

Versão 1.1

**Controle de Versão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Razões para alteração** | **Responsável** |
| 1.0 | 21/03/2015 | Versão inicial | André e Douglas |
| 1.1 | 02/04/2015 | Ajuste caso de uso, regras para arquivos e ajustes gerais | André e Douglas |

**Envolvidos na elaboração do Documento de Arquitetura**

|  |  |
| --- | --- |
| **Nome** | **Área** |
| André Luiz dos Santos | Analista de TI |
| Douglas Alexandre da Silva | Analista de TI |

**Índice**

[1. Introdução 5](#_Toc415736057)

[1.1. Objetivo do Documento 5](#_Toc415736058)

[1.2. Escopo 5](#_Toc415736059)

[1.3. Siglas, Abreviações e Acrônimos 6](#_Toc415736060)

[1.4. Referências 8](#_Toc415736061)

[2. Composição da Arquitetura 9](#_Toc415736062)

[2.1. Tecnologias 9](#_Toc415736063)

[2.2. Ferramentas 10](#_Toc415736064)

[2.3. Componentes corporativos 11](#_Toc415736065)

[2.4. Frameworks 11](#_Toc415736066)

[2.5. Padrões do projeto 12](#_Toc415736067)

[3. Metas e Restrições da Arquitetura 12](#_Toc415736068)

[3.1. Tecnologia de desenvolvimento 12](#_Toc415736069)

[3.2. Servidor de Banco de dados 12](#_Toc415736070)

[3.3. Protocolo de rede 12](#_Toc415736071)

[3.4. Padrões corporativos 12](#_Toc415736072)

[3.5. Segurança 12](#_Toc415736073)

[3.6. Ambiente de desenvolvimento 13](#_Toc415736074)

[3.7. Geração de relatórios 13](#_Toc415736075)

[3.8. Armazenamento e processamento de imagens 13](#_Toc415736076)

[3.9. Prototipação 14](#_Toc415736077)

[4. Visão de Casos de Uso 21](#_Toc415736078)

[4.1. Fazer Login no Sistema 21](#_Toc415736079)

[4.2. Manter Departamento 22](#_Toc415736080)

[4.3. Manter Gestor 22](#_Toc415736081)

[4.4. Manter Funcionário 22](#_Toc415736082)

[4.5. Manter Categoria 23](#_Toc415736083)

[4.6. Manter Documento 23](#_Toc415736084)

[4.7. Carregar Departamentos 23](#_Toc415736085)

[4.8. Carregar Categorias 23](#_Toc415736086)

[4.9. Emitir Relatórios de Documentos 23](#_Toc415736087)

[5. Visão Lógica 25](#_Toc415736088)

[5.1. Visão Geral 25](#_Toc415736089)

[5.2. Nomenclatura da Aplicação 27](#_Toc415736090)

[6. Visão de Implementação 27](#_Toc415736091)

[6.1. Componentes e frameworks a serem reutilizados 27](#_Toc415736092)

[6.2. Componentes e frameworks a serem criados 28](#_Toc415736093)

[6.3. Tratamento de Erros e Exceções 28](#_Toc415736094)

[6.3.1. Erros e exceções do sistema 28](#_Toc415736095)

[6.3.2. Erros e exceções do usuário 28](#_Toc415736096)

[7. Visão de Segurança 29](#_Toc415736097)

[8. Visão de Implantação 30](#_Toc415736098)

[8.1. Visão Geral 30](#_Toc415736099)

[8.2. Servidor web 31](#_Toc415736100)

[8.3. Servidor de Banco de Dados 31](#_Toc415736101)

[9. Tamanho e Desempenho 32](#_Toc415736102)

# Introdução

## Objetivo do Documento

Este documento apresenta uma visão geral abrangente da arquitetura do projeto Sistema de Informatização de Documento, gerado pela área de Tecnologia de Informação, utilizando uma série de visões arquiteturais diferentes para ilustrar seus diversos aspectos.

Sua intenção é capturar e transmitir as decisões significativas do ponto de vista da arquitetura que foram tomadas em relação ao projeto.

## Escopo

Este projeto teve origem da observação do ambiente de trabalho e das dificuldades identificadas no processo descrito abaixo.

O processo de cadastramento e armazenamento de documentação corporativa é manual, lento e necessita de uma área relativamente grande para manter armários e pastas físicas. No modelo atual, alguns setores têm um volume de documentos extremamente alto, chegando a algumas centenas de processos e alguns milhares de documentos arquivados no total. A empresa não tem um número preciso de documentos arquivados devido à impossibilidade de indexação dos mesmos. Os documentos que estão ligados a processos estão registrados em planilhas de EXCEL, com data de cadastro e breve descrição dos processos. Neste registro é cadastrada uma identificação de localização no arquivo físico que indica a gaveta onde o documento está armazenado.

Documentos que não estão ligados a processos, que servem como referência ou apenas para constar em relatórios do departamento, como protocolos, atestados médicos, comprovantes bancários ou mesmo algumas correspondências, são armazenados no arquivo, por ordem alfabética sem ser categorizados.

Somente funcionários do mesmo setor podem acessar os documentos por este setor registrados.

O projeto não prevê integração com outros sistemas.

## Siglas, Abreviações e Acrônimos

**Apache Software Foundation - ASF:** Denominada *Apache Foundation* ou Fundação Apache é uma organização sem fins lucrativos responsável por administrar e dsponibilizar projetos de código aberto relacionado o Servidor Web Apache.

**Browser**: Browser é o navegador de internet utilizado pelo usuário para acessar as páginas dos sites. Os principais navegadores são: Internet Explorer, Chrome, Firefox, Opera, IOS e Android.

**CSS:** *Cascade Style Sheets,* para padronização de estilos.

**Design Pattern:** Solução de projeto de implementação desenvolvida por terceiros, que pode ser reaproveitada em vários projetos.

**EXCEL:** É um editor de planilhas produzido pela Microsoft utilizado amplamente por sua capacidade de fazer cálculos, ter uma interface intuitiva e contrução de gráficos.

**Hibernate:** É um framework para o mapeamento objeto-relacional escrito na linguagem Java.

**HTML:** *Hiper Text Markup Language*. Linguagem para marcação de texto usa em páginas da internet.

**HTTP:** Significa *Hypertext Transfer Protocol ou* Protocolo de Transferência de Hipertexto. É o protocolo de comunicação que transfere dados por camadas, servindo como base para a comunicação pela internet.

**IDE:** *Integrated Development Environment* Ambiente Integrado de Desenvolvimento.

**JavaScript:** É uma linguagem de programação interpretada pelo browser utilizada amplamente para tornar os sites mais dinâmicos sem a necessidade de processamento do servidor web.

**JDBC:** Java Database Connectivity é um conjunto de classes e interfaces (API) escritas em Java que faz o envio de instruções SQL para qualquer banco de dados relacional.

