



(Análise de Sistemas)

Aula 5

Prof. Dr. Eliseu LS



(ADS)
***Análise de Sistemas,
Metodologias Agile e Rational RUP***

Prof. Dr. Eliseu LS

ADO 2 / Elucidação de Requisitos

CENÁRIO - CLÍNICA MÉDICA ON-LINE (CENÁRIO HIPOTÉTICO)

Você foi contratado(a) para fazer o sistema online para agendamento de consultas, o cliente pode fazer o agendamento tanto por telefone quanto pelo próprio app/site. A clínica possui 3 filiais, mas somente dois funcionários que fazem os agendamentos para as filiais. São equipadas com internet, mas não possuem domínio de internet e nem serviço de hospedagem, o domínio deverá ser criado e um provedor de hospedagem também. O sistema inicialmente deverá rodar no browser com tela de login, cadastro de usuários, o cliente irá fazer irá se logar no início com o cpf, uma senha será mandada via email para o cliente que irá selecionar a especialidade, depois a clínica e finalmente o médico. No formulário de marcação de consultas ou exames deverá aparecer os dados dos pacientes, nome, cpf, etc, dados dos médicos, nome, crm, dados da especialidade, dia, horário, e dados da filial, endereço, telefone, horário de atendimento, etc.

ATIVIDADES:

I) Faça a Elucidação de Requisitos Funcionais

II) Incremente o modelo Conceitual já feito por você (D.C.) e Faça o modelo Lógico do projeto de banco de dados. (D.E.R.) contendo o maior número de atributos possíveis na versão beta.

NOTA - Relatório da atividade deverá ser entregue pelo Blackboard.

SOBRE O CRUD (Cadastro de Entidades)

CRUD (Create, Read, Update, Delete) é um acrônimo para as maneiras de se operar em informação armazenada. É um mnemônico para as quatro operações básicas de armazenamento persistente.

CREATE -> trata-se da inclusão dos dados da entidade na tabela, a criação de um novo registro da entidade.

READ -> Trata-se da consulta dos dados de das entidades ou em um formulário ou mesmo em um relatório denominado visão, por exemplo. dados de um determinado produto, cliente ou empresa fornecedora.

UPDATE -> Atualização dos dados da entidade após a consulta pela chave primária ou por um campo de descrição qualquer.

DELETE -> Exclusão do registro da entidade no banco de dados após uma consulta por chave ou por um campo de descrição qualquer.

CENÁRIO - CLÍNICA MÉDICA ON-LINE (CENÁRIO HIPOTÉTICO)

