#), 42% dos 578 entrevistados acham que há poucos espaços de coworking na sua cidade.

Talvez em uma primeira observação esses números não sejam tão impressionantes, porém, é válido reforçar, com dados da mesma pesquisa, que uma grande parcela dos entrevistados faz parte de centros urbanos que concentram o maior número de coworkings no Brasil. Sendo a cidade de São Paulo a principal: concentrando 273 dos negócios, quase 24%

de todos os coworkings do Brasil e, somando o top 10 cidades com mais estabelecimentos, esta porcentagem chega a 56% ([COWORKING BRASIL, 2019a](#)). É natural então supor que a maioria que respondeu “não” é morador de algum centro urbano. Dessa forma, a demanda por novos locatários se faz ainda mais notória em cidades de médio e grande porte que não chegam ao nível das colossais capitais paulista e carioca.

Ademais, diferente do que se espera a partir dos dados, os coworkings não são espaços que só se integram bem em grandes centros urbanos, com ambientes permeados com um corporativismo consolidado. É justamente o oposto, assim como é possível aferir com dados do [COWORKING BRASIL \(2019a\)](#), que aponta 32% dos coworkings sendo de bairros não tradicionais; mostra que 82% das empresas contam com até 10 funcionários nos seus recursos humanos; e aponta que 39% das empresas clientes entrevistas ainda estão dando o pontapé inicial no mercado.

O coworking então é perfeito para micro ou pequenas empresas, que compõem 53,4% do produto interno bruto do país ([SEBRAE, 2011](#)).

Com isso tudo dito, é possível afirmar que o Brasil ainda tem muita área de crescimento do coworking dentro de seu território. Contudo, ainda existe um outro possível problema a ser solucionado dentro do contexto do coworking brasileiro: existe um perfil de investidores que dispõem de um conjunto de imóveis de aspecto comercial e preferem investir no mercado de aluguel comercial tradicional.

Levando em conta o preço médio em reais por metro quadrado para locação de imóveis comerciais no Brasil de 37,42 R\$/m² em maio de 2021 ([FIZEZAP, 2021](#)), investir no mercado tradicional aspira um retorno do valor médio de 5,43% do valor do imóvel a.a. (ao ano).

Todavia, dependendo do apetite ao risco do investidor, pode ser mais lucrativo entrar no crescente setor do aluguel de coworking, como um locador de espaços, levando em conta que em um aluguel de coworking se ganha também com a permanência do cliente no local – sendo possível também comercializar diversos produtos dentro do espaço também. O receio acontece por desconhecimento das oportunidades deste mercado, ou da falta do conhecimento necessário para fazer uma gestão adequada que possa trazer lucros reais para seu investimento, considerando um investimento de alto risco e outras implicações, como visto anteriormente.

O mercado de coworking por ser voltado a um público com sede de mudanças, não exige escritórios tradicionais caros, além disso, é possível conseguir vendas rápidas, dinamizando os pacotes fornecidos de locação, alugando uma mesa por exemplo, em vez de uma sala.

Em termos gerais, o coworking possui uma escalabilidade maior que os aluguéis tradicionais. Com a locação bem-sucedida dos seus imóveis, uma boa parcela de investidores

poderia facilmente expandir para novos imóveis, estes não precisando de grande capital para o investimento inicial. Somado a isso, 73% dos clientes finais da locação do coworking mantém alguma interação benéfica com o dono ([COWORKING BRASIL, 2019b](#)), e mesmo que só 27% que tenha confirmado isso, em consideração de potenciais parceiros é uma boa parcela. Nesse sentido, a prosperidade dos clientes do negócio reflete também nos lucros da empresa mãe.

1.3 Objetivos

O objetivo principal da equipe NewGen é desenvolver um software que ajude a administrar os locais de coworking, trazendo um público que tem vontade de fazer um investimento mais ousado no setor imobiliário, mas não sabe por onde começar. Com um sistema pronto que atenda às necessidades de gestão, esse grupo de pessoas poderia analisar concretamente suas oportunidades. Eventualmente uma parcela que se afugentava desse setor, seja por desconhecimento de oportunidades, ou por medo de tomar riscos em investimentos com grande abstração, sem estratégias concretas de gestão, entraria no mercado em uma relação de ganha-ganha com a NewGen.

Cabe destacar que este é um setor recente que ainda não tem nem duas décadas de existência, existe um grande nicho a ser explorado e ainda há um certo ar de pioneirismo no coworking. Neste engatinhar de um novo mercado, há uma grande riqueza de oportunidades e geração de valor.

Por este ângulo, o objetivo secundário da equipe do NewGen é o de contribuir com o crescimento saudável de um setor que abre portas para a revitalização de espaços decadentes ([LUO, 2020](#)), auxilia na expansão de economias emergentes - como as de cidades de pequeno ou médio porte - e ajuda a distribuir valor em comunidades carentes (32% dos coworkings são de bairros não tradicionais ([COWORKING BRASIL, 2019a](#))).

Em outras palavras, a equipe da NewGen, por meio da sua aplicação, pretende ser uma peça fundamental na jornada dos investidores que ainda estão começando no mercado do coworking, maximizando suas chances de sucesso. Além de tudo, ao fazer sua parte dentro da comunidade do Coworking, a NewGen ajuda a criar um modelo social de trabalho mais humanizado e centrado nas pessoas.

1.4 Solução Proposta

O sistema NewGen é uma aplicação web flexível, assim como o modelo do coworking, que possa ser vendida como um produto via assinatura para uma empresa que pretenda ingressar ou já está na área de locação de espaços para coworking. A aplicação já vai

apresentar soluções reais prontas e poderá ser customizada pelo administrador do sistema de modo a atender às regras de negócio de sua empresa.

A estratégia desse modelo de software aplicação é chamada de White-label ou private-label. White-label de uma tradução livre do inglês significa rótulo branco, ou seja, um produto sem marca já pronto que é vendido para uma empresa colocar sua marca eomercial por cima e revender como sua. modelada para ser um Software as a Service. SaaS descreve uma forma de distribuição de um software como um serviço de assinatura com preços variados. Exemplos de SaaS famosos são o Microsoft Office 365, o Google Docs, o dropbox, o Evernote, etc.

Ao realizar a compra, os custos físicos do sistema (manutenção e eletricidade) ficarão por conta do cliente. A ideia por trás deste tipo de comercialização é que o usuário pode assinar e parar de usar quando quiser. A vantagem é que há a flexibilidade de parar de pagar quando quiser. Há também esperado que o cliente tenha uma infraestrutura mínima própria para rodar a aplicação a vantagem que o cliente pode enxugar os prejuízos caso não goste do produto, pois diferente de uma compra única, ou um contrato com uma empresa que o faça. No futuro, possa ser que haja um estudo de parcerias da NewGen com empresas que fornecam esses serviços só cancelar a assinatura.

Apesar de haver essas responsabilidades prévias por parte do cliente, uma vez que o sistema se tornou propriedade dele, ele poderá hospedar da forma que quiser e fazer as modificações que bem desejar. Além disso, a disponibilidade, segurança e manutenção do sistema fica a cargo do prestador de serviços SaaS. Sendo possível o cliente abrir mão de custos operacionais de infraestrutura e de principalmente de aquisição, já que ele não terá que comprar um servidor próprio para sua aplicação, basta que ele assine o serviço.

Caso haja cancelamento da assinatura, o NewGen se compromete a passar para o cliente os dados do banco de dados do seu coworking, sob pagamento de uma taxa, ou não dependendo do seu plano de assinatura. Dessa forma o cliente não precisa ficar preso à nossa plataforma se for este seu interesse. O pagamento da taxa se refere aos custos de exportação do banco de dados.

Como o sistema será modelado como um web-service, a aplicação da NewGen teria alguns direitos reservados, por exemplo Dependendo do serviço de assinatura, o cliente terá um domínio e um banco de dados próprio, mas esta funcionalidade ainda está sendo estudada pela equipe do NewGen, podendo ser implementada somente em uma terceira fase do projeto.

Voltando ao cancelamento da assinatura, o NewGen não seria permitido se compromete a abrir mão de algumas propriedades intelectuais. É possível que haja exportação de alguns arquivos relativos ao front-end para o cliente, mas ao cancelar o serviço do NewGen, é previsto que o cliente se apossasse da ferramenta pensando em revender dentro do nicho

~~de aplicações de White-label para coworking. tenha um outro sistema para realizar a transição, então os arquivos do back-end (com extensão .py) não serão exportados e são direitos reservados da NewGen. Porém, ao que se refere a dados e personalização do cliente, são negociáveis.~~

A proposta é que uma vez que o cliente tenha contratado a aplicação da NewGen, ele não precise se preocupar com erros ou atualizações, infraestrutura e manutenção do sistema que constariam dentro do seu pacote de assinatura completa.

A criação da instância do sistema do cliente dentro do sistema da aplicação da equipe NewGen é feita via online e em alguns planos conta suporte para personalização mais avançada, como alterações específicas de escolha do proprietário do coworking. Com o serviço de assinatura diversificado, o comprador escolhe o pacote mais flexível que se adapta à sua necessidade, corta custos de risco com o desenvolvimento de um sistema que possa não ser de seu agrado - considerando que a NewGen oferece plataformas já montadas - e ganha um sistema do zero. O público alvo da NewGen são justamente os proprietários que não têm ou estão insatisfeitos com seus sistemas de gerenciamento e que precisam de soluções inteligentes sob medida.

Em outros termos, com esta proposta, a equipe NewGen pretende lucrar contribuindo para o aumento do mercado de coworking, realizando vendas justas para clientes que querem novidades no investimento imobiliário em uma relação benéfica que ambas as partes consigam retirar uma margem interessante de lucro.

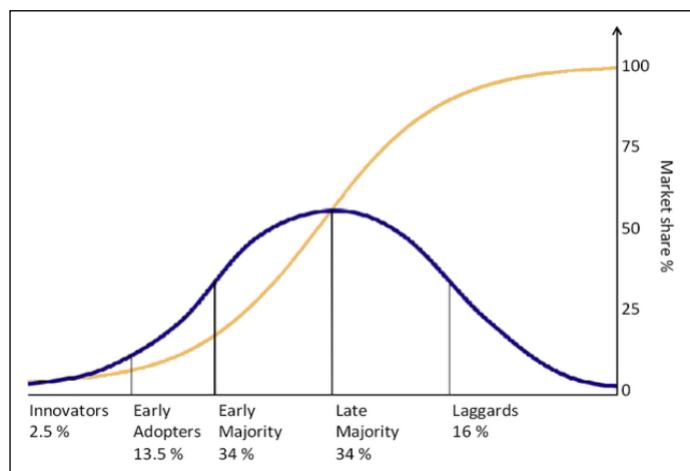
1.5 Justificativa

Para justificar o software planejado pela equipe NewGen, se baseia em 3 principais pontos: Demanda de mercado, melhoria socioeconômica e falta de opções que atendem a proposta exata sugerida pela NewGen.

1.5.1 Demanda

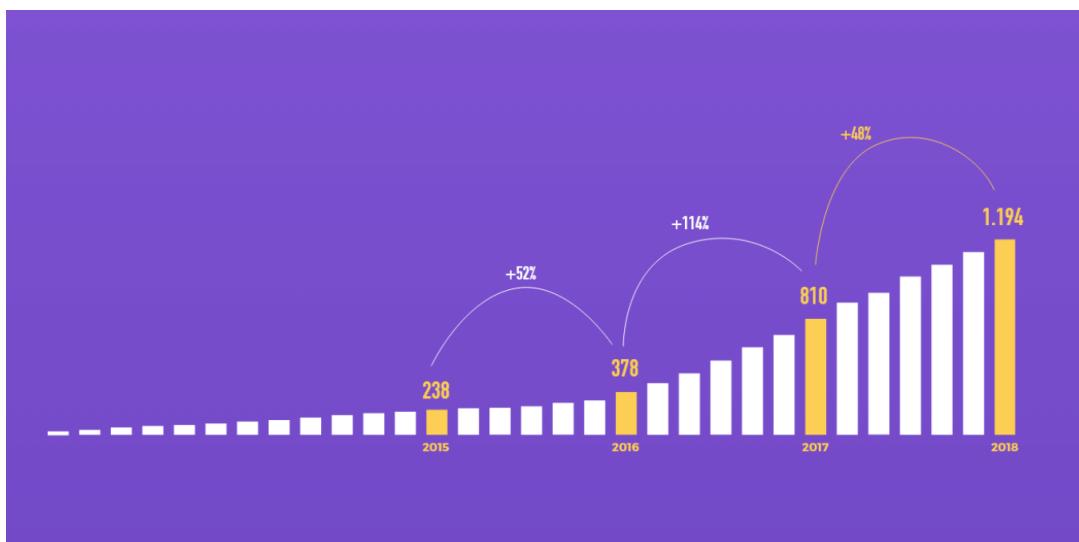
~~Como dito anteriormente, o~~ O coworking é um mercado em expansão e por se tratar de um modelo relativamente recente, ainda tem uma boa taxa de crescimento antes que se torne um modelo defasado. Traçando um paralelo com a famosa curva difusão de um produto (Figura 1), dividida em: Inovadores, adeptos iniciais, maioria inicial, maioria tardia e retardatários; dá para dizer que o setor do coworking está no ponto mais interessante de crescimento: a dos adeptos iniciais (WATERS-LYNCH, 2018).

Figura 1 – Curva evolutiva de adoção de um produto



Fonte: ([DUBE; GUMBO, 2017](#))

Figura 2 – Crescimento de empresas que disponibilizam espaços de coworking dentro do Brasil ao longo dos anos



Fonte: Censo Coworking do [COWORKING BRASIL](#) (2019a)

Segundo a [DESKMAG](#) (2013) o primeiro coworking do Brasil apareceu em 2008 na cidade de São Paulo e, como é possível ver na [Figura 2](#), e desde de então vem sofrendo um crescimento expressivo dentro do território brasileiro, principalmente na segunda metade da década de 2010.

É nos Nos estágios iniciais que a de um mercado crescente, muitos investidores conseguem tirar seus lucros boas oportunidades de margem de lucro. Como o caso descrito na [DESKMAG](#) (2013), do Pto de Contato, que foi pioneiro no mercado nacional e ainda em 2013, 5 anos depois de abrir o negócio, já contavam com 2 pontos de coworking

Figura 3 – Distribuição das empresas de coworking pelos estados e cidades brasileiras no ano de 2018



Fonte: Censo Coworking do COWORKING BRASIL (2019a)

somando mais de 50 estações de trabalho. Nesta perspectiva, o coworking desponta como a alternativa ideal para quem quer assumir um pouco mais de risco no mercado imobiliário com boa chance de tirar um bom proveito da fatia lucrativa que o crescimento inicial oferece.

A Figura 3 mostra que o Coworking no Brasil ainda está muito concentrado nas grandes metrópoles, principalmente São Paulo, que ostenta 24% dos estabelecimentos em relação ao país todo. Ainda sim, levando em conta o porte da capital paulista, ainda existe uma boa demanda de crescimento e não é de surpreender que este número venha a crescer ainda mais.

Porém, o que os dados da Figura 3 também mostram é que a demanda por coworkings fora de São Paulo está ainda muito alta e que há muito mercado a ser explorado. Portanto, o público alvo do sistema da NewGen podem abocanhar ampla fração do mercado brasileiro enquanto ele ainda está nas suas fases iniciais.

Os dados do COWORKING BRASIL (2019a) juntamente dão uma dimensão das quantidades envolvidas dentro do empório brasileiro: 127 milhões de reais movimentados, 214 mil pessoas distintas circulando, 88 mil estações de trabalho disponíveis (estaçõa se referindo a uma unidade mínima ofertada para um cliente trabalhar) e a criação de 7 mil empregos diretos. Tais números demonstram que o coworking brasileiro já saiu da fase de risco inicial de "morrer na praia", mas ao mesmo tempo tem um grande potencial de expansão. Em essência: algo já foi consolidado, porém, há grande espaço para mais gente no setor.

Adentrando nas capitais do interior

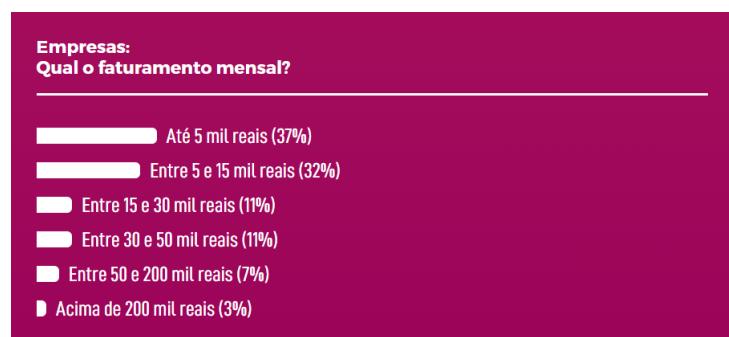
1.5.2 Visão social

O coworking surge como planta em um ecossistema: ele é um ator importante para que empresas que estão iniciando no mercado, e.g. start-ups, possam se cortar gastos dispendiosos e em um mutualismo peculiar, essas empresas dão a vida ao coworking enquanto o coworking dá a vida à elas. Essa relação é descrita por Moriset (2013) e mostra como que a década de 2000, contextualizada por crises como a do atentado às torres gêmeas ou a de 2008, o coworking surge com a intenção de ser um modelo flexível e moderno. Um modelo que, por meio dos custos adaptados a cada situação, permitiu e permite que muitas empresas que estão apenas iniciando possam dar continuidade a seus negócios.

Segundo dados do COWORKING BRASIL (2019b) em termos de maturidade do negócio: 39% das empresas que usufruem do coworking estão iniciando, 30% passaram da fase inicial e estão indo bem, 19% estão maduros e estáveis; em termos de tamanho da empresa, 36% é composta de somente uma pessoa, 34% de até 5 pessoas e 12% 5 e 10 pessoas; no perfil da empresa cliente: 43% são proprietários de empresa, 31% são profissionais independentes e 26% são funcionários de uma empresa. Logo, a relação de empresas de menor porte com empresas disponibilizadoras dos espaços é perceptível.

A Figura 4 demonstra como uma expressiva fração do negócios clientes do coworking ainda são de micro ou pequeno porte.

Figura 4 – Rendimento das empresas clientes de coworking no Brasil em 2018



Fonte: Censo Coworkers do COWORKING BRASIL (2019b)

Figura 5 – Informações dos clientes de coworking no Brasil no ano de 2018 acerca da pergunta “Como considera a sua relação com os administradores do espaço?”



Fonte: Censo Coworkers do COWORKING BRASIL (2019b)

Além disso, na [Figura 5](#) com 27% considerando o proprietário do estabelecimento um amigo, é apontado um outro acontecimento detalhado por [COWORKING BRASIL \(2019b\)](#): o ambiente do coworking muitas vezes favorece parcerias cliente-locatário, de forma que um emaranhado de associações possa acabar surgindo dentro do próprio coworking, quase como uma economia interna ao próprio ambiente. Incontestavelmente este é um caso muito lucrativo para o gestor do negócio e, fora a parceria formalmente firmada, existe o networking intercorporativo. O ato de uma companhia dar uma recomendação da sua parceira por exemplo. Tudo isso conquistado sob o senso comunitário do contexto. Contata-se que o coworking e as micro e pequenas empresas são interdependentes e que o coworking, mesmo que possa atender as necessidades de uma grande empresa, se encaixa nos moldes e necessidades de uma novata.

Sabendo disso, com outros dados [COWORKING BRASIL \(2019a\)](#), é visto que existe grande demanda para o crescimento de cidades no interior. 55% dos estabelecimentos se encontram nas capitais dos estados e 51% fica em municípios de mais de 1 milhão de habitantes. 30% sobra para cidades de médio porte (100 mil a 1 milhão de habitantes). É inegável que o coworking traga um ar de metrópole moderna criativa como conta [Gandini \(2015\)](#), porém, o coworking sobretudo se encaixa em economias emergentes (tal qual cidades de médio porte), por fazer a conexão de pequenas empresas ou recém criadas, portanto é inegável também que há uma carência de estabelecimentos nas cidades interior.

Mais dados do [COWORKING BRASIL \(2019b\)](#) mostram que 32% dos estabelecimentos de coworking estão em bairros de perfil não tradicional, demonstrando uma integração do setor com comunidades mais carentes e mostrando acima de tudo como a versatilidade do modelo traz oportunidades para todas as áreas da sociedade.

Somado a isso, 78% dos clientes consideram o preço justo e 72% apontaram uma melhora do networking pessoal, corroborando com a hipótese do próprio [COWORKING BRASIL \(2021a\)](#) da comunidade humanizada e inovadora dos ambientes de coworking.

Para finalizar, cabe uma análise do perfil do coworker (cliente final coworking). Como a [Figura 6](#) mostra, qualidade da internet e salas de reuniões são bem importantes, porém, também no mesmo nível de importância os clientes consideram espaço de convivência, copa/cozinha (para interagir com colegas do espaço) e eventos organizados no local; levantando a importância de disponibilizar um espaço que se enquadre nas exigências para ser um espaço proveitoso aos clientes e que possa render lucros de fato ao proprietário.

Cabe lembrar que a idade média do coworker é de 33 anos e 36% preferem mesas rotativas, 29% mesas fixas e 28% salas privadas ([COWORKING BRASIL, 2019b](#)). Significando esses dados, é válido concluir que um cliente médio de coworking é um perfil jovem que não necessariamente vai se importar com a estação de trabalho, visto que as porcentagens estão razoavelmente pareadas umas com as outras, mas sim na maneira e

Figura 6 – ~~Informações dos~~ Avaliação de serviços essenciais no coworking para os clientes de coworking no Brasil no ano de 2018



Fonte: Censo Coworkers do COWORKING BRASIL (2019b)

criatividade que o proprietário vai conseguir trazer um ambiente produtivo e comunitário dentro do seu espaço. O importante é fazer o cliente sentir que aquele local apropriado para oportunidades.

1.5.3 Vantagens do Sistema NewGen

O sistema NewGen se diferencia dos outros, sendo uma ferramenta com um pacote completo de utilidades essenciais para a administração de um negócio de coworking. De alguma forma, a aplicação que a equipe NewGen busca trazer com suas características e peculiaridades é única e não existia uma com uma proposta igual, dando sentido no desenvolvimento do software proposto. É elencado abaixo 5 pontos que apontam suas vantagens:

- **A aplicação promove uma gestão digitalizada que incentiva a prática do coworking.**

Um dos diferenciais da NewGen é trazer dentro de um sistema de aluguel voltado a coworking uma gama de ferramentas de auxílio de gestão. Dessa forma, o cliente interessado a entrar na área do coworking que ~~comprar assinar~~ o sistema web não precisa ter grande conhecimento prévio de administração do modelo coworking. Pois dentro da ferramenta, com algumas funções básicas previamente ajustadas, o

cliente consegue ter um workflow básico. Conforme ele for necessitando de novas especificações, ele pode configurar o sistema para ir se ajustando às suas preferências. Em uma análise de concorrência, não foi possível detectar se os sistemas concorrentes de fato tinham ferramentas avançadas de gestão que não apenas tratassem o aluguel ([vide 1.6 Análise Refinada de Concorrência](#)[1.6 Análise de concorrência](#)). Mas o que foi identificado é que o agendamento (tanto reserva como aluguel) não era 100% digitalizado, era necessário entrar em contato com um responsável da empresa, mesmo que fosse na primeira vez, para dar prosseguimento no agendamento de um espaço.

Algumas das funcionalidades que entram nesse tópico são: Relatórios periódicos com feedback do cliente, Sistema de Fidelidade que auxiliam o administrador saber quem é seu cliente mais valioso e o manter no negócio, Controle de estoque que automatiza a gestão dos recursos materiais e energéticos de acordo às demandas do espaço de coworking e um serviço de locação de equipamentos, que fornece ao cliente final (o cliente do nosso cliente) uma flexibilidade extra de alocação de novos recursos de maneira automática.

Isso tudo contando que haverá a possibilidade do agendamento 100% digital via web com a validação presencial via QR Code, se o cliente assim desejar.

- **Utilização de recursos de maneira a promover o networking, a alma do coworking.**

Como visto na [Problematização 1.2](#)[1.2 Problematização](#), nem todo espaço que se vende como coworking é necessariamente um espaço de coworking. Porque espaços de coworking tem o seu diferencial acima dos outros espaços centrados no networking profissional.

É verdade que o sistema NewGen ([Solução Proposta 1.4](#)[1.4 Solução Proposta](#)) traga customização livre para o cliente do sistema usar como bem entender, assim sendo possível a utilização do sistema com uma abordagem que foge um pouco da proposta pelo modelo.

No entanto, o sistema por trazer [já na primeira assinatura](#) as funcionalidades básicas modeladas para um gerenciamento padrão de coworking, guiam um potencial cliente que poderia fugir da proposta, o ajudando a promover integração social dentro do seu negócio. Segue algumas vantagens trazidas pelas funcionalidades do NewGen que podem coordenar o negócio do cliente a ser um ambiente frutífero tanto para ele quanto para seu próprio cliente:

1. *Confiança e senso de pertencimento ao cliente:*

Reserva e Lista de Espera: o primeiro dá a confiança ao cliente que ele vai pagar e o segundo confere senso de importância do cliente com o sistema de

coworking, pois o próprio sistema o avisa perguntando sobre se ele quer alugar a vaga que ele tentou uma vez alugar uma primeira vez, mas sem sucesso.

2. *Importância da voz do cliente no negócio;*

Relatórios periódicos que conferem a satisfação do cliente dentro do espaço de trabalho, podendo também ouvir sugestões de melhoria do atendimento.

3. *Grande liberdade na hora de fazer o pedido;*

Com funções como Locação de Máquinas ou Transferência de responsabilidade, o cliente final do coworking possui grande flexibilidade na hora de com suas reservas, permitindo que ele faça algumas alterações futuras pertinentes, mesmo com o pedido já pago.

4. *Recompensas ao cliente fiel, fomentando possíveis parcerias entre cliente e proprietário;*

O sistema de fidelidade propõe ser um processo que recomende que um cliente fiel de valor para empresa receba recompensas por parte do locatário, tais recompensas podem ser fornecidas automaticamente também.

- **Fornece à empresa um sistema interno integrado fácil de ser encontrado.**

Com sistema integrado subtende a integração de um sistema de divulgação online, cujo site é usado para mostrar o espaço de Coworking do cliente, além de criar presença no ambiente virtual a fim de ser encontrado por um maior público; com um sistema de compras, que carrega a responsabilidade de criar certa automatização para empresa de processos como agendamento ou a aquisição de planos; e com um sistema de gerenciamento interno que mostra métricas e instrumentos que permitam ao proprietário um controle e planejamento do seu negócio.

Por estar na proposta do webservice, o sistema do NewGen pode ser facilmente encontrado por mecanismos de pesquisa da internet. Além disso, é possível e recomendado que o cliente utilize ferramentas comunitárias (comunitárias no sentido de possuir um público considerável) de exposição de locação espacos para divulgar o seu domínio (site) website, onde ocorre a interação com o cliente final para as locações automáticas digitais.

Vale ressaltar que este ponto da justificativa esta funcionalidade do NewGen é um diferencial em relação a sistemas comunitários de promoção de exposição de espacos de aluguel. Sistemas internos privados de gerenciamento desenvolvidos sob demanda, como o 2Work, ConvivaCoworking e GoWork2Work, ConvivaCoworking e GoWork, retratados na tabela abaixoQuadro 1, possuem o mesmo ponto positivo.

Posteriormente é válido também o estudo de parcerias com ferramentas tais ferramentas de exposição de locação comunitárias espacos, que propunham que os clientes dela

utilizem o nosso sistema para gerenciamento e locação automatizada do coworking, recebendo digitalizada do coworking. Ao fazer isso, a empresa por trás dessa ferramenta, receberia uma parcela de comissão por nos fechar uma venda, ao trazer um novo cliente ao NewGen. Ao mesmo tempo que a empresa publicitária estaria completando nesse de exposição estaria completando o seu sistema com algo que não é fornecido no dela: gerenciamento completo do negócio do coworking. Pondo em vista que as preocupações de já que fornecer um sistema de locação comunitária são estritamente divulgação e o tratamento para que ocorra a localização do negócio do seu cliente dentro do sistema gerenciamento não é o foco dela.

- Redução de custos e riscos ao cliente.

Dentre O NewGen oferece modelos personalizáveis que já podem ser usados pelo cliente no dia da própria compra, caso não seja desejado nenhuma configuração personalizada muito específica. E dentre as principais vantagens de uma aplicação White-label, como o NewGen, sobre uma aplicação sob demanda estão SaaS sob demanda customizável como este está principalmente a diminuição dos custos e riscos. Com uma aplicação adaptável a este modelo de negócio específico do coworking já pronta, os custos seriam consideravelmente menores, já que a NewGen não precisa fazer grandes modificações para adaptar à necessidade do cliente, pois em teoria tudo está pronto para ser modelado rapidamente e posto em prática.

Além disso, o cliente teria uma noção de como ficaria o sistema final e poderia fazer um “test-drive”, com termos ainda não definidos, antes da compra para certeza se é isso o que ele está procurando podendo cancelar sua assinatura e dentro de um prazo estipulado, ainda a ser analisado, ter seu dinheiro de volta. Nesse âmbito, os riscos de receber um produto de forma não desejável seriam pequenos. Em suma, ao adquirir o sistema NewGen, o cliente compraria um produto barato com poucos riscos de investimento pagaria menos do que seria para manter uma infraestrutura própria, além de poder cancelar a qualquer hora, caso o sistema não seja mais de seu agrado.

- Trazer mais empresas para os negócios de locação dinâmica, estimulando a entrada de mais clientes no coworking.

Partido do que foi abordado, é possível perceber que o sistema do NewGen será uma ferramenta “compre assine e faça o seu negócio” com um passo a passo guiado dentro da sua gama de funcionalidades. Se bem implementado implementada, a ferramenta tem o potencial de contribuir com o prolongamento do crescimento do coworking no Brasil, por facilitar de uma forma prática a entrada de novas empresas neste ramo.

Como abordado em 1.1 Contextualização Como abordado na 1.1 Contextualização, e mais detalhadamente em 2.1 Coworking Coworking; esse modelo de “coopetição” revitaliza

setores que estavam estagnados, promove um modelo de negócio mais humano com o palco principal sendo a interação, floresce uma rede de conhecimento rica entre profissionais, ajudam o crescimento de pequenas empresas como startups em uma associação de mutualismo, além de promover cenários de inovação, por causa de toda bagagem de criatividade. Em outras palavras, a aplicação sugerida pelo NewGen carrega um grande potencial para promover relações de ganho-ganho com diversos atores sociais, que vão desde a esfera individual, até a de grandes instituições, governamentais ou privadas.

1.6 Análise de concorrência

Com uma pesquisa de mercado, notou-se que por mais que haja ferramentas que solucionem alguns problemas pertinentes ao aluguel de escritórios, seja coworking ou não, não foi encontrada uma ferramenta específica que propunha atender exatamente o que a aplicação da NewGen propõe.

A aplicação da NewGen tem o objetivo de ser um software que atenda a maior parte das necessidades de uma empresa que tem escritórios para alugar. Resolvendo tanto a parte de gestão do negócio, com por exemplo funcionalidades de gestão de estoque, até sanando a parte de aluguéis, que é o cerne do negócio.

As ferramentas analisadas, em primeira análise, parecem focar principalmente na parte de aluguel, deixando o gerenciamento do negócio principalmente para os recursos humanos da empresa. O NewGen se propõe a ser uma aplicação que atende tanto o **MVP** (*Minimum Viable Product*) necessário para empresa, como funcionalidades extras que facilitem a gestão do coworking. Permitindo assim que uma empresa nova ou que queira ingressar na área de negócios possa montar um time pequeno de funcionários, depositando a maior parte dos seus processos básicos para a aplicação, que tratará deles **automaticamente** **programaticamente**.

Os concorrentes selecionados para o comparativo foram: o 2Work, ConvivaCoworking e GoWork2Work, ConvivaCoworking e GoWork, como sistemas privados, ou seja, sistemas próprios desenvolvidos pela empresa para atender as demandas de aluguéis; já para sistemas “públicos” que se assemelham a um Airbnb para coworking, ou seja, um serviço online e comunitário de exposição para as pessoas disponibilizarem seus espaços disponíveis para aluguel de coworking, foram analisados o CoworkingBrasil e WebEscrítorios CoworkingBrasil e WebEscrítorios.

Segue abaixo o quadro O 2Work é uma empresa de coworking que atua na cidade de São Paulo e de Campinas. Ela possui um sistema próprio de agendamento digital e planos de assinatura para Coworking. Há também um serviço de gerenciamento para o Coworking, mas não foi possível saber quais são as métricas e instrumentos desse serviço.

O ConvivaCoworking é uma empresa de coworking que atua na cidade de Aracaju. Ela possui um sistema próprio para lidar com o agendamento, mas diferente da 2Work, o cadastro dentro do site não é feito de forma automática. Em outras palavras, é necessário entrar em contato com um funcionário para ser cadastrado no site. Também como a 2Work, ela deve possuir uma ferramenta de gerenciamento, já que é um sistema interno, mas não é possível constatar as métricas e instrumentos de controle de gerenciamento.

O CoworkingBrasil é uma organização que oferece visibilidade para espaços de coworking. Ou seja, eles não são uma empresa com um espaço de coworking, mas sim disponibilizam um cadastro de espaços para os clientes anunciar seus espaços dentro da sua plataforma. O sistema do coworkingBrasil não oferece um sistema de gerenciamento para os seus clientes, se focando na divulgação.

O GoWork é uma empresa que disponibiliza espaços de coworking gerenciados por ela. É possível cadastrar um imóvel seu no site deles para ganhar com o lucro do gerenciamento deles, tal sendo analisado e acordado de antemão. Existem planos de assinatura que podem ser comprados digitalmente dentro do site, com algum grau de automação, o qual a empresa não precisa dispor um funcionário propriamente para gerenciar a compra das assinaturas. O site deles possui um sistema de gerenciamento próprio para coordenar os seus diversos estabelecimentos de Coworking. Um ponto negativo encontrado foi a super quantidade de informação apresentada de uma vez só.

A WebEscritórios é uma empresa focada em disponibilizar um espaço de visualização de escritórios disponíveis para aluguel. Eles possuem uma parte para Coworking dentro do seu site, mas em essência não é esse o único foco. Por ser uma ferramenta de procura de Coworkings, eles se centram no contato entre o cliente e o coworking. Neste sentido, assim como o CoworkingBrasil, não oferecem aos proprietários de coworking ferramentas de gerenciamento, com métricas e instrumentos de controle, e o agendamento é mais manual, sendo necessário haver um funcionário para o cliente entrar em contato, seja via email, telefone ou whatsapp.

O Quadro 1 é um comparativo das funcionalidades dos concorrentes com a aplicação do NewGen, onde X aponta existência da funcionalidade:

Quadro 1 – Comparação de Funcionalidades Entre Concorrentes.

Aplicações	Agendamento Programático	Planos Flexíveis	Ferramentas de Gerenciamento	Aluguel por Assinatura	Locação de Equipamentos
2Work		x		x	
ConvivaCoworking		x		x	
CoworkingBrasil		x			
GoWork		x		x	
WebEscritórios		x			
NewGen	x	x	x	x	x

Fonte: Os Autores

A fim de esclarecimentos, ~~encontra-se abaixo~~ são listadas algumas explicações pertinentes sobre algumas funcionalidades que foram selecionadas como necessárias de detalhamento, expandindo o que é retratado do gráfico.

1.6.1 Agendamento ~~Automático~~Programático

~~A funcionalidade~~ A funcionalidade *agendamento automático*programático se refere a uma reserva ou locação que não necessite o contato direto com algum funcionário. Nos sistemas analisados todos pedem que o usuário entre em contato com alguém da empresa, seja com um número de telefone ou email.

O NewGen não planeja descartar essa funcionalidade, mas adicioná-la como opcional, deixando a locação totalmente ~~automática~~caso o usuário queira, ~~programática~~caso o cliente proprietário queira. De forma que o usuário possa ir navegando por opções que aponte para ele quais são as salas disponíveis ou os horários que ele pode alugar o pacote que ele deseja ~~sem que haja necessidade de entrar em contato direto com algum funcionário responsável~~.

Contudo, caso haja maiores dúvidas, haverá a disponibilidade de um contato para suporte do usuário com o responsável da empresa cliente do NewGen.

1.6.2 Ferramentas de Gerenciamento

As *ferramentas de gerenciamento* se referem às ferramentas que contribuem com a gestão do negócio do coworking, não se limitando apenas a automatizar a locação.

O NewGen, se propondo a ser ~~um white-label~~customizável para empresas que queiram ingressar no mercado de coworking, oferece uma gestão completa de todos os recursos que a empresa possa querer. Em ferramentas de gestão, no quadro descrito, inclui-se ferramentas que mostram para o gestor quais são as salas alugadas, equipamentos usados, como está o estoque, quais são os clientes mais valiosos para empresa, entre outras.

As ferramentas privadas desenvolvidas pelas empresas com o propósito exclusivo

de atender o seu próprio negócio, marcadas pelas aplicações 2Work, ConvivaCoworking e GoWork, 2Work, ConvivaCoworking e GoWork, foram selecionadas como um ponto de incerteza, pois não é possível constatar se há dentro da ferramenta web-service ferramentas de gerenciamento que não se limitem apenas a locações. No entanto, as aplicações CoworkingBrasil e WebEscrítorios, CoworkingBrasil e WebEscrítorios, que são fundamentadas em serem um aluguel comunitário, onde qualquer empresa pode anunciar seu espaço de locação ali, não têm ferramentas de gerenciamento disponíveis para as suas empresas clientes, por causa do seu propósito de terceirizar somente a responsabilidade de aluguel do coworking.

1.6.3 Locação de Equipamentos

Já a funcionalidade *locação de equipamentos* se refere à personalização de equipamentos usados na sala, podendo ser depois do agendamento. Em outras palavras, se um usuário desejar alugar um equipamento para sua sala, mesmo posterior à sua alocação, é possível que ele use um sistema fornecido pela equipe NewGen que deixe ele selecionar os itens que estão disponíveis para contribuir com o seu expediente. Não é possível com uma análise crua saber se os outros sistemas de coworking têm este sistema, pois se tiverem é só para clientes internos que já reservaram algum espaço, porém, considerando que não há agendamento automático em primeira vista, não é muito absurdo supor que se houver tal locação, o mesmo ocorre de forma manual, com contato prévio do usuário com o responsável da empresa para pedido de equipamentos extras nas suas salas.

O sistema do-NewGen propõe um gerenciamento automático destes recursos e diminuir dentro do possível a interação do cliente com o funcionário, automatizando o gerenciamento do negócio.

1.6.4 Funcionalidades RetiradasAdiadas

Para finalizar, o recurso do *chatbot* era um recurso cogitado para entrar como uma funcionalidade adicional do NewGen, como forma de automatizar ainda mais o suporte de solução de eventuais dúvidas do usuário. Ele entrou como um recurso a ser avaliado nas plataformas concorrentes, mas posteriormente foi retirado do planejado para a aplicação dentro da etapa de desenvolvimento acadêmico por expandir o escopo proposto pela disciplina demasiadamente. Posteriormente, após o lançamento da aplicação com o seu uso comercial, esta função poderá ser cogitada novamente.

O mesmo ocorreu com a funcionalidade de *Eventos de Integração*, com a diferença que este não foi incluído na tabela final, considerando que a inclusão deste poluiria mais ainda a tabela (o chatbot só foi incluído como exemplificação de que o NewGen não trará todas as funcionalidades que todas as plataformas possuem e que existem outras

funcionalidades que o NewGen não ~~trata~~tratará em sua primeira versão).

2 Revisão de Literatura

Este capítulo consiste em expor o principal conceito abordado pela plataforma do sistema NewGen: O coworking. Aqui é dado um maior enriquecimento bibliográfico dentro do assunto do coworking, analisando suas diferentes facetas com maiores detalhes, fornecendo um entendimento mais rico das suas causas, imprescindível para a compreensão da importância da ferramenta proposta pela NewGen.

2.1 Coworking

Com o cenário do mundo atual sendo altamente digitalizado, é exigido grande flexibilidade às mudanças, pois, por permitir grande rapidez nas trocas de informações, é requisitado que as empresas atendam as atualizações das demandas de forma dinâmica também. Junto a isso, houve também a crise de 2008 que aprofundou um sentimento de necessidade de renovação dentro dos administradores ([GANDINI, 2015](#)). Neste contexto, o qual os trabalhadores são incentivados a adotarem ritmos igualmente dinâmicos, surge o Coworking como um método de trabalho alternativo.

Coworkings, segundo [Gandini \(2015\)](#), são espaços compartilhados utilizados por diferentes tipos de profissionais de diversas áreas do conhecimento, a maioria freelancers; trabalhando em vários graus de especialização no vasto domínio do conhecimento industrial. Ainda de acordo ele espaços de coworking são concebidos como lugares de escritórios de aluguel onde os trabalhadores pagam uma taxa para dispor-se de uma mesa e uma conexão wireless, que são os dois fatores mais cruciais quando se trata de um lugar produtivo hoje em dia. Nestes locais, esses profissionais podem viver suas rotinas de trabalho normalmente ao lado de outros profissionais, sendo da mesma área de especialização ou não.

2.1.1 Desenvolvimento do Coworking Moderno

O coworking contemporâneo surgiu em 2005, em São Francisco ([GANDINI, 2015](#)). Conforme o autor, o modelo de trabalho nasceu para ser uma "terceira via": entre o modelo padrão de estilo de trabalho com um local de trabalho bem delimitado com comunidade e cenários característicos, e o estilo de trabalho independente de um freelancer, caracterizado por liberdade e independência, onde o local de trabalho é na sua maioria baseado na isolação do ambiente doméstico. Esta terceira foi cunhada como coworking indica a prática de trabalhar individualmente em um ambiente compartilhado.

De acordo com [Pratt \(2008\)](#), a península de São Francisco foi uma das áreas que lideraram a produção de nova mídia no começo dos anos como um resultado de

infraestrutura híbrida de interação capaz de conectar tecnologias, pessoas e espaços. Pratt aponta que São Francisco, localizado no famoso polo de inovação Vale do Silício, com um alto grau de concentração de empresas de tecnologia e de indústrias de hardware, satisfaçõs os requisitos para uma nova visão contemporânea de valor do espaço. Isso se deu inclusive por causa da cultura vibrante com ativismo político e padrões de trabalhos organizados centrados na estrutura social, baseados em redes sociais e compartilhamento de conhecimento (PRATT, 2008).

Após pouco tempo da sua concepção, a ideia de coworking se espalhou rapidamente, se tornando hoje em dia um tópico quente com grandes expectativas acerca do futuro do jeito de se trabalhar (GANDINI, 2015). Segundo Gandini, a definição de coworking como uma terceira onda de trabalho virtual procura restaurar a ideia de compartilhamento de espaços, atendendo os modos de produção digital, cujas tarefas podem ser realizadas em qualquer lugar a qualquer hora.

Gandini (2015) conta que a proliferação das iniciativas de coworking podem ser testemunhadas em diferentes cidades ao redor do globo, pois de alguma forma o movimento coworking está alinhado com um outro tópico tendência, que são o cenário das startups, da inovação social e da economia compartilhada - os quais floresceram após a crise de 2008.

Não à toa, na última década, o desenvolvimento de espaços de coworking tornou-se um fenômeno global. A deskmag, uma revista online sobre coworking, traz dados acerca de tal crescimento: no começo de 2008, havia 160 espaços que se auto proclamavam espaços de coworking, em 2018 este número passou para 19000. E a tendência de crescimento ainda é contínua (LUO, 2020).

Luo (2020), traçando uma perspectiva oriental, remontam também como a China pós-socialista está usando o movimento do coworking para revitalizar espaços abandonados devido à crise de transição de modelos econômicos. Mostrando como o modelo do coworking evoluiu de uma tendência alternativa, para um modelo avaliado por grandes estados para revitalização de seus espaços urbanos e crescimento econômico no setor das pequenas empresas.

2.1.2 Benefícios do coworking

A ideia mais atraente do coworking é justamente o "sharing" ou compartilhamento que o trabalhador tem no seu ambiente de trabalho (GAUGER ANDREAS PFNÜR, 2021), pois inicialmente um freelancer, um profissional que trabalha independentemente sob demanda, não teria acesso. Dessa forma, é possível criar um networking - uma rede de comunicações com vários outros profissionais - que é um ponto crucial que as empresas analisam na hora da contratação, considerando que essa rede de comunicação cria possibilidades e perspectivas para a empresa. Um profissional que conhece o pessoal

da empresa parceira da sua pode criar pontes sociais e acordos financeiros valiosos.

A prevalência de demandas robustas por trabalhadores de coworking explica o crescimento disruptivo dos espaços de coworking ([SPINUZZI, 2012](#)):

1. Procurar um lugar de trabalho formal com ferramentas de trabalho básica;
2. Evitar o senso de isolamento por trabalhar sozinho, procurar vizinhos e um senso de comunidade;
3. Acessar o transbordamento de conhecimento.

De acordo com [Gauger Andreas Pfniür \(2021\)](#), os espaços de coworkings se tornaram avenidas de troca de conhecimento, estimulando fertilização cruzada e colaboração por meio da intensiva coexistência de diversos atores sociais. A vida diária, como comer junto ou trabalhar à noite, como também a intensidade de diversos eventos (seminários, palestras, encontros semanais e conferências) aumentam a possibilidade de encontros frutíferos. Estas interações face-a-face promovem confiança e compartilhamento tácito de informações. Portanto, os espaços de coworking possuem um grande papel dentro da inovação microescala entre indivíduos e firmas, fomentando não só um ambiente de troca de conhecimento individual, como também um ecossistema empreendedor ([LUO, 2020](#)).

Um ecossistema empreendedor é comumente definido como um conjunto de fatores tangíveis e intangíveis que modelam a performance de startups e trabalham em conjunto para crescimento econômico e imersão através de interações bem-sucedidas ([GAUGER ANDREAS PFNÜR, 2021](#)). É justamente o que ambientes de coworking podem prover com seus espaços de trabalho compartilhados e multi locatários. Eles fornecem a infraestrutura institucional para o empreendedorismo intensivo acerca do conhecimento, além da infraestrutura física, como espaço mobiliado, salas apropriadas para cada trabalho, internet, entre outros recursos.

Aproveitando o link empresarial, é notável comentar também que o coworking é ligado ao crescimento neoliberal urbano devido ao caráter individualista, à tendência para empresas tecnológicas com fins lucrativos nas decisões públicas e às relações público-privadas vibrantes nas práticas locais. O crescimento de indivíduos empreendedores em espaços de coworking ecoa valores neoliberais que identificam a liberdade individual e a liberdade como sacrossanto - como os valores centrais da civilização ([GAUGER ANDREAS PFNÜR, 2021](#)).

Ainda segundo os autores, o cenário atual de competição global incentiva o empreendedorismo nas cidades. O coworking foi recentemente reconhecido por planejadores urbanos como lugares importantes para este empreendedorismo e inovação. O desenvolvimento desses hubs criativos empreendedores fomenta a regeneração e o planejamento urbano.

Inúmeros espaços de coworking são localizados em construções antigas, galpões e fábricas de metrópoles centrais, revitalizando o uso de construções uma vez decadentes para um cenário totalmente oposto de criação e inovação.

Além disso tudo, o coworking é um negócio lucrativo para empresas privadas. Os espaços providos por estas empresas se tornam, para startups e empresas pequenas, provedores de serviços essenciais para a prática benéfica do coworking. ([GAUGER ANDREAS PFNÜR, 2021](#))

Não só isso, as empresas provedoras podem se tornar parceiras, clientes ou investidoras de suas empresas clientes, se for constatado que existem interesses industriais covalentes. Ao financiar estes espaços de coworking, as empresas mãe lucram com pagamentos de aluguel, melhoram os perfis de marketing e estendem as conexões externas para iniciativas emergentes ([MORISET, 2013](#)).

Em resumo, a prática do coworking produz benefícios para diversos atores sociais, seja um trabalhador independente, um trabalhador de pequena empresa, uma startup, empresas que provém o aluguel ou até mesmo as cidades preocupadas em se expandir tornando-se um centro de referência de tecnologia e cultura - como foi o caso de São Francisco, visto no [tópico 2.1.1 Desenvolvimento do Coworking Moderno](#) - existem lucros interessantes para levar em consideração.

2.1.3 Porque o coworking

Para uma maior análise desta questão, catalogando e expandindo sob perspectiva os assuntos já abordados, cabe considerar que existem duas abordagens para esta pergunta: a visão empreendedora e a visão do trabalhador. O primeiro considerando o ponto de vista da empresa e o segundo considerando o ponto de vista do cliente final do coworking.

2.1.3.1 Ponto de vista do empregador

Primeiro analisando do ponto de vista empreendedor, que além de雇用, precisa se preocupar com custos de manutenção e colocar em pauta outros custos de gestão que possam passar despercebidos através de uma análise breve. Dentre as opções de mercado, o administrador pode escolher: o home office, o coworking, business centers ou até mesmo um escritório próprio como lugar para deslanchar a produção ([COWORKING BRASIL, 2017](#)).

Antes de tudo, o home office sem sombra de dúvida aparece como a solução mais flexível e barata que existe para as empresas que querem utilizar serviços distribuídos dentro de sua matriz. Outrossim, a possibilidade de os funcionários trabalharem de suas casas exclui problemas de logística, como o trânsito, permitindo que eles concentrem a

maior parte do tempo apenas no que deve ser produzido para agregar valor à empresa ([COWORKING BRASIL, 2017](#)).

Ainda sobre o home office, [COWORKING BRASIL \(2017\)](#) conta como é notório o grande corte de gastos com aluguel, instalações e manutenção, uma vez que uma parcela dos trabalhadores da empresa seja autônoma, grande parte dos serviços da empresa podem ficar distribuídos na nuvem. No geral é mais barato a virtualização da matriz corporativa que a utilização de espaços físicos, considerando a infraestrutura necessária para prover os direitos dos trabalhadores.

Todavia, apesar de mais econômico, pesquisas mostram que home-offices falham em trazer o estímulo e sinergia necessários para colaboração ([SPINUZZI, 2012](#)); em outras palavras, a interação entre trabalhadores fisicamente favorece a construção de um trabalho colaborativo de forma mais produtiva do que a do trabalho remoto. Como Aristóteles dizia, o homem é um animal social, então precisamos de interações físicas para nos integrarmos ao grupo efetivamente. Assim, trabalhos de espaços adequados incentivam não só a produtividade bruta por causa da infraestrutura, mas também por causa da conexão humana.

Pensando em resolver este problema, para uma startup ou uma empresa ainda não consolidada, é possível também a utilização de *business centers* (escritórios virtuais), conferindo espaços mais apropriados para a interação humana e para o fluxo de trabalho, bem como trazendo ainda certo grau de flexibilidade de custos. Porém, a rede de comunicação nestes espaços, por trazerem uma roupagem mais privativa, própria da empresa, ficaria restringida aos colegas da mesma empresa ([COWORKING BRASIL, 2017](#)).

Soma-se a isso, é primordial que companhias que estejam iniciando no mercado, e.g. startups, tenham o acesso a uma rede colaborativa que permita o surgimento de novas possibilidades e o desenvolvimento de estratégias mais inteligentes ([GAUGER ANDREAS PFNÜR, 2021](#)). Assim sendo de suma importância não só o fortalecimento das interconexões, interações dentro da própria empresa, mas como também o fortalecimento das interconexões de empresas.

Para finalizar, o coworking costuma apresentar custos mais adaptáveis ao cenário da empresa. Segundo o [COWORKING BRASIL \(2017\)](#), a burocracia destes ambientes é quase zero, sendo as maiores dificuldades centradas em encontrar um espaço adequado. Além disso, não há preocupação com a manutenção, pois essas responsabilidades ficam terceirizadas com a empresa que aluga o local de trabalho. É atrativo também considerar que há escalabilidade neste modelo, porque conforme o time da empresa vai aumentando, é possível reajustar o aluguel para salas ou opções que suportem mais funcionários ou uma maior infraestrutura. Ele é válido para casos em que houve a diminuição do escopo. Ademais, o coworking é considerado um espaço de classe mais criativa que incentiva o

desenvolvimento de ideias e corrobora com trabalhos que demandam pensamentos "fora da curva", como desenvolvimento de Software, por exemplo. ([GANDINI, 2015](#))

Após estas considerações, nota-se que o coworking se faz o modelo ideal às empresas que busquem o networking e a troca de experiências, com uma boa flexibilização de custos, que é o que startups ou empresas que ainda estão se consolidando no mercado precisam. Subir a popularidade no mercado corporativo, por regra se dá com as interações em redes de comunicação profissional, reforçando a importância do coworking para o crescimento de empresas pequenas ou até para o surgimento de um cenário econômico mais próspero.

2.1.3.2 Ponto de vista do trabalhador

O tópico anterior centrou-se na visão estratégica de um empregador para a escolha do coworking, porém, fica claro que não é só as corporações que têm a ganhar com o coworking.

Aproveitando o gancho do aspecto social do tópico anterior, em um ensaio teórico baseado no vídeo de [Akita \(2019\)](#), suponha que você é desenvolvedor de uma empresa e esbarrou em um problema crucial que tenha interrompido seu fluxo de produção. Uma das opções seria ir à internet e procurar soluções específicas para seu problema, na maior parte dos casos isso será resolvido. No entanto, eventualmente existirão problemas tão específicos que serão interessantes de serem revistos por um maior especialista ou por uma segunda opinião. No meio web, é possível contactar pessoas para ajudar no problema, seja a partir de mensagens ou chamadas; porém, não será tão efetivo quanto a interação presente dentro de um ambiente físico, o qual, seja por gestos ou maior taticidade, a comunicação flua mais espontaneamente. Além disso, pelo meio físico, se cria uma maior intimidade, conferindo um possível acréscimo do networking, com o bônus do aumento da bagagem profissional para solução de problemas.

Não só isso, para freelancers ou pessoas independentes que buscam um ambiente mais adequado para o trabalho, o coworking não se trata apenas de aferir profissionalidade, há a criação de grandes oportunidades. Com a rede de informações, o profissional independente que adota o coworking ganha para si novas oportunidades de emprego, reflexões ou perspectivas novas acerca do mercado de trabalho, uma melhor visão das tendências atuais e motivações para produzir o que foi definido ([GANDINI, 2015](#)). Os últimos dois, obviamente, não são apenas benéficos para freelancers, mas também para contratados.

Além disso, um cenário com a criatividade em pauta, fomenta novas perspectivas e ideias para o próprio trabalho do profissional, como mostrado por [Gandini \(2015\)](#). Sendo possível ser mais produtivo em um ambiente de coworking do que em um ambiente tradicional, dependendo do perfil do trabalhador.

Em síntese, o ambiente do coworking expande o olhar dos trabalhadores e pode

fomentar um ambiente saudável de produção que os outros meios mais convencionais não conseguam.

Definições Gerais

3

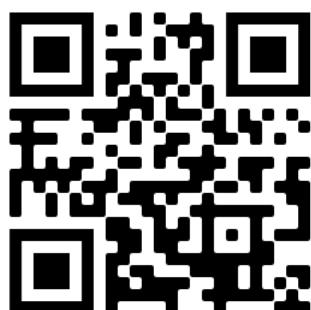
Desenvolvimento do projeto

Neste tópico, trataremos da definição da metodologia escolhida para capítulo, trataremos a evolução do desenvolvimento do projeto e o motivo para utilizá-la, apontando suas principais características e benefícios; o papel de cada integrante na equipe para melhor dividir as atividades durante o desenrolar da disciplina e para cada um se dedicar naquilo que mais tem domínio; a arquitetura da da equipe NewGen, referente a aplicação que irá verificar o atendimento sobre as tecnologias a serem utilizadas ; como iremos fazer a gestão do tempo das entregas do projeto durante o semestre em que teremos a disciplina, visto que estamos com um prazo muito menor devido a pandemia; os . Serão detalhadas a organização da equipe e o uso da metodologia, as tecnologias utilizadas no projeto, a manutenibilidade da plataforma e design patterns utilizados, a viabilidade financeira, os requisitos levantados e critérios que utilizaremos para de segurança e privacidade dos dados dos usuários, tudo dentro dos parâmetros , e outros pontos importantes para compreensão da legislação; e por fim, as tecnologias que aplicação.

3.1 Links do projeto

Para apresentar o desenvolvimento da aplicação, nesta seção serão utilizadas para o desenvolvimento disponibilizados os links do projeto, tanto para back-end (lógica, funcionalidades, regras, segurança e integridade do banco de dados) como front-end (parte visível da aplicação para o usuário)em QR code, para acesso ao site, repositório do SVN, repositório do GitHub, blog da equipe e o canal do Youtube.

Figura 7 – Código QR da aplicação.



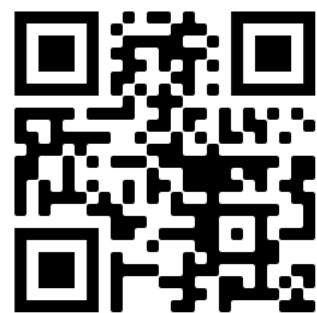
Fonte: Os autores

Figura 8 – Código QR do repositório SVN.



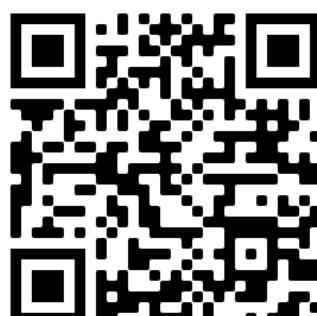
Fonte: Os autores

Figura 9 – Código QR do repositório no GitHub.



Fonte: Os autores

Figura 10 – Código QR do Blog da equipe.



Fonte: Os autores

Figura 11 – Código QR do canal do Youtube.



Fonte: Os autores

3.2 Metodologia de Gestão de Projetos

Após analisarmos as alternativas mais comuns de metodologia ágil disponíveis no mercado, optamos por escolher e utilizar a metodologia [Scrum](#). Segundo [Lamelas \(2018\)](#), é um dos métodos mais utilizados atualmente, se destacando pela liberdade de implementação. Sendo bem implementado, funciona como um contêiner para outras técnicas e metodologias.

Suas principais características consistem em ser um método fácil de aprender e usar já que temos papéis bem definidos do que cada cargo terá como função. Também é mais flexível na coleta de requisitos, permitindo iniciar o desenvolvimento logo no início do projeto e aumenta o controle das atividades, o que otimiza a eficiência e eficácia da equipe. ([SCHWABER, 2020](#))

Através de Sprints o Scrum vai desenvolvendo o projeto, onde em cada sprint a equipe de desenvolvimento foca em codificar uma funcionalidade específica em um prazo determinado. Assim, o projeto vai ganhando corpo conforme cada Sprint é realizada.

3.3 Organização da equipe

Por ser um projeto que terá a metodologia ágil [Scrum](#) como sua base, é preciso dividir os integrantes do grupo em papéis para melhor desenvolvimento da aplicação.

De acordo com todos os membros, Camila França será a Scrum Master do projeto, garantindo que a cultura e as práticas do Scrum sejam seguidas, tendo o conhecimento do que fazer para ajudar a definir como fazer.

O papel de Product Owner será de Fernando Sousa, responsável por definir o que deve ser feito em cada Sprint, realizando as reuniões semanais, verificando com os desenvolvedores o andamento da programação e formalizando a documentação.

Os outros integrantes ficarão responsáveis pela programação da aplicação como um todo.

Apesar das definições dos papéis, é importante destacar que todos os membros do grupo terão passagem por todos os papéis definidos, tanto para experiência de como é possuir determinada função, como para aprendizado do projeto como um todo, com a finalidade de todos possuírem o mesmo nível de conhecimento da aplicação em sua totalidade.

3.4 Sprints

As sprints, períodos de tempo estabelecidos pelo Ágil, foram feitas com o auxílio da ferramenta Jira. A equipe trabalhou com sprints de duração compatíveis com as entregas, variando entre uma a três semanas. Para maiores detalhes sobre elas, verificar o Apêndice A.

3.5 Métricas do projeto

Nessa seção serão apresentadas as métricas do projeto. Os valores das métricas foram atualizadas no dia 09/08/2021 e são referentes ao total de commits e linhas de código, como mostrado em Figura 12.

Os dados foram coletados com o uso da ferramenta StatSVN, as informações e dados presentes na tabela Figura 13 demonstram as linhas de código e as modificações realizadas por cada membro da equipe. Mesmo após a entrega desse documento, a equipe continua trabalhando no desenvolvimento da aplicação.

Figura 12 – Contribuição ao SVN em total de linhas

Author	Author Id	Changes	Lines of Code	Lines per Change
Gabriel	Gabriel	304 (72.2%)	42342 (57.1%)	139.2
Bruna	Bruna	52 (12.4%)	31630 (42.7%)	608.2
Beatriz	Beatriz	3 (0.7%)	138 (0.2%)	46.0
Lucas	Lucas	62 (14.7%)	0 (0.0%)	0.0
Totals		421 (100.0%)	74110 (100.0%)	176.0

Fonte: Os Autores

Figura 13 – Contribuição ao SVN mensalmente

Last 6 Months						
Author	2/2021	3/2021	4/2021	5/2021	6/2021	7/2021
Gabriel	0 (-)	0 (-)	0 (-)	0 (-)	1671 (73.2%)	40671 (56.6%)
Bruna	0 (-)	0 (-)	0 (-)	0 (-)	473 (20.7%)	31157 (43.4%)
Beatriz	0 (-)	0 (-)	0 (-)	0 (-)	138 (6.0%)	0 (0.0%)
Lucas	0 (-)	0 (-)	0 (-)	0 (-)	0 (0.0%)	0 (0.0%)
Totals	0 (-)	0 (-)	0 (-)	0 (-)	2282 (100.0%)	71828 (100.0%)

Fonte: Os Autores

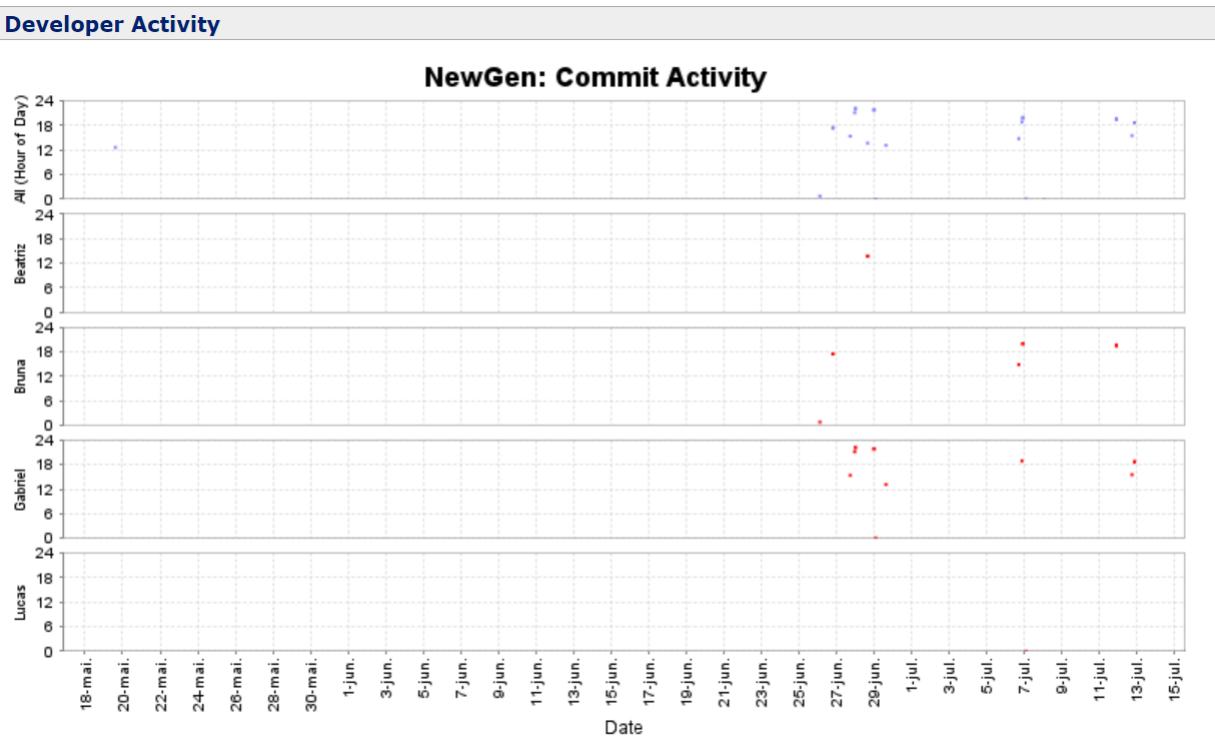
3.5.1 Estatísticas SVN

Nesta sub-seção estão os gráficos estatísticos do projeto em relação os material versionado pelo SVN. Os gráficos que serão referidos nesta seção são referentes a valores e dados que envolvem tanto códigos quanto documentos desenvolvidos pela equipe. Os gráficos foram gerados com a ferramenta StatSVN, que faz a análise do repositório e gera gráficos relacionados a diferentes aspectos do mesmo, como quantidade de commits realizados por cada integrante e equipe, quantidades totais e específicas de linhas de códigos de cada arquivo.

É importante considerar o fato de que são analisados todos os arquivos do repositório, incluindo comentários de código e os relacionados a documentação.

A Figura 14 exibe um gráfico de dispersão que exibe as datas de cada envio no eixo horizontal, dividido na vertical por cada integrante da equipe com a distribuição total na parte superior do eixo vertical, representados pelo nome de cada membro da equipe em cada hora do dia. Cada integrante da equipe ficou responsável pela publicação do seu desenvolvimento no repositório durante todo o projeto.

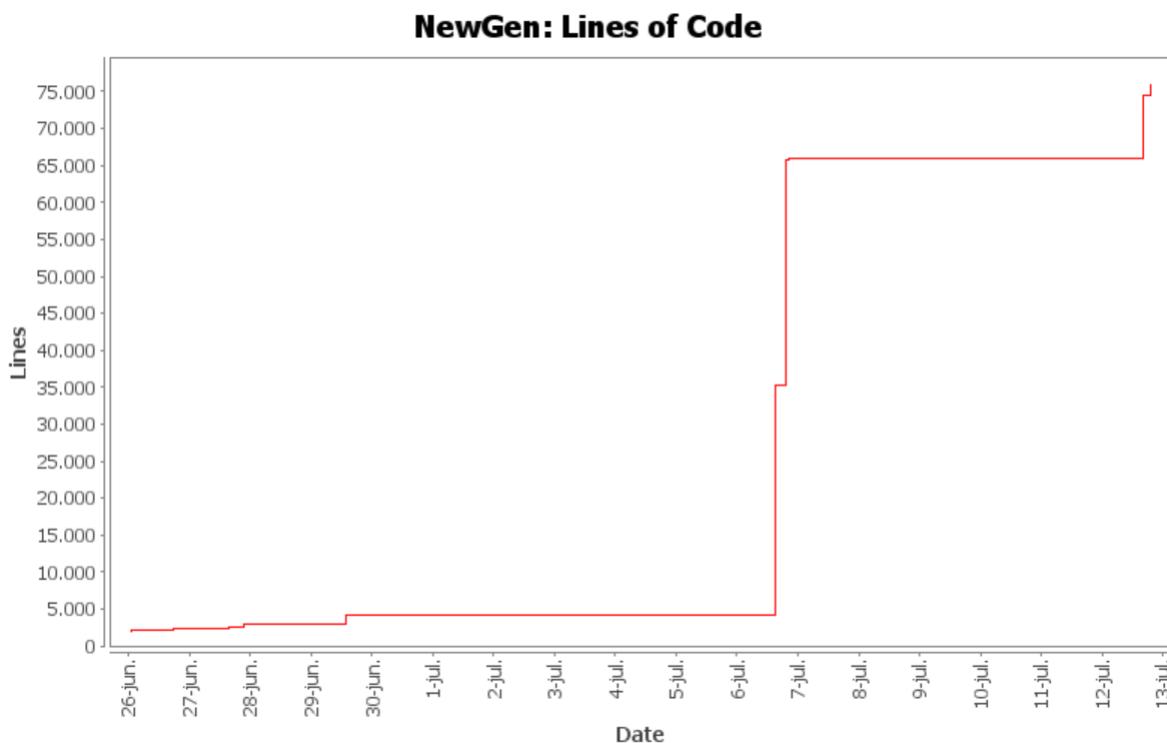
Figura 14 – Dispersão de commits



Fonte: Os Autores

Na Figura 15, é possível ver que a maior parte do desenvolvimento se deu inicio no mês de Junho.

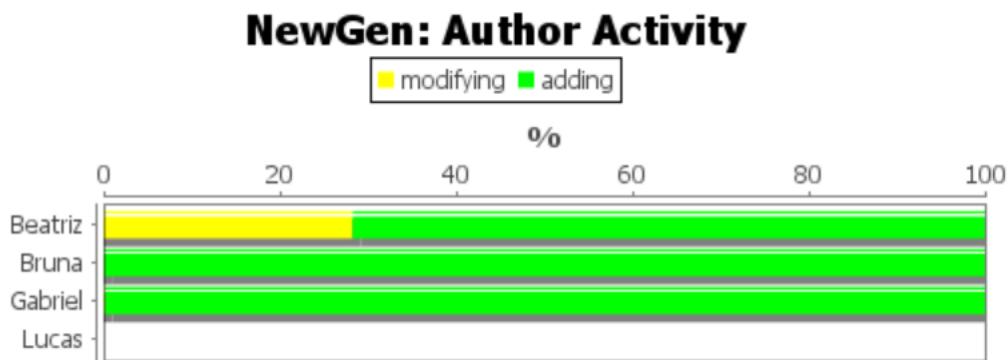
Figura 15 – Quantidade de linhas de código



Fonte: Os Autores

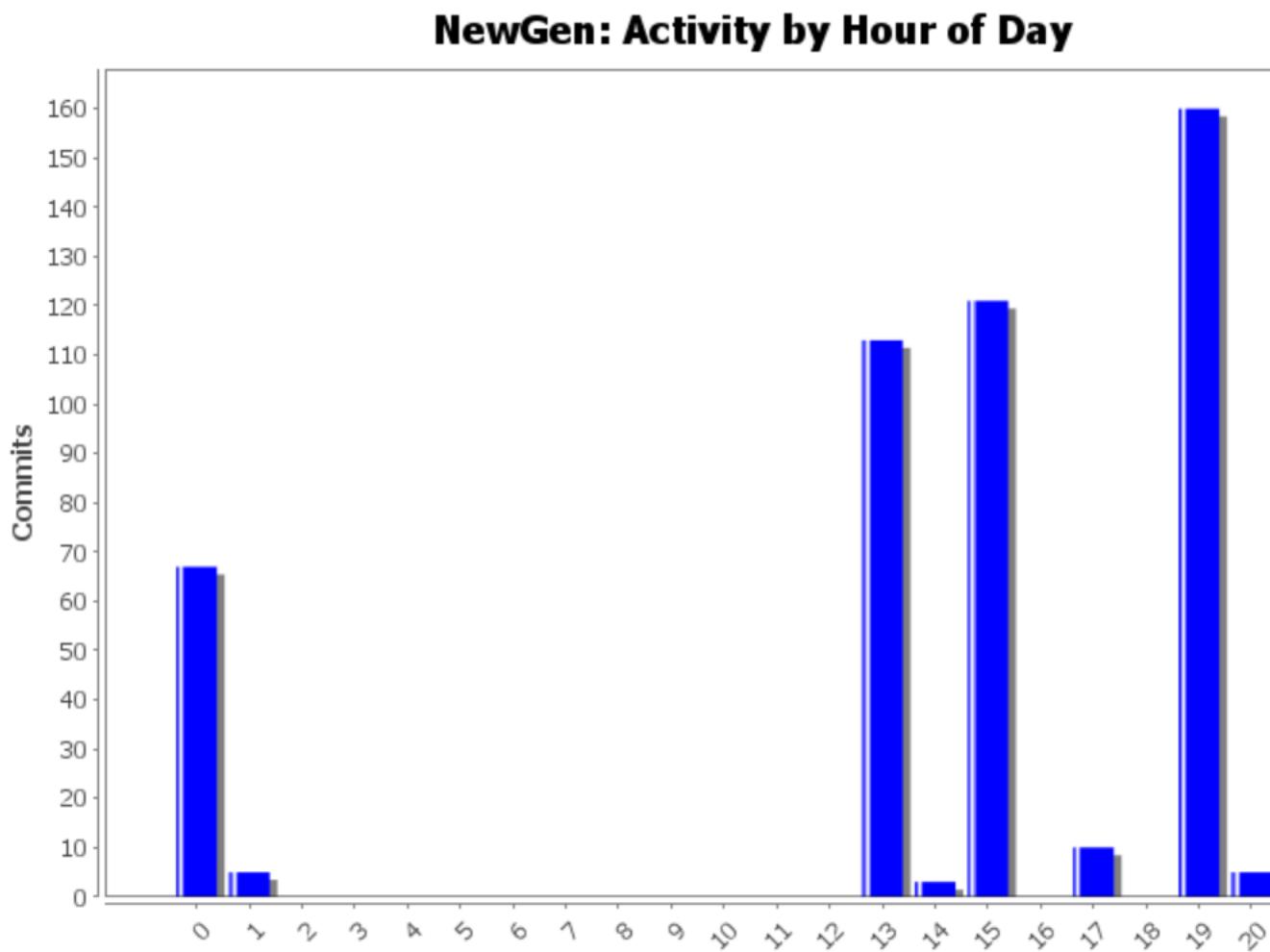
A Figura 16 exibe a atividade por autor. A parcela amarela da barra representa as modificações realizadas em arquivos pré-existentes e a parcela verde, as adições de arquivos novos ao projeto.

Figura 16 – Tipos de atividades dos autores



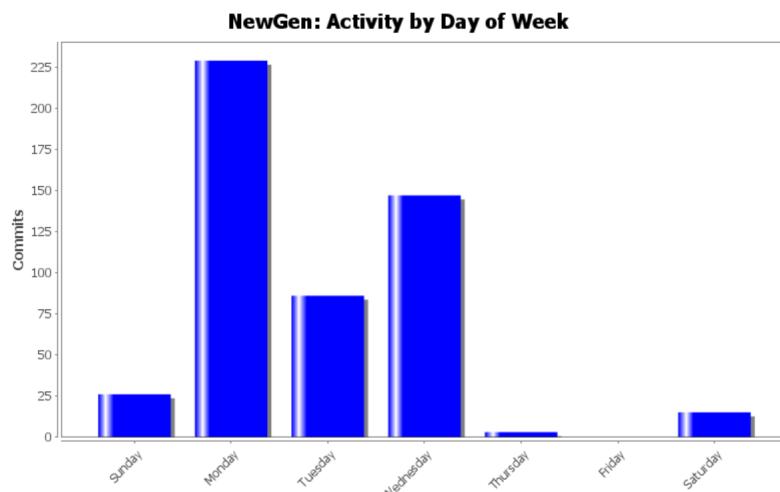
Fonte: Os Autores

A Figura 17 apresenta os horários onde houveram mais commits.

Figura 17 – Commits por horário

Fonte: Os Autores

Na Figura 18, é demonstrado a quantidade de commits no eixo vertical por dias da semana no eixo horizontal.

Figura 18 – Commits por dia da semana

Fonte: Os Autores

3.6 Definição de Entregas

Após mapear e descrever todas as funcionalidades que nossa aplicação terá, foi definido quais processos iriam ser possíveis de serem desenvolvidos para as entregas planejadas, para alinhamento do andamento do projeto.

O Quadro 2 demonstra o cronograma a ser utilizado para as entregas das funcionalidades, e em qual entrega cada uma estará presente (*Proof of Concept* (POC), MVP, ou Produto Oficial/Final).

3.7 Tecnologias utilizadas

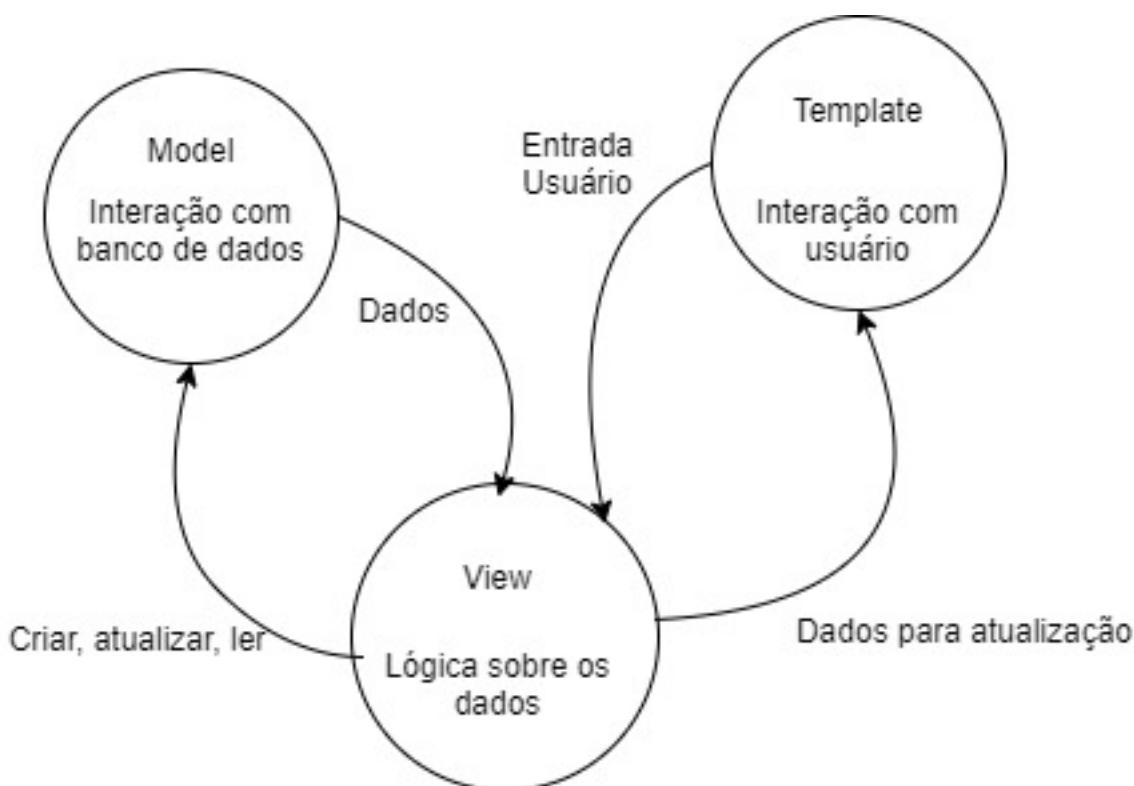
- **Python:** É uma linguagem de programação de alto nível, robusta, dinâmica, interpretada, multiplataforma, com sintaxe de fácil compreensão e de acordo com o paradigma da programação orientada a objetos. O Python possui uma vasta variedade de bibliotecas e grande comunidade de desenvolvedores. Sendo assim, possibilitando a disponibilidade de uma aplicação sofisticada. Deve ser a linguagem responsável pelo back-end da aplicação.
- **Django:** É um framework que deve ser utilizado em conjunto com a linguagem Python para facilitar no desenvolvimento da aplicação, a qual disponibiliza um conjunto de bibliotecas para ser efetuada a criação de base da aplicação. Portanto, permite otimizar os recursos para facilitar na arquitetura e configuração do sistema. O Django deverá utilizar o padrão MTV para a aplicação. As camadas são respectivamente:

Quadro 2 – Escopo do projeto.

Funcionalidades	POC	MVP	Produto final
Cadastro de usuário	X		
Login de usuário		X	
Reserva		X	
Aluguel			X
Lista de Espera			X
Autorização de Entrada			X
Locação de Máquinas			X
Relatórios Periódicos			X
Sistema de Fidelidade			X
Transferência de Responsabilidade			X
Controle de Estoque			X

Fonte: Os Autores

1. **Model** – Para realizar a interação com o banco de dados.
 2. **Template** – Onde realiza a renderização dos dados para interação do usuário com o site.
 3. **View** – É a camada responsável pelas regras de negócios.
- **Javascript**: É uma linguagem de programação de alto nível, dinâmica, permite realizar o controle de multimídia e animação de imagens. Esta linguagem, dedicada a web é a terceira camada para o desenvolvimento de aplicações, juntamente com o HTML (linguagem de marcação) e CSS (Linguagem para regras de estilo).
 - **Bootstrap**: É um framework front-end que fornece componentes preparados para utilização na aplicação. Sendo possível criar e personalizar site web responsivo, com a estrutura de HTML e CSS preparada para utilização. Com isto, permite que o site possua um aspecto visual elegante por disponibilizar componentes e plugins para o uso no projeto.
 - **MySQL**: Trata-se de um banco de dados relacional com um modelo de cliente servidor, local onde os dados serão armazenados e gerenciados. O MySQL deve ser instanciado pelo serviço RDS da Amazon Web Services.

Figura 19 – Modelo MTV no Django

Fonte: Caderno de Laboratório (2021)

- Serviços Amazon Web Services (AWS): A aplicação deve estar disponibilizada através da plataforma de serviços web da Amazon, os serviços utilizados devem ser:

Amazon CodePipeline: Serviço de integração contínua para automatização do processo de lançamento da aplicação.

Amazon EC2: Serviço web para disponibilizar capacidade computacional na nuvem. Neste serviço, é possível ter controle dos recursos computacionais, na qual devemos utilizar uma configuração de processamento, armazenagem e de sistema operacional para instanciação de virtualização para o ambiente do servidor.

Amazon S3: Serviço para armazenagem em nuvem que deve ser utilizado para backup, arquivamento de dados e aplicações na Amazon Web Services.

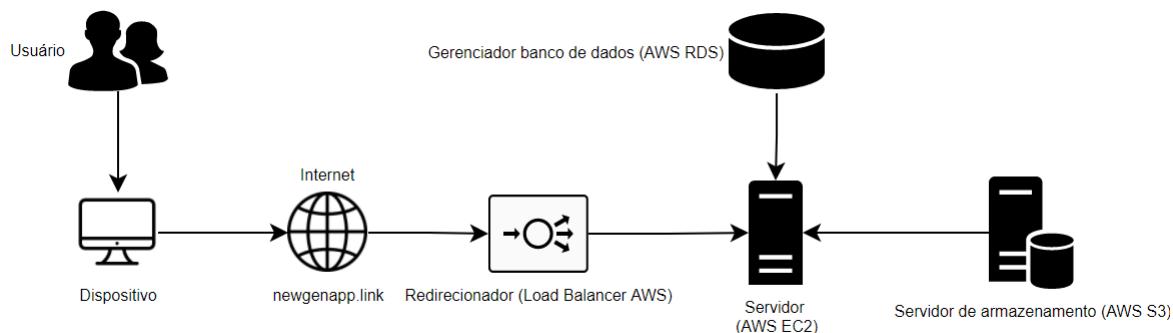
Amazon RDS: Serviço que deve facilitar a configuração, operação e implantação do MySQL no ambiente AWS. Deve possibilitar instanciar o banco de dados MySQL da aplicação.

3.8 Arquitetura do sistema

Os usuários devem ter acesso ao site a partir do seu dispositivo via web. A aplicação é disponibilizada e hospedada por meio de uma instância de servidor na Amazon Web Services (Amazon Elastic Cloud Computing - EC2) e utilização do banco de dados MySQL Server via AWS RDS.

O servidor alocado, realiza o processo de requisições de usuário, conforme [Figura 20](#):

Figura 20 – Servidor para processo de requisições de usuário.



Fonte: Os Autores

3.8.1 Configurações do Servidor EC2

O servidor Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - [ami-09e67e426f25ee0d7](#) (~~64 bits x86~~) / [ami-00d1ab6b335f217cf](#) (64 bits Arm) foi escolhido para instanciar a aplicação, e suas configurações são respectivamente:

Região: us-east-1 (Norte da Virgínia)

Quantidade de CPU: 1

Memória: 1GB

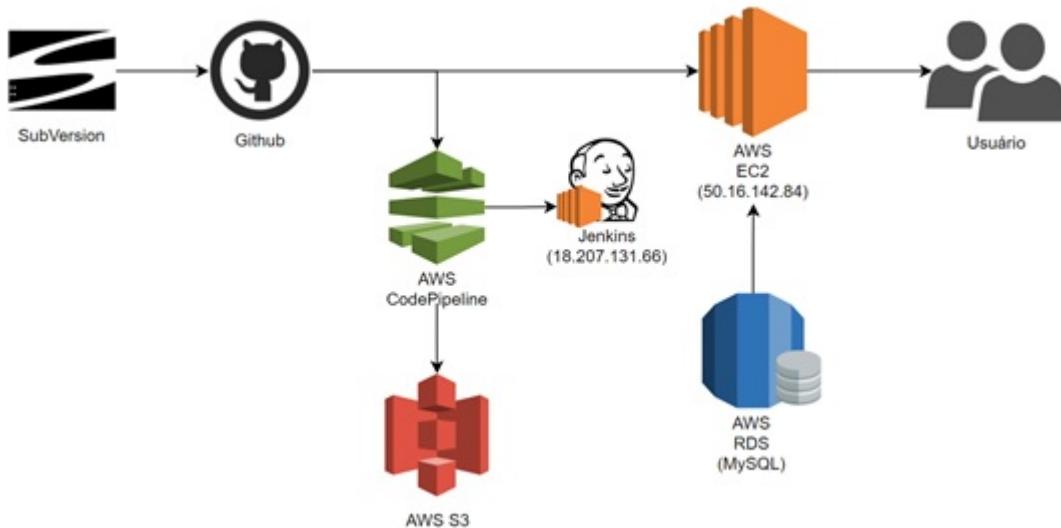
Armazenamento de disco: 8 GB

Processamento: 2.5 GHz

3.8.2 Comunicação Entre Ambientes

A aplicação deve seguir o modelo [Figura 21](#) para um desenvolvimento confiável durante todo o processo:

Figura 21 – Esquematização de fluxo para comunicação entre ambientes.



Fonte: Os Autores

3.8.3 Escalabilidade

A escalabilidade diz respeito a promover a expansão de nosso projeto sem comprometer sua qualidade e seu desempenho. Esse resultado deve ser alcançado de maneira a evitar um grande aumento de especificação e estabilizar o custo benefício a longo prazo.

Para garantir essa escalabilidade é necessário que haja uma utilização estratégica da mão de obra envolvida. Ou seja, alocá-los da melhor maneira possível, respeitando as individualidades e skills de cada um, atentando-se às necessidades do projeto.

Por consequência, a escalabilidade trará melhores condições de adaptação para os envolvidos, por incentivar a mão de obra na flexibilização de tarefas. O que implica na otimização de prazos internos, desburocratização de processos, trazendo uma maior quantidade de tarefas realizadas em um menor tempo.

Em termos gerais, em via de escalabilidade, pretendemos criar um aplicativo mobile para manter o projeto mais acessível na palma da mão de nosso usuário. Mesmo tendo em vista que uma aplicação web é sim versátil tanto em meios desktops quanto mobile recebendo o devido acesso a uma rede Wi-Fi, um app seria de maior comodidade e modernidade, para sustentar os avanços e necessidades do mercado. Esse aplicativo teria um custo fixo agregado à taxa de assinatura, a qual se submeteria a uma modalidade “plus” com o advento dessa facilitação.

Também pode-se agregar uma maior capacitação de nosso sistema a uma massa crescente de usuários, pois inicialmente, ele não terá capacidade para tal. Nossos sistemas de banco de dados, hosting e automatizações seriam otimizados para comportar esses usuários.

Além disso, o processo de personalização se tornaria algo mais automatizado, com funcionalidades mais acessíveis e dinâmicas para facilitar esse design exclusivo sendo gerado pelo próprio cliente. Assim, sua interface poderia tomar um rumo mais intuitivo.

3.9 Segurança da Informação Ferramentas de testes

~~Nosso projeto está sendo construído de forma que estará seguindo todos os padrões que a determina, legislação essa muito significativa e que preza pela privacidade dos dados. Esta legislação possui versões similares em outros países, todas voltadas a proteger os dados pessoais dos cidadãos~~ AWS Codepipeline com Jenkins : Uma instância de servidor EC2 é criada com a imagem do Jenkins e integrada com o serviço CodePipeline para realizar a Build no processo de integração contínua.

~~Sancionada em 2018, estabelece regras que as empresas devem seguir ao coletar, armazenar e utilizar os dados dos clientes, sendo passível de multas às empresas que não se adequarem à legislação~~ Python unittest : Framework de testes unitários para realizar a automação dos testes. Deve ser utilizado para a testagem do código back-end da aplicação.

~~Procuramos um ambiente que permita controlar todos os processos que ocorrem dentro dele, visando entregar um produto de qualidade para os clientes e com rastreabilidade. Esse sistema pode ser acessado por colaboradores em suas rotinas de trabalho, e para que~~

3.10 Modelagem do Banco de Dados

~~Para a modelagem do Banco de Dados, foram desenvolvidos o Modelo Entidade-Relacionamento (MER) e o Diagrama Entidade-Relacionamento (DER) da aplicação. Nestes, é descrito como é feita a estruturação do banco, com o MER apresentando as tabelas e suas conexões e o DER detalhando as colunas de cada tabela e suas particularidades, como tipo de dados e chaves primárias e estrangeiras.~~

~~Ambos podem ser consultados em Apêndice B.~~

3.11 Coding Convention

~~O coding convention é um conjunto de diretrizes que recomendam o estilo, as práticas e os métodos de se programar em determinada linguagem específica.~~

3.11.1 Back end

Para a linguagem de programação de backend (optada por Python) temos as convenções estabelecidas pelo PEP 8, que visa melhorar a legibilidade e padronização da codificação na linguagem citada. Foi escolhido como norma padrão de nossa codificação por ser a referência de mercado na programação em Python atualmente.

- **Indentação:** Na PEP8 temos como referência o uso de tabulações ou espaços em momentos específicos da codificação. Além de ser essencial realizar a indentação nessa linguagem, que exige espaços antes do código para que este rode fluidamente, sua aparência se torna mais limpida quando utilizamos com frequência essa prática, fazendo da sua leitura algo mais fácil.

O guia define por padrão a ~~segurança em banho de dados~~ indentação com 4 espaços. Se houver quebra de linha dentro de algum parênteses, colchetes ou chaves, se faz necessário que seu primeiro argumento esteja alinhado com o elemento da primeira linha.

Figura 22 – Exemplo de Identação em Código Simples.

```
# Válido
foo = funcao_qualquer(
    variavel1, variavel2,
    variavel3, variavel4)

# Válido
foo = funcao_qualquer(variavel1, variavel2,
                      variavel3, variavel4)

# Válido
def funcao(variavel1, variavel2,
           variavel3, variavel4):
    pass
```

Fonte: Os Autores

Figura 23 – Exemplo de Identação em Array.

```

minha_lista = [
    1, 2, 3
    4, 5, 6
    7, 8, 9
]

# Ou ainda
minha_lista = [
    1, 2, 3
    4, 5, 6
    7, 8, 9
]

```

Fonte: Os Autores

Em listas e tuplas, o mesmo processo deve ser seguido para realização de indentação.

- Limitação do tamanho de linhas: O limite de tamanho de linhas se dá por 79 colunas. em blocos de texto, o ideal é que não seja comprometida por acidentes ou ameaças intencionais, se passe de 72 caracteres. É recomendável também, caso se faça necessário quebrar uma linha no decorrer de um bloco de texto, utilizar uma contra-barra ou a utilização de parênteses.

Figura 24 – Exemplo de Tamanho de Linhas adequado.

```

with open("path/to/file/one/file1.txt") as file_one, \
    open("path/to/file/two/file2.txt") as file_two:
    file_two.write(file_one.read())

```

Fonte: Os Autores

- Imports: Para os imports, se estabelece a regra de fazê-los em linhas únicas, a não ser que os objetos ou funções sejam pertencentes a uma mesma classe. Neste último caso, tem-se uma exceção onde os imports são feitos em maior quantidade e em uma única linha.

Figura 25 – Exemplo de Imports Simples.

```
# Correto:
import os
import sys

# Incorreto:
import sys, os
```

Fonte: Os Autores

Figura 26 – Exemplo de Imports Múltiplos.

```
from subprocess import Popen, PIPE
```

Fonte: Os Autores

O layout também sofre algumas interferências pelas convenções, que podemos aplicar às demais linguagens também. Um código limpo é preciso garantir o bom funcionamento de três pilares: a integridade, disponibilidade e confiabilidade essencial para uma leitura mais fluida do código, linhas em branco para marcar a finalização do que podemos considerar os parágrafos da língua portuguesa em sua referência na programação. Detalhes simples como os citados podem gerar uma grande diferença para a visibilidade e fluidez do código.

Outro ponto a destacar com questão de segurança-

- **Nomenclatura:** A padronização já começa pela nomenclatura de tudo o que foi criado no código (variáveis, métodos, funções). Elas não devem ser genéricas e precisam indicar claramente ao que aquela ação se refere.
A convenção mais comum é a snake case, na qual as palavras são separadas por sublinhados e completamente em caixa baixa. Quando a variável é o, utilizado em nosso front-end, método de criptografar todas as informações contidas em uma página da web. Isso inclui informações sobre a própria página, bem como dados inseridos pelo usuário. Tudo isso de uso interno, ela se inicia com essa sublinhada, como seria em uma linguagem de encapsulamento o referente a uma variável privada. Para módulos e pacotes convém-se que é criptografado de uma forma que os interceptores melhor utilizar nomes curtos e em caixa baixa, podendo utilizar a sublinhada apenas para facilitar a legibilidade em módulos. Em classes, segue-se o padrão CamelCase, com a primeira letra de cada palavra em caixa alta.

3.11.2 Front End

Para o front end, optamos por seguir a ITCSS que estabelece algumas convenções de codificação para o CSS. Ela ajuda a organizar a arquitetura CSS, tornando sua visibilidade em camadas separadas que, quando juntas, fazem parte de uma pirâmide de elementos (Settings, Tools, Generic, Elements, Objects, Components, Trumps).

Essa pirâmide invertida segue a ideologia de especificidade. O topo tem códigos mais amplos, com pouca ou nenhuma especificidade. Ao contrário da ponta, onde se concentram códigos bem específicos.

- **Settings:** Trata-se das configurações básicas de nosso projeto, onde definimos variáveis globais (tanto para disposição de cores, espaçamentos, entre outros).

Figura 27 – Exemplo de Settings.

```
$primary-color: #0099ff  
$border-default: 1px solid #dcdcdc  
$gap-container: 1rem
```

Fonte: Os Autores

- **Tools:** Local de criação de funções e mixins para começar a construir um estilo e layout.

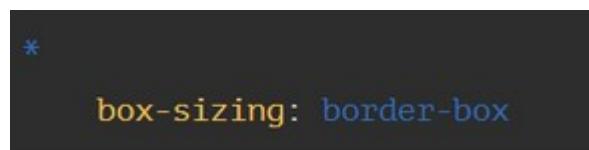
Figura 28 – Exemplo de Tools.

```
@mixin full()  
  height: 100vh  
  width: 100vw
```

Fonte: Os Autores

- **Generic:** Nessa camada da pirâmide começamos a desenvolver os códigos CSS. É onde passamos a informar regras mais genéricas (ou de menor especificidade), e onde podemos aplicar os resets.

Figura 29 – Exemplo de Generic.



Fonte: Os Autores

- **Elements:** Podemos definir seletores básicos e é a última camada onde se permite utilizar seletores por tag, nas demais o uso de id e classes é liberado, aqui não conseguem entender.

3.12 Modelagem

Para iniciar o desenvolvimento da aplicação, foi necessário a construção da modelagem do banco de dados, pois estes auxiliam na construção da aplicação e também oferecem um panorama geral do sistema antes de sua construção, podendo através deles prever

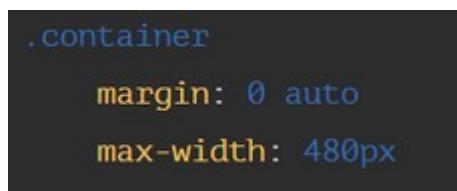
Figura 30 – Exemplo de Elements.



Fonte: Os Autores

- **Object:** Onde já iniciamos a implementação de pequenos objetos, somente permitindo a utilização de seletores de classes.

Figura 31 – Exemplo de Object.



Fonte: Os Autores

- **Components:** Como diz-se o próprio nome, começa-se a utilização de componentes de uma forma um pouco mais específica. É restrito para o uso exclusivo de classes.

Figura 32 – Exemplo de Components.

```
.btn-success
background: #00cc99
border-radius: 4px
color: #fdfdfd
```

Fonte: Os Autores

- **Trumps:** É a ponta da pirâmide e a última etapa de codificação. Determina os campos de maior especificidade, permitindo o uso de !important. Eles devem seguir a ideia de alocação de conteúdos onde sua utilização seja aplicada e não sobrescrita.

Figura 33 – Exemplo de Trumps.

```
.hidden
display: none !important
```

Fonte: Os Autores

3.12 Design Patterns

É de suma importância seguir uma estrutura de padronização Design Pattern no processo de desenvolvimento de um projeto. Ele precisa ser estruturado nesse padrão para ajudar a manter uma coerência no desenvolvimento e ajudar na solução de possíveis erros na arquitetura que poderiam gerar problemas no decorrer do desenvolvimento mesmo.

Para este projeto, foram desenvolvidos e História de Usuário Pensando nisso, os Designs Patterns que mais convergem com esse método de desenvolvimento e vão de encontro com a linguagem Python e do framework Django são Adapter, Strategy, Factory Method.

3.12.1 Modelagem do Banco de Dados

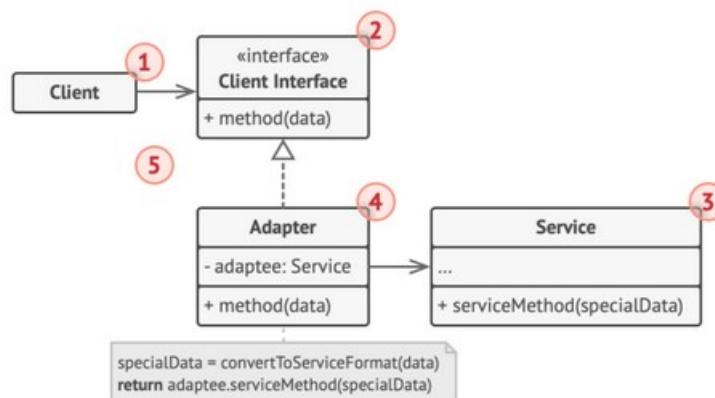
Para a modelagem do Banco de Dados, foram desenvolvidos o e

- **Adapter:** É um padrão do tipo estrutural de projeto que possibilita que uma classe com objetos de interfaces não compatíveis com o desenvolvimento, se encaixe nele com maior facilidade. Se, por exemplo, for necessário que uma biblioteca passe

dados para o formato JSON, porém o da aplicação. Nestes, código fonte não esteja sendo capaz de gerar esse arquivo no formato solicitado, o Adapter pode ajudar a solucionar seu problema. Como podemos observar na imagem, a interface do cliente não se adaptaria com excelência no servidor se não houvesse a classe Adapter para fazer seu intermédio.

A classe implementa a interface à qual a ela se adapta e encapsula o objeto de outra interface. Desta forma, através do modo adaptador, podemos gerenciar uma classe X, operada por um objeto que só conhece Y, pois passamos um método ao nosso socket para controlar um objeto que ele não conhece e não é deserto como adequado para ele. O mais importante é feita a estruturação do banco, com o apresentando as tabelas e suas conexões e o detalhando as colunas de cada tabela e suas particularidades, como tipo de dados e chaves primárias e estrangeiras que não precisamos mudar a classe de destino de forma alguma. O que será muito útil nos métodos e classes utilizados no back-end de nosso projeto, onde, a grande quantidade de dados de diferentes tipos poderá causar certa incompatibilidade entre classes.

Figura 34 – Exemplo de um Design Pattern Adapter



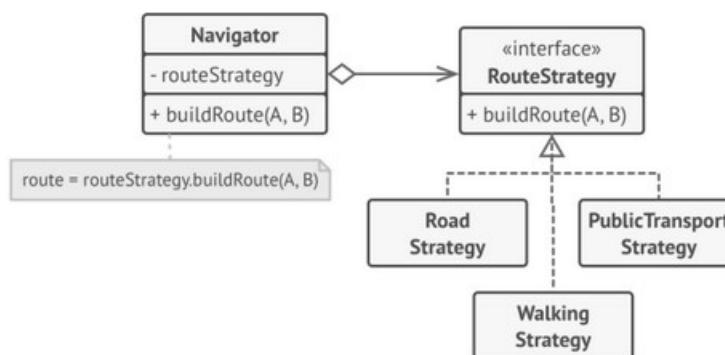
Fonte: Refactoring.guru (2021a)

- **Strategy:** Um padrão do tipo comportamental que define uma família de algoritmos, separando-os em classes e tornando seus objetos intercambiáveis. Ou seja, ele fornece opções de classes diferentes que exercem a mesma função de maneiras distintas, separando esses algoritmos em classes denominadas estratégias. Na imagem podemos ver, de maneira bem didática essa prática, já que a classe de “rota” é separada em classes de “de carro”, “transporte público” e “a pé”. Ou seja, ela nos fornece 3 tipos de classes distintas mas que exercem a mesma função: chegar a um destino comum, seja ele de carro, por transporte público ou a pé.

~~Ambos podem ser consultados em Deve-se ter em vista que o objetivo dessa Design~~

Pattern é definir uma família de algoritmos, encapsular cada um, e fazê-los intercambiáveis, assim permitindo que algoritmos variem independentemente entre clientes que os utilizam. Por tanto, seria cabível em classes do código onde haverá mudanças de acordo com a necessidade do cliente, pois quando agregada em nosso desenvolvimento, permitirá esse tipo de alteração automática. Assim, quando houver qualquer mudança no tipo de cálculo de estoque, por exemplo, não será necessário alterar a classe Navigator , apenas implementar uma nova interface (Route Strategy).

Figura 35 – Exemplo de um Design Pattern Strategy



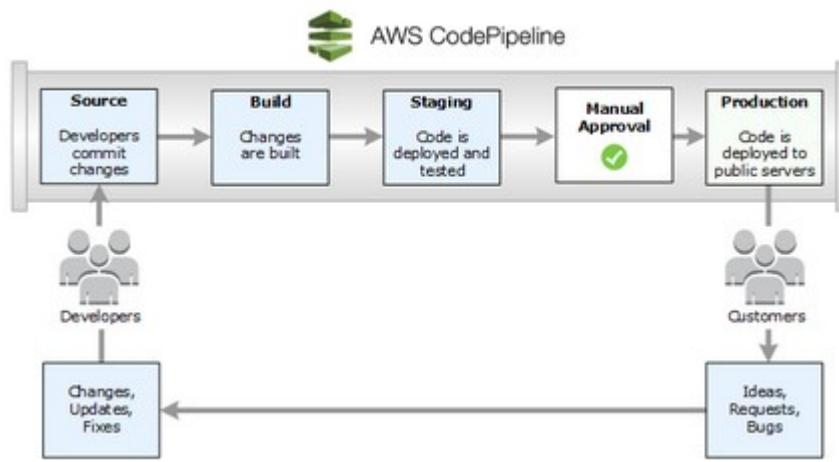
Fonte: Refactoring.guru (2021b)

- **Factory Method:** Pertencente ao grupo de design pattern criacional, é um padrão de projeto que possibilita a criação de uma interface para criar objetos em uma superclasse. Com um adendo, as subclasses têm direito a alterar objetos criados. É útil em casos onde não se sabe os tipos de dependências exatas de objetos que o código funciona.

3.13 Integração continua

Neste projeto, iremos utilizar um serviço de integração contínua que pode ser usado para visualizar e automatizar etapas necessárias para realizar o lançamento do sistema; o AWS CodePipeline, onde podemos automatizar o processo de lançamento a partir do repositório de origem no Github até a etapa de build no Jenkins, com testes. Ele serve como esteira de desenvolvimento, onde será possível testar para lançar progressivamente.

Figura 36 – Esquematização do Amazon Web Services CodePipeline.



Fonte: [AWS \(2021\)](#)

3.14 Product Backlog

Conforme a metodologia Scrum, devemos criar uma lista de priorização das funcionalidades que devem ser implementadas no produto, ou seja, houve necessidade do uso de Product Backlog, a qual possibilita a criação das sprints de 1 a 4 semanas e a geração do Kanban para ter um melhor gerenciamento das entregas contínuas.

3.14.1 Histórias de Usuário

A ferramenta Jira Software foi escolhida para ser utilizada no gerenciamento do desenvolvimento dos sprints, conforme o modelo ágil.

Conforme Figura 37, é possível visualizar uma esquematização de uma sprint exemplo:

Figura 37 – Esquema de Sprint na Ferramenta Jira

Fonte: Os Autores

3.15 Histórias de Usuário

As histórias de usuários descrevem como funcionará o projeto a ser desenvolvido, e como ele interage com os usuários do sistema, sendo criadas a partir dos requisitos da aplicação.

Uma vez que o projeto utiliza da abordagem Scrum como metodologia de desenvolvimento, as classes do sistema podem ser desenhadas com base nessas histórias, uma vez que ao utilizar dessa metodologia de desenvolvimento as classes e processos dentro do sistema refletem as histórias de usuários.

3.15.0.1 Épicos

3.15.1 Épicos

- **POC**

- Eu como administrador quero que o cliente se cadastre no meu sistema para que eu possa ter um maior controle de dados.

- **MVP**

- Eu como cliente quero me logar no sistema para que eu possa ter agilidade na hora de alugar um próximo espaço;
- Eu como cliente quero fazer uma reserva para poder selecionar o meu espaço previamente mesmo que eu não tenha o dinheiro no momento.

- **Produto Final**

- Eu como administrador quero visualizar relatórios e estatísticas para ter informações cruciais sobre o meu negócio;
- Eu como administrador, gerente ou recepcionista quero consultar uma agenda para descobrir quais são os aluguéis que foram ou estão marcados;
- Eu como gerente, quero uma interface de monitoramento para gerenciar os recursos disponíveis para locação;
- Eu como gerente quero uma interface de pacotes de locação para configurar os planos disponíveis para os clientes;
- Eu como gerente quero uma ferramenta de gerenciamento de usuários para caso seja necessário fazer alguma modificação manualmente nas informações de um usuário;
- Eu como cliente quero escolher um espaço para locação para que eu possa ter um lugar de qualidade dentro do que eu preciso para trabalhar;
- Eu como cliente quero conseguir consultar os recursos disponíveis para minha locação para saber se eu terei tudo o que preciso;
- Eu como cliente quero conseguir alugar recursos para minha reunião;
- Eu como administrador ou gerente quero que meu estoque seja gerenciado de forma automática para que eu possa focar estritamente na gestão de vendas, cortando os custos.

3.15.1.1 Histórias Divididas

3.15.2 Histórias Divididas

- **POC**

- Eu como gerente ou administrador quero ter um controle de dados de novos clientes que se cadastraram no sistema;
- Eu como gerente ou administrador quero que o cliente forneça informações necessárias no cadastro para facilitar as compras no negócio;
- Eu como cliente quero me cadastrar no sistema para poder utilizar as funções básicas de locação;
- Eu como cliente quero uma opção de me cadastrar pelo Google para maior agilidade.

- **MVP**

- Eu como cliente quero ter uma opção de esqueci minha senha para caso eu precise na entrada do sistema;
- Eu como gerente ou administrador quero que o usuário esteja logado no sistema na hora da compra para ter maior controle;
- Eu como gerente ou administrador quero que o cliente forneça uma fatia mínima na hora de realizar uma reserva como alguma garantia que ele virá;
- Eu como gerente ou administrador quero uma maneira de visualizar as reservas que estão disponíveis para que eu possa ter controle;
- Eu como cliente quero visualizar os planos de reserva por tempo para que eu possa escolher uma reserva no momento que eu preciso;
- Eu como cliente quero visualizar os planos de reserva por preços para que eu possa escolher um pacote que seja econômico para mim;
- Eu como cliente quero visualizar os planos de reserva por recursos para que eu possa escolher um pacote de reserva que tenha tudo o que eu precise;
- Eu como cliente quero que a reserva seja confirmada pelo sistema para que eu não tenha que ficar aguardando uma resposta do suporte da empresa para concluir minha reserva;
- Eu como cliente quero que tenha a opção de suporte para garantir que eu não tenha dúvidas ao realizar a minha reserva.

- **Produto Final**

- Eu como administrador quero visualizar relatórios e estatísticas para ter informações cruciais sobre o meu negócio;
- Eu como administrador, gerente ou recepcionista quero consultar uma agenda para descobrir quais são os aluguéis que foram ou estão marcados;
- Eu como gerente quero uma interface de monitoramento para gerenciar os recursos disponíveis para locação;
- Eu como gerente quero uma interface de pacotes de locação para configurar os planos disponíveis para os clientes;
- Eu como gerente quero uma ferramenta de gerenciamento de usuários para caso seja necessário fazer alguma modificação manualmente nas informações de um usuário.

3.16 ~~Definições de Escopo~~

Neste tópico abordaremos os casos de uso da aplicação (forma de descrever uma funcionalidade do sistema); diagrama de requisitos (identifica as funcionalidades a serem implementadas); product backlog (lista com todos requisitos); e definição de entregas (quais funcionalidades estarão disponíveis nas principais entregas).

Neste tópico, serão descritos os casos de uso da aplicação.

3.15.1 Casos de Uso

Os casos de uso foram construídos para estabelecer quais as necessidades dos usuários a serem abordadas no sistema, pensando num uso diário.

Caso de Uso do Sistema na Visão de um Administrador

Caso de Uso do Sistema na Visão de um Gerente

Caso de Uso do Sistema na Visão de um Usuário

3.16 Análise de requisitos

A análise de requisitos do projeto consiste na visão da equipe em relação ao projeto como um todo, incluindo os problemas a serem solucionados e suas funcionalidades. A análise de requisitos é vital para o desenvolvimento do sistema, determinando o sucesso ou fracasso do projeto.

Os requisitos colhidos devem ser quantitativos, detalhados e relevantes para o projeto, pois eles fornecerão a referência para validar o produto final, estabelecerão o acordo entre cliente e fornecedor sobre o que o software fará e consequentemente reduzirão os custos de desenvolvimento, pois requisitos mal definidos implicam em retrabalho.

Dentro ~~deste capítulo será feita a listagem~~ desta seção, é feita a relação de requisitos funcionais e não funcionais do produto.

3.16.1 Requisitos Funcionais

Os requisitos funcionais são os responsáveis por descrever as funções que serão implementadas na aplicação. Por se tratar de um desenvolvimento ágil, cada um desses requisitos deve ser tratado como uma funcionalidade a ser atendida durante o desenvolvimento do projeto.

Os quadros **Quadro 3** e **Quadro 4** apresentam uma listagem relação dos requisitos funcionais, separados através de um código de identificação RF e um sequencial para facilitação de referências futuras.

Será listado também o nome e uma breve descrição da funcionalidade a ser abordada neste requisito.

Listagem-**Quadro 3 – Relação de requisitos funcionais**

Código	Nome	Descrição
RF01	Forma de pagamento do sistema O sistema poderá ser adquirido em formato de assinatura. Podendo ser com renovação mensal, semestral ou anual. RF02 Gerenciamento de usuários	O sistema deve permitir a criação, leitura e edição de usuários.
RF03 <u>RF02</u>	Autenticar usuários	O acesso ao sistema deve ser projetado protegido por mecanismo de autenticação das informações do usuário.
RF04 <u>RF03</u>	Gerenciar salas e recursos	O sistema deve permitir a criação, leitura, edição e exclusão de espaços e recursos (datashow, dvd, quadro branco, etc.).
RF05 <u>RF04</u>	Autenticação de entrada e saída	O sistema enviará um QR code ao usuário (locatário) recebe um QR code até 2 horas antes do horário da reserva para validar sua para validação de entrada e saída do espaço alocado, que será validado pela recepcionista do local.
RF06 <u>RF05</u>	Exibir disponibilidade de espaço	O sistema deverá apresentar aos usuários status de espaços, datas e horários, além dos tipos de espaços disponíveis para locação.
RF07 <u>RF06</u>	Cancelar locação	O sistema deve ter a opção de cancelamento da locação, o período de cancelamento será de responsabilidade do cliente (locador).
RF08 <u>RF07</u>	Lista de espera	O sistema dará uma opção para o locatário entrar na lista de espera, caso a data, hora ou espaço desejado não estajam disponíveis.
RF09 <u>RF08</u>	Confirmação de reserva	O sistema deverá enviar um e-mail de confirmação de locação após aprovação de pagamento da mesma, com as informações da locação.
RF10 <u>RF09</u>	Transferência de responsabilidade	O sistema terá a opção de transferência da responsabilidade pela da locação pelo próprio host. Onde o usuário (locatário) indica a pessoa para quem essa responsabilidade está sendo atribuída e este deve confirmar a atribuição.
RF11 <u>RF10</u>	Alerta de tempo limite	O sistema mostrará um pop-up quando faltar 15 minutos para o fim do tempo que foi reservado.
RF12 <u>RF11</u>	Confirmação de alerta de tempo limite	O usuário poderá apenas confirmar no pop-up ou solicitar mais horas para dar continuidade.
RF13 <u>RF12</u>	Solicitação de tempo extra	Caso Se o horário seguinte esteja estiver disponível, o usuário pode continuar na mesma sala, caso contrário, ele terá opção de selecionar outra sala para dar continuidade à sua locação.
RF14 <u>RF13</u>	Formulário de Feedback	Após finalização do processo de locação, serão disponibilizados dois tipos de formulários, um para o locatário, para avaliar sua experiência, e outro para a recepcionista, onde será avaliado se o locatário manteve o espaço dentro dos conformes

Quadro 4 – Continuação da listagem de requisitos funcionais

Código	Nome	Descrição
RF18 <u>RF17</u>	Recebimento de pagamento	O sistema terá integração com mercadoPago para que seja confirmado o recebimento de pagamento do locatário.
RF19 <u>RF18</u>	Tela do Administrador	O sistema deve disponibilizar informações sobre seus espaços, como status de cada um, deve conter a opção de cadastro de usuários (funcionários internos), recursos e salas, além de consultas de relatórios.
RF20 <u>RF19</u>	Tela de Recepcionista	O Sistema deve apresentar a opção de verificar os status de locação do dia atual e dos próximos 30 dias, opção de validação de pagamento, validação de QR code do locatário.
RF21 <u>RF20</u>	Tela do Locatário	O sistema deve apresentar as locações que o usuário possui nos próximos 15 dias, caso haja. Também haverá: status de aprovação de locação; campos para informar o local que quer fazer a locação, tipo de espaço, calendário mostrando datas e horários com informação de status (disponível, indisponível), em caso de disponíveis, ter a opção de entrar na Lista de espera.
RF22 <u>RF21</u>	Tela do Gerente	O gerente será responsável pelo gerenciamento da reserva (confirmação, edição e cancelamento) a partir da confirmação do pagamento, terá acesso a agenda do dia e dos próximos 30 dias.

Fonte: Os Autores

3.16.2 Requisitos ~~não funcionais~~Não Funcionais

Requisitos não funcionais são aqueles que descrevem como a aplicação deve funcionar, com condições mínimas de funcionamento e restrições que o sistema deve conter.

Dentro do **Quadro 5** os requisitos não funcionais foram classificados a partir de um código sequencial com prefixo RNF, e cada requisito possui um nome e uma breve descrição.

Quadro 5 – Listagem de requisitos não funcionais.

Código	Nome	Descrição
RNF01	Protocolos de segurança	O sistema deve utilizar o protocolo <i>Hypertext Transfer Protocol (HTTPS)</i> para fazer a transferência segura dos dados inseridos no sistema.
RNF02	Backup de dados	O sistema deve garantir o acesso e a integridade dos dados, por meio de BackUps rotineiros.
RNF03	Responsividade do Layout	O sistema deve ter um layout adaptável ao dispositivo usado na visualização.
RNF04	Criptografia de senhas	O sistema deve armazenar senhas com criptografia adequada.
RNF05	Performance	O sistema deve ser construído a fim de garantir sua possível expansão.
RNF06	Disponibilidade	O sistema deve estar disponível aos usuários sem interrupções.

Fonte: Os Autores

Product Backlog

3.17 Regras de negócio

Conforme a metodologia Serum, devemos criar uma lista de priorização das funcionalidades que devem ser implementadas no produto, ou seja, houve necessidade do uso de Product Backlog, a qual possibilita a criação das sprints de 1 a 4 semanas e a geração do Kanban para ter um melhor gerenciamento das entregas contínuas. A partir das regras de negócio que são especificadas as particularidades das funcionalidades a serem desenvolvidas. No processo de desenvolvimento de sistemas, a regra de negócio visa detalhar as particularidades de um software.

A ferramenta Jira Software foi escolhida para ser utilizada no gerenciamento do desenvolvimento dos sprints, conforme o modelo ágil. O Quadro 6 apresenta uma listagem dos requisitos funcionais, separados através de um código de identificação RN e um sequencial para facilitação de referências futuras.

Conforme, é possível visualizar uma esquematização de uma sprint exemplo:

Esquema de Sprint na Ferramenta Jira

3.17.1 Definição de Entregas

Após mapear e descrever todas as funcionalidades que nossa aplicação terá, foi definido quais processos iriam ser possíveis de serem desenvolvidos para as entregas planejadas, para alinhamento do andamento do projeto. Será listado também o nome uma breve descrição da regra a ser seguida e os requisitos funcionais relacionados.

O cronograma a ser utilizado para as entregas das funcionalidades, e em qual entrega cada uma estará presente (, , ou Produto Oficial/Final).

Escoço do projeto

Quadro 6 – Listagem de regras de negócios.

Funcional	Nodo Código	Descrição	Produto final Relacionado
Cadastro de usuário RN01	X Obrigatoriedade de login para acesso ao sistema	Login de Para acessar o sistema será necessário a realização de login do usuário em caso de primeiro acesso, é necessária a realização do cadastro.	X RF02, RF03
Reserva RN02	Perfis de usuários	X O Sistema terá 4 perfis diferentes: Administrador (locador), Usuario comum (locatario), Gerente, recepcionista.	RF19, RF20, RF21, RF22
Aluguel RN03	Telas por perfil	Cada perfil possui um tipo de fluxo de telas personalizadas.	X RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF19, RF20, RF21, RF22
Lista de Espera RN04	Possibilidade de reserva	Será possível realizar uma reserva do espaço, data e hora que o usuário deseja alugar. Para a reserva, será cobrada uma taxa a cargo do Locador sobre o valor, podendo realizar o pagamento total dentro de 24hrs após a reserva.	X RF09
Autorização de Entrada RN05	Realizar locação	A locação será realizada somente após confirmação de pagamento da mesma.	X RF6, RF07, RF08, RF09
Lotação de Máquinas RN06	Forma de pagamento Empresa Contratante	A forma de pagamento da empresa contrarrente do sistema para a NewGen será por boleto ou cartão de crédito.	X RF01
Relatórios Periódicos RN07	validação de QR code	QR code para validação de entrada e saída do locatário.	X RF05
Sistema de Fidelidade RN08	Lista de espera	O sistema deve ser responsivo, se adequando ao dispositivo de origem do acesso.	X RF22
Transferência de Responsabilidade RN09	Pagamento do locatário	O locatário terá a opção de pagamento via boleto, PIX ou cartão de crédito.	X RF18
Controle de Estoque X-			

Fonte: Os Autores

4

Viabilidade Financeira e Manutenção da Aplicação

O projeto New Gen busca ser um diferencial de mercado no ramo de coworkings e, para isso, ele se baseia em um sistema de mensalidade. A proposta desse método é trazer flexibilidade para o consumidor do sistema, onde ele não se verá preso em uma anuidade e poderá ter sua mensalidade cancelada a qualquer momento. Devido a esse diferencial, torna-se um serviço viável financeiramente.

Além disso, não dependerá de grandes licenças para seu funcionamento convencional e, se necessário, pode ter seu plano incrementado de acordo com a preferência do cliente.

O intuito é ser o mais flexível possível. Sua interface, seu layout e, como não poderia deixar de ser, seu custo também são cambiáveis de acordo com as necessidades do consumidor.

3.1 Viabilidade Financeira

Após o usuário adquirir o serviço de gerenciamento da New Gen, ele terá direito a optar por um plano que mais se enquadra em sua realidade no coworking e, caso haja interesse do mesmo, realizar as devidas mudanças de plano, adicionando mais funcionalidades ou adquirindo mais licenças que se fazem essenciais em seu negócio.

Nosso projeto inclui custos tais como a assinatura com o MercadoPago e o sistema hosting da aplicação web. Tais taxas, já são agregadas no processo de assinatura mensal ofertada.

O que não engloba (ou seja, fica a encargo do cliente) as responsabilidades pelo custeamento de estoque, reposição de qualquer material do seu coworking, depredação de seu estabelecimento, entre outros casos onde não há correlação direta com o sistema da New Gen. Além disso, não arcamos tampouco com falhas provocadas pelo usuário, tendo um custo a parte para manutenção. Em situações onde a falha é em nosso desenvolvimento, o cliente se ausenta do financiamento do concerto.

Um preço médio que o projeto cobraria por assinatura seria R\$ 550,00. Levando em consideração que empresas de coworking geralmente cobram mais da metade do valor citado por apenas um aluguel, seria viável (e até muito acessível) o custo de nosso projeto mensal. Desse valor, certamente por volta de 5% seria destinado às taxas de aplicações terceiras de pagamento (MercadoPago), que cobram em torno dessa porcentagem (que

pode variar conforme a forma de pagamento, valor da compra, entre outros) por compra gerada mais sua taxa de parcelamento (caso haja).

Deve-se também levar em consideração o fato do NewGen ser um sistema personalizável, o que garante exclusividade a cada cliente e sua preservação de identidade de marca. Tal fato é um dos diferenciais que agregam maior valorização ao serviço. Isso porque, a identidade visual no marketing pode causar tanto impacto e fidelização de clientes a ponto de fazê-los optar pela empresa apenas pelo fato de ela estar concentrada nesse valor.

As assinaturas, como mencionado, serão estabelecidas de forma mensal, podendo haver a solicitação de cancelamento em qualquer momento. Um detalhe importante a ser exaltado é que, uma vez realizado o pagamento de uma mensalidade, seu valor não será reembolsado em caso de cancelamento sem justa causa. Se não for apresentado qualquer dano por parte do sistema New Gen, o valor desembolsado na mensalidade seguirá sem reembolso, mesmo que o cancelamento seja realizado antes do término do período ofertado.

Com variante de gastos, as mensalidades são cobradas a partir de R\$ 109,90, em um plano mais simples, com menos funcionalidades e/ou facilidades de serviço. Podem chegar até R\$ 999,90.

Tendo em vista a quantidade de coworkings por metro quadrado, vide a cidade de São Paulo, território pioneiro dessa prática atualmente no Brasil, pode-se entender que, com uma boa campanha de marketing, podemos chegar a captar metade dos administradores de coworking vigentes, apenas na Avenida Paulista. Localidade esta que comporta centro de coworking desde os menos conhecidos até os mais renomados, tais como GoWork, WeWork, Y4 Coworking, WorkInOffice, 2Work, entre outros.

3.2 ~~Manutenibilidade Segurança~~ da aplicação Informação

~~Durante a criação da aplicação, será importante utilizar ferramentas que auxiliem no desenvolvimento do sistema, enquanto garantem sua qualidade.~~

~~Padrões como Coding Convention e Design Patterns foram estudados e definidos pela equipe, assim como a ferramenta de , responsáveis pela qualidade final da aplicação Nossos projeto está sendo construído de forma que estará seguindo todos os padrões que a Lei Geral de Proteção de Dados (LGPD) determina, legislação essa muito significativa e que preza pela privacidade dos dados. Esta legislação possui versões similares em outros países, todas voltadas a proteger os dados pessoais dos cidadãos.~~

3.2.1 ~~Coding Convention~~

~~O coding convention é um conjunto de diretrizes que recomendam o estilo, as práticas e os métodos de se programar em determinada linguagem específica Sancionada~~

em 2018, estabelece regras que as empresas devem seguir ao coletar, armazenar e utilizar os dados dos clientes, sendo passível de multas às empresas que não se adequarem à legislação.

3.2.0.1 Back-end

Para a linguagem de programação de backend (optada por) temos as convenções estabelecidas pelo , que visa melhorar a legibilidade e padronização da codificação na linguagem citada. Foi escolhido como norma padrão de nossa codificação por ser Procuramos um ambiente que permita controlar todos os processos que ocorrem dentro dele, visando entregar um produto de qualidade para os clientes e com rastreabilidade. Esse sistema pode ser acessado por colaboradores em suas rotinas de trabalho, e para que a referência de mercado na programação em Python atualmente.

Indentação: Na PEP8 temos como referência o uso de tabulações ou espaços em momentos específicos da codificação segurança em banco de dados não seja comprometida por acidentes ou ameaças intencionais, é preciso garantir o bom funcionamento de três pilares: a integridade, disponibilidade e confiabilidade. Além de ser essencial realizar a indentação nessa linguagem, que exige espaços antes do código para que este rode fluidamente, sua aparência se torna mais límpida quando utilizamos com frequência essa prática, fazendo da sua leitura algo mais fácil.

O guia define por padrão a indentação com 4 espaços. Se houver quebra de linha dentro de algum parênteses, colchetes ou chaves, se faz necessário que seu primeiro argumento esteja alinhado com o elemento da primeira linha.

Exemplo de Indentação em Código Simples.

Exemplo de Indentação em Array.

Em listas e tuplas, o mesmo processo deve ser seguido para realização de indentação.

Limitação do tamanho de linhas: O limite de tamanho de linhas se dá por 79 colunas. em blocos de texto, o ideal Outro ponto a destacar com questão de segurança é que o HTTPS, utilizado em nosso front-end, método de criptografar todas as informações contidas em uma página da web. Isso inclui informações sobre a própria página, bem como dados inseridos pelo usuário. Tudo isso é criptografado de uma forma que os interceptores não se passem de 72 caracteres. É recomendável também, caso se faça necessário quebrar uma linha no decorrer de um bloco de texto, utilizar uma contra-barra ou a utilização de parênteses consigam entender.

Exemplo de Tamanho de Linhas adequado.

3.3 Mudanças ao longo do projeto

Imports: Para os imports, se estabelece a regra de fazê-los em linhas únicas, a não ser que os objetos ou funções sejam pertencentes a uma mesma classe. Neste último caso, tem-se uma exceção onde os imports Nesta seção, são feitos em maior quantidade e em uma única linha registradas as evoluções e tomadas de decisões que a equipe do NewGen tomou ao longo do percurso de desenvolvimento do Software.

Exemplo de Imports Simples.

Exemplo de Imports Múltiplos.

O layout também sofre algumas interferências pelas convenções, que podemos aplicar às demais linguagens também. Um código limpo é essencial para uma leitura mais fluida do código, linhas em branco para marcar a finalização do que podemos considerar os parágrafos da língua portuguesa em sua referência na programação. Detalhes simples como os citados podem gerar uma grande diferença para a visibilidade e fluidez do código.

Nomenclatura: A padronização já começa pela nomenclatura de tudo o que foi criado no código (variáveis, métodos, funções). Elas

3.3.1 White-Label para SaaS

A aplicação web desenvolvida pela NewGen inicialmente tinha a proposta de ter o seu modelo de comercialização como um WhiteLabel, não devem ser genéricas e precisam indicar claramente ao que aquela ação se refere como um SaaS.

A convenção mais comum é a, na qual as palavras são separadas por sublinhados e completamente em caixa baixa. Quando a variável é de uso interno, ela se inicia com essa sublinhada, como seria em uma linguagem de encapsulamento o referente a uma variável privada. Para módulos e pacotes convém-se que é melhor utilizar nomes curtos e em caixa baixa, podendo utilizar a sublinhada apenas para facilitar a legibilidade em módulos. Em classes, segue-se o padrão CamelCase, com a primeira letra de cada palavra em caixa alta. Ideia era que o cliente do sistema pudesse realizar uma compra única e adaptasse o produto da NewGen dentro de sua infraestrutura. O Software viria pronto para ser implementado, haveria uma área de customização para o cliente colocar o seu logo e deixar o site ao seu gosto, mas seria possível que ele modificasse completamente o site da maneira que preferisse.

Após o prosseguir do projeto, com as análises de viabilidade financeira (presentes em 3.1) e algumas discussões de infraestrutura, foi optado por adaptar a aplicação para um modelo SaaS.

3.3.1.1 Front End

Para o front end, optamos por seguir a que estabelece algumas convenções de codificação para o CSS. Ela ajuda a organizar a arquitetura CSS, tornando sua visibilidade em camadas separadas que, quando juntas, fazem parte de uma pirâmide de elementos (Settings, Tools, Generic, Elements, Objects, Components, Trumps).

Essa pirâmide invertida segue a ideologia de especificidade. O topo tem códigos mais amplos, com pouca ou nenhuma especificidade. Ao contrário da ponta, onde se concentram códigos bem específicos.

Settings: Trata-se das configurações básicas de nosso projeto, onde definimos variáveis globais (tanto para disposição de cores, espaçamentos, entre outros).

Exemplo de Settings.

Tools: Local de criação de funções e mixins para começar a construir um estilo e layout.

Exemplo de Tools.

Generic: Nessa camada da pirâmide começamos a desenvolver os códigos CSS. É onde passamos a informar regras mais genéricas (ou de menor especificidade), e onde podemos aplicar os resets.

Exemplo de Generic.

Elements: Podemos definir seletores básicos. Nele, o cliente entra em contato com o portal online da NewGen, cujas opções de assinatura dão a escolha de um sistema que atenda as suas necessidades. Haverá um suporte para que as personalizações do cliente sejam feitas, dependendo do pacote de assinatura escolhido. A partir da realização da assinatura e da personalização de suas necessidades, o cliente terá um link dentro do domínio da NewGen, ou um domínio próprio cadastrado pela NewGen para ele (ainda a ser estudado) e é a última camada onde se permite utilizar seletores por tag, nas demais o uso de id e classes é liberado, aqui não.

Exemplo de Elements.

Object: Onde já iniciamos a implementação de pequenos objetos, somente permitindo a utilização de seletores de classes. Poderá começar a rodar a instância digital de seu sistema coworking.

Exemplo de Object.

3.3.2 Alterações na lógica do banco

3.3.2.1 Uso do ORM

Components: Como diz-se Durante a implementação inicial do banco de dados, foi definido o próprio nome, começa-se a utilização de componentes de uma forma um pouco mais específica. É restrito para o uso exclusivo de classes.

Exemplo de Components.

Trumps: É a ponta da pirâmide e a última etapa de codificação. Determina os campos de maior especificidade, permitindo o uso de !important. Eles devem seguir a ideia de alocação de conteúdos onde sua utilização seja aplicada e não sobreescrita. Um ORM para poder se adaptar a bancos pré-existentes do cliente. Após estudos mais detalhados e discussões entre a equipe, essa decisão foi trocada em favor do uso de apenas um banco de dados, o MySQL, por questões de facilidade na manutenção.

Exemplo de Trumps.

3.3.3 Design Patterns

É de suma importância seguir uma estrutura de padronização Design Pattern no processo de desenvolvimento de um projeto. Ele precisa ser estruturado nesse padrão para ajudar a manter uma coerência no desenvolvimento e ajudar na solução. Para clientes que possuam bancos pré-existentes, será dada a opção de possíveis problemas no decorrer do mesmo.

Pensando nisso, os Design Patterns que mais convergem com esse método de desenvolvimento e vão de encontro com a linguagem e do framework são Adapter, Strategy, Factory Method.

Adapter: É um padrão do tipo estrutural de projeto que possibilita que uma classe com objetos de interfaces não compatíveis com o desenvolvimento, se encaixe nele com maior facilidade. Se, por exemplo, for necessário que uma biblioteca passe importação dos dados para o formato JSON, porém o código fonte não esteja sendo capaz de gerar esse arquivo no formato solicitado, o Adapter pode ajudar a solucionar seu problema. Como podemos observar na imagem, a interface do cliente não se adaptaria com excelência no servidor se não houvesse a classe Adapter para fazer seu intermédio. A classe implementa a interface à qual a ela se adapta e encapsula o objeto de outra interface. Desta forma, através do modo adaptador, podemos gerenciar uma classe X, operada por um objeto que só conhece Y, pois passamos um método ao nosso socket para controlar um objeto que ele não conhece e não é adequado para ele. O mais importante é que não precisamos mudar a classe de destino de forma alguma. O que será muito útil nos métodos e classes utilizados no back-end de nosso projeto, onde, a grande quantidade

~~de dados de diferentes tipos poderá causar certa incompatibilidade entre classes. banco da aplicação, mediante estudo da estrutura do banco e das informações contidas dentro dele para garantir compatibilidade.~~

LABEL NECESSÁRIA

3.3.2.1 Adição de novas colunas

~~Strategy:~~ Um padrão do tipo comportamental que define uma família de algoritmos, separando-os em classes e tornando seus objetos intercambiáveis. Ou seja, ele fornece opções de classes diferentes que exercem a mesma função de maneiras distintas, separando esses algoritmos em classes denominadas estratégias. Na imagem podemos ver, de maneira bem didática essa prática, já que a classe de “rota” é separada em classes de “de carro”, “transporte público”. Após reunião com os professores e apresentação do MER e “DER, a questão de registro de alterações nas tabelas foi levantada. Portanto, para as tabelas ‘tipoEspaco’, ‘espaco’ e ‘equipamento’, foram adicionadas novas colunas para identificar o funcionário responsável pela última edição e a pé”. Ou seja, ela nos fornece 3 tipos de classes distintas mas que exercem a mesma função: chegar a um destino comum, seja ele de carro, por transporte público ou a pé. Deve-se ter em vista que o objetivo dessa Design Pattern é definir uma família de algoritmos, encapsular cada um, e fazê-los intercambiáveis, assim permitindo que algoritmos variem independentemente entre clientes que os utilizam. Por tanto, seria cabível em classes do código onde haverá mudanças de acordo com a necessidade do cliente, pois quando agregada em nosso desenvolvimento, permitirá esse tipo de alteração automática. Assim, quando houver qualquer data desta.

~~O novo formato das tabelas pode ser consultado no Apêndice C.~~

3.3.2.2 Estratégia de modelagem

~~Após a reunião do dia 28 de Julho, disponível no Apêndice E, ficou acordado que o produto seria um SaaS que o cliente entraria em um site fornecido pela NewGen para realizar a assinatura e cadastro de seu sistema customizado. No entanto, tal mudança no tipo de cálculo de estoque, por exemplo, não será afeta a maneira que o banco de dados estava estruturado até este dia, pois é necessário alterar a classe Navigator (vida a imagem), apenas implementar uma nova interface (Route Strategy). levar em conta dos dados das tabelas agora não serem de um único Coworking especificamente.~~

~~LABEL NECESSÁRIA~~ A estratégia de modelagem que se seguiria se tornou confusa dentro do grupo. Como já havia ocorrido falha de comunicação sobre a definição do produto uma vez, o integrante Gabriel P. criou um estudo (Estudo informal sobre migração do banco) a fins de comunicar quais são as possibilidades de realizar tal modelagem e ao mesmo tempo conferir que as informações estavam pareando. Foram estudadas 3 escolhas:

- **Factory Method:** Pertencente ao grupo de design pattern criacional, é Adicionar uma chave estrangeira em todas as tabelas do banco, referentes a uma tabela principal que contém os dados das empresas de coworking assinantes;
- Criar um banco de dados único para cada instância de coworking;
- Analisar a troca para um padrão de projeto que possibilita a criação de uma interface para criar objetos em uma superclasse. Com um adendo, as subclasses têm direito a alterar objetos criados. É útil em casos onde banco de dados não se sabe os tipos de dependências exatas de objetos que o código funciona—relacional, como o MongoDB por exemplo.

3.4 Ferramentas de testes

AWS Codepipeline com Jenkins —Uma instância de servidor EC2 é criada com a imagem do Jenkins e integrada com o serviço CodePipeline para realizar a Build no processo de integração contínua. No momento, o grupo escolheu a primeira opção, porém ainda está aberto o debate para outras opções, conforme futuras implementações.

Python unittest —Framework de testes unitários para realizar a automação dos testes. Deve ser utilizado para a testagem do código back-end da aplicação.

4 Considerações Finais

4.0.1 Integração continua

Neste projeto, iremos utilizar um serviço de integração contínua que pode ser usado para modelar, visualizar e automatizar etapas necessárias para realizar o lançamento do sistema. Trata-se do CodePipeline, na qual podemos automatizar o processo de lançamento a partir do repositório de origem no Github até a etapa de build no Jenkins, com testes. Ele serve como esteira de desenvolvimento, onde será possível testar para lançar progressivamente.

Esquematização do Amazon Web Services CodePipeline.

4.1 Tecnologia utilizadas

É uma linguagem de programação de alto nível, robusta, dinâmica, interpretada, multiplataforma, com sintaxe de fácil compreensão e de acordo com o paradigma da programação orientada a objetos. O Python possui uma vasta variedade de bibliotecas e grande comunidade de desenvolvedores. Sendo assim, possibilitando disponibilizarmos De todas as propostas estudadas pela equipe, a idéia de uma aplicação sofisticada. Deve ser a linguagem responsável pelo back-end da aplicação. voltada para oferecer suporte a um espaço de Coworking foi a mais aceita, tanto entre a equipe quanto pelos professores. No entanto, apenas o desenvolvimento da ideia se mostrou como uma tarefa fácil.

Para o desenvolvimento, a equipe escolheu linguagens atuais e relevantes ao mercado de TI, aproveitando o desenvolvimento do projeto como uma forma de se obter maior conhecimento prático destas. Sendo estas linguagens conhecidas por membros do grupo, o compartilhamento de conhecimento foi mais fluido, auxiliado por pesquisas e videoaulas on-line.

É um framework que deve ser utilizado em conjunto com a linguagem Python para facilitar no desenvolvimento da aplicação, a qual disponibiliza um conjunto de bibliotecas para ser efetuada a criação de base da aplicação. Portanto, permite otimizar os recursos para facilitar na arquitetura e configuração do sistema. O Django deverá utilizar o padrão MTV para a aplicação. As camadas são respectivamente:

1. **Model** — Para realizar a interação com o banco de dados.
2. **Template** — Onde realiza a renderização dos dados para interação do usuário com o site.

3. **View** — É a camada responsável pelas regras de negócios.

No entanto, houve problemas. O maior deles, a dificuldade na comunicação e compreensão entre os membros da equipe, pode ser atribuída à situação de pandemia da COVID-19, visto que todas as reuniões precisavam ser feitas através de videochamadas, e não possuíam o mesmo detalhamento de reuniões presenciais. Muitas mudanças foram feitas em último momento, e a dificuldade de se encontrar suporte para erros nos sistemas também afetou as entregas.

Modelo MTV no Django

Javascript Dado as dificuldades encontradas, a equipe ainda vê os resultados atuais como positivos: um website que, mesmo possuindo o ocasional bug, é uma linguagem de programação de alto nível, dinâmica, permite realizar o controle de multimídia e animação de imagens. Esta linguagem, dedicada à web é a terceira camada para o desenvolvimento de aplicações, juntamente com o HTML (linguagem de marcação) e CSS (Linguagem para regras de estilo).

Bootstrap é um framework front-end que fornece componentes preparados para utilização na aplicação. Sendo possível criar e personalizar site web responsivo, com a estrutura de HTML e CSS preparada para utilização. Com isto, permite que o site possua um aspecto visual elegante por disponibilizar componentes e plugins para o uso no projeto.

Trata-se de um banco de dados relacional com um modelo de cliente-servidor, local onde os dados serão armazenados e gerenciados. O MySQL deve ter funcional, utiliza diversas funções que conversam entre si e faz uma comunicação na plataforma da Amazon Web Services para instanciação da aplicação.

Serviços A aplicação deve estar disponibilizada através da plataforma de serviços web da Amazon, entre todas as camadas, pronto para ser expandido no próximo semestre. Mesmo com os serviços utilizados devem ser:

Amazon CodePipeline: Serviço de integração contínua para automatização do processo de lançamento da aplicação operações encontrados no caminho, para muitos, a criação de um site do completo zero é motivo de orgulho.

Amazon EC2: Serviço web para disponibilizar capacidade computacional na nuvem. Neste serviço, é possível ter controle dos recursos computacionais, na qual devemos utilizar uma configuração de processamento, armazenagem e de sistema operacional para instanciação de virtualização para o ambiente do servidor.

Amazon S3: Serviço para armazenagem em nuvem que deve ser utilizado para backup, arquivamento de dados e aplicações na Amazon Web Services.

Amazon RDS: Serviço que deve facilitar a configuração, operação e implantação do MySQL no ambiente AWS. Deve possibilitar instanciar o banco de dados MySQL da aplicação. A equipe, em retrospecto, entende que aprendeu com os erros e falhas no caminho, e vê agora a possibilidade de expandir o projeto, com novas funcionalidades, e refinar o website atual para criação de um produto final de qualidade.

Referências

AKITA, F. *Refletindo sobre RESOLUÇÃO de Problemas / O bug do Premiere*. 2019. Disponível em: <<https://www.youtube.com/watch?v=3W6xc4Yh2P0>>. Acesso em: 07 Jun. 2021. Citado na página 37.

AWS. *Breve panorama do CodePipeline*. 2021. Disponível em: <https://docs.aws.amazon.com/pt_br/codepipeline/latest/userguide/welcome-introducing.html>. Acesso em: 06 Jun. 2021. Citado na página 61.

Caderno de Laboratório. *O modelo MTV no Django*. 2021. Disponível em: <<https://cadernodelaboratorio.com.br/o-modelo-mtv-no-django/>>. Acesso em: 02 Jun. 2021. Citado na página 49.

Carrick-Davies. 2020 ou 2021. Acesso em: 07 Jun. 2021. Citado na página 13.

COSTA, A. *O que aprendemos com a história que o WeWork nos contou*. 2018. Disponível em: <<https://movebla.com/o-que-aprendemos-com-a-hist%C3%B3ria-que-o-wework-nos-contou-556ba12af74d>>. Acesso em: 10 Jul. 2021. Citado na página 14.

COWORKING BRASIL. *Um guia definitivo para você escolher o espaço ideal para construir a sua empresa*. 2017. Disponível em: <<https://coworkingbrasil.org/guia/home-office-coworking-escritorio-virtual-sala-comercial/>>. Acesso em: 07 Jun. 2021. Citado 6 vezes nas páginas 14, 35 e 36.

COWORKING BRASIL. *Censo Coworking Brasil 2018*. 2019. Disponível em: <<https://coworkingbrasil.org/censo/2018/>>. Acesso em: 07 Jun. 2021. Citado 8 vezes nas páginas 12, 15, 16, 19, 20 e 22.

COWORKING BRASIL. *Censo Coworking Brasil 2018*. 2019. Disponível em: <<https://coworkingbrasil.org/censo/2018/coworkers/>>. Acesso em: 07 Jun. 2021. Citado 9 vezes nas páginas 14, 16, 21, 22 e 23.

COWORKING BRASIL. *Manifesto Coworking / Coworking Brasil*. 2021. Disponível em: <<https://coworkingbrasil.org/manifesto/>>. Acesso em: 07 Jun. 2021. Citado 5 vezes nas páginas 12, 13 e 22.

COWORKING BRASIL. *O que é Coworking?* 2021. Disponível em: <<https://coworkingbrasil.org/como-funciona-coworking/>>. Acesso em: 07 Jun. 2021. Citado na página 12.

DELTA BC. *Coworking no Brasil: entenda o crescimento desse modelo*. 2021. Disponível em: <<https://www.deltabc.com.br/blog/coworking-no-brasil>>. Acesso em: 07 Jun. 2021. Citado na página 12.

DESKMAG. *Coworking in Brazil*. 2013. Disponível em: <<https://www.deskmag.com/en/coworking-city-country-profiles/coworking-spaces-in-brazil-sao-paulo-812>>. Acesso em: 9 jul 2021. Citado 3 vezes nas páginas 13 e 19.

DESKMAG. *The 2018 Global Coworking Survey*. 2018. Disponível em: <<https://www.deskmag.com/en/background-of-the-2018-global-coworking-survey-market-research>>. Acesso em: 9 jul 2021. Citado na página 12.

DUBE, C.; GUMBO, V. Diffusion of innovation and the technology adoption curve: Where are we? the zimbabwean experience. 2017. Disponível em: <https://www.researchgate.net/profile/Victor-Gumbo/publication/317704759_Diffusion_of_Innovation_and_the_Technology_Adoption_Curve_Where_Are_We_The_Zimbabwean_Experience/links/595ae3fea6fdcc36b4d7bed0/Diffusion-of-Innovation-and-the-Technology-Adoption-Curve-Where-Are-We-The-Zimbabwean-Experience.pdf>. Acesso em: 09 Aug. 2021. Citado na página 19.

FIZEZAP. *Preço do aluguel de salas e conjuntos comerciais sobe 0,32% em abril*. 2021. Disponível em: <<https://fipezap.zapimoveis.com.br/preco-do-aluguel-de-salas-e-conjuntos-comerciais-sobe-032-em-abril/>>. Acesso em: 07 Jun. 2021. Citado na página 15.

GANDINI, A. *The rise of coworking spaces: A literature review*. 2015. Disponível em: <<https://air.unimi.it/retrieve/handle/2434/618721/1152525/Ephemera ASN.pdf>>. Acesso em: 07 Jun. 2021. Citado 10 vezes nas páginas 14, 22, 32, 33 e 37.

GAUGER ANDREAS PFNÜR, J.-O. S. F. Coworking spaces and start-ups: Empirical evidence from a product market competition and life cycle perspective. 2021. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0148296321002460>>. Acesso em: 07 Jun. 2021. Citado 6 vezes nas páginas 33, 34, 35 e 36.

LAMELAS, A. As 5 principais metodologias agile: vantagens e desvantagens. 2018. Disponível em: <<https://www.xpand-it.com/pt-pt/blog/5-metodologias-agile/>>. Citado na página 41.

LUO, R. C. K. C. Y. Production of coworking spaces: Evidence from shenzhen, china. 2020. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0016718520300087>>. Acesso em: 07 Jun. 2021. Citado 6 vezes nas páginas 12, 16, 33 e 34.

MORISET, B. Building new places of the creative economy. the rise of coworking spaces. 2013. Disponível em: <<https://halshs.archives-ouvertes.fr/halshs-00914075/>>. Acesso em: 07 Jun. 2021. Citado 2 vezes nas páginas 21 e 35.

PRATT, A. C. Creative cities: the cultural industries and the creative class. 2008. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0467.2008.00281.x>>. Acesso em: 07 Jun. 2021. Citado 2 vezes nas páginas 32 e 33.

Refactoring.guru. *Adapter*. 2021. Disponível em: <<https://refactoring.guru/pt-br/design-patterns/adapter>>. Acesso em: 11 Jul. 2021. Citado na página 59.

Refactoring.guru. *Strategy*. 2021. Disponível em: <<https://refactoring.guru/pt-br/design-patterns/strategy>>. Acesso em: 11 Jul. 2021. Citado na página 60.

SCHWABER, J. S. K. *The 2020 Scrum Guide*. [S.l.], 2020. Disponível em: <<https://scrumguides.org/scrum-guide.html>>. Citado na página 41.

SEBRAE. *Micro e pequenas empresas geram 27% do PIB do Brasil - Sebrae.* 2011. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/ufs/mt/noticias/micro-e-pequenas-empresas-geram-27-do-pib-do-brasil,ad0fc70646467410VgnVCM2000003c74010aRCRD>>. Acesso em: 07 Jun. 2021. Citado na página 15.

SPINUZZI, C. Working alone together. 2012. Disponível em: <<https://journals.sagepub.com/doi/10.1177/1050651912444070>>. Acesso em: 07 Jun. 2021. Citado 2 vezes nas páginas 34 e 36.

WAN, J. Prevention and landing of bubble. international review of economics & finance. 2018. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S105905601730816X>>. Acesso em: 10 Jul. 2021. Citado na página 14.

WATERS-LYNCH, J. A theory of coworking: entrepreneurial communities, immaterial commons and working futures. 2018. Disponível em: <<https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/A-theory-of-coworking-entrepreneurial-communities-immaterial-commons-and-working-futures/9921864026601341>>. Acesso em: 09 Aug. 2021. Citado na página 18.

Glossário

Airbnb	Sistema online de hospedagem, onde o usuário pode alugar, principalmente, casas de família para aluguel e atividades turísticas. - Citado em 27
AWS	Uma plataforma de serviços de computação em nuvem, pertencente a Amazon. - Citado em 60
Bootstrap	Framework de desenvolvimento de aplicações cliente, que é desenvolvida sobre a linguagem de programação CSS e, opcionalmente, JavaScript. - Citado em 48
Django	Framework de desenvolvimento de websites, focado no desenvolvimento rápido, seguro e de fácil manutenção, que é desenvolvido sobre a linguagem de programação Python. - Citado em 47, 58
ITCSS	Forma de organização para arquivos de CSS, tornando eles mais fáceis de manipular e cuidar. - Citado em 56
Javascript	Linguagem de programação de script para front end, uma das mais comuns de uso. - Citado em 48
MySQL	Sistema gerenciador de banco de dados objeto relacional, desenvolvido como projeto de código aberto. - Citado em 48, 77
PEP 8	Trata-se de documentações que demonstram novas funcionalidades e procedimentos da linguagem Python. A versão 8 foi criada por Guido Van Rossum, Barry Warsaw e Nick Coghlan. Seu objetivo é estabelecer um estilo da linguagem. - Citado em 53
Python	Python é uma linguagem de programação alto nível, orientada a objeto, front e back-end, que prioriza a legibilidade do código. - Citado em 47, 53, 58
Scrum	É um framework ágil, voltado a gerenciamento de projetos, se baseando em aprendizado e ajuste aos fatores variantes do projeto. - Citado em 41
snake case	Se refere a um estilo de escrita em programação, voltado a substituir espaços por underscores. - Citado em 55

Apêndices

APÊNDICE A – Relatórios das Sprints

Neste apêndice encontram-se os relatórios das sprints gerados pela ferramenta Jira Software. Neste, estão contemplados os gráficos de story points por tempo, descrição das atividades, bem como identificação das atividades que foram concluídas e as que não foram concluídas na respectiva sprint.

Seu trabalho ▾ Projeto Integrado: TCC ▾ Filtros ▾ Painéis de controle ▾ Pessoas ▾ Aplic ▾ 🔍 ? 🔍 🔍

Projetos / Projeto Integrado: TCC / quadro PIT / Relatórios

Relatório de Sprint

Sprint semana 1

Sprint fechado, finalizado em Fernando Sousa Silva 12/mai/21 2:27 AM - 06/jul/21 2:30 AM View linked pages

The chart shows a horizontal red line at 0 story points from May 12 to July 6, indicating no progress or work remaining.

STORY POINTS

TEMPO

Relatório de Estado

* Item adicionada ao sprint depois do tempo de início

Items concluídas

Visualizar no Navegador de Itens

Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-74 *	Definir tema	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-75 *	Criação de blog	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-76 *	Criar SVN	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-77 *	Criar canal no Youtube	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-

Seu trabalho ▾ Projeto Integrado: TCC ▾ Filtros ▾ Painéis de controle ▾ Pessoas ▾ Aplic. Q ? 🔍 ⚙

Projetos / Projeto Integrado: TCC / quadro PIT / Relatórios

Relatório de Sprint

Sprint semana 2

Sprint fechado, finalizado em Fernando Sousa Silva 18/mai/21 3:08 AM - 06/jul/21 3:11 AM View linked pages

STORY POINTS

TEMPO

Relatório de Estado

* Item adicionada ao sprint depois do tempo de início

Items concluídas

Visualizar no Navegador de Itens

Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-78 *	Realizar levantamento de requisitos funcionais e não funcionais	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-79 *	Realizar levantamento das regras de negócios	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-80 *	Definir o escopo de projeto	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-

Seu trabalho ▾ Projeto Integrado: TCC Filtros ▾ Painéis de controle ▾ Pessoas ▾ Aplic. Q ? 🔍 ⚙

Projetos / Projeto Integrado: TCC / quadro PIT / Relatórios

Relatório de Sprint

Sprint semana 4

Sprint fechado, finalizado em Beatriz Harumi 15/jun/21 3:32 AM - 07/jul/21 12:42 AM View linked pages

The chart shows a horizontal red line at 0 story points from June 15 to July 7, indicating no progress or work remaining.

STORY POINTS

TEMPO

Relatório de Estado

* Item adicionada ao sprint depois do tempo de início

Items concluídas

Visualizar no Navegador de Items

Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-63 *	Criar o banco de dados	<input checked="" type="checkbox"/> Tarefa	↗ Highest	CONCLUÍDO	-
PIT-84 *	Criação do front-end da tela de login de cadastro	<input checked="" type="checkbox"/> Tarefa	➡ Medium	CONCLUÍDO	-
PIT-85 *	Realizar integração do front-end com back-end	<input checked="" type="checkbox"/> Tarefa	➡ Medium	CONCLUÍDO	-
PIT-86 *	Efetuar a criação do back-end da tela do login de cadastro	<input checked="" type="checkbox"/> Tarefa	➡ Medium	CONCLUÍDO	-

Items não Completas

Visualizar no Navegador de Items

Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-61 *	Criar instâncias de servidores AWS	<input checked="" type="checkbox"/> Tarefa	↗ Highest	EM ANDAMENTO	-

Seu trabalho ▾ Projeto Integrado: TCC ▾ Filtros ▾ Painéis de controle ▾ Pessoas ▾ Aplic ▾ 🔍 ? 🔍 🔍

Projetos / Projeto Integrado: TCC / quadro PIT / Relatórios

Relatório de Sprint

Sprint semana 5

Sprint fechado, finalizado em Fernando Sousa Silva 08/jun/21 9:00 PM - 26/jun/21 4:22 PM View linked pages
Preparar a apresentação do desenho do projeto do dia 15/06

STORY POINTS

TEMPO

Jun 8 Jun 15 Jun 26

Legend: Diretriz (Baseline), Valores restantes (Remaining values), Dias não úteis (Non-working days)

Relatório de Estado

Items concluídas

Visualizar no Navegador de Items

Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-16	Criar apresentação do desenho do projeto	<input checked="" type="checkbox"/> Tarefa	= Medium	CONCLUÍDO	-
PIT-19	Criação do gource	<input type="checkbox"/> História	▼ Low	CONCLUÍDO	-

Projetos / Projeto Integrado: TCC / quadro PIT / Relatórios

Relatório de Sprint

Sprint semana 6

Sprint fechado, finalizado em Fernando Sousa Silva 15/jul/21 7:07 PM - 08/ago/21 7:28 PM

[View linked pages](#)

The chart displays Story Points on the Y-axis (ranging from -1 to 1) against TEMPO (Time) on the X-axis (ranging from Jul 15 to Aug 08). A legend indicates three series: Diretriz (Grey line), Valores restantes (Red line), and Dias não úteis (White area).

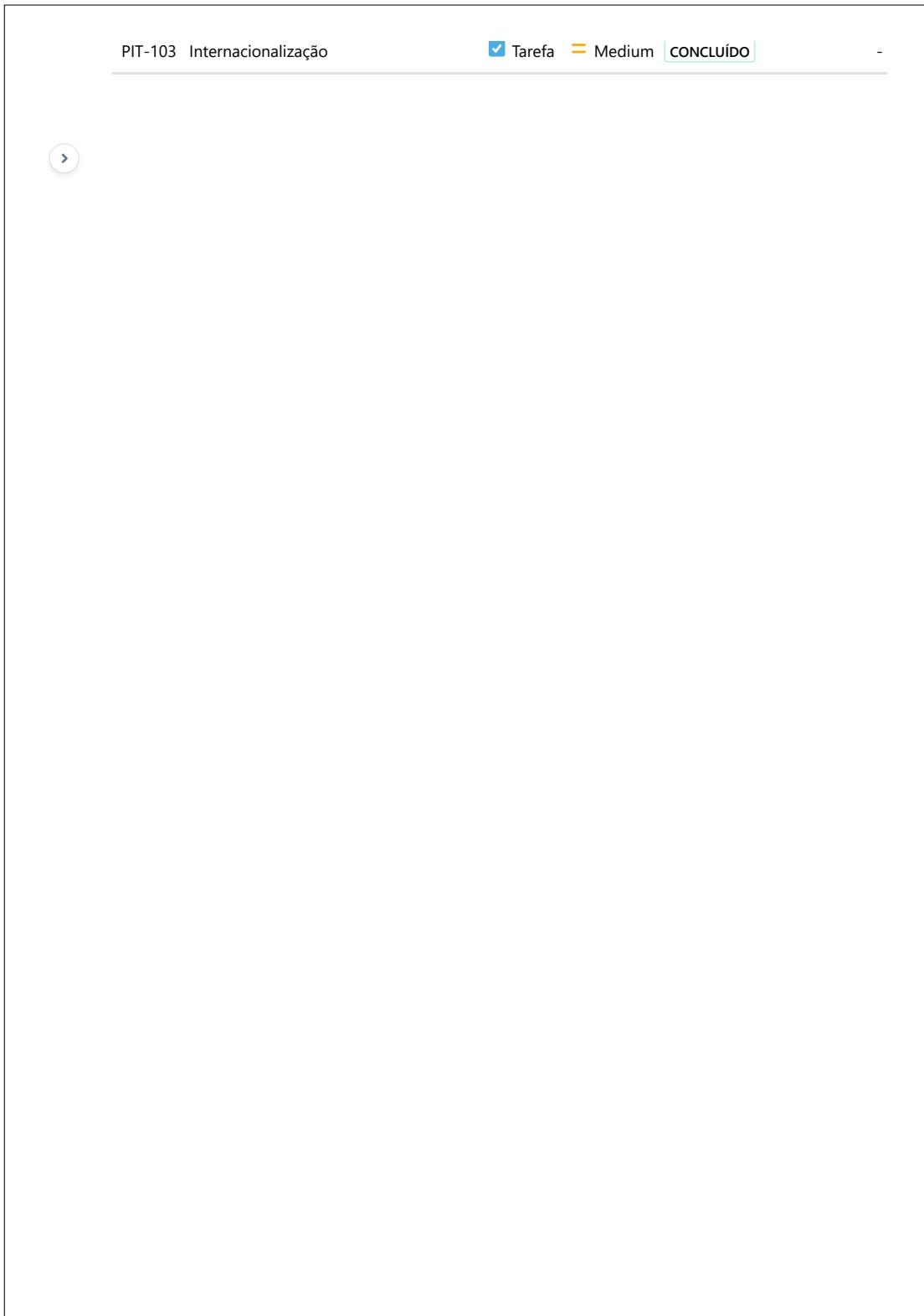
Relatório de Estado

* Item adicionada ao sprint depois do tempo de início

Items concluídas

[Visualizar no Navegador de Items](#)

Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-61	Criar instâncias de servidores AWS	<input checked="" type="checkbox"/> Tarefa	↗ Highest	CONCLUÍDO	-
PIT-95 *	Criar o front-end da tela de reserva	<input checked="" type="checkbox"/> Tarefa	▬ Medium	CONCLUÍDO	-
PIT-96 *	Criar o back-end da tela de reserva	<input checked="" type="checkbox"/> Tarefa	▬ Medium	CONCLUÍDO	-
PIT-98	Edição das Tabelas do MySQL	<input checked="" type="checkbox"/> Tarefa	▬ Medium	CONCLUÍDO	-
PIT-99	Adicionar botão para troca de cadastro Cliente PF/Cliente PJ	<input checked="" type="checkbox"/> Tarefa	▬ Medium	CONCLUÍDO	-
PIT-102	Validação de CPF e CNPJ	<input checked="" type="checkbox"/> Tarefa	▬ Medium	CONCLUÍDO	-



Projetos / Projeto Integrado: TCC / quadro PIT / Relatórios

Relatório de Sprint

Sprint semana 7

Sprint fechado, finalizado em Fernando Sousa Silva 28/jul/21 12:03 AM - 10/ago/21 12:03 AM

[View linked pages](#)

STORY POINTS

TEMPO

Relatório de Estado

* Item adicionada ao sprint depois do tempo de início

[Visualizar no Navegador de Items](#)

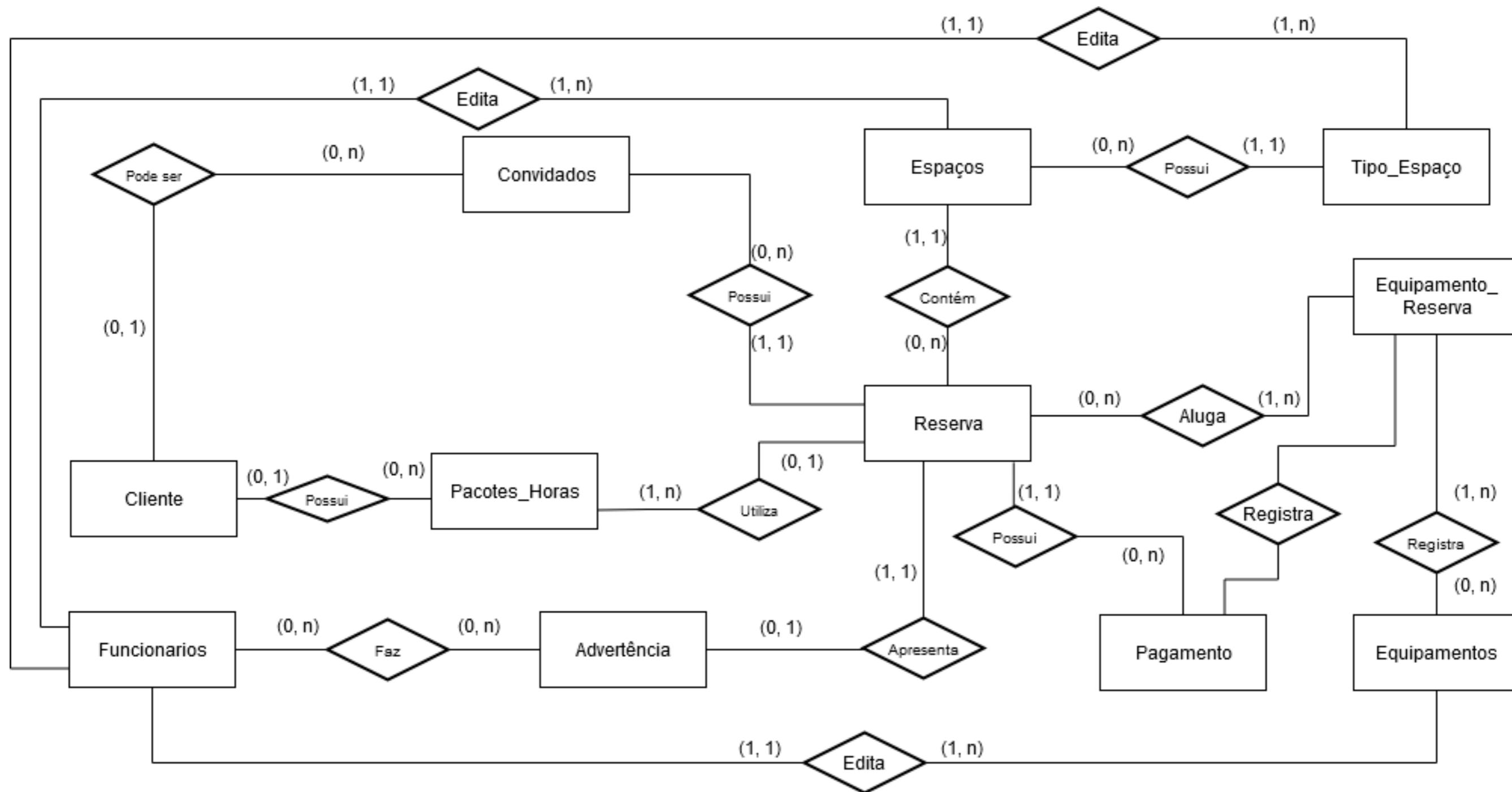
Chave	Resumo	Tipo de item	Prioridade	Status	Story Points (-)
PIT-94 *	Realizar ajustes da documentação final	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-115 *	Realizar revisão da sprint	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-116 *	Refatorar o código	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-117 *	Gerar certificado SSL e disponibilizar HTTPS	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-118 *	Atualizar o dicionário de dados	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-
PIT-119 *	Realizar testes unitários	<input checked="" type="checkbox"/> Tarefa	Medium	CONCLUÍDO	-

Relatório de Sprint - Sprint 1				
Identificação	Descrição	Tarefa	Complexidade	Status
PIT-120 *	Realizar alteração da página index	<input checked="" type="checkbox"/> Tarefa	= Medium	CONCLUÍDO
PIT-121 *	Realizar alterações na página administrador	<input checked="" type="checkbox"/> Tarefa	= Medium	CONCLUÍDO
PIT-122 *	Realizar alterações na pagina de customização	<input checked="" type="checkbox"/> Tarefa	= Medium	CONCLUÍDO
PIT-123 *	Realizar modificações na internacionalização	<input checked="" type="checkbox"/> Tarefa	= Medium	CONCLUÍDO
PIT-124 *	Realização correções nas conexões entre as páginas	<input checked="" type="checkbox"/> Tarefa	= Medium	CONCLUÍDO

APÊNDICE B – MER e DER

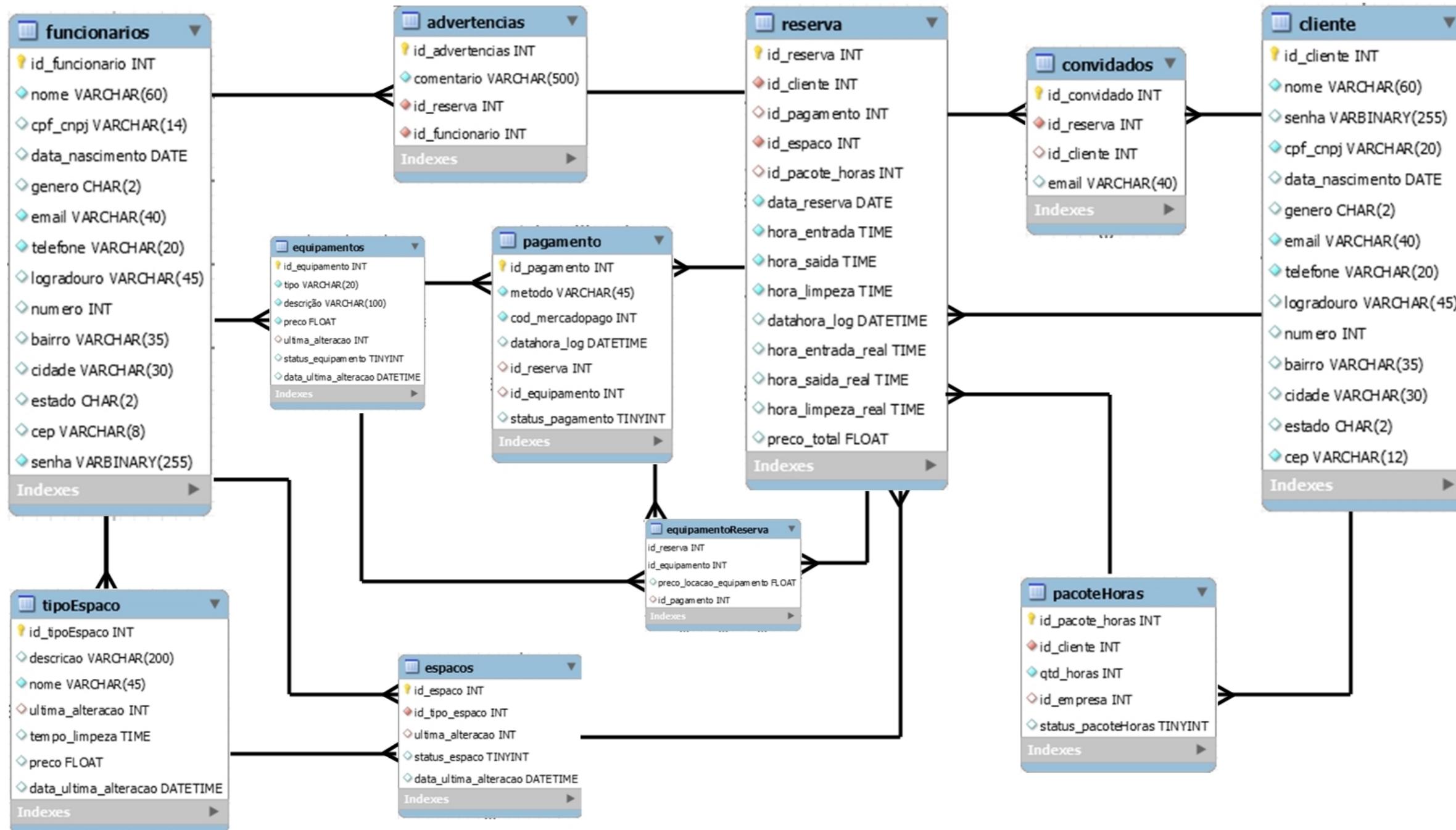
Para a definição do banco de dados, foi necessária a criação do MER ([Figura 38](#)) e DER ([Figura 39](#)) da aplicação.

Figura 38 – MER - Modelo Entidade-Relacionamento



Fonte: Os Autores

Figura 39 – DER - Diagrama Entidade-Relacionamento



Fonte: Os Autores

APÊNDICE C – Dicionário de Dados

Neste apêndice encontram-se as definições de todas as informações das tabelas do banco de dados da aplicação, incluindo os nomes das colunas, tipo e tamanho, definição de Primary e Foreign Keys e descrições de suas funções.

Quadro 7 – Descrição dos dados da tabela 'tipoEspaco'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_tipoEspaco	INT	15	PK	Deve ser gerada pelo sistema.
ultima_alteracao	INT	15	FK	ID do funcionário que atualizou.
data_ultima_alteracao	DATETIME			Deve ser puxado o horário do sistema na alteração.
nome	VARCHAR	45		Nome do tipo de espaço
preco	FLOAT	7		Preço do tipo de espaço
descrição	VARCHAR	200		Descrição do espaço apresentada no site.
tempo_limpeza	TIME			Quanto tempo se leva para limpar a sala.

Quadro 8 – Descrição dos dados da tabela 'espaco'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_espaco	INT	15	PK	Deve ser gerada pelo sistema.
id_tipo_espaco	INT	15	FK	ID do tipoEspaco que define.
ultima_alteracao	INT	15	FK	ID do funcionário que atualizou.
data_ultima_alteracao	DATETIME			Deve ser puxado o horário do sistema na alteração.
status	TINYINT	2		0 - Ativo, 1 - Desativado

Quadro 9 – Descrição dos dados da tabela 'equipamentos'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_equipamento</u>	<u>INT</u>	<u>15</u>	<u>PK</u>	<u>Deve ser gerada pelo sistema.</u>
<u>ultima_alteracao</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do funcionário que atualizou.</u>
<u>data_ultima_alteracao</u>	<u>DATETIME</u>			<u>Deve ser puxado o horário do sistema na alteração.</u>
<u>tipo</u>	<u>VARCHAR</u>	<u>45</u>		<u>Tipo do equipamento</u>
<u>descricao</u>	<u>VARCHAR</u>	<u>200</u>		<u>Descrição sucinta do equipamento, apresentada no site.</u>
<u>preco</u>	<u>float</u>			<u>Preço do equipamento</u>
<u>status</u>	<u>TINYINT</u>	<u>3</u>		<u>0 - Ativo, 1 - Desativado, 2 - Extraviado</u>

Quadro 10 – Descrição dos dados da tabela 'pagamento'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_pagamento</u>	<u>INT</u>	<u>15</u>	<u>PK</u>	<u>Deve ser gerada pelo sistema.</u>
<u>id_reserva</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID da reserva a ser paga.</u>
<u>id_equipamento</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do equipamento a ser reservado.</u>
<u>cod_mercadopago</u>	<u>INT</u>	<u>15</u>		<u>Código da transação via Mercado Pago.</u>
<u>metodo</u>	<u>TINYINT</u>	<u>3</u>		<u>0 - PIX, 1 - Cartão de Crédito e 2 - Boleto.</u>
<u>status</u>	<u>TINYINT</u>	<u>3</u>		<u>0 - Aguardando pagamento, 1 - Pagamento aprovado, 2 - Pagamento negado, 3 - Pacote de horas, 4 - Cancelado pelo cliente, 5 - Expirado, 6 - Cancelado via sistema</u>
<u>datahora_log</u>	<u>DATETIME</u>			<u>Deve ser puxado o horário do sistema quando foi feito o pedido de pagamento.</u>

Quadro 11 – Descrição dos dados da tabela 'funcionario'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_funcionario</u>	<u>INT</u>	<u>15</u>	<u>PK</u>	<u>Deve ser gerada pelo sistema.</u>
<u>nome</u>	<u>VARCHAR</u>	<u>60</u>		<u>Nome completo do funcionário.</u>
<u>cpf_cnpj</u>	<u>VARCHAR</u>	<u>14</u>		<u>CPF do funcionário ou CNPJ da empresa.</u>
<u>senha</u>	<u>VARBINARY</u>	<u>255</u>		<u>Senha de acesso, criptografada.</u>
<u>data_nascimento</u>	<u>DATE</u>			<u>Data de nascimento do funcionário.</u>
<u>genero</u>	<u>CHAR</u>	<u>2</u>		<u>M - Masculino, F - Feminino, O - Outro.</u>
<u>email</u>	<u>VARCHAR</u>	<u>14</u>		<u>E-mail do funcionário.</u>
<u>telefone</u>	<u>VARCHAR</u>	<u>14</u>		<u>Telefone do funcionário.</u>
<u>logradouro</u>	<u>VARCHAR</u>	<u>45</u>		<u>Logradouro do endereço do funcionário.</u>
<u>numero</u>	<u>INT</u>	<u>5</u>		<u>Número do endereço do funcionário.</u>
<u>bairro</u>	<u>VARCHAR</u>	<u>35</u>		<u>Bairro do endereço do funcionário.</u>
<u>cidade</u>	<u>VARCHAR</u>	<u>30</u>		<u>Cidade do endereço do funcionário.</u>
<u>estado</u>	<u>CHAR</u>	<u>2</u>		<u>Sigla do estado do funcionário.</u>
<u>cep</u>	<u>VARCHAR</u>	<u>8</u>		<u>CEP do endereço do funcionário, apenas números.</u>

Quadro 12 – Descrição dos dados da tabela 'cliente'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_funcionario	INT	15	PK	Deve ser gerada pelo sistema.
nome	VARCHAR	60		Nome completo do cliente
cpf_cnpj	VARCHAR	14		CPF do cliente ou CNPJ da empresa.
senha	VARBINARY	255		Senha de acesso, criptografada.
data_nascimento	DATE			Data de nascimento do cliente.
genero	CHAR	2		M - Masculino, F - Feminino, O - Outro.
email	VARCHAR	14		E-mail do cliente.
telefone	VARCHAR	14		Telefone do cliente.
logradouro	VARCHAR	45		Logradouro do endereço do cliente.
numero	INT	5		Número do endereço do cliente.
bairro	VARCHAR	35		Bairro do endereço do cliente.
cidade	VARCHAR	30		Cidade do endereço do cliente.
estado	CHAR	2		Sigla do estado do cliente.
cep	VARCHAR	8		CEP do endereço do cliente, apenas números.

Quadro 13 – Descrição dos dados da tabela 'pacoteHoras'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_pacote_horas	INT	15	PK	Deve ser gerada pelo sistema.
id_cliente	INT	15	FK	ID do cliente que pode utilizar.
id_empresa	INT	15	FK	ID da empresa que disponibiliza o pacote.
qtd_horas	INT	5		Quantidade de horas totais da locação.
status_pacoteHoras	TINYINT	3		0 - Ativo, 1 - Consumido, 2 - Cancelado

Quadro 14 – Descrição dos dados da tabela 'reserva'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_reserva</u>	<u>INT</u>	<u>15</u>	<u>PK</u>	<u>Deve ser gerada pelo sistema.</u>
<u>id_cliente</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do cliente que fez a reserva.</u>
<u>id_espaco</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do espaço reservado.</u>
<u>id_pagamento</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do pagamento da reserva.</u>
<u>id_pacoteHoras</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do pacote de horas utilizado.</u>
<u>datahora_log</u>	<u>DATETIME</u>			<u>Deve ser puxado o horário do sistema quando foi feito o pedido.</u>
<u>data_reserva</u>	<u>DATE</u>			<u>Data da reserva.</u>
<u>hora_entrada_reserva</u>	<u>TIME</u>			<u>Hora da entrada na reserva.</u>
<u>hora_entrada_real</u>	<u>TIME</u>			<u>Hora da entrada real.</u>
<u>hora_saida_reserva</u>	<u>TIME</u>			<u>Hora da saída na reserva.</u>
<u>hora_saida_real</u>	<u>TIME</u>			<u>Hora da saída real.</u>
<u>hora_limpeza_reserva</u>	<u>TIME</u>			<u>Tempo de término da limpeza na reserva.</u>
<u>hora_limpeza_real</u>	<u>TIME</u>			<u>Tempo de término da limpeza real.</u>
<u>preco_total</u>	<u>FLOAT</u>	<u>8</u>		<u>Preço da reserva do espaço.</u>

Quadro 15 – Descrição dos dados da tabela 'equipamentoReserva'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_equipamento</u>	<u>INT</u>	<u>15</u>	<u>FK/PK</u>	<u>Parte da PK, ID do equipamento a ser reservado.</u>
<u>id_reserva</u>	<u>INT</u>	<u>15</u>	<u>FK/PK</u>	<u>ID da reserva a qual pertence.</u>
<u>id_pagamento</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do pagamento da reserva.</u>
<u>preco_locacao_equipamento</u>	<u>FLOAT</u>	<u>8</u>		<u>Preço da locação do equipamento.</u>

Quadro 16 – Descrição dos dados da tabela 'convidados'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_convidado</u>	<u>INT</u>	<u>15</u>	<u>PK</u>	<u>Deve ser gerada pelo sistema.</u>
<u>id_reserva</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID da reserva a qual pertence.</u>
<u>id_cliente</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>Se o e-mail for compatível com algum cliente, insere o ID deste cliente.</u>
<u>email</u>	<u>VARCHAR</u>	<u>40</u>		<u>E-mail informado pelo responsável da reserva, do convidado.</u>

Quadro 17 – Descrição dos dados da tabela 'advertencia'

<u>NOME DA COLUNA</u>	<u>TIPO</u>	<u>TAMANHO</u>	<u>CHAVE</u>	<u>DEFINIÇÃO</u>
<u>id_advertencia</u>	<u>INT</u>	<u>15</u>	<u>PK</u>	<u>Deve ser gerada pelo sistema.</u>
<u>id_reserva</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID da reserva sofrendo advertência.</u>
<u>id_funcionario</u>	<u>INT</u>	<u>15</u>	<u>FK</u>	<u>ID do funcionário que realizou advertência.</u>
<u>comentario</u>	<u>VARCHAR</u>	<u>500</u>		<u>Comentários da advertência.</u>

APÊNDICE D – Publicações do Blog

Neste tópico encontram-se as publicações realizadas no blog da equipe, na íntegra.

Onde tudo começou...

 newgenprojetointegrado.blogspot.com/2021/05/onde-tudo-comecou.html

Olá! Somos o grupo New Gen, formado pela aula de Projeto Integrado I do curso de Análise e Desenvolvimento de Sistemas do IFSP.

A equipe é formada pelos alunos Camila França, Beatriz Harumi, Bruna Rodrigues, Fernando Sousa, Lucas Ferreira e Gabriel Pinheiro, que, durante a primeira reunião (a qual ocorreu no dia 11 de maio durante e após o horário da aula remota) estabeleceram um brain storm de ideias para utilizar no projeto.

Foi nesse momento que tiramos as principais dúvidas sobre o que era necessário para contemplar todos os requisitos solicitados e os fatores imprescindíveis para a aprovação deles.

Muitas ideias foram surgindo mas nenhuma delas teve tanto sucesso pela escassez de processos. Dentre elas: sistemas penitenciário, aplicação web de gerenciamento de estudos, aplicação de divulgação e busca de profissionais voltados a mídias sociais, entre dezenas de outras.

No entanto, nosso obstáculo foi se prender a uma função conhecida como CRUD, de busca, seleção de apresentação. E, para organizar a busca por um tema relevante e viável para o trabalho, estabelecemos o seguinte acordo: todos trariam uma ideia bem desenvolvida para a reunião seguinte, onde detalhariam processos e trâmites da aplicação.

Dificuldade pela frente

 newgenprojetointegrado.blogspot.com/2021/05/dificuldade-pela-frente.html

É claro que não esperávamos menos de um trabalho de tamanha magnitude. Os impasses surgem a todo momento e depende de nós saber desviá-los.

Em uma segunda reunião (marcada para o dia 16/05) levantamos as ideias do encontro anterior, só que dessa vez, com mais processos e detalhes inexistentes anteriormente. Foi um sucesso, pois conseguimos entender melhor nossas falhas de perspectiva e focar em no cerne do problema.

Camila e Fernando uniram suas ideias e propuseram uma aplicação parecida com a rede Linkedin, porém voltada a Social Medias (gerenciamento de mídias sociais, designers de posts, criadores de filtros, entre outros). Seu propósito seria realizar um contato entre pessoas que desejam divulgar seu trabalho em uma rede centrada no assunto e, no polo oposto, pessoas que se interessam por tais serviços.

Bruna levantou a possibilidade de gerarmos um sistema de agendamento de consultas baseado em um chatbot. Visando facilitar o atendimento de pacientes de um estabelecimento clínico, nossa aplicação iria poupar esforços de uma secretaria, por exemplo, realizando agendamentos simples, evitando aglomerações na linha telefônica ou filas desnecessárias.

Mas, a ideia que teve nossa aprovação por unanimidade foi a de Beatriz. Ela propôs que desenvolvêssemos um sistema de gerenciamento de um coworking que traria maior comodidade e organização para realizar reservas de salas e laboratórios de uma empresa.

O tema, por fim, foi escolhido pelo grupo, que passou a enfocar pensamentos em como detalhá-lo ainda mais para poder expô-lo para os professores Braz e Ivan durante a aula do dia 18/05. Agora o que nos resta é aguardar as cenas dos próximos capítulos...

Configurações e afins

 newgenprojetointegrado.blogspot.com/2021/06/configuracoes-e-afins.html

Nessa semana, estávamos um pouco atrasados, confessamos. Devíamos já ter as configurações de SVN devidamente estabelecidas para realizar commits e compartilhar nossos documentos com os professores.

Alguns detalhes no próprio blog também estavam aquém, então resolvemos tudo isso durante esse período de 7 (sete) dias. Na realidade, no total, só levou 2 para que conseguíssemos todos os acessos.

Bruna e Fernando foram os que tomaram a frente para configurar o Tortoise (mesmo que todos devessem possuí-lo em sua máquina pessoal, foram os dois alunos que desbravaram a plataforma para auxiliar o restante posteriormente).

Já Camila optou por gerenciar as mídias do projeto (blog e canal no YouTube), criando email, posts e pensando em futuros vídeos para alimentar as redes.

Não ocorreram maiores problemas, apesar de alguns pequenos infortúnios com o acesso de um dos alunos, mas que no geral foram bem solucionados por Bruna e pelos professores.

Beatriz e Lucas ficaram responsáveis pelas revisões do documento e por realizar as apresentações por slides, agregando todo o conteúdo em um compilado de cenas.

Documentação

 newgenprojetointegrado.blogspot.com/2021/06/documentacao.html

Durante esses 10 dias que se passaram, nós escrevemos o desenho da aplicação.

As tarefas foram divididas entre o cada membro do grupo da seguinte maneira:

Camila: Viabilidade financeira, manutenção da aplicação, escalabilidade, coding conventions e design patterns.

Fernando: Arquitetura, product backlog, tecnologias utilizadas, processo de integração e lançamento.

Beatriz: Metodologia Scrum, critérios de segurança, privacidade e legislação.

Bruna: Definição de escopo, casos de uso, regras de negócio, requisitos funcionais, histórias de usuário e gestão de tempo.

Lucas: Fases de entrega e papéis da equipe.

Gabriel: Contextualização, problematização, objetivos, justificativa e análise de concorrência.

Percebemos uma certa dificuldade de comunicação entre a equipe, já que parecíamos perdidos em relação a nossas tarefas e o que deveríamos atender nessa etapa do Projeto Integrado.

Ao final, tudo correu bem, todos colaboraram, Lucas e Bia como sendo, respectivamente, PO e Scrum Master do projeto, se encarregaram de revisar os tópicos dos demais integrantes e formatá-lo seguindo as coordenadas do Latex.

Foi de extrema importância essa pequena desavença no time, pois foi com ela que, após a entrega do desenho da aplicação, pudemos entender ainda melhor o quanto devemos nos manter unidos e nos organizarmos melhor para futuras entregas.

No geral, os professores Braz e Ivan aprovaram nosso desenho, apontando devidamente alguns erros a serem corrigidos, mas sem grandes equívocos.

Apresentação do Desenho da Aplicação

 newgenprojetointegrado.blogspot.com/2021/06/apresentacao-do-desenho-da-aplicacao.html

Nessa semana foi tempo de apresentar a nossos colegas de turma e professores nosso Desenho da Aplicação.

Em uma apresentação Power Point, elencamos os principais pontos do documento para expor nosso projeto e seus objetivos.

Como mencionado no post anterior, foram atribuídos a cada integrante do grupo alguns tópicos, os mesmos que seriam apresentados pelos membros nessa apresentação.

Em alguns dos ensaios realizados pela equipe, gravamos os discursos e disponibilizamos no canal do You Tube.

Nessa etapa, Beatriz e Lucas ficaram encarregados, além de estudar seus tópicos para apresentar, de elaborar a apresentação e formatá-la devidamente.

Camila também colaborou de maneira ativa com a criação de uma logo para o sistema. Como ela, já foi possível determinar uma paleta de cores para a marca NewGen e estabelecer uma identidade visual atrativa.



Desenvolvimento da POC

 newgenprojetointegrado.blogspot.com/2021/06/desenvolvimento-da-poc.html

Durante essa semana, todos se focaram a desenvolver a POC do nosso projeto.

Para isso, resolvemos dividir o grupo em 2 subgrupos: o de front-end e o de back-end. No front, se encontravam Bruna, Beatriz e Gabriel, enquanto no back ficaram Camila, Fernando e Lucas. Fernando se dispôs a realizar todas as configurações referentes à AWS, cujo qual enfrentamos alguns problemas de acesso. Com ela, não conseguíamos conectar nosso banco de dados MySQL devido a ausência de algumas permissões do plano.

Gabriel, Camila e Fernando começaram a prover a modelagem de dados, peça chave que estava atrasada desde a semana passada na entrega do desenho da aplicação. Os 3 começaram a desenhar o MER e Bruna, juntamente com Gabriel e Beatriz, deram continuidade. Camila e Lucas se uniram para configurar as instalações necessárias para o back-end como o ambiente de desenvolvimento Python com Visual Studio, Xampp e Django.

APÊNDICE E – Atas de Reunião

Neste tópico encontram-se as atas das reuniões realizadas pela equipe.

New Gen
Ata da reunião 01

14/05/2021, 20:30 às 22:08

I. Abrir sessão

Beatriz H. abriu a sessão da reunião regular do New Gen às 20:30, em 14 de maio, sexta-feira, através do Google Meets.

II. Questões da última reunião

Não houve reuniões prévias.

III. Questões debatidas

- a) Escolha de tema e descrição do projeto:
 - (1) Beatriz H.; Sistema de administração de coworking vinculado a aplicação web para locação de espaços.
 - (2) Camila F. e Fernando S.; Rede social em estilo LinkedIn, voltada a freelancers de redes sociais, com gamificação entre usuários.
 - (3) Bruna R.; Sistema de agendamento para consultas, via chatbot.
- b) Votação de escolha de tema:
 - (1) Votação unânime ao projeto (1), Sistema de administração de coworking.
 - (2) Conversa sobre pesquisa de competição, definição de escopo do projeto e exemplificação de conceitos.

IV. Questões para próxima reunião

- i) Reunião do dia 16 de maio, domingo, às 21:00.
- b) Criação de, ao menos, um processo adaptável ao sistema.
- c) Definições de habilidades individuais.

V. Adiamento

Beatriz H. adiou a reunião às 22:11.

Ata enviada por: Beatriz H.

Atas aprovadas por: Nome

New Gen
Ata da reunião 02

16/05/2021, 21:00 às 23:38

I. Abrir sessão

Camila França abriu a sessão da reunião regular do New Gen às 21:00, em 16 de maio, domingo, através do Google Meets.

II. Questões da última reunião

Trazer à tona detalhes relevantes para agregar à ideia vencedora da votação (sistema de gerenciamento de coworking).

III. Questões debatidas

- a) Incrementos para o tema do projeto;
 - (1) Camila F.; Declarar características de cada sala para locação de acordo com as necessidades de cada locador (datashow, televisão, notebooks, entre outros).
- b) Habilidades individuais;
 - (1) Camila F.; Programação em Python e gerenciamento de rede blogspot.

IV. Questões para próxima reunião

- i) Reunião do dia 16 de maio, domingo, às 21:00.
- b) Criação de, ao menos, um processo adaptável ao sistema.
- c) Definições de habilidades individuais.

V. Adiamento

Não houve.

Ata enviada por: Camila F.

New Gen
Ata da reunião 03

18/05/2021, 19:20 às 20:00

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do New Gen às 19:20, em 18 de maio, terça-feira, através do Google Meets.

II. Questões da última reunião

Detalhamento de processos dentro da ideia escolhida.

III. Questões debatidas

a) Organização e planejamento dos feitos do grupo;

(1) Criação do Subversion

Responsáveis: Bruna R., Fernando S. e Gabriel P.

(2) Montagem da apresentação

Responsáveis: Beatriz O. e Lucas

(3) Atas da reunião e relatório gerencial da semana

Responsáveis: Camila F. e Gabriel P.

b) Debate de questão da política do grupo;

Com intenção de deixar acertado como será feito as atividades do grupo, meios de comunicação e o comportamento perante os professores. Assim conferindo ao grupo uma postura mais unida e profissional.

IV. Questões para próxima reunião

i) Reunião do dia 21 de maio, sexta-feira, às 21:00.

a) Validação das atividades

b) Modificação da apresentação

c) Revisar o conteúdo dado e garantir que todo mundo está atualizado quanto aos requisitos e feitos do grupo

d) Divisão de tópicos da apresentação

ii) A ser debatido em reuniões futuras

- a) Criação de, ao menos, um processo adaptável ao sistema.
- b) Definições de habilidades individuais.

V. Adiamento

Reunião aberta em via de urgência. (Não foi previamente marcada)

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 04

23/05/2021, 17:30 às 20:00

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do New Gen às 19:20, em 23 de maio, domingo, através do Google Meets.

II. Questões da última reunião

Conversa rápida para ater todo mundo nas questões do grupo e também divisão das responsabilidades individuais da semana.

III. Questões debatidas

a) Divisão dos tópicos na apresentação

- 1 introdução
 - o que é coworking?
 - A tendência de crescimento dos coworkings
 - Por que um Coworking e não um escritório?

2 Proposta

- Aplicação síncrona, conectada ao site de fácil domínio
- Website White Label com agenda online

3 Diferenciais

- Sistema 100% portátil
- Chatbot pra locação
- Agendamentos
- Lista de espera automática
- Valicação de entrada através de QR Code
- Eventos no coworking

4 Tecnologias

- Back-End
- Front-End
- Colaboradores da NewGen

b) Montagem de Slides da Apresentação

IV. Questões para próxima reunião

i) Reunião do dia 24 de maio, segunda-feira, às 20:00

 a) Revisão dos tópicos

b) Apresentação teste

V. Adiamento

Reunião previamente marcada para sexta-feira dia 21/05/2021 e adiada para domingo às 18:00

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 05

24/05/2021, 17:30 às 20:00

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do New Gen às 21:20, em 24 de maio, segunda-feira, através do Google Meets.

II. Questões da última reunião

Divisão dos tópicos da apresentação.

III. Questões debatidas

- a) Treino prático cronometrado da apresentação da proposta
- b) Alterações em algumas falas para ajustar nos tempos previstos

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Divisão das próximas tarefas
 - b) Debate acerca dos resultados da apresentação do dia 25/05/21

V. Adiamento

Houve atraso de 1 hora em relação ao prazo demarcado

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 06

25/05/2021, 19:30 às 20:57

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do New Gen às 19:30, em 25 de maio, terça-feira, através do Google Meets.

II. Questões da última reunião

Treinamento da apresentação.

III. Questões debatidas

- a) Organização do documento da apresentação para entrega
- b) Criação de pedaços de documento que ficaram faltando no documento anterior.
Exemplo: tabela de concorrência

IV. Questões para próxima reunião

- i) Sem data marcada
- a) Divisão das próximas tarefas
- b) Debate acerca dos resultados da apresentação do dia 25/05/21

V. Adiamento

Reunião aberta em tom de emergência, portanto não foi considerada planejada.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 07

01/06/2021, 20:00 às 21:50 e 22:20 – 00:30

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do NewGen às 20:00, em 01 de junho, terça-feira, através do Google Meets.

II. Questões da última reunião

Reunião de emergência para terminar partes incompletas do documento de apresentação.

III. Questões debatidas

- a) Definições de processos para apresentar aos professores na reunião às 22:00

Processos definidos:

Processo: Cadastro

Processo: Aluguel por PF

Processo: Aluguel por empresa

Processo: Lista de Espera

Processo: Autorização de Entrada

Processo: Evento de Integração do Coworking

Processo: Finalização de Locação

Processo: Controle de estoque

Processo: Transferência de responsabilidade

Processo: Sistema de fidelidade

Processo: Relatórios periódicos

Processo: Locação de Máquinas

- b) Debate dos resultados da apresentação passada, sugestão de melhorias e combinados acerca da política do grupo.

- c) Utilização da ferramenta Jira

IV. Questões para próxima reunião

- i) Sem data marcada
- a) Divisão das próximas tarefas

V. Adiamento

Não houve atrasos

VI. Observação

A reunião acabou 21:50 para começar a reunião com os professores, mas após isso, Gabriel e Fernando deram continuidade para criação do que se tornou a nossa ferramenta de organização de equipe. Antes íamos testar Trello, mas devido a algumas limitações, o Fernando sugeriu o Jira.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 08

03/06/2021, 23:13 às 00:46

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do NewGen às 00:46, em 03 de junho, quinta-feira, através do Google Meets.

II. Questões da última reunião

Definições de processos, debates de política do grupo e criação do Jira

III. Questões debatidas

a) Criação dos TODOs da semana no Jira

- Introdução
 - Contextualização
 - Problematização
 - Objetivos
 - Justificativa
 - Análise de concorrência refinada
 - Revisão bibliográfica pertinente
- Definições de escopo
 - Casos de uso
 - Diagrama de requisitos
 - Histórias de usuário
 - Product backlog
 - Definir entregas (POC, MVP, Produto oficial)
- Definições gerais
 - Definir metodologia
 - Organização dos papéis da equipe
 - Arquitetura

- Gestão de tempo: Diagrama de Gantt, Cronograma ou Formato de Sprints
- Critérios de Segurança/Privacidade/Legislação
- Tecnologias a serem utilizadas
- Viabilidade financeira e manutenção da aplicação
 - Lançamento da ferramenta
 - Necessidade de escalabilidade
 - Um processo de integração contínua
 - Ferramentas compatíveis com as tecnologias escolhidas para testes automatizados e análise
 - Especificação do "Codingn Convention"
 - Design Patterns pertinentes à aplicação

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Atualização do que foi feito
 - b) Revisão das entregas

V. Adiamento

Não houve, reunião criada de última hora para discutir o assunto da divisão de tópicos do Jira.

VI. Observação

Devido ao horário tarde não planejado, somente Camila, Gabriel e Fernando participaram da reunião.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 09

08/06/2021, 21:00 às 21:20

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do NewGen às 21:00, em 08 de junho, terça-feira, através do Google Meets.

II. Questões da última reunião

Divisão dos tópicos do desenho da aplicação no Jira.

III. Questões debatidas

- a) Atualização do que foi feito
- b) Revisão das entregas
- c) Debate sobre problemas que ocorrem para atrasos na documentação

IV. Questões para próxima reunião

- i) Reunião do dia 14 de junho, segunda-feira, às 21:00.
 - a) Revisão dos slides
 - b) Treino da apresentação

V. Adiamento

Não houve atrasos e a reunião ocorreu dentro do planejado pelo WhatsApp.

Ata enviada por: Gabriel P.

New Gen

Ata da reunião 10

14/06/2021, 21:04 às 22:40

I. Abrir sessão

Nome do Facilitador abriu a sessão da reunião regular do NewGen às 21:04, em 14 de junho, segunda-feira, através do Google Meets.

II. Questões da última reunião

Atualizações, revisões e melhorias sobre a Sprint passada

III. Questões debatidas

- a) Revisão dos slides em conjunto
- b) Apresentação teste do desenho da aplicação com cronometragem

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Considerações pós-apresentação
 - b) Planejamento dos tópicos para POC

V. Adiamento

A reunião ocorreu na data e hora estipuladas sem atrasos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 11

21/06/2021, 22:28 às 22:46

I. Abrir sessão

Lucas F. abriu a sessão da reunião regular do NewGen às 22:28, em 21 de junho, segunda-feira, através do Google Meets.

II. Questões da última reunião

Treino da apresentação do desenho da proposta.

III. Questões debatidas

- a) Preparação e organização para POC
- b) Discussão técnica sobre implementação Python de um curso Udemy adquirido pelo grupo para auxílio do desenvolvimento.

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Divisão dos tópicos da POC
 - b) Criação do banco de dados

V. Adiamento

Reunião ocorrida sem planejamentos, portanto não houve atrasos ou adiamentos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 12

22/06/2021, 18:58 às 23:23

I. Abrir sessão

Fernando S. abriu a sessão da reunião regular do NewGen às 18:58, em 22 de junho, às 21:08 terça-feira, através do Google Meets, mas por problemas de dupla comunicação, migramos para a plataforma Discord e a reunião foi terminada ali.

II. Questões da última reunião

Debate pincelado para os próximos passos em relação à entrega da prova de conceito (POC)

III. Questões debatidas

- a) Passagem do banco de dados do rascunho para o Modelo de Entidade Relacional e o Diagrama de Entidade Relacional

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Subir o banco de dados na plataforma da AWS
 - b) Integrar back-end com front-end

V. Adiamento

Reunião ocorrida sem planejamentos, portanto não houve atrasos ou adiamentos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 13

23/06/2021, 21:37 às 22:30

I. Abrir sessão

Camila F. abriu uma sessão de reunião não planejada do NewGen às 18:58, em 23 de junho, às 21:37 quarta-feira, através do Google Meets.

II. Questões da última reunião

Criação dos modelos do banco. MER terminado e DER incompleto.

III. Questões debatidas

- a) DER terminado e script para o banco pronto

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Subir o banco de dados na plataforma da AWS
 - b) Integrar back-end com front-end
 - c) Integrar banco de dados no models (ORM) do Django

V. Adiamento

Reunião não planejada, portanto, sem adiamentos ou atrasos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 14

26/06/2021, 18:07 às 22:57

I. Abrir sessão

Beatriz H. abriu a sessão de reunião regular do NewGen às 18:07, em 26 de junho, Sábado, através do Google Meets.

II. Questões da última reunião

Término do Diagrama de Entidade Relacional do banco de dados

III. Questões debatidas

- a) Arrumado alguns erros que estavam no script auto gerado a partir do DER
- b) Instância do banco na AWS
- c) Integração do banco com o framework Django

IV. Questões para próxima reunião

- i) Sem data marcada
 - a) Integrar back-end com front-end

V. Adiamento

Sem adiamentos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 15

27/06/2021, 18:38 às 00:26

I. Abrir sessão

Beatriz H. abriu a sessão de reunião regular do NewGen às 18:38, em 27 de junho, Domingo, através do Google Meets.

II. Questões da última reunião

Integração do banco com o Django e conexão com a instância RDS da AWS.

III. Questões debatidas

- a) Tentando conectar o Front e o Backend

IV. Questões para próxima reunião

- i) Reunião do dia 28 de junho, domingo, às 19:00
 - a) Integrar back-end com front-end

V. Adiamento

Sem adiamentos.

VI. Adiamento

Não obtivemos sucesso na integração do front com o back pelo Django ainda.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 16

28/06/2021, 18:39 às 00:10

I. Abrir sessão

Camila F. abriu a sessão de reunião regular do NewGen às 18:39, em 28 de junho, segunda-feira, através do Google Meets.

II. Questões da última reunião

Tentando conectar o Front e o Backend

III. Questões debatidas

a) Tentando conectar o Front e o Backend

IV. Questões para próxima reunião

i) Data indefinida

a) Rever problemas da POC

V. Adiamento

Sem adiamentos.

VI. Adiamento

Não obtivemos sucesso na integração do front com o back pelo Django.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 17

29/06/2021, 16:47 às 18:47

I. Abrir sessão

Camila F. abriu a sessão de reunião emergencial do NewGen às 16:47, em 29 de junho, terça-feira, através do Google Meets.

II. Questões da última reunião

Tentando conectar o Front e o Backend

III. Questões debatidas

- a) Montagem rápida de slides para POC
- b) Gravação do vídeo da POC
- c) Divisão rápida das partes e tópicos que cada um vai abordar na POC

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Rever problemas da POC

V. Adiamento

Reunião emergencial, sem adiamentos.

VI. Observação

De última hora houve sucesso da integração do front com o back, mas sobraram lacunas por causa do atraso para resolver este problema. Faltou a internacionalização e a infraestrutura EC2 da AWS.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 18

05/07/2021, 22:25 às 00:47

I. Abrir sessão

Camila F. abriu uma reunião não planejada do NewGen às 22:25, em 05 de julho, segunda-feira, através do Google Meets.

II. Questões da última reunião

Reunião emergencial de ajustes finais para POC.

III. Questões debatidas

- a) Relatórios sobre o andamento da atualização do documento
- b) Arrumação das sprints desde o início do projeto

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Discutir planejamento do MVP

V. Adiamento

Reunião não planejada, portanto, sem agendamentos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 19

06/07/2021, 17:40 às 20:40

I. Abrir sessão

Fernando F. abriu uma reunião regular do NewGen às 22:25, em 06 de julho, terça-feira, através do Google Meets.

II. Questões da última reunião

Organização das Sprints do projeto todo.

III. Questões debatidas

- a) Tentar configurar o HTTP e o HTTPS na AWS
- b) Debate sobre o que vamos apresentar para os professores
- c) Revisão da política do grupo para melhorar as apresentações

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Arrumar HTTP e HTTPS da AWS
 - b) Discutir processo de reserva do MVP

V. Adiamento

Reunião sem atrasos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 20

12/07/2021, 21:20 às 21:40

I. Abrir sessão

Bruna R. abriu uma reunião breve do NewGen às 21:20, em 12 de julho, segunda-feira, através do Google Meets.

II. Questões da última reunião

Configurações da AWS.

III. Questões debatidas

- a) Comunicação das necessidades do Front-end para a equipe de Back-end

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Arrumar HTTP e HTTPS da AWS
 - b) Discutir processo de reserva do MVP

V. Adiamento

Reunião improvisada na hora, portanto sem atrasos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 21

13/07/2021, 20:43 às 22:00

I. Abrir sessão

Beatriz H. abriu uma reunião não planejada do NewGen às 19:53, em 13 de julho, terça-feira, através do Google Meets.

II. Questões da última reunião

Discussão dos papéis da equipe e sobre o que faltou no documento

III. Questões debatidas

- a) Discussão dos papéis da equipe
- b) Revisão do que faltou na entrega da versão 2 do documento

IV. Questões para próxima reunião

- i) Reunião do dia 14 de julho, quarta-feira, às 20:00
 - a) Ensinar o básico de Latex para todos da equipe conseguirem mexer no doc
 - b) Ensinar o básico de Jira para todos da equipe
 - c) Configurar o Django nos PCs de quem estava com problemas por causa das dependências
- ii) Data indefinida
 - a) Arrumar HTTPS da AWS
 - b) Atualizar o grupo sobre como está o MVP
 - c) Treinar para apresentação do MVP

V. Adiamento

Reunião não planejada, portanto não houve adiamentos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 22

14/07/2021, 20:43 às 22:00

I. Abrir sessão

Beatriz H. abriu uma reunião regular do NewGen às 20:43, em 14 de julho, quarta-feira, através do Google Meets.

II. Questões da última reunião

Discussão dos papéis da equipe e sobre o que faltou no documento

III. Questões debatidas

- a) Ensinamento dos básicos do Latex para equipe toda
- b) Ensinamento dos básicos do Jira para equipe toda
- c) Configuração do Django no computador de quem não tava conseguindo rodar

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Arrumar HTTPS da AWS
 - b) Atualizar o grupo sobre como está o MVP
 - c) Treinar para apresentação do MVP

V. Adiamento

Reunião adiada das 20:00 para as 20:40.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 23

26/07/2021, 21:42 às 23:00

I. Abrir sessão

Camila F. abriu uma reunião regular do NewGen às 21:42, em 26 de julho, segunda-feira, através do Google Meets.

II. Questões da última reunião

Ensinamento do básico de Latex e Jira e configuração de dependências no Django.

III. Questões debatidas

- a) Criação dos Slides para MVP
- b) Divisão das falas

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Arrumar HTTP e HTTPS da AWS
 - b) Atualizar o grupo sobre como está o MVP

V. Adiamento

Reunião adiada das 21:00 para as 21:42.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 24

27/07/2021, 16:31 às 18:45

I. Abrir sessão

Camila F. abriu uma reunião de emergência do NewGen às 16:31, em 27 de julho, terça-feira, através do Google Meets.

II. Questões da última reunião

Criação dos slides para apresentação do MVP

III. Questões debatidas

- a) Finalizações do que ficou pendente para o MVP
- b) Montagem final do Slide com algumas partes que ficaram faltando
- c) Atualizar o grupo sobre como está o MVP

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Arrumar e HTTPS da AWS
 - b) Discutir sobre a entrega do MVP e o que ficou faltando
 - c) Refletir sobre pontos de melhoria do grupo

V. Adiamento

Reunião aberta com tom de emergência, portanto não houve atrasos.

Ata enviada por: Gabriel P.

New Gen
Ata da reunião 25

28/07/2021, 21:04 às 22:30

I. Abrir sessão

Camila F. abriu uma reunião regular do NewGen às 21:04, em 28 de julho, quarta-feira, através do Google Meets.

II. Questões da última reunião

Finalizações do MVP

III. Questões debatidas

- a) Oportunidades de melhoria
- b) Pontos de erro
- c) Discussão acerca do que será o produto da equipe
- d) Separação de tarefas para entrega final
- e) Discussão de papéis

IV. Questões para próxima reunião

- i) Data indefinida
 - a) Discutir o que foi feito e arrumar o que ficou faltando até a entrega final.

V. Adiamento

Não houve adiamentos.

Ata enviada por: Gabriel P.

APÊNDICE F – Estudo informal sobre migração do banco

Este documento apresenta linguagem lúdica e certo grau de informalidade, pois o seu objetivo inicial era ser uma maneira de comunicar uma mensagem complicada que era a de alteração do Banco de dados de um sistema WhiteLabel para um sistema SaaS. Feito pelo integrante Gabriel Pinheiro, o documento possui as interpretações dele e o ponto de vista dele sobre o que pode ser feito pela equipe como um todo.

1 Glossário

Clientes proprietários: clientes do nosso sistema que são donos de um estabelecimento de coworking.

Clientes finais: clientes dos clientes proprietários, os clientes que usarão os estabelecimentos de coworking.

2 Introdução do problema

Antes de tudo, eu creio que o nosso sistema é basicamente focado para o público que não entrou no coworking e têm interesse no mercado imobiliário, ou seja, pessoas que não vão ter banco de dados e a gente que criará um para eles.

Agora partindo para o problema em si, vamos conferir se estamos na mesma página. Nós vamos hospedar um domínio na internet com algo como <http://www.newgenapp.link/>. Este domínio será o nosso domínio principal e será onde teremos uma página de cadastro para os **clientes proprietários** registrarem a sua empresa e efetuarem uma assinatura no nosso sistema. A partir dessa assinatura iremos gerar uma página para eles, no momento será algo como <http://www.newgenapp.link/coworkingJoao>, mas que no futuro podemos visualizar um plano de registro de domínio DNS especializado para cada cliente. Correto? Se a gente concorda até aí, beleza, bora dar continuidade.

Para cada cliente proprietário, precisamos hospedar seus dados específicos do seu coworking, mas ao mesmo tempo precisamos armazenar em um banco de dados principal alguma(s) tabela(s) que relate uma empresa com uma página e uma instância de dados. Imagine algo como:

Tabela empresas				
id_coworking	nome da empresa	configurações das págs	...	dados do seu coworking
1	JoãoEnterprises	{'agendamento': 'digital'}		TABELAS COWORKING1
2	IvanEmpresas	{'agendamento': 'digital'}		TABELAS COWORKING2
3	BrazCompany	{'agendamento': 'manual'}		TABELAS COWORKING3
4	CaixaFederal	{'agendamento': 'digital'}		TABELAS COWORKING4

No campo “configurações das págs” teríamos acesso à customização personalizada para cada coworking, assim quando o usuário abrir uma página ele carregaria uma página default aplicando as configurações daquele coworking específico.

No campo “dados do seu coworking” estaria as informações daquele coworking. Seria como se dentro desse campo tivesse o banco de dados atual nosso.

Porém, a única coisa que é certa é que temos que realizar um cadastro de uma empresa no coworking e dali gerar uma página para esta empresa poder trabalhar. A maneira de como faremos a ligação dos dados específicos a uma empresa a partir de um cadastro é um leque de soluções. As que eu pensei eu listei abaixo:

3 Possíveis Soluções

3.1 Tabela nova + campos novos em todas as tabelas

Vamos apelidar esta solução de solução 1. Se o título “Tabela nova mais campos novos em todas as tabelas” não ficou claro, bora explicar melhor no desenho.

Imagina que as tabelas do nosso banco tenham mais ou menos este formato:

Tabela Funcionário			
<i>id_func</i>	<i>nome</i>	<i>cpf</i>	...
51	Rogério	111.111.111-35	
52	Gabriel	222.222.222-94	
53	Mariela	333.333.333-77	
54	Daniel	444.444.444-14	
55	Isabela	555.555.555-15	
56	Iago	666.666.666-27	

Tabela Cliente			
<i>id_clien</i>	<i>nome</i>	<i>cpf</i>	...
457	Cláudia	114.775.945-95	
458	Matheus	184.893.894-47	
459	Felipe	194.975.786-14	
460	Carla	757.425.542-45	
461	Daniel	755.744.341-45	
462	Paula	455.453.123-94	

No momento todas as tabelas criadas por nós são: Funcionarios, Cliente, Tipoespaco, Equipamentos, Pacotehoras, Espacos, Advertencias, Convidados, Pagamento, Equipamentoreserva, Reserva

O que esta solução 1 propõe é adicionar um campo novo para cada tabela e atribuir cada campo com um coworking específico. Desse modo, teríamos algo como:

Tabela Funcionário				
<i>id_func</i>	<i>nome</i>	<i>cpf</i>	...	<i>id_coworking</i>
51	Rogério	111.111.111-35		2
52	Gabriel	222.222.222-94		2
53	Mariela	333.333.333-77		1
54	Daniel	444.444.444-14		3
55	Isabela	555.555.555-15		1
56	Iago	666.666.666-27		4

Tabela Cliente				
<i>id_clien</i>	<i>nome</i>	<i>cpf</i>	...	<i>id_coworking</i>
457	Cláudia	114.775.945-95		2
458	Matheus	184.893.894-47		1
459	Felipe	194.975.786-14		3
460	Carla	757.425.542-45		4
461	Daniel	755.744.341-45		1
462	Paula	455.453.123-94		4

As tabelas seriam uma bagunça e na hora de dar um select sem um “Where *id_coworking = 2*”, por exemplo, veríamos essa mistura de campos com dados de diferentes coworkings.

É válido lembrar também que eu mostrei apenas 2 tabelas simples, mas esse campo novo “id_coworking” teria que estar em TODAS as outras tabelas que temos no nosso banco atualmente. Como por exemplo “tipoespaco”:

Tabela tipo_espaco			
id_tipoespaco	nome	...	id_coworking
15	Sala privada		1
16	Mesa compartilhada		2
17	Mesa compartilhada		1
18	Sala privada		3
19	Sala de fotografia		4
20	Sala de reunião		3

O que eu entendi da proposta da Bia naquele áudio “Camila” de 3 minutos dela, seria pegar essa solução 1 e adicionar a primary key e o id_coworking e juntar as duas para serem compostas.

Agora vamos para os prós e contras dessa solução:

PRÓS

- Fácil de implementar
- Vai funcionar para nossa MVP

CONTRAS

- Jeito porco de fazer.

Misturando os dados de todos os clientes em uma mesma tabela causará grandes problemas de otimização, as consultas ficarão gradativamente mais lentas, não há fácil entendimento das tabelas e o gerenciamento delas ficaria exclusiva do backend.

- Possível vulnerabilidade

Misturar dados de clientes em uma só tabela pode violar aquele código de leis que Ivan pediu pra gente ler, além de se tornar mais fácil que dados de todos os nossos clientes sejam corrompidos de uma só vez.

3.2 Instância de banco de dados individual para cada cliente

Vamos chamar esta solução de solução 2. Em resumo, esta solução se baseia em fazer um banco de dados de banco de dados.

Estão lembrados desta tabela?

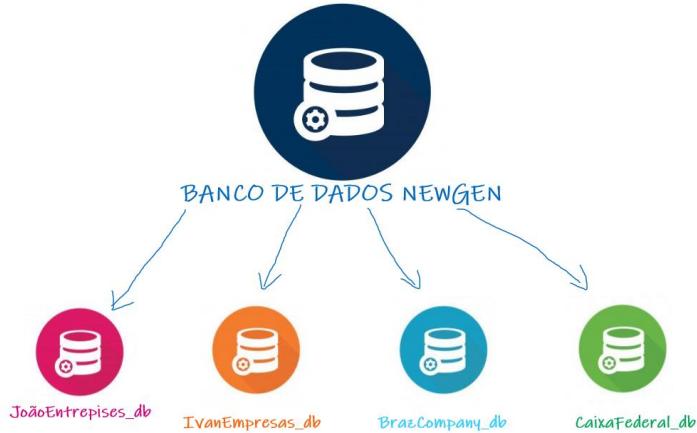
Tabela empresas				
id_coworking	nome da empresa	configuracoes das pags	...	dados do seu coworking
1	JoãoEnterprises	{'agendamento': 'digital'}		TABELAS COWORKING1
2	IvanEmpresas	{'agendamento': 'digital'}		TABELAS COWORKING2
3	BrazCompany	{'agendamento': 'manual'}		TABELAS COWORKING3
4	CaixaFederal	{'agendamento': 'digital'}		TABELAS COWORKING4

Agora vamos supor que em vez do campo “dados do seu coworking” tenhamos algo como:

Tabela empresas				
id_coworking	nome da empresa	configuracoes das pags	...	Key do Banco
1	JoãoEnterprises	{'agendamento': 'digital'}		Key1
2	IvanEmpresas	{'agendamento': 'digital'}		Key2
3	BrazCompany	{'agendamento': 'manual'}		Key3
4	CaixaFederal	{'agendamento': 'digital'}		Key4

E este campo “key do banco” se refere a uma chave de acesso do administrador àquele banco específico que será gerado pela gente assim que o processo de assinatura dele no nosso sistema for concluído.

O que eu imagino para esta solução é algo mais ou menos como:



Onde o banco de dados newgen, ou o newgen_db, armazenaria as informações de acesso para cada um dos bancos dos clientes proprietários. Já o cliente final acessaria algo como o "IvanEmpresas_db" por exemplo.

Lembrando que cada instância de banco de dados apresentada, seja o "JoãoEnterprises_db" ou o "BrazCompany_db", todos teriam as tabelas que temos atualmente, eles só seriam instanciados com dados próprios de cada coworking. É como se o banco de dados atual nosso** fosse uma classe de java e um banco de dados do cliente proprietário, ex: "BrazCompany_db", fosse um objeto instanciado. A gente ainda farçaria os clientes proprietários a usar um banco dentro dos nossos moldes, o que mudaria é que cada molde pode ser instanciado para cada empresa.

Eu não sei como fazer isso, mas já ouvi falar de conceitos como "Docker" ou "Virtualização de banco de dados", eu teria que ir atrás com o Fernando para saber como que isso pode ser viável.

**(aquele que tem todas aquelas tabelas: Funcionarios, Cliente, Tipoespaco, Equipamentos, Pacotehoras, Espacos, Advertencias, Convidados, Pagamento, Equipamentoreserva, Reserva)

PRÓS

- Maneira profissional de fazer

- Otimização eficiente

- Proteção dos dados

CONTRAS

- Difícil implementação

Sendo bem sincero não faço ideia de como implementar isso

- Talvez não seja possível de fazer isso usando a AWS

Ou talvez seja, usando Docker. Mas de novo, eu não sei.

- Poderá consumir bastante tempo para ser implementado

3.3 Utilização do Mongo DB

Por último e não menos importante temos a solução 3, a solução de trocar o tipo de banco de dados.

Tá... mas por que o mongoDB?

O mongoDB é inteiro estruturado como um grande JSON. JSON para quem não sabe são objetos que tem a seguinte roupagem:

```
1  {
2    'nome': 'Gabriel',
3    'cpf': '111.111.111.11',
4    'idade': 20,
5    'altura': 1.94,
6 }
```

Em Python objetos JSON são conhecidos como dicionários e exatamente este formato. Então traduzir de JSON pra dicionário é como traduzir um `int` para um `Integer` (quem sabe java entendeu).

No mongoDB todas as tabelas são JSONs, mas cada campo é um JSON. Ficou confuso né? Melhor mostrando na prática.

```
1 { 'id1': {
2     'nome': 'Gabriel',
3     'cpf': '111.111.111.11',
4     'idade': 20,
5     'altura': 1.94,
6   },
7   'id2': {
8     'nome': 'Isabela',
9     'cpf': '981.784.954.13',
10    'idade': 19,
11    'altura': 1.70,
12  },
13   'id3': [
14     {
15       'nome': 'Pedro',
16       'cpf': '111.111.111.12',
17       'idade': 21,
18       'altura': 1.78,
19     }
20 }
```

Mas, nada impede que o campo 'id4' seja algo como:

```
'id4': {
    'endereco': 'Rua dos bobos nro 0',
    'cidade': 'Sao Paulo',
    'preco_hora_default': 49.99,
}
```

Eu posso estar falando bobagem, porque eu usei o mongoDB só para fazer um trabalhinho de BD2A3, mas pelo que eu me lembre, isso é válido dentro do mundo dos bancos de dados não relacionais.

Dessa forma, o que impediria a gente de ter um descontrole total das tabelas seria o backEnd, mas visto que no mundo real ninguém inseriria dados à mão, não teria problema.

Agora, sabendo que podemos inserir JSONs dentro de JSONs, poderíamos inserir tabelas dentro de tabelas e ter algo como:

```

28 { '1': {
29     'nome da empresa': 'JoãoEnterprises',
30     'configurações das pags': {'agendamento':'digital'},
31     'database': {
32         'Clientes': {'457': {
33             'nome': 'Cláudia',
34             'cpf': '114.775.945-95',
35             'sexo': 'F'
36         },
37             '458': {
38                 'nome': 'Matheus',
39                 'cpf': '184.893.894-47',
40                 'sexo': 'M'
41             }
42         },
43         'Funcionarios': {'14': {
44             'nome': 'Rogério',
45             'cpf': '111.111.111-35',
46             'sexo': 'M'
47         }
48     },
49     'Tipo_Espaco':{'1': {
50         'nome': 'Sala privada',
51         'descrição': 'Uma sala particular para sua equipe trabalhar mais reservadamente.'
52         'compartilhado': False,
53     }
54     }
55   },
56   '2': {
57     'nome da empresa': 'IvanEmpresas_db',
58     'configurações das pags': {'agendamento':'digital'},
59     'database': {
60         'Tipo_Espaco':{'1': {
61             'nome': 'Mesa compartilhada',
62             'descrição': 'Uma mesa onde várias pessoas podem desfrutar da inovação do coworking'
63             'compartilhado': True,
64         }
65     }
66   }
67 }
68 }
```

Explicando melhor, temos a imagem abaixo:

```

23 { '1': {
24     'nome da empresa': 'JoãoEnterprises',
25     'configurações das pags': {'agendamento':'digital'},
26     'database': {
27         'Clientes': {'457': {
28             'nome': 'Cláudia',
29             'cpf': '114.775.945-95',
30             'sexo': 'F'
31         },
32             '458': {
33                 'nome': 'Matheus',
34                 'cpf': '184.893.894-47',
35                 'sexo': 'M'
36             }
37         },
38         'Funcionarios': {'14': {
39             'nome': 'Rogério',
40             'cpf': '111.111.111-35',
41             'sexo': 'M'
42         }
43     },
44     'Tipo_Espaco':{'1': {
45         'nome': 'Sala privada',
46         'descrição': 'Uma sala particular para sua equipe trabalhar mais reservadamente.'
47         'compartilhado': False,
48     }
49     }
50   },
51   '2': {
52     'nome da empresa': 'IvanEmpresas_db',
53     'configurações das pags': {'agendamento':'digital'},
54     'database': {
55         'Tipo_Espaco':{'1': {
56             'nome': 'Mesa compartilhada',
57             'descrição': 'Uma mesa onde várias pessoas podem desfrutar da inovação do coworking'
58             'compartilhado': True,
59         }
60     }
61   }
62 }
```

Na imagem acima, o preenchido de vermelho representa o banco de dados de cada coworking e o de rosa seria as tabelas desses bancos de dados. O melhor disso é que o mongoDB é flexível e otimizado para comportar tabelas desse nível, então acredito que poderíamos sustentar algum negócio utilizando isso.

PRÓS

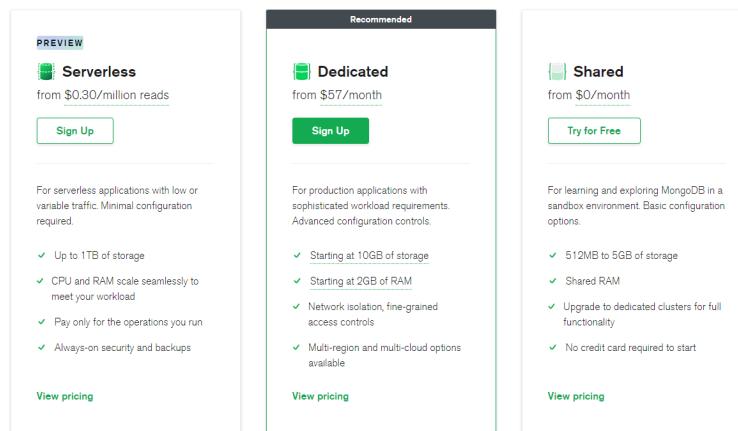
- Ganho de otimização
- Mais profissional que a solução 1

CONTRAS

- Pouca experiência da equipe com o mongoDB
- Possíveis problemas na implementação devido ao problema de experiência
- Alteração radical do nosso paradigma
- A AWS não aceita suporte para isso

Porém eu vi que existem soluções gratuitas tranquilas para fazer essa hospedagem

<https://www.mongodb.com/pricing>



E provavelmente o acoplamento disso com a AWS seja quase nulo de dificuldade, já que é só alterar dentro da nossa aplicação Django o ip do novo database.

Na verdade, isso aliviaria o gasto de créditos da AWS.

4 Prioridades atuais

Eu imagino que a gente não deva se centrar nesta parte de cadastrar um coworking por agora e faça um sistema de agendamento como se só tivéssemos cadastrado no momento um cliente proprietário. Ou seja, vamos deixar o nosso SQL como está e deixar estas alterações pendentes para as férias ou para o semestre 2. No momento eu acho bom a gente só fazer a simulação de um coworking real e terminar o processo de agendamento, já que essa simulação se encaixa bem no paradigma atual que o código está setado (White Label).

Como eu disse, eu pensei que o nosso sistema era WhiteLabel, não SaaS (Software as a Service), então o nosso código atualmente está funcionando como um whiteLabel, só teria que alterar as views que a Bruna mandou pq ela setou um Saas.

É basicamente só trocar texto, não abandonar o modelo SaaS e analisar melhor a implementação dessas implicações para depois do dia 10/08 e se concentrar somente no agendamento por enquanto. É essa a minha sugestão.

5 Bibliografia

<https://dicas.ivanfm.com/aulas/pi1a5.html>