

# **WEBSERVER INTERATIVO**

***DL* softwares**



**Cliente: UTFPR  
Responsáveis no cliente: Laudelino  
Bastos**

**Projeto: WEBSERVER INTERATIVO  
Versão: 1.11**

**Curitiba  
2015**

## ***WEBSERVER* INTERATIVO**

**Responsáveis no cliente: Laudelino Bastos**

**Responsáveis pelo projeto e desenvolvimento:**

**Diego Wesley Braga  
Luís Fernando Guerreiro**

**Curitiba  
2015**

## Histórico de Modificações

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
13/05/2015	1.0	Primeira Entrega	Diego W. Braga. Luís F. Guerreiro
01/06/2015	1.1	Correção do professor	Diego W. Braga. Luís F. Guerreiro
24/06/2015	1.2	Versão Final	Diego W. Braga. Luís F. Guerreiro

# SUMÁRIO

SUMÁRIO .....	4
1 Lista de abreviaturas, siglas e termos em língua estrangeira .....	6
2 Introdução.....	7
2.1 Objetivo Geral.....	7
2.2 Objetivos Específicos.....	7
2.3 Conteúdo do Plano do Projeto.....	7
3 Levantamento de requisitos.....	9
3.1 Questões Organizacionais.....	9
3.2 Questões Econômicas .....	10
3.3 Questões Técnicas .....	11
3.4 Questões Operacionais .....	13
3.5 Requisitos Funcionais, Não Funcionais e Restrições de Projeto ....	14
4 Estudo de viabilidade.....	16
4.1 Viabilidade Organizacional .....	16
4.2 Viabilidade Econômica.....	16
4.3 Viabilidade Técnica.....	17
4.4 Viabilidade Operacional .....	17
4.5 Recursos a serem utilizados.....	18
5 Diagrama de casos de uso .....	20
5.1.1 Caso de uso: Requisição HTTP .....	21
5.1.2 Caso de uso: Mostrar informações dos dados transmitidos.....	22
5.1.3 Caso de uso: Fechar conexão/ Salvar dados.....	23
5.1.4 Caso de uso: Consultar banco de dados .....	24
5.1.5 Caso de uso: Iniciar o servidor .....	25
5.1.6 Caso de uso: Reiniciar o servidor.....	26
6 Diagrama de classes .....	29
6.1 Dicionário de informações .....	29
6.2 Diagrama de Objetos (Instâncias). .....	34
7 Diagramas de Sequência.....	35
7.1 UC01: Realizar Conexão .....	36
7.2 UC02: Mostrar informações dos dados transmitidos .....	37
7.3 UC03: Salvar dados da conexão .....	38
7.4 UC04: Consultar banco de dados.....	39
7.5 UC05: Iniciar Servidor .....	40
7.6 UC06: Reiniciar Servidor .....	41
8 Diagramas de comunicação .....	42
8.1 UC01: Realizar conexão.....	42

8.2	UC02: Mostrar informações dos dados transmitidos .....	43
8.3	UC03: Salvar dados da conexão .....	44
8.4	UC04: Consultar banco de dados.....	45
8.5	UC05: Inicar servidor .....	46
8.6	UC06: Reiniciar servidor.....	47
9	Mapeamento do diagrama de classes para Diagrama Entidade-Relacionamento .....	48
10	Diagrama de estados .....	49
11	Diagrama de atividades .....	61
12	Conclusões .....	73
13	Referências .....	74
14	Apêndices: .....	75
14.1	Diagrama de classes.....	75
14.2	Diagrama de objetos .....	76

# 1 LISTA DE ABREVIATURAS, SIGLAS E TERMOS EM LÍNGUA ESTRANGEIRA

**HTML:** Hyper Text Mark up Language

**HTTP:** Hyper Text Transfer Protocol

**FAQs:** Frequently asked questions

**SGBD:** Sistema gerenciador de banco de dados

**Software:** Programa de computador

**Hardware:** Partes físicas de um computador

**Web:** sinônimo de rede, pode ser usado como rede de computadores

**Browser:** navegador, usado para navegadores de Internet

**Parser:** decodificador ou analisador sintático

**Debug:** Ato de analisar o programa ou sistema em busca de erros.

## 2 INTRODUÇÃO

Este projeto tem por finalidade desenvolver um *servidor web*, inicialmente para hospedar páginas HTML. O *software* deve, além de desempenhar a função de servidor, detalhar o processo de funcionamento deste de tal forma que fique transparente para o usuário. O motivo desse detalhamento é a utilização do programa como ferramenta didática ou para *debug* de redes de computadores. Essa é uma boa escolha para este tipo de projeto, pois o modelo de *software* de um servidor já é um caso bem resolvido na comunidade de computação e além disso é uma estrutura usada diariamente na vida moderna. Portanto é um tema relevante e adequado para aplicar os conhecimentos de análise e projeto de sistemas.

### 2.1 OBJETIVO GERAL

Desenvolver um servidor para páginas web capaz de funcionar de forma similar aos sistemas comerciais utilizados na prática. O sistema desenvolvido também deve atuar como ferramenta complementar de ensino e aprendizagem na área de redes de computadores.

### 2.2 OBJETIVOS ESPECÍFICOS

Permitir ao usuário a visualização do funcionamento de um servidor, exibindo detalhes das conexões ocorridas;

Mostrar ao usuário indicadores de funcionamento do servidor;

Salvar informações das conexões realizadas e permitir ao usuário consultas ao banco de dados;

### 2.3 CONTEÚDO DO PLANO DO PROJETO

Este documento está dividido nas seguintes partes:

Capítulo 1: Lista de abreviaturas, siglas e termos em língua estrangeira

Capítulo 2: Introdução

Capítulo 3: Levantamento de requisitos

Capítulo 4: Estudo de viabilidade

Capítulo 5: Diagrama de Casos de Uso

Capítulo 6: Diagrama de Classes

Capítulo 7: Diagramas de Sequência

Capítulo 8: Diagramas de Comunicação

Capítulo 9: Mapeamento do Diagrama de Classes para Diagrama

Entidade-Relacionamento

Capítulo 10: Diagramas de Estados

Capítulo 11: Diagramas de Atividades

Capítulo 12: Conclusões

Capítulo 13: Referências

Capítulo 14: Apêndices



### **3 LEVANTAMENTO DE REQUISITOS**

Os requisitos do projeto foram definidos através das seguintes questões:

#### **3.1 QUESTÕES ORGANIZACIONAIS**

##### **1-Quais características do *software* são importantes para o ensino e aprendizagem de redes?**

A principal característica é o detalhamento do processo de uma requisição HTTP através de uma conexão TCP/IP. O *software* detalha os elementos do sistema operacional envolvidos no processo e o princípio básico que um *webserver* usa para decodificar uma requisição HTTP. Outra característica importante é o armazenamento do histórico de conexões com o servidor e a produção de informações personalizadas baseadas nesses dados, com isso o usuário pode estimar o comportamento de um *webserver* real.

##### **2-É preciso algum tipo de conhecimento prévio para utilizar o sistema?**

Para realmente entender o que está acontecendo o usuário precisa saber alguns conceitos de redes como conexões TCP, portas do sistema, endereço IP e uma noção de programação distribuída (comunicação interprocessos). Mas um usuário leigo no assunto pode usar o *software* como ferramenta de estudo em conjunto com um material teórico sobre redes.

##### **3- Quais casos o sistema consegue exemplificar/simular?**

O sistema funciona respondendo requisições HTTP através de conexões TCP/IP, ou seja, quando uma pessoa acessa um site através de um navegador e o servidor responde com uma página *web* e o seu respectivo conteúdo.

##### **4- Em que tipo de instituição o sistema pode ser utilizado?**

Escolas, universidades e qualquer outra instituição de ensino que ensina conceitos de redes de computadores ou computação.

### **5-Como o *software* será distribuído?**

O *software* será distribuído via *download*, disponibilizado em um repositório GitHub.

### **6-Será oferecido suporte técnico?**

A princípio não, pois o *software* não tem objetivo comercial. Pode ser fornecido algum tipo de manual ou ajuda geral como uma página de wiki ou FAQs.

### **7-Quantas pessoas são necessárias para operar o sistema?**

É necessário apenas um operador para iniciar o sistema e monitorar os possíveis erros.

## **3.2 QUESTÕES ECONÔMICAS**

### **1-Qual o custo (consumo de processamento e memória) por tempo de utilização?**

O custo de processamento e memória deverá ser proporcional ao número de conexões ativas no momento.

$$C_{total} = C_{conexão} \times N \times t$$

- $C_{conexão}$  = custo máximo de uma conexão;
- $N$  = número de conexões ativas;
- $t$  = tempo de utilização

### **2-Qual o custo por acesso ou por conexão do servidor?**

Pode-se estimar esse custo calculando a quantidade de memória/processamento um processo em Python consome do sistema. Além disso a conexão tem um valor variável de bytes transmitidos. Então o custo também depende do custo por utilização da rede.

$$C_{conexão} = C_{sistema} + C_{rede}$$

- $C_{sistema}$  = custo de memória e processamento para rodar o sistema;
- $C_{rede}$  = custo proporcional à utilização que o sistema faz da rede.

### **3-Existe um possível mercado futuro para esse projeto?**

Sim, o *software* contém o princípio básico de funcionamento de qualquer *webserver* comercial. É possível aumentar o projeto para ser usado profissionalmente no futuro.

### **4-Existe algum produto semelhante disponível?**

Sim. Apache, IIS, nginx, GWS.

### **5-Qual o tempo estimado para projetar e implementar o sistema?**

Considerando que o projeto será desenvolvido semanalmente, usando 4 horas de trabalho por semana, é estimado um período de 2 meses para sua conclusão.

### **6-Quantas pessoas serão necessárias para implementar/testar o sistema?**

Considerando o tempo planejado e o escopo do projeto apenas duas pessoas são suficientes.

### **7-Para esse tipo de sistema qual a ordem de importância dos itens: velocidade de processamento, processamento paralelo, espaço de memória RAM, velocidade de acesso a disco, infraestrutura de rede?**

Como o objetivo de um *webserver* é responder a várias requisições simultaneamente, o ideal é que ele consiga manter muitos clientes ao mesmo. Além disso a duração de uma transação (requisição de página e resposta do servidor) depende mais do tipo de página pedida e seu conteúdo do que das capacidades de *hardware* do servidor. Portanto a ordem de importância dos itens fica:

- 1° infraestrutura de rede;
- 2° processamento paralelo;
- 3° espaço de memória RAM;
- 4° velocidade de processamento;
- 5° velocidade de acesso a disco;

## **3.3 QUESTÕES TÉCNICAS**

**1-Qual o máximo de conexões simultâneas que o servidor pode suportar?**

Não existe um limite definido, pois pode-se dimensionar um servidor constituído de inúmeros computadores e uma grande infraestrutura de rede. Em média um servidor trabalha com um limite arbitrário entre 500 e 1000 conexões por endereço IP, porém esse limite depende de vários fatores, entre eles:

- Tipo da requisição HTTP;
- Sistema operacional do servidor;
- Quantidade de memória disponível;
- Utilização da rede por outros serviços.

**2-Que tipo de páginas *web* ou conteúdo o programa pode servir?**

Todos os tipos de página HTML e seu respectivo conteúdo (texto, imagem, etc)

**3-Quais os requisitos mínimos de *hardware* para o sistema rodar o *software*?**

Requisitos mínimos para suportar a linguagem Python e uma conexão com a internet.

**4-Em quais sistemas operacionais o programa funciona?**

A princípio qualquer sistema que suporta a linguagem Python

**5-Que tipo de acesso será usado para o SGBD?**

O acesso será feito indiretamente, ou seja, o usuário deve realizar consultas através da interface gráfica que por sua vez utiliza uma classe específica do sistema para acessar o banco de dados. O usuário poderá consultar uma tabela completa, parte dela ou uma consulta específica juntando partes de tabelas diferentes.

**6-Qual seria uma situação ótima para o funcionamento do sistema? E uma crítica?**

Situação ótima: apenas um cliente acessando uma página *web* que contenha apenas texto ou uma página que não esteja hospedada no servidor.

Situação crítica: o máximo de clientes suportados acessando páginas que contenham conteúdo pesado, como uma página *web* com muitas imagens ou *download* de arquivos grandes.

#### **7-O sistema pode ser operado remotamente?**

Sim, a princípio usando SSH ou VPN (Unix).

### **3.4 QUESTÕES OPERACIONAIS**

#### **1-Quais seriam o os usuários diretos do *software*?**

São as pessoas que hospedarão conteúdo,

#### **2-Quais as informações do funcionamento do sistema que o usuário tem acesso?**

Número de conexões ativas, uso de memória, e todas as informações referentes a uma conexão (bytes transferidos, páginas acessadas, porta utilizada, endereço IP de origem, data e hora).

#### **3-Quais parâmetros do sistema ou variáveis o usuário definirá?**

Número máximo de conexões, porta do servidor.

#### **4-Quais os modos de operação ou número de opções que o usuário pode escolher no sistema?**

Com ou sem suporte a arquivos(*Download*);

#### **5-Que tipo de tratamento de erros será usado?**

Será baseado no tratamento de exceções, se o erro estiver previsto nas exceções do sistema uma mensagem é apresentada ao usuário.

#### **6-Como serão apresentados ao usuário os dados guardados no SGBD?**

O usuário pode acessar os dados através da interface gráfica do sistema. Ele poderá consultar todas as tabelas salvas. Além disso o sistema vai gerar indicadores em tempo real.

**7-Quais seriam os usuários indiretos do *software* (clientes finais que farão as conexões com o servidor)?**

*Serão web browsers* ou qualquer outro programa de computador que implemente uma requisição HTTP. Geralmente esses programas são operados por pessoas acessando uma página web.

### **3.5 REQUISITOS FUNCIONAIS, NÃO FUNCIONAIS E RESTRIÇÕES DE PROJETO**

**Requisitos Funcionais:**

**RF01:** O usuário pode definir os parâmetros de funcionamento do servidor (portas, máximo de conexões, tipo de arquivos a hospedar, etc);

**RF02:** O usuário pode visualizar as informações recebidas e enviadas pelo servidor;

**RF03:** O usuário pode receber avisos sobre o funcionamento do sistema;

**RF04:** O usuário pode pesquisar todo ou um subconjunto do banco de dados;

**RF05:** O sistema gera automaticamente indicadores de uso baseados nas conexões ativas.

**Requisitos Não Funcionais:**

**RNF01:** Para possibilitar a execução do **RF01** o sistema apresenta uma janela inicial para o usuário preencher campos que vão definir suas respectivas variáveis no sistema. Caso o usuário não escolha os parâmetros o sistema começa com uma configuração padrão;

**RNF02:** Para possibilitar a execução do **RF02** o sistema toma as seguintes ações:

- Mostrar uma lista de todas as conexões feitas no sistema (várias portas diferentes);
- Detalhar cada conexão através das informações disponíveis e calculadas de cada uma (endereço IP de origem, porta do sistema utilizada, bytes enviados e recebidos, página acessada, etc);

**RNF03:** Para possibilitar a execução do **RF03** o sistema apresenta algumas informações em tempo real, baseadas nas variáveis que definem o “estado do sistema” (número de conexões ativas, uso de memória, tempo ocioso, etc);

**RNF04:** Para possibilitar a execução do **RF04** o sistema deve, a cada conexão feita, possibilitar o registro das informações obtidas como uma entrada no banco de dados. Associar esses dados a informações adicionais como data e hora para facilitar a análise do histórico do sistema futuramente.

**RNF05:** Para possibilitar a execução do **RF05** o sistema deve a partir dos dados guardados na tabela do SGBD gerar informações relacionadas ao uso do sistema ao longo do tempo (tempo médio de conexão, média de clientes usando o servidor, páginas mais acessadas, horário de pico, etc);

### **Restrições de Projeto**

**RP01:**Utilizar uma linguagem orientada a objetos;

**RP02:**Apresentar uma interface gráfica para o usuário;

**RP03:**Acessar um banco de dados referente ao tema do projeto;

**RP04:**Deve ser realizado no tempo de um semestre letivo;

## 4 ESTUDO DE VIABILIDADE

A análise da viabilidade do projeto foi dividida nas mesmas quatro áreas das perguntas utilizadas para levantar os requisitos.

### 4.1 VIABILIDADE ORGANIZACIONAL

O sistema poderá ser utilizado por qualquer pessoa que o queira, mas a princípio é planejado sua distribuição para professores e alunos de disciplinas de redes de computadores, em escolas técnicas e universidades, onde conhecimentos prévios serão importantes para o entendimento total das informações. O *software* será capaz de simular requisições HTTP através de conexões TCP/IP, detalhando todas características das conexões envolvidas, bem como os elementos do sistema utilizado enquanto o servidor é acessado. O sistema será portanto uma grande ferramenta de visualização de conceitos apresentados nas devidas disciplinas. Visando essa maior utilização acadêmica, a distribuição livre via *download* atende de forma ampla e fácil os interessados.

### 4.2 VIABILIDADE ECONÔMICA

Em relação a viabilidade econômica deste projeto pode-se concluir que seu desenvolvimento não acarretará grandes custos, levando em consideração os recursos e o número de pessoas necessários. No entanto, nesta versão, também não visa lucros, sendo distribuído gratuitamente para a utilização acadêmica. Os recursos necessários para utilização do *software* em boa performance são pouco exigentes, sendo viável o seu desenvolvimento em laboratórios comuns de universidades e escolas técnicas, sem gastos extras. Para o desenvolvimento serão utilizados dois computadores domésticos, com preço médio de R\$1.580, de acordo com um levantamento do instituto IDC Brasil em 2014, sistema operacional gratuito Linux, e plataforma de desenvolvimento gratuita. Em relação ao custo de mão de obra, serão necessários dois profissionais de análise e desenvolvimento de sistemas, com salário médio de 15,625 reais/ hora de



trabalho. Levando em consideração os 4 meses de desenvolvimento, com uma dedicação de 4 horas semanais, é possível estipular um custo total de R\$2000 para mão de obra.

### **4.3 VIABILIDADE TÉCNICA**

O projeto é facilmente realizável tecnicamente, pois o modelo de um *webserver* já foi amplamente discutido teoricamente, e há vários exemplos de *software* disponíveis. Portanto a modelagem do servidor em si já está praticamente resolvida.

Além disso a linguagem escolhida (Python) oferece um bom suporte para os outros requisitos do sistema (acesso ao banco de dados, processamento multitarefa, comunicação interprocessos e tratamento de erros) que também já são considerados problemas resolvidos pela comunidade de *software*. E também, ambos os membros da equipe já possuem conhecimento suficiente (ministrados nas disciplinas do curso de Engenharia de Computação) para atacar esses problemas.

A dificuldade maior será integrar cada parte do sistema e criar a interface gráfica (única parte do sistema que deve ser feita por inteiro). Além disso é preciso estimar o número máximo de conexões para que o sistema tenha um desempenho satisfatório.

Em relação ao hardware o projeto também é bastante simples, pois necessita apenas de um computador comum conectado à internet. Não é necessário nenhum conhecimento específico de hardware na área de redes ou cabeamento para implementar o projeto. É necessário apenas um computador com placa de rede e um ponto de acesso à internet.

### **4.4 VIABILIDADE OPERACIONAL**

Como já foi dito, o usuário direto do sistema terá acesso a várias informações referentes ao processo de hospedar páginas web. Portanto pode-se dizer que o *software* descreve com detalhes uma situação que pode ser abordada no estudo de redes de computadores, e portanto pode ser usado como ferramenta didática.

Para operar o sistema o usuário não necessita de nenhum conhecimento específico de software, deve apenas saber rotinas básicas no uso de computadores, por exemplo: iniciar e encerrar programas em seu

sistema operacional e interagir com elementos de interface gráfica(botões, barras de rolagem, listas de seleção, campos de texto).

Um fator que melhora a interação do usuário com o programa é saber interpretar os dados apresentados na forma de gráficos e tabelas, dessa forma, mesmo que o usuário não possua conhecimentos sobre redes e servidores, ele pode aprender alguns conceitos simplesmente ao analisar os dados do programa.

A aceitação do sistema vai depender do nível de conhecimento do usuário e os objetivos de aprendizado dele. Para o usuário indiretos o sistema é transparente, ou seja, irá se comportar como qualquer outro *webserver*, desde que funcione adequadamente.

