



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA



Mestrado Profissional em Ciência da Computação

Vídeo Locadora Imperial

Relatório final

Disciplina: Engenharia de Software

Professor: Vinicius Cardoso Garcia

Equipe:

lnff@cin.ufpe.br

maos2@cin.ufpe.br

lmq@cin.ufpe.br

rhs@cin.ufpe.br

ocsf@cin.ufpe.br

rmva@cin.ufpe.br

Recife, 14 de fevereiro de 2019

Sumário

1. INTRODUÇÃO	4
2. GERÊNCIA DE CONFIGURAÇÃO E AMBIENTE	5
2.1 Ferramentas de desenvolvimento	5
2.2 Linguagem utilizada	5
2.3 Banco de dados	5
2.4 Bibliotecas	5
2.5 Repositórios de gerenciamento de versões e mudanças	5
2.6 Servidores de implantação	5
2.6.1 Configurações do servidor	5
3. CARACTERÍSTICAS DA APLICAÇÃO	6
3.1 Elicitação das características da aplicação	6
3.2 Funcionalidades da aplicação	6
3.3 Diferenciais	7
4. VISÃO DE ANÁLISE E PROJETO (ARQUITETURA)	8
4.1 Especificações de casos de uso	8
4.1.1 User Store – Locação	8
4.1.2 User Store - Devolução	10
4.2 Diagramas de apoio ao projeto	11
4.2.1 DER	11
4.2.2 Diagrama Lógico	12
4.3 Visão de Implantação	14
4.3.1 Visão e instruções da instalação do sistema	14
4.3.2 Um release notes com os erros, falhas e etc.	14
4.3.3 Endereço de implantação da aplicação	15
4.3.4 Vídeo de demonstração promovendo o produto (youtube).	15
4.4 Visão de Uso	15
Ponto de Vista do Gerente / Funcionário	15
4.4.1 Para efetuar o Cadastro de Clientes Titular	15
4.4.2 Para efetuar o Cadastro de Clientes Dependente	15
4.4.3 Para efetuar o Cadastro de Distribuidora	15
4.4.4 Para efetuar Locação	16
4.4.5 Para efetuar Pagamento	16
4.4.6 Para efetuar Cadastro de Filmes	16
4.4.7 Para efetuar Cadastro de Endereço	16
4.4.8 Para efetuar Cadastro de Tipo de Item	16
4.4.9 Para efetuar Cadastro de Tipo de Pagamento	17
Ponto de Vista do Cliente	17

4.4.10 Para efetuar Reserva	17
5. Revisão do Projeto	17
5.1 Descrição do processo de desenvolvimento, principais problemas e tomadas de decisão em relação ao projeto.	17
5.2 Revisão Individual	20
5.2.1 Rafaela Helena Santos - [rhs] [Scrum Master]	21
5.2.2 Rogério Mangabeira Vicente de Aquino - [rmva] [Dev Team]	21
5.2.3 Osmario de Carvalho Santos Filho - [ocsf] [Dev Team]	21
5.2.4 Mônica Auricélia Oliveira Santana - [maos2] [Product Owner]	22
5.2.5 Lilian Maria Gonçalves - [lmg] [Dev Team]	22
5.2.6 Leonardo Navarro F. Freire - [lnff] [Dev Team]	23
6 Referências Bibliográficas	24

1. INTRODUÇÃO

O uso das tecnologias digitais tornou-se cada vez mais comuns no dia a dia das pessoas, logo as empresas sentiram a necessidade de atualizar-se para se manter inseridas em um mercado cada vez mais competitivo. Para acompanhar essa tendência, a vídeo locadora Imperial percebeu a necessidade da informatização dos seus processos, sendo extremamente importante o desenvolvimento de um sistema com a implantação em uma plataforma web e que ficasse disponível 24 horas por dia aos seus clientes. Disponibilizando rotinas como cadastro de clientes, reservas, locações, disponibilidade de catálogos. Com isso a vídeo locadora Imperial passa a ter mais agilidade nas suas atividades, um maior controle sobre seus ativos e disponibilidade das informações.

Para o desenvolvimento deste sistema foi elaborado um projeto que segue as técnicas de engenharia de software (documentação, análise, projeto, implementação e implantação), sendo apresentado informações do andamento das etapas aos seus stakeholders, reduzindo os riscos e aumentando a satisfação do cliente.

A proposta deste sistema é funcionar 24 horas por dia, contendo os requisitos solicitados pelo cliente, para otimizar os serviços e beneficiar o processo de locação, devolução e reservas de filmes com maior rapidez e eficiência. O sistema conta com a implantação de um ambiente seguro para manipulação das informações, reduzindo custos com retrabalhos e insumos, além de aumentar a produtividade com o seu manuseio.

O projeto é constituído da solução construída de acordo com a proposta e todos os artefatos que a compõem, versionada e compartilhada por meio de um repositório no GitHub onde será possível fazer a rastreabilidade do ciclo de vida. O relatório técnico final contém a contextualização, escopo, tomadas de decisão, documentação da arquitetura da solução e projeto de implantação para melhor transparência do trabalho que foi realizado.

2. GERÊNCIA DE CONFIGURAÇÃO E AMBIENTE

Descrição das ferramentas e ambientes de desenvolvimento utilizados.

