

PÓS-GRADUAÇÃO MIT ENGENHARIA DE SOFTWARE COM .NET

Projeto de Bloco: Engenharia de Software e Modelagem

Integrantes:

ALUNO: ALEXANDER SILVA

ALUNO: MARLON BRAGA

ALUNO: PEDRO NOVAES

MODELAGEM ESTRATÉGICA

Aplicar os conceitos de DDD

https://www.eduardopires.net.br/2016/08/ddd-nao-e-arquitetura-em-camadas

Qual problema vai resolver?; O que vai implementar?; Para que serve?; Quem vai atender? Quando implementar o DDD?



Aplicar os conceitos de DDD

Pontos Chave

Ubiquitous Language

Bounded Context

Context Map

Uma linguagem onipresente, universal de um negócio dentro da empresa, compartilhada por todas as partes envolvidas no projeto, ou seja, termos específicos do domínio.

É uma parte da empresa ou do domínio do negócio que possui elementos ou conceitos com significados bem definidos, com linguagem própria, arquitetura e implementação específica.

É um mapa que representa todos os contextos mapeados através dos elementos da linguagem ubíqua, bem como seus subdomínios e relacionamentos.

Aplicar os conceitos de DDD

Atenção!!!
Deixar de extrair a linguagem ubíqua é um dos piores erros.

Primeiro passo extrair a linguagem ubíqua



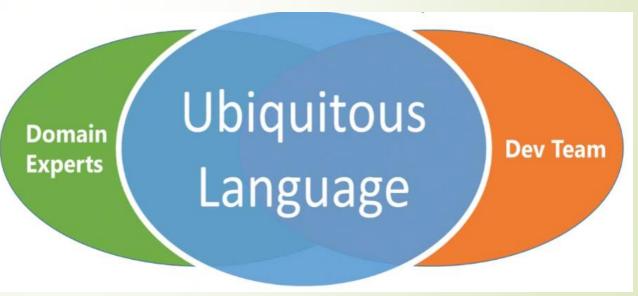
Domain Expert

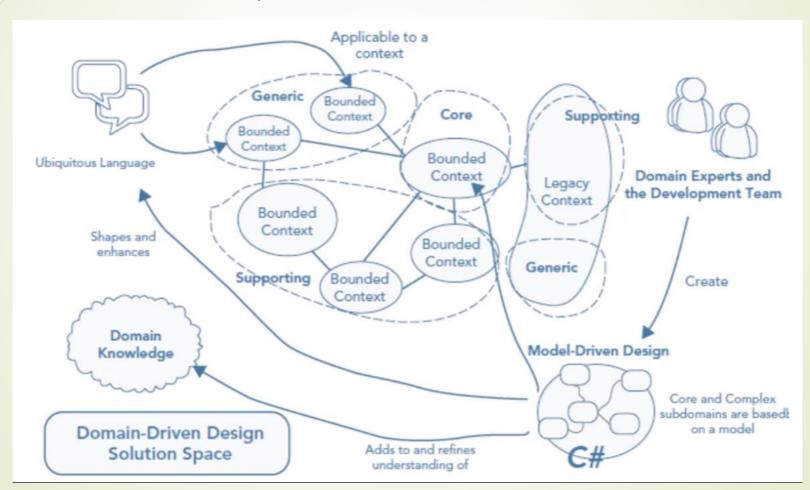
- Conhece do negócio, os processos e seus termos
- Define novos termos, processos e regras
- Tem conhecimento da operação



