

**RELATÓRIO TÉCNICO**

1. **Identificação**

Equipe Técnica:

Carlos Henrique Vieira Marques - RA: 18720367

Erick Matheus Lopes Pacheco - RA: 18711630

Hiago Silva Fernandes - RA: 18726455

João Henrique Pereira - RA: 18712919

Leonardo Sanavio - RA: 18054395

Cliente: Rede ACME de Cinemas

Referência: S.O.

Nome: Sistemas de Ocupação.

Descrição: O sistema visa a solução do problema de filas dos cinemas com a criação de um aplicativo onde não mais terá-se a necessidade de ir no caixa do cinema.

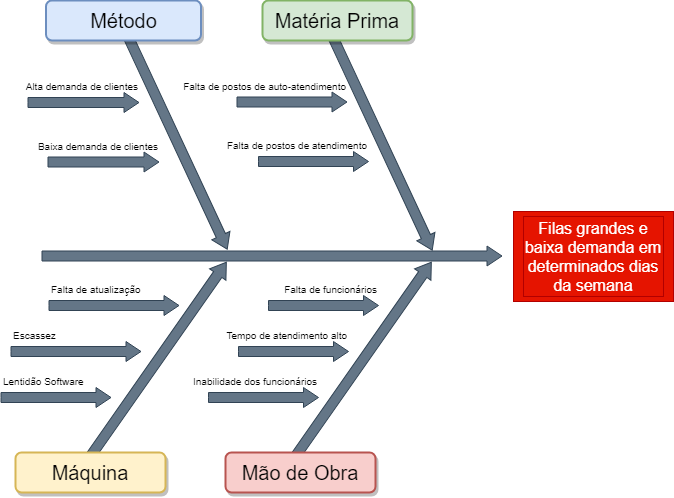
**2 Propósito e Motivação**

Nos cinemas da Rede ACME, é comum as filas serem grandes em determinados dias e horários da semana, como em sábados, domingos e feriados, e a situação se inverte nos demais dias gerando muitas salas vazias, com poucos clientes, por causa da pequena quantidade de procura.

Devido aos prejuízos sofridos pelas extensas filas e salas vazias, a empresa visa adquirir uma solução para enfrentar a desmotivação das pessoas nesses momentos de grande fluxo de clientes e sanar a falta destes em períodos de baixo fluxo. Sendo assim, nossa solução consiste em propor um sistema de compra de ingressos mais ágil, por meio do desenvolvimento de um aplicativo mobile que será disponibilizado para download para os clientes baixarem e, por meio dele, realizar a compra e pagamento de ingressos de filmes em cartaz da maneira mais dinâmica possível: permitindo escanear um QR Code referente ao filme desejado pelo cliente disponível no cartaz do filme, podendo optar por meios de pagamento virtuais, como o uso de cartão de crédito e outros. Tal ferramenta tornaria possível a redução das filas de compra, fazendo com que somente os que pagarão em dinheiro ou não queiram baixar o aplicativo a enfrente.

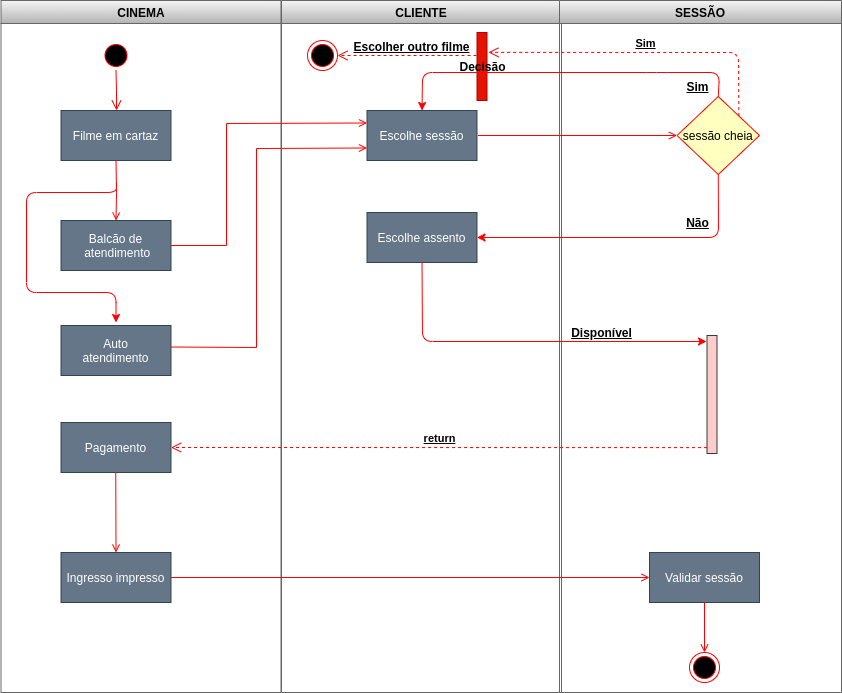
Surge, a partir daí, um problema: como incentivar o uso do aplicativo? Além do uso de placas dentro do próprio cinema com chamadas atrativas como “Deixe de enfrentar filas hoje mesmo, baixe o app e compre seu ingresso em um só passo!”, entramos, então, com a solução do outro problema, o da falta de demanda, que não só poderia aumentar a demanda em alguns dias e horários da semana como também incentivará o uso da aplicação. A proposta parte da premissa da criação de um sistema de bonificações dentro do aplicativo, onde cada ingresso comprado utilizando o app geraria um cashback em “cinepoints”, como chamaremos a moeda virtual. O usuário do aplicativo poderá utilizá-los para obter descontos de até 100% na compra de ingressos e em itens da bomboniere. Para forma de estimular ainda mais o uso do aplicativo e a demanda durante a semana, o aplicativo também contará com a presença de descontos exclusivos para ingressos de filmes no meio de semana.

