CÉLIA HANAKO KANO
DAVID FERNANDES DE CARVALHO GOMES

# IMPLEMENTAÇÃO E INTEGRAÇÃO DO SUBSISTEMA DE TRANSPORTE EM UM SISTEMA PRODUTIVO DISPERSO

São Paulo
2010

CÉLIA HANAKO KANO
DAVID FERNANDES DE CARVALHO GOMES

# IMPLEMENTAÇÃO E INTEGRAÇÃO DO SUBSISTEMA DE TRANSPORTE EM UM SISTEMA PRODUTIVO DISPERSO

Monografia apresentada à Escola Politécnica da Universidade de São Paulo referente à disciplina PMR 2550 – Projeto de Conclusão de Curso II

Curso de Graduação:
Engenharia Mecatrônica

Orientador:
Professor Dr. Paulo Eigi Miyagi

São Paulo
2010

FICHA CATALOGRÁFICA

**Kano, Célia Hanako**
**Implementação e integração do subsistema de transporte em um sistema produtivo disperso / C.H. Kano, D.F.C. Gomes. -- São Paulo, 2010.**
**176 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.**

**1. Sistemas de produção 2. Serviços 3. Internet 4. Redes de Petri I. Gomes, David Fernandes de Carvalho II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III. t.**

# DEDICATÓRIA

Aos meus pais, Sonia e Celso, pelos esforços em proporcionar uma boa formação e incentivos na conclusão de mais uma etapa na minha vida; e aos amigos, pelos momentos de alegria que compartilhamos nesses anos.

Célia Hanako Kano

Aos meus pais Maria Luisa Gonçalves Fernandes Gomes e Luís Filipe de Carvalho Gomes e minha irmã Joana Fernandes de Carvalho Gomes pelo apoio e incentivo ao longo de meus estudos

David Fernandes de Carvalho Gomes

# AGRADECIMENTOS

Agradecimento especial ao professor Paulo Eigi Miyagi, pela orientação e pelo grande estímulo e ensinamento transmitidos em sala de aula e durante todo o Trabalho de Formatura.

Aos colegas José Isidro Garcia Melo, Marcosiris Amorim de Oliveira Pessoa, Reinaldo Squillante Júnior, Samira Souit e ao professor Fabrício Junqueira pelo apoio que viabilizaram o desenvolvimento do presente trabalho. Agradecemos também aos demais colegas do Laboratório de Sistemas de Automação (LSA) que indiretamente nos ajudaram com o trabalho.

Por fim, um agradecimento à Escola Politecnica da USP, em especial ao Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, que institucionalmente viabilizaram este trabalho, e à CNPq, FAPESP e CAPES por financiarem este projeto.

# RESUMO

O presente projeto consiste no desenvolvimento de uma interface homem-computador para o subsistema de transporte, bem como sua integração aos demais subsistemas que fazem parte de um sistema de manufatura automatizado instalado na Escola Politécnica da Universidade de São Paulo (EPUSP). Este projeto faz parte de um projeto de pesquisa do desenvolvimento de uma arquitetura para a teleoperação e monitoração remota via internet de um sistema de manufatura disperso. O subsistema de transporte realiza a movimentação de *pallets* para o transporte de peças do sistema de manufatura para a montagem mecânica de um produto. O sistema de manufatura é assim composto por vários subsistemas produtivos fisicamente distribuídos, mas com os respectivos controladores ligados à internet.

**Palavras chave:** sistema produtivo, ambiente distribuído, *web service*, teleoperação.

# ABSTRACT

This project develops a human-computer interface for the transport subsystem and its integration with other subsystems that are part of an automated manufacturing system installed at the Escola Politécnica da Universidade de São Paulo (EPUSP). This project is part of a research project to develop an architecture for teleoperation and remote monitoring via the Internet of a dispersed manufacturing system. The transport subsystem performs the movement of pallets that transport parts from a manufacturing system to a mechanical assembly system. So then, the manufacturing system is composed of several distributed productive subsystems with their drivers connected to the Internet.

**Keywords:** production system, distributive environment, web service, teleoperation

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| **ASI** | *Actuator Sensor Interface* |
| **BPEL** | *Business Process Execution Language* |
| **B2B** | *Business-To-Business* |
| **CLP** | Controlador Lógico Programável |
| **COM** | *Component Object Model* |
| **DCOM** | *Distributed Component Object Model* |
| **DB** | *Data Base* |
| **EPUSP** | Escola Politécnica da Universidade de São Paulo |
| **FB** | *Function Block* |
| **I/O** | *Input/Output* |
| **LED** | *Light Emitting Diode* |
| **LSA** | Laboratório de Sistemas de Automação |
| **OPC** | *Object linking and embedding for Process Control* |
| **PC** | *Personal Computer* |
| **PFS** | *Production Flow Schema* |
| **SED** | Sistema a Evento Discreto |
| **SOA** | *Service Oriented Architecture* |
| **SOCRADES** | *Service Oriented Cross-layer Infrastructure for Distributed Smart Embedded Devices* |
| **SP** | Sistema Produtivo |
| **UML** | *Unified Modeling Language* |
| **VAT** | *Variable Table* |
| **XML** | *Extensible Markup Language* |
| **WS** | *Web Service* |
| **WSDL** | *Web Service Description Language* |

# SUMÁRIO

# 1. INTRODUÇÃO

Na chamada ‘era da Informação’, o homem tem testemunhado avanços tecnológicos significativos na computação e na mecatrônica aplicado a áreas como telecomunicações, robotização e automação. Isto tem proporcionado mudanças na percepção de distâncias e tempos. O conceito de ‘globalização’ pode ser entendido como uma relação de interdependência entre os países e dentre seus efeitos, cita-se a acirrada competição entre as empresas, que requer uma adaptação destes aos diferentes mercados e, conseqüentemente, a evolução dos processos produtivos.

Os sistemas produtivos (SPs) geograficamente dispersos (MIYAGI et al., 2009) surgem como uma estrutura organizacional alternativa, com plantas em diferentes lugares ou até países, com objetivos que envolvem (a) reduzir o custo através do acesso a mão-de-obra, cultura técnica, recursos e matérias primas locais, (b) assegurar a qualidade explorando as condições e infraestrutura de cada região e, (c) garantir o atendimento das necessidades específicas de clientes de cada região. Ressalta-se que SPs são entendidos aqui como sistemas com processos que produzem bens ou realizam serviços (adaptado de VILLANI et al., 2007).

## 1.1. Motivação

O presente projeto está sendo conduzido no Laboratório de Sistemas de Automação (LSA) localizado na EPUSP. A linha de pesquisa do laboratório é o desenvolvimento de técnicas e tecnologias para o estudo, projeto e implementação de sistemas construídos pelo homem (*man made systems*) sob a abordagem de sistemas a eventos discretos.

Adicionalmente, ressalta-se que este projeto está inserido no contexto dos seguintes projetos: “BASys: Sistemas de Automação Balanceada”, apoiado pelo CNPq, e “Telecomando e Monitoramento Remoto de Sistemas de Manufatura” e “Desenvolvimento de um sistema colaborativo de tele-operação produtivo”, ambos participantes do programa TIDIA-Kyatera e apoiados pela FAPESP.

O primeiro projeto, apoiado pelo CNPQ, teve início em 2004, e envolve o estudo de metodologias de projeto de sistemas de automação balanceada, focado nos problemas de construção de modelos e análise destes sistemas. As aplicações são generalizadas para o caso de sistemas produtivos que envolvem não apenas sistemas de manufatura, mas

também sistemas/processos onde existe um tipo de agregação de valor aos itens (materiais ou informações).

Os demais projetos são apoiados pelo programa TIDIA-Kyatera. Lançado em 2001 pela FAPESP, o projeto Kyatera do programa TIDIA (Tecnologia da Informação no Desenvolvimento da Internet Avançada) tem como objetivo formar um ambiente de trabalho colaborativo à distância para a geração de conhecimento e inovações tecnológicas entre empresas, institutos de pesquisa, universidades e agências de fomento. Para isso, desde abril de 2004, foram feitas parcerias com empresas do setor de telecomunicações e a criação de uma rede de fibras ópticas de alta velocidade *(fiber-to-the-lab)* que chega diretamente ao interior dos laboratórios (KYATERA, 2010). Dentre os beneficiados com a rede, temos: as universidades Unicamp, USP São Carlos, USP São Paulo (Figura 1), PUC Campinas, UFSCar, IPEN, Inatel, Incor e a empresa Telefônica.


*Figura 1 - Mapa da rede de fibra óptica do projeto Kyatera no campus USP São Paulo*

Na Europa foi desenvolvido o projeto SOCRADES (*Service Oriented Cross-layer infRAstructure for Distributed smart Embedded deviceS*) um trabalho do *Information Society Technologies*, que integra o *European Union's 6th Framework Programme*. O SOCRADES tem como objetivo desenvolver uma nova geração de sistemas automatizados industriais através da SOA (*Service Oriented Architecture*). Baseado no paradigma da automação colaborativa, o SOCRADES considera o uso de ferramentas e métodos flexíveis que permitem a reconfiguração e operação em rede colaborativa, através da descentralização e sistemas embarcados distribuídos. Para isso, propõe se que os sistemas devem ser projetados segundo o conceito de orientação a serviços, ou seja, uma arquitetura que apresente uma alta adaptabilidade e rápida configuração de seus processos (adaptado de SIXTH FRAMEWORK PROGRAMME, 2010).

## 1.2. Justificativa

No presente contexto de globalização e busca pela redução de custos através da distribuição geográfica de diferentes unidades produtivas torna-se latente o problema de integrar estas plantas produtivas de forma eficiente e assegurando a devida flexibilidade.

Embora a distribuição da planta em diferentes partes possa apresentar vantagem competitiva, a coordenação e controle de movimentação dos insumos e produtos não são triviais. Desta forma o desenvolvimento de meios práticos e baratos para a integração dessas plantas é de suma importância para o desenvolvimento competitivo deste meio de produção.

Além das dificuldades relacionadas à movimentação destes materiais é necessário integrar os sistemas de forma a encadear as atividades produtivas de forma coerente e minimizando o tempo de espera e de retrabalho. Neste âmbito a exploração da aplicação de web services se demonstra uma alternativa viável em um momento em que as redes de internet estão crescendo em sua capacidade e sua confiabilidade.

## 1.3. Objetivo

O presente projeto pretende contribuir na implementação de um *framework* de sistemas colaborativos teleoperados para sistemas produtivos.

Este projeto visa especificamente o desenvolvimento da interface para as funções de monitoração remota e teleoperação do subsistema de transporte. A implementação consiste no desenvolvimento de um programa computacional que possa ser utilizado por um teleoperador em qualquer PC ligado à internet. Este programa computacional deve assegurar a reusabilidade em outras aplicações, dado que será desenvolvido na forma de um módulo funcional.