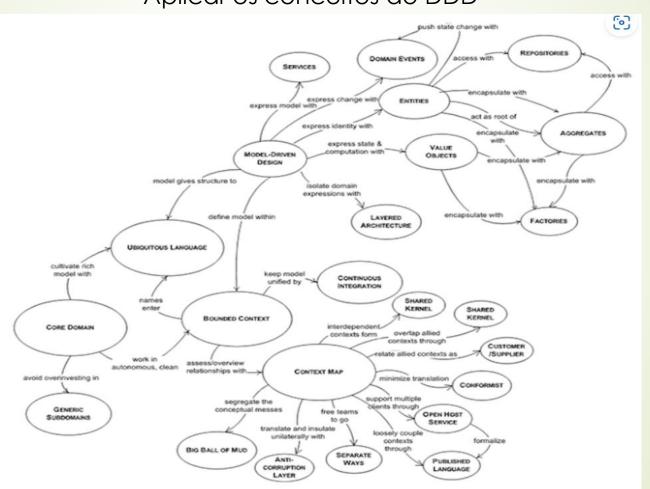
Time de Desenvolvimento

Precisa do apoio do Domain Expert para entender o processo, e as terminologias do negócio





Mapa de Aprendizado





- A loja virtual exibirá um catálogo de produtos de diversas categorias.
- Um cliente pode realizar um pedido contendo 1 ou N produtos.
- A loja realizará as vendas através de pagamento por cartão de crédito.
- O cliente irá realizar o seu cadastro para poder fazer pedidos.
- O cliente irá confirmar o pedido, endereço de entrega, escolher o tipo de frete e realizar o pagamento.
- Após o pagamento o pedido mudará de status conforme resposta da transação via cartão.
- Ocorrerá a emissão da nota fiscal logo após a confirmação de pagamento do pedido.



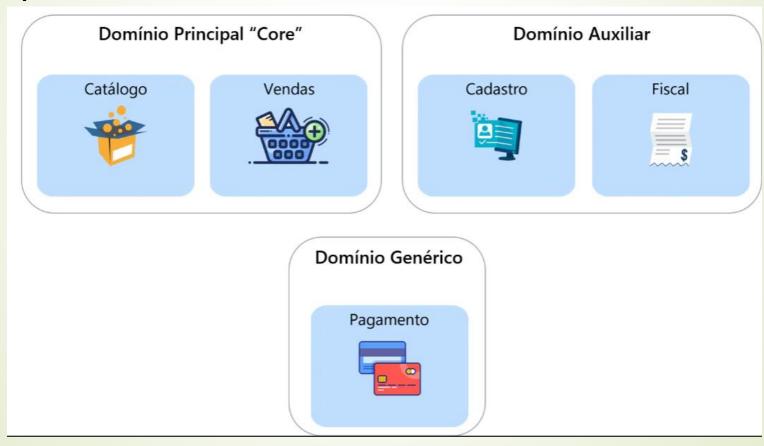
- A loja virtual exibirá um catálogo de produtos de diversas categorias.
- Um cliente pode realizar um pedido contendo 1 ou N produtos.
- A loja realizará as vendas através de pagamento por cartão de crédito.
- O cliente irá realizar o seu cadastro para poder fazer pedidos.
- O cliente irá confirmar o pedido, endereço de entrega, escolher o tipo de frete e realizar o pagamento.
- Após o pagamento o pedido mudará de status conforme resposta da transação via cartão.
- Ocorrerá a emissão da nota fiscal logo após a confirmação de pagamento do pedido.