2.1 Ferramentas de desenvolvimento

As seguintes ferramentas de desenvolvimento foram utilizadas: IDE Netbeans, JSF, Eclipse Mars, Maven e Tomcat 9.0

2.2 Linguagem utilizada

JAVA: Versão 1.8_201

2.3 Banco de dados

Postgresql: versão 11 no servidor e 9.6 para desenvolvimento.

2.4 Bibliotecas

Java PrimeFaces, JSF, JDBC e XHTML.

2.5 Repositórios de gerenciamento de versões e mudanças

O GitHub foi utilizado por ser uma plataforma de hospedagem de código-fonte com controle de versão assim permitindo que todos os membros da equipe pudessem contribuir no projeto.

2.6 Servidores de implantação

Foi utilizado o ambiente computacional do Instituto Federal do Mato Grosso para hospedagem do sistema.

2.6.1 Configurações do servidor

Servidor com as seguintes configurações:

IDE: Eclipse Mars JAVA: Versão 1.8_201 Tomcat: 9.0 Postgresql: versão 11 no servidor e 9.6 para desenvolvimento.

3. CARACTERÍSTICAS DA APLICAÇÃO

3.1 Elicitação das características da aplicação

A partir dos requisitos fornecidos pelo cliente se obteve dados necessários para construção do sistema e o desenvolvimento do projeto.

3.2 Funcionalidades da aplicação

- Cadastrar cliente

Permitir o armazenamento das informações de seus clientes e dependentes para que possam efetuar reservas, locações e devoluções.

- Cadastrar Filmes

Armazenar as informações de Filmes a serem disponibilizados para os clientes.

- Cadastrar Itens

Permitir o armazenamento das informações dos itens pertencentes aos filmes.

- Cadastrar Distribuidora

Permitir o armazenamento das informações das distribuidoras que fornecem os filmes, para posterior consulta e atualização dos dados.

- Cadastrar Reserva

Quando houver item disponível para reserva, permitir o armazenamento informações de reservas de seus acervos, gerenciando as solicitações e desejos de locação de seus clientes.

- Registrar locação

Permitir o controle e armazenamento da locação realizada e gerenciar a saída de itens para controle do acervo da locadora.

- Registrar devolução

Permitir o controle e armazenamento da devolução realizada e gerenciar a entrada de itens para controle do acervo da locadora.

- Realizar o controle financeiro

O sistema possibilita o controle financeiro de todos os pagamentos efetuados pelos clientes, sendo possível armazenar informações como: valor, tipo de

pagamento realizado, cobrança de multa e a garantia que o pagamento será realizado na locação ou na devolução do filme.

- Diferenciar o valor da locação de acordo com o tipo de filme

O sistema deve alterar o valor da locação de acordo com o tipo de filme se lançamento ou catálogo.

- Alterar valor e prazo de locações

O sistema deve possibilitar a alteração do valor e prazo de locações de acordo com a política da empresa.

- Expirar reserva

A reserva expira quando passadas mais de 24h de sua comunicação com o cliente através de e-mail.

- Impedir reservas e locações para clientes inadimplentes

Clientes inadimplentes não podem efetuar locações ou reservas.

- Controle de acesso

O sistema deve controlar o acesso das funcionalidades do sistema de acordo com cada perfil informado.

- Utilizar dispositivo de leitura de código de barras

Para agilizar o atendimento ao cliente.

- Realizar pagamento com cartão de crédito

Permitir que o cliente possa realizar o pagamento com cartão de crédito.

- Realizar consultas

O cliente ou o atendente da locadora poderão efetuar consultas com o uso do auto completar, evitando a digitalização desnecessária de informações.

3.3 Diferenciais

Software desenvolvido com tecnologia de aplicações Web e possuindo uma customização da aplicação com base nos requisitos do cliente, que facilita a usabilidade e recursos otimizados para ganho de performance.

4. VISÃO DE ANÁLISE E PROJETO (ARQUITETURA)

4.1 Especificações de casos de uso

4.1.1 User Store – Locação

Registrar locações
Como atendente da locadora, Quero registrar locações, indicando o cliente e os itens locados, bem como a data e o valor da locação e a data da devolução prevista de cada item, Para que eu possa manter o histórico da locação realizada e gerenciar a saída de itens para controle do acervo da locadora.

Figura 1

- Locação de Itens (Filmes)

