



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA



Mestrado Profissional em Ciências da Computação



Vídeo Locadora Imperial

Relatório de Projeto

(Versão 1.0)

ÍNDICE

1.	INTRODUÇÃO	4
1.1	Motivação	4
1.2	Sobre a Organização	4
2.	OBJETIVOS	5
2.1	O Problema Identificado	5
3.	ESCOPO NEGATIVO	6
4.	IDENTIFICAÇÃO DE ATORES DO SISTEMA	7
4.1	ATORES E FUNÇÕES	7
4.1.1	Administradores	7
4.1.2	Atendentes	7
4.1.3	Clientes / Visitantes do web site	7
5.	METODOLOGIA	8
6.	EQUIPE DO PROJETO	11
6.1	MEMBROS	11
6.2	PAPEIS	12
7.	DESENHO E IMPLANTAÇÃO	13
7.1	AMBIENTE DE DESENVOLVIMENTO	13
7.2	CICLO DE DESENVOLVIMENTO	14
7.3	CICLO DE ENTREGA	15
7.4	ARQUITETURA	17
7.5	CRONOGRAMA E EXECUÇÃO	18
8.	FUNCIONALIDADES	21
8.1	IMPLEMENTADAS	21
8.1.1	Requisitos Funcionais	21
8.1.2	Regras de Negócio	22
8.1.3	Requisitos Não-Funcionais	23
8.2	NÃO IMPLEMENTADAS	24
8.2.1	Requisitos Funcionais	24
8.2.2	Regras de Negócio	24
8.2.3	Requisitos Não-Funcionais	25
9.	CONSIDERAÇÕES	26
9.1	O Projeto	26
9.2	A Metodologia	27
9.3	Expectativas e experiência adquirida	28

10. REFERÊNCIAS	29
11. APÊNDICES E ANEXOS	30
11.1 <u>Documento de requisitos</u> (fornecido pelo cliente)	30
11.2 Documentos de desenvolvimento	30
11.2.1 <u>Atributos de Qualidade</u>	30
11.2.2 <u>User Stories</u>	30
11.2.3 <u>Dicionário de dados e Diagrama ER</u>	30
11.2.4 <u>Fluxo de desenvolvimento</u>	30
11.2.5 <u>Ciclo de entrega</u>	30
11.2.6 <u>Arquitetura da Aplicação</u>	30
11.2.7 <u>Testes de aceitação realizados no sistema</u>	30
11.2.8 Postmortem das Sprints	30

1. INTRODUÇÃO

O objetivo deste documento é descrever o problema que foi identificado e especificar os processos de negócio atuais e futuros da Vídeo Locadora Imperial.

O nosso objeto de estudo é a Vídeo Locadora Imperial localizada na Região Metropolitana da Grande Recife. A vídeo locadora Imperial tem como finalidade maior a prestação de serviço de locação de filmes que é composta por várias etapas, como gestão do acervo e dos clientes, consulta do acervo, reserva, locação, pagamento e devolução. Tendo em vista o grande número de tarefas a ser realizadas pela vídeo locadora Imperial o projeto vem com o objetivo de ajudar a mesma no gerenciamento das informações.

1.1 MOTIVAÇÃO

A Vídeo Locadora Imperial necessita de um sistema de informação para apoiar a realização de suas atividades principais, a saber: empréstimo e devolução de itens e reserva de filmes. Para que essas atividades sejam apoiadas, é necessário controlar as informações acerca de filmes e clientes, dentre outros. Além disso, devem ser fornecidas facilidades de consulta ao acervo da locadora, permitindo consultas por diversas informações dos filmes.

1.2 SOBRE A ORGANIZAÇÃO

O público-alvo do sistema é constituído pelos funcionários da Locadora Imperial e seus clientes.

Uma locadora de vídeo guarda um pouco da história do cinema. Pelas prateleiras, é possível encontrar títulos de épocas, gêneros e estilos diferentes. Desde os clássicos até os lançamentos mais recentes que ainda não estão disponíveis nem na TV paga nem no Netflix. Neste contexto, a Vídeo Locadora Imperial atua no mercado de vídeo há mais de 10 anos, com o foco de trabalho na Região Metropolitana da Grande Recife. Visando constantemente o crescimento e a satisfação dos clientes, os resultados desse trabalho são a confiança e apoio de nossos vários parceiros e clientes.

2. OBJETIVOS

2.1 O PROBLEMA IDENTIFICADO

As atividades fundamentais de uma locadora consiste em realizar a locação e receber a devolução de filmes, bem como, o gerenciamento dos clientes e fornecedores, atualmente a Vídeo Locadora Imperial não dispõe de uma ferramenta para auxiliar essas atividades, causando-lhe inúmeros problemas tais como: inconsistências nas informações gerenciais, ineficiência no controle das locações, devoluções e reservas, dos clientes, bem como, de todo o acervo da locadora e os fornecedores.

Por outro lado, a falta de visibilidade do acervo da locadora impede o crescimento no número de clientes e de locações de filmes.

Desta forma o projeto deve satisfazer os objetivos da organização e definir porque o sistema é necessário.

- Proporcionar o acesso web do catálogo de filmes da locadora a fim de atrair novos clientes;
- Oferecer o serviço de reserva, com o envio de e-mail de notificações com o objetivo de promover a satisfação dos clientes;
- Gerenciar o catálogo, os clientes e os serviços de locação e devolução de filmes para melhorar as atividades gerenciais da locadora;

3. ESCOPO NEGATIVO

Para melhor compreensão da abrangência do sistema, abaixo encontram-se os problemas da empresa que não serão solucionados com o sistema.

- O sistema não vai possibilitar a reserva on-line de filmes;
- O sistema não permitirá a locação on-line e delivery de filmes;
- Outras atividades gerenciais além das atividades fundamentais listadas pelo cliente não serão contempladas pelo sistema, como por exemplo, a gestão de RH, Gestão Financeira, Contábil etc.

4. IDENTIFICAÇÃO DE ATORES DO SISTEMA

4.1 ATORES E FUNÇÕES

4.1.1 Administradores

Grupo de usuários do sistema com permissões para gerenciamento de todo o acervo e os usuários do sistema.

4.1.2 Atendentes

Grupo de usuários do sistema com permissões para gerenciamento das atividades de reserva, locação e devolução de itens além da gestão de clientes e seus dependentes.