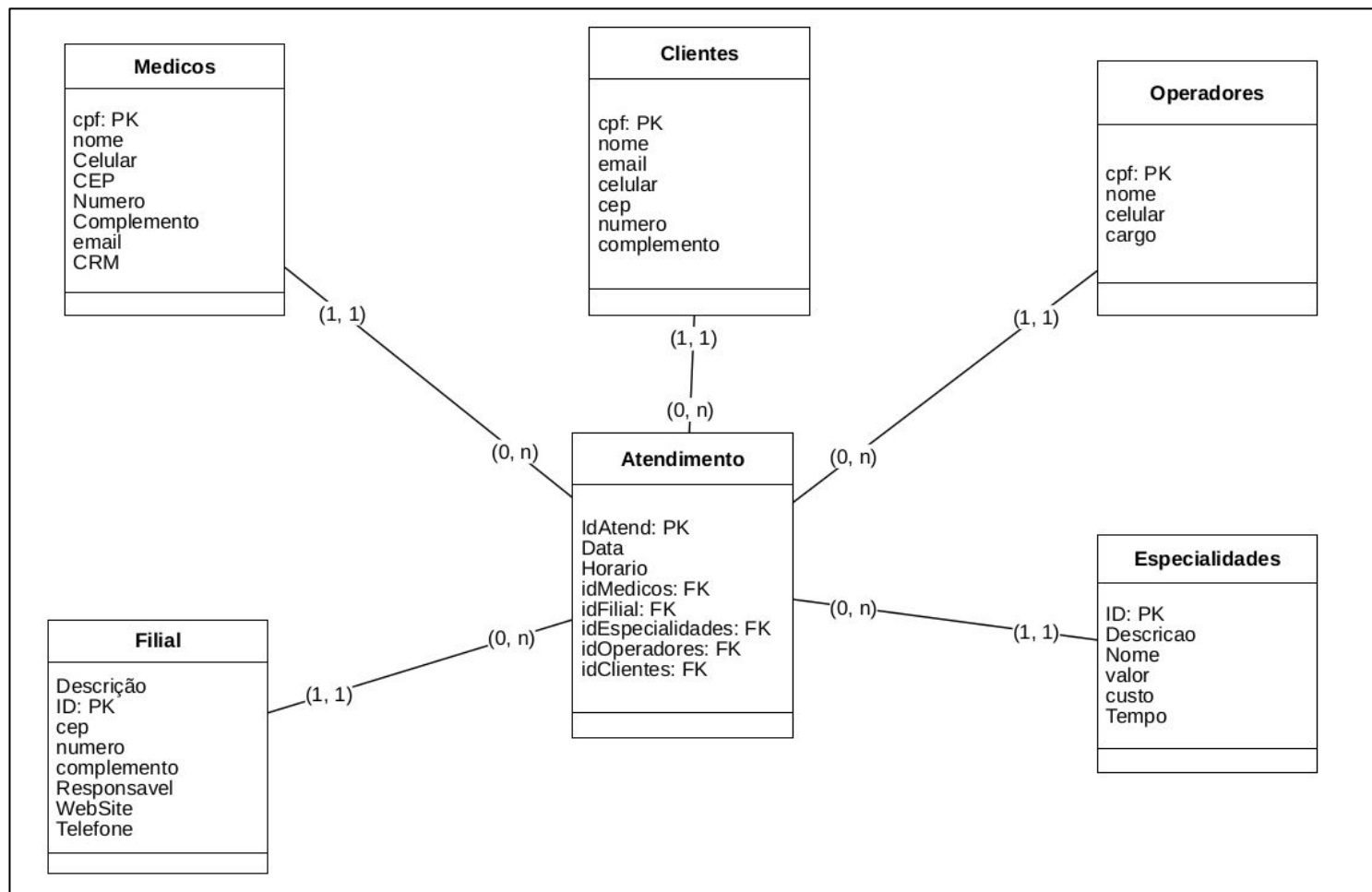
REQUISITOS FUNCIONAIS

- Cadastrar **Usuários** (Administrador, Operador, Cliente)
- Cadastrar **Funcionários**
- Cadastrar **Clientes/Pacientes**
- Cadastrar **Filiais** - Dados das unidades, endereço, telefone, cnpj, site, etc.
- Cadastrar **Especialidades** - Dados de Especialidades e demais serviços prestados
- Cadastrar **Finanças** - Cadastro dos preços das consultas e dos demais serviços
- Relatórios de consultas por médico e/ou especialidade

REQUISITOS NÃO FUNCIONAIS

- O sistema será hospedado em um provedor de internet (Hostinger)
- Será utilizado a linguagem PHP para Backend
- Para FrontEnd apenas usará html, css, javascript
- Será utilizado o SGBD Mysql

Diagrama Entidade Relacionamento - Clínica Médica On Line (Ferramenta : BrModelo Web)



Fonte: SILVA, Eliseu Lemes, 2023.

Modelos de Projeto de Software (MPS)

Um processo de desenvolvimento de software é formado por um conjunto de fases que devem ser seguidas para que o produto (software) seja produzido. Cada uma das fases é composta por tarefas com entradas (O que é necessário para que a tarefa seja realizada) e saídas (O que é produzido pela tarefa) específicas e com o papel associado (Quem deverá realizar a tarefa). Cada empresa acaba por desenvolver sua cultura para desenvolvimento de software mas de maneira geral todo processo de desenvolvimento de software contempla as fases a seguir:

***ELUCIDAÇÃO DE REQUISITOS -> ANÁLISE -> PROJETO ->
DESENVOLVIMENTO -> TESTE -> VALIDAÇÃO -> IMPLANTAÇÃO***

RATIONAL UNIFIED PROCESS (RUP)

Este MPS se baseia no Unified Process (Processo Unificado) desenvolvido pela Rational, oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento propondo metas para garantia da qualidade do software desenvolvido dentro de um cronograma e orçamentos previsíveis

É tanto iterativo como incremental, desenvolvido e implementado em partes sendo que a cada iteração ocorrerá em quatro fases:

Concepção -> Elaboração -> Construção -> Transição.