Além disso, o projeto consiste na integração do subsistema de transporte com os outros três subsistemas (de alimentação, inspeção e montagem) de um sistema produtivo disperso. Para a integração, considera-se a SOA (Arquitetura Orientada a Serviços), que garante uma alta adaptabilidade e rápida configuração de seus processos através da criação de módulos funcionais.

### 1.4. Organização do Texto

No capítulo 2 apresentam-se os fundamentos abordados ao longo do trabalho, como os sistemas produtivos dispersos, *web services* e a descrição de protocolos de comunicação (ASI e Profibus). Em seguida, o capítulo 3 especifica o sistema flexível de montagem automatizado instalado no LSA-EPUSP, através da descrição dos componentes e funcionamento de cada subsistema. A seguir, o capítulo 4 apresenta o levantamento de informações e etapas adotadas para a implementação do subsistema de transporte e a integração dos quatro subsistemas. Este capítulo inclui descrições da parte de hardware (listagem de sensores, atuadores, por exemplo), da parte de software (mapeamento com o OPC Server, criação de *web services*) e da integração (funcionamento do Coordenador, diagramas de seqüência, interfaces do cliente). Por fim, o capítulo 5 encerra com a discussão dos resultados obtidos e das simplificações consideradas.

# 2. FUNDAMENTOS

Nesse capítulo, inicialmente apresentamos conceitos a serem discutidos ao longo do trabalho. Em seguida, uma visão geral do funcionamento do sistema flexível de montagem e uma descrição de cada subsistema e da arquitetura adotada.

Os fundamentos abordados foram divididos em 4 assuntos, sendo eles sistemas, modelagem de sistemas, comunicação e ferramentas. No primeiro tópico serão discutidos assuntos relacionados aos sistemas produtivos dispersos, a emulação de sistemas, sistemas colaborativos e *web services*. No segundo tópico serão abordados temas relevantes para a emulação de sistemas, de forma a focar nos aspectos relevantes para este trabalho, onde se destacam os sistemas a eventos discretos, redes PFS e a *Unified Modeling Language*. No terceiro tópico discorre-se sobre os principais aspectos de comunicação, onde é possível destacar a rede ASI, a Profibus e comunicação síncrona e assíncrona. Por ultimo será discutido o Simatic Manager e o OPC Server.

## 2.1. Sistemas

Um sistema pode ser definido como um conjunto de partes e componentes, isolado de maneira conveniente do ambiente que o circunda, com um único objetivo, com um ou mais aspectos variantes no tempo, a ser representado na forma de Equações de Estado (BARROS, 2008).

Dentre os principais assuntos relacionados a sistema o presente trabalho aborda: os sistemas produtivos dispersos, a emulação de sistemas, sistemas colaborativos e a arquitetura orientada a serviços e os *web services*.

### 2.1.1. Sistema produtivo disperso

Um SP disperso consiste em plantas/unidades produtivas cujas instalações estão geograficamente dispersas e que necessitam se comunicar e trabalhar de forma colaborativa. Esta comunicação normalmente lida com uma grande quantidade de informações entre centros produtivos de uma mesma entidade (empresa ou algum tipo de consórcio de empresas), mas em diferentes localizações geográficas (MIYAGI et al., 2009).

Nesse trabalho considera-se que as entidades não são isoladas, mas sim algum tipo de consórcio de diferentes unidades. Desse modo, um sistema industrial automatizado pode

ser constituído de diferentes unidades produtivas responsáveis pela recepção e distribuição da matéria-prima, testes de controle de qualidade, transporte e montagem dos subprodutos intermediários e do produto final, e por outras funções. Assim, as unidades produtivas podem ter localizações geográficas diferentes, como em cidades ou países afastados, contudo, uma integração entre essas diferentes unidades deve ser garantida pelo SP disperso.

O sistema industrial automatizado é considerado um SED, pois seus eventos não são contínuos no tempo, mas sim instantâneos e assíncronos. Ou seja, as diferentes unidades produtivas possuem interdependência. Por exemplo, o teste de controle de qualidade é feito somente após a distribuição da matéria prima; e por sua vez, o transporte somente é requisitado após o teste de controle de qualidade.

### 2.1.2. Emulação de sistemas

No presente projeto considera-se que o sistema de montagem automatizado instalado no Laboratório de Sistemas de Automação (LSA) na EPUSP emula um SP disperso. Define-se “emulação” como o casamento de duas disciplinas: simulação e projeto do sistema de controle (SCHEISS, 2001).

Neste projeto, a primeira disciplina, simulação, ocorre quando o modelo de SED do SP disperso é recriado para ser verificado e analisado utilizando uma técnica de simulação (SCHEISS, 2001). Por exemplo, um programa de simulação discreta que utiliza de métodos numéricos e empregam procedimentos computacionais para executar o modelo. As operações matemáticas são realizadas em momentos determinados utilizando as informações disponíveis de forma que é possível obter uma execução através de passos e visualizar os estados no modelo do sistema. Dessa forma, enquanto um SP disperso real pode apresentar uma dinâmica relativamente complexa, a execução de um programa de simulação discreta pode executar passos padronizados de diferentes cenários de forma a obter dados significativos para a análise da dinâmica do SP disperso.

A segunda disciplina se refere ao projeto do sistema de controle. Geralmente, em SPs o controle é implementado por meio de CLPs (controladores lógicos programáveis) ou PCs (*personal computers*) com um software de controle, ambos com *inputs* (entradas) dos sinais dos dispositivos de detecção e dispositivos de comando e, *outputs* (saídas) para os dispositivos de atuação e de monitoração. No CLP e PCs, os softwares de controle processam estes *inputs* e *outputs* convertendo comandos lógicos em sinais para atuadores executarem ações reais do sistema. Como exemplos de *inputs*, temos sinais de *switches* e

botões; dentre os *outputs*, sinais para os cilindros pneumáticos, LEDs (*light emitting diode*), etc (SCHIESS, 2001).

No presente projeto considera-se a emulação de um SP disperso. Um SP disperso conforme descrito anteriormente é constituído de unidades produtivas geograficamente dispersas, ou seja, cada uma em uma localização geográfica diferente, como em cidades ou países afastados. Nesse cenário, uma rede de transporte é responsável pela movimentação dos subprodutos intermediários pelas unidades produtivas. As unidades produtivas executam operações produtivas, como testes de controle de qualidade e montagem dos subprodutos. Um esboço de um SP disperso é mostrado na Figura 2. Na figura, o subsistema de transporte é representado pelos caminhos traçados entre as unidades produtivas.

Para a emulação do subsistema de transporte, de um SP disperso, o SP instalado no LSA-EPUSP é utilizado (Figura 3). Nessa instalação, o subsistema de transporte é emulado por um subsistema de movimentação de *pallets*, responsável pelo transporte das peças entre as unidades produtivas (SP1/SP2 e SP3).

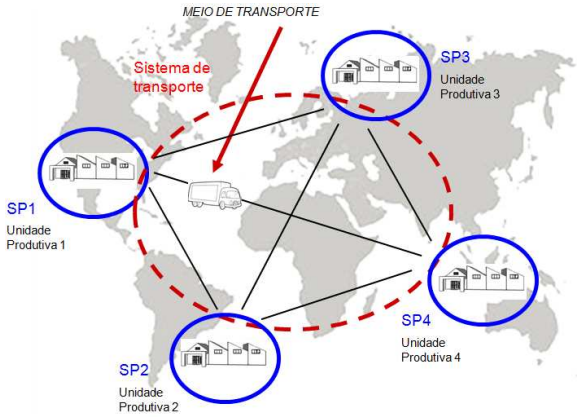



*Figura 2 - Sistema produtivo disperso*

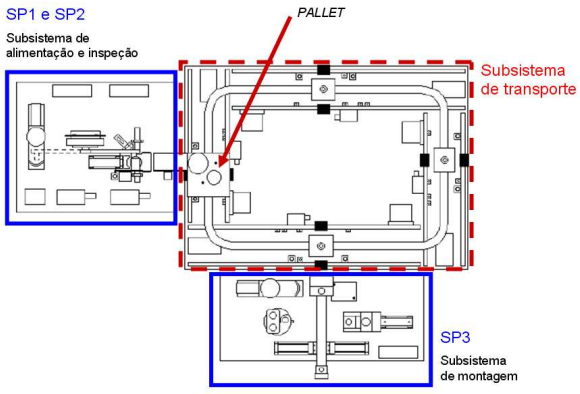



*Figura 3 - Sistema produtivo emulado*

Dentre as vantagens da emulação, citam-se: (a) diminuição de riscos de acidentes reais, (b) possibilidade de criar testes extensivos, (c) diminuição do custo em cada teste através do modelo experimental (CHWIF, 2002), (d) não envolvimento de perdas de produção e (e) não envolvimento de gastos com funcionários, e outros.

### 2.1.3. Sistema colaborativo e arquitetura orientada a serviços

Neste trabalho considera-se a emulação de um SP disperso e colaborativo. Um sistema colaborativo considera um conjunto de unidades como um conglomerado de entidades distribuídas, autônomas, inteligentes e reutilizáveis que cooperam entre si para realização de uma tarefa. Essas entidades devem ser pró-ativas, isto é, devem ter iniciativa nas ações e interagirem com as outras entidades para atingir os objetivos locais e globais (SIXTH FRAMEWORK PROGRAMME, 2010).

A SOA (*Service Oriented Architecture*) é uma definição de arquitetura para sistemas distribuídos baseados no conceito de “serviço”. Nessa arquitetura, serviços requerem simples funções para atender os processos computacionais a serem executados. Na SOA, cada bloco de programação do software pode ser interpretado como um serviço programado ou de configuração (MIYAGI et al., 2009).

Dentre as vantagens da SOA estão: (a) reconfiguração dos elementos, ao invés da reprogramação deles; (b) interoperabilidade, através da adição de novos serviços/elementos ao sistema, independente da plataforma e da tecnologia (permitindo o uso de dispositivos fabricados por empresas diferentes); (c) integração vertical, pela implementação da SOA através de técnicas como o de *Web Services* que permite o tratamento comum (unificado) de todas as informações (dados e sinais), desde os sinais/dados de sensores e atuadores até os dados dos processos de negócios; (d) escalabilidade, permitindo a expansão do sistema sem que necessite modificar significativamente o sistema original; (e) diagnósticos de falhas (em sensores, fontes, controladores) e reparo em tempo real; (f) reuso dos componentes, que gera uma redução de custos e sustentabilidade; e (g) proteção intelectual da arquitetura que expõe a funcionalidade dos serviços e não a tecnologia por trás (SIXTH FRAMEWORK PROGRAMME, 2010).

### 2.1.4. Web service

O *web service* é uma das técnicas para implementação da SOA. O *web service* (WS) consiste em um conjunto de protocolos padrões que permitem a criação, distribuição e

integração de aplicações na internet, através da comunicação entre sistemas computacionais, sendo muito utilizado em aplicações B2B (*business-to-business*) (PAMPLONA, 2010). Define-se o B2B como a linguagem de negócios que faz uso das tecnologias baseadas em *Web* para conduzir os negócios entre empresas. "Negócios" podem significar compra e venda ou também troca de informações. As transações B2B podem ocorrer diretamente entre empresas ou através de uma terceira entidade (intermediária) para ajudar a compatibilizar as informações de compradores e vendedores (TALLARD GROUP, 2010).