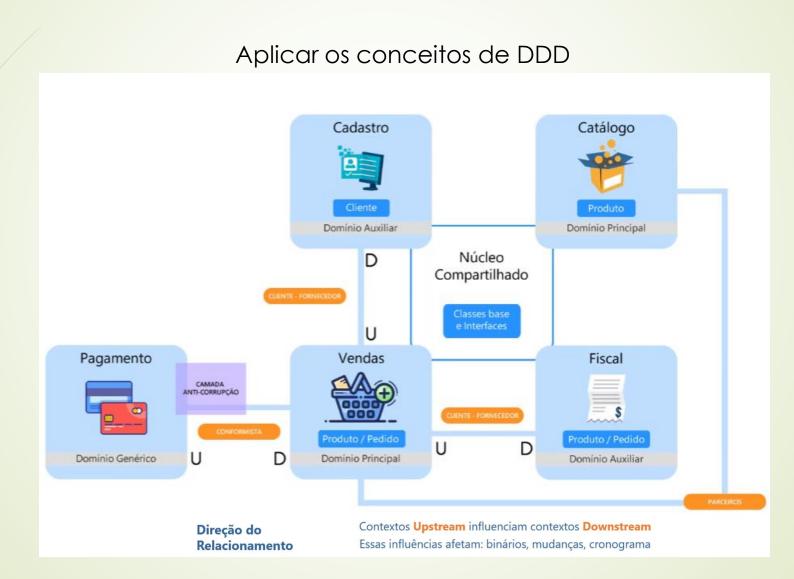
- A loja virtual exibirá um catálogo de produtos de diversas categorias.
- Um cliente pode realizar um pedido contendo 1 ou N produtos.
- A loja realizará as vendas através de pagamento por cartão de crédito.
- O cliente irá realizar o seu cadastro para poder fazer pedidos.
- O cliente irá confirmar o pedido endereço de entrega, escolher o tipo de frete e realizar o pagamento.
- Após o pagamento o pedido mudará de status conforme resposta da transação via cartão.
- Ocorrerá a emissão da nota fiscal logo após a confirmação de pagamento do pedido.

