

UNIVERSIDADE FEDERAL DO PARANÁ
BRUNO PEREIRA MARTINS
GUSTAVO BENJAMIN VIRMOND PORTELA FERREIRA
HUGO HIDEO FUJIWARA

FUP – FILA DE UNIDADE PRESENCIAL

CURITIBA

2016

BRUNO PEREIRA MARTINS
GUSTAVO BENJAMIN VIRMOND PORTELA FERREIRA
HUGO HIDEO FUJIWARA

FUP – FILA DE UNIDADE PRESENCIAL

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Msc. Razer Anthom Nizer Rojas Montaña.

CURITIBA

2016

TERMO DE APROVAÇÃO

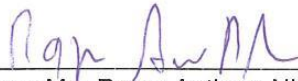
BRUNO PEREIRA MARTINS
GUSTAVO BENJAMIN VIRMOND PORTELA FERRERIRA
HUGO HIDEO FUJIWARA

FUP – FILA DE UNIDADE PRESENCIAL

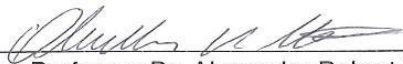
Trabalho apresentado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná.

Curitiba, 28 de Setembro de 2016.

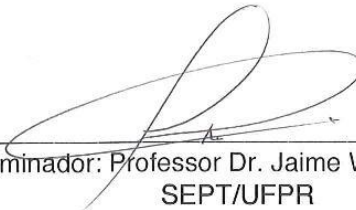
BANCA EXAMINADORA



Orientador: Professor Me. Razer Anthom Nizer Rojas Montaña
SEPT/UFPR



Examinador: Professor Dr. Alexander Robert Kutzke
SEPT/UFPR



Examinador: Professor Dr. Jaime Wojciechowski
SEPT/UFPR

AGRADECIMENTOS

Gostaríamos de agradecer a Deus por ter-nos abençoado ao longo da jornada acadêmica. Agradecemos muito ao nosso orientador, Professor Razer Anthom Nizer Rojas Montaña, por ter aceitado o desafio de nos guiar neste projeto de um prazo absurdo de um mês de duração. Não podemos esquecer-nos de agradecer às nossas respectivas esposas/namoradas pela paciência e compreensão durante essas exaustivas semanas. Por fim, agradecemos a nós mesmos, pelo companheirismo e dedicação deste grupo onde os integrantes se conheceram há pouco, mas houve entrosamento e união frente ao desafio.

*“No meio de cada dificuldade
existe uma oportunidade.”*

Albert Einstein

RESUMO

Nesta época a tecnologia vem ganhando cada vez mais espaço no mercado e no dia a dia dos cidadãos. Conectando-se com as diversas áreas e agilizando o acesso a informação, ajudando na evolução da saúde, lazer, segurança, educação, entre outras. Considerando pesquisas e relatos do dia a dia das pessoas em postos de atendimentos, que buscam um melhor tempo de espera e gerenciamento do fluxo de filas, encontrou-se a oportunidade de desenvolver o sistema FUP – Fila de Unidade Presencial – aliado ao grande crescimento na utilização de armazenamento em nuvem, este trabalho de conclusão de curso descreve o desenvolvimento de controle de filas e ao gerenciar o fluxo de atendimento, o sistema apresenta uma série de recursos que auxiliam na gerência e administração das unidades de atendimento, tornando-o mais ágil e dinâmico possível. Este documento compreende a descrição dos conceitos nos quais o projeto se baseia e das tecnologias utilizadas no desenvolvimento do sistema, plano de atividades e cronograma, divisão de responsabilidades, documentação e descrição das funcionalidades do sistema desenvolvido.

Palavras-chave: Controle de filas, tempo de espera, gerenciamento do fluxo, postos de atendimento.

ABSTRACT

At this time the technology is gaining more space in the market and in the daily lives of citizens. Connecting to the various areas and speeding access to information, helping in the development of health, leisure, security, education, among others. Considering research and reports the daily lives of people in attendance stations, seeking a better time and managing the flow queues, met the opportunity to develop the FUP – *Fila de Unidade Presencial* – system ally with strong growth in the use of cloud storage, this work completion of course describes the developed queue control and manage the flow of service, the system has a number of features that help in the management and administration of the service units, making it more agile and dynamic as possible. This document comprehends the description of the concepts on which the project stands and the technologies used on the system's development, activity plan and schedule, division of responsibilities, documentation and description of all the developed system's functionalities.

Keywords: queue control, waiting time, flow management, service stations.

LISTA DE FIGURAS

FIGURA 1 – ELEMENTOS DE UMA FILA.....	17
FIGURA 2 – ESTRUTURA DAS FILAS	19
FIGURA 3 – MVC	24
FIGURA 4 – FASES DE CADA ITERAÇÃO	27
FIGURA 5 – WORK BREAKDOWN STRUCTURE.....	29
FIGURA 6 – GRÁFICO DE GANTT COM TABELA DE PRECEDÊNCIAS.....	30
FIGURA 7 – ANÁLISE DO CAMINHO CRÍTICO	31
FIGURA 8 – PLANO DE RISCOS	32
FIGURA 9 – MÓDULO RECEPÇÃO.....	39
FIGURA 10 – PAINEL DE SENHAS	40
FIGURA 11 – MÓDULO ATENDIMENTO.....	41
FIGURA 12 – ESTADOS DA SENHA	42
FIGURA 13 - DASHBOARD	43
FIGURA 14 – TABELA DE DETALHE DA REGIAO DO DASHBOARD	44
FIGURA 15 – MAPA.....	44
FIGURA 16 – RELATÓRIO GERAL.....	45
FIGURA 17 – RELATÓRIO TMA	46
FIGURA 18 – DATASOURCE DO FUP	47
FIGURA 19 – DIAGRAMA DE CASOS DE USO	59
FIGURA 20 – TELA GERAR SENHA	60
FIGURA 21 – TELA CHAMAR SENHA.....	62
FIGURA 22 – TELA INICIAR E CANCELAR ATENDIMENTO	62
FIGURA 23 – TELA CONSULTA FILA	65
FIGURA 24 – TELA USUÁRIOS CADASTRADOS	66
FIGURA 25 – TELA ADICIONAR USUÁRIOS.....	67
FIGURA 26 – TELA EDITAR USUÁRIO.....	67
FIGURA 27 – TELA POSTOS CADASTRADOS	70
FIGURA 28 – TELA ADICIONAR POSTO	70
FIGURA 29 – TELA REGIÕES CADASTRADAS	72
FIGURA 30 – TELA ADICIONAR REGIÃO.....	73
FIGURA 31 – TELA RELATÓRIO GERAL.....	75
FIGURA 32 – TELA LOGIN	77

FIGURA 33 – TELA DASHBOARD	79
FIGURA 34 – TELA COM TABELA DE DETALHE DA REGIÃO NO DASHBOARD.	79
FIGURA 35 – TELA TIPO DE SENHAS CADASTRADAS	81
FIGURA 36 – TELA ADICIONAR TIPO DE SENHA	81
FIGURA 37 – TELA CONFIGURAÇÕES CADASTRADAS.....	84
FIGURA 38 – TELA EDITAR CONFIGURAÇÃO.....	84
FIGURA 39 – TROCAR SENHA	86
FIGURA 40 - DER: DIAGRAMA DE ENTIDADE RELACIONAMENTO	88
FIGURA 41 – DIAGRAMA DE ENTIDADE RELACIONAMENTO (LÓGICO).....	89
FIGURA 42 – DIAGRAMA DE CLASSE.....	90
FIGURA 43 – DIAGRAMA DE SEQUÊNCIA: GERAR SENHA.....	91
FIGURA 44 – DIAGRAMA DE SEQUÊNCIA: ATENDER CLIENTE.....	92
FIGURA 45 – DIAGRAMA DE SEQUÊNCIA: CONSULTAR FILA	93
FIGURA 46 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR USUÁRIOS.....	94
FIGURA 47 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR POSTO	95
FIGURA 48 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR REGIÃO.....	96
FIGURA 49 – DIAGRAMA DE SEQUÊNCIA: GERAR RELATÓRIO	97
FIGURA 50 – DIAGRAMA DE SEQUÊNCIA: EFETUAR LOGIN.....	98
FIGURA 51 – DIAGRAMA DE SEQUÊNCIA: ANALISAR GRÁFICOS	99
FIGURA 52 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR TIPOS DE SENHA.....	100
FIGURA 53 – DIAGRAMA DE SEQUÊNCIA: MOVER SENHAS PARA HISTÓRICO	101
FIGURA 54 – DIAGRAMA DE SEQUÊNCIA: CONFIGURAR SISTEMA	102
FIGURA 55 – DIAGRAMA DE SEQUEÊNCIA: TROCAR SENHA	103

SUMÁRIO

RESUMO	6
ABSTRACT	7
1 INTRODUÇÃO	13
1.1 OBJETIVO GERAL	13
1.2 OBJETIVOS ESPECÍFICOS	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 FILAS DE ESPERA	15
2.2 TEORIA DAS FILAS	16
2.3 ELEMENTOS DO SISTEMA DE FILAS	16
2.3.1 Fonte populacional finita ou infinita.....	17
2.3.2 Características das filas.....	17
2.3.3 Disciplina da fila.....	18
2.3.4 Estruturas das instalações de serviço	18
2.3.5 Saída do sistema	19
2.4 LEGISLAÇÃO	20
2.4.1 Carta de serviço ao cidadão	20
2.4.2 Fila nos bancos e supermercados	20
2.4.3 Concessionárias de distribuição de energia elétrica.....	21
2.5 APLICAÇÕES SEMELHANTES	21
2.6 UML	22
2.7 ARQUITETURA MVC	23
2.8 TESTES	24
2.8.1 Caixa-branca	25
2.8.2 Caixa-preta	25
2.9 BANCO DE DADOS	26
3 METODOLOGIA	27
3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE	27
3.2 PLANO DE ATIVIDADES.....	28
3.3 PLANO DE RISCOS	31
3.4 RESPONSABILIDADES	32
3.5 MATERIAIS.....	33
3.5.1 Documentação e Análise:.....	33

3.5.2	Desenvolvimento:.....	34
3.5.3	Hardware:.....	34
3.6	DESENVOLVIMENTO DO PROJETO	35
3.6.1	Linguagem de programação	35
3.6.2	Arquitetura do software	36
4	APRESENTAÇÃO DO SOFTWARE	38
4.1	MÓDULO RECEPCIONISTA.....	39
4.1.1	Tela de recepção	39
4.1.2	Painel de senha	40
4.2	MÓDULO ATENDIMENTO.....	41
4.3	MÓDULO ADMINISTRADOR.....	43
4.3.1	Dashboard.....	43
4.3.2	Mapa	44
4.3.3	Relatórios	45
4.4	MÓDULO INTEGRAÇÃO	46
4.5	MONTAGEM DO AMBIENTE E INSTALAÇÃO DO SISTEMA.....	47
5	CONSIDERAÇÕES FINAIS.....	49
5.1	CONCLUSÃO.....	49
	REFERÊNCIAS.....	51
	APÊNDICES.....	52
	APÊNDICE A – ANÁLISE DOS REQUISITOS.....	52
	Requisitos Funcionais	52
	Requisitos não funcionais	57
	APÊNDICE B – DIAGRAMA DE CASO DE USO.....	59
	APÊNDICE C – ESPECIFICAÇÕES DE CASOS DE USO.....	60
	UC001 – Gerar Senha.....	60
	UC002 – Atender Cliente	61
	UC003 - Consultar fila	64
	UC004 – Gerenciar Usuários	66
	UC005 – Gerenciar Posto	69
	UC006 – Gerenciar Região	72
	UC007 – Gerar relatório	74
	UC008 – Efetuar login	76
	UC009- Analisar Gráficos.....	78

UC010 – Gerenciar tipos de senha.....	80
UC011 – Mover senhas para histórico.....	82
UC012 – Configurar Sistema	83
UC013 – Trocar senha.....	85
APÊNDICE D – DIAGRAMA DE ENTIDADE RELACIONAMENTO	88
APÊNDICE E – DIAGRAMA DE ENTIDADE RELACIONAMENTO (LÓGICO)	89
APÊNDICE F – DIAGRAMA DE CLASSES.....	90
APÊNDICE G – DIAGRAMAS DE SEQUÊNCIA	91
Diagrama de Sequência 001 – Gerar Senha.....	91
Diagrama de Sequência 002 – Atender Cliente.....	92
Diagrama de Sequência 003 - Consultar Fila	93
Diagrama de Sequência 004 – Gerenciar Usuários.....	94
Diagrama de Sequência 005 – Gerenciar Posto.....	95
Diagrama de Sequência 006 – Gerenciar Região	96
Diagrama de Sequência 007 – Gerar Relatório	97
Diagrama de Sequência 008 – Efetuar Login	98
Diagrama de Sequência 009 – Analisar Gráficos	99
Diagrama de Sequência 010 – Gerenciar tipos de senha.....	100
Diagrama de Sequência 011 – Mover senhas para histórico.....	101
Diagrama de Sequência 012 – Configurar Sistema.....	102
Diagrama de Sequência 013 – Trocar Senha.....	103
APÊNDICE H – WEB SERVICE RESTFUL.....	104

1 INTRODUÇÃO

As empresas, tanto de iniciativa pública como privada, prestam serviços que necessitam de atendimento ao público. Mesmo atualmente existindo canais de atendimento alternativos, como por exemplo telefone, chat e mídias-sociais, para determinados serviços ainda é necessário que as pessoas tenham que comparecer até um estabelecimento. Determinadas situações são imprescindíveis a presença *in loco*, seja para apresentar algum documento oficial ou realizar um exame médico. Nos postos de atendimento, quando a procura é maior que a demanda, há a formação de filas.

Para estabelecimentos de cunho comercial ou de prestação de serviços, caso o problema das filas não receba o devido tratamento, pode ocorrer na insatisfação e perda de um cliente, além de publicidade negativa.

Neste trabalho, foi desenvolvido o FUP – Fila de Unidade Presencial – que é uma aplicação com intuito de ser um software livre para ser utilizado como solução para atender diversas demandas de atendimento ao cliente. Sua aplicação pode ser estendida a qualquer posto de atendimento que necessite de um gerenciamento de qualidade.

1.1 OBJETIVO GERAL

O principal objetivo deste trabalho é desenvolver um sistema web para um melhor gerenciamento das filas de atendimento, tendo como foco principal o aumento da qualidade, organização e diminuir o tempo de espera seja no âmbito das redes públicas ou privadas.

1.2 OBJETIVOS ESPECÍFICOS

Este projeto tem como objetivos específicos:

- Projetar, desenvolver e implantar um sistema web, utilizando tecnologias como Java e JSF e metodologia de desenvolvimento iterativo e incremental;
- Trabalhar em equipe, visando adquirir experiência para o mercado de trabalho;
- Gerar gráficos e relatórios para o melhor controle das filas e dos atendentes nos postos de atendimento
- Colocar em prática os conhecimentos adquiridos durante o curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

2 FUNDAMENTAÇÃO TEÓRICA

Com os avanços da tecnologia, tanto na arquitetura dos computadores quanto nas linguagens de programação, considerando os diferentes ambientes de trabalho e diferentes formas de programar, se faz necessária uma constante evolução nas metodologias e engenharias utilizadas para desenvolver software, tendo como objetivo facilitar a manutenção dos sistemas, diminuir seus erros e riscos e encontrar as melhores formas de produzir, atendendo aos diferentes objetivos das aplicações. Cada software possui características que o fazem único, tal que, para desenvolvê-lo da melhor forma possível, é necessário avaliar qual a melhor tecnologia, o melhor paradigma, a melhor metodologia a ser seguida.

Este capítulo contempla uma apresentação dos conceitos aplicados no desenvolvimento deste projeto, tanto para a gerência e o planejamento, utilizando um modelo de desenvolvimento iterativo e incremental; quanto para a sua execução, utilizando a programação orientada a objetos – pois permite que o sistema seja programado para crescer, facilitando a reutilização de código e aproximando o sistema do mundo real – aliada ao modelo de arquitetura MVC e ao banco de dados PostgreSQL.

2.1 FILAS DE ESPERA

No momento em que os guichês não conseguem atender de forma efetiva a demanda de clientes que chegam ao estabelecimento, surgem as filas de espera. Um problema muito comum que tende a não ocorrer a todo momento, mas apenas em determinados instantes do dia. Apesar disso, o problema necessita ser tratado de alguma maneira.

Quando há as filas de clientes, para mantê-los e cumprir o atendimento com qualidade e agilidade, deixando o cliente satisfeito, há necessidade de se estabelecer programas para tratar de filas (MARQUES, 2012).

2.2 TEORIA DAS FILAS

A teoria das filas é uma mistura de probabilidade e formação das filas no dia a dia. Seu foco está em avaliar todo o comportamento do sistema de serviços de uma empresa que possua demanda aleatória, conseguindo assim medir de forma a satisfazer os clientes, evitando desentendimentos com a situação (GUEDES e ARAÚJO 2013).

A teoria das filas de espera tem como objetivo maximizar o fluxo das filas, encontrando uma boa solução, dentre diversos problemas como: tempo, qualidade do atendimento e uma boa organização para que os funcionários não fiquem muito tempo inativos do serviço. Pereira (2009)

Uma boa gerência de filas é importante para se ter publicidade positiva, saindo em vantagem quando se trata da competição com outras empresas, uma vez que, satisfaz as necessidades dos clientes.

2.3 ELEMENTOS DO SISTEMA DE FILAS

Os elementos de uma situação de fila são os clientes e os servidores, como mostrado na FIGURA 1. Ao chegarem a um estabelecimento podem iniciar o serviço imediatamente ou esperar em uma fila quando os guichês estão ocupados. Quando um guichê conclui o atendimento é chamado outro cliente da fila, se houver algum. Se a fila estiver vazia, o estabelecimento ficará ocioso até chegar um novo cliente (TAHA, 2008).

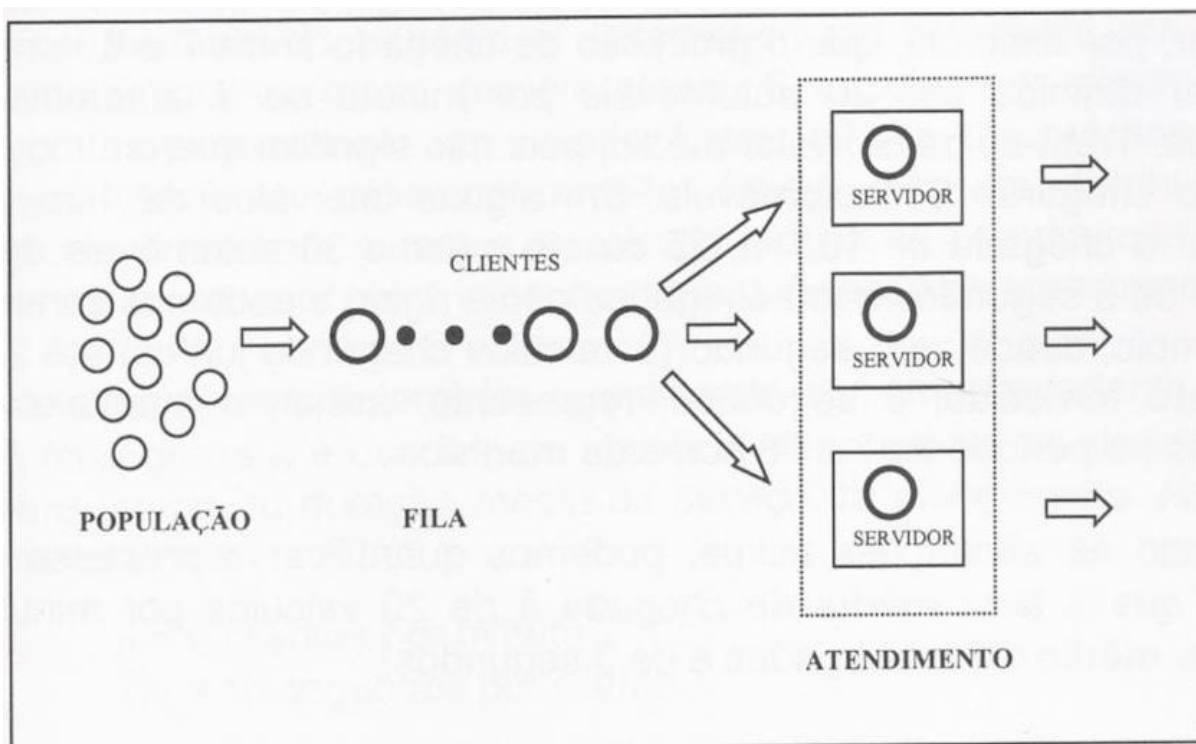


FIGURA 1 – ELEMENTOS DE UMA FILA
 FONTE: MARQUES (2012)

2.3.1 Fonte populacional finita ou infinita

Fonte Populacional infinita é uma população muito grande para não sofrer com desistências dos clientes que entram no sistema de filas. População finita representa um sistema com população pequena e fica pouco afetada pelos clientes que estão sendo atendidos pelos servidores (CORRÊA e CORRÊA, 2008).

2.3.2 Características das filas

As filas caracterizam-se de acordo com a extensão e número de filas. Quanto ao tamanho, por lei, pode ser com limite ou sem, dependendo da situação tratada.

As filas podem ser únicas ou filas múltiplas. Fila única é aquela em que todos os clientes aguardam em uma única fila. Filas múltiplas referem-se a várias filas

únicas, ou as filas que passam por um ponto central de redistribuição (DAVIS et al, 2001).

2.3.3 Disciplina da fila

É feito nas filas uma regra de prioridades ou um conjunto de regras que determinam a ordem que ocorrerá o atendimento dos clientes. As regras selecionadas podem aumentar muito o desempenho no sistema. O número do cliente na fila, o tempo médio de espera, a quantidade de cliente na fila e a eficiência no local de serviço são alguns fatores influenciados pela escolha das regras de prioridades. Geralmente a maioria dos sistemas de atendimento utilizam a regra primeiro a chegar, primeiro a ser atendido (PCPS). O cliente em primeiro lugar na fila tem a maior prioridade, enquanto o cliente que chegou por último tem a menor prioridade.

2.3.4 Estruturas das instalações de serviço

As instalações de serviços possuem as seguintes classificações, como mostrado na FIGURA 2:

- a) Fase única: todos os serviços prestados para o cliente podem ser feitos por um único atendente;
- b) Fases múltiplas: Os clientes formam uma fila única e seguem para outro caixa de atendimento, assim por diante.
- c) Arranjo de canais múltiplos e fase única: usado quando os serviços oferecidos pelo caixa são diferentes. Os clientes formam um ou mais filas, dependendo do projeto;
- d) Arranjo canais múltiplos e fases múltiplas: Os clientes podem ser atendidos por um dos caixas da primeira fase e depois precisam do atendimento de um caixa da próxima fase, assim por diante.

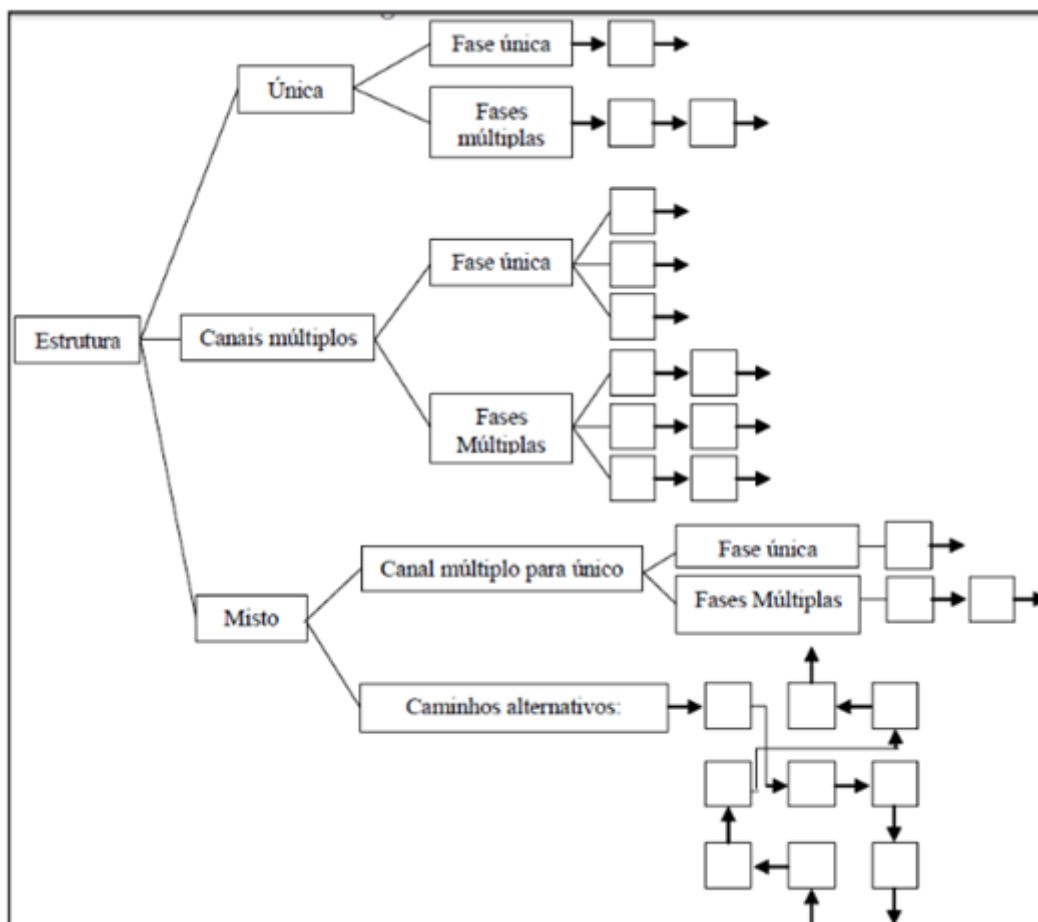


FIGURA 2 – ESTRUTURA DAS FILAS
 FONTE: ADAPTADO DE DAVIS ET AL (2001)

2.3.5 Saída do sistema

Após o cliente ser atendido, dois fluxos de saída são possíveis: o cliente pode retornar à fonte populacional e imediatamente tornar-se candidato para novo atendimento ou pode existir uma pequena probabilidade de retrabalho (DAVIS et al, 2001).