No WS, os softwares são serviços, e por isso chamado de *e-services* (CHENG et al., 2006). Além disso, outra característica de um WS é a independência em relação à plataforma de desenvolvimento e a linguagem de programação.

Numa economia globalizada com constantes exigências por interoperabilidade e flexibilidade nas estruturas dos SPs, o WS se apresenta como uma solução adequada. Caracterizado pela flexibilidade em encapsular funções e a interoperabilidade por suportar integração de aplicações diversas, essa tecnologia tem chamado a atenção por reduzir o custo na integração de aplicativos e facilitar a implementação de uma arquitetura orientada a objetos (CHENG et al., 2006).

## 2.2. Modelagem de sistemas

De forma a permitir a análise sistemática de um sistema é necessário primeiramente obter um modelo matemático para seu funcionamento, sendo posterior realizada a análise de seu desempenho.

Na obtenção de um modelo devemos conciliar simplicidade e precisão nos resultados da análise. Com freqüência, propriedades físicas inerentes ao sistema são ignoradas. Nesse caso, se os efeitos que essas têm forem pequenos, obtêm-se boa aproximação na análise do modelo e os resultados experimentais (OGATA, 2006).

Neste presente trabalho foram abordados os temas de sistemas a eventos discretos e as ferramentas *Production Flow Schema* (PFS) e a *Unified Modeling Language* (UML).

### 2.2.1. Sistemas a eventos discretos

Os sistemas a eventos discretos (SEDs) são sistemas cujo comportamento dinâmico é determinado pela ocorrência de estímulos discretos, ou seja, os eventos que ocorrem no sistema não são contínuos no tempo, mas instantâneos e assíncronos (CURY, 2001). Estes

eventos determinam os estados dos SEDs que são mantidos até a ocorrência de novos eventos.

### 2.2.2. Production Flow Schema

Na modelagem conceitual de um SP, considera-se o PFS (*Production Flow Schema*), um grafo que representa as principais atividades realizadas no sistema, pois a identificação destas atividades facilita a compreensão do sistema. A idéia é que um sistema visto como um SED pode ser caracterizado com base no fluxo de itens e qualquer processo produtivo pode ser decomposto em três elementos básicos (MIYAGI, 1996): (a) elemento ativo (atividades); (b) elemento distribuidor (distribuições) e (c) arco (relações entre os componentes anteriores. Os elementos básicos são representados na Figura 4:

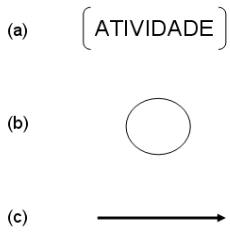



*Figura 4 - Elementos do PFS: (a) Elemento ativo; (b) Elemento distribuidor; (c) Arco (Adaptado de MIYAGI, 1996)*

### 2.2.3. Unified Modeling Language

O PFS, anteriormente descrito, é uma ferramenta de apoio para a especificação das funcionalidades e operações do subsistema de transporte. Com base neste material, para a especificação dos requisitos da interface com o operador deste subsistema optou-se pelo uso da UML. A UML (*Unified Modeling Language*) é uma linguagem semi-formal amplamente utilizada no processo de desenvolvimento de softwares para a especificação de programas computacionais baseados no paradigma da “orientação a objeto”. Nessa linguagem, diferentes tipos de diagramas permitem a representação dos diferentes aspectos do programa computacional a ser desenvolvido (BERNARDI et al., 2002).

A UML 2.0 possui diferentes diagramas, classificados em três grupos: (a) diagramas estruturais (de classes, de objetos, de componentes, de instalação, de pacotes e de

estrutura); (b) diagramas comportamentais (de casos de uso, de transição de estados e de atividade); (c) diagramas de interação (de seqüência, de interatividade, de colaboração ou comunicação e de tempo).

## 2.3. Comunicação

Os conceitos mais relevantes para a implementação deste foram a rede PROFIBUS, protocolo de rede, a rede ASI, utilizada na implementação da rede de dispositivos de detecção e atuação e a comunicação síncrona e assíncrona.

### 2.3.1. Profibus

Os conceitos de SOA e WS se aplicam para a implementação da parte lógica e de software do sistema de controle do SP disperso, que envolve a interação via internet entre as unidades produtivas (subsistemas), entretanto, para a comunicação entre os dispositivos de controle que estão dentro das unidades produtivas é necessário considerar também um protocolo de comunicação. Assim no presente caso considerou-se o PROFIBUS, que permite monitorar e controlar múltiplos dispositivos de I/O's.

O PROFIBUS é um protocolo de rede de comunicação de campo, aberto e independente de fornecedores, no qual a interface entre dispositivos permite uma ampla aplicação em processos de manufatura e automação (KANEKO, 2008). Este protocolo de comunicação começou como um projeto apoiado por autoridades públicas, em 1987 na Alemanha. Dentro do contexto deste projeto, 21 companhias e institutos uniram forças e criaram um projeto estratégico chamado de FIELDBUS. O objetivo era a realização e manutenção de um barramento de campo bitserial, tendo como requisito básico a padronização da interface do dispositivo de campo. Por esta razão, os membros das companhias do ZVEI (Associação Central da Indústria Alemã) concordaram em adotar um único conceito técnico para manufatura e automação de processos (MAIBASHI; SAITO, 2009).

Hoje, existem diferentes versões PROFIBUS baseadas nas aplicações usuais de automação: PROFIBUS DP (manufatura), PROFIBUS PA (processo), PROFIdrive (drivers) e PROFIsafe (universal) (PROFIBUS, 2010), e este protocolo pode utilizar como meio físico de transmissão qualquer um dos seguintes padrões: RS-485, ISC 61158-2 ou fibra óptica (KANEKO, 2008).

O PROFIBUS é o protocolo de rede responsável pela comunicação entre o computador local e o controlador remoto. Adicionalmente a este protocolo, o subsistema de transporte utiliza uma comunicação com porta serial RS-232 para o dispositivo de identificação dos *pallets* e uma comunicação através da ASI (*Actuator Sensor Interface)* para os demais dispositivos. Esses dois tipos de comunicação serão descritos a seguir.

### 2.3.2. ASI

O SP tem entre seus componentes diversos dispositivos de atuação e de detecção. Desta maneira, para organizar como estes dispositivos devem ser conectados ao dispositivo de realização de controle (controladores). Com exceção dos dispositivos de detecção responsáveis pela identificação do *pallet* em cada posição todos os demais dispositivos de detecção a atuação se comunicam no sistema através da ASI (*Actuator Sensor Interface*).

A ASI é uma forma de implementar uma rede de dispositivos de atuação e detecção. Normalmente, os sinais desses dispositivos nos processos industriais são transmitidos através de um grande número de fios elétricos. A ASI permite a simplificação desta fiação, substituindo então malhas de cabos elétricos, por apenas um par de fios, que são usados por todos esses dispositivos. Este par de fios é responsável pela alimentação (de energia) dos dispositivos e pela transmissão dos dados binários. A Figura 5 ilustra a posição da ASI numa estrutura de comunicação industrial (adaptado de KANEKO, 2008).

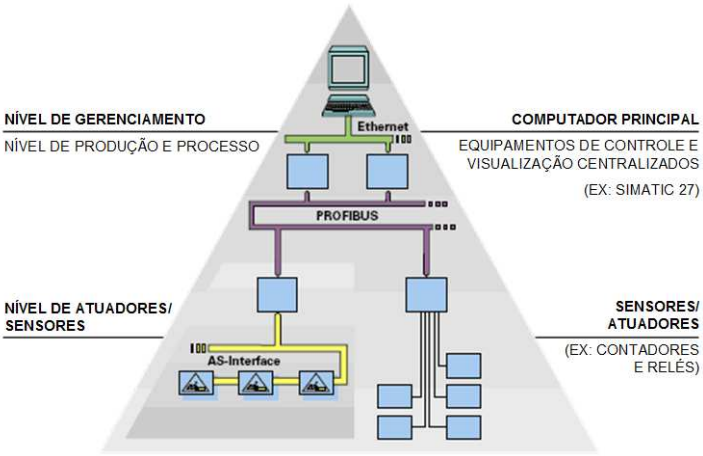



*Figura 5 - Hierarquia da comunicação industrial (Adaptado de SIEMENS, 2008).*

A ASI foi concebida para complementar as redes de comunicação e tornar mais simples e rápida a conexão dos dispositivos de atuação e detecção com os seus respectivos

controladores (dispositivos de realização do controle). O sistema baseia-se numa comunicação mestre-escravo, cujo mestre é responsável pelo direcionamento das "perguntas" e tratamento das "respostas" dos escravos. O mestre pode gerenciar até 31 escravos. A comunicação entre o mestre e os escravos é feita serialmente, através de um par de fios não trançados e nem blindados. Inicialmente, o mestre "fala" com o primeiro escravo, atualiza as saídas do mesmo (se existirem) e pergunta o estado binário das entradas deste escravo. Imediatamente o escravo responde e, após um pequeno intervalo, o mestre "fala" com o próximo escravo. Após o último escravo, o ciclo se completa e o mestre começa a conversar novamente com o primeiro escravo. O ciclo de varredura completo tem duração de até 5 ms (contendo 31 escravos na rede). Um escravo na ASI pode possuir no máximo 4 entradas digitais e no máximo 4 saídas digitais (ALTUS, 2008). A ASI permite a implementação de uma rede de comunicação com vários tipos de topologias, permitindo ainda que a qualquer momento possa se incluir uma nova derivação, possibilitando a conexão de novos dispositivos. Cada usuário pode escolher sua topologia, conforme sua necessidade e disposição física dos elementos no campo. O cabo da rede de comunicação não necessita de resistor de terminação, sua única limitação está relacionada com o comprimento do fio, que deve ser de no máximo de 100 m de comprimento. Caso necessário, o cabo pode ter um acréscimo de 200 m com a utilização de repetidores (*boosters*), ficando, assim, com um comprimento total de 300 m (ALTUS, 2008).

O cabo (Figura 6), padrão da AS-Interface, tornou-se um tipo de marca registrada. Ele possui uma seção geometricamente determinada e transmite ao mesmo tempo dados e energia auxiliar para os sensores. Para os dispositivos de atuação é em geral, necessária uma tensão auxiliar alimentada adicionalmente ($24V_{CC}$). Desta forma, o cabo para a energia auxiliar $24V_{CC}$ é um cabo perfilado preto (Figura 6) (adaptado de KANEKO, 2008).

Para aplicações com exigências maiores de isolação elétrica podem se utilizar cabos com TPE perfilado (elastômetro termoplástico) ou PUR perfilado (poliuretano). Como condutor de transmissão podem ser utilizados também cabos redondos com sistema de condução duplo. Uma blindagem do condutor não é necessária em função da técnica de transmissão de sinais empregada (SIEMENS, 2008).