4.1.3 Clientes / Visitantes do web site

Grupo de usuários do sistema que terá acesso apenas a consulta do acervo da vídeo locadora através do web site.

5. METODOLOGIA

Quando tratasse de metodologia de desenvolvimento de sistemas existem diversos métodos disponíveis que são divididos em 2 categorias: Metodologias estruturadas/tradicionais e Metodologias ágeis, contudo, podemos afirmar que não existe uma metodologia perfeita que possa ser empregada em todos os casos de desenvolvimento, todos os projetos e que trará os melhores resultados.

Tendo em vista o tamanho e as habilidades da equipe, as definições de necessidades do sistema apresentado pelo cliente, assim como outras características do projeto, foi definido que para o desenvolvimento deste projeto deveríamos adotar uma metodologia híbrida, que consiste no uso de técnicas e características de diversos métodos de desenvolvimento de software, de forma que pudesse apoiar, acompanhar e gerenciar melhor o projeto. Desta forma a equipe optou pelo uso de metodologia ágil de desenvolvimento como base e a utilização de técnicas e boas práticas de outras metodologias que pudessem proporcionar o melhor acompanhamento e gestão do projeto.

Uma característica que reforça o uso de uma metodologia ágil é a entrega contínua do projeto ou partes dele, visto que após uma interação o que foi desenvolvido já pode ser disponibilizada para o cliente sem a necessidade de finalização do projeto.

Assim, a equipe optou por usar o Scrum como metodologia de desenvolvimento, isso porque o Scrum adota um conjunto de valores, práticas e princípios compatíveis com a proposta do projeto.

O Scrum não define uma técnica específica para o desenvolvimento de software durante a etapa de implementação, ele se concentra em descrever como os membros da equipe devem trabalhar para produzir um sistema flexível, num ambiente de mudanças constantes. A ideia central do Scrum é que o desenvolvimento de sistemas envolve diversas variáveis (ambientais e técnicas) e elas possuem grande probabilidade de mudar durante a execução do projeto (por exemplo: requisitos, prazos, recursos, tecnologias etc.). Estas características tornam o desenvolvimento do sistema de software uma tarefa complexa e imprevisível, necessitando de um processo flexível e capaz de responder às mudanças (FRANCO 2007).

O Scrum possui 3 pilares em sua estrutura que consiste de: Transparência dos processos, requisitos de entrega e status do projeto; a Inspeção constante de tudo que está sendo feito e a Adaptação, tanto do processo quanto do produto às mudanças.

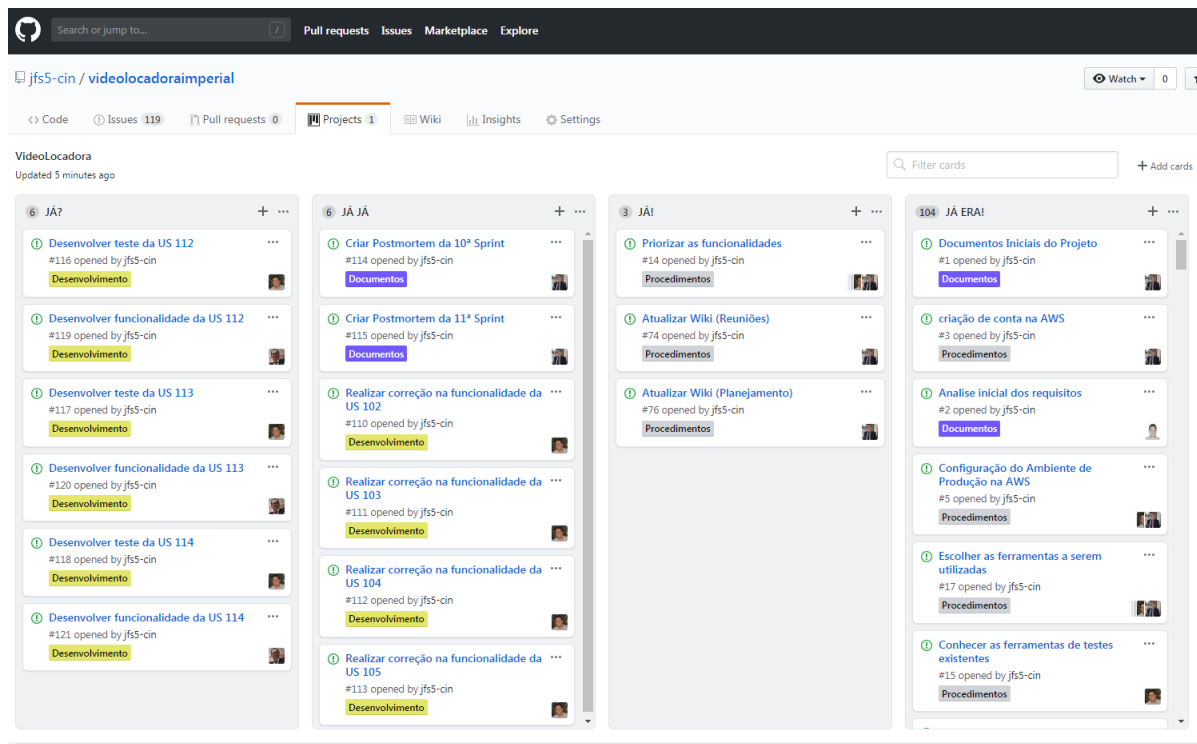
(Scrum 2014) afirma que no Scrum, os projetos são divididos em ciclos chamados de Sprints. O Sprint representa um período no qual um conjunto de atividades deve ser executadas. Desta forma o projeto é dividido em um determinado número de interações (Sprints) que deve possuir uma quantidade limitada de atividades a serem realizadas, facilitando a gestão, o acompanhamento e o planejamento do projeto.

Segundo (Scrum 2014), as funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como Product Backlog. No início de cada Sprint, faz-se uma reunião de planejamento na qual o Product Owner prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas em um Sprint são transferidas do Product Backlog para o Sprint Backlog.

A cada dia de uma Sprint, a equipe deve fazer uma breve reunião, chamada Daily Scrum. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia (Scrum 2014). Contudo, a equipe do projeto optou por não realizar a Daily Scrum, realizando as reuniões iniciais de cada Sprint (on-line). Devido a distribuição geográfica dos membros bem como a incompatibilidade de agenda dos membros da equipe.

Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting. Finalmente, faz-se uma Sprint Retrospective e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo(Scrum 2014).

Para facilitar o acompanhamento das atividades do projeto por todos, foi utilizado o Task board, que consiste na utilização de cartões (*pos-it*) para indicar o andamento de cada atividade do projeto. Porém, foi substituído os cartões (Post-it) pelo uso da ferramenta de projetos disponível no gitHub. Veja a ilustração abaixo:



A task board do projeto foi personalizada em 4 status: **JÁ?** (status inicial da tarefa, onde ficam alocadas todas as atividades identificadas do projeto que ainda não foram iniciadas); **JÁ JÁ** (status secundário das tarefas, onde ficam alocadas as atividades que foram distribuídas entre os membros da equipe que estão como prioridade para execução pela equipe); **JÁ!** (status onde fica alocada as atividades que estão sendo executadas por algum membro da equipe) e por fim **JÁ ERA!** (status onde são alocadas todas as atividades finalizadas e entregues do projeto)

6. EQUIPE DO PROJETO

6.1 MEMBROS

- **José Fernando da Silva** (jfs5@cin.ufpe.br) é graduado em Gestão de Tecnologia da Informação pela Universidade Estácio de Sá, com especialização em Informática, atuando desde de 2010 como Técnico de Tecnologia da Informação, exercendo a função Coordenador de Gestão de Tecnologia da Informação de uma Instituição Federal de Ensino.
- **Anderson França Ferreira** (aff2@cin.ufpe.br) é graduado em Ciências da Computação pela Universidade Católica de Pernambuco com especialização em Gestão de Tecnologia da Informação, atuando desde de 2014 como Técnico de Tecnologia da Informação, exercendo a função Coordenador de Gestão de Tecnologia da Informação e Comunicação de uma Instituição Federal de Ensino.
- **Jobson Tenório do Nascimento** (jtn@cin.ufpe.br) é graduado em Análise e Desenvolvimento de Sistemas pela Faculdade de Filosofia, Ciências e Letras de Caruaru, com especialização em Gestão Pública, atuando desde de 2010 como Analista de Tecnologia da Informação, exercendo a função Coordenador de Gestão de Tecnologia da Informação de uma Instituição Federal de Ensino.
- **Wellyson Fernando Nunes Souza** (wfn@cin.ufpe.br) é graduado em Análise e Desenvolvimento de Sistemas Faculdade de Filosofia, Ciências e Letras de Caruaru, com Engenharia de Software, atuando desde de 2011 como Técnico de Tecnologia da Informação de uma Instituição Federal de Ensino.

Após levantamento das habilidades dos membros da equipe e objetivando o melhor aproveitamento das habilidades de cada membro da equipe, foram definidos os papéis de cada membro no projeto da seguinte forma:

6.2 PAPEIS

- **José Fernando da Silva** será o gerente do projeto, que tem a função de coordenar as atividades do projeto, bem como, ser o interlocutor entre o cliente e a equipe de desenvolvimento.
- **Anderson França Ferreira** será o analista de requisitos, que tem a função de realizar o levantamento e análise de todas as demandas apresentadas pelo cliente para o sistema, entende qual o problema a ser automatizado, o que pode ser melhorado, descreve em documentos formais como o sistema deve se comportar de acordo com cada ação do usuário para, posteriormente, serem projetados e codificados pela equipe de desenvolvimento(Hatano 2012).
- **Jobson Tenório do Nascimento** e **Wellyson Fernando Nunes Souza** serão os Desenvolvedores da aplicação, que possui a atribuição de desenvolver e implementar os testes e as funcionalidades especificadas do sistema pelo o analista de requisitos, bem como, a manutenção e gestão do banco de dados. Cabendo ainda aos desenvolvedores a realização de correções das funcionalidades quando detectadas no sistema ou banco de dados.

7. DESENHO E IMPLANTAÇÃO

7.1 AMBIENTE DE DESENVOLVIMENTO

Como ambiente de desenvolvimento a equipe utilizou notebook com Sistema Operacional Microsoft Windows 10 PRO com diversas ferramentas e bibliotecas conforme relacionadas abaixo com suas respectivas versões e descrições para o projeto:

- Ferramentas**

Ferramenta	Versão	Descrição
Apache	2.4.X	Servidor de conteúdo web
PHP	7.3.X	Interpretador da linguagem de programação de script executada no servidor
Mysql	5.7.X	SGDB-Sistema Gerenciador de Banco de Dados
Laravel	5.7.X	Framework de desenvolvimento PHP
Composer	1.8.X	Gerenciador de dependências para aplicações PHP
Visual Studio Code	1.31.X	Editor de código-fonte/IDE
Cmder	1.3.X	Emulador de console para Windows (aceita comandos Linux e Windows)

- Bibliotecas PHP (Composer)**

Biblioteca	Versão	Descrição
guzzlehttp/guzzle	6.X	Biblioteca que facilita o envio de solicitações HTTP para integração com serviços da web.
Jeroennoten/laravel-adminlte	1.X	Painel de administração que fornece uma variedade de componentes responsivos, reutilizáveis.
league/flysystem-aws-s3-v3	1.X	Biblioteca para integração de sistemas PHP com o serviço S3 da AWS.
simplesoftwareio/simple-qrcode	2.X	Biblioteca de criação de QRCode's.
laravel/dusk	4.X	API para aplicação de testes automatizados em sistemas PHP/Laravel.

