

**CENTRO UNIVERSITÁRIO SENAC
CAMPUS SANTO AMARO**

**CESAR CORDEIRO
HENRIQUE MACHADO CHAVES
JOSÉ HENRIQUE TEIXEIRA**

GORDÃO BARBEARIA: SRS (System Requirement Specification)

**SÃO PAULO – SP
2016**

Especificação de Requisitos de Software Para Gordão Barbearia

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
29/03/2016	0.1	Realizado primeiro esboço respeitando o template escolhido	Cesar Quadros Henrique Chaves José Henrique
12/04/2016	0.2	Ajustado informações de RF, RNF e RN	Cesar Quadros Henrique Chaves José Henrique
19/04/2016	0.3	Atualização de RNF, RN e Caso de uso	Cesar Quadros Henrique Chaves José Henrique
03/05/2016	0.4	Inclusão de novas requisitos	Cesar Quadros Henrique Chaves José Henrique
17/05/2016	0.5	Ajustes de requisitos e regras de negócio.	Henrique Chaves
20/05/2016	0.6	Ajuste nos requisitos e DDL	Cesar Quadros
04/06/2016	0.7	Ajustes nas regras de negócio	Cesar Quadros Henrique Chaves
06/06/2016	0.8	Ajustes no documento e nos itens 3 e 5	Henrique Chaves
07/06/2016	0.9	Ajustes no item 4 e formatação.	Henrique Chaves José Henrique
11/06/2016	1.0	Ajustes finais	Cesar Quadros Henrique Chaves José Henrique

Sumário

1. Introdução	4
1.1 Finalidade	4
1.2 Escopo	4
1.3 Definições, Acrônimos e Abreviações	4
2. Requisitos e Regra de Negócios	5
2.1 Requisitos Funcionais	5
2.2 Requisitos Não Funcionais.....	5
2.3 Regras de Negócios	5
3. Relatórios de Caso de Uso.....	7
3.1 Diagrama de Caso de Uso.....	7
3.2 Caso de Uso	7
4. Matrizes de Relacionamento	10
4.1 Matriz Requisitos Funcionais X Caso de Uso	10
4.2 Matriz Requisitos Funcionais X Regra de Negócios.....	10
5. Modelagem Banco de Dados.....	11
5.1 Modelo Conceitual	11
5.2 Diagrama Entidade Relacionamento.....	11
5.3 DDL - Data Definition Language	12
5.4 DML - Data Manipulation Language.....	13

Especificação de Requisitos de Software

1. Introdução

1.1 Finalidade

Desenvolver um sistema de agendamento para a Gordão Barbearia de forma que o cliente possa melhorar seu controle/gerenciamento aumentando sua receita.

1.2 Escopo

Será desenvolvido uma agenda eletrônica com o objetivo de permitir ao usuário além de realizar os agendamentos necessários obter diversas outras informações, trazendo com isso um mapeamento melhor do negócio e suas reais necessidades.

1.3 Definições, Acrônimos e Abreviações

RF	Requisito Funcional
RN	Regra de Negócio
RNF	Requisito não Funcional
Java	Linguagem de programação
BD	Banco de dados
SQL	Linguagem de pesquisa declarativa para BD
UML	Linguagem para visualização, especificação, construção e documentação.

2. Requisitos e Regra de Negócios

2.1 Requisitos Funcionais

Requisitos	Importância	Complexidade
RF01 - Tela com as opções de cadastro dos clientes	Alta	Média
RF02 - Tela que permite extrair relatórios, por data, cliente, funcionário e unidade	Média	Alta
RF03 - Tela que permite realizar o agendamento	Alta	Alta
RF04 - Tela com a opção de visualização dos horários agendados, em espera e cancelado de acordo com o dia selecionado	Alta	Média
RF05 - Tela inicial que permite escolher qual recurso será utilizado	Média	Alta

2.2 Requisitos Não Funcionais

Requisitos	Importância	Complexidade
RN01 - Sistema desenvolvido em Java	Alta	Alta
RN02 - Sistema desenvolvido com Banco de Dados SQLite.	Alta	Alta
RN03 - Interface intuitiva.	Alta	Média
RN04 - O Sistema deve funcionar em sistemas operacional Windows.	Media	Alta

