

MONALYSA KLAUCK YOSHIKAWA

MATRÍCULA 40473204800



ASSESSMENT - PROJETO DE BLOCO

Nome: Monalysa Klauck Yoshikawa

Curso: Análise e Desenvolvimento de Sistema

Disciplina: Projeto de Bloco

Professor: Armênio

São Paulo, 20 de setembro de 2021

Sumário

Fases do modelo cascata. 3

Problemas do modelo Cascata. 5

Modelo Iterativo e incremental 7

Relação entre OO, UML e RUP.. 8

Bibliografia. 9

Fases do modelo cascata

O modelo de cascata ou linear é uma das formas de modelo de gestão de projetos, já muito utilizado para a elaboração de software ou programas. Esse modelo assume que as informações sobre a meta e a solução do projeto são quase que perfeitamente completas. Uma coisa a ser observada nesse modelo é que para se começar uma fase do projeto a anterior tem que estar concluída, ou seja ele sempre segue em frente, diferente do ágil, inclusive esse é a referência para o nome “cascata” o fato de sempre seguir em frente.

No caso do modelo em cascata para software ele pode ser explicado em suas fases da seguinte forma:

Análise – coletar informações e estudar todos os pontos do projeto. Estudar as requisições, analisar a documentação, esquematizar o planejamento do projeto, arquitetura do programa.

Projeto – Esta fase está associada mais na criação do projeto, no desenvolvimento do projeto ou da requisição. Essa fase somente se inicia quando a parte de análise finalizou, a ideia está pronta e estrategicamente montada.

Implementação – fase do projeto que ele está desenvolvido e está sendo implantado, ou seja “colocado em prática”, subindo para o ambiente de execução do projeto para acesso do cliente.

Teste – Fase para testar a implementação e verificar se todos os campos estão corretos.

Manutenção – Essa fase está associada a manutenção, levantar os erros e subir para análise novamente.

Problemas do modelo Cascata

As fraquezas do modelo são as seguintes:

- · Não aceita muito bem as mudanças
- · O custo é muito alto
- · Demora muito tempo antes de produzir quaisquer entregas
- · Requer planos completos e detalhados;
- · Tem de seguir uma rígida sequencia de processos;
- · Não esta focado no valor para o cliente

Não aceita muito bem as mudanças:

O problema é que praticamente qualquer solicitação de mudança de escopo aprovada causara problemas para o cronograma programado. Qualquer alteração no projeto deve-se adicionar uma Declaração de Impacto sobre o Projeto, que pode inicialmente afetar o prazo do projeto.

O custo é muito alto

Como o cliente não terá nenhuma entrega ate o ultimo momento da programação do projeto, que é o momento que os testes estão avaliando se os requisitos foram satisfeitos. Geralmente acontecerão ajuste ou modificações que irá gerar um custo extras.

Demora muito tempo antes de produzir quaisquer entregas

Nesse modelo o cliente não recebe entregas ate que o projeto esteja bem avançado. Então, não sobra muito tempo para modificações mesmo havendo recursos. Como as entregas serão feitas próximo da data de finalização do projeto, os recursos muitas vezes estão já organizados para em x data iniciar em outro projeto.

Um projeto simples, sem alterações constantes, ou sem influencia do mercado para funcionar, não sofrera tanto os impactos. Entretanto, hoje em dia é difícil encontrar um projeto que não passe por algum influencia do mercado.

Requer planos completos e detalhados.

Para o modelo de projetos em cascata ou linear requer um planejamento muito bem detalhado e explicado, situação que atualmente se torna inviável, já que requer muito tempo e podem surgir atualizações que podem interferir no planejamento

Tem que seguir uma rígida sequência de processos

Para esse modelo prosseguir ele tem que seguir uma sequência de passos, sempre em frente, por isso a referência a “cascata”, e para seguir esse projeto tem que seguir o planejamento a risco, qualquer alteração acarretará em um novo planejamento e um maior custo.

Não está focado no valor ao cliente

Esse modelo ele é focado na programação do projeto dentro de prazo, dentro do orçamento de acordo com as especificações do cliente. Porém nenhum deles está se referindo em entregar valor ao cliente e sim em entregar o projeto.

Modelo Iterativo e incremental

Um modelo de CVGP Incremental consiste em vários incrementos dependentes e que são concluídos em uma sequência predefinida. Cada Incremento inclui as etapas de Lançamento, Monitoramento e Controle e Encerramento para as funções, características e funcionalidades esperadas para aquele respectivo incremento. Cada incremento integra partes adicionais da solução até o incremento final, no qual as partes remanescentes da solução são integradas.

Um modelo de CVGP Iterativo São projetos que possuem meta claramente definida e documentada, mas as características e funcionalidades ainda não podem ser claramente definidas e documentadas e cuja solução é completa a ponto de especificar a definição final de uma ou mais características e funcionalidades.