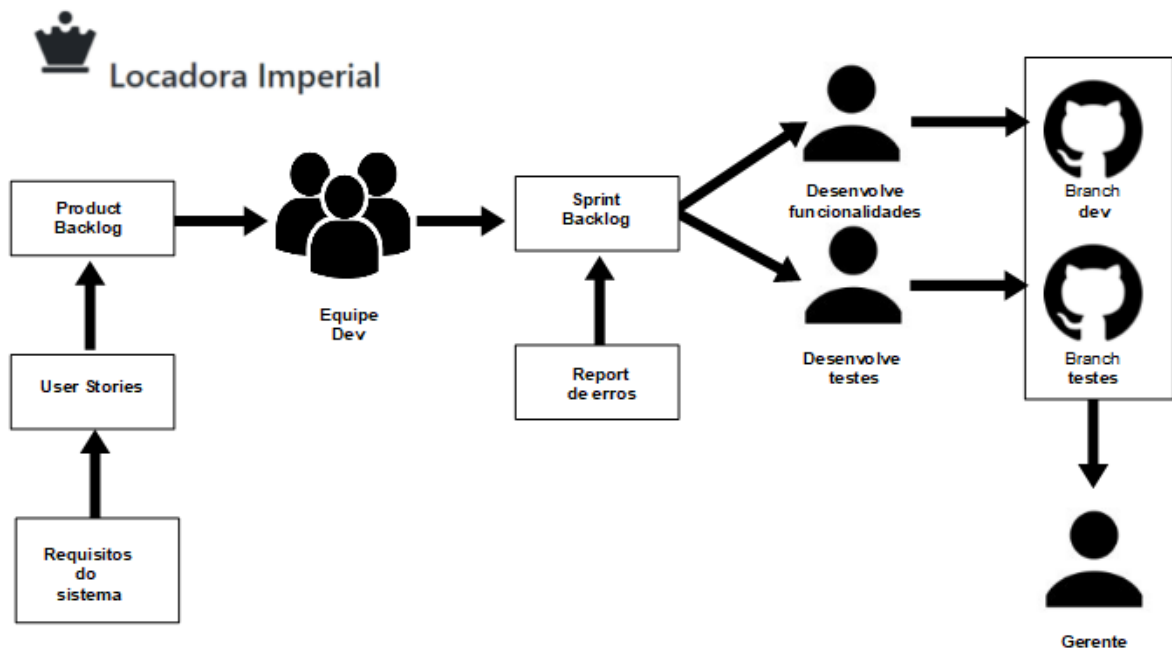
- **Bibliotecas Java Script**

Biblioteca	Versão	Descrição
DataTables	1.10.X	Biblioteca para filtros e ordenações de tabelas HTML.
icheck-1.0.2	1.X	Biblioteca que fornece um conjunto de botões e caixas de seleção personalizáveis.
select2-4.0.5	4.0.X	Biblioteca que fornece caixa de seleção personalizável com suporte para pesquisa.
webcodecamjs	2.1.X	Biblioteca para leitura de QRCode.

7.2 CICLO DE DESENVOLVIMENTO

A Sprint Backlog será composta pelas funcionalidades priorizadas pela equipe, derivadas dos requisitos do sistema, em conjunto com o Report de erros da sprint anterior. Em seguida a atividade de desenvolvimento é dividida entre os dois desenvolvedores do grupo de modo que um desenvolve os testes e o outro a funcionalidade, estes papéis de desenvolvimento de testes e funcionalidade são alternados entre os desenvolvedores a cada nova sprint. Em princípio o grupo tinha optado pela adoção da metodologia **TDD - Test Driven Development**, onde primeiro deveriam ser feitos os testes para posteriormente serem desenvolvidas as funcionalidades, no entanto devida a falta de experiência da equipe com testes, optamos por desenvolver testes e funcionalidades em paralelos e auxiliarmos-nos mutuamente. Finalizados testes e funcionalidades realizamos o merge da **branch test** com a **branch dev** e executamos os testes para assim notificarmos o gerente do projeto da conclusão das funcionalidades.

- **Ciclo de Desenvolvimento**



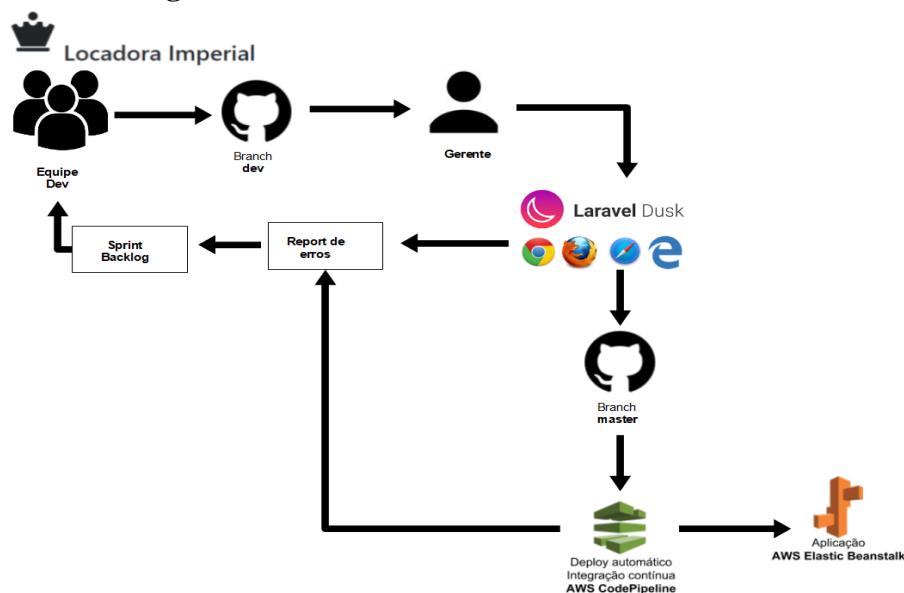
7.3 CICLO DE ENTREGA

O ciclo de entrega do sistema é realizado através de etapas definidas pela equipe de desenvolvimento, no qual após o desenvolvimento dos testes e funcionalidades do sistema, os desenvolvedores notificam o gerente do projeto sobre as funcionalidades concluídas, que são disponibilizadas na branch “dev” da plataforma de hospedagem de código-fonte (GitHub), em seguida o gerente do projeto realiza a cópia do código-fonte disponibilizado na branch “dev” do GitHub para o seu ambiente de testes e validação, assim o gerente realiza todos os testes automatizados utilizando o Laravel Dusk além de verificar as funcionalidades nos principais navegadores (Microsoft Edge, Mozilla Firefox, Google Chrome e Safari). Estando todas as funcionalidades funcionando conforme especificado e todos os testes automatizados de validação sendo realizados sem erros o gerente executa o merge das funcionalidades para a branch “master” do GitHub, caso contrário os erros são reportados pelo gerente do projeto e inseridos no próximo Sprint backlog para correção.