**JQuery:** É uma biblioteca JavaScript suportada em diferentes browsers desenvolvida para simplificar os scripts que são interpretados na máquina do cliente que interagem com o HTML.

**JS:** Ver JavaScript.

**MVC:** *Model View Controller*. É o design pattern utilizado na camada de apresentação

**Segurança declarativa:** Cada recurso pode ser acessado por uma ou mais classes de regras. O usuário pode ter um ou mais papeis dentre um aplicativo. Isso é equivalente a estrutura organizacional de uma empresa.

**SGDB:** Sistema de Gerenciamento de Banco de Dados

**SIDOC:** É o nome escolhido para designar este projeto. Significa Sistema de Informatização de Documentos.

**UML:** *Unified Modeling Language*,. Linguagem de Modelagem Unificada.

**URL:** *Uniform Resource Locator* significa Localizador padrão de Recursos. É o endereçamento de uma página na rede, que pode ser local ou na internet.

**Web Service:** Servidor Web é responsável por armazenar os dados dos sites e disponibilizá-los na rede.

**WEB:** Sistema de comunicação e transmissão de dados redes baseado em HTML interligados e executados por meio de um browser.

**XML:** eXtensible Markup Language.

## Referências

1. Proposta preliminar, versão 1.0 de 24/02/2015 em:

https://github.com/alsantos10/doc\_info/tree/master/Documentacao/Analise/Atividade\_01\_Proposta\_Preliminar.docx.

1. Visão das necessidades, versão 1.0 de 04/03/2015 em:

https://github.com/alsantos10/doc\_info/tree/master/Documentacao/Analise/VisaoDasNecessidades.doc.

1. Documento de requisistos, versão 1.1 de 20/03/2015 em:

https://github.com/alsantos10/doc\_info/tree/master/Documentacao/Analise/DocumentoDeRequisitos.docx.

1. Prazos obrigatórios de guarda de documentos, de 2011 em:

https://github.com/alsantos10/doc\_info/tree/master/Documentacao/Analise/guarda\_documentos.pdf.

1. Bootstrap: http://www.getbootstrap.com.
2. Bootstrap Template: Progressus - Sergey Pozhilov: http://www.gettemplate.com.
3. Eclipse: http://www.eclipse.org
4. Datatables: https://www.datatables.net.
5. Documentação Javadoc: http://www.java.sun.com/j2se/javadoc.
6. GitHub: http://www.github.com.
7. Hibernate: http://objectgeneration.com.
8. Jaspersoft:http://jasperstudio.sourceforge.net.
9. JQuery: http:www.jquery.com.

# Composição da Arquitetura

## Tecnologias

Tabela - Tecnologias utilizadas

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Versão** | **Descrição** | **Utilização** |
| Annotation |  | Recurso da plataforma Java que fornece opção do uso de metadata ao longo do código que podem ser interpretadas por um compilador ou pré-compilador que poderá realizar alguma tarefa pré-definida. | Utilizada em todas as camadas. |
| J2EE | 1.4 | Plataforma com uma série de especificações bem detalhadas sobre cada serviço de infraestrutura para construção de um software com caracteristicas transacionais, multicamadas e modulares. | Toda a aplicação é baseada nesta tecnologia. |
| J2SE | 1.4 | Plataforma de desenvolvimento para a linguagem Java. Composto por maquina virtual, bibliotecas, interface gráfica etc. | Utilizada em toda a aplicação. |
| JQuery | 2.1 | Componente utilizado para simplificar scripts JavaScript e diminuir a incompatibilidade entre navegadores. | Visão. |
| JSP |  | *JavaServer Pages,* é a Tecnologia para aplicações Web. | Visão. |
| JSTL | 1.2 | *JavaServer Pages Standard Tag Library*, é um componente da plataforma de desenvolvimento web Java EE que disponibiliza ao JSP uma biblioteca de tags JSP para tarefas como processamento de dados XML, execução condicional, loops e internacionalização. | Visão. |
| MySQL | 3 | Servidor de banco de dados. | SGDB utilizado como repositório de dados. |

## Ferramentas

Tabela - Ferramentas

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Versão** | **Descrição** | **Utilização** |
| Apache |  | Servidor local para transmissão de dados via intranet. | Disponibiliza o acesso ao sistema via rede local. |
| Astah Community | 6.8 | Ferramenta utilizada para construção dos diagramas UML. | Construir os diagramas UML. |
| Eclipse | 4 ou superior | Proporciona o ambiente de desenvolvimento (IDE), com um editor de código e seus plugins. | Editor, depurador e gerador de códigos. |
| Firefox | 7 ou superior | Navegador de Internet produzido pela Mozilla Foundation. | Acesso a documentos de visualização no servidor web. |
| GitHub | 2.12 | Serviço de web hosting compartilhado para projetos que usam o controle de Sistema de controle de versão Git. | Possibilita o desenvolvimento concorrente do sistema por diferentes programadores simultaneamente. |
| Internet Explorer | 7 ou superior | Navegador de Internet produzido pela Microsoft | Acesso a documentos de visualização no servidor web. |
| Java JDK | 7 | Disponibiliza o ambiente de criação e execução das aplicações em Java. Inclui a máquina virtual Java (JVM), o compilador Java, as APIs do Java e outros utilitários. | Possibilita desenvolvimento de um software na plataforma Java. |
| MySQL | 3 | Disponibiliza um servidor de banco de dados. | Fornece o repositório de dados. |

## Componentes corporativos

Tabela 3 – Componentes corporativos

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Versão** | **Descrição** | **Utilização** |
| Util | 1.0 | Componente corporativo composto por classes base para outros componentes. | Classes utilitárias diversas. |
| Security | 1.0 | Componente corporativo centralizador para a utilização de autenticação, autorização e auditoria. | Autenticação, autorização e auditoria. |

## Frameworks

Tabela 4 - Frameworks utilizados

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Versão** | **Descrição** | **Utilização** |
| Bootstrap | 3.3 | Framework de desenvolvimento em HTML, CSS e JS de forma ágil para diferentes dispositivos. | Construir as páginas de navegação responsivas. |
| Hibernate | 4.2 | Framework para mapeamento objeto-relacional em Java. Utilizado para facilitar a transição de dados do modelo tradicional para objetos da aplicação por meio de *Annotations* para permitir esta relação. | Transformação das classes Java para tabelas de dados. |
| JasperReports | 6 | Framework Java para gerar relatórios em diversos formatos. | Gerar relatórios no formato PDF. |
| JSF | 2.2 | *JavaServer Faces* é um framework para construção de interfaces de usuário no padrão MVC em um servidor Java EE | Controle e Visão |
| Progressus |  | Template escolhido que utiliza o Bootstrap | Visão. |

## Padrões do projeto

Tabela – Descrição dos padrões de projeto

|  |  |
| --- | --- |
| **Nome** | **Utilização** |
| DAO – Data Acess Object | Todo acesso à base de dados ficará posicionado em classes que utilizam o padrão DAO. |
| MVC - Model View Controller | Direcionar a construção da interface gráfica da camada de apresentação. |

# Metas e Restrições da Arquitetura

## Tecnologia de desenvolvimento

* A arquitetura da aplicação deve ser baseada em Java EE.
* Construção dos componentes será baseada em camadas.