**3 Possíveis causas do problema**

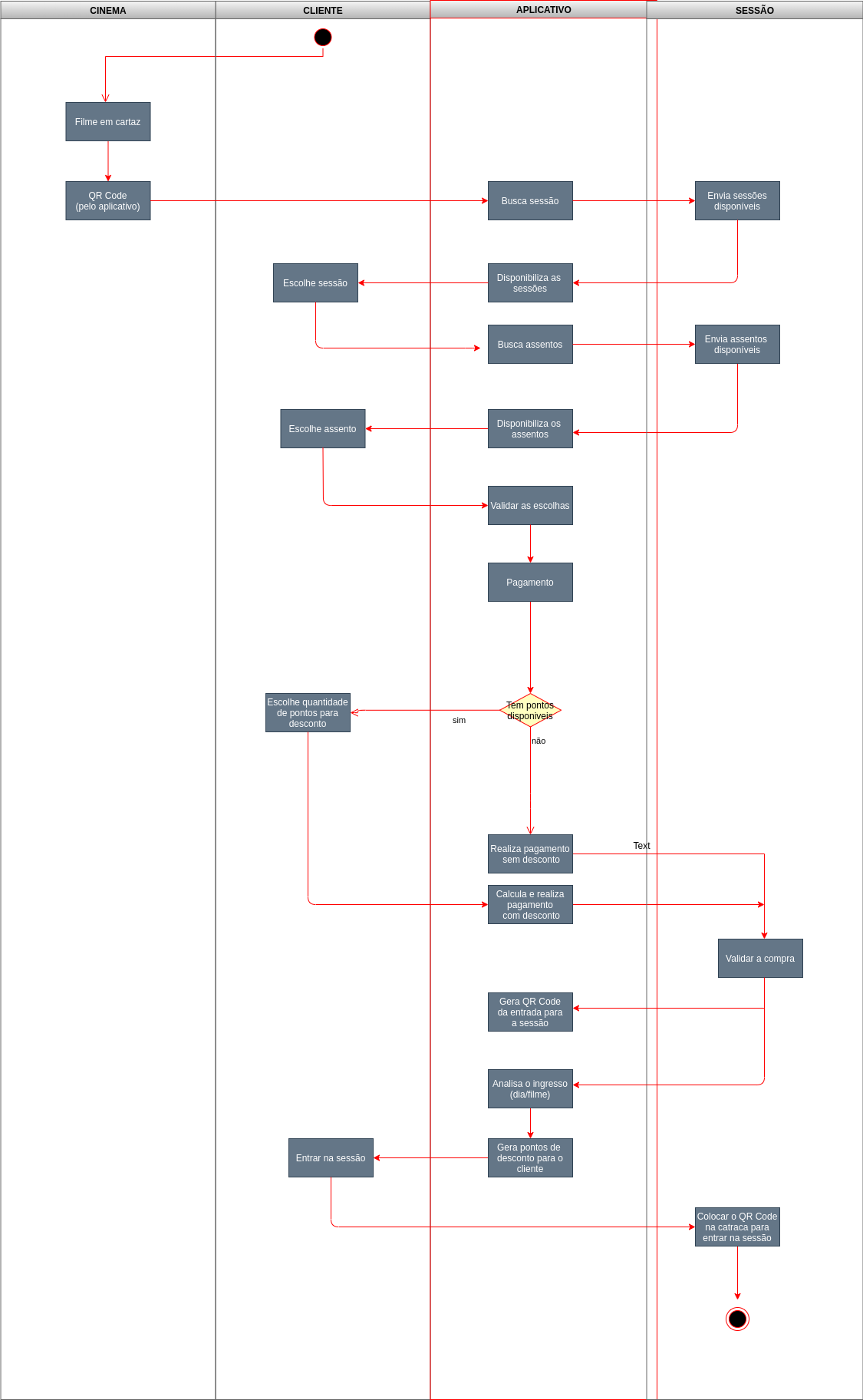


**4 Mapeamento dos processos do sistema**

**4.1 Mapeamento dos processos (atual)**



**4.2 Mapeamento dos processos (proposto)**



**5 Escopo**

O projeto á implementar levará em conta o fornecimento de um aplicativo para facilitar a compra dos ingressos e disponibilizar uma outra opção de compra para o cliente. Será criado também, um sistema de acúmulo de pontos para quem escolher comprar os seus ingressos pelo aplicativo, trocando assim esses pontos por descontos em compras futuras como ingressos, produtos e combos ACM.

O sistema agora passará a se comunicar, via internet, com um aplicativo que estará instalado no dispositivo de cada cliente/usuário. E desse modo, enviará os dados de cada cliente para o sistema utilizado no cinema, que haverá mudanças necessárias para se integrar com o sistema do aplicativo.

O sistema implementado terá que ser capaz de coletar informações de primeira mão do usuário e encaminhar para o sistema utilizado no cinema, para ser confirmado as respectivas compras realizadas. Será coletado dados como a identificação da conta do usuário, identificação do QR Code escaneado, método de pagamento utilizado, informações da compra que será realizada (dados de pagamentos, valor, pontos possuídos, desconto aplicado), informações dos ingressos adquiridos (filme escolhido, sessão escolhida, horário, número da sala, lugares escolhidos). O sistema do cinema irá se encarregar de confirmar, aplicar, armazenar e retornar as compras e ações realizadas pelo cliente no aplicativo.

O cliente irá escanear o QR Code disponibilizado no cartaz e será redirecionado para a parte de compras, no caso a compra do respectivo filme identificado pelo QR Code escolhido, em seguida o cliente realizará a compra, pelo método que preferir (cartões de crédito e débito, PagSeguro, PayPal, Mercado Pago, etc), e os dados serão enviados para o sistema do cinema, que irá retornar a confirmação da compra feita, junto com suas informações (confirmação do valor cobrado, filme escolhido, sessão, sala, horário, desconto aplicado, pontos consumidos), e enviar outro QR Code para o aplicativo do cliente que representará o ingresso para entrar na sessão. E assim o aplicativo atualizará as informações necessárias no dispositivo do cliente, e armazenar os ingressos comprados (QR Codes) na conta do usuário, que a qualquer momento poderá visualizá-los.