Em relação às questões legais de operação do *software* não há nenhum impedimento, pois inicialmente o projeto não terá fins comerciais e portanto o conteúdo hospedado no servidor serão páginas HTML de teste para fins didáticos, contendo apenas texto e imagens de domínio público.

## **4.5 RECURSOS A SEREM UTILIZADOS**

Este projeto não requer muitos recursos, visto que exige apenas a implementação do software descrito no documento de projeto. Pode inclusive ser feito por uma única pessoa, dessa forma a análise dos recursos será feita com base nos recursos utilizados por pessoa que irá desenvolver o projeto. Cada desenvolvedor precisa de um computador com acesso a uma rede e o documento de especificação do projeto. Além disso o tempo estimado para o projeto é de 2 meses, com regime de trabalho de 4 horas semanais. Então o tempo total de trabalho no projeto é de 32 horas.

Considerando o salário médio de um analista e desenvolvedor de sistemas , podemos estimar o valor de sua hora de trabalho:

2500 reais mensais /160 horas por mês = 15,625 reais/ hora de trabalho;

Custo por pessoa =  $32 * 15,625 = 500$  reais;

Custo total = Custo por pessoa \* número de programadores.

Dessa forma pode-se estipular o custo do projeto caso ele fosse desenvolvido utilizando mão de obra de terceiros.

Em relação a distribuição, o software será gratuito e de código aberto. Para sua utilização, o usuário deverá dispor de um computador doméstico para o servidor, o qual tem um custo médio de R\$1.580. Outros computadores deverão ser utilizados como clientes do servidor, e a quantidade fica a critério do usuário. Se o usuário possuir acesso a uma rede, poderá utilizá-la para comunicação entre os computadores. Se não, deverá ser adquirido um comutador simples, o qual pode ser encontrado por cerca de R\$ 50,00.

## 5 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso do sistema, apresentado na **Figura** a seguir, é um resumo das funcionalidades do sistema, ou seja, as ações que ele pode executar resultantes ou não da intervenção do usuário. Esta figura inicial é apenas uma visão geral do funcionamento do sistema, para ter uma visão a respeito da estrutura do sistema e suas partes ver o **capítulo 6 diagrama de classes**

A Figura 1 representa o funcionamento do sistema através de seus casos de uso. Cada um deles é descrito em detalhes nas seções seguintes.

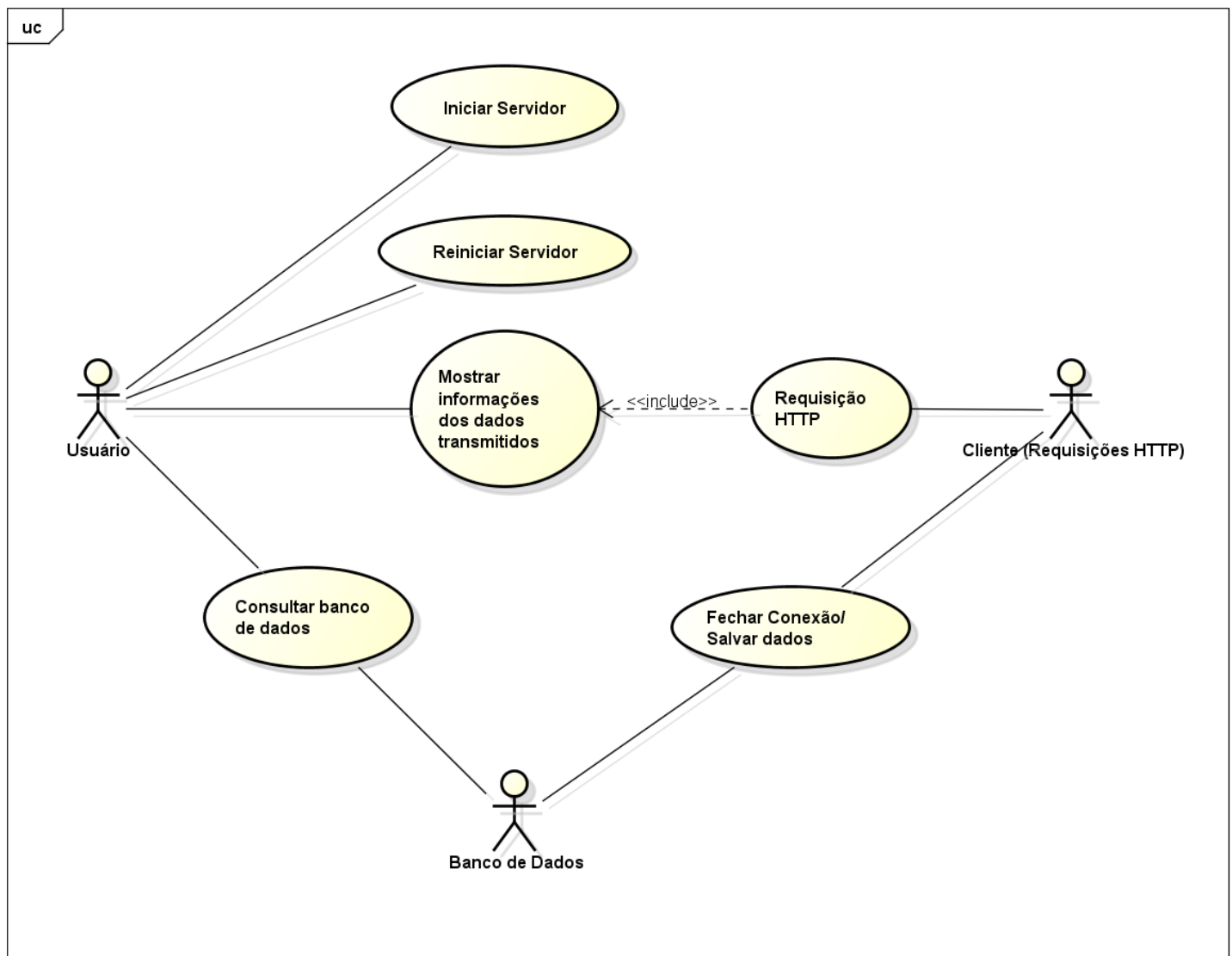


Figura 1: Diagrama de casos de uso. Fonte:autor

### 5.1.1 Caso de uso: Requisição HTTP

Nome	UC01: Requisição HTTP	
Atores	Ator principal: Cliente (Web request);	
Descrição	Executado a cada vez que o sistema recebe uma nova conexão ou requisição HTTP de um cliente.	
Pré-condições	-O sistema já deve ter iniciado o servidor para esperar conexões na respectiva porta;	
Pós-Condições	-Caso não exista conexão referente ao IP do cliente, uma nova é adicionada à lista de conexões do sistema; - A conexão referente ao IP do cliente tem seus dados atualizados.	
Fluxo Básico		
Ações dos atores	Ações do Sistema	
1-Cliente solicita uma conexão com servidor.		
	2-Sistema detecta uma mensagem do cliente, cria uma conexão e espera os dados (requisição HTTP).	
	4-Sistema decodifica a requisição e responde apropriadamente de acordo com RN01	
5-Cliente recebe a resposta do sistema (Página HTML e seus respectivos conteúdos)		
	6-Sistema mantém a conexão aberta e fica aguardando a próxima requisição HTTP do cliente.	
Regras de negócio		
RN01: Os dados transmitidos devem respeitar o protocolo HTTP		
Fluxo Alternativo		

<b>Ações dos atores</b>	<b>Ações do Sistema</b>
1-Cliente solicita que a conexão seja encerrada.	
	2-Encerra a conexão e a remove de sua lista interna de conexões.
<b>Fluxo de exceção</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
3.a Cliente envia dados fora do protocolo HTTP ou uma requisição inválida	
	4.a Sistema responde apropriadamente e aguarda uma requisição válida

### 5.1.2 Caso de uso: Mostrar informações dos dados transmitidos

<b>Nome</b>	UC02: Mostrar informações dos dados transmitidos.
<b>Atores</b>	Ator principal: Usuário (observa as informações). Ator secundário: Cliente (faz uma nova requisição HTTP).
<b>Descrição</b>	Executado a cada vez que uma conexão recebe uma requisição HTTP e o sistema responde.
<b>Pré-condições</b>	-Deve existir pelo menos uma conexão ativa para receber requisições.
<b>Pós-Condições</b>	-A respectiva conexão na lista do sistema, que é exibida na interface gráfica, é atualizada com os dados de cada requisição HTTP recebida.
<b>Fluxo Básico</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
	1-Sistema recebe uma requisição em uma de suas conexões.
	2-A respectiva conexão tem seus dados atualizados.

	3-A lista de conexões, identificadas por endereço IP, que aparece na aba “Conexões Ativas” da janela principal é atualizada com os novos valores.
4-Usuário pode navegar pela lista de conexões usando uma barra de rolagem vertical. E selecionar uma das conexões da lista para observar seus detalhes.	
	5-Sistema mostra na tela, ao lado da lista de conexões, os detalhes da conexão selecionada pelo usuário. Os detalhes são os dados de cada requisição : “Data”, “Hora” e “Bytes recebidos”, também organizados em forma de uma lista navegável através de barra de rolagem vertical
<b>Regras de negócio</b>	
Não se aplica a este caso.	
<b>Fluxo Alternativo</b>	
Não se aplica a este caso.	
<b>Fluxo de exceção</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
	1.a- O servidor nunca recebe uma requisição (nenhum cliente acessou o servidor).
4.a- Usuário observa uma lista em branco de conexões.	

### 5.1.3 Caso de uso: Fechar conexão/ Salvar dados

<b>Nome</b>	UC03: Fechar conexão/ Salvar dados
<b>Atores</b>	Ator principal: Cliente (solicita que uma conexão seja encerrada) Ator secundário: Banco de dados (guarda os dados referentes à conexão encerrada)
<b>Descrição</b>	Executado a cada vez que uma conexão é encerrada

<b>Pré-condições</b>	-Deve existir pelo menos uma conexão ativa para fornecer os dados a serem salvos.
<b>Pós-Condições</b>	-A conexão (e suas respectivas informações) é salva no banco de dados.
<b>Fluxo Básico</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
1- Cliente solicita que a conexão seja encerrada.	
	2- A conexão é removida da lista de conexões do sistema.
	3- Os dados da conexão são formatados e enviados ao banco de dados.
<b>Regras de negócio</b>	
Não se aplica a este caso.	
<b>Fluxo Alternativo</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
1- A conexão atinge tempo limite de inatividade (sem receber requisições HTTP).	
	2- Sistema executa o passo 2 do fluxo básico
<b>Fluxo de exceção</b>	
Não se aplica a este caso.	

#### 5.1.4 Caso de uso: Consultar banco de dados

<b>Nome</b>	UC05: Consultar banco de dados
<b>Atores</b>	Ator principal: Usuário (solicita consulta) Ator secundário: Banco de dados (fornece os dados referentes a consulta)
<b>Descrição</b>	Permite ao usuário consultar no banco de



	dados informações das conexões armazenadas.
<b>Pré-condições</b>	– O sistema já deve ter iniciado o servidor.
<b>Pós-Condições</b>	– A interface gráfica apresenta os dados requeridos na consulta.
<b>Fluxo Básico</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
1-Usuário escolhe na interface quais informações deseja consultar.	
	2- Sistema busca (realiza uma consulta) no banco de dados.
3- Banco de dados retorna o resultado da consulta.	
	4- Sistema apresenta na interface gráfica o resultado da busca.
<b>Regras de negócio</b>	
Não se aplica a este caso.	
<b>Fluxo Alternativo</b>	
Não se aplica a este caso.	
<b>Fluxo de exceção</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
	1- Nenhuma conexão foi salva no banco de dados, então a busca não obtém resultados.
2- O usuário visualiza área de resultados vazia.	

### 5.1.5 Caso de uso: Iniciar o servidor

<b>Nome</b>	UC05: Iniciar servidor
<b>Atores</b>	Ator principal: Usuário (escolhe configurações do sistema)

<b>Descrição</b>	Permite ao usuário definir parâmetros do servidor
<b>Pré-condições</b>	- O servidor ainda não foi iniciado
<b>Pós-Condições</b>	- O sistema passa a operar com os parâmetros definidos.
<b>Fluxo Básico</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
	1-Sistema mostra a sua janela inicial com os parâmetros : “Máximo de conexões”, “Porta”, “Time Out” e “Habilitar Donwload” descritos no dicionário de informações da Classe Server.
2- Usuário determina os valores de cada parâmetro e clica no botão “Iniciar Servidor” para passar os parâmetros para o sistema e iniciar o servidor.	
	3- O sistema cria o servidor e outros componentes de acordo com os parâmetros estabelecidos.
<b>Regras de negócio</b>	
Não se aplica a este caso.	
<b>Fluxo Alternativo</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
2.a- Usuário a qualquer momento clica no botão “SAIR” ou fecha a janela inicial.	
	3.a- O sistema encerra a interface gráfica e termina seu processo de execução
<b>Fluxo de exceção</b>	
Não se aplica a este caso.	

### 5.1.6 Caso de uso: Reiniciar o servidor

<b>Nome</b>	UC06: Reiniciar servidor
<b>Atores</b>	Ator principal: usuário.
<b>Descrição</b>	A cada vez que o usuário mudar um dos parâmetros iniciais o servidor deve encerrar todas as conexões, salvar as respectivas informações e esperar novas conexões de acordo com os novos parâmetros definidos.
<b>Pré-condições</b>	- O servidor já deve estar em funcionamento.
<b>Pós-Condições</b>	- Todas as conexões são encerradas e seus respectivos dados são enviados ao banco de dados.
<b>Fluxo Básico</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
1- Com o servidor em funcionamento, o usuário escolhe novos parâmetros na aba “Configurações” da janela principal e clica no botão “Reiniciar Servidor”	
	2- Todas conexões são encerradas.
	3- As informações das conexões encerradas são armazenadas no banco de dados.
	4- O sistema redefine os parâmetros de funcionamento do servidor e o reinicia para esperar novas conexões.
	5- As abas “Conexões Ativas” e “Indicadores” reiniciam(zeram) os valores de seus componentes
<b>Fluxo Alternativo</b>	
<b>Ações dos atores</b>	<b>Ações do Sistema</b>
Não se aplica neste caso.	
<b>Fluxo de exceção</b>	
1- Com o servidor em funcionamento, o usuário escolhe novos parâmetros na aba “Configurações” da	

janela principal e clica no botão "Reiniciar Servidor"	
	2.e- O servidor ainda não tem nenhuma conexão ativa.
	3.e- Executa o passo 4 do fluxo básico.

## 6 DIAGRAMA DE CLASSES

Nesta sessão é apresentado o diagrama de classes do projeto, apresentando uma estruturação das partes a serem implementadas no software. O diagrama de classes está presente em:, **página 75**.Cada classe é descrita mais detalhadamente na seção seguinte. Além disso as interações entre as classes são detalhadas posteriormente em seções individuais.

### 6.1 Dicionário de informações

Cada classe apresentada no diagrama de classes é descrita em detalhes nas tabelas na seguir, obedecendo à seguinte relação: classe “JanelaInicial” na **tabela 1**; “JanelaPrincipal” na **tabela 2**; classe “Server” na **tabela 3**; classe “Indicadores” na **tabela 4**; classe “AcessoSGBD” na **tabela 5**; classe “Conexão” na **tabela 6**, classe “HTTPParser” na **tabela 7**, classe “Requisição” na **tabela 8** e classe “Resposta” na **tabela 9**.

\***Atributos** referentes à interface gráfica ainda não foram definidos, pois dependem do tipo de implementação, estes serão definidos em uma seção a parte quando a interface gráfica estiver implementada. Apenas os atributos essenciais para o funcionamento do sistema estão descritos no dicionário de informações.

**Tabela 1: Informações da classe JanelaInicial**

<b>JanelaInicial:</b> Primeiro elemento da interface gráfica apresentado ao usuário. Através dessa classe o usuário pode definir os parâmetros iniciais do sistema. Não possui nenhum atributo pois apenas repassa as informações coletadas do usuário para a <b>classe Server</b> .					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
Ainda não definido*					

**Tabela 2: Informações da classe JanelaPrincipal**

<p><b>Janela Principal:</b> Elemento da interface gráfica que permite ao usuário interagir com todas as partes do sistema. O usuário pode ver a lista de conexões do sistema e seus respectivos detalhes; analisar o funcionamento do sistema o longo do tempo através dos indicadores redefinir os parâmetros de funcionamento e fazer uma consulta aos dados salvos no banco de dados.</p> <p>Não possui nenhum atributo pois apenas mostra as informações coletadas a partir de atributos de outras classes.</p>					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
Ainda não definido*					

**Tabela 3: Informações da classe Server**

<p><b>Server:</b> Classe responsável por iniciar o servidor com os parâmetros fornecidos pelo usuário. Também cria as outras classes que compõem o sistema (<b>JanelaPrincipal</b>, <b>AcessoSGBD</b>, <b>Indicadores</b>) e instâncias da classe <b>Conexão</b> quando necessário.</p>					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
maxConeoxoes	Número máximo de conexões que o sistema pode servir ao mesmo tempo	4	Numérico	{9}4	Contínuo
Porta	Porta utilizada pelas conexões do sistema	5	Numérico	{9}5	Contínuo
Download	Define se será permitido suporte a download ou não	1	Booleano	Não se aplica	Discreto True ou False
timeOut	Tempo máximo de inatividade (em segundos) de uma conexão antes que ela seja encerrada pelo sistema	5	Numérico	{9}5	Contínuo
listaConexoes	Lista de todas as conexões ativas no sistema	1000 (máximo de conexões do servidor)	Classe Conexão	Não se aplica	Não se aplica

**Tabela 4: Informações da classe Indicadores**

<p><b>Indicadores:</b> Classe responsável por gerar alguns indicadores referentes ao uso do sistema ao longo do tempo. Basicamente coleta informações de outras classes e as organiza em forma de gráficos e valores médios.</p>					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio

mediaBytesEnviados	Valor médio dos bytes enviados por uma conexão do sistema	9	Numérico	{9}9	Contínuo
tempoMedioCoexao	Tempo médio de duração de uma conexão do sistema	5	Numérico	{9}5	Contínuo
pagMaisAcessada	Página HTML mais acessada pelos clientes	Variável	Alfanumérico	Não se aplica	Contínuo
mediaBytesRecebidos	Média de bytes recebidos por uma conexão do sistema	9	Numérico	{9}9	Contínuo

**Tabela 5: Informações da classe AcessoSGBD**

<b>AcessoSGBD:</b> Classe responsável por fazer a interação com o banco de dados. Não possui nenhum atributo, pois apenas repassa informações.					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
Não se aplica					

**Tabela 6: Informações da classe Conexão**

<b>Conexão:</b> Classe responsável por fazer a comunicação com o cliente e armazenar temporariamente os dados resultantes desta ação.					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
IP	endereço IP de origem da conexão (IP do cliente)	12	Numérico	{9}12	Contínuo
Porta	Porta do sistema usada pela conexão	5	Numérico	{9}5	Contínuo
data	Data de início da conexão	14	Alfanumérico	DD/MM/AAAA hh:mm:ss	Contínuo D=dia D=mês A=ano h=hora m=minuto s=segundo

requisiçõesRecebidas	Conjunto de todas as requisições recebidas até o momento na conexão	Variável	Classe Requisição	Não se aplica	Contínuo
paginasAcessadas	Conjunto de todas as páginas acessadas pelo cliente na conexão	Variável	Alfanumérico	Não se aplica	Contínuo
bytesEnviados	Total de bytes enviados até o momento	9	Numérico	{9}9	Contínuo
bytesRecebidos	Total de bytes recebidos até o momento	9	Numérico	{9}9	Contínuo
tempoInativo	Tempo sem que a conexão receba uma requisição HTTP	6	Numérico	{9}6	Contínuo

**Tabela 7: Informações da classe HTTPParser**

<b>HTTPParser:</b> Não possui atributos pois é uma classe auxiliar para decodificar as requisições salvas no <b>atributo requisiçõesRecebidas</b> da classe Conexão.					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
Não se aplica					

**Tabela 8: Informações da classe Requisições**

<b>Requisição:</b> Classe responsável por guardar as informações de cada requisição que chega em uma conexão ativa do sistema. Esta classe é associada à <b>classe Conexão</b> como um atributo.					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
tipo	Método do protocolo HTTP contido na requisição	Variável	Alfanumérico	Não se aplica	Contínuo
paginaAcessada	Endereço da página	Variável	Alfanumérico	Não se aplica	Contínuo



	solicitada				
versaoProtocoloHTTP	Versão do protocolo HTTP utilizada na requisição	3	Alfanumérico	X.Y	Contínuo
hostname	Servidor do endereço solicitado	Variável	Alfanumérico	Não se aplica	Contínuo
data	Data da requisição	14	Alfanumérico	DD/MM/AAA A hh:mm:ss	Contínuo D=dia D=mês A=ano h=hora m=minuto s=segundo
respostas	Lista de objetos da classe Resposta contendo as respostas à requisição	Variável	Classe Resposta	Não se aplica	Contínuo

**Tabela 9: Informações da classe Resposta**

<b>Resposta:</b> Classe responsável por armazenar as respostas de cada requisição que chega em uma conexão ativa no sistema. Esta classe é associada à <b>classe Requisição</b> como um atributo					
Atributo	Descrição	Tamanho	Tipo	Formato	Domínio
header	Cabeçalho da mensagem de resposta à requisição	Variável	Alfanumérico	Não se aplica	Contínuo
content	Conteúdo da resposta do servidor à requisição	Variável	Alfanumérico	Não se aplica	Contínuo

## 6.2 Diagrama de Objetos (Instâncias).

Nesta sessão é apresentado o diagrama de objetos do projeto. Este diagrama se trata de uma variação do diagrama de classes apresentado anteriormente e tem como objetivo mostrar os objetos instanciados de cada classe. A figura em apêndice na **seção , página 76**, apresenta esse diagrama.

