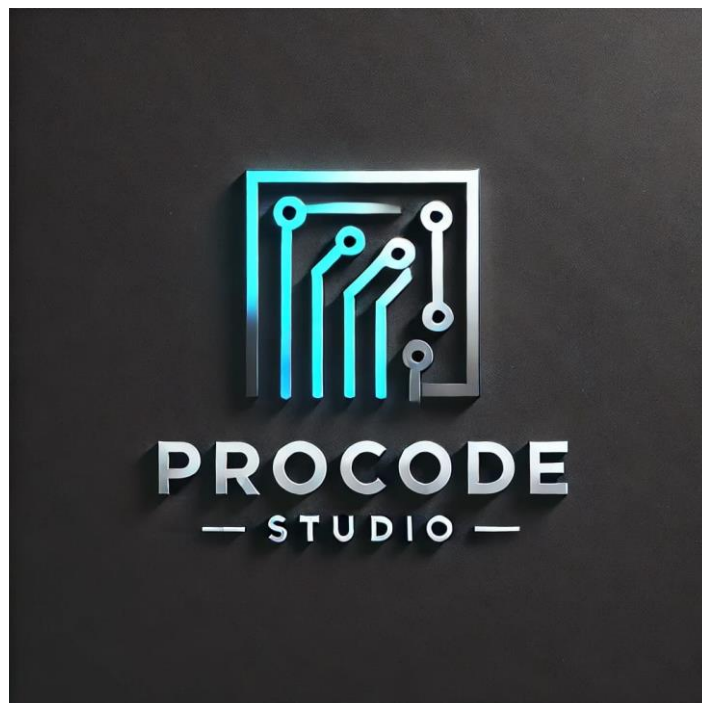


UNINOVE CAMPUS SANTO AMARO

KAIKY BRAGA NOGUEIRA



TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ORIENTADOR: PROFº FELIPE SANTOS DE JESUS

TURMA 50

SÃO PAULO - SP

2024

## SUMÁRIO

APRENDIZADO DE MÁQUINA (ALGORITMOS DE ML) .....	3
Exploração de Dados e Pré-processamento .....	3
Implementação de Modelos de Aprendizado de Máquina .....	5
Otimização e Validação do Modelo .....	8
Modelagem Conceitual .....	10
Modelagem Lógica e Física .....	11
Dicionário de Dados uma Simulação de Cadastro .....	12
REDES DE COMPUTADORES .....	13
Planta baixa de Rede da Empresa .....	13
Configuração de IP de Todos os Equipamentos .....	14
No arquivo abaixo você pode conferir a planta e toda a infraestrutura de rede: .....	16
SEGURANÇA DA INFORMAÇÃO .....	17
Análise de Riscos .....	17
Implementação de Medidas de Segurança .....	19
PROCEDURE STUDIO – CONSULTORIA E DESENVOLVIMENTO .....	21
Serviços Oferecidos Pela Empresa .....	21
Descrição Do Projeto .....	21
Escopo Do Projeto .....	21
Plano De Negócios .....	21
Tecnologias Utilizadas .....	22
LINK GITHUB .....	22

# APRENDIZADO DE MÁQUINA (ALGORITMOS DE ML)

## Exploração de Dados e Pré-processamento

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_squared_error, accuracy_score, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

# Expansão dos Dados
np.random.seed(42)
projetos = np.random.choice(["Pequenos", "Intermediários", "Complexos"], size=50)
quantidade_contratos = np.random.randint(1, 10, size=50)
valor_projeto = np.random.choice([10000, 70000, 200000], size=50)
lucro_mensal = quantidade_contratos * valor_projeto

# DataFrame
data = pd.DataFrame({
    "TipoProjeto": projetos,
    "QuantidadeContratos": quantidade_contratos,
    "ValorMedio": valor_projeto,
    "LucroMensal": lucro_mensal
})

# Visualização inicial
sns.scatterplot(data=data, x="QuantidadeContratos", y="LucroMensal", hue="TipoProjeto")
plt.title("Distribuição de Lucro por Contratos e Tipo de Projeto")
plt.show()

# Preparo dos dados
X = data[["QuantidadeContratos", "ValorMedio"]]
y_reg = data["LucroMensal"]
y_clf = data["TipoProjeto"]

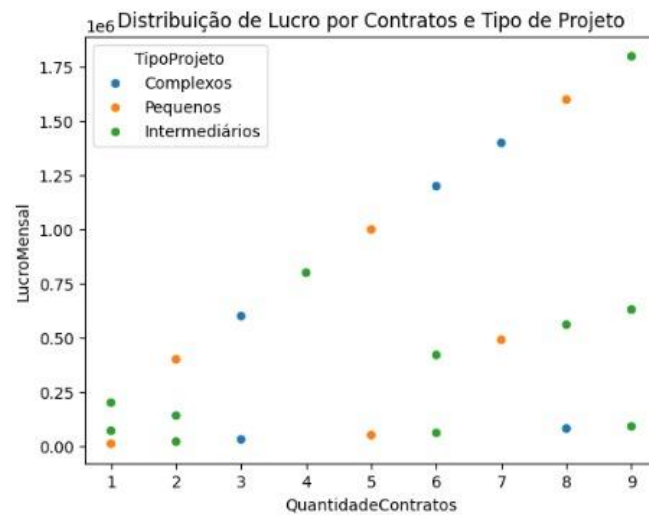
# Divisão entre treino e teste (mantendo as classes balanceadas)
X_train, X_test, y_reg_train, y_reg_test, y_clf_train, y_clf_test = train_test_split(
    X, y_reg, y_clf, test_size=0.3, random_state=42, stratify=y_clf
)

# 1. Regressão Linear
reg_model = LinearRegression()
reg_model.fit(X_train, y_reg_train)
y_reg_pred = reg_model.predict(X_test)

# Avaliação da regressão
mse = mean_squared_error(y_reg_test, y_reg_pred)
print(f"Erro quadrático médio (MSE): {mse:.2f}")

# 2. Classificação com Random Forest
clf_model = RandomForestClassifier(random_state=42)
clf_model.fit(X_train, y_clf_train)
y_clf_pred = clf_model.predict(X_test)