Em seguida, na entrada para a sessão, será escaneado o QR Code do aplicativo do cliente, através de uma catraca posicionada, e enviado para o sistema do cinema, para confirmar a entrada do cliente e atualizar o status do ingresso (QR Code) com já utilizado e não válido novamente, e assim já exclui o ingresso utilizado do aplicativo do cliente.

**6 Usuário-Chave**

Nosso sistema será composto por diversos usuários. Os próprios clientes da Rede ACME, e equipe técnica e os responsáveis e gerentes da Rede. O sistema se divide de maneira hierárquica sendo cada grau de hierarquia um limite de acesso sobre os dados do sistema. Os níveis de hierarquia se dividem em Master, Middle e User. Sendo o master o acesso livre a todos os dados como filmes, gasto da empresa, pontos acumulados por todos os usuários e gastos gerais da Rede e também seu Lucro, este nível é dado somente para os Donos da Rede e a Equipe de manutenção do sistema. O middle é limitado em acesso apenas ao login dos clientes da rede, sua quantidade de pontos e os filmes, pois caso ocorra algum problema com os pontos eles sejam hábeis a resolver e os. Usuários têm acesso somente a seus dados e aos filmes que deseja ver. Dessa forma, para um bom funcionamento é no mínimo necessário 1 usuário-chave para cada Nível, sendo de suma importância uma equipe de coordenação para o sistema, para ocorrência de feedbacks, análises e reuniões visando melhorias no Software. Uma equipe de gerentes também é essencial para resolver problemas com os Usuários no dia-a-dia do cinema. E por fim, junto destas equipes de suporte uma equipe para Apoio técnico seria ótimo para suporte em casa de problemas com o sistema.

OBS: O Link da tabela e uma imagem com o seu modelo está apresentado a seguir:

<https://docs.google.com/spreadsheets/d/1Z67YfCto081_bhDjQS41G_aBFm2cSka3_U3vVeSPFP4/edit?usp=sharing>



**7 Definições, siglas ou abreviaturas**

**Cinema:**

Filmes em Cartaz: Filmes que foram lançados recentemente e estão disponíveis para assistir.

Filmes em breve: Filmes que foram anunciados, mas não estão disponíveis ainda.

Sessão: Determinada sala com sua respectiva identificação, data e horário da apresentação de um filme.

Programação: Lista de filme que estão em cartaz no cinema ao longo da semana.

**Sistema:**

Sistema de Entrada do Sistema:

Qr Code: Código de barras bidimensional que pode ser facilmente escaneado usando a maioria dos telefones celulares equipados com câmera. Permite que o cliente evite filas e já vá para opção de compra da sessão do filme escaneado.

Compra Online: Opção de o cliente comprar ingressos pela internet via Browser ou Aplicativo de Celular.

Cine Points: Mecanismo de premiação sempre que uma compra é realizada online (Cash Back).

Níveis da Hierarquia:

Master- Acesso Ilimitado.

Middle - Acesso Intermediário.

User - Acesso de Usuário.

Dashboard: Fazer o controle dos filmes, cadastrando e fazendo alterações necessárias no um banco de dados.

**8 Referências**

Documentação da Linguagem JavaScript:

Javascript - Mozilla Web Docs. JavaScript, 2020. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em 21 de abril de 2020.

Documentação da Linguagem utilizada para o desenvolvimento mobile:

React Native - A framework for building native apps. React Native, 2020.Disponível em: <<https://reactnative.dev/>>. Acesso em: 18 de abril de 2020.

Código de Defesa do Consumidor:

Código de Defesa do Consumidor. Planalto, 2020. Disponível em: <<http://www.planalto.gov.br/ccivil_03/leis/l8078.htm> >. Acesso em: 22 de abril de 2020.

Marco Civil da Internet:

Marco Civil da Internet. Planalto, 2020. Disponível em: <<http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm>>. Acesso em: 21 de abril de 2020.

Lei de Registro de Software:

Portal da Câmara dos Deputados. Câmara dos Deputados, 2020. Disponível em: <<https://www2.camara.leg.br/legin/fed/lei/1998/lei-9609-19-fevereiro-1998-364738-norma-pl.html>>. Acesso em: 25 de abril de 2020.

Registro da Marca:

Lei 9279. Planalto, 2020. Disponível em: <<http://www.planalto.gov.br/ccivil_03/Leis/L9279.htm>>. Acesso em 18 de abril de 2020.

Lei da Relação de Consumo:

Lei 8078. Planalto, 2020. Disponível em: <<http://www.planalto.gov.br/ccivil_03/leis/l8078.htm>>. Acesso em: 18 de abril de 2020.

**9. Descrição do sistema**

A tela inicial da aplicação contará com três opções: login, cadastro e “escanear QR”, tendo em vista que a aplicação possibilitara seu uso sem a necessidade de cadastro, apenas perdendo a vantagem de cashback via sistema de pontuação, mantendo um dos propósitos vigentes da aplicação, que é possibilitar uma compra mais prática, rápida e longe das filas.

Cada filme em cartaz possui seu respectivo QR code, ao escaneá-lo, o usuário é redirecionado para uma tela contendo as informações sobre o filme (sinopse, classificação indicativa) e as sessões disponíveis com seus respectivos horários.

Selecionando a sessão, caso a mesma esteja cheia, uma pop-up de alerta informando “sessão cheia, favor selecionar outra” aparecerá e o cliente deverá escolher outra. Em caso de sessão com assentos disponíveis, o usuário será redirecionado para uma tela onde selecionará as poltronas não ocupadas e o número de ingressos se ajustará automaticamente de acordo com a quantidade de poltronas selecionadas, e, após enviar a requisição, o usuário será redirecionado para uma próxima tela onde deverá discriminar quantos dos ingressos são entradas inteiras e quantos são meia entrada, se deseja algum item da bomboniere, bem como a forma de pagamento (débito, crédito, Google Pay, etc.) sendo informado sobre o valor unitário de cada item e o valor total da compra.