## 7 DIAGRAMAS DE SEQUÊNCIA

Esta seção descreve com mais detalhes o que ocorre no sistema cada vez que um caso de uso é executado. Os diagramas de sequência mostram qual a relação entre as classes (partes) do sistema, atores (usuário direto, indireto e banco de dados) e a respectiva funcionalidade (caso de uso). Em cada diagrama as classes e os atores são representados na parte superior da figura, cada interação entre as partes pode ser uma mensagem ou chamada de método, que são representados por setas e devem seguir uma ordem na linha do tempo (representada pelas linhas tracejadas).

## 7.1 UC01: Realizar Conexão

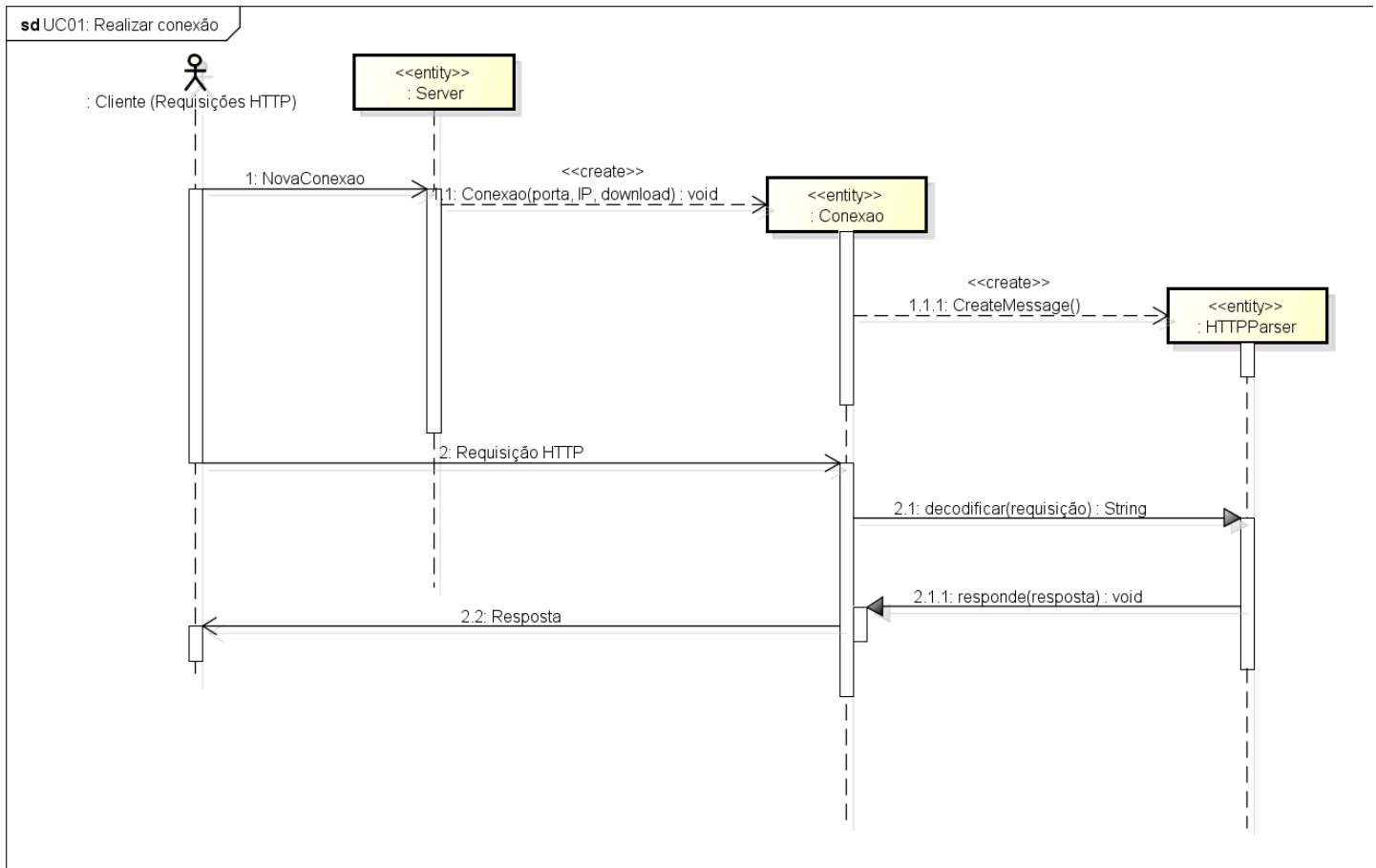


Figura 3: Diagrama de sequência para UC01:Realizar Conexão Fonte: autor

## 7.2 UC02: Mostrar informações dos dados transmitidos

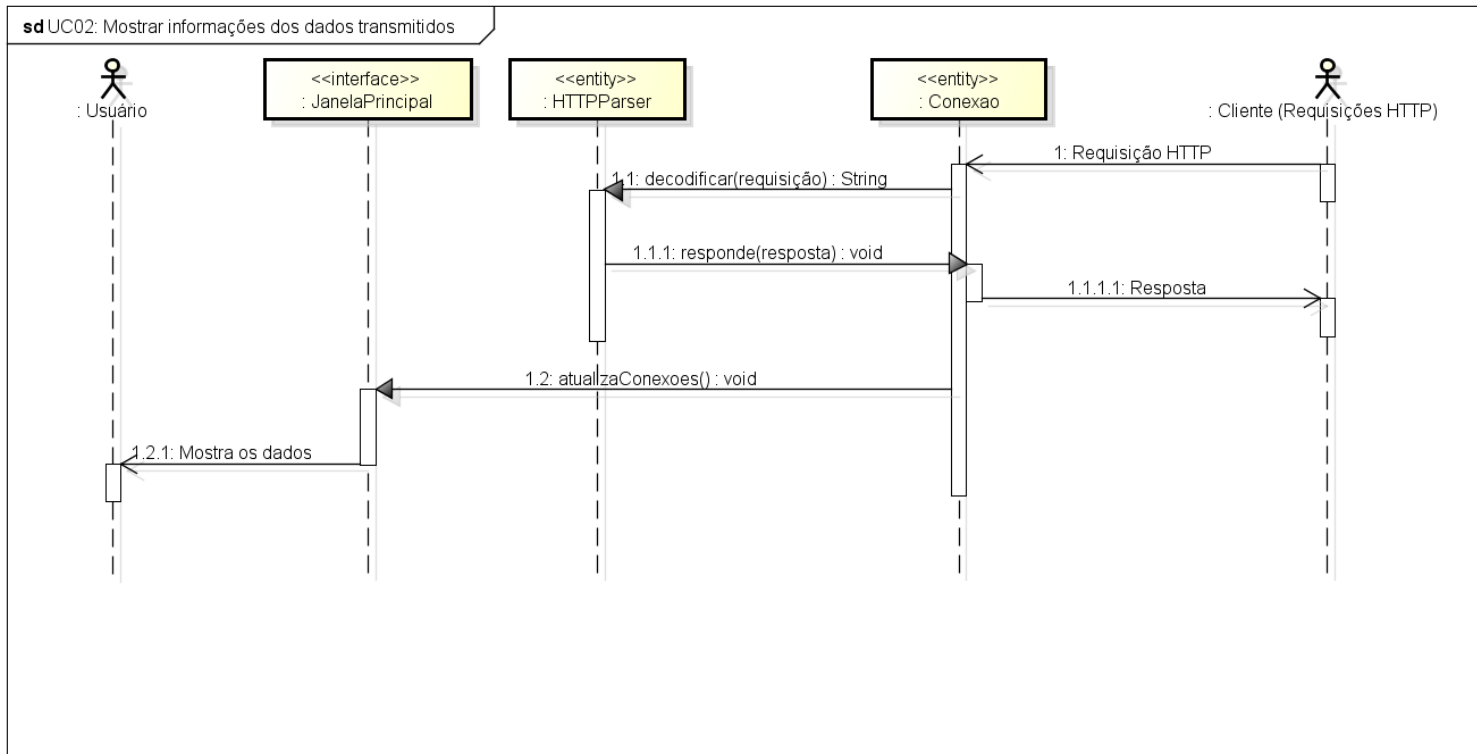


Figura 4 :Diagrama de sequência para UC02:Mostrar informações dos dados transmitidos  
Fonte: autor