Aplicar os conceitos de DDD

Tipos de Domínios





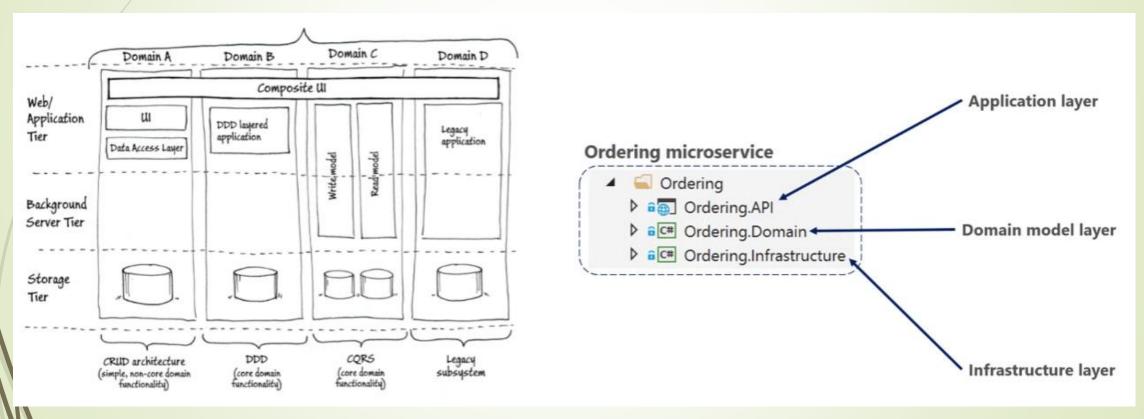


Mapa de

Contextos

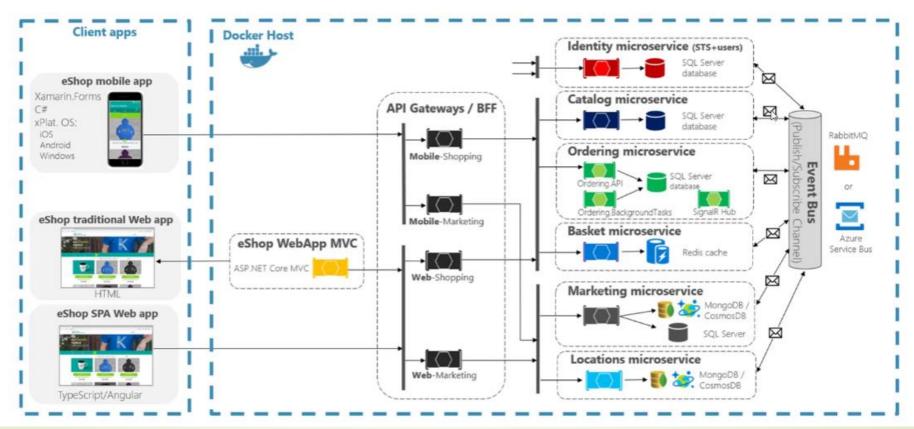
Aplicar os conceitos de DDD

Definir arquitetura dos contextos

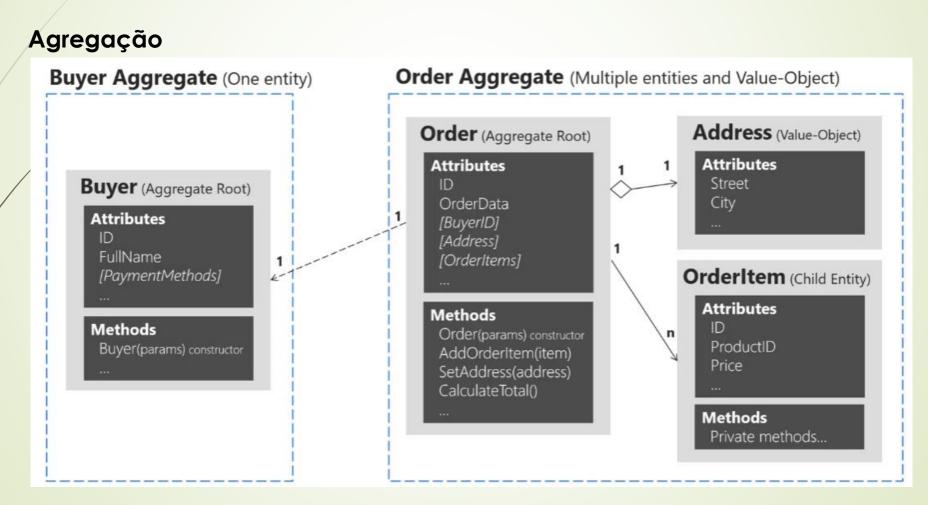


Aplicar os conceitos de DDD

Exemplo de arquitetura dos contextos

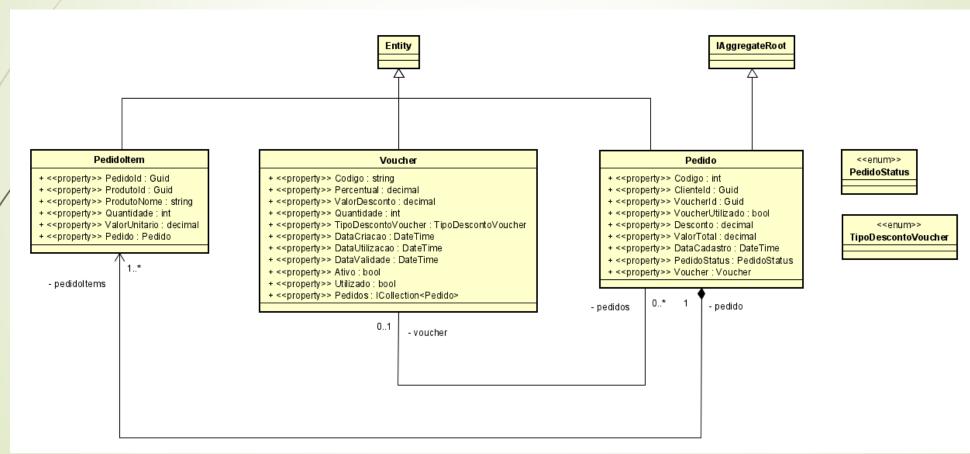


MODELAGEM TÁTICA

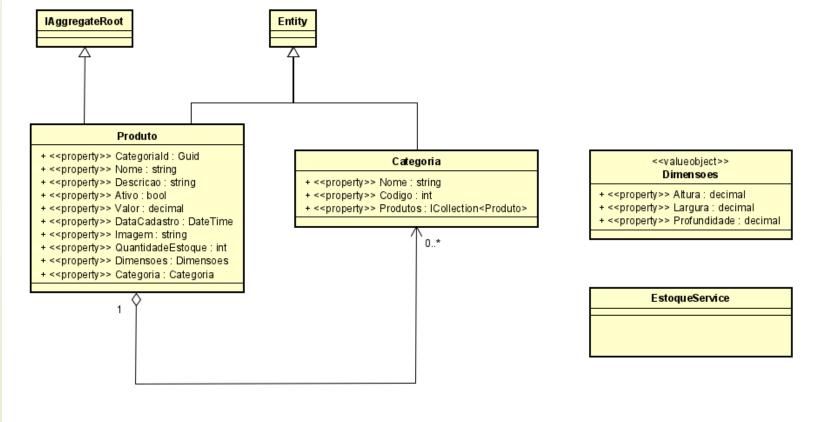


Aplicar os conceitos de DDD

Modelo Conceitual – Contexto Vendas







Aplicar os conceitos de DDD

Interface de marcação

```
# □namespace NerdStore.Core.Data

| Images | Im
```

Raiz de agregação

```
7 — namespace NerdStore.Vendas.Domain
 8
         public class Pedido : Entity, IAggregateRoot
10
             3 referências
11
             public int Codigo { get; private set; }
             public Guid ClienteId { get; private set; }
12
             public Guid? VoucherId { get; private set; }
13
             public bool VoucherUtilizado { get; private set; }
14
15
             public decimal Desconto { get; private set; }
             public decimal ValorTotal { get; private set; }
16
             public DateTime DataCadastro { get; private set; }
17
             public PedidoStatus PedidoStatus { get; private set; }
18
19
             private readonly List<PedidoItem> _pedidoItems;
20
             public IReadOnlyCollection<PedidoItem> PedidoItems => _pedidoItems;
21
22
             // EF Rel.
23
24
             public Voucher Voucher { get; private set; }
25
```

Aplicar os conceitos de DDD

Objeto de valor

```
1 ⊡using System;
2
    using NerdStore.Core.DomainObjects;
3
4 = namespace NerdStore.Catalogo.Domain
5
        public class Produto : Entity, IAggregateRoot
6
8
             public Guid CategoriaId { get; private set; }
             4 referências
9
             public string Nome { get; private set; }
10
             public string Descricao { get; private set; }
             public bool Ativo { get; private set; }
11