Confirmado, é enviada uma requisição de compra de ingressos para o sistema, bloqueando temporariamente os assentos selecionados pelo usuário e, assim, garantindo que dois usuários não comprem ingressos para um mesmo assento ao mesmo tempo. O usuário é redirecionado para uma tela de pagamentos em conformidade com a opção de pagamento selecionada onde é informado novamente sobre os valores da compra, filme e assentos selecionados, sala e quantidade de cinepoints que receberá (desde que logado). Nesta tela o usuário também terá a opção de aplicar a quantidade desejada de “cinepoints” (desde que disponível) para obter desconto (de até 100%, de acordo com a quantidade de cinepoins) e deverá informar os dados de pagamentos e, então, confirmar a compra enviando uma requisição de finalização de compra.

Após confirmada a compra pelo sistema, o usuário receberá os cinepoints (se logado) e o(s) código(s) do(s) seu ingresso(s) juntamente com o(s) QR code(s) que serão escaneados na entrada, e, validando o(s) ingresso(s), é liberada a catraca para passagem.

**10 Requisitos**

**10.1 Requisitos funcionais**

|  |  |
| --- | --- |
| Identificador | RF 001 |
| Nome | Cadastros |
| Descrição/Regras | O aplicativo possibilitará o cadastro de usuários e no mesmo a classificação em usuário e um mestre (seria o usuário do cinema) |

|  |  |
| --- | --- |
| Identificador | RF 002 |
| Nome | Alteração de dados |
| Descrição/Regras | Por meio do aplicativo, quando logado, o usuário poderá fazer a modificação e alteração de seus dados. |

|  |  |
| --- | --- |
| Identificador | RF 003 |
| Nome | Login |
| Descrição/Regras | O acesso ao aplicativo será por meio de um login (e-mail) e senha que será determinada no cadastro. |

|  |  |
| --- | --- |
| Identificador | RF 004 |
| Nome | Atividades/Usuário |
| Descrição/Regras | O usuário poderá fazer compras pelo app. |

|  |  |
| --- | --- |
| Identificador | RF 005 |
| Nome | Atividades/Usuário |
| Descrição/Regras | Depois de escolhido um filme (sessão, poltrona etc) o mesmo será direcionado para o pagamento onde poderá obter desconto caso esteja logado e tenha “cinepoints”. |

|  |  |
| --- | --- |
| Identificador | RF 006 |
| Nome | Atividades/Usuário |
| Descrição/Regras | O usuário poderá visualizar os novos filmes que estão em cartaz, e os que tem sessões disponíveis com assentos. |

|  |  |
| --- | --- |
| Identificador | RF 007 |
| Nome | Atividades/Usuário |
| Descrição/Regras | Será disponibilizado em uma aba os pontos acumulados pelo cliente(usuário) para que este tenha total controle dos seus bônus. |

|  |  |
| --- | --- |
| Identificador | RF 008 |
| Nome | Seleção de assentos Favoritos |
| Descrição/Regras | Após selecionar o filme e horário no ato de comprar o ingresso, serão sinalizados os assentos mais utilizados/favoritos do cliente, para selecionar.  - Caso todos os assentos mais utilizados/favoritos do cliente não estejam disponíveis, uma mensagem de alerta dirá: "Infelizmente não será possível escolher seu assento favorito :(". |

|  |  |
| --- | --- |
| Identificador | RF 009 |
| Nome | Busca por Filmes |
| Descrição/Regras | - Critérios de busca, qualidade de separação de critérios, seletividade e sistema inteligente, retorna os filmes mais rentáveis.  1- Filtros a serem inseridos:   * Gênero * Idade * Data de Lançamento * Data de estreia * Nome * Ano |

|  |  |
| --- | --- |
| Identificador | RF 010 |
| Nome | Histórico de Compras |
| Descrição/Regras | Será disponibilizado uma tela com o histórico de compras do usuário com a data de compra e a data de apresentação e os pontos obtidos como bônus em cada. |

|  |  |
| --- | --- |
| Identificador | RF 011 |
| Nome | Mostra do QR Code |
| Descrição/Regras | Após a confirmação de pagamento do usuário quando o mesmo acessar a tela de histórico terá a possibilidade de clicar no filme para pegar o QR Code para passar na catraca. |

**10.2 Requisitos não funcionais**

|  |  |
| --- | --- |
| Identificador | RNF001 |
| Nome | Linguagem De Programação |
| Descrição/Regras | Utilização da linguagem JavaScript com o framework para desenvolvimento mobile React Native com possibilidade de criação de apps nativas com um único código. |

|  |  |
| --- | --- |
| Identificador | RNF002 |
| Nome | Requisito de Portabilidade |
| Descrição/Regras | O aplicativo deverá ser compatível com múltiplos sistemas mobile, ou seja, android e IOS. |

|  |  |
| --- | --- |
| Identificador | RNF003 |
| Nome | Documentação |
| Descrição/Regras | Para o desenvolvimento ser implementado serão utilizados diagramas UML para que se tenha uma melhor visualização do sistema de software |

|  |  |
| --- | --- |
| Identificador | RNF004 |
| Nome | Padronização |
| Descrição/Regras | Será utilizado uma estratégia para que os termos utilizados pelos desenvolvedores sejam “entendíveis” pelos profissionais de outras áreas de TI e vice-versa. |

|  |  |
| --- | --- |
| Identificador | RNF005 |
| Nome | Interface |
| Descrição/Regras | Para um melhor entendimento por diversas pessoas a interface será desenvolvida de forma que seja intuitiva e autoexplicativa, focando em um maior número de imagens. |