## 7.3 UC03: Salvar dados da conexão

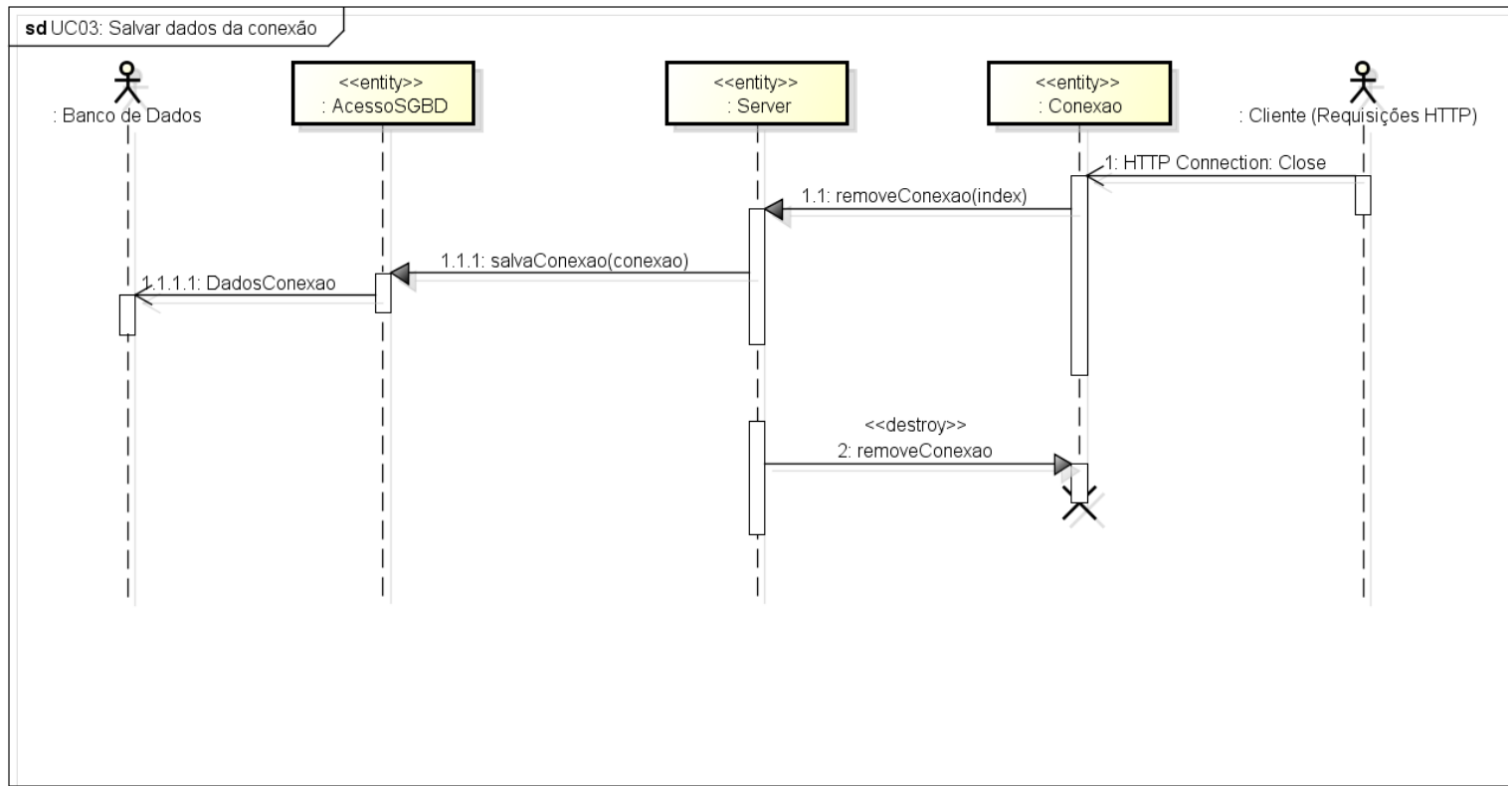


Figura 5 Diagrama de sequência para UC03: Salvar dados da conexão Fonte: autor

## 7.4 UC04: Consultar banco de dados

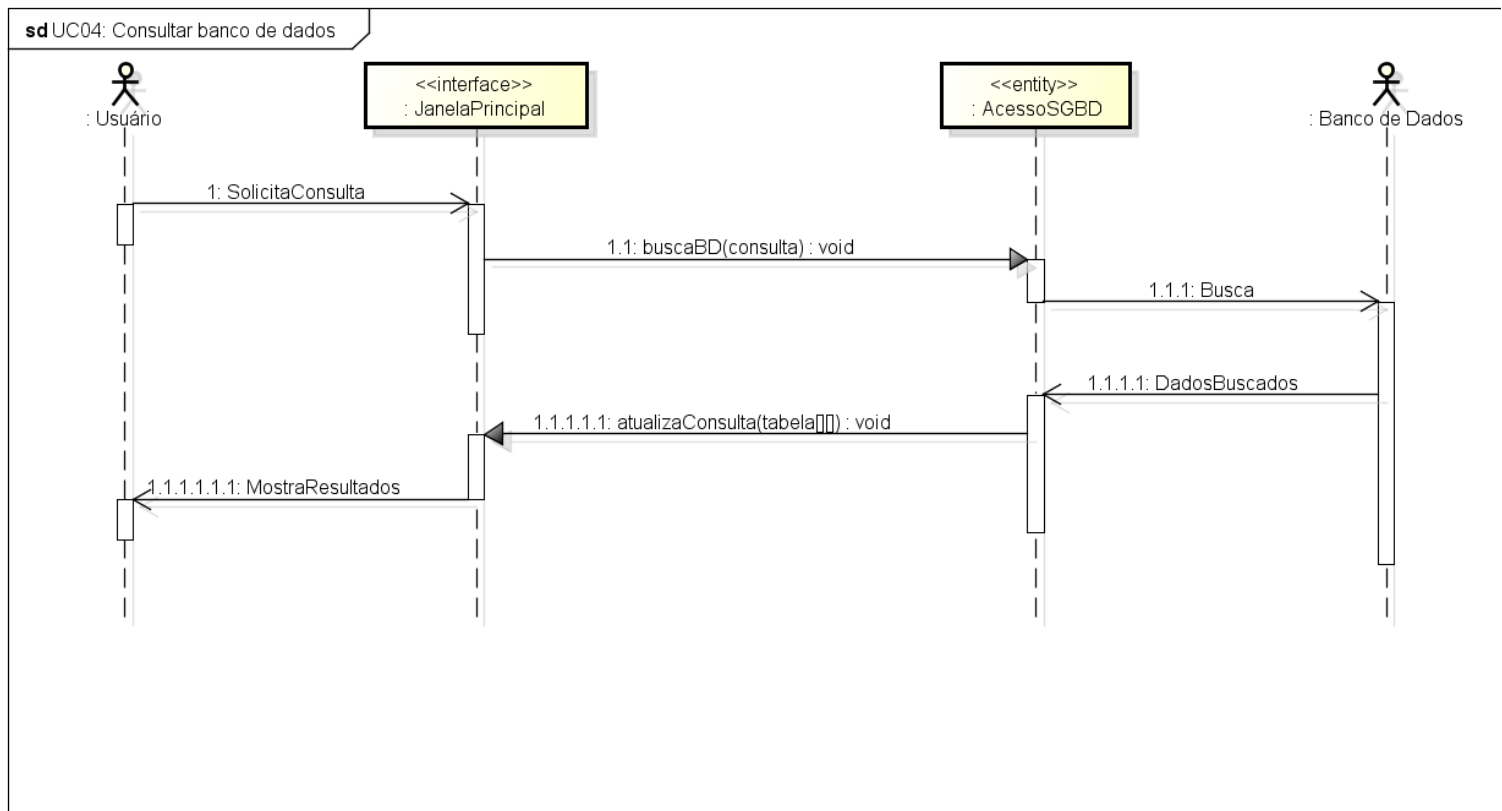


Figura 6: Diagrama de sequência para UC04:Consultar banco de dados Fonte: autor

## 7.5 UC05: Iniciar Servidor

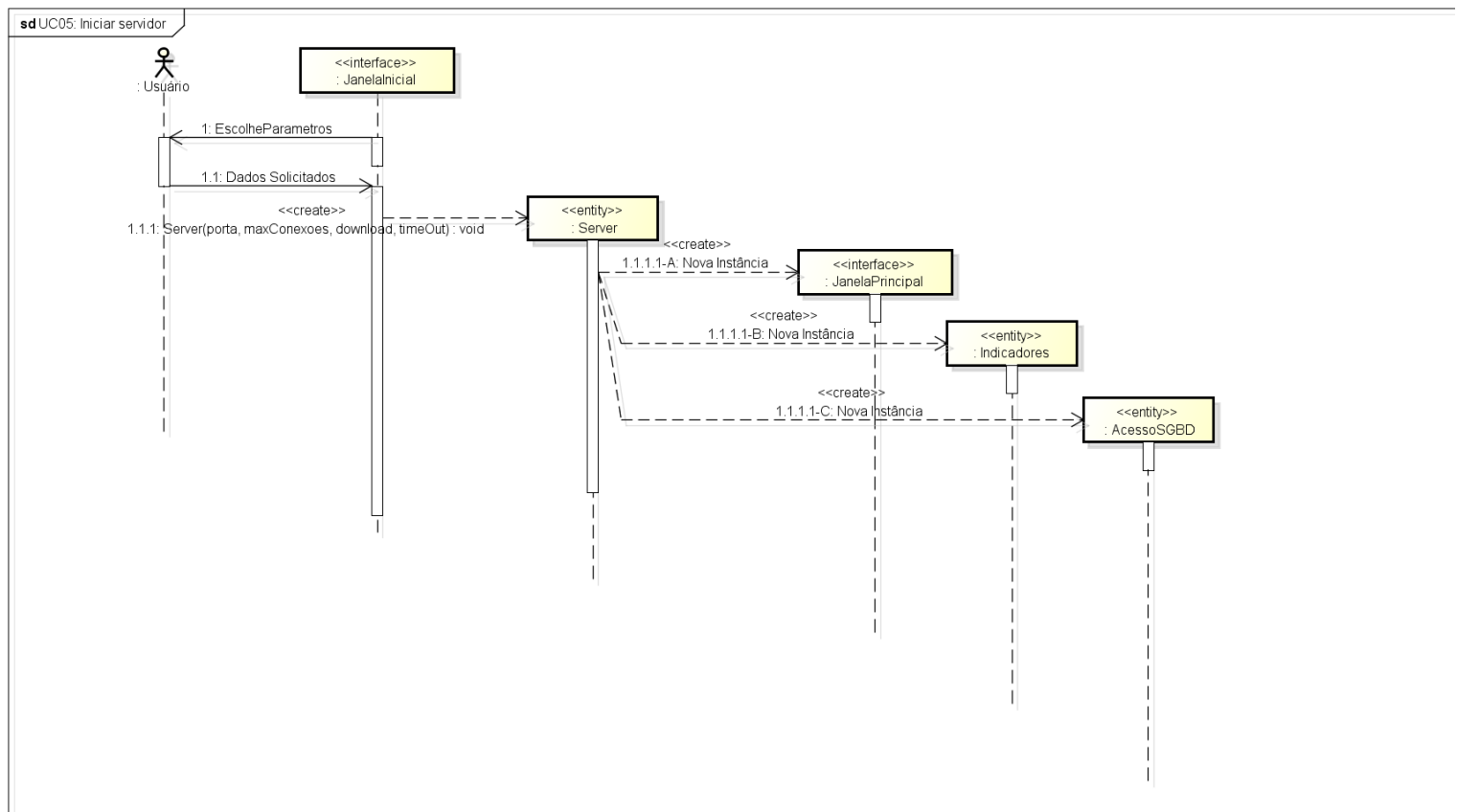


Figura 7: Diagrama de sequência para UC05:Iniciar servidor Fonte: autor



## 7.6 UC06: Reiniciar Servidor

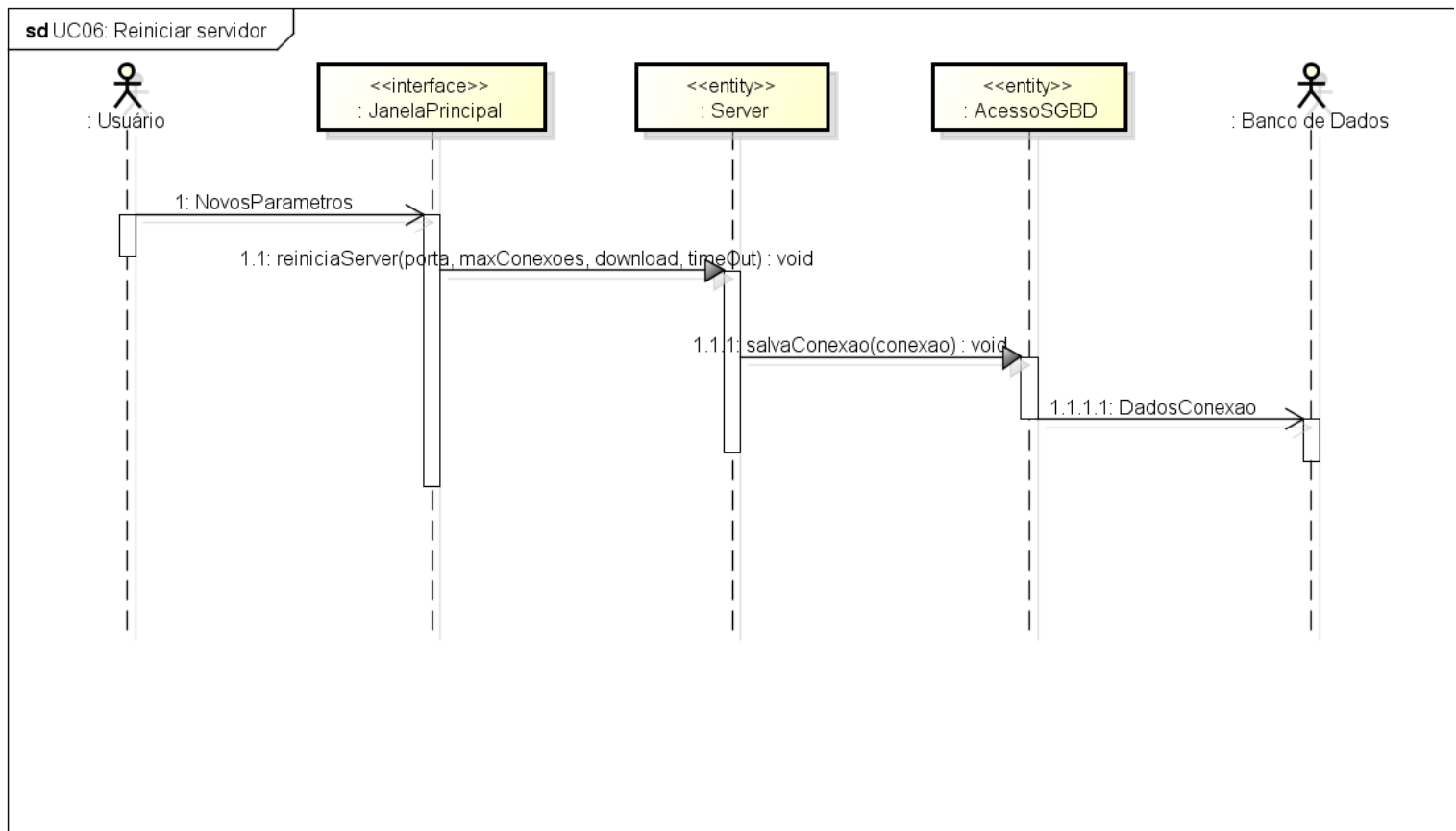
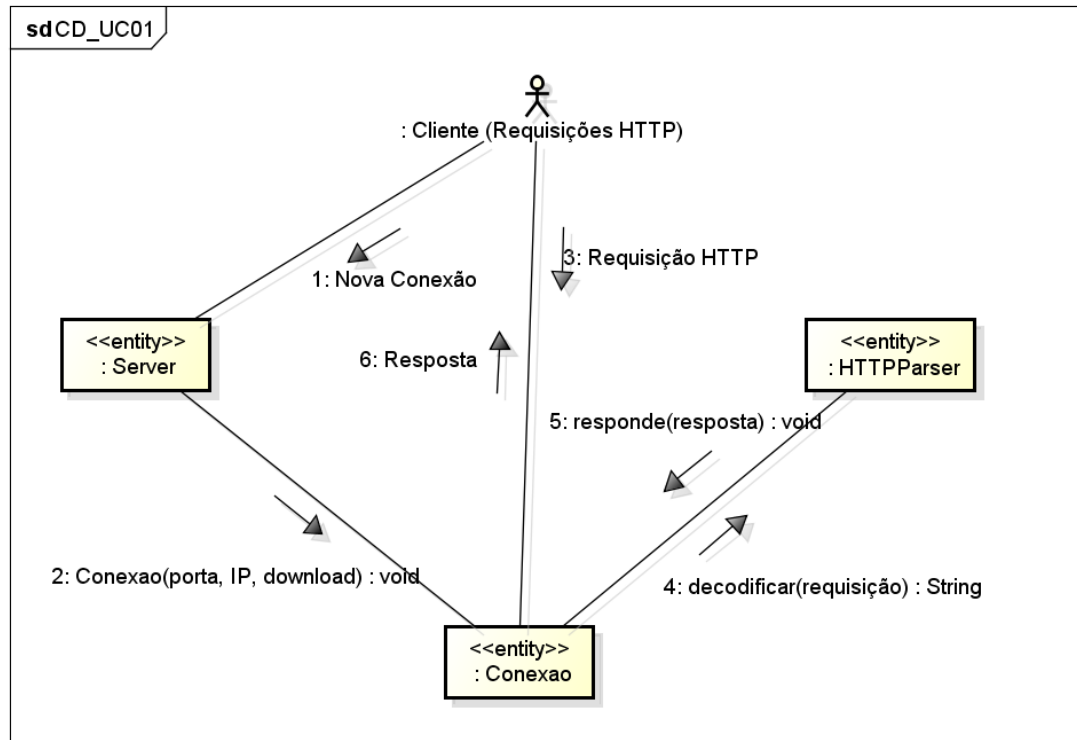


Figura 8: Diagrama de sequência para UC06:Reiniciar servidor Fonte: autor

## 8 DIAGRAMAS DE COMUNICAÇÃO

Nesta seção serão apresentados os diagramas de comunicação elaborados a partir dos diagramas de sequência apresentados na seção anterior. Estes diagramas apresentam a organização dos objetos que participam de cada caso de uso e a comunicação entre os mesmos.

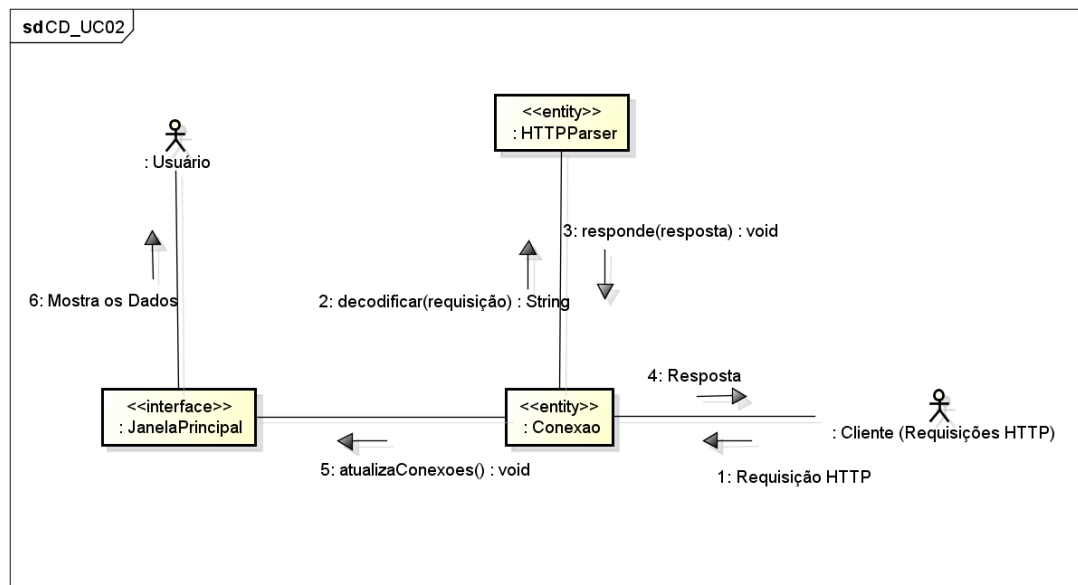
### 8.1 UC01: Realizar conexão



powered by Astah

Figura 9: Diagrama de sequência para UC01:Realizar conexão. Fonte: autor

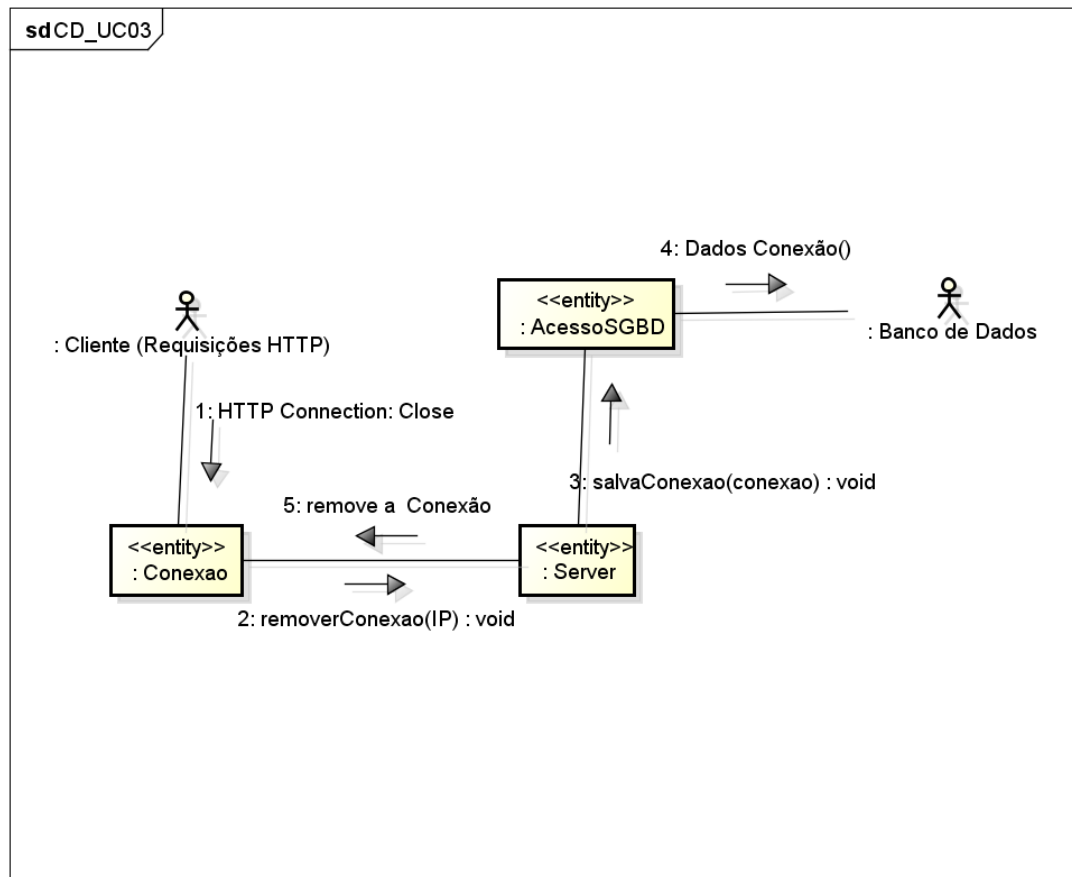
## 8.2 UC02: Mostrar informações dos dados transmitidos



powered by Astah

**Figura 10: Diagrama de sequência para UC02: Mostrar informações dos dados transmitidos.**  
Fonte: autor

### 8.3 UC03: Salvar dados da conexão



powered by Astah

Figura 11: Diagrama de sequência para UC03: Salvar dados da conexão. Fonte: autor

## 8.4 UC04: Consultar banco de dados

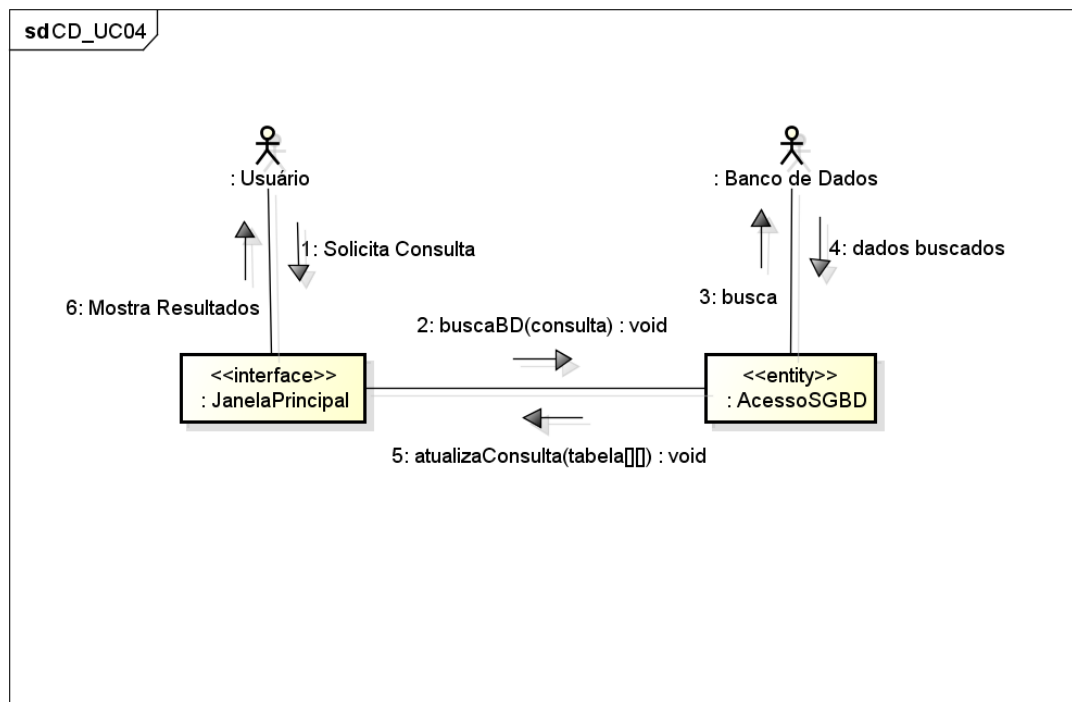
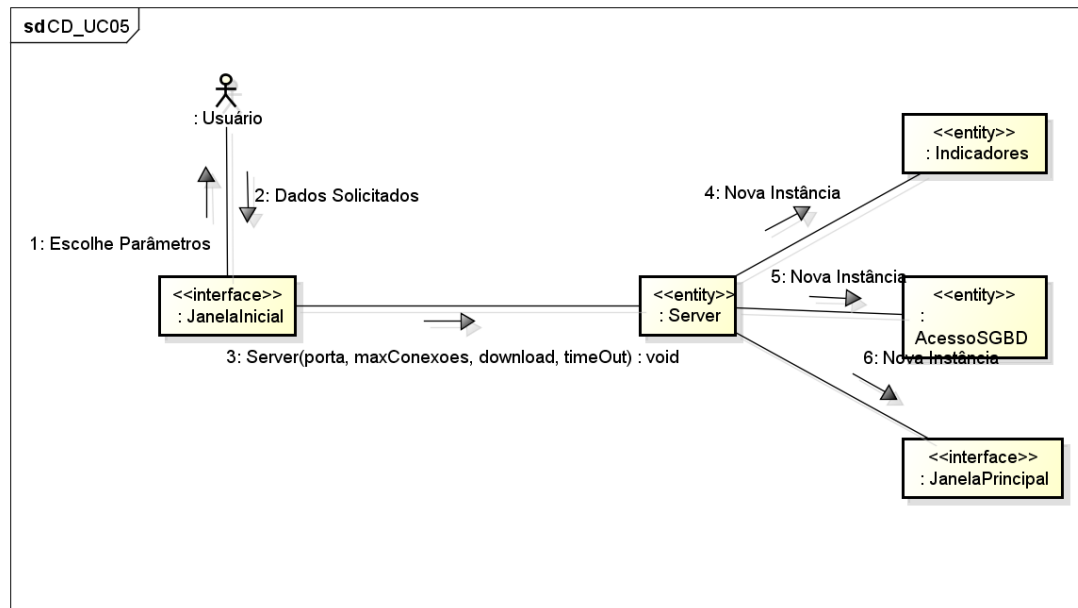


Figura 12: Diagrama de sequência para UC04: Consultar banco de dados. Fonte: autor