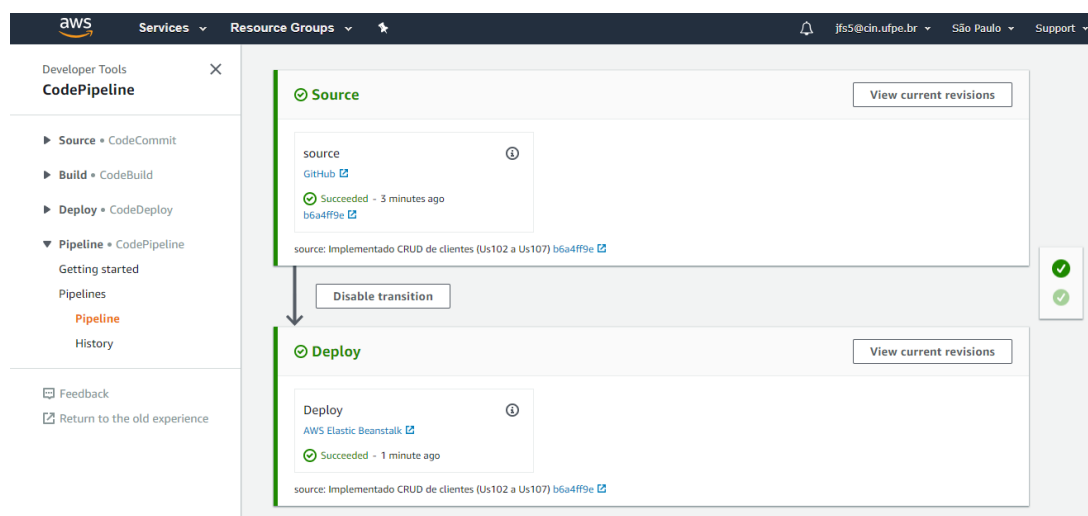
A branch “master” do projeto no GitHub está integrado ao serviço AWS CodePipeline, desta forma, ao ser detectado uma nova versão do sistema ou parte dele no GitHub pelo AWS CodePipeline, o mesmo realiza cópia desta nova versão, verifica se existe erros de sintaxe do código-fonte, estando o código correto o AWS CodePipeline realiza a implantação do código no serviço AWS Elastic Beanstalk que também está integrado ao CodePipeline, caso contrário os erros são reportados pelo gerente do projeto e inseridos no próximo Sprint backlog para correção.

O AWS Elastic Beanstalk é o serviço onde está hospedado a versão de produção do sistema. Desta forma é concluído o ciclo de verificação e entrega das versões que são desenvolvidas do sistema ao cliente.

• Ciclo de Entrega



• Integração com AWS CodePipeline



Integração com AWS Elastic Beanstalk

Visão geral

Integridade OK

Versão em execução code-pipeline-1549131846225-32b52e08c2dd84789c3ec3c57e-df8b0a12ee30

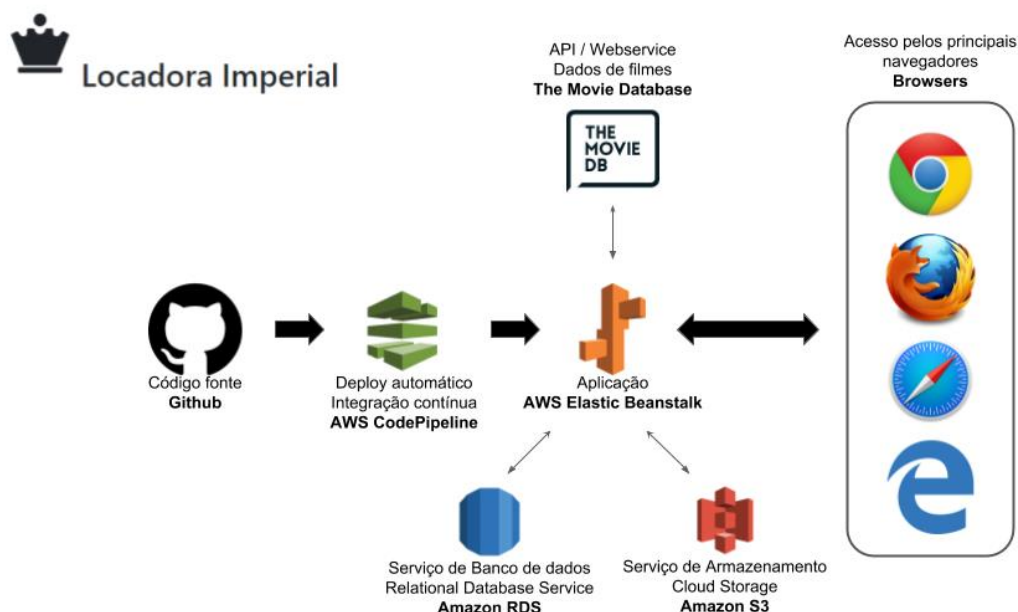
Configuração PHP 7.2 running on 64bit Amazon Linux/2.8.5

Eventos recentes

Tempo	Tipo	Detalhes
02-02-2019 15:26:23 UTC-0300	INFO	Environment health has transitioned from Info to Ok. Application update completed 55 seconds ago and took 48 seconds.
02-02-2019 15:25:01 UTC-0300	INFO	Environment update completed successfully.
02-02-2019 15:25:01 UTC-0300	INFO	New application version was deployed to running EC2 instances.

7.4 ARQUITETURA

O sistema está estruturado de forma que possibilite a integração entre a plataforma de código-fonte com controle de versão (GitHub) e o serviço AWS CodePipeline que realiza a verificação de compilação do código do sistema e em seguida realiza a implementação do código na instância o AWS Elastic Beanstalk (PaaS) onde está hospedado o sistema. O AWS Elastic Beanstalk ainda está integrado com os serviço de Amazon RDS de banco de dados e o Amazon S3 de armazenamento de objetos, além da integração com o serviço externo TMDb que disponibiliza uma base de dados grátis com informações sobre filmes e series de TV.



Considerando a arquitetura do sistema, bem como, a linguagem de programação utilizada para seu desenvolvimento, o sistema é capaz de ser usado nos principais navegadores web, para plataformas Desktop.

7.5 CRONOGRAMA E EXECUÇÃO

Após a finalização do projeto, o mesmo foi executado em 13 sprints com metas e períodos determinados, conforme descrito no quadro abaixo.