O RUP é composto por nove disciplinas e quatro fases!!!!

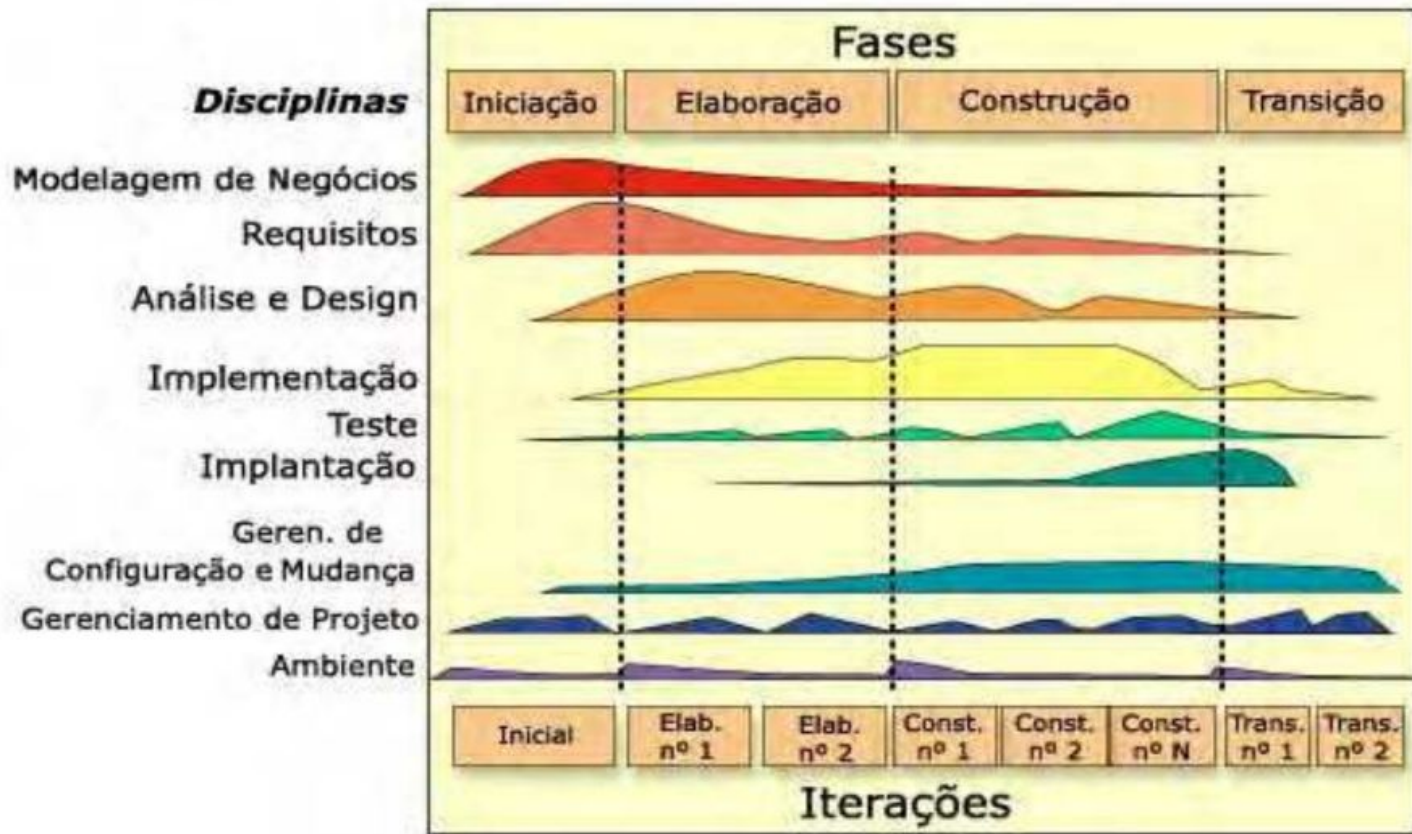


Figura. Fases e disciplinas do RUP. Fonte: (KRUCHTEN, 2003)

MPS SCRUM

O Scrum é um processo de desenvolvimento ágil de software baseado em grupos de práticas e papéis pré-definidos, sendo iterativo e incremental permite o gerenciamento baseado em fases chamadas de sprints que são interações que seguem o ciclo do acrônimo **PDCA (Plan - Planejar, Do - Fazer, Check - Verificar, Act - Agir)** e entrega um incremento de software pronto. Os principais papéis do Scrum são: **Product Owner** (Visionário/Técnico), **Scrum Master** (Coach) e **Scrum Team** (equipe do projeto).