## 8.5 UC05: Inicar servidor



powered by Astah

Figura 13: Diagrama de sequência para UC05: Inicar servidor. Fonte: autor

## 8.6 UC06: Reiniciar servidor

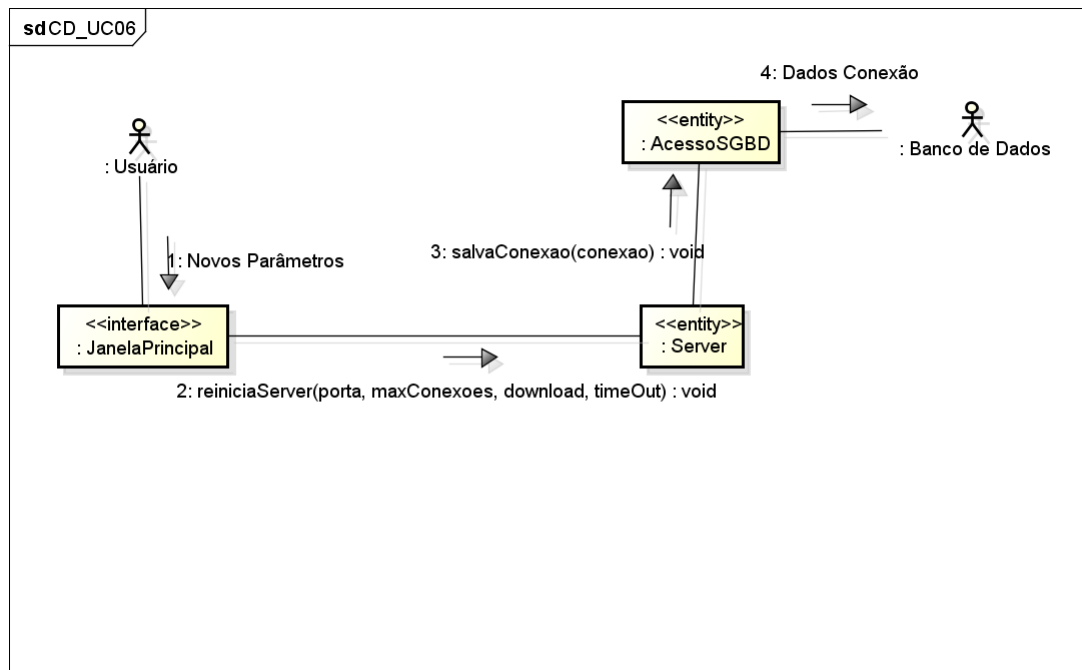
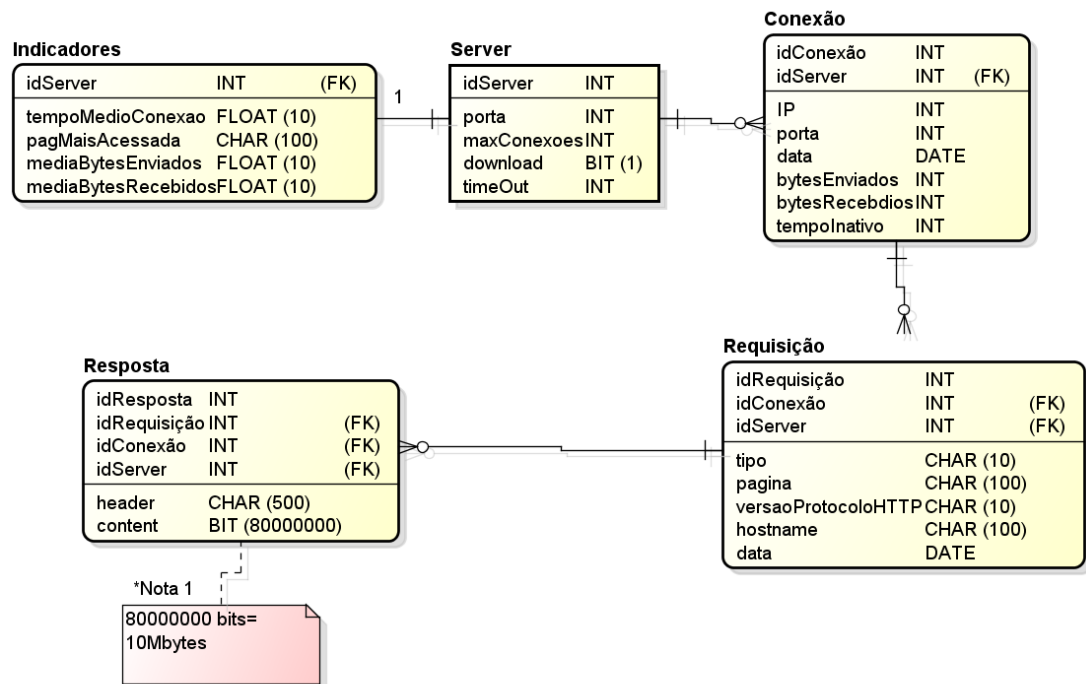


Figura 14: Diagrama de sequência para UC06: Reiniciar servidor. Fonte: autor

## 9 MAPEAMENTO DO DIAGRAMA DE CLASSES PARA DIAGRAMA ENTIDADE-RELACIONAMENTO

Abaixo é apresentado o mapeamento do Diagrama de Classes, mostrado na seção 6, para o Diagrama Entidade-Relacionamento (DER). Este diagrama em nível lógico tem como objetivo a sincronização entre o sistema e o banco de dados.



powered by Astah

Figura 15: Diagrama Entidade-Relacionamento. Fonte: autor

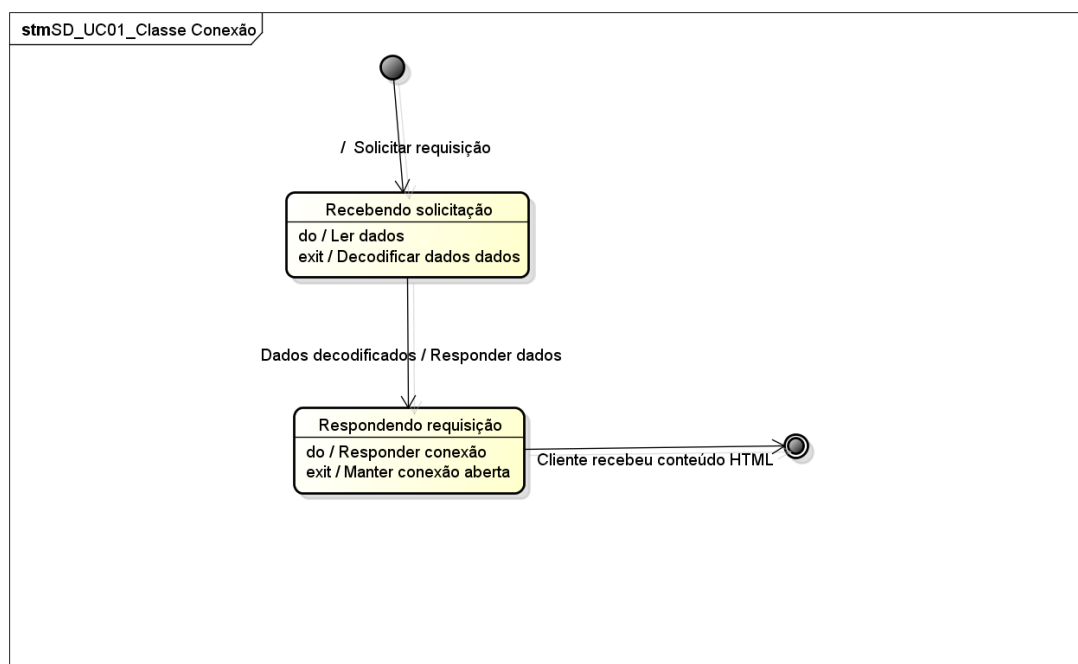


## 10 DIAGRAMA DE ESTADOS

Nesta seção serão apresentados Diagramas de Estados presentes no projeto. Serão mostrados dois Diagramas de Estado referentes à duas classes presentes em cada Diagrama de Estado. Estes diagramas têm objetivo de apresentar o comportamento de uma classe em um determinado caso de uso.

### 10.1 UC01: Realizar conexão

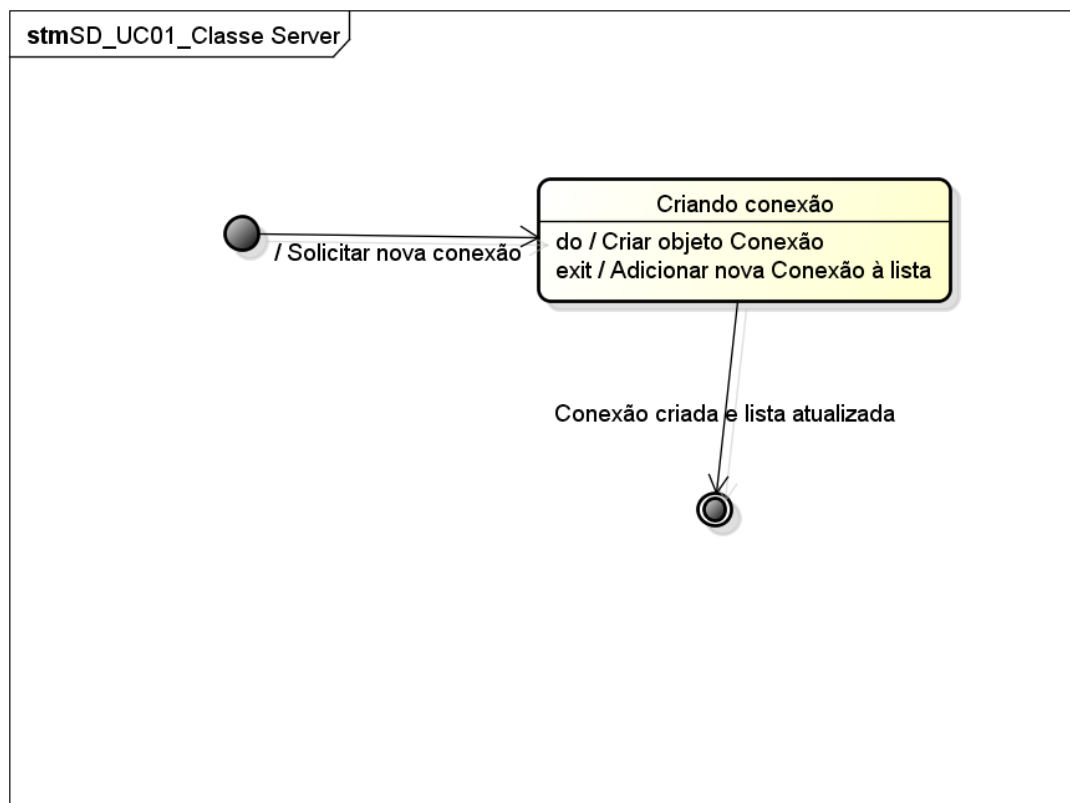
#### 10.1.1 Classe Conexao



powered by Astah

Figura 17: Diagrama de Estados UC01: Realizar conexão - Classe Conexao. Fonte: Autor.

## 10.1.2 Classe Server

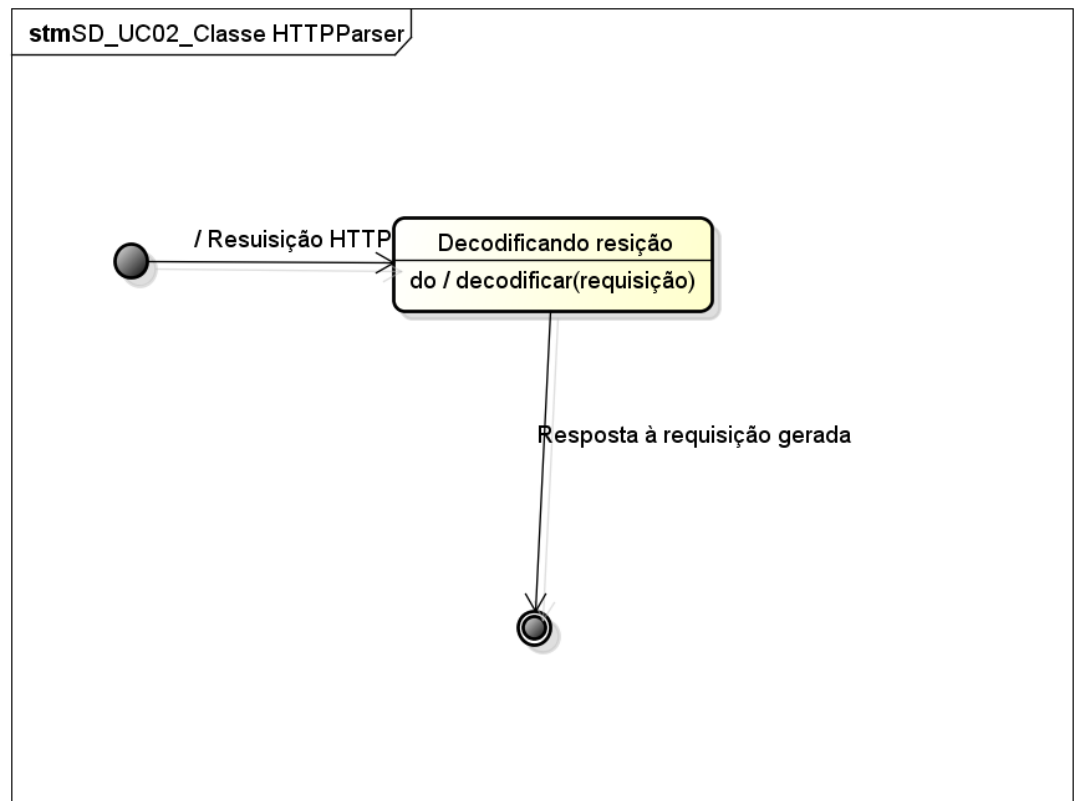


powered by Astah

Figura 18: Diagrama de Estados UC01: Realizar conexão - Classe Server. Fonte: Autor.

## 10.2 UC02: Mostrar informações dos dados transmitidos

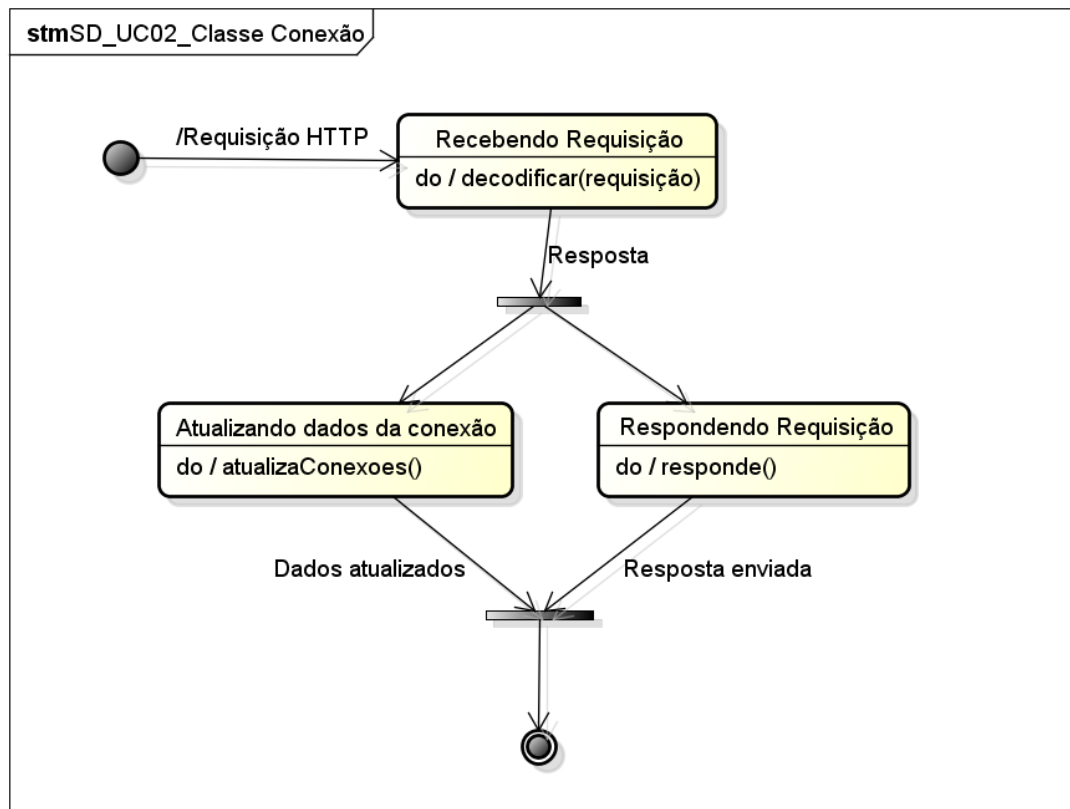
### 10.2.1 Classe Conexao



powered by Astah

Figura 19: Diagrama de Estados UC02: Mostrar informações dos dados transmitidos - Classe Conexao. Fonte: Autor.

## 10.2.2 Classe HTTPParser

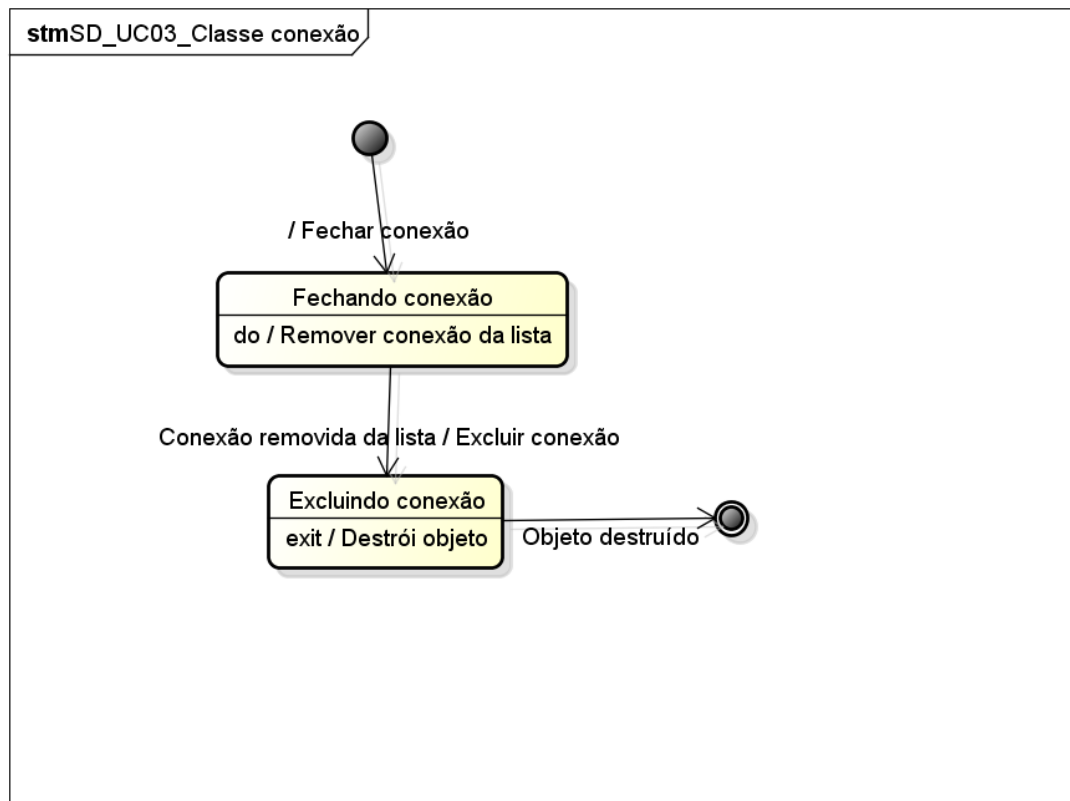


powered by Astah

Figura 20: Diagrama de Estados UC02: Mostrar informações dos dados transmitidos - Classe HTTPParser. Fonte: Autor.

## 10.3 UC03: Salvar dados da conexão

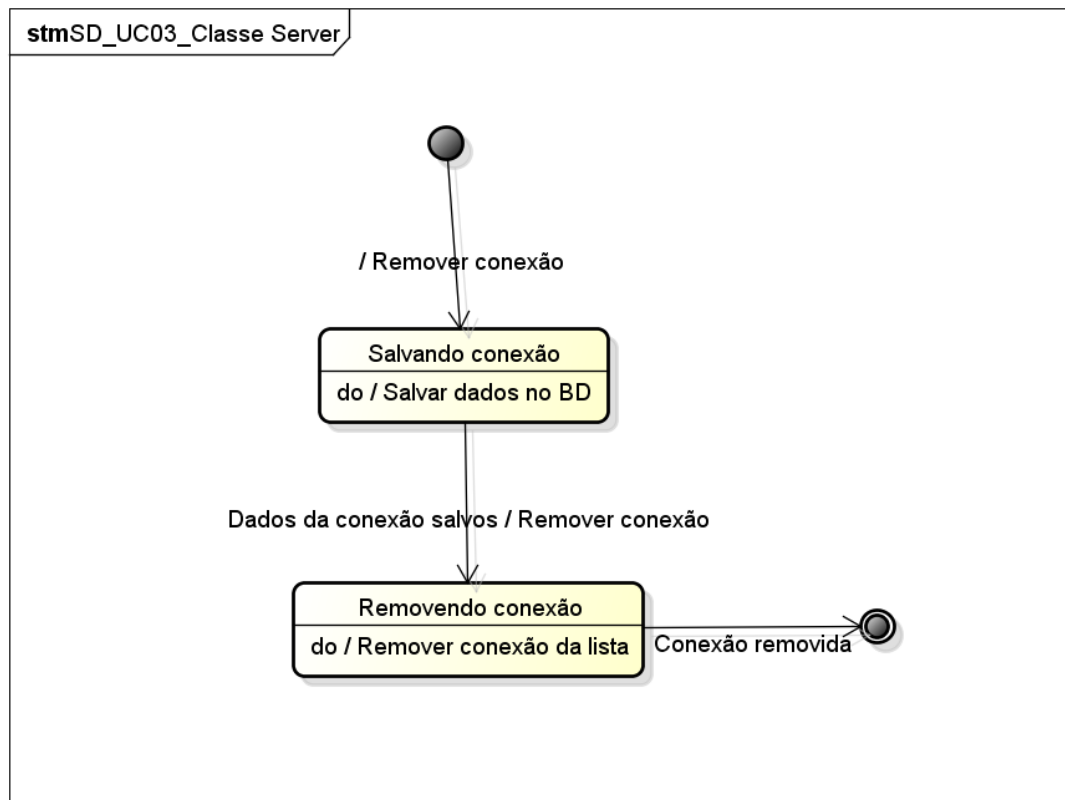
### 10.3.1 Classe Conexao



powered by Astah

Figura 21: Diagrama de Estados UC03: Salvar dados da conexão - Classe Conexao. Fonte: Autor.