# Avaliação da classificação
acc = accuracy_score(y_clf_test, y_clf_pred)
print(f"Acurácia da classificação: {acc:.2f}")
print("Relatório de Classificação:")
print(classification_report(y_clf_test, y_clf_pred))
```



Erro quadrático médio (MSE): 40212925364.07

Acurácia da classificação: 0.47

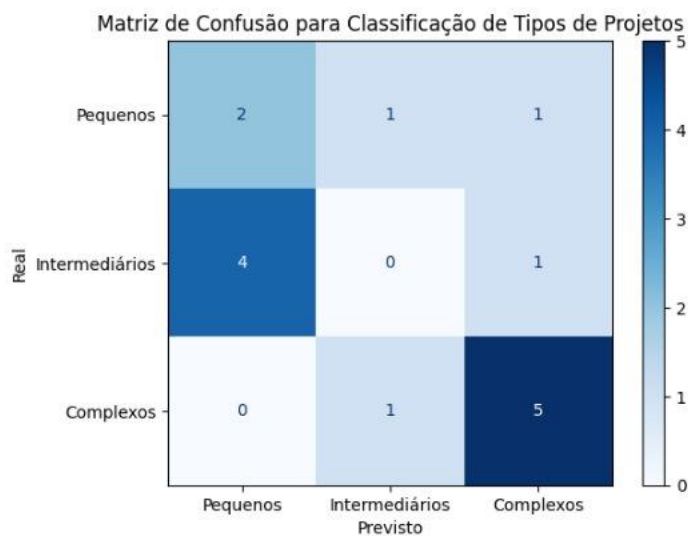
Relatório de Classificação:

	precision	recall	f1-score	support
Complexos	0.71	0.83	0.77	6
Intermediários	0.00	0.00	0.00	5
Pequenos	0.33	0.50	0.40	4
accuracy			0.47	15
macro avg	0.35	0.44	0.39	15
weighted avg	0.37	0.47	0.41	15

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Matriz de confusão
labels = ["Pequenos", "Intermediários", "Complexos"]
matriz_confusao = confusion_matrix(y_clf_test, y_clf_pred, labels=labels)

# Exibir matriz de confusão
disp = ConfusionMatrixDisplay(confusion_matrix=matriz_confusao, display_labels=labels)
disp.plot(cmap='Blues')
plt.title("Matriz de Confusão para Classificação de Tipos de Projetos")
plt.xlabel("Previsto")
plt.ylabel("Real")
plt.show()
```



## Implementação de Modelos de Aprendizado de Máquina

```
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV, cross_val_predict
from sklearn.metrics import mean_squared_error, classification_report, confusion_matrix, ConfusionMatrixDisplay
from sklearn.utils import resample
import matplotlib.pyplot as plt
import seaborn as sns

# Geração de Dados Balanceados
np.random.seed(42)
projetos = np.random.choice(["Pequenos", "Intermediários", "Complexos"], size=100)
quantidade_contratos = np.random.randint(1, 15, size=100)
valor_projeto = np.random.choice([10000, 70000, 200000], size=100)
lucro_mensal = quantidade_contratos * valor_projeto

# Criando o DataFrame
data = pd.DataFrame({
    "TipoProjeto": projetos,
    "QuantidadeContratos": quantidade_contratos,
    "ValorMedio": valor_projeto,
    "LucroMensal": lucro_mensal
})

# Balanceamento do dataset
data_balanced = pd.concat([
    resample(data[data["TipoProjeto"] == "Pequenos"], replace=True, n_samples=40, random_state=42),
    resample(data[data["TipoProjeto"] == "Intermediários"], replace=True, n_samples=40, random_state=42),
    resample(data[data["TipoProjeto"] == "Complexos"], replace=True, n_samples=40, random_state=42)
])

# Preparo dos dados
X = data_balanced[["QuantidadeContratos", "ValorMedio"]]
y_clf = data_balanced["TipoProjeto"]
y_reg = data_balanced["LucroMensal"]

# Otimização do modelo de classificação com Random Forest
param_grid = {
    "n_estimators": [10, 50, 100],
    "max_depth": [None, 5, 10],
    "min_samples_split": [2, 5],
    "min_samples_leaf": [1, 2]
}
clf = RandomForestClassifier(random_state=42)
grid_clf = GridSearchCV(clf, param_grid, cv=3, scoring="accuracy") # cv reduzido para evitar erros
grid_clf.fit(X, y_clf)

# Melhor modelo de classificação
best_clf = grid_clf.best_estimator_
print(f"Melhores parâmetros para classificação: {grid_clf.best_params_}")

# Avaliação com validação cruzada
y_clf_pred = cross_val_predict(best_clf, X, y_clf, cv=3)

# Relatório de classificação
print("Relatório de Classificação (Modelo Otimizado):")
print(classification_report(y_clf, y_clf_pred, zero_division=0))

# Matriz de confusão
cm = confusion_matrix(y_clf, y_clf_pred, labels=data_balanced["TipoProjeto"].unique())
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=data_balanced["TipoProjeto"].unique())
disp.plot(cmap="viridis")
plt.title("Matriz de Confusão")
plt.show()
```

```

# Otimização do modelo de regressão com Ridge
param_grid_reg = {"alpha": [0.01, 0.1, 1.0, 10.0]}
ridge = Ridge()
grid_reg = GridSearchCV(ridge, param_grid_reg, cv=3, scoring="neg_mean_squared_error")
grid_reg.fit(X, y_reg)

# Melhor modelo de regressão
best_reg = grid_reg.best_estimator_
print(f"Melhores parâmetros para regressão: {grid_reg.best_params_}")

# Avaliação com validação cruzada
y_reg_pred = cross_val_predict(best_reg, X, y_reg, cv=3)
mse = mean_squared_error(y_reg, y_reg_pred)
print(f"Erro Quadrático Médio (MSE) após otimização: {mse:.2f}")

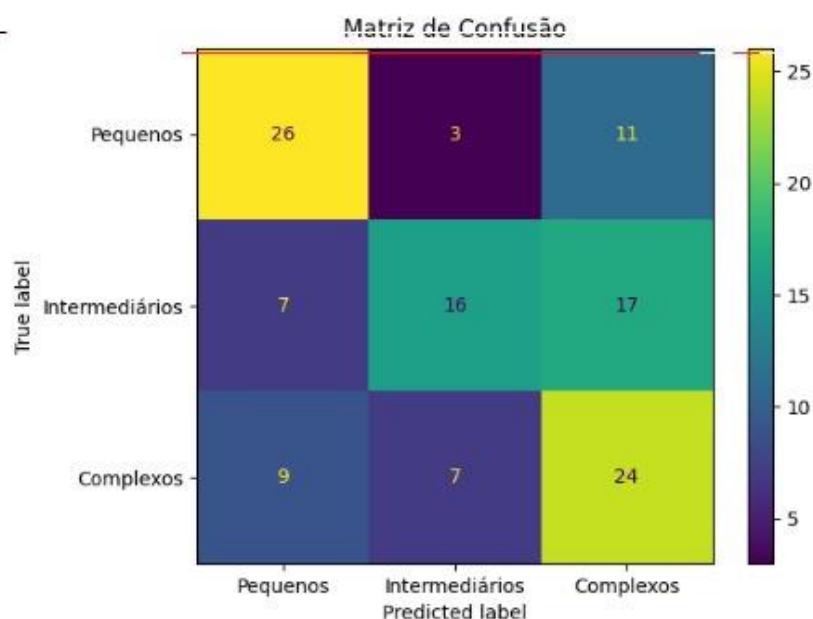
# Gráfico de predições vs valores reais
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_reg, y=y_reg_pred, color="blue")
plt.plot([min(y_reg), max(y_reg)], [min(y_reg), max(y_reg)], color="red", linestyle="--", linewidth=2)
plt.title("Predições vs Valores Reais (Regressão Ridge)", fontsize=16)
plt.xlabel("Valores Reais", fontsize=12)
plt.ylabel("Predições", fontsize=12)
plt.tight_layout()
plt.show()