2.4 LEGISLAÇÃO

Há uma preocupação por parte do poder público em tentar garantir a qualidade do atendimento ao público, e não apenas do atendimento telefônico, mas também do atendimento presencial. Serão apresentados a seguir alguns exemplos de normas sobre o assunto.

2.4.1 Carta de serviço ao cidadão

O Decreto Nº 6.932 de 11 de agosto de 2009 instituiu a “Carta de serviços ao cidadão”, documento que cada órgão e entidade do Poder Executivo Federal que presta serviço ao cidadão deveria elaborar. A Lei determinou que a carta deveria conter informações que aumentassem a transparência dos sobre os serviços prestados, além de determinar padronizações na sistemática do atendimento. A Lei ainda determinou procedimentos para reduzir a burocracia no atendimento.

O Artigo 11 contém definições sobre conteúdos para aumentar a qualidade no atendimento, como por exemplo: serviços ofertados, requisitos para solicitações, prazo máximo para os serviços, locais onde os serviços podem ser solicitados, prioridades de atendimento e também tempo de espera.

A Lei demonstrou não apenas a preocupação do Estado com o atendimento ao público, mas também a necessidade que havia de melhorar.

2.4.2 Fila nos bancos e supermercados

No Estado do Paraná foi sancionada a Lei nº 13.400, de 21 de dezembro de 2001 que dispõe que as instituições bancárias e supermercados deverão providenciar medidas para efetivar atendimento a seus usuários em no máximo 20 minutos, sendo tolerável o tempo de até 30 minutos em vésperas e após feriados prolongados.

No Artigo 2º, a Lei determina que deve ser dado prioridade para os idosos, gestantes, pessoas portadoras de deficiência física e pessoas com crianças no colo. Diz ainda que o atendimento preferencial deverá ser realizado com senhas numéricas e o estabelecimento deve possuir no mínimo 15 assentos com encosto. No parágrafo único do mesmo artigo, há a determinação de que todo atendimento em bancos seja realizado através de senhas numéricas com registro de data e hora de emissão, para auxiliar no controle do tempo de espera.

2.4.3 Concessionárias de distribuição de energia elétrica

Em sua Resolução Normativa Nº 414, de 9 de setembro de 2010, a Agência Nacional de Energia Elétrica – ANEEL, define, do Artigo 177 até o 182, regras exclusivas sobre a estrutura do atendimento presencial das empresas concessionárias de distribuição de energia elétrica.

Um dos itens, por exemplo, é a determinação que as empresas devem possuir postos de atendimento em todos os municípios de sua área de concessão. Tomando como exemplo local, a COPEL – Companhia Paranaense de Energia Elétrica (2016), empresa que presta atendimento a quase todo território paranaense, possui 402 postos de atendimento, onde alguns municípios maiores possuem mais de um posto em seu território.

A Resolução Normativa definiu ainda como 45 minutos o tempo máximo que um cliente deve aguardar na fila de espera. Por ser o maior tempo que encontramos sendo referenciado em alguma norma, utilizamos este prazo como parâmetro de tempo de espera no aplicativo FUP.

2.5 APLICAÇÕES SEMELHANTES

Tendo como foco os preceitos que desejamos para o FUP (aplicação web e software livre), foi realizada a busca de aplicações semelhantes disponíveis no mercado. Existe uma oferta expressiva de softwares embarcados e pagos, enquanto

apenas um encontrado possuía os requisitos desejados, trata-se do Novo SGA (NOVO SGA, 2016).

O Novo SGA é uma aplicação web já em sua segunda versão. Foi desenvolvido em PHP, assemelha-se bastante em relação ao Fila de Unidade Presencial. Em comparação, verificamos os seguintes aspectos:

Vantagens:

1. Arquitetura mais leve, indicado para uso moderado;
2. Mecanismo operacional de atendimento com mais recursos, as senhas são classificadas na conclusão;

Desvantagens:

1. Não indicado para uso mais pesado, a estrutura Java EE do FUP é mais adequada por possibilitar maior escalabilidade;
2. Relatório estático em PDF. O FUP exporta arquivo em formato de planilha, é possível trabalhar com as informações facilmente;
3. Não possui Dashboard;
4. Não possui visualização de mapa;

Devido ao aspecto não gratuito das demais soluções verificadas, não foi possível realizar testes e comparar as aplicações. Seguem alguns exemplos:

1. Specto (SPECTO, 2016): Software embarcado, senhas são chamadas através de controles-remoto;
2. Gautek (GAUTEK, 2016): Empresa possui soluções próprias e também realiza acoplamento com o SGA;
3. Guarda Fila (GUARDA FILA, 2016): Aplicação web paga;

2.6 UML

A UML (Unified Modeling Language) é uma linguagem-padrão usada para a composição da estrutura de projetos de software. Ela é adequada para a modelagem de sistemas, cuja abrangência inclui desde sistemas simples, como uma aplicação web simples, até sistemas complexos embutidos em tempo real. UML é apenas uma linguagem, ou seja, é só um pedaço de um método para desenvolvimento de software e é independente do processo, podendo ser perfeitamente utilizada em processo

orientado a casos de usos, centrado na arquitetura, iterativo e incremental (BOOCH et al., 2006).

Um diagrama é a apresentação gráfica de um conjunto de elementos, desenhados para permitir a visualização de um sistema sob diferentes perspectivas. Em UML, podemos encontrar 13 tipos diferentes de diagramas, mas os mais comumente utilizados são (BOOCH et al., 2006):

- *Diagrama de classe*: exibe um conjunto de classes, interfaces e colaborações, bem como seus relacionamentos. Encontrados com muita frequência em sites de modelagem orientados a objeto.
- *Diagrama de componentes*: abrange a visão de implementação do projeto estático do sistema. São importantes para criar grandes sistemas a partir de partes menores.
- *Diagrama de caso de uso*: abrange a visão estática de casos de uso do sistema, sendo importantes para a organização e modelagem de comportamentos do sistema.
- *Diagrama de sequência*: exibe uma interação através de um conjunto de objetos ou papéis, incluindo as mensagens que podem ser trocadas entre eles. É um diagrama de interação cuja ênfase está na ordenação temporal dessas mensagens.
- *Diagrama de atividade*: abrange a visão dinâmica do sistema, tornando-se importante principalmente para a modelagem da função de um sistema e ao mesmo tempo dando ênfase ao fluxo de controle entre objetos.

2.7 ARQUITETURA MVC

O modelo de arquitetura adotado foi o modelo de três camadas que tem por objetivo separar um projeto em camadas, o mais conhecido é o modelo MVC que possui as camadas e ligações representadas na figura abaixo.

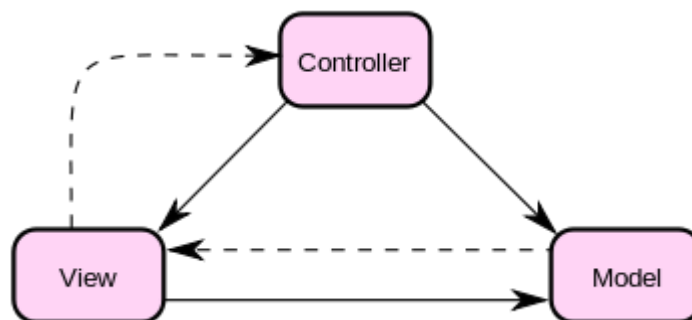


FIGURA 3 – MVC

A camada de modelo (Model) é responsável pelo acesso aos locais de armazenamento, isolando-os de tal forma que se for trocado, apenas as classes pertencentes a essa camada deverão ser alteradas. Essa camada é responsável em manter os dados da camada de apresentação atualizados.

A camada de visão (View) é a camada de interface, na qual o sistema é apresentado ao usuário. Será por meio dela que haverá as interações entre homem/máquina. Essa camada é responsável em notificar a camada de controle referente as mudanças solicitadas pelos usuários.

A camada de controle (Controller) é onde são definidas as regras de negócio do sistema, essa é a camada intermediária situada entre as camadas de apresentação e o modelo. As requisições da camada de apresentação para a camada de modelo devem passar pela camada de controle.

A arquitetura MVC apresenta diversas vantagens, como por exemplo, faz com que o código seja desenvolvido de forma mais rápida e com menor custo. As informações trocadas entre as classes são formadas de acordo com um padrão para que exista troca de mensagens entre elas. Esse tipo de arquitetura tem por objetivos aumentar a modularização do sistema e reduzir os custos de manutenção diminuindo o tempo para que correções e alterações sejam efetuadas

2.8 TESTES

O objetivo do teste é encontrar erros, tal que um bom teste é aquele que tem alta probabilidade de encontrar um erro. Neste trabalho foram executadas as técnicas de testes conhecidas como caixa-branca e caixa-preta. Os testes caixa-

branca foram realizados a nível de unidade, de forma que a cada funcionalidade programada, testes básicos eram executados para verificar o código, ao passo que os testes caixa-preta foram utilizados ao final do desenvolvimento de cada módulo, testando as funcionalidades através da interface, e também como testes de integração, verificando o funcionamento do sistema entre os módulos.

2.8.1 Caixa-branca

De acordo com Pressman (PRESSMAN, 1995), o teste de caixa-branca é baseado em um exame rigoroso do detalhe procedimental do software. Examinam-se caminhos lógicos internos e colaborações entre componentes.

Seu objetivo é garantir que todos os caminhos independentes de um módulo tenham sido executados pelo menos uma vez;

- Exercitam todas as decisões lógicas em seus lados verdadeiro e falso;
- Executem todos os ciclos (loops) nos seus limites e dentro de seus intervalos operacionais;
- Executem suas estruturas de dados internas.

2.8.2 Caixa-preta

Teste de caixa-preta refere-se a testes que são conduzidos na interface do software. Examina-se a funcionalidade do sistema, sem se preocupar com a estrutura interna (PRESSMAN, 1995).

- Focaliza os requisitos funcionais do software;
- Permite ao engenheiro de software derivar conjuntos de condições de entrada que exercitem plenamente todos os requisitos funcionais de um programa;
- É complementar ao teste de caixa-branca.

2.9 BANCO DE DADOS

O PostgreSQL, escolhido para o desenvolvimento deste projeto, é um sistema gerenciador de banco de dados relacional que utiliza a linguagem SQL (Structured Query Language) como interface. O PostgreSQL é otimizado para aplicações complexas, isto é, que envolvem grandes volumes de dados ou que tratam de informações críticas. Assim, para um sistema de comércio eletrônico de porte médio/alto, por exemplo, o PostgreSQL é mais interessante, já que esse SGBD é capaz de lidar de maneira satisfatória com grandes volumes de dados.

Tecnicamente falando, o PostgreSQL é um banco de dados relacional e orientado a objetos. Um de seus atrativos é possuir recursos comuns a banco de dados de grande porte, o que o deixa apto a trabalhar, inclusive, com operações de missão crítica. Além disso, trata-se de um banco de dados versátil, seguro, gratuito e de código aberto.

Entre suas características, tem-se:

- Compatibilidade multi-plataforma, ou seja, executa em vários sistemas operacionais, como Windows, Mac OS X, Linux e outras variantes de Unix;
- Compatibilidade com várias linguagens, entre elas, Java, PHP, Python, Ruby, e C/C++;
- Base de dados de tamanho ilimitado;
- Suporte a recursos como triggers, views, stored procedures, SSL, MVCC, schemas, transactions, savepoints, referential integrity e expressões regulares;
- Instruções em SQL, como indica o nome.

3 METODOLOGIA

Como descrito anteriormente, para o desenvolvimento desse trabalho optou-se pelo desenvolvimento iterativo e incremental, método onde o desenvolvimento do software é dividido em ciclos com a presença das fases de análise, projeto, implementação e testes em cada iteração.

Utilizou-se para planejamento das atividades a WBS (Work Breakdown Structure) e a tabela de precedências junto ao gráfico de Gantt como cronograma, acompanhamento dos prazos e a organização da divisão de responsabilidades para a realização das atividades.

3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

Partindo do modelo iterativo e incremental, o projeto foi organizado em iterações que abordam as seguintes fases: levantamento de requisitos, análise dos requisitos, projeto, implementação, testes e implantação.

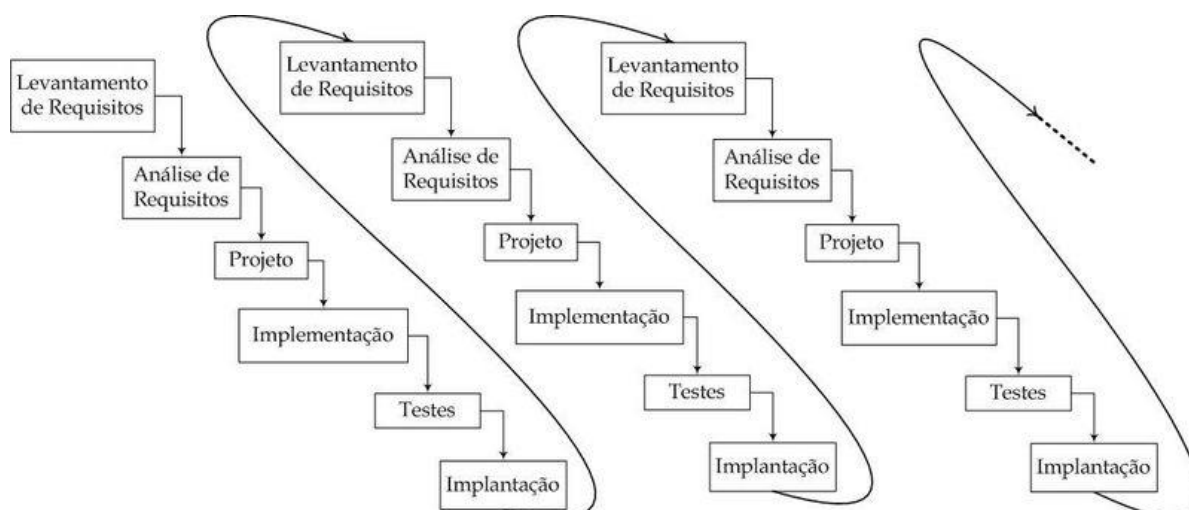


FIGURA 4 – FASES DE CADA ITERAÇÃO
FONTE: IFSC

No início de cada iteração é realizado o levantamento dos requisitos pertinentes, cria-se um projeto, é realizada a implementação e no fim do ciclo há a

verificação se os requisitos foram atingidos. Caso os objetivos estejam de acordo, inicia-se a próxima iteração, caso contrário, realiza-se uma nova iteração com os mesmos requisitos utilizando uma abordagem diferente.

O modelo iterativo incremental é considerado uma evolução para suprir falhas do método cascata, apresentando por exemplo as seguintes vantagens:

- Baixa complexidade;
- Pode ser realizado com número reduzido de profissionais;
- Cada iteração produz resultados concretos;
- Redução nos riscos por ser mais fácil administrar iterações de curta duração e também pelo fato de não ser possível especificar todas as necessidades logo no início de um projeto.

Entretanto, as características deste modelo também apresentam algumas desvantagens:

- Risco de o usuário ficar com a expectativa muito alta por começar a perceber resultados muito rápido;
- Não é possível definir a quantidade de iterações necessárias de um projeto, logo, o fim do projeto também tende a ser imprevisível;
- Dificuldade de gerenciamento pois algumas fases da iteração podem ocorrer simultaneamente.

As duas próximas seções apresentam a WBS e o Gráfico de Gantt, utilizados para organização das atividades acima descritas, demonstrando também sua distribuição ao longo do tempo.

3.2 PLANO DE ATIVIDADES

Para a organização das atividades realizadas, foi utilizada a WBS (Work Breakdown Structure), também conhecida como EAP (Estrutura Analítica do Projeto), apresentando as fases supra descritas do projeto (Figura 4). Cada uma das fases contém as atividades realizadas nelas e/ou os produtos gerados, estimou-se um total aproximadamente de um mês para o projeto, e as responsabilidades foram atribuídas

conforme as aptidões dos membros do grupo. Essa divisão de tempo e recursos está detalhada no Gráfico de Gantt (Figura 6).

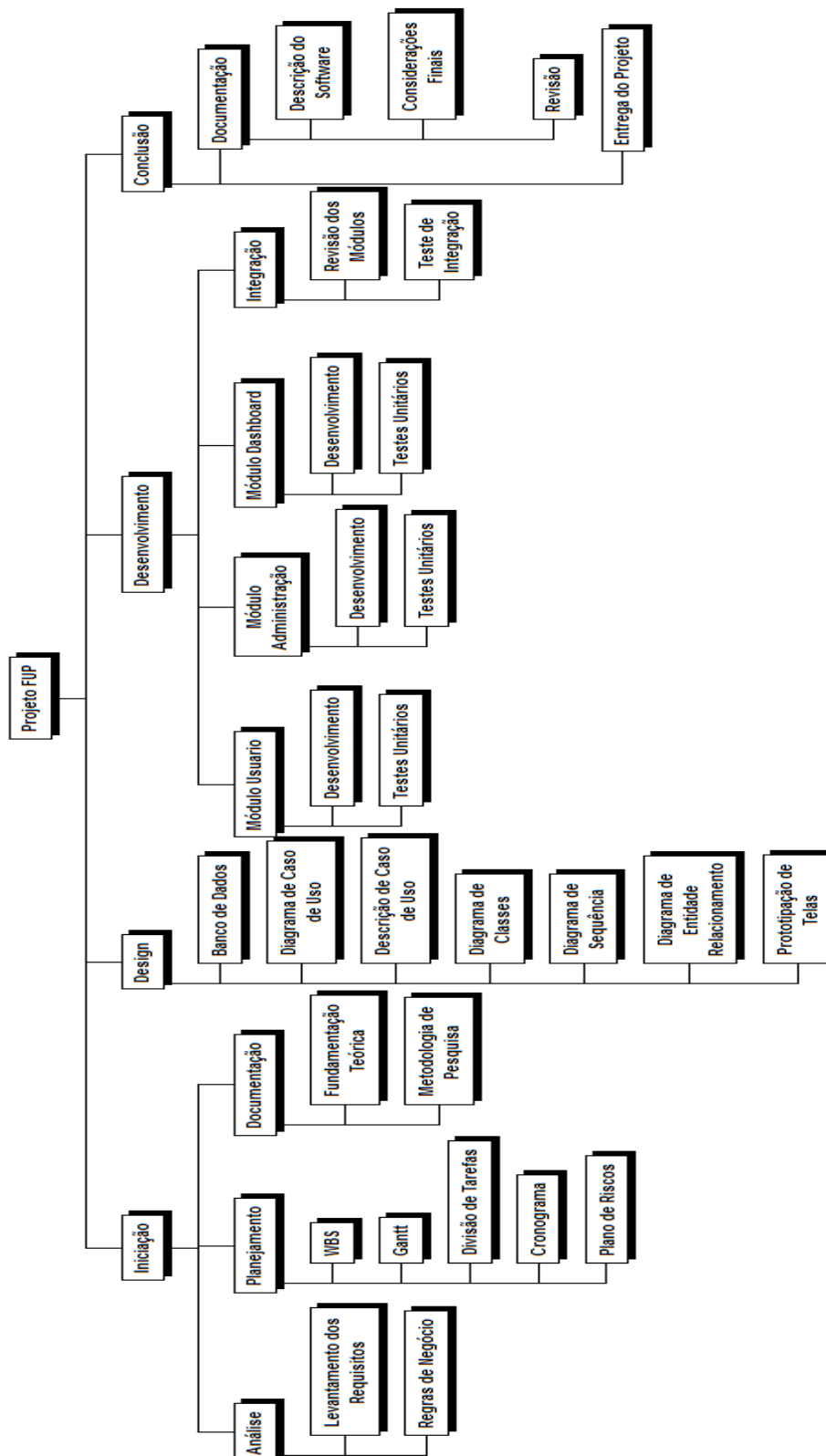


FIGURA 5 – WORK BREAKDOWN STRUCTURE
 FONTE: OS AUTORES (2016)

	📍	Nome	Duração	Início	Término	Predecessoras
1		FUP	20 dias?	01/09/16 08:00	28/09/16 17:00	
2		Iniciação	6 dias?	01/09/16 08:00	08/09/16 17:00	
3		Análise	1 dia?	01/09/16 08:00	01/09/16 17:00	
4		Regras de Negócios	1 dia?	01/09/16 08:00	01/09/16 17:00	
5		Levantamento de Requisitos	1 dia?	01/09/16 08:00	01/09/16 17:00	
6		Planejamento	3 dias?	01/09/16 08:00	05/09/16 17:00	
7		WBS	0,5 dias?	05/09/16 08:00	05/09/16 13:00	10
8		Gráfico de Gantt	0,5 dias?	05/09/16 13:00	05/09/16 17:00	7
9		Divisão de Tarefas	0,5 dias?	01/09/16 08:00	01/09/16 13:00	
10		Plano de Riscos	0,5 dias?	02/09/16 13:00	02/09/16 17:00	11
11		Cronograma	1 dia?	01/09/16 13:00	02/09/16 13:00	9
12		Documentação	3 dias?	06/09/16 08:00	08/09/16 17:00	
13		Fundamentação Teórica	1,5 dias?	06/09/16 08:00	07/09/16 13:00	8
14		Metodologia de Pesquisa	1,5 dias?	07/09/16 13:00	08/09/16 17:00	13
15		Design	13,5 dias?	01/09/16 08:00	20/09/16 13:00	
16		Banco de Dados	5 dias?	01/09/16 08:00	07/09/16 17:00	
17		Diagrama de Caso de Uso	0,5 dias?	12/09/16 13:00	12/09/16 17:00	21
18		Diagrama de Classes	0,5 dias?	13/09/16 08:00	13/09/16 13:00	17
19		Diagrama de Sequência	5 dias?	13/09/16 13:00	20/09/16 13:00	18
20		Diagrama de Entidade Relacionamento	0,5 dias?	09/09/16 08:00	09/09/16 13:00	14
21		Prototipação de Telas	1 dia?	09/09/16 13:00	12/09/16 13:00	20
22		Especificação Caso de Uso	5 dias?	12/09/16 13:00	19/09/16 13:00	21
23		Desenvolvimento	13 dias?	01/09/16 08:00	19/09/16 17:00	
24		Módulo Cliente	8 dias?	01/09/16 08:00	12/09/16 17:00	
25		Desenvolvimento	7 dias?	01/09/16 08:00	09/09/16 17:00	
26		Testes Unitários	1 dia?	12/09/16 08:00	12/09/16 17:00	25
27		Módulo Administração	8 dias?	01/09/16 08:00	12/09/16 17:00	
28		Desenvolvimento	7 dias?	01/09/16 08:00	09/09/16 17:00	
29		Testes Unitários	1 dia?	12/09/16 08:00	12/09/16 17:00	28
30		Módulo Dashboard	8 dias?	01/09/16 08:00	12/09/16 17:00	
31		Desenvolvimento	7 dias?	01/09/16 08:00	09/09/16 17:00	
32		Testes Unitários	1 dia?	12/09/16 08:00	12/09/16 17:00	31
33		Integração	5 dias?	13/09/16 08:00	19/09/16 17:00	
34		Revisão dos Módulos	2 dias?	13/09/16 08:00	14/09/16 17:00	32
35		Testes de Integração	3 dias?	15/09/16 08:00	19/09/16 17:00	34
36		Conclusão	7 dias?	20/09/16 08:00	28/09/16 17:00	
37		Documentação	7 dias?	20/09/16 08:00	28/09/16 17:00	
38		Descrição do Software	1,5 dias?	20/09/16 08:00	21/09/16 13:00	35
39		Considerações Finais	1,5 dias?	21/09/16 13:00	22/09/16 17:00	38
40		Revisão	3 dias?	23/09/16 08:00	27/09/16 17:00	39
41		Entrega do Projeto	1 dia?	28/09/16 08:00	28/09/16 17:00	40

FIGURA 6 – GRÁFICO DE GANTT COM TABELA DE PRECEDÊNCIAS
 FONTE: OS AUTORES (2016)

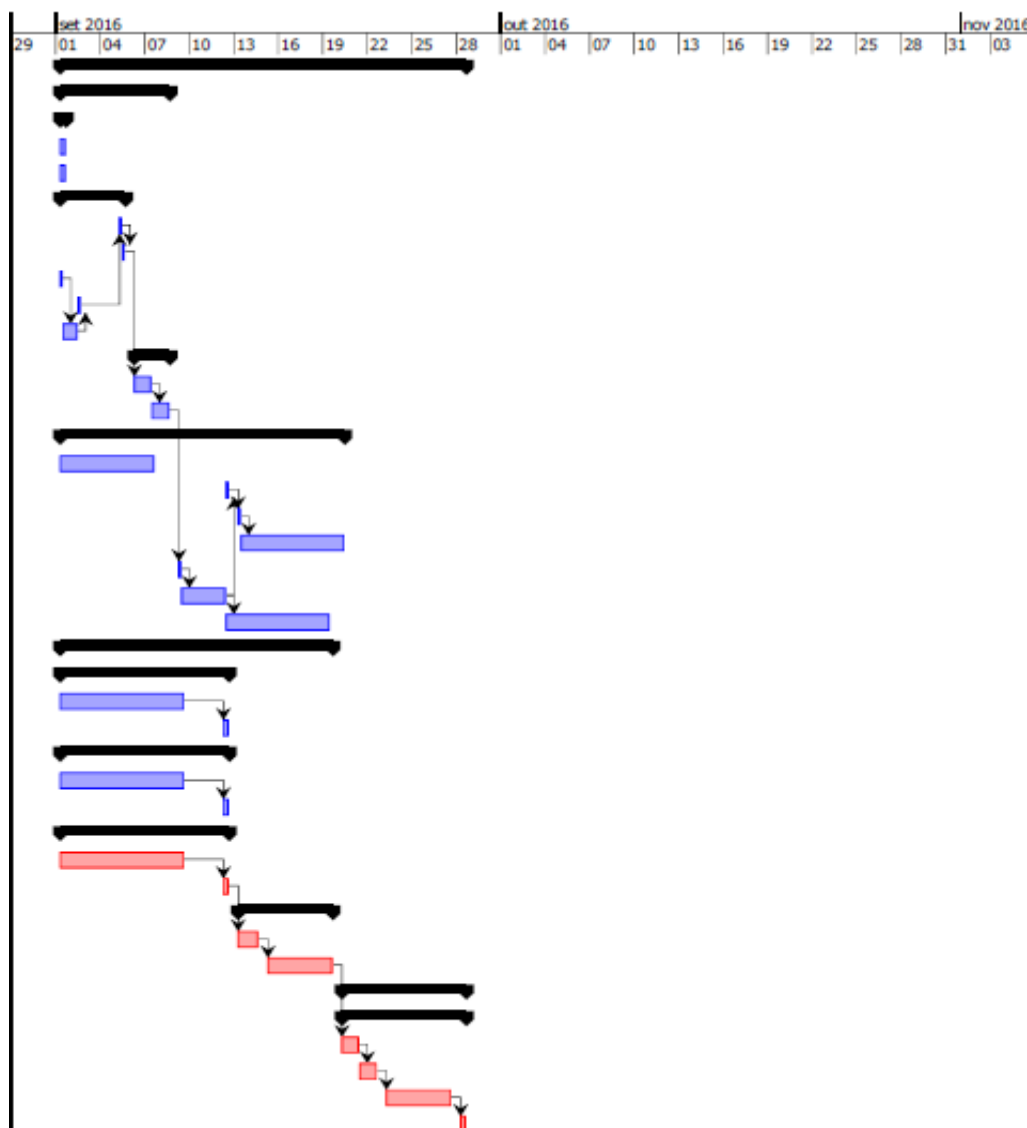


FIGURA 7 – ANÁLISE DO CAMINHO CRÍTICO
 FONTE: OS AUTORES (2016)

3.3 PLANO DE RISCOS

O Gerenciamento de Riscos é o meio pelo qual as incertezas em relação ao projeto são sistematicamente gerenciadas, permitindo avaliar e enfrentar os riscos, evitando o evitável, controlando o controlável e minimizando as imprevisibilidades. As principais formas de identificar riscos são: brainstorming com os participantes do projeto, analisando premissas e restrições, consultando especialistas ou pessoas com experiências em projetos similares (MARTINS, 2007).