## Servidor de Banco de dados

* O servidor escolhido é o MySQL.

## Protocolo de rede

* O acesso ao sistema deverá ser realizado através da intranet (via protocolo HTTP).

## Padrões corporativos

* O sistema deverá contar com uma interface WEB.

## Segurança

* Será utilizada a segurança declarativa, por meio de autenticação, autorização de usuário (login/ senha) através do componente de segurança da aplicação.
* No caso de auditoria, o sistema utilizará o componente de segurança que fará registro de log.
* Todos os arquivos de documentos digitalizados serão armazenados somente pelo sistema e não poderão ser acessados diretamente pela URL do arquivo.

## Ambiente de desenvolvimento

* Codificação utilizando a IDE Eclipse.
* Linguagem Java.

## Geração de relatórios

* O sistema permitirá que cada responsável de departamento possa emitir relatórios de documentos cadastrados pela sua equipe por período ao fornecer obrigatoriamente data inicial e final. Caso deseje, esse relatório poderá ser filtrado por funcionário responsável e/ou por categoria.
* O sistema ainda permitirá que cada responsável de departamento possa emitir relatórios de documentos que já possam ser excluídos do arquivo físico, mediante informação de período inicial e final.
* Os relatórios de documentos em arquivo poderão ser gerados no formato XLS e/ou PDF.

## Armazenamento e processamento de imagens

* O sistema armazenará os dados de cada arquivo digitalizado referente a um documento.
* Cada registro de documento poderá conter mais de uma imagem digitalizada de documento caso o mesmo seja composto por mais de uma página.
* O sistema somente registrará arquivos de documentos no formato PDF ou JPG.
* Os arquivos de documentos registrados serão copiados para o sistema e armazenados em uma pasta de documentos.
* Cada arquivo de documento digitalizado armazenado no sistema não poderá ultrapassar o tamanho de 50 Megabytes.
* Para garantir que cada arquivo não seja gravado em duplicidade ou substitua outro já existente, cada arquivo adicionado ao sistema deverá ser renomeado utilizando a seguinte regra:
* O nome do arquivo será substituído pelo título do documento preenchido pelo usuário no momento do cadastro. Serão retirados todos os caracteres especiais e os espaços serão substituídos pelo caractere “\_” *underline*. O campo título do documento deverá ter no mínimo 3 caracteres. Todos os caracteres deverão ser convertidos para letra minúscula.
* Serão colocados seis dígitos na frente do nome no seguinte formato 000000, iniciando do número 1 da direita para a esquerda. Ou seja, o primeiro arquivo será precedido pelos seguintes dígitos 000001 e assim por diante.
* O nome do arquivo no final deverá ter no mínimo 9 caracteres, sendo 6 dígitos e 3 para o título do documento e no máximo 32 caracteres. Caso haja mais caracteres, o sistema deverá ignorá-los.
* A re-indexação das imagens deverá ser realizada diariamente após as 18:00 de cada dia.
* A transmissão de dados dos arquivos de documentos registrados será realizada via FTP ou HTTP.