```

Melhores parâmetros para classificação: {'max\_depth': None, 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 50}

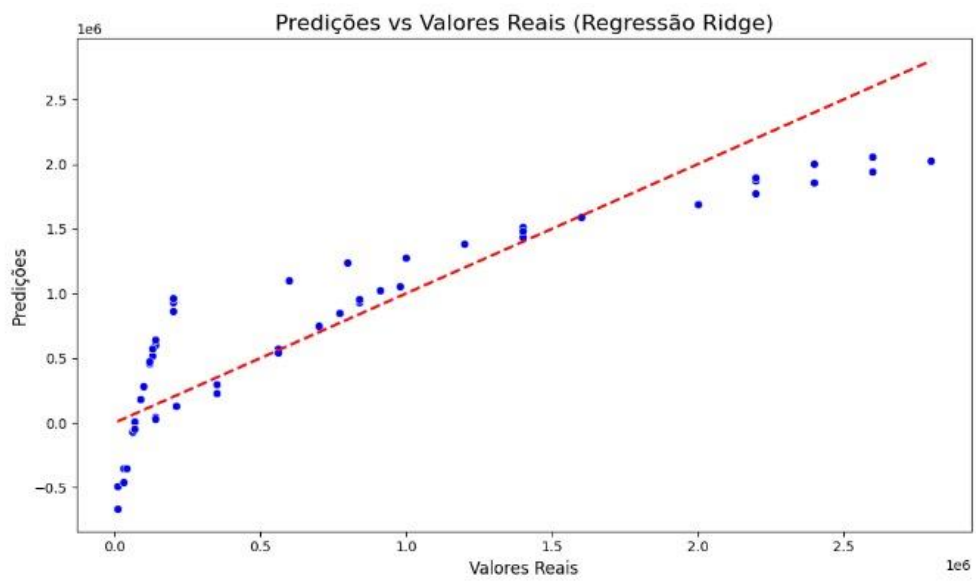
Relatório de Classificação (Modelo Otimizado):

	precision	recall	f1-score	support
Complexos	0.46	0.60	0.52	40
Intermediários	0.62	0.40	0.48	40
Pequenos	0.62	0.65	0.63	40
accuracy			0.55	120
macro avg	0.57	0.55	0.55	120
weighted avg	0.57	0.55	0.55	120



Melhores parâmetros para regressão: {'alpha': 0.01}

Erro Quadrático Médio (MSE) após otimização: 121856816496.37





## Otimização e Validação do Modelo

```
J: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Geração de dados balanceados para "projetos"
np.random.seed(42)
projetos = np.random.choice(["Pequenos", "Intermediários", "Complexos"], size=100)
quantidade_contratos = np.random.randint(1, 15, size=100) # Entre 1 e 15 contratos por mês
valor_projeto = np.random.choice([10000, 70000, 200000], size=100) # Valores típicos para os tipos de projeto
lucro_mensal = quantidade_contratos * valor_projeto

# Criando o DataFrame
data = pd.DataFrame({
    "TipoProjeto": projetos,
    "QuantidadeContratos": quantidade_contratos,
    "ValorMedio": valor_projeto,
    "LucroMensal": lucro_mensal
})

# Resumo do dataset balanceado
print(data.head())

# Análise agregada
summary = data.groupby("TipoProjeto").agg({
    "QuantidadeContratos": "mean",
    "LucroMensal": "mean"
}).reset_index()

# Preparação para o gráfico
tipos_projetos = summary["TipoProjeto"]
quantidade_media_contratos = summary["QuantidadeContratos"]
lucro_medio_mensal = summary["LucroMensal"]

# Visualização
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

axes[0].plot(tipos_projetos, quantidade_media_contratos, marker='o', color='blue', label="Clientes")
axes[0].set_title("Quantidade Média de Clientes por Tipo de Projeto")
axes[0].set_xlabel("Tipo de Projeto")
axes[0].set_ylabel("Quantidade Média de Clientes")
axes[0].grid(True)
axes[0].legend()

axes[1].plot(tipos_projetos, lucro_medio_mensal, marker='o', color='green', label="Lucro Mensal")
axes[1].set_title("Lucro Médio Mensal por Tipo de Projeto")
axes[1].set_xlabel("Tipo de Projeto")
axes[1].set_ylabel("Lucro Médio (R$)")
axes[1].grid(True)
axes[1].legend()

plt.tight_layout()
plt.show()
```