             2 referências
             public decimal Valor { get; private set; }
12
             public DateTime DataCadastro { get; private set; }
13
14
             public string Imagem { get; private set; }
             public int QuantidadeEstoque { get; private set; }
15
             public Dimensoes Dimensoes { get; private set; }
16
17
             public Categoria Categoria { get; private set; }
18
```

```
using NerdStore.Core.DomainObjects;
3 = namespace NerdStore.Catalogo.Domain
        public class Dimensoes
             public decimal Altura { get; private set; }
             public decimal Largura { get; private set; }
8
             public decimal Profundidade { get; private set; }
9
10
             public Dimensoes(decimal altura, decimal largura, decimal profundidade)
11 Ė
12
13
                Validacoes. Validar Se Menor Que (valor: altura, minimo: 1, mensagem: "O campo Altura não pode ser menor ou igual a 0");
                Validacoes. Validar Se Menor Que (valor: largura, minimo: 1, mensagem: "O campo Largura não pode ser menor ou igual a 0");
14
                Validacoes. Validar Se Menor Que (valor: profundidade, minimo: 1, mensagem: "O campo Profundidade não pode ser menor ou igual a 0");
15
16
17
                 Altura = altura;
                Largura = largura;
18
19
                 Profundidade = profundidade;
20
21
22 Ė
             public string DescricaoFormatada()
23
                return $"LxAxP: {Largura} x {Altura} x {Profundidade}";
24
25
26
27 Ė
             public override string ToString()
28
                return DescricaoFormatada():
29
30
31
32 }
```

Requisitos de Alto Nível - RANs



Os RANs são descrições do que o sistema/produto deve ou não fazer. Eles não definem como deve ser feito, pois isto já compete a área técnica. Eles geralmente se concentram nos resultados desejados e nas funcionalidades gerais que o sistema ou produto deve ter para atender às necessidades dos usuários e das partes interessadas.