## Prototipação

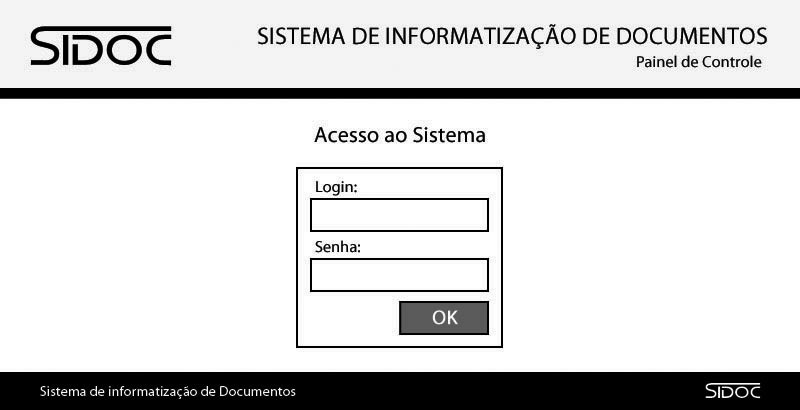


Figura - Tela de Login

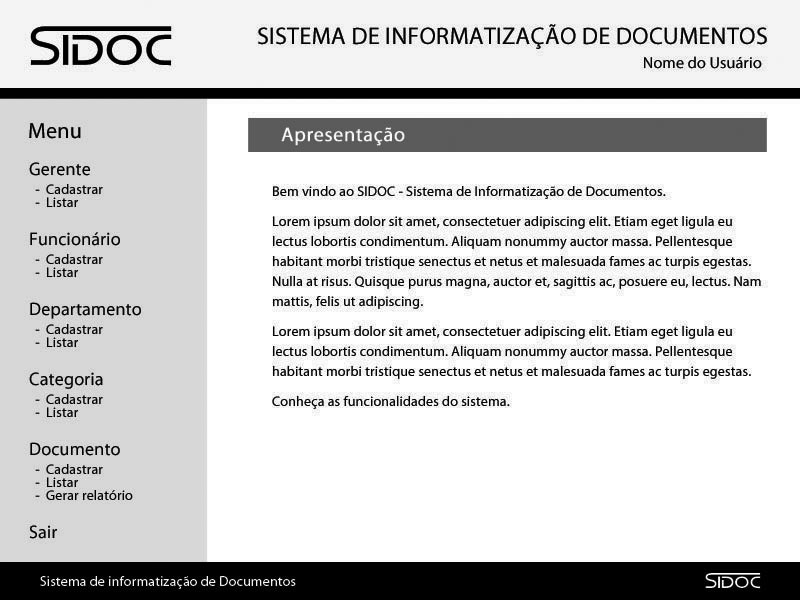


Figura - Tela de Apresentação - Página Inicial



Figura - Tela Lista de Gerentes

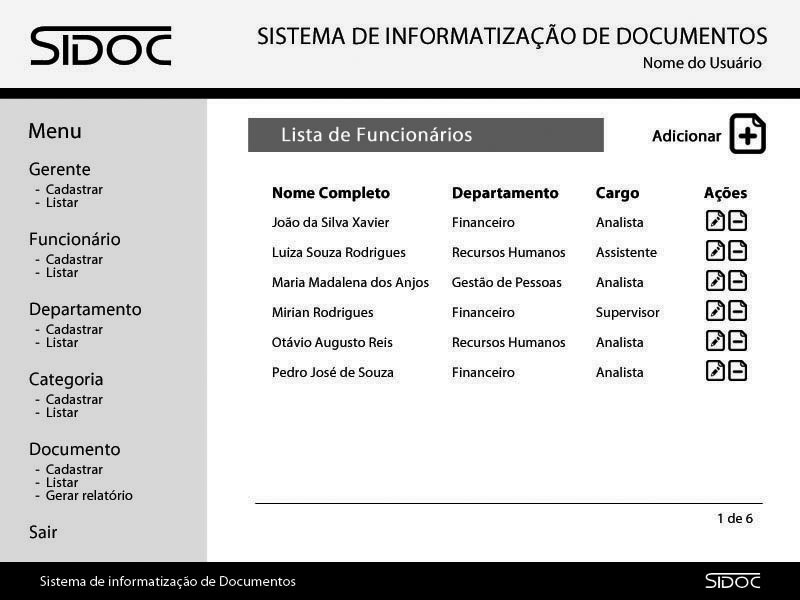


Figura - Tela Lista de Funcionários

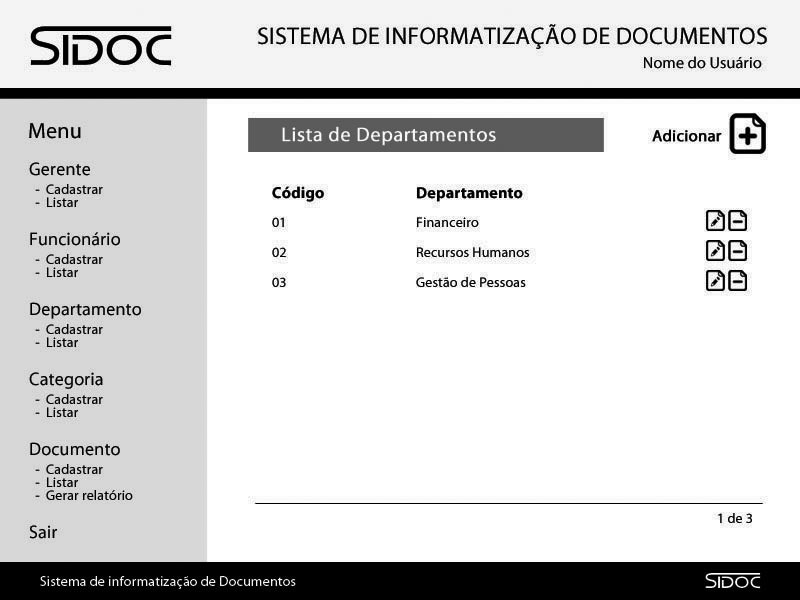


Figura - Tela Lista de Departamentos

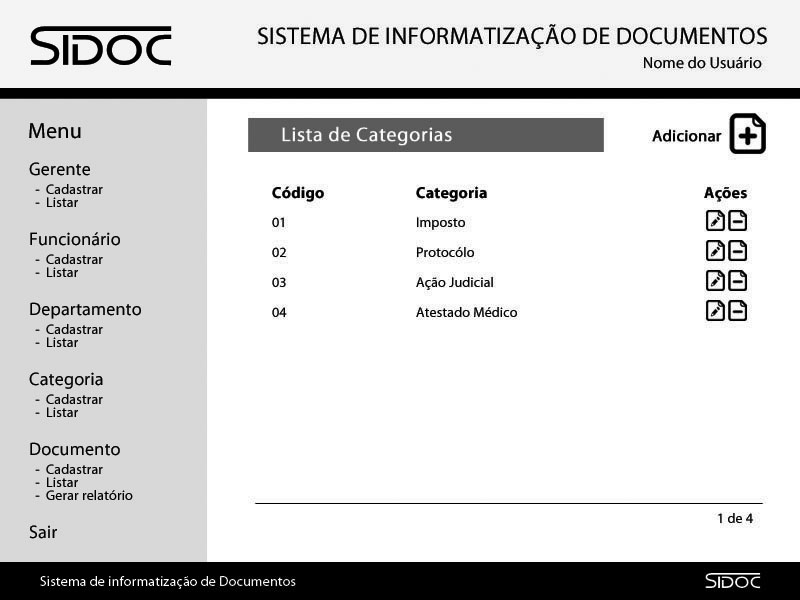


Figura - Tela Lista de Categorias

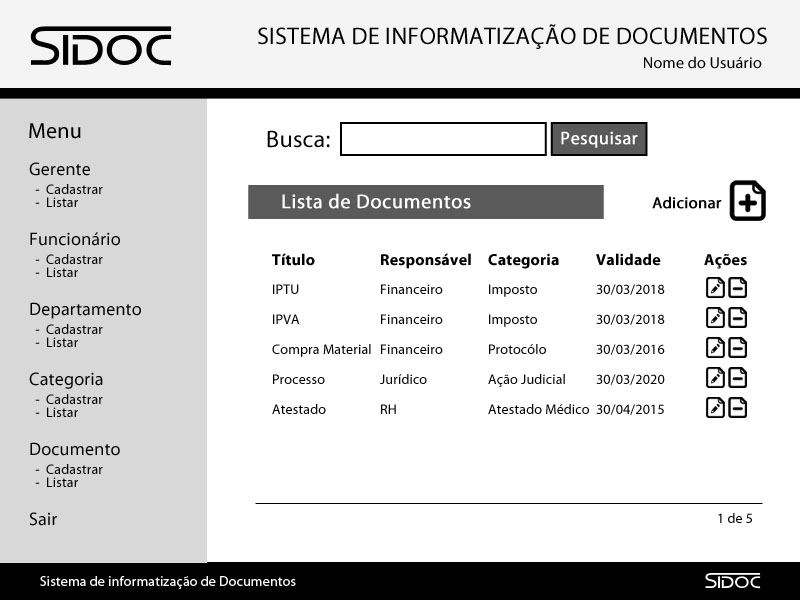


Figura - Tela Lista de Documentos

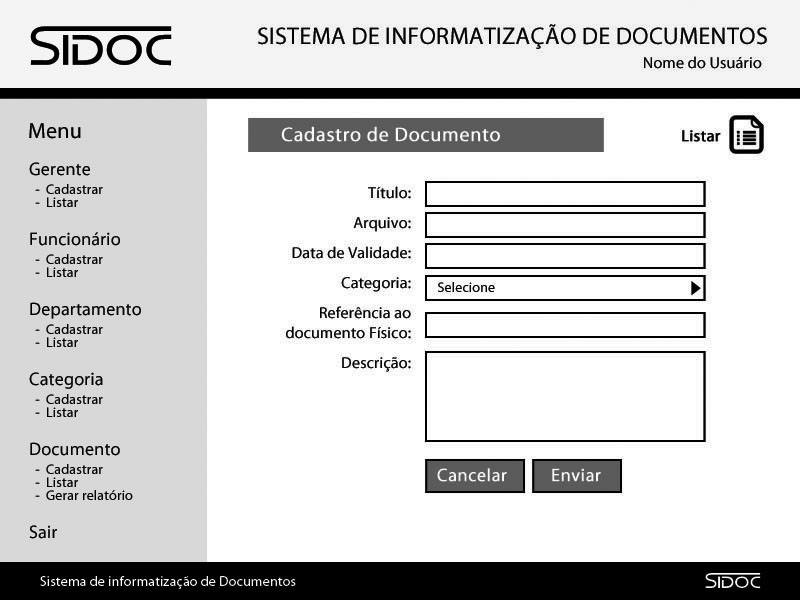


Figura - Tela Cadastrar/Alterar Documento

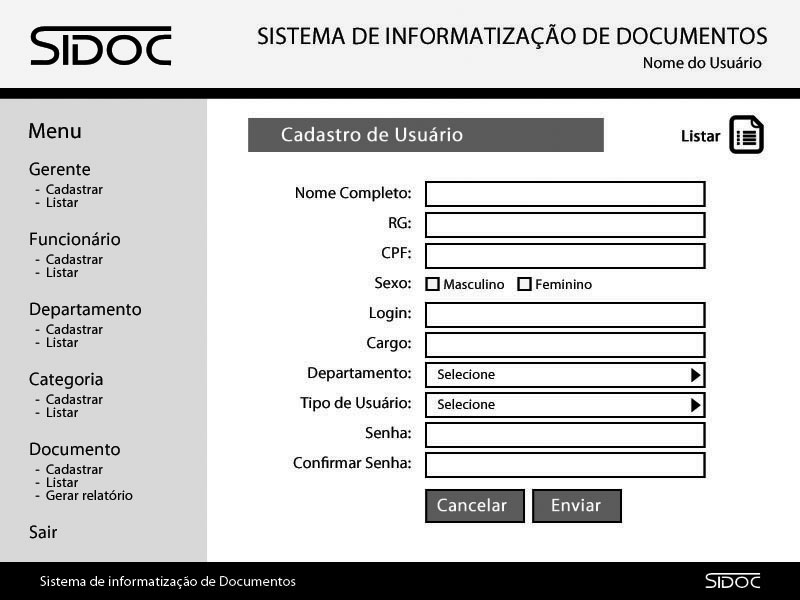


Figura - Tela Cadastrar/Alterar Usuário

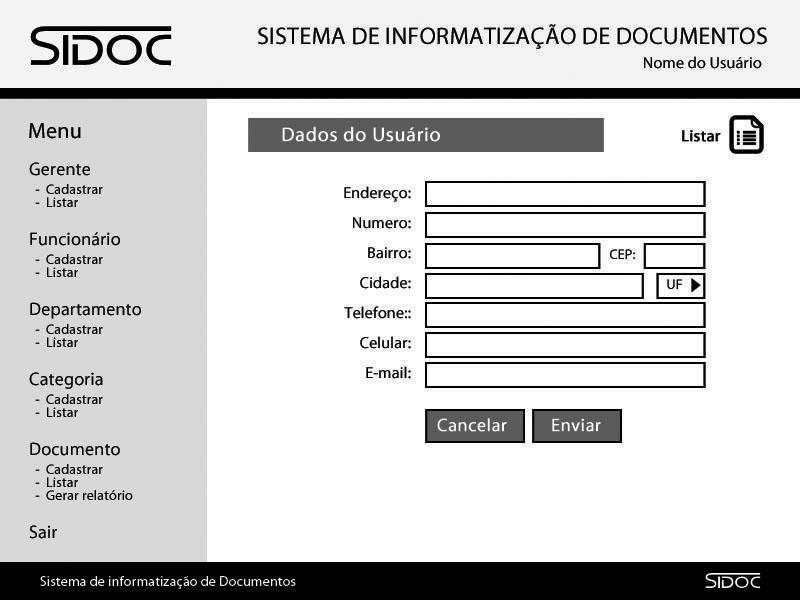


Figura - Tela Cadastrar/Alterar Usuário continuação

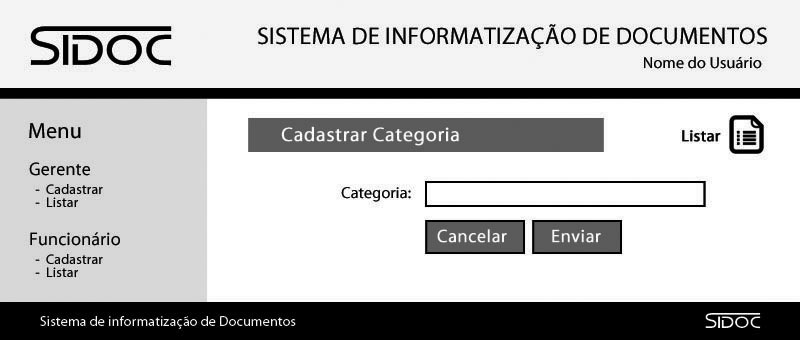


Figura - Tela Cadastrar/Alterar Categoria

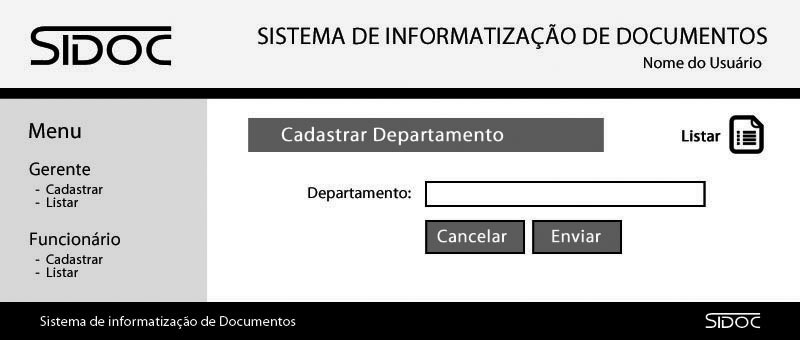


Figura - Tela Cadastrar/Alterar Departamento

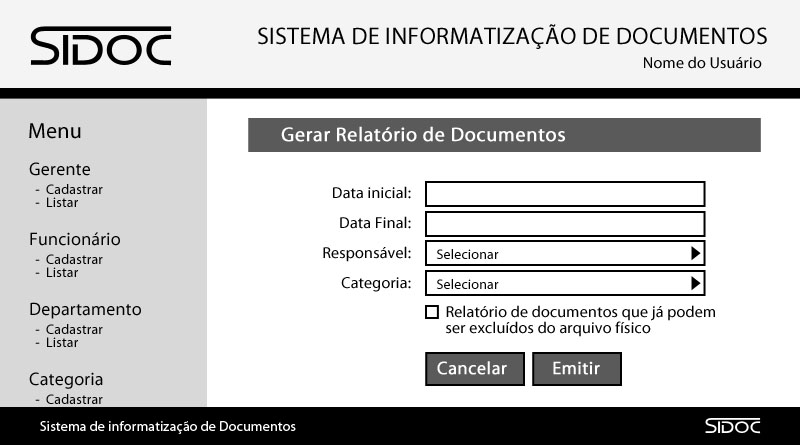


Figura - Tela Gerar Relatório de Documentos

# Visão de Casos de Uso

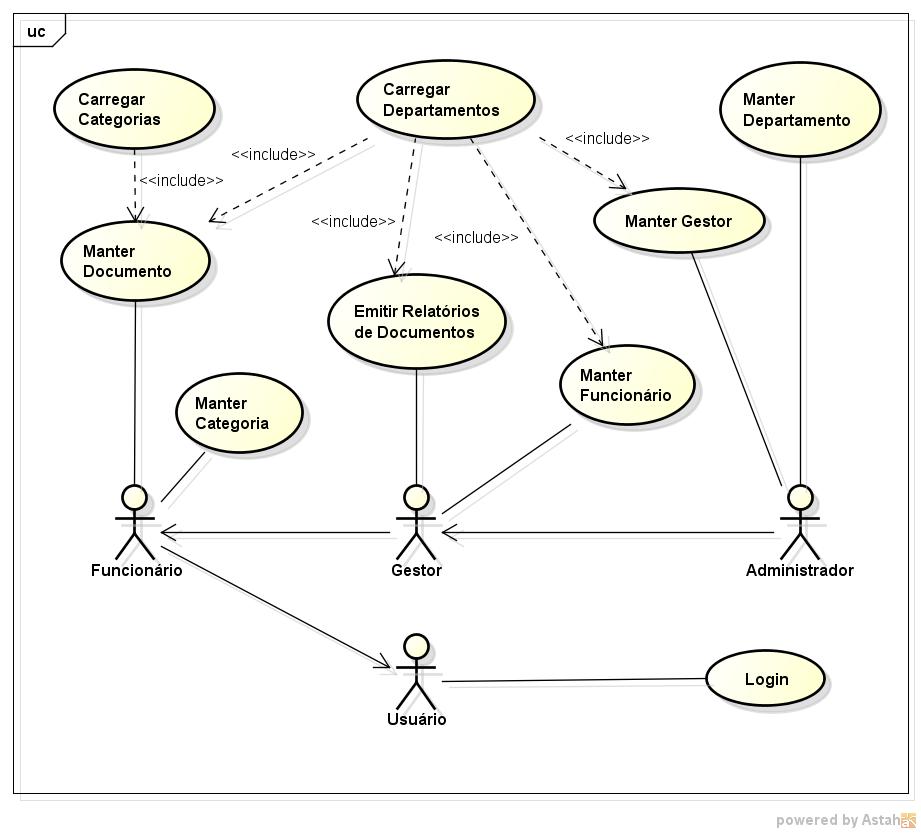


Figura - Caso de Uso SIDOC

## Fazer Login no Sistema

**Atores:** Usuário

**Pré-condição:** N/A.

**Descrição:** Todos os usuários do sistema devem realizar LOGIN, com validação de usuário e senha para ter acesso ao sistema SIDOC.

## Manter Departamento

**Ator:** Administrador.

**Pré-condição:** Login.