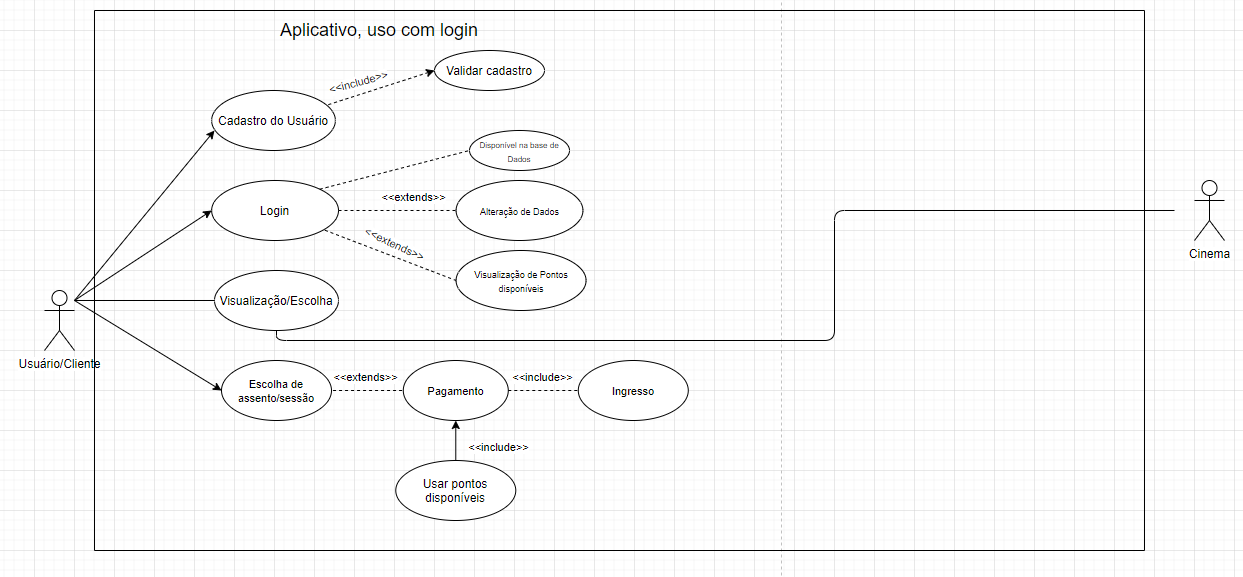
|  |  |
| --- | --- |
| Identificador | RNF 006 |
| Nome | Dados |
| Descrição/Regras | Para uma maior segurança não deve ser utilizada o armazenamento de dados locais (armazenamento interno do aparelho). |

**11 Restrições iniciais**

Primeiramente uma das restrições iniciais seria o tempo, já que a solução tem um prazo máximo, caso esse prazo seja ultrapassado geraria mais percas a empresa. Outra restrição está na linguagem utilizada para o desenvolvimento do mesmo, já que pode ocorrer de ter a necessidade de treinamento para a equipe responsável pelo desenvolvimento. Um outro problema seria ainda o “linguajar” utilizado pela empresa contratante e pelos profissionais de desenvolvimento ou o responsável pela obtenção dos requisitos necessários ao desenvolvimento. Problemas com a equipe de desenvolvimento causaria problemas no prazo de entrega.

**12 Modelagem dos Casos de Uso**

Diagrama de casos de uso apresentado abaixo:



**13 Combinando Paradigmas de Engenharia de Software**

**13.1 Preliminares:**

Abaixo está uma tabela representando as fases realizadas em cada modelo em questão.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Cascata** | **Incremental** | **Evolutivo** | **Prototipação** | **Baseado em componentes** | **Espiral tradicional** | **Espiral Atualizado** | **Espiral para desenvolvimento baseado em componentes** | **RUP** | **XP** | **SCRUM** |
| **Solicitação do cliente** |  |  |  |  |  |  | **X** |  |  |  |  |
| **Avaliar a contribuição do sistema para o negócio** |  |  |  |  |  |  |  |  | **X** |  |  |
| **Análise de requisitos** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **Refinar protótipo** |  |  |  | **X** |  |  |  |  |  |  |  |
| **Análise dos componentes** |  |  |  |  | **X** |  |  |  |  |  |  |
| **Modificação dos requisitos** |  |  |  |  | **X** |  |  |  |  |  | **X** |
| **Análise de riscos** |  |  |  |  |  | **X** | **X** | **X** | **X** |  |  |
| **Modelagem** |  |  |  |  |  |  | **X** | **X** | **X** |  |  |
| **Projetos de software** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **Codificação (algoritmos e linguagens de programação)** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **Entrega** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **Testes (cliente e equipe)** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  | **X** |
| **Manutenção** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |

É evidente como os modelos de ciclos de vida de software se modificaram ao longo do tempo. Podendo ser lineares ou cíclicos, com diferentes tipos e graus de interatividade cliente-desenvolvedor. Apresentam fases em comum, como análise de requisitos, projetos de software, codificação e testes, entretanto, tais tópicos não se organizam necessariamente de maneira linear, seguindo um fluxo sequencial de acontecimentos.

De acordo com a tabela apresentada acima, referente às fases que cada modelo segue para o desenvolvimento de projetos, podemos retirar as seguintes informações sobre eles:

Praticamente todos os modelos apresentam um objetivo semelhante, que é a análise e especificação dos requisitos, na qual os responsáveis pela parte apresentada, verifica quais são os requisitos necessários que o projeto possui e divide as tarefas que serão realizadas em cada parte para o desenvolvimento do software em questão. Porém, alguns modelos se diferenciam nessa fase, como o *Espiral Atualizado* que realiza uma solicitação do cliente antes de dar início a fase de análise de requisitos. Outro modelo que se destaca é o *RUP* que faz uma avaliação da contribuição do sistema para o negócio, para analisar se compensa aderir ao projeto.

Todos os modelos apresentados possuem e realizam essa etapa de Projetos de Software, que se refere ao desenvolvimento de um projeto do produto que será o projeto básico a ser construído. Todos passam por essa parte, porém alguns modelos como o *Espiral Atualizado*, *Espiral* *para desenvolvimento baseado em componentes* e o *RUP* optam antes por desenvolver a modelagem do projeto que se referem a avaliar os negócios usando casos de uso de negócios.