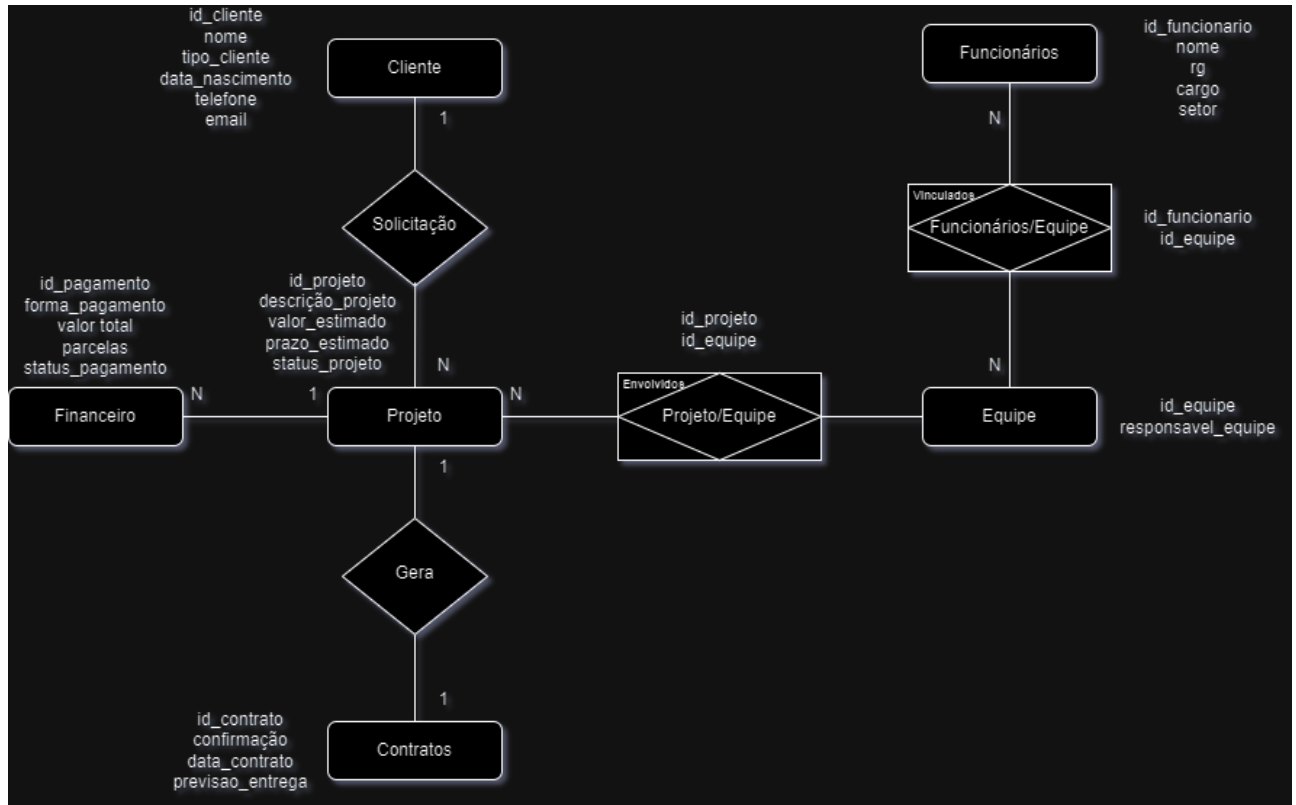
	TipoProjeto	QuantidadeContratos	ValorMedio	LucroMensal
0	Complexos	12	10000	120000
1	Pequenos	8	200000	1600000
2	Complexos	8	200000	1600000
3	Complexos	11	70000	770000
4	Pequenos	3	200000	600000



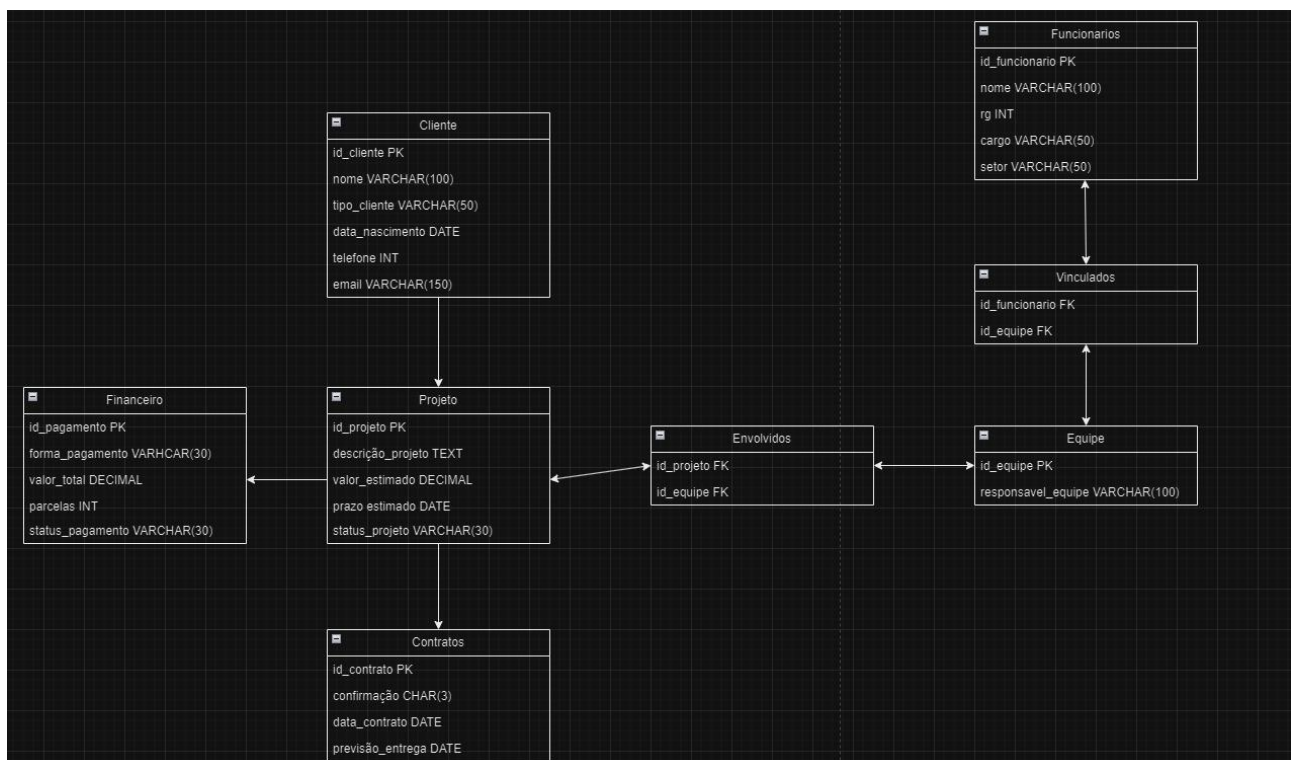
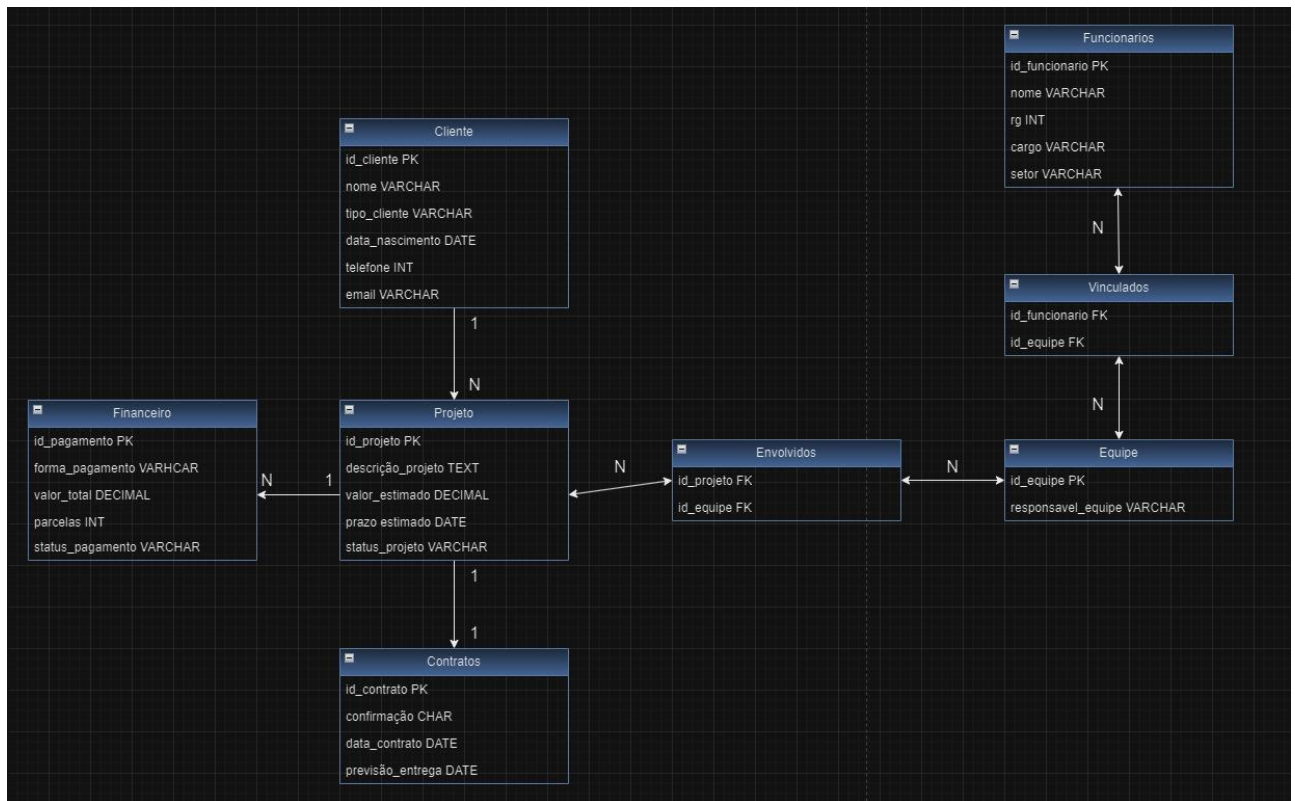