- O sistema deve permitir que os usuários façam login e gerenciem suas contas.
- 2. O produto deve ser fácil de usar para usuários com habilidades limitadas em tecnologia.
- 3. O sistema deve ser capaz de processar um grande volume de transações em tempo real.
- 4. O produto deve ser seguro e proteger os dados confidenciais dos usuários.
- 5. O sistema deve ser escalável e capaz de lidar com o crescimento futuro da empresa.
- 6. O Produto enviado deverá ser rastreável.

Requisitos Funcionais

Requisitos Funcionais							
[RF-001]	Manter Produto						
[RF-002]	Manter Estoque						
[RF-003]	Manter Categoria						
[RF-004]	Gerir venda						
[RF-005]	Aplicar voucher pedido						
[RF-006]	Estornar estoque						

Requisito	Regra de negócio
RF 01	QUANTIDADE: produto é considerado disponível quando está ativo e tem estoque > 0
RF 01	PRODUTO: tem que ter obrigatoriamente id, nome, descrição, valor e imagem
RF 01	PRODUTO: dimensões devem ter altura, largura e profundidade maior ou igual a zero
RF 01	PRODUTO: a formatação das dimensões deve ser "L x A x P"
RF 01	PRODUTO: Sistema deve exibir detalhes de produto
RF 02	ESTOQUE: Ao concluir uma compra, sistema deve debitar quantidade de itens de estoque
RF 03	PRODUTO: Sistema deve listar todos os produtos disponíveis
RF 03	PRODUTO: Sistema deve listar produtos disponíveis filtrados por categoria
RF 03	CATEGORIA: deve ter obrigatoriamente Nome e código
RF 04	PAGAMENTO: Pagamento pode ser realizado por cartão de crédito e boleto
RF 04	CARRINHO: Precisa ter 1 ou mais itens para ser fechado
RF 04	PAGAMENTO: Status de transação pode ser Pago ou Recusado
RF 04	COMPRA: Sistema deve recuperar resumo de compra desejado
RF 05	VOUCHER: Carrinho de compras - Desconto absoluto e/ou percentual.
RF 06	PAGAMENTO: Se o pagamento não for realizado com sucesso, a compra não deve ser concluída
RF 06	ESTORNO: Em casos de devolução de produto, estoque deve ser acrescido da quantidade devolvida

Requisitos Não Funcionais

	Código	Requisitos Não Funcionais	Indicadores
/			
	RNF 01	Desempenho	Tempo de Resposta das requisições(1 a cada 100 ms)
/	RNF 02	Disponibilidade	Funcional em tempo integral (24h/7 dias por semana)
	RNF 03	Escalabilidade	picos de consumo de até 1k acessos/min
	D) [0 4		
	RNF 04	Segurança	Autenticação JWT / criptografia Char 256
	RNF 05	Privacidade	Cumprimento da LGPD

Matriz de Rastreabilidade

Requesitos X Casos de uso Obter Produto	- 1																				
Requestion X Cosos de oso Produto Produto Produto Produto em Estoque [RF-001] Manter Produto [RF-002] Manter Estoque [RF-003] Manter Categoria [RF-004] Gerir venda [RF-004] Gerir venda [RF-005] Aplicar voucher pedido Produto Produto Produto em Estoque [RF-005] Aplicar voucher pedido Produto Produto Produto em Estoque RF-005] Aplicar voucher pedido Produto Produto Produto em Estoque RF-005] Aplicar voucher pedido Produto Produto Produto em Estoque RF-005] Aplicar voucher pedido Produto Produto Produto em Estoque RF-005] Aplicar voucher pedido Produto Produto Produto em Estoque RF-006] Produto Produto em Estoque RF-007] Produto Produto em Estoque RF-008] RF-008] Aplicar voucher pedido Produto Produto Produto em Estoque RF-008] Aplicar voucher pedido RF-008] Aplicar voucher pedido Produto Produto Produto Produto Estoque Produtos Prod				[UC-001]	[UC-002]	[UC-003]	[UC-004]	[UC-005]	[UC-006]	[UC-007]	[UC-008]	[UC-009]	[UC-010]	[UC-011]	[UC-012]	[UC-013]	[UC-014]	[UC-015]	[UC-015]	[UC-015]	[UC-015]
[RF-002] Manter Estoque [RF-003] Manter Categoria [RF-004] Gerir venda [RF-005] Aplicar voucher pedido	ſ	Requesi	itos X Casos de uso	Produto por	Produto	Todos os			Produto em	Produto em		Lista Produtos		Lista Produtos	Opter	item	processa mento		IIIICIUI	item	Obtém pagamento
[RF-003] Manter Categoria [RF-004] Gerir venda [RF-005] Aplicar voucher pedido	[RF-001]	Manter Produto																		
[RF-004] Gerir venda [RF-005] Aplicar voucher pedido	[RF-002]	Manter Estoque																		
[RF-005] Aplicar voucher pedido	[RF-003]	Manter Categoria																		
pedido pedido	[RF-004]	Gerir venda																		
[RF-006] Estornar estoque	[RF-005]																			
	[RF-006]																			

Processo de estimativa



código	Requisito Funcional	Complexidade Estimada
[RF-001]	Manter Produto	5
[RF-002]	Manter Estoque	5
[RF-003]	Manter Categoria	2
[RF-004]	Gerir venda	13
[RF-005]	Aplicar voucher pedido	1
[RF-006]	Estornar estoque	8
<i>i</i>	·	

COMPLEXIDADE ESTIMADA



Cronograma de implementação

Do quisitos	Semanas										
Requisitos	1	2	3	4	5	6	7	8			
Manter Produto											
Manter Estoque											
Manter Categoria											
Gerir venda											
Aplicar voucher pedido											
Estornar estoque											

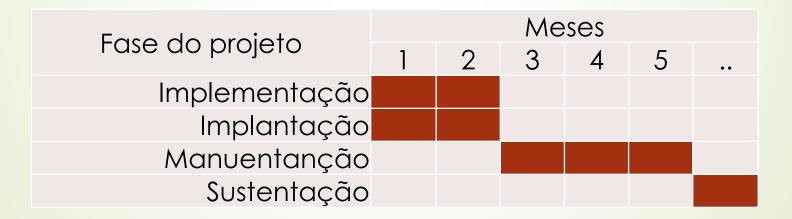
Cronograma de lançamento



Proposta de implantação



Cronograma geral



Custo do software

Manutenção	R\$	3.750,00
Implantação	R\$	6.000,00
Implementação	R\$	15.454,55
Gerenciamento	20%	
Margem de lucro	40%	
CUSTO TOTAL	R\$	30.246,86
Manutenção extra	R\$	1.250,00 / mês

Referências

MODELAGEM de Domínios Ricos. desenvolvedor.io/. Disponível em: https://desenvolvedor.io/curso/modelagem-de-dominios-ricos . Acesso em: 02 jun. 2022.

MODELAGEM de Software. Ims.infnet.edu.br. Disponível em: https://lms.infnet.edu.br/moodle/course/view.php?id=5928 . Acesso em: 10 fev. 2022.

ENGENHARIA de Software Aplicada. Ims.infnet.edu.br. Disponível em: https://lms.infnet.edu.br/moodle/course/view.php?id=5927 . Acesso em: 11 dez. 2022.

MODELAGEM de Dados UML. udemy.com. Disponível em: https://www.udemy.com/course/uml-diagrama-de-classes/learn/lecture/9881660?start=255#overview. Acesso em: 17 abr. 2022.