No plano de risco situado abaixo, foram descritas as condições de risco que poderiam ter efeito no desenvolvimento do trabalho, suas consequências e o plano de ação que deveria ser empregado caso houvesse necessidade. Também foi descrito sua probabilidade de ocorrência e classificação

	Condição	Data Limite	Consequência	Ação	Probabilidade	Impacto	Classificação
1	Falta de comunicação	Sem data	Atraso na execução de certas atividades	Fazer reuniões de membros da equipe	Baixa	Alto	5
2	Cronograma incorreto	Sem data	Atraso nas entregas e aumento da rotina de trabalho	Reajuste do cronograma	Moderada	Alto	6
3	Doença de membro da equipe	Sem data	Atraso na execução das atividades do membro doente	Dividir as atividades do membro doente até que ele retorne	Moderada	Moderado	5
4	Luto Familiar	Sem data	Possível atraso nas entregas das atividades do integrante em luto	Aumento da carga de trabalho; reorganização de horários	baixa	baixo	3
5	Saída de membro da equipe	Sem data	Sobrecarga de trabalho e atraso nas entregas	Realocação de atividades	Baixa	Muito alto	6
6	Requisitos pouco precisos	Sem data	Baixa especificação de Desenvolvimento	Reuniões para melhor esclarecimento	Baixa	Moderado	4
7	Mudança no prazo final de entrega	30 set	Aumento na rotina de trabalho e risco de atraso	Reajuste do cronograma e aumento da carga de trabalho	Alta	Moderado	6

FIGURA 8 – PLANO DE RISCOS
FONTE: OS AUTORES (2016)

3.4 RESPONSABILIDADES

As responsabilidades pessoais dos integrantes da equipe foram atribuídas da seguinte maneira:

- Bruno Pereira Martins: Levantamento de requisitos, elaboração dos casos de uso, especificação dos casos de uso, execução dos testes de integração, elaboração da documentação.

- Hugo Hideo Fujiwara: levantamento de requisitos, desenvolvimento, prototipação das telas, execução dos testes unitários.
- Gustavo Benjamin Virmond Ferreira: Levantamento de requisitos, modelagem de dados, confecção do diagrama de sequência, confecção do diagrama de classes execução dos testes de integração, elaboração da documentação.

3.5 MATERIAIS

Descrição de hardware e software utilizados para o desenvolvimento e documentação do projeto:

3.5.1 Documentação e Análise:

- **Microsoft Office Word 2013**, editor de textos utilizado para a documentação do projeto;
- **Astah Community 7.0.0**, utilizado para a elaboração do diagrama de casos de uso e diagrama de sequência;
- **NClass 2.4.1823**, utilizado para a criação do Diagrama de Classe.
- **brModelo**, utilizado para a elaboração do diagrama de entidade relacionamento;
- **WBS Chart Pro**, utilizado para a confecção do gráfico de atividades do projeto;
- **OpenProj**, utilizado para o desenvolvimento do gráfico de Gantt, com o planejamento das atividades do projeto e tabela de precedências.

3.5.2 Desenvolvimento:

- **JDK** (Java SE Development Kit) 8 update 102.
- **pgAdmin III versão 1.22.1**, utilizado para desenvolvimento do banco de dados do sistema, tendo como base o PostgreSQL 9.5.
- **Red Hat JBoss Developer Studio versão 10.0.0.GA**, IDE utilizada para o desenvolvimento do software.
- **JBoss Enterprise Application Platform (JBoss EAP) versão 7**, utilizado como servidor de aplicações Java Enterprise Edition (Java EE).
- **PrimeFaces Community 6.0**, biblioteca javascript com componentes JSF.
- **Hibernate 5.0.9 Final**, framework utilizado como interface com o banco de dados.

3.5.3 Hardware:

- Notebook Dell com sistema operacional Windows 7 Ultimate 64-bits, processador Intel Core i5 CPU M 2430 2.40GHz, memória RAM de 6GB DDR3, placa de vídeo Intel HD Graphics;
- Notebook LG com sistema operacional Windows 10 Home Single Version, processador Intel Core i7 3517U 2.40GHz, memória RAM de 12GB DDR3, placa de vídeo Intel HD Graphics 4000.
- Notebook Acer com sistema operacional Windows 10 Pro 64-bits, processador Intel Core i3-4010U 1.7GHz, memória RAM de 4GB DDR3, placa de vídeo Intel HD Graphics 4400.

3.6 DESENVOLVIMENTO DO PROJETO

Na fase inicial do projeto foram realizadas reuniões para levantar os requisitos do sistema, bem como discutir regras de negócio. Após a definição destes, houve a atribuição de tarefas levando em conta o conhecimento e familiaridade dos membros com os processos de desenvolvimento do sistema. Durante o restante dessa fase inicial, ocorreram reuniões semanais entre os membros da equipe para a definição do escopo e confecção inicial dos diagramas. As reuniões com o orientador ocorreram de forma semanal, apresentando o que foi desenvolvido até o momento da reunião e definindo as próximas atividades para cada membro até a próxima reunião.

Após essa etapa de planejamento e definições de escopo, todas as atividades já estavam bem definidas para cada membro e as reuniões entre estes passaram a ser semanais, nas terças-feiras, para que pudessem debater entre si o que cada um havia feito e realizar algum desenvolvimento em conjunto. As reuniões com o orientador tornaram-se toda quarta-feira, para acompanhar o andamento do projeto e sanar as dúvidas que foram surgindo. Na semana em que não ocorria o encontro com o professor orientador, era encaminhado um e-mail para ele com a situação atual e os documentos atualizados.

Este ritmo de andamento dos trabalhos fluiu até aproximadamente a terceira semana, quando por um motivo pessoal de um dos integrantes, houve a necessidade de antecipar o prazo final do projeto para no máximo quatro semanas, contadas a partir daquele momento. Em um ato de compreensão dos integrantes da equipe e do coordenador, o desafio foi aceito e houve uma alteração brusca em todo o planejamento e escopo do projeto.

A etapa de desenvolvimento do sistema deu-se da seguinte maneira:

3.6.1 Linguagem de programação

Houve consenso na escolha do Java como linguagem de programação para o FUP, pois é de uso gratuito e possui uma grande oferta de bibliotecas, frameworks, IDEs e demais acessórios também gratuitos e muito poderosos.

Outro motivo foi a intimidade com a linguagem que todos os integrantes da equipe apresentavam.

3.6.2 Arquitetura do software

Foi decidido que a aplicação seria construída com a utilização de padrões de desenvolvimento que fossem coerentes e que fossem de conhecimento dos desenvolvedores, reduzindo assim o risco do não cumprimento do prazo de entrega.

A aplicação foi montada na IDE utilizando quatro projetos:

1. Projeto Client: Possui as entidades de JPA (Java Persistence API), POJOs (Plain Old Java Objects), DTOs (Data Transfer Objects) e interfaces. Este projeto é utilizado como uma biblioteca para os demais.
2. Projeto Web: É a camada da interface com o usuário. Utiliza JSF (Java Server Faces) e a biblioteca Primefaces. Os textos, nomes de campos e mensagens de tela foram separados em um arquivo de propriedades (Message Bundle). A segurança do controle de acesso à aplicação e também aos Web Services foram realizadas implementando Filtro do pacote Servlet. Os campos que necessitam de validação, foram implementados com validações utilizando AJAX e as telas foram confeccionadas com responsividade para utilização em dispositivos móveis.
3. Projeto EJB: Possui os DAOs (Data Access Object) e Serviços, que são EJBs (Singletons e Stateless) e realizam Injeção de Dependência entre si e na camada web. Implementam interfaces do projeto client.
4. Projeto EAR: Tem a função de unificar os três projetos anteriores.

Como IDE foi utilizado o Red Hat JBoss Developer Studio devido ao seu pacote completo de ferramentas e também de ser gratuito para o desenvolvimento de aplicações. A escolha do JBoss EAP como servidor de aplicação foi pelo mesmo motivo, muito embora possa ser utilizado o JBoss Wildfly ou outro container Java EE sem problemas, pois não houve a utilização de biblioteca exclusiva do JBoss.

O banco de dados escolhido foi o PostgreSQL devido suas características de capacidade de processamento de grandes volumes de dados, integridade referencial

e gratuidade. A camada de persistência do FUP utiliza Data Source com JTA (Java Transaction API) e Hibernate. Com isso, as conexões são provenientes de um pool de conexões configuradas no container, nesse caso o JBoss.

4 APRESENTAÇÃO DO SOFTWARE

O sistema Fila de unidade presencial foi desenvolvido tendo como base quatro módulos: usuário, administrador, dashboard e por último o de integração. Após o desenvolvimento, os módulos foram ajustados conforme funcionalidades por perfil, onde o módulo de usuário foi separado em recepcionista e atendente, enquanto os módulos administrador e dashboard foram unificados, ficando da seguinte maneira:

1. Módulo recepcionista: Módulo com menos funcionalidades;
2. Módulo atendente: Possui as funcionalidades do módulo recepcionista e também outras próprias;
3. Módulo administrador: Possui as mesmas funcionalidades que os módulos anteriores e mais funcionalidades próprias;
4. Módulo integração: Conjunto de serviços web que possibilitam a integração do FUP com outros sistemas.

Ao realizar o login na aplicação, o sistema exibe apenas as funções pertinentes à cada perfil de acesso.

As funcionalidades do sistema para cada perfil de usuário estão presentes em um menu objetivo, com opções pertinentes a cada módulo, permitindo que o usuário as acesse facilmente a qualquer momento. As telas possuem um padrão intuitivo, simples e com responsividade, facilitando a interação do usuário com o sistema, mantendo, nos três módulos, um mesmo padrão de estrutura de páginas. Todos esses atributos contribuem para agregar valor ao uso do sistema, por isso, ele poderia ser facilmente utilizado por diversos perfis de usuários, considerando as suas personalidades e formas de usar o sistema.

4.1 MÓDULO RECEPCIONISTA

4.1.1 Tela de recepção

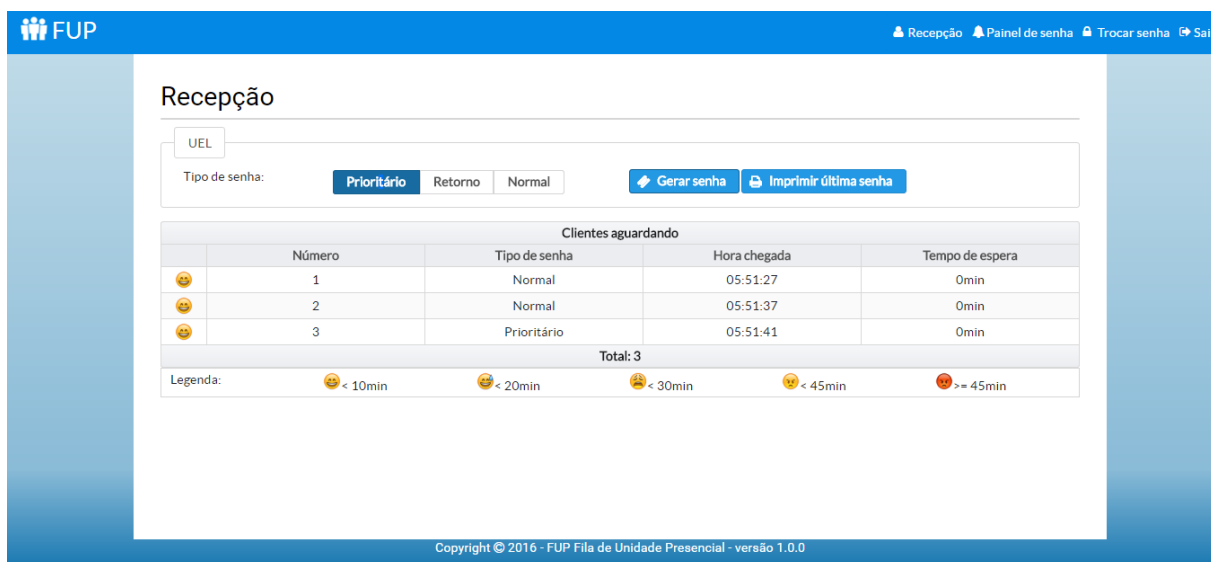


FIGURA 9 – MÓDULO RECEPÇÃO
 FONTE: OS AUTORES (2016)

No módulo recepcionista, o usuário interage com o cliente para realizar uma triagem, gera uma senha adequada e a entrega ao cliente. Além disso, ele pode na mesma tela de geração de senha visualizar a fila de espera, observando a sua prioridade e a quanto tempo cada um está aguardando.

4.1.2 Painel de senha

Últimas chamadas	
Número	Guichê
3	1
9	1
8	2
5	1
2	1
1	1

06:02:08

FIGURA 10 – PAINEL DE SENHAS
FONTE: OS AUTORES (2016)

No painel de senha o cliente pode visualizar facilmente a última senha chamada em destaque, além do histórico das últimas seis senhas. Além disso, o painel emite um alerta sonoro.

4.2 MÓDULO ATENDIMENTO

Atendimento

UFPR SEPT

Número: 4
Tipo de senha: Normal
Status da senha: Chamada
Tempo de espera: 6min

Guichê

Iniciar atendimento
Cancelar senha

Clientes aguardando

	Número	Tipo de senha	Hora chegada	Tempo de espera
😊	7	Normal	06:00:31	7min
😊	10	Normal	06:00:35	7min
😊	11	Normal	06:07:13	0min
😊	12	Prioritário	06:07:19	0min
Total: 4				

Legenda: 😊 < 10min 😊 < 20min 😊 < 30min 😊 < 45min 😊 >= 45min

Clientes em atendimento

	Número	Tipo de senha	Guichê	Hora chamada	Hora início atendimento	Tempo de espera	Tempo em atendimento
😊	4	Normal	1	06:06:31		6min	N/A
😊	6	Normal	2	06:07:07	06:07:07	6min	0min
Total: 2							

Atendimentos finalizados

Usuário	Concluídos	Cancelados
atendente	5	0
hugo	1	0

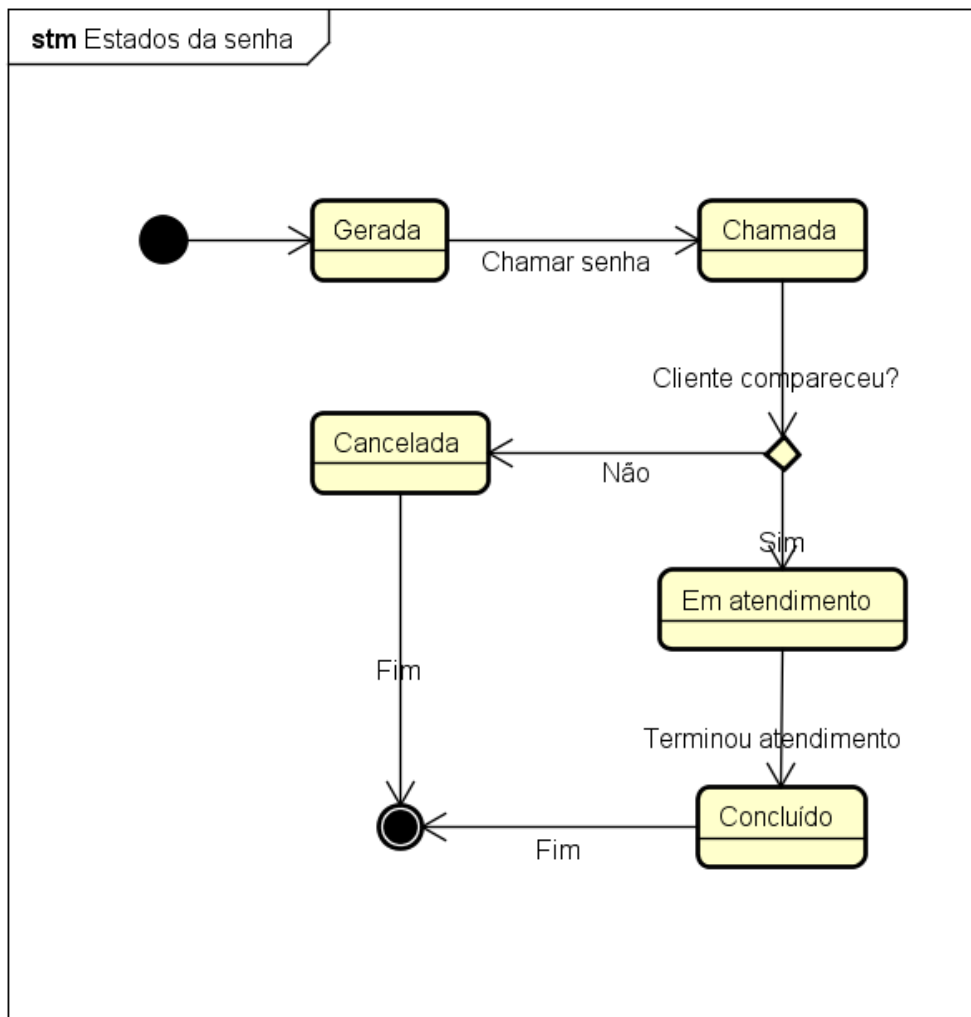
Copyright © 2016 - FUP Fila de Unidade Presencial - versão 1.0.0

FIGURA 11 – MÓDULO ATENDIMENTO
FONTE: OS AUTORES (2016)

O usuário atendente tem a função de “consumir” senhas. Ao chamar uma senha, o FUP verifica qual das senhas que estão aguardando possui a maior prioridade e a chama. A partir desse momento, quando o cliente chega ao guichê para o atendimento, o atendente deve pressionar o botão de “início de atendimento”, e ao término, o botão de conclusão. Caso o cliente não compareça com a chamada, o atendente deve utilizar a função de cancelamento de senha.

Ainda na tela de atendimento, os funcionários pertencentes ao posto podem visualizar as senhas que estão aguardando, as que estão sendo atendidas (por quem e a quanto tempo) e também visualizar a quantidade de atendimento realizado por cada funcionário. Com isso, há uma certa competição entre os funcionários devido à publicidade dos dados e informações, gerando um aumento na produtividade.

O ciclo de vida da senha é representado pelos seguintes estados:



powered by Astah

FIGURA 12 – ESTADOS DA SENHA
 FONTE: OS AUTORES (2016)

Os intervalos de tempo entre as alterações de estado da senha geram informações relevantes da senha:

1. Tempo de espera: Gerada -> Chamada;
2. Tempo de deslocamento: Chamada -> Em atendimento;
3. Tempo de atendimento: Em atendimento -> Concluído;
4. Tempo de permanência: Gerada -> Concluído;

4.3 MÓDULO ADMINISTRADOR

O usuário do módulo administrador é responsável pelo cadastramento das regiões, postos de atendimento, usuários, tipos de senha e também de parametrizar as configurações base da aplicação. O administrador também pode transferir um atendente para outro posto de atendimento.

4.3.1 Dashboard

No Dashboard o administrador pode analisar gráficos de fácil visualização e interpretação, que auxiliam a transformação de dados em informações pertinentes. Com esse acesso, o administrador tem a possibilidade de atuar em tempo real assim que identificar focos de lentidão no atendimento.

Os gráficos do tipo pizza possuem interação, onde ao selecionar a fatia da região desejada, outras informações são mais detalhadas ao nível de posto de atendimento.

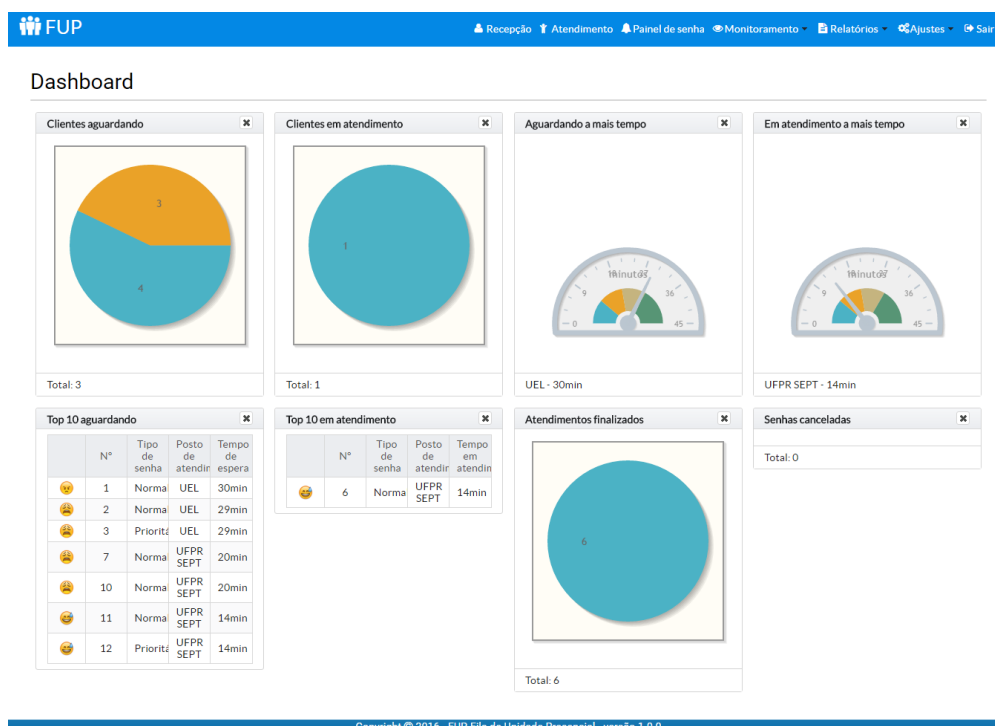


FIGURA 13 - DASHBOARD
FONTE: OS AUTORES (2016)



FIGURA 14 – TABELA DE DETALHE DA REGIAO DO DASHBOARD
 FONTE: OS AUTORES (2016)

4.3.2 Mapa

A visualização dos postos de atendimento em um mapa possibilita ao gestor visualizar a senha mais crítica de cada local, tendo ainda a possibilidade de visualizar mais detalhes clicando no ícone.