# MODELAGEM DE DADOS

## Modelagem Conceitual




## Modelagem Lógica e Física



## Dicionário de Dados uma Simulação de Cadastro

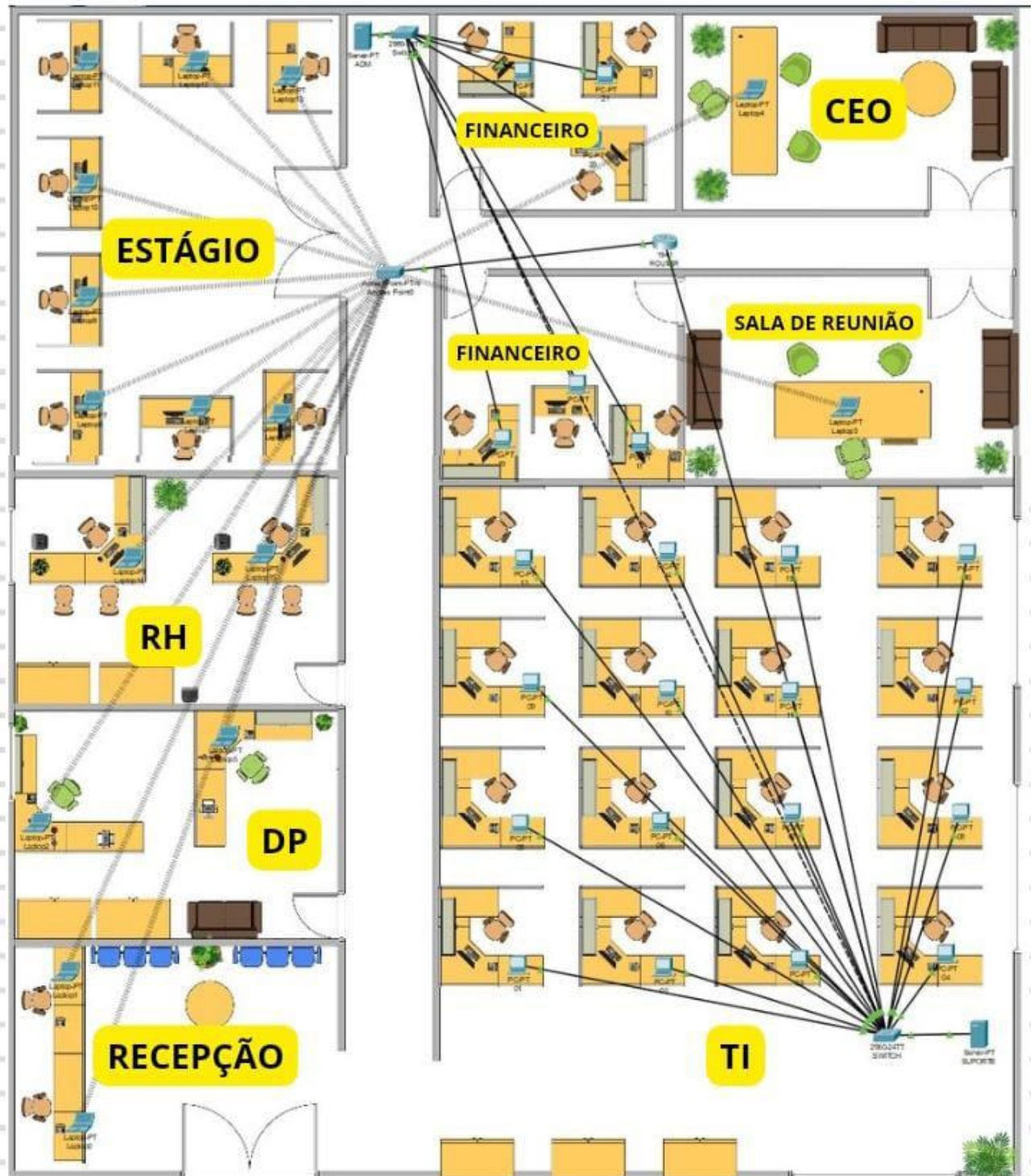
/

SIMULAÇÃO DE CADASTRO			
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_cliente	PRIMARY KEY	Identificação do cliente	PK
nome	VARCHAR	Nome do cliente	
tipo_cliente	VARCHAR	Tipo de cliente (físico ou jurídico)	
data_nascimento	DATE	Data de nascimento	
telefone	INT	Telefone	
email	VARCHAR	Email	
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_projeto	PRIMARY KEY	Identificação do projeto	PK
descricao_projeto	TEXT	Descrição do projeto	
valor_estimado	DECIMAL	Valor estimado do projeto	
prazo_estimado	DATE	Prazo de entrega do projeto	
status_projeto	VARCHAR	Status andamento do projeto	
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_pagamento	PRIMARY KEY	Identificação do pagamento	PK
forma_pagamento	VARCHAR	Forma de pagamento	
valor_total	DECIMAL	Valor total do projeto	
parcelas	INT	Quantas vezes foi parcelado	
status_pagamento	VARCHAR	Status andamento do pagamento	
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_contrato	PRIMARY KEY	Identificação do contrato	PK
confirmação	CHAR	Contrato em aberto ou concluído	
data_contrato	DATE	Data do fechamento do contrato	
previsão_entrega	DATE	Previsão de entrega do projeto	
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_projeto	FOREIGN KEY	Identificação do projeto	FK
id_equipe	FOREIGN KEY	Identificação da equipe	FK

ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_equipe	PRIMARY KEY	Identificação da equipe	PK
responsavel_equipe	VARCHAR	Responsável pela equipe	
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_funcionário	FOREIGN KEY	Identificação do funcionário	FK
id_equipe	FOREIGN KEY	Identificação da equipe	FK
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_funcionário	PRIMARY KEY	Identificação do funcionário	PK
nome	VARCHAR	Nome do funcionário	
rg	INT	Rg do funcionário	
cargo	VARCHAR	Cargo do funcionário	
setor	VARCHAR	Setor do funcionário	

## REDES DE COMPUTADORES

### Planta baixa de Rede da Empresa



## Configuração de IP de Todos os Equipamentos

Nossa empresa tem um escritório de aproximadamente 450m<sup>2</sup>, com 25m de comprimento e 18m de largura. Pensando na estrutura do local, foram instalados no escritório 22 computadores, 1 roteador, 2 switches, 1 access point, 1 servidor ADM, 1 servidor SUPORTE e 16 notebooks.

SERVIDOR SUPORTE						
Setor	Dispositivo	Porta	IP	Rede	Broadcast	Máscara e CIDR
TI	PC1	1	192.168.0.129	128 á 254	192.168.0.255	255.255.255.0
TI	PC2	2	192.168.0.130			255.255.255.0
TI	PC3	3	192.168.0.131			255.255.255.0
TI	PC4	4	192.168.0.132			255.255.255.0
TI	PC5	5	192.168.0.133			255.255.255.0
TI	PC6	6	192.168.0.134			255.255.255.0
TI	PC7	7	192.168.0.135			255.255.255.0
TI	PC8	8	192.168.0.136			255.255.255.0
TI	PC9	9	192.168.0.137			255.255.255.0
TI	PC10	10	192.168.0.138			255.255.255.0
TI	PC11	11	192.168.0.139			255.255.255.0
TI	PC12	12	192.168.0.140			255.255.255.0
TI	PC13	13	192.168.0.141			255.255.255.0
TI	PC14	14	192.168.0.142			255.255.255.0
TI	PC15	15	192.168.0.143			255.255.255.0
TI	PC16	16	192.168.0.144			255.255.255.0
SERVIDOR ADM						
Setor	Dispositivo	Porta	IP	Rede	Broadcast	Máscara e CIDR
FINANCEIRO	PC17	1	192.168.0.2	0 á 127	192.168.0.128	255.255.255.0
FINANCEIRO	PC18	2	192.168.0.3			255.255.255.0
FINANCEIRO	PC19	3	192.168.0.4			255.255.255.0
FINANCEIRO	PC20	4	192.168.0.5			255.255.255.0
FINANCEIRO	PC21	5	192.168.0.6			255.255.255.0
FINANCEIRO	PC22	6	192.168.0.7			255.255.255.0



ADM					
Primeiro válido	Último válido	Broadcast			
192.168.0.1	192.168.0.126	.127			
SUPORTE					
Primeiro válido	Último válido	Broadcast			
192.168.0.129	192.168.0.254	.255			
ProCode Studio					
Dispositivo	Porta	IP	Rede	Broadcast	Máscara e CIDR
Switch ADM	Gig0/1				
Switch Suporte	Gig0/2				
Access-point	Port1				
Roteador	Gig0/1; Gig0/0	192.168.0.1			255.255.255.0
Server ADM	Fa0/24	192.168.0.2			255.255.255.128
Server SUPORTE	Fa0/24	192.168.0.129			255.255.255.254
NOTEBOOKS CONECTADOS AO ACCESS POINT					
Dispositivo	Setor	IP	MAC		
Notebook 0	RECEPÇÃO	168.254.28.93	0090.213E.1C5A		
Notebook 1	RECEPÇÃO	168.254.24.33	142.081.820		
Notebook 2	DP	168.254.148.13	0060.7045.942B		
Notebook 3	SALA DE REUNIÃO	192.168.20.24	0004.9A58.B09D		
Notebook 4	CEO	192.168.20.25	00D0.BA8E.9087		
Notebook 5	DP	168.254.238.135	0001.9608.EE87		
Notebook 6	ESTÁGIO	168.254.98.124	0001.6462.627B		
Notebook 7	ESTÁGIO	168.254.13.71	0040.0BC8.0D47		
Notebook 8	ESTÁGIO	168.254.228.237	000D.BD6E.E4EC		
Notebook 9	ESTÁGIO	168.254.81.152	0003.E4BE.5198		
Notebook 10	ESTÁGIO	168.254.44.45	0003.E441.2C2D		
Notebook 11	ESTÁGIO	168.254.201.21	0001.63E7.C915		
Notebook 12	ESTÁGIO	168.254.96.4	00D0.BAE7.6004		
Notebook 13	ESTÁGIO	192.168.20.26	000C.CFA4.56BE		
Notebook 14	RH	168.254.43.25	0009.7CB6.2B19		
Notebook 15	RH	168.254.154.118	00D0.586D.9A76		

O documento apresenta a configuração da rede da nossa empresa, que foi dividida em duas sub-redes: ADM e SUPORTE. A sub-rede ADM utiliza a faixa de IPs de 192.168.0.1 a 192.168.0.126, com o broadcast em 192.168.0.127 e máscara 255.255.255.128 (/25). Já a sub-rede SUPORTE ocupa a faixa de 192.168.0.129 a 192.168.0.254, com o broadcast em 192.168.0.255 e a mesma máscara. Os PCs da sub-rede SUPORTE estão identificados de 1 a 16, com IPs consecutivos entre 192.168.0.129 e 192.168.0.144, enquanto na sub-rede ADM estão os PCs 17 a 22 com IPs 192.168.0.2 a 192.168.0.7. Além disso, há servidores e switches configurados com IPs e portas específicos.

A rede conta com um roteador conectado às duas sub-redes, utilizando o IP 192.168.0.1 na interface Gig0/0 (sub-rede ADM). A divisão em sub-redes melhora a organização e o controle de tráfego, separando dispositivos administrativos e de suporte. A máscara 255.255.255.128 foi usada para dividir a rede principal em dois blocos iguais, facilitando o gerenciamento. Essa configuração segue boas práticas, garantindo eficiência e evitando problemas como conflitos de IPs.