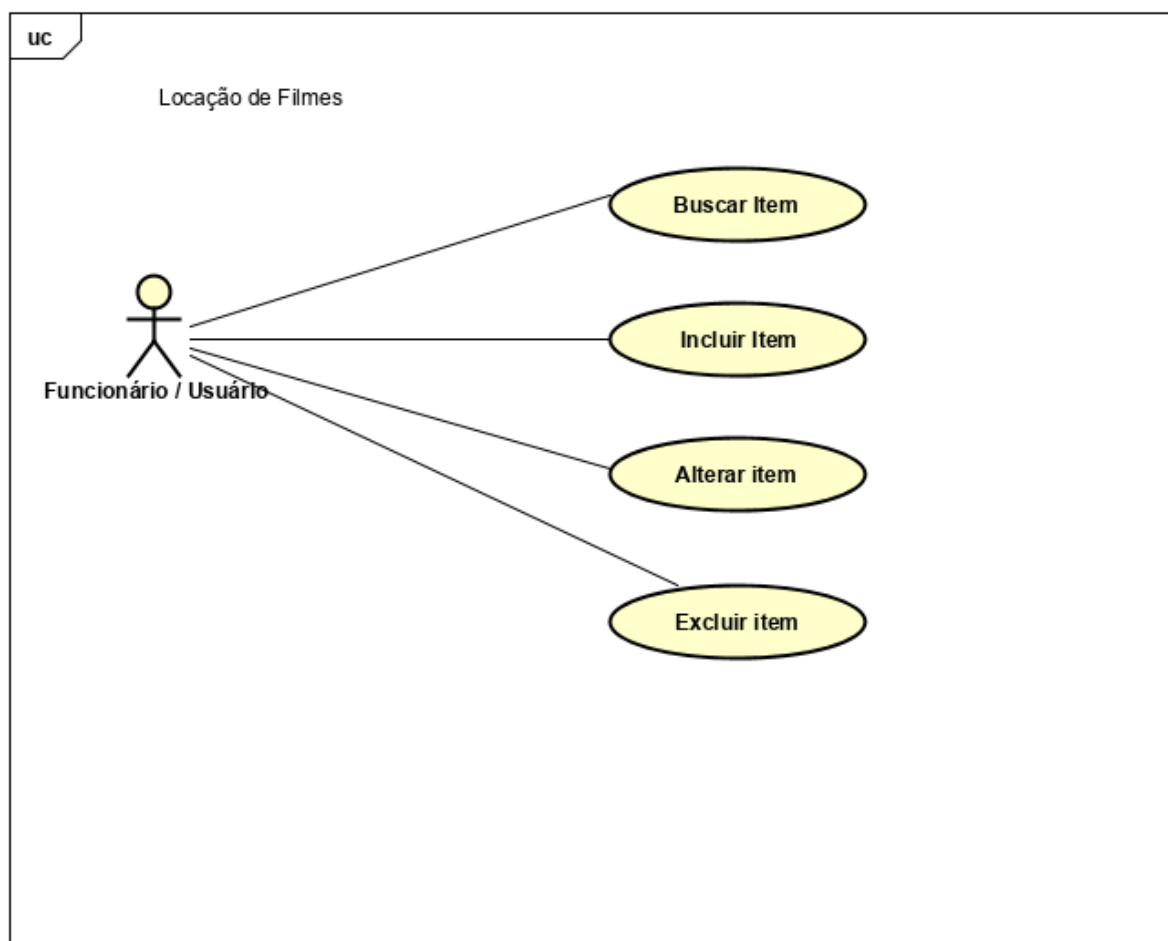


Figura 2

Nome da Use Case: Buscar Item

Ator: Atendente

Descrição: Durante a locação o cliente poderá solicitar a busca de um item.

Cenário:

- O cliente informa o Filme que deseja locar;
- O atendente realiza a busca pelo Item (Dvd, Blu-ray) do filme disponível;
- O atendente após achar o Item daquele filme poderá realizar a inclusão.

Nome da Use Case: Incluir Item

Ator: atendente

Descrição: Durante a locação o cliente poderá solicitar a inclusão de item.

Cenário:

- O Cliente informa qual filme deseja locar;

- O atendente realiza a busca pelo item daquele filme escolhido, e inclui caso esteja disponível;
- O atendente poderá incluir mais de um item, caso o cliente deseje;
- O atendente, após inclusão, poderá finalizar a locação

Nome da Use Case: Alterar Item

Ator: atendente

Descrição: Durante a locação o cliente poderá realizar alteração de itens.

Cenário:

- O Cliente informa qual filme deseja alterar;
- O atendente realiza a busca pelo item daquele filme escolhido, e realizar alteração;
- O atendente, após alteração, poderá finalizar a locação

Nome da Use Case: Excluir Item

Ator: atendente

Descrição: Durante a locação o cliente poderá solicitar a exclusão de um ou mais itens.

Cenário:

- O atendente poderá realizar a buscar ou selecionar o item ou itens do filme que o cliente deseja excluir.

4.1.2 User Store - Devolução

Registrar devolução
Como atendente da locadora, Quero registrar as devoluções, indicando os itens devolvidos e a data de devolução. Para que eu possa gerenciar a entrada de itens para controle do acervo da locadora.

Figura 3

- Devolução de Itens (Filmes)

