**FACULDADE DE TECNOLOGIA DE GUARATINGUETÁ**

**D2VS: Sistema Colaborativo para Promoção Estratégica de Serviços Autônomos**

**Denise Cristina Tobias Sant’Anna**

**Diogo Torres Bianchi**

**Sofia Lunkes da Silva**

**Vitor Barbosa Shimura**

Relatório Técnico-científico apresentado à Faculdade de Tecnologia de Guaratinguetá, para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

**Guaratinguetá - SP**

**2016**

**FACULDADE DE TECNOLOGIA DE GUARATINGUETÁ**

**D2VS: Sistema Colaborativo para Promoção Estratégica de Serviços Autônomos**

**Denise Cristina Tobias Sant’Anna**

**Diogo Torres Bianchi**

**Sofia Lunkes da Silva**

**Vitor Barbosa Shimura**

Relatório Técnico-científico apresentado à Faculdade de Tecnologia de Guaratinguetá, para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Área de Concentração: Análise de Sistemas

Orientadores: Allbert Velleniche D. A. Almeida, Bruno Donizeti da Silva, Claudemir Santos Pinto, José Geraldo de Moraes.

**Guaratinguetá - SP**

**2016**

SANT’ANNA, Denise; BIANCHI, Diogo, SILVA, Sofia, SHIMURA, Vítor. **D2VS**: Sistema Colaborativo para Promoção Estratégica de Serviços Autônomos. Guaratinguetá, 2016. 84p. Relatório Técnico-científico, Faculdade de Tecnologia de Guaratinguetá.

Eu acho justo pensar que computadores pessoais se tornaram a ferramenta mais poderosa que nós já criamos. Eles são ferramentas de comunicação, ferramentas de criatividades e podem ser moldadas pelo usuário. (Bill Gates) 

SANT’ANNA, Denise; BIANCHI, Diogo, SILVA, Sofia, SHIMURA, Vítor. **D2VS**: sistema colaborativo para promoção estratégica de serviços autônomos. Guaratinguetá, 2016. 84p. Relatório Técnico-científico, Faculdade de Tecnologia de Guaratinguetá.

**Resumo**

O Brasil enfrenta, nos últimos anos, uma grande crise econômica. Devido a essa crise, muitas empresas têm demitido seus funcionários e muitos têm se tornado trabalhadores autônomos. Dado o atual cenário brasileiro, este trabalho tem por objetivo apresentar o desenvolvimento de um sistema *Web* paraatender esses trabalhadores autônomos e ajudá-los a divulgarem seus serviços on-line e a participarem de leilões de serviço. Por se tratar de um sistema on-line, oferece facilidade nas pesquisas comparativas e está aberto 24 por dia levando comodidade aos usuários. Para o desenvolvimento do sistema, utilizou-se a pesquisa bibliográfica exploratório para que, por meio de livros, revistas e artigos, informações pertinentes ao problema em questão fosse solucionado. O desenvolvimento foi dividido em duas partes, especificamente, nos dois semestres do ano de 2016. No primeiro semestre, levantou-se os requisitos do sistema, as regras de negócio e os casos de uso. Além disso, trabalhou-se na definição das cores, a tipografia, os ícones e a relação entre as ações das páginas do sistema para a criação dos *wireframes.* Concomitantemente, iniciou-se o planejamento do Modelo Entidade Relacionamento (MER). Ao final do primeiro semestre, criou-se uma versão não funcional do sistema para analisar seu comportamento. A partir do segundo semestre, com orientação dos professores, aplicou-se a metodologia *Scrum* para dividir o desenvolvimento das funcionalidades entre os membros da equipe. A equipe utilizou o sistema de controle de versão de software *Git* para que cada membro desenvolvesse e entregasse, dentro do prazo, e definidas na reunião *Scrum*, as funcionalidades a serem desenvolvidas. Conclui-se que o sistema terá boa utilidade e aceitação pelos próximos anos. Além de ser uma boa oportunidade para trabalhadores autônomos, pessoas com pouca ou nenhuma experiência profissional encontram uma oportunidade de iniciar suas atividades profissionais. O sistema não faz a intermediação do pagamento entre os usuários e percebeu-se que isso poderia trazer consequências negativas. Sugere-se futuramente que isso seja um módulo do sistema para trazer mais confiança e segurança para os usuários e evitar a má imagem do sistema.

**Palavras-chave:** Serviço on-line, Leilão reverso, Autônomo.

SANT’ANNA, Denise; BIANCHI, Diogo, SILVA, Sofia, SHIMURA, Vítor. **D2VS**: sistema colaborativo para promoção estratégica de serviços autônomos. Guaratinguetá, 2016. 84p. Relatório Técnico-científico, Faculdade de Tecnologia de Guaratinguetá.

**Abstract**

In the last years, Brazil has faced a major economic crisis. Due to this crisis, many companies have laid off their employees and many have become self-employed worker. Given the current Brazilian scenario, this paper aims to present the development of a Web system to assist these self-employed workers and to help them advertise their services on-line and participate in service auctions. Because it is an on-line system, it offers ease in comparative searches and is open 24h a day, giving convenience to users. For the development of the system, the exploratory bibliographical research was used so that, through books, magazines and articles, information about the problem in question could be raised. The development was divided in two parts, specifically, in the two semesters of 2016. In the first semester, the requirements of the system, the business rules and the use cases were raised. In addition, the team worked on the definition of colors, typography, icons and the relationship between the actions of the system pages for the creation of wireframes. Concomitantly, the planning of the Relationship Entity Model (MER) was started. At the end of the first semester, a non-functional version of the system was created to analyze its behavior. At the beginning of the second semester, with teacher orientation, the Scrum methodology was applied to divide the development of the functionalities between the members of the team. The team used Git, a distributed version control system, so that each member developed and delivered, within the timeframe, and defined in the Scrum meeting, the functionalities to be developed. It is concluded that the system will be of good use and acceptance for the coming years. Besides being a good opportunity for self-employed workers, people with little or no professional experience find an opportunity to start their professional activities. The system does not mediate payment between users and it has been realized that this could have negative consequences. It is suggested in the future that could be a system module to bring more confidence and security to the users and to avoid the bad image of the system.

**Keyword:** On-line Service, Service auction, Self-employed worker.

LISTA DE ILUSTRAÇÕES

[Figura 1 – Diagrama de caso de uso do sistema D2VS 20](#_Toc468035899)

[Figura 2 – Caso de uso do ator administrador 21](#_Toc468035900)

[Figura 3 – Caso de uso do ator cliente 21](#_Toc468035901)

[Figura 4 – Caso de uso do ator autônomo 22](#_Toc468035902)

[Figura 5 – Diagrama de classes do sistema D2VS 23](#_Toc468035903)

[Figura 6 – Classes Estado, Cidade, TipoMidia e Multimidia 24](#_Toc468035904)

[Figura 7 – Classes TipoUsuario, Favorito, AlterarSenha e Ocorrencia 25](#_Toc468035905)

[Figura 8 – Classes Especialidade, Motivo, AreaAtuacao e Profissao 26](#_Toc468035906)

[Figura 9 – Classes Agendamento, Leilao, AutonomoLeilao e UsuarioProfissao 26](#_Toc468035907)

[Figura 10 – Classes AutonomoLeilao e UsuarioProfissao 27](#_Toc468035908)

[Figura 11 – Classe Usuario 27](#_Toc468035909)

[Figura 12 – Processo de cadastro no sistema D2VS 28](#_Toc468035910)

[Figura 13 – Processo de criação de leilão no sistema 29](#_Toc468035911)

[Figura 14 – Processo de fazer oferta por um leilão 30](#_Toc468035912)

[Figura 15 – Modelo Entidade Relacionamento do sistema D2VS 31](#_Toc468035913)

[Figura 16 - Paleta de cores aplicada no sistema 39](#_Toc468035914)

[Figura 17 – Logotipo do sistema D2VS 40](file:///C:\Users\Diogo\Documents\documentacao\relatorio_tecnico-cientifico.docx#_Toc468035915)

[Figura 18 - Cartela de ícones do sistema D2VS 41](#_Toc468035916)

[Figura 19 - Tela de protótipo da Index 48](#_Toc468035917)

[Figura 20 - Tela de protótipo de cadastro de dados pessoais 49](#_Toc468035918)

[Figura 21 - Tela de protótipo de cadastro de profissional 50](#_Toc468035919)

[Figura 22 - Tela de protótipo de cadastro de mídia 51](#_Toc468035920)

[Figura 23 - Tela de protótipo da Área de Leilão 52](#_Toc468035921)

[Figura 24 - Tela de protótipo de Criação de Leilão 52](#_Toc468035922)

[Figura 25 - Tela de protótipo de Histórico de Leilão 53](#_Toc468035923)

[Figura 26 - Tela de protótipo de Enviar Lance 53](#_Toc468035924)

[Figura 27 - Tela de protótipo de Visualizar Lances 54](#_Toc468035925)

[Figura 28 - Tela de protótipo do Menu Lateral do Autônomo 55](#_Toc468035926)

[Figura 29 - Index do sistema D2VS 56](#_Toc468035927)

[Figura 30 – Validação do tipo de pesquisa 56](#_Toc468035928)

[Figura 31 - Tela de protótipo de cadastro de dados pessoais 57](#_Toc468035929)

[Figura 32 - Tela de cadastro profissional do autônomo 57](#_Toc468035930)

[Figura 33 - Tela de cadastro de mídia 58](#_Toc468035931)

[Figura 34 – Verificação de usuário e preenchimento de formulário 59](#_Toc468035932)

[Figura 35 – Cadastro de especialidade do autônomo 59](#_Toc468035933)

[Figura 36 - Tela da área de leilão dos usuários Cliente e Autônomo 60](#_Toc468035934)

[Figura 37 – Formulário de criação de leilão 60](#_Toc468035935)

[Figura 38 – Componente validador de campos 61](#_Toc468035936)

[Figura 39 – Desenvolvimento *back-end* para a criação do leilão 62](#_Toc468035937)

[Figura 40 - Tela de histórico de leilão 62](#_Toc468035938)

[Figura 41 – Tela de envio de lances 63](#_Toc468035939)

[Figura 42 – Código para gerar as informações sobre o leilão 63](#_Toc468035940)

[Figura 43 – Código para gerar o campo do lance e os botões 64](#_Toc468035941)

[Figura 44 – Cartão gerado a partir de um lance 64](#_Toc468035942)

[Figura 45 – Lance de um autônomo em forma de cartão 65](#_Toc468035943)

[Figura 46 - Tela de menu lateral 65](#_Toc468035944)

[Figura 47 – LinkButton na interface gráfica do usuário 66](#_Toc468035945)

[Figura 48 – Redirecionamento do botão para área de leilão 66](#_Toc468035946)

[Figura 49 – Parametrização para prevenção de injeção SQL 67](#_Toc468035947)

[Figura 50 - Método de aplicação de criptografia à senha do usuário 68](#_Toc468035948)

[Figura 51 - Validação do perfil logado no sistema 69](#_Toc468035949)

LISTA DE QUADROS

[Quadro 1 – Regras de negócio do sistema D2VS 18](#_Toc468035950)

[Quadro 2 – Requisitos funcionais do sistema D2VS 19](#_Toc468035951)

[Quadro 3 - Alternativa textual para conteúdo não textual 36](#_Toc468035952)

[Quadro 4 - Taxa de contraste apropriada para conteúdos textuais 37](#_Toc468035953)

[Quadro 5 - Título na página para facilitar a atual localização no sistema. 37](#_Toc468035954)

[Quadro 6 - Uso de rótulos em formulários 37](#_Toc468035955)

[Quadro 7 - Navegação consistente para rápida interação com conteúdo 38](#_Toc468035956)

**SUMARIO**

[**INTRODUÇÃO 12**](#_Toc468035957)

[**1 DELIMITAÇÃO DO OBJETO 12**](#_Toc468035958)

[**2 JUSTIFICATIVA 13**](#_Toc468035959)

[**3.1 Objetivos Gerais 14**](#_Toc468035960)

[**3.2 Objetivos Específicos 14**](#_Toc468035961)

[**4 MÉTODOS E METODOLOGIA 15**](#_Toc468035962)

[**5 FUNDAMENTAÇÃO TEÓRICA 16**](#_Toc468035963)

[**6 DELIMITAÇÃO DO SISTEMA 18**](#_Toc468035964)

[**6.1 Regras de Negócios 18**](#_Toc468035965)

[**6.2 Requisitos funcionais 19**](#_Toc468035966)

[**6.3 Modelagem do Sistema 20**](#_Toc468035967)

[**6.3.1 Casos de Uso 20**](#_Toc468035968)

[**6.3.2 Diagrama de Classe 22**](#_Toc468035969)

[**6.3.3 Diagrama de Atividades 28**](#_Toc468035970)

[**6.4 Modelagem do Banco de Dados 30**](#_Toc468035971)

[**6.4.1 Modelo Entidade e Relacionamento 30**](#_Toc468035972)

[**6.4.2 Scripts do Banco de Dados 31**](#_Toc468035973)

[**7 INTERAÇÃO HUMANO COMPUTADOR 36**](#_Toc468035974)

[**7.1 Aplicação de práticas de acessibilidade no sistema 36**](#_Toc468035975)

[**7.2 Tecnologia assistiva e reconhecimento de Voz 38**](#_Toc468035976)

[**7.3 Paleta de Cores 38**](#_Toc468035977)

[**7.4 Logotipo 39**](#_Toc468035978)

[**7.5 Cartela de ícones 40**](#_Toc468035979)

[**7.6 Tipologia 42**](#_Toc468035980)

[**8 IMPLEMENTAÇÃO 43**](#_Toc468035981)

[**8.1 Linguagens de Desenvolvimento 43**](#_Toc468035982)

[**8.1.1 SQL 43**](#_Toc468035983)

[**8.1.2 C# 43**](#_Toc468035984)

[**8.1.3 HTML5 43**](#_Toc468035985)

[**8.1.4 CSS3 44**](#_Toc468035986)

[**8.1.5 JavaScript 44**](#_Toc468035987)

[**8.2 Padrão de Desenvolvimento 44**](#_Toc468035988)

[**8.2.1 Web Forms 44**](#_Toc468035989)

[**8.3 Ferramentas Utilizadas 45**](#_Toc468035990)

[**8.3.1 Pencil Project 45**](#_Toc468035991)

[**8.3.2 MySQL Workbench 45**](#_Toc468035992)

[**8.3.3 Visual Studio 2013 46**](#_Toc468035993)

[**8.3.4 Astah Community 46**](#_Toc468035994)

[**8.3.5 Trello 46**](#_Toc468035995)

[**8.3.6 Git e SourceTree 47**](#_Toc468035996)

[**8.4 Prototipação e Desenvolvimento 47**](#_Toc468035997)

[**8.4.1 Protótipo 47**](#_Toc468035998)

[**8.4.2 Desenvolvimento 55**](#_Toc468035999)

[**9 SEGURANÇA 67**](#_Toc468036000)

[**9.1 Aplicação 67**](#_Toc468036001)

[**9.2 A1 – Injeção 67**](#_Toc468036002)

[**9.3 A2 - Quebra de autenticação e gerenciamento de sessão 67**](#_Toc468036003)

[**9.3 A6 – Exposição de Dados Sensíveis 68**](#_Toc468036004)

[**9.4 A7 – Falta de função para controle do nível de acesso 68**](#_Toc468036005)

[**9.5 Estratégia de backup 69**](#_Toc468036006)

[**10 TESTE E IMPLANTAÇÃO 70**](#_Toc468036007)

[**SUGESTÕES PARA IMPLEMENTAÇÕES FUTURAS 72**](#_Toc468036008)

[**CONSIDERAÇÕES FINAIS 73**](#_Toc468036009)

[**APÊNDICE A 74**](#_Toc468036010)

[**REFERÊNCIAS BIBLIOGRÁFICAS 82**](#_Toc468036011)

# 

# INTRODUÇÃO

# 1 DELIMITAÇÃO DO OBJETO

O sistema D2VS, composto por três atores, têm como objetivo contribuir na divulgação on-line e posicionamento estratégico dos trabalhadores autônomos. Há muitos sistemas semelhantes no mercado, no entanto, esse possui um diferencial, o leilão reverso de serviços. O cliente, ator que realiza a pesquisa por esses trabalhadores, também cria leilões, ou seja, divulga uma necessidade de serviço e recebe lances por ela. O principal ator, o trabalhador autônomo, registra-se no sistema com suas informações pessoais e profissionais, e o sistema se encarrega de exibir seu perfil profissional. Ele terá acesso a uma agenda para gerenciar seus compromissos diários e, além disso, poderá visualizar a quantidade de acessos ao seu perfil e seu ganho mensal. Ainda, pesquisará e participará de leilões de serviços dando o seu preço por um serviço solicitado pelo cliente. Por último, o administrador é o ator encarregado de receber reclamações feitas pelos usuários e, ainda, inativar usuários que comprometam o bom funcionamento do sistema.

Não é de responsabilidade do D2VS intermediar a negociação, ou seja, cliente e autônomo, após contato, combinarão a maneira de execução do serviço bem como a forma de pagamento.

# 2 JUSTIFICATIVA

O sistema D2VS oferece diversos benefícios ao cliente que busca um serviço e para o autônomo que divulga o seu. Para o cliente, o sistema oferece comodidade e disponibilidade na busca por trabalhadores, pois este estará aberto 24h por dia e pode ser acessado por diferentes dispositivos de processamento de dados. Além da possibilidade de buscar trabalhadores, o cliente pode solicitar um serviço por meio de um leilão o qual os autônomos participarão dando lances, deste modo, o cliente escolhe o melhor preço pelo serviço solicitado.

Para o autônomo, o sistema disponibilizará uma área de exibição de perfil profissional contendo fotos de serviços, vídeos, suas especialidades e observações extras. A área de leilão de serviço aumenta as chances do trabalhador autônomo, por meio de envio lances, conseguir um serviço.

Pelo fato do autônomo ter que se organizar, pois a prestação de serviço é eventual e não habitual, o D2VS disponibiliza uma agenda de compromissos a fim de auxiliá-lo em sua organização diária. Além da agenda, o trabalhador poderá acompanhar a quantidade de visualizações em seu perfil e o ganho mensal com seus serviços.

Com base nos argumentos apresentados, acredita-se que o sistema, principalmente em tempos de crise econômica, período em que há um aumento no número desses trabalhadores, será de grande ajuda para pessoas desempregadas ou mesmo para aquelas que estão iniciando suas atividades profissionais.

**3 OBJETIVOS**

## 3.1 Objetivos Gerais

O objetivo do sistema é permitir que o trabalhador autônomo participe de leilões de serviços, divulgar seus serviços on-line, auxiliá-lo em sua rotina diária e emitir relatório ganho mensal. Além disso, o sistema permite que seus clientes o avaliem e o recomende.

## 3.2 Objetivos Específicos

* Criar leilão reverso;
* Exibir perfil profissional;
* Agendar horário de serviço;
* Visualizar comanda mensal;
* Visualizar relatório de visualizações de perfil;
* Avaliar autônomo;
* Recomendar autônomo;

# 4 MÉTODOS E METODOLOGIA

O processo de desenvolvimento do sistema foi dividido em dois semestres no ano de 2016.

No primeiro semestre, com base em pesquisa exploratória sobre um crescente aumento de trabalhadores autônomos no mercado de trabalho brasileiro, projetou-se um sistema atendê-los. A fase inicial do projeto consistiu-se em definir a ferramenta de gerenciamento do projeto, levantar os requisitos e regras negócio, bem como os casos de uso que compuseram o sistema e o diagrama de classes. Após essa fase, a equipe trabalhou na parte *front-end* desenvolvendo *wireframes,* definindo as cores, a tipografia, os ícones e a relação entre as ações das páginas do sistema. Concomitantemente, foi dado início ao planejamento do Modelo Entidade Relacionamento (MER), ou seja, o levantamento dos dados que devem persistir no banco de dados e suas relações. Por último, a equipe codificou o protótipo, versão final e não funcional do sistema.

No segundo semestre, com a orientação dos professores, aplicou-se a metodologia *Scrum* para dar início ao desenvolvimento das funcionalidades. Em quatro *sprints*, foram desenvolvidas as funcionalidades e aplicadas correções e melhorias no sistema. Além da metodologia *Scrum*, a equipe utilizou o sistema de controle de versão de software Git para que cada membro desenvolvesse e entregasse rapidamente algumas funcionalidades.

# 5 FUNDAMENTAÇÃO TEÓRICA

Há, em termos de exibição de perfil profissional, muitos sistemas on-line semelhantes ao D2VS. O LinkedIN, por exemplo, rede social profissional, permite ao usuário construir uma identidade profissional on-line e manter contato com outros usuários. No entanto, O D2VS, além de exibir o perfil profissional, possui um grande diferencial, especificamente para trabalhadores autônomos, o leilão reverso de serviço. Essa nova maneira de fazer negócio beneficia, principalmente, trabalhadores desempregados, pois terão à disposição mais ofertas de serviço.

Para o projeto e desenvolvimento desse sistema diferenciado, a equipe utilizou diversas abordagens, ferramentas e adotou o *framework* *Scrum* para gerenciá-lo.

A elicitação e análise dos requisitos do sistema, processo de listagem de requisitos do sistema por meio de observações, auxiliou a equipe a definir o sistema. “Eles ajudam o analista a compreender o sistema a ser especificado.” (SOMMERVILLE, 2007, p.50).

A escolha do Sistema Gerenciador de Banco de Dados (SGBD), ferramenta mantenedora de Banco de Dados (BD), gerou grandes dúvidas no projeto do sistema. Muitos fatores foram levados em consideração como, por exemplo, preço, qualidade e segurança. A equipe optou pelo MySQL. Nas palavras de Carvalho, Além de ser uma ferramenta gratuita, oferece qualidade, robustez e segurança (CARVALHO, 2015).

Atualmente, muitos produtos requerem a interação dos usuários e os sistemas *Web* não são diferentes. Uma das preocupações da equipe foi tornar a navegação usável e o mais agradável possível para o usuário. Para isso, aplicaram-se conceitos do design de interação. De acordo com ROGERS (2013, p.2),

Uma preocupação crucial do design de interação é desenvolver produtos interativos que sejam usáveis, o que genericamente significa produtos que sejam fáceis de aprender a usar, eficazes e que proporcionem ao usuário uma experiência agradável.

Por oferecer diversos benefícios para a equipe e para o desenvolvimento do sistema, decidiu-se por adotar a metodologia ágil *Scrum*. De acordo com Sabbagh (2013, p.4), Scrum pode ser usado “em projetos para produção de softwares comerciais, de sites da Internet, de softwares embarcados, de aplicativos para dispositivos móveis, de softwares financeiros e de jogos”. A equipe, diariamente, realizava reuniões para manter a comunicação, auxiliar nas dificuldades uns dos outros e acompanhar o progresso de cada membro, isso manteve a equipe produtiva e motivada durante todo o desenvolvimento. Nas palavras de Sabbagh (2013, p.13), cada membro gera uma parte do sistema para atingir sua totalidade, essa parte guia e dá propósito ao seu trabalho.

# 6 DELIMITAÇÃO DO SISTEMA

Para o melhor entendimento do sistema, demonstra-se por meio de regras de negócio e requisitos funcionais as regras seguidas das funcionalidades. Tanto as regras de negócio e os requisitos funcionais serão explicados em seus respectivos capítulos.

## 6.1 Regras de Negócios

Conforme GUEDES (2011, p.22), as regras de negócio representam “as políticas, normas e condições estabelecidas pela empresa que devem ser seguidas na execução de uma funcionalidade.”. O levantamento das regras de negócio é de extrema importância para os desenvolvedores, visto que eles desenvolverão as validações e o funcionamento das funcionalidades. Apresenta-se no quadro 1 todas as regras de negócio levantadas pelos desenvolvedores.

Quadro – Regras de negócio do sistema D2VS

|  |  |
| --- | --- |
| ***Código*** | ***Descrição*** |
| **RN01** | Para usar o sistema, o profissional autônomo deve se cadastrar |
| **RN02** | CNPJ não pode ser cadastrado |
| **RN03** | O profissional autônomo deve ter pelo menos um telefone para contato |
| **RN04** | O profissional autônomo pode avaliar o trabalho de outro profissional autônomo |
| **RN05** | O sistema não gerencia o serviço prestado pelo profissional autônomo |
| **RN06** | O profissional autônomo pode gerar relatório de visualização de perfil e de serviços prestados |
| **RN07** | O profissional autônomo pode recomendar outros profissionais |
| **RN08** | O profissional autônomo deve ter uma foto de perfil |
| **RN09** | O autônomo não pagante não terá acesso ao leilão |
| **RN10** | O autônomo pagante poderá participar de leilões ilimitadamente |
| **RN11** | O autônomo poderá inserir, ilimitadamente, fotos de seus serviços |
| **RN12** | O autônomo poderá adicionar links de vídeos do youtube |
| **RN13** | O contato feito pelo usuário do sistema deve ser respondido em até 2 dias, salvo em feriados e finais de semana |
| **RN14** | O administrador pode inativar usuários do sistema |
| **RN15** | O administrador pode ativar usuários do sistema |
| **RN16** | O cliente só poderá avaliar ou recomendar o autônomo após o término do leilão |
| **RN17** | O administrador do sistema pode cadastrar outros administradores |
| **RN18** | Cliente e autônomo podem abrir uma quantidade ilimitada de leilões |
| **RN19** | O leilão poderá ficar aberto por até 30 dias |
| **RN20** | O leilão deverá ter uma duração mínima de 1 dia |
| **RN21** | Cliente e autônomo que criaram um leilão serão notificados 2h antes do seu término |
| **RN22** | O cliente pode encerrar o leilão mesmo que este não tenha um vencedor |
| **RN23** | O leilão deve conter data de abertura, término, data de previsão de encerramento, título, descrição do serviço |
| **RN24** | O cliente será notificado de novos lances |
| **RN25** | O autônomo poderá fazer pesquisa de leilões. |
| **RN26** | Os lances mais novos do leilão devem ser exibidos no topo da lista dos lances |
| **RN27** | Autônomo não pode fazer lances em áreas de atuação que não sejam pertencentes a ele |
| **RN28** | O carrossel exibirá leilões aleatórios |
| **RN29** | No leilão, os lances mais novos serão exibidos primeiro para o cliente |
| **RN30** | Uma vez criado o leilão, este não poderá ter suas informações editadas |
| **RN31** | Um lance não precisa ser aceita aceito ou negado pelo cliente |

## 6.2 Requisitos funcionais

Por meio de entrevista, técnica de levantamento de requisitos funcionais, buscou-se saber o que o cliente gostaria para o sistema e como ele deveria funcionar. Para GUEDES (2011, p.22), “o engenheiro de software busca compreender as necessidades do usuário e o que ele deseja que o sistema a ser desenvolvido realize.”. Ainda de acordo com o GUEDES(2011, p.22), “o engenheiro tenta compreender como funciona atualmente o processo a ser informatizado e quais serviços o cliente precisa que o software forneça.”. O quadro 2 apresenta todos os requisitos coletados e avaliados pela equipe de desenvolvimento.

Quadro – Requisitos funcionais do sistema D2VS

|  |  |
| --- | --- |
| ***Código*** | ***Descrição*** |
| **RF01** | O sistema deve permitir a inscrição de usuários ao sistema |
| **RF02** | O sistema deve permitir ao usuário atualizar suas informações profissionais e/ou profissional |
| **RF03** | O sistema deve permitir ao autônomo incluir fotos e vídeos |
| **RF04** | O sistema deve permitir acesso ao sistema mediante e-mail e senha |
| **RF05** | Após login, o sistema direcionará o usuário para Index. |
| **RF06** | O sistema deve permitir ao usuário recuperar sua senha por meio do e-mail cadastrado |
| **RF07** | O sistema deve permitir ao usuário encerrar a sua sessão no sistema |
| **RF08** | O sistema permite ao usuário pesquisar trabalhadores autônomos por uma profissão ou especialidade |
| **RF09** | O sistema deve permitir filtrar a pesquisa por autônomos pelo estado e cidade |
| **RF10** | O sistema deve permitir ao usuário ordenar a pesquisa por autônomo mais buscado ou melhor avaliado |
| **RF11** | O sistema deve permitir ao autônomo agendar, atualizar ou excluir um serviço de sua agenda |
| **RF12** | O sistema deve permitir ao autônomo visualizar sua comanda mensal |
| **RF13** | O sistema deve permitir ao autônomo visualizar a quantidade de acessos ao seu perfil num período mensal |
| **RF14** | O sistema deve permitir ao cliente/autônomo criar um leilão de serviço, porém não sendo possível modificá-lo |
| **RF15** | O sistema permite ao cliente/autônomo encerrar um leilão antes do prazo final |
| **RF16** | O sistema permite ao cliente/autônomo filtrar lances por: nome, avaliação ou valor do lance |
| **RF17** | O sistema permite ao autônomo dar lance em leilões no qual ele participa |
| **RF18** | O sistema deve permitir a um administrador cadastrado cadastrar outros administradores |
| **RF19** | O sistema deve permitir a listagem de usuários cadastrados para o administrador |
| **RF20** | O sistema deve permitir a ordenação dos usuários cadastrados por tipo de usuário ou status de ativação |
| **RF21** | O sistema deve permite ao administrador ativar ou inativar usuários cadastrados no sistema |
| **RF22** | O sistema deve permitir ao usuário cadastrado ou não entrar em contato com os administradores do sistema |
| **RF23** | O sistema deve permitir ao usuário, após o término do leilão, caso este tenha um vencedor, avaliar e recomendar um autônomo |

## 6.3 Modelagem do Sistema

A modelagem é uma interpretação visual dos múltiplos aspectos do sistema utilizados para o entendimento de suas partes como, por exemplo, as ações que os atores podem executar no sistema e a relação entre os objetos. Os modelos usados e apresentados nos subtópicos subsequentes deste capítulo são: diagrama de caso de uso, diagrama de classes, diagrama de atividades e o Modelo Entidade Relacionamento. Cada modelo será explicado no seu subtópico. Esses modelos serão explicados em seus respectivos tópicos.

### 6.3.1 Casos de Uso do sistema D2VS

Identificaram-se, durante a fase de levantamento de requisitos, os três atores que integram o sistema, o ator Administrador, Autônomo e o Cliente. Por meio do diagrama de caso de uso, descreveu-se o que cada ator faz no sistema facilitando a visão dos desenvolvedores. Nas palavras de Guedes (2011), o diagrama de casos de uso identifica os atores do sistema, bem como as funcionalidades que o sistema ter. Por se tratar de um diagrama grande, ele foi divido por ator, entretanto foi mantido sua versão completa para que fique claro as funcionalidades dos atores e suas relações. A figura 1 ilustra o caso de uso completo do sistema.

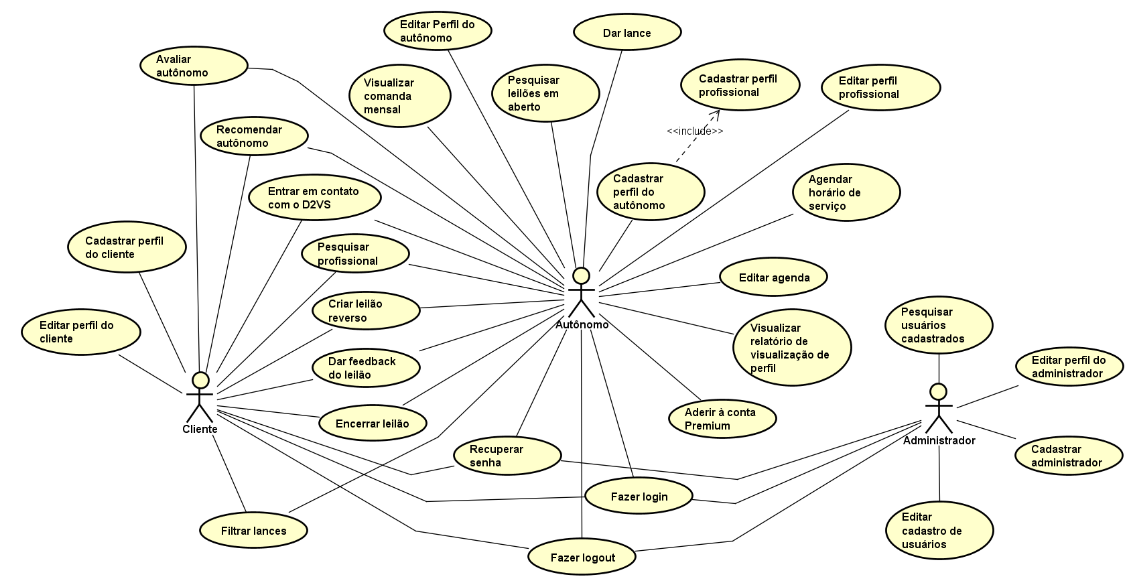


Figura – Diagrama de caso de uso do sistema D2VS

As descrições completas de cada funcionalidade estão em anexo no Apêndice A.

#### 6.3.1.1 Administrador

O Administrador é o ator responsável por cadastrar novos administradores no sistema, assim como gerenciar as contas ativas e inativas. A figura 2 exibe todas as funcionalidades que o administrador pode executar no sistema:

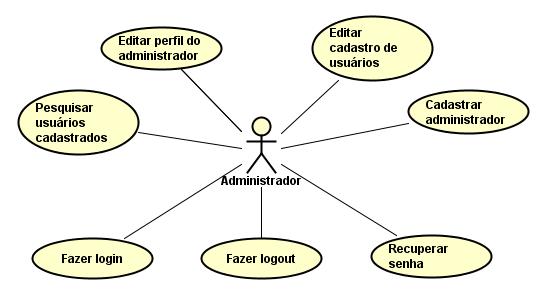


Figura – Caso de uso do ator administrador

#### 6.3.1.2 Cliente

O Cliente é o usuário que acessa o sistema para procurar por trabalhadores autônomos. Além disso, ele pode, por meio do leilão reverso, requisitar um serviço e receber propostas por ele. A figura 3 exibe todas as funcionalidades que o cliente pode executar no sistema.

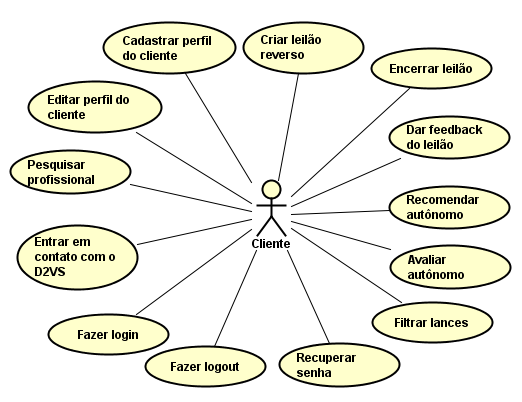


Figura – Caso de uso do ator cliente

#### 6.3.1.3 Autônomo

O Autônomo, ator com o maior número de funcionalidades, é o trabalhador que se cadastra no sistema para divulgar seus serviços e participar de leilões. Além de conter a maior parte das funcionalidades do cliente, no caso as funcionalidades de leilão, há duas ferramentas de auxílio para sua rotina diária, a agenda de serviços e relatório de visualizações de perfil e comanda mensal. A figura 4 exibe todas as que o autônomo pode executar no sistema.

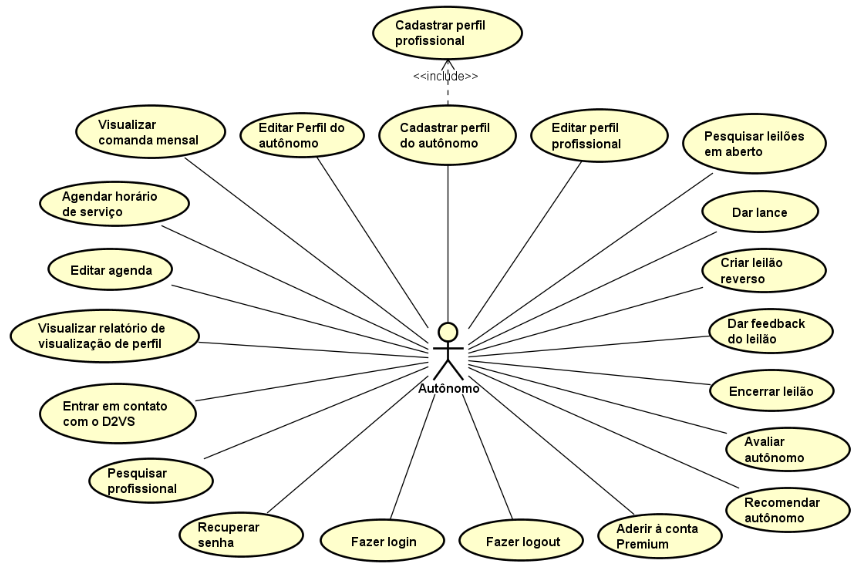


Figura – Caso de uso do ator autônomo

### 6.3.2 Diagrama de Classe

Para definir a estrutura de classes identificadas para o sistema, elaborou-se o diagrama de classes. Segundo GUEDES (2011, p.31), o diagrama de classes

“define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si Nele estão contidos os atributos, métodos e os relacionamento entre as classes que trocam informações.

A fim de reduzir o tamanho do diagrama e criar uma versão mais simples, os métodos chamados de *Setters* e *Getters,* metódos de atribuição e acesso de valores de atributos, respectivamente*,* foram nomeados como metodosDeAcesso(). A relação, representada por uma linha que interliga duas classes, que contém um “\*” representa a relação de vários objetos de uma classe para 0, 1 ou vários objetos de outra classe. Vale ressaltar que todas as figuras serão explicadas seguindo uma ordem de leitura, de cima para baixo e da esquerda para a direita. A figura 5 exibe o diagrama completo do sistema, porém, devido ao seu tamanho, ele será divido em partes e explicado. Mesmo embora a figura do diagrama completo não seja nítida, ela representa, claramente, todas as relações da classe.

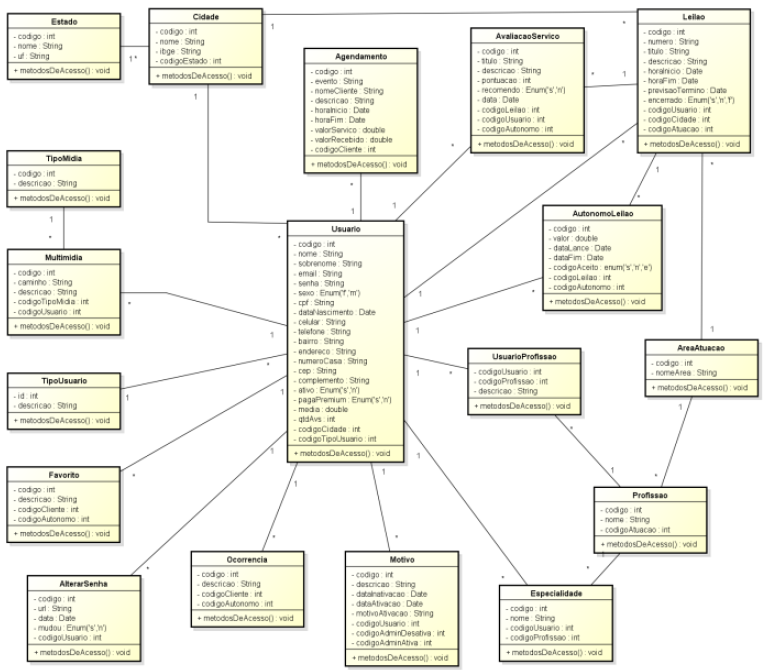


Figura – Diagrama de classes do sistema D2VS

As classes Estado e Cidade contém os atributos de localização de um autônomo e também o local onde um leilão ocorre. Já a classe TipoMidia informa se uma mídia é uma foto ou um vídeo, e a classe Multimidia informa o diretório de armazenamento no servidor e a legenda de cada mídia. A figura 6 ilustra todos os atributos dessas classes.

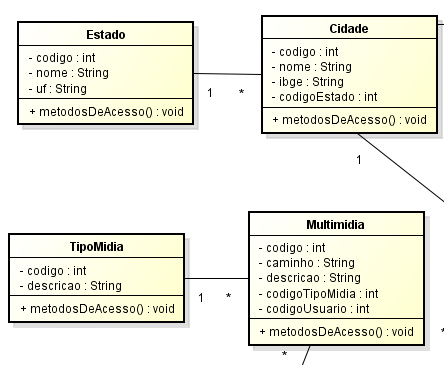


Figura – Classes Estado, Cidade, TipoMidia e Multimidia

A classe TipoUsuario identifica o perfil do usuário no sistema podendo ser: cliente, autônomo ou administrador. A classe Favorito contém informações para que um cliente possa guardar informações de um autônomo e acessar rapidamente na próxima vez que entrar no sistema. Por questão de tempo, a função de adicionar aos favoritos não foi implementada neste projeto, mas a equipe decidiu por manter esse diagrama e implementá-lo futuramente. A AlterarSenha mantém os registros de alterações de senha do usuário, o link para qual ele é redirecionado para muda-la e a data de alteração. Por último, a Ocorrencia representa um registro de uma ocorrência feita por um usuário. A figura 7 ilustra todos os atributos dessas classes.

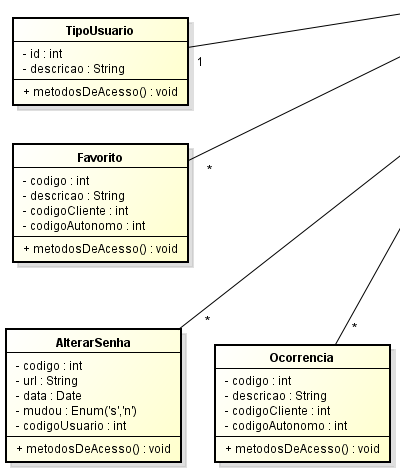


Figura – Classes TipoUsuario, Favorito, AlterarSenha e Ocorrencia

A Especialidade representa todas as especialidades associadas ao um autônomo e as suas profissões. A classe Motivo armazena as informações sobre o administrador que reativou um usuário do sistema e qual o motivo da reativação.

A classe AreaAtuacao mantém o registro de todas áreas que um autônomo pode estar inserido. Por último, a Profissao, que se relaciona com a AreaAtuacao, contém todas as profissões cadastradas por um autônomo. A figura 8 ilustra todos os atributos dessas classes.

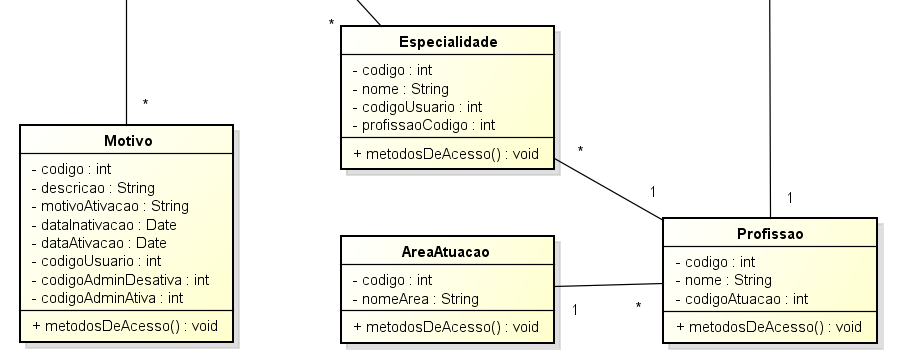


Figura – Classes Especialidade, Motivo, AreaAtuacao e Profissao

A classe Agendamento armazena os dados contidos na agenda do autônomo. A AvaliacaoServico contém as informações sobre a avaliação feita pelo cliente após o leilão. Por último, o Leilao contém as informações necessárias para que o usuário crie um leilão e para que outros visualizem. A figura 9 ilustra todos os atributos dessas classes.

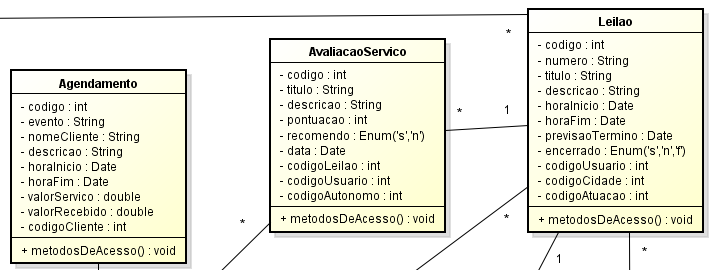


Figura – Classes Agendamento, Leilao, AutonomoLeilao e UsuarioProfissao

A classe AutonomoLeilao contém os atributos referentes a um lance como, por exemplo, o valor dado e a data em que ele foi dado. Já a classe UsuarioProfissao se refere a descrição de cada uma das profissões registradas por um autônomo. A figura 10 ilustra os atributos das duas classes.

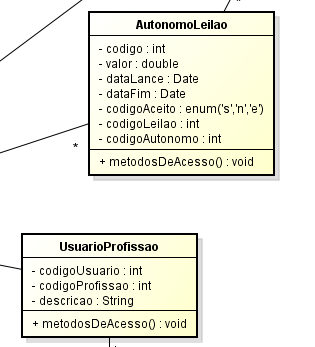


Figura – Classes AutonomoLeilao e UsuarioProfissao

Para finalizar a exibição das partes do diagrama de classes, a figura 11 ilustra a classe Usuario que representa todas as informações pertinentes aos atores do sistema, sendo que o autônomo terá de preencher a maior parte das informações.

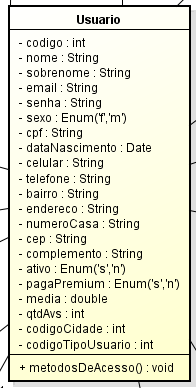


Figura – Classe Usuario

### 6.3.3 Diagrama de Atividades

O diagrama de atividades pode ser usado para representar a modelagem de um processo de negócio assim como a sequência lógica de um único caso de uso. Sendo assim, utilizou-se esse diagrama para representar os principais processos de negócio do sistema D2VS. Para GUEDES (2011, p.36),

O diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo.

A primeira atividade é o cadastro de usuário. O usuário deve acessar o sistema e clicar em cadastrar devendo escolher entre cadastro de autônomo ou cliente. Se ele preencher corretamente o formulário, o acesso ao sistema será liberado. A figura 12 ilustra esse processo.

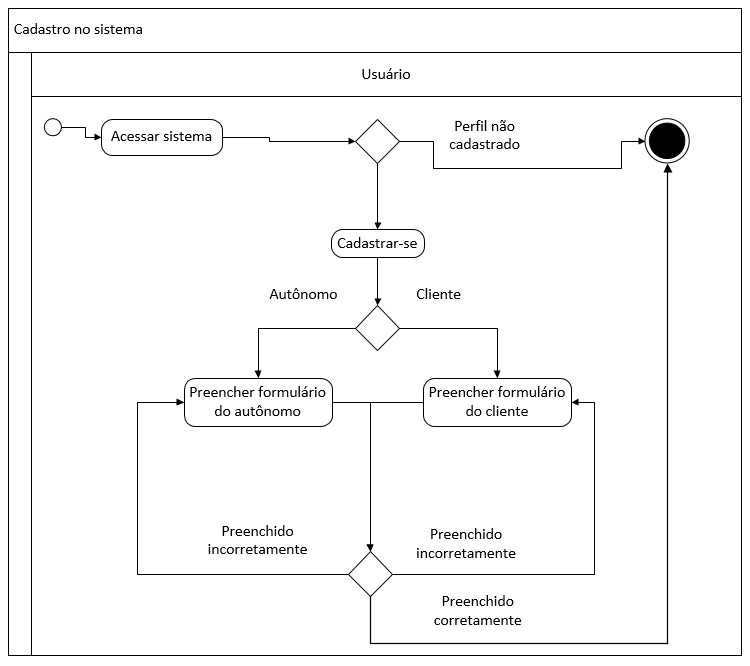


Figura – Processo de cadastro no sistema D2VS

A criação de leilão reverso, principal processo do sistema D2VS, acontece seguindo as seguintes etapas; primeiro, o cliente deve acessar o sistema e realizar o login. Após acesso, deve-se navegar até a área de criação de leilão e clicar em criar leilão. Será solicitado ao usuário um formulário o qual deve ser preenchido corretamente para que o leilão seja exibido aos demais usuários do sistema. Caso tenha sido preenchido incorretamente o usuário deverá preenchê-lo novamente. A figura 13 ilustra o processo completo.

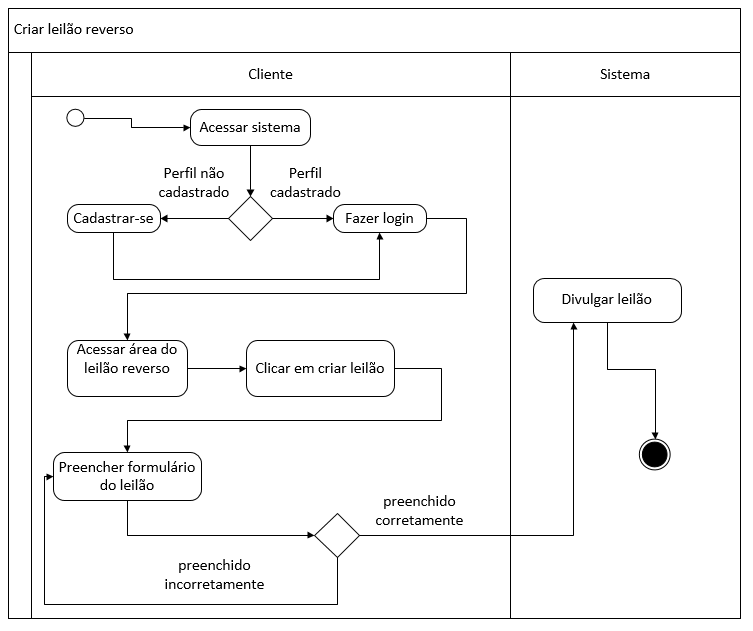


Figura – Processo de criação de leilão no sistema

Outro processo muito importante e que dá continuidade ao leilão reverso é a oferta de lance do autônomo. O autônomo deverá acessar o sistema, que verificará se os dados são válidos. Após validação, ele deverá navegar até a área do leilão reverso. No campo de busca, deverá digitar o leilão que busca. Caso seja encontrado ele deve acessá-lo, caso contrário, deverá realizar uma nova pesquisa. Juntamente com as informações do leilão, haverá um botão chamado “enviar lance” que deverá ser clicado e passado um valor para o leilão. A figura 14 ilustra as etapas pertinentes a esse processo.

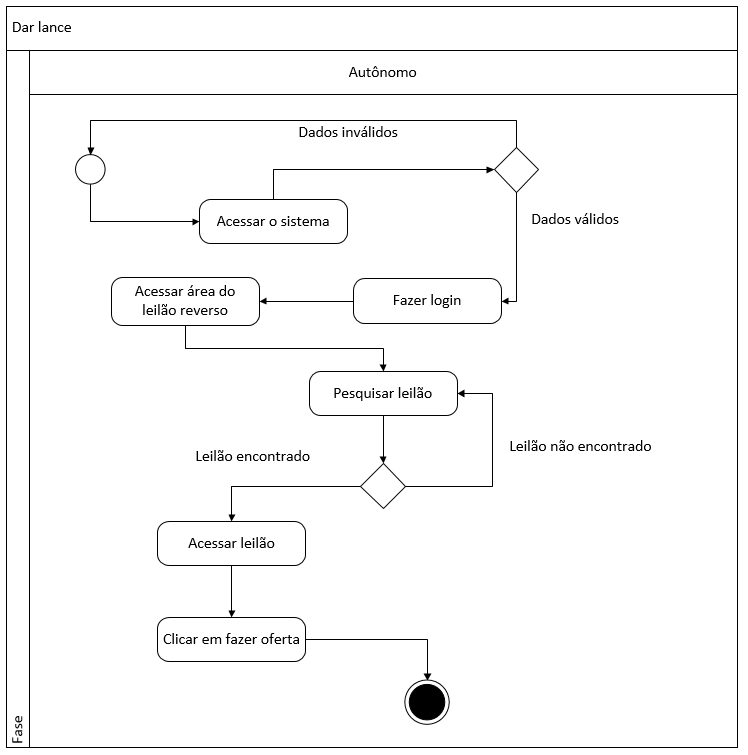


Figura – Processo de fazer oferta por um leilão

### 6.3.4 Modelo Entidade e Relacionamento

Para que fosse desenvolvido o script do banco de dados do sistema D2VS, a equipe projetou o Modelo Entidade Relacionamento (MER) contendo os dados dos objetos que devem persistir no sistema e suas relações. Nas palavras de Heuser (2009, p.72), o modelo ER (Entidade-Relacionamento) pode ser usado como uma ferramenta CASE (Computer Aided Software Engineering) para gerar um banco de dados relacional.

Em relação ao diagrama de classes apresentado no capítulo 6.3.2, a única diferença entre as tabelas e as classes é a ausência do metodoDeAcesso(). No modelo apresentado na figura 15 são apresentados, em forma de quadros, com o nome da tabela em negrito, as tabelas, suas colunas e seus relacionamentos igual ao diagrama de classes. Visto que as colunas e os relacionamentos já foram explicados no diagrama de classes, esse MER não será divido.

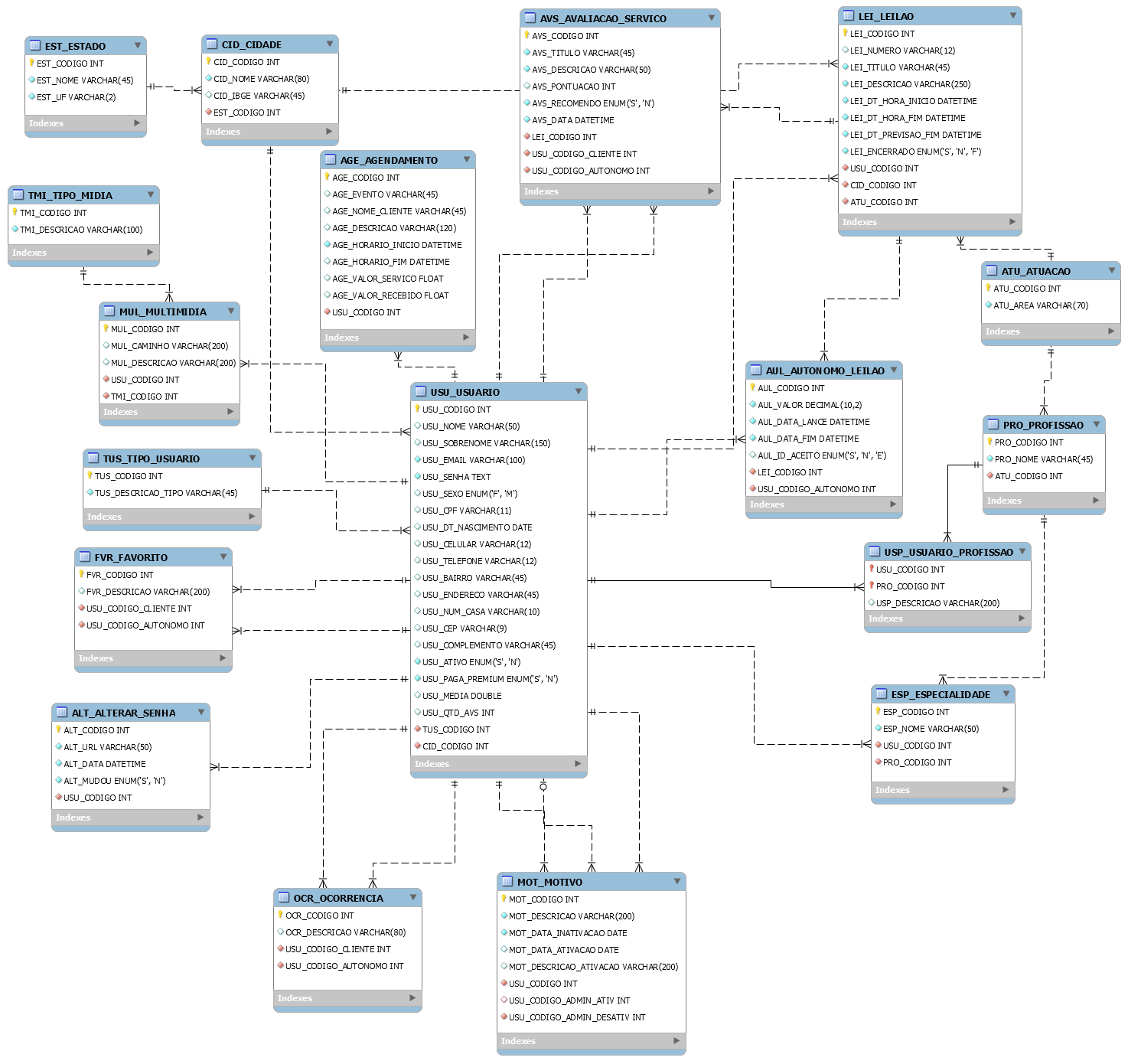


Figura – Modelo Entidade Relacionamento do sistema D2VS

### 6.4.2 Scripts do Banco de Dados

Baseado no MER, gerou-se os comandos necessários para a criação do banco de dados. Para uma melhor visualização dos comandos, as palavras estão escritas em cores diferentes. A cor preta representa o nome do banco de dados, o nome das tabelas e dos campos contidos nela, ou seja, suas colunas. Em cor azul estão as declarações SQL para criar o banco e a definição do tipo de dado das colunas. A cor laranja identifica o tamanho dos campos definidos como varchar, ou seja, um conjunto de *strings*. Por último, identificado na cor verde, estão os campos do tipo enum, objeto *string* que aceita valores definidos pelos desenvolvedores. A seguir, serão apresentados os comandos utilizados para criar cada tabela do banco de dados do sistema D2VS.

CREATE DATABASE D2VS DEFAULT CHARACTER SET utf8;

USE D2VS;

CREATE TABLE ATU\_ATUACAO(

ATU\_CODIGO INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

ATU\_AREA VARCHAR(70) NOT NULL);

CREATE TABLE PRO\_PROFISSAO(

PRO\_CODIGO INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

PRO\_NOME VARCHAR(45) NULL,

ATU\_CODIGO INT NOT NULL,

FOREIGN KEY (ATU\_CODIGO) REFERENCES ATU\_ATUACAO (ATU\_CODIGO));

CREATE TABLE TUS\_TIPO\_USUARIO(

TUS\_CODIGO INT NOT NULL PRIMARY KEY,

TUS\_DESCRICAO\_TIPO VARCHAR(45) NOT NULL);

CREATE TABLE EST\_ESTADO (

EST\_CODIGO INT PRIMARY KEY,

EST\_NOME VARCHAR(45) NOT NULL,

EST\_UF VARCHAR(2) DEFAULT NULL);

CREATE TABLE CID\_CIDADE(

CID\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

CID\_NOME VARCHAR(80) NOT NULL,

CID\_IBGE VARCHAR(45),

EST\_CODIGO INT NOT NULL,

FOREIGN KEY (EST\_CODIGO) REFERENCES EST\_ESTADO (EST\_CODIGO));

CREATE TABLE TMI\_TIPO\_MIDIA (

TMI\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

TMI\_DESCRICAO VARCHAR(100) NOT NULL);

CREATE TABLE TUS\_TIPO\_USUARIO (

TUS\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

TUS\_DESCRICAO\_TIPO VARCHAR(45) NOT NULL);

CREATE TABLE USU\_USUARIO (

USU\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

USU\_NOME VARCHAR(50) NOT NULL,

USU\_SOBRENOME VARCHAR(150) NULL,

USU\_EMAIL VARCHAR(100) NOT NULL UNIQUE,

USU\_SENHA TEXT NOT NULL,

USU\_SEXO ENUM('F', 'M') NULL,

USU\_CPF VARCHAR(11) NULL UNIQUE,

USU\_DT\_NASCIMENTO DATE NULL,

USU\_CELULAR VARCHAR(12) NULL,

USU\_TELEFONE VARCHAR(12) NULL,

USU\_BAIRRO VARCHAR(45) NULL,

USU\_ENDERECO VARCHAR(45) NULL,

USU\_NUM\_CASA VARCHAR(10) NULL,

USU\_CEP VARCHAR(9) NULL,

USU\_COMPLEMENTO VARCHAR(45) NULL,

USU\_ATIVO ENUM('S', 'N') NOT NULL,

USU\_PAGA\_PREMIUM ENUM('S', 'N') NOT NULL,

USU\_PAGA\_PREMIUM ENUM('S', 'N') NOT NULL,

USU\_MEDIA DOUBLE NULL, CID\_CODIGO INT NOT NULL,

TUS\_CODIGO INT NOT NULL,

FOREIGN KEY (TUS\_CODIGO) REFERENCES TUS\_TIPO\_USUARIO (TUS\_CODIGO),

FOREIGN KEY (CID\_CODIGO) REFERENCES CID\_CIDADE (CID\_CODIGO));

CREATE TABLE MUL\_MULTIMIDIA (

MUL\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

MUL\_CAMINHO VARCHAR(200) NULL,

MUL\_DESCRICAO VARCHAR(200) NULL,

USU\_CODIGO INT NOT NULL,

TMI\_CODIGO INT NOT NULL,

FOREIGN KEY (USU\_CODIGO) REFERENCES USU\_USUARIO (USU\_CODIGO),

FOREIGN KEY (TMI\_CODIGO) REFERENCES TMI\_TIPO\_MIDIA (TMI\_CODIGO));

CREATE TABLE FVR\_FAVORITO (

FVR\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

FVR\_DESCRICAO VARCHAR(200) NULL,

USU\_CODIGO\_CLIENTE INT NOT NULL,

USU\_CODIGO\_AUTONOMO INT NOT NULL,

FOREIGN KEY (USU\_CODIGO\_CLIENTE) REFERENCES USU\_USUARIO (USU\_CODIGO),

FOREIGN KEY (USU\_CODIGO\_AUTONOMO) REFERENCES USU\_USUARIO USU\_CODIGO));

CREATE TABLE AVS\_AVALIACAO\_SERVICO (

AVS\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

AVS\_TITULO VARCHAR(29) NOT NULL,

AVS\_DESCRICAO VARCHAR(50) NOT NULL,

AVS\_PONTUACAO INT NULL,

AVS\_RECOMENDA ENUM('S', 'N') NOT NULL,

USU\_CODIGO\_CLIENTE INT NOT NULL,

USU\_CODIGO\_AUTONOMO INT NOT NULL,

FOREIGN KEY (LEI\_CODIGO) REFERENCES LEI\_LEILAO (LEI\_CODIGO),

FOREIGN KEY (USU\_CODIGO\_CLIENTE) REFERENCES USU\_USUARIO (USU\_CODIGO),

FOREIGN KEY (USU\_CODIGO\_AUTONOMO) REFERENCES USU\_USUARIO (USU\_CODIGO));

CREATE TABLE OCR\_OCORRENCIA (

OCR\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

OCR\_DESCRICAO VARCHAR(80) NULL,

USU\_CODIGO\_CLIENTE INT NOT NULL,

USU\_CODIGO\_AUTONOMO INT NOT NULL,

FOREIGN KEY (USU\_CODIGO\_CLIENTE) REFERENCES USU\_USUARIO (USU\_CODIGO),

FOREIGN KEY (USU\_CODIGO\_AUTONOMO) REFERENCES USU\_USUARIO (USU\_CODIGO));

CREATE TABLE MOT\_MOTIVO (

MOT\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

MOT\_DESCRICAO VARCHAR(200) NOT NULL,

MOT\_DATA\_INATIVACAO DATE NOT NULL,

MOT\_DATA\_ATIVACAO DATE NULL,

MOT\_DESCRICAO\_ATIVACAO VARCHAR(200) NULL,

USU\_CODIGO INT NOT NULL,

USU\_CODIGO\_ADMIN\_DESATIV INT NOT NULL,

USU\_CODIGO\_ADMIN\_ATIV INT NULL,

FOREIGN KEY (USU\_CODIGO) REFERENCES USU\_USUARIO (USU\_CODIGO),

FOREIGN KEY (USU\_CODIGO\_ADMIN\_DESATIV) REFERENCES USU\_USUARIO USU\_CODIGO),

FOREIGN KEY (USU\_CODIGO\_ADMIN\_ATIV) REFERENCES USU\_USUARIO (USU\_CODIGO));

CREATE TABLE ESP\_ESPECIALIDADE (

ESP\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

ESP\_NOME VARCHAR(50) NOT NULL,

USU\_CODIGO INT NOT NULL,

PRO\_CODIGO INT NOT NULL,

FOREIGN KEY (PRO\_CODIGO) REFERENCES PRO\_PROFISSAO (PRO\_CODIGO),

FOREIGN KEY (USU\_CODIGO) REFERENCES USU\_USUARIO (USU\_CODIGO));

CREATE TABLE USP\_USUARIO\_PROFISSAO (

PRO\_CODIGO INT NOT NULL,

USU\_CODIGO INT NOT NULL,

USP\_DESCRICAO VARCHAR(200) NULL,

PRIMARY KEY (PRO\_CODIGO, USU\_CODIGO),

FOREIGN KEY (PRO\_CODIGO) REFERENCES PRO\_PROFISSAO (PRO\_CODIGO),

FOREIGN KEY (USU\_CODIGO) REFERENCES USU\_USUARIO (USU\_CODIGO));

CREATE TABLE AGE\_AGENDAMENTO (

AGE\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

AGE\_EVENTO VARCHAR(45) NULL,

AGE\_NOME\_CLIENTE VARCHAR(45) NULL,

AGE\_HORARIO DATETIME NULL,

AGE\_DESCRICAO VARCHAR(120) NULL,

AGE\_VALOR\_SERVICO FLOAT NULL,

AGE\_VALOR\_RECEBIDO FLOAT NULL,

USU\_CODIGO\_CLIENTE INT NULL,

FOREIGN KEY (USU\_CODIGO\_CLIENTE) REFERENCES USU\_USUARIO USU\_CODIGO));

CREATE TABLE LEI\_LEILAO (

LEI\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

LEI\_NUMERO VARCHAR(12) NOT NULL,

LEI\_TITULO VARCHAR(29) NOT NULL,

LEI\_DESCRICAO VARCHAR(250) NOT NULL,

LEI\_DT\_HORA\_INICIO DATE NOT NULL,

LEI\_DT\_HORA\_FIM DATE NULL,

LEI\_DT\_PREVISAO\_FIM DATE NOT NULL,

LEI\_DT\_PREVISAO\_FIM DATE NOT NULL,

LEI\_ENCERRADO ENUM('S', 'N', ‘F’) NOT NULL

USU\_CODIGO INT NOT NULL,

ATU\_CODIGO INT NOT NULL,

CID\_CODIGO INT NOT NULL,

FOREIGN KEY (ATU\_CODIGO) REFERENCES ATU\_ATUACAO (ATU\_CODIGO),

FOREIGN KEY (ESP\_CODIGO) REFERENCES ESP\_ESPECIALIDADE (ESP\_CODIGO),

FOREIGN KEY (CID\_CODIGO) REFERENCES CID\_CIDADE (CID\_CODIGO));

CREATE TABLE AUL\_AUTONOMO\_LEILAO (

AUL\_CODIGO INT PRIMARY KEY AUTO\_INCREMENT,

AUL\_VALOR DECIMAL(10,2) NOT NULL,

AUL\_DATA\_LANCE DATETIME NOT NULL,

AUL\_DATA\_FIM DATETIME NOT NULL,

AUL\_ID\_ACEITO ENUM('S', 'N', ‘E’) NULL,

USU\_CODIGO\_AUTONOMO INT NOT NULL,

LEI\_CODIGO INT NOT NULL,

FOREIGN KEY (USU\_CODIGO\_AUTONOMO) REFERENCES USU\_USUARIO (USU\_CODIGO),

FOREIGN KEY (LEI\_CODIGO) REFERENCES LEI\_LEILAO (LEI\_CODIGO));

CREATE TABLE ALT\_ALTERAR\_SENHA (

ALT\_CODIGO INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

ALT\_URL VARCHAR(50) NOT NULL,

ALT\_DATA DATETIME NOT NULL,

ALT\_MUDOU ENUM('S','N') NOT NULL,

USU\_CODIGO INT NOT NULL,

FOREIGN KEY (USU\_CODIGO) REFERENCES USU\_USUARIO(USU\_CODIGO));

# 7 INTERAÇÃO HUMANO COMPUTADOR

A disciplina de Interação Humano Computador (IHC) foi utilizada nesse projeto para tratar o design do sistema e garantir a melhor experiência de usabilidade ao usuário.

De acordo com ROGERS (2013, p.9, APUD ACM SIGCHI, 1992, p.6), “a IHC tem um foco mais estreito, trata do design, da avaliação e da implementação de sistemas de computação interativos para uso humano e estuda fenômenos que os rodeiam.”.

A fim de garantir essa experiência, alguns itens foram abordados e apresentados nos seguintes subtópicos, são eles: acessibilidade, tecnologia assistiva, paleta de cores, logotipo, cartela de ícones e a tipologia.

## 7.1 Aplicação de práticas de acessibilidade no sistema

A fim de garantir que os usuários tenham acesso a informação e promover o desenvolvimento da sociedade, adotaram-se boas práticas de acessibilidade na *web* promovidas pela W3C. A W3C, consórcio de empresas governamentais e não governamentais, que tem a finalidade de desenvolver padrões para web, desenvolve diretrizes, recursos e estratégias para ajudar pessoas com deficiência[[1]](#footnote-1).

Baseado numa lista desses recursos promovido pelo consórcio[[2]](#footnote-2), utilizou-se 5 aplicações práticas para promover acessibilidade no sistema D2VS. Essas práticas são apresentadas logo abaixo, em formato de quadro, constando o número da prática e o nome na primeira coluna, uma breve definição na segunda coluna e como ela foi utilizada no sistema na última coluna.

O quadro 3 apresenta a primeira prática que beneficia cegos e deficientes auditivos, pois provê alternativas textuais para contextos não textuais, por exemplo, imagens.

Quadro - Alternativa textual para conteúdo não textual

|  |  |  |
| --- | --- | --- |
| **Nome da prática** | **Definição** | **Utilização** |
| 1.1.1 Conteúdo não textual | Fazer com que as informações transmitidas através de conteúdo não-textual sejam acessíveis através de uma alternativa de texto. | Uso do atributo html "alt" em imagens. |

A segunda prática, apresentada no quadro 4, tem por objetivo promover uma visualização agradável do conteúdo textual para pessoas com pouca visão e pessoas idosas.

Quadro - Taxa de contraste apropriada para conteúdos textuais

|  |  |  |
| --- | --- | --- |
| **Nome da prática** | **Definição** | **Utilização** |
| 1.4.3 Contraste (mínimo) | Prover contraste entre cor da frente e cor de fundo para ser lido por idosos e pessoas com baixa visão. | Apresentação visual do texto com taxa de contraste de luminosidade de 4.5:1 |

Beneficiando a todos os usuários, o quadro 5 traz informações sobre o uso de título nas páginas para que os usuários possam se orientar no sistema.

Quadro - Título na página para facilitar a atual localização no sistema.

|  |  |  |
| --- | --- | --- |
| **Nome da prática** | **Definição** | **Utilização** |
| 2.4.2 Título na página | Orientar os usuários sobre sua localização no sistema. | Inserir o título da página na tag html “<title>” |

O uso de rótulos ao lado ou acima de campos de preenchimento em formulários previne erros e otimiza o uso do tempo dos usuários. O quadro 6 refere-se ao uso de rótulos como prática de acessibilidade.

Quadro - Uso de rótulos em formulários

|  |  |  |
| --- | --- | --- |
| **Nome da prática** | **Definição** | **Utilização** |
| 3.3.2 Instruções e rótulos | Identificar tipos de formulários para que os usuários saibam o tipo de dado a ser inserido. | Formulário indicado por rótulos acima de seus campos de preenchimento |

Por último, sem necessidade de recorrer a lentes de aumento, pessoas com pouca visão utilizam pistas visuais para localizar conteúdo rapidamente. A rápida localização de conteúdo ocorre devido ao uso padronizado de elementos que se repetem por um conjunto de páginas do sistema. O quadro 7 traz a definição e a utilização dessa prática.

Quadro - Navegação consistente para rápida interação com conteúdo

|  |  |  |
| --- | --- | --- |
| **Nome da prática** | **Definição** | **Utilização** |
| 3.2.3 Navegação consistente | Uso consistente de elementos os quais o usuário interage repetidamente num conjunto de páginas. | Padronização da posição de elementos nas páginas do sistema. |

## 7.2 Tecnologia assistiva e reconhecimento de Voz

De acordo com o CAT, Comitê de Ajudas Técnicas (2009, p.9),

Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social.

Um exemplo de tecnologia assistiva é o reconhecedor de voz, software que recebe e interpreta a voz de uma pessoa e o transforma em texto. Visto que o D2VS é um sistema de busca, optou-se por usar essa tecnologia. A intenção é facilitar a pesquisa de profissionais autônomos para que estes não tenham dificuldade de acesso a eles. A aplicação da tecnologia foi feita através de uma API[[3]](#footnote-3) *JavaScript*. A API testa se há suporte por parte do navegador do usuário. Além disso, a privacidade do usuário é garantida, uma vez que este deverá permitir acesso ao microfone.

## 7.3 Paleta de Cores

Para a construção visual do sistema, elaborou-se uma cartela com 6 cores que serão aplicadas do sistema, assim como seu código em hexadecimal. A cor principal irá compor o cabeçalho e o rodapé das páginas do sistema e algumas vezes será usado como *hover*, efeito que muda a cor de um elemento HTML quando o cursor do mouse for passado sobre ele. A cor secundária servirá como contraste da cor principal para dar destaque aos elementos do cabeçalho, por exemplo, o logotipo e botões. A cor base é o *background* do sistema e será um leve tom de cinza. O texto terá a cor preta e será usado para descrição do serviço do autônomo assim como em outros momentos de escrita. O alerta representa a cor de mensagens de erro que os usuários possam ter durante a utilização do sistema. Por último, o sucesso representa as mensagens que confirmam que determinada etapa foi concluída sem problemas. A figura 16 representa a paleta de cores aplicada no sistema.

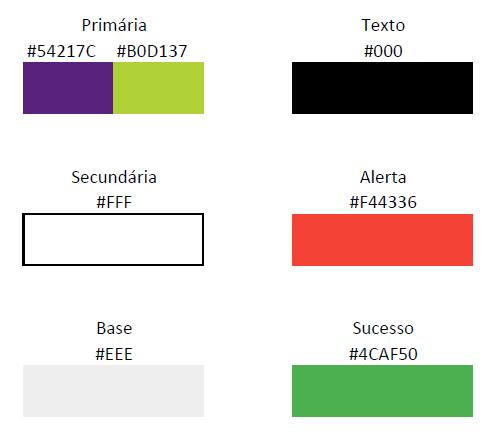


Figura - Paleta de cores aplicada no sistema

## 7.4 Logotipo

O logotipo simboliza um acordo entre um cliente e um autônomo por meio de um aperto de mão. A cor roxa estimula a imaginação, altos ideais e a inspira a fé, comportamento e características essenciais para quem busca prestar o melhor serviço possível. A figura 17 a ilustra



Figura – Logotipo do sistema D2VS

## 7.5 Cartela de ícones

Para a construção do sistema, identificou-se a utilização de 30 ícones gratuitos do Material Design Icon, conjunto de ícones da Google.Serão 5 ícones de serviços para representar as áreas de atuação que o autônomo pode ser inserido, são elas: serviços de gastronomia, serviços de tecnologia da informação, serviços domésticos, serviços de construção e serviços de educação.

Há, também, ícones para que os usuários compartilhem o sistema nas redes sociais, são eles: Facebook, Twitter e Google plus. Na área do autônomo, estão presentes ícones que representam funcionalidades específicas desse ator: serviços do autônomo, relatório de serviços, comanda, calendário de serviços e editar perfil. Na área de cadastro de serviço estarão presentes os ícones: área de imagem, que representa um espaço de adicionar imagens de serviços, link para vídeo e adicionar, que também estará no calendário de serviços. A estrela de avaliação representa a avaliação do autônomo no sistema e poderá ser visualizada por qualquer usuário. Para o administrador, o único ícone específico será cadastrar administrador. Comum para todos os atores, estarão os ícones perfil de usuário, ícone de pesquisa, menu de navegação, *login* e *logout*, que dependerá do estado do usuário no sistema. O ícone comanda servirá para exibir ao autônomo os seus ganhos com serviços. O leilão reverso terá um ícone de martelo. Os botões de confirmação terão um sinal de confirmação, os botões de cancelar, um “X”. O autônomo terá um ícone de contato em sua página para que seus clientes vejam seus contatos. Para simbolizar o leilão reverso, escolheu-o ícone de um martelo. Por último, as opções do leilão terão ícones em forma de caixa de diálogo. A fim de exibir os ícones e suas definições, elaborou-se uma cartela deles exibidas na figura 18.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serviço de gastronomia | Serviço de Tecnologia da informação | Serviços domésticos | Serviços de construção | Serviço de educação |
| Confirmar | Perfil do usuário | Ícone de pesquisa | Menu de navegação | Cancelar |
| Área de imagem | Link para vídeo | Estrela de avaliação | Adicionar | Contato |
| Login | Logout | Serviços do autônomo | Editar perfil | Criar leilão |
| Relatório de serviço | Calendário de serviços | Cadastrar administrador | Comanda | Leilão em andamento |
| http://simpleicon.com/wp-content/uploads/twitter-2-128x128.png  Twitter | https://cdn3.iconfinder.com/data/icons/picons-social/57/80-google-plus-512.png  Google plus | http://simpleicon.com/wp-content/uploads/facebook-2-128x128.png  Facebook | Leilão reverso | Histórico de leilão |

Figura - Cartela de ícones do sistema D2VS

## 7.6 Tipologia

Escolheu-se a fonte calibri para o desenvolvimento do sistema. Por não ter serifas, pequenos prolongamentos de linha nas extremidades das letras, facilita a legibilidade de textos online tanto em computadores desktop como em dispositivos móveis, reforça a confiança e a credibilidade dos leitores em relação ao conteúdo, valorizando, pois, o sistema.

# 8 IMPLEMENTAÇÃO

Nos subtópicos subsequentes tratam-se das linguagens e padrões de desenvolvimento, ou seja, trata do conteúdo usado pelos desenvolvedores para transformar esses itens num sistema. Serão

## 8.1 Linguagens de Desenvolvimento

Para que o sistema D2VS fosse desenvolvido, utilizaram-se algumas linguagens de desenvolvimento sendo elas: C#, HMTL5, CSS3, *JavaScript* e MySQL. A seguir, será apresentada uma breve descrição desses padrões e linguagens.

### 8.1.1 SQL

Antes de falar sobre SQL, é necessário definir o que é um banco de dados. Banco de dados pode ser definido como “um local no qual é possível armazenar informações para consulta ou utilização, quando necessário” (CARVALHO, 2015, p.3). Essas informações são armazenadas em tabelas com colunas e linhas. As tabelas representam um objeto do qual se guarda dados, as colunas identificam o tipo de dado que é guardado e as linhas representam as informações sobre cada registro. Cada banco é um conjunto de tabelas relacionadas.

Sendo assim, para que um banco de dados exista, é necessária uma linguagem para definir, criar e manipular os dados. Está linguagem chamada de *Structured Query Language* (SQL) “é a linguagem padrão utilizada pelos bancos de dados relacionais” (CARVALHO, 2015, p.5).

### 8.1.2 C#

Por se tratar de uma linguagem simples e poderosa, toda a parte do desenvolvimento das funcionalidades foi feita com a linguagem C# (pronuncia-se C *sharp*). Nas palavras de LIMA (2002), é uma linguagem poderosa como o C++ e simples como o Visual Basic. Além disso, ela é completamente orientada a objetos e fortemente tipada o que ajuda evitar erros de manipulação de tipos e atribuições incorretas. Foi a linguagem usada no desenvolvimento das funcionalidades.

### 8.1.3 HTML5

É comum nos dias de hoje que o projeto de desenvolvimento de aplicações locais ou remotas tenham desenvolvedores trabalhando na parte *front-end* do projeto. Mazza (2012, p.2) define o termo *front-end* como “um dos termos usados para se referenciar a interface de uma aplicação”. E para o desenvolvimento *front-end,* utilizou-se o *HTML*. *HyperText* *Markup* *Language* (HTML), conhecida como Linguagem de Marcação de Hipertexto, é uma linguagem baseada em marcas em que palavras-chave indicam o início e o fim de cada elemento hipertexto (BISPO et al., 2012). Esta é apropriada para o projeto de interfaces e, portanto, sua versão mais nova foi utilizada para o desenvolvimento do sistema.

### 8.1.4 CSS3

Para dar uma aparência refinada e moderna ao sistema, utilizou-se o padrão CSS (Folhas de Estilo em Cascata). Folhas de Estilo em Cascata são um mecanismo utilizado para se controlar a aparência das páginas como a cor, posição dos elementos, efeitos de negrito, tamanho e estilo da letra (BISPO et al, 2012).

### 8.1.5 JavaScript

*JavaScript* é uma linguagem interpretada pelo navegador, isso significa que um script é executado quando acessado por ele (AIRES; RIBEIRO; ORLOVSKI 2013). Essa linguagem traz diversos recursos para tornar as páginas mais interativas como, por exemplo, mensagens de alerta e caixas de interação. Ainda, algumas de suas funcionalidades são: adicionar efeitos visuais baseadas em condições, busca de informações sem a necessidade de um recarregamento da página e modificação de conteúdo textual. Além de trazer diversos recursos, o uso dela acrescenta usabilidade e acessibilidade para que o usuário tenha a melhor experiência possível, conceitos adotados para o desenvolvimento do D2VS.

## 8.2 Padrão de Desenvolvimento

Os padrões de desenvolvimento se referem às ferramentas e frameworks para o desenvolvimento e serão apontadas e explicadas nos capítulos subsequentes.

### 8.2.1 Web Forms

Tendo em vista que o presente produto é um sistema *Web*, decidiu-se por utilizar o Web Forms. Web Forms é uma página baseada na tecnologia ASP.NET para criar a interface do usuário para aplicações Web. Ele apresenta informações para o usuário em qualquer navegador e implementa a lógica do aplicativo usando o código no lado do servidor (server-side). Um arquivo Web Forms pode conter HTML estático e também controles de servidor ASP.NET facilitando o desenvolvimento. A lógica para a página reside num arquivo chamado de *code-behind* podendo ser escrita em Visual Basic ou em C#.

### 8.3.1 Pencil Project

Foram feitos *wireframes*, protótipo inicial das telas sem uso de cores, das telas do sistema D2VS para que os desenvolvedores tivessem uma base de como desenvolvê-las. Para isso, utilizou-se a ferramenta a Pencil Project. Além de ser gratuita, essa ferramenta de prototipação possui diversos recursos como uma coleção de formas prontas para desenvolvimento de aplicativos móveis. Ainda, o resultado do trabalho pode ser exportado para um arquivo de imagem .png ou mesmo como uma página *WEB.*

### 8.3.2 MySQL Workbench

O MySQL Workbench é uma ferramenta de gerenciamento de bancos de dados (SGBD) usada por desenvolvedores. Ele permite aos desenvolvedores criar um modelo visual (MER) para que, antes de executar um script, todos os objetos e relacionamentos do banco pudessem ser criados. O MySQL Workbench fornece um ambiente para configurar servidores, administrar usuários e realizar backup de bancos de dados.

### 8.3.3 Visual Studio 2013

Microsoft Visual Studio 2013 é um Ambiente de Desenvolvimento Integrado (IDE) para o desenvolvimento de aplicativos para o sistema operacional Windows e também aplicações *Web*. Ele inclui um editor de código com suporte ao *IntelliSense*, técnica que completa parte do código digitado pelo desenvolver otimizando o tempo de desenvolvimento. Além disso, fornece também um depurador, ferramenta que ajuda os desenvolvedores a encontrar problemas com o código.

### 8.3.4 Astah Community

Durante toda a fase de levantamento de requisitos, utilizou-se a Linguagem de Modelagem Unificada (UML). Nas palavras de Guedes (2011), UML é uma linguagem visual para modelar softwares auxiliando os engenheiros de software a definirem as características do sistema.

A equipe precisou criar diagrama de classes e de caso de uso e para criá-los, decidiu-se pelo software Astah Community. Astah Community é um software gratuito de modelagem UML, criada por uma empresa Japonesa, muito útil no desenvolvimento de sistemas.

### 8.3.5 Trello

Todo o projeto foi gerenciado por meio do Trello, ferramenta on-line de colaboração que organiza o projeto em quadro. O trello oferece um quadro para organizar as tarefas, e a equipe dividiu o quadro em quatro partes indicando em qual etapa está uma determinada tarefa, representada por um cartão semelhante as notas adesivas *posts-its*.

A primeira divisória, chamada Backlog, contém todas as tarefas do projeto. A segunda divisória, *Sprint* X, em que X representa o número da *sprint,* estão os cartões que serão desenvolvidos pela equipe num determinado período de, geralmente, 20 a 30 dias. A terceira divisória, Em Andamento, contém os cartões que estão de fato sendo desenvolvido e qual membro da equipe está trabalhando nele. Por último, a divisória Feito contém todos os cartões que a equipe desenvolveu.

### 8.3.6 Git e SourceTree

*SourceTree* é uma Interface gráfica do usuário (GUI) para o *Git*, sistema de controle de versão de arquivos. O Git permite controlar a versão do código fonte do sistema bem como todos os documentos. Esse sistema permite que cada desenvolvedor trabalhe numa parte do projeto e, posteriormente, integrá-las. Cada alteração de código é submetida com uma mensagem sucinta sobre o que foi feito. Deste modo, todos os desenvolvedores puderam acompanhar o progresso geral do projeto. Além disso, se uma submissão recente conter erros, facilmente ela pode ser revertida. Por ter uma interface amigável e permitir realizar todas essas operações com poucos cliques, escolheu-se esse software.

## 8.4 Prototipação e Desenvolvimento

A fase de implementação se deu em duas partes, o desenvolvimento da prototipação e o desenvolvimento das funcionalidades. Na prototipação, desenvolveram-se as telas de protótipo do sistema auxiliando os desenvolvedores chegarem ao modelo final. Na fase de desenvolvimento, quando os protótipos foram analisados, os desenvolvedores construíram a versão funcional dessas telas.

### 8.4.1 Protótipo

Neste tópico, serão mostradas as telas de protótipo que foram desenvolvidas para que pudesse serem desenvolvidas as telas reais do sistema D2VS. A figura 19 representa a Index, página inicial do sistema, em que um usuário pode pesquisar por um autônomo ou leilão utilizando a barra de pesquisa na parte superior da tela. Ao lado direito da barra de pesquisa, o usuário tem a opção de se cadastrar ou entrar no sistema, caso já seja cadastrado. Abaixo da barra de pesquisa, há um carrossel, espécie de slide automático, que exibe leilões em destaque. Por último, abaixo do carrossel, há uma lista de autônomos em destaque por área de atuação.

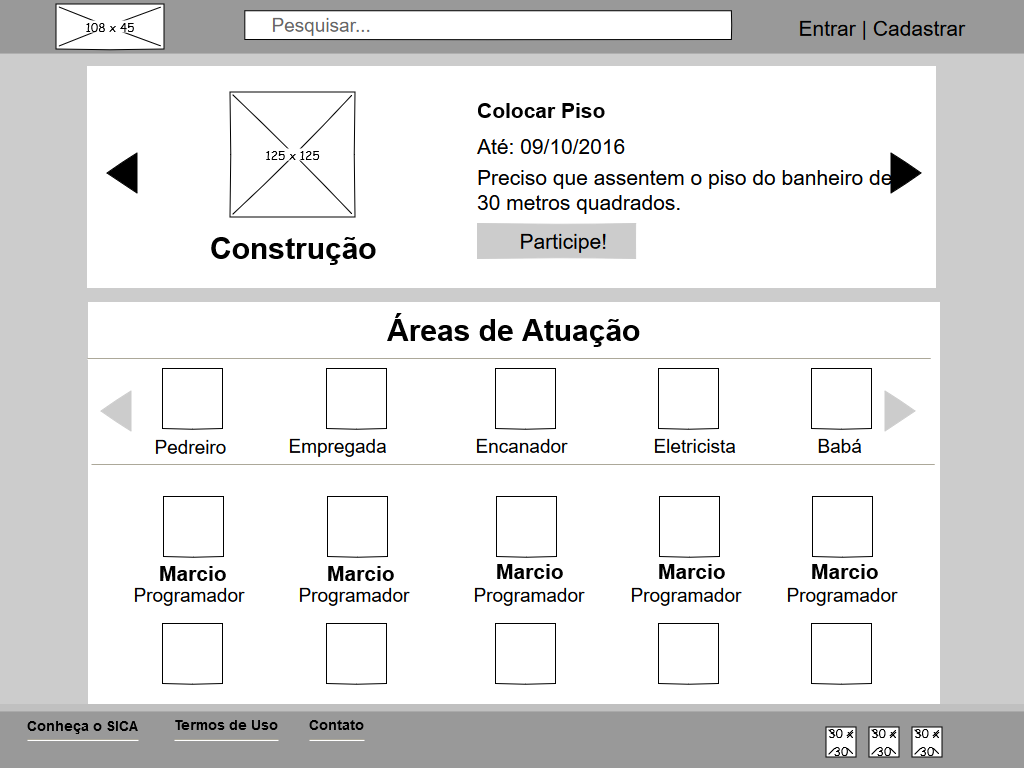


Figura - Tela de protótipo da Index

A figura 20 ilustra o protótipo da tela cadastro de perfil profissional. Essa tela contém um formulário em que ele deve inserir seus dados pessoais. A função desse cadastro é gerar um perfil para que ele seja encontrado por clientes. Além disso, o administrador gerencia melhor o sistema podendo ativar e inativar autônomos. Pelo fato do cadastro ser um processo extenso, a tela foi dividida em três partes: cadastro de dados pessoais, cadastro de perfil profissional e cadastro de mídia.

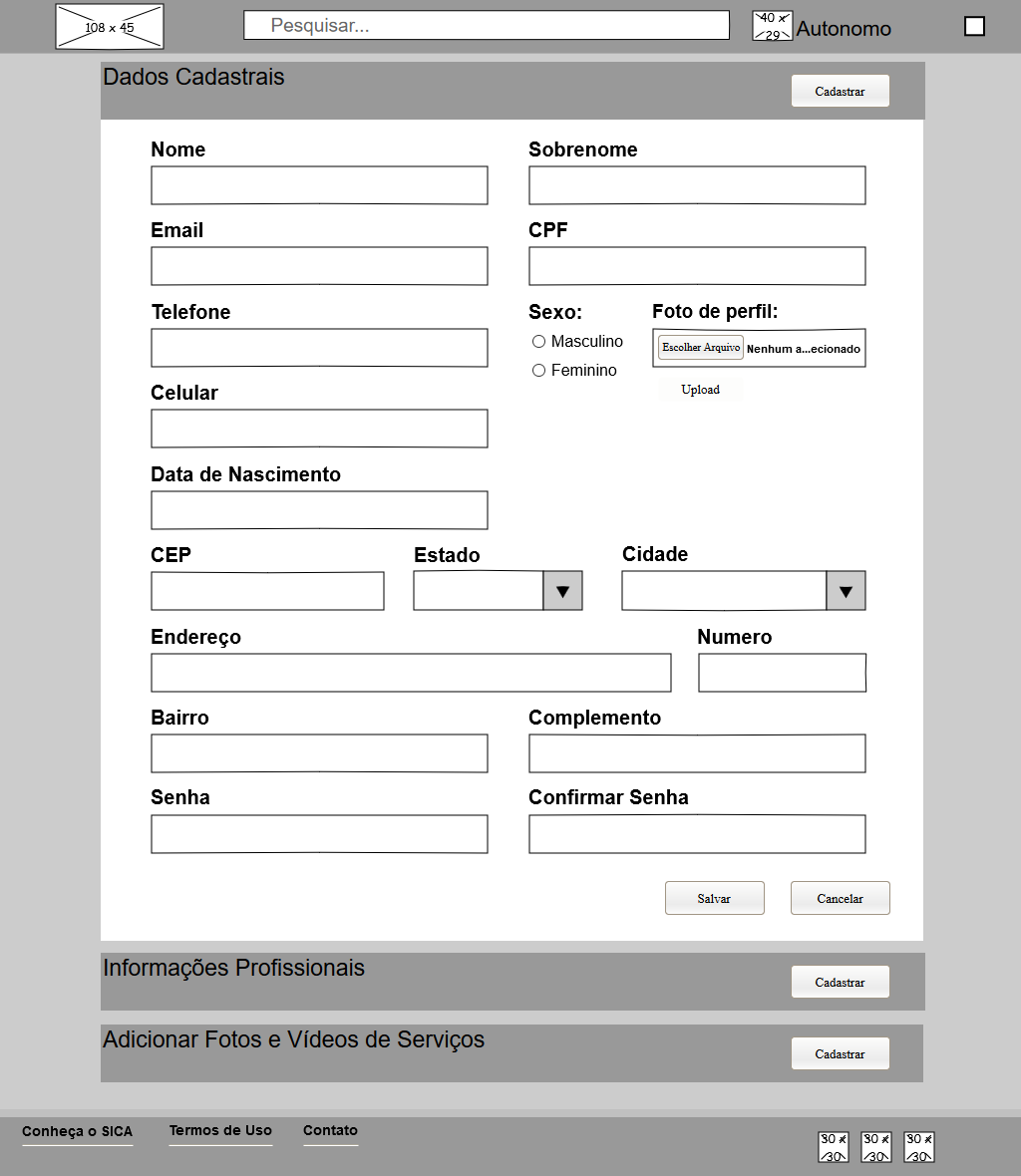


Figura - Tela de protótipo de cadastro de dados pessoais

A figura 21 ilustra a continuação da tela do cadastro do perfil profissional, o cadastro de perfil profissional. Neste momento, é necessário cadastrar a área de atuação, a profissão e suas especialidades. A área de atuação é um conjunto de profissões e o autônomo deverá escolher uma para que os clientes consigam filtrar o resultado de suas pesquisas.

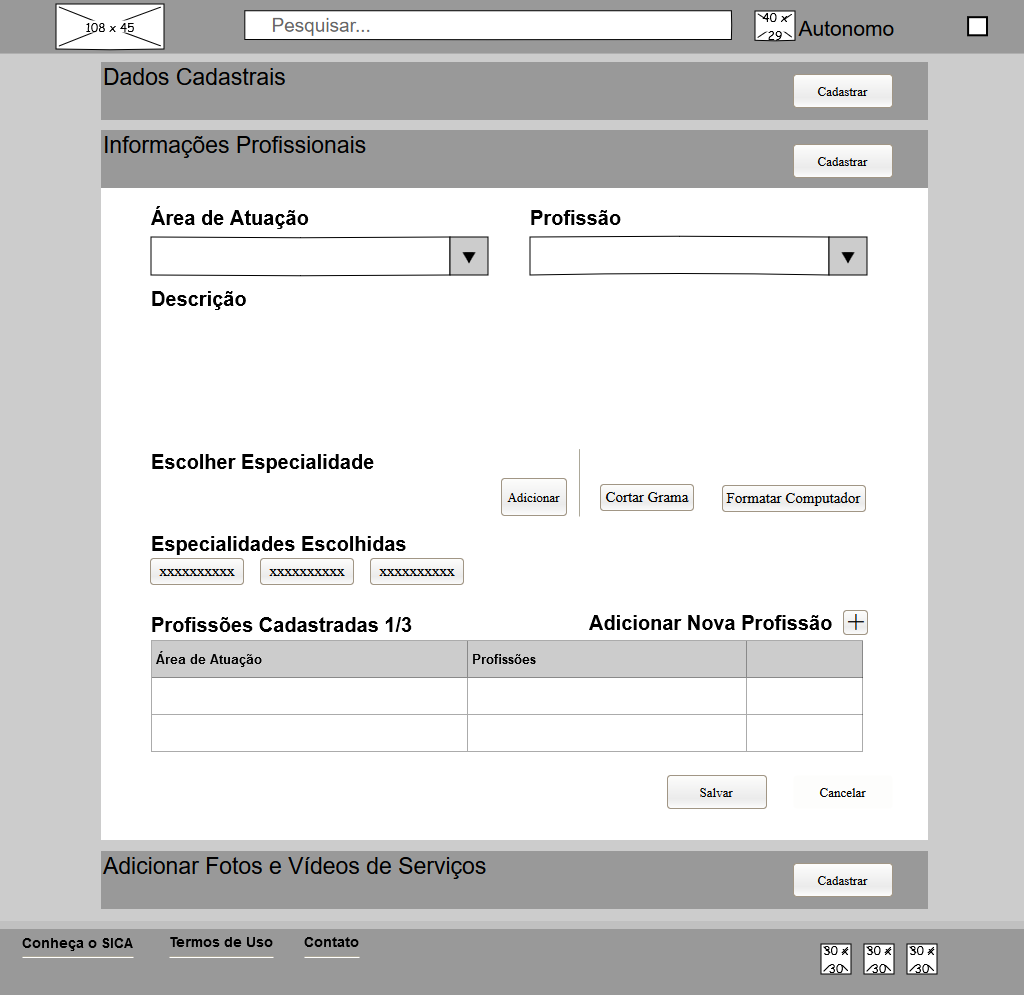


Figura - Tela de protótipo de cadastro de profissional

Por último, a figura 22 ilustra o cadastro de mídia. Quando todas as informações já estiverem preenchidas, o autônomo poderá incluir no seu perfil dois tipos de mídia, foto e vídeo. Essas mídias estarão disponíveis no perfil do autônomo e poderão ser editadas posteriormente.

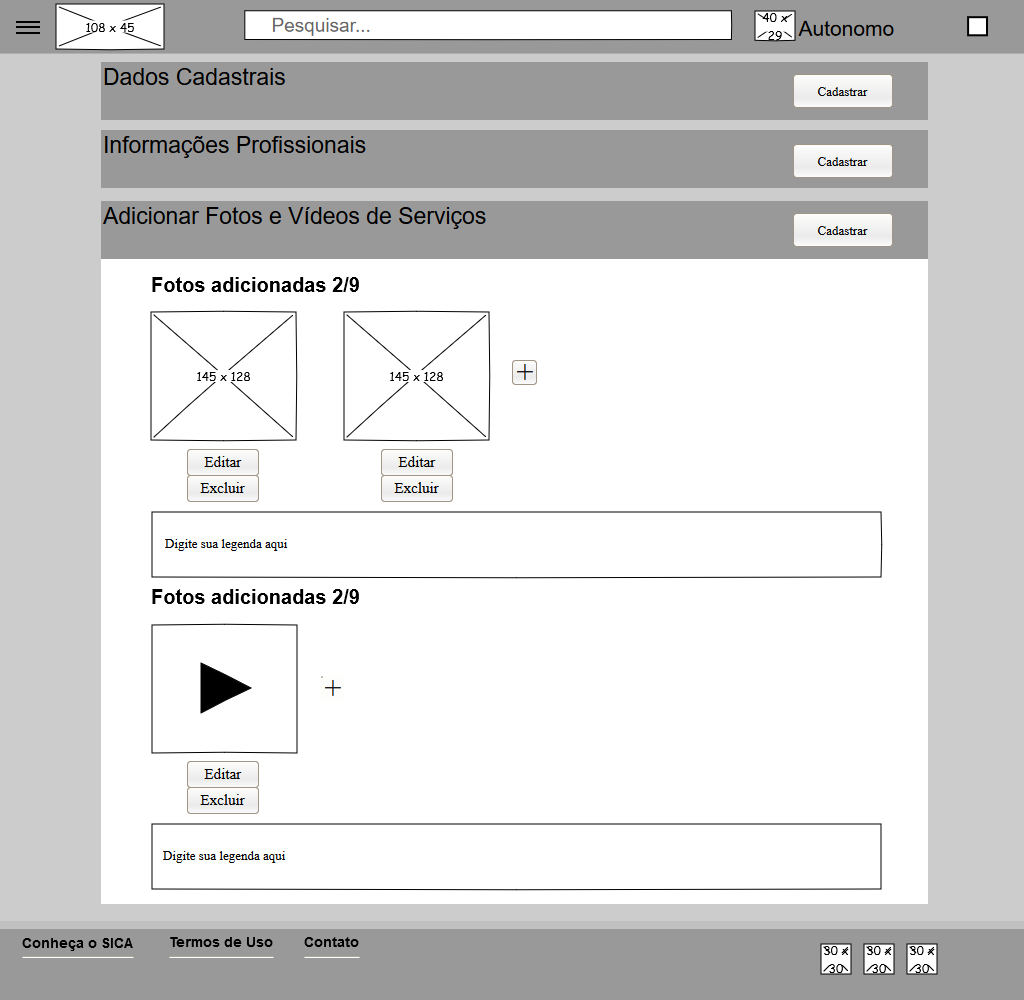


Figura - Tela de protótipo de cadastro de mídia

A figura 23 ilustra a área de leilão, espécie de painel de controle do usuário permitindo que ele faça busca por leilões, crie, visualize quais ele está participando e todos que ele já participou. Clicando-se na opção “leilão reverso”, opção encontrada no menu lateral, ilustrada na figura 28, o autônomo pode acessar essa área e executar a operação que deseja.



Figura - Tela de protótipo da Área de Leilão

A figura 24 mostra como o usuário criará seu leilão. Deve-se preencher um formulário contendo um título do leilão, um prazo final, a área de atuação, seu estado e cidade e, por fim, uma breve descrição de sua real necessidade.

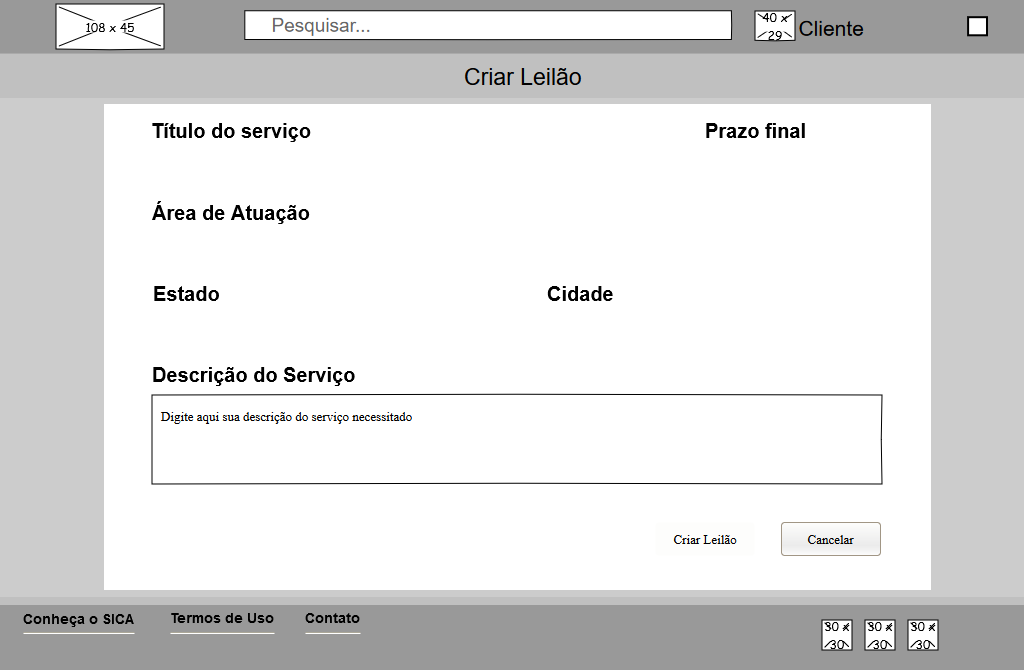


Figura - Tela de protótipo de Criação de Leilão

A figura 25 ilustra a tela de histórico de leilão. Nessa tela, o usuário pode filtrar os leilões em aberto, os encerrados ou mesmo os dois juntos.

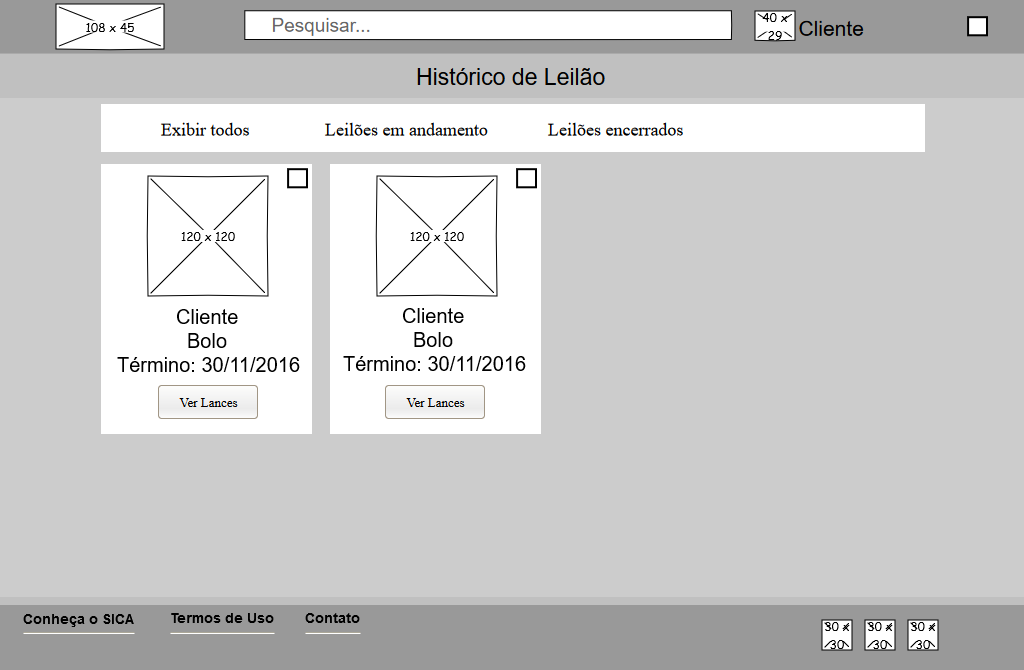


Figura - Tela de protótipo de Histórico de Leilão

Na figura 26, apresenta-se a tela de dar lance em um leilão. Após o autônomo pesquisar o leilão e escolher o leilão que o agrade, ele poderá enviar um lance. Uma vez que o lance foi dado, ele poderá ser atualizado.



Figura - Tela de protótipo de Enviar Lance

Na figura 27, o usuário Cliente e o Autônomo poderão visualizar os lances dados pelos autônomos. Para cada lance enviado é gerado um cartão contendo o nome, a avaliação e o valor do lance. Esse cartão pode ser visto na figura 27 logo abaixo da escrita “Ordenar por”, espécie de filtro disponível ao usuário, e assim aceitar o lance mais adequado.

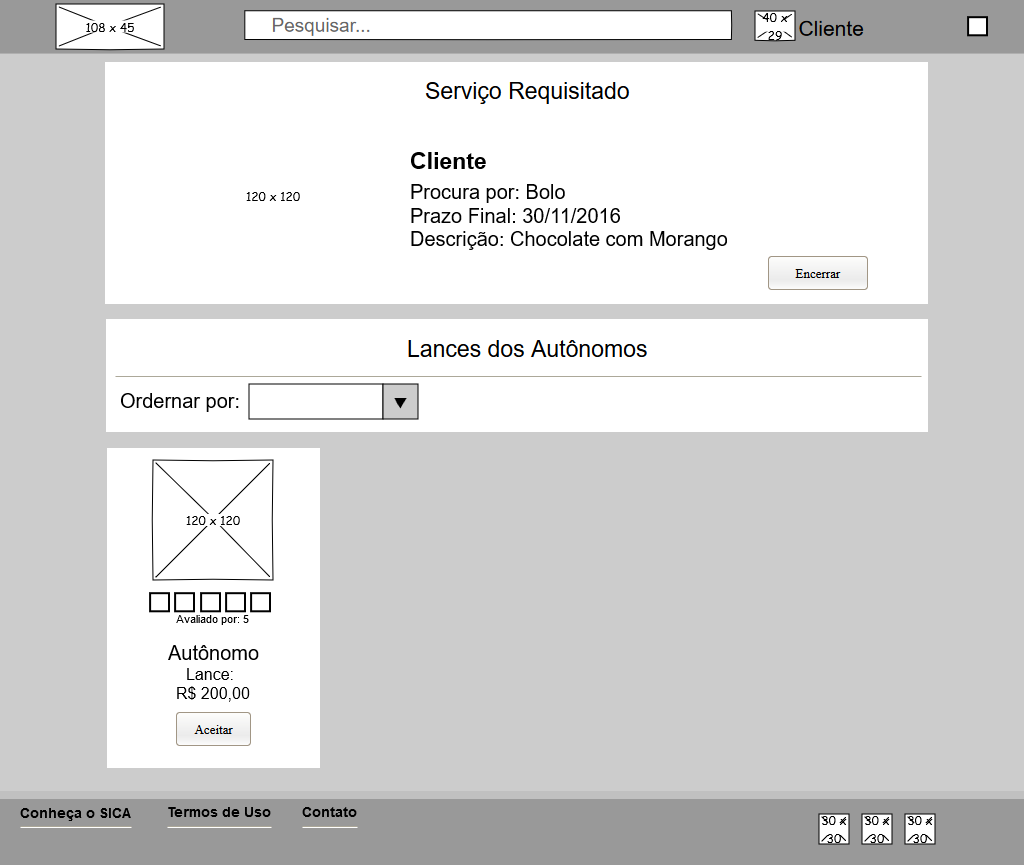


Figura - Tela de protótipo de Visualizar Lances

Por último, a figura 28 ilustra o menu do autônomo. Esse menu facilita a sua navegação para todas as partes do sistema. Todo usuário do sistema terá um menu e, dependendo do seu perfil, outras opções estarão disponíveis. Por exemplo, o usuário Administrador tem acesso a uma área exclusiva de gerenciamento de usuários e, portanto, não poderá ser vista no menu do usuário Cliente ou Autônomo.

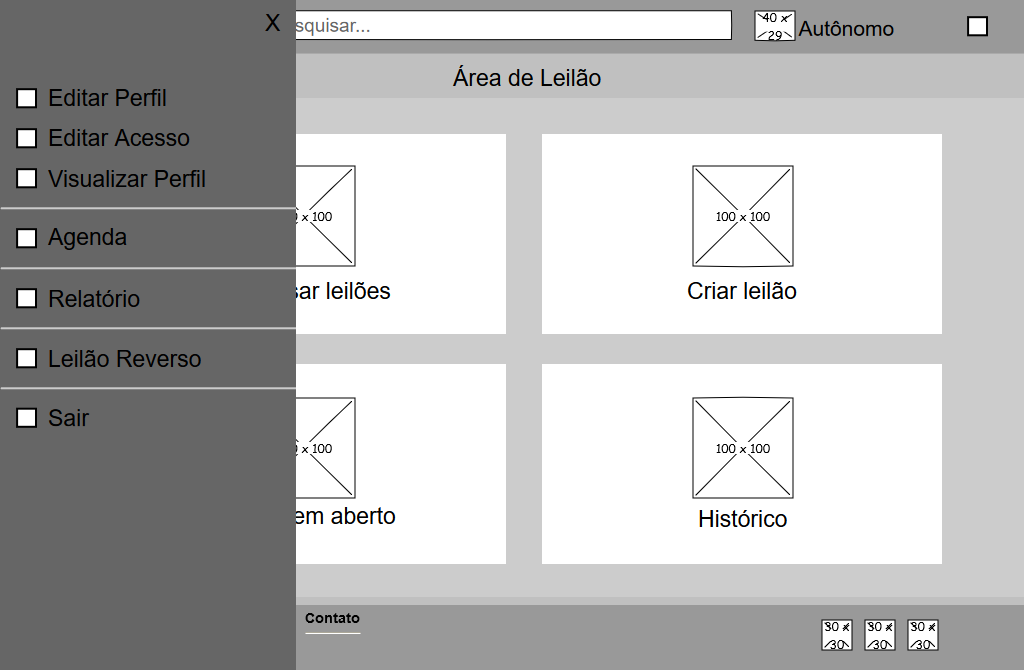


Figura - Tela de protótipo do Menu Lateral do Autônomo

### 8.4.2 Desenvolvimento

Serão explicadas no subtópico 8.4.2 as principais funcionalidades, das principais telas do sistema, e como foi feita sua programação. Vale ressaltar que as funcionalidades se encontram no Apêndice A deste documento.

A seguir, ilustrado na figura 29, está a tela a index, página inicial do sistema, que poderá ser visualizada por qualquer usuário que estiver acessando o sistema, independente se este é cadastrado ou não. Nessa tela, ao lado do logo D2VS, está a barra de pesquisa com o ícone de uma lupa. Essa barra permite pesquisar autônomos e leilões filtrado por estado e cidade. Será explicado, pois, como foi feita a sua programação.

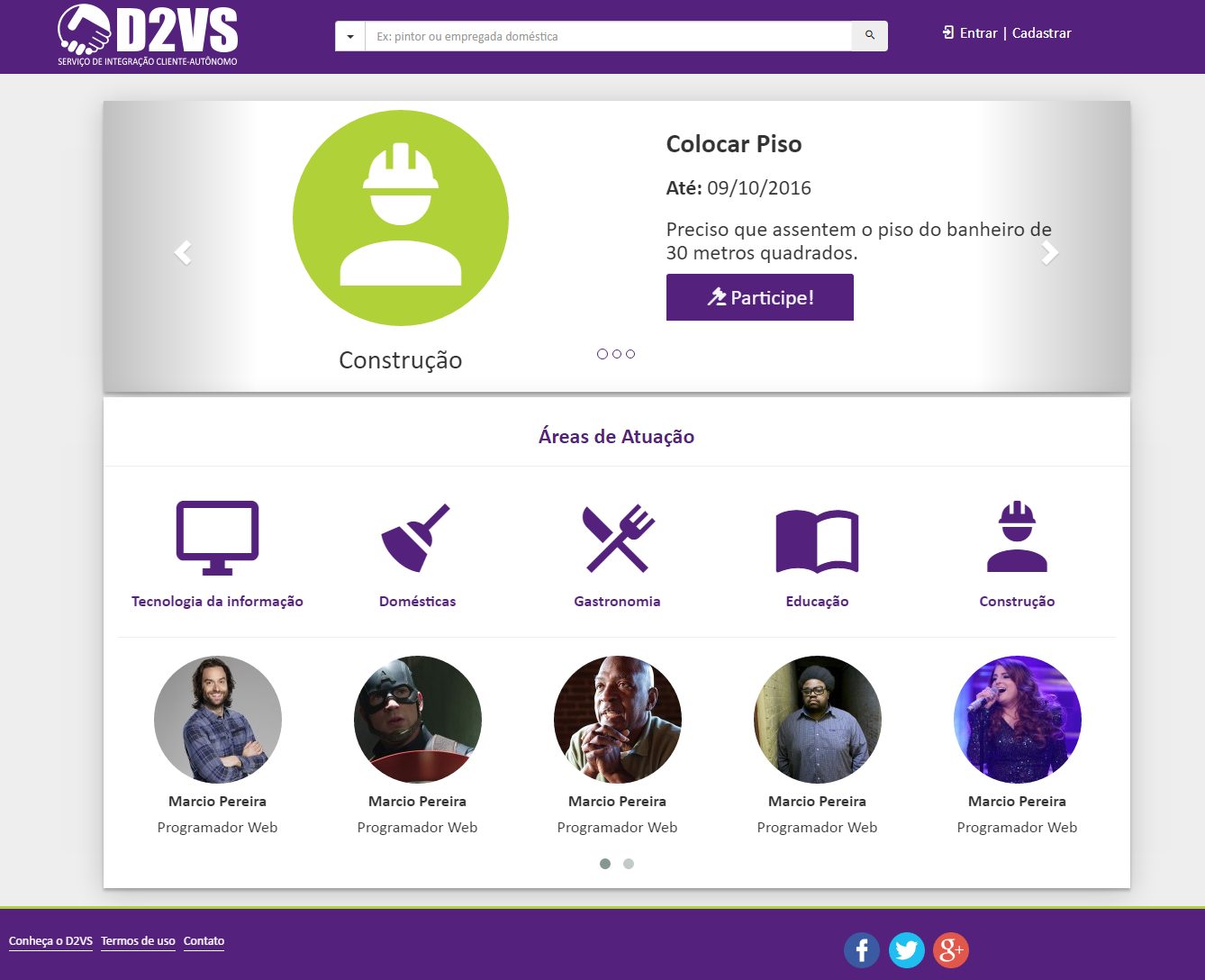


Figura 29 - Index do sistema D2VS

Ao clicar no ícone da lupa, é chamado um método que valida o tipo de pesquisa, pesquisa por autônomo ou por leilão, conforme ilustra a figura 30. Se a pesquisa por autônomo for escolhida, o sistema irá direcionar o usuário para a tela de busca de profissionais, caso contrário, o usuário será redirecionado para a pesquisa de leilões.

protected void lbtPesquisar\_Click(object sender, EventArgs e)

{

if (rblTipoPesquisa.SelectedValue == "1")

{

Response.Redirect("BuscaDeProfissional.aspx?prof=" + txtPesquisa.Text + "&cid=" + ddlCidade.SelectedValue + "&t=" + rblTipoPesquisa.SelectedValue);

}

else

{

Response.Redirect("AutonomoOferta.aspx?l=" + txtPesquisa.Text + "&c=" + ddlCidade.SelectedValue + "&t=" + rblTipoPesquisa.SelectedValue);

}

}

Figura – Validação do tipo de pesquisa

A figura 31 identifica a tela de cadastro de dados pessoais autônomo.

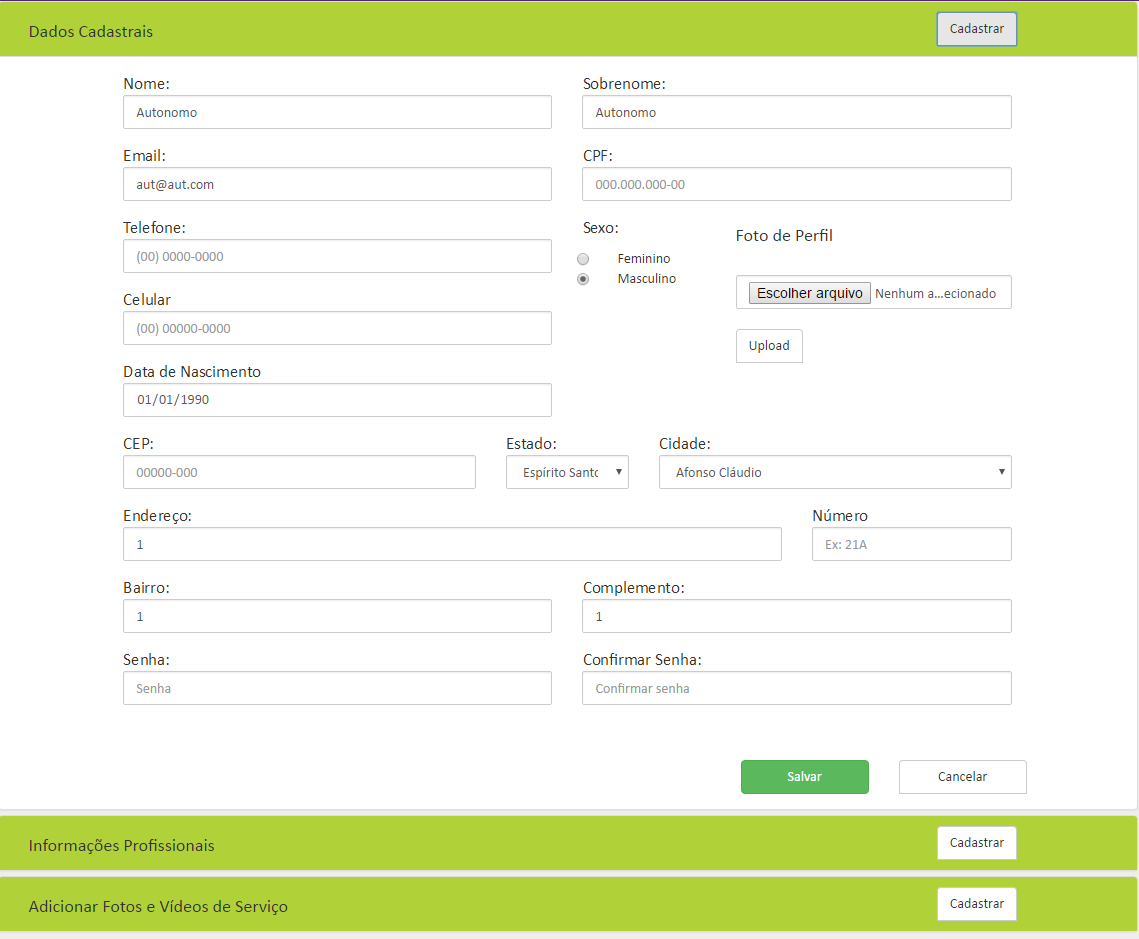


Figura - Tela de protótipo de cadastro de dados pessoais

A figura 32 representa a segunda parte do cadastro profissional do autônomo.

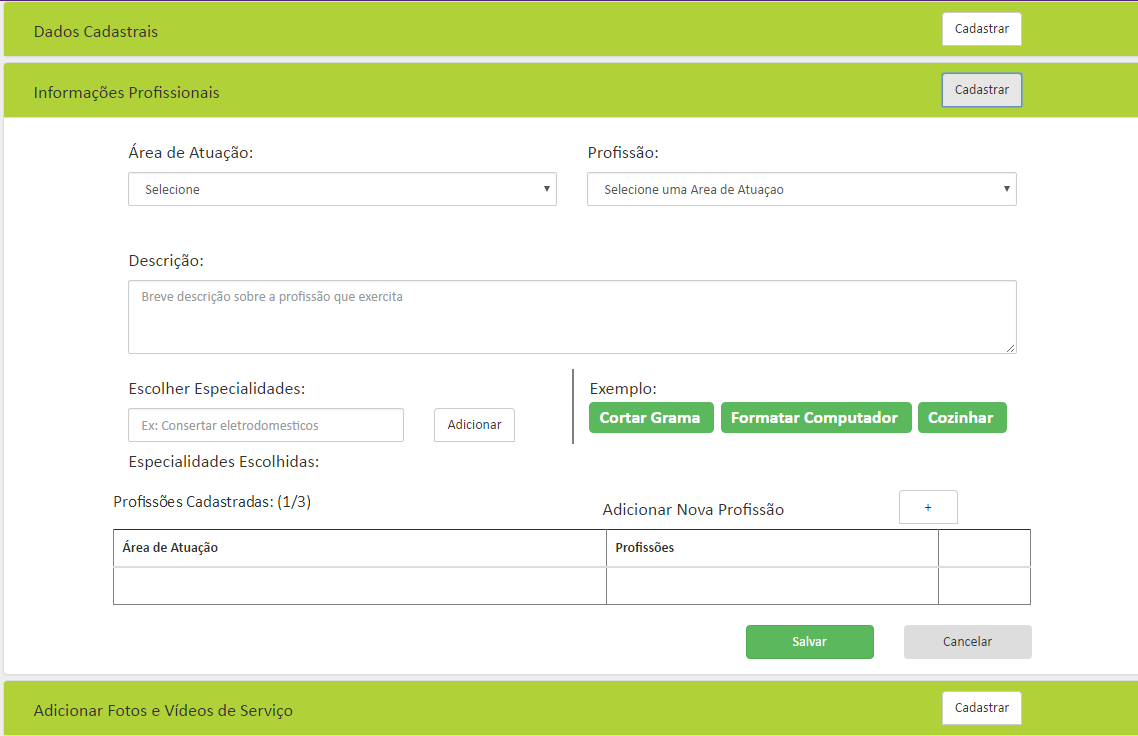


Figura - Tela de cadastro profissional do autônomo

A figura 33 ilustra a última parte do cadastro do autônomo, o cadastro de mídia.

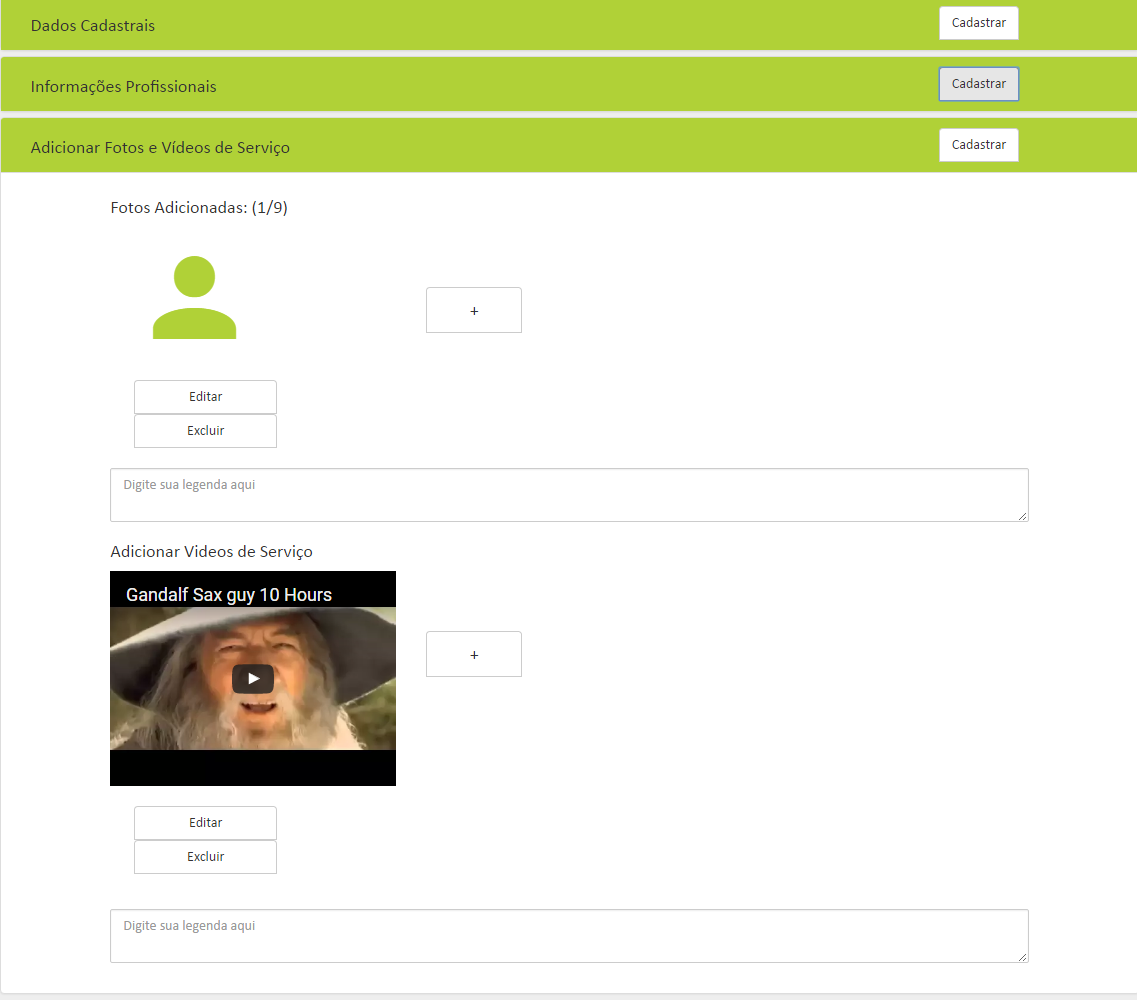


Figura - Tela de cadastro de mídia

Todo esse desenvolvimento cadastral do autônomo teve uma programação bem semelhante durante as três fases ilustradas nas figuras 31, 32 e 33. Houve mudanças apenas em nome de objetos e nos seus atributos, portanto, será explicado apenas o trecho do cadastro de dados profissionais do autônomo, especificamente, de suas especialidades.

Ao ser carregada, a página de cadastro de dados pessoais do autônomo verifica se o autônomo está logado, caso esteja, três funções são chamadas: CarregarFormulario(), CarregarEspecialidade() e CarregarProfissoes(). Seus dados pessoais, profissionais e as mídias serão preenchidas automaticamente nos campos dos formulários facilitando, no futuro, a atualização de alguma informação. A figura 34 ilustra como esse processo foi feito.

if (Session["LOGIN"] != null) {

Usuario usuario = (Usuario)Session["LOGIN"];

if (usuario.TipoUsuario.Codigo== 1) {

CarregarFormulario();

CarregarEspecialidades();

CarregarProfissoes();

}

Figura – Verificação de usuário e preenchimento de formulário

Caso o usuário estiver fazendo um cadastro, o processo será diferente. Será criado um objeto “especialidade” do tipo “Especialidade” que receberá todas as informações do formulário. Após o objeto especialidade ter sido preenchido, um método da classe de persistência, método responsável por inserir as informações no banco de dados do sistema, para armazenar essas informações no banco de dados. Destacado em amarelo na primeira linha da figura 35 está a criação do objeto especialidade, na última, a chamada do método de persistência.

Especialidade especialidade = new Especialidade();

especialidade.Nome = txtEscolherEspecialidade.Text;

especialidade.Autonomo = autonomo;

Profissao profissao = new Profissao();

profissao.Codigo = Convert.ToInt32(ddlProfissao.SelectedValue);

especialidade.Profissao= profissao;

EspecialidadeBD.Insert(especialidade);

Figura – Cadastro de especialidade do autônomo

Pelo fato do módulo leilão reverso ser a principal funcionalidade do sistema, decidiu-se por explicar as principais etapas do desenvolvimento. A seguir, ilustrado na figura 36, verifica-se que os usuários Cliente e Autônomo têm acesso através de uma área às pesquisas de leilões, criação, visualizar leilões abertos e seus históricos. Vale ressaltar que a área de leilão oferece todas as funcionalidades para ambos os usuários, salvo “Dar lance”, funcionalidade exclusiva do Autônomo.

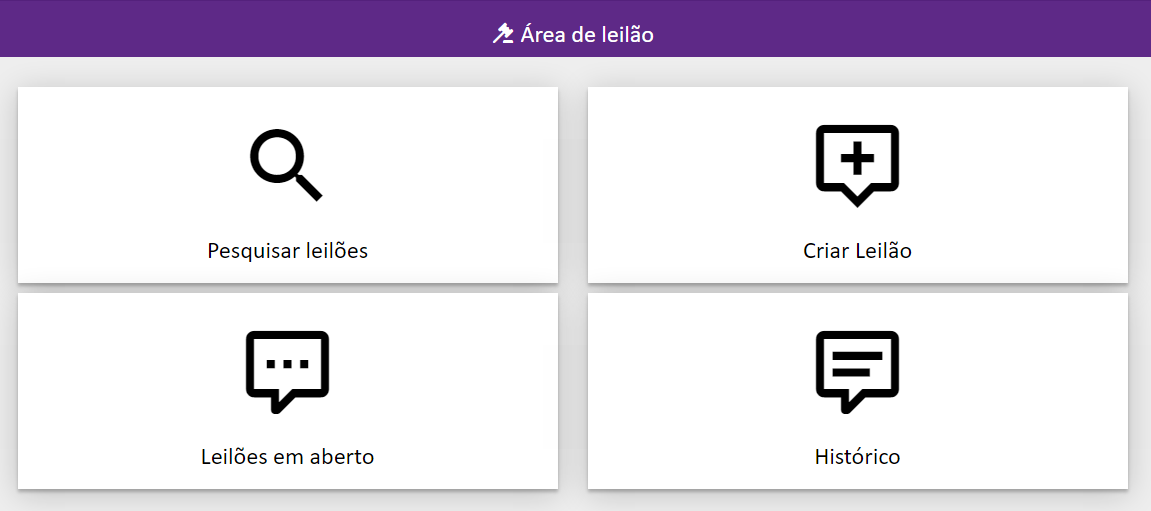


Figura - Tela da área de leilão dos usuários Cliente e Autônomo

Uma das principais funcionalidades do Cliente e do Autônomo é a capacidade de criar leilão reverso preenchendo um formulário como ilustra a figura 37.



Figura – Formulário de criação de leilão

No desenvolvimento *front-end*, não só na criação do leilão, mas em todas as telas que contenham formulário, utilizou-se o controle *RequiredFieldValidator*. Esse controle é aplicado em campos específicos do formulário e garante que estes não poderão estar vazios. Desse modo, o usuário preenche corretamente as informações prevenindo erros de preenchimento. Destacado em amarelo, a linha do código ilustrada na figura 38 mostra a aplicação desse controle sendo aplicado no campo “Título do Serviço” do formulário da figura 18. Parte interessante desse controle é o uso do atributo “*ErrorMessage”* que permite definir uma mensagem de erro ao usuário.

<div class="col-lg-6 col-lg-offset-1">

<asp:ValidationSummary ID="vlsSumario" runat="server" CssClass="alert alert-danger" HeaderText="Erros:" />

<asp:Label ID="lblServicoReq" runat="server" Text="Título do serviço:" Font-Size="Large"></asp:Label>

<asp:RequiredFieldValidator ID="rfvServicoReq" runat="server" ErrorMessage="Título obrigatório" ControlToValidate="txtServicoReq" Text="\*" ForeColor="Red"></asp:RequiredFieldValidator>

<div class="input-group">

<asp:TextBox ID="txtServicoReq" CssClass="form-control col-xs-12" runat="server" placeholder="Ex: pintar casa, formatar computador" MaxLength="29" Width="350"></asp:TextBox>

</div>

</div>

Figura – Componente validador de campos

No *back-end,* as informações preenchidas e validadas no formulário estão sendo coletadas e armazenadas num banco de dados. Quando o usuário finaliza o preenchimento e clica no botão “Criar leilão”, botão verde localizado ao final do formulário, é ativado uma sequência de etapas para fazer a inserção dos dados no banco. O código da figura 39 exibe como foi feita essa sequência de etapas. Ao clicar no botão “Criar leilão”, o método chamado lbtCriarLeilao\_Click é acionado. Esse método instancia um objeto chamado “lei” do tipo “Leilao”. Esse objeto recebe os valores vindo do formulário. Destacado em amarelo, a primeira linha mostra como é feita a instanciação de um objeto, a segunda, a atribuição do valor a ele.

Na primeira linha destacada em verde, está a chamada de um método de uma classe de persistência. Inserido numa estrutura de decisão “if”, o método valida se as informações foram corretamente inseridas. Caso tudo esteja correto, o método retorna o valor 0. Ao final da estrutura de decisão será exibida uma mensagem ao usuário informando que o leilão começou como mostra a segunda linha destacada em verde.

protected void lbtCriarLeilao\_Click(object sender, EventArgs e)

{

Leilao lei = new Leilao();

lei.Titulo = txtServicoReq.Text;

lei.DataHoraInicio = DateTime.Now;

lei.DataPrevisao = DateTime.Now.AddMonths(1);

lei.DataHoraFim = Convert.ToDateTime(txtPrazoFinal.Text);

lei.Descricao = txtDescricaoCompleta.Text;

lei.Encerrado = 'N';

Atuacao atu = new Atuacao();

atu.CodigoAtuacao = Convert.ToInt32(ddlAtuacao.SelectedValue);

lei.Atuacao = atu;

Cidade cid = new Cidade();

cid.CodigoCidade = Convert.ToInt32(ddlCidade.SelectedValue);

lei.CodigoCid = cid;

Usuario usu = (Usuario)Session["LOGIN"];

lei.CodigoUser = usu;

if (LeilaoDB.InsertLeilao(lei) == 0)

{

txtServicoReq.Text = string.Empty;

txtDescricaoCompleta.Text = string.Empty;

txtPrazoFinal.Text = string.Empty;

ddlEstado.SelectedIndex = 0;

ddlCidade.SelectedIndex = 0;

ddlAtuacao.SelectedIndex = 0;

Response.Write("<script>alert('Começou o leilão!');</script>");

Response.Write("<script>window.location.href='HistoricoLeilao.aspx';</script>");

}

}

Figura – Desenvolvimento *back-end* para a criação do leilão

Após a criação do leilão, tanto o Autônomo e o Cliente terão acesso aos leilões criados por meio da tela “Histórico de leilão”. A figura 40 ilustra essa com dois leilões criado.

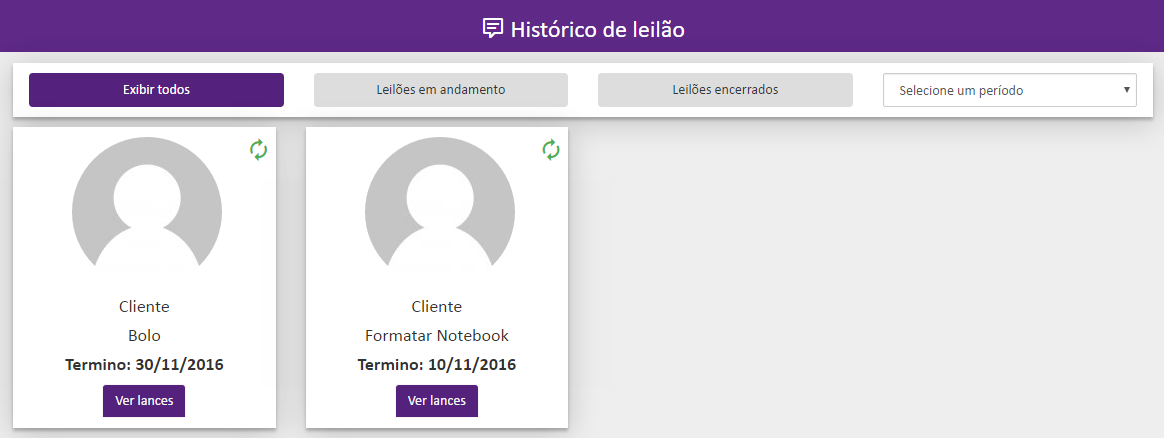


Figura - Tela de histórico de leilão

Outra funcionalidade importante e exclusiva do autônomo é dar lance em leilões em aberto. A figura 41 ilustra a tela em que um autônomo envia um lance a um leilão.

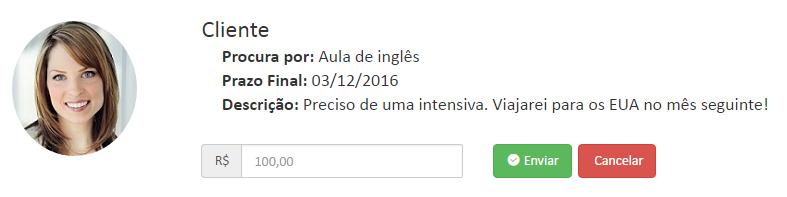


Figura – Tela de envio de lances

No desenvolvimento *back-end*, todas as informações sobre o leilão como, por exemplo, o serviço procurado e a descrição estão sendo resgatadas do banco de dados do sistema.

O código para gerar a tela da figura 41 foi divido em duas partes. A primeira parte, ilustrada na figura 42, refere-se às informações do leilão como a imagem do cliente e todas as informações escritas a direita dela. As informações resgatadas do banco de dados foram armazenadas num objeto “lei” do tipo “Leilao” e colocadas entre um código HTML, destacado em amarelo na figura, para que a tela fosse gerada corretamente.

lblLeilao.Text =

"<div class='col-xs-2 col-xs-offset-1'>" +

"<img src = 'img/imgOfertaServicoCliente.jpg' class='img1' />" +

"</div>" +

"<div class='col-xs-8'>" +

"<h3 class='textos2'>" + lei.CodigoUser.Nome + "</h3>" +

"<h4 class='textos2'><b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Procura por: </b> " + lei.Titulo.ToString() + "</h4>" +

"<h4 class='textos2'><b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Prazo Final: </b>" + lei.DataHoraFim.ToShortDateString().ToString() + "</h4>" +

"<h4 class='textos2'><b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Descrição: </b>" + lei.Descricao.ToString() + "</h4>" +

"</div>";

Figura – Código para gerar as informações sobre o leilão

A segunda parte, ilustrada na figura 43, refere-se ao campo em que um valor será passado e aos dois botões, Enviar e Cancelar em verde e vermelho respectivamente. Esses elementos, a princípio, ficam escondidos do autônomo até que sejam validados. É necessário verificar se o autônomo está tentando dar lance no próprio leilão, visto que não há sentido nisso.

string darLance =

"<input id='ValorLance' name='ValLance' class='form-control' aria-label='Amount (to the nearest dollar)' placeholder='100,00' />" +

"<button id='btnEnviarOferta' class='btn btn-success' onclick='lanceEnvia(); return false;'\"><i class='mdi mdi-check-circle' style='color: white'></i>&nbsp;Enviar</button>&nbsp;&nbsp;" +

"<button id='btnCancelarOferta' class='btn btn-danger' data-dismiss='modal' onclick='limpaLance(); return false;'>&nbsp;Cancelar</button>" +

Figura – Código para gerar o campo do lance e os botões

Uma vez que o lance foi enviado, ele será exibido para o cliente ou autônomo que iniciou o leilão em forma de um cartão que pode ser visto ao lado esquerdo da seta vermelha na figura 44.

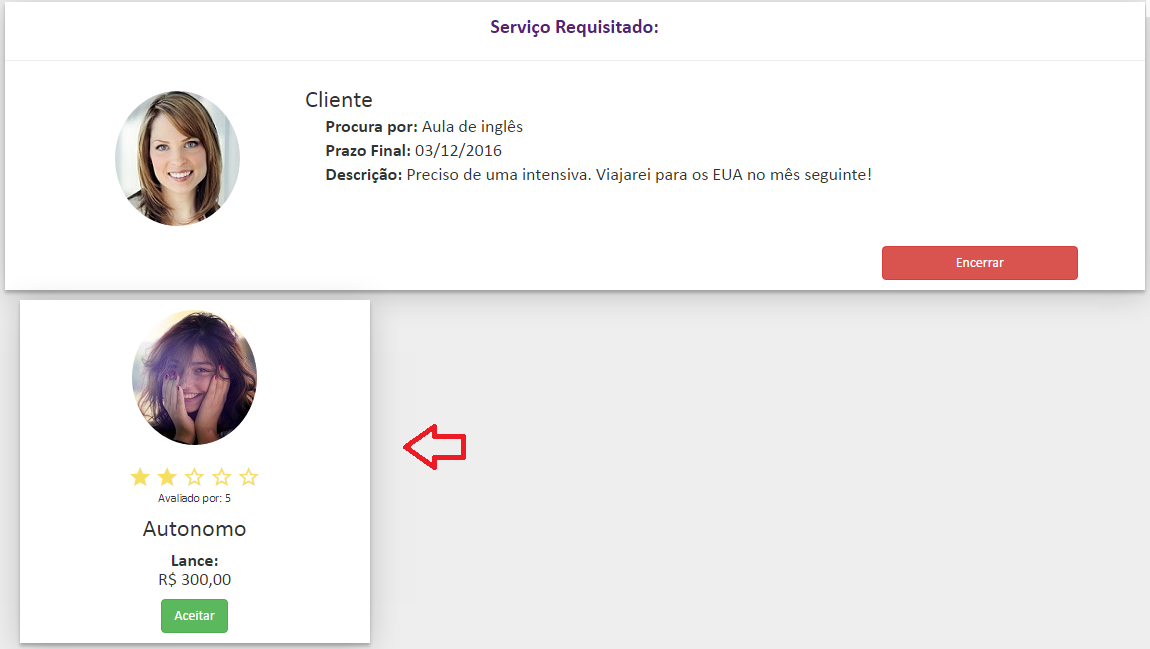


Figura – Cartão gerado a partir de um lance

Todos os lances estão sendo armazenado num *DataSet* chamado ds, objeto que armazena um conjunto de dados vindo do banco de dados. Os dados armazenados nele são percorridos por um uma estrutura de repetição *foreach*. No código destacado em amarelo, na figura 45, por meio do comando dr[“USU\_NOME”].toString(), o nome do autônomo é acessado no DataSet e exibido numa estrutura HTML.

DataSet ds = LeilaoDB.SelecionarLeilaoUsuarioPorCodigoLeilao(leilao);

foreach (DataRow dr in ds.Tables[0].Rows)

"<h3 style='margin-top: 0px'><div id='nomeAut'>" + dr["USU\_NOME"].ToString() + "</div></h3> " +

"<h4><b>Lance:</b></br> <div id='lanceAut'>R$ " + dr["AUL\_VALOR"].ToString() + "</div> </h4> " + "</div> " +

"<div class='row'> " +

botao +

"</div> " +

Figura – Lance de um autônomo em forma de cartão

Por último, a figura 46 ilustra o menu lateral do autônomo fornecendo-o acesso a todas as funcionalidades do sistema disponível para esse perfil de usuário, neste caso, Autônomo.

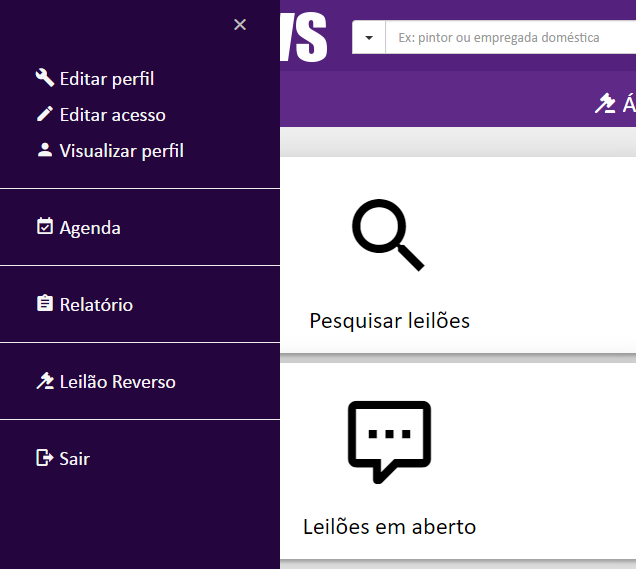


Figura - Tela de menu lateral

Por se tratar de um desenvolvimento semelhante, a figura 47 ilustra um exemplo de como é feito a programação de um item do menu, a opção “Leilão Reverso”. Esse botão foi feito usando o “*LinkButton*”, componente *asp* pois permite usar ícone e outras funcionalidades. A figura 46 ilustra a aplicação desse botão na interface gráfica do usuário. Destacado em amarelo, entre a tag <asp:LinkButton> </asp:LinkButton>, está a tag <i>, marcação HTML para uso de ícones.

<asp:LinkButton ID="btnLeilao" runat="server" CssClass="btn btn-menu" OnClick="btnLeilao\_Click" CausesValidation="false"><span style="font-size:20px"><i class="mdi mdi-gavel"></i>&nbsp;Leilão Reverso </span></asp:LinkButton>

Figura – LinkButton na interface gráfica do usuário

Quando o botão “Leilão reverso” é clicado, um método chamado “btnLeilao\_Click*”* executa uma ação de redirecionamento para a página “AreaLeilao.aspx” por meio do comando *Response.Redirect* ilustrado na figura 48.

protected void btnLeilao\_Click(object sender, EventArgs e) {

Response.Redirect("AreaLeilao.aspx");

}

Figura – Redirecionamento do botão para área de leilão

# 9 SEGURANÇA

Atualmente, a informação é o ativo mais importante de uma empresa e não há informação completamente segura. Baseado nessa premissa, aplicaram-se técnicas de segurança da informação no sistema D2VS a fim de proteger as informações dos usuários. De acordo com Moraes (2010, p.19), “a segurança da informação pode ser definida como um processo de proteger a informação do mau uso tanto acidental como intencional, por pessoas internas ou externas à organização, incluindo empregados, consultores e hackers.”

## 9.1 Aplicação

A segurança do sistema foi baseada na OWASP, “uma comunidade aberta dedicada a capacitar as organizações a desenvolver, adquirir e manter aplicações confiáveis.” (OWASP, 2013, p.3). A comunidade apresenta, em forma de lista, os riscos de segurança em aplicações e o impacto do risco para o negócio. Baseado nessa lista, foram escolhidos alguns itens que possuem relação direta com o desenvolvimento do presente sistema. Os itens retirados do OWASP que serão descritos no decorrer dos próximos tópicos são: A1 – Injeção, A2 – Quebra de autenticação e gerenciamento de sessão, A6 – Exposição de dados sensíveis e A7 – Falta de função para o controle do nível de acesso.

## 9.2 A1 – Injeção

Considerando-se que ataques do tipo injeção SQL são muito comuns, decidiu-se por aplicar uma técnica de prevenção no sistema. As falhas de injeção SQL ocorrem quando dados não confiáveis são enviados para um interpretador como parte de uma consulta (OWASP, 2013). Visto que há muitos formulários no sistema e que dados confiáveis pudessem ser passados como parte de uma consulta, adotou-se o uso de parâmetros. Qualquer texto que se entre em um parâmetro será tratado como uma informação de campo e não como parte de uma declaração, deste modo, não há como inserir um código para praticar a injeção SQL. Destacado em amarelo, a figura 49 ilustra a aplicação da parametrização.

string sql = "SELECT \* FROM USU\_USUARIO WHERE USU\_EMAIL=?EMAIL and USU\_SENHA=?SENHA;";

conexao = Mapped.Connection();

comando = Mapped.Command(sql, conexao);

comando.Parameters.Add(Mapped.Parameter("?EMAIL", usu.Email));

Figura – Parametrização para prevenção de injeção SQL

## 9.3 A2 - Quebra de autenticação e gerenciamento de sessão

Segundo a OWASP (2013), as funções da aplicação relacionadas com autenticação e gerenciamento de sessão, implementadas de forma incorreta, permite que atacantes comprometam senhas e *tokens* de sessão para assumir a identidade de outros usuários. Para evitar que um atacante comprometa a senha dos usuários do sistema, aplicou-se uma função *Hash,* funçãoque“embaralha” a senha do usuário e muito difícil de ser “desembaralhada”, mostrada na figura 50.

public static string HashTexto(string texto)

{

HashAlgorithm algoritmo = HashAlgorithm.Create("SHA512");

if (algoritmo == null)

{

throw new ArgumentException("Nome da hash incorreta", "nomehash");

}

byte[] hash = algoritmo.ComputeHash(System.Text.Encoding.UTF8.GetBytes(texto));

return Convert.ToBase64String(hash);

}

Figura - Método de aplicação de criptografia à senha do usuário

## 9.3 A6 – Exposição de Dados Sensíveis

Durante um cadastro de usuário no sistema, muitos dados são inseridos no banco de dados. Uma informação importante armazenada no banco é a senha do usuário. A fim de evitar que ela seja descoberta durante um ataque do tipo *man-in-the-middle*, ataque em que um usuário intercepta os dados entre cliente e servidor, utilizou-se o método de criptografia de senha *SHA-512* mostrado na figura 50.

## 9.4 A7 – Falta de função para controle do nível de acesso

Visto que há três perfis de acesso no sistema, surgiu a necessidade de implementar um controle para que cada perfil acessasse apenas as funções designadas a cada um. Por meio de uma validação no carregamento da página, foi verificado o perfil logado no sistema. Caso o perfil logado tente acessar outra página a qual não tem acesso, este será redirecionado para a tela principal do sistema. A figura 51 ilustra uma validação feita no carregamento de uma página do Administrador do sistema. A estrutura de decisão if (usu.CodigoTipo.CodigoTipo != 3), destacada em cor amarela, está verificando se o usuário logado possui o código de perfil diferente de 3, que se refere ao Administrador do sistema. Se qualquer outro usuário logado com o código 1 ou 2, Autônomo ou Cliente, respectivamente, tentar acessar a área do administrador, ele será redirecionado para a tela principal do sistema. Então qualquer acesso a funcionalidade do administrador será interrompido.

public partial class AreaAdministrador : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["LOGIN"] != null)

{

Usuario usu = (Usuario)Session["LOGIN"];

if (usu.CodigoTipo.CodigoTipo != 3)

{

Response.Redirect("Index.aspx");

}

}

Figura - Validação do perfil logado no sistema

## 9.5 Estratégia de backup

Para que toda a equipe de desenvolvimento do sistema tivesse acesso ao código do sistema, bem como a documentação, utilizou-se a ferramenta GIT, uma ferramenta de controle de versão. Todo código e documentação está armazenado num repositório on-line no site [www.bitbucket.com](http://www.bitbucket.com) facilitando, para todos os membros autorizados, o acesso a qualquer hora e lugar. Cada membro possui em seu computador pessoal todo o conteúdo presente no repositório, portanto, se algum imprevisto surgir com um membro, os outros têm uma cópia.

Além disso, durante todo o desenvolvimento, um membro da equipe manteve uma cópia do projeto salvo em uma unidade externa de armazenamento permitindo, pois, uma restauração.

# 10 TESTE

Com o objetivo de analisar o comportamento do sistema e planejar futuras implementações, realizou-se um teste com um usuário da Fatec Guaratinguetá totalmente leigo no sistema. Além desse teste, o sistema foi submetido a quatro avaliações pelos professores orientadores ao longo do segundo semestre 2016, fornecendo sugestões de correções e melhorias para o sistema. As observações coletadas serão listadas e comentadas e, no caso de detecção de problemas, comenta-se se já foram corrigidas, caso contrário, a observação será tratada no tópico subsequente.

* Observou-se que o sistema possui boa funcionalidade de pesquisa tanto para trabalhadores autônomos quanto para leilão. Além disso, o acesso ao formulário de cadastro está visível e seu preenchimento não está simples. Os desenvolvedores já melhoraram os rótulos e também simplificaram os formulários de cadastro.
* Mesmo embora as cores sejam agradáveis e compatíveis com a proposta do sistema, aconselha-se que a cor roxa (#54217C), presente na cartela de ícones da figura 16, seja usada em menor quantidade para tornar o visual mais agradável.
* Após efetuar o login, o usuário é redirecionado para a página principal do sistema, a Index.aspx. Tendo em vista a melhora da usabilidade, esse usuário poderia ser redirecionado para uma tela de painel de controle (*dashboard*), facilitando, pois, a navegação para as demais funcionalidades do sistema.
* Uma vez cadastrado, percebeu-se certa desorganização ao encerrar a sessão do usuário com o sistema. Há um botão chamado **sair,** que encerra a sessão, localizado no menu lateral, no entanto, a sugestão foi para que ele aparecesse quando o usuário clicasse em sua foto de perfil, pois sistemas semelhantes trabalham assim e isso melhoraria a usabilidade.
* Na tela de envio de lances para leilões, sugeriu-se que houvesse, além do valor, uma caixa de texto para escrever uma justificativa para o lance, facilitando a escolha do usuário que iniciou o leilão.
* Por último, observou-se que os autônomos que deram lances em leilões poderiam receber uma média de lances para que ele pudesse dar um lance com um valor compatível com os demais e aumentar sua chance de vitória.

# SUGESTÕES PARA IMPLEMENTAÇÕES FUTURAS

Baseada nas informações coletadas pelo teste e pelas observações dos professores, muitas sugestões foram propostas para aplicar no sistema. Outras sugestões, no entanto, foram concebidas após o teste e as observações dos professores como, por exemplo, permitir que eu sistema tenha suporte a um novo tipo de mídia. A seguir, serão discutidas essas sugestões.

O profissional autônomo pode inserir fotos e vídeos de seus serviços para atrair clientes. No entanto, esse sistema limita-se a esses dois tipos de mídias. A partir dessa observação, considerou-se adotar novas mídias, como, por exemplo, *podcasts*. Desse modo, esse sistema cresce com ainda mais diferenciais.

As novas tendências de desenvolvimento *front-end* sugerem visuais menos pesados e mais limpos. As mesmas cores adotadas no planejamento continuarão a ser usadas, mas, especificamente a cor roxa (#54217C), será usada com menos frequência para tornar o visual mais “leve” para o usuário.

A fim de melhorar a navegação dos usuários cadastrados no sistema, considerou-se o desenvolvimento de uma tela que funciona como um painel de controle, isso deve facilitar a navegação dos usuários pelas diversas funcionalidades do sistema.

Pretende-se alocar a funcionalidade fazer logout, nomeada como **Sair** para o usuário do sistema, ao ser clicada em sua foto ao invés de estar localizada no menu lateral.

Outra sugestão para implementação foi inserir uma área de texto junto ao valor do lance para que um autônomo justifique o valor dado pelo serviço em questão.

A equipe de desenvolvimento preocupa-se com a usabilidade do sistema. Baseada nessa preocupação, percebeu-se que leilões encerrados não poderiam ser reabertos gerando um grande retrabalho para o usuário. Para ele isso é ruim, pois deve preencher novamente um conjunto de informações repetidas, algo que não aconteceria se ele pudesse reabrir o leilão. Juntamente com a reabertura, surgiu a necessidade de editar as informações do leilão, caso o usuário decida por mudar algum detalhe.

Por último, é interessante que o sistema mantenha atualizado para o autônomo a média do valor dos lances de um leilão.

# CONSIDERAÇÕES FINAIS

Observou-se que o desenvolvimento desse sistema se deu em momento oportuno devido a atual situação econômica do Brasil. Sem uma previsão exata de melhora, e com o contínuo aumento no número de trabalhadores autônomos, acredita-se que esse sistema terá boa utilidade e aceitação pelos próximos anos. Ainda, pessoas com pouca experiência profissional encontram uma oportunidade de iniciar suas atividades profissionais utilizando o sistema.

Não é parte do escopo desse projeto fazer a intermediação do pagamento entre cliente e autônomo, isso é algo combinado entre as partes após comunicação. Entretanto, o pagamento, por algum motivo, pode não acontecer e se essa ação ocorrer constantemente, a imagem do sistema poderá ficar comprometida. A partir disso, observou-se que a intermediação entre eles poderia se tornar um módulo do sistema, a fim de passar mais segurança e confiança ao usuário, que poderá trabalhar de forma mais segura com o pagamento. Consequentemente, isso evitará a má imagem do sistema.

Muitas funcionalidades foram propostas para o desenvolvimento, porém, os desenvolvedores estavam em fase de aprendizagem e de domínio das tecnologias usadas no sistema e, portanto, algumas não puderam ser concluídas. Isso abriu a possibilidade para os desenvolvedores conhecer e aprofundar o estudo dessas tecnologias e, partir desse conhecimento, ser capaz de desenvolver o que não foi possível.

Por fim, mesmo embora todas as funcionalidades não pudessem ser desenvolvidas, o objetivo de desenvolver um sistema de divulgação on-line de serviços e de leilões, bem como ajudá-lo em sua rotina foi alcançado.

# 

# REFERÊNCIAS BIBLIOGRÁFICAS

AIRES, João.; ORLOVSKI, Regiane.; RIBEIRO, Sergio. Desenvolvimento de sistema de gerenciamento e controle para academias. Disponível em:< http://semanaacademica.org.br/artigo/desenvolvimento-de-sistema-de-gerenciamento-e-controle-para-academias>. Acesso em: 26 out. 2016.

BISPO, Diogo.UMBUZEIRO, Gisela.; ZABEU, Márcio.; BORGES, Marcos. REVISTA BRASILEIRA DE INFORMÁTICA NA EDUCAÇÃO. Desenvolvimento de Jogo Educacional sobre Ecotoxicologia Utilizando HTML5, v.20, n.1, 2012.

CARVALHO, Vinícius. **MySQL: comece com o principal banco de dados open source do mercado**. São Paulo: Casa do Código, 2015.

CASTELLI, Ian. Trello: como esta ferramenta pode ajudar você a organizar a sua vida

Disponível em: <http://www.tecmundo.com.br/organizacao/75128-trello-ferramenta-ajudar-voce-organizar-vida.htm> Acesso em 26 out. 2016.

GUEDES, Gilleanes T. A. **UML 2: guia prático**. São Paulo: Novatec Editora, 2007.

GUEDES, Gilleanes. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec Editora, 2011.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegra: Bookman, 2009.

LIMA, Edwin. **C# e .Net para desenvolvedores**. Rio de Janeiro: Campus, 2002.

MAZZA, Lucas. **HTML5 E CSS3: Domine a web do futuro**. São Paulo: Casa do Código, 2012.

MORAES, Alexandre Fernandes de. **Segurança em Redes**.São Paulo: Érica, 2010.

OWASP: os dez riscos de segurança mais críticos em aplicações web. Disponível em: <https://www.owasp.org/images/9/9c/OWASP\_Top\_10\_2013\_PT-BR.pdf> Acesso em 10 nov. 2016

ROGERS, Y.; SHARP, H.; PREECE, J. **Design de interação: além da interação humano-computador.** 3. ed. Porto Alegre: Bookman, 2013.

SABBAGH, Rafael. **Scrum: gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 8 ed. São Paulo, 2007.

**AGRADECIMENTOS**

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Aos nossos pais, que proporcionaram toda condição necessária para a conclusão desta faculdade.

Aos professores Allbert, Bruno, Claudemir e José, pelo inestimável apoio na orientação deste trabalho.

Aos colegas de classe, com quem nesses anos de estudo tivemos a felicidade de conviver.

# APÊNDICE A

Neste apêndice, encontram-se as descrições dos casos de uso do autor cliente, autônomo e administrador do sistema D2VS.

|  |  |
| --- | --- |
| Caso de Uso | Fazer login |
| Visão geral | O usuário acessa o sistema. |
| Objetivo | Acessar o perfil. |
| Ator primário | Todos os atores. |
| Pré-condição | O cliente deve acessar o sistema e clicar em “Entrar”. |
| Pós-condição | Será exibida uma tela para preencher o e-mail e a senha. |
| Sequências típica de eventos | O cliente deverá acessar o sistema e clicar em “entrar”. O usuário deverá inserir seu e-mail e sua senha para ter acesso ao sistema. |
| Sequência alternativa de eventos | O usuário não possui um cadastro ou o e-mail e a senha são inválidos. |

|  |  |
| --- | --- |
| Caso de Uso | Fazer logout |
| Visão geral | O usuário encerra sua sessão no sistema. |
| Objetivo | Encerrar a sessão do usuário. |
| Ator primário | Todos os atores. |
| Pré-condição | O usuário deve estar logado e clicar em “Sair”. |
| Pós-condição | O usuário encerrará sua sessão com o sistema. |
| Sequências típica de eventos | O usuário deve navegar até a opção “Sair” e dar um clique. |
| Sequência alternativa de eventos | O botão “Sair” pode não estar funcionando. |

|  |  |
| --- | --- |
| Caso de Uso | Recuperar senha |
| Visão geral | O usuário recupera sua senha. |
| Objetivo | Recuperar senha do usuário. |
| Ator primário | Todos os atores. |
| Pré-condição | O usuário deve acessar o sistema e clicar em “Entrar”. |
| Pós-condição | O usuário deve clicar em “Esqueci minha senha”. |
| Sequências típica de eventos | O usuário acessa o sistema e clica no botão “Entrar”. Após acesso, o usuário deve clicar em “Esqueci minha senha”. O usuário deverá informar o e-mail cadastrado no sistema para receber uma nova senha. |
| Sequência alternativa de eventos | O e-mail pode não estar cadastrado no sistema. |

|  |  |
| --- | --- |
| Caso de Uso | Pesquisar profissional |
| Visão geral | Busca um profissional autônomo no sistema. |
| Objetivo | Buscar profissionais no sistema. |
| Ator primário | Cliente e Autônomo. |
| Pré-condição | O cliente deve acessar o sistema. |
| Pós-condição | A tela inicial do sistema exibirá um campo de pesquisa de profissional. |
| Sequências típica de eventos | O cliente ou autônomo deverá acessar o sistema digitar, no campo de pesquisa, a profissão ou a especialidade do autônomo. |
| Sequência alternativa de eventos | Se não houver o profissional na pesquisa, a pesquisa não retornará profissionais. |

|  |  |
| --- | --- |
| Caso de Uso | Entrar em contato com o D2VS |
| Visão geral | O usuário entra em contato com os responsáveis do sistema. |
| Objetivo | Contatar os responsáveis do sistema. |
| Ator primário | Cliente e Autônomo. |
| Pré-condição | O usuário deve acessar o sistema. |
| Pós-condição | O usuário deve clicar, na tela inicial, no link contato que se encontrar no rodapé do sistema. |
| Sequências típica de eventos | O usuário acessa o sistema e navega até o rodapé da tela inicial. Após acesso, o usuário deve clicar em “contato”. O usuário deverá informar seu nome, e-mail, um assunto e uma descrição. O usuário deve clicar em enviar. |
| Sequência alternativa de eventos | O sistema pode estar com problema e não enviar a mensagem do usuário. |

|  |  |
| --- | --- |
| Caso de Uso | Recomendar autônomo |
| Visão geral | O autônomo ou cliente recomenda um autônomo |
| Objetivo | Recomendar um autônomo |
| Ator primário | Autônomo e Cliente. |
| Pré-condição | Um dos usuários precisa criar um leilão. |
| Pós-condição | Após o termino do leilão, o usuário poderá recomendar um autônomo. |
| Sequências típica de eventos | O usuário cria um leilão. Caso haja um vencedor, o usuário deverá dar um feedback ao sistema informando se o serviço foi concluído. Durante o feedback, o usuário deverá informar se recomenda o autônomo. Caso sim, essa informação será exibida no perfil do autônomo. |
| Sequência alternativa de eventos | O autônomo a ser recomendado pode estar com a conta inativa. |

|  |  |
| --- | --- |
| Caso de Uso | Avaliar autônomo |
| Visão geral | O cliente ou autônomo avalia um autônomo. |
| Objetivo | Avaliar um trabalhador autônomo. |
| Ator primário | Cliente e autônomo. |
| Pré-condição | Um dos usuários precisa criar um leilão. |
| Pós-condição | Após o termino do leilão, o usuário poderá recomendar um autônomo. |
| Sequências típica de eventos | O usuário cria um leilão. Caso haja um vencedor, o usuário deverá dar um feedback ao sistema informando se o serviço foi concluído. Durante o feedback, o usuário poderá fazer uma avaliação ao autônomo. Caso tenha feito, essa informação será exibida no perfil do autônomo. |
| Sequência alternativa de eventos | O autônomo a ser avaliado pode estar com a conta inativa. |

|  |  |
| --- | --- |
| Caso de Uso | Cadastrar administrador |
| Visão geral | Cadastra um usuário como administrador. |
| Objetivo | Cadastrar administrador no sistema. |
| Ator primário | Administrador |
| Pré-condição | O administrador deverá fazer o login no sistema. |
| Pós-condição | O administrador deverá navegar até a tela de cadastro de administrador. |
| Sequências típica de eventos | O Administrador deve acessar o sistema e realizar o login. O administrador deverá clicar no menu de navegação e clicar em “cadastrar administrador”. O administrador deve fornecer o CPF, nome, e-mail e senha. |
| Sequência alternativa de eventos | O sistema pode não estar funcionando. |

|  |  |
| --- | --- |
| Caso de Uso | Editar perfil do administrador |
| Visão geral | O administrador edita seu perfil. |
| Objetivo | Editar o perfil do administrador. |
| Ator primário | Administrador |
| Pré-condição | O administrador deverá fazer o login no sistema. |
| Pós-condição | O administrador deverá navegar até a tela de edição de perfil. |
| Sequências típica de eventos | O Administrador deve acessar o sistema e realizar o login. O administrador deverá clicar no menu de navegação e clicar em “editar perfil”. O administrador deve editar os campos que deseja. |
| Sequência alternativa de eventos | A tela de edição pode estar com problemas. |

|  |  |
| --- | --- |
| Caso de uso | Pesquisar usuários cadastrados |
| Visão geral | O administrador pesquisa clientes e autônomos cadastrados no sistema. |
| Objetivo | Pesquisar usuários no sistema. |
| Ator primário | Administrador |
| Pré-condição | O administrador precisa estar logado no sistema para pesquisar. |
| Pós-condição | O administrador entrará numa tela de busca de usuários. |
| Sequências típica de eventos | O administrador deve acessar o sistema e realizar o login. Após o acesso, o administrador estará numa tela de pesquisa. O administrador deverá digitar o usuário que ele deseja pesquisar podendo usar filtros para agilizar a busca. |
| Sequência alternativa de eventos | Uma falha no banco de dados pode comprometer a lista de usuários do sistema. |

|  |  |
| --- | --- |
| Caso de Uso | Editar cadastro de usuários |
| Visão geral | O administrador edita os usuários cadastrados no sistema. |
| Objetivo | Ativar e inativar os usuários do sistema. |
| Ator primário | Administrador |
| Pré-condição | O administrador precisa estar cadastrado no banco de dados do sistema e fazer o login no sistema. |
| Pós-condição | O administrador entrará numa tela de visualização de usuários cadastrados. |
| Sequências típica de eventos | O Administrador precisa acessar o sistema e inserir seu login e a senha. O administrador deverá selecionar o usuário que ele deseja ativar ou inativar a conta. |
| Sequência alternativa de eventos | O administrador inseriu o login ou senha errada. A pesquisa pelo usuário pode resultar negativa. |

|  |  |
| --- | --- |
| Caso de Uso | Cadastrar perfil do cliente |
| Visão geral | O cliente cadastra o seu perfil no sistema. |
| Objetivo | Cadastrar o perfil do cliente. |
| Ator primário | Cliente. |
| Pré-condição | O cliente deve acessar o site e clicar em “Cadastrar”. |
| Pós-condição | Será exibido um o formulário de preenchimento de dados pessoais. |
| Sequências típica de eventos | O cliente acessa o sistema e clica em Cadastre-se. O cliente deverá preencher um formulário com os seguintes campos: cpf, nome, e-mail e senha. |
| Sequência alternativa de eventos | O cliente pode não inserir os dados corretamente. |

|  |  |
| --- | --- |
| Caso de Uso | Editar perfil do cliente |
| Visão geral | O cliente edita seu perfil. |
| Objetivo | Editar o perfil do cliente. |
| Ator primário | Cliente. |
| Pré-condição | O cliente deve ser cadastrado e se logar no sistema. |
| Pós-condição | O cliente deverá acessar a página de edição de perfil. |
| Sequências típica de eventos | O cliente precisa acessar o sistema e inserir seu e-mail e senha. O cliente deverá clicar no menu de navegação e clicar em editar perfil. O cliente deverá selecionar o campo que ele deseja editar e clicar em salvar. |
| Sequência alternativa de eventos | O cliente inseriu o login ou senha errada e não conseguiu acessar o sistema. |

|  |  |
| --- | --- |
| Caso de Uso | Criar leilão reverso |
| Visão geral | O cliente cria um leilão reverso. |
| Objetivo | Criar um leilão reverso e receber lances por ele. |
| Ator primário | Cliente e autônomo. |
| Pré-condição | O cliente ou autônomo deve ser cadastrado e se logar no sistema. |
| Pós-condição | O cliente deverá acessar a página do leilão reverso. |
| Sequências típica de eventos | O cliente precisa acessar o sistema e inserir seu e-mail e senha. O cliente deverá clicar no menu de navegação e clicar em criar leilão. O cliente deverá fornecer informações como: título do serviço, descrição e data de término, estado e cidade. |
| Sequência alternativa de eventos | O leilão pode se encerrar antes da data de término. |

|  |  |
| --- | --- |
| Caso de Uso | Encerrar leilão |
| Visão geral | O cliente encerra um leilão. |
| Objetivo | Encerrar leilões em aberto. |
| Ator primário | Cliente e Autônomo. |
| Pré-condição | O cliente deve ser cadastrado e se logar no sistema. |
| Pós-condição | O cliente deverá acessar a página de leilão de serviços. |
| Sequências típica de eventos | O cliente precisa acessar o sistema e inserir seu e-mail e senha. O cliente deverá clicar no menu de navegação e clicar em leilão reverso. Uma vez que o leilão for iniciado, o cliente poderá clicar num botão encerrar leilão para finalizá-lo. |
| Sequência alternativa de eventos | O cliente pode ou não encerrar o leilão antes do prazo. |

|  |  |
| --- | --- |
| Caso de Uso | Cadastrar perfil do autônomo |
| Visão geral | O autônomo cadastra o seu perfil no sistema. |
| Objetivo | Cadastrar o perfil do autônomo. |
| Ator primário | Autônomo. |
| Pré-condição | O cliente deve acessar o site e clicar em “Cadastrar”. |
| Pós-condição | Será exibido um o formulário de preenchimento de dados pessoais. |
| Sequências típica de eventos | O autônomo acessa o sistema e clica em “Cadastrar”. O autônomo deverá preencher um formulário com os seguintes campos: nome, sobrenome, e-mail, cpf, telefone, celular, data de nascimento, sexo, cep, estado, cidade, endereço, número, bairro, complemento e senha. |
| Sequência alternativa de eventos | O cliente pode não inserir os dados corretamente. |

|  |  |
| --- | --- |
| Caso de Uso | Editar perfil do autônomo |
| Visão geral | O autônomo edita seu perfil. |
| Objetivo | Editar o perfil do autônomo. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve ser cadastrado e se logar no sistema. |
| Pós-condição | O autônomo deverá acessar a página de edição de perfil. |
| Sequências típica de eventos | O autônomo precisa acessar o sistema e inserir seu e-mail e senha. O autônomo deverá clicar no menu de navegação e clicar em editar perfil. O autônomo deverá selecionar o campo que ele deseja editar e clicar em salvar. |
| Sequência alternativa de eventos | O autônomo inseriu o login ou senha errada e não conseguiu acessar o sistema. |

|  |  |
| --- | --- |
| Caso de Uso | Cadastrar perfil profissional |
| Visão geral | O autônomo cadastra seu serviço. |
| Objetivo | Cadastrar o serviço do autônomo. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve acessar o sistema e realizar o login. |
| Pós-condição | Será exibida uma tela com as informações pertinentes ao cadastro do serviço. |
| Sequências típica de eventos | O autônomo deverá preencher os seguintes campos: área de atuação, profissão, descrição do serviço e especialidade. Opcionalmente, o autônomo poderá inserir fotos e link de vídeos de serviços. |
| Sequência alternativa de eventos | O autônomo pode não encontrar sua área de atuação ou especialidades. O autônomo poderá se cadastrar se sua área de atuação não for listada. |

|  |  |
| --- | --- |
| Caso de Uso | Editar serviço |
| Visão geral | O autônomo edita seu serviço. |
| Objetivo | Editar o serviço do autônomo. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve acessar o sistema para editar seu serviço. |
| Pós-condição | Será exibida a página do serviço do autônomo. |
| Sequências típica de eventos | O autônomo deverá se logar no sistema usando seu e-mail e sua senha. Após acesso, o autônomo deverá clicar no menu de navegação e clicar em editar serviços. |
| Sequência alternativa de eventos | A conta do autônomo pode ter sido inativa pelo Administrador. |

|  |  |
| --- | --- |
| Caso de Uso | Agendar horário de serviço |
| Visão geral | O autônomo agenda seus serviços. |
| Objetivo | Agendar horário de serviços. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve acessar o sistema para agendar horário de serviços. |
| Pós-condição | Será exibida uma agenda contendo os serviços do autônomo. |
| Sequências típica de eventos | Após o acesso ao sistema, o autônomo deverá preencher a agenda conforme os dados: título do evento, cliente, local, descrição, hora de início, hora do término e valor do serviço. |
| Sequência alternativa de eventos | O sistema pode estar fora do ar e impossibilitar o autônomo acessar sua agenda. |

|  |  |
| --- | --- |
| Caso de Uso | Editar agenda |
| Visão geral | O autônomo edita sua agenda. |
| Objetivo | Editar serviços da agenda. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve acessar o sistema e acessar sua agenda. |
| Pós-condição | O autônomo poderá atualizar ou excluir seus serviços na agenda. |
| Sequências típica de eventos | O autônomo deverá acessar sua agenda e selecionar o serviço a ser editado, podendo alterar os mesmos campos preenchidos. |
| Sequência alternativa de eventos | O sistema pode estar fora do ar e impossibilitar o autônomo acessar sua agenda. |

|  |  |
| --- | --- |
| Caso de Uso | Visualizar relatório de visualização |
| Visão geral | O autônomo faz a visualização do relatório de visualizações do seu perfil. |
| Objetivo | Visualizar o relatório de visualizações do perfil. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve estar logado no sistema. |
| Pós-condição | Será exibida o relatório de visualização. |
| Sequências típica de eventos | O autônomo deverá se logar no sistema usando seu e-mail e sua senha. Após o acesso, o autônomo deverá clicar no menu de navegação e clicar em relatório. O cliente deverá selecionar o relatório de acesso de perfil e selecionar um período mensal. |
| Sequência alternativa de eventos | O sistema pode estar fora do ar e impossibilitar o autônomo acessar sua agenda. |

|  |  |
| --- | --- |
| Caso de Uso | Visualizar Comanda Mensal |
| Visão geral | Visualização do relatório de comanda mensal. |
| Objetivo | Visualizar o relatório de comanda mensal. |
| Ator primário | Autônomo. |
| Pré-condição | O autônomo deve estar logado no sistema. |
| Pós-condição | O autônomo deverá clicar no menu de navegação e clicar em relatório. |
| Sequências típica de eventos | O autônomo deverá se logar no sistema usando seu e-mail e sua senha. Após o acesso, o autônomo deverá clicar no menu de navegação e selecionar a opção relatório. Na tela de relatório, o autônomo deverá selecionar relatório de comanda mensal. |
| Sequência alternativa de eventos | Se o autônomo não inserir o valor do serviço na agenda, não haverá relatório de comanda mensal. |

|  |  |
| --- | --- |
| Caso de Uso | Aderir à conta premium |
| Visão geral | O autônomo assina um plano premium. |
| Objetivo | Tornar a conta premium. |
| Ator primário | Autônomo. |
| Pré-condição | O Autônomo precisa acessar o sistema e fazer o login. |
| Pós-condição | O autônomo deverá clicar no menu de navegação e clicar em conta premium. |
| Sequências típica de eventos | O Autônomo acessa o sistema e realiza o login. O autônomo deverá clicar no menu de navegação e clicar em conta premium. O autônomo deverá clicar em adquirir conta premium. |
| Sequência alternativa de eventos | O sistema pode não estar funcionando corretamente e, assim, o autônomo não conseguirá aderir à conta premium. |

|  |  |
| --- | --- |
| Caso de Uso | Dar lance |
| Visão geral | O autônomo dá um lance em um leilão aberto. |
| Objetivo | Dar lance em leilões. |
| Ator primário | Autônomo. |
| Pré-condição | O Autônomo precisa acessar o sistema e fazer o login. |
| Pós-condição | O autônomo deverá clicar no menu de navegação e clicar em leilão em aberto. |
| Sequências típica de eventos | O Autônomo acessa o sistema e realiza o login. O autônomo deverá clicar no menu de navegação e clicar em leilões em aberto. O autônomo deverá escolher um dos leilões, clicar em dar lance e inserir o valor. |
| Sequência alternativa de eventos | O sistema pode não estar funcionando corretamente e, assim, o autônomo não conseguirá aderir à conta premium. |

|  |  |
| --- | --- |
| Caso de Uso | Pesquisar leilões em aberto |
| Visão geral | O autônomo pesquisa por um leilão aberto. |
| Objetivo | Pesquisar leilões em aberto. |
| Ator primário | Autônomo. |
| Pré-condição | O Autônomo precisa acessar o sistema e fazer o login. |
| Pós-condição | O autônomo deverá clicar no menu de navegação e clicar em leilão reverso. |
| Sequências típica de eventos | O Autônomo acessa o sistema e realiza o login. O autônomo deverá clicar no menu de navegação e clicar em leilão reverso. O autônomo deverá clicar em pesquisar leilão. Ele deverá pesquisar por área de atuação ou especialidade. |
| Sequência alternativa de eventos | A busca por um leilão pode não retornar resultados. |

|  |  |
| --- | --- |
| Caso de Uso | Filtrar lances |
| Visão geral | O autônomo filtra um leilão. |
| Objetivo | Filtrar leilões. |
| Ator primário | Cliente e Autônomo. |
| Pré-condição | O usuário precisa acessar o sistema e fazer o login. |
| Pós-condição | O usuário deverá clicar no menu de navegação e clicar em leilão reverso. |
| Sequências típica de eventos | O usuário acessa o sistema e realiza o login. O autônomo deverá clicar no menu de navegação e clicar em leilão reverso. O autônomo deverá clicar em histórico de leilão. Ele poderá filtrar por leilões em aberto ou pesquisar por área de atuação ou especialidade. |
| Sequência alternativa de eventos | Se o usuário não estiver participando de leilão, nada poderá ser filtrado. |

|  |  |
| --- | --- |
| Caso de Uso | Dar feedback do leilão |
| Visão geral | O cliente ou autônomo dão feedback de um leilão. |
| Objetivo | Dar feedback do leilão. |
| Ator primário | Cliente e Autônomo. |
| Pré-condição | O usuário precisa acessar o sistema e fazer o login. |
| Pós-condição | O usuário deverá clicar no menu de navegação e clicar em leilão reverso. |
| Sequências típica de eventos | O usuário acessa o sistema e realiza o login. O autônomo deverá clicar no menu de navegação e clicar em leilão reverso. O autônomo deverá no leilão no botão “Encerrar leilão” caso este esteja em aberto ou expirado, porém sem feedback. O feedback deve conter a avaliação e a recomendação do autônomo, bem como um |
| Sequência alternativa de eventos | O usuário pode não realizar o feedback do leilão. |

1. https://www.w3.org/WAI/gettingstarted/Overview.html [↑](#footnote-ref-1)
2. Guia de boas práticas de acessibilidade: https://www.w3.org/TR/UNDERSTANDING-WCAG20/Overview.html#contents [↑](#footnote-ref-2)
3. APIs (*application programming interfaces*) conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web [↑](#footnote-ref-3)