Os servidores da rede foram configurados para oferecer serviços essenciais, como DHCP, DNS, HTTP e FTP, garantindo a distribuição automática de IPs, resolução de nomes, hospedagem de páginas web e transferência de arquivos.

Em nossa rede, a segmentação foi realizada por meio de VLANs, separando as áreas de ADM e SUPORTE. A VLAN 10 foi atribuída à parte administrativa (ADM), a VLAN 20 ficou destinada à parte de suporte (SUPORTE). Essa divisão em VLANs proporciona maior organização, segmentação de tráfego e segurança, separando os dispositivos de cada setor e otimizando o desempenho da rede.

Para a conectividade sem fio, utilizamos 1 Access Points. O Access Point foi configurado com o SSID "Escritório 5" e a senha "123456789", ao qual conectamos os notebooks. Essa configuração permitiu a segmentação adequada dos dispositivos móveis e garantiu uma conexão estável e segura para os usuários do local.

A escolha da classe C foi ideal para essa configuração, pois atende às necessidades de uma organização de pequeno a médio porte de forma eficiente e organizada. Com capacidade para até 254 endereços utilizáveis, a divisão em sub-redes com máscara /25 permitiu separar os dispositivos em setores distintos (ADM e SUPORTE), garantindo clareza na estrutura e espaço para expansão futura.

A classe C simplifica o gerenciamento da rede, tornando a configuração mais prática e reduzindo a complexidade em comparação com classes maiores, como A ou B, que ofereceriam mais endereços do que o necessário e gerariam desperdício de recursos. Além disso, a segmentação por sub-redes e VLANs reforça a segurança e otimiza o controle do tráfego, facilitando a organização e o isolamento entre setores.

Com essa infraestrutura, garantimos a separação eficiente dos fluxos de dados entre os diferentes stands, a segurança dos dispositivos conectados e a possibilidade de expansão futura.

No arquivo abaixo você pode conferir a planta e toda a infraestrutura de rede:



OBS: basta dar 2 cliques no item ao lado.

## SEGURANÇA DA INFORMAÇÃO

### Análise de Riscos

- Riscos de mercado: A concorrência acirrada, as constantes mudanças tecnológicas e a instabilidade econômica são fatores que podem impactar diretamente no nosso negócio. Para mitigar esses riscos, é essencial estarmos acompanhando esse nicho de mercado, investir em inovação e marketing digital, além de ter um planejamento financeiro sólido

- Riscos operacionais: Erros no desenvolvimento de sistemas ou atrasos podem comprometer a entrega dos projetos, impactando a confiança e satisfação dos clientes. Além disso, a perda de profissionais-chave pode afetar a continuidade e qualidade do trabalho, especialmente em projetos complexos. Problemas de comunicação interna também representam um risco, já que mal-entendidos podem gerar retrabalho, desperdício de recursos e atrasos adicionais.

- Riscos financeiros: Projetos sob demanda podem gerar receitas variáveis, dificultando a previsão de caixa e investimentos, especialmente nos estágios iniciais do nosso negócio. Custos elevados para aquisição de ferramentas, licenças e treinamentos podem impactar a margem de lucro. Embora investir continuamente seja essencial para manter a competitividade, realizar grandes saídas financeiras sem retornos rápidos pode comprometer a saúde financeira. Além de existir a possibilidade de pagamentos indevidos ou erros no gerenciamento financeiro, o que pode resultar em prejuízos significativos para nossa empresa.

- Riscos regulatórios: O descumprimento de leis, como a LGPD (Lei Geral de Proteção de Dados) e regulamentações trabalhistas, pode expor nossa empresa a multas, sanções e ações judiciais, afetando tanto a saúde financeira quanto a reputação. Além disso, em plataformas desenvolvidas para clientes, pode ocorrer de sermos responsabilizados caso os sistemas não estejam em conformidade com normas de proteção de dados ou acessibilidade.

- Riscos de reputação: Erros em projetos ou problemas na comunicação com os clientes podem prejudicar a imagem da empresa. Ademais, uma violação grave de segurança que exponha dados sensíveis pode causar danos irreparáveis à credibilidade, dificultando a confiança de novos e antigos clientes.

- Riscos de segurança: A proteção de dados é fundamental nesse setor. Ataques cibernéticos e vazamentos de informações podem causar danos irreparáveis a nossa reputação. Para se proteger, implementaremos medidas de segurança robustas, como firewalls, antivírus e criptografia, além de criar uma política de segurança da informação e treinar seus colaboradores.

- Riscos de saúde e segurança: O ritmo acelerado do desenvolvimento de software pode levar ao burnout e a doenças ocupacionais como a LER. Para prevenir esses problemas, promoveremos a saúde e o bem-estar dos nossos colaboradores, oferecendo um ambiente de trabalho ergonômico e incentivando a prática de atividades físicas.

- Riscos tecnológicos: Falhas de software e dependência de terceiros são desafios comuns no desenvolvimento de software. Para mitigá-los, realizaremos testes rigorosos antes de lançar um produto, mantendo sempre um plano de contingência para lidar com falhas e problemas que possam surgir no caminho.

- Riscos ambientais: O desenvolvimento de software também tem um impacto ambiental. Para minimizar seus efeitos, procuraremos investir em práticas sustentáveis, como o uso de energia renovável e a reciclagem de equipamentos eletrônicos.

- Riscos de gestão: A falta de experiência em gestão e a dificuldade em contratar e reter talentos são desafios comuns para startups como a nossa. Para superá-los, buscaremos sempre a orientação de pessoas do ramo, estaremos sempre dispostos a networkings, com o objetivo de criar uma cultura organizacional forte e atrativa.

## Implementação de Medidas de Segurança

- **Treinamento em segurança:** O treinamento contínuo da equipe é essencial para garantir que todos os colaboradores estejam preparados para identificar e responder a ameaças. Isso inclui treinar os funcionários para reconhecer tentativas de phishing, ataques de engenharia social e práticas de segurança digital, como o uso de senhas fortes e a proteção de dados sensíveis. A conscientização de segurança deve ser parte integrante do processo de integração de novos funcionários e de atualizações regulares.

- **Instalações seguras:** A segurança física das instalações onde a nossa empresa opera é fundamental. Isso envolve garantir que servidores, equipamentos e áreas críticas sejam protegidos contra acessos não autorizados. É importante termos um controle rigoroso de entrada e saída de pessoas, com o uso de sistemas de segurança como biometria, catracas e câmeras de monitoramento. Além disso, procuraremos garantir que o ambiente tenha proteção contra incêndios, falhas elétricas e outros desastres que possam comprometer a infraestrutura da empresa.