Não existe a figura única do Gerente de Projetos. Suas responsabilidades estão diluídas entre os papéis citados. Cada um conhece sua participação frente ao projeto e trabalha em conjunto para conseguir alcançar o goal definido.

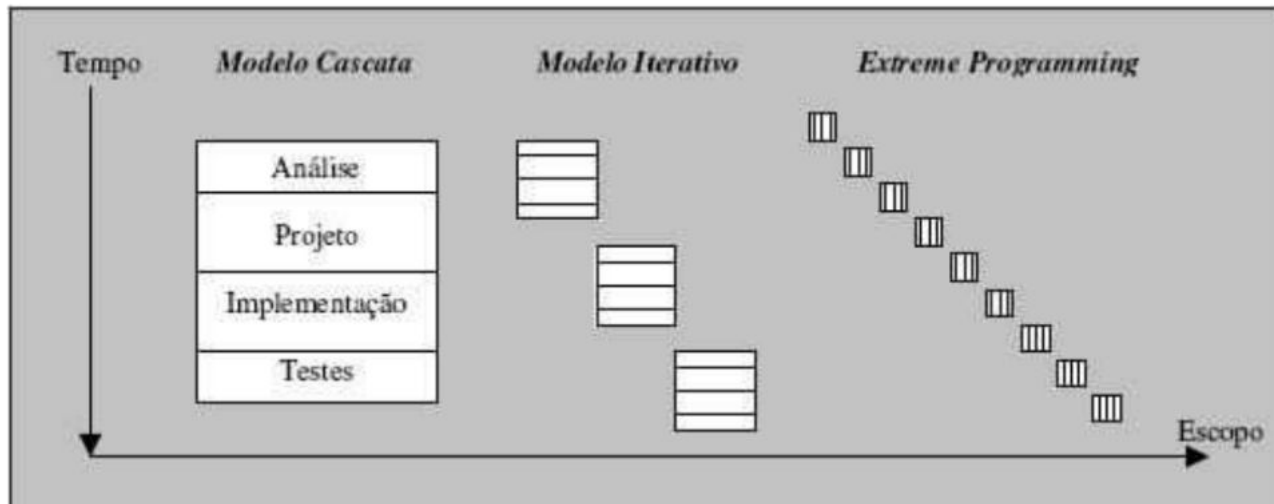
Seus artefatos principais são: o **Product Backlog** (atividades/fases do projeto) e **Sprint Backlog** – (Versão do Product Backlog) artefatos que representam seus requisitos/atividades, além de **Burndown charts e impediment backlogs**.

Maiores informações sobre certificação Scrum:
<https://www.scrum.org/resources/blog/15-dicas-para-tirar-certificacao-professional-scrum-master-psm-1>

MPS EXTREME PROGRAMMING

Fazendo parte da metodologias ágeis (agile) que inicialmente propõe métodos leves (**lightweight**) para diferenciá-los dos métodos tradicionais de desenvolvimento considerados pesados (**heavyweight**), os quais seriam baseados na produção de uma grande quantidade de documentação e modelos para guiar a programação.

Ao contrário dos métodos tradicionais que são orientados ao planejamento, previsibilidade e ao controle, os métodos ágeis são orientados à construção, testes e principalmente, às pessoas.



Como se vê na figura, o modelo em cascata possui ordenação linear nas etapas, já o modelo iterativo propõe contornar algumas limitações do modelo em cascata através da repetição. Já a XP trabalha com iterações de menor tamanho possível com os requisitos de maior importância, a equipe produz e já coloca algumas funcionalidades em prática onde o cliente acaba também contribuindo para todas as etapas.

VALORES MPS EXTREME PROGRAMMING

COMUNICAÇÃO: é o principal valor da XP. Grande parte das técnicas da XP está baseada na comunicação, e se esta não for eficiente, pode causar problemas e atrasos no desenvolvimento do sistema.