**Descrição:** Cadastrar (incluir, consultar, alterar e excluir) Departamentos no sistema. Os campos necessários para preenchimento são: Nome do departamento (campo obrigatório) e o sistema registra automaticamente o código do departamento durante a persistência dos dados.

## Manter Gestor

**Ator:** Administrador.

**Pré-condição:** Manter Departamento.

**Descrição:** Cadastrar (incluir, consultar, alterar e excluir) Gestor no sistema. Os campos necessários para preenchimento são: nome, RG, CPF, login, senha, cargo, tipo de usuário (que deve ser gestor), selecionar o código departamento. Todos os campos são obrigatórios. O campo de contato email também é obrigatório.

Os outros campos de preenchimento são endereço, telefone comercial, telefone residência, celular e sexo. O sistema gera automaticamente o código do usuário do Gestor.

## Manter Funcionário

**Atores:** Administrador e Gestor.

**Pré-condição:** Manter Gestor e Manter Departamento.

**Descrição:** Cadastrar (incluir, consultar, alterar e excluir) Funcionário no sistema. Os campos necessários para preenchimento são: nome, email, RG, CPF, login, senha, cargo, tipo de usuário (que deve ser funcionário), gestor responsável, selecionar o código departamento. Todos estes campos são obrigatórios.

Os outros campos de preenchimento são endereço, telefone comercial, telefone residência, celular e sexo. O sistema gera automaticamente o código de usuário do Funcionário.

