

UNIVERSIDADE DE PASSO FUNDO - UPF	
Projeto Ambientche	Data: 23/08/2018
Documento de Análise e Projeto do Produto	Versão: 1.0

Projeto Ambientche

Documento de Análise e Projeto do Produto

Ciência da Computação - UPF

Disciplina de Análise e Projeto de Software

Prof.: Jeangrei Veiga / José Maciel

Aluno(s): Ricardo M. Fachinello, Tiago L. B. Seben

Novembro /2019

Histórico de Revisões

[illegible]

Sumário

1	INTRODUÇÃO	4
1.1	<i>IDENTIFICAÇÃO DO DOMÍNIO (ESCOPO)</i>	4
1.2	<i>PRINCIPAIS STAKEHOLDERS</i>	4
2	REQUISITOS	4
2.1	<i>REQUISITOS FUNCIONAIS</i>	5
2.1.1	<i>RF01 – Alterar status da luz</i>	5
2.1.2	<i>RF02 – Status da luz</i>	5
2.1.3	<i>RF03 – Salvar o log de alterações no status da luz</i>	6
2.1.4	<i>RF04 – Consultar Log de alterações</i>	6
2.1.5	<i>RF05 – Configurar o sistema</i>	6
2.2	<i>REQUISITOS NÃO-FUNCIONAIS</i>	7
2.2.1	<i>Requisitos de Segurança</i>	7
2.2.2	<i>Requisitos de Interface</i>	7
2.2.3	<i>Requisitos Operacionais</i>	7
3	ESPECIFICAÇÃO	8
3.1	<i>DIAGRAMA DE CASOS DE USO GERAL (OBRIGATÓRIO)</i>	8
3.2	<i>ESPECIFICAÇÃO DO CASO DE USO</i>	9
3.2.1	<i>UC01 – Configurar cômodo</i>	9
3.2.2	<i>UC02 – Consultar log</i>	10
3.2.2	<i>UC03 – Gerenciar configuração do sistema</i>	10
3.2.2	<i>UC04 – Realizar troca de mensagens</i>	11
4	PROJETO	12
4.1	<i>DIAGRAMA DE ATIVIDADES GERAL</i>	13
4.2	<i>DIAGRAMA DE CLASSES</i>	13
4.3	<i>DIAGRAMA DE SEQUÊNCIA</i>	13
4.3.1	<i>DIAGRAMA DE SEQUÊNCIA – CONSULTAR LOG</i>	14
4.4	<i>DIAGRAMA DE COMPONENTES</i>	15
4.5	<i>DIAGRAMA DE MÁQUINA DE ESTADOS</i>	15
4.6	<i>DIAGRAMA DE IMPLANTAÇÃO</i>	16
4.7	<i>DIAGRAMA ER</i>	17

1 INTRODUÇÃO

1.1 Identificação do Domínio (escopo)

Acesso ao sistema:

Uma interface Web

Controle de luzes:

O usuário deve acessar o link.

O usuário pode selecionar um cômodo em uma planta baixa onde deseja acender ou apagar a luz.

A interface web deve enviar um parâmetro de acordo com o cômodo para o servidor.

O servidor deve receber o parâmetro e mandar para o arduino.

O arduino deve ligar ou desligar a luz do cômodo referente ao parâmetro.

A interface web deve mostrar se a luz foi apagada ou ligada.

Controle de log:

O usuário deve acessar o sistema de log.

O usuário deve inserir os dados de acesso ao banco de dados.

O sistema deve mostrar o log do uso da luzes.

1.2 Principais Stakeholders

Nome	Responsabilidade	Dados para contato
Usuário	Ligar ou desligar as luzes	654654653213
Técnico	Responsável por montar o setup	tecnico@gmail.com

2 REQUISITOS

"Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferecem e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações."

SOMMERVILLE, I. Engenharia de Software. 9ª. ed. São Paulo: Person Education do Brasil, 2011 P. 57.

Identificação dos Requisitos

Cada requisito será unicamente identificado no formato [tipoRequisito.numero]. Para requisitos funcionais, o código do tipo de requisito será RF, e para requisitos não funcionais, RNF. Um número será assinalado a cada requisito de forma incremental, na ordem que forem mencionados neste documento.

Prioridade dos Requisitos

Para estabelecer a prioridade do requisito, será utilizado uma escala que inicia com número [1] até [5]. Quanto menor o número, maior a prioridade do requisito.

Importância do Requisito

Para estabelecer a importância do requisito, serão utilizadas as opções:

Essencial - Realmente é fundamental para o sistema, sem o qual o sistema não pode ser dado como "completo", ou "apto para produção". São requisitos que se não são implementados impedem uma implantação ou a conclusão do sistema. São compulsórios, não sendo possível aplicar soluções de contorno ou paliativos para eles.

Importante - Deve ser parte do escopo, mas não bloqueia o sistema a entrar em produção. É como se o sistema ficasse com uma "pendência" de escopo – criando débito técnico – que será atendido em momento oportuno. Sem um requisito importante, o sistema poderá rodar, funcionar, ser utilizado. Pode ser simplesmente postergado para pós-implantação, ou ser atendido temporariamente por soluções de contorno ou paliativos.

Desejável - Não é indispensável para o sistema estar completo, para entrar em produção. Também não é algo que, mesmo postergado, deverá ser feito obrigatoriamente. Sem um requisito desejável o sistema deve funcionar de maneira satisfatória, atendendo completamente seu objetivo. Por ser algo que não precisa ser feito para que o sistema esteja completo, é a menor das prioridades, e deve ser postergado para, se possível ser viabilizado no futuro.