SIMPLICIDADE: a XP utiliza-se da simplicidade para implementar apenas aquilo que é suficiente para o cliente, não se procura fazer especulações sobre necessidades futuras, pois quase sempre são especulações errôneas, deixamos os problemas do futuro para o futuro.

FEEDBACK: o cliente deve receber o sistema o quanto antes, a fim de poder dar um feedback rápido, guiando assim o desenvolvimento do software.

CORAGEM: é preciso muita coragem para mudar a maneira pela qual desenvolve-se sistemas. Colocar um sistema em produção assim que ele tenha valor para o cliente, fazer apenas o que se precisa para o momento é calçar o processo de análise principalmente na comunicação não é fácil, é precisa que a equipe esteja realmente decidida a mudar o seu processo de desenvolvimento.

PRÁTICAS MPS EXTREME PROGRAMMING

Cliente Presente (On-site customer)

Na XP todas as decisões sobre o rumo do projeto devem ser tomadas pelo cliente. Ele deve priorizar as tarefas, ser responsável pelos testes de aceitação, e, acima de tudo, orientar e tirar dúvidas dos desenvolvedores durante o processo de programação.

Jogo do Planejamento (The Planning Game)

Na XP o planejamento é um processo contínuo, e o mesmo é constantemente refinado pelo cliente e pela equipe de desenvolvimento, deixando assim a metodologia bastante flexível e entregando para o cliente sempre o máximo valor pelo investimento dele.

Pequenos Lançamentos (Small Releases)

Para que o cliente possa fornecer constantemente feedback sobre o andamento do sistema, fazendo possível que o jogo do planejamento (planning game) seja executado, é importante que o sistema tenha releases pequenos, a fim de ajustar o desenvolvimento às necessidades que vão surgindo no decorrer do processo. Normalmente,

Desenvolvimento Guiado pelos testes (Test First Design)

O desenvolvimento guiado pelos testes define que qualquer método de um objeto que possa falhar deve ter um teste automatizado que garanta o seu funcionamento.

PRÁTICAS MPS EXTREME PROGRAMMING

Integração Contínua (Continuous Integration) - Integração entre as partes do sistema que está sendo desenvolvido.

Projeto Simples (Simple Design) - Designer simples para acesso dos usuários.

Refatoração (Refactoring) - Alterações rápidas no sistema quando o cliente solicitar.

Programação em pares (Pair Programming) - Programação sempre em pares de programadores para cada Artefato.

Propriedade Coletiva (Collective Ownership) - Todos são donos do código e podem dar sugestões de melhoria e apontar possíveis erros.

Padrões de Codificação (Coding Standards) - O uso de frameworks é importante e o reuso do código de vital importância em todas as funcionalidades possíveis.

- **Ritmo Sustentável (40 Hour Week)** - A jornada de trabalho deve ser controlada para evitar as pressões e estresses da equipe na construção dos artefatos.
- **Metáfora (Metaphor)** - Usar metáforas para explicar o sistema ao usuário é membros das equipes de trabalho.

Responda as questões da Revisão Teórica (Próxima ADO 3 - Parte 1)

1. Descreva as atividades do ciclo de vida de um modelo de software em cascata.
2. Descrevas as atividades do fluxo de processo de um modelo genérico de processo de software.
3. Qual a diferença entre as metodologias convencionais e as metodologias Agile para desenvolvimento de Software ?
4. Qual a diferença entre banco de dados relacional e Banco de Dados hierárquico ?
5. Qual a diferença entre o modelo lógico e o modelo conceitual de banco de dados ?
6. Defina o que seria Entidades e qual o papel das entidades na produção de softwares e nos modelos de dados.
7. Explique a diferença entre Primary Key e Foreign Key.
8. Explique o conceito dos três tipos de relacionamentos do modelo relacional e como na prática é feito cada um deles.
9. Explique o que você entende sobre gerência de projetos e qual a sua importância para a produção de projetos de softwares.
10. Explique e dê exemplos de requisitos funcionais e requisitos não funcionais.

NOTA: Questões de revisão para ajudar na AV1, cada questão deverá ser respondida em um parágrafo de no mínimo 5 linhas com exemplos feita por pequenos grupos, mas a entrega será individual. .