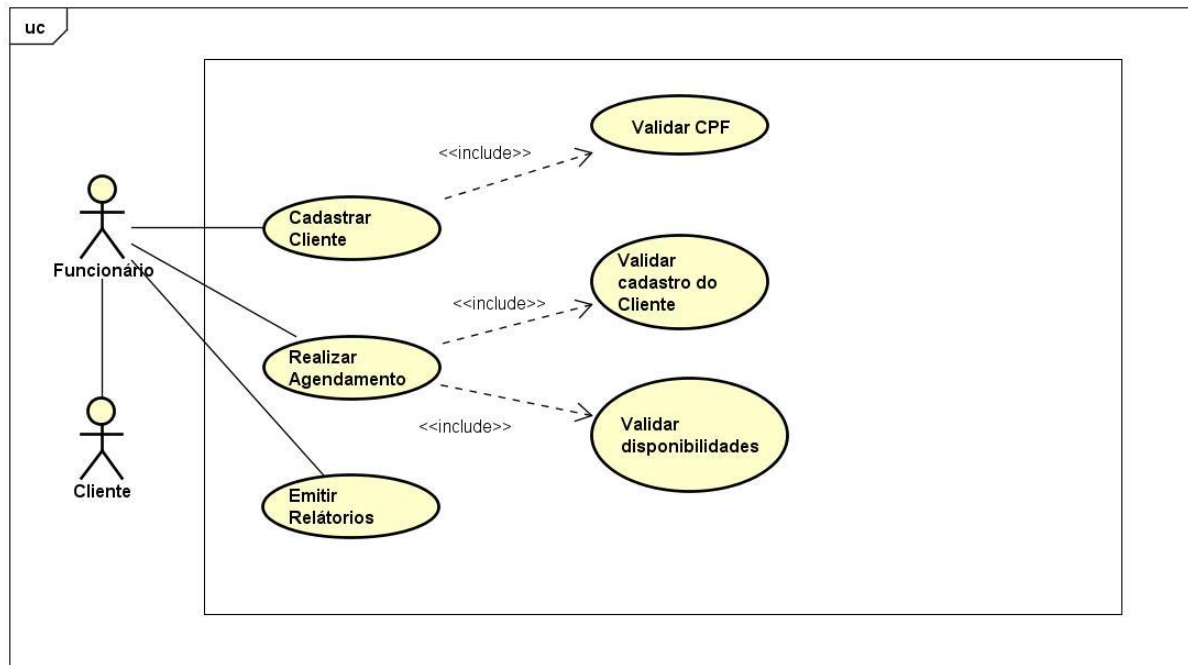
2.3 Regras de Negócios

Regras de Negócio	Definição
RN01 - O sistema poderá gerar diversos tipos de relatórios.	O usuário deve conseguir extrair relatórios por período selecionando cliente, funcionário ou unidade, além de poder escolher os status (todos, agendados, atendidos, cancelados e em espera).
RN02 - O sistema é manipulado por um único funcionário.	Todas as alterações/manipulações do sistema são realizado por um único funcionário não existindo acesso simultâneo.
RN03 - Todos os funcionários realizam todos os serviços oferecidos.	Todos os funcionários são capacitados para atender os clientes podendo realizar qualquer um dos tipos de serviços cadastrados no sistema
RN04 - Todas as unidades oferecem todos os serviços.	Todas as unidades devem oferecer os serviços que estão cadastrados no sistema.
RN05 - Horário de atendimento dos Funcionários	Cada funcionário possui seu horário de entrada e saída, porem, o horário de almoço é definido pelo

	próprio funcionário, desta forma o não agendamento para este período fica de responsabilidade de quem fará os agendamentos.
RN06 - Fila de espera	É permitido cadastrar no sistema mais de um cliente para o mesmo horário, sendo que o primeiro cadastro é o válido e os demais são marcados como “Em espera”, fica a cargo do funcionário validar com o cliente agendado se ele irá comparecer, assim como, se for o caso promover este para cancelado e o que está em espera para agendado. O sistema sempre permitirá realizar agendamentos em espera independente se este já exista.

3. Relatórios de Caso de Uso

3.1 Diagrama de Caso de Uso



powered by Astah

3.2 Caso de Uso

Cadastrar cliente. (UC01)

Nome Ator	Funcionário.
Descrição	O funcionário interage com o sistema para realizar o cadastro de novos clientes.
Pré condição	É necessário que o sistema esteja instalado e acessando o banco de dados.
Cenário principal	O funcionário deve acessar o sistema e efetuar o cadastro do cliente, para isso será necessário o Nome, telefone e CPF, só pode ser feito um cadastro por CPF, logo o funcionário poderá validar se o cliente já é cadastrado consultando o CPF.
Cenário alternativo	Caso o funcionário não valide o CPF todo o preenchimento realizado será perdido, neste ponto é exibido uma mensagem notificando que se trata de um cliente já cadastrado.

Validar CPF. (UC02)

Caso de uso associado	Cadastrar cliente.
Descrição	O funcionário interage com o sistema para realizar a validação do CPF.
Pré condição	É necessário que o sistema esteja instalado e acessando o banco de dados.
Cenário principal	O funcionário deve acessar o sistema e validar o CPF do cliente.
Cenário alternativo	Quando o funcionário digitar o CPF e solicitar a consulta é retornado uma mensagem informando se o CPF está disponível ou não disponível para ser cadastrado.

Realizar Agendamento. (UC03)

Nome Ator	Funcionário.
Descrição	O funcionário interage com o sistema para realizar o agendamento.
Pré condição	É necessário que o sistema esteja instalado e acessando o banco de dados. Neste ponto é esperado que o cliente já esteja cadastrado assim como todos os funcionários elegíveis
Cenário principal	O funcionário deve acessar o sistema e efetuar o agendamento, que pode ser realizado por disponibilidade de hora ou do profissional que realizará o atendimento.
Cenário alternativo	O sistema não realizar agendamentos com horários iguais para o mesmo profissional, caso o funcionário tente fazer esta ação deve ser retornado uma mensagem notificando o erro.

Validar cadastro do cliente. (UC04)

Caso de uso associado	Realizar Agendamento.
Descrição	O funcionário valida se o cliente já possui cadastro.
Pré condição	É necessário que o sistema esteja instalado e acessando o banco de dados.
Cenário principal	O funcionário deve acessar o sistema e efetuar com o CPF do cliente se ele já é cadastrado.
Cenário alternativo	O sistema retorna uma mensagem notificando o status do CPF consultado.

Validar disponibilidade. (UC05)

Caso de uso associado	Realizar Agendamento.
Descrição	O funcionário validar se a hora e/ou profissional escolhido possui disponibilidade para o agendamento
Pré condição	É necessário que o sistema esteja instalado e acessando o banco de dados e também que o cliente já esteja cadastrado.
Cenário principal	O funcionário deve acessar o sistema e checar na tela de agendamento se o horário e/ou profissional estão disponíveis.
Cenário alternativo	Estas informações devem ser exibidas na tela, caso o funcionário tente agendar horários/profissionais conflitantes deve ser retornado uma mensagem de erro.

Relatórios (UC06)

Nome Ator	Funcionário.
Descrição	O funcionário interage com o sistema para realizar a extração de relatórios.
Pré condição	É necessário que o sistema esteja instalado e acessando o banco de dados.
Cenário principal	O funcionário deve acessar o sistema e selecionar o tipo de relatório que deseja (período, funcionário, cliente ou unidade).
Cenário alternativo 1	Caso o relatório solicitado não tenha informação nenhuma é retornado uma mensagem caso contrário o relatório é gerado.

4. Matrices de Relacionamento

4.1 Matriz Requisitos Funcionais X Caso de Uso

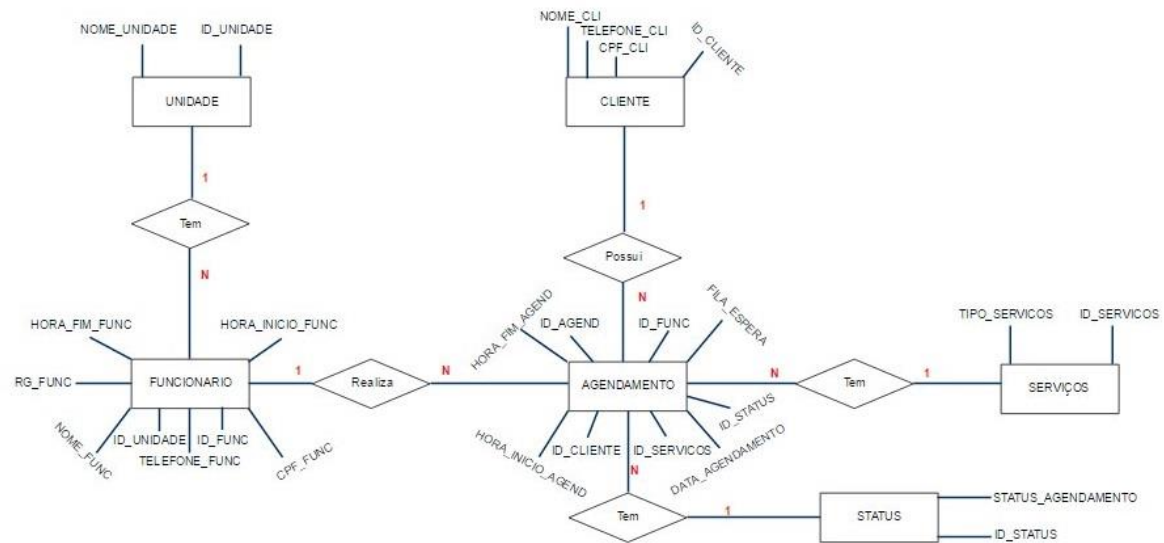
Requisitos/Caso de Uso	UC01	UC02	UC03	UC04	UC05	UC06
RF01	X	X				
RF02						X
RF03			X	X	X	
RF04	X					
RF05						

4.2 Matriz Requisitos Funcionais X Regra de Negócios

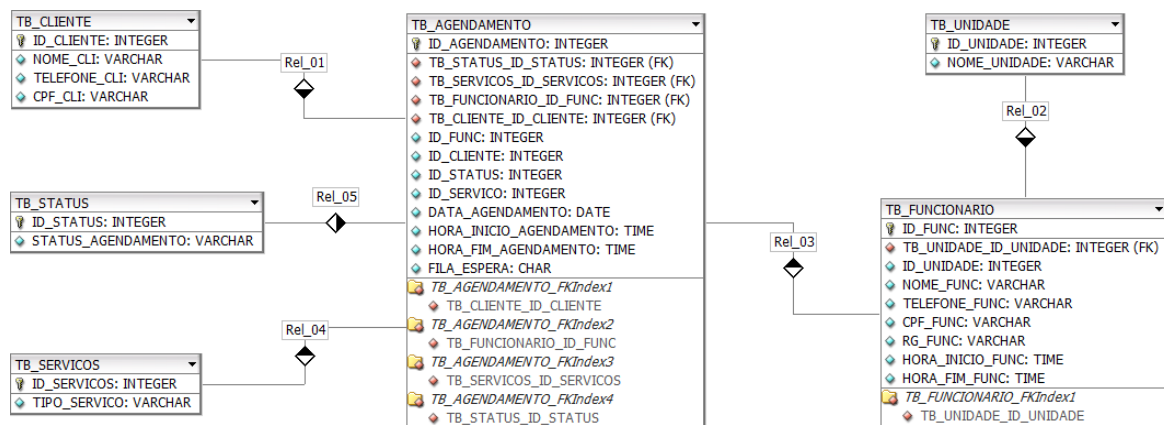
Requisitos/Regra de Negócios	RN01	RN02	RN03	RN04	RN05	RN06
RF01						
RF02	X					
RF03						
RF04						
RF05		X	X	X	X	X