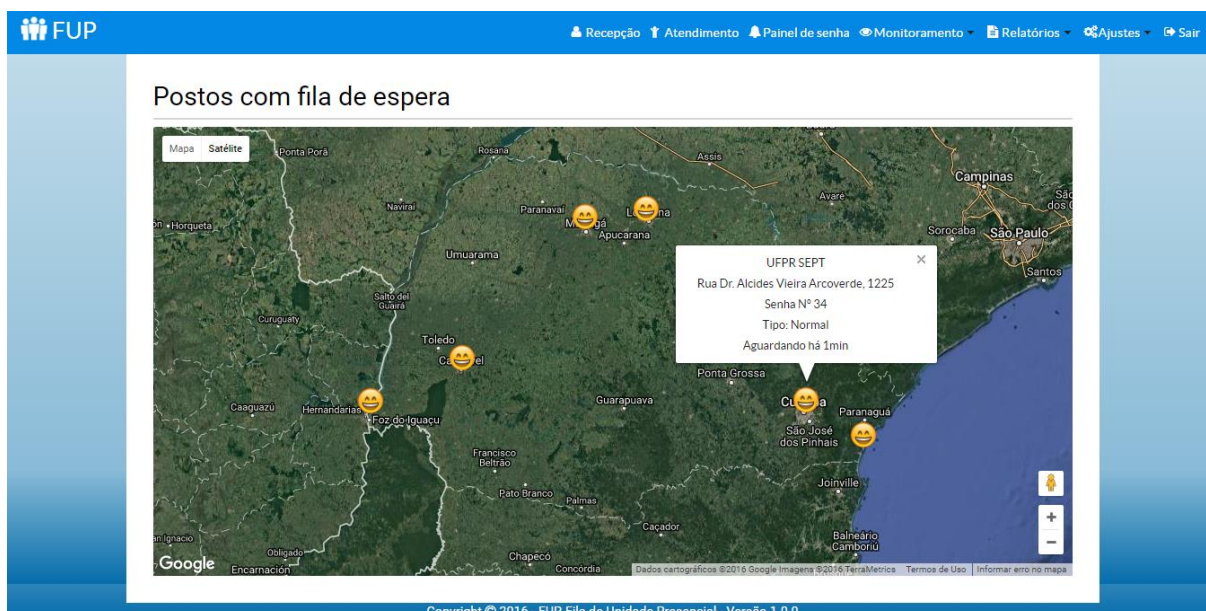


FIGURA 15 – MAPA
 FONTE: OS AUTORES (2016)

4.3.3 Relatórios

Por fim, ele pode também gerar relatórios com informações de um período desejado. Após a emissão do relatório, o usuário pode realizar com facilidade filtros e ordenações nos dados, além de exportar uma planilha em formato XLS para armazenagem ou utilização em outras análises.

FUP

Recepção Atendimento Painel de senha Monitoramento Relatórios Ajustes Sair

Relatório geral

Parâmetros

Data início: 01/09/2016 Data fim: 21/09/2016

Número	Região	Posto de at	Tipo de ser	Status da si Status	Guichê	Usuário	Data emiss	Hora emiss	Hora cham	Hora início	Hora fim	Tempo de e	Tempo desl	Tempo em :	Tempo pern
4	Leste (CWI)	UFPR SEP	Retorno	Cancelada	A1	hugo	18/09/201	00:58:35	00:58:44		00:58:45	00:00:08		00:00:01	00:00:14
3	Leste (CWI)	UFPR SEP	Prioritário	Concluída	A1	hugo	18/09/201	00:58:03	00:58:14	00:58:16	00:58:18	00:00:10	00:00:02	00:00:01	00:00:05
2	Leste (CWI)	UFPR SEP	Normal	Concluída	A1	hugo	18/09/201	00:57:59	00:58:46	00:58:47	01:01:04	00:00:47	00:00:01	00:02:16	00:03:05
1	Leste (CWI)	UFPR SEP	Normal	Concluída	A1	hugo	18/09/201	00:57:51	00:58:19	00:58:21	00:58:31	00:00:28	00:00:01	00:00:10	00:00:40
30	Leste (CWI)	UFPR SEP	Prioritário	Gerada		hugo	17/09/201	23:40:36							
29	Leste (CWI)	UFPR SEP	Normal	Gerada		hugo	17/09/201	23:40:30							
28	Leste (CWI)	UFPR SEP	Normal	Gerada		hugo	17/09/201	23:15:35							
27	Leste (CWI)	UFPR SEP	Normal	Em atendi	b12	hugo	17/09/201	23:15:33	23:39:44	23:39:45		00:24:11	00:00:00		
26	Leste (CWI)	UFPR SEP	Normal	Concluída	b12	hugo	17/09/201	22:47:45	23:17:51	23:17:52	23:39:43	00:30:05	00:00:01	00:21:50	00:51:57
25	Leste (CWI)	UFPR SEP	Normal	Concluída	b12	hugo	17/09/201	22:47:41	23:15:43	23:15:46	23:17:50	00:28:02	00:00:02	00:02:03	00:30:09
24	Leste (CWI)	UFPR SEP	Normal	Concluída	1	hugo	17/09/201	22:47:36	23:13:06	23:13:07	23:14:50	00:25:30	00:00:00	00:01:42	00:27:13
23	Leste (CWI)	UFPR SEP	Prioritário	Concluída	f	hugo	17/09/201	22:14:53	22:36:21	22:40:12	22:40:14	00:21:27	00:03:51	00:00:01	00:25:20
22	Leste (CWI)	UFPR SEP	Normal	Concluída	1	hugo	17/09/201	20:34:04	22:47:53	22:47:54	23:08:49	02:13:48	00:00:01	00:20:54	02:34:44
21	Leste (CWI)	UFPR SEP	Normal	Concluída	f	hugo	17/09/201	20:28:15	22:41:31	22:42:38	22:47:49	02:13:15	00:01:06	00:05:10	02:19:33
20	Leste (CWI)	UFPR SEP	Normal	Concluída	f	hugo	17/09/201	20:13:43	22:41:00	22:41:14	22:41:16	02:27:17	00:00:14	00:00:01	02:27:32
19	Leste (CWI)	UFPR SEP	Normal	Concluída	1	hugo	17/09/201	20:13:33	20:34:51	20:36:07	22:13:56	00:21:18	00:01:16	01:37:48	02:00:23
18	Leste (CWI)	UFPR SEP	Normal	Concluída	1	hugo	17/09/201	20:11:21	20:17:25	20:17:26	20:32:13	00:06:04	00:00:00	00:14:46	00:20:51
17	Leste (CWI)	UFPR SEP	Normal	Concluída	1	hugo	17/09/201	20:11:16	20:14:25	20:14:34	20:15:38	00:03:08	00:00:09	00:01:04	00:04:22
16	Leste (CWI)	UFPR SEP	Normal	Concluída	1	hugo	17/09/201	20:08:10	20:14:13	20:14:14	20:14:22	00:06:03	00:00:00	00:00:08	00:06:12
15	Leste (CWI)	UFPR SEP	Normal	Cancelada	1	hugo	17/09/201	20:05:37	20:14:13		20:14:13	00:08:35			

Copyright © 2016 - FUP Filial de Unidade Presencial - versão 1.0.0

FIGURA 16 – RELATÓRIO GERAL
FONTE: OS AUTORES (2016)

Existe também a opção de verificar os TMAs (Tempos Médios de Atendimento), ideais para a o uso de benchmark entre atendentes, postos de atendimento e até mesmo das regiões.

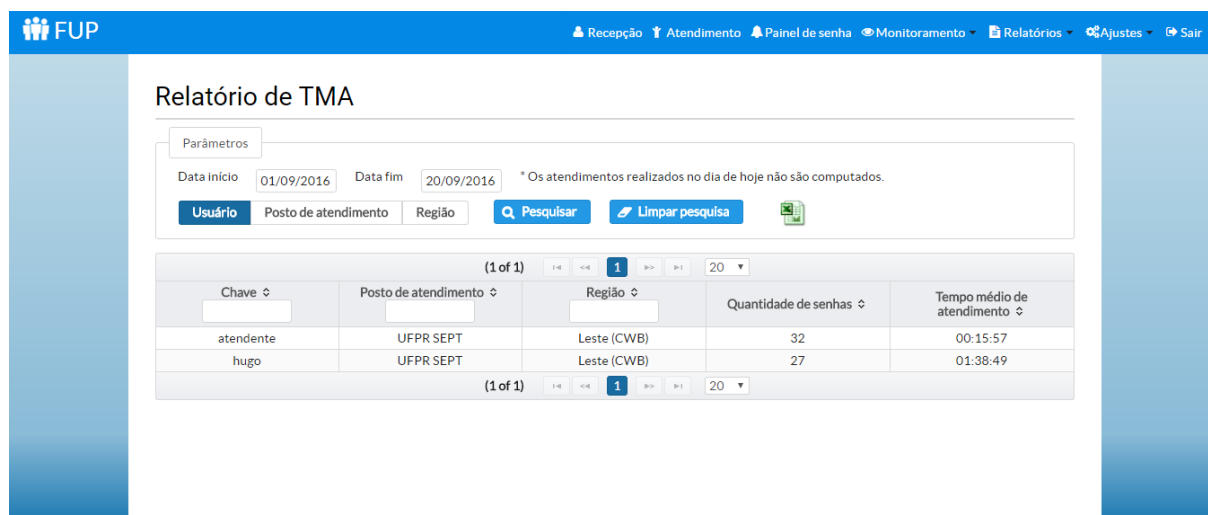


FIGURA 17 – RELATÓRIO TMA
 FONTE: OS AUTORES (2016)

Todas informações obtidas nos gráficos e relatórios auxiliam o administrador a identificar pontos onde cabem melhoria no processo, o que reflete em aumento na qualidade do atendimento e satisfação do cliente.

4.4 MÓDULO INTEGRAÇÃO

No FUP foram disponibilizados serviços web do tipo REST que possibilitam realizar integração com sistemas legados e outros dispositivos, onde é possível realizar operações de geração de senha e também toda a série de ações para as alterações dos estados das senhas.

Foi realizada a opção pela arquitetura REST com JSON em vez do protocolo SOAP com XML devido ser mais leve e eficiente.

Foi utilizado como segurança, autenticação BASIC em todas as requisições. A configuração do usuário e senha é realizada pelo administrador do FUP.

O funcionamento e detalhes dos recursos estão descritos no APÊNDICE H.

4.5 MONTAGEM DO AMBIENTE E INSTALAÇÃO DO SISTEMA

Para efetuar a montagem do ambiente e instalação do sistema Fila de Unidade Presencial, recomenda-se seguir os passos descritos abaixo:

1. Instalar Java JDK 8;
2. Configurar a variável de ambiente JAVA_HOME;
3. Instalar versão recente da IDE Eclipse;
4. Instalar servidor de aplicação JBoss EAP 7 ou Wildfly 10;
5. Criar usuário administrador do JBoss;
6. Adicionar o JBoss no Eclipse e inicializar;
7. Com o usuário criado, instalar o driver do PostgreSQL no JBoss. O método aconselhado é simplesmente realizar o deploy do jar do driver através do acesso do menu de administração do JBoss, geralmente configurada na url: <http://localhost:9990>;
8. Instalar o PostgreSQL, de preferência versão 9.5 e acima.
9. Criar no banco usuário e senha com acesso de superadmin;
10. Criar tabelas e realizar carga de dados utilizando script fornecido no CD do FUP;
11. Configurar arquivo “standalone.xml” do JBoss, criando o datasource para acesso ao banco, utilizando qualquer usuário que possua acesso para leitura, escrita, inserção e deleção nas tabelas do FUP. Segue exemplo de como deve ficar:

```
<datasource jta="true" jndi-name="java:/fupDS" pool-name="fupDS" enabled="true" use-ccm="true">
  <connection-url>jdbc:postgresql://localhost:5432/fupdb</connection-url> <!-- ajustar -->
  <driver-class>org.postgresql.Driver</driver-class>
  <driver>postgresql-9.4.1208.jar</driver> <!-- ajustar para o nome do driver instalado -->
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>30</max-pool-size>
    <prefill>true</prefill>
  </pool>
  <security>
    <user-name>** usuario **</user-name> <!-- substituir -->
    <password>** senha **</password> <!-- substituir -->
  </security>
</datasource>
```

FIGURA 18 – DATASOURCE DO FUP
 FONTE: OS AUTORES (2016)

12. Importar os quatro projetos no Eclipse;
13. Adicionar as bibliotecas fornecidas no CD;
14. Fazer deploy do FUP no JBoss;
15. Acessar a aplicação com o contexto “/fup”;
16. Utilizar usuário “admin” com a senha padrão “admin”;
17. Trocar a senha por questão de segurança.

5 CONSIDERAÇÕES FINAIS

Na próxima seção serão apresentadas as dificuldades encontradas durante todo o processo de desenvolvimento do *software*, o aprendizado obtido e também os requisitos e melhorias determinados como implementações futuras no sistema existente.

5.1 CONCLUSÃO

Durante a execução do projeto do sistema FUP, houve uma dificuldade que esteve presente em todas as etapas: o prazo para entrega. O limite era de aproximadamente um mês, que resultou em uma compactação agressiva no cronograma das atividades.

O desenvolvimento deste projeto agregou muita experiência aos membros do grupo, como novos conhecimentos, trabalho em equipe, gerenciamento de tempo e aperfeiçoamento dos conhecimentos já obtidos com o curso de Análise e Desenvolvimento de Sistemas na Universidade Federal do Paraná. Alguns desses novos conhecimentos englobam alguns *design patterns*, o método iterativo incremental e o framework JSF. Houve também aprofundamento dos conhecimentos em Java 8 e Java EE.

Algumas características do Fila de Unidade Presencial, em seu funcionamento como um todo, dão espaço para aperfeiçoamentos futuros.

1. Categorias para serem utilizadas para classificação no momento da conclusão do atendimento;
2. Possibilidade de múltiplas filas, onde os atendentes podem ser relacionados para atender uma ou mais delas. Este método de atendimento pode ser utilizado para especializar atendentes em determinadas tarefas;
3. Agendamento online de atendimento. Envio de e-mail com lembrete;
4. Versão mobile, Android e/ou iOS. Envio de push message como lembrete;

5. Incrementar painel de senha para possibilitar customizações como logotipos, cores, imagens e vídeos institucionais;
6. Interface de autoatendimento (web ou mobile) para o cliente gerar e imprimir senha sozinho, sem a necessidade de um recepcionista.
7. Testar o leiaute de impressão da senha em uma impressora térmica típica para tal função.

REFERÊNCIAS

BOOCH, G. **The Unified Modeling Language User Guide, 2/E**. Pearson Education India, 2005.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **Uml**. Elsevier Brasil, 2006.

BRASIL. **Decreto Nº 6.932, de 11 de agosto de 2009**. Disponível em <http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6932.htm>. Acesso em: 16/09/2016.

ANEEL, Agência Nacional de Energia Elétrica. **Resolução Normativa Nº 414, de 9 de setembro de 2010**. Disponível em <<http://www2.aneel.gov.br/cedoc/ren2010414.pdf>>. Acesso em 16/09/2016.

COPEL, Companhia Paranaense de Energia Elétrica. **Website Oficial**, 2016. Disponível em: <<http://www.copel.com>>. Acesso em 24/09/2016.

SUPERIOR, SECRETARIA DE EDUCAÇÃO. **Carta de Serviços ao Cidadão**. Brasília: MINISTÉRIO DA EDUCAÇÃO, 2009. 15-19 p.

BRASIL, Assembléia Legislativa do Estado do Paraná. **LEI Nº 13.400** . Paraná: MINISTÉRIO DA EDUCAÇÃO, 2001. 1 - 2 p.

AVELINO, Guilherme Amaral . **Teoria das Filas**. [S.l.: s.n.], 2015. 2 - 37 p.

VENTURA, Plínio . **Caso de Uso – Include, Extend e Generalização** . [S.l.: s.n.], 2014. 1 - 2 p.

SIQUEIRA, RAFAELA RIBEIRO . **TEORIA DAS FILAS APLICADO A UMA CASA LOTÉRICA NO MUNICÍPIO DE MARABÁ**. Fortaleza: Enegep, 2015. 1 - 15 p.

NOVO SGA. **Website Oficial**, 2016. Disponível em: <<http://www.novosga.org>>. Acesso em 17/09/2016.

SPECTO. **Website Oficial**, 2016. Disponível em: <<http://www.specto.com.br>>. Acesso em 18/09/2016.

GAUTEK. **Website Oficial**, 2016. Disponível em: <<http://www.gautek.com.br>>. Acesso em 18/09/2016.

GUARDA FILA. **Website Oficial**, 2016. Disponível em: <<http://www.guardafila.com.br>>. Acesso em 18/09/2016.

APÊNDICES

APÊNDICE A – ANÁLISE DOS REQUISITOS

Requisitos Funcionais

ID:	RF1	Tipo de requisito:	Funcional
Dependências:	N/A	Prioridade:	Alta
Nome:	Gerenciar região		
Descrição	Permite ao administrador realizar o CRUD da entidade Região.		

ID:	RF02	Tipo de requisito:	Funcional
Dependências:	RF01	Prioridade:	Alta
Nome:	Gerenciar posto		
Descrição	Permite ao administrador realizar o CRUD da entidade Posto. É obrigatório vincular o posto com uma região.		

ID:	RF03	Tipo de requisito:	Funcional
Dependências:	RF02	Prioridade:	Alta
Nome:	Gerenciar usuários		
Descrição	Permite ao administrador realizar o CRUD dos usuários. Será obrigatório atribuir um perfil e deverá ser possível atribuir um posto de atendimento ao usuário. Perfis possíveis: Administrador, atendente e recepcionista.		

ID:	RF04	Tipo de requisito:	Funcional
Dependências:	N/A	Prioridade:	Alta
Nome:	Gerenciar tipos de senha		
Descrição	Permite ao administrador cadastrar tipos diferenciados de senhas. Necessário para atender legislação de prioridade para determinadas situações no atendimento.		

ID:	RF05	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03 e RF04	Prioridade:	Alta
Nome:	Tela de triagem de cliente e geração de senha		
Descrição	Tela para o recepcionista. Demais perfis também poderão acessar. Tem a função de gerar senha em um determinado posto (ao qual o usuário está vinculado).		

ID:	RF06	Tipo de requisito:	Funcional
Dependências:	RF05	Prioridade:	Alta
Nome:	Tela de atendimento de clientes		
Descrição	Tela para atendente acompanhar fila, chamar, cancelar e/ou concluir senha. O recepcionista não deve ter acesso e o administrador pode ter acesso. Se possível o atendente deve poder visualizar quais senhas estão em atendimento no momento no posto que ele está vinculado e também visualizar quantos atendimentos foram realizados por cada atendente no dia.		

ID:	RF07	Tipo de requisito:	Funcional
Dependências:	RF06	Prioridade:	Média
Nome:	Web Service para integração		
Descrição	Deve possuir Web Services que possibilitem a outro sistema legado de atendimento integrar com o FUP. Deve ser possível realizar as operações com senha, realizado as suas alterações de status.		

ID:	RF08	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03 e RF04	Prioridade:	Alta
Nome:	Gráficos em tempo real com dados de senhas por região		
Descrição	Gráficos e/ou tabelas com dados das senhas segregadas por região que possibilitem a rápida visualização e análise da situação do momento em um escopo mais amplo. Os gráficos devem ser atualizados em tempo real.		

ID:	RF09	Tipo de requisito:	Funcional
Dependências:	RF08	Prioridade:	Alta
Nome:	Gráficos em tempo real com dados de senhas por posto		
Descrição	Deve possuir a mesma função da RF08, porém o escopo são as senhas segregadas por posto.		

ID:	RF10	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03 e RF04	Prioridade:	Baixa
Nome:	Gráficos de quantidade de senhas concluídas e canceladas		
Descrição	Deve possibilitar a análise do fluxo de atendimento já realizado nos postos e regiões.		

ID:	RF11	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03 e RF04	Prioridade:	
Nome:	Gráfico ou tabela com as senhas de clientes aguardando e sendo atendidos a mais tempo		
Descrição	Deve exibir as 10 senhas que representam os piores cenários possíveis das senhas que estão aguardando atendimento (demora para ser atendido) e que estão sendo atendidos no momento (lentidão durante o atendimento).		

ID:	RF12	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03, RF04 e RNF05	Prioridade:	Alta
Nome:	Relatório de senhas por período		
Descrição	Deve gerar relatórios tendo como parâmetros de entrada um intervalo de tempo. Deve ser possível filtrar os resultados e exportar o relatório em forma de arquivo, de preferência em formato de planilha. Este relatório deve ser granulado para possibilitar a confecção de outros relatórios a partir destes dados.		

ID:	RF13	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03, RF04 e RNF05	Prioridade:	Média
Nome:	Relatório de TMA por período e atendente		
Descrição	Deve gerar relatório de TMA (Tempo Médio de Atendimento) em um dado intervalo de tempo para os atendimentos realizados. Os dados devem ser segregados por atendente. Serve para comparar o desempenho entre os atendentes.		

ID:	RF14	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03, RF04 e RNF05	Prioridade:	Média
Nome:	Relatório de TMA por período e posto		
Descrição	Idem ao RF13, porém neste o TMA deve ser segregado por posto de atendimento. Neste caso servirá para avaliar o desempenho entre os postos de atendimento.		

ID:	RF15	Tipo de requisito:	Funcional
Dependências:	RF01, RF02, RF03, RF04 e RNF05	Prioridade:	Média
Nome:	Relatório de TMA por período e região		
Descrição	Idem ao RF13 e RF14, porém neste o TMA deve ser segregado por regiões. Neste caso servirá para avaliar o desempenho entre as regiões.		

ID:	RF16	Tipo de requisito:	Funcional
Dependências:	RF06	Prioridade:	Média
Nome:	Painel de senha		
Descrição	Tela para os clientes do posto de atendimento poderem visualizar qual senha foi chamada. Deve exibir qual atendente chamou a senha (número/nome do guichê). Se possível, exibir histórico das últimas senhas chamadas para auxiliar clientes desatentos.		

Requisitos não funcionais

ID:	RNF01	Tipo de requisito:	Não funcional
Dependências:	N/A	Prioridade:	Alta
Nome:	Linguagem de desenvolvimento		
Descrição	Deverá ser utilizado Java como a linguagem de desenvolvimento do sistema. É melhor escolha devido suas qualidades e os benefícios que proporciona, além de ser linguagem onde os desenvolvedores são mais fluentes.		

ID:	RNF02	Tipo de requisito:	Não funcional
Dependências:	RNF01	Prioridade:	Alta
Nome:	Arquitetura do sistema / Design pattern		
Descrição	Deverá ser utilizado Java EE, EJB (Injeção de dependência), JSF, JPA (Hibernate) e MVC.		

ID:	RNF03	Tipo de requisito:	Não funcional
Dependências:	RF07	Prioridade:	Média
Nome:	Web Service com autenticação		
Descrição	O Web Service do RF07 deve ser implementado com a exigência de usuário e senha para possuir segurança de acesso ao FUP.		

ID:	RNF04	Tipo de requisito:	Não funcional
Dependências:	RF03	Prioridade:	Alta
Nome:	Criptografia das senhas dos usuários		
Descrição	As senhas dos usuários do FUP devem ser criptografadas antes de serem persistidas no banco. Não utilizar hash ou MD5, utilizar método um pouco mais seguro.		

ID:	RNF05	Tipo de requisito:	Não funcional
Dependências:	N/A	Prioridade:	Alta
Nome:	Separar as senhas em duas tabelas		
Descrição	O banco de dados deve possuir duas tabelas distintas: uma para as senhas do dia e outra que deve ser utilizada como repositório de senhas dos dias anteriores. Devido ao perfil de muitas consultas no banco do sistema, trabalhar com uma tabela reduzida aumentará o desempenho dos acessos.		

ID:	RNF06	Tipo de requisito:	Não funcional
Dependências:	RNF05	Prioridade:	Alta
Nome:	Rotina para mover senhas para a tabela de histórico		
Descrição	Rotina programada para toda madrugada realizar a movimentação das senhas da tabela do dia para a tabela de histórico. Por garantia (caso a aplicação não esteja rodando no momento da execução da rotina), a aplicação deverá rodar a rotina sempre que ela for inicializada.		

ID:	RNF07	Tipo de requisito:	Não funcional
Dependências:	N/A	Prioridade:	Média
Nome:	Utilização de dados em cache		
Descrição	Utilizar dados em cache quando houver muita consulta repetida à dados que possuam pouca mutabilidade. Isso ajudará a aliviar a carga de consultas ao banco de dados.		

APÊNDICE B – DIAGRAMA DE CASO DE USO

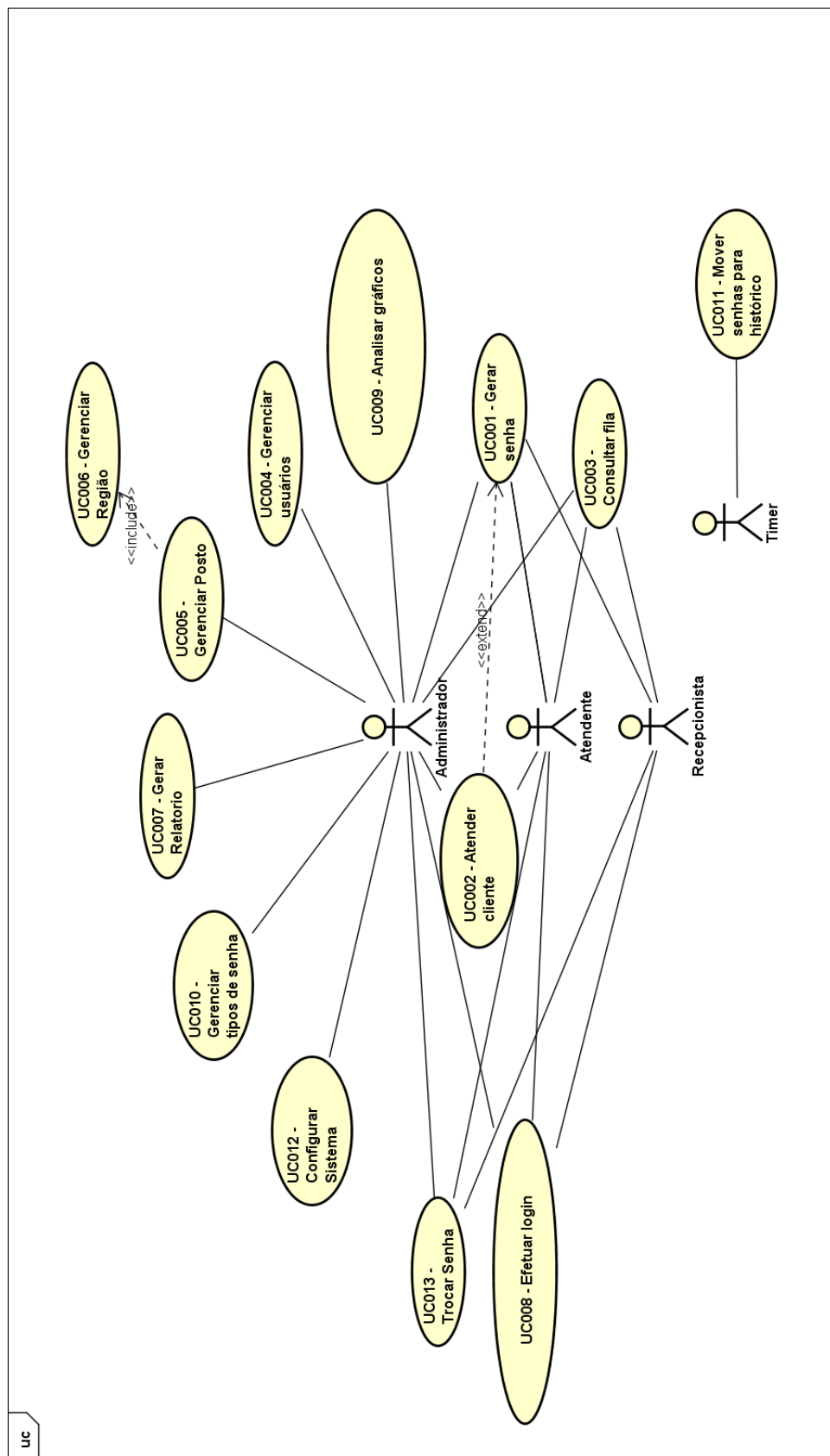


FIGURA 19 – DIAGRAMA DE CASOS DE USO
 FONTE: OS AUTORES (2016)

APÊNDICE C – ESPECIFICAÇÕES DE CASOS DE USO

UC001 – Gerar Senha

Descrição:

Caso de uso para gerar senha do cliente no sistema

Data View:

DV1. Geração de senha

Recepção

Londrina

Tipo de senha: Prioritário Retorno Normal Gerar senha Imprimir última senha

Clientes aguardando					
	Número	Tipo de senha	Hora chegada	Tempo de espera	
😞	1	Normal	15:07:35	26min	
😞	2	Normal	15:07:39	26min	
Total: 2					
Legenda:	😊 < 10min	😄 < 20min	😓 < 30min	😞 < 45min	😡 >= 45min

FIGURA 20 – TELA GERAR SENHA
FONTE: OS AUTORES (2016)

Pré-condições:

Nenhuma

Pós-condições:

Após o fim normal deste caso de uso o sistema deve:

1. Cadastrar a nova senha no banco de dados
2. Acionar o caso de uso UC – Consultar Fila

Ator primário:

Recepcionista e atendente

Fluxo principal:

1. O sistema exibe a tela DV1
2. O usuário seleciona o tipo de senha que será gerado
3. O usuário clica no botão 'Gerar senha' (E1)(A1)(R3)
4. A senha é mostrada na tabela 'clientes aguardando' (R1)(R2)
5. O sistema persiste os dados e finaliza o caso de uso

Fluxos alternativos

A1. O usuário clica no botão “imprimir última senha”

1. O Sistema imprime a última senha

Fluxos de exceção

E1. O usuário deixa de escolher um dos campos:

1. O sistema exibe a mensagem “Obrigatório tipo de senha”.

Regra de Negocio

R1. A geração de senhas começa zerada no começo do dia de trabalho

R2. O painel 'clientes aguardando' é constantemente atualizado

R3. Tanto recepcionista como atendente podem gerar senhas, caso necessário o administrador pode gerar também.

UC002 – Atender Cliente

Descrição:

Caso de uso para realizar a chamada de senha do cliente

Data View:

DV1 Chamar senhas

Atendimento

SEPT UFPR

Número: Chamar senha

Tipo de senha:
Status da senha:
Tempo de espera:

Clientes aguardando				
	Número	Tipo de senha	Hora chegada	Tempo de espera
🔴	6	Normal	00:58:38	14h58min
🔴	7	Normal	01:00:59	14h55min
Total: 2				
Legenda: 🟢 < 10min 🟡 < 20min 🟠 < 30min 🟡 < 45min 🔴 >= 45min				

Clientes em atendimento							
	Número	Tipo de senha	Guichê	Hora chamada	Hora início atendimento	Tempo de espera	Tempo em atendimento
🔴	5	Normal	A1	01:01:09	01:01:11	2min	14h55min
Total: 1							

Atendimentos finalizados		
Usuário	Concluídos	Cancelados
hugo	3	1

FIGURA 21 – TELA CHAMAR SENHA
FONTE: OS AUTORES (2016)

DV2. Botões 'iniciar atendimento' e 'cancelar senha' habilitados

Atendimento

SEPT UFPR

Número: ▶ Iniciar atendimento
✖ Cancelar senha

Tipo de senha: 6 Normal
Status da senha: Chamada
Tempo de espera: 15h11min

Clientes aguardando				
	Número	Tipo de senha	Hora chegada	Tempo de espera
🔴	7	Normal	01:00:59	15h10min
Total: 1				
Legenda: 🟢 < 10min 🟡 < 20min 🟠 < 30min 🟡 < 45min 🔴 >= 45min				

Clientes em atendimento							
	Número	Tipo de senha	Guichê	Hora chamada	Hora início atendimento	Tempo de espera	Tempo em atendimento
🟢	6	Normal	C02	16:10:21		15h11min	N/A
🔴	5	Normal	A1	01:01:09	01:01:11	2min	15h9min
Total: 2							

Atendimentos finalizados		
Usuário	Concluídos	Cancelados
hugo	3	1

FIGURA 22 – TELA INICIAR E CANCELAR ATENDIMENTO
FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login

2. O usuário deve ser atendente ou administrador

Pós-condições:

1. Persistir os dados da senha chamada no banco de dados.

Ator primário:

Atendente

Fluxo principal

1. O sistema exibe a tela DV1(R2)(R3)(E2)
2. O usuário preenche a identificação do seu guichê(E1)
3. O usuário clica no botão 'chamar senha'(R1)
4. O sistema atualiza a tela DV1
5. O sistema habilita o botão "Cancelar senha"
6. O sistema habilita o botão "Iniciar atendimento"
7. O sistema apresenta o número da senha chamada
8. O sistema apresenta o tipo de senha chamado
9. O sistema apresenta o status da senha
10. O sistema apresenta o tempo de espera
11. O sistema atualiza o painel 'Clientes Aguardando', tirando a senha chamada da fila
12. O usuário clica no botão 'iniciar atendimento'(A1)
13. O sistema atualiza o 'status da senha' para 'em atendimento'
14. O sistema atualiza o painel 'Clientes em atendimento'
15. O usuário clica no botão 'concluir atendimento'
16. O sistema atualiza o painel 'Clientes em atendimento', limpando o registro do atendimento
17. O sistema atualiza atendimentos finalizados, contando mais um atendimento para o respectivo atendente
18. O Caso de uso é encerrado

Fluxos alternativos:

- A1. O usuário clica no botão 'cancelar atendimento'

1. O sistema limpa os campos 'número', 'Tipo de senha', 'Status da senha' e 'tempo de espera'
2. O sistema retira o respectivo registro da senha do painel 'Clientes em atendimento'
3. O Sistema contabiliza uma senha cancelada para o respectivo atendente no painel 'atendimentos finalizados'

Fluxos de exceção:

E1. O usuário tenta chamar senha sem preencher o campo 'Guichê'

1. O sistema exibe a mensagem 'Obrigatório Guichê '

E2. O usuário não atende ao R2 e R3

1. O sistema exibe mensagem de erro

Regras de negócio

R1. Necessário ter senhas aguardando para chamar

R2. O usuário deve ser atendente ou administrador

R3. O usuário deve estar cadastrado em um posto

UC003 - Consultar fila

Descrição:

Caso de uso para visualizar o gerenciamento das senhas.

Data View:

DV1. Consultar fila perfil recepcionista

Recepção

Londrina

Tipo de senha:

Prioritário
Retorno
Normal

🔑 Gerar senha
🖨️ Imprimir última senha

Clientes aguardando				
	Número	Tipo de senha	Hora chegada	Tempo de espera
😞	1	Normal	15:07:35	26min
😞	2	Normal	15:07:39	26min
Total: 2				
Legenda:				
😊	< 10min	😄	< 20min	😓
😓	< 30min	😞	< 45min	🔴
				>= 45min

FIGURA 23 – TELA CONSULTA FILA
 FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login

Pós-condições:

Nenhuma

Ator primário:

Qualquer usuário

Fluxo principal:

1. O sistema apresenta a tela DV1 do recepcionista (R1)(E1)
2. O usuário visualiza o numero da senha
3. O usuário visualiza o tipo de senha
4. O usuário visualiza a hora que foi gerada a senha para o cliente
5. O usuário visualiza quanto tempo o cliente esta esperando para ser atendido

Fluxos de Exceção:

E1. O usuário não esta cadastrado em um posto

1. O sistema emite mensagem de erro

Regras de negócio

R1. O usuário deve estar cadastrado em um posto de atendimento

UC004 – Gerenciar Usuários

Descrição:

Caso de uso para realizar o cadastro de usuários do sistema

Data View:

DV1 Usuários cadastrados

Usuário

Usuários cadastrados						+ Adicionar
Nome ↕	Sobrenome ↕	Chave ↕	E-mail ↕	Posto de atendimento ↕	Tipo de usuário ↕	Ações
Administrador do sistema		admin	admin@fup.com.br		administrador	Editar Excluir
Hugo	Fujiwara	hugo	hugo.fujiwara@ema	SEPT UFPR	administrador	Editar Excluir
Fulano	de Tal	atendente	fulano@email.com	SEPT UFPR	atendente	Editar Excluir
Ciclano	de Tal	recepcionista	recep@email.com	Londrina	recepcionista	Editar Excluir

(1 of 1) |< << 1 >> >| 10 ▾

Total: 4

FIGURA 24 – TELA USUÁRIOS CADASTRADOS
 FONTE: OS AUTORES (2016)

DV2 Adicionar usuário

Usuário

Nome*	<input type="text"/>
Sobrenome	<input type="text"/>
Chave*	<input type="text"/>
Posto de atendimento	Selecione ▼
Tipo de usuário*	Selecione ▼
E-mail*	<input type="text"/>
Senha*	<input type="password"/>
Confirme a senha*	<input type="password"/>
Lembrete	<input type="text"/>

FIGURA 25 – TELA ADICIONAR USUÁRIOS
FONTE: OS AUTORES (2016)

DV3 Editar usuário

Usuário

Nome*	Administrador do sistem
Sobrenome	<input type="text"/>
Chave*	admin
Posto de atendimento	Selecione ▼
Tipo de usuário*	administrador ▼
E-mail*	admin@fup.com.br

FIGURA 26 – TELA EDITAR USUÁRIO
FONTE: OS AUTORES (2016)

Pré-condições:

1. Estar logado com perfil de administrador

Pós-condições:

1. Persistir os dados do usuário no banco de dados

Ator primário:

Administrador

Fluxo principal:

1. O sistema exibe a tela DV1
2. O usuário clica no botão 'Adicionar'(A1)(A2)
3. O sistema exibe a tela DV2
4. O usuário preenche o campo "Nome"
5. O usuário preenche o campo 'Sobrenome'
6. O usuário preenche o campo 'Chave'
7. O usuário escolhe a opção do campo 'Posto de atendimento'
8. O usuário escolhe a opção do campo 'tipo de usuário'
9. O usuário preenche o campo "E-mail"
10. O usuário preenche o campo "Senha"
11. O usuário preenche o campo "Confirme a senha"
12. O usuário preenche o campo "Lembrete"
13. O usuário clica no botão 'salvar' (E1)(E2)(E3)(A3)
14. O sistema persiste os dados do usuário e finaliza o caso de uso

Fluxos alternativos:**A1.** O usuário clica no botão 'editar'

1. O sistema apresenta a tela (DV3)
2. O usuário altera o campo que achar conveniente
3. O usuário clica em salvar (A3)(E1)

A2. O usuário clica no botão 'Excluir'

1. O sistema exibe a mensagem "Deseja realmente excluir o registro?"
2. O usuário confirma a exclusão
3. O sistema remove o registro selecionado
4. O sistema atualiza a lista de usuários cadastrados

A3. O usuário clica no botão 'Sair'

1. O sistema finaliza a seção

Fluxos de exceção:

E1. O usuário não preenche algum dos campos obrigatórios

1. O sistema exibe uma mensagem indicando que todos os campos obrigatórios devem ser preenchidos

E2. O usuário informa senhas diferentes.

1. O sistema exibe uma mensagem indicando que as senhas não conferem

E3. O administrador tenta incluir um usuário já cadastrado

1. O sistema exibe a mensagem 'Erro'

UC005 – Gerenciar Posto

Descrição:

Caso de uso para gerenciar o posto de atendimento

Data View:

DV1. Cadastro de postos de atendimento

Posto de atendimento

Postos cadastrados						+ Adicionar
Nome ↕	Descrição ↕	Região ↕	Endereço ↕	Latitude ↕	Longitude ↕	Ações
UEL	Universidade Estadual de Londrina	Norte	Rodovia Celso Garcia Cid, Km 380	-23.323348	-51.196874	✎ Editar ✕ Excluir
UFPR SEPT	Setor de Educação Profissional e Tecnológica	Leste (CWB)	Rua Dr. Alcides Vieira Arcoverde, 1225	-25.45707	-49.235869	✎ Editar ✕ Excluir

(1 of 1) [◀](#) [◀◀](#) **1** [▶▶](#) [▶](#) 10 ▾

Total: 2

FIGURA 27 – TELA POSTOS CADASTRADOS
FONTE: OS AUTORES (2016)

DV2. Adicionar Posto

Posto de atendimento

Nome*

Descrição*

Região*

Endereço

Latitude

Longitude

[💾 Salvar](#) [✕ Sair](#)

FIGURA 28 – TELA ADICIONAR POSTO
FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login
2. O usuário deve ser Administrador

Pós-condições:

Nenhuma

Ator primário:

Administrador

Fluxo principal:

1. O sistema exibe a tela DV1
2. O usuário clica no botão “Adicionar”(A1)(A2)
3. O sistema exibe a tela (DV2)
4. O usuário preenche o campo “Nome”
5. O usuário preenche o campo “descrição”
6. O usuário escolhe dentre as opções a “Região”
7. O usuário preenche o campo “Latitude”
8. O usuário preenche o campo “Longitude”
9. O usuário clica no botão “Salvar ”(A3)(E1)
10. O sistema persiste os dados.

Fluxos alternativos:

A1. O usuário clica no botão 'Editar'

1. O sistema exibe a tela DV2
2. O usuário edita o campo que achar conveniente
3. O usuário clica no botão ‘Salvar’ (A3)(E1)
4. O sistema persiste os dados

A2. O usuário clica no botão ‘Excluir’

1. O sistema exibe a mensagem “Deseja realmente excluir o registro?”
2. O usuário confirma a exclusão
3. O sistema remove o registro do posto selecionado
4. O sistema atualiza a lista de postos cadastrados

A3. O usuário clica no botão ‘Sair’

1. O sistema exibe a tela DV1.

Fluxos de exceção:

E1. O usuário clica no botão “Salvar” com campo obrigatório vazio

1. O sistema emite mensagem “Obrigatório nome, descrição e região”

UC006 – Gerenciar Região

Descrição:

Caso de uso para gerenciar a região do posto de atendimento

Data View:

DV1. Cadastrar região

Região

Regiões cadastradas		+ Adicionar
Nome ↕	Descrição ↕	Ações
<input type="text"/>	<input type="text"/>	
Região Metropolitana	Região Metropolitana de Curitiba	✎ Editar ✖ Excluir
Norte	Norte do PR	✎ Editar ✖ Excluir
Oeste	Oeste do PR	✎ Editar ✖ Excluir
Central	PG e região	✎ Editar ✖ Excluir
aaa	bbbb	✎ Editar ✖ Excluir
sul	birrrlo	✎ Editar ✖ Excluir
Leste (CWB)	Unidades de Curitiba	✎ Editar ✖ Excluir

(1 of 1) [◀](#) [◀◀](#) **1** [▶▶](#) [▶](#) 10 ▾

Total: 7

FIGURA 29 – TELA REGIÕES CADASTRADAS
FONTE: OS AUTORES (2016)

DV2. Adicionar região

Região

Nome*

Descrição*



 **Salvar**  **Sair**

FIGURA 30 – TELA ADICIONAR REGIÃO
FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login
2. O usuário deve ser Administrador

Pós-condições:

Nenhuma

Ator primário:

Administrador

Fluxo principal:

1. O sistema exibe a tela DV1
2. O usuário clica no botão “Adicionar”(A1)(A2)
3. O sistema exibe a tela (DV2)
4. O usuário preenche o campo “Nome”
5. O usuário preenche o campo “Região”
6. O usuário clica no botão “Salvar ”(A3)(E1)
7. O sistema persiste os dados.

Fluxos alternativos:

A1. O usuário clica no botão 'Editar'

1. O sistema exibe a tela DV2
2. O usuário edita o campo que achar conveniente
3. O usuário clica no botão ‘Salvar’ (A3)(E1)
4. O sistema persiste os dados

A2. O usuário clica no botão 'Excluir'

1. O sistema exibe a mensagem "Deseja realmente excluir o registro?"
2. O usuário confirma a exclusão
3. O sistema remove o registro da região
4. O sistema atualiza a lista de postos cadastrados

A3. O usuário clica no botão 'Sair'

1. O sistema exibe a tela DV1.

Fluxos de exceção:**E1.** O usuário clica no botão "Salvar" com campo obrigatório vazio

1. O sistema emite mensagem "Obrigatório nome e descrição"

UC007 – Gerar relatório

Descrição:


Caso de uso para gerar os relatórios do posto de atendimento.

Data View:

DV1. Relatórios

Relatório geral

Parâmetros

Data início 16/09/2016 Data fim 18/09/2016 

(1 of 3) 1 2 3 20 Colunas

Número	Região	Posto de	Tipo de se	Status da	Guichê	Usuário	Data emis	Hora emis	Hora cha	Hora inici	Hora fim	Tempo de	Tempo de	Tempo em	Tempo pe
2	Norte	UEL	Normal	Gerada		repcionis	18/09/2016	15:07:39				01:58:45			
1	Norte	UEL	Normal	Gerada		repcionis	18/09/2016	15:07:35				01:58:49			
7	Leste (CWE	UFPR SEPT	Normal	Gerada		hugo	18/09/2016	01:00:59				16:05:25			
6	Leste (CWE	UFPR SEPT	Normal	Cancelada	C02	atendente	18/09/2016	00:58:38	16:10:21		16:14:03	15:11:43			
5	Leste (CWE	UFPR SEPT	Normal	Em atendim	A1	hugo	18/09/2016	00:58:36	01:01:09	01:01:11		00:02:32	00:00:02	16:05:12	
4	Leste (CWE	UFPR SEPT	Retorno	Cancelada	A1	hugo	18/09/2016	00:58:35	00:58:44		00:58:45	00:00:08			
3	Leste (CWE	UFPR SEPT	Prioritário	Concluída	A1	hugo	18/09/2016	00:58:03	00:58:14	00:58:16	00:58:18	00:00:10	00:00:02	00:00:01	00:00:14
2	Leste (CWE	UFPR SEPT	Normal	Concluída	A1	hugo	18/09/2016	00:57:59	00:58:46	00:58:47	01:01:04	00:00:47	00:00:01	00:02:16	00:03:05
1	Leste (CWE	UFPR SEPT	Normal	Concluída	A1	hugo	18/09/2016	00:57:51	00:58:19	00:58:21	00:58:31	00:00:28	00:00:01	00:00:10	00:00:40
30	Leste (CWE	UFPR SEPT	Prioritário	Gerada		hugo	17/09/2016	23:40:36							
29	Leste (CWE	UFPR SEPT	Normal	Gerada		hugo	17/09/2016	23:40:30							
28	Leste (CWE	UFPR SEPT	Normal	Gerada		hugo	17/09/2016	23:15:35							
27	Leste (CWE	UFPR SEPT	Normal	Em atendim	b12	hugo	17/09/2016	23:15:33	23:39:44	23:39:45		00:24:11	00:00:00		
26	Leste (CWE	UFPR SEPT	Normal	Concluída	b12	hugo	17/09/2016	22:47:45	23:17:51	23:17:52	23:39:43	00:30:05	00:00:01	00:21:50	00:51:57
25	Leste (CWE	UFPR SEPT	Normal	Concluída	b12	hugo	17/09/2016	22:47:41	23:15:43	23:15:46	23:17:50	00:28:02	00:00:02	00:02:03	00:30:09
24	Leste (CWE	UFPR SEPT	Normal	Concluída	1	hugo	17/09/2016	22:47:36	23:13:06	23:13:07	23:14:50	00:25:30	00:00:00	00:01:42	00:27:13

FIGURA 31 – TELA RELATÓRIO GERAL
 FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login
2. O usuário deve ser Administrador

Pós-condições:

Nenhuma

Ator primário:

Administrador

Fluxo principal:

1. O sistema exibe a tela DV1
2. O usuário preenche o campo “data de inicio”
3. O usuário preenche o campo “data de fim”
4. O usuário clica no botão “pesquisar” (A1)(A2)(E1)
5. O sistema busca todas as senhas geradas no intervalo de tempo
6. O usuário clica no ícone do Excel

7. O sistema Exporta os dados pesquisados para o Excel
8. O caso de uso é encerrado

Fluxos alternativos:

A1. O usuário clica no botão 'Limpar pesquisa'

1. O sistema limpa os dados pesquisados

A2. O usuário clica no botão 'Colunas'

1. O usuário escolhe quais colunas de informação devem ser buscadas

Fluxos de exceção:

E1. O usuário clica no botão "Pesquisar" com campo de datas obrigatório vazio

1. O sistema emite mensagem e não permite a busca

E2. O usuário clica no botão "pesquisar" e não é encontrado nenhum registro

1. O sistema emite mensagem "nenhuma senha encontrada no período"

UC008 – Efetuar login

Descrição:

Este caso de uso efetua o login no sistema.

Data View:

DV1. Tela de login

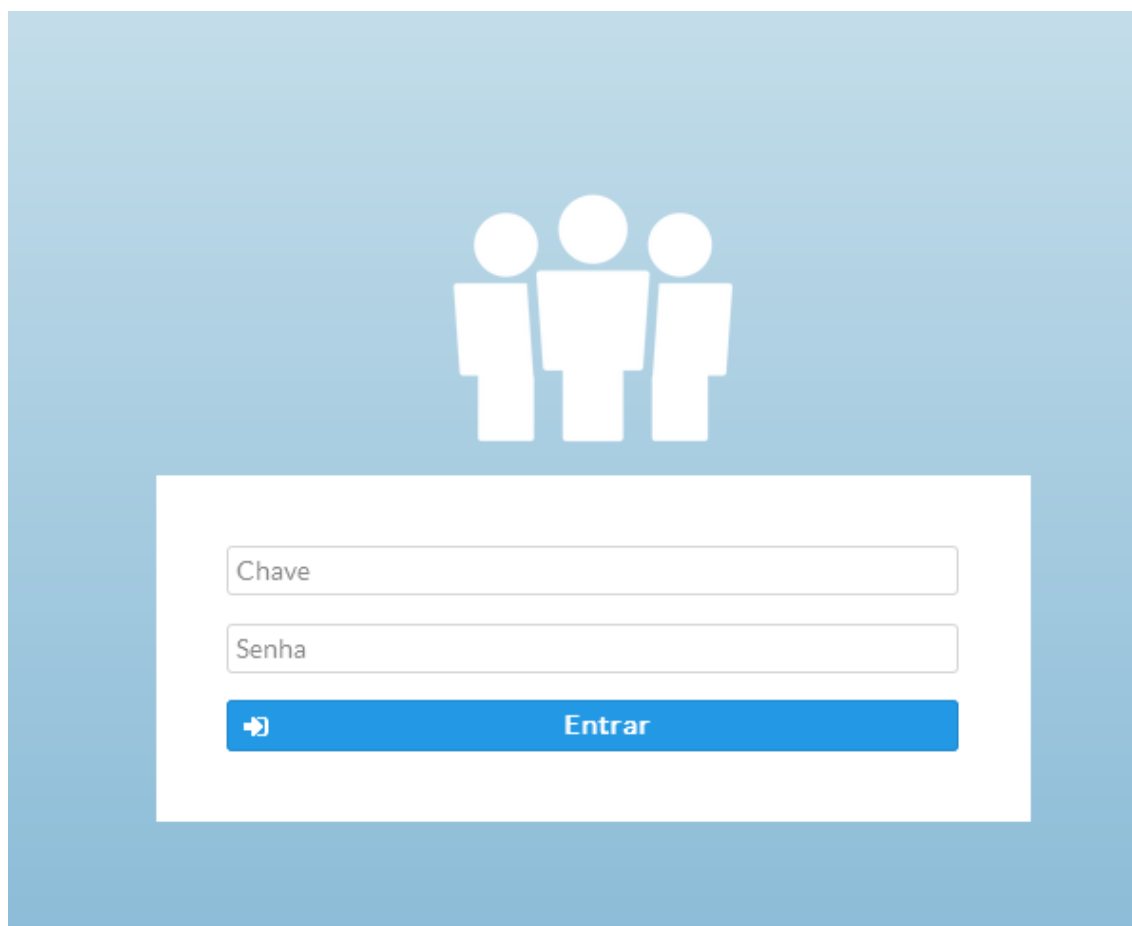


FIGURA 32 – TELA LOGIN
FONTE: OS AUTORES (2016)

Ator Principal:

Usuário

Pré-condições:

Não há pré-condições para o início desse caso de uso.

Pós-condições:

1. Após o fim normal deste caso de uso o sistema deve;
Exibir as funcionalidades conforme o perfil.

Fluxo de Eventos Principal:

1. O sistema apresenta a tela (DV1)
2. O usuário preenche o campo “Chave”
3. O usuário preenche o campo “Senha”
4. O usuário pressiona o botão “Entrar”

5. O sistema consiste os dados informados (E1)
6. O caso de uso é encerrado

Fluxos de Exceção:**E1. Login ou senha inválidos**

1. O sistema verifica inconsistência nos dados informados (R1) (R2)
2. O sistema mostra o aviso “Usuário inválido”.

Regras de Negócio:

R1. O login informado deve estar previamente cadastrado no banco de dados.

R2. A senha informada deve ser idêntica à senha cadastrada no banco de dados para o login informado.

UC009- Analisar Gráficos

Descrição:

Caso de uso para visualizar os gráficos dos clientes nos posto de atendimento

Data View:

DV1. Dashboard

Dashboard

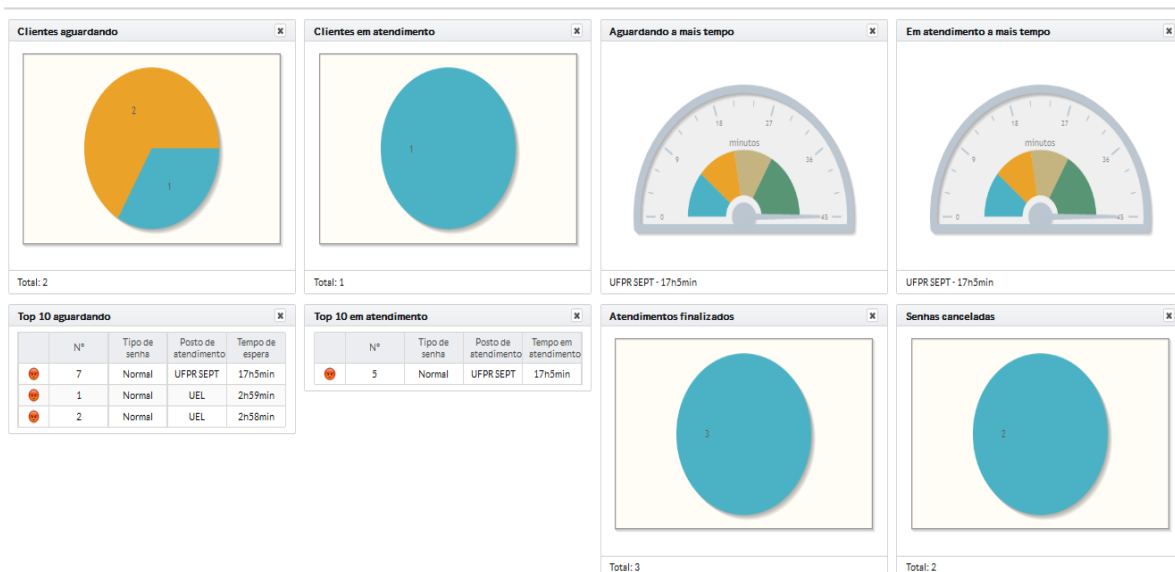


FIGURA 33 – TELA DASHBOARD
 FONTE: OS AUTORES (2016)

DV2. Tabela do respectivo gráfico

[← Voltar](#)

Norte - Clientes aguardando

Posto de atendimento ⇅	Quantidade de senhas					
	Total ⇅	< 10min	< 20min	< 30min	< 45min	>= 45min
UEL	2					🔴🔴

FIGURA 34 – TELA COM TABELA DE DETALHE DA REGIÃO NO DASHBOARD
 FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login
2. O usuário deve ser Administrador

Pós-condições:

Nenhuma

Ator primário

Administrador

Fluxo principal

1. O sistema apresenta a tela DV1
2. O sistema apresenta gráfico estatístico de clientes aguardando (A1)
3. O sistema apresenta gráfico estatístico de clientes em atendimento (A1)
4. O sistema apresenta gráfico estatístico de aguardando a mais tempo
5. O sistema apresenta gráfico estatístico de em atendimento a mais tempo
6. O sistema apresenta gráfico estatístico de top 10 aguardando
7. O sistema apresenta gráfico estatístico de top 10 em atendimento
8. O sistema apresenta gráfico estatístico de atendimentos finalizados (A1)
9. O sistema apresenta gráfico estatístico de senhas canceladas (A1)
10. O caso de uso é encerrado

Fluxos alternativos:

A1. O usuário clica numa das regiões do gráfico

1. O sistema apresenta a tela DV2 com a quantidade de senhas por tempo de espera na região escolhida(A2)

A2. O usuário clica no botão 'Voltar'

1. O sistema retorna para a tela DV1

UC010 – Gerenciar tipos de senha

Descrição:

Caso de uso gerencia as prioridades das senhas

Data View:

DV1. Tipo senha

Tipo de senha

Tipos de senha cadastrados		+ Adicionar
Nome ↕	Prioridade ↕	Ações
Prioritário	1	✎ Editar ✖ Excluir
Retorno	2	✎ Editar ✖ Excluir
Normal	4	✎ Editar ✖ Excluir

(1 of 1) ⏪ << 1 >> ⏩ 10

Total: 3

FIGURA 35 – TELA TIPO DE SENHAS CADASTRADAS
 FONTE: OS AUTORES (2016)

DV2. Adicionar tipo de senha

Tipo de senha

Nome*

Prioridade*

[💾 Salvar](#) [✖ Sair](#)

FIGURA 36 – TELA ADICIONAR TIPO DE SENHA
 FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login
2. O usuário deve ser Administrador

Pós-condições:

Nenhuma

Ator primário

Administrador

Fluxo principal:

1. O sistema apresenta a tela DV1

2. O usuário clica no botão “adicionar”(A1)(A2)
3. O sistema apresenta a tela DV2
4. O usuário preenche o campo “Nome”
5. O usuário preenche o campo “Prioridade(R1)(R2)
6. O usuário clica em “salvar”(A3) (E1)
7. O caso de uso é encerrado

Fluxos alternativos:

A1. O usuário clica no botão “Editar”

1. O sistema apresenta a tela DV2

A2. O usuário clica no botão ‘Excluir’

1. O sistema apresenta a mensagem “Deseja mesmo excluir Prioritário?”

A3. O usuário clica em “Sair”

1. O sistema apresenta a tela DV1

Fluxo de Exceção:

E1. O usuário clica em “salvar” sem preencher os campos

1. O sistema emite mensagem “Obrigatório Nome” e “ Obrigatório Prioridade”

Regra de Negócio:

R1. Quanto menor a prioridade mais prioritária o tipo de senha será.

R2. O campo prioridade só aceita números

UC011 – Mover senhas para histórico

Descrição:

Caso de uso que move as senhas para a tabela de histórico de senhas

Data View

Não se aplica

Pré-condições:

1. Alcançar o horário programado

Pós-condições

Nenhuma

Ator primário:

Sistema

Fluxo principal:

1. O sistema move todas as senhas geradas antes do dia atual para a tabela de histórico de senhas

Fluxo de Exceção:

E1. O sistema não esta rodando no horário programado

1. O sistema não faz nada

Regra de Negócio:

R1. Horário deve estar programado para 00:05:00

R2. O sistema deve executar o UC-011 sem respeitar a regra do horário

UC012 – Cofigurar Sistema

Descrição:

Caso de uso serve para alterar parâmetros do sistema

Data View

DV1. Configurar

Configuração

Configurações cadastradas			
Chave ↕	Descrição ↕	Valor ↕	Ações
USUARIO_WS	Nome do usuário para utilizar o Web Service das senhas	ping	✎ Editar
SENHA_WS	Senha do usuário para utilizar o Web Service das senhas	pong	✎ Editar
POLL_RECEPCAO	Intervalo em segundos que leva para atualizar a tela de recepção	10	✎ Editar
POLL_ATENDIMENTO	Intervalo em segundos que leva para atualizar a tela de atendimento	10	✎ Editar
NOME_EMPRESA_SENHA	Nome da empresa a ser impresso no papel da senha	FUP	✎ Editar
POLL_PAINEL_SENHA	Intervalo em segundos que leva para atualizar o painel de senha	5	✎ Editar
POLL_MAPA	Intervalo em segundos que leva para atualizar o mapa	30	✎ Editar
POLL_DASHBOARD	Intervalo em segundos que leva para atualizar a tela do dashboard	30	✎ Editar

(1 of 1) 1 10

Total: 8

FIGURA 37 – TELA CONFIGURAÇÕES CADASTRADAS
FONTE: OS AUTORES (2016)

DV2. Editar configuração

Configuração

Chave	USUARIO_WS
Descrição	Nome do usuário para utilizar o Web Service das senhas
Valor	<input type="text" value="ping"/>

[💾 Salvar](#)
[✕ Sair](#)

FIGURA 38 – TELA EDITAR CONFIGURAÇÃO
FONTE: OS AUTORES (2016)

Pré-condições:

1. Executar UC – Login como administrador

Pós-condições:

Nenhuma

Ator primário:

Administrador

Fluxo principal:

1. O sistema apresenta a tela DV1
2. O usuário clica no botão “Editar”
3. O sistema apresenta a tela DV2
4. O usuário altera o campo “Valor”
5. O usuário clica no botão “Salvar”(A1)
6. O caso de uso é encerrado

Fluxo alternativo:

A1. O usuário clica no botão “Sair”

1. O sistema retorna para DV1

Regra de Negócio:

R1. Somente o administrador pode editar a configuração

UC013 – Trocar senha

Descrição:

Caso de uso para alterar senha

Data View:

DV1. Trocar senha

Trocar senha

Usuário	
Nome	Fulano
Sobrenome	de Tal
Chave	atendente
Posto de atendimento	UFPR SEPT
Tipo de usuário*	atendente
E-mail	fulano@email.com

Trocar senha	
Senha*	<input type="text"/>
Confirme a senha*	<input type="text"/>
Lembrete	<input type="text" value="atendente"/>

[Salvar](#)

FIGURA 39 – TROCAR SENHA
 FONTE: OS AUTORES (2016)

Pré-condições:

1. O sistema tiver executado o UC-Login

Pós-condições:

1. Nenhuma

Ator primário:

Usuário

Fluxo principal:

1. O sistema apresenta a tela DV1
2. O usuário preenche o campo “Senha”
3. O usuário preenche o campo “Confirme a senha”
4. O usuário preenche o campo “Lembrete”
5. O usuário clica no botão “Salvar”(E1)
6. O caso de uso é encerrado

Fluxo de Exceção

E1. O usuário clica em “salvar” sem preencher os campos

1. O sistema emite mensagem “Obrigatório Senha” e “Obrigatório Confirme Senha”

Regra de Negócio:

R1. A confirmação da senha tem que ser igual a senha digitada

APÊNDICE D – DIAGRAMA DE ENTIDADE RELACIONAMENTO

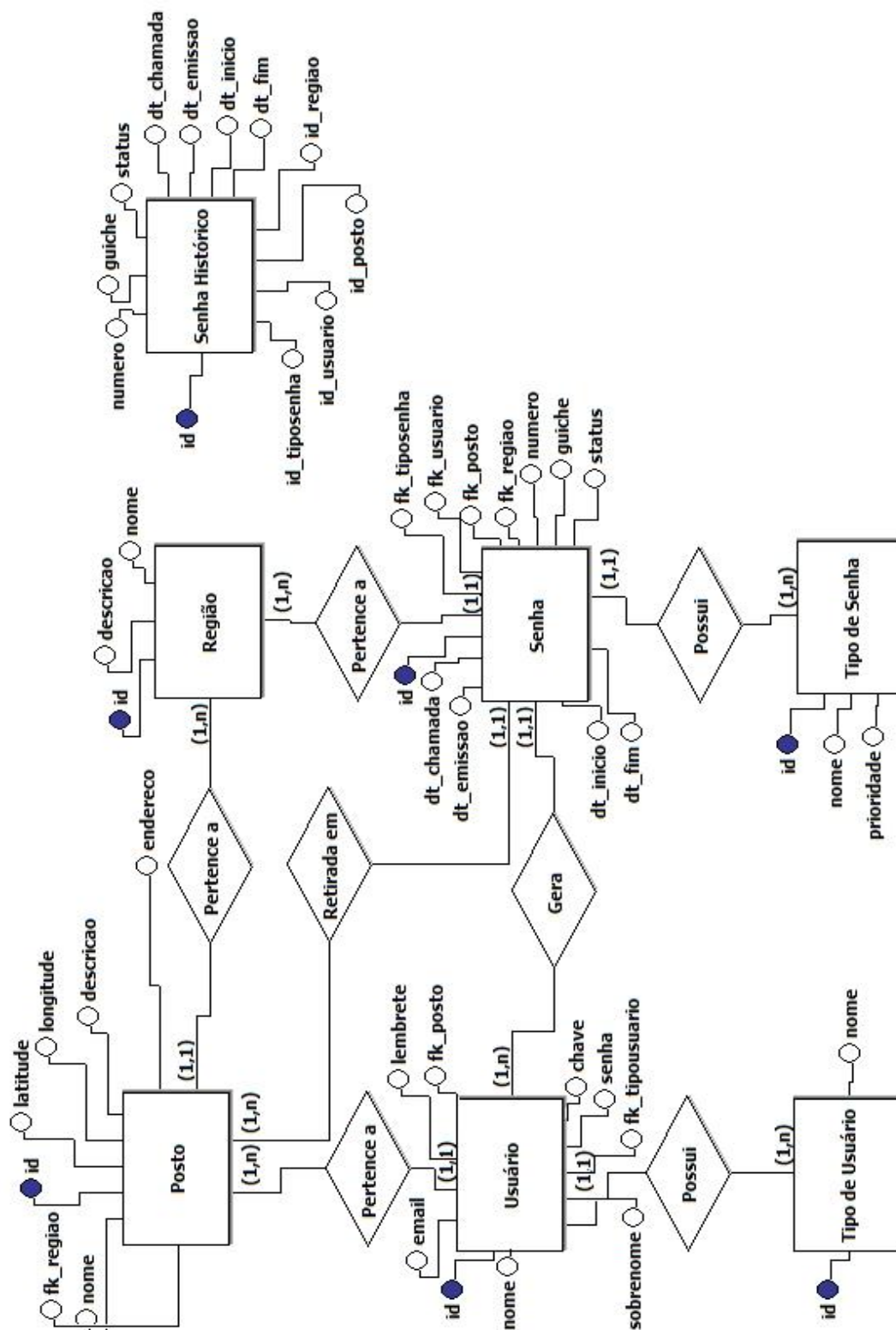


FIGURA 40 - DER: DIAGRAMA DE ENTIDADE RELACIONAMENTO
 FONTE: OS AUTORES (2016)

APÊNDICE E – DIAGRAMA DE ENTIDADE RELACIONAMENTO (LÓGICO)

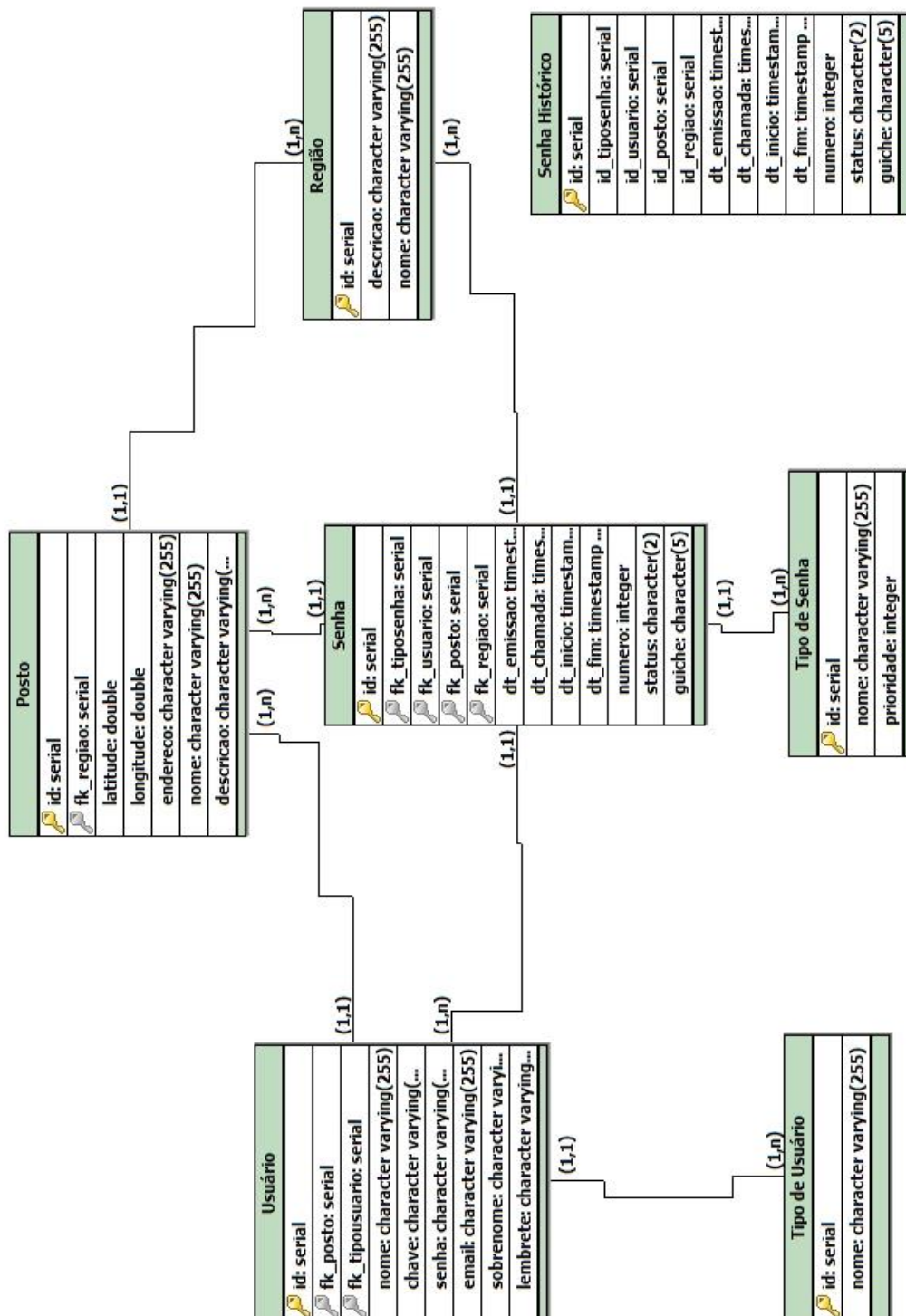


FIGURA 41 – DIAGRAMA DE ENTIDADE RELACIONAMENTO (LÓGICO)
 FONTE: OS AUTORES (2016)

APÊNDICE F – DIAGRAMA DE CLASSES

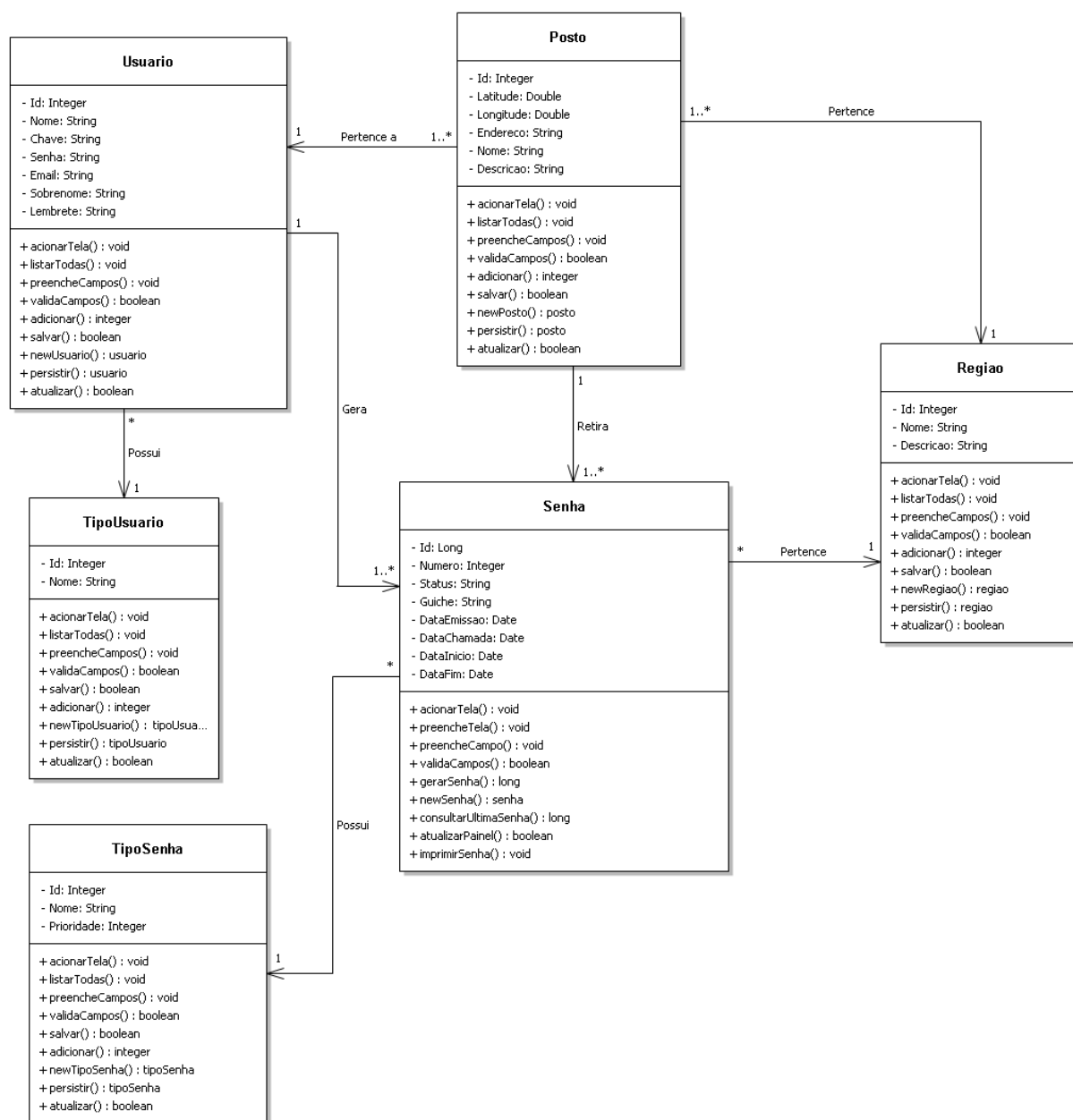
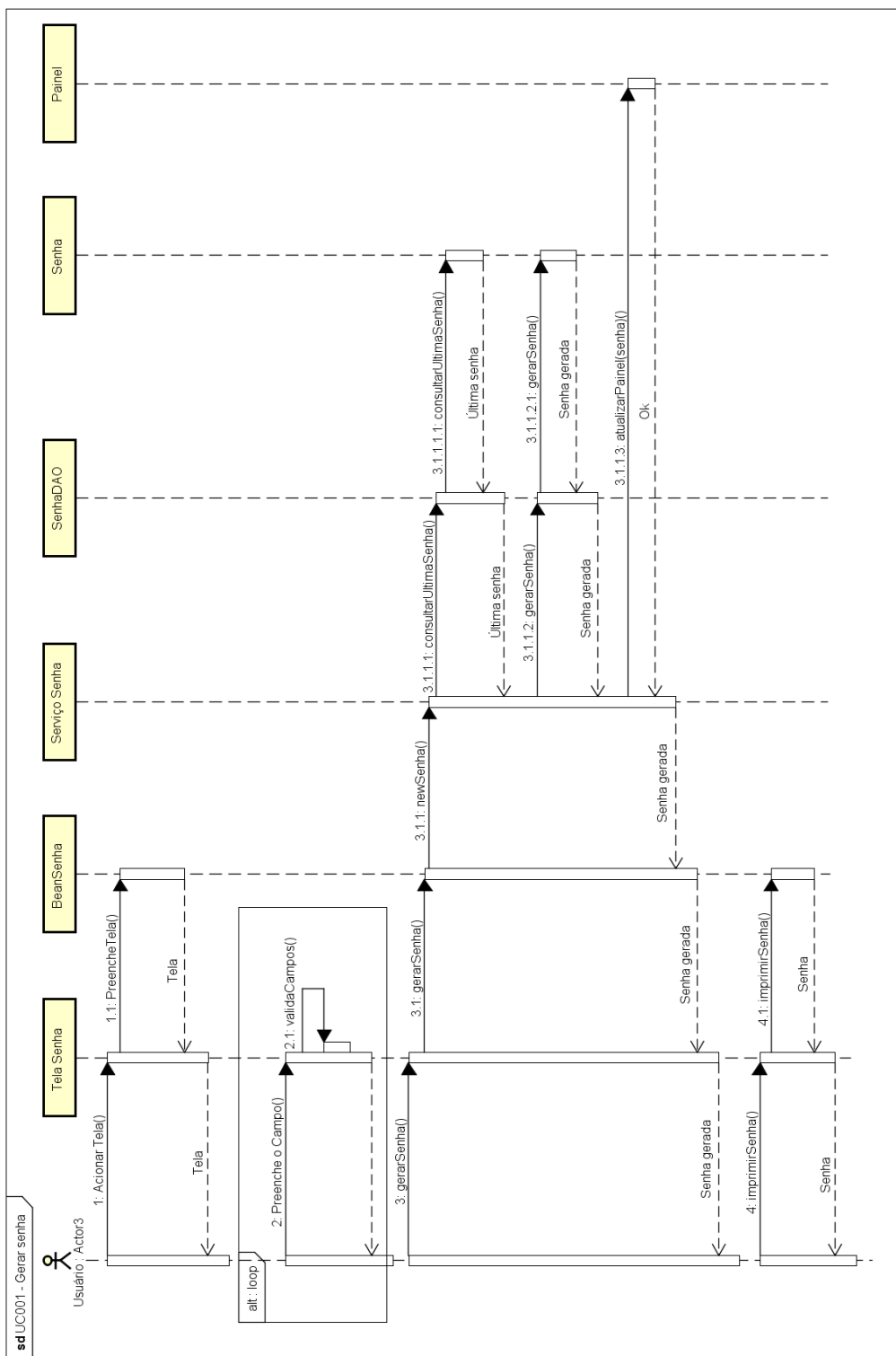


FIGURA 42 – DIAGRAMA DE CLASSE
FONTE: OS AUTORES (2016)

APÊNDICE G – DIAGRAMAS DE SEQUÊNCIA

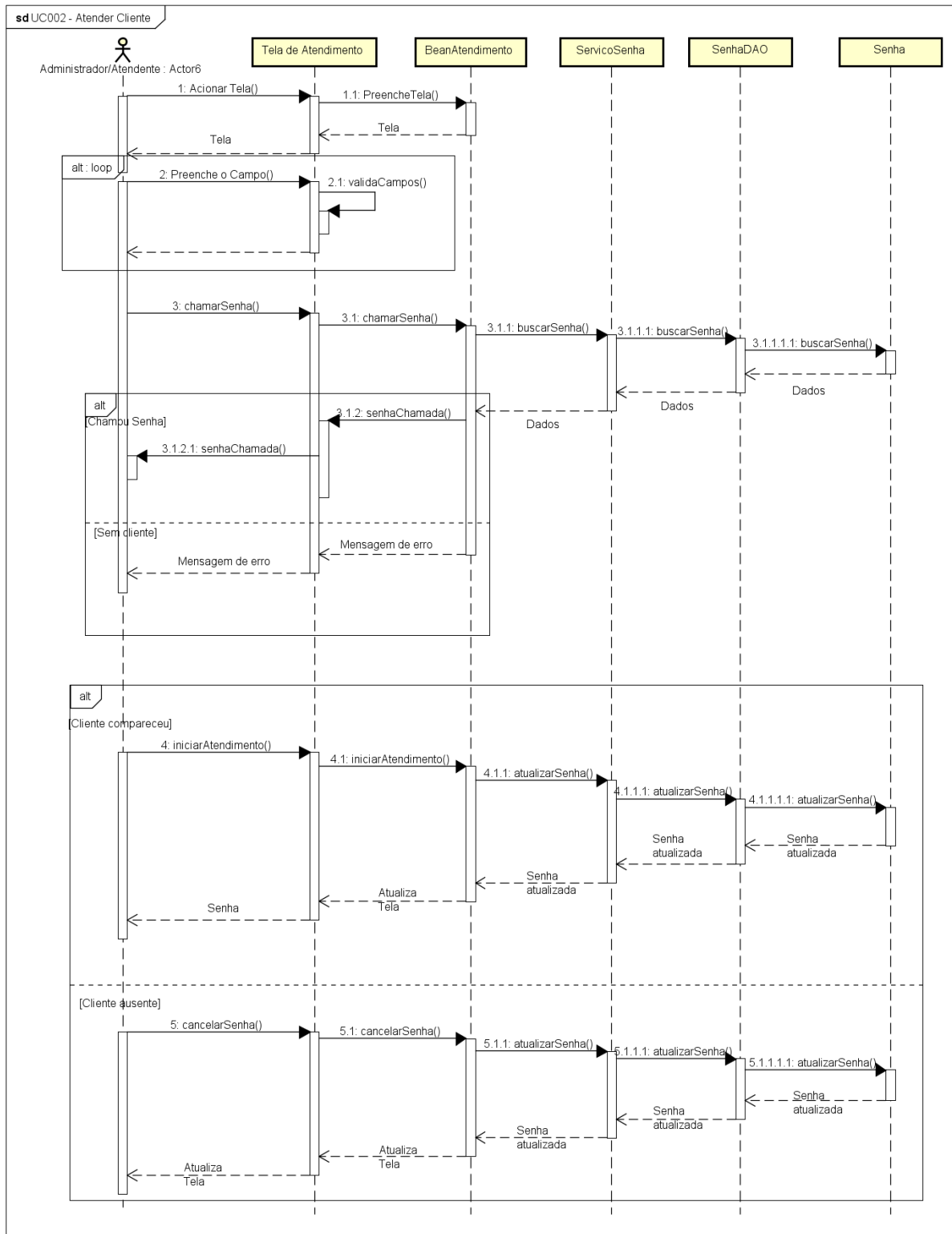
Diagrama de Sequência 001 – Gerar Senha



powered by Astah

FIGURA 43 – DIAGRAMA DE SEQUÊNCIA: GERAR SENHA
 FONTE: OS AUTORES (2016)

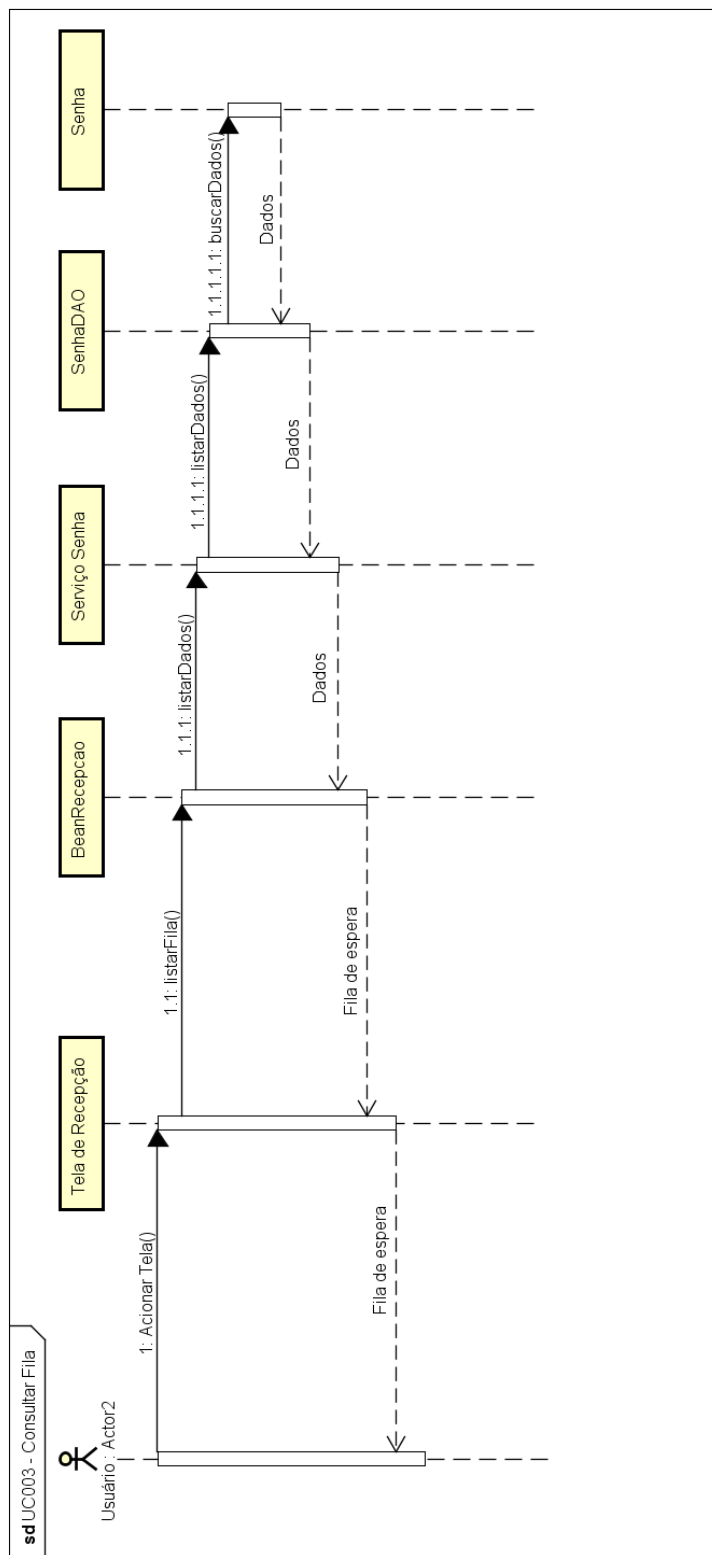
Diagrama de Sequência 002 – Atender Cliente



powered by Astah

FIGURA 44 – DIAGRAMA DE SEQUÊNCIA: ATENDER CLIENTE
 FONTE: OS AUTORES (2016)

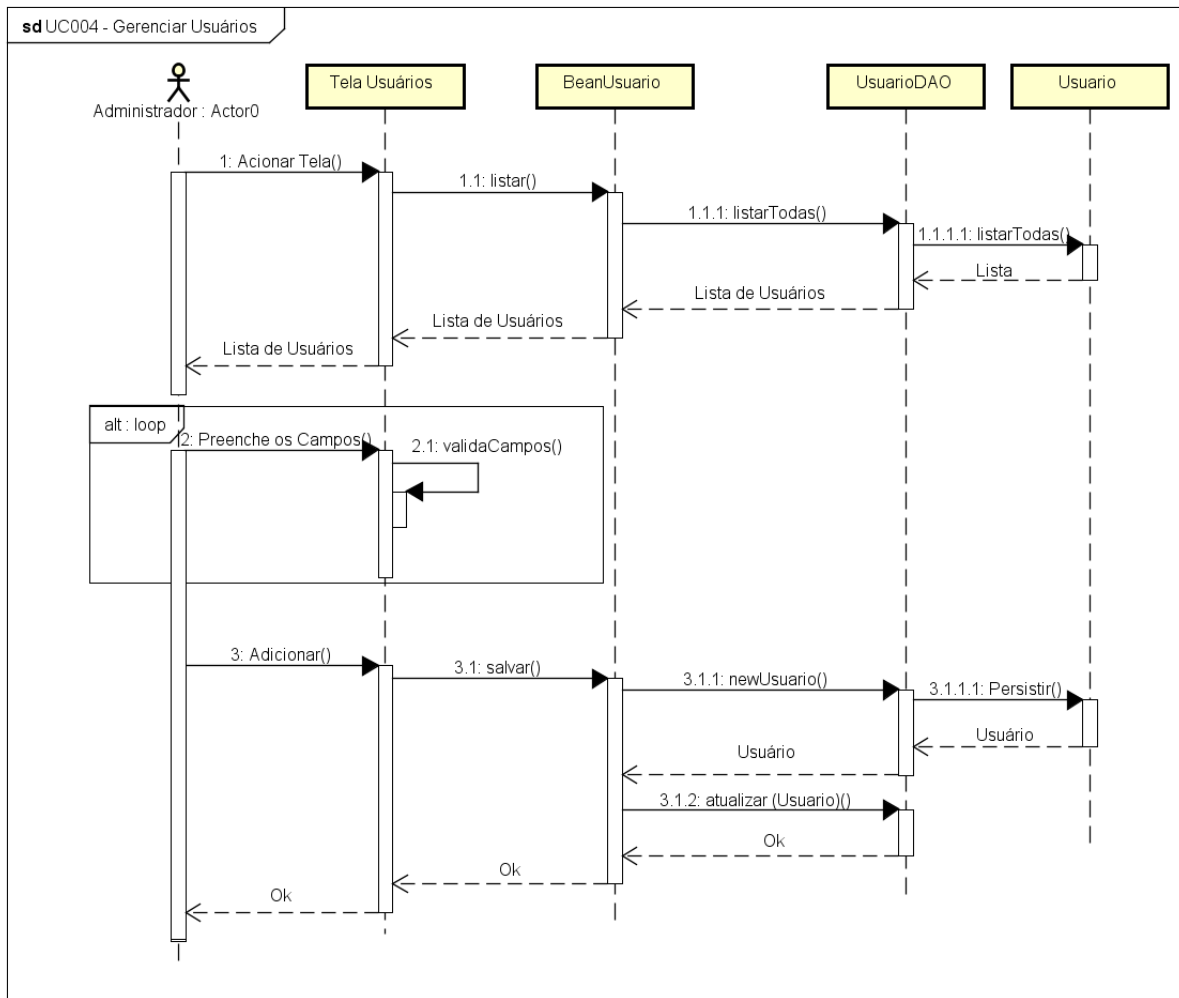
Diagrama de Sequência 003 - Consultar Fila



powered by Astah

FIGURA 45 – DIAGRAMA DE SEQUÊNCIA: CONSULTAR FILA
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 004 – Gerenciar Usuários



powered by Astah

FIGURA 46 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR USUÁRIOS
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 005 – Gerenciar Posto

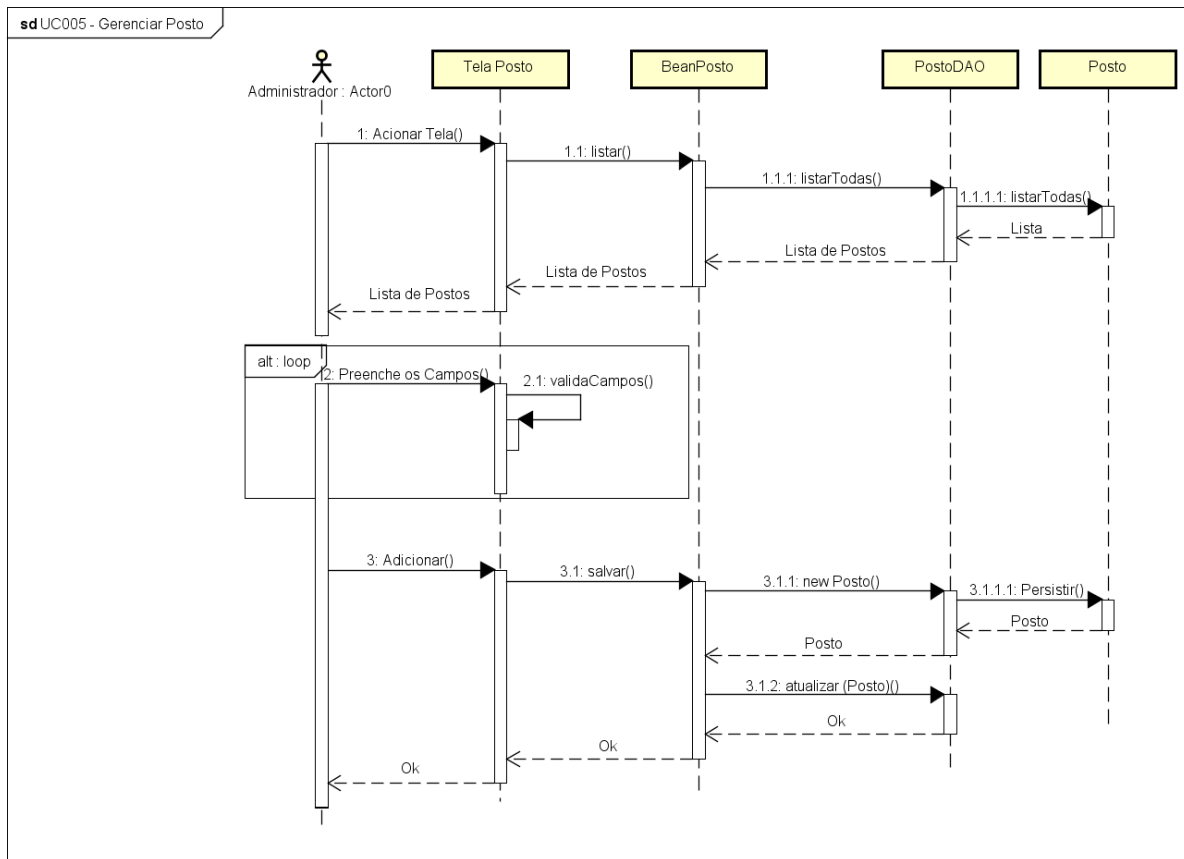
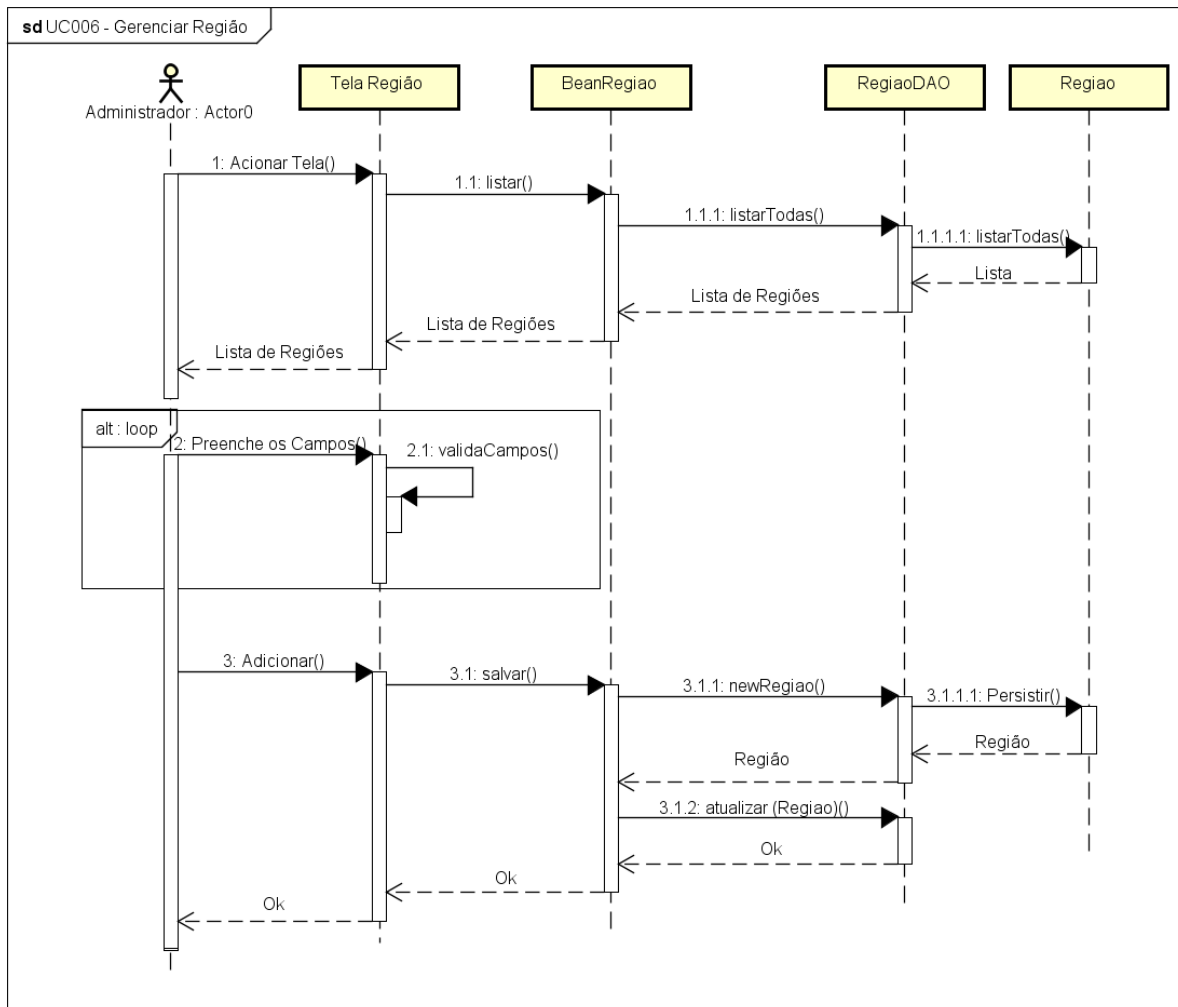


FIGURA 47 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR POSTO
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 006 – Gerenciar Região



powered by Astah

FIGURA 48 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR REGIÃO
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 007 – Gerar Relatório

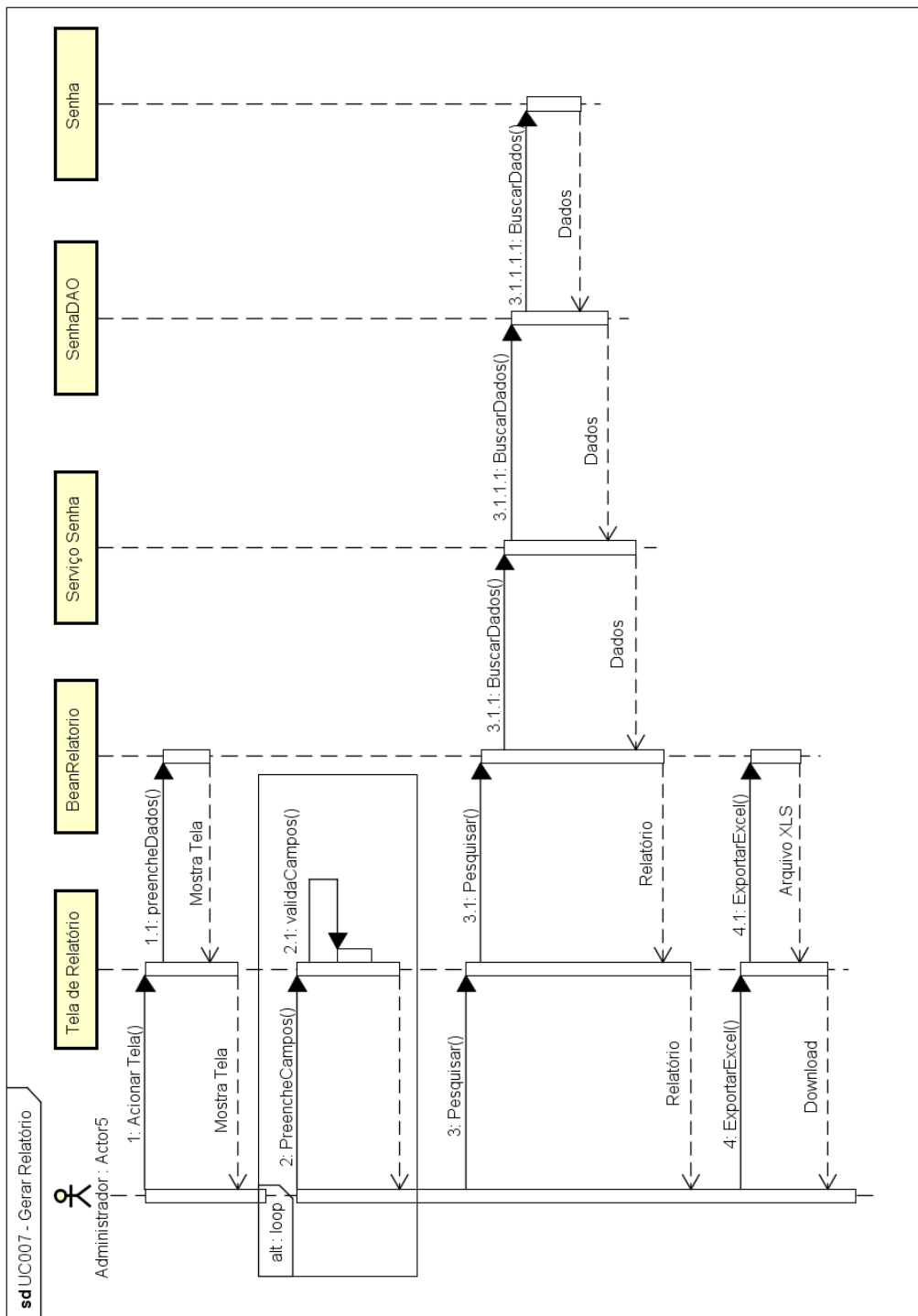


FIGURA 49 – DIAGRAMA DE SEQUÊNCIA: GERAR RELATÓRIO
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 008 – Efetuar Login