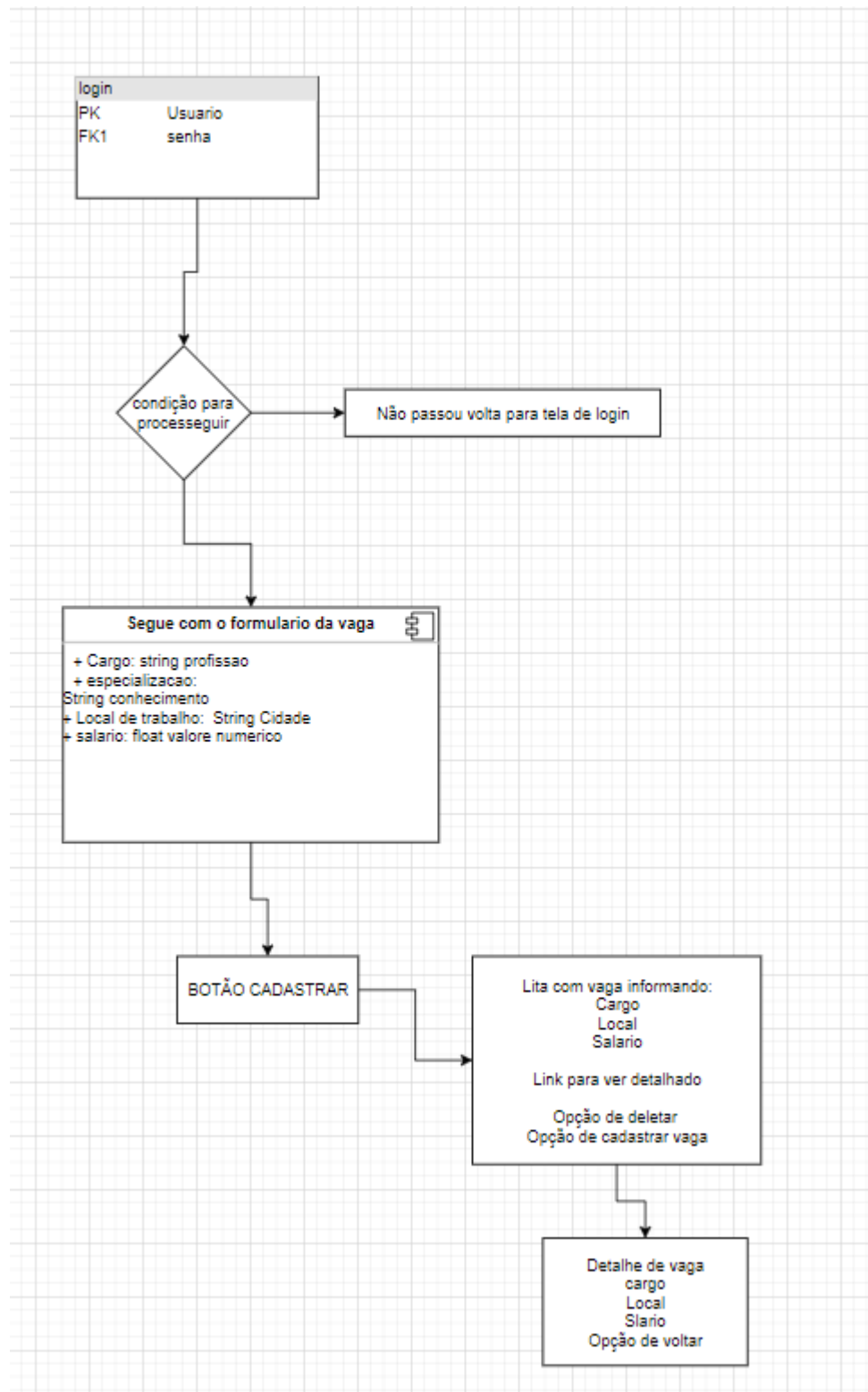
Relação entre OO, UML e RUP

O RUP se espalha sobre uma grande área do panorama de projetos. Quando a complexidade e a incerteza são baixas, mas a solução não completamente definida, o RUP é um processo pesado. Requer uma documentação considerável, especialmente para reuso de código.

Orientação a Objetos (OO) é uma abordagem de programação que procura explorar nosso lado intuitivo. Os objetos da computação são análogos aos objetos existentes no mundo real. No enfoque de OO, os átomos do processo de computação são os objetos que trocam mensagens entre si. Essas mensagens resultam na ativação de métodos, os quais realizam as ações necessárias.

UML (Unified Modelling Language) É uma linguagem para especificação, construção, visualização e documentação de sistemas. É uma evolução das linguagens para especificação de conceitos de Booch, OMT e OOSE e também de outros métodos de especificação de requisitos de software orientados a objetos ou não.