*Figura 6 - Cabos ASI (SIEMENS, 2008).*

### 2.3.3. Comunicação síncrona e assíncrona

Ressalta-se que entre os dispositivos de detecção e de atuação, e o dispositivo de realização do controle, existem dois tipos de comunicação: síncrona e assíncrona.

A comunicação síncrona funciona baseada na sincronia do emissor e receptor, desta forma a comunicação depende de um sinal de *clock* externo e ao enviar uma mensagem o sistema fica inativo até obter uma resposta.

Na comunicação assíncrona não é necessário haver sincronia entre o emissor e o receptor da mensagem, desta forma o sistema fica livre para realizar outras tarefas até que receba uma resposta. Esse tipo de comunicação leva à exigência de um identificador, de forma que o receptor saiba identificar a que se refere a mensagem que acaba de chegar.

Ao se comparar estes diferentes tipos de comunicação nota-se que a comunicação assíncrona é simples, barata e de fácil implementação, porém gera um tráfego grande de informações com propósitos unicamente de controle e processos secundários. A comunicação síncrona por sua vez apresenta uma melhor relação de mensagens entregues corretamente, sendo, porém mais complexa e exigindo hardware mais caro.

## 2.4. Ferramentas

De forma a integrar as funcionalidades dos diferentes aplicativos e permitir a escrita dos programas responsáveis pela lógica de funcionamento do sistema foi necessário utilizar duas ferramentas em especial. O SIMATIC Manager é o programa da Siemens que permite a implementação da lógica enquanto o OPC Server facilita a integração entre dispositivos de diferentes lógicas e fabricantes.

### 2.4.1. SIMATIC Manager

O SIMATIC Manager é o software da Siemens responsável pela integração entre a parte física e a parte lógica dos sistemas da empresa. Essa ferramenta permite a configuração, programação, compilação e envio de programas para o CLP. O SIMATIC Manager permite o uso de três linguagens para sua programação, sendo elas: Ladder, Instruction Lists e Function Blocks.

A transferência da lógica de controle do PC para o CLP é feita através de rede PROFIBUS, enquanto comunicação de entradas e saídas do programa, referentes aos dispositivos de detecção e atuação, é realizada através de rede ASI.

Uma vez transferido, o CLP pode executar as funções criadas independentemente do PC. Para que seja possível o acesso remoto das funções programadas foi feito o mapeamento das variáveis de memória do CLP através da aplicação OPC Scout.

### 2.4.2. OPC Server

O OPC (*OLE for Process Control*) define um tipo de comunicação padrão utilizado na indústria que garante a interoperabilidade, ou seja, permite a troca de informações entre equipamentos e aplicações de controle independente dos fabricantes (OPC Foundation, 2010). Esta tecnologia foi desenvolvida pelo OPC Foundation (organização sem fins lucrativos, patrocinada pela Intellution, Microsoft e outras empresas). Em 1994, o grupo foi formado por fornecedores de diversos ramos da indústria e tomaram como desafio eliminar a necessidade do uso em conjunto do aplicativo cliente e servidor do mesmo fornecedor.

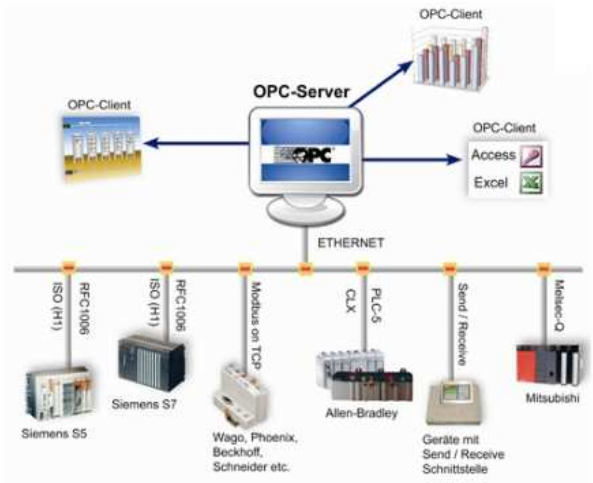



*Figura 7 - Esquema da estrutura cliente/servidor da comunicação via OPC (OPC Foundation, 2010)*

Esta é uma tecnologia baseada em padrões abertos, tecnologias Microsoft de OLE (*Object Linking and Embedding*), COM (*Component Object Model*) e DCOM (*Distributed Component Object Model*). Essa tecnologia é baseada em uma estrutura cliente/servidor (Figura 7), no qual uma aplicação atua como servidor provendo informações e a outra atua como cliente recebendo as informações (MARCHENTA, 2009). Quanto à transferência de dados, o servidor OPC atua como um “Portal de Software”, lendo informação dos dispositivos de campo e transformando-as em dados OPC, com o propósito de integração dos diversos sistemas e dispositivos (LING,CHEN,YU, 2004).

# 3. SISTEMA PRODUTIVO EMULADO

O projeto considera o subsistema de transporte de um sistema industrial automatizado e geograficamente disperso, ou seja, um sistema no qual as unidades produtivas podem estar em locais fisicamente afastados, como em cidades ou países diferentes. Por esse motivo, as funções de monitoração e operação devem permitir que o operador execute essas funções de forma remota, ou seja, sem a necessidade de estar no mesmo local físico da unidade produtiva.

O sistema flexível de montagem automatizada instalado no LSA-EPUSP emula um SP disperso. Neste sistema, o produto final é composto por quatro tipos de peças: “corpo” (na cor preta, prata ou rosa), “pino” (na cor preta ou cinza), “mola” e “tampa” (na cor azul). As peças são ilustradas na Figura 8.

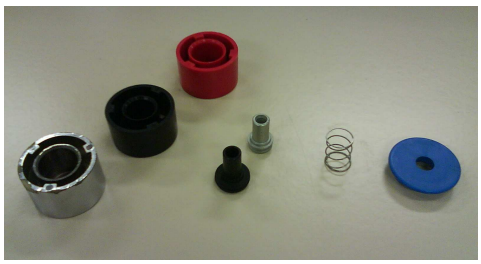


*Figura 8 - Foto do corpo nas três cores (preta, prata e rosa), do pino nas duas cores (preta e cinza), mola e tampa (azul).*

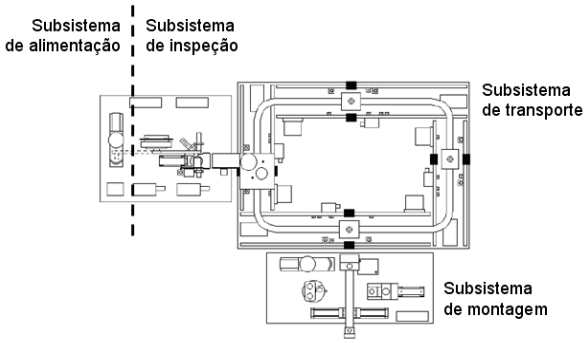



*Figura 9 - Subsistemas do sistema (adaptado de KANO et al., 2009)*

Este SP na versão atualmente instalada é um conjunto de quatro subsistemas, cada qual capaz de ser operada individualmente (KANEKO, 2008). Os subsistemas são:

subsistema de alimentação, subsistema de inspeção, subsistema de transporte e subsistema de montagem, indicados na Figura 9.

Neste SP, o subsistema de alimentação é responsável pelo armazenamento dos “corpos”, e quando solicitado pelo operador, este subsistema libera estes para o subsistema de inspeção. A seguir, o subsistema de inspeção é responsável pela identificação da cor (prateada, rosa ou preta) e teste da altura dos “corpos” (KANEKO, 2008). Caso o “corpo” não possua a cor ou altura desejada pelo operador, este subsistema é responsável pelo descarte da peça. Em seguida, os corpos são enviados via subsistema de transporte para o subsistema de montagem, no qual cada “corpo” recebe um “pino”, “mola” e “tampa”, sendo a cor do “pino” dependente da cor do “corpo”, ou seja: “corpo” prata com “pino” preto, “mola” e “tampa”; “corpo” rosa com “pino” preto, “mola” e “tampa”; e “corpo” preto com “pino” prata, “mola” e “tampa”, como mostra a Figura 10.

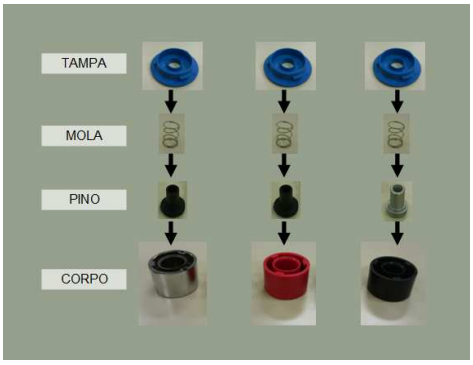



*Figura 10 - Os três tipos de produtos (adaptado de KANEKO, 2008)*

A seguir, a descrição dos quatro subsistemas (alimentação, inspeção, montagem e transporte) e da arquitetura do SP disperso a ser implementada.

### 3.1. Alimentação

O subsistema de alimentação é responsável pelo início da montagem, que ocorre a partir da seleção do tipo correto de “corpo” para o produto desejado. Ele é composto por um *buffer* com capacidade para 10 “corpos”, podendo as mesmas ser de cor prateada (metal), preta (plástico), ou rosa (plástico). Através de um pistão e um braço basculante o “corpo “é enviado para o subsistema de inspeção, que será responsável pela identificação do corpo.

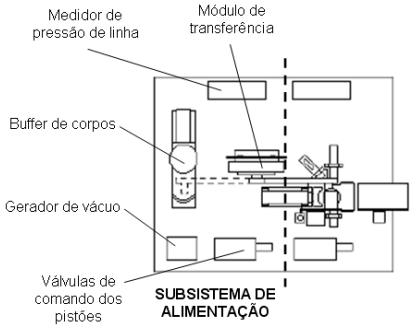



*Figura 11 - Subsistema de alimentação (adaptado de MARCHENTA, 2009)*

### 3.2. Inspeção

Após a liberação do “corpo” pelo subsistema de alimentação, o “corpo” é levado para o subsistema de inspeção por um braço articulado do módulo de transferência para identificação das características do material e dimensões do “corpo” (identificação da cor – prata, rosa ou preta; e teste da altura dos “corpos”). Para a inspeção, o subsistema utiliza três tipos de sensores: indutivo (que identifica “corpos” metálicos), óptico (identifica os corpos rosa ou prata) e capacitivo (que identifica a presença do corpo).

Assim, estas características são comparadas com as características do pedido feito pelo usuário, resultando no descarte ou disponibilização do “corpo” para o subsistema de transporte.