### 10.3.2 Classe Server

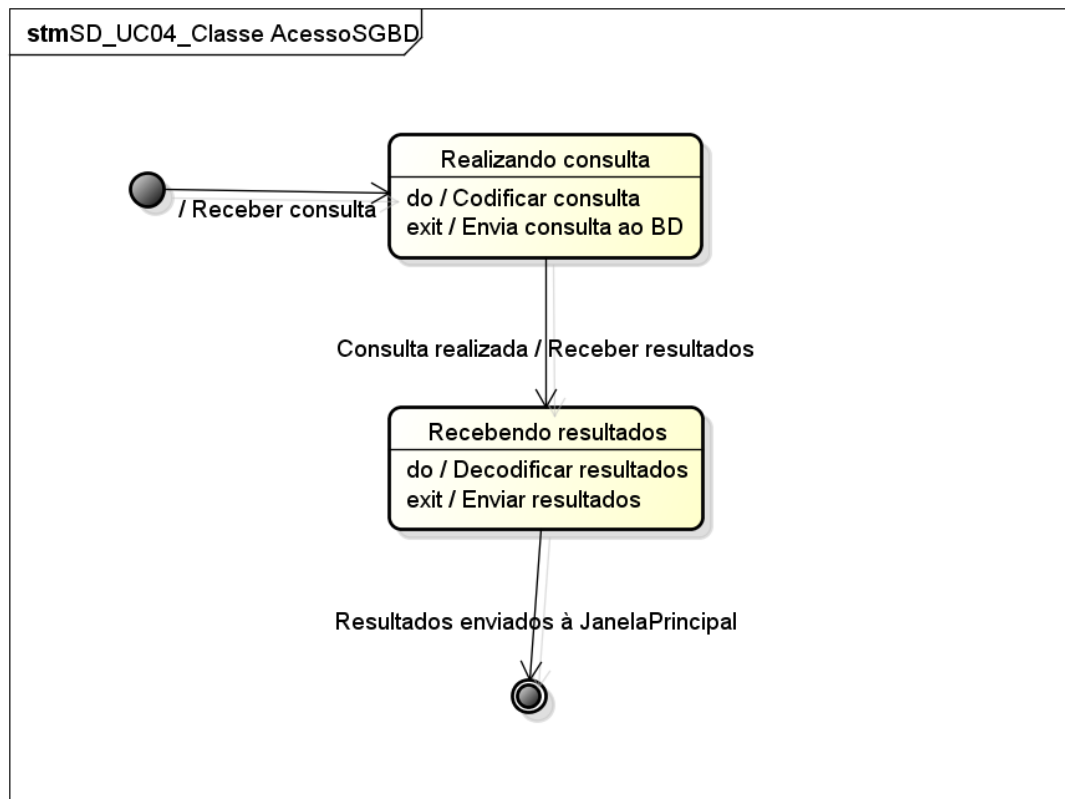


powered by Astah

Figura 22: Diagrama de Estados UC03: Salvar dados da conexão - Classe Server. Fonte: Autor.

## 10.4 UC04: Consultar banco de dados

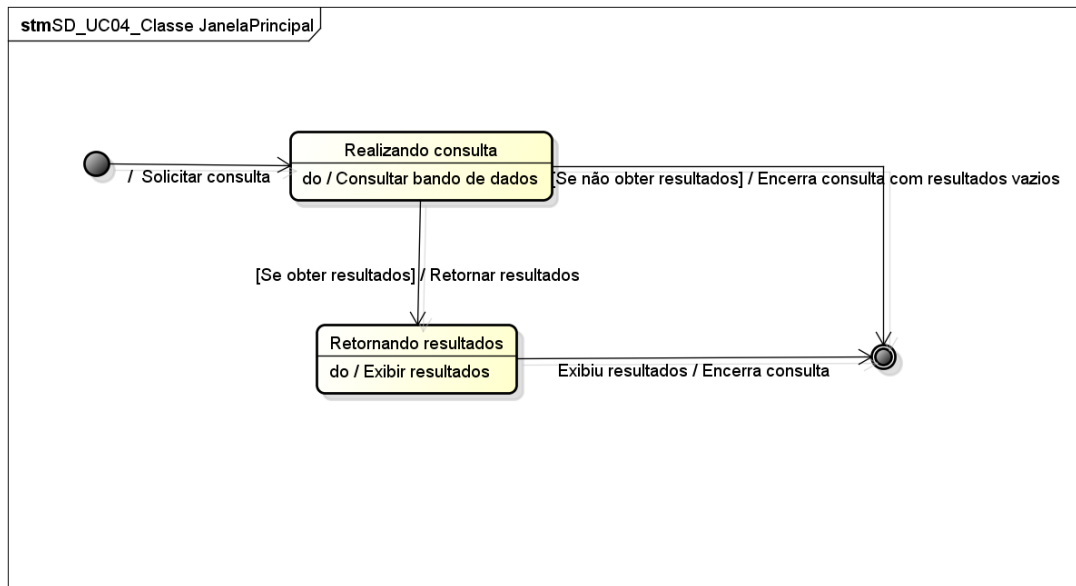
### 10.4.1 Classe AcessoSGBD



powered by Astah

Figura 23: Diagrama de Estados UC04: Consultar banco de dados - Classe AcessoSGBD. Fonte: Autor.

## 10.4.2 Classe JanelaPrincipal



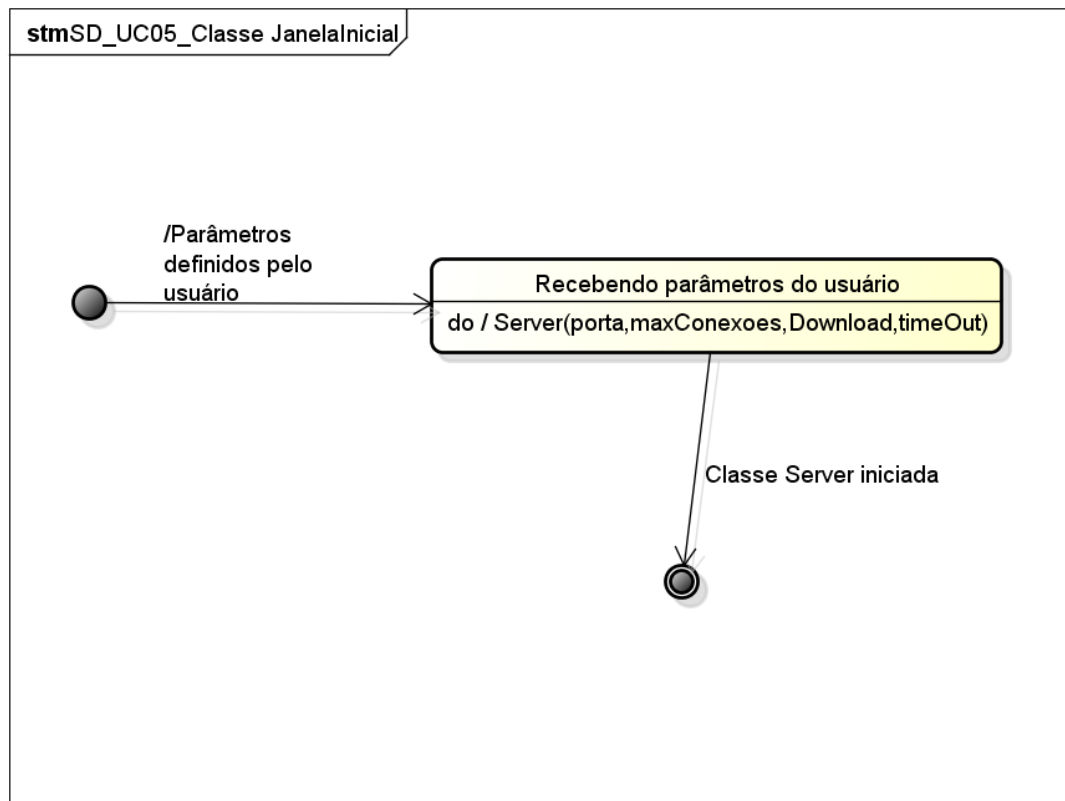
powered by Astah

Figura 24: Diagrama de Estados UC04: Consultar banco de dados - Classe JanelaPrincipal.  
Fonte: Autor.



## 10.5 UC05: Iniciar servidor

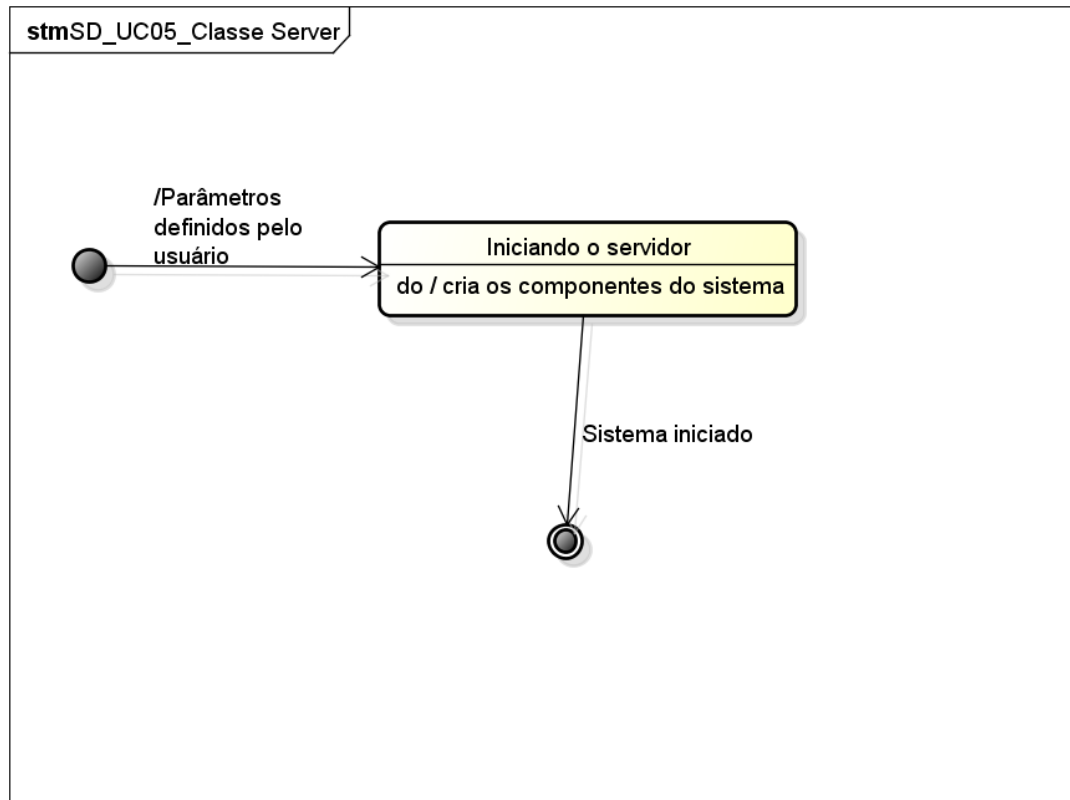
### 10.5.1 Classe JanelaInicial



powered by Astah

Figura 25: Diagrama de Estados UC05: Iniciar servidor - Classe JanelaInicial. Fonte: Autor.

## 10.5.2 Classe Server

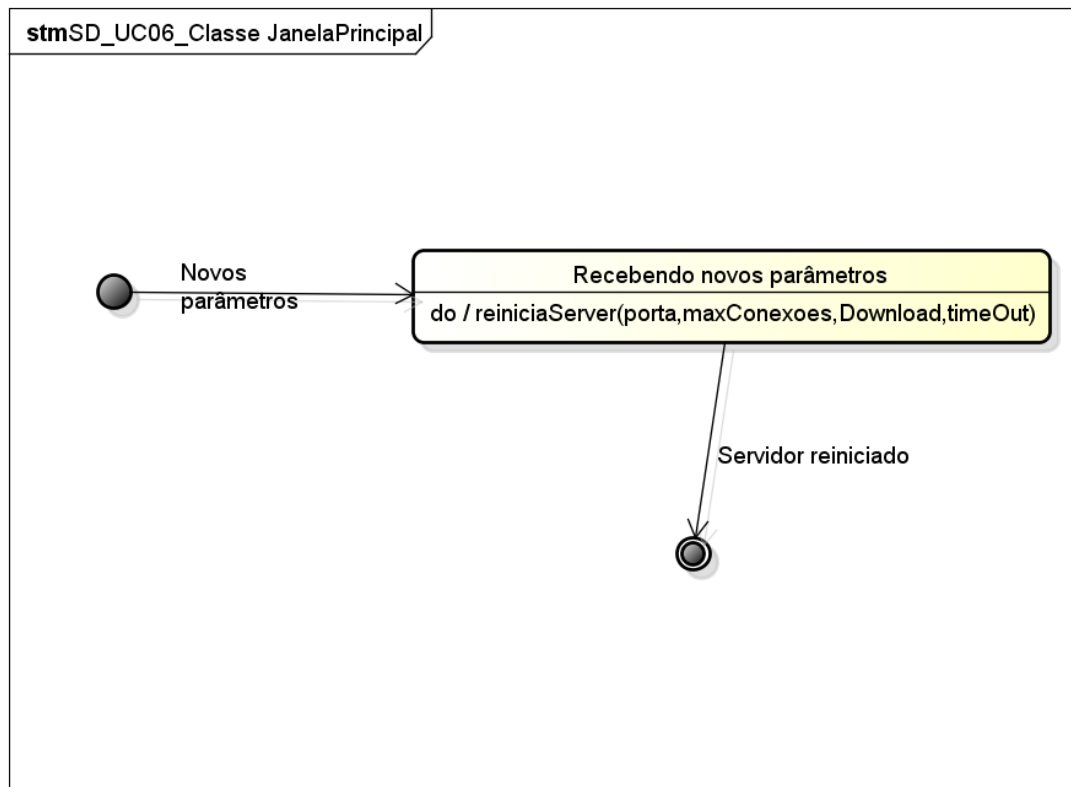


powered by Astah

Figura 26: Diagrama de Estados UC05: Iniciar servidor - Classe Server. Fonte: Autor.

## 10.6 UC06: Reiniciar servidor

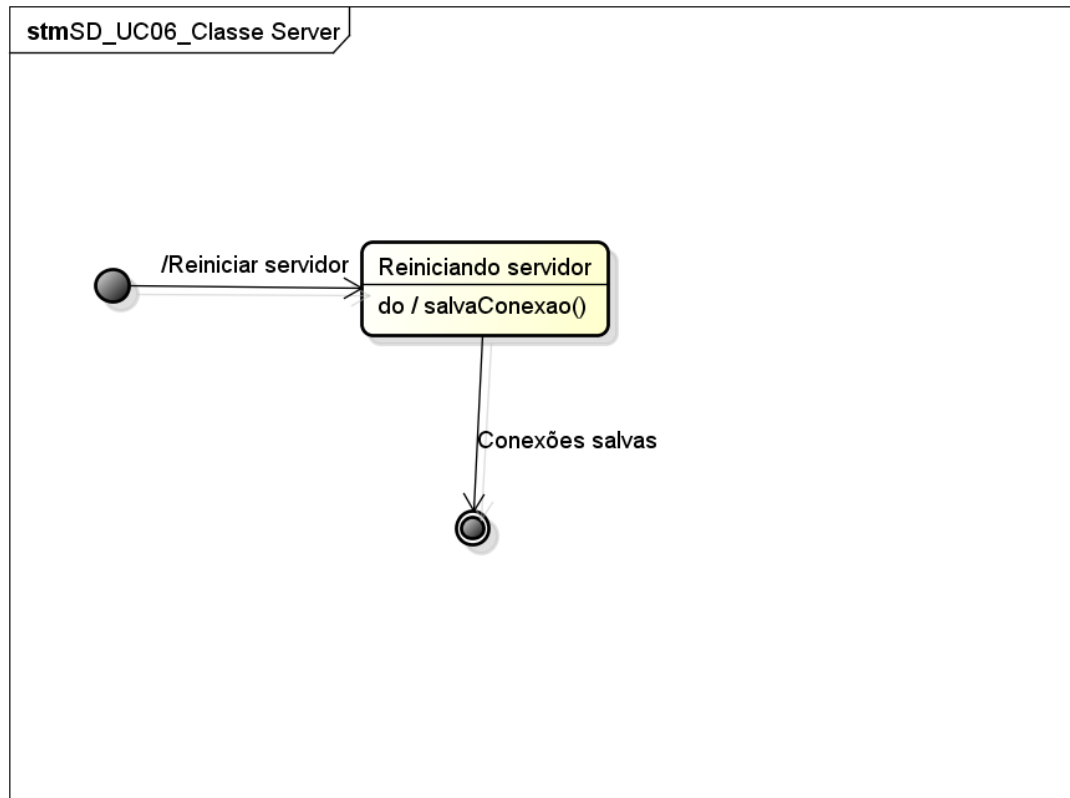
### 10.6.1 Classe JanelaPrincipal



powered by Astah

Figura 27: Diagrama de Estados UC06: Reiniciar servidor - Classe JanelaPrincipal. Fonte: Autor.

## 10.6.2 Classe Server



powered by Astah

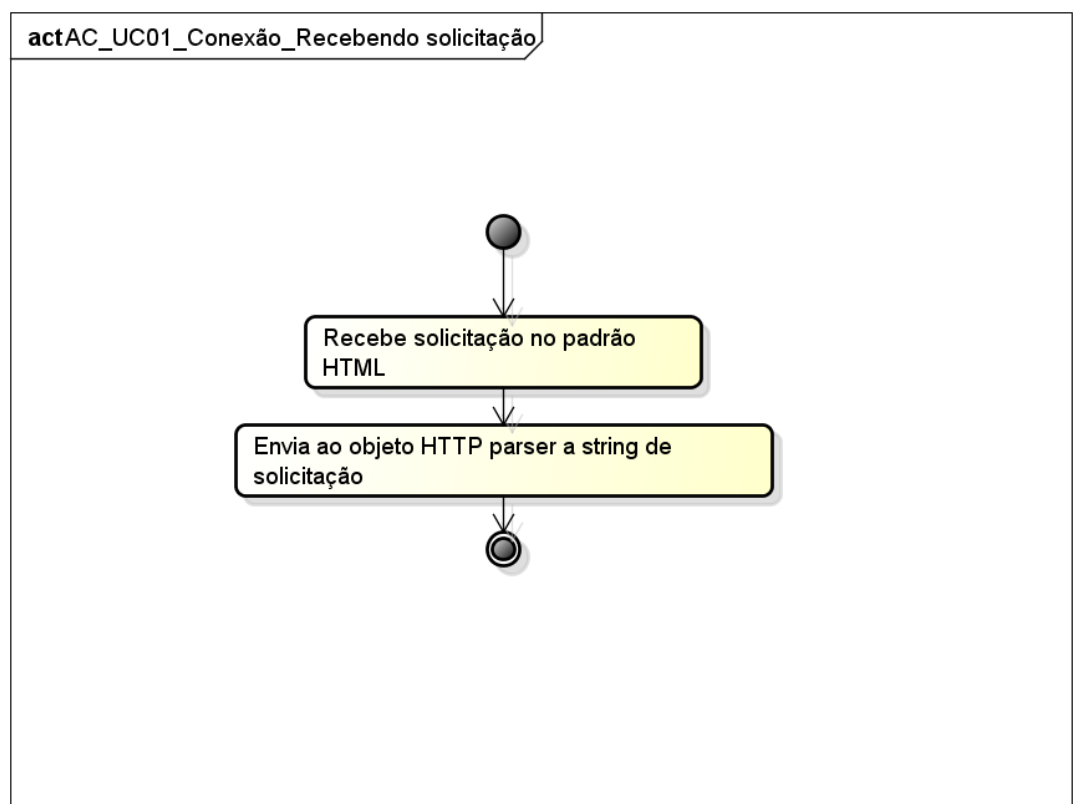
**Figura 28: Diagrama de Estados UC06: Reiniciar servidor - Classe JanelaPrincipal. Fonte: Autor.**

## 11 DIAGRAMA DE ATIVIDADES

Nesta seção serão apresentados Diagramas de Atividades que compõem o projeto. Será mostrado um diagrama para cada Diagrama de Estados presente na seção anterior, para um único estado de sistema que o compõe. Este tipo de diagrama tem objetivo de indicar os passos realizados por um objeto em um determinado estado.