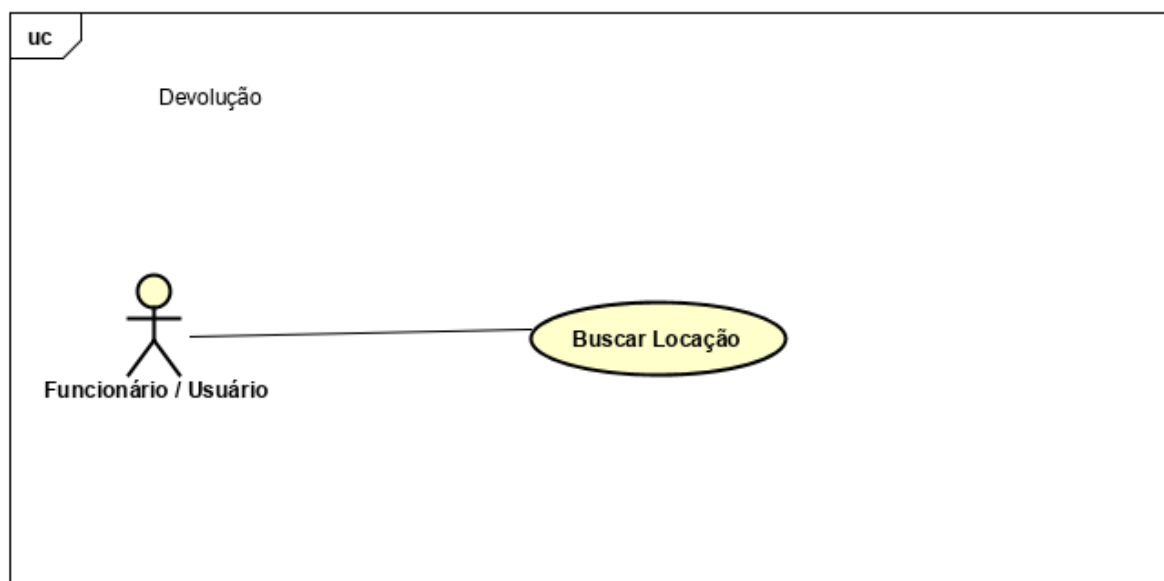


Figura 4

Nome da Use Case: Devolução Item ou itens

Ator: atendente

Descrição: O atendente poderá realizar a devolução do item ou itens

Cenário:

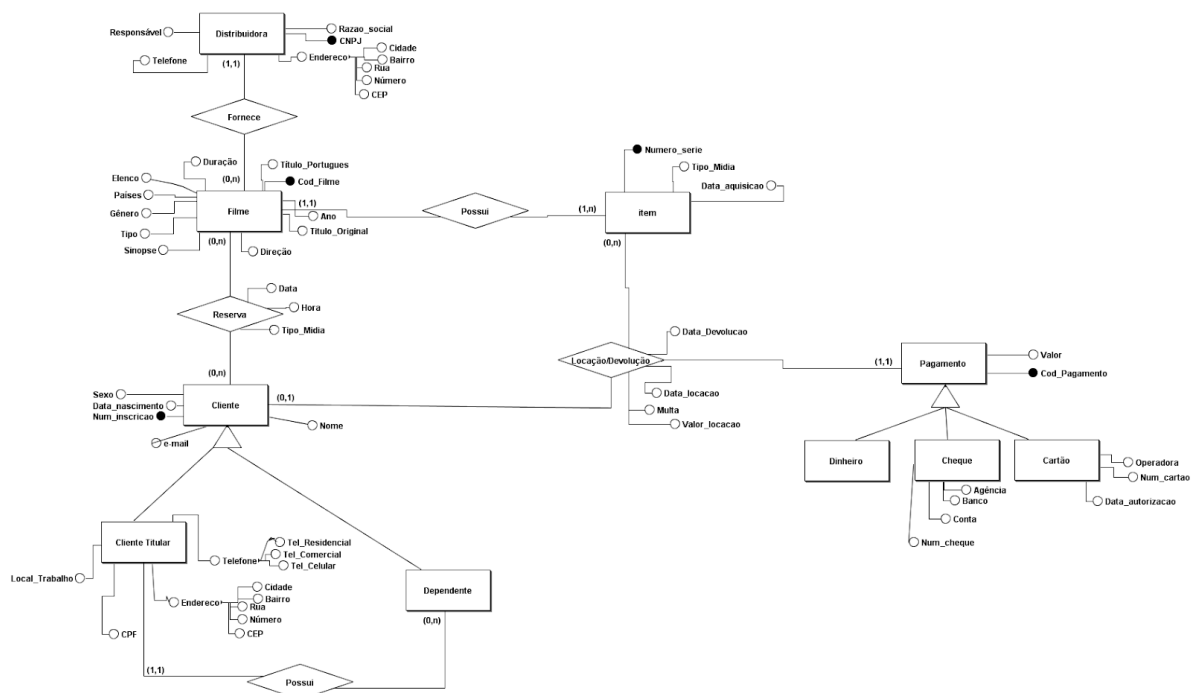
- O Cliente devolve o item ou os itens;
- O atendente realiza a busca da locação feita pelo cliente;
- O atendente após localizar ou buscar, verifica se existe algum atraso, caso sim, solicita que o cliente efetue o pagamento atrasado, caso contrário, o atendente dar baixa nos itens e finaliza a devolução.

4.2 Diagramas de apoio ao projeto

4.2.1 DER

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos) (RODRIGUES,2014).