## Manter Categoria

**Ator:** Gestor e Funcionário.

**Pré-condição:** Manter Gestor ou Manter Funcionário.

**Descrição:** Cadastrar (incluir, consultar, alterar e excluir) categorias de documentos no sistema. Os campos necessários para preenchimento são: Nome da Categoria (campo obrigatório). O sistema gera o código da Categoria automaticamente.

## Manter Documento

**Ator:** Gestor e Funcionário.

**Pré-condição:** Manter Gestor ou Manter Funcionário e Manter Categoria.

**Descrição:** Cadastrar (incluir, consultar, alterar e excluir) Documentos no sistema. Os campos necessários para preenchimento são:

Campos obrigatórios: Título do documento, código da categoria, data de validade (que significa o período de guarda obrigatória), arquivo (para enviar arquivo(s)) e descrição. Os outros campos são: descrição, referência de documento físico, caso exista e campo booleano documento descartado. O sistema gera o código do Documento e registra a data de cadastro, o código do usuário e o código da categoria automaticamente.

## Carregar Departamentos

**Ator:** N/A.

**Pré-condição:** Manter Departamento.

**Descrição:** Exibir nomes de Departamentos cadastrados.

## Carregar Categorias

**Ator:** N/A.

**Pré-condição:** Manter Categoria.

**Descrição:** Exibir nomes de Categorias cadastradas.

## Emitir Relatórios de Documentos

**Ator:** Gestor.

**Pré-condição:** Manter Documento, Manter Categoria, Manter Gestor e Manter Departamento.

**Descrição:** Emissão de relatório de Documentos cadastrados em um determinado período de tempo. Os campos obrigatórios necessários para preenchimento são: Data inicial e Data final, Código do Departamento.