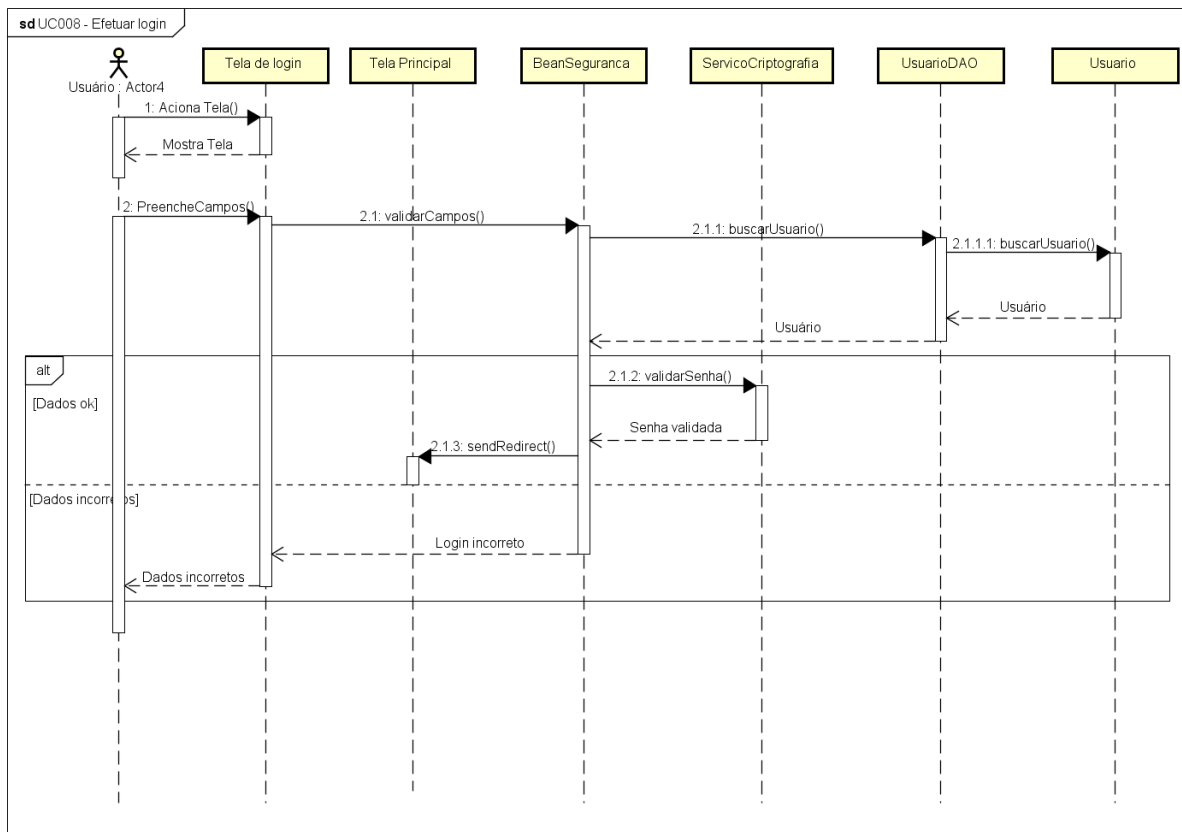


FIGURA 50 – DIAGRAMA DE SEQUÊNCIA: EFETUAR LOGIN
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 009 – Analisar Gráficos

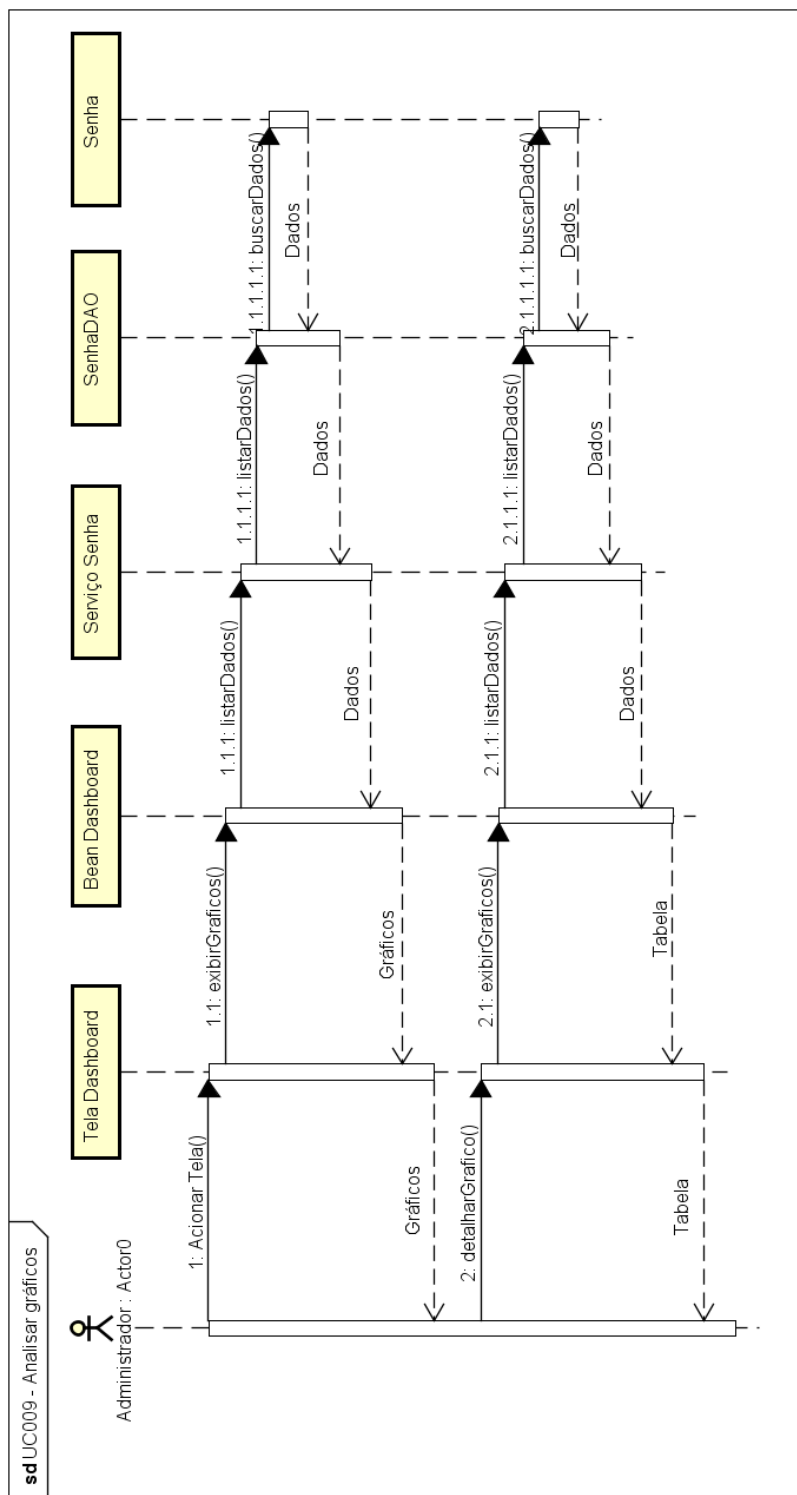
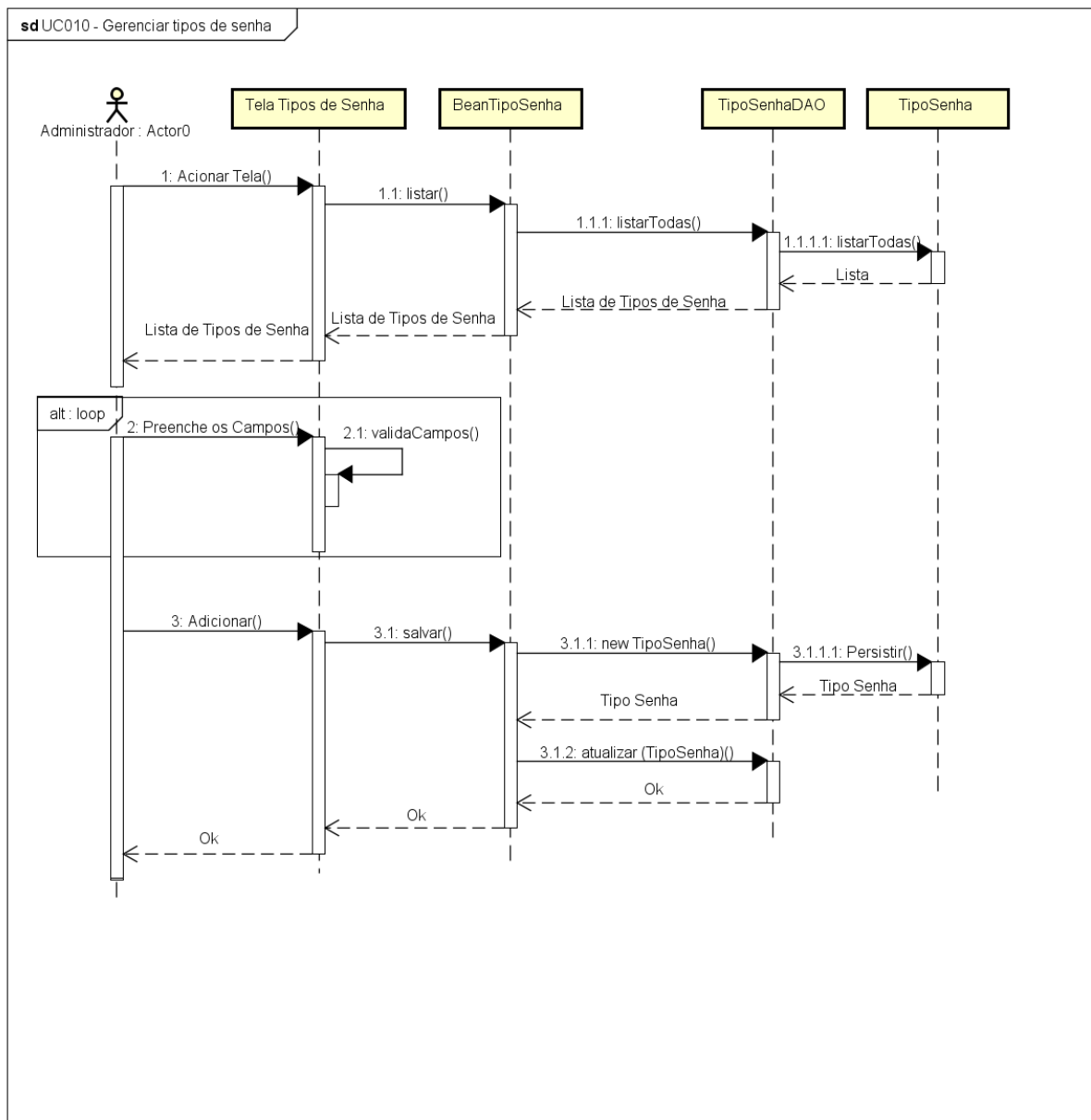


FIGURA 51 – DIAGRAMA DE SEQUÊNCIA: ANALISAR GRÁFICOS
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 010 – Gerenciar tipos de senha



powered by Astah

FIGURA 52 – DIAGRAMA DE SEQUÊNCIA: GERENCIAR TIPOS DE SENHA
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 011 – Mover senhas para histórico

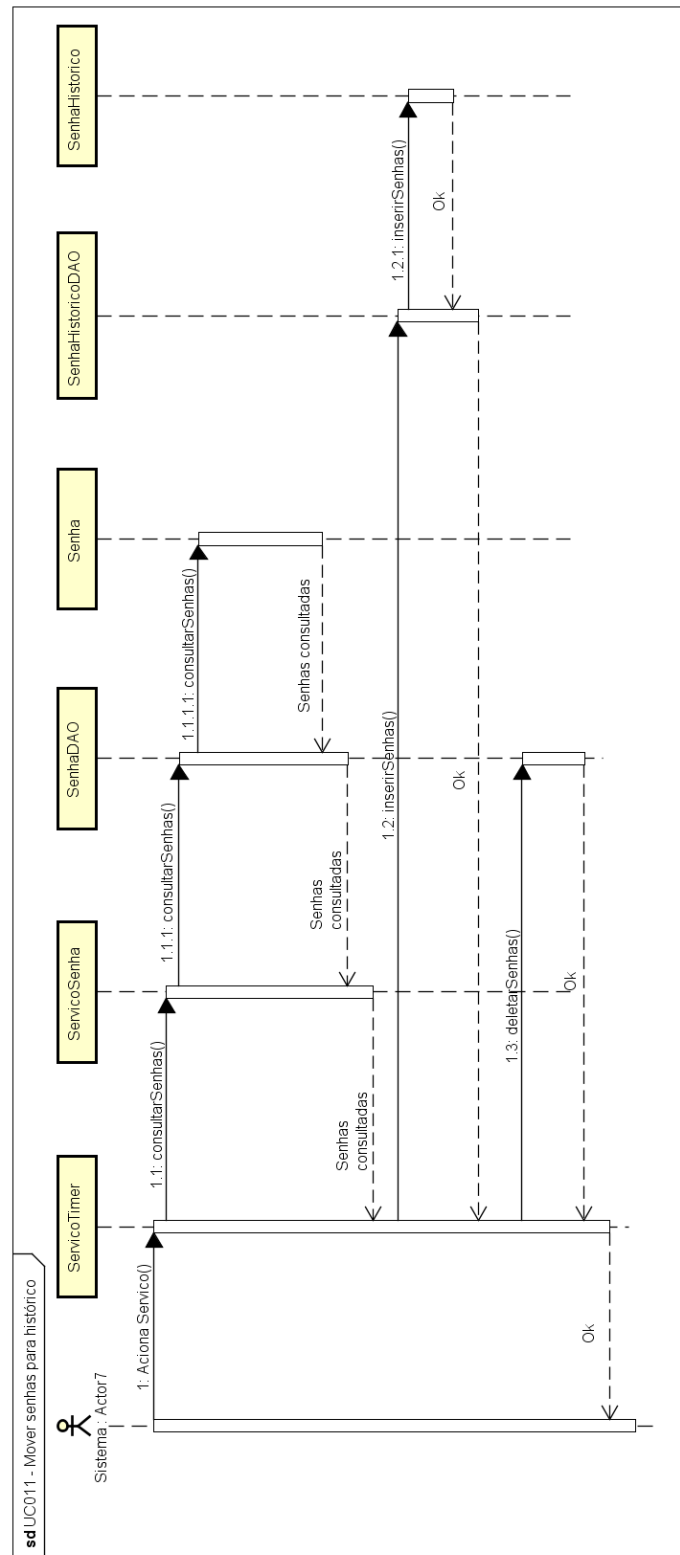
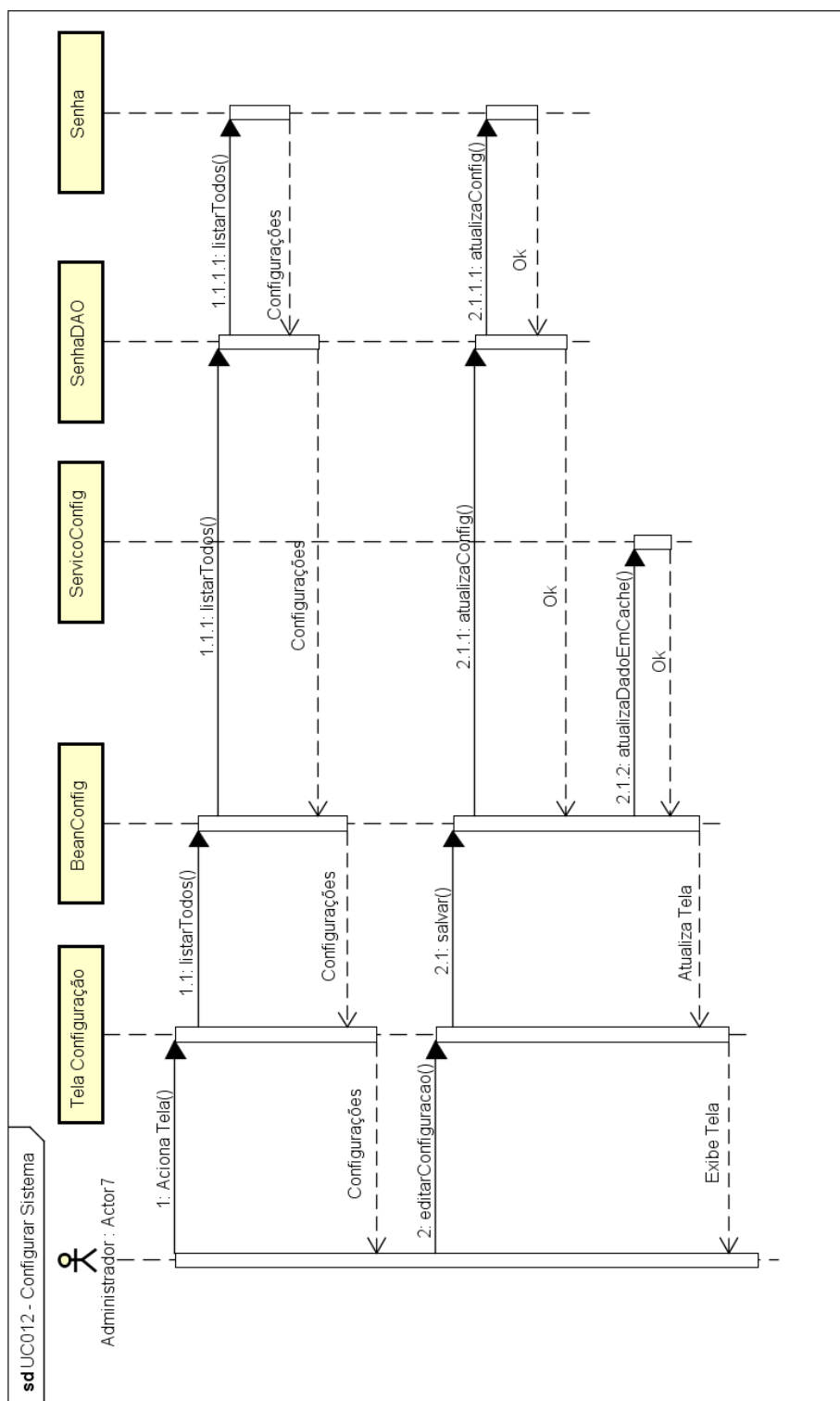


FIGURA 53 – DIAGRAMA DE SEQUÊNCIA: MOVER SENHAS PARA HISTÓRICO
 FONTE: OS AUTORES (2016)

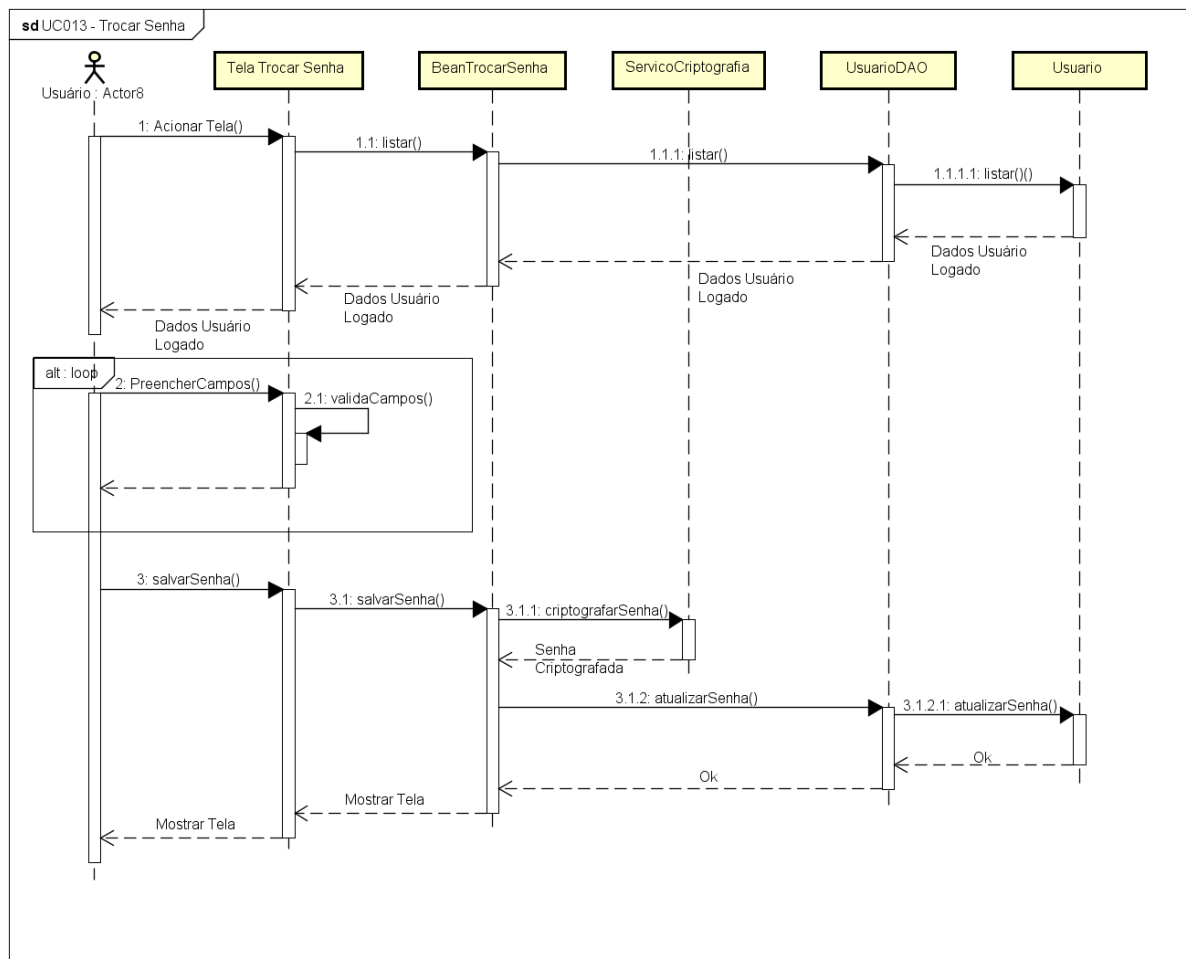
Diagrama de Sequência 012 – Configurar Sistema



powered by Astah

FIGURA 54 – DIAGRAMA DE SEQUÊNCIA: CONFIGURAR SISTEMA
 FONTE: OS AUTORES (2016)

Diagrama de Sequência 013 – Trocar Senha



powered by Astah

FIGURA 55 – DIAGRAMA DE SEQUEÊNCIA: TROCAR SENHA
 FONTE: OS AUTORES (2016)

APÊNDICE H – WEB SERVICE RESTFUL

Descrição: Teste do REST	
URI: http://{host}/fup/rest/senha/teste	Método: GET
Recebe: Nada	Retorna: Texto
Detalhes: O retorno é a mensagem "PONG!" seguindo da data do servidor no formato ISSO 8601. Exemplo: PONG! 2016-09-24T21:58:04.960Z	

Descrição: Retorna uma lista dos tipos de senha cadastrados pelo administrador do sistema.	
URI: http://{host}/fup/rest/senha/getListaTipoSenha	Método: GET
Recebe: Nada	Retorna: JSON – Lista de objetos de tipo de senha
Detalhes: Exemplo de retorno: <pre>[{ "id": 3, "nome": "Prioritário", "prioridade": 1 }, { "id": 4, "nome": "Retorno", "prioridade": 2 }, { "id": 1, "nome": "Normal", "prioridade": 4 }]</pre>	

Descrição: Valida usuário e retorna alguma senha caso o usuário esteja atendido. Datas no formato ISO 8601.	
URI: http://{host} /fup/rest/senha/validarUsuario/{chaveUsuario}	Método: GET
Recebe: Parâmetro {chaveUsuario} tipo string	Retorna: JSON – Objeto usuário
Detalhes: Exemplo de retorno com objeto senha:	
<pre> { "id": 9, "nome": "Hugo", "chave": "hugo", "sobrenome": "Fujiwara", "tipoUsuario": "administrador", "posto": "UFPR SEPT", "senhaEmAtendimento": { "id": 438, "numero": 1, "tipoSenha": { "id": 1, "nome": "Normal", "prioridade": 4 }, "status": "AT", "guiche": "1", "dataEmissao": "2016-09-24T22:01:45.196Z", "dataChamada": "2016-09-24T22:01:56.082Z", "dataInicio": "2016-09-24T22:02:31.666Z" } } </pre>	

Descrição: Gera senha para o posto de atendimento onde o usuário está relacionado.	
URI: http://{host} /fup/rest/senha /gerarSenha/{chaveUsuario}/{idTipoSenha}	Método: POST
Recebe: Parâmetros {chaveUsuario} tipo string e {idTipoSenha} tipo número	Retorna: JSON – Objeto senha
Detalhes: Exemplo: <pre>{ "id": 439, "numero": 2, "tipoSenha": { "id": 1, "nome": "Normal", "prioridade": 4 }, "status": "GE", "guiche": null, "dataEmissao": "2016-09-24T22:15:25.040Z", "dataChamada": null, "dataInicio": null }</pre>	

Descrição: Chama uma senha para o usuário, caso haja alguma disponível.	
URI: http://{host} /fup/rest/senha /chamarSenha/{chaveUsuario}/{guiche}	Método: PUT
Recebe: Parâmetro {chaveUsuario} tipo string e {guiche} tipo string com comprimento máximo de 5 caracteres	Retorna: JSON – Objeto senha
Detalhes: Troca o status da senha de “GE” para “CH”. Exemplo do retorno: <pre>{ "id": 439, "numero": 2, "tipoSenha": { "id": 1, "nome": "Normal", "prioridade": 4 } }</pre>	

```

    },
    "status": "CH",
    "guiche": "A1",
    "dataEmissao": "2016-09-24T22:15:25.040Z",
    "dataChamada": "2016-09-24T22:18:53.034Z",
    "dataInicio": null
  }

```

Descrição: Inicia o atendimento para a senha informada	
URI: http://{host} /fup/rest/senha/iniciarAtendimento/{idSenha}	Método: PUT
Recebe: Parâmetro {idSenha} tipo número	Retorna: JSON - Booleano
Detalhes: Troca o status da senha de “CH” para “AT”. Somente funciona para senhas no status “CH”. Exemplo de resposta: true	

Descrição: Conclui o atendimento da senha informada.	
URI: http://{host} /fup/rest/senha/concluirAtendimento/{idSenha}	Método: PUT
Recebe: Parâmetro {idSenha} tipo número	Retorna: JSON – Booleano
Detalhes: Troca o status da senha de “AT” para “CO”. Somente funciona para senhas no status “AT”. Exemplo de resposta: true	

Descrição: Cancela a senha, situação de quando o cliente se evadiu do local.	
URI: http://{host} /fup/rest/senha/cancelarAtendimento/{idSenha}	Método: PUT
Recebe: Parâmetro {idSenha} tipo número	Retorna: JSON – Booleano
Detalhes: Troca o status da senha de “CH” para “CA”. Somente funciona para senhas no status “CH”. Exemplo de resposta: true	