SPRINT	PERÍODO	META
1 ^a	03/12/18 a 09/12/18	Levantamento das habilidades dos membros, construção e revisão dos documentos iniciais e criação de configuração de ambiente de desenvolvimento.
2 ^a	17/12/18 a 23/12/18	Configurar o ambiente de produção na AWS (codepipeLine, Elastic BeanStalk etc.), realizar integração do repositório gitHub com AWS para entrega contínua do sistema, Listar as funcionalidades do sistema, definir os critérios de priorização de desenvolvimento das funcionalidades, priorizar as funcionalidades para desenvolvimento, conhecer ferramentas de teste automatizado escolher ferramentas de testes automatizados para uso no desenvolvimento.
3 ^a	02/01/19 a 05/01/19	Criar o banco de dados da aplicação, popular o banco de dados para desenvolvimento inicial, criar template (modelo) das telas do sistema, desenvolver testes e funcionalidades do sistema (autenticação de administrador e atendente), manter o repositório e os arquivos atualizados e continuar o levantamento e priorização das funcionalidades.
4 ^a	07/01/19 a 09/01/19	Desenvolver testes e funcionalidades do sistema (CRUD de mídia, gênero, tipo e consulta simples de acervo pelo cliente), manter o repositório e os arquivos atualizados e continuar o levantamento e priorização das funcionalidades.
5 ^a	10/01/19	Finalizar o desenvolvimento de testes e funcionalidades do sistema (CRUD de distribuidor e filmes), desenvolver testes e funcionalidades do

	a 12/01/19	sistema (Consulta avançada do acervo pelo cliente), manter o repositório e os arquivos atualizados e continuar o levantamento e priorização das funcionalidades.
6 ^a	14/01/19 a 19/01/19	Atualizar as configurações do ambiente de produção (GitHub + AWS), atualizar o repositório GitHub, disponibilizar versão inicial do sistema no ambiente de produção, Finalizar o desenvolvimento de testes e funcionalidades do sistema (CRUD de distribuidor e filmes), desenvolver testes e funcionalidades do sistema (CRUD de itens), manter o repositório e os arquivos atualizados e continuar o levantamento e priorização das funcionalidades.
7 ^a	21/01/19 a 23/01/19	Elaborar documentos de post-mortem das sprint's do projeto, manter o repositório e os arquivos atualizados, Criar documento de atributos de qualidade do sistema, finalizar o desenvolvimento de testes e funcionalidades do sistema (CRUD de filmes), desenvolver testes e funcionalidades do sistema (CRUD de usuários) e continuar o levantamento e priorização das funcionalidades.
8 ^a	24/01/19 a 26/01/19	Atualizar o cliente do status do projeto, manter o repositório e os arquivos atualizados, elaborar o documento de atributos de qualidade do sistema, finalizar o desenvolvimento de testes e funcionalidades do sistema (CRUD de filmes), desenvolver testes e funcionalidades do sistema (CRUD de clientes).
9 ^a	28/01/19 a 30/01/19	Atualizar o cliente do status do projeto, manter o repositório e os arquivos atualizados, elaborar o documento de atributos de qualidade do sistema, finalizar o desenvolvimento de testes e funcionalidades do sistema (CRUD de filmes), desenvolver testes e funcionalidades do sistema (CRUD de clientes)
10 ^a	31/01/19 a 02/02/19	Atualizar o cliente do status do projeto, manter o repositório e os arquivos atualizados, elaborar o documento de atributos de qualidade do sistema, realizar correção nas funcionalidades do sistema (CRUD de Gênero e Item), desenvolver testes e funcionalidades do sistema (CRUD

		de clientes).
11ª	04/02/19 a 06/02/19	Manter o repositório e os arquivos atualizados, desenvolver testes e funcionalidades do sistema (CRUD de locação) e realizar correções e melhorias nas funcionalidades do sistema (CRUD de clientes).
12ª	07/02/19 a 09/02/19	Realizar upload do post-mortem das sprints para o repositório, desenvolver o relatório final do projeto para entrega ao cliente, apresentar o projeto ao cliente, finalizar e enviar o projeto para o cliente, manter o repositório e os arquivos atualizados, desenvolver testes e funcionalidades do sistema (Locação e Devolução).
13ª	11/02/19 a 15/02/19	Realizar upload do post-mortem das sprints para o repositório, finalizar o relatório final do projeto para entrega ao cliente, apresentar o projeto ao cliente, finalizar e enviar o projeto para o cliente, manter o repositório e os arquivos atualizados, desenvolver testes e funcionalidades do sistema (Locação e Devolução).

O detalhamento das atividades executadas e não executadas de cada Sprint está presente em seus respectivos postmortem.

8. FUNCIONALIDADES

8.1 IMPLEMENTADAS

Segue abaixo tabela contendo as funcionalidades implementadas no sistema, distribuídas de acordo com os requisitos apresentados pelo cliente.

8.1.1 Requisitos Funcionais

ID	Descrição	Prioridade
[RF01]	O sistema deve registrar locações, indicando o cliente e os itens locados, bem como a data e o valor da locação e a data de devolução prevista de cada item.	ALTA
[RF02]	O sistema deve registrar devoluções, indicando os itens devolvidos e a data de devolução.	ALTA
[RF06]	O sistema deve efetuar o controle do acervo da locadora.	ALTA
[RF07]	O sistema deve efetuar o controle de distribuidoras de filmes.	BAIXA
[RF08]	O sistema deve permitir consultas ao acervo. As consultas poderão ser feitas informando uma (ou uma combinação) de informações dos filmes, dentre elas: título (ou parte dele), gênero, tipo de mídia disponível, ator.	MÉDIA
[RF09]	O sistema deve permitir o gerenciamento do conjunto de clientes da locadora, abrangendo tanto clientes titulares quanto seus dependentes.	ALTA
[RF10]	O sistema deve permitir desativar clientes.	MÉDIA
[RF11]	O sistema deve permitir reativar clientes.	BAIXA

8.1.2 Regras de Negócio

ID	Descrição	Prioridade
[RN01]	O sistema deve permitir que sejam dados descontos nas locações, bem como que sejam ampliados os prazos de devolução de itens, em função da política da empresa.	MÉDIA
[RN02]	O sistema deve cobrar multa para itens devolvidos com atraso, segundo a seguinte fórmula: $M = n * VL$, onde M é o valor da multa, n é o número de dias de atraso e VL é o valor de locação do item.	ALTA
[RN04]	O sistema deve manter o histórico de locações e, portanto, clientes que tenham feito locações não poderão ser excluídos.	ALTA
[RN05]	Quando um cliente titular for desativado, todos os seus dependentes também devem ser desativados.	MÉDIA
[RN06]	O sistema deve gerar um número de inscrição único para cada cliente.	ALTA
[RN07]	Um titular pode ter de zero a três dependentes em um ponto no tempo.	ALTA
[RN11]	O valor da locação de um item é dado pelo tipo de mídia do item. Cada tipo de mídia tem um valor de locação associado. Um acréscimo de 50% do valor da locação do tipo de mídia deve ser aplicado no caso do filme do item ser um lançamento.	ALTA
[RN12]	A data de devolução prevista é definida em função do filme do item ser lançamento ou não. Lançamentos têm prazo de um dia; filmes do catálogo têm três dias de prazo.	ALTA
[RN14]	A data de devolução prevista de um item locado deve ser superior à data da locação.	ALTA

8.1.3 Requisitos Não-Funcionais

ID	Descrição	Prioridade
[RNF01]	O sistema deve controlar o acesso às funcionalidades. Funcionalidades para controlar o acervo da locadora devem ser restritas a administradores. Funcionalidades de atendimento a clientes devem estar restritas a atendentes. Funcionalidades de consulta ao acervo devem estar disponíveis na Internet.	ALTA
[RNF02]	A consulta ao acervo deve estar disponível pela Internet, a partir dos principais navegadores disponíveis no mercado.	MÉDIA
[RNF03]	Os itens devem ser identificados por um código de barras, sendo possível a leitura dos mesmos usando dispositivos de leitores de código de barras.	ALTA
[RNF04]	O tempo para a realização das funções de atendimento ao cliente deve ser inferior a cinco segundos, a partir da correta entrada de dados.	MÉDIA
[RNF07]	A persistência das informações deve ser implementada, em um primeiro momento, em um Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR) livre (Postgres ou MySQL). Contudo, no futuro deve ser possível utilizar outros SGBDRs ou até mesmo outra tecnologia de bancos de dados.	ALTA
[RNF08]	O sistema deve ser fácil de usar, devendo-se evitar a digitação desnecessária de informações, de modo a dar agilidade ao processo.	ALTA
[RNF09]	O sistema deve ser fácil de manter, de modo a acomodar novas funcionalidades ou até mesmo adaptação para outras vídeo locadoras.	ALTA

8.2 NÃO IMPLEMENTADAS

Segue abaixo tabela contendo as funcionalidades não implementadas no sistema, distribuídas de acordo com os requisitos apresentados pelo cliente, conforme listadas abaixo.

8.2.1 Requisitos Funcionais

ID	Descrição	Prioridade
[RF03]	O sistema deve registrar os pagamentos de locações	ALTA
[RF04]	O sistema deve registrar a reserva de filmes a clientes, permitindo indicar, ainda, o tipo de mídia desejado.	MÉDIA
[RF05]	O sistema deve permitir o cancelamento de uma reserva, tanto pelo usuário, quanto automaticamente pelo sistema, quando expirado o prazo para retirada do item, de acordo com política da empresa.	MÉDIA
[RF12]	O sistema deve notificar clientes quando um item reservado estiver disponível.	BAIXA

8.2.2 Regras de Negócio

ID	Descrição	Prioridade
[RN03]	Reservas canceladas pelo sistema não deverão ser efetivamente excluídas pelo sistema, mas sim marcadas como expiradas.	MÉDIA
[RN04]	O sistema deve manter o histórico de locações e, portanto, clientes que tenham feito locações não poderão ser excluídos.	ALTA
[RN08]	Pagamentos podem ser feitos no ato da locação ou da devolução de um item. Caso uma locação de um item não tenha sido paga no ato da locação, terá de ser paga obrigatoriamente na devolução.	ALTA
[RN09]	Pagamentos podem ser feitos em dinheiro, cheque ou cartão.	ALTA
[RN10]	O atendimento às reservas deve se dar pela ordem temporal em que as reservas são feitas, considerando os tipos de mídia solicitados nas reservas.	MÉDIA
[RN13]	Locações pagas não podem ser canceladas.	ALTA
[RN15]	Uma reserva expira quando passadas mais do que 24h de sua comunicação para o cliente.	ALTA

[RN16]	Clientes em atraso não podem efetuar nem locações nem reservas.	ALTA
[RN17]	Não se devem aceitar reservas quando houver itens disponíveis na locadora que atender à reserva.	ALTA

8.2.3 Requisitos Não-Funcionais

ID	Descrição	Prioridade
[RNF05]	O sistema deve estar integrado ao Sistema de Operadoras de Cartão de Crédito para enviar e receber informações para pagamento com cartão de crédito.	ALTA
[RNF06]	A comunicação de disponibilidade de uma reserva deverá ser feita automaticamente por e-mail	BAIXA

9. CONSIDERAÇÕES

9.1 O PROJETO

No início do projeto a equipe decidiu o período de 1 (uma) semana para cada Sprint, contudo, após o passar do 1º (primeiro) mês do cronograma do projeto com poucas entregas e realizações das atividades destinadas para as Sprints, e buscando ajustar o Product backlog do projeto ao cronograma, foi realizada a mudança do período das Sprint's de 1 (uma) semana para 3 (três) dias, com essa alteração no período das Sprint's foi possível tanto dimensionar melhor a quantidade de tarefas para cada Sprint quando realizar um acompanhamento mais fiel das atividades realizadas pelos membros da equipe.

Em virtude da constância de realizações das Sprint's com intervalos de 3 (três) dias, bem como, a distribuição geográfica da equipe, a mesma optou por não realizar as Daily Scrum das Sprint's, realizando apenas a Sprint Review Meeting e Sprint Retrospective em conjunto com a Sprint Planning Meeting da próxima Sprint e registrando as deliberações e considerações nas atas das reuniões, ressaltando que nas reuniões não havia a participação do Product Owner, conforme prevê o Scrum.

Visando a transparência do status do projeto para todos os membros do projeto e o cliente, a equipe decidiu publicizar todos os artefatos construídos para o projeto, bem como, as pautas e Atas das reuniões da equipe do projeto e o planejamento das Sprints através da wiki do projeto disponível na ferramenta de hospedagem de códigos-fontes e versionamento GitHub. Conforme é possível acessar através do link <https://github.com/jfs5-cin/videolocadoraimperial/wiki>, onde existe a página de 1. Início que possui informações do projeto, a página 2. Documentos que possui todos os documentos e artefatos do projeto, a página 3. Reuniões onde está todas as Pautas de Atas das reuniões realizadas pela equipe e a página 4. Planejamento onde está distribuído todo o planejamento das Sprint's do projeto, contendo todas as atividades que estava prevista para a Sprint e seus respectivos responsáveis, as atividades que foram executadas e as atividades não executadas na Sprint.

A [RNF03] foi implementada utilizando não código de barras convencionais e sim utilizando QRCode, além dos itens o sistema é capaz de realizar a leitura/identificação dos clientes através de QRCode de identificação, disponível para cada cliente em sua carteira de cliente da locadora.

Como forma de complementar o uso da tecnologia de QRCode conforme expresso na [RNF03] e facilitar a implementação e uso no sistema, os desenvolvedores incluíram no sistema uma biblioteca que cria o QRCode automaticamente no cadastro de Cliente (criando o QRCode do cliente e seus dependentes, assim, o usuário no próprio sistema realiza os cadastro e gera os QRCode, sem a necessidade de uso de outra ferramenta para essa finalidade.

No momento a funcionalidade de leitura de QRCode só está funcionando no Navegador Mozilla Firefox, isto porque o acesso a webcam exige uma conexão segura, protocolo HTTPs, nos demais navegadores. Assim seria necessário apenas a compra do domínio e de um certificado de segurança válido para a Vídeo Locadora Imperial.

Conforme solicitado pelo cliente nos requisitos do sistema pelo cliente, o sistema deveria realizar a integração com serviços externos, e para atender este requisito a equipe do projeto realizou a integração de 3 (três) serviços externos, sendo eles: O Amazon RDS que disponibiliza diversas plataformas de SGBD's, o Amazon S3 serviço de armazenamento de objetos na nuvem e o TMDB que consiste numa base de dados grátis com informações sobre filmes e series de TV.

9.2 A METODOLOGIA

Embora a equipe tenha escolhido a metodologia ágil Scrum como base para o desenvolvimento e gestão do projeto, algumas características da metodologia tiveram que ser adaptadas para a realização do projeto. Conforme previsto na metodologia o Product Owner deveria estar presente nas reuniões, outro aspecto de destaque são os artefatos que devem ser criados durante o projeto de acordo com o Scrum, visando otimizar os artefatos do projeto a equipe optou por usar a wiki do projeto para disponibiliza-los, assim foi possível em uma única plataforma (wiki) colocar todos os documentos do projeto, como: Pautas e Atas das reuniões da equipe, o planejamento das Sprints do projeto, os documentos fornecidos pelo cliente e elaborados durante o projeto, como postmortens e outros e um resumo do projeto e sua finalidade.

9.3 EXPECTATIVAS E EXPERIÊNCIA ADQUIRIDA

Conforme foi apresentado a proposta do projeto para a equipe, os membros se depararam com um desafio, visto que o projeto abordou um conjunto de requisitos para obtenção do êxito, pois algumas exigências eram desconhecidas por alguns membros, e outros que detinham algum conhecimento também era superficial, contudo, através de técnicas e ferramentas recomendadas e apresentadas em sala de aula pelo professor foi um passo inicial de como seria possível a conclusão do projeto.

Podemos destacar como pontos positivo para o êxito do projeto na visão da equipe a exposição em aula das ferramentas de hospedagem de conteúdo (o produto final e o código-fonte) e suas possibilidades, outro ponto que destacamos positivamente foi a demonstração das metodologias de desenvolvimento e as várias possibilidades de uso, outro aspecto que contribuiu para o êxito foi o cliente/professor deixar a equipe livre para escolha de técnicas, ferramentas e metodologias para o projeto, sem restrições e limitações, ficando a equipe a vontade para uso de técnicas, ferramentas e metodologias que julga-se melhor para conclusão do projeto.

A equipe julgou como aspecto negativo foi a quantidade de conteúdo exposto na disciplina, com breves explicações de forma superficial, pois julgamos que o conteúdo da disciplina deveria ser menor e esses conteúdos serem mais explorados e aprofundados.

Por fim, diante do desafio apresentado, das tecnologias, metodologias e ferramentas descobertas e estudadas para a execução do projeto, consideramos que este projeto foi de grande importância para a trajetória acadêmica e profissional de todos os membros, pois muitas lições aprendidas para uso no projeto já estão sendo usadas em outros projetos acadêmicos do projeto e abrindo novas possibilidades nas atividades profissionais que cada um desempenha nos campi onde trabalhamos e até mesmo em projetos de desenvolvimento do IFPE.

10. REFERÊNCIAS

- FRANCO, E. F. 2007. “Um Modelo de Gerenciamento de Projetos Baseado Nas Metodologias Ágeis de Desenvolvimento de Software e Nos Princípios Da Produção Enxuta. Escola Politécnica Da Universidade de São Paulo (Dissertação de Mestrado).” Universidade de São Paulo.
- Hatano, Daniel. 2012. “As Qualidades de Um Excelente Analista de Requisitos.” <https://www.tiespecialistas.com.br/as-qualidades-de-um-excelente-analista-de-requisitos/> (February 12, 2019).
- Scrum. 2014. “Desenvolvimento Ágil.” <http://www.desenvolvimentoagil.com.br/scrum/> (February 8, 2019).

11. APÊNDICES E ANEXOS

11.1 DOCUMENTO DE REQUISITOS (FORNECIDO PELO CLIENTE)

11.2 DOCUMENTOS DE DESENVOLVIMENTO

11.2.1 Atributos de Qualidade

11.2.2 User Stories

11.2.3 Dicionário de dados e Diagrama ER

11.2.4 Fluxo de desenvolvimento

11.2.5 Ciclo de entrega

11.2.6 Arquitetura da Aplicação

11.2.7 Testes de aceitação realizados no sistema

11.2.8 Postmortem das Sprints

- 1ª Sprint
- 2ª Sprint
- 3ª Sprint
- 4ª Sprint
- 5ª Sprint
- 6ª Sprint
- 7ª Sprint
- 8ª Sprint
- 9ª Sprint
- 10ª Sprint
- 11ª Sprint
- 12ª Sprint
- 13ª Sprint