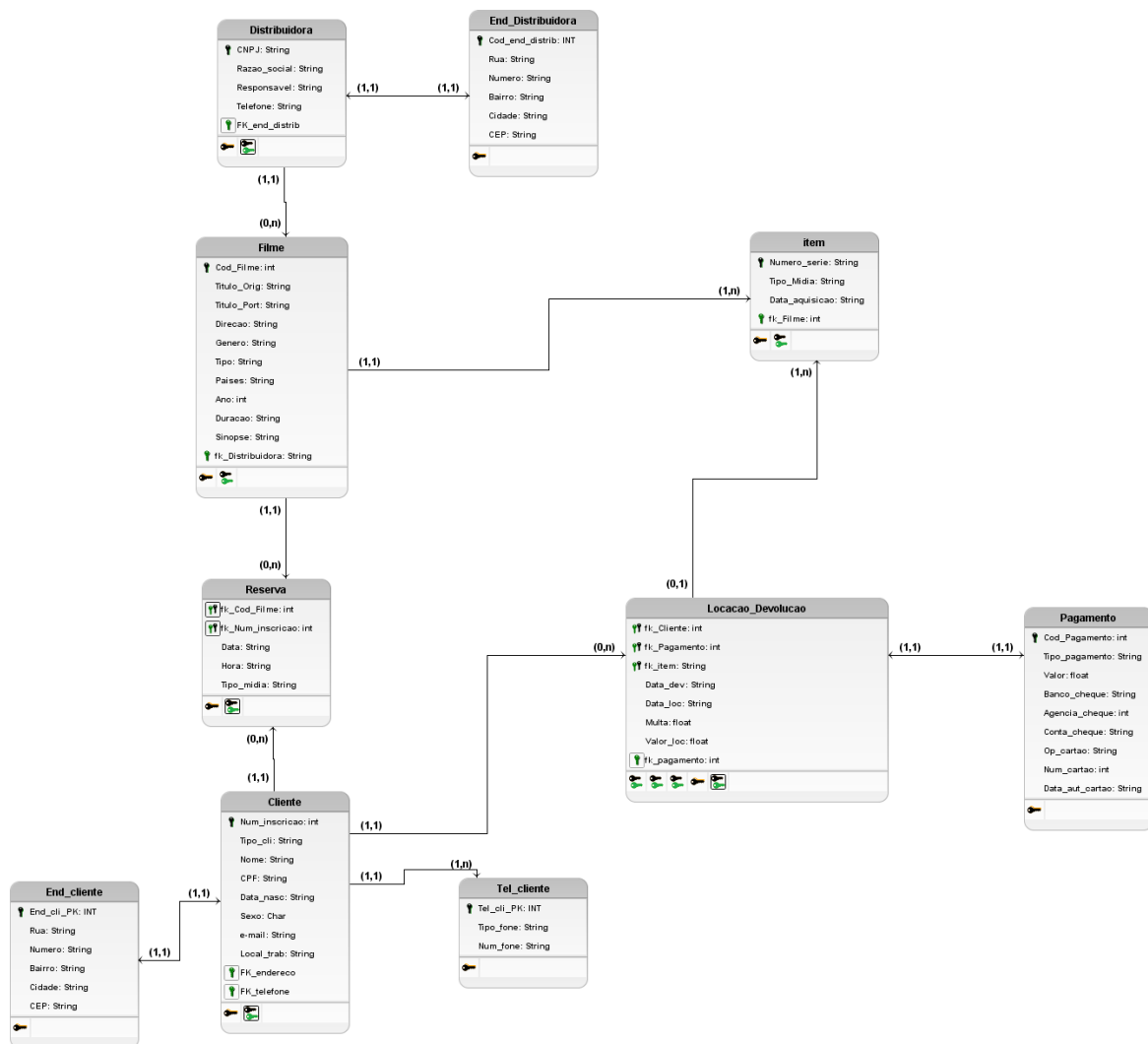
Segue abaixo o DER do sistema de Vídeo Locadora Imperial:



4.2.2 Diagrama Lógico

Compreende uma descrição das estruturas que serão armazenadas no banco e que resulta em uma representação gráfica dos dados de maneira lógica, inclusive nomeando os componentes e ações que exercem uns sobre os outros (LOPES, 2012).

Segue abaixo o diagrama lógico do sistema de Vídeo Locadora Imperial:



4.3 Visão de Implantação

4.3.1 Visão e instruções da instalação do sistema

..		
settings	gravando no banco, mas falta só o retorno. Finalmente. Sem Hibernate	7 days ago
WebContent	Correção de erro	2 hours ago
bin	Atualização index com menu	10 days ago
src	Tipos de Pagamentos e Tipos de Mídia	3 hours ago
target	Atualização da aplicação - montagem dos pagamentos e tipos de mídia	3 hours ago
.classpath	Alteração do Projeto	7 days ago
.gitignore	Atualização index com menu	10 days ago
.project	gravando no banco, mas falta só o retorno. Finalmente. Sem Hibernate	7 days ago
.tern-project	Atualização index com menu	10 days ago
pom.xml	mudança do pom.xml	5 days ago
videolocadoraeng.sql	Atualização da aplicação - montagem dos pagamentos e tipos de mídia	3 hours ago

O projeto da Vídeo Locadora Imperial foi implantado em um servidor localizado no Instituto Federal do Mato Grosso, com disponibilidade web 24 horas, o código fonte foi armazenado no github e no servidor web, sendo compostos por diversos arquivos e diretórios. Sendo observados na implantação alguns requisitos como disponibilidade, desempenho, escalabilidade e confiabilidade.

4.3.2 Um release notes com os erros, falhas e etc.

Durante o desenvolvimento do projeto foram encontrados erros, incompatibilidades e dificuldades, tais como: erro de persistência de relacionamento, servidor não realizando inserção no banco que para ser consertado teve que ser ajustada a versão e a senha.

Alguns códigos gerados pela ferramenta Netbeans causavam inconsistências, sendo feito os ajustes para correção. Dificuldade também em acertar versão das bibliotecas para não haver incompatibilidade entre as ferramentas.