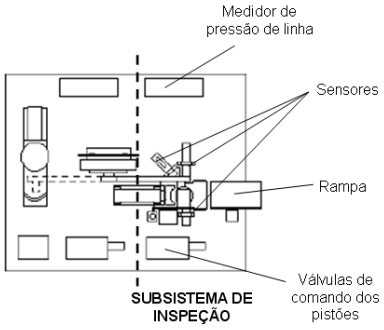



*Figura 12 - Subsistema de inspeção (adaptado de MARCHENTA, 2009)*

### 3.3. Montagem

Em seguida, os “corpos” são enviados para o subsistema de montagem através de *pallets* do subsistema de transporte. Quando o *pallet* chega ao local para processamento do

subsistema de montagem, o “corpo” é suspenso e colocado no apoio para *pallet*. Em seguida, a unidade de execução de montagem se move e o “corpo” é pego pela garra dedicada e transferido para o módulo de travamento. A garra retira um “pino” do *buffer*, sendo ele de alumínio para peças pretas ou de plástico preto para peças prateadas ou rosas (MARCHENTA, 2009), e o posiciona no interior do “corpo”. Depois, uma “mola” é retirada do seu *buffer* e pressionada sob o “pino”. Por fim, uma “tampa” é retirada do seu *buffer*, rotacionada pelo dispositivo de montagem rotativo e fixada no “corpo”. Em seguida, a peça montada é liberada pelo módulo de travamento e a garra leva a peça novamente até o *pallet*.

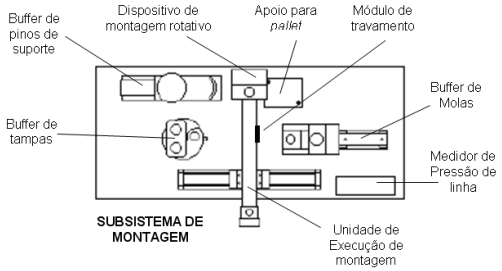



*Figura 13 - Subsistema de montagem (adaptado de MARCHENTA, 2009)*

### 3.4. Transporte

O subsistema de transporte possui quatro locais específicos para a parada de *pallets*: estação 1, estação 2, estação 3 e estação 4. A movimentação dos *pallets* é feita por esteiras. Em cada estação existem dispositivos de detecção e de atuação para identificar os *pallets* e para o devido posicionamento destes.

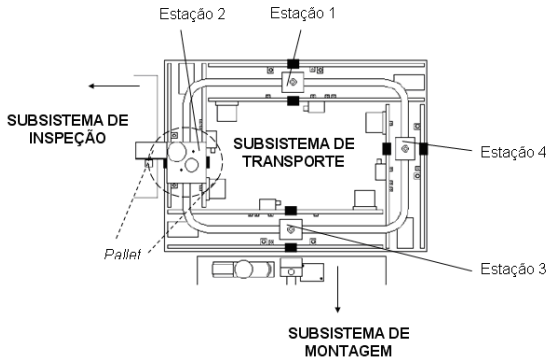



*Figura 14 - Subsistema de transporte (adaptado de MARCHENTA, 2009)*

### 3.5. Arquitetura do sistema

A arquitetura considerada neste caso é a SOA que apresenta uma alta adaptabilidade e rápida configuração de seus processos (SIXTH FRAMEWORK PROGRAMME, 2010). Neste sentido os subsistemas podem ser vistos como módulos funcionais. A arquitetura do SP (Figura 15) visa assegurar a automatização e coordenação das diferentes atividades dos subsistemas de montagem (em destaque), subsistema de transporte, subsistema de inspeção e subsistema de alimentação, que são interligados através da internet, fornecendo serviços específicos (SOUIT, 2009).

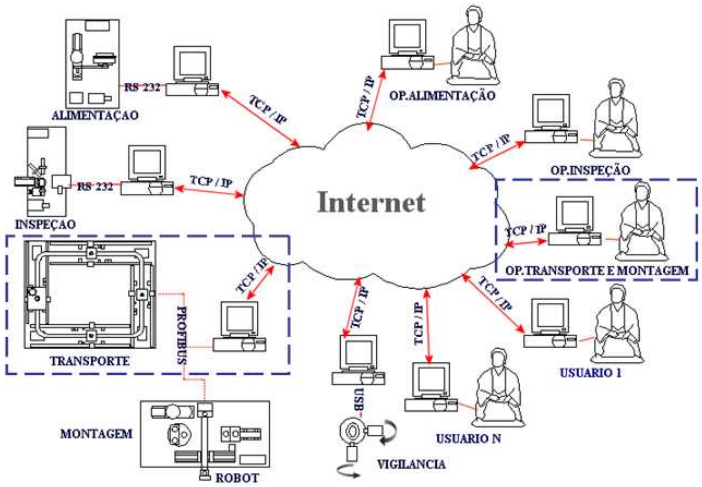



*Figura 15 - Arquitetura do sistema (Adaptado de MELO, 2008)*

# 4. PROJETO

A fim de especificar o subsistema de transporte, o item 4.1 descreve o funcionamento do subsistema e a parte de hardware, apresentando os dispositivos de detecção e atuação que o compõem e o controlador programável utilizado. O item 4.2 apresenta a parte de software, que envolve toda tecnologia que permite a comunicação via Internet do subsistema de transporte, como o OPC Server e o *web service*. A efetiva integração dos subsistemas é descrita no item 4.3, onde se apresenta a arquitetura adotada, a SOA (Arquitetura Orientada a Serviços). Por fim, os itens 4.4 e 4.5 descrevem a metodologia adotada para a modelagem e implementação das funções de controle, os testes realizados e, finalmente, o funcionamento das interfaces do cliente.

## 4.1. SUBSISTEMA DE TRANSPORTE - HARDWARE

A seguir, a Figura 16 apresenta um diagrama esquemático do subsistema de transporte (hardware).

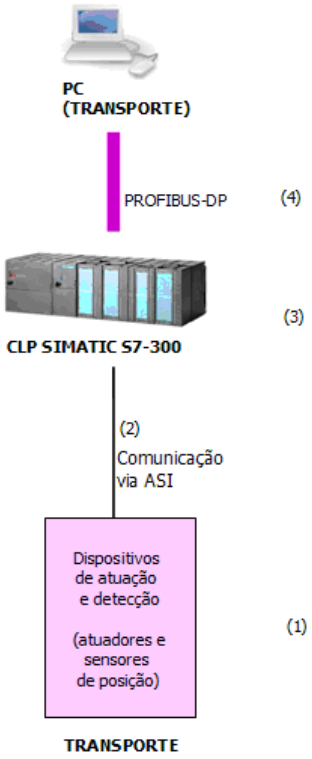



*Figura 16 - Diagrama esquemático do subsistema de transporte (hardware)*

A Figura 16 foi enumerada para facilitar o entendimento dos próximos itens do relatório. A descrição dos dispositivos de detecção e atuação (1), assim como a listagem dos sensores e atuadores será feito nos itens 4.1.1, 4.1.2, 4.1.3 e 4.1.4. A comunicação via ASI (2) será abordado no item 4.1.5. Em seguida, no item 4.1.6, faz-se uma breve introdução do controle programável (3) utilizado na implementação. Por fim, o item 4.1.7 conclui com uma abordagem da comunicação PROFIBUS empregada para a comunicação do PC e o controlador programável (4).

### 4.1.1. Planta do subsistema de transporte

O SP disperso aqui estudado pode ser dividido em quatro subsistemas (Figura 17), de alimentação, de inspeção, de montagem e de transporte. Alguns dos dispositivos de controle são utilizados por apenas um desses subsistemas, enquanto outros são compartilhados.

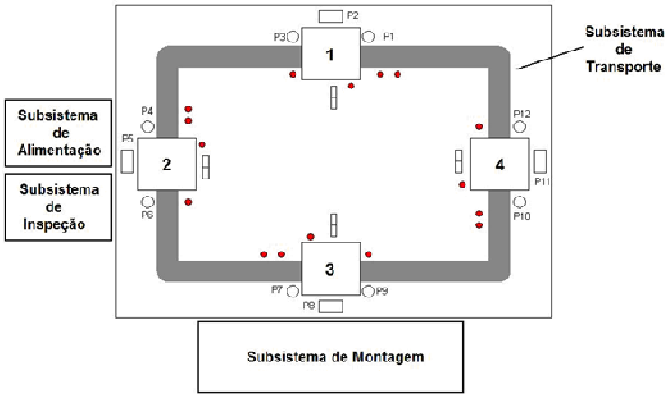



*Figura 17 - Esquema do subsistema de transporte (KANEKO, 2008)*

O subsistema de transporte, foco deste trabalho, é composto por uma série de esteiras de transporte e cinco *pallets* que transportam os “corpos” aprovados do subsistema de inspeção até o de montagem.

O “*pallet*” aqui é um tipo de “bandeja” ou suporte feito de alumínio que possui um formato próprio para o seu encaixe nas esteiras. Para facilitar o transporte das peças, o *pallet* possui uma estrutura auxiliar em formato de “copo” com base circular, que permite o

encaixe das peças “corpo”. Esta estrutura auxiliar em formato de “copo” é exposta na Figura 18.

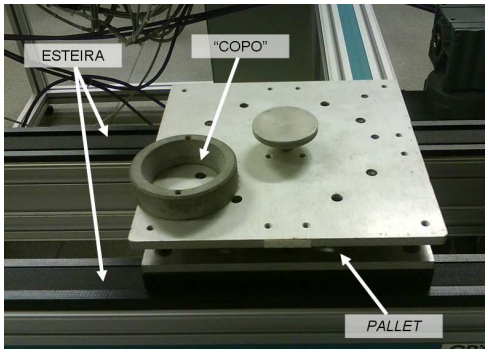



*Figura 18 - “Copo” do Pallet*

O subsistema de transporte possui quatro esteiras lineares dispostas, cada uma, perpendicularmente entre as duas esteiras vizinhas. Quando os *pallets* estão apoiados nas esteiras, elas se movimentam juntas. Guias curvadas em 90° facilitam a movimentação dos *pallets* nas junções (Figura 19). A movimentação das esteiras é feita por quatro motores elétricos auxiliares. Nos locais de carregamento e descarregamento de peças (estação 1, estação 2, estação 3 e estação 4), existem dispositivos que desacoplam os *pallets* das esteiras e pistões que quando acionados limitam a parada dos *pallets* que assim ficam em posições fixas independentemente da movimentação das esteiras.

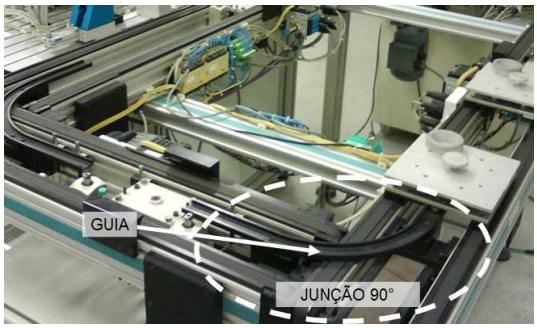



*Figura 19 - Guias curvadas nas junções*

O subsistema de transporte tem definido quatro posições distintas para a parada dos *pallets*: estação 1, estação 2, estação 3 e estação 4 (ver Figura 20). Para o controle de movimentação desses *pallets* tem-se disponível um controlador (Dispositivo de realização de controle) e dispositivos de atuação e detecção.