### 11.1 UC01: Realizar conexão

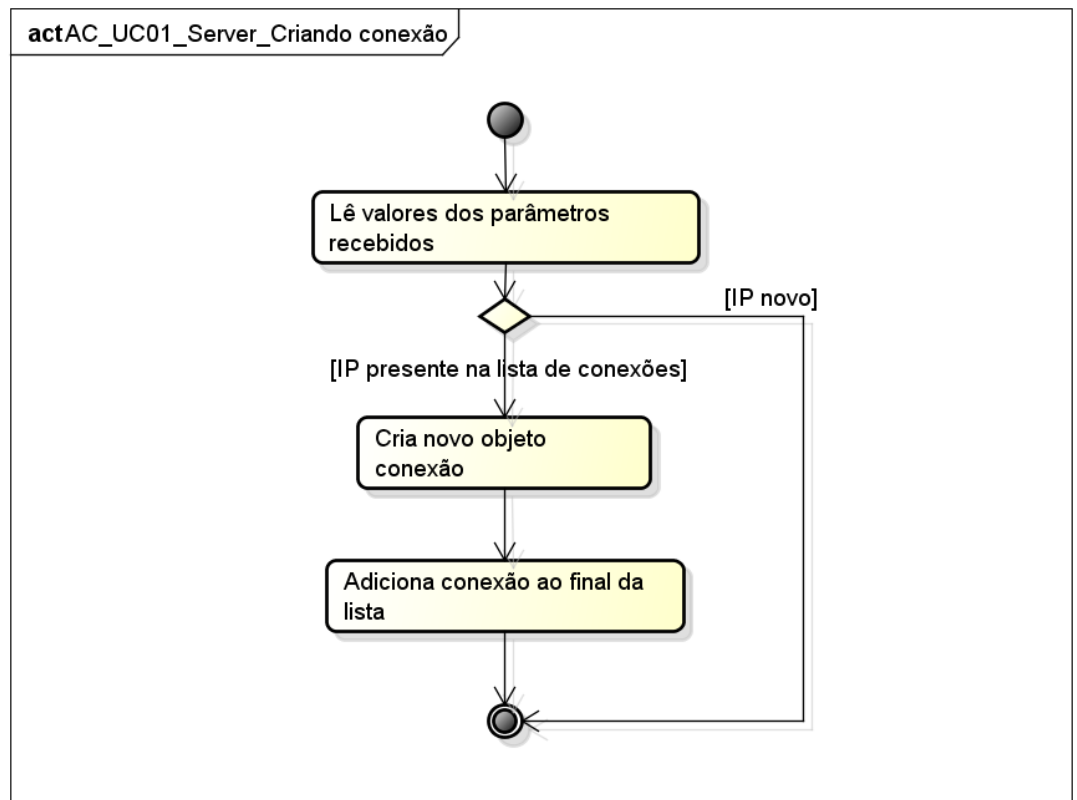
#### 11.1.1 Classe Conexao: Recebendo solicitação



powered by Astah

Figura 29: Diagrama de Atividades UC01: Realizar conexão - Classe Conexao – Recebendo solicitação. Fonte: Autor.

### 11.1.2 Classe Server: Criando conexão

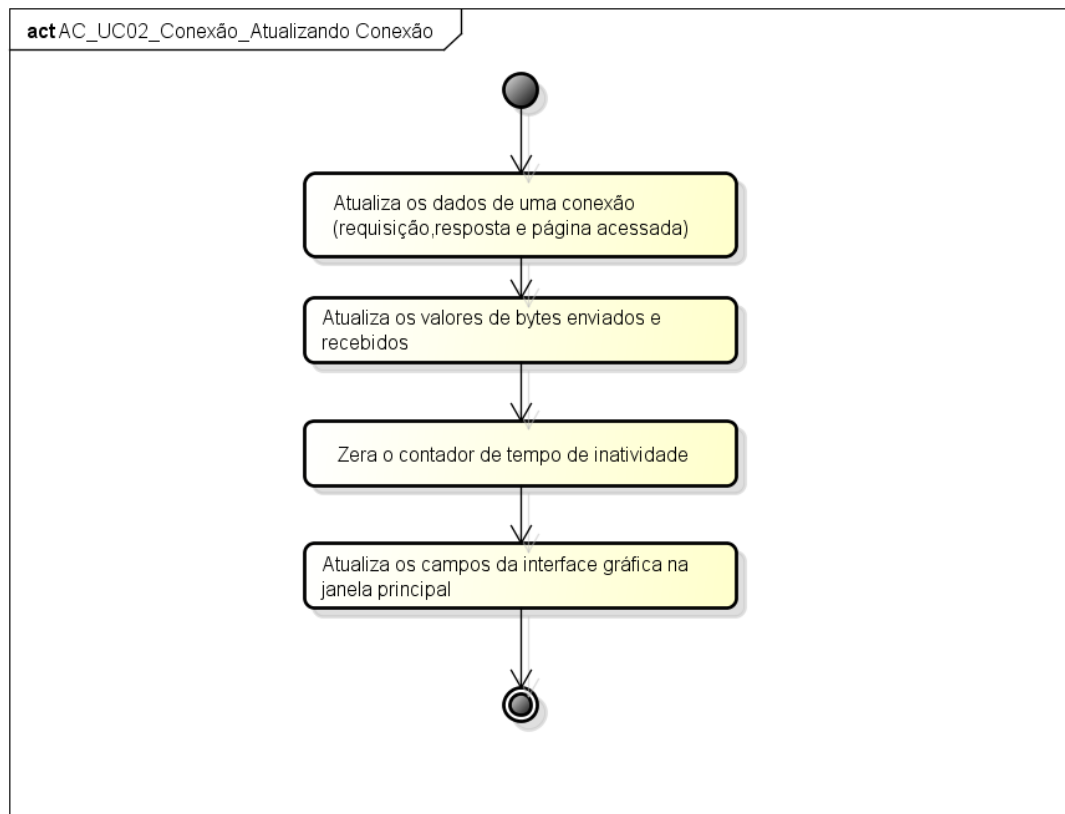


powered by Astah

**Figura 30: Diagrama de Atividades UC01: Realizar conexão - Classe Server – Criando conexão. Fonte: Autor.**

## 11.2 UC02: Mostrar informações dos dados transmitidos

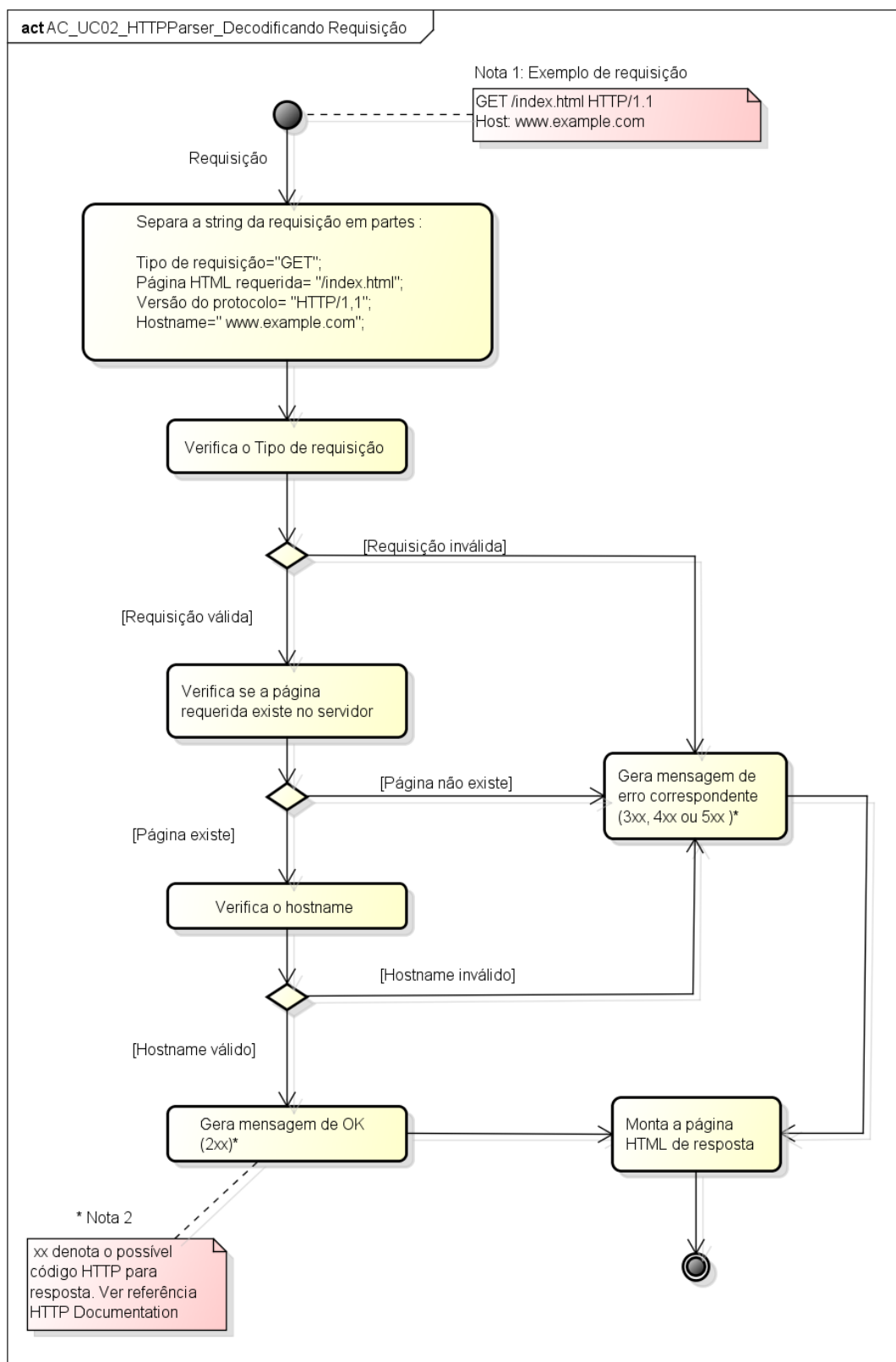
### 11.2.1 Classe Conexao: Atualizando conexão



powered by Astah

**Figura 31: Diagrama de Atividades UC02: Mostrar informações dos dados transmitidos - Classe Conexao – Atualizando conexão. Fonte: Autor.**

## 11.2.2 Classe HTTPParser: Decodificando requisição



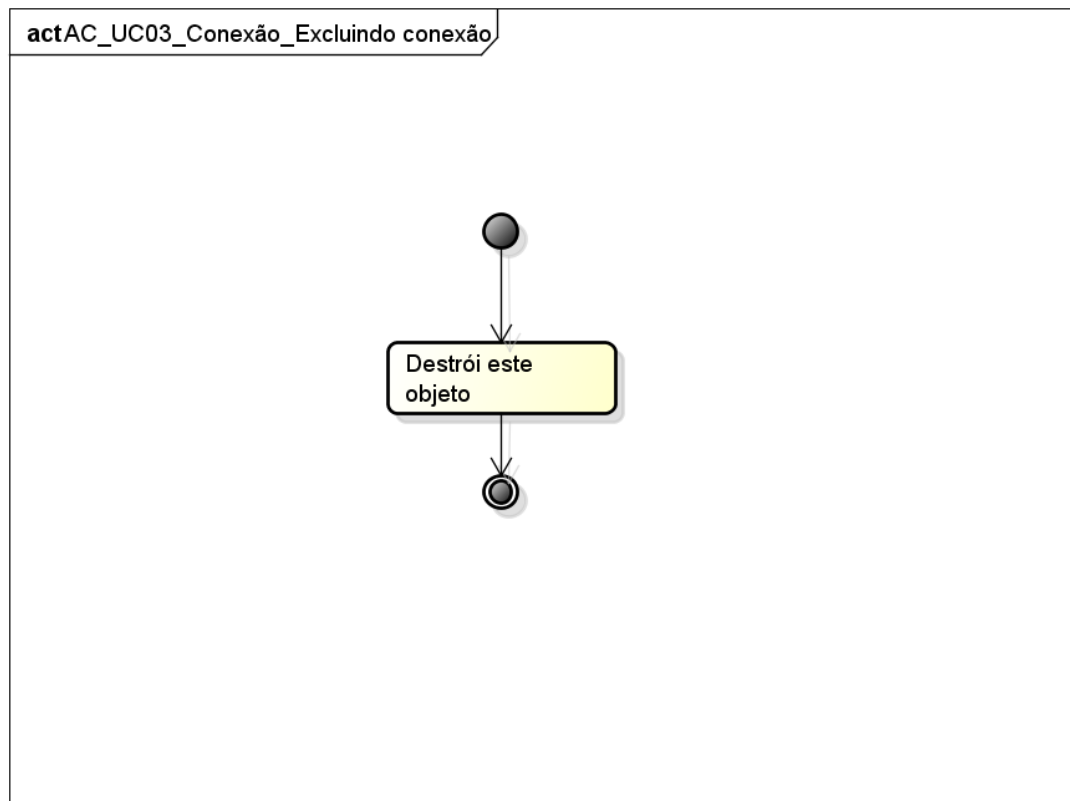
powered by Astah

Figura 32: Diagrama de Atividades UC02: Mostrar informações dos dados transmitidos - Classe HTTPParser – Decodificando requisição. Fonte: Autor.



## 11.3 UC03: Salvar dados da conexão

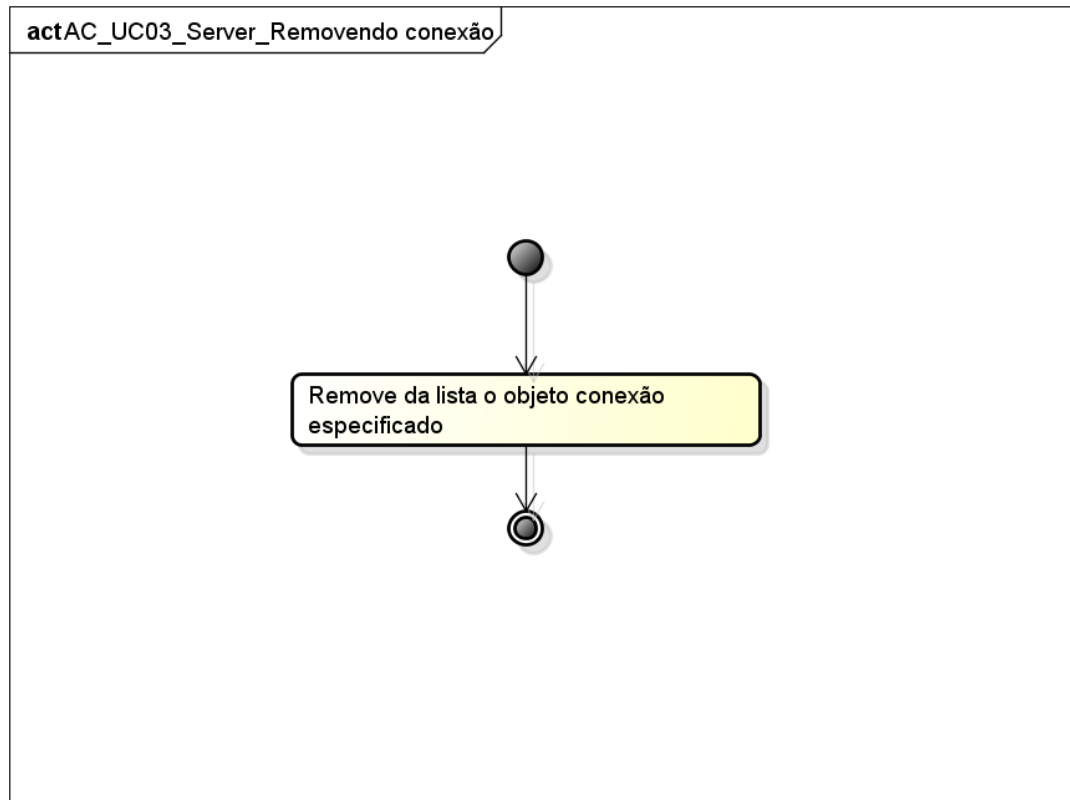
### 11.3.1 Classe Conexao: Excluindo conexão



powered by Astah

Figura 33: Diagrama de Atividades UC03: Mostrar informações dos dados transmitidos - Classe HTTPParser – Decodificando requisição. Fonte: Autor.

### 11.3.2 Classe Server: Removendo conexão

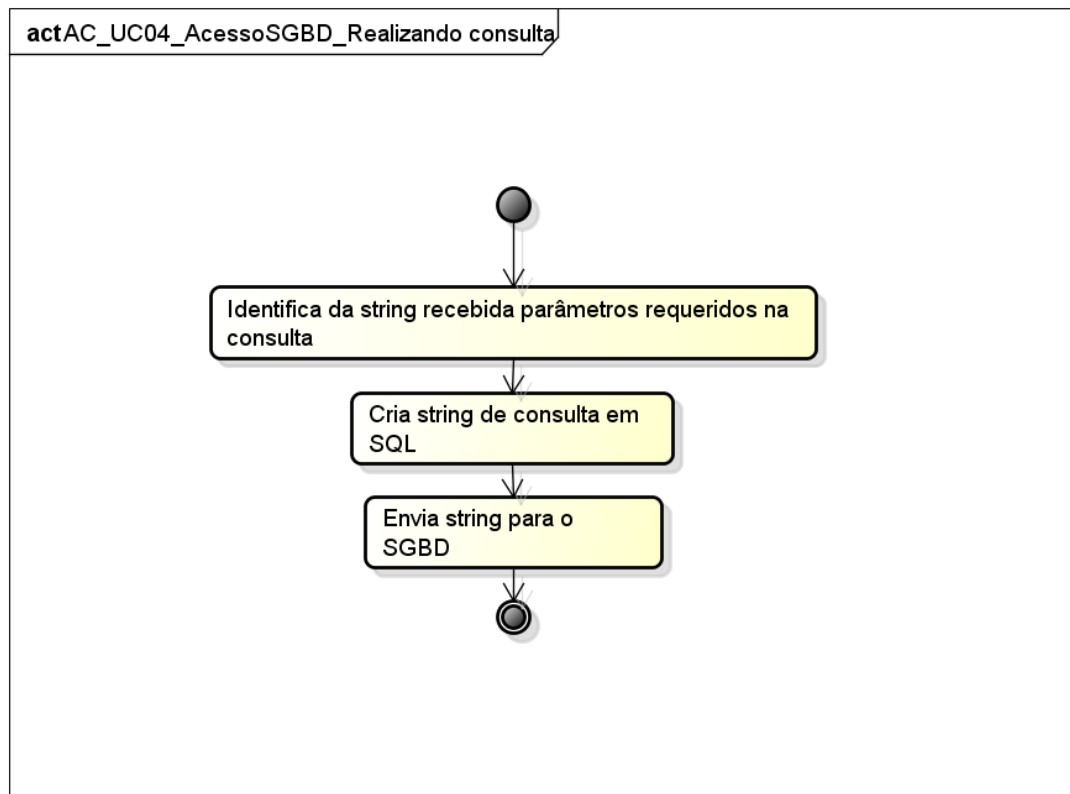


powered by Astah

Figura 34: Diagrama de Atividades UC03: Mostrar informações dos dados transmitidos – Classe Server – Removendo conexão. Fonte: Autor.