DIAGRAMA DE CASO



Testes da execução da rota no Json

http://localhost:8080/api/vagas

POST http://localhost:8080/api/vagas

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data **x-www-form-urlencoded** raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	cargo	Professora			
	Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 58 ms Size: 250 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 0,
3   "cargo": "Professora",
4   "especializacao": null,
5   "localTrabalho": null,
6   "salario": 0.0
7 }
```

http://localhost:8080/api/vagas

POST http://localhost:8080/api/vagas

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookie

none form-data **x-www-form-urlencoded** raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Ed
<input checked="" type="checkbox"/>	cargo	Professora			
	Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 493 ms Size: 250 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "cargo": "Professora",
4   "especializacao": null,
5   "localTrabalho": null,
6   "salario": 0.0
7 }
```


http://localhost:8080/api/vagas

POST http://localhost:8080/api/vagas

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	cargo	Diretora			
	Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 36 ms Size: 248 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "cargo": "Diretora",
4   "especializacao": null,
5   "localTrabalho": null,
6   "salario": 0.0
7 }
```

Esta puxando todas as dependências

http://localhost:8080/api/vagas

POST http://localhost:8080/api/vagas

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	cargo	Desenvolvedora	
<input checked="" type="checkbox"/>	especializacao	Java	
<input checked="" type="checkbox"/>	localTrabalho	Remoto	
<input checked="" type="checkbox"/>	salario	5500.45	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 36 ms Size: 248 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 6,
3   "cargo": "Desenvolvedora",
4   "especializacao": "Java",
5   "localTrabalho": "Remoto",
6   "salario": 5500.45
7 }
```

RETORNANDO A LISTA DE VAGAS CADASTRADAS DO BANCO DE DADOS

http://localhost:8080/api/vagas

Save

GET http://localhost:8080/api/vagas Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	BU
-----	-------	-------------	-----	----

body Cookies Headers (5) Test Results Status: 200 OK Time: 662 ms Size: 845 B Save Resp

Pretty Raw Preview Visualize JSON

```
6      "localTrabalho": null,
7      "salario": 0.0
8    },
9    {
10     "id": 2,
11     "cargo": "Diretora",
12     "especializacao": null,
13     "localTrabalho": null,
14     "salario": 0.0
15   },
16   {
17     "id": 3,
18     "cargo": "Desenvolvedora",
19     "especializacao": "Java",
20     "localTrabalho": "Remoto",
21     "salario": 5500.0
```

Bootcamp Runner

Inclusão do método PUT para alteração de elementos

PUT http://localhost:80... + ... No En

http://localhost:8080/api/vagas

PUT http://localhost:8080/api/vagas

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

<input checked="" type="checkbox"/>	cargo	Professora	
<input checked="" type="checkbox"/>	especializacao	JavaScript	
<input checked="" type="checkbox"/>	localTrabalho	hibrido	
<input checked="" type="checkbox"/>	salario	7500	
<input checked="" type="checkbox"/>	id	1	
	Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 497 ms Size:

Pretty Raw Preview Visualize JSON

```

2  "id": 1,
3  "cargo": "Professora",
4  "especializacao": "JavaScript",
5  "localTrabalho": "hibrido",
6  "salario": 7500.0

```

Inclusão do método Delete por Id

DEL http://localhost:80... + ... No Environment

http://localhost:8080/api/vagas/7 Save

DELETE http://localhost:8080/api/vagas/7 Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit

Body Cookies Headers (4) Test Results Status: 200 OK Time: 150 ms Size: 123 B Save Response

Pretty Raw Preview Visualize Text

1

Consulta de elemento por string no caso Cargo

GET http://localhost:80... + ... No Environment

http://localhost:8080/api/vagas/cargo/professora Save

GET http://localhost:8080/api/vagas/cargo/professora Send

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION	...	E
--	-----	-------	-------------	-----	---

Body Cookies Headers (5) Test Results Status: 200 OK Time: 68 ms Size: 268 B Save Res

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "cargo": "Professora",
5     "especializacao": "JavaScript",
6     "localTrabalho": "hibrido",
7     "salario": 7500.0
8   }
9 }
```

Conexão com banco de dados

▼ springboot

- ▼ Tables
 - ▶ hibernate_sequence
 - ▶ role
 - ▶ usuario
 - ▶ usuarios_rols
 - ▶ vaga
- Views
- Stored Procedures
- Functions

<						
Result Grid						
Filter Rows:						
Edit:						
Export/Import:						
Wrap						
	id	especializacao	local_trabalho	nome_vaga	salario	cargo
▶	3	Java	Remoto	NULL	5500	Desenvolvedora
	4	Java	Remoto	NULL	5500	Desenvolvedora
	5	Java	Remoto	NULL	5500	Desenvolvedora
	40	JavaScript	remoto	NULL	753	Desenvolvedora
	41	matematica	remoto	NULL	758	professora
	51	NULL	NULL	NULL	0	NULL
	52	NULL	NULL	NULL	0	NULL