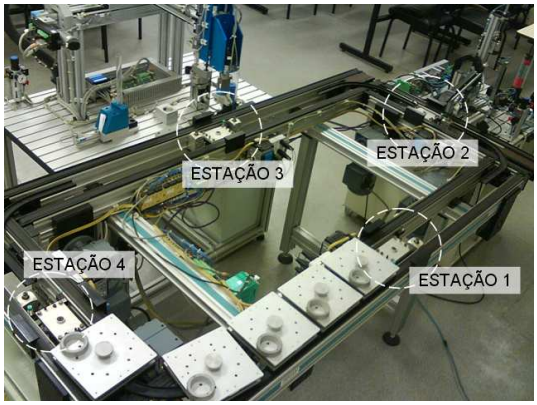



*Figura 20 - Estações de transporte do subsistema*

### 4.1.2. Dispositivos de detecção e atuação

A localização dos *pallets* é feita através de sinais dos seis dispositivos de detecção disponíveis em cada estação. Essa informação, associada aos comandos e programa de controle do controlador, resulta em sinais para os dispositivos de atuação para o acionamento ou recolhimento das três travas (atuadores) de cada estação. As travas e os sensores são apresentados na Figura 21.

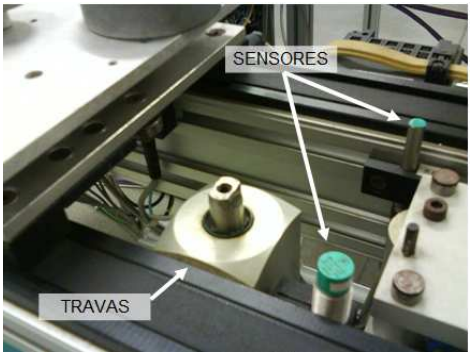



*Figura 21 - Detalhe da trava na entrada da estação e os sensores*

### 4.1.3. Listagem dos sensores

A seguir pode ser vista a listagem dos dispositivos de detecção (sensores) em cada estação (ver Figura 22 e Tabela 1).

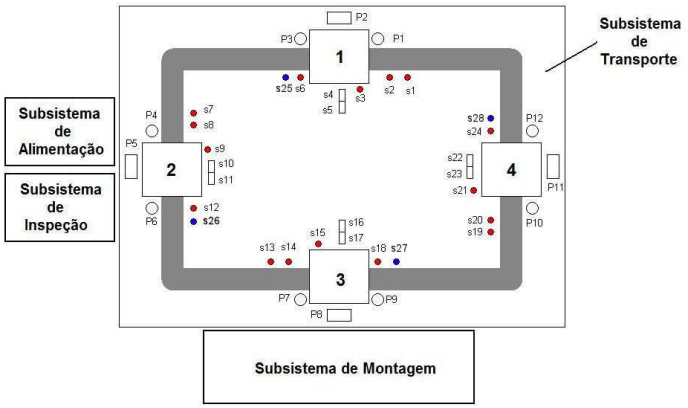



*Figura 22 - Nomenclatura adotada para os atuadores e sensores da estação de transporte (adaptado de KANEKO, 2008)*

*Tabela 1 - Tabela de sensores*

| Sinal | Flag | Tipo | Descrição | Sensor |
|---|---|---|---|---|
| S_Cil_Estend_1 | M 10.0 | BOOL | Informa que o cilindro da estação 1 esta estendido | s4 |
| S_Cil_Estend_2 | M 11.0 | BOOL | Informa que o cilindro da estação 2 esta estendido | s10 |
| S_Cil_Estend_3 | M 12.0 | BOOL | Informa que o cilindro da estação 3 esta estendido | s16 |
| S_Cil_Estend_4 | M 13.0 | BOOL | Informa que o cilindro da estação 4 esta estendido | s22 |
| S_Cil_Recuado_1 | M 10.1 | BOOL | Informa que o cilindro da estação 1 esta recuado | s5 |
| S_Cil_Recuado_2 | M 11.1 | BOOL | Informa que o cilindro da estação 2 esta recuado | s11 |
| S_Cil_Recuado_3 | M 12.1 | BOOL | Informa que o cilindro da estação 3 esta recuado | s17 |
| S_Cil_Recuado_4 | M 13.1 | BOOL | Informa que o cilindro da estação 4 esta recuado | s23 |
| S_Ent1-Est1 | M 10.2 | BOOL | informa que o *pallet* esta na entrada da estação 1 | s1 |
| S_Ent1-Est2 | M 11.2 | BOOL | informa que o *pallet* esta na entrada da estação 2 | s7 |
| S_Ent1-Est3 | M 12.2 | BOOL | informa que o *pallet* esta na entrada da estação 3 | s13 |
| S_Ent1-Est4 | M 13.2 | BOOL | informa que o *pallet* esta na entrada da estação 4 | s19 |
| S_Ent2-Est1 | M 10.3 | BOOL | informa que o *pallet* esta entrando na estação 1 | s2 |
| S_Ent2-Est2 | M 11.3 | BOOL | informa que o *pallet* esta entrando na estação 2 | s8 |
| S_Ent2-Est3 | M 12.3 | BOOL | informa que o *pallet* esta entrando na estação 3 | s14 |
| S_Ent2-Est4 | M 13.3 | BOOL | informa que o *pallet* esta entrando na estação 4 | s20 |
| S_Est1 | M 10.4 | BOOL | informa que o *pallet* esta na estação 1 | s3 |
| S_Est2 | M 11.4 | BOOL | informa que o *pallet* esta na estação 2 | s9 |
| S_Est3 | M 12.4 | BOOL | informa que o *pallet* esta na estação 3 | s15 |
| S_Est4 | M 13.4 | BOOL | informa que o *pallet* esta na estação 4 | s21 |
| S_Saida-Est1 | M 10.5 | BOOL | informa que o *pallet* esta saindo da estação 1 | s6 |
| S_Saida-Est2 | M 11.5 | BOOL | informa que o *pallet* esta saindo da estação 2 | s12 |
| S_Saida-Est3 | M 12.5 | BOOL | informa que o *pallet* esta saindo da estação 3 | s18 |
| S_Saida-Est4 | M 13.5 | BOOL | informa que o *pallet* esta saindo da estação 4 | s24 |

Os sensores em vermelho são responsáveis pela identificação da posição dos *pallets* nas estações 1 a 4 e se comunicam através da rede ASI. Os sensores em azul são responsáveis pela identificação do *pallet* que está em cada uma destas 4 estações e se comunicam através do protocolo RS-232

### 4.1.4. Listagem dos atuadores

Em cada estação temos três dispositivos de atuação (atuadores). Todos os atuadores são pistões de acionamento pneumático, sendo o primeiro responsável pela retenção do *pallet* antes da estação, o segundo pela retenção dentro do *pallet* da estação e o terceiro pela estabilização deste quando dentro da estação. Abaixo é possível ver a listagem dos atuadores do subsistema de transporte (Tabela 2).

*Tabela 2 - Tabela de Atuadores*

| Sinal | Flag | Tipo | Descrição |
|---|---|---|---|
| A_Cil_Ent_Est1 | M 20.0 | BOOL | Permite a entrada do *pallet* a estação 1 |
| A_Cil_Ent_Est2 | M 21.0 | BOOL | Permite a entrada do *pallet* a estação 2 |
| A_Cil_Ent_Est3 | M 22.0 | BOOL | Permite a entrada do *pallet* a estação 3 |
| A_Cil_Ent_Est4 | M 23.0 | BOOL | Permite a entrada do *pallet* a estação 4 |
| A_Cil_Saida_Est1 | M 20.2 | BOOL | Permite a saída do *pallet* a estação 1 |
| A_Cil_Saida_Est2 | M 21.2 | BOOL | Permite a saída do *pallet* a estação 2 |
| A_Cil_Saida_Est3 | M 22.2 | BOOL | Permite a saída do *pallet* a estação 3 |
| A_Cil_Saida_Est4 | M 23.2 | BOOL | Permite a saída do *pallet* a estação 4 |
| A_Cil_Trav_Est1 | M 20.1 | BOOL | Permite segurar o *pallet* a estação 1 |
| A_Cil_Trav_Est2 | M 21.1 | BOOL | Permite segurar o *pallet* a estação 2 |
| A_Cil_Trav_Est3 | M 22.1 | BOOL | Permite segurar o *pallet* a estação 3 |
| A_Cil_Trav_Est4 | M 23.1 | BOOL | Permite segurar o *pallet* a estação 4 |

### 4.1.5. Comunicação via ASI - Identificação da posição dos pallets e acionamento dos pallets

A comunicação do controlador com os dispositivos de atuação e detecção utiliza a comunicação via ASI e as funções de “mestre” é implementado por um módulo CP342-2

(Figura 23) do fabricante Siemens, que permite a interligação física dos sensores e atuadores.


*Figura 23 - Módulo ASI utilizado no projeto (SIEMENS, 2008).*

### 4.1.6. Controlador Programável

O hardware do dispositivo de realização do controle local (controlador) do subsistema de transporte é composto por diversas partes. O CLP SIMATIC S7-300 é um controlador modular amplamente utilizado em aplicações industriais centralizadas ou distribuído de pequeno a médio porte. A Figura 24 apresenta uma foto do SIMATIC S7-300 e a indicação dos seguintes elementos: (1) Fonte de energia; (2) Bateria de reserva; (3) Conector 24 V DC; (4) Chave de operação; (5) Status e falhas (LEDs); (6) Cartão de memória; (7) MPI (Interface multi-ponto); (8) Conector frontal e (9) Porta da frente.

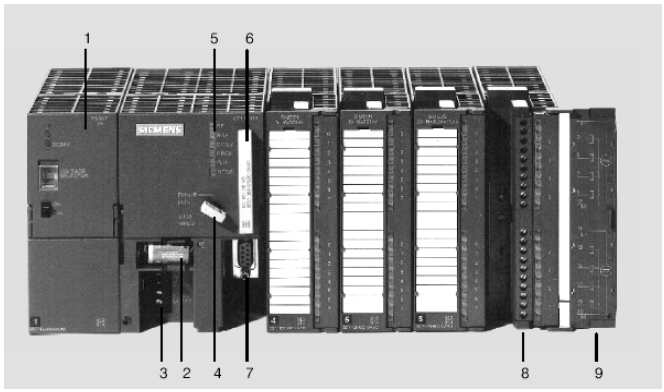



*Figura 24 - SIMATIC S7-300 design (PINHEIRO, 2006).*

### 4.1.7. Comunicação PC - CLP

Uma rede de comunicação é a parte responsável pela comunicação entre o PC e o CLP. Atualmente, existe uma tendência nos sistemas de controle de processos distribuídos

de conectarem os elementos distribuídos através de um único barramento com pontos de acesso espalhados, ao invés de conexões tradicionais *point-to-point*. Nesse contexto, as redes FIELDBUS surgiram como solução para comunicar controladores, sensores e atuadores, no nível mais baixo da automação (FIELDBUS, 2010).