Campos opcionais: Selecionar responsável, selecionar categoria e opção para gerar um relatório de documentos que já podem ser excluídos do arquivo físico

# Visão Lógica

## Visão Geral

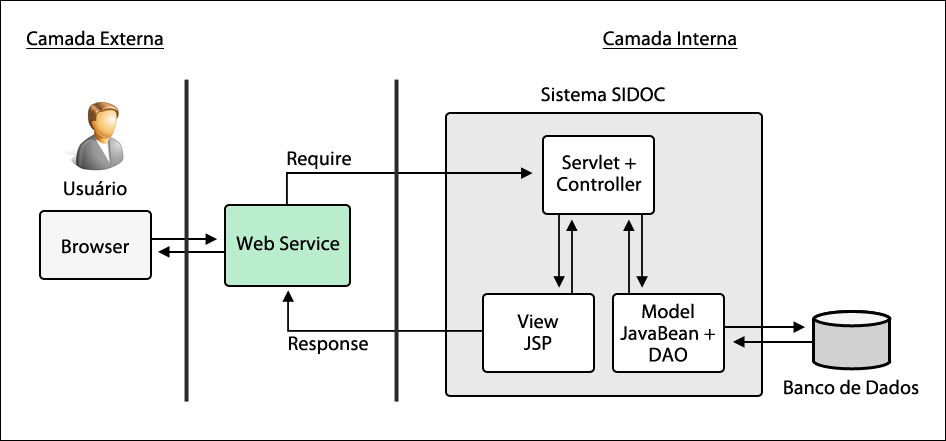


Figura - Principais Elementos da Solução

A figura acima ilustra de uma forma geral os principais elementos da solução a ser desenvolvida. O usuário acessa a *URL* do sistema e faz uma requisição ao *Web Service* por meio de um *Browser*. Esta requisição é interpretada pelo *Controller* da aplicação que vai solicitar à camada de modelo os dados para formar a resposta desejada. Na camada de modelo - *Model* - existem as regras de negócio, os componentes e as informações para conexão com o banco de dados que poderá gravar ou retornar dados. Toda requisição deverá obter um retorno positivo ou negativo da aplicação. Este retorno da camada de modelagem será interpretado pelo *Controller* novamente que o encaminha para a camada de visão - *View -* que se encarregará de construir a página com a resposta a solicitação do usuário. A resposta passa pelo *Web Service* até chegar ao navegador para que o usuário tenha acesso.

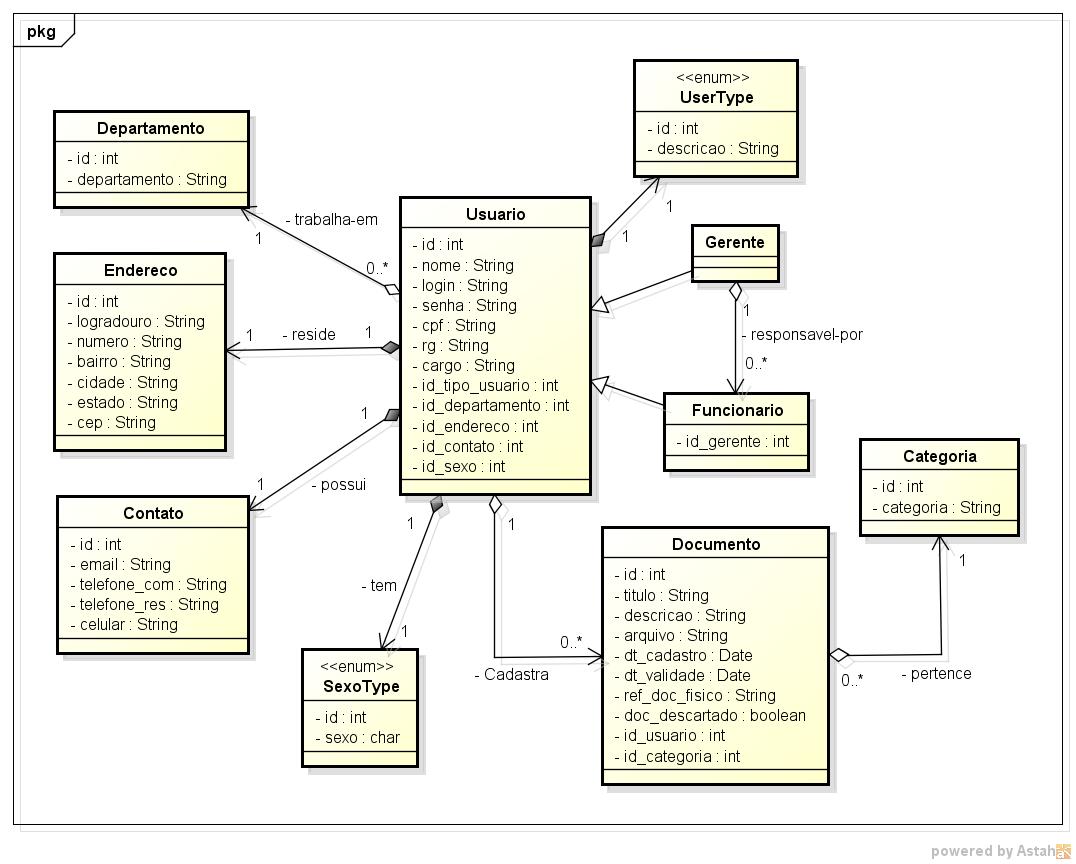


Figura - Diagrama de Entidades e Relacionamento SIDOC

## Nomenclatura da Aplicação

A aplicação a ser desenvolvida terá as seguintes identificações:

|  |  |  |
| --- | --- | --- |
| Nome da aplicação | sidoc\_application |  |
| Pacotes | br.com.sidoc.rules | Contém as regras do negócio. |
| br.com.sidoc.controller | Camada de controle da aplicação. |
| br.com.sidoc.model.bean | Armazena as endtidades. |
| br.com.sidoc.model.dao | Responsável pelo acesso ao banco de dados. |
| br.com.sidoc.common | Contém as classes de suporte e comuns. |

## Visão de Implementação

## Componentes e frameworks a serem reutilizados

Tabela - Componentes e Frameworks que serão utilizados

| Identificação | Responsável | Descrição |
| --- | --- | --- |
| Bootstrap | GitHub | Framework de desenvolvimento em HTML, CSS e JS de forma ágil para diferentes dispositivos. |
| Progressus | GetTemplate | *Template* escolhido que utiliza Bootstrap. |
| Hibernate | Hibernate.org | Framework para mapeamento objeto-relacional em Java. Utilizado para facilitar a transição de dados do modelo tradicional para objetos da aplicação por meio de *Annotations* para permitir esta relação. |
| JasperReports | Jaspersoft Corporation | Framework Java para gerar relatórios em diversos formatos. |
| JSF | Oracle Technology Network | É um framework para construção de interfaces de usuário no padrão MVC em um servidor Java EE |
| Servidor Apache | Apache Software Foundation | Disponibiliza um servidor local para transmissão de dados via intranet com protocolo HTTP. |

## Componentes e frameworks a serem criados

Tabela - Componentes e Frameworks que serão criados

|  |  |  |
| --- | --- | --- |
| **Nome** | **Responsável** | **Descrição** |
| Util | Aplicação | Componente corporativo composto por classes base para outros componentes. |
| Security | Aplicação | Componente corporativo centralizador para a utilização de autenticação, autorização e auditoria. |

## Tratamento de Erros e Exceções

### Erros e exceções do sistema

Os erros e exceções de sistema que possam ocorrer por falha do sistema deverão ser armazenados um arquivo de LOG com código, descrição do erro, data em que ocorreu o erro e o código de identificação do usuário que estava acessando o sistema no momento do erro.

### Erros e exceções do usuário

Todos os erros e excessões ocorridos por erro na utilização do sistema deverão exibir mensagens de orientação e aviso para o usuário. Caso o erro seja de procedimento, o usuário deverá ser encaminhado para a tela onde ocorreu o erro e orientado devidadmente para que possa ser direcionado para a forma correta e consiga finalizar a tarefa esperada pelo sistema.

Para todos os erros que não pertençam ao fluxo natural da aplicação deverão encaminhar o usuário para uma página de erro padrão.

# Visão de Segurança

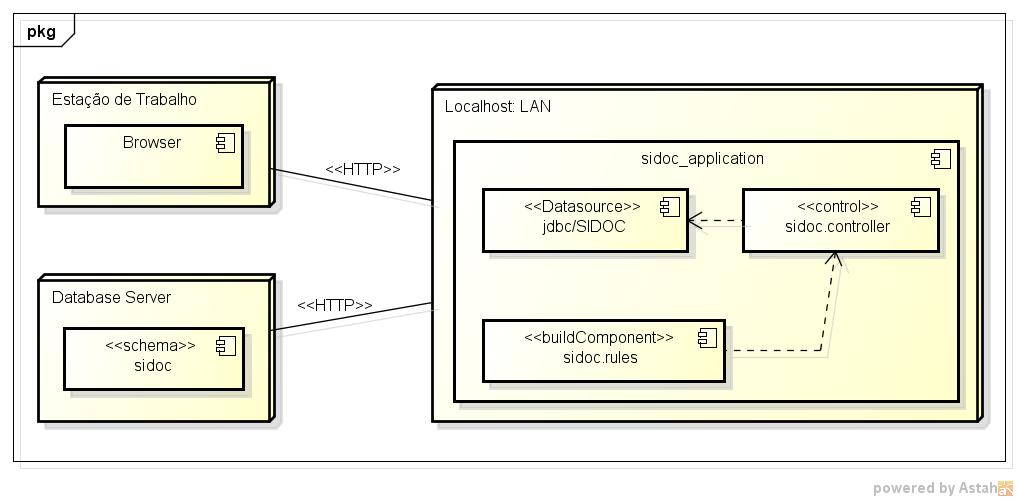
Tabela - Formas de acesso ao sistema

| Usuários | Tipo | Portal | Acesso |
| --- | --- | --- | --- |
| Administrador | Tecnologia da Informação | Sistema Corporativo | Interno |
| Gerente | Gerente do Departamento | Sistema Corporativo | Interno |
| Funcionário | Analistas do Departamento | Sistema Corporativo | Interno |

* O cadastramento dos usuários do tipo Gerentes e de suas permissões de acesso deve ser realizado diretamente no Sistema SIDOC pelo Administrador.
* O cadastramento dos usuários do tipo Funcionários e suas permissões deverão ser cadastrados pelo usuário Gerente responsável por ele.
* É obrigatório que todos os usuários do tipo Funcionários tenham um usuário do tipo Gerente responsável por ele.
* Um usuário do tipo Gerente pode ser responsável por um ou mais Departamentos.
* É obrigatório o preenchimento do campo “Categoria” em todos os Documentos cadastrados.
* Cada Documento cadastrado precisa obrigatoriamente ter o campo “data de validade” preenchido com uma data váida e posterior à data de cadastro.
* Cada usuário do tipo Gerente tem acesso a todos os Documentos que ele cadastrou e também pode acessar os Documentos de todos os integrantes de sua(s) equipe(s).
* Cada usuário do tipo Funcionário tem acesso somente aos Documentos que ele mesmo cadastrou.
* O sistema não permite o acesso a Documentos cadastrados por usuários de um departamento a outros usuários que não pertençam a este departamento.

# Visão de Implantação

## Visão Geral

 Tabela - Diagrama de Implantação

A quantidade de estações de trabalho que terão acesso ao sistema inicialmente é inferior a 50 divididas em 3 departamentos.

A forma de acesso ao sistema será via protocolo HTTP pela rede local - LAN. Sua distribuição ocorrerá através do roteador via intranet a todos os computadores conectados via cabeamento ou via transmissão sem fio (wi-fi).

A aplicação para acessar o sistema será o browser do usuário na estação de trabalho. Cada usuário terá acesso somente a interface gráfica do sistema em seu computador de acordo com as restrições determinadas pelas regras de negócio e será obrigatório o fornecimento de login e senha.

Os componetes contendo as regras de negócio ficarão dentro do servidor local escolhido – Apache, protegidos de acesso direto para manter a integridade dos dados.

A camada jdbc conterá o conjunto de classes que fará a conexão com o banco de dados.

## Servidor web

Tabela - dados do servidor web

| Datasource | Provider | XA | Versão | Servidor | Banco | Login |
| --- | --- | --- | --- | --- | --- | --- |
| jdbc/sidoc | MySQL | SIM | 5 | Localhost:8080 | Sidoc\_application | admin |

## Servidor de Banco de Dados

Tabela - Dados do servidor de Banco de Dados

|  |  |
| --- | --- |
| **Nome do Banco de Dados** | Sidoc\_application |
| **Endereço do servidor** | Localhost:8080 |
| **Dados de Acesso** | User: admin  Password: psw123  Datatable: sidoc |
| **Collation** | Utf-8 |

# Tamanho e Desempenho

* O pico de utilização do sistema deve ser de 100 pessoas utilizando simultaneamente.
* O sistema deve ser intuitivo e de fácil utilização, deve ser aprendido pelo manuseio das funcionalidades básicas em, no máximo, uma hora de utilização.
* O sistema deve ser capaz de indexar arquivos para aumentar a velocidade durante a pesquisa de dados.
* O sistema deve apresentar resultados de pesquisa em 100 itens indexados por 1 segundo.
* O sistema deve permitir o manuseio de arquivos de extensão PDF e JPG.
* O tamanho dos arquivos enviados não deve ultrapassar 50 Megabytes para não sobrecarregar o sistema.
* O sistema precisa ser capaz de registrar um endereçamento do documento físico em arquivo real, caso exista.
* Para não sobrecarregar o sistema e que haja manutenção de desempenho do mesmo, o sistema deve ter a obrigatoriedade de colocar data inicial e final para cada pesquisa por documentos a ser realizada.
* O sistema deve registrar data de descarte de documentos físicos, caso seja possível