

Trabalho Final - Engenharia de Software

Professora: Rebeca Schroeder Freitas

Alunos: Mateus Roberto da Silva Correia, Marco Antonio Reche Rigon

Descrição do problema

O sistema de ERP é voltado para a gestão de um consultório de dentista. O objetivo do sistema é gerenciar as consultas que serão realizadas no consultório, e o estoque do consultório. Uma consulta deve ser realizada em pacientes cadastrados, ela deve ter sido agendada previamente. O sistema também deverá notificar um lembrete ao paciente 24 horas antes do horário marcado para a consulta. Deve ser possível consultar o histórico do paciente, como consultas realizadas e outros dados médicos. O gerenciamento de estoque serve para relacionar quais fornecedores servem quais produtos, a fim de facilitar a compra dos mesmos. O sistema de estoque não irá gerenciar quantos produtos estão em estoque, por causa da natureza descartável de boa parte dos materiais, e a dificuldade de anotação de quantos foram utilizados por consulta.

Requisitos do software

Requisito Funcional				
Nome	F1 Compra de produtos			
Descrição:	A função irá exibir informações sobre os fornecedores do estoque			
Requisitos Não funcionais				
Nome	Descrição	Categoria	Desejável	Permanente
N.F. 1.1. Consulta	A consulta ao estoque irá exibir dados dos produtos	Especificação	(x)	(x)
N.F. 1.2. Acesso rápido ao fornecedor	Clicar no nome do fornecedor levará aos dados do mesmo.	Interface	(x)	()
N.F. 1.3. Montagem de lista de compras	O usuário poderá montar um carrinho de compras.	Especificação	(x)	(x)
N.F. 1.4.	Ao finalizar o carrinho, exportar a lista como um arquivo PDF próprio para impressão	Usabilidade	()	(x)

Requisito Funcional				
Nome		F2 Agendar consultas		
Descrição:		A função irá agendar consultas no consultório		
Requisitos Não funcionais				
Nome	Descrição	Categoria	Desejável	Permanente
N.F. 2.1. Tempo de Registro	O tempo para registro do agendamento deverá ser inferior a três segundos.	Performance	(x)	()
N.F. 2.2. Cadastrar Paciente	A interface deve ter um botão de acesso rápido para cadastrar um novo paciente.	Interface	()	(x)
N.F. 2.3 Choque de horário	Caso a sala ou dentista já esteja realizando uma consulta nesse horário, impede o agendamento da consulta.	Especificação	()	(x)
N.F. 2.4 Calendário	Deve ter um botão para escolher a data do agendamento no calendário.	Interface	(x)	(x)
N.F. 2.5 Acesso ao cadastro	Com exceção de recepcionistas, um funcionário só poderá agendar uma consulta que ele mesmo realizará.	Segurança	()	(x)

Requisito Funcional				
Nome		F3 Consulta de agenda		
Descrição:		A função irá exibir todas as consultas agendadas		
Requisitos Não funcionais				
Nome	Descrição	Categoria	Desejável	Permanente
NF. 3.1 Mostrar Calendário	As consultas agendadas devem ser mostradas em um calendário.	Interface	(x)	(x)
N.F. 3.2 Filtrar	Deve ser possível filtrar as consultas por paciente, funcionário ou sala.	Usabilidade	(x)	()
N.F. 3.3 Controle de Acesso	Com exceção de recepcionistas, os funcionários só podem ver as consultas marcadas para eles mesmos.	Segurança	()	(x)
N.F. 3.4 Acesso a informações	O funcionário poderá selecionar a consulta e ver os dados do cliente e o que será realizado numa consulta.	Interface	()	()

Requisito Funcional	
Nome	
F4 Consulta de paciente	
Descrição:	
A função irá exibir a ficha médica dos pacientes	

Requisitos Não funcionais				
Nome	Descrição	Categoria	Desejável	Permanente
NF. 4.1 Ocultar Informações	Dados sensíveis, como CPF, serão ocultados para usuários que não sejam administradores.	Segurança	(x)	(x)
N.F. 4.2 Acesso a informações	Funcionários não administradores só podem acessar dados dos pacientes que já tiveram ou têm consultas agendadas com o mesmo.	Segurança	(x)	()
N.F. 4.3 Histórico	Deverá ser exibido o histórico de consultas do paciente.	Usabilidade	(x)	(x)
N.F. 4.4 Exportar ficha	Será possível exportar a ficha do paciente para um arquivo PDF, num formato próprio para impressão	Usabilidade	(x)	(x)
N.F. 4.5. Identificação do Paciente	O paciente deverá ser localizado pelo seu nome.	Interface	(x)	()
N.F. 4.6. Contato rápido	O usuário poderá clicar em um botão para entrar em contato com o paciente	Usabilidade	()	(x)

Requisito Funcional				
Nome		F5 Consulta de Funcionário		
Descrição:		A função irá exibir a ficha médica dos pacientes		
Requisitos Não funcionais				
Nome	Descrição	Categoria	Desejável	Permanente
N.F. 5.1 Ocultar Informações	Apenas usuários administradores podem acessar a ficha dos funcionários	Segurança	(x)	(x)
N.F. 5.2. Contato rápido	O usuário poderá clicar em um botão para entrar em contato com o funcionário	Usabilidade	()	(x)

Estimativa de duração do projeto completo

A estimativa de duração usará o método COCOMO. Para usarmos o COCOMO, precisamos da quantidade de linhas de código em KLOC, para isso usaremos o APF. Primeiro iremos separar os elementos do software:

- Número de Arquivos Lógicos Internos (ALI): 7 (Pacientes, Funcionários, Consultas, Salas, Estoque, Fornecedores, Compras)

- Número de Entradas Externas (EE): 6 (Essencialmente, o cadastro de cada tabela do banco de dados, exceto compras)
- Número de Saídas Externas (SE): 1 (Exportar ficha dos pacientes)
- Número de Consultas Externas (CE): 1 (Exportar lista de compras montada)
- Número de Arquivos de Interface Externos (AIE): 0

Ou seja:

- EE: 6
- SE: 1
- CE: 1
- ALI: 7
- AIE: 0

Agora, usaremos a seguinte tabela para definir a complexidade dos elementos.

		Campos de Dados		
		1 a 19	20 a 50	51 ou mais
Entidades Agrupadas	1	Baixa	Baixa	Média
	2 a 5	Baixa	Média	Alta
	6 ou mais	Média	Alta	Alta

Como as tabelas para cada função de dados terão menos de 20 colunas, e uma entidade não terá mais do que 5 relacionamentos, a complexidade do sistema é baixa. A partir disso, definimos o peso de cada elemento:

Elemento\Complexidade	Baixa	Média	Alta
Entradas Externas (EE)	3	4	6
Saídas Externas (SE)	4	5	7
Consultas Externas (CE)	3	4	6
Arquivos Lógicos Internos (ALI)	7	10	15
Arquivos de Interface Externos (AIE)	5	7	10

Pesos:

- EE: 3
- SE: 4
- CE: 3
- ALI: 7
- AIE: 5

Usando os dados acima, vamos definir os Pontos de Função Não Ajustados (PFNA), sendo ele o somatório de quantidade de cada elemento vezes o seu peso:

$$PFNA = 6*3 + 1*4 + 1*3 + 7*7 + 0*5 = 74$$

Para chegarmos na quantidade de linhas de código, multiplicamos pela complexidade da linguagem usada. O sistema será feito em Java, que tem complexidade 53:

$$LOC = 74 * 53 = 3922$$

$$KLOC = 3922 / 100 = 3,922$$

Voltando ao COCOMO. Usaremos a seguinte tabela para buscarmos os valores de duração, assumindo que é um projeto simples:

COCOMO Básico				
Complexidade	α	β	ϵ	θ
Simples	2,4	1,05	2,5	0,38
Média	3,0	1,12	2,5	0,35
Alta	3,6	1,20	2,5	0,32

$$\text{Esforço (E)} = \alpha \cdot \text{KLOC}^\beta$$

$$\text{Tempo (T)} = \epsilon \cdot E^\theta$$

KLOC = 3,922

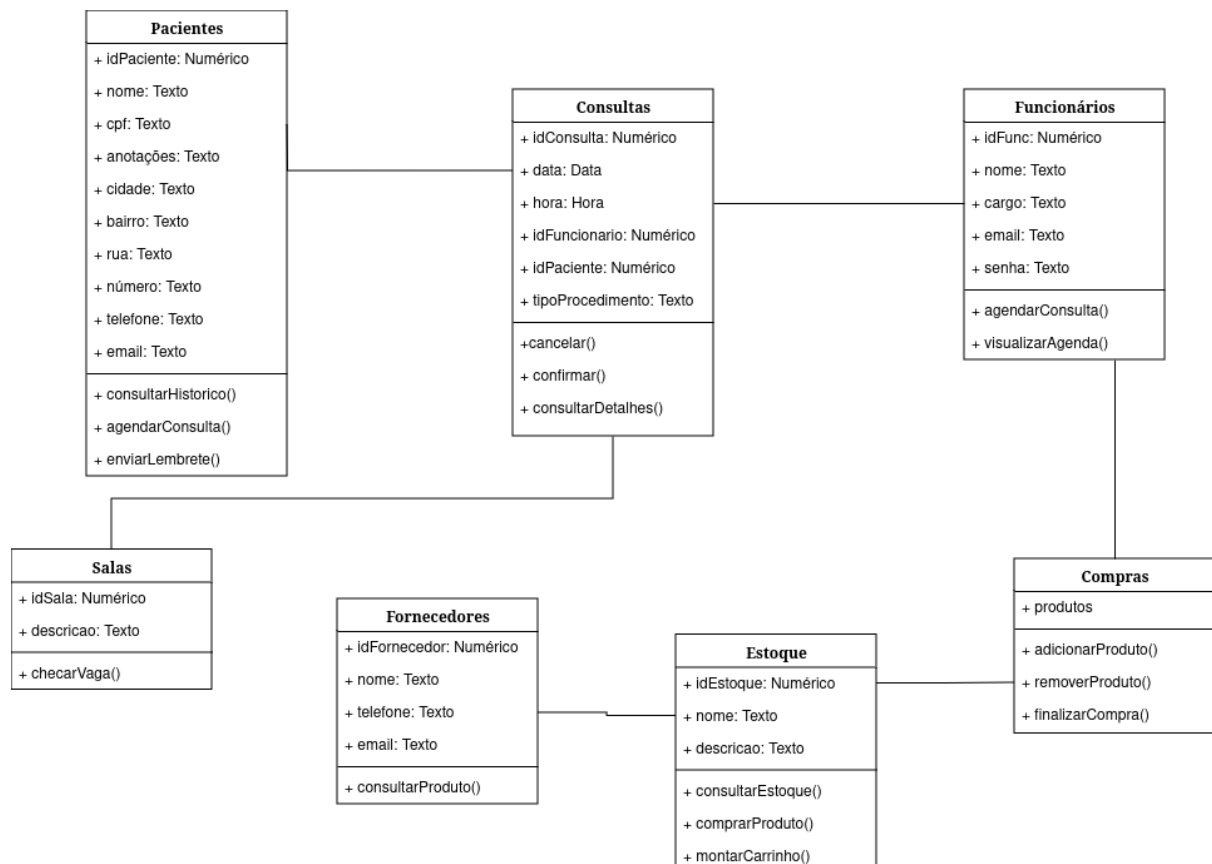
Complexidade: Simples

$$E = 2,4 \cdot 3,922^{1,05} = 10,07$$

$$T = 2,5 \cdot 10,07^{0,38} = 6,01.$$

Assim, sabemos que o projeto levará um pouco mais de 6 meses para ser concluído.

Diagrama de Classes do Projeto UML



Testes Unitários

[Tentar 3 testes]