5. Modelagem Banco de Dados

5.1 Modelo Conceitual



5.2 Diagrama Entidade Relacionamento



5.3 DDL - Data Definition Language

```
--BEGIN TRANSACTION;

DROP TABLE IF EXISTS TB_AGENDAMENTO;
DROP TABLE IF EXISTS TB_CLIENTE;
DROP TABLE IF EXISTS TB_FUNCIONARIO;
DROP TABLE IF EXISTS TB_SERVICOS;
DROP TABLE IF EXISTS TB_STATUS;
DROP TABLE IF EXISTS TB_UNIDADE;

CREATE TABLE TB_UNIDADE (
    ID_UNIDADE      INTEGER PRIMARY KEY AUTOINCREMENT,
    NOME_UNIDADE    VARCHAR(30) NOT NULL
);

CREATE TABLE TB_STATUS (
    ID_STATUS       INTEGER PRIMARY KEY AUTOINCREMENT,
    STATUS_AGEND    VARCHAR(15) NOT NULL
);

CREATE TABLE TB_SERVICOS (
    ID_SERVICO      INTEGER PRIMARY KEY AUTOINCREMENT,
    TIPO_SERVICO    VARCHAR(20) NOT NULL
);

CREATE TABLE TB_FUNCIONARIO (
    ID_FUNC         INTEGER PRIMARY KEY AUTOINCREMENT,
    ID_UNIDADE      INT NOT NULL,
    NOME_FUNC       VARCHAR(30) NOT NULL,
    CPF_FUNC        VARCHAR(15) NULL UNIQUE,
    RG_FUNC         VARCHAR(15) NOT NULL,
    TELEFONE_FUNC   VARCHAR(15) NOT NULL,
    HORA_INICIO_FUNC TIME(7) NOT NULL,
    HORA_FIM_FUNC    TIME(7) NOT NULL,
    CONSTRAINT FK_ID_UNIDADE FOREIGN KEY (ID_UNIDADE) REFERENCES TB_UNIDADE(ID_UNIDADE)
);

CREATE TABLE TB_CLIENTE (
    ID_CLIENTE      INTEGER PRIMARY KEY AUTOINCREMENT,
    NOME_CLI        VARCHAR(30) NOT NULL,
    TELEFONE_CLI    VARCHAR(15) NOT NULL,
    CPF_CLI         VARCHAR(15) NOT NULL UNIQUE
);

CREATE TABLE TB_AGENDAMENTO (
    ID_AGENDAMENTO  INTEGER PRIMARY KEY AUTOINCREMENT,
    ID_FUNC         INT NOT NULL,
    ID_CLIENTE      INT NOT NULL,
    ID_STATUS       INT NOT NULL,
    ID_SERVICO      INT NOT NULL,
    DATA_AGENDAMENTO DATE NOT NULL,
    HORA_INICIO_AGEND TIME(7) NOT NULL,
    HORA_FIM_AGEND  TIME(7) NOT NULL,
    CONSTRAINT FK_FUNCIONARIO FOREIGN KEY (ID_FUNC) REFERENCES TB_FUNCIONARIO(ID_FUNC),
    CONSTRAINT FK_CLIENTE FOREIGN KEY (ID_CLIENTE) REFERENCES TB_CLIENTE(ID_CLIENTE),
    CONSTRAINT FK_STATUS FOREIGN KEY (ID_STATUS) REFERENCES TB_STATUS(ID_STATUS),
    CONSTRAINT FK_SERVICO FOREIGN KEY (ID_SERVICO) REFERENCES TB_SERVICOS(ID_SERVICO)
);

COMMIT;
```

5.4 DML - Data Manipulation Language

```
--BEGIN TRANSACTION;

INSERT INTO `TB_UNIDADE` (ID_UNIDADE,NOME_UNIDADE)
VALUES (1,'SANTO AMARO'), (2,'BERRINI'), (3,'PINHEIROS');

INSERT INTO `TB_STATUS` (ID_STATUS,STATUS_AGEND)
VALUES (1,'AGENDADO'), (2,'ATENDIDO'), (3,'CANCELADO'), (4,'ESPERA');

INSERT INTO `TB_SERVICOS` (ID_SERVICO,TIPO_SERVICO)
VALUES (1,'CORTE'), (2,'BARBA');

INSERT INTO `TB_FUNCIONARIO`
(ID_FUNC,ID_UNIDADE,NOME_FUNC,CPF_FUNC,RG_FUNC,TELEFONE_FUNC,HORA_INICIO_FUNC,HORA_FIM_FUNC)
VALUES (1,1,'CHAVES','31974047857','325794777','976227960','10:00','19:00'),

(2,2,'HENRIQUE','24024024024','242424240','242424240','07:00','16:00'),
(3,2,'NINAO','37765289829','347055990','56214658','09:00','18:00'),
(4,3,'BOZO','01234567890','098765432','15973085','07:00','22:00');

INSERT INTO `TB_CLIENTE` (ID_CLIENTE,NOME_CLI,TELEFONE_CLI,CPF_CLI)
VALUES (1,'BRUCE WAYNE','11112233445','96796485831'),
(2,'TONY STARK','11223344556','42420780051'),
(3,'BRUCE BANNER','11334455667','34705599000'),
(4,'CLARK KENT','11445566778','42147516805'),
(5,'PETER PARKER','11556677889','37294284890'),
(6,'MARTY MCFLY','11667788990','26164345120'),
(7,'CHEWBACCA','11778899001','86329714339');

INSERT INTO `TB_AGENDAMENTO`
(ID_AGENDAMENTO,ID_FUNC,ID_CLIENTE,ID_STATUS,ID_SERVICO,DATA_AGENDAMENTO,HORA_INICIO_AGEND,HORA_FIM_AGEND)
VALUES (1,1,1,1,2,'04/06/2016','13:00','14:00'),
(2,1,5,3,1,'04/06/2016','14:00','15:00'),
(3,2,2,1,2,'04/06/2016','08:00','09:00'),
(4,3,3,2,1,'04/06/2016','12:00','13:00'),
(5,2,6,4,1,'04/06/2016','08:30','09:00'),
(6,4,6,1,2,'04/06/2016','08:30','09:00'),
(7,4,7,1,1,'04/06/2016','13:00','20:00');

COMMIT;
```