Na parte de finalização do projeto, destas etapas citadas acima, é feita a entrega do produto para o cliente. Mas alguns modelos específicos apresentam outras etapas a mais, antes ou depois da entrega para o cliente, como é o caso do modelo Cascata e o RUP, que possuem uma parte específica para realizarem a manutenção do produto entregue para o cliente.

Apesar das fases em comum nos distintos modelos, vale exemplificar as especificidades de alguns ciclos.

Refinamento do protótipo: O modelo de prototipação usa, evidentemente, protótipos antes do produto final. O refinamento é a etapa onde, após o uso e avaliação do protótipo pelo cliente, se ressignifica o conceito do produto, reavaliando os requisitos que se tornaram mais claros a partir do uso do protótipo.

Análise dos Componentes e Modificação dos Requisitos: Presentes no modelo Baseado em Componentes, tais etapas compõem a centralidade do modelo. Por meio da fase de Análise dos Componentes, se avalia quais componentes já existentes, poderão ser reutilizados para fornecer parte das funcionalidades do produto. A partir desse estudo, será redefinido os requisitos na etapa de Modificação, de maneira a se adequar com os componentes encontrados, de forma que tal modificação, não se distancie radicalmente dos requisitos definidos inicialmente.

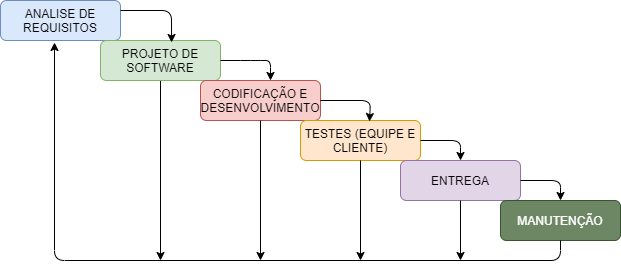
**Solicitação do cliente:**

**Avaliar A Contribuição Do Sistema Para O Negócio:**

As especificidades não se restringem somente às etapas do modelo, mas também em como são implementadas. Modelos denominados ágeis, como o XP, apesar de apresentarem etapas comuns à outros ciclos, tem na centralidade de seu funcionamento a interatividade cliente-desenvolvedor, promovendo constante presença do cliente no decorrer do desenvolvimento do produto.

**13.2 Criar o Modelo de Processo Resumido:**

Diagrama baseado nas atividades comuns encontradas em todos os modelos:



Analisando a tabela criada no tópico 13.1, foi possível comparar e construir um diagrama próprio, com as atividades em comuns que aparecem em todos os modelos. Explicando mais sobre cada fase, temos:

Análise de Requisitos: Fase onde acontece a identificação, por meio de entrevistas com o cliente, e a análise dos requisitos e conceitos do novo projeto a ser tratado, com foco no usuário, no processo e nas restrições e riscos, que são validados juntamente com o usuário, assim como as ferramentas utilizadas, metodologia e técnicas. Fica a cargo de um profissional específico, que realizará a ponte entre clientes e desenvolvedores, garantindo a preservação dos interesses do usuário diante do desenvolvimento do projeto.

Projeto de Software: Onde são definidos os componentes de softwares que serão utilizados, as funções contidas no produto, mantendo o foco nos dados e no projeto arquitetural do produto de software. Assim como também é definida a prototipação das interfaces para o usuário validar.

Codificação e Desenvolvimento: Onde é definido as sintaxes e linguagens que serão utilizadas para a construção do produto, assim como analisadas as funcionalidades disponíveis e limitações do projeto.

Testes (equipe e cliente): Onde é realizado os testes, tanto da equipe, quanto do cliente. Com foco nas especificações de saídas do produto, utilizando técnicas de testagem, procedimentos de qualidade, procedimentos de instalação e ferramentas de testagem.

Entrega: É feita a entrega do produto por completo, porém aberto ainda à manutenção se forem necessárias.

Manutenção: É a fase mais longa dos ciclos. Onde é realizadas as manutenções necessárias após a entrega do produto. Contendo vários tipos de manutenção, como corrigir erros remanescentes, adaptar a novas situações e necessidades, preparar para futuras alterações, entre outros.

**13.3 Criar o Modelo de Processo Universal:**

· Crie um segundo modelo gráfico onde apareça o máximo de atividades específicas detectadas nos ciclos de vida estudados, de forma a termos um modelo genérico (amplo) e atualizado, capaz de dar conta das preocupações tratadas nos modelos estudados (análise de riscos, negociação entre os interessados, uso de componentes, etc.).

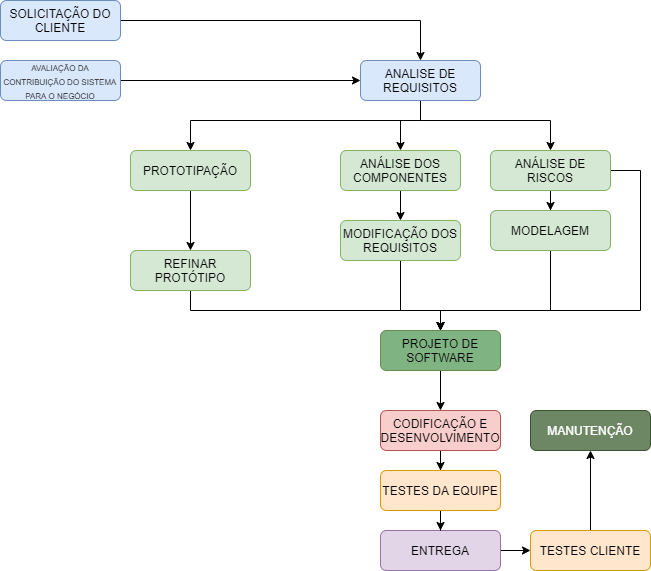
· Esse segundo modelo será então um Modelo de Processo Universal.

· Represente esse segundo modelo por meio de um diagrama do tipo “Workflow”.