Neste trabalho, a versão de PROFIBUS utilizada é o PROFIBUS DP, otimizado para alta velocidade de transmissão e conexão de baixo custo, e que foi projetado especialmente para a comunicação entre sistemas de controle de automação e elementos distribuídos no nível de dispositivo (KANEKO, 2008).

### 4.1.8. DEMAIS SUBSISTEMAS - HARDWARE

Os demais subsistemas que compõem o sistema produtivo (isto é, subsistemas de alimentação, de inspeção e de montagem) possuem uma configuração de hardware similar ao do subsistema de transporte, anteriormente descrito. A seguir, a Figura 25 apresenta um diagrama esquemático dos quatro subsistemas (hardware).

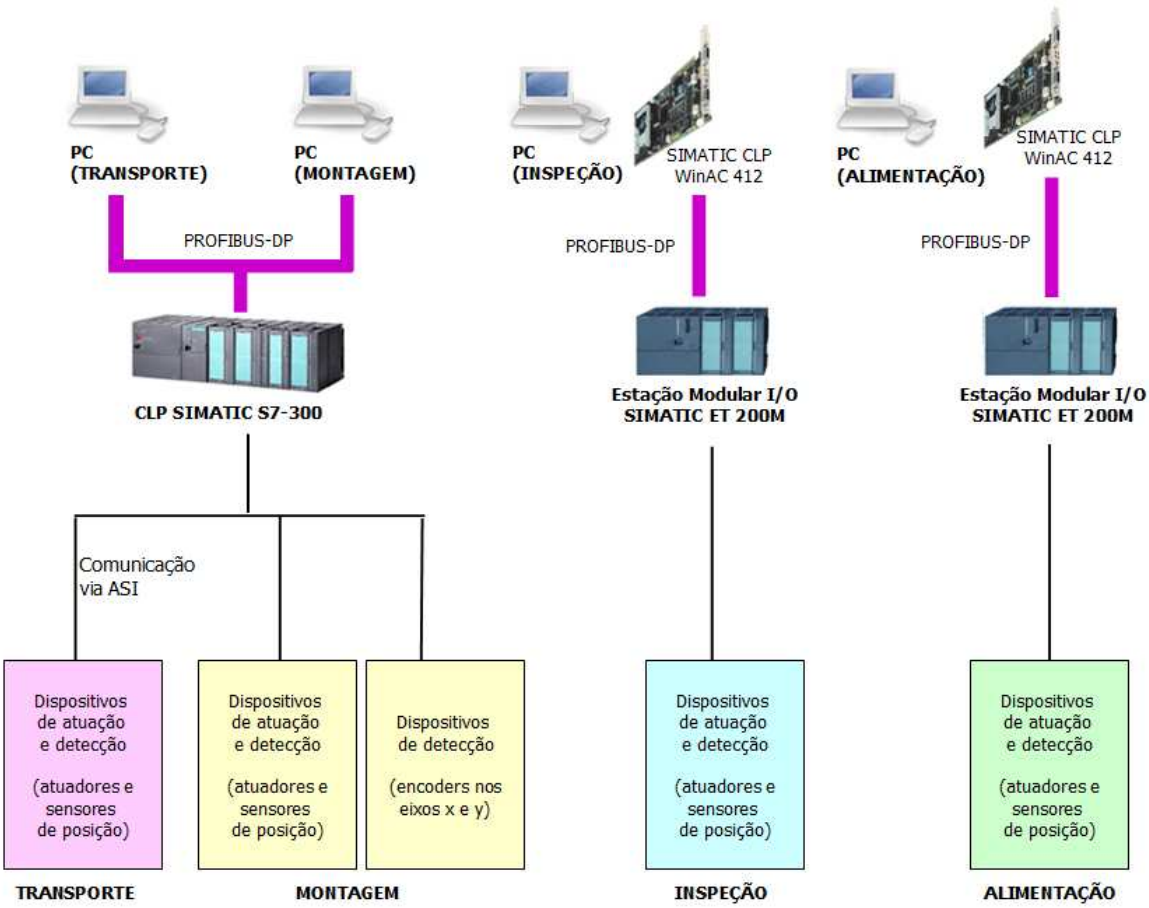



*Figura 25 - Diagrama esquemático dos quatro subsistemas (hardware)*

No sistema produtivo em estudo, o CLP Simatic S7-300 (Figura 25) é o controlador programável dos subsistemas de transporte e montagem. Similar ao subsistema de transporte, o PC se comunica com o CLP via PROFIBUS. Paralelamente, os dispositivos de atuação e detecção (sensores de posição e encoders nos eixos x e y do braço da montagem) são controlados pelo CLP.

Ressalta-se que o CLP Simatic S7-300, o PC do subsistema de transporte e o PC do subsistema de montagem estão ligados em uma rede PROFIBUS. Nesse caso, o CLP controla os dois subsistemas, mas, caso tivéssemos dois CLPs, os PCs poderiam estar em redes PROFIBUS separadas.

Os subsistemas de inspeção e alimentação possuem uma diferença quanto ao local físico do CLP. Diferentemente dos outros dois subsistemas, estes possuem um CLP (Simatic WinAC 412) instalado nos PCs e via PROFIBUS é feita a comunicação com uma estação modular I/O (Simatic ET200M). Na inspeção e alimentação, os dispositivos de atuação e detecção se ligam diretamente à estação modular.

## 4.2. SUBSISTEMA DE TRANSPORTE - SOFTWARE

A seguir, a Figura 26 apresenta um diagrama esquemático do subsistema de transporte (software).

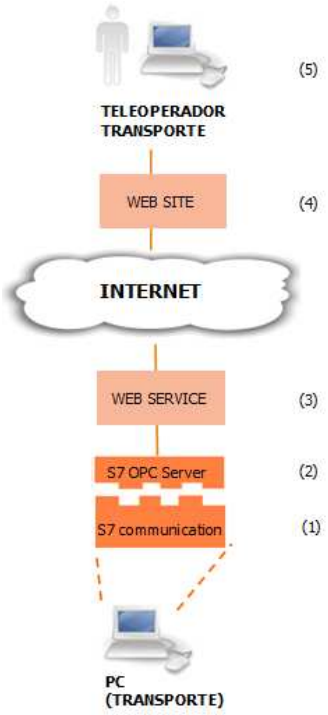



*Figura 26 - Diagrama esquemático do subsistema de transporte (software)*

A Figura 26 foi enumerada para facilitar o entendimento dos próximos itens do relatório. A descrição da tecnologia OPC Server (2), utilizada para a comunicação entre o controlador (1) e a WS (3), é descrita nos itens 4.2.1 e 4.2.2. A seguir, descreve-se o funcionamento da WS (3), no item 4.2.3; da aplicação Web (4), no item 4.2.4; e por fim, os modos de operações do usuário (5) no item 4.2.5.

### 4.2.1. Transferência da lógica de controle para o CLP

A lógica de controle é transferida ao CLP através do software SIMATIC Manager, uma ferramenta computacional que permite a configuração, programação, compilação e envio dos programas para o CLP. O SIMATIC Manager permite o uso de três linguagens de programação: Ladder, Lista de Instruções e Function Block.

Uma vez transferido, o CLP pode executar as funções criadas independentemente do PC, contudo, no presente trabalho, a conexão com o PC é fundamental para o controle via internet do CLP. Para que seja possível o acesso remoto das funções programadas foi feito o mapeamento das variáveis de memória do CLP através da aplicação OPC Scout.

### 4.2.2. OPC Server

No presente trabalho, o aplicativo OPC Scout foi utilizado para o mapeamento das variáveis de memória do CLP no PC. Através do servidor OPC criado, o WS (cliente) é capaz de acessar as informações (status dos sensores e atuadores) e acionar os atuadores.

### 4.2.3. Web Service do subsistema de transporte

Como já descrito no item 2.1.4, o WS consiste em um conjunto de protocolos padrões que permitem a criação, distribuição e integração de aplicações na internet. Neste trabalho, utilizou-se essa tecnologia para a disponibilização das funções de controle do CLP na internet.

### 4.2.4. Web Site

O Web Site consiste na interface web pela qual o usuário (no caso, o teleoperador) se comunica com o *Web Service* por meio da internet, permitindo a leitura dos dispositivos de detecção e o acionamento dos dispositivos de atuação do sistema flexível. A Figura 27, a seguir, ilustra a interface criada para o subsistema de transporte.


*Figura 27 - Web Site criado para o subsistema de transporte*

### 4.2.5. Teleoperador

Definem-se quatro modos de operação para o usuário: modo automático e manual (ambos para controle local); e modo de monitoração e teleoperação (ambos para controle remoto).

No modo local, o usuário pode optar por dois tipos de modos de operação: automático e manual. No modo manual, o usuário pode executar a mesma sequencia de funções do modo automático, porém, passo a passo. A implementação desse modo é possível através do painel de controle disponível nas proximidades do CLP Simatic S7-300, na estação de montagem e transporte.

O presente trabalho aborda o modo de operação remoto. Nesse modo, o usuário é denominado como “Teleoperador”. O teleoperador é responsável pela “monitoração”, definida pelo acompanhamento das atividades do subsistema através de uma interface

(Web Site) e deve ser capaz, também, de fazer a “teleoperação”, ou seja, comandar diretamente as funções do subsistema em local remoto.

### 4.2.6. DEMAIS SUBSISTEMAS - SOFTWARE

Os demais subsistemas que compõem o sistema produtivo (isto é, os subsistemas de alimentação, de inspeção e de montagem) possuem uma configuração de software similar ao do subsistema de transporte, anteriormente descrito. A seguir, a Figura 28 apresenta um diagrama esquemático dos quatro subsistemas (software).