- **Controle de acesso:** Dentro da empresa, o controle de acesso digital e físico deve ser rigoroso. Para o ambiente digital, implementaremos sistemas de autenticação multifatorial (MFA) e controle de acesso baseado em funções (RBAC), garantindo que apenas os colaboradores com as permissões adequadas tenham acesso a informações sensíveis. Em termos físicos, nós iremos adotar sistemas de segurança como cartões de acesso e monitoramento eletrônico para limitar o acesso a áreas restritas e sensíveis, como servidores ou documentos confidenciais.

- **Procedimentos de emergência:** É importante termos um plano de emergência bem estruturado para lidar com situações inesperadas, como falhas de sistema, ataques cibernéticos ou desastres naturais. Esse plano deve incluir

ações claras sobre como a equipe deve reagir em diferentes cenários, quem são os responsáveis pela comunicação e a recuperação dos dados e como as operações críticas serão restabelecidas. Testes regulares e simulações de emergências são necessários para garantir que todos saibam o que fazer em momentos de crise.

- **Limpeza e saneamento:** Além da segurança digital, a limpeza e o saneamento básico também são importantes. Isso envolve garantir que os ambientes de trabalho estejam limpos, organizados e livres de objetos ou materiais que possam representar riscos, como papéis com informações sensíveis. É fundamental adotarmos políticas para o descarte adequado de documentos e dispositivos antigos, com a destruição segura de dados de dispositivos eletrônicos e a eliminação de papéis com informações confidenciais

- **Monitoramento de câmeras de segurança:** As câmeras devem ser instaladas em pontos estratégicos, como entradas, corredores e áreas de servidores, para garantir a segurança física. Além disso, o acesso às imagens de segurança deve ser restrito a pessoas autorizadas e as imagens devem ser armazenadas de maneira segura.

- **Seguro adequado:** Ter um seguro adequado é uma medida essencial para proteger nossa empresa contra danos financeiros decorrentes de incidentes de segurança, como invasões de dados, danos materiais ou responsabilidades civis. O seguro de responsabilidade cibernética, por exemplo, pode cobrir custos relacionados a vazamentos de dados ou ataques cibernéticos, enquanto o seguro de propriedade pode proteger contra danos físicos às instalações ou equipamentos.

## PROCEDURE STUDIO – CONSULTORIA E DESENVOLVIMENTO

### Serviços Oferecidos Pela Empresa

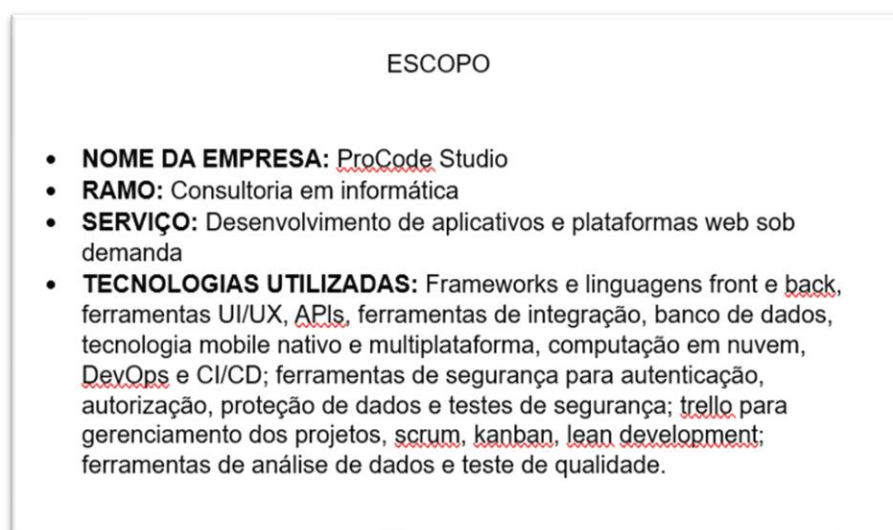
Consultoria e Planejamento, Desenvolvimento de Aplicativos, Desenvolvimento de Plataformas Web, UX/UI Design, Manutenção e Suporte, Tecnologias Avançadas, Gerenciamento de Projetos, Testes e Qualidade, Publicação e Lançamento, Personalização e Escalabilidade.

### Descrição Do Projeto

A ProCode Studio é uma startup especializada no desenvolvimento de aplicativos móveis, plataformas web e soluções tecnológicas sob medida. Com foco na inovação e na experiência do usuário, oferecemos serviços personalizados para transformar ideias em produtos digitais de alto impacto.

Nossa equipe combina expertise técnica, criatividade e metodologias ágeis para atender às necessidades de clientes de diversos setores. Desde a consultoria e planejamento inicial até a entrega final e suporte contínuo, trabalhamos para garantir eficiência, qualidade e escalabilidade em cada projeto.

### Escopo Do Projeto



### Plano De Negócios

O nosso plano de negócios inclui estratégias para alcançar os seguintes objetivos:

1. Expansão de Mercado: Identificar e atingir novos públicos-alvo, ampliando nossa presença no mercado digital.

2. Fidelização de Clientes: Desenvolver soluções de alta qualidade que atendam às necessidades específicas dos clientes, promovendo confiança e parcerias de longo prazo.
3. Inovação Tecnológica: Investir continuamente em tecnologias avançadas, como Inteligência Artificial, IoT e blockchain, para oferecer soluções modernas e competitivas.
4. Escalabilidade dos Produtos: Criar sistemas e plataformas robustas capazes de acompanhar o crescimento dos negócios dos nossos clientes.
5. Sustentabilidade Financeira: Maximizar a lucratividade através de uma gestão eficiente de recursos e estratégias de precificação.
6. Fortalecimento da Marca: Consolidar a ProCode Studio como referência em desenvolvimento de aplicativos e plataformas web sob demanda.
7. Aprimoramento Contínuo: Implementar práticas ágeis para garantir entregas rápidas, com qualidade e alinhadas às expectativas do mercado.
8. Suporte ao Cliente: Garantir uma experiência de atendimento completa e eficiente, do planejamento inicial à manutenção pós-lançamento.

## Tecnologias Utilizadas

Assim como citado no escopo, algumas tecnologias serão essenciais dentro da nossa empresa, como frameworks e linguagens front e back, ferramentas UI/UX, APIs, ferramentas de integração, banco de dados, tecnologia mobile nativo e multiplataforma, computação em nuvem, DevOps e CI/CD; ferramentas de segurança para autenticação, autorização, proteção de dados e testes de segurança; trello para gerenciamento dos projetos, scrum, kanban, lean development; ferramentas de análise de dados e teste de qualidade.

## LINK GITHUB

Neste link, você terá acesso ao nosso GitHub!