## 11.4 UC04: Consultar banco de dados

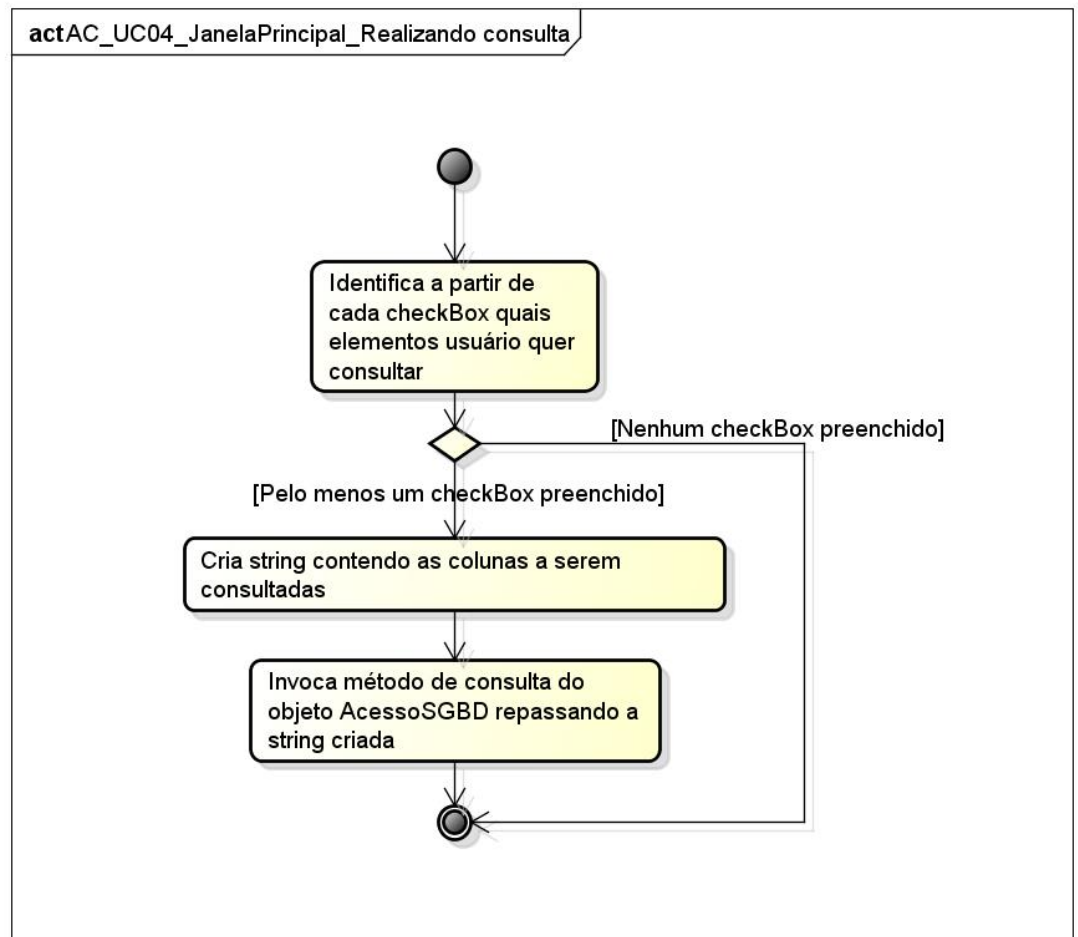
### 11.4.1 Classe AcessoSGBD: Realizando consulta



powered by Astah

Figura 35: Diagrama de Atividades UC04: Consultar banco de dados – Classe Acesso SGBD – Realizando consulta. Fonte: Autor.

### 11.4.2 Classe JanelaPrincipal: Realizando consulta

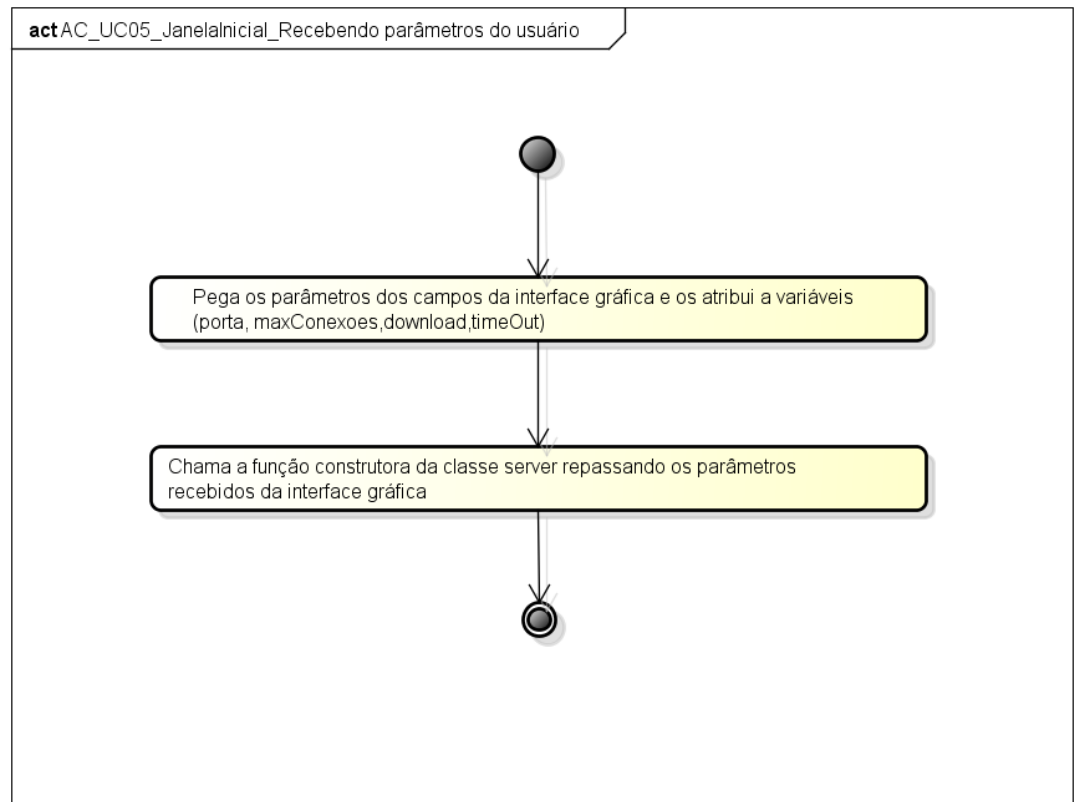


powered by Astah

Figura 36: Diagrama de Atividades UC04: Consultar banco de dados – Classe JanelaPrincipal – Realizando consulta Fonte: Autor.

## 11.5 UC05: Iniciar servidor

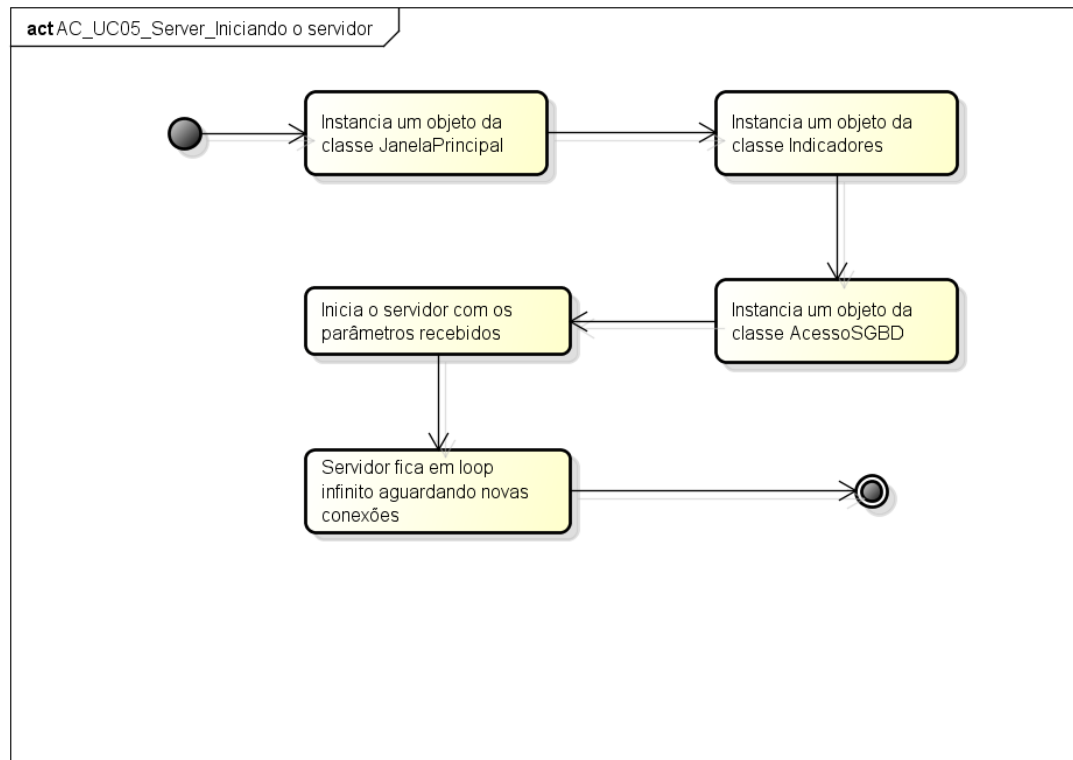
### 11.5.1 Classe JanelaInicial: Recebendo parâmetros do usuário



powered by Astah

**Figura 37: Diagrama de Atividades UC05: Iniciando servidor – Classe Janela Inicial – Recebendo parâmetros do usuário. Fonte: Autor.**

## 11.5.2 Classe Server: Iniciando o servidor

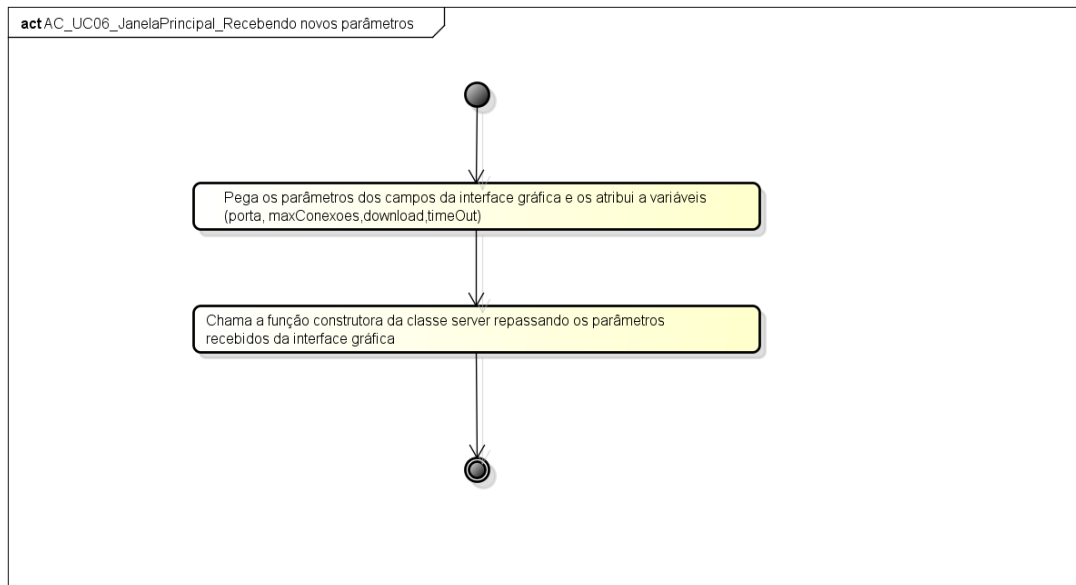


powered by Astah

**Figura 38: Diagrama de Atividades Diagrama de Atividades UC05: Iniciando servidor – Classe Server – Iniciando servidor. Fonte: Autor.**

## 11.6 UC06: Reiniciar servidor

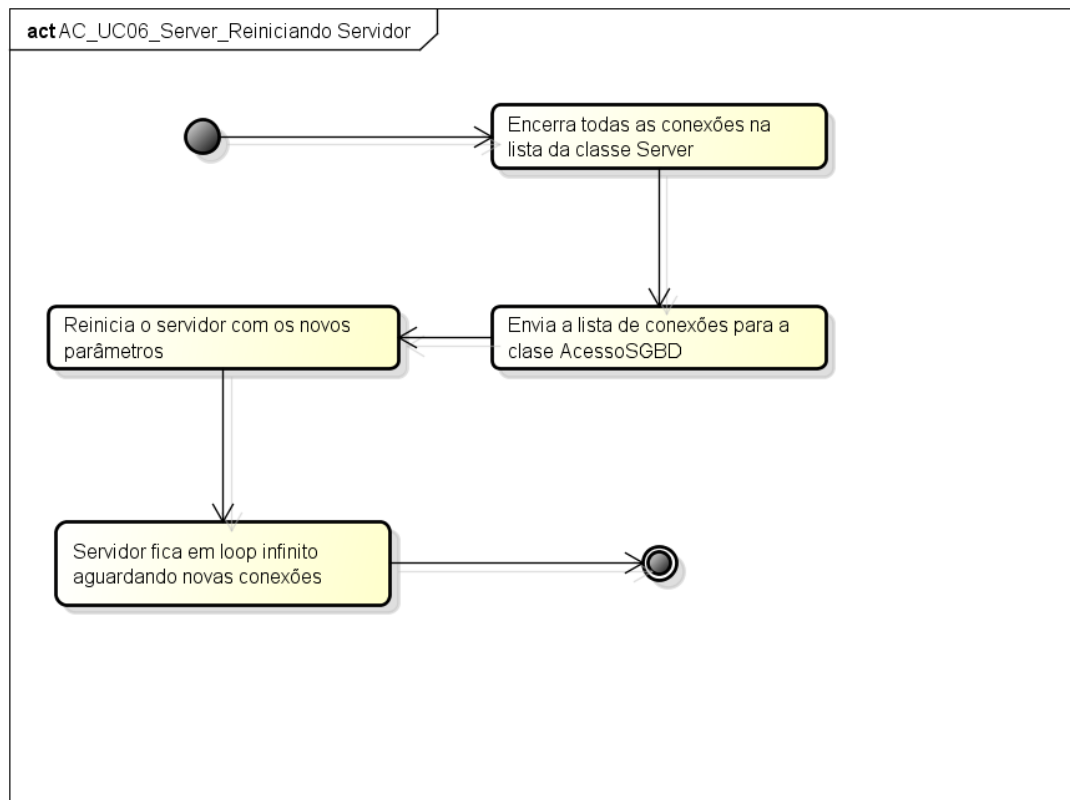
### 11.6.1 Classe JanelaPrincipal: Recebendo novos parâmetros



powered by Astah

**Figura 39: Diagrama de Atividades Diagrama de Atividades UC06: Reiniciando servidor – Classe JanelaPrincipal – Recebendo novos parâmetros. Fonte: Autor.**

## 11.6.2 Classe Server: Reiniciando servidor



powered by Astah

**Figura 40: Diagrama de Atividades Diagrama de Atividades UC06: Reiniciando servidor – Classe Server – Reiniciando Servidor. Fonte: Autor.**



## 12 CONCLUSÕES

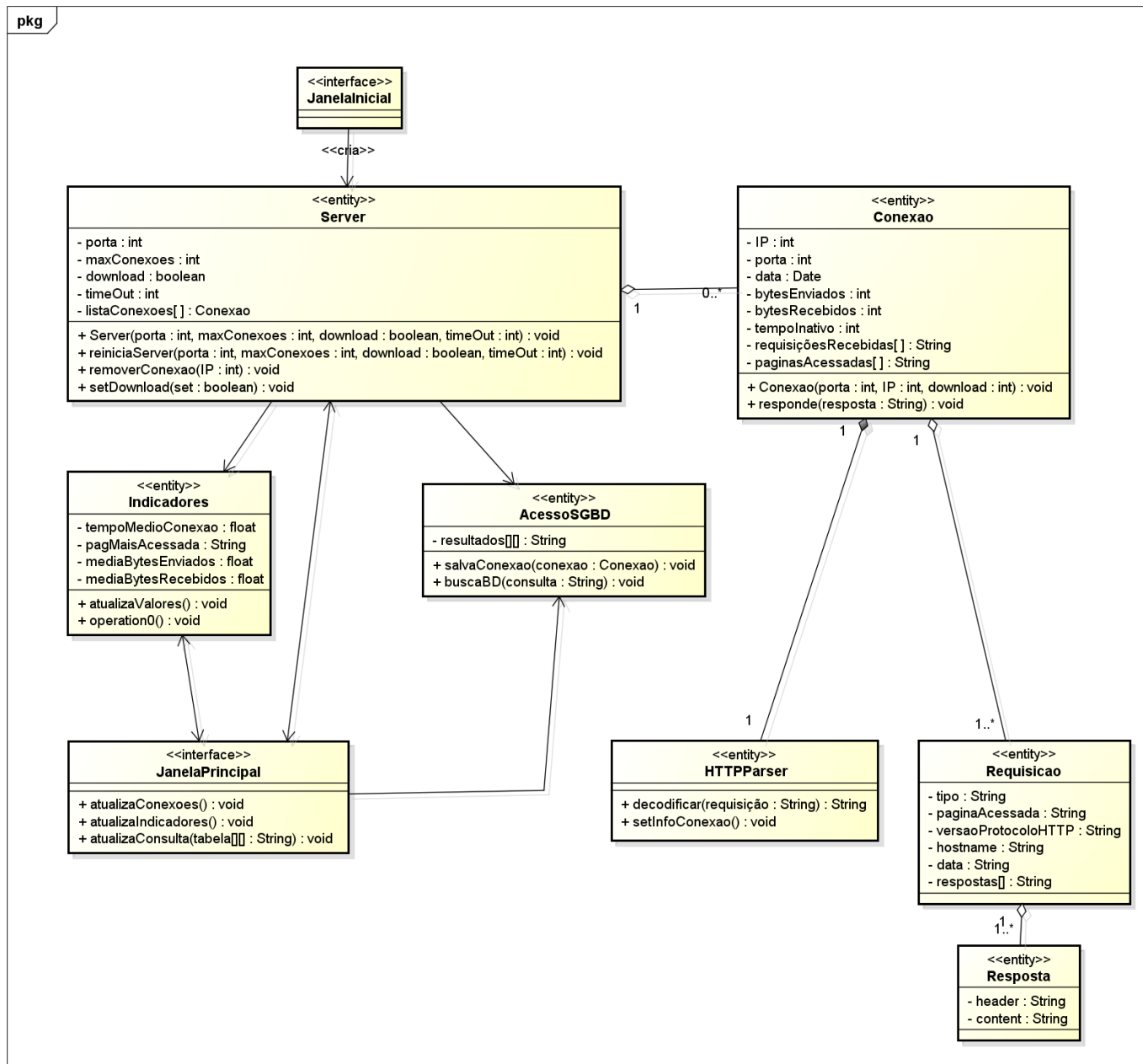
Levando em consideração que o projeto ainda não está concluído (ainda restam alguns bugs para serem corrigidos) não é possível concluir se os objetivos específicos foram atingidos. Entretanto, a implementação seguiu exatamente os modelos desse documento, então é altamente provável que os objetivos sejam cumpridos. Pois, como dito anteriormente na seção de levantamento de requisitos, cada uma das partes do sistema é um problema bem definido e resolvido na comunidade de software atual. Logo, cada objetivo específico do projeto já foi atingido separadamente, restando apenas o desafio de integrar essas partes em um sistema completo para alcançar um resultado satisfatório. O objetivo geral de criar um servidor web que mostra as informações de seu funcionamento foi atingido, pois conseguimos realizar conexões de teste com sucesso e extrair algumas informações do processo. Agora resta apenas corrigir os bugs restantes para permitir que o sistema tenha todas as funcionalidades descritas nos documento.

## 13 REFERÊNCIAS

Exame, R. (2014). Acesso em 12 de 05 de 2014, disponível em <http://exame.abril.com.br/carreira/ferramentas/tabela-de-salarios-rh/?empres>

## 14 APÊNDICES:

### 14.1 DIAGRAMA DE CLASSES



## 14.2 DIAGRAMA DE OBJETOS