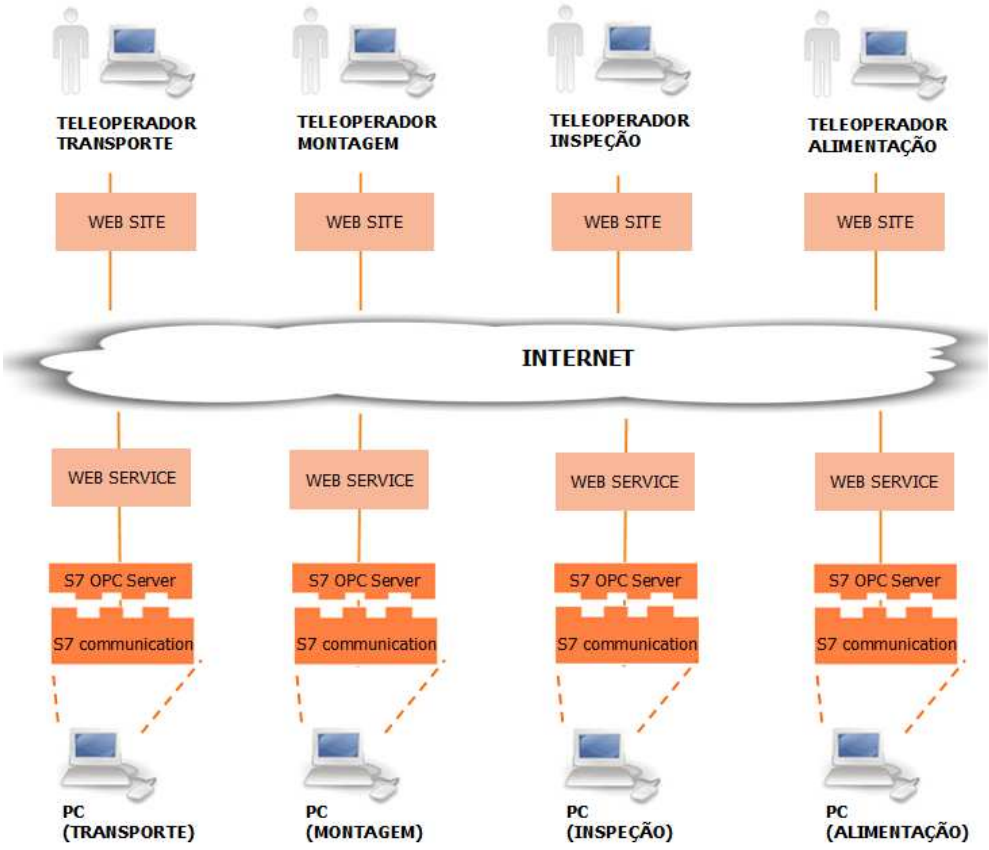



*Figura 28 - Diagrama esquemático dos quatro subsistemas (software)*

Resumidamente, os quatro subsistemas utilizam-se do software SIMATIC Manager para a transferência das lógicas de controle para os CLPs. Neles, o OPC Server garante que as WSs criadas para cada subsistema acessem as variáveis de memória dos CLPs. Por fim, no sistema flexível, considera-se que cada subsistema possui um teleoperador exclusivo que acessa a interface criada, no modo de monitoração ou teleoperação.

### 4.3. INTEGRAÇÃO

Os itens de 4.1 a 4.2 descreveram o funcionamento do hardware e software dos subsistemas de alimentação, inspeção, transporte e montagem. Essa parte do relatório foca-se na integração dos quatro subsistemas. A seguir, a Figura 29 ilustra os componentes hardware e software do SP.

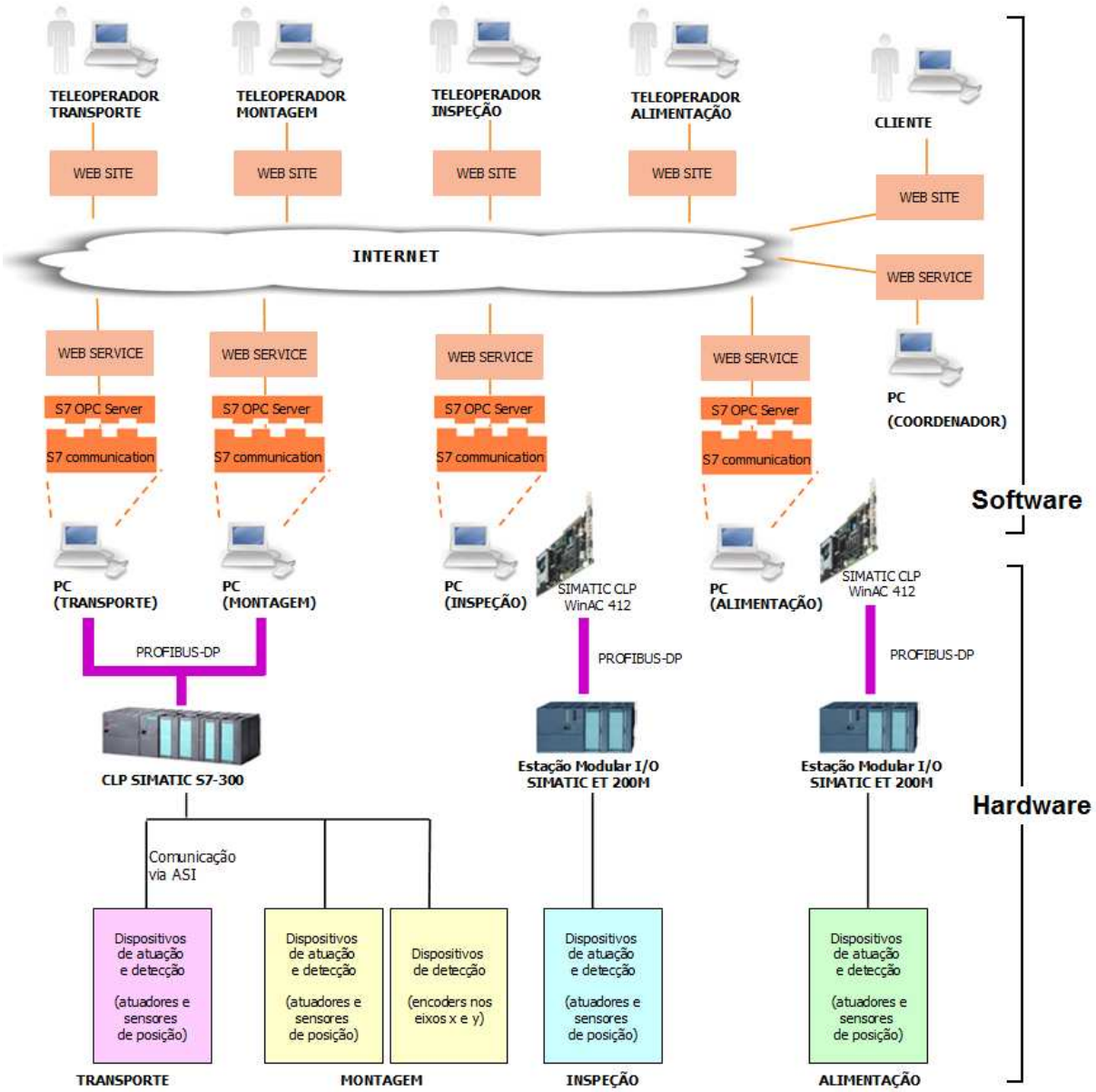



*Figura 29 - Componentes hardware e software do sistema flexível*

Na integração do sistema produtivo, considera-se que os subsistemas são WSs (A), ou seja, módulos funcionais que disponibilizam serviços para outros WSs. Os serviços dos

WSs são mostrados aos teleoperadores através de uma interface, um Web Site (B). Este teleoperador (C) pode acessar as funcionalidades de dois modos: monitoração e teleoperação, anteriormente descritos no item 4.2.5.

Para a coordenação dos serviços disponibilizados por cada WSs, um WS Coordenador (D) é criado. Sua função é consumir os serviços de cada subsistema de forma ordenada, a fim de atender aos pedidos do cliente (F). Da mesma forma que os WSs dos subsistemas, o WS Coordenador possui um Web Site (E) cuja função é disponibilizar ao cliente as funções do sistema flexível.

### 4.3.1. Estrutura dos web services

A Figura 30 apresenta os WSs implementados, seus respectivos serviços e os bancos de dados necessários.

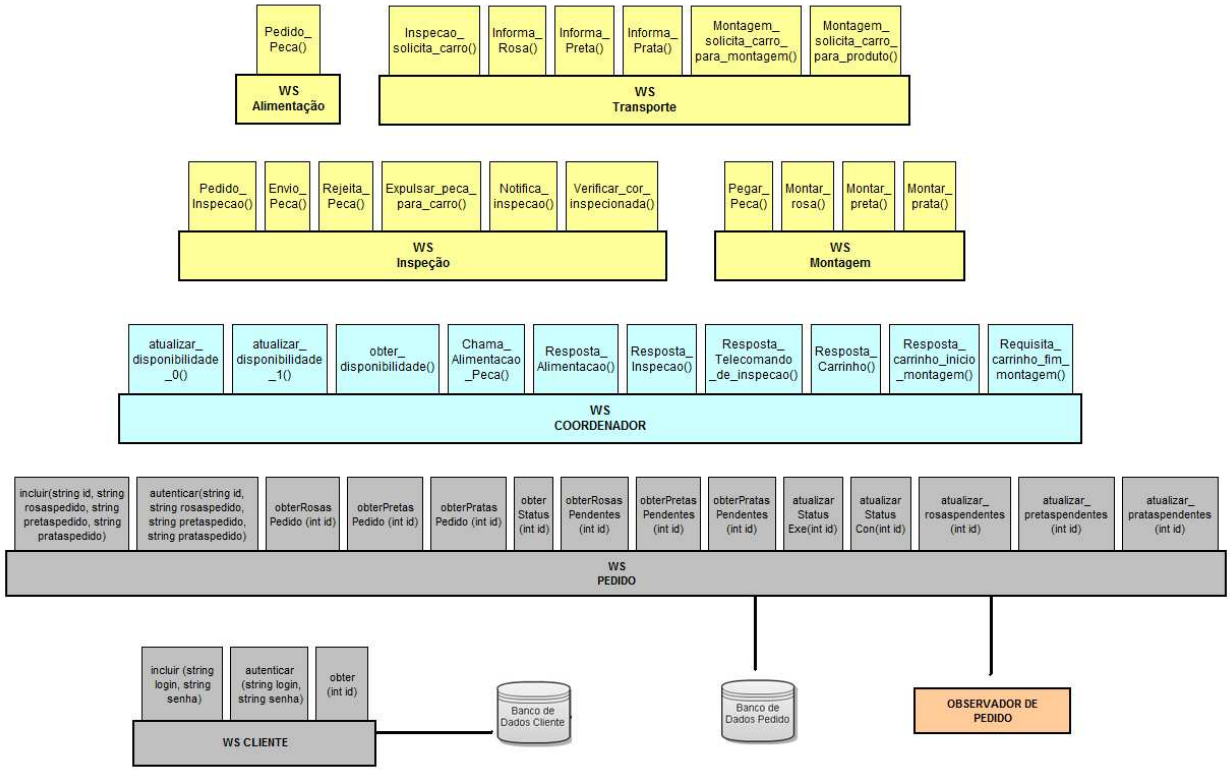



*Figura 30 - Estrutura dos WSs*

Existem sete WSs: (1) WS Cliente, para administração das informações do cliente; (2) WS Pedido, para administração das informações dos pedidos; (3) WS Coordenação, responsável pela integração dos WSs; (4) WS de Alimentação, com os serviços do subsistema de alimentação; (5) WS de Inspeção, com os serviços do subsistema de inspeção; (6) WS de Transporte, com os serviços do subsistema de transporte; e (7) WS de Montagem, com os serviços do subsistema de montagem.

Considerou-se, também, necessário a criação de dois bancos de dados: para registrar as informações do cliente e pedido.

### 4.3.2. Mapeamento OPC de cada subsistema

Os WSs criados utilizam-se de funções definidas no OPC Server. A seguir, a listagem do OPