

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 *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 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	20
Figura 2 – Crescimento de empresas que disponibilizam espaços de coworking dentro do Brasil ao longo dos anos	20
Figura 3 – Distribuição das empresas de coworking pelos estados e cidades brasileiras no ano de 2018	21
Figura 4 – Rendimento das empresas clientes de coworking no Brasil em 2018 . . .	22
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?”	23
Figura 6 – Avaliação de serviços essenciais no coworking para os clientes de coworking, Brasil, 2018	24
Figura 7 – Código <i>Quick Response</i> (QR) da aplicação.	39
Figura 8 – Código QR do repositório Subversion (SVN).	39
Figura 9 – Código QR do repositório no GitHub.	40
Figura 10 – Código QR do blog da equipe.	40
Figura 11 – Código QR do canal do YouTube.	40
Figura 12 – Contribuição ao SVN em total de linhas	42
Figura 13 – Contribuição ao SVN mensalmente	42
Figura 14 – Dispersão de commits	43
Figura 15 – Quantidade de linhas de código	44
Figura 16 – Tipos de atividades dos autores	44
Figura 17 – Commits por horário	45
Figura 18 – Commits por dia da semana	45
Figura 19 – Atividade por autor	46
Figura 20 – Atividade por commit em dispersão	46
Figura 21 – Atividade por dia da semana	47
Figura 22 – Quantidade de código total feito	47
Figura 23 – Contribuição por linha de código	48
Figura 24 – Desenvolvedor com maior contribuição do mês	48
Figura 25 – Atividade por hora do dia	49
Figura 26 – Edições de código por pessoa	49
Figura 27 – Linhas de código por pessoa	49
Figura 28 – Modelo MTV no Django	51
Figura 29 – Servidor para processo de requisições de usuário.	53
Figura 30 – Esquematização de fluxo para comunicação entre ambientes.	54

Figura 31 – Esquematização de fluxo para comunicação NGINX - Gunicorn - Django.	54
Figura 32 – Exemplo de Identação em Código Simples.	56
Figura 33 – Exemplo de Identação em Array.	57
Figura 34 – Exemplo de Identação em Array.	57
Figura 35 – Exemplo de Tamanho de Linhas adequado.	57
Figura 36 – Exemplo de Imports Simples.	58
Figura 37 – Exemplo de Imports Múltiplos.	58
Figura 38 – Exemplo de Settings.	59
Figura 39 – Exemplo de Tools.	59
Figura 40 – Exemplo de Generic.	60
Figura 41 – Exemplo de Elements.	60
Figura 42 – Exemplo de Object.	60
Figura 43 – Exemplo de Components.	60
Figura 44 – Exemplo de Trumps.	61
Figura 45 – Exemplo de um Design Pattern Adapter	62
Figura 46 – Exemplo de um Design Pattern Strategy	63
Figura 47 – Esquematização do Amazon Web Services CodePipeline.	63
Figura 48 – Esquema de Sprint na Ferramenta Jira	64
Figura 49 – Planos de assinatura do sistema da NewGen	74
Figura 50 – Security Headers do domínio <i>newgenapp</i> .	75
Figura 51 – Security Headers do domínio <i>milleniacoworking</i> .	75
Figura 52 – Security Headers do domínio <i>nonworking</i> .	75
Figura 53 – Security Headers do domínio <i>creativework</i> .	76
Figura 54 – Security Headers do domínio <i>overworking</i> .	76
Figura 55 – Resultado do teste SSL do domínio <i>newgenapp</i> .	76
Figura 56 – Resultado do teste SSL do domínio <i>creativecoworking</i> .	77
Figura 57 – Resultado do teste SSL do domínio <i>milleniacoworking</i> .	77
Figura 58 – Resultado do teste SSL do domínio <i>nonworking</i> .	77
Figura 59 – Exemplo de dados inseridos dentro da tabela <i>auth_user</i>	81
Figura 60 – Formulário de pessoa física dentro do template .html	81
Figura 61 – Seleção de opção de pessoa física ou pessoa jurídica.	82
Figura 62 – Erros, validação de máscara e campos nos formulários.	83
Figura 63 – Login no sistema, com máscara de cliente.	83
Figura 64 – Login no sistema, com máscara de empresa.	84
Figura 65 – Navbar de login.	84
Figura 66 – Tela inicial da NewGen.	86
Figura 67 – Tela de cadastro de coworking.	87
Figura 68 – Domínios de coworkings, disponíveis no banco de dados.	87
Figura 69 – Configuração do e-mail para envios.	88

Figura 70 – Opção de recuperação de senha, na área de login.	89
Figura 71 – Requisição do e-mail para envio do link.	89
Figura 72 – Mensagem informada ao cliente, caso o e-mail esteja presente na base.	90
Figura 73 – E-mail com link para redefinição de senha.	90
Figura 74 – Página de redefinição, com as regras para novas senhas.	91
Figura 75 – Conclusão da alteração de senha.	91
Figura 76 – Tela de seleção do tipo de espaço.	93
Figura 77 – Tela de seleção de espaço.	94
Figura 78 – Seleção de data e hora de entrada e saída.	94
Figura 79 – Calendário para seleção da data.	95
Figura 80 – Exemplo de eventos no calendário.	95
Figura 81 – Exemplo de reservas	96
Figura 82 – Lógica de Pagamento	98
Figura 83 – Formulário de pagamento	98
Figura 84 – Código para envio de e-mail	100
Figura 85 – Botão para envio do código QR, na aba de reservas	100
Figura 86 – URL identificando para qual reserva foi feita o envio	100
Figura 87 – Código hash, salvo no banco de dados	100
Figura 88 – E-mail com código anexo em PDF	101
Figura 89 – Valor inserido no código QR	101
Figura 90 – DER das informações do cliente	103
Figura 91 – Tag de templates para tradução.	104
Figura 92 – Nova tag de templates para tradução.	105
Figura 93 – Cadastro das informações do cliente administrador	106
Figura 94 – Edição da tela principal	107
Figura 95 – Edição da imagem e Sobre Nós	108
Figura 96 – Edição da paleta de cores do site	109
Figura 97 – Exemplo de tela de relatórios na aplicação	111
Figura 98 – MER - Modelo Entidade-Relacionamento - NewGen	140
Figura 99 – DER - Diagrama Entidade-Relacionamento - NewGen	140
Figura 100 – MER - Modelo Entidade-Relacionamento - Coworking	141
Figura 101 – DER - Diagrama Entidade-Relacionamento - Coworking	142

Lista de quadros

Quadro 1 – Comparação de Funcionalidades Entre Concorrentes	30
Quadro 2 – Escopo do projeto	50
Quadro 3 – Relação de requisitos funcionais	69
Quadro 4 – Continuação da relação de requisitos funcionais	70
Quadro 5 – Relação de requisitos não funcionais	71
Quadro 6 – Listagem de regras de negócios	72
Quadro 7 – Descrição dos dados da tabela 'administrador'	143
Quadro 8 – Descrição dos dados da tabela 'domain'	143
Quadro 9 – Descrição dos dados da tabela 'tipoEspaco'	143
Quadro 10 – Descrição dos dados da tabela 'espaco'	144
Quadro 11 – Descrição dos dados da tabela 'equipamentos'	144
Quadro 12 – Descrição dos dados da tabela 'pagamento'	144
Quadro 13 – Descrição dos dados da tabela 'funcionario'	145
Quadro 14 – Descrição dos dados da tabela 'cliente'	145
Quadro 15 – Descrição dos dados da tabela 'pacoteHoras'	146
Quadro 16 – Descrição dos dados da tabela 'reserva'	146
Quadro 17 – Descrição dos dados da tabela 'equipamentoReserva'	146
Quadro 18 – Descrição dos dados da tabela 'convidados'	146
Quadro 19 – Descrição dos dados da tabela 'advertencia'	147
Quadro 20 – Descrição dos dados da tabela 'clienteConvidado'	147

Listas de abreviaturas e siglas

AWS	<i>Amazon Web Services</i> - Serviços Web da Amazon - Citado em 52
DER	Diagrama Entidade-Relacionamento - Citado em 55 , 79
HTTPS	<i>Hypertext Transfer Protocol</i> - Protocolo de Transferência de Hipertexto - Citado em 71 , 75
LGPD	Lei Geral de Proteção de Dados - Citado em 74
MER	Modelo Entidade-Relacionamento - Citado em 55 , 79
MVP	<i>Minimum Viable Product</i> - Mínimo Produto Viável - Citado em 28 , 50
POC	<i>Proof of Concept</i> - Prova de Conceito - Citado em 50
QR	<i>Quick Response</i> - Resposta Rápida - Citado em 4 , 39 , 40 , 70
SaaS	<i>Software as a Service</i> - Software como serviço - Citado em 13 , 18 , 27 , 78 , 79 , 203
SVN	Subversion - Citado em 4 , 39

Sumário

1	INTRODUÇÃO	13
1.1	Contextualização	13
1.2	Problematização	15
1.2.1	A necessidade da gestão correta	15
1.2.2	Alta demanda e potencial desperdiçado	16
1.3	Objetivos	17
1.4	Solução Proposta	18
1.5	Justificativa	19
1.5.1	Demanda	20
1.5.2	Visão social	22
1.5.3	Vantagens do Sistema NewGen	24
1.6	Análise de concorrência	28
1.6.1	Agendamento Programático	30
1.6.2	Ferramentas de Gerenciamento	30
1.6.3	Locação de Equipamentos	31
1.6.4	Funcionalidades Adiadas	31
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
3.5.1.1	Primeiro Semestre	43
3.5.1.2	Segundo Semestre	45
3.6	Definição de Entregas	50

3.7	Tecnologias utilizadas	50
3.8	Arquitetura do sistema	52
3.8.1	Configurações do Servidor EC2	53
3.8.2	Comunicação Entre Ambientes	53
3.8.3	Arquitetura do servidor	54
3.8.4	Escalabilidade	54
3.9	Testes	55
3.9.1	Ferramentas de Testes	55
3.10	Modelagem do Banco de Dados	55
3.11	Coding Convention	56
3.11.1	Back-End	56
3.11.2	Front-End	59
3.12	Design Patterns	61
3.13	Integração continua	63
3.14	Product Backlog	64
3.15	Histórias de Usuário	64
3.15.1	Épicos	65
3.15.2	Histórias Divididas	66
3.16	Análise de requisitos	67
3.16.1	Requisitos Funcionais	68
3.16.2	Requisitos Não Funcionais	70
3.17	Regras de Negócio	71
3.18	Viabilidade Financeira	72
3.19	Segurança da Informação	74
3.20	Escolhas e Descartes	78
3.20.1	White-Label para SaaS	78
3.20.2	Alteração da lógica do banco	78
3.20.2.1	Uso do ORM	78
3.20.2.2	Adição de novas colunas	79
3.20.2.3	Estratégia de modelagem	79
3.20.3	Projeção para Terceiro Semestre	79
4	FUNCIONALIDADES	80
4.1	Cadastro Inicial	80
4.1.1	Estudos	80
4.1.2	Implementação	80
4.1.3	Revisões	82
4.1.4	Telas	82
4.2	Cadastro e Gerenciamento de Instâncias de Coworking	85

4.2.1	Estudos	85
4.2.2	Implementação	85
4.2.3	Revisões	86
4.2.4	Telas	86
4.3	Recuperação de Senha	87
4.3.1	Estudos	87
4.3.2	Implementação	88
4.3.3	Revisões	88
4.3.4	Telas	88
4.4	Reserva	92
4.4.1	Estudos	92
4.4.2	Implementação	92
4.4.3	Revisões	93
4.4.4	Telas	93
4.5	Pagamento	97
4.5.1	Estudos	97
4.5.2	Implementação	97
4.5.3	Revisões	98
4.5.4	Telas	98
4.6	Autorização de Entrada	99
4.6.1	Estudos	99
4.6.2	Implementação	99
4.6.3	Revisões	99
4.6.4	Telas	99
4.7	Personalização de Tela	102
4.7.1	Estudos	102
4.7.2	Implementação	102
4.7.3	Revisões	105
4.7.4	Telas	105
4.8	Relatórios	110
4.8.1	Estudos	110
4.8.2	Implementação	110
4.8.3	Revisões	110
4.8.4	Telas	111
5	CONSIDERAÇÕES FINAIS	112
5.1	Primeiro Semestre	112
5.2	Segundo Semestre	112

REFERÊNCIAS	114
GLOSSÁRIO	117
APÊNDICES	120
APÊNDICE A – RELATÓRIOS DAS SPRINTS	121
A.1 Primeiro Semestre	121
A.2 Segundo Semestre	130
APÊNDICE B – MER E DER	140
APÊNDICE C – DICIONÁRIO DE DADOS	143
APÊNDICE D – PUBLICAÇÕES DO BLOG	148
APÊNDICE E – ATAS DE REUNIÃO	163
APÊNDICE F – ESTUDO INFORMAL SOBRE MIGRAÇÃO DO BANCO	203

1 Introdução

Este é 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.

O objetivo da equipe é construir uma aplicação web, um *Software as a Service (SaaS)* que auxilie nosso público alvo (administradores de coworings) a gerenciar seu ambiente. Em outras palavras, traremos facilidades 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.

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.

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 [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 empresarial brasileiro. No entanto, 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 \(2019a\)](#), 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, 2019b](#)). 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 \(2019b\)](#), 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, 2019a](#)), 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, 2019b](#))).

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 aplicação é 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.

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 a vantagem que o cliente pode enxugar os prejuízos caso não goste do produto, pois diferente de uma compra única, é só cancelar a assinatura.

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.

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 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 tenha um outro sistema para realizar a transição, então os arquivos do back-end (como extension.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, atualizações, infraestrutura e manutenção do sistema que constariam dentro do seu pacote de assinatura.

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.

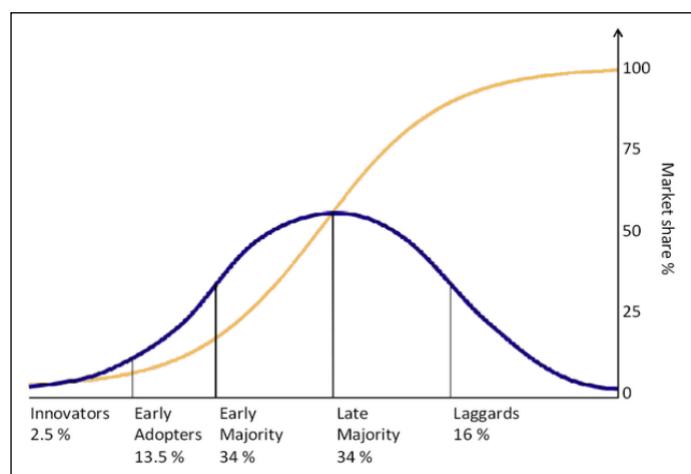
1.5 Justificativa

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

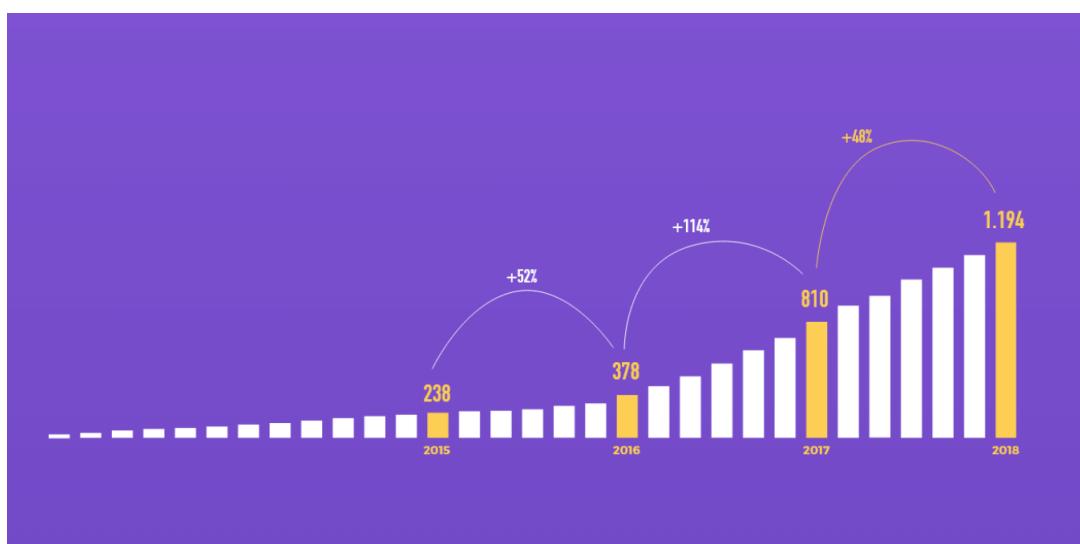
O coworking é um mercado em expansão e por se tratar de um modelo relativamente recente, ainda tem uma boa taxa de crescimento antes que se torne um modelo defasado. Traçando um paralelo com a famosa curva difusão de um produto (Figura 1), dividida em: Inovadores, adeptos iniciais, maioria inicial, maioria 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 ([2019b](#))

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



Fonte: Censo Coworking do COWORKING BRASIL (2019b)

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.

Nos estágios iniciais de um mercado crescente, muitos investidores conseguem 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 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 (2019b) 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 (estaçao 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.

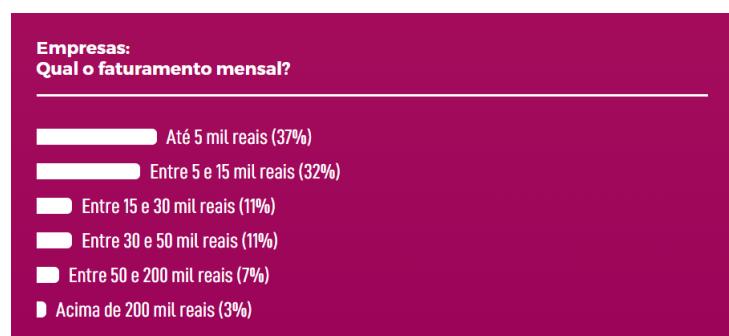
1.5.2 Visão social

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 a década de 2000, contextualizada por crises como o 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 (2019a) 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.

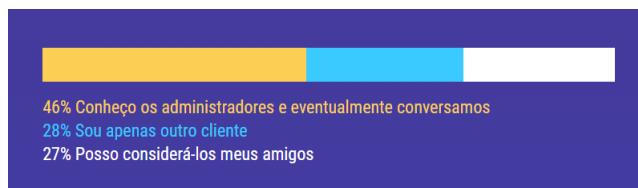
A Figura 4 demonstra como uma expressiva fração do negócios clientes do coworking ainda são de micro ou pequeno porte.

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



Fonte: Censo Coworkers do COWORKING BRASIL (2019a)

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 \(2019a\)](#)

Além disso, na [Figura 5](#) com 27% considerando o proprietário do estabelecimento um amigo, é apontado um outro acontecimento detalhado por [COWORKING BRASIL \(2019a\)](#): 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 \(2019b\)](#), 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 \(2019a\)](#) 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.

Figura 6 – Avaliação de serviços essenciais no coworking para os clientes de coworking, Brasil, 2018



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

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, 2019a](#)). 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 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 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 ([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 [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 ([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 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 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 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 espaços para divulgar o seu website, onde ocorre a interação com o cliente final para as locações digitais.

Vale ressaltar que esta funcionalidade do NewGen é um diferencial em relação a sistemas de exposição de espaços de aluguel. Sistemas de gerenciamento desenvolvidos sob demanda, como o [2Work](#), [ConvivaCoworking](#) e [GoWork](#), retratados na [Quadro 1](#), possuem o mesmo ponto positivo.

Posteriormente é válido também o estudo de parcerias com tais ferramentas de exposição de espaços, que propunham que os clientes dela utilizem o nosso sistema para gerenciamento e locação digitalizada do coworking. Ao fazer isso, a empresa por trás dessa ferramenta, receberia uma parcela de comissão por trazer um novo cliente ao NewGen. Ao mesmo tempo que a empresa de exposição estaria completando o seu sistema com algo que não é fornecido no dela: gerenciamento completo do negócio do coworking; já que fornecer um sistema de gerenciamento não é o foco dela.

- **Redução de custos e riscos ao cliente.**

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 [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”, 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 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 “assine e faça o seu negócio” com um passo a passo guiado dentro da sua gama de funcionalidades. Se bem 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 na [1.1 Contextualização](#), e mais detalhadamente em [2.1 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 [*Minimum Viable Product \(MVP\)*](#) 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 programaticamente.

Os concorrentes selecionados para o comparativo foram: o [2Work](#), [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 de exposição para as pessoas disponibilizarem seus espaços disponíveis para aluguel de coworking, foram analisados o [CoworkingBrasil](#) e [WebEscritórios](#).

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, 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 Programático

A funcionalidade *agendamento 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 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 customizável para empresas que querem 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](#), 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 [WebEscritórios](#), 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 um responsável da empresa para pedido de equipamentos extras nas suas salas.

O sistema 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 Adiadas

O recurso do *chatbot* foi 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 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 [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.

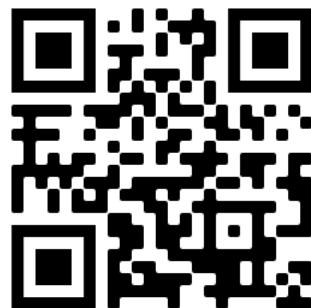
3 Desenvolvimento do projeto

Neste capítulo, trataremos a evolução do desenvolvimento do projeto da equipe NewGen, referente a aplicação. 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 de segurança, e outros pontos importantes para compreensão da aplicação.

3.1 Links do projeto

Para apresentar o desenvolvimento da aplicação, nesta seção serão disponibilizados os links do projeto, 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.



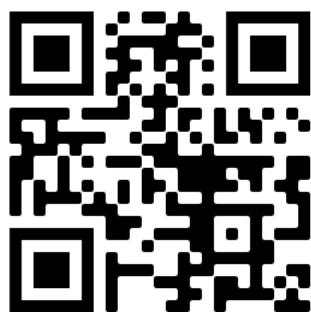
Fonte: Os autores

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 \(2018\)](#), este 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, com 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 ganha novas funcionalidades conforme a finalização de cada Sprint.

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 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

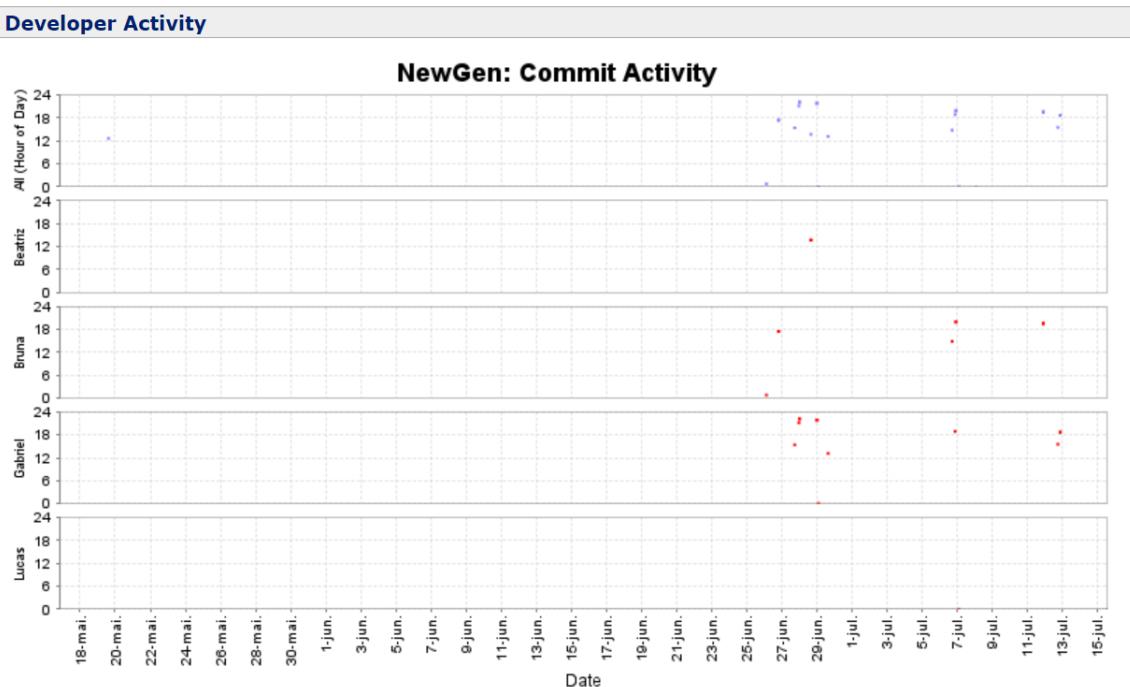
Nesta sub-seção constam os gráficos estatísticos do projeto em relação ao material versionado pelo SVN. Os gráficos que serão referidos nesta seção são referentes a valores e dados que envolvem tanto códigos quanto documentos desenvolvidos pela equipe. Os gráficos foram gerados com a ferramenta StatSVN, 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.

3.5.1.1 Primeiro Semestre

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.

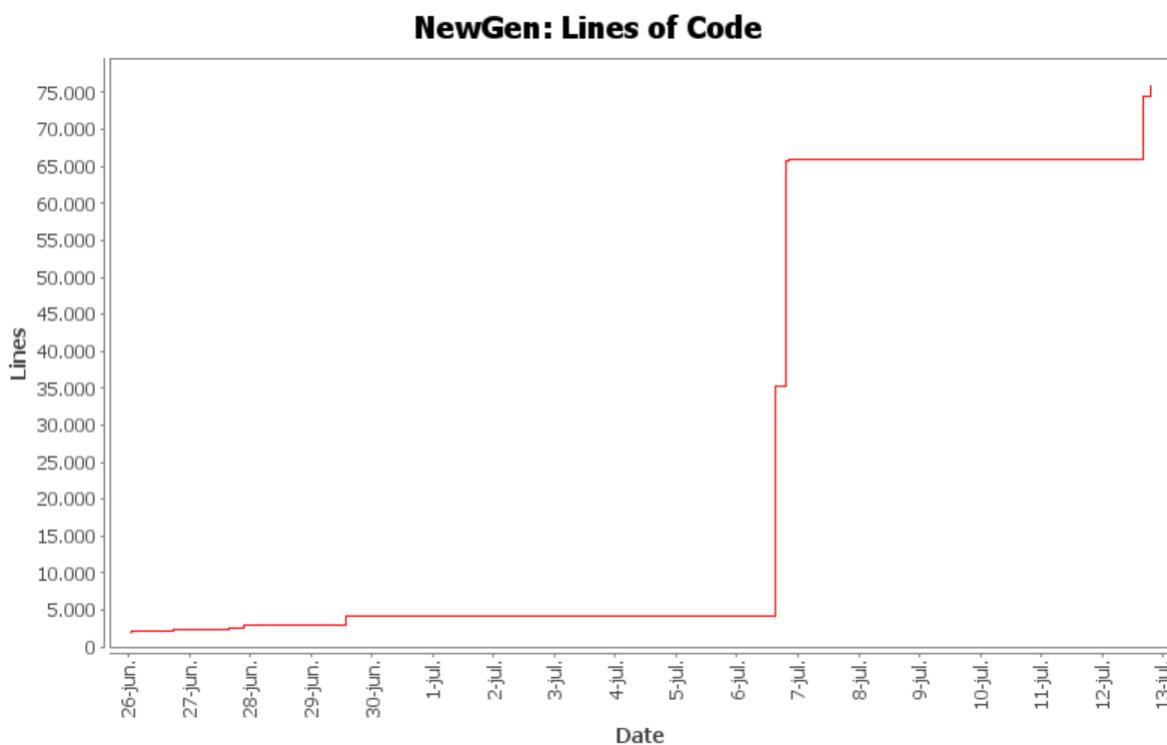
Figura 14 – Dispersão de commits



Fonte: Os Autores

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

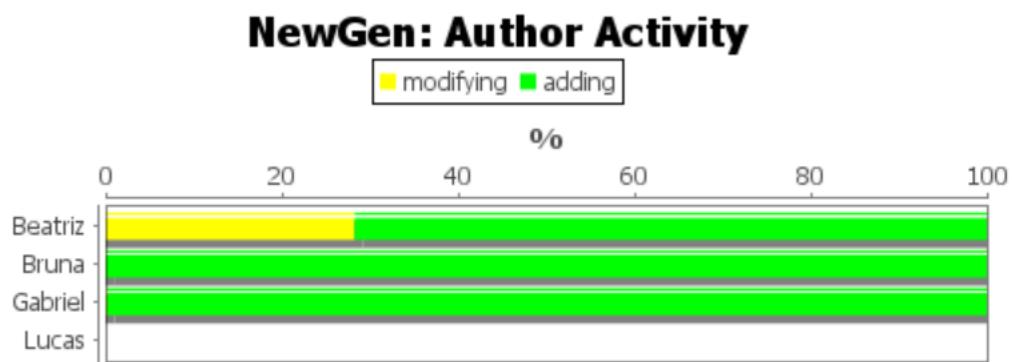
Figura 15 – Quantidade de linhas de código



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.

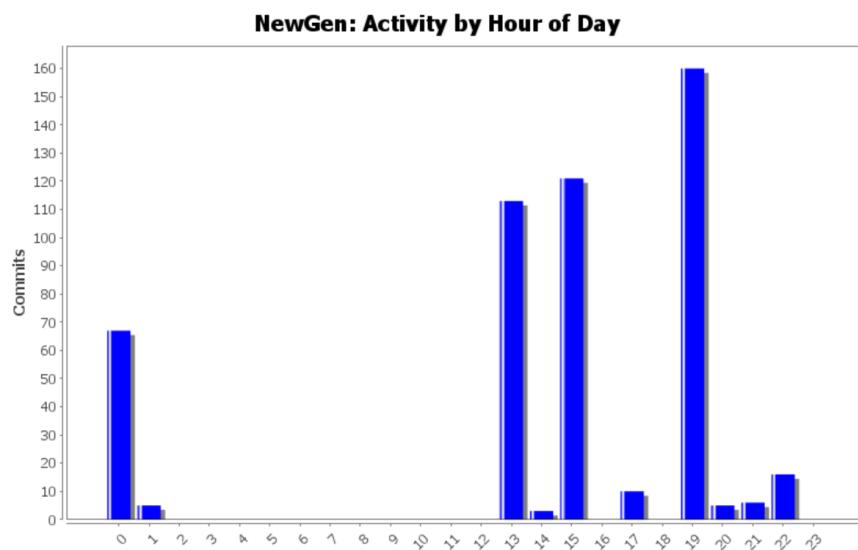
Figura 16 – Tipos de atividades dos autores



Fonte: Os Autores

A Figura 17 apresenta os horários onde houveram mais commits.

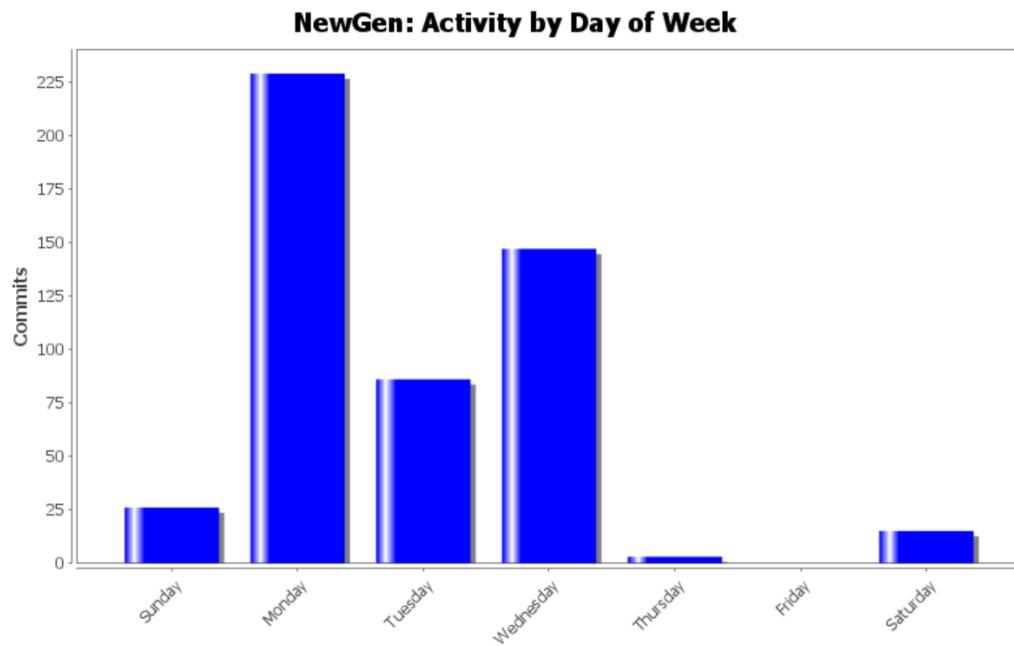
Figura 17 – Commits por horário



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



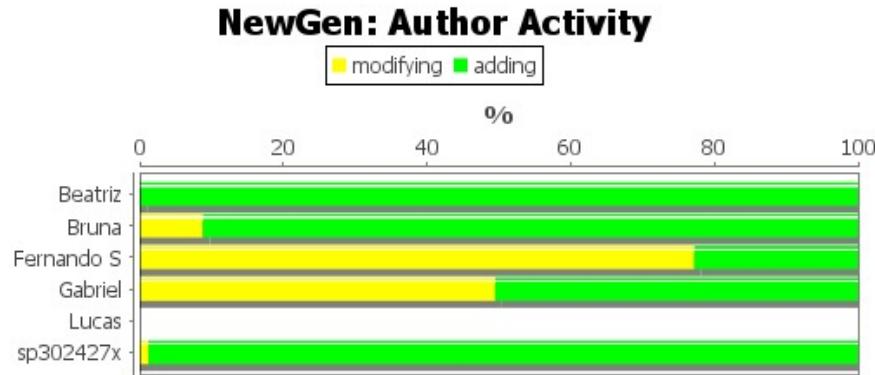
Fonte: Os Autores

3.5.1.2 Segundo Semestre

Para o segundo semestre, foram extraídas novas estatísticas do SVN, na data de 18 de novembro de 2021. Para o contribuinte Lucas, houve problemas de conexão com o

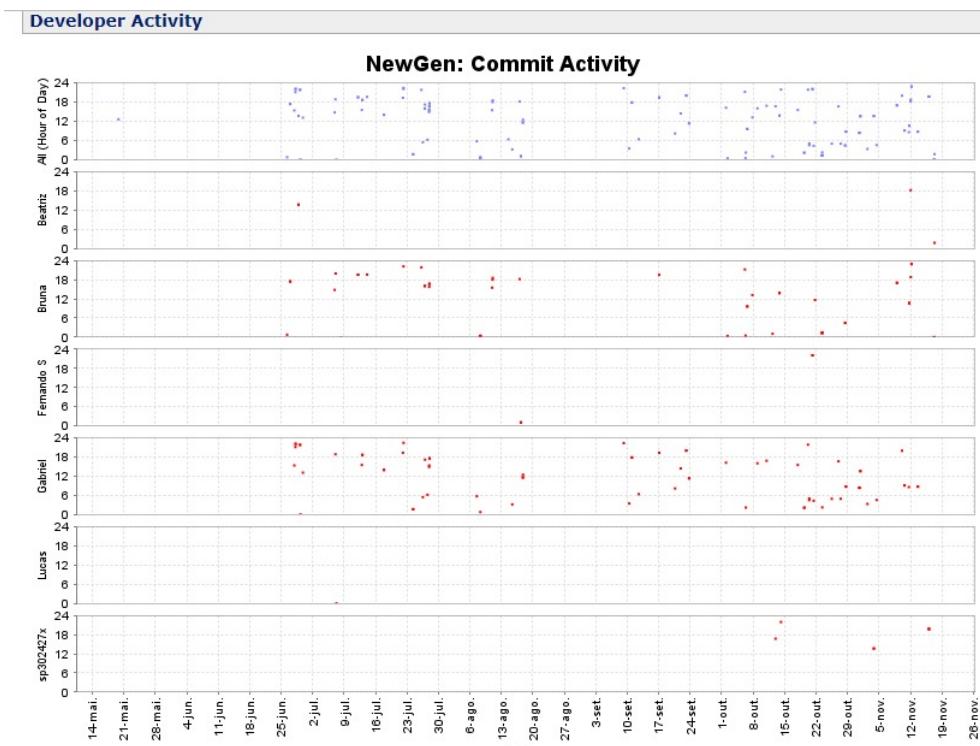
SVN por questão de usar um MacBook, portanto suas atualizações foram feitas por outras pessoas da equipe.

Figura 19 – Atividade por autor



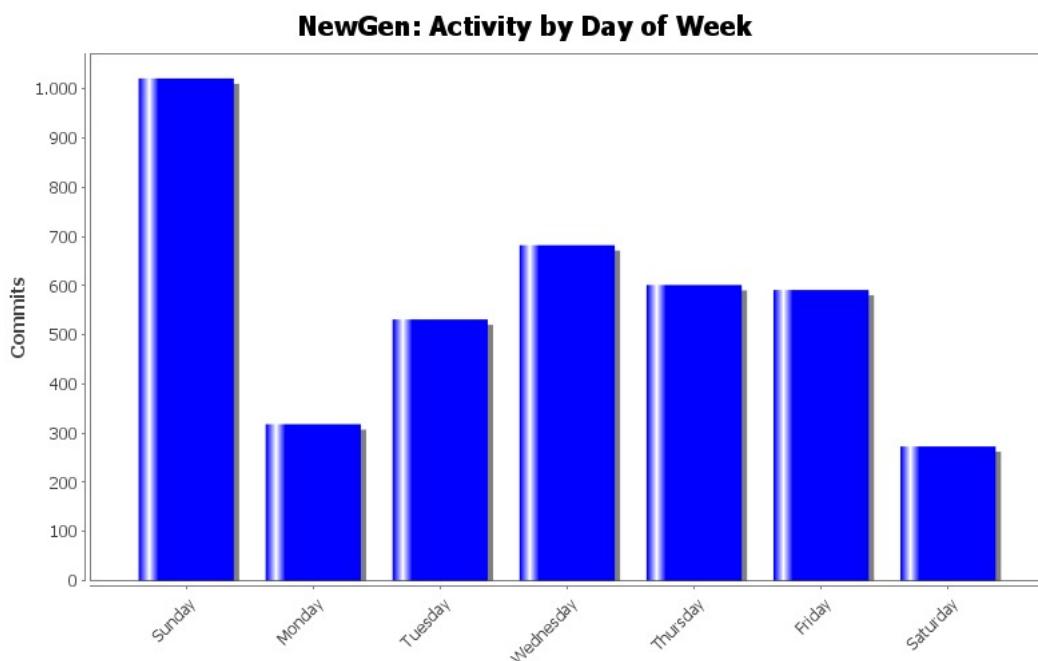
Fonte: Os Autores

Figura 20 – Atividade por commit em dispersão



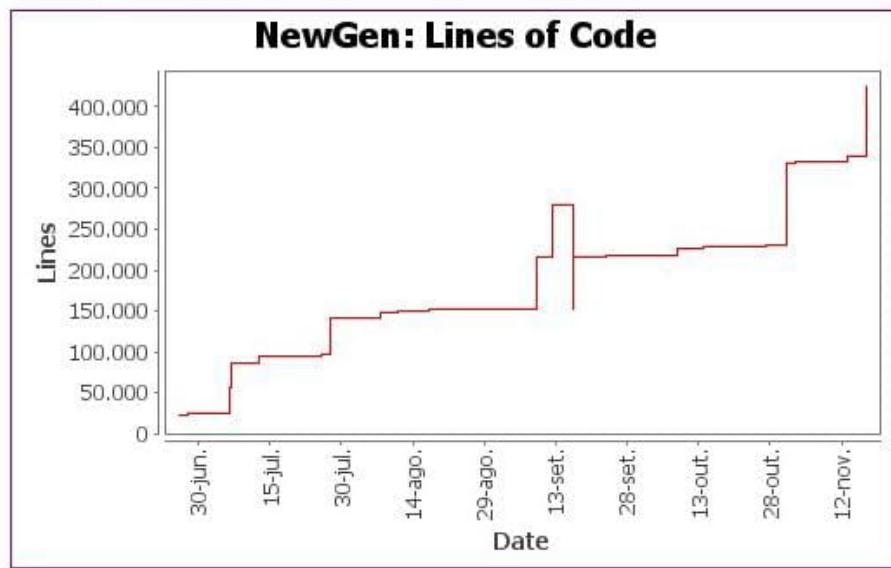
Fonte: Os Autores

Figura 21 – Atividade por dia da semana



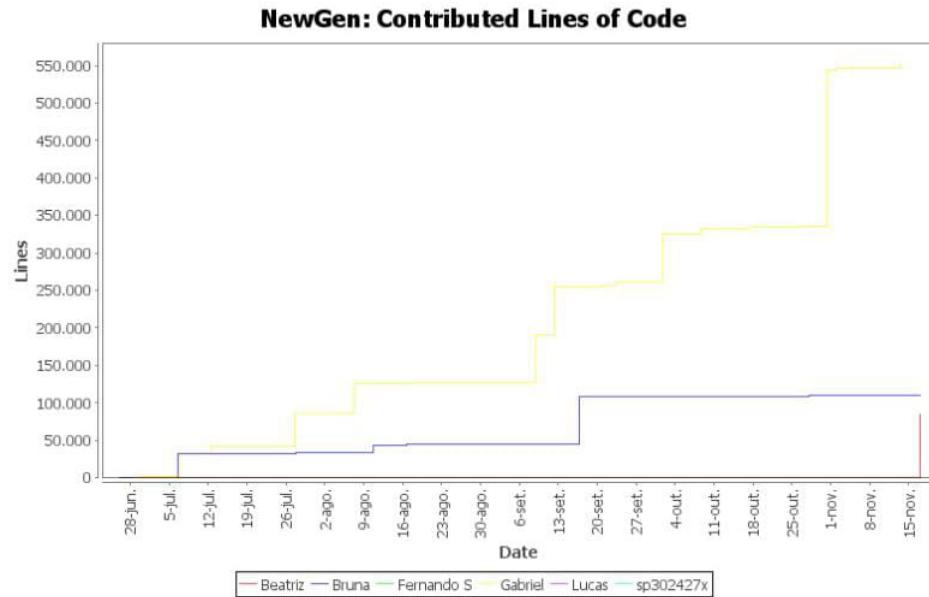
Fonte: Os Autores

Figura 22 – Quantidade de código total feito



Fonte: Os Autores

Figura 23 – Contribuição por linha de código



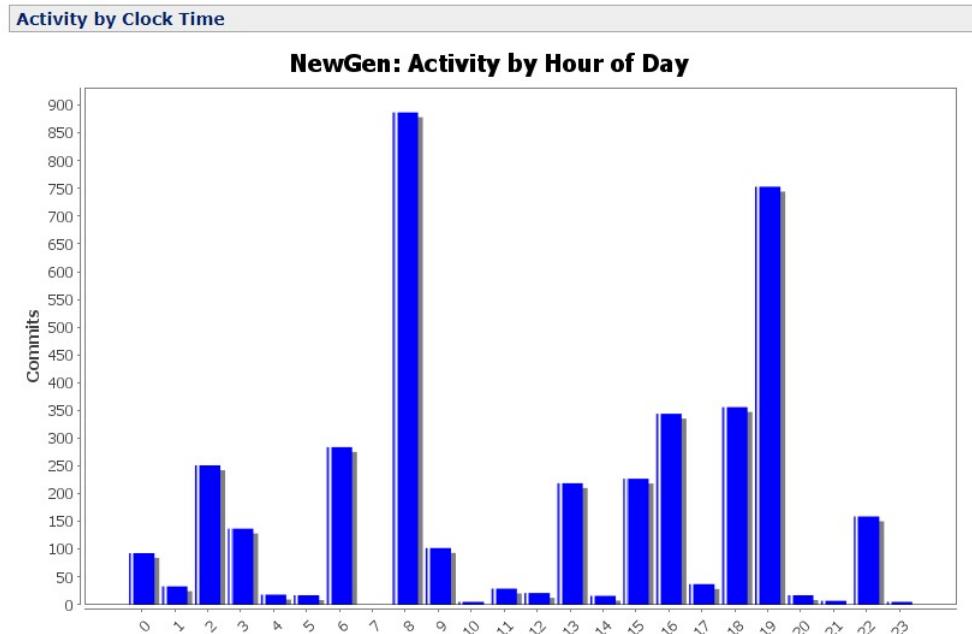
Fonte: Os Autores

Figura 24 – Desenvolvedor com maior contribuição do mês

Developer of the Month		
Month	Author	Lines
<u>November 2021</u>	<u>Beatriz</u>	85086
<u>October 2021</u>	<u>Gabriel</u>	283469
<u>September 2021</u>	<u>Gabriel</u>	133750
<u>August 2021</u>	<u>Gabriel</u>	41868
<u>July 2021</u>	<u>Gabriel</u>	84036
<u>June 2021</u>	<u>Gabriel</u>	1671

Fonte: Os Autores

Figura 25 – Atividade por hora do dia



Fonte: Os Autores

Figura 26 – Edições de código por pessoa

Author	Author Id	Changes	Lines of Code	Lines per Change
Gabriel	Gabriel	2923 (76.3%)	552409 (73.9%)	188.9
Bruna	Bruna	568 (14.8%)	109519 (14.7%)	192.8
Beatriz	Beatriz	247 (6.5%)	85224 (11.4%)	345.0
sp302427x	sp302427x	11 (0.3%)	246 (0.0%)	22.3
Fernando S	Fernando S	18 (0.5%)	79 (0.0%)	4.3
Lucas	Lucas	62 (1.6%)	0 (0.0%)	0.0
Totals		3829 (100.0%)	747477 (100.0%)	195.2

Fonte: Os Autores

Figura 27 – Linhas de código por pessoa

Author	Lines of Code
Gabriel	552409 (73.9%)
Bruna	109519 (14.7%)
Beatriz	85224 (11.4%)
sp302427x	246 (0.0%)
Fernando S	79 (0.0%)
Lucas	0 (0.0%)

Fonte: Os Autores

3.6 Definição de Entregas

Após mapear e descrever todas as funcionalidades que nossa aplicação terá, foram definidos 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*, Segundo Semestre ou Terceiro Semestre).

Quadro 2 – Escopo do projeto.

Funcionalidades	POC	MVP	Segundo Semestre	Terceiro Semestre
Cadastro Inicial	X			
Login		X		
Reserva		X		
Cadastro de Coworking			X	
Recuperação de Senha			X	
Aluguel			X	
Pagamento			X	
Autorização de Entrada			X	
Relatórios Periódicos			X	
Transferência de Responsabilidade			X	
Sistema de Fidelidade				X
Controle de Estoque				X
Lista de Espera				X
Locação de Máquinas			X	

Fonte: Os Autores

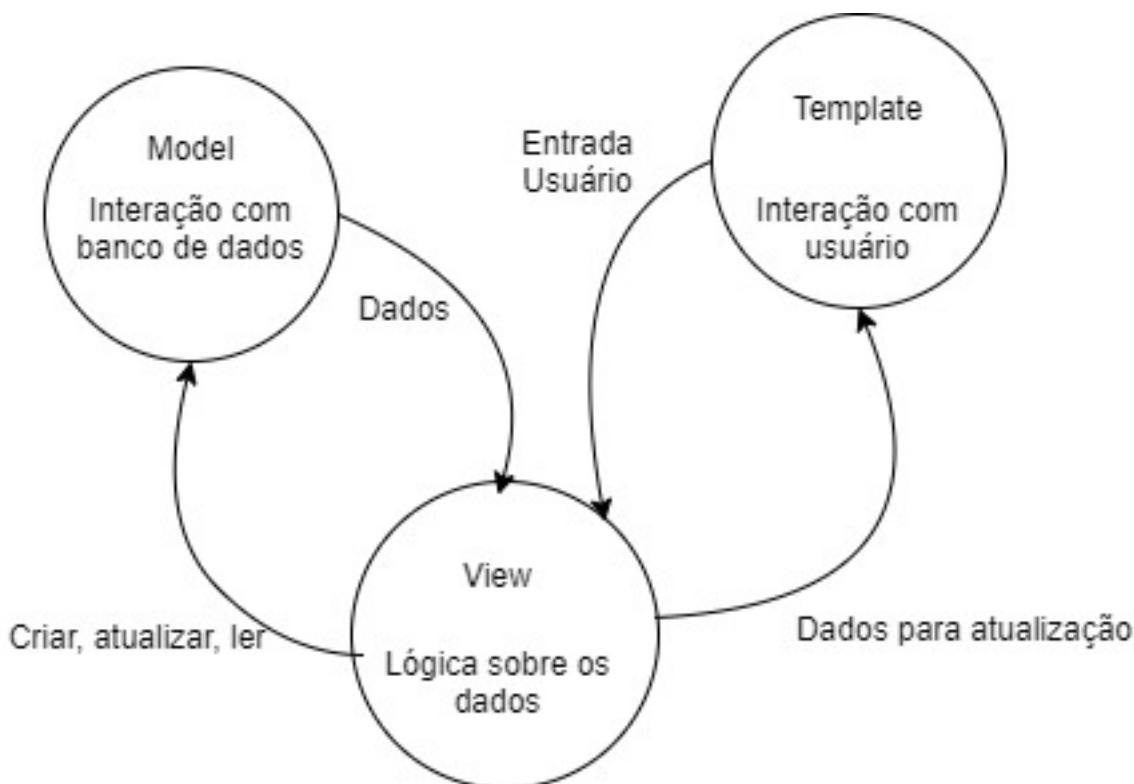
3.7 Tecnologias utilizadas

- **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:

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.

Figura 28 – Modelo MTV no Django



Fonte: [Caderno de Laboratório \(2021\)](#)

- **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.

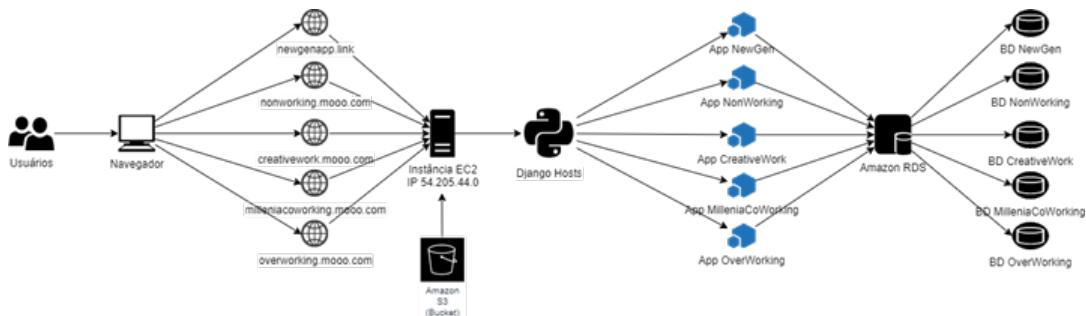
- Serviços **Amazon Web Services (AWS)**: A aplicação deve estar disponibilizada através da plataforma de serviços web da Amazon, os serviços utilizados devem ser:
 - Amazon CodePipeline**: Serviço de integração contínua para automatização do processo de lançamento da aplicação.
 - Amazon CodeBuild**: Serviço de integração contínua que compila o código-fonte, realiza os testes e produz pacote de software preparado para implantação.
 - Amazon CodeDeploy**: Serviço gerenciado para automatizar implantação de software e servirá para realizar implantação na instância do EC2.
 - 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 [AWS](#).
- **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 [AWS](#).
- **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.
- **NGINX**: Trata-se de um software servidor web de código aberto. Com esta tecnologia, poderemos utilizar como proxy reverso e balanceador de carga HTTP.
- **Gunicorn**: Servidor de aplicação dedicada Python WSGI HTTP, o qual deveremos utilizar para servir diretamente a 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), possuindo também a utilização do banco de dados MySQL Server via Amazon Web Services (AWS) RDS e com os arquivos armazenados em um bucket na Amazon Web Services (S3).

O servidor alocado, realiza o processo de requisições de usuário, conforme [Figura 29](#):

Figura 29 – Servidor para processo de requisições de usuário.



Fonte: Os Autores

O Django Hosts recebe a requisição do usuário e instancia a aplicação que corresponde ao domínio requisitado, que por sua vez, a aplicação instanciada realiza a comunicação com o serviço RDS da Amazon e instancia o banco de dados correspondente.

O bucket no serviço S3 da Amazon contém os arquivos da aplicação e, portanto, é integrado com a instância EC2 por meio de uma política de acesso criada no IAM (recurso de gerenciamento de acesso dos serviços da AWS).

3.8.1 Configurações do Servidor EC2

O servidor Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - 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

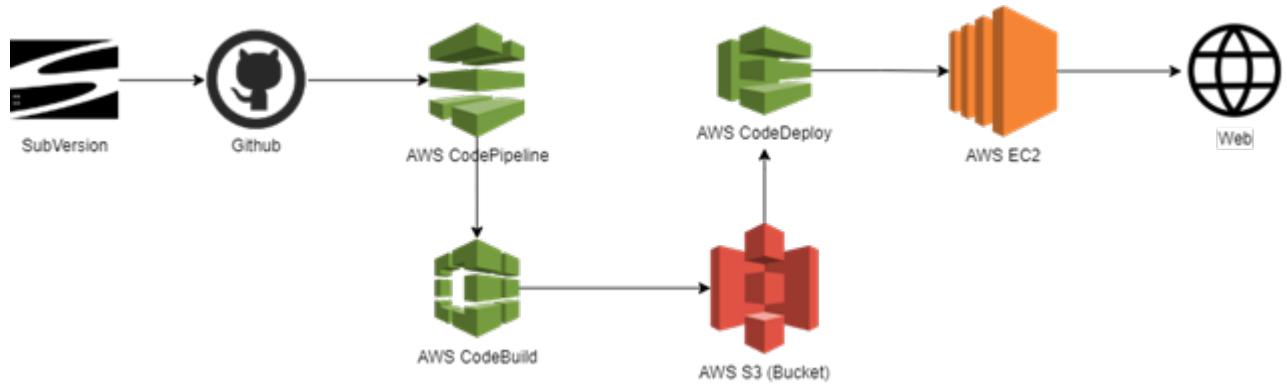
Armazenamento de disco: 8 GB

Processamento: 2.5 GHz

3.8.2 Comunicação Entre Ambientes

A aplicação deve seguir o modelo Figura 30 para um desenvolvimento confiável durante todo o processo:

Figura 30 – Esquematização de fluxo para comunicação entre ambientes.

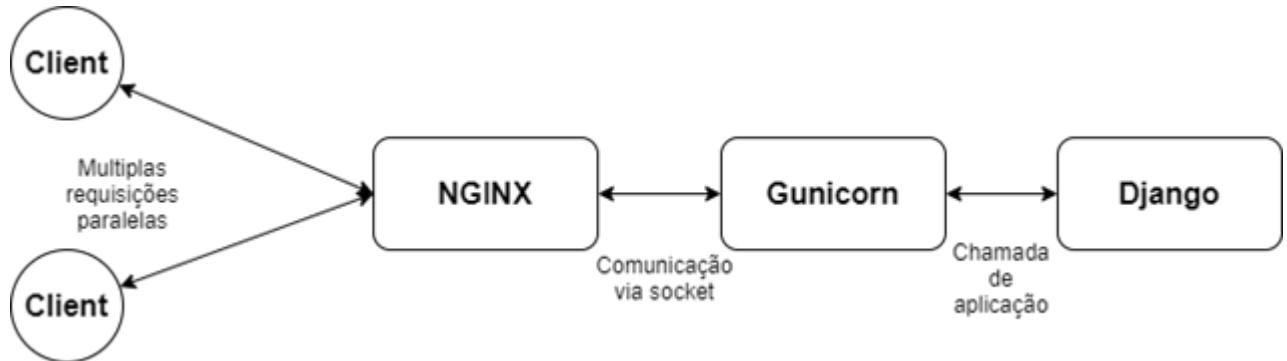


Fonte: Os Autores

3.8.3 Arquitetura do servidor

Conforme [Figura 31](#), o NGINX deve se comunicar com as requisições exteriores, ou seja, de client, enquanto o Gunicorn deve executar a aplicação Django e retornar as respostas:

Figura 31 – Esquematização de fluxo para comunicação NGINX - Gunicorn - Django.



Fonte: Os Autores

3.8.4 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 facilização.

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 Testes

Para garantir a entrega de um sistema robusto, que trata a maior quantia de erros possíveis, é necessária a aplicação de testes na maior quantidade de áreas possíveis. Serão listados as ferramentas utilizadas, e quais os testes rodados no sistema da NewGen.

3.9.1 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.

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

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 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.

Figura 32 – Exemplo de Indentação 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

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

Figura 33 – 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

Para seguir o padrão da PEP8, foi utilizado o pacote *pycodestyle*, que faz a verificação de estilo em todos os arquivos .py, conforme exemplo da figura [Figura 34](#).

Figura 34 – Exemplo de Identação em Array.

```

PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle app
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle com
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle loo
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle loo
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle sta
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle tem
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8> pycodestyle tes
PS C:\Users\k\Documents\Faculdade\PI2A6\Desenvolvimento\aplicacao_pep8>

```

Fonte: Os Autores

- 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 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 35 – 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 36 – Exemplo de Imports Simples.

```
# Correto:  
import os  
import sys  
  
# Incorreto:  
import sys, os
```

Fonte: Os Autores

Figura 37 – 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 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 começa pela nomenclatura de tudo o que for criado no código (variáveis, métodos, funções); não devem ser genéricas e precisam indicar claramente ao que aquele item 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 de uso interno, ela se inicia com essa sublinhada, como seria em uma linguagem de encapsulamento 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.

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 38 – 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 39 – 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 40 – Exemplo de Generic.

```
*  
  box-sizing: border-box
```

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.

Figura 41 – Exemplo de Elements.

```
li  
  list-style: none
```

Fonte: Os Autores

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

Figura 42 – Exemplo de Object.

```
.container  
  margin: 0 auto  
  max-width: 480px
```

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.

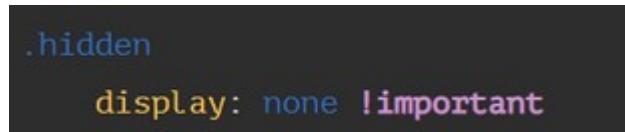
Figura 43 – Exemplo de Components.

```
.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 44 – Exemplo de Trumps.



Fonte: Os Autores

3.12 Design Patterns

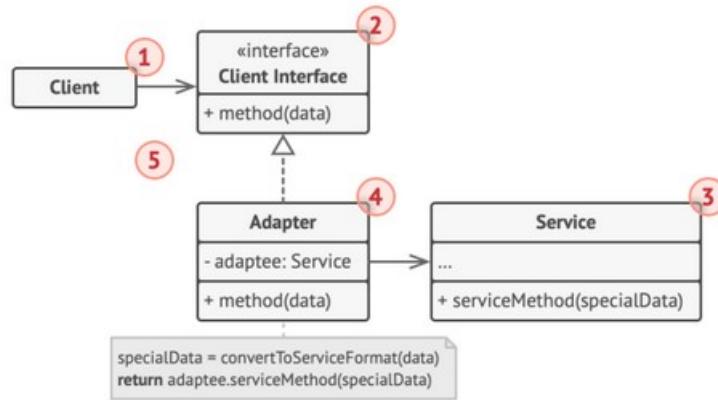
É de suma importância seguir uma estrutura de padronização Design Pattern no processo de desenvolvimento de um projeto. Ele precisa ser estruturado nesse padrão para ajudar a manter uma coerência no desenvolvimento e ajudar na solução de possíveis problemas no decorrer do mesmo.

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.

- **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 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.

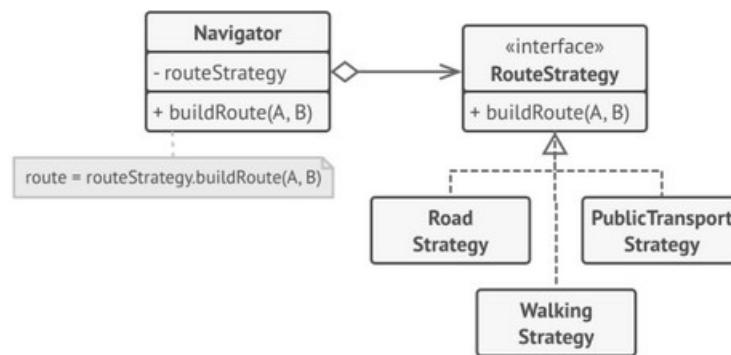
Figura 45 – Exemplo de um Design Pattern Adapter

Fonte: [Refactoring.guru \(2021a\)](https://refactoring.guru/design-patterns/adapter)

- **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.

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 46 – Exemplo de um Design Pattern Strategy

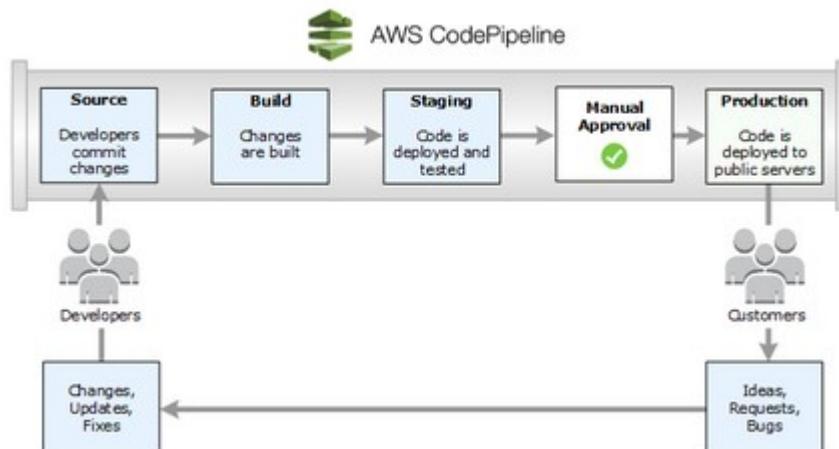
Fonte: [Refactoring.guru \(2021b\)](#)

- **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 47 – 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.

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

Conforme Figura 48, é possível visualizar uma esquematização de uma sprint exemplo:

Figura 48 – Esquema de Sprint na Ferramenta Jira

The screenshot shows the Jira interface for a project titled 'Projeto Integrado: TCC / quadro-PIT'. At the top, there are filters for 'Somente meus itens' and 'Recentemente atualizados', and a 'Share' button. Below the header, a sidebar on the left lists 'Sprint semana 4' (5 items) and 'Backlog' (5 items). The 'Sprint semana 4' section shows five tasks with checkmarks and their respective assignees and IDs: PIT-63, PIT-85, PIT-86, PIT-61, and PIT-84. The 'Backlog' section shows four tasks: 'Criar relatório da POC' (PIT-92), 'Criar vídeo da POC' (PIT-93), 'Realizar ajustes da documentação final' (PIT-94), and 'Criar o front-end da tela de reserva' (PIT-95).

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.

3.15.1 Épicos

- **POC**
 - Eu como administrar 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.
- **Segundo Semestre**
 - 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.
- **Terceiro Semestre**
 - 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;
 - 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.

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.

- **Segundo Semestre**

- Eu como administrador quero visualizar relatórios e estatísticas para ter informações cruciais sobre o meu negócio;
- Eu como administrador, gerente ou recepcionista quero ter uma agenda para verificar as reservas de hoje;
- Eu como gerente, quero uma interface que me permita ler QR Codes e aprovar a entrada de clientes;
- Eu como gerente quero personalizar minha página para atrair o público;
- Eu como cliente quero pagar minha reserva através de uma fonte confiável.
- Eu como cliente quero requisitar meu QR Code a qualquer momento.
- Eu como cliente quero poder acessar meu local reservado mostrando apenas o QR Code.

- **Terceiro Semestre**

- 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.

3.16 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 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 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.

Quadro 3 – Relação de requisitos funcionais

Código	Nome	Descrição
RF01	Gerenciamento de usuários	O sistema deve permitir a criação, leitura e edição de usuários.
RF02	Autenticar usuários	O acesso ao sistema deve ser protegido por mecanismo de autenticação das informações do usuário.
RF03	Gerenciar salas e recursos	O sistema deve permitir a criação, leitura, edição e exclusão de espaços e recursos (datashow, dvd, quadro branco, etc.).
RF04	Autenticação de entrada e saída	O sistema enviará um QR code ao usuário (locatário) para validação de entrada e saída do espaço alocado, que será validado pela recepcionista do local.
RF05	Exibir disponibilidade de espaço	O sistema deverá apresentar aos usuários status de espaços, datas e horários, além dos tipos de espaços disponíveis para locação.
RF06	Cancelar locação	O sistema deve ter a opção de cancelamento da locação, o período de cancelamento será de responsabilidade do cliente (locador).
RF07	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.
RF08	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.
RF09	Transferência de responsabilidade	O sistema terá a opção de transferência da responsabilidade da locação. Onde o usuário (locatário) indica a pessoa para quem essa responsabilidade está sendo atribuída e este deve confirmar a atribuição.
RF10	Alerta de tempo limite	O sistema mostrará um pop-up quando faltar 15 minutos para o fim do tempo que foi reservado.
RF11	Confirmação de alerta de tempo limite	O usuário poderá apenas confirmar no pop-up ou solicitar mais horas para dar continuidade.
RF12	Solicitação de tempo extra	Se o horário seguinte estiver disponível, o usuário pode continuar na mesma sala, caso contrário, ele terá opção de selecionar outra sala para dar continuidade a sua locação.
RF13	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
RF14	Sistema de Fidelidade	O sistema terá a opção de fidelizar clientes.
RF15	Relatórios	O sistema disponibilizará ao Administrador, relatórios de Volumetria de locação, status de ocupação passadas, presentes e futuras.
RF16	Pacote de horas	O sistema disponibilizará a opção de pacote de horas, onde o locatário poderá reservar horas semanais, quinzenais e mensais.

Fonte: Os Autores

Quadro 4 – Continuação da listagem de requisitos funcionais

Código	Nome	Descrição
RF17	Recebimento de pagamento	O sistema terá integração com mercadoPago para que seja confirmado o recebimento de pagamento do locatário.
RF18	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.
RF19	Tela de Recepcionista	O Sistema deve apresentar a opção de verificar os status de locação do dia atual e dos próximos dias, opção de validação de pagamento, validação de QR code do locatário.
RF20	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.
RF21	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 dias.

Fonte: Os Autores

3.16.2 Requisitos 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.

Fonte: Os Autores

3.17 Regras de negócio

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.

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.

Será listado também o nome uma breve descrição da regra a ser seguida e os requisitos funcionais relacionados.

Quadro 6 – Listagem de regras de negócios.

Código	Nome	Descrição	Relacionado
RN01	Obrigatoriedade de login para acesso ao sistema	Para acessar o sistema será necessário a realização de login do usuário, em caso de primeiro acesso, é necessária a realização do cadastro.	RF02, RF03
RN02	Perfis de usuários	O Sistema terá 4 perfis diferentes: Administrador (locador), Usuario comum (locatario), Gerente, recepcionista.	RF19, RF20, RF21, RF22
RN03	Telas por perfil	Cada perfil possui um tipo de fluxo de telas personalizadas.	RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF19, RF20, RF21, RF22
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.	RF09
RN05	Realizar locação	A locação será realizada somente após confirmação de pagamento da mesma.	RF6, RF07, RF08, RF09
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.	RF01
RN07	validação de QR code	QR code para validação de entrada e saída do locatário.	RF05
RN08	Lista de espera	O sistema deve ser responsivo, se adequando ao dispositivo de origem do acesso.	RF22
RN09	Pagamento do locatário	O locatário terá a opção de pagamento via boleto, PIX ou cartão de crédito.	RF18

Fonte: Os Autores

3.18 Viabilidade Financeira

Após o usuário adquirir o serviço de gerenciamento da NewGen, 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 NewGen. 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 de assinatura seria por volta de R\$ 509,00. 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 sistema da NewGen ser 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; 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 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 NewGen, 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\$ 250,00, em um plano mais simples, com menos funcionalidades, podendo chegar até R\$ 800,00, conforme exemplificado na [Figura 49](#).

Figura 49 – Planos de assinatura do sistema da NewGen

Plano Básico	Plano Pro	Plano Prime
R\$ 109,00 / mês	R\$ 509,00 / mês	R\$ 800,00 / mês
Gestão de clientes e funcionários	Gestão de clientes e funcionários	Gestão de clientes e funcionários
Gestão de salas e equipamentos	Gestão de salas e equipamentos	Gestão de saças e equipamentos
Agendamento On-line	Agendamento On-line	Agendamento On-line
Checkin e Checkout via QR-Code	Checkin e Checkout via QR-Code	Checkin e Checkout via QR-Code
Customização do sistema	Pacote de Horas	Customização do sistema
Pacote-de-Horas	Customização do sistema	Customização do sistema
controle-de-Estoque	Trasnferência de responsabilidade	Trasnferencia de responsabilidade
Fila-de-Espera	controle-de-Estoque	Fila de Espera
Trasnferencia de responsabilidade	Fila-de-Espera	Pacote de Horas
Assine Agora	Em breve	Em breve

Fonte: Os Autores

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.

3.19 Segurança da Informação

Nosso 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.

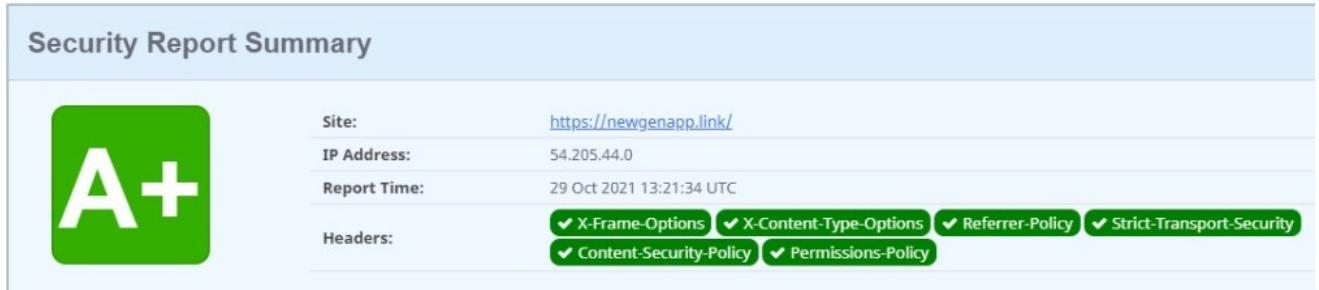
Sancionada em 2018, estabelece regras que as empresas devem seguir ao coletar, armazenar e utilizar os dados dos clientes, sendo passível de multas às empresas que não se adequarem à legislação.

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 segurança em banco de dados não seja comprometida por acidentes ou ameaças intencionais, é preciso garantir o bom funcionamento dos três pilares: integridade, disponibilidade e confiabilidade.

Outro ponto a destacar com questão de segurança é 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 consigam entender.

Como forma de provar a segurança do HTTPS, foram realizados scans dos domínios através do Security Headers, mostrando uma ótima nota que garante proteção dos dados.

Figura 50 – Security Headers do domínio *newgenapp*.

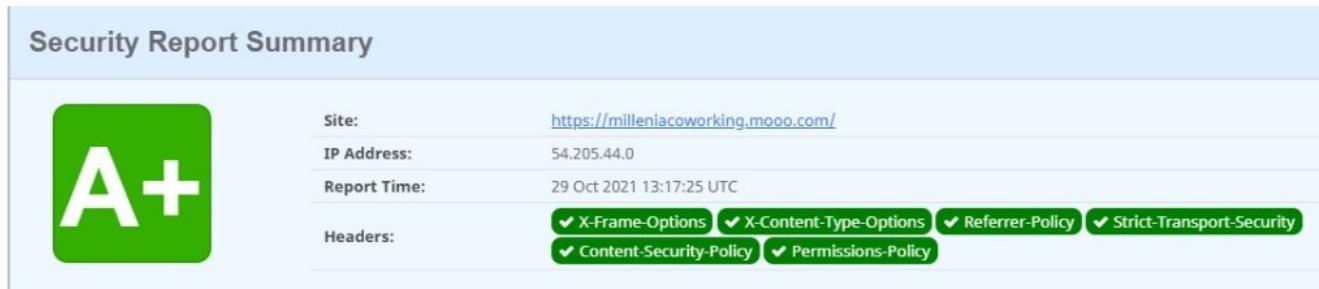


The screenshot shows a 'Security Report Summary' for the domain <https://newgenapp.link/>. The report is graded with an **A+**. Key details include:

- Site:** <https://newgenapp.link/>
- IP Address:** 54.205.44.0
- Report Time:** 29 Oct 2021 13:21:34 UTC
- Headers:** A list of checked items: X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Strict-Transport-Security, Content-Security-Policy, and Permissions-Policy.

Fonte: Os Autores

Figura 51 – Security Headers do domínio *milleniacoworking*.

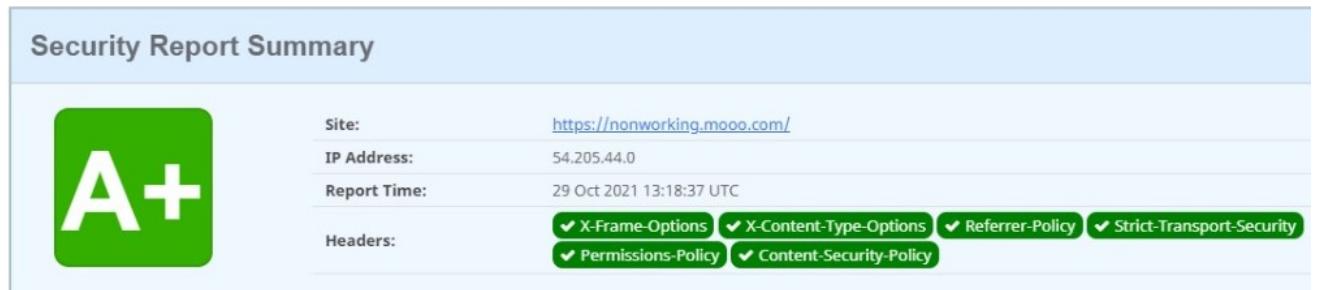


The screenshot shows a 'Security Report Summary' for the domain <https://milleniacoworking.mooo.com/>. The report is graded with an **A+**. Key details include:

- Site:** <https://milleniacoworking.mooo.com/>
- IP Address:** 54.205.44.0
- Report Time:** 29 Oct 2021 13:17:25 UTC
- Headers:** A list of checked items: X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Strict-Transport-Security, Content-Security-Policy, and Permissions-Policy.

Fonte: Os Autores

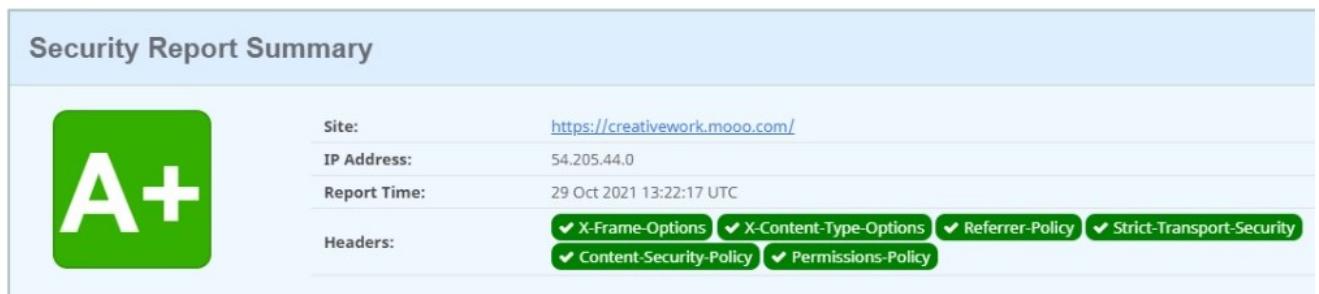
Figura 52 – Security Headers do domínio *nonworking*.



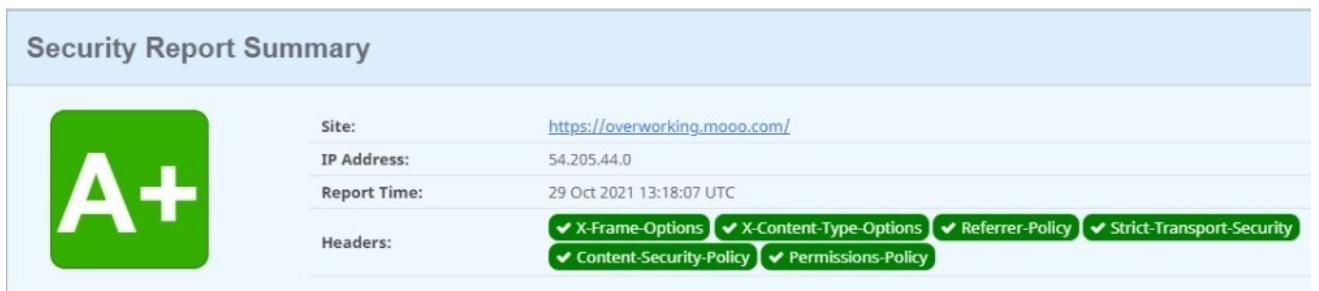
The screenshot shows a 'Security Report Summary' for the domain <https://nonworking.mooo.com/>. The report is graded with an **A+**. Key details include:

- Site:** <https://nonworking.mooo.com/>
- IP Address:** 54.205.44.0
- Report Time:** 29 Oct 2021 13:18:37 UTC
- Headers:** A list of checked items: X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Strict-Transport-Security, Permissions-Policy, and Content-Security-Policy.

Fonte: Os Autores

Figura 53 – Security Headers do domínio *creativework*.

Fonte: Os Autores

Figura 54 – Security Headers do domínio *overworking*.

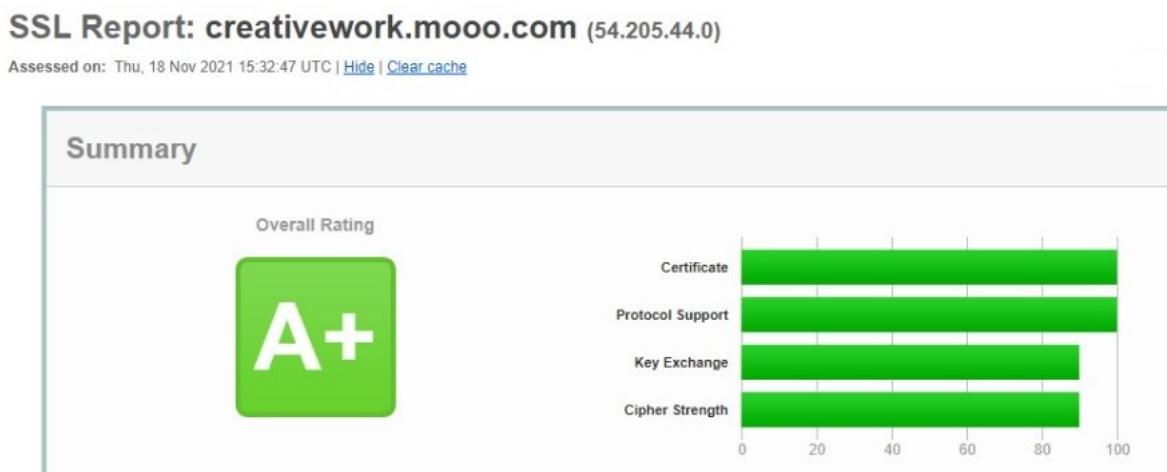
Fonte: Os Autores

De forma adicional ao HTTPS, também foram feitos testes de SSL nos domínios, para garantir a criptografia entre as páginas da aplicação.

Figura 55 – Resultado do teste SSL do domínio *newgenapp*.**SSL Report: newgenapp.link (54.205.44.0)**Assessed on: Thu, 18 Nov 2021 15:28:07 UTC | [Hide](#) | [Clear cache](#)

Fonte: Os Autores

Figura 56 – Resultado do teste SSL do domínio *creativecoworking*.



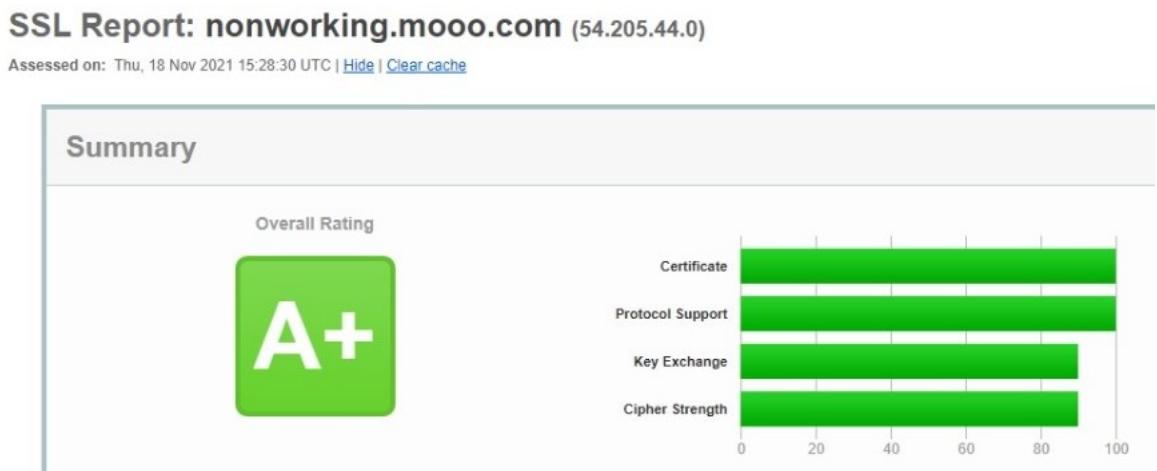
Fonte: Os Autores

Figura 57 – Resultado do teste SSL do domínio *milleniacoworking*.



Fonte: Os Autores

Figura 58 – Resultado do teste SSL do domínio *nonworking*.



Fonte: Os Autores

3.20 Escolhas e Descartes

Nesta seção, são registradas as evoluções e tomadas de decisões que a equipe do NewGen tomou ao longo do percurso de desenvolvimento do Software.

3.20.1 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 como um [SaaS](#).

A 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 [3.18](#)) e algumas discussões de infraestrutura, foi optado por adaptar a aplicação para um modelo [SaaS](#).

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 já poderá começar a rodar a instância digital de seu sistema coworking.

3.20.2 Alterações na lógica do banco

3.20.2.1 Uso do ORM

Durante a implementação inicial do banco de dados, foi definido o uso de 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.

Para clientes que possuam bancos pré-existentes, será dada a opção de importação dos dados para o banco da aplicação, mediante estudo da estrutura do banco e das informações contidas dentro dele para garantir compatibilidade.

3.20.2.2 Adição de novas colunas

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 data desta.

O novo formato das tabelas pode ser consultado no [Apêndice C](#).

3.20.2.3 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 afeta a maneira que o banco de dados estava estruturado até este dia, pois é necessário levar em conta dos dados das tabelas agora não serem de um único Coworking especificamente.

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:

- 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 banco de dados não relacional, como o MongoDB por exemplo.

No momento, o grupo escolheu a primeira opção, porém ainda está aberto o debate para outras opções, conforme futuras implementações.

3.20.3 Projeção para Terceiro Semestre

Durante os preparativos para o início do Segundo Semestre, a equipe fez um novo estudo sobre as funcionalidades a serem implementadas e decidiu por diminuir o escopo a ser cumprido no segundo semestre, movendo as funcionalidades de Controle de Estoque e Sistema de Fidelidade para um teorizado terceiro semestre.

As funcionalidades ainda possuirão Histórias de Usuário e espaço para implementação no projeto final, caso o projeto seja retomado por equipes futuras.

4 Funcionalidades

Neste capítulo, serão tratadas as funcionalidades da aplicação desenvolvida pela equipe NewGen, incluindo o estudo de implementação, informações do back-end e front-end, testes e revisões posteriores, além de exemplos das funcionalidades através de imagens das telas.

4.1 Cadastro Inicial

Para fazer o uso e acesso a plataforma, o cadastro do usuário é uma parte importante. Através do cadastro inicial, é possível dizer se o usuário é um Cliente Administrador, Cliente Funcionário ou Cliente Final, com cada um possuindo níveis de acesso e permissões diferentes.

4.1.1 Estudos

Sendo a primeira funcionalidade do sistema, houve um estudo inicial não apenas de templates individuais, mas da linguagem Python como um todo. Como forma de estudo, a equipe NewGen adquiriu um curso básico, voltado a criação de telas iniciais para um sistema hospitalar em Django. ([COSTA, 2021](#)) Junto ao curso, também foram utilizadas as documentações oficiais do Django, principalmente a documentação sobre a biblioteca de autenticações. ([Django, 2021c](#))

Também foram estudadas quais os tipos de validações a serem utilizados em campos numéricos, como CPF, CEP e telefones. Para validação dos dois primeiros, houve um estudo de uso da API dos Correios, porém ela foi descartada por fazer cobrança de taxas para cada consulta. Na validação de telefone, a equipe escolheu por utilizar um port da biblioteca *libphonenumbers*, originalmente do Google, para o Python, a *python-phonenumbers*. ([DRYSDALE, 2021](#))

4.1.2 Implementação

Usando os conhecimentos do curso e documentações, a principal biblioteca de suporte para o cadastro de usuário foi a biblioteca de autenticação *auth*, usada para garantir a validade do cadastro como um todo.

Para criar a validação de campos, a equipe de back-end usou o algoritmo de verificação de CPF, adaptando ele dentro do sistema, de forma a não depender de APIs terceiras. ([MACORATTI, 2021](#)) Para a validação do CEP, foi utilizado uma API que

disponibiliza uma lista de CEPs não-criptografada, para garantir a autenticidade do CEP inserido. (OLIVAES, 2005) Ambas as implementações foram escolhidas para evitar cobranças de taxas por uso.

Conforme o usuário insere os valores no formulário, o Django Form recebe eles e separa os valores de CPF/CNPJ e e-mail para serem enviados a tabela *auth_user*, onde será determinado se o usuário novo é funcionário (através do campo "*is_staff*") ou administrador (através do campo "*is_superuser*"). Também há a interação com a tabela *auth_group* que define outros grupos padrões, como administrador, funcionario, clienteEmpresa e clientePessoa.

Figura 59 – Exemplo de dados inseridos dentro da tabela *auth_user*



The screenshot shows a MySQL Workbench interface with a query editor at the top containing the SQL command: "42 • select * from auth_user;". Below the query is a "Result Grid" table with the following data:

	id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined
1	pbkdf2_s...	2021-10-0...	0	42768138835	Gabriel	Pinheiro Brants Gonçalves	x.gabriel12@gmail.com	0	1	2021-10-09...	
2	pbkdf2_s...	NULL	0	80867023406	Gabriel	Agora	x.gabriel12@gmail.com	0	1	2021-10-12...	
3	pbkdf2_s...	2021-10-1...	0	30426238000144	Jojo Company	NULL	NULL	x.gabrie12@gmail.com	0	1	2021-10-12...

Fonte: Os Autores

Conforme visto na Figura 59, a encriptação dos dados é feita via PBKDF2, considerado seguro contra ataques de dicionário ou força bruta. Além disto, o csrf_token, utilizado no Django, garante a segurança do transporte de dados via POST, através de uma chave de encriptação que evita ataques do tipo Man In The Middle.

Para as máscaras de campos de CPF, CNPJ, telefone e CEP, foram utilizadas as bibliotecas do JQuery com módulo de masks. Os outros campos sofrem tratamentos diretamente do Django, como no campo de logradouro, onde o limite de 5 caracteres impede que o usuário digite valores maiores que o estabelecido.

Figura 60 – Formulário de pessoa física dentro do template .html

```
<div id="BodyRegister">
    <form method="POST" class="form-group" name="formPessoaFisica">
        {% csrf_token %}
        {{ pessoaForm|crispy }}

        <input type="hidden" class="form-selecionado" name="form-selecionado" value="{{ selecionado }}>

        <button type="submit" id="enviar_form_pessoa" class="btn btn-primary">{{ translate "Registrar" }}</button>
    </form>
```

Fonte: Os Autores

A Figura 60 mostra a implementação do formulário de pessoa física em HTML, utilizando o pacote crispy-forms. O *pessoaForm* indica o objeto formulário que é enviado

do Python ao comando *crispy*, que resulta no formulário formatado. Também constam o csrf_token e a tag de internacionalização do texto.

4.1.3 Revisões

Após a implementação inicial, houve revisões para adicionar a validação dos campos de CPF, CNPJ e CEP, para adicionar uma diferenciação dos formulários de cliente pessoa e cliente empresa (antes sendo páginas separadas) como visível na [Figura 61](#).

Houve também mudanças no redirecionamento das telas para o início do sistema, além da criação do menu referente a login e cadastro no navbar das páginas.

4.1.4 Telas

Algumas das telas disponíveis no acesso a aplicação, visíveis aos usuários do sistema.

Figura 61 – Seleção de opção de pessoa física ou pessoa jurídica.



Fonte: Os Autores

Figura 62 – Erros, validação de máscara e campos nos formulários.

Nome Completo*
Gabriel 981981981 Pinheiro Brants Gonçalves ⓘ
Certifique-se que o nome só contém letras. Números e símbolos não são permitidos.

Senha*
[campo vazio]

CPF*
968.119.819-19 ⓘ
CPF inválido

Data de Nascimento*
18/02/1800 ⓘ
Data de nascimento inválida! Escolha entre 1900 e 2010.

Gênero*
Masculino

Email*
x.gabriel12@gmail.com

Telefone*
(11) 00000-0000 ⓘ
Telefone inválido!

Fonte: Os Autores

Figura 63 – Login no sistema, com máscara de cliente.

LOGIN

CPF/CNPJ: 427.681.388-35

Senha:

Entrar

Primeiro Acesso

Esqueceu a senha?

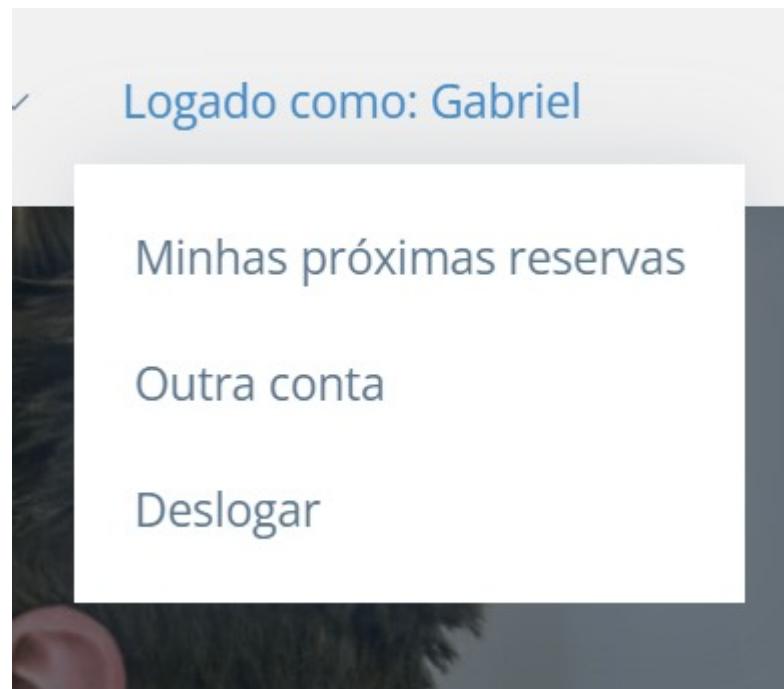
Fonte: Os Autores

Figura 64 – Login no sistema, com máscara de empresa.



Fonte: Os Autores

Figura 65 – Navbar de login.



Fonte: Os Autores

4.2 Cadastro e Gerenciamento de Instâncias de Coworking

Após o cadastro inicial do Administrador, será dada a opção de cadastro dos espaços, que consiste em fornecer as informações do Coworking para criação da página no sistema.

4.2.1 Estudos

Para o cadastro das informações do coworking, houve uma nova carga de estudos a ser feita. Na questão do banco de dados, foram estudadas três frentes: o uso de um banco compartilhado com todos os clientes do sistema; o uso de um banco de dados não-relacional, como o MongoDB (vide [Apêndice F](#)); e a geração de um banco de dados individual para cada cliente.

Após conversação com os professores, foi decidido seguir a via da geração de bancos individuais dentro do MySQL, com a alternativa de migrar o sistema para o PostgreSQL, onde há a função de multi-schema. Para a conexão com o Python, foi estudado o conector específico para bancos de dados múltiplos, de modo a garantir que eles funcionariam corretamente na aplicação. ([Django, 2021a](#))

Para garantir o gerenciamento inteligente dos sistemas, foi estudado o conceito de *multi-tenancy*, uma arquitetura inteligente que permite alternar entre domínios específicos conforme a escolha de conteúdo, algo necessário para os múltiplos sites a serem disponibilizados. No Django, esse conceito foi aplicado através do *django-hosts*. ([Jazzband, 2016](#))

4.2.2 Implementação

Utilizando o pacote *django-hosts*, é possível definir as URLs válidas para acesso de cada domínio em apenas um arquivo, o *hosts.py*; para a URL proprietária da NewGen, por exemplo, são disponibilizadas URLs como "registrar_coworking", "assinatura_de_coworking" e "sobre_o_newgen", enquanto um domínio de uma empresa terceira teria URLs diferentes, como "pagamento_reserva", "agendamento" ou "listar_espacos".

No arquivo *router.py* são informadas as rotas do banco de dados para cada domínio disponibilizado. Conforme cada domínio é cadastrado, é criado um banco com nome único para acesso; uma concatenação do nome do cliente com o nome da empresa em formato ASCII, de forma a evitar conflitos. Também há uma função para mitigar a chance de nomes de banco de dados idênticos, renomeando o banco mais recente em casos positivos.

Dentro da arquitetura do Django, o arquivo *settings.py* mantém todas as configurações globais do projeto, incluindo as configurações de acesso de cada banco de dados.

Sendo um arquivo estático, cada instância do banco de dados forçaria o sistema a ser reiniciado para atualizar.

Para solucionar este problema, foi criado o *db.json*, para armazenamento das conexões relevantes necessárias durante a criação de um banco de dados novo; assim, cada implementação precisa apenas de um comando que recarrega os objetos de banco de dados já presentes na memória do sistema.

O *db.json* também solucionou o problema de armazenamento das informações sobre os bancos utilizados; sem esse arquivo, seria necessário inserir todas as informações de cada banco em um banco centralizado, o que seria contraprodutivo, pois seria necessário a inicialização de dois bancos diferentes a cada uso do sistema, e traria uma vulnerabilidade desnecessária ao sistema, pois deixaria os bancos todos disponíveis em um local só, enquanto no *db.json* essas informações são visíveis apenas ao Django.

Para oferecer mais apoio, foi criada uma nova aplicação no back-end, chamada de *cria_coworking* usando o domínio da *newgenapp.link*, com a aba de cadastro de coworking que será refatorada futuramente para adição dos planos de assinatura. [Figura 66](#)

4.2.3 Revisões

Até a última atualização do documento, não foram feitas revisões na implementação ou lógica desta funcionalidade.

4.2.4 Telas

Figura 66 – Tela inicial da NewGen.



Fonte: Os Autores

Figura 67 – Tela de cadastro de coworking.

Fonte: Os Autores

Figura 68 – Domínios de coworkings, disponíveis no banco de dados.

	id	domain	name	isActive
▶	1	newgenapp.link	newgenapp	1
	2	millenniacoworking.mooo.com	millenia coworking	1
	3	creativework.mooo.com	creative work	0
	4	nonworking.mooo.com	nonworking	0
	5	overworking.mooo.com	overworking	0
	6	testtest.com	test	1
*	NULL	NULL	NULL	NULL

Fonte: Os Autores

4.3 Recuperação de Senha

A recuperação de senha pode ser utilizada por qualquer usuário do sistema, para redefinir a senha de acesso utilizada. Deve ser informado o CPF do usuário para recebimento do e-mail com o link para recuperação. Neste link, será requisitada a nova senha de acesso, e após a confirmação a senha anterior do usuário será substituída pela nova.

4.3.1 Estudos

Utilizando o aprendizado de funcionalidades anteriores, o estudo para recuperação de senha foi mais rápido, precisando apenas identificar a necessidade do uso de outras bibliotecas para implementação.

Foi usado um artigo que explica a implementação da recuperação de senha através do Django como base de estudo, buscando compreender quais partes já haviam sido adaptadas anteriormente e, portanto, poderiam ser reaproveitadas, e quais precisavam de implementações a serem feitas. ([OrdinaryCoders, 2020](#))

4.3.2 Implementação

Foram reutilizados os templates da biblioteca de autenticação do Django, utilizada previamente nas funcionalidades de [Cadastro Inicial](#) e [Cadastro Inicial](#).

Figura 69 – Configuração do e-mail para envios.

```
# Configurações de envio do e-mail

EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = "newgenoficial2021@gmail.com"
```

Fonte: Os Autores

Através destas views, é possível criar um link único para recuperação de senha, que é encaminhado para o e-mail do requisitor.

4.3.3 Revisões

Durante a finalização da implementação, houve problemas na funcionalidade de PasswordResetReview, que gerava um link incorreto para o usuário. A solução escolhida pela equipe foi a edição da string antes do envio, corrigindo para uma URL correta.

4.3.4 Telas

Figura 70 – Opção de recuperação de senha, na área de login.



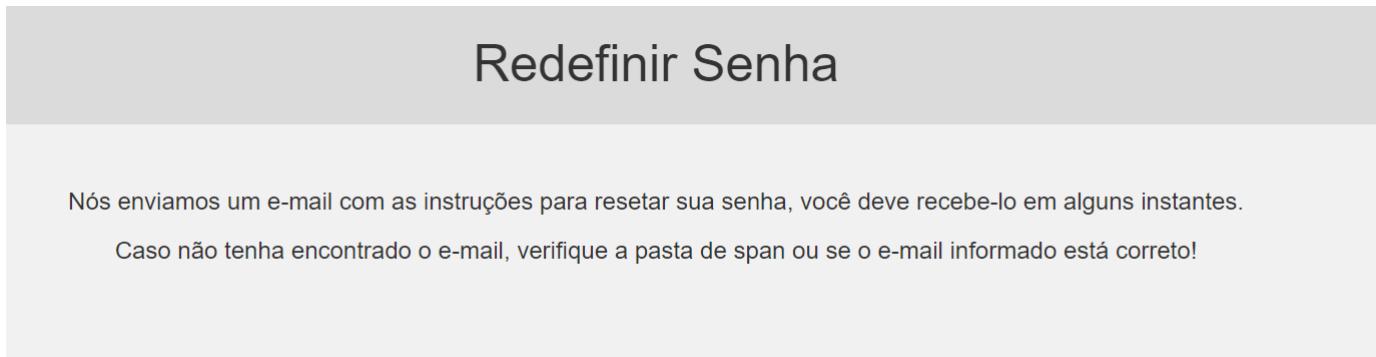
Fonte: Os Autores

Figura 71 – Requisição do e-mail para envio do link.



Fonte: Os Autores

Figura 72 – Mensagem informada ao cliente, caso o e-mail esteja presente na base.



Fonte: Os Autores

Figura 73 – E-mail com link para redefinição de senha.



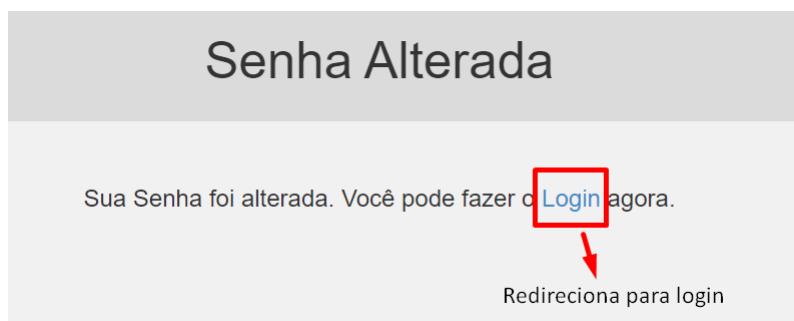
Fonte: Os Autores

Figura 74 – Página de redefinição, com as regras para novas senhas.

The screenshot shows a web page with a light gray header containing the text "Altere sua senha". Below the header is a form field labeled "Nova senha:" followed by a redacted input box. Underneath this is a list of four rules: "Sua senha não pode ser muito parecida com o resto das suas informações pessoais.", "Sua senha precisa conter pelo menos 8 caracteres.", "Sua senha não pode ser uma senha comumente utilizada.", and "Sua senha não pode ser inteiramente numérica.". Below the rules is another form field labeled "Confirmação da nova senha:" followed by a redacted input box. At the bottom of the form is a blue button labeled "Alterar Senha".

Fonte: Os Autores

Figura 75 – Conclusão da alteração de senha.



Fonte: Os Autores

4.4 Reserva

A reserva é a principal funcionalidade da aplicação, sendo voltada a permitir que o Cliente Final faça a reserva no seu coworking de escolha, dando prioridade a horário ou data, e escolha o espaço mais adequado para sua necessidade.

4.4.1 Estudos

De início, foi estudado o uso do pacote *django-scheduler*, porém a falta de documentação e exemplos práticos de uso do pacote resultaram no descarte da opção. Após maiores pesquisas, foi escolhido o pacote *full-calendar*, que correspondia melhor as necessidades de exibir uma interface para escolha de reservas por datas e horários ao usuário.

Sendo uma interface gratuita e bem documentada, foi necessário apenas o estudo desta para avaliar a melhor forma de implementação no sistema da NewGen, sendo este colocado na parte de front-end. ([FullCalendar, 2021](#))

4.4.2 Implementação

Inicialmente, foi feita a decisão de passar todas as reservas disponíveis por uma lógica que transformasse eles em objetos JSON, a serem reconhecidos pelo *full-calendar*, baseando sua estrutura na escolha do tipo de espaço feito pelo usuário. Durante o MVP, esse sistema acontecia antes da reserva do espaço, o que causava atrasos no processamento do modelo do negócio antes que a requisição chegasse ao usuário.

Para mitigar o tempo de processamento, foi estudado a possibilidade da criação de uma API que fornecesse tais dados de forma dinâmica, se baseando apenas nas telas que o usuário fizesse acesso. Esta foi criada a partir do framework *django rest api*, que facilitou a criação dos endpoints para se conectar com as outras partes do sistema. No momento, a [API](#) de reserva foi criada aceitando apenas métodos GET, e está passando por melhorias de segurança.

No front-end, o *full-calendar* está sendo utilizado para a lógica de escolha de espaços. O calendário é uma função [Javascript](#) processada sobre um [div HTML](#), com os elementos criados sob ele. A funcionalidade traz diversas configurações disponíveis para uso, sendo uma delas o campo de *eventos*, que são os objetos a serem mostrados no calendário como um agendamento específico, como visível na [Figura 79](#).

Para a lógica de seleção de um horário, foi escolhido a interface *date-picker*, outra dependência [Javascript](#) que possuí configurações básicas para funcionamento. Para a implementação na aplicação da NewGen, foi dispensado o uso da escolha de data, usando apenas a de horário, e a seleção de horas foi inserida dentro de um [modal](#), que é chamado quando o usuário clica em um horário disponível.

Por último, foi utilizado o pacote *owl-carousel*, que cria um **carousel** para o display de espaços para locação, tornando a visão dos espaços mais dinâmica para o cliente.

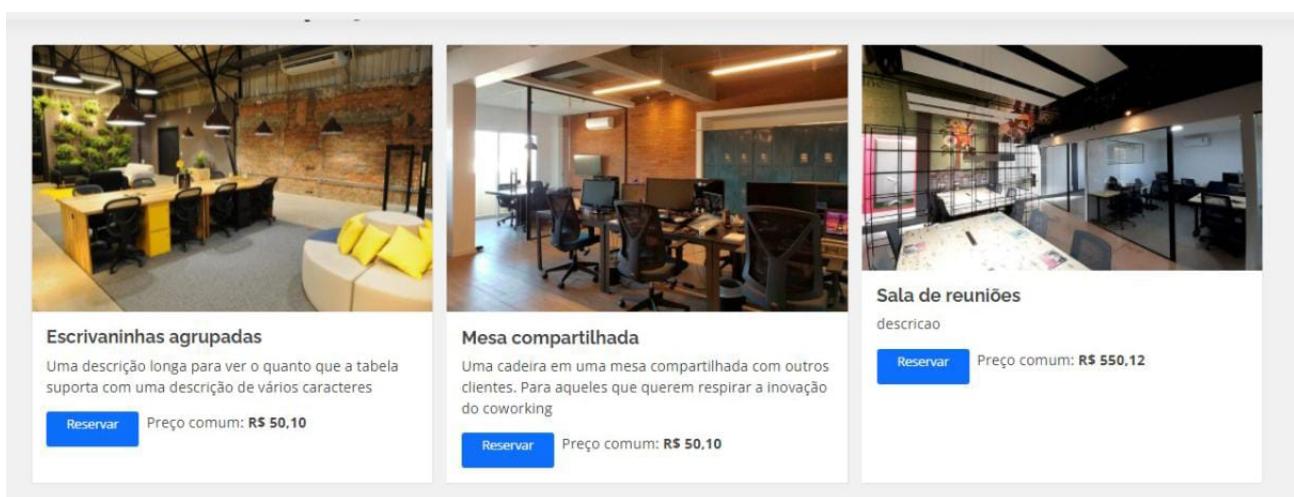
4.4.3 Revisões

Da implementação inicial até o presente momento, a funcionalidade de reserva é uma das que mais passou por modificações. A escolha de espaços, antes um **HTML** fixo, agora recebe os espaços atribuidos pelo proprietário em [Cadastro e Gerenciamento de Instâncias de Coworking](#).

Também foi adicionada a parte de pagamentos ao final da reserva, desenvolvida em [Pagamento](#). As melhorias de visualização, como a adição do carousel, e processamento, o [API](#) de leitura dinâmica, também foram adicionados em revisões posteriores da funcionalidade.

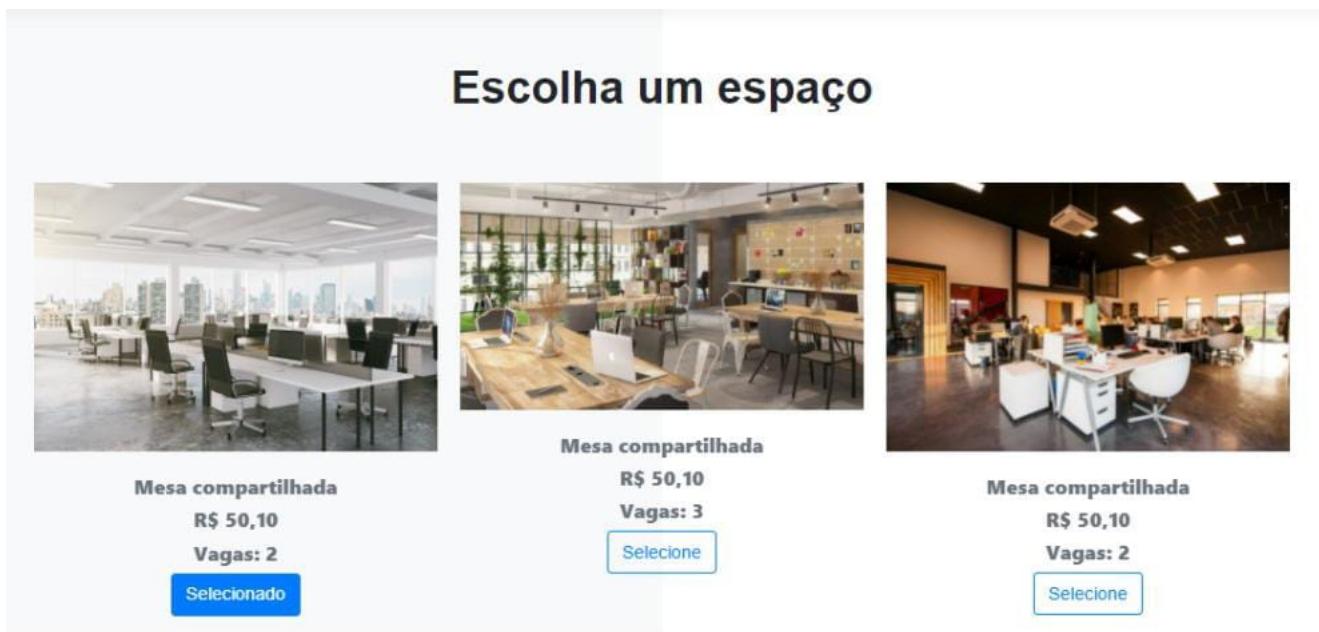
4.4.4 Telas

Figura 76 – Tela de seleção do tipo de espaço.



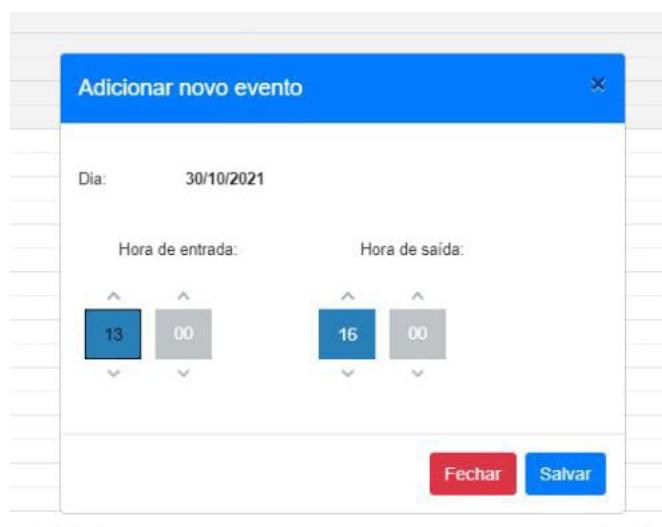
Fonte: Os Autores

Figura 77 – Tela de seleção de espaço.



Fonte: Os Autores

Figura 78 – Seleção de data e hora de entrada e saída.



Fonte: Os Autores

Figura 79 – Calendário para seleção da data.



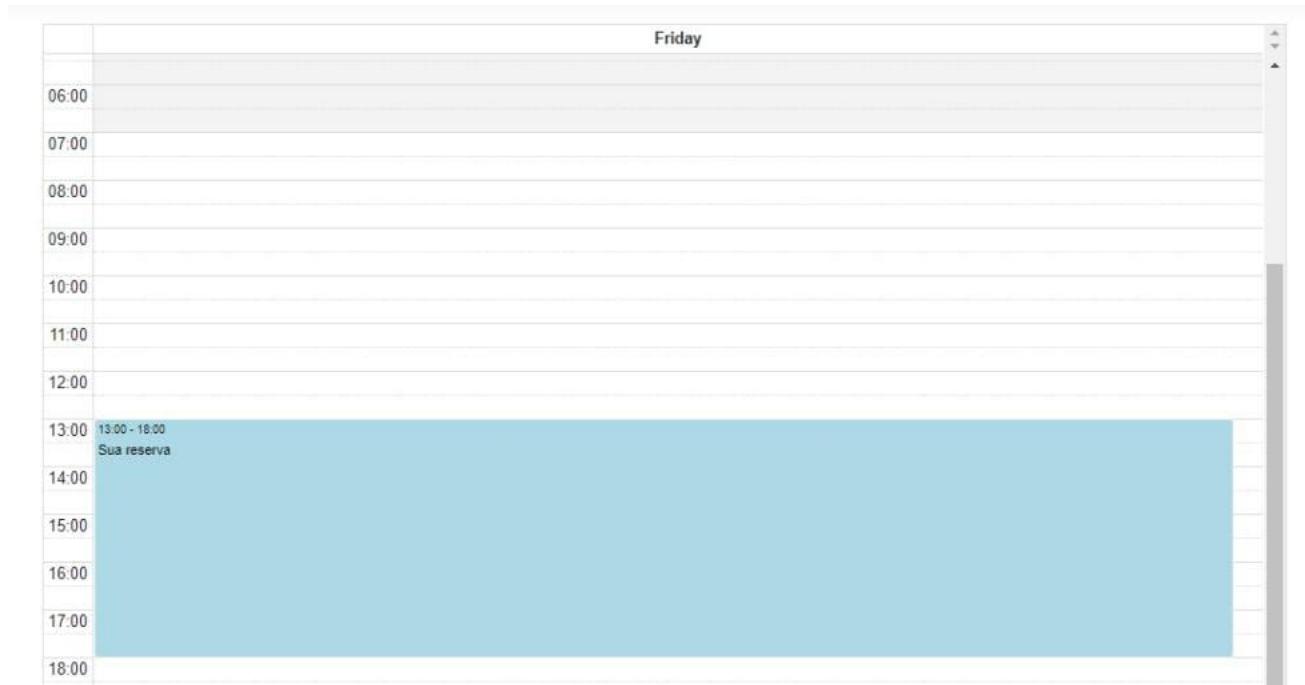
Fonte: Os Autores

Figura 80 – Exemplo de eventos no calendário.



Fonte: Os Autores

Figura 81 – Exemplo de reservas



Fonte: Os Autores

4.5 Pagamento

Partindo da funcionalidade de [Reserva](#), a funcionalidade de pagamento é, através da plataforma de pagamentos Mercado Pago, responsável por definir se a locação feita é uma Reserva (pago um sinal de 10% para reservar o horário) ou um Aluguel (pago o valor completo e feita a locação exclusiva).

4.5.1 Estudos

Para um pagamento eficiente e com segurança ao cliente, foi selecionada a [API](#) de pagamento do Mercado Pago, tanto pela questão de segurança com as informações de cartão de crédito e dados pessoais, quanto pela flexibilidade permitida pela [API](#), que se adaptava ao nosso sistema. ([Mercado Pago Developers, 2021](#))

Dentre os métodos de implementação disponibilizados na aba de desenvolvedores do Mercado Pago, foi escolhido o Checkout Transparente, que oferece todos os requisitos necessários para cumprir com um processo de pagamento seguro e sem excessivas customizações; além de ser funcional ao cliente, também trazia implementação rápida para a equipe. ([Mercado Pago, 2021](#))

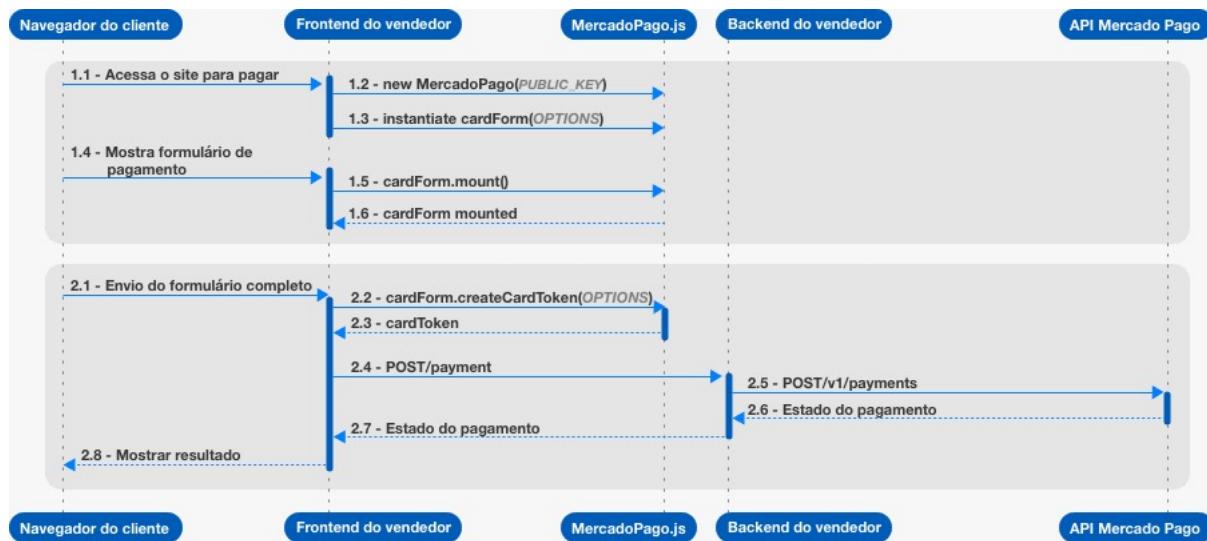
4.5.2 Implementação

Para o backend do pagamento foi feito a criação de um endpoint ([API](#)) dentro do Django pelo *django-rest*. O cliente entra na tela de pagamento, termina o formulário e no momento de enviar, acontece a consulta com a API do MercadoPago por meio do JavaScript do próprio navegador do cliente. Isso ocorre porque o JavaScript do navegador do cliente contém dados sensíveis a respeito da sua conta bancária, portanto o MercadoPago, com intenção de garantir a segurança no processo envolvido com seu nome, obriga que a consulta de API seja do FrontEnd diretamente com a consulta dos dados deles.

Na hora de enviar a API para o Django, o mercado pago limpa os campos sensíveis (número do cartão, bandeira, código de validação, etc) e envia as informações públicas referentes ao pagamento (se ele foi recusado, quantidade cobrada, quem é o pagador, etc) junto com um token.

É nessa hora que o endpoint entra em ação, interceptando esses dados pós-pagamento para fazer uma segunda consulta na API junto com o token fornecido e, a partir das informações recebidas via front do JavaScript, realizando as lógicas de negócio, com o registro das informações nos campos de pagamento e validando a reserva como paga.

Figura 82 – Lógica de Pagamento



Fonte: Os Autores

4.5.3 Revisões

Até a última atualização do documento, não foram feitas revisões na implementação ou lógica desta funcionalidade.

4.5.4 Telas

Figura 83 – Formulário de pagamento

Formulário de pagamento

Número do cartão*

Mês vencimento*

Ano vencimento*

Nome descrito no cartão*

E-mail*

Código de segurança*

Fonte: Os Autores

4.6 Autorização de Entrada

A validação de entrada se trata do envio de um código QR para entrada no coworking. Este código QR é gerado a partir do momento que o usuário locador realiza o pagamento completo da reserva, podendo gerar o código a qualquer momento, que será enviado para o e-mail registrado no cadastro do cliente.

4.6.1 Estudos

Os estudos iniciais foram ao redor da geração do código QR e do envio deste ao cliente. Para o uso da biblioteca *qrcode*, do Django, foi estudado apenas a documentação oficial desta. ([Python Wiki Brasil, 2020](#))

Para o envio de e-mails, foi usada a biblioteca de envio de e-mail do Django em conjunto com frameworks voltados para esta funcionalidade. ([Django, 2021b](#)) Também foram utilizados tutoriais de implementação para montar um *roadmap* para a implementação da funcionalidade. ([VICO, 2021](#))

4.6.2 Implementação

Para a implementação dessa funcionalidade, foi necessário a junção da geração do código QR com o envio de e-mail.

A geração do código QR foi baseada em uma lógica onde a opção de geração é disponibilizada apenas se o valor da reserva for pago por completo; através da verificação no banco de dados, a opção de geração do código é disponibilizada na reserva para o cliente. No código, é disponibilizado um hash e o número de identificação da reserva, sendo a hash da imagem salva no banco de dados, sobre o número de identificação da mesma reserva.

Para o envio do e-mail com a imagem, foram feitas configurações para identificar quem faz o envio do e-mail, quem receberá, o corpo do e-mail, o assunto dele e o anexo, como visível na [Figura 84](#).

4.6.3 Revisões

Até a última atualização do documento, não foram feitas revisões na implementação ou lógica desta funcionalidade.

4.6.4 Telas

Figura 84 – Código para envio de e-mail

```

corpo = "Use este QRcode para confirmar sua reserva no momento da entrada no estabelecimento."

email_msg = MIME Multipart()
email_msg ['From'] = login
email_msg ['To'] = email
email_msg ['Subject'] = "Qrcode para validação de entrada"
email_msg.attach(MIMEText(corpo, "html"))

caminhoArquivo = "static/media/qrcode.png"
attachment = open(caminhoArquivo, 'rb')

att = MIMEBase('application', 'octet-stream')
att.set_payload(attachment.read())
encoders.encode_base64(att)

att.add_header('Content-Disposition', f'attachment; filename=qrcode.png')
attachment.close()

```

Fonte: Os Autores

Figura 85 – Botão para envio do código QR, na aba de reservas

Suas próximas reservas				
DATA	HORÁRIO	ESPAÇO	RESPONSÁVEL	QR CODE
29 de Outubro de 2021	13:00	Mesa compartilhada	Gabriel Pinheiro Brants Gonçalves	<button>Gerar</button>
31 de Outubro de 2021	13:40	Mesa compartilhada	Gabriel Pinheiro Brants Gonçalves	
17 de Novembro de 2021	11:00	Mesa compartilhada	Gabriel Pinheiro Brants Gonçalves	

Fonte: Os Autores

Figura 86 – URL identificando para qual reserva foi feita o envio



Fonte: Os Autores

Figura 87 – Código hash, salvo no banco de dados

	<u>id_reserva</u>	<u>is_aluguel</u>	<u>hash_qrcode</u>
▶	7	1	4db0b381629a212a23050c5055e14ce6
✖	NULL	NULL	NULL

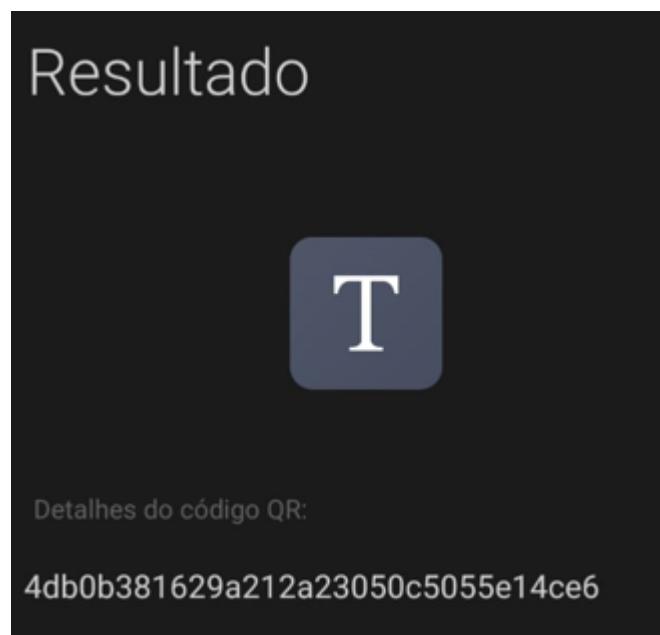
Fonte: Os Autores

Figura 88 – E-mail com código anexo em PDF



Fonte: Os Autores

Figura 89 – Valor inserido no código QR



Fonte: Os Autores

4.7 Personalização de Tela

A personalização, parte essencial do White Label, serve para garantir que o administrador possa deixar seu site mais adequado a sua marca, alterando cores, imagens e textos a vontade, o que atrai uma maior clientela ao produto da NewGen.

4.7.1 Estudos

Antes de implementar a funcionalidade de customização - que se refere à capacidade do cliente proprietário escolher cores, logo e textos para áreas específicas do site dele - foram estudados 3 tópicos de pesquisa: uma função interna do django, a biblioteca *os* do python e uma tag interna do django.

A primeira, a função interna do django, é a "render_to_string" do pacote *django.template.loader.render_to_string*. Ela serve para renderizar uma página html dinâmica aplicando as modificações dos templates. Os templates no Django são páginas html com tags que o Django reconhece como funções, e dependendo das variáveis que transitam pelas requests e responses, esses valores são alterados na tela. A função render_to_string pega uma página dinâmica, aplica as modificações e os comandos django dentro da página e devolve uma string com o resultado estático destas modificações.

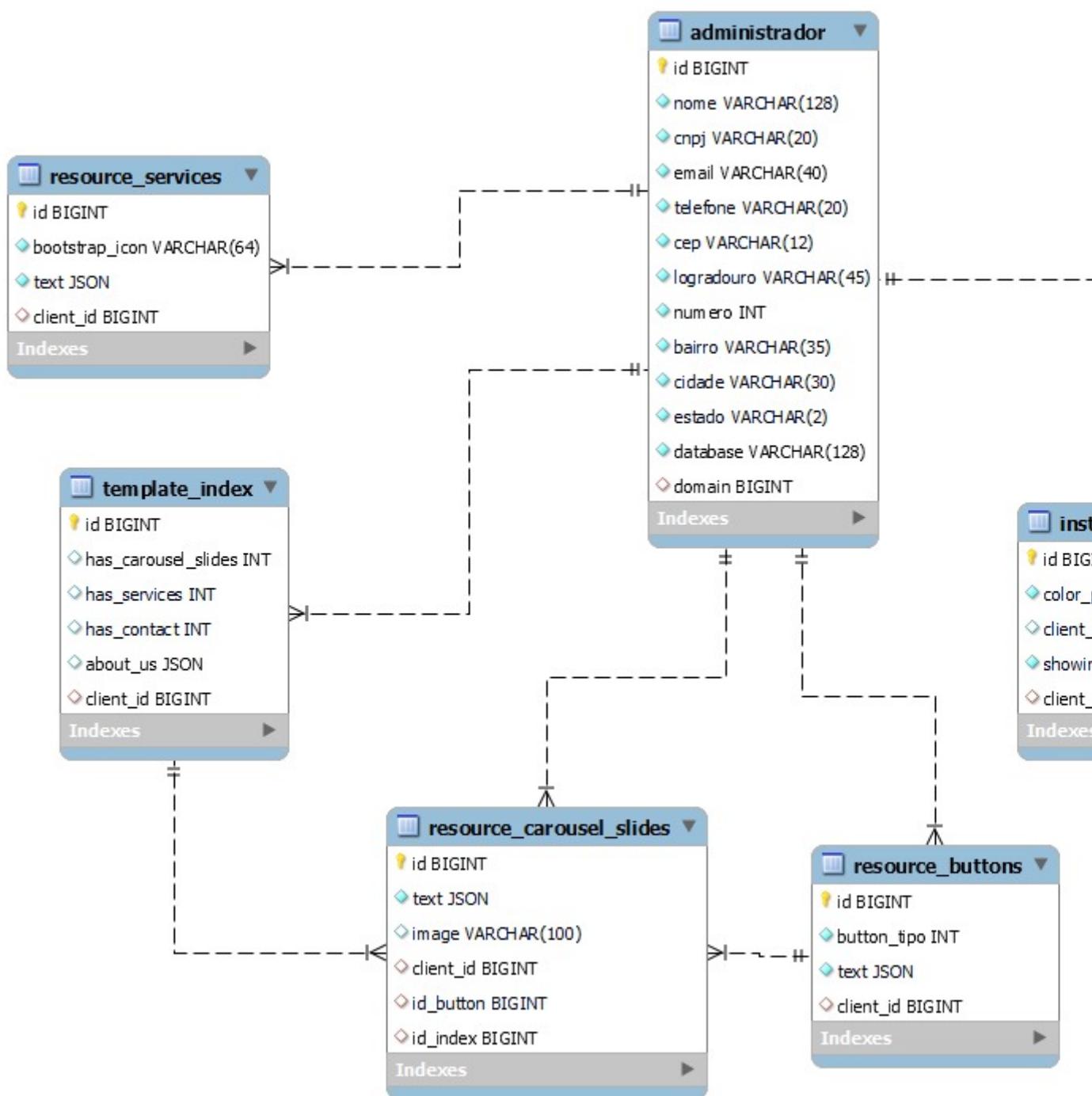
A segunda é a biblioteca *os* do python. Ela é uma das principais bibliotecas do Python e atua junto com o sistema operacional. Para a customização ela foi importante para criar diretórios personalizados que conserva os templates de cada cliente.

A terceira, mas não menos importante, é a tag customizada *templatetag*. Ela serve para escapar da renderização HTML do Django, retornando uma outra tag no final. A utilização desta tag foi essencial para que no final da renderização com a função render_to_string fossem criados páginas dinâmicas reconhecidas pelo Django (templates), não apenas HTMLs estáticos. O funcionamento da templatetag é parecido com o sinal barra invertida ' ' visto em outras linguagens de programação, que dizem ao compilador para ignorar o próximo caractere. ([Python, 2021](#))

4.7.2 Implementação

Para o back-end foram criadas tabelas novas no banco de dados, específicas para guardar as informações customizadas de cada cliente.

Figura 90 – DER das informações do cliente



Fonte: Os Autores

Cada tabela guarda um tipo de informação diferente, mas todas são consultadas na hora de montar um template para o cliente. Partiu-se com a ideia de usar o banco de dados para guardar informações simples como imagem e texto e montar dinamicamente conforme houver alterações as páginas dos clientes em arquivos.html dentro do diretório de templates do Django. Criou-se também uma função que copula as tabelas do cliente com algumas informações básicas para serem alteradas posteriormente nas páginas de

customização.

Para etapa do ambiente acadêmico, as funções de customização alteram a estrutura básica do site e as informações da página inicial, porém, há espaço para se criar uma ferramenta de customização que dê liberdade para o cliente proprietário deixar o site dele o mais perto do seu agrado.

Uma outra questão que precisou ser resolvida também foi a passagem destas informações entre as telas. Visto que gostaríamos de dar a oportunidade do cliente em mudar a própria internacionalização, terceirizando e aliviando também a necessidade de ter um tradutor para rever as traduções de cada cliente de acordo as customizações, foi decidido utilizar o formato JSON para passar as informações em várias línguas.

Para buscar as informações dentro do Django, utilizou-se a funcionalidade de tags personalizadas do Django, passando as informações de qual é a língua selecionada do usuário e qual é o texto a ser impresso por dentro do próprio template, como visível em ??

Figura 91 – Tag de templates para tradução.

```
<!-- ===== Contact Us Section ===== -->
<section id="contact" class="contact">
  <div class="container" data-aos="fade-up">

    <div class="section-title">
      <h2>{% templatetag openblock %} translate "Entre e contato" {% templatetag closeblock %}
    </div>

    <div class="row">

      <div class="col-lg-6 d-flex" data-aos="fade-up">
        <div class="info-box">
          <i class="bx bx-map"></i>
          <h3>{% templatetag openblock %} translate "Endereço" {% templatetag closeblock %}
          <p>{{ endereco }}</p>
        </div>
      </div>
    </div>
```

Fonte: Os Autores

Figura 92 – Nova tag de templates para tradução.

```
<!-- ===== Contact Us Section ===== -->
<section id="contact" class="contact">
  <div class="container" data-aos="fade-up">

    <div class="section-title">
      <h2>{% translate "Entre e contato" %}</h2>
    </div>

    <div class="row">

      <div class="col-lg-6 d-flex" data-aos="fade-up">
        <div class="info-box">
          <i class="bx bx-map"></i>
          <h3>{% translate "Endereço" %}</h3>
          <p>Rua Santo Antônio, 561, São Pedro, Vitória - ES, 29030-035</p>
        </div>
      </div>
    </div>
  </div>
```

Fonte: Os Autores

Uma questão que precisou ser levada em conta também foi a encriptação dos caracteres sensíveis do JSON (como as aspas por exemplo) em query string http. Nessa notação, um caractere aspa é codificado em '"'.

Para o front foi utilizado o bootstrap e a maneira que o bootstrap se comunica para trazer elementos novos. Por exemplo, para customizar o botão principal do coworking, foi aproveitado as classes de botões (e.g.: primary, secondary, danger, warning) e essas informações salvas como strings para cada proprietário. Na hora de montar o botão, basta apenas o django renderizar a classe do elemento com o tipo do botão por dentro do próprio HTML.

Além disso, na parte dos formulários para editar as informações presentes, utilizou-se muito a mesclagem de Bootstrap com o objeto "form" do Django, aplicando também as funcionalidades do HTML5 como o "color-picker".

4.7.3 Revisões

Até a última atualização do documento, não foram feitas revisões na implementação ou lógica desta funcionalidade.

4.7.4 Telas

Figura 93 – Cadastro das informações do cliente administrador

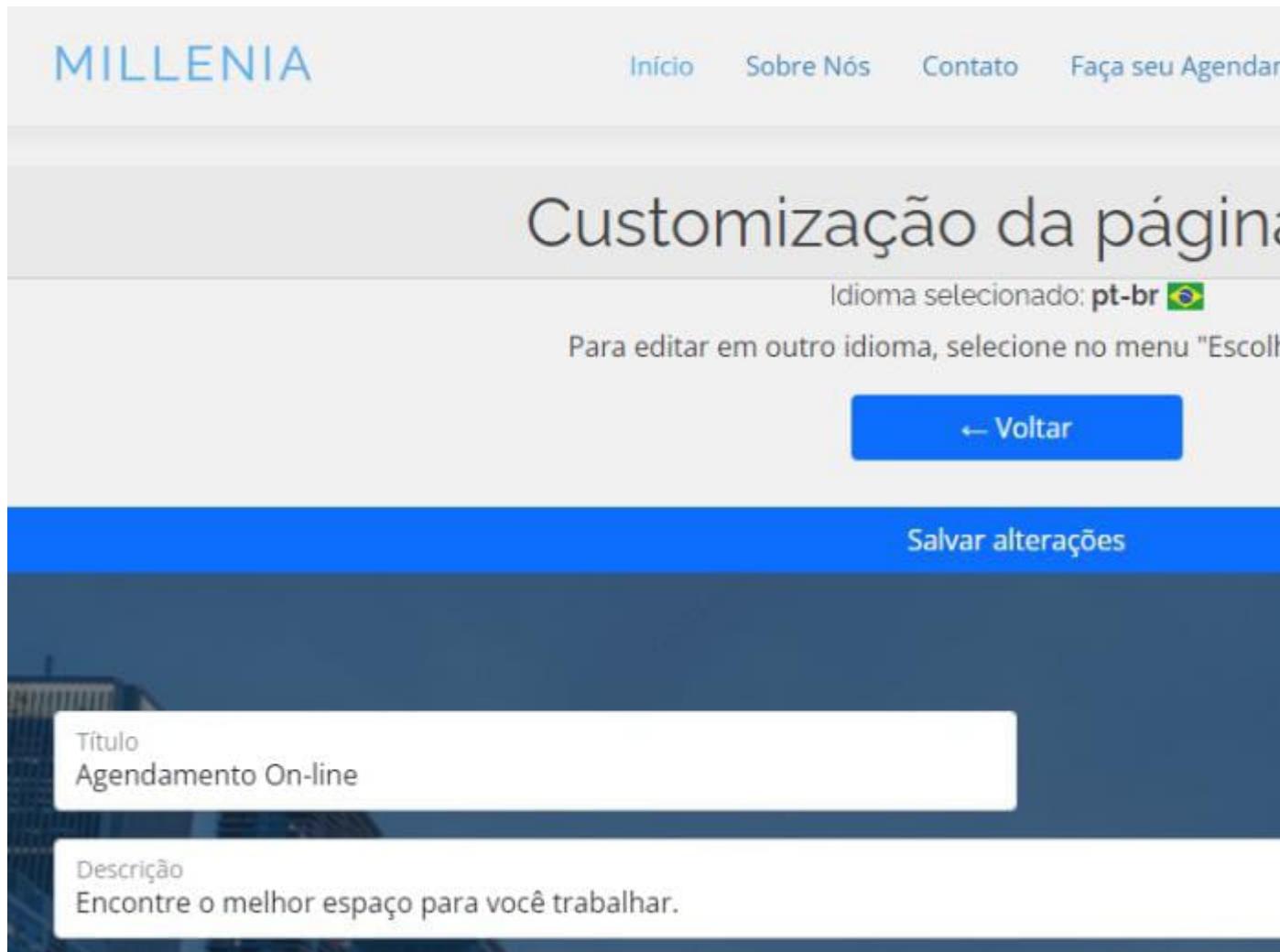
The screenshot shows a web-based form for registering client administrator information. The fields and their values are:

- Telefone***: (11) 97786-6159
- Cep***: 29030-035
- Logradouro***: Rua Santo Antônio
- Número***: 561
- Bairro***: São Pedro
- Estado***: Espírito Santo
- Cidade***: Vitória

At the bottom of the form are two buttons: a blue "— Voltar" button on the left and a blue "Salvar alterações" button on the right.

Fonte: Os Autores

Figura 94 – Edição da tela principal



Fonte: Os Autores

Figura 95 – Edição da imagem e Sobre Nós



Fonte: Os Autores

Figura 96 – Edição da paleta de cores do site



Fonte: Os Autores

4.8 Relatórios

Os relatórios são apresentados apenas na tela do administrador, servindo para demonstrar de forma ágil o andamento do coworking; de reservas de salas, com filtro para datas, ao valor arrecadado, e até os maiores clientes do local.

4.8.1 Estudos

Antes de iniciar os estudos, foram pesquisadas as alternativas para implementação; devido a experiências profissionais de alguns membros da equipe, o PowerBI da Microsoft foi escolhido, tanto pela facilidade de implementação quanto pela conexão fácil com o Django. ([Microsoft, 2021](#))

Após a decisão, foram estudadas as documentações proprietárias, incluindo a de conexão ao banco de dados através do Pandas [Pandas Development Team \(2021\)](#), uma ferramenta de extração e manipulação de dados, e revisão da extração de informações do MySQL pelo Django. ([Oracle Corporation, 2021](#))

4.8.2 Implementação

Foram seguidas as melhores práticas para a construção do dashboard como; a criação de uma tabela específica para as datas, centralizando assim as movimentações dos registros; os relacionamentos foram baseados no modelo estrela, sendo este o mais responsável e eficiente; E a utilização de parâmetros em todas as consultas às bases de dados, permitindo assim a alteração rápida e ágil da fonte de informações.

Para busca dos dados foi utilizado um script Python onde, inicialmente, foi instanciado um objeto MySQL de conexão ao banco. Em seguida, cada linha retornada (no formato de tupla), foi adicionada em uma lista que, por fim, foi convertida em um objeto Panda.DataFrame aceito pelo PowerBI. Em cada consulta a base, foi selecionado uma view que representa ou uma dimensão (tabelas de descrições de clientes, empresas, entre outros) ou uma fato (entidades com movimentações como, no presente caso, as reservas).

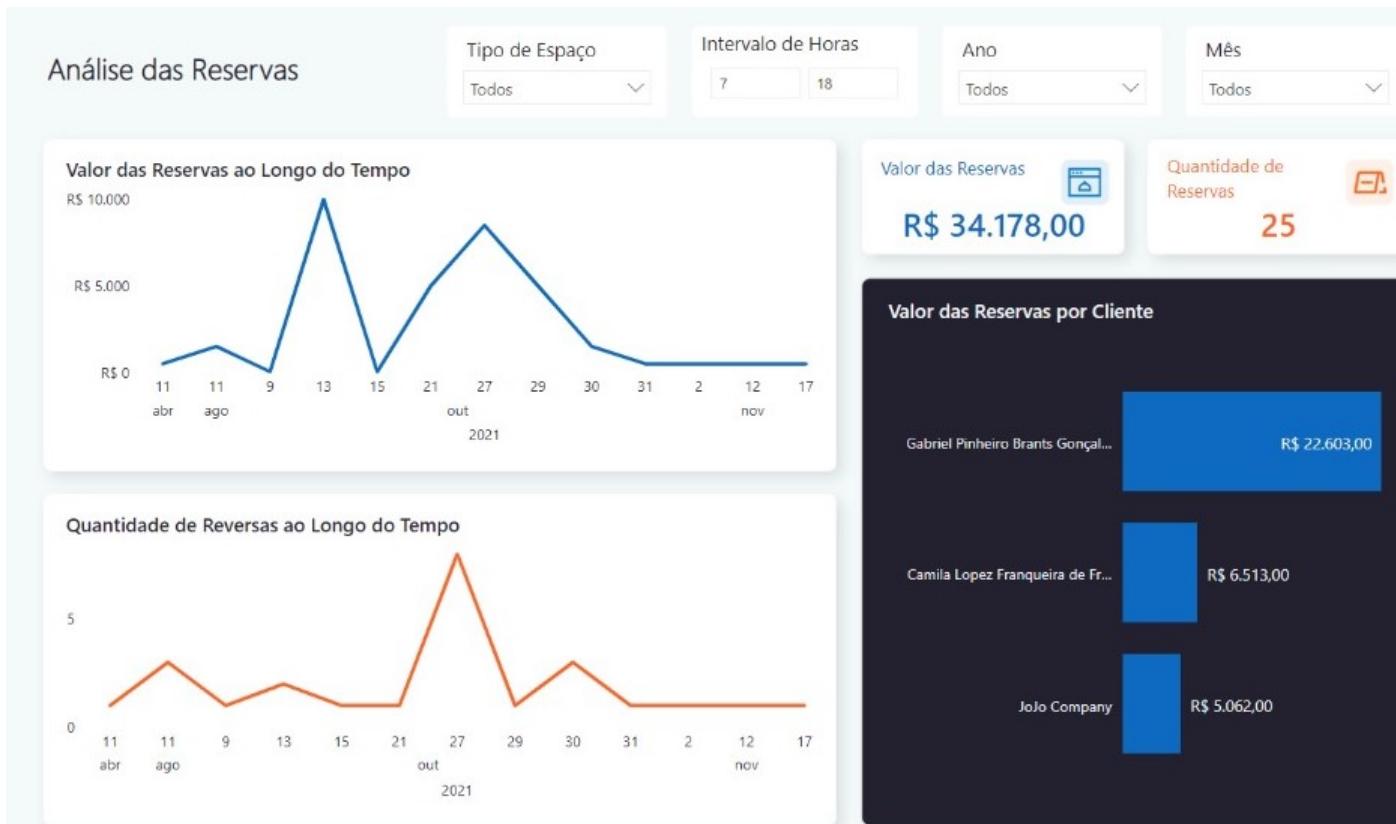
Para a criação das métricas de análise, priorizou-se o valor monetário e a quantidade de reservas feitas, sendo estas segmentadas ao longo do tempo e por cliente, sendo ainda possível a aplicação de filtros das características acima e do tipo de espaço.

4.8.3 Revisões

Até a última atualização do documento, não foram feitas revisões na implementação ou lógica desta funcionalidade.

4.8.4 Telas

Figura 97 – Exemplo de tela de relatórios na aplicação



Fonte: Os Autores

5 Considerações Finais

5.1 Primeiro Semestre

De todas as propostas estudadas pela equipe, a idéia de uma 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.

No entanto, houve problemas. O maior deles, a dificuldade na comunicação e compreensão entre os membros da equipe, pode ser atribuída a situação de pandemia da COVID-19, visto que todas as reuniões precisavam ser feitas através de videochamadas, e não possuiam 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.

Dado as dificuldades encontradas, a equipe ainda vê os resultados atuais como positivos; um website que, mesmo possuindo o ocasional bug, é funcional, utiliza diversas funções que conversam entre si e faz uma comunicação entre todas as camadas, pronto para ser expandido no próximo semestre. Mesmo com os percalços encontrados no caminho, para muitos, a criação de um site do completo zero é motivo de orgulho.

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.

5.2 Segundo Semestre

Após quase um ano de desenvolvimento, a equipe da NewGen se encontra nos momentos finais do projeto. Após a entrega de um MVP que foi considerado insatisfatório, junto aos problemas na entrega da documentação, a segunda parte do projeto começou com tropeços.

De início, foi necessária a revisão das atividades estipuladas ao segundo semestre;

um período mais curto cobrava a escolha de cumprir as entregas, ao custo da qualidade, ou diminuir o escopo em favor de um projeto bem-feito. A equipe escolheu fazer uma entrega menor, deixando aberta a oportunidade de melhorias e novas implementações a equipes futuras.

Os estudos foram intensificados, com cada parte da aplicação necessitando de novos conhecimentos, não apenas de Django, mas de HTML, CSS, AWS, e até mesmo PowerBI. A cada funcionalidade, se via a necessidade de continuar os estudos, buscar exemplos teóricos e práticos e até, em alguns casos, requisitar a ajuda de colegas de classe ou serviço para tirar dúvidas.

A equipe tentou se dividir igualmente, com todos fazendo as mesmas coisas no trabalho, mas foi como equipe que a realização veio; nem todos tem afinidade a programação, nem todos tem afinidade a documentação, nem todos são capazes de mexer em todas as áreas de um projeto. Portanto, a equipe escolheu por segregar as diferentes tarefas para as pessoas mais adequadas.

Para garantir que todos tivessem um nível básico de conhecimento do projeto, a equipe se reuniu durante fins de semana para "aulões" de Django, explicando as funcionalidades do sistema e como aplicar novas funcionalidades se aproveitando de conhecimento pré-existente, o que garantiu uma distribuição de conhecimento do projeto entre aqueles que tiveram menor interação com o código.

Mesmo com os percalços, a equipe se vê esperançosa com o projeto; após revisões, decisões difíceis e muitas discussões sobre responsabilidades, a aplicação da NewGen está funcional, cumprindo o que promete, e ainda aberta a melhorias.

Referências

AKITA, F. *Refletindo sobre RESOLUÇÃO de Problemas / O bug do Premiere*. 2019. Disponível em: <<http://web.archive.org/web/20211030220403/><https://www.youtube.com/watch?v=3W6xc4Yh2P0>>. Acesso em: 30 Oct. 2021. Citado na página 37.

AWS. *Breve panorama do CodePipeline*. 2021. Disponível em: <<http://web.archive.org/web/20211030223652/>https://docs.aws.amazon.com/pt_br/codepipeline/latest/userguide/welcome-introducing.html>. Acesso em: 30 Oct. 2021. Citado na página 63.

Caderno de Laboratório. *O modelo MTV no Django*. 2021. Disponível em: <<http://web.archive.org/web/20211030223732/><https://cadernodelaboratorio.com.br/o-modelo-mtv-no-django/>>. Acesso em: 30 Oct. 2021. Citado na página 51.

Carrick-Davies. 2020 ou 2021. Acesso em: 07 Jun. 2021. Citado na página 15.

COSTA, A. *O que aprendemos com a história que o WeWork nos contou*. 2018. Disponível em: <<http://web.archive.org/web/20211030211438/><https://movebla.com/o-que-aprendemos-com-a-hist%C3%B3ria-que-o-wework-nos-contou-556ba12af74d?gi=120fd2855306>>. Acesso em: 30 Oct. 2021. Citado na página 16.

COSTA, R. *Sistema de Medicina Ocupacional em Python/Django na prática*. 2021. Disponível em: <<http://web.archive.org/web/20211030232706/><https://www.udemy.com/course/sistema-de-medicina-ocupacional-em-python-django-na-pratica/>>. Acesso em: 30 Oct. 2021. Citado na página 80.

COWORKING BRASIL. *Um guia definitivo para você escolher o espaço ideal para construir a sua empresa*. 2017. Disponível em: <<http://web.archive.org/web/20211030211929/><https://coworkingbrasil.org/guia/escritorio-compartilhado/>>. Acesso em: 30 Oct. 2021. Citado 6 vezes nas páginas 15, 35 e 36.

COWORKING BRASIL. *Censo Coworking Brasil 2018*. 2019. Disponível em: <<http://web.archive.org/web/20211030223508/><https://coworkingbrasil.org/censo/2018/coworkers/>>. Acesso em: 30 Oct. 2021. Citado 9 vezes nas páginas 16, 17, 22, 23 e 24.

COWORKING BRASIL. *Censo Coworking Brasil 2018*. 2019. Disponível em: <<http://web.archive.org/web/20211030223404/><https://coworkingbrasil.org/censo/2018/>>. Acesso em: 30 Oct. 2021. Citado 7 vezes nas páginas 16, 18, 20, 21 e 23.

COWORKING BRASIL. *Manifesto Coworking / Coworking Brasil*. 2021. Disponível em: <<http://web.archive.org/web/20211030211807/><https://coworkingbrasil.org/manifesto/>>. Acesso em: 30 Oct. 2021. Citado 5 vezes nas páginas 14, 15 e 23.

COWORKING BRASIL. *O que é Coworking?* 2021. Disponível em: <<http://web.archive.org/web/20211030211542/><https://coworkingbrasil.org/como-funciona-coworking/>>. Acesso em: 30 Oct. 2021. Citado na página 14.

DELTA BC. *Coworking no Brasil: entenda o crescimento desse modelo*. 2021. Disponível em: <<http://web.archive.org/web/20211030213637/><https://www.deltabc.com.br/blog/coworking-no-brasil>>. Acesso em: 30 Oct. 2021. Citado na página 14.

DESKMAG. *Coworking in Brazil*. 2013. Disponível em: <<http://web.archive.org/web/20211030213534/https://www.deskmag.com/en/coworking-city-country-profiles/coworking-spaces-in-brazil-sao-paulo-812>>. Acesso em: 30 Oct. 2021. Citado 3 vezes nas páginas 14 e 21.

DESKMAG. *The 2018 Global Coworking Survey*. 2018. Disponível em: <<http://web.archive.org/web/20211030213502/https://www.deskmag.com/en/background-of-the-2018-global-coworking-survey-market-research>>. Acesso em: 30 Oct. 2021. Citado na página 13.

Django. *Documentation - Multiple databases*. 2021. Disponível em: <<http://web.archive.org/web/20211030233107/https://docs.djangoproject.com/en/3.2/topics/db/multi-db/>>. Acesso em: 30 Oct. 2021. Citado na página 85.

Django. *Documentation - Sending email*. 2021. Disponível em: <<https://web.archive.org/web/20211104050056/https://docs.djangoproject.com/en/3.2/topics/email/>>. Acesso em: 04 Nov. 2021. Citado na página 99.

Django. *Documentation - User authentication in Django*. 2021. Disponível em: <<http://web.archive.org/web/20211030232827/https://docs.djangoproject.com/en/3.2/topics/auth/>>. Acesso em: 30 Oct. 2021. Citado na página 80.

DRYSDALE, D. *phonenumbers Python Library*. 2021. Disponível em: <<http://web.archive.org/web/20211030233045/https://github.com/daviddrysdale/python-phonenumbers>>. Acesso em: 30 Oct. 2021. Citado na página 80.

DUBE, C.; GUMBO, V. Diffusion of innovation and the technology adoption curve: Where are we? the zimbabwean experience. 2017. Disponível em: <http://web.archive.org/web/20211030220741/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: 30 Oct. 2021. Citado na página 20.

FIZEZAP. *Preço do aluguel de salas e conjuntos comerciais sobe 0,32% em abril*. 2021. Disponível em: <<http://web.archive.org/web/20211030220528/https://fipezap.zapimoveis.com.br/preco-do-aluguel-de-salas-e-conjuntos-comerciais-sobe-032-em-abril/>>. Acesso em: 30 Oct. 2021. Citado na página 17.

FullCalendar. *Documentation - FullCalendar*. 2021. Disponível em: <<http://web.archive.org/web/20211030233525/https://fullcalendar.io/docs>>. Acesso em: 30 Oct. 2021. Citado na página 92.

GANDINI, A. The rise of coworking spaces: A literature review. 2015. Disponível em: <<http://web.archive.org/web/20211030220609/https://air.unimi.it/retrieve/handle/2434/618721/1152525/Ephemera ASN.pdf>>. Acesso em: 30 Oct. 2021. Citado 10 vezes nas páginas 15, 23, 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: <<http://web.archive.org/web/20211030220708/https://www.sciencedirect.com/>>.

science/article/abs/pii/S0148296321002460>. Acesso em: 30 Oct. 2021. Citado 6 vezes nas páginas 33, 34, 35 e 36.

Jazzband. *Documentation - django-hosts*. 2016. Disponível em: <<http://web.archive.org/web/20211030233147/https://django-hosts.readthedocs.io/en/latest/>>. Acesso em: 30 Oct. 2021. Citado na página 85.

LAMELAS, A. As 5 principais metodologias agile: vantagens e desvantagens. 2018. Disponível em: <<http://web.archive.org/web/20211030230957/https://www.xpand-it.com/blog/top-5-agile-methodologies/>>. Citado na página 41.

LUO, R. C. K. C. Y. Production of coworking spaces: Evidence from shenzhen, china. 2020. Disponível em: <<http://web.archive.org/web/20211030221025/https://www.sciencedirect.com/science/article/abs/pii/S0016718520300087>>. Acesso em: 30 Oct. 2021. Citado 6 vezes nas páginas 14, 18, 33 e 34.

MACORATTI, J. C. *CSharp - Validando CNPJ , CPF e PIS*. 2021. Disponível em: <http://web.archive.org/web/20211030232857/http://www.macoratti.net/alg_cpf.htm>. Acesso em: 30 Oct. 2021. Citado na página 80.

Mercado Pago. *Checkout API samples*. 2021. Disponível em: <<http://web.archive.org/web/20211030232500/https://github.com/mercadopago/card-payment-sample>>. Acesso em: 30 Oct. 2021. Citado na página 97.

Mercado Pago Developers. *Criar um pagamento*. [S.l.], 2021. Disponível em: <http://web.archive.org/web/20211030232424/https://www.mercadopago.com.br/developers/pt/reference/payments/_payments/post>. Acesso em: 30 Oct. 2021. Citado na página 97.

Microsoft. *Executar scripts do Python no Power BI Desktop*. [S.l.], 2021. Disponível em: <<http://web.archive.org/web/20211118190930/https://docs.microsoft.com/pt-br/power-bi/connect-data/desktop-python-scripts>>. Acesso em: 18 Nov. 2021. Citado na página 110.

MORISET, B. Building new places of the creative economy. the rise of coworking spaces. 2013. Disponível em: <<http://web.archive.org/web/20211030221120/https://halshs.archives-ouvertes.fr/halshs-00914075/>>. Acesso em: 30 Oct. 2021. Citado 2 vezes nas páginas 22 e 35.

OLIVAES, F. *Busca de CEP*. 2005. Disponível em: <<http://web.archive.org/web/20211030233013/https://www.republicavirtual.com.br/cep/index.php>>. Acesso em: 30 Oct. 2021. Citado na página 81.

Oracle Corporation. *MySQL Connector/Python Developer Guide*. [S.l.], 2021. Disponível em: <<http://web.archive.org/web/20211118190716/https://dev.mysql.com/doc/connector-python/en/>>. Acesso em: 18 Nov. 2021. Citado na página 110.

OrdinaryCoders. *Reset User Passwords in Django*. 2020. Disponível em: <<http://web.archive.org/web/20211030232351/https://www.ordinarycoders.com/blog/article/django-password-reset>>. Acesso em: 30 Oct. 2021. Citado na página 88.

- Pandas Development Team. *Pandas Documentation*. [S.l.], 2021. Disponível em: <<http://web.archive.org/web/20211118191015/https://pandas.pydata.org/docs/>>. Acesso em: 18 Nov. 2021. Citado na página 110.
- 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.
- Python. *Built-in template tags and filters*. [S.l.], 2021. Disponível em: <<http://web.archive.org/web/20211118200752/https://docs.djangoproject.com/en/dev/ref/templates/builtins/>>. Acesso em: 18 Nov. 2021. Citado na página 102.
- Python Wiki Brasil. *COMO GERAR IMAGEM DE QR CODE COM PYTHON*. 2020. Disponível em: <<http://web.archive.org/web/20211030233525/https://fullcalendar.io/docs>>. Acesso em: 04 Nov. 2021. Citado na página 99.
- Refactoring.guru. *Adapter*. 2021. Disponível em: <<http://web.archive.org/web/20211030230808/https://refactoring.guru/pt-br/design-patterns/adapter>>. Acesso em: 30 Oct. 2021. Citado na página 62.
- Refactoring.guru. *Strategy*. 2021. Disponível em: <<http://web.archive.org/web/20211030230907/https://refactoring.guru/pt-br/design-patterns/strategy>>. Acesso em: 30 Oct. 2021. Citado na página 63.
- SCHWABER, J. S. K. *The 2020 Scrum Guide*. [S.l.], 2020. Disponível em: <<http://web.archive.org/web/20211030231141/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: <<http://web.archive.org/web/20211030221433/https://www.sebrae.com.br/sites/PortalSebrae/ufs/mt/noticias/micro-e-pequenas-empresas-geram-27-do-pib-do-brasil,ad0fc70646467410VgnVCM2000003c74010aRCRD>>. Acesso em: 30 Oct. 2021. Citado na página 16.
- SPINUZZI, C. Working alone together. 2012. Disponível em: <<http://web.archive.org/web/20211030221956/https://journals.sagepub.com/doi/10.1177/1050651912444070>>. Acesso em: 30 Oct. 2021. Citado 2 vezes nas páginas 34 e 36.
- VICO, R. *Como enviar e-mail com python via SMTP em HTML e com Anexos - Tutorial COMPLETO*. 2021. Disponível em: <https://web.archive.org/web/20211104050313/https://www.youtube.com/watch?v=umvzsQLZYD4&t=684s&ab_channel=RonanVico>. Acesso em: 04 Nov. 2021. Citado na página 99.
- WAN, J. Prevention and landing of bubble. *international review of economics & finance*. 2018. Disponível em: <<http://web.archive.org/web/20211030205630/https://www.sciencedirect.com/science/article/pii/S105905601730816X>>. Acesso em: 30 Oct. 2021. Citado na página 15.
- WATERS-LYNCH, J. A theory of coworking: entrepreneurial communities, immaterial commons and working futures. 2018. Disponível em: <<http://web.archive.org/web/20211030210614/https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/A-theory-of-coworking-entrepreneurial-communities-immaterial-commons-and-working-futures/9921864026601341>>. Acesso em: 30 Oct. 2021. Citado na página 20.

Glossário

Airbnb	Sistema online de hospedagem, onde o usuário pode alugar, principalmente, casas de família para aluguel e atividades turísticas. - Citado em 28
API	Um intermediário de software que permite que dois aplicativos conversem entre si. - Citado em 92 , 93 , 97
AWS	Uma plataforma de serviços de computação em nuvem, pertencente a Amazon. - Citado em 63
Bootstrap	Framework de desenvolvimento de aplicações cliente, que é desenvolvida sobre a linguagem de programação CSS e, opcionalmente, JavaScript. - Citado em 51
carousel	Uma interface do bootstrap que possui a funcionalidade de "slides", onde o usuário pode movimentar os objetos para os lados. - Citado em 93
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 50 , 61
HTML	Uma linguagem de marcação utilizada na construção de páginas na Web. - Citado em 92 , 93
ITCSS	Forma de organização para arquivos de CSS, tornando eles mais fáceis de manipular e cuidar. - Citado em 59
Javascript	Linguagem de programação de script para front end, uma das mais comuns de uso. - Citado em 51 , 92
modal	Uma janela modal coloca o usuário em um modo onde ele pode interagir apenas com ela, tornando os comandos de menu e as outras janelas da aplicação inacessíveis. - Citado em 92
MySQL	Sistema gerenciador de banco de dados objeto relacional, desenvolvido como projeto de código aberto. - Citado em 52 , 78
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 56

Python	Python é uma linguagem de programação alto nível, orientada a objeto, front e back-end, que prioriza a legibilidade do código. - Citado em 50 , 56 , 61
QR	Um código de barras, ou barrametrico, bidimensional, que pode ser facilmente escaneado usando a maioria dos telefones celulares equipados com câmera. - Citado em 6 , 25 , 39 , 69 , 72 , 99 , 100 , 101
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 espacos por underscores. - Citado em 58
SSL	Um certificado digital que autentica a identidade de um site e possibilita uma conexão criptografada; protocolo de segurança que cria um link criptografado entre um servidor Web e um navegador Web. - Citado em 76

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.

A.1 Primeiro Semestre

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

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

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 towards a goal. The Y-axis ranges from -1 to 1, and the X-axis shows dates from Jun 15 to Jul 7. A legend indicates: Diretriz (Goal) at 1, Valores restantes (Remaining values) at 0, and Dias não úteis (Non-working days) at -1.

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. Q ? 🔍 ⚙

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 (Grey line), Valores restantes (Red line), Dias não úteis (Grey area)

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 shows a single horizontal red line at the 0 mark on the Y-axis, spanning from July 15 to August 08. The legend indicates three series: 'Diretriz' (grey line), 'Valores restantes' (red line), and 'Dias não úteis' (white area).

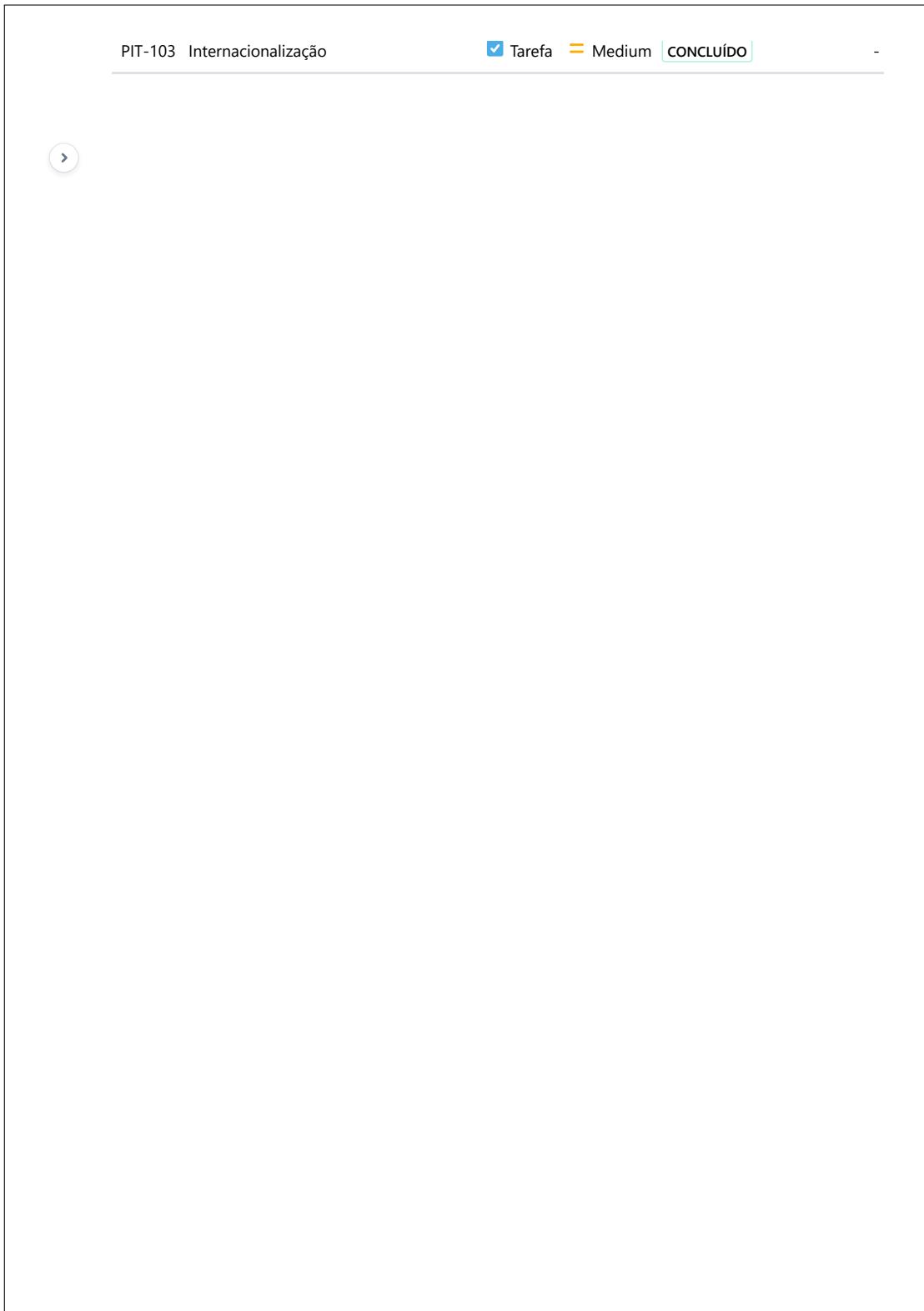
Relatório de Estado

* 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 Itens](#)

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

A.2 Segundo Semestre

Your work Projects Filters Dashboards People Apps

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 2

Closed sprint, ended by Camila Lopez Franqueira de França 27/Sep/21 3:20 AM - 30/Sep/21 9:14 PM View linked pages

Terminar planejamentos para apresentação da aula 3.

The burndown chart displays Story Points on the Y-axis (0 to 50) against TIME on the X-axis (Sep 27 3:20 to Sep 30 21:14). A red line shows the progress of work completed. A legend indicates: Guideline (grey shaded area), Remaining Values (red line), and Non-Working Days (white space). The burndown chart shows that work began on Sep 27 at 3:20 and progressed steadily until Sep 30 at 21:14, where all story points were completed.

Status Report

* Issue added to sprint after start time

Completed Issues

Key	Summary	Issue Type	Priority	Status	Story Points (- → 57)
PIT-136	Apresentação powerpoint	Story	Medium	DONE	- → 5
PIT-137	Planejamento em conjunto	Story	Medium	DONE	- → 7
PIT-138	Criação de planilha de planejamento	Story	Medium	DONE	- → 5
PIT-139	Planejamento de testes	Story	Medium	DONE	- → 15
PIT-140	Planejamento de deploy	Story	Medium	DONE	- → 15
PIT-142 *	Planejamento scrum	Story	Medium	DONE	- → 10

Your work ▾ Projects ▾ Filters ▾ Dashboards ▾ People ▾ Apps ▾

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 3

Closed sprint, ended by Camila Lopez Franqueira de França 30/Sep/21 9:14 PM - 07/Oct/21 10:31 PM View linked pages

Status Report

* Issue added to sprint after start time

Completed Issues						View in Issue Navigator
Key	Summary	Issue Type	Priority	Status	Story Points (82 → 78)	
PIT-143	Consertar migração backend para banco de dados	Story	Medium	DONE	13	
PIT-144	Criar mais testes	Story	Medium	DONE	13	
PIT-145	Recuperação de senha	Story	Medium	DONE	11	
PIT-147	Estudo - Integração com o pagamento (API pagamento)	Story	Medium	DONE	15 → 11	
PIT-150	Atualização da documentação	Story	Medium	DONE	17	
PIT-151	Acompanhamento individual do projeto	Story	Medium	DONE	13	

Issues Not Completed						View in Issue Navigator
Key	Summary	Issue Type	Priority	Status	Story Points (61)	
PIT-130 *	Dar deploy da aplicação / servidor	Story	High	IN PROGRESS	16	
PIT-146	Layout telas do Aluguel	Story	Medium	IN PROGRESS	17	
PIT-148	Ajustar calendário	Story	Medium	TO DO	13	
PIT-149	Criar esteira de integração contínua	Story	High	IN PROGRESS	15	

Your work ▾ Projects ▾ Filters ▾ Dashboards ▾ People ▾ Apps ▾

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 4

Closed sprint, ended by Fernando Sousa Silva 07/Oct/21 10:33 PM - 15/Oct/21 10:01 PM View linked pages

Status Report

* Issue added to sprint after start time

Completed Issues

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (86)
PIT-130	Dar deploy da aplicação / servidor	Story	High	DONE	16
PIT-149	Criar esteira de integração contínua	Story	High	DONE	15
PIT-155	Revisão de recuperação de senha	Story	Medium	DONE	-
PIT-156	Implementação da API do pagamento	Story	Medium	DONE	19
PIT-184 *	Suporte ao backend	Story	Medium	DONE	10
PIT-185 *	Acompanhamento individual do progresso	Story	Medium	DONE	13
PIT-186 *	Alterações no documento	Story	Medium	DONE	13
PIT-187 *	Analise e implementação "coverage"	Story	Medium	DONE	-

Issues Not Completed

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (30)
PIT-146	Layout telas do Aluguel	Story	Medium	IN PROGRESS	17
PIT-148	Ajustar calendário	Story	Medium	IN PROGRESS	13

Your work Projects Filters Dashboards People Apps

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 5

Closed sprint, ended by Camila Lopez Franqueira de França 15/Oct/21 10:59 PM - 22/Oct/21 11:43 PM View linked pages

STORY POINTS

TIME

Status Report

Completed Issues

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (95)
PIT-157	Revisão API pagamento	Story	Medium	DONE	15
PIT-188	Estudo Autorização de Entrada	Story	Medium	DONE	10
PIT-189	Criar backend da tela de customização	Story	Medium	DONE	17
PIT-190	Atualização do documento	Story	Medium	DONE	15
PIT-191	Investigação de erro 500	Story	Medium	DONE	13
PIT-193	Criar a tela de customização	Story	Medium	DONE	10
PIT-196	Criar customização de espaços	Story	Medium	DONE	15

Issues Not Completed

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (56)
PIT-146	Layout telas do Aluguel	Story	Medium	IN PROGRESS	9
PIT-148	Ajustar calendário	Story	Medium	IN PROGRESS	19
PIT-194	Pesquisar sistema de log	Story	Medium	TO DO	15
PIT-197	Documentação	Story	Medium	TO DO	13

Your work ▾ Projects ▾ Filters ▾ Dashboards ▾ People ▾ Apps ▾

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 6

Closed sprint, ended by Camila Lopez Franqueira de França 23/Oct/21 12:58 AM - 28/Oct/21 11:16 PM View linked pages

Status Report

Completed Issues

View in Issue Navigator					
Key	Summary	Issue Type	Priority	Status	Story Points (100)
PIT-148	Ajustar calendário	Story	= Medium	DONE	19
PIT-165	Autorização de entrada	Story	= Medium	DONE	17
PIT-194	Pesquisar sistema de log	Story	= Medium	DONE	15
PIT-197	Documentação	Story	= Medium	DONE	17
PIT-205	Funcionalidade de espaço	Story	= Medium	DONE	18
PIT-212	Tela principal do administrador	Story	= Medium	DONE	14

Issues Not Completed

View in Issue Navigator					
Key	Summary	Issue Type	Priority	Status	Story Points (17)
PIT-204	Criar telas de reserva de equipamentos	Story	= Medium	IN PROGRESS	17

Issues completed outside of this sprint

View in Issue Navigator					
Key	Summary	Issue Type	Priority	Status	Story Points (9)
PIT-146	Layout telas do Aluguel	Story	= Medium	DONE	9

Your work Projects Filters Dashboards People Apps

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 7

Closed sprint, ended by Camila Lopez Franqueira de França 29/Oct/21 12:47 AM - 04/Nov/21 11:17 PM View linked pages

Status Report

Completed Issues

Key	Summary	Issue Type	Priority	Status	Story Points (157)
PIT-232	Ajuste de tela de pagamento	Story	Medium	DONE	10
PIT-234	Refatorar Design Pattern do projeto	Story	Medium	DONE	13
PIT-235	Refatorar Django hosts	Story	Medium	DONE	8
PIT-236	Pesquisa Custom Templates	Story	Medium	DONE	20
PIT-242	Verificar cumprimento dos tópicos do Dicas Ivan	Story	Medium	DONE	10
PIT-244	Avaliar dados para relatório de coworking	Story	Medium	DONE	10
PIT-245	Planejar telas editáveis pelo cliente	Story	Medium	DONE	15
PIT-246	Fazer a apresentação de slides	Story	Medium	DONE	7
PIT-247	Finalizar funcionalidades no documento	Story	Medium	DONE	10
PIT-248	Mover bibliografia para web archive	Story	Medium	DONE	5
PIT-249	Revisar siglas e glossários	Story	Medium	DONE	5
PIT-250	Revisar interface de gerenciamento de espaços	Story	Medium	DONE	8
PIT-252	Atualizar dicionario de dados	Story	Medium	DONE	10
PIT-253	Criar MER e DER	Story	Medium	DONE	8
PIT-255	Criar novas tabelas de relação	Story	Medium	DONE	12
PIT-256	"Arrastar pro lado no Jira"	Story	Medium	DONE	1
PIT-257	Postagens no Blog	Story	Medium	DONE	5

PIT-204	Criar telas de reserva de equipamentos	Story	Medium	IN PROGRESS	17	
PIT-230	Seleção de espaços do calendário	Story	Medium	IN PROGRESS	13	
PIT-231	Integrar back e front no calendário	Story	Medium	TO DO	19	
PIT-233	Sistema de log	Story	Medium	TO DO	17	
PIT-237	Revisão da documentação da arquitetura/tecnologias	Story	Medium	TO DO	11	
PIT-238	Criar lógica do dashboards na tela do adm	Story	Medium	IN PROGRESS	12	
PIT-239	Pegar dados da API do dashboards	Story	Medium	IN PROGRESS	12	
PIT-240	Criar interface de dashboard	Story	Medium	IN PROGRESS	8	
PIT-241	Editar equipamentos	Story	Medium	TO DO	15	
PIT-243	Deletar equipamentos	Story	Medium	IN PROGRESS	10	
PIT-251	Modificação do código Django	Story	Medium	IN PROGRESS	15	
PIT-254	Resolver problemas de hardware pessoal	Story	Medium	IN PROGRESS	17	
PIT-258	Criar equipamentos	Story	Medium	TO DO	13	
PIT-259	Integrar tela de equipamentos com backend	Story	Medium	TO DO	17	

Your work Projects Filters Dashboards People Apps

Projects / Projeto Integrado: TCC / quadro PIT / Reports

Sprint Report

PI2 - Sprint semana 8

Closed sprint, ended by Camila Lopez Franqueira de França 04/Nov/21 11:10 PM - 12/Nov/21 1:31 AM View linked pages

Status Report

* Issue added to sprint after start time

Completed Issues

Key	Summary	Issue Type	Priority	Status	Story Points (32 → 251)
PIT-168	Entrega de relatórios periódicos	Story	Medium	DONE	- → 17
PIT-238 *	Criar lógica do dashboards na tela do adm	Story	Medium	DONE	12 → 19
PIT-239 *	Pegar dados da API do dashboards	Story	Medium	DONE	12
PIT-240 *	Criar interface de dashboard	Story	Medium	DONE	8
PIT-260	Criar lógica de realizar reserva ou pagamento total	Story	Medium	DONE	- → 19
PIT-262 *	Criar post para o blog	Story	Medium	DONE	- → 5
PIT-263 *	Terminar interface de customização da página index	Story	Medium	DONE	- → 19
PIT-264 *	Criar lógica de pegar template do cliente	Story	Medium	DONE	- → 15
PIT-265 *	Criar interface de customizar a aparência do sistema	Story	Medium	DONE	- → 15
PIT-266 *	Criar lógica da interface de aparência	Story	Medium	DONE	- → 19
PIT-267 *	Criar interface do funcionário	Story	Medium	DONE	- → 16
PIT-268 *	Leitura de qrcode	Story	Medium	DONE	- → 15
PIT-270 *	Atualizar funcionalidades	Story	Medium	DONE	- → 11
PIT-271 *	Completar traduções da internacionalização	Story	Medium	DONE	- → 11
PIT-272 *	Preparar apresentação	Story	Medium	DONE	- → 5
PIT-274 *	Garantir que a nota da aplicação é favorável	Story	Medium	DONE	- → 17
PIT-275 *	Análise estática do código	Story	Medium	DONE	- → 15

The screenshot shows a Jira interface with the following details:

Header navigation: Your work, Projects, Filters, Dashboards, People, Apps.

Search bar and other icons: magnifying glass, refresh, help, gear.

Section title: Issues Not Completed.

Table header with columns: Key, Summary, Issue Type, Priority, Status, Story Points (164 → 196).

Table data:

Key	Summary	Issue Type	Priority	Status	Story Points (164 → 196)
PIT-169	Tentativa de Transferência de responsabilidade	Story	Medium	IN PROGRESS	- → 19
PIT-204 *	Criar telas de reserva de equipamentos	Story	Medium	IN PROGRESS	17
PIT-230 *	Seleção de espaços do calendário	Story	Medium	IN PROGRESS	13
PIT-231 *	Integrar back e front no calendário	Story	Medium	TO DO	19
PIT-233 *	Sistema de log	Story	Medium	IN PROGRESS	17
PIT-237 *	Revisão da documentação da arquitetura/tecnologias	Story	Medium	TO DO	11
PIT-241 *	Editar equipamentos	Story	Medium	IN PROGRESS	15
PIT-243 *	Deletar equipamentos	Story	Medium	IN PROGRESS	10
PIT-251 *	Modificação do código Django	Story	Medium	IN PROGRESS	15
PIT-254 *	Resolver problemas de hardware pessoal	Story	Medium	IN PROGRESS	17
PIT-258 *	Criar equipamentos	Story	Medium	IN PROGRESS	13
PIT-259 *	Integrar tela de equipamentos com backend	Story	Medium	TO DO	17
PIT-273 *	Popular banco	Story	Medium	IN PROGRESS	- → 13

APÊNDICE B – MER e DER

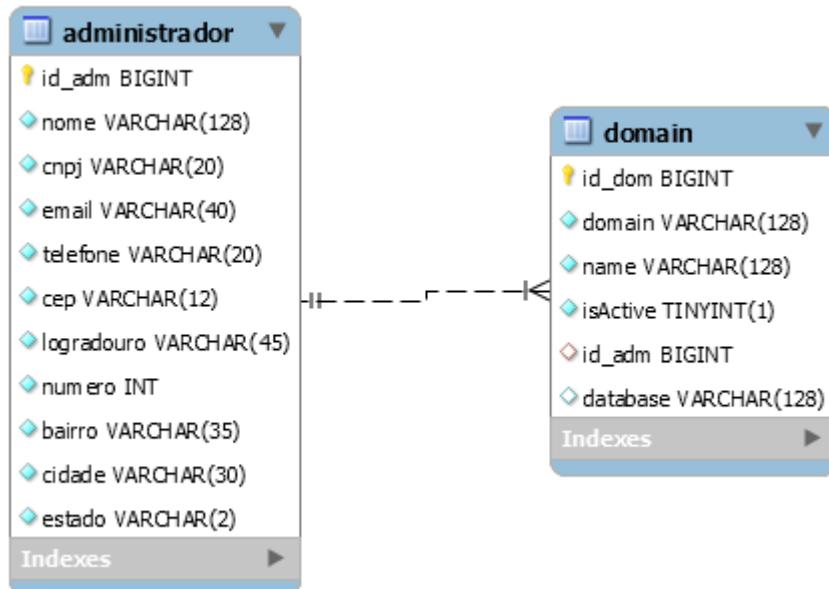
Para a definição do banco de dados, foi necessária a criação de MERs ([Figura 98](#) e [Figura 100](#)) e DERs ([Figura 99](#) e [Figura 101](#)) da aplicação, divididas entre Banco de Dados da NewGen e o Banco de Dados do Coworking.

Figura 98 – MER - Modelo Entidade-Relacionamento - NewGen



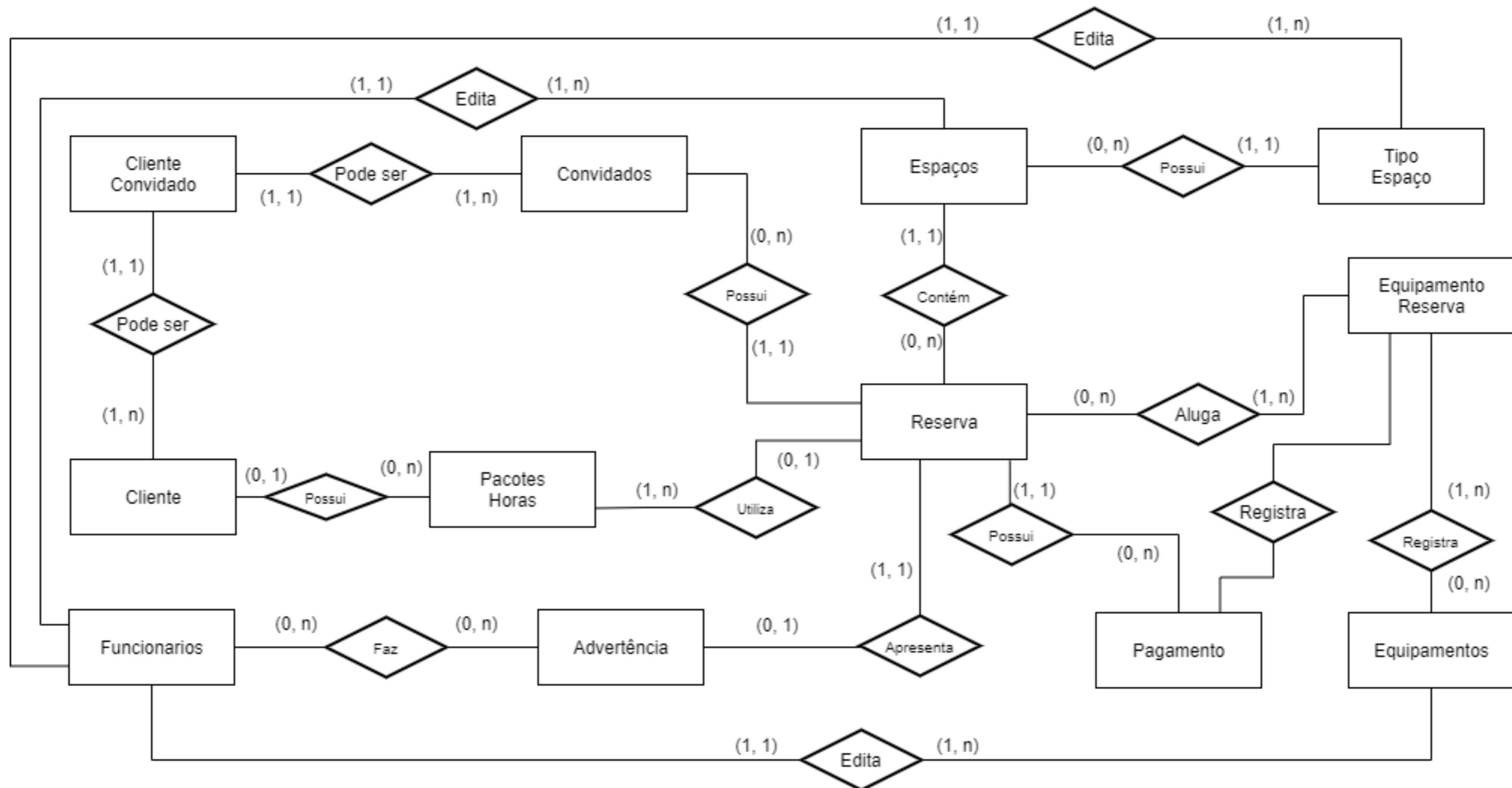
Fonte: Os Autores

Figura 99 – DER - Diagrama Entidade-Relacionamento - NewGen



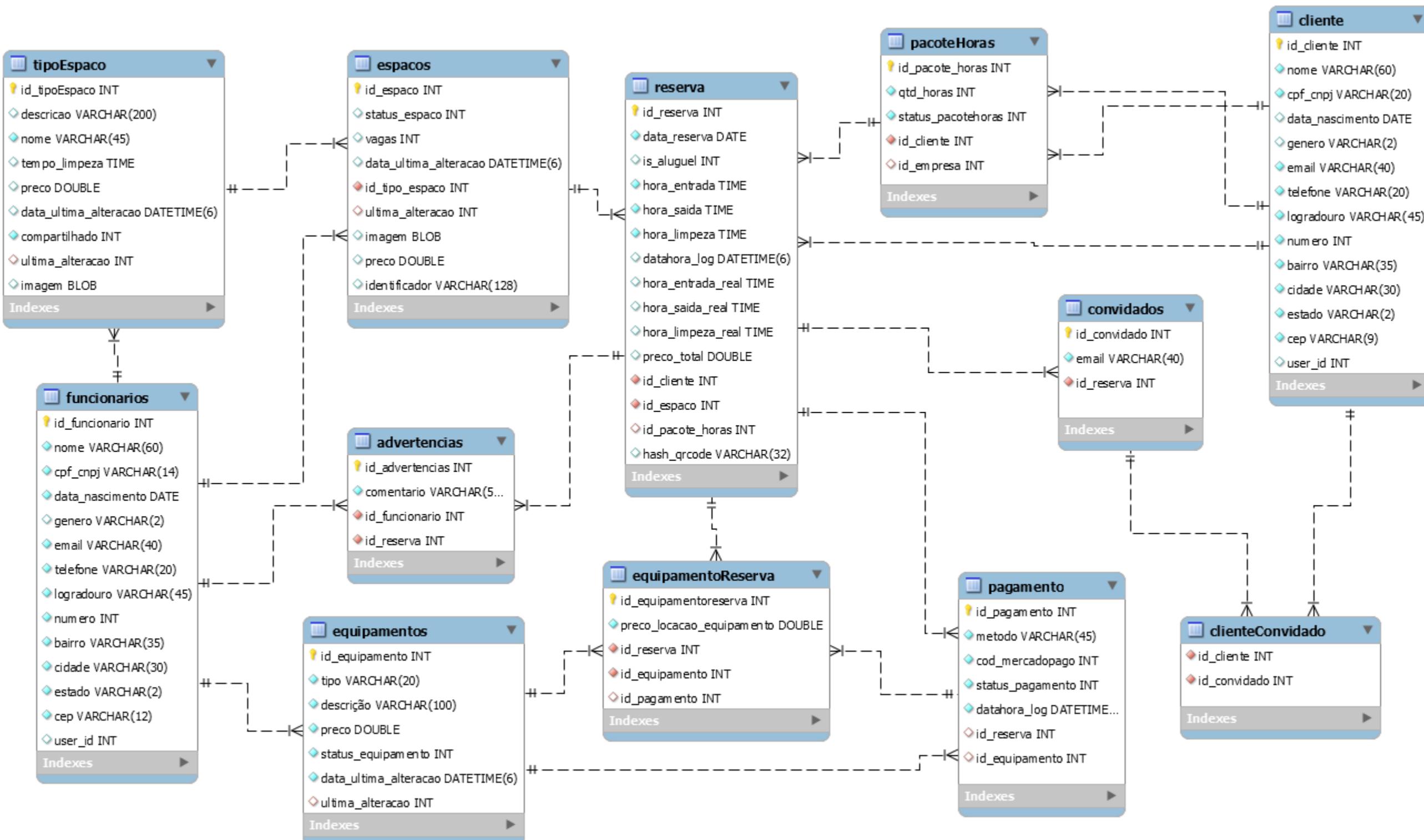
Fonte: Os Autores

Figura 100 – MER - Modelo Entidade-Relacionamento - Coworking



Fonte: Os Autores

Figura 101 – DER - Diagrama Entidade-Relacionamento - Coworking



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 'administrador'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_adm	BIGINT		PK	Deve ser gerada pelo sistema.
nome	VARCHAR	128		Nome completo da empresa.
cnpj	VARCHAR	20		CNPJ da empresa.
email	VARCHAR	40		E-mail da empresa.
telefone	VARCHAR	20		Telefone da empresa.
cep	VARCHAR	12		CEP do endereço da empresa.
logradouro	VARCHAR	45		Logradouro do endereço da empresa.
numero	INT	5		Número do endereço da empresa.
bairro	VARCHAR	35		Bairro do endereço da empresa.
cidade	VARCHAR	30		Cidade do endereço da empresa.
estado	CHAR	2		Sigla do estado da empresa.

Quadro 8 – Descrição dos dados da tabela 'domain'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_dom	BIGINT		PK	Deve ser gerada pelo sistema.
domain	VARCHAR	128		Endereço Web do domínio.
name	VARCHAR	128		Nome do domínio.
isActive	TINYINT	1		Indica se o domínio está ativo.
id_adm	BIGINT		FK	Chave do administrador responsável pelo domínio.
database	VARCHAR	128		Acesso a base de dados do domínio.

Quadro 9 – 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 10 – Descrição dos dados da tabela 'espaço'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_espaço	INT	15	PK	Deve ser gerada pelo sistema.
id_tipo_espaço	INT	15	FK	ID do tipoEspaço que define.
ultima_alteração	INT	15	FK	ID do funcionário que atualizou.
identificador	VARCHAR	128		Identificador da sala.
id_tipo_espaço	INT	15	FK	ID do tipoEspaço que define.
status	TINYINT	2		0 - Ativo, 1 - Desativado

Quadro 11 – Descrição dos dados da tabela 'equipamentos'

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

Quadro 12 – Descrição dos dados da tabela 'pagamento'

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

Quadro 13 – Descrição dos dados da tabela 'funcionario'

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

Quadro 14 – Descrição dos dados da tabela 'cliente'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_cliente	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 15 – 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 16 – Descrição dos dados da tabela 'reserva'

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

Quadro 17 – Descrição dos dados da tabela 'equipamentoReserva'

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

Quadro 18 – Descrição dos dados da tabela 'convidados'

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

Quadro 19 – Descrição dos dados da tabela 'advertencia'

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

Quadro 20 – Descrição dos dados da tabela 'clienteConvidado'

NOME DA COLUNA	TIPO	TAMANHO	CHAVE	DEFINIÇÃO
id_cliente	INT	15	FK	Deve ser gerada pelo sistema.
id_convidado	INT	15	FK	Deve ser gerada pelo sistema.

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 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.

Configurações e afins

 newgenprojetointegrado.blogspot.com/2021/08/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.

05/06/2021

Durante as férias...

 newgenprojetointegrado.blogspot.com/2021/09/durante-as-ferias.html

Nosso projeto não parou de sofrer mudanças ao longo do período de recesso acadêmico.

Até então, a aplicação estava contendo um banco de dados único para todos os supostos clientes NewGen e assim, não seguia os acordos da LGPD. Resolvemos então transformar nosso banco de dados único para cada cliente, exclusividade garantida para eles.

Nesse ímpeto, desenvolvemos também a ideia de manter para cada cliente um domínio único. Ainda estamos pensando se essa escolha é de bom tom, se será mesmo necessário essa funcionalidade para preservar o estilo WhiteLabel ou se apenas um subdomínio bastaria.

Outra questão que otimizamos foi a criação de um banco de dados automaticamente ao cadastrar um domínio próprio do cliente no sistema. No formulário de cadastro, colocamos uma barra de escolha desse domínio, onde registramos 5 até agora.

A internacionalização, que antes enfrentava alguns bugs do tipo não guardar informação de idioma na troca de páginas, foi corrigida.

Camila França, 27/09/2021

Atualizações de retomada

 newgenprojetointegrado.blogspot.com/2021/10/atualizacoes-de-retomada_4.html

Durante esse período de retomada dos trabalhos no projeto integrado tivemos alguns problemas com a conexão do site.

Ele apresentou dificuldade de reconhecimento do HTTPS em certas ocasiões, portanto, Gabriel e Fernando se encarregaram de rever esse quesito.

Mantivemos as funções de front com Lucas e Bruna, sendo essa última também responsável por parte do backend nessa segunda etapa do projeto devida a grande demanda.

Camila e Gabriel seguem no backend, sendo eu encarregada principalmente de rever a parte de trâmites do aluguel e a inserção da API de pagamento. Nessa semana foi apenas realizados estudos para aderir à melhor abordagem a seguir e implementar nas demais semanas.

Beatriz segue encarregada da documentação como um todo.

04/10/21

A parte 2 é mais corrida

 newgenprojetointegrado.blogspot.com/2021/10/a-parte-2-e-mais-corrida_11.html

Percebendo o quanto esse semestre seria extremamente corrido em relação às entregas, decidimos dar um ultimato no grupo: ou todos cooperavam ou era melhor desistir naquele momento.

As sprints estavam todas erradas, em sua maioria, não sendo cumpridas todas as tarefas e aquilo não poderia ficar assim.

Foi a partir dessa semana que todos se reuniram para decidir o que seria feito dali em diante.

Muitas revisões precisavam ser feitas: o calendário estava incompleto, alguns erros de ortografia no site, bugs de internacionalização e detalhes no projeto como um todos estavam desfalcados.

Cada um sentou de frente para discutir quem seria o encarregado de cada ponto e decidimos seguir em frente, mesmo sem a cooperação de todos.

11/10/21

Implementações e afins

 newgenprojetointegrado.blogspot.com/2021/10/implementacoes-e-afins.html

Essa semana se sucedeu como as demais desde que retomamos o projeto integrado 2: com sprints bagunças, com tarefas sem entregar e excesso de trabalho em poucas mãos.

É inegável que nossa equipe está desfalcada, se mostrando nítido inclusive durante as reuniões com os professores. Mas não ousamos desistir.

Apesar de sobrecarregados, Gabriel e Camila fizeram a integração com a API de pagamento, ele se mostrando sempre muito disposta a realizar muito além de suas próprias atividades e ela dando o máximo de si para aprender habilidades não existentes antes para suprir a demanda do grupo.

Bruna também fez por merecer realizando atividades de seu parceiro do front end, Lucas, além de suas próprias na aprovação de entrada por meio de QR Code e customização de telas.

Fernando investigou durante essa semana erros do projeto relacionados à conexão e tratou de colocar em andamento os logs até então não realizados.

Bia faltou com a comunicação antes acordada pelo grupo, mas tentou realizar suas tarefas referentes à documentação em tempo, o que nos foi notificado não ser possível apenas no final da sprint.

20/10/21

Tirando alguns atrasos

 newgenprojetointegrado.blogspot.com/2021/10/tirando-alguns-atrasos.html

Essa semana foi muito proveitosa.

Apesar de algumas tarefas ainda terem ficado incompletas, nosso gráfico de burndown se mostrou bem mais coerente. Gabriel e Bruna foram protagonistas nessa sprint, conseguindo terminar a maioria de suas tarefas, assim como Fernando e Beatriz.

Lucas e Camila ficaram mais atrás com suas atividades, se esforçaram para entregar tudo a tempo, mas não puderam contar tanto com a equipe, já muito atarefada com suas próprias atividades.

Bruna se ocupou de terminar a funcionalidade de Qr code, onde o sistema gera esse código para o cliente do coworking no intuito de validar sua entrada no estabelecimento. Este é enviado por email ao mesmo.

Gabriel por fim conseguiu implementar as alterações necessárias no calendário e entre outras partes defasadas do back end.

Camila estava encarregada de criar a funcionalidade de aluguel de equipamentos/recursos no coworking porém não conseguiu integrar de fato o código.

Lucas agora se ocupou do banco de dados, já muito defasado devido a quantidade de mudanças que ocorreram na lógica do sistemas, porém a maioria de suas tarefas não foram entregues.

Bia e sua ocupação com tudo que diz respeito ao documento, ainda não concluiu suas atividades nessa sprint, faltando poucos tópicos para tal.

Fernando, com sua obrigação referente à arquitetura do sistema, estudou para implementar o sistema de log, mas ainda não o fez de fato. Deixaremos para a sprint seguinte.

Camila França 28/10/2021

2 semanas restantes...

 newgenprojetointegrado.blogspot.com/2021/11/2-semanas-restantes.html

Não podemos negar que agora estamos na reta final e isso traz a sensação de ansiedade à flor da pele.

Foi uma semana de muitos empecilhos, tarefas inacabadas por dificuldade de desenvolvimento, mas ainda sim conseguimos progredir em alguns pontos.

Revisamos todo o site do professor Ivan para ter certeza de que estávamos seguindo o caminho correto e saber se não deixamos algo escapar.

Claro que contamos com o apoio de todos, porém nem sempre atingimos resultados esperados.

Nossa aplicação gerou um erro misterioso ao ser rodada e acessada na web. Problema este que impossibilitou grande parte do progresso acontecer. Não conseguíamos testar nosso desenvolvimento e vê-lo em prática, rodando.

Resolvemos ao final dessa semana (hoje) e pretendemos dar continuidade e recuperar o tempo perdido nessa sprint.

Camila França - 03/11/2021

Reta final

 newgenprojetointegrado.blogspot.com/2021/11/reta-final.html

Agora, a uma semana da entrega final, os ânimos estão ainda mais tensos.

Tranquilos com os progressos que já efetuamos mas ansiosos para saber se serão suficientes para uma boa classificação.

Durante essa sprint, muitas modificações foram feitas, principalmente sobre a tela de customizações dos clientes e ao dashboards disponíveis para administradores do coworking.

A parte de leitura de QR code também avançou muito assim como a tela de pagamento, que agora se faz possível realizar apenas uma reserva (pagamento um sinal de 10% do valor do aluguel) ou realizar o pagamento total.

Camila França 11/11/2021

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 "Coding 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

Ensínamento 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.

Reunião 01 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 16 de setembro de 2021

Horário da Reunião: 19:30 às 21:20

Pautas:

- Controle de Estoque e Sistema de Fidelidade são jogadas para um teorizado 3º semestre (alunos que queiram assumir o projeto em turmas futuras);
- Adicionada nova funcionalidade: Cadastro e Personalização de Coworking;
- Edição do Banco de Dados para acompanhar a nova definição (Bancos individuais para cada Cliente);
- Relatórios Periódicos definido como último processo a ser implementado (dependente dos status e dados das implementações anteriores);
- Avaliar correção da internacionalização do sistema no servidor AWS, para apresentação na Quinta-Feira.
- Updates iniciais para o JIRA.

Reunião 02 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 23 de setembro de 2021

Horário da Reunião: 18:30 às 21:55

Pautas:

- Finalização da apresentação para professores
- Apresentação do planejamento do segundo semestre para professores
 - Discussão sobre melhorias no planejamento de implementação
- Definição dos próximos passos, anotações no Jira

Reunião 03 – NewGen

Escritura da Ata: Gabriel P.

Data da Reunião: 27 de setembro de 2021

Horário da Reunião: 21:00 às 22:20

Pautas:

- Atualização das necessidades da equipe;
- Criação da sprint da semana no Jira;
- Discussão de planejamento da AWS:
 - Criação de 3 planos para o gerenciamento multi-tenancy de domínios:
 - Plano A: seguir com a ideia de multidomínios
 - Plano B: usar subdomínios e gerar ao usuário um subdomínio pela rota 53
 - Plano C: usar o URI
 - Análise de exportação da arquitetura para o GIT por conta da integração contínua
- Repassagem de informações da reunião passada para os que estavam ausente;
- Divisão de 3 grandes planejamentos necessários para a apresentação do dia 30:
 - Planejamento de testes
 - Planejamento de deploy
 - Planejamento Scrum
- Agendamento de uma nova reunião importante para definir vários tópicos de funcionalidades e responsabilidades.

Reunião 04 – NewGen

Escritura da Ata: Beatriz H. F. Oliveira

Data da Reunião: 04 de outubro de 2021

Horário da Reunião: 21:25 às 21:45

Pautas:

- Atualização de progresso da equipe
 - Gabriel: correção do calendário de reserva; finalização da correção do arquivo que trata a conexão entre banco e back-end; início do processo de testes para banco de dados
 - Bruna: implementação da funcionalidade de recuperação da senha; encontrando problemas com link enviado via e-mail
 - Fernando: correção do deploy na arquitetura
 - Camila: estudo na API de pagamento do MercadoPago
 - Lucas: produzindo telas de aluguel.

Reunião 05 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 07 de outubro de 2021

Horário da Reunião: 18:20 às 22:35

Pautas:

- Apresentação de progresso para os professores
- Implementação do Coverage na aplicação
- Discussão sobre redistribuição de papéis, cobrança da equipe
- Planejamento para próxima sprint
- Verificação de progresso da equipe

Reunião 06 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 14 de outubro de 2021

Horário da Reunião: 18:05 às 22:35

Pautas:

- Apresentação do progresso semanal aos professores
- Definição de próximos passos para implementação da API de pagamento
- Verificação de acessos
- Criação da próxima sprint no Jira

Reunião 07 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 21 de outubro de 2021

Horário da Reunião: 18:10 às 22:55

Pautas:

- Discussão sobre entregas e novo planejamento de distribuição de atividades
- Verificação de progresso
- Mockup de apresentação com a própria equipe, em questão de não haver aula.
- Criação da próxima sprint do Jira, redefinição de metas.

Reunião 08 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 28 de outubro de 2021

Horário da Reunião: 17:30 às 23:10

Pautas:

- Apresentação do progresso para os professores
- Pequena celebração
- Estipulação de próximos objetivos, verificação do progresso da equipe
- Criação das tarefas da próxima semana no Jira, buscando evitar mais atrasos.

Reunião 09 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 04 de novembro de 2021

Horário da Reunião: 18:10 às 22:20

Pautas:

- Apresentação do progresso semanal aos professores
- Revisão dos tópicos do Dicas Ivan a serem cumpridos
- Reprojeção de objetivos para o semestre: mais funcionalidades descartadas
- Preparo da próxima sprint e inicio desta

Reunião 10 – NewGen

Escritura da Ata: Beatriz H.

Data da Reunião: 11 de novembro de 2021

Horário da Reunião: 18:25 às 23:45

Pautas:

- Verificação de partes não-abordadas do projeto, para finalizar durante a última sprint
- Preparo da sprint a ser iniciada após apresentação
- Apresentação para os professores
- Definição final das entregas possíveis

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 White-Label 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			
id_func	nome	cpf	...
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			
id_clien	nome	cpf	...
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				
id_func	nome	cpf	...	id_coworking
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				
id_clien	nome	cpf	...	id_coworking
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?

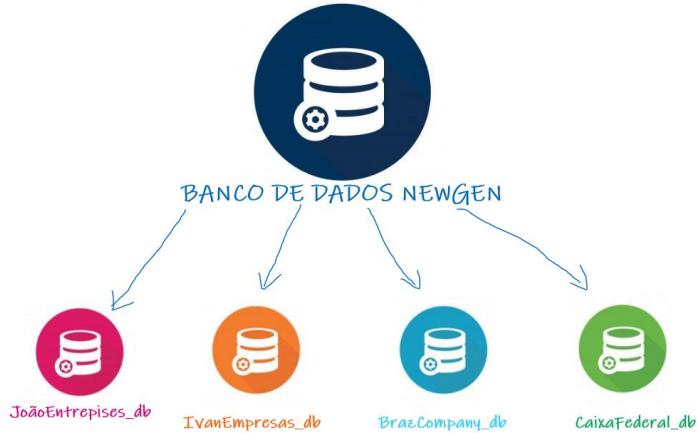
Tabela empresas				
id_coworking	nome da empresa	configuracoes das pags	...	dados do seu coworking
1	JoãoEnterprises	{'agendamento': 'digital'}		TABELAS COWORKING1
2	IvanEmpresas	{'agendamento': 'digital'}		TABELAS COWORKING2
3	BrazCompany	{'agendamento': 'manual'}		TABELAS COWORKING3
4	CaixaFederal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

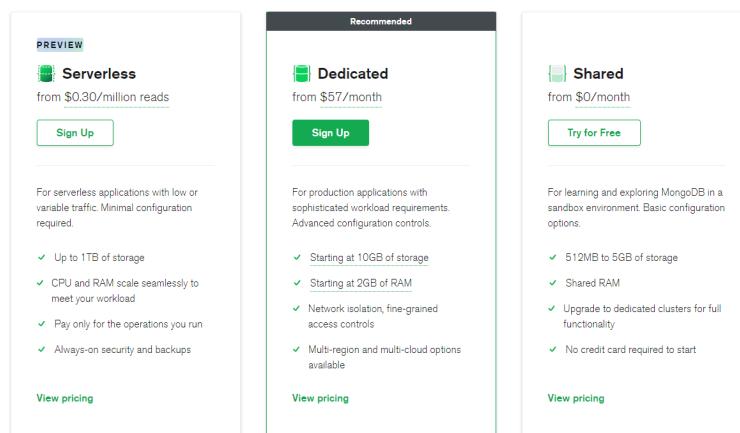
- Ganho de otimização
- Mais profissional que a solução 1

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>