Devido o grande desgaste, foi feito uma consulta com outros desenvolvedores (Thiago, Raimundo Fagner e o professor da disciplina - Vinícius Garcia) e foi feita a alteração da estrutura do projeto e resolveu-se não se utilizar

mais o Hibernate e utilizar JDBC para persistência no banco. Além da aplicação utilizada na capacitação, foi utilizado o Maven para controle de dependências.

Não foi utilizado o Heroku, como pretendido anteriormente, devido ainda a incompatibilidade de bibliotecas, que ao fazer o build da aplicação no eclipse, ainda há problemas de compatibilidade de versionamento do JAVA.

4.3.3 Endereço de implantação da aplicação

<http://svnproen.ifmt.edu.br:80/vimperial/>

4.3.4 Vídeo de demonstração promovendo o produto (youtube).

Segue Link para assistir o vídeo demonstrativo: <https://youtu.be/ww0Aav4xaMo>

4.4 Visão de Uso

Ponto de Vista do Gerente / Funcionário

4.4.1 Para efetuar o Cadastro de Clientes Titular

- Cadastro >> Cliente

O Cliente Titular deverá informar ou fornecer as informações:

(Nome, Cpf, Data de Nascimento, e-mail, sexo, Endereço, telefone residencial, telefone comercial, local de trabalho)

4.4.2 Para efetuar o Cadastro de Clientes Dependente

- Cadastro >> Dependente

O Cliente Dependente deverá informar ou fornecer as informações:

(Nome, tipo, e-mail, selecionar cliente)

4.4.3 Para efetuar o Cadastro de Distribuidora

- Cadastro >> Distribuidora

A Distribuidora deverá informar ou fornecer as informações:

(Razão social, CNPJ, responsável, endereço e telefone)

4.4.4 Para efetuar Locação

- Locação >> Locação

O Gerente ou Funcionário deverá buscar ou fornecer as informações:

(Número de inscrição do cliente, tipo de item, data de locação e devolução, valor, multa e forma de pagamento)

4.4.5 Para efetuar Pagamento

- Pagamento >> Pagamento

O Gerente ou funcionário vai buscar informações da locação e cliente vai informar as informações de pagamento.

(Locação, tipo de pagamento, valor, valor recebido, troco)

4.4.6 Para efetuar Cadastro de Filmes

- Cadastro >> Filmes

O funcionário deverá informar ou fornecer as informações:

(Código do Filme, Título Original, Título português, direção, gênero, tipo, países, ano, duração, distribuidora, sinopse)

4.4.7 Para efetuar Cadastro de Endereço

- Cadastro >> Endereço

O Gerente ou Funcionário para realizar cadastro de endereço dos seus clientes precisa das seguintes informações:

(Endereço)

4.4.8 Para efetuar Cadastro de Tipo de Item

- Cadastro >> Tipo de Item

O gerente ou funcionário para registrar o cadastro precisa que das seguintes informações:

(Tipo de Item)

4.4.9 Para efetuar Cadastro de Tipo de Pagamento

- Cadastro >> Tipo Pagamento

O gerente ou funcionário para registrar o tipo de pagamento precisa que o usuário informe as seguintes informações:

(Tipo de Pagamento)

Ponto de Vista do Cliente

4.4.10 Para efetuar Reserva

- Reserva >> Reserva

O cliente e o sistema deveram buscar ou fornecer as informações:

(Código Filme, número de inscrição do cliente, data e hora e tipo item)

5. Revisão do Projeto

5.1 Descrição do processo de desenvolvimento, principais problemas e tomadas de decisão em relação ao projeto.

Processos de desenvolvimento rápido de software são projetados para criar software útil rapidamente. Geralmente, eles são processos iterativos nos quais a especificação, o projeto, o desenvolvimento e o teste são intercalados. O software não é desenvolvido e disponibilizado integralmente, mas em uma série de

incrementos e, cada incremento inclui uma nova funcionalidade do sistema (SOMMERVILLE, 2007).

Para gerenciamento do desenvolvimento do projeto utilizamos a metodologia ágil Scrum, sendo o trabalho dividido em iterações ou sprints, definindo os papéis de cada membro da equipe e as tarefas que iriam ser realizadas em cada sprint. Utilizamos alguns softwares como BrModelo para criação dos diagramas, Pencil na prototipação de telas e Astah Professional para caso de uso. Também foi utilizado o Google Drive para compartilhar arquivos e organizar os documentos do projeto.

O Scrum fornece uma maneira para que as equipes trabalhem juntas para desenvolver um produto. O desenvolvimento de produtos, utilizando Scrum, ocorre em pequenos pedaços, com cada peça sobre peças criadas anteriormente. A construção de uma peça pequena de cada vez estimula a criatividade e permite que as equipes correspondam ao feedback e mudança, para construir exatamente e somente o que é necessário (SCRUM.ORG, 2015).

Com a utilização da metodologia ágil Scrum é possível reduzir o prazo de entrega e permitir um melhor acompanhamento das atividades realizadas pela equipe. Nosso projeto foi desenvolvido em vários segmentos ou séries de incrementos, a fim de entregar as funcionalidades após testes intercalares.

Apesar dos benefícios com a inserção da metodologia Scrum no projeto, tornando mais fácil o gerenciamento e cronograma das atividades a serem desenvolvidas, tendo transparência no andamento do projeto e melhor envolvimento da equipe, houveram algumas dificuldades no início do projeto de entender como funcionava o Scrum na prática, sendo resolvido através de capacitação, estudo e análise por parte de toda a equipe. Logo iniciamos definindo os papéis de cada membro da equipe por afinidade e conhecimento. Após isso, tivemos problema como seria o andamento das nossas reuniões, após alguns testes com algumas ferramentas de videoconferência, por questões de conectividade não deu certo, decidimos então realizar as reuniões via whatsapp, através da participação dos membros por meio de envio de áudio, textos e outros.

Para a utilização do GitHub, fizemos um estudo da ferramenta pois se tratava de algo novo para todos os integrantes da equipe. Na abertura e fechamento dos issues, além das atribuições, classificações, tivemos dificuldade de entender o seu uso, mas após entendimento fizemos todo processo por ele. Erramos em alguns processos no início da sua utilização, mas logo começamos a entender o seu funcionamento.

No desenvolvimento dos documentos começamos a utilizar word e enviando arquivos para github, mas vimos a necessidade de realizar alterações nestes documentos e logo utilizamos o wiki para realizar a documentação, as revisões e leitura dos arquivos. A parte de documentação do projeto foi dividida em partes, sendo desenvolvido nas ferramenta do Github (Wiki), além do googledocs para colaboração e revisão dos documentos.

O desenvolvimento da aplicação começou com a escolha de qual IDE, linguagem, banco de dados levando em consideração os requisitos do cliente. No desenvolvimento do código foi utilizada a linguagem de programação Java e a hospedagem da aplicação foi realizada no servidor do Instituto Federal do Mato Grosso. Para atender melhor aos requisitos do usuário, criamos Users store, prototipação de telas, testes de BDD e TDD até a entrega dos artefatos.

Tivemos diversas reuniões com registros em ata para decidir, ver pendências e ter o acompanhamento do projeto. Foi feita uma documentação (post mortem) ao final de cada sprint com resumo do que foi feito e o planejamento da próxima sprint.

No desenvolvimento do projeto foi utilizada a ferramenta Netbeans como IDE, bem como JSF e banco de dados Postgresql. A escolha do Netbeans se deu devido sua facilidade no desenvolvimento, porém apesar disto, verificou-se que a alteração do código gerado pela ferramenta causava algumas inconsistências e isto fez com que o desenvolvimento atrasasse bastante. Como o membro escolhido para fazer o desenvolvimento, tinha capacitação inicial em JSF e Postgresql utilizando Eclipse, mudou-se a ferramenta justamente para se ter maior agilidade e reaproveitamento de aplicações utilizadas. Uma das primeiras dificuldades foi a

falta de experiência em desenvolver uma aplicação do início ao fim, visto que por mais que foram feitos investimentos em capacitação ao longo da vida profissional, nunca tinha sido feito o desenvolvimento de uma aplicação completa, mas sim coisas pontuais de manutenção e criação de módulos pequenos em aplicações desktop com ferramentas case (não havia a necessidade de criação de telas, códigos dos CRUDs - Create/Read/Update/Delete). Uma dificuldade encontrada foi acertar a versão das bibliotecas, visto que devido a incompatibilidade entre as ferramentas utilizadas - principalmente relacionada a versão do JAVA, que ocasionou um refazimento do projeto de forma demasiada. Com o auxílio de outros desenvolvedores, tutoriais e vídeo-aulas, conseguiu-se chegar a uma configuração que o projeto finalmente saiu do papel.

Atribuição das atividades e técnicas de gerenciamento, monitoramento e controle.

Como as atividades foram atribuídas conforme a experiência, conhecimento do membro da equipe, tivemos um grupo responsável pela documentação, implementações e uma pessoa na programação do sistema. Documentação foi divididas em partes, que pode ser observada pela rastreabilidade do GitHub, assim como programação foi também separada em etapas e criados os sprints dentro do modelo Scrum.

O Gerenciamento, monitoramento e controle foram feitos no (Git Hub- Kanban), onde pudemos realizar a organização do projeto e desenvolvimento por etapas: o DO (o que tinha para fazer), o DOING (o que estava fazendo) e o DONE (o que foi finalizado ou concluído). Foi criada a aba Sprint Backlog para demonstrar o que estava planejado para aquela sprint.

5.2 Revisão Individual

Lições aprendidas do ponto de vista individual dos membros do time em relação a execução do projeto em si.

5.2.1 Rafaela Helena Santos - [rhs] [Scrum Master]

O projeto em si foi bem desafiador no meu ponto de vista, pois como não sou da área de desenvolvimento, tenho pouco contato com as ferramentas que precisamos utilizar. A nossa escolha em misturar a metodologia Scrum com Kanban foi bem assertiva. Primeiramente tive que familiarizar com github e depois tudo foi ficando mais fácil. Era possível verificar rapidamente a situação do trabalho. Todas as tarefas necessárias estão expostas a todos, permitindo melhor comunicação e maior integração. Outra dificuldade que encontrei foi em marcar reuniões, por mais que fossem reuniões rápidas e pragmáticas, a equipe não possuía uma agenda compatível, a logística da equipe dificultou um pouco na interação, mas não atrapalhou, pois definimos bem os papéis o que ajudou bastante. Gostei muito de realizar esse projeto e tenho para mim que contribui bastante para minha vida profissional.

5.2.2 Rogério Mangabeira Vicente de Aquino - [rmva] [Dev Team]

As experiências vividas na prática durante o desenvolvimento da atividade na disciplina de engenharia de software, tornou-se um dos principais instrumentos para refletir sobre a necessidade constante dos estudos. As descobertas dos requisitos que funcionaram ou que não funcionaram durante a implementação das atividades permitem pensar sobre o que se fez e sobre o que se pode melhorar. Apesar de desafiador, é o desenvolvimento de práticas que possibilitam o amadurecimento enquanto pessoa, estudante ou profissional.

5.2.3 Osmario de Carvalho Santos Filho - [ocsf] [Dev Team]

Apesar de não ter pouca experiência na prática de desenvolvimento de software, o que tornou o projeto um desafio para entender como funciona todo seu ciclo de vida e a sua essência na prática. Achei muito interessante a utilização da metodologia ágil Scrum, com a divisão em ciclos (Sprint), a rastreabilidade dos processos ou atividades com ajuda controle de versões GitHub e com componente Kanban (DO, Doing, done) foi possível organizar toda sequência para criação dos Sprints. As reuniões foram importantes para alinhar o andamento do projeto e tirar

dúvidas com os membros da equipe. Foi possível entender algumas etapas de desenvolvimento mesmo sendo atribuídas para outros membros da equipe, enfim, o projeto foi muito desafiador e ao mesmo tempo educativo, sendo possível agregar novos conhecimento e que serão importantes para esta jornada do Mestrado Profissional e experiência pessoal na área de engenharia de Software.

5.2.4 Mônica Auricélia Oliveira Santana - [maos2] [Product Owner]

A primeira dificuldade enfrentada pela equipe foi a divisão das tarefas do projeto, pois tínhamos que quebrar em partes menores algo que, ao meu ver, era bastante extenso principalmente a parte de documentação. Solucionamos o problema dividindo a equipe em duas partes: uma responsável pela documentação e outra pelo desenvolvimento do sistema. Outra dificuldade encontrada foi a não familiaridade com Scrum por parte da equipe causando paradas no início do projeto para alinhar os conhecimentos a respeito da técnica. Nossa equipe tem a particularidade de que apenas um membro da equipe possuía experiência na área de desenvolvimento, o restante sendo da área de infraestrutura. A comunicação entre a equipe foi, na minha opinião, o maior gargalo do projeto. Devido a distância e incompatibilidade de agenda, era difícil marcar uma reunião onde todos pudessem participar ou até mesmo uma comunicação individual. Ressalto que, apesar das dificuldades enfrentadas, o projeto foi de grande aprendizado contribuindo para um riquíssimo aprendizado.

5.2.5 Lilian Maria Gonçalves - [lmg] [Dev Team]

O maior aprendizado foi a questão de desenvolver uma aplicação do zero até a entrega do cliente. A primeira experiência obtida foi a determinação de qual IDE, linguagem, banco de dados, sempre respeitando os requisitos do cliente. A outra experiência foi tentar utilizar ferramentas, que exigiu uma capacitação autodidata e que nem sempre dá os resultados esperados suficientes para cumprir os prazos estabelecidos, sendo necessário sempre se basear nos conhecimentos já adquiridos antes em capacitações. Outro aprendizado foi a incompatibilidade entre as bibliotecas que causou um grande desgaste mental e físico e que a falta de experiência refletiu nesta hora. Mesmo com auxílio de outras pessoas, não foi

alcançado o resultado esperado. Ou seja, foi aplicada todas as disciplinas foram aplicadas em um ambiente de produção real, onde foi observada que as capacitações que fazemos no mercado não nos preparam de forma completa e o trabalho em grupo - com contato com a comunidade e/ou auxiliado por pessoas de maior experiência é fundamental para o andamento de projetos que temos no dia a dia.

5.2.6 Leonardo Navarro F. Freire - [Inff] [Dev Team]

Acredito que dentre todas as dificuldades encontradas durante o desenvolvimento da atividade, o “tempo” foi um dos mais difíceis de mensurar, logo a experiência da equipe na área de desenvolvimento de software, deixa uma pouco a desejar, devido a área de atuação profissional de cada um, dificultando a atividade de estabelecer prazos para etapas de desenvolvimento, mas que de toda maneira nos serviram de ensinamentos.

6 Referências Bibliográficas

BECK, K. (2008) Programação Extrema (XP) Explicada. Porto Alegre: Bookman.

LOPES, Abraão. Modelo Conceitual, Lógico e Físico, Entidade-Relacionamento. 2012 Disponível em: <http://docente.ifrn.edu.br/abrahamlopes/semestre-2012.1/4.401.1m-banco-de-dados/slides-modelo-conceitual-fisico-logico-er> Acesso em: 05/02/2019.

RODRIGUES, Joel Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). 2014. Disponível em: <https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332> Acesso em: 05/02/2019.

SCRUM.ORG. (2015) What is Scrum? Disponível em: <https://www.scrum.org> Acesso: 5 de Fevereiro de 2019.

SOMMERVILLE, I. (2007) Engenharia de Software. 8. ed. São Paulo: Pearson Addison Wesley.