Disponível: < <http://www.ateomomento.com.br/priorizacao-de-requisitos/>>. Acesso ago. 2017.

2.1 Requisitos Funcionais

"Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o sistema realize em benefício dos usuários. Eles variam de acordo com o tipo de software em desenvolvimento, com usuários e com o tipo de sistema que está sendo desenvolvido. Requisitos funcionais podem ser expressos de diversas maneiras e, como já foi dito acima, em diferentes níveis de detalhamento. Os requisitos funcionais de usuários definem recursos específicos que devem ser fornecidos pelo sistema."

SOMMERVILLE, I. **Engenharia de Software**. 9ª. ed. São Paulo: Person Education do Brasil, 2011. p. 59.

2.1.1 RF01 – Alterar status da luz

Importância: [X] essencial [] importante [] desejável

Dependência do(s) requisito(s): []

Priorização: [1]

PROBLEMAS/NECESSIDADES IDENTIFICADAS

Necessidade de controlar a iluminação da casa a distância.

SOLUÇÃO

Uma interface web conectado a um servidor que se conecta a um arduino para controle das luzes.

RESTRIÇÕES / EXCEÇÕES

2.1.2 RF02 – Status da luz

Importância: [] essencial [X] importante [] desejável

Dependência do(s) requisito(s): [RF01]

Priorização: [1]

PROBLEMAS/NECESSIDADES IDENTIFICADAS

Necessidade do usuário saber o status atual da luz.

SOLUÇÃO

Mostrar na interface web o status da luz.

RESTRIÇÕES / EXCEÇÕES

2.1.3 RF03 – Salvar o log de alterações no status da luz

Importância: [X] essencial [] importante [] desejável

Dependência do(s) requisito(s): [RF01]

Priorização: [1]

PROBLEMAS/NECESSIDADES IDENTIFICADAS

Necessidade de manter as informações de alteração dos status da luz.

SOLUÇÃO

O servidor salva o status de alterações da luz em um banco de dados.
O banco possui os seguintes dados:

- Data da alteração.
- Status da luz.
- Cômodo da casa.

RESTRIÇÕES / EXCEÇÕES

-Todos os dados são obrigatórios.

2.1.4 RF04 – Consultar Log de alterações

Importância: [] essencial [X] importante [] desejável

Dependência do(s) requisito(s): [RF03]

Priorização: [1]

PROBLEMAS/NECESSIDADES IDENTIFICADAS

Necessidade de consultar as alterações feitas no status da luz

SOLUÇÃO

O usuário ou o técnico pode consultar os status da luz, o sistema busca no banco de dados e retorna os seguintes dados:

- Data da alteração.
- Status da luz.
- Cômodo da casa.

RESTRIÇÕES / EXCEÇÕES

2.1.5 RF05 – Configurar o sistema

Importância: [X] essencial [] importante [] desejável

Dependência do(s) requisito(s): []

Priorização: [1]

PROBLEMAS/NECESSIDADES IDENTIFICADAS

Necessidade de configurar o sistema para o cliente

SOLUÇÃO

O técnico pode mapear as luzes de acordo com a planta baixa da casa do cliente.

RESTRIÇÕES / EXCEÇÕES

2.1.6 Requisitos Não-Funcionais

"Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo sistema. Podem estar relacionados a propriedades de sistemas emergentes, como confiabilidade, tempo de resposta, espaço em disco, desempenho e outros atributos de qualidade do produto. Às vezes podem dizer respeito ao sistema como um todo. Isso significa que na maioria das vezes eles são mais importantes que os requisitos funcionais individuais. Se uma falha em cumprir um requisito funcional pode comprometer parte do sistema, uma falha em cumprir um requisito não funcional pode tornar todo o sistema inútil".

SOMMERVILLE, I. **Engenharia de Software**. 9ª. ed. São Paulo: Person Education do Brasil, 2011. p. 60.

Os requisitos que descrevem os aspectos não-funcionais do sistema são apresentados a seguir:

2.1.7 Requisitos de Segurança

Ident.	Descrição
RNF/SEG-01	

2.1.8 Requisitos de Interface

Ident.	Descrição
RNF/INT-01	O sistema deve ter uma interface visual de fácil utilização.

2.1.9 Requisitos Operacionais

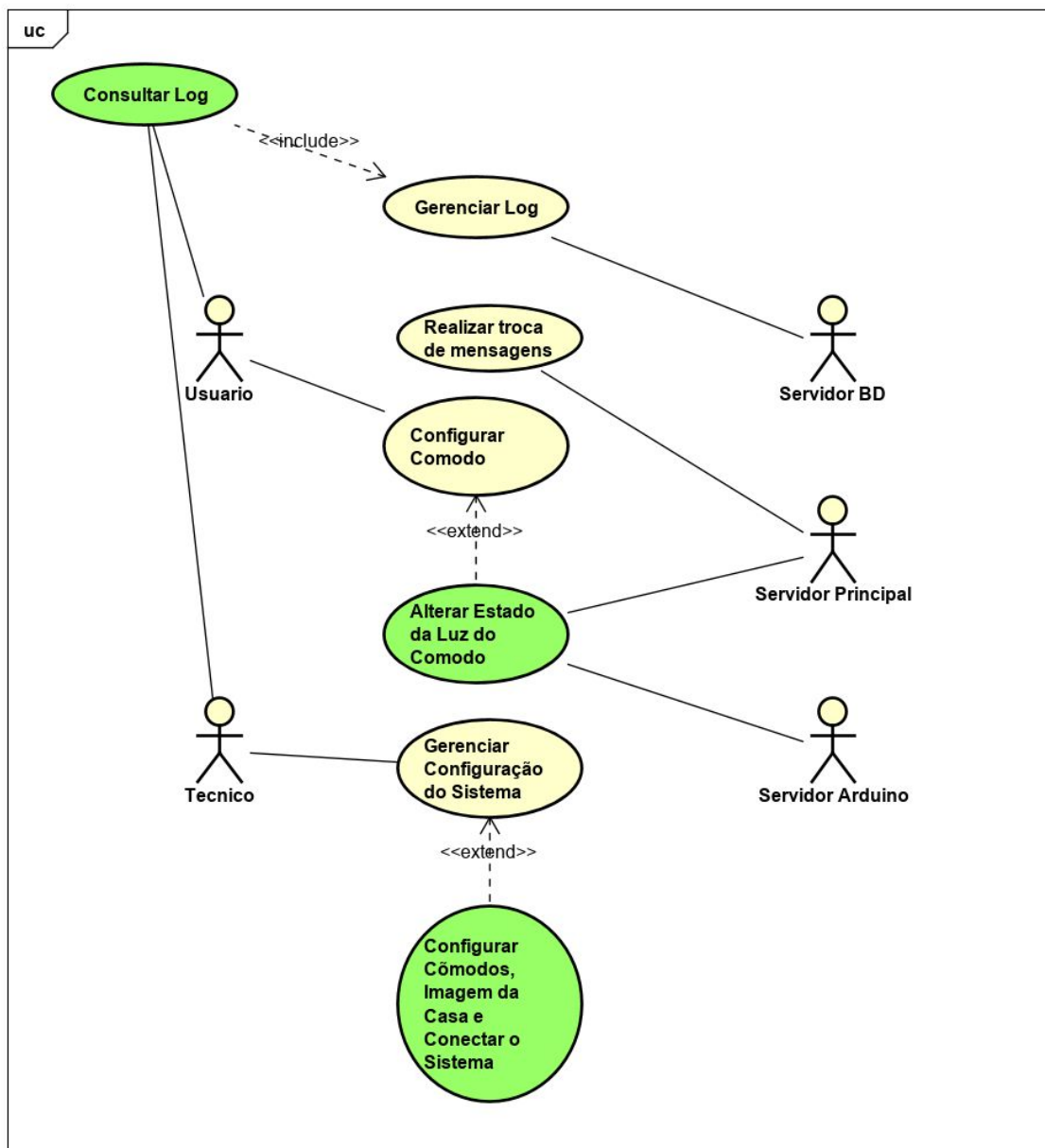
Ident.	Descrição
RNF/OPE-01	<p>A camada de aplicação para web compatível com os principais browsers de mercado (<i>Microsoft Edge, Google Chrome e Firefox</i>) e compatível com os sistemas operacionais Windows e Linux.</p> <p>O sistema deve verificar se a conexão foi estabelecida com sucesso.</p> <p>O sistema deve verificar se o código enviado para o servidor está correto.</p>

3 ESPECIFICAÇÃO

3.1 Diagrama de Casos de Uso Geral (Obrigatório)

O diagrama de casos de uso, expresso em UML (Unified Modeling Language), expressa os requisitos funcionais do sistema na forma de casos de uso. Segundo o RUP (Rational Unified Process), para cada requisito funcional tem-se um caso de uso. A descrição textual detalhada dos requisitos funcionais, seus fluxos de atividades e requisitos não funcionais associados pode ser encontrada na próxima seção.

Na figura abaixo mostramos a representação gráfica em UML dos casos de uso do sistema.



3.2 Especificação do caso de uso

"Uma especificação de caso de uso é um documento funcional, ou seja, ele descreve as funcionalidades que um sistema deve ter. No caso de uso é apresentado como o sistema deve reagir ao conjunto de entradas que recebe qual o comportamento desse sistema e as saídas que serão geradas."

Disponível: < <http://www.devmedia.com.br/elementos-da-especificacao-de-casos-de-uso/34391>>. Acesso 02 ago. 2017.

Dentre os casos de uso do sistema mostrados no diagrama de casos de uso, foram escolhidos cinco para serem detalhados e trabalhados nas fases de análise e projeto do sistema.

3.2.1 UC01 – Configurar cômodo

Descrição:	Este caso de uso descreve a ação para o usuário configurar o cômodo		
Atores:	Usuário		
Prioridade:	[1]		
Requisito associado:	RF01		
Entradas e pré-condições:	● Acesso ao sistema.		
Saídas e pós-condições:	● Status da luz atualizado.		
Fluxos de eventos			
Fluxo principal:	<div>1. O usuário acessa a interface web.</div> <div>2. A interface exibe a planta para o usuário selecionar qual cômodo configurar. [FA01] [FA02]</div>		
Fluxo alternativo:	<div>FA01 – Ligar Luz</div> <div>1. O usuário clica no cômodo que deseja ligar a luz.</div> <div>2. O sistema envia uma mensagem para o servidor principal.</div> <div>3. O servidor principal envia uma mensagem para o servidor do arduino e para o servidor do banco de dados.</div> <div>4. O arduino liga a luz do cômodo.</div> <div>5. A interface web mostra o estado atual do cômodo alterado.</div> <div>6. O sistema salva o log no banco de dados.</div> <div>7. O banco de dados replica o log para outro banco de dados.</div> <div>FA02 – Desligar a luz</div> <div>1. O usuário clica no cômodo que deseja desligar a luz.</div> <div>2. O sistema envia uma mensagem para o servidor principal.</div> <div>3. O servidor principal envia uma mensagem para o servidor do arduino e para o servidor do banco de dados.</div> <div>4. O arduino desliga a luz do cômodo.</div> <div>5. A interface web mostra o estado atual do cômodo alterado.</div> <div>6. O sistema salva o log no banco de dados.</div> <div>7. O banco de dados replica o log para outro banco de dados.</div>		

Fluxo de exceção:	FE01 – Comando incorreto O sistema verifica se o comando feito na interface web está correto, caso contrário, o sistema exibe um erro interno no servidor.
Esboço de tela:	
Cenários:	CEN01 – Usuário consegue configurar o cômodo. O usuário clica na interface web e o estado da luz do cômodo é alterado com sucesso. CEN02 – Usuário não consegue configurar o cômodo. O usuário clica na interface web e o estado da luz do cômodo não é alterado com sucesso.
Regras de Negócio:	

3.2.2 UC02 – Consultar log

Descrição:	Este caso de uso descreve a ação para consultar o log do banco de dados.		
Atores:	Usuário, Técnico, Servidor BD		
Prioridade:	[1]		
Requisito associado:	RF04		
Entradas e pré-condições:	● Banco de dados conectado		
Saídas e pós-condições:	● Log do uso do sistema		
Fluxos de eventos			
Fluxo principal:	<div>1. O usuário informa os dados para conectar ao banco de dados.</div> <div>2. O sistema solicita ao banco de dados os dados da tabela log.</div> <div>3. O sistema mostra na tela os logs do banco de dados.</div>		
Fluxo alternativo:	<div>FA01 – O banco de dados não possui dados de log</div> <div>1. O sistema não mostra nada ao usuário.</div>		

3.2.3 UC03 – Gerenciar configuração do sistema

Descrição:	Este caso de uso descreve a ação para configurar o sistema.	
Atores:	Técnico	
Prioridade:	[1]	
Requisito associado:	RF05	
Entradas e pré-condições:	<ul style="list-style-type: none"> • Planta baixa da casa 	
Saídas e pós-condições:	<ul style="list-style-type: none"> • Mapeamento das luzes da casa 	

Fluxos de eventos	
Fluxo principal:	<ol style="list-style-type: none"> 1. O técnico insere a planta baixa da casa na interface web. 2. O técnico mapeia as luzes da casa no arduino.
Fluxo alternativo:	

3.2.4 UC04 – Realizar troca de mensagens

Descrição:	Este caso de uso descreve a ação para troca de mensagens do servidor.	
Atores:	Servidor principal	
Prioridade:	[1]	
Requisito associado:	RF01	
Entradas e pré-condições:	<ul style="list-style-type: none">● Mensagem da interface	
Saídas e pós-condições:	<ul style="list-style-type: none">● Mensagem para os outros servidores	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O usuário seleciona um cômodo na interface web.2. A interface web envia uma mensagem para o servidor principal.3. O servidor principal envia a mensagem para o servidor BD.4. O servidor principal envia a mensagem para o servidor Arduino.	
Fluxo alternativo:		

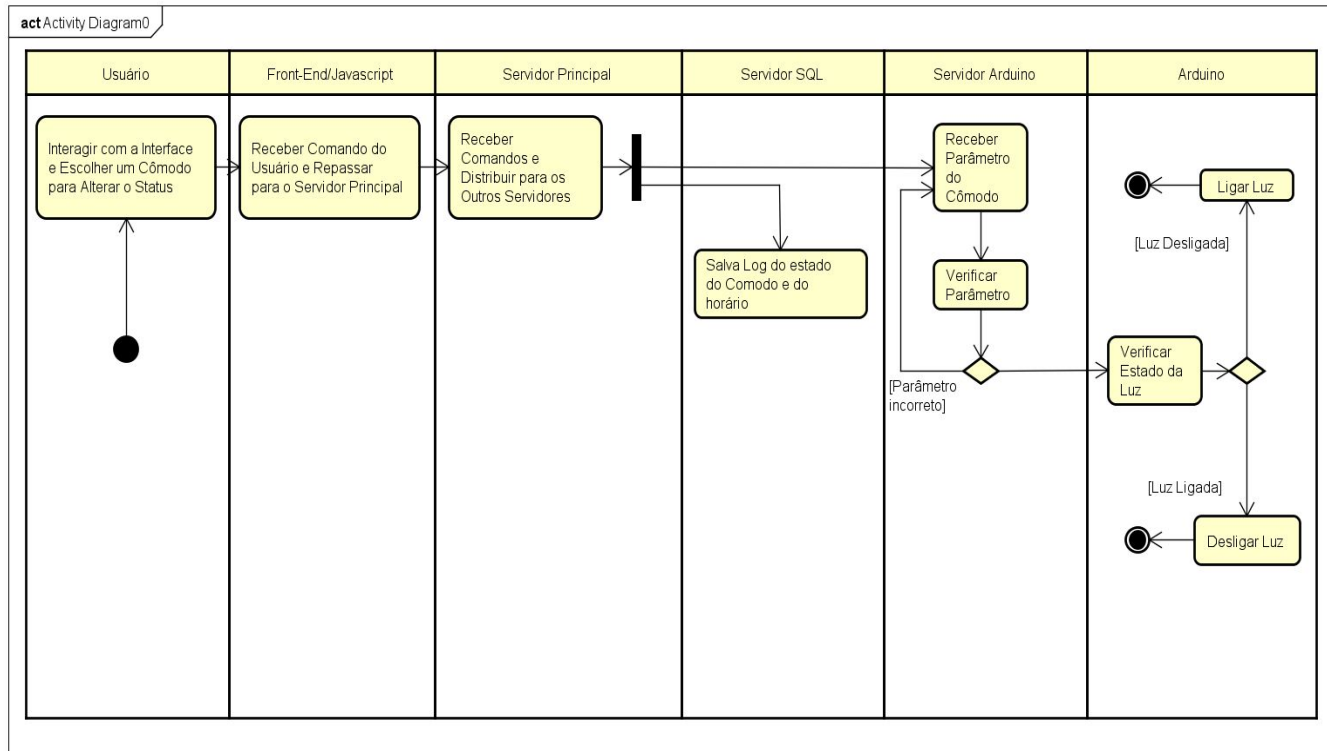
4 PROJETO

4.1 Diagrama de Atividades Geral

"O diagrama de atividade é um diagrama com maior ênfase ao nível de algoritmo da UML e provavelmente um dos mais detalhistas...

Esse diagrama é utilizado, como o próprio nome diz, para modelar atividades, que podem ser um método ou um algoritmo, ou mesmo um processo completo. Atividades podem descrever computação procedural; neste contexto elas são os métodos correspondentes as operações sobre classes."

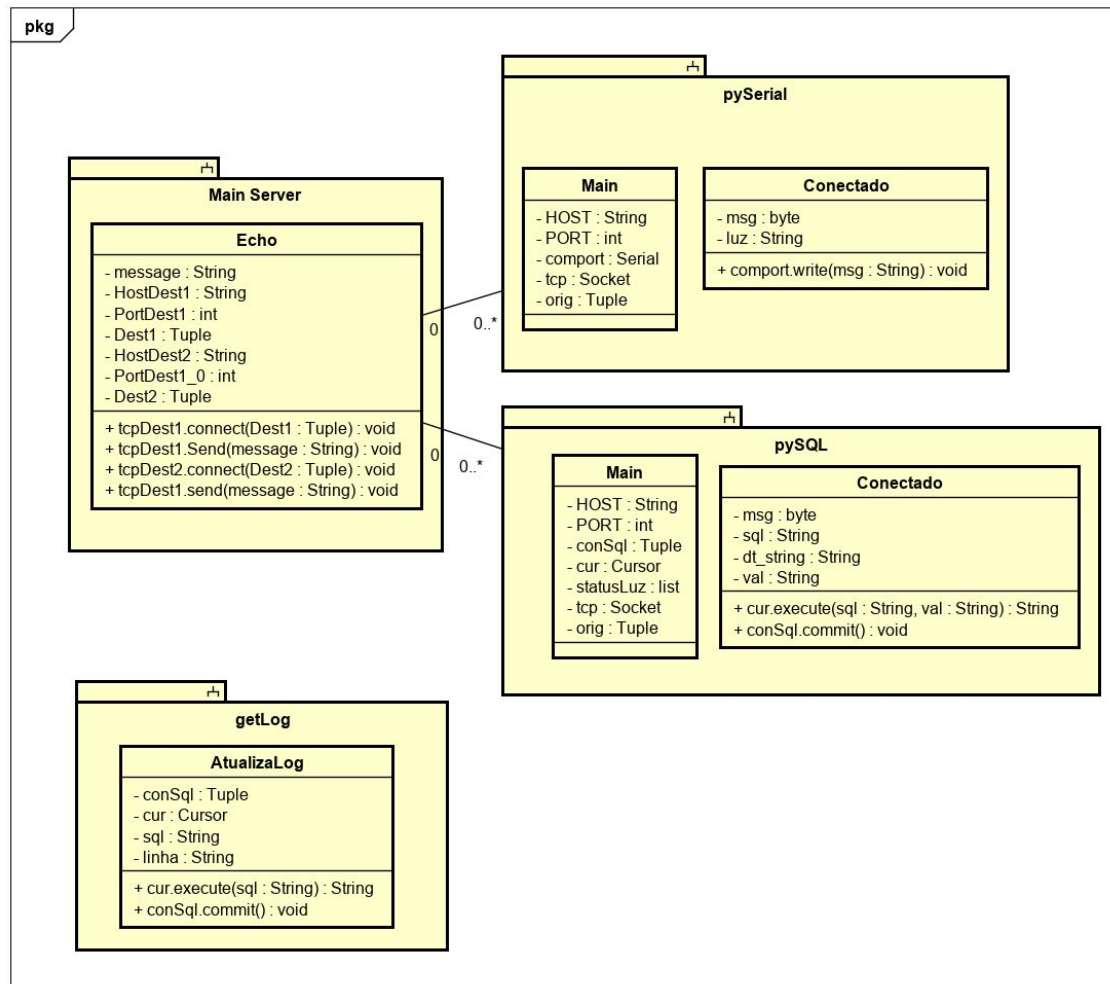
Guedes, Gilleanes T.A., **UML 2 Guia Prático**. São Paulo: Novatec Editora, 2007. p. 133.



4.2 Diagrama de classes

"O diagrama de classes demonstra a estrutura estática das classes de um sistema onde estas representam as "coisas" que são gerenciadas pela aplicação modelada. O diagrama de classes é considerado estático já que a estrutura descrita é sempre válida em qualquer ponto do ciclo de vida do sistema".

A figura abaixo demonstra a representação gráfica em UML das classes do sistema.

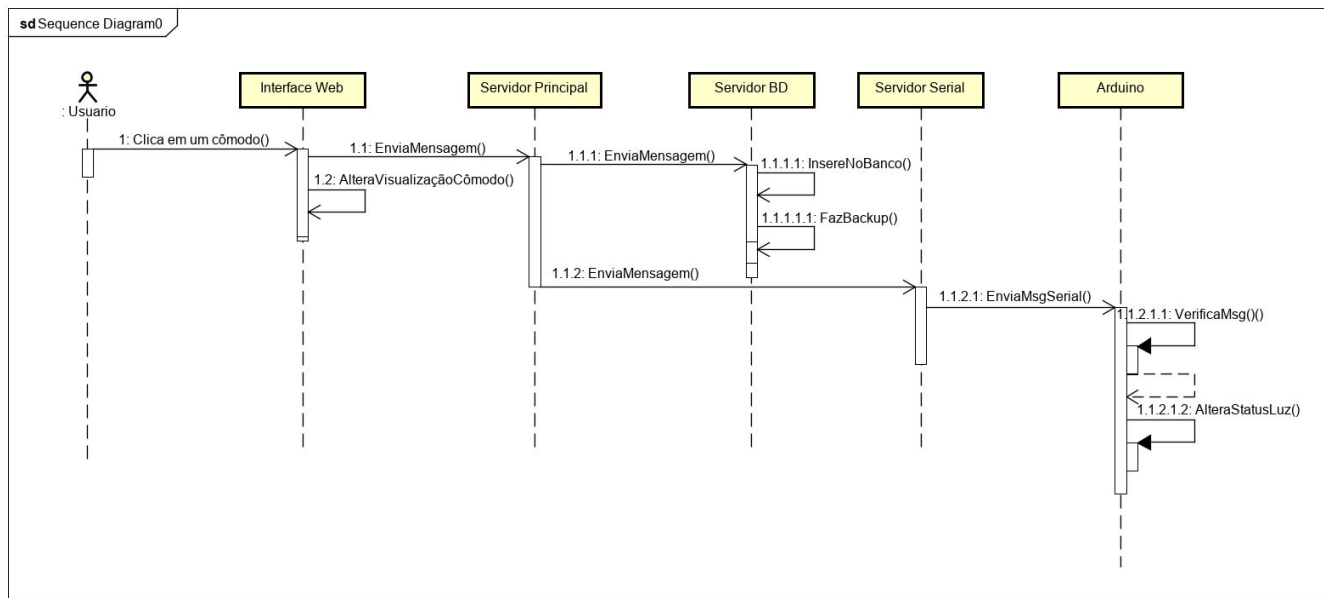


4.3 Diagrama de Sequência

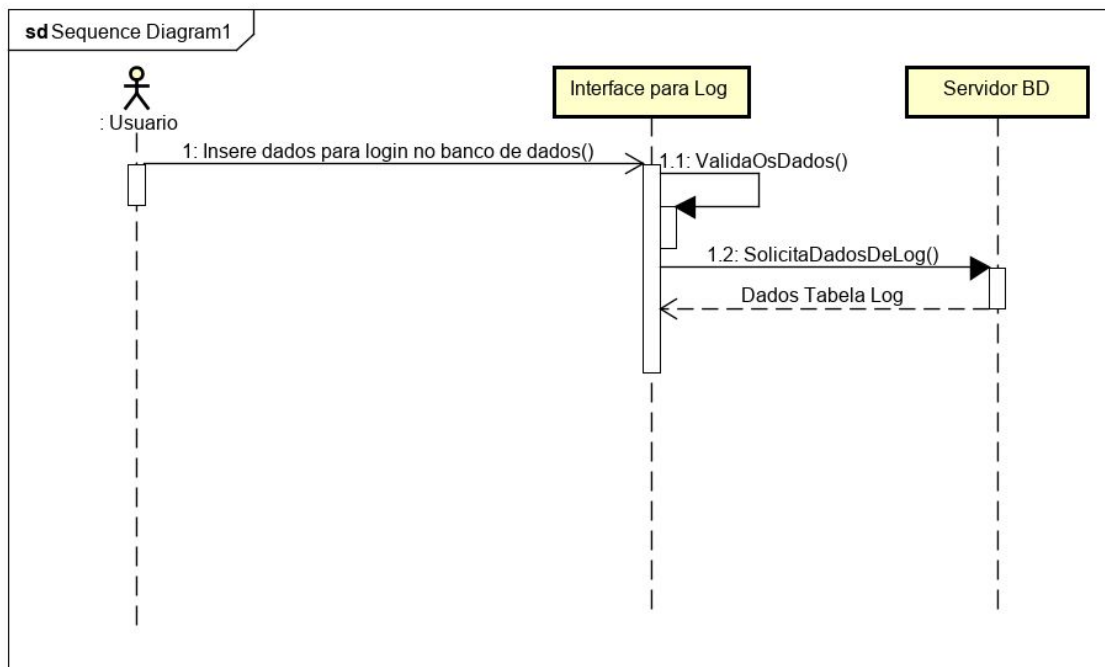
"Este diagrama procura determinar a sequência de eventos que ocorre em um determinado processo, identificando quais métodos devem ser disparados entre os atores e objetos envolvidos e em que ordem. O diagrama de sequência baseia-se no diagrama de casos de uso, havendo normalmente um diagrama de sequência para cada diagrama de caso de uso, uma vez que um caso de uso, em geral, refere-se a um processo disparado pelo autor."

Guedes, Gilleanes T.A., **UML 2 Guia Prático**. São Paulo: NOvatec Editora, 2007. p. 84.

Na figura abaixo demonstra a representação gráfica do diagrama de sequência do principal caso de uso.



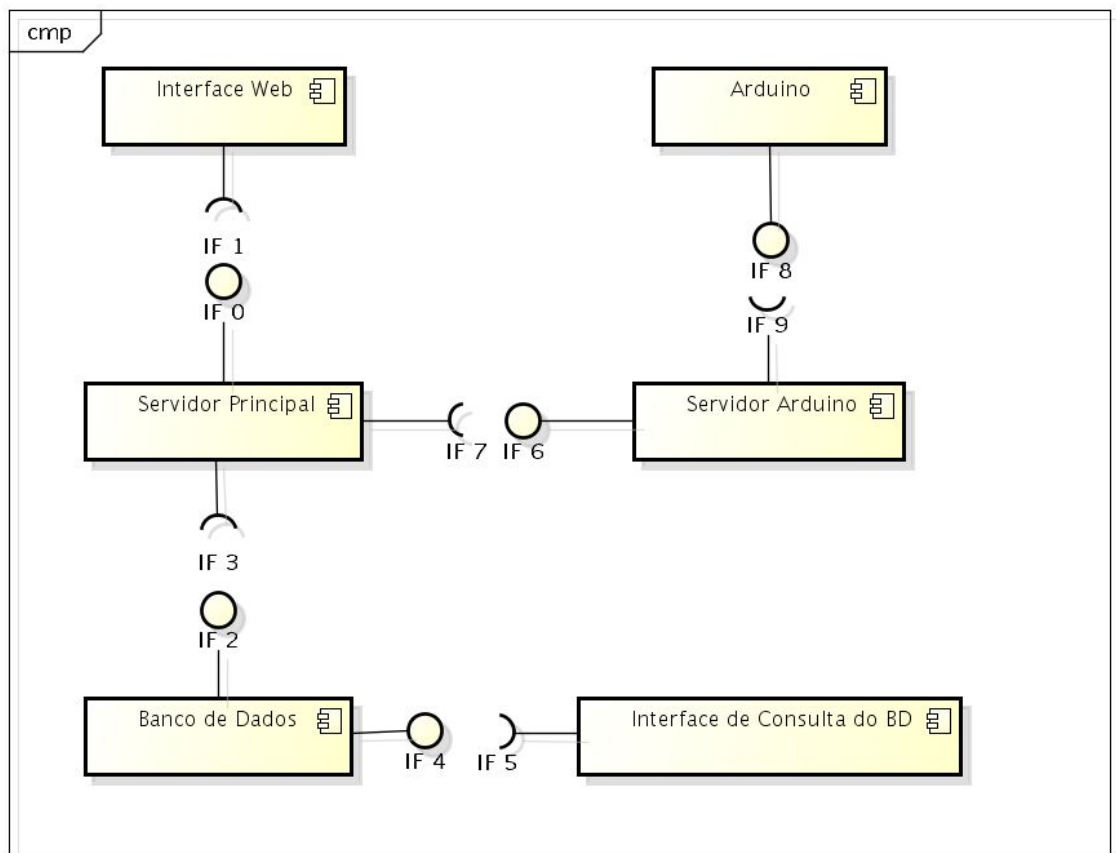
4.3.1 Diagrama de Sequência – Consultar log



4.4 Diagrama de Componentes

"O Diagrama de Componentes está amplamente associado à linguagem de programação que será utilizada para desenvolver o sistema modelado. Este diagrama representa componentes do sistema quando este for implementado em termos de módulos de código-fonte, bibliotecas, formulários, arquivos de ajuda, módulos executáveis, etc. e determina como os componentes estarão estruturados e interagirão para que o sistema funcione de maneira adequada. O Diagrama de Componentes pode ser utilizado para modelar o código-fonte, os módulos executáveis de um sistema, a estrutura física de um banco de dados ou mesmo os componentes necessários para a construção de interface."

Guedes, Gilleanes T.A., **UML 2 Guia Prático**. São Paulo: Novatec Editora, 2007. p. 24.

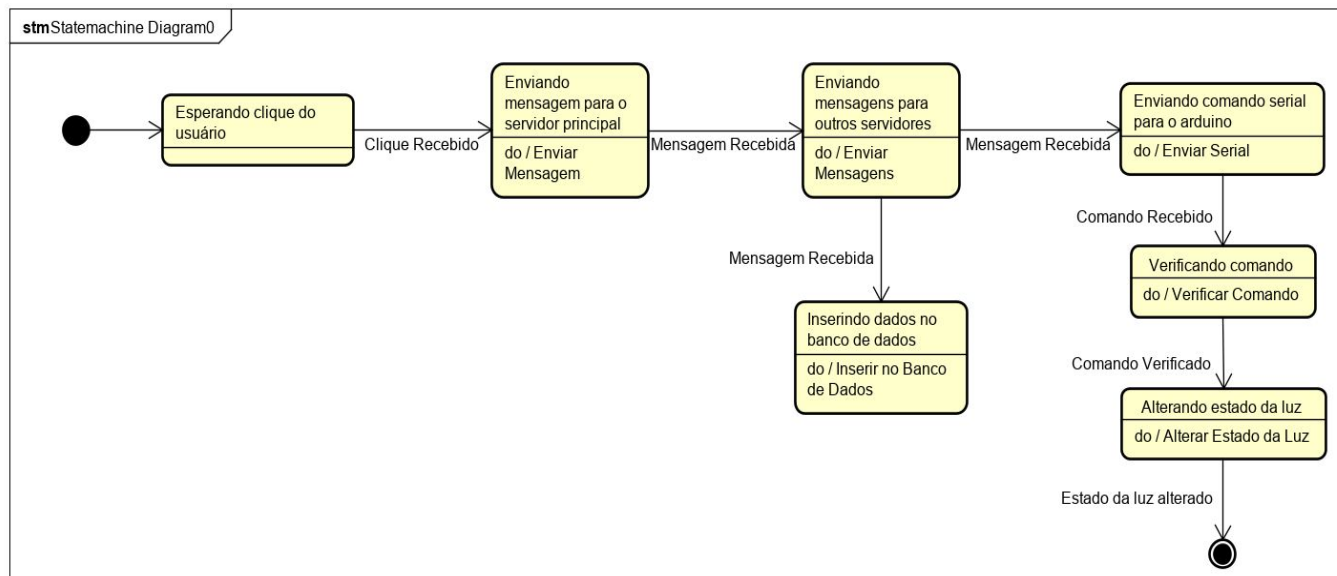


powered by Astah

4.5 Diagrama de Máquina de Estados

"Esse diagrama procura acompanhar as mudanças sofridas nos estados de uma instância de uma classe, de um Caso de Uso ou mesmo de um subsistema completo. Com o Diagrama de Sequência, o Diagrama de Máquina de Estados muitas vezes se baseia em um Caso de Uso e se apoia no Diagrama de Classes."

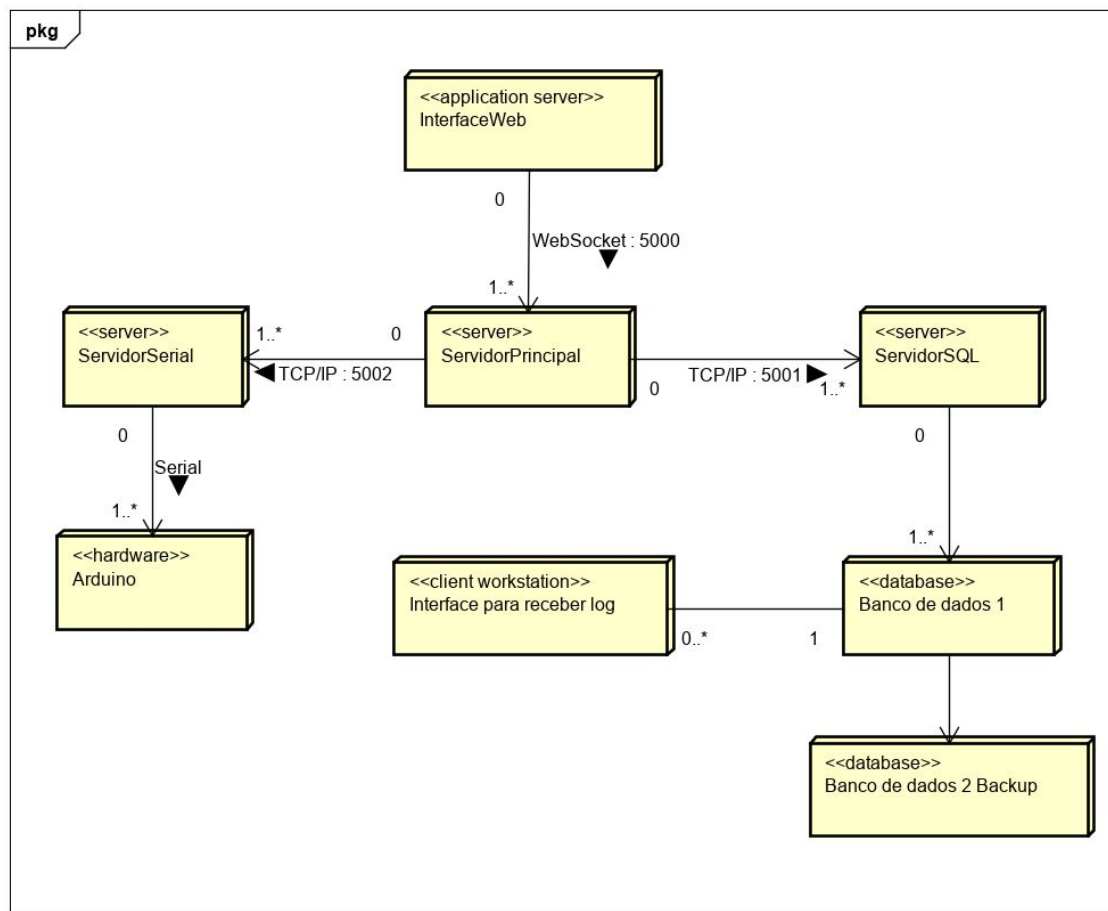
Guedes, Gilleanes T.A., **UML 2 Guia Prático**. São Paulo: Novatec Editora, 2007. p. 22.



4.6 Diagrama de Implantação

"Este diagrama determina as necessidades de hardware do sistema, as características físicas como servidor, estação, topologia e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado."

Guedes, Gilleanes T.A., **UML 2 Guia Prático**. São Paulo: Novatec Editora, 2007. p. 25.



4.7 Diagrama ER