· Esse modelo permite fluxos paralelos, loops, decisões etc. facilitando a criação de opções para produtos simples e complexos, com e sem prototipação, com uma única iteração ou quantas se fizerem necessário, etc.

De acordo com os modelos e suas respectivas etapas analisadas, foi possível criar um Modelo de Processo Universal que tenta abranger o máximo de atividades detectadas nos ciclos de vida estudados, e tenta lidar com praticamente todas as preocupações tratadas.

Abaixo está o nosso Modelo de Processo Universal representado através do diagrama WorkFlow:



O modelo representado acima, foi criado com base em diferentes caminhos que a empresa ou quem estiver utilizando-o poderá seguir e adotar. Atividades que são comuns em todos os modelos estudados, são apresentadas também neste novo modelo, e a partir das que são distintas, foi criado diversos caminhos, dando a opção de escolha, dependendo do projeto que será desenvolvido e da maneira que gostaria e que seria mais prática para a construção do mesmo.

Na fase inicial, o utilizador do modelo pode escolher se prefere dar ênfase para executar uma solicitação do cliente ou dependendo do modo em que trabalha, analisar se o projeto a ser adquirido vale ou não a pena para seu negócio.

Em seguida o modelo passa por uma frase essencial, que é a análise de requisitos, onde realiza-se uma identificação e análise dos requisitos e conceitos do novo projeto a ser tratado, com foco no usuário, no processo e nas restrições e riscos, que são validados juntamente com o usuário, assim como as ferramentas utilizadas, metodologia e técnicas.

A partir desse momento, há uma opção para se escolher como irá funcionar e qual caminho irá seguir para a construção e desenvolvimento do projeto. Há três caminhos para se escolher, o de prototipação, que são feitos protótipos e realizado testes e refinamentos no próprio, antes da construção em si do projeto de software. Outra opção é seguir o modelo de análise dos componentes, que se avalia quais componentes já existentes, poderão ser reutilizados para fornecer parte das funcionalidades do produto. A partir desse estudo, será redefinido os requisitos na etapa de Modificação, de maneira a se adequar com os componentes encontrados. E a última opção disponibilizada no modelo é a de análise de riscos, que consiste na verificação dos pontos críticos que possam vir a apresentar não conformidade durante a execução de determinado projeto ou atividade. E a partir desse momento, é realizado a modelagem do projeto que é a construção de modelos para comunicar a estrutura e o comportamento desejado do sistema, visualizar e controlar a arquitetura do mesmo e compreender melhor o sistema que está sendo elaborado. Essa fase do modelo, não precisa ser seguida obrigatoriamente, ela pode não ser realizada e assim continuar para a atividade seguinte do modelo.

Seguindo adiante, temos a fase do projeto de software, onde são definidos os componentes de softwares que serão utilizados, as funções contidas no produto, mantendo o foco nos dados e no projeto arquitetural do produto de software. Assim como também é definida a prototipação das interfaces para o usuário validar. Fase essencial e necessária para os modelos.

Em seguida temos a parte do desenvolvimento e codificação em si do projeto. E que também são definidas as linguagens e sintaxe utilizadas para a construção do projeto, e definidas suas funcionalidades que serão estarão disponíveis no produto.

Após isso, a equipe realiza o seus testes, analisando todas as funcionalidades que foram implementadas no produto, e se estão totalmente responsivas assim como o cliente desejou, para então realizar a entrega do produto para o cliente, que realizará testes e irá utilizar o seu produto, podendo dar o seu feedback, sugerindo melhorias, correções, mudanças, adaptações e outras escolhas que serão implementadas na fase de manutenção do modelo, onde é a fase mais longa dos ciclos.

**13.4 Responda as questões:**

1) Qual dos paradigmas apresentados é o mais eficiente? Explique.

Não se pode afirmar logo de início qual o melhor ou mais eficiente dos paradigmas discutidos, pois essa classificação depende muito do propósito do projeto,se os desafios se situam na interface com o usuário ou na integração entre subsistemas, e como a empresa deseja trabalhar em cima dos modelos existentes, e quais serão as fases que serão adotadas e implementadas, desde o início, adesão do projeto, até a entrega do produto final.

Como dito anteriormente não existe um paradigma perfeito, mas sim o paradigma que melhor atende a necessidade e a maneira em que se deseja trabalhar e desenvolver um projeto. Deve-se prestar atenção na quantidade de fase, pois indica para que tipo de projeto o modelo é direcionado. É interesse começar com modelos que são adaptativos, exemplo disso é o modelo evolutivo que se desenvolve através da interação com o cliente. Desta forma, se torna mais fácil na hora de de implementar o modelo pois em caso de mudanças extremas do cliente ainda sim pode-se fazer adaptações para alcançar os objetivos.

2) Quais paradigmas apresentados são adequados ao desenvolvimento baseado em componentes (COTS)?

O desenvolvimento de sistemas através de componentes consiste em criar o software utilizando componentes pré-construídos, não precisando assim partir do zero. Duas consequências diretas dessa definição são: a possibilidade de aumentar a produtividade da equipe e de reduzir os custos do projeto.

Entretanto, a utilização de componentes requer uma nova forma de pensar da equipe do projeto. Arquitetos, analistas e programadores devem preocupar-se agora com a integração do sistema com os componentes que serão utilizados, necessitando assim de uma atenção maior na definição de interfaces internas e externas. Esses pontos podem gerar mudanças no processo de desenvolvimento utilizado.

Existem dois modelos de paradigmas que se adequam ao desenvolvimento baseado em componentes. **O modelo Espiral Para Desenvolvimento** e o **Modelo RUP (Rational Unified Processing)**

3) Quais os critérios para a escolha de um paradigma, face a uma situação real de desenvolvimento de um produto de software?

Há muitos critérios na escolha de um paradigma, pois temos que levar em consideração muitas decisões da empresa. O modo em que ela deseja realizar todas as partes do desenvolvimento do projeto em questão, e um dos principais critérios antes de se escolher um modelo para seguir, podemos dizer que são: tempo de entrega , a taxa e feedback que deseja ser recebida pelo cliente, como a solicitação do cliente deseja ser feita, que se definem em reuniões periódicas com o cliente para definição de funcionalidades contidas no produto, avaliação e teste do cliente; Como a empresa deseja manter o cliente atualizado sobre como está ficando o produto e se há algo que precisa alterar, acrescentar ou remover, isso pode ser realizado em maneiras diferentes, apresentando para o cliente e mantendo-o atualizado a cada versão do produto construída, ou se a avaliação do cliente será solicitada apenas quando o produto estiver completamente finalizado.

Depende também se a empresa deseja realizar prototipação ou modelagem do produto antes da construção em si do mesmo. Existem muitos modelos diferentes e variados processos e fases adotadas até a finalização do projeto, e a escolha de algum, depende muito de como a empresa deseja realizar o desenvolvimento ao todo do produto.

* Modelo Waterfall (cascata):

Modelo didático que divide o ciclo de vida de 5 a 12 fases.

**Adequação:**

Projetos grandes (cobre todas as fases do ciclo de vida),

Os requisitos estão claramente definidos no início do desenvolvimento

Complexidade baixa

Riscos técnicos e de projeto bem entendidos.

* Modelo Incremental:

É uma adaptação (para melhor) do modelo de desenvolvimento linear, pois trata-se de um arranjo de vários pequenos ciclos em cascata.

A versão do produto de software tem uma nova funcionalidade (incremento), definida no início do ciclo. Depois de desenvolvida, cada versão será entregue ao usuário para testes. Cada versão implementa uma nova funcionalidade (incremento) que é acrescida à funcionalidade anterior.

* Modelo Evolutivo ou Desenvolvimento Exploratório:

Tem como objetivo promover um desenvolvimento conjunto (desenvolvedor trabalhando junto com o usuário) a fim de descobrir os requisitos de maneira incremental.

Diferente do modelo incremental, esse modelo volta sempre à fase de definição de requisitos, num movimento exploratório de conceitos do produto e necessidades do usuário.

Pode ampliar e/ou modificar as funcionalidades anteriores

* Modelo de Prototipação ou Protótipo Descartável:

Tem como objetivo obter uma melhor definição dos requisitos do sistema.

Diferentemente do Modelo Incremental, que faz uma exploração para a definição dos requisitos enquanto constroem o produto final, a prototipação exploratória usa os protótipos apenas para descobrir a funcionalidade desejada e os riscos implícitos.

Depois de definido os requisitos, o protótipo é descartado e inicia-se a construção do produto final.

* Modelo Baseado em Componentes

Na maioria dos casos de desenvolvimento de software, pensa-se, informalmente, em incorporar algum componente já desenvolvido anteriormente. Esta prática acelera o desenvolvimento do software.

Nos últimos anos, com a intensificação desta prática emergiu o modelo CBSE (Component-Based Software Engineering ou Engenharia de Software Baseada em Componentes).

A CBSE é centrada no reuso de componentes de software e tem sido cada vez mais utilizada. O desenvolvimento baseado em componentes, é uma consequência da adoção da Orientação a Objetos.

* Modelo Espiral (tradicional)

O Modelo Espiral (Boehm, 88) dá início a uma nova tendência na modelagem, através de ciclos repetitivos em espiral.

Engenharia de Software Ciclos de Vida de Software Modelo Espiral No modelo espiral tradicional, existem quatro quadrantes, onde se executa prioritariamente as seguintes tarefas: 1o quadrante- Planejar 2o quadrante- Analisar riscos 3o quadrante- Construir 4o quadrante- Avaliar.

A espiral inicia de um centro onde o movimento ainda é pouco acentuado (o produto está em sua versão inicial; é apenas um cerne do que será o produto final).

Com o passar dos ciclos, o movimento fica maior em cada quadrante, o que representa bem o fato de que, à medida que a espiral cresce, o trabalho e/ou a complexidade do produto de software aumenta.

* Modelo Espiral Atualizado

1. Passou de 4 para 6 quadrantes: • Solicitação do cliente • Planejamento • Análise de risco • Modelagem • Construção, teste, instalação e suporte para o cliente testar • Avaliação do cliente.

Outra alteração do modelo espiral atualizado em relação ao tradicional é a divisão do curso do desenvolvimento (curva) em 4 tempos (espiral dividida em quatro tempos, que definem quatro objetivos diferentes para o trabalho), que são:

• Projeto de concepção de desenvolvimento

• Projeto de desenvolvimento do novo produto

• Projeto de melhoria do produto

• Projeto de manutenção do produto

Como cada fase tem um objetivo diferente, gera um produto diferente no seu término (vide figura a seguir)

* Modelo Espiral para Desenvolvimento Baseado em Componentes

Trata-se de uma variação do modelo de desenvolvimento em espiral atualizado, onde as fases 4 e 5 (modelagem e construção-teste-instalação-suporte para teste) tem duas possíveis formas de desenvolvimento:

1. Identificar componentes candidatos:

Identificar componentes candidatos (produtos de software que tem a funcionalidade requerida)

Procurar o componente (bibliotecas, outros projetos, mercado, internet etc)

2. Obter e disponibilizar componentes:

Caso não existam componentes prontos, a fase será desenvolvida como no modelo espiral adaptado (neste ciclo da espiral), com a construção do componente necessário.

Se existirem componentes, eles serão recuperados e catalogados na biblioteca do projeto. Em seguida serão aplicados na construção do produto, encerrando essa fase.

O ciclo continua na fase de avaliação do cliente.

* Modelo RUP (rational unified processing)

Desenvolvimento de software a partir de um framework genérico, que pode ser especializado para diferentes aplicações, organizações, níveis de competência e tamanho de projetos.

Características:

Baseado em componentes reutilizáveis

Usa UML para modelar o produto

Dirigido a casos de uso

Centrado na arquitetura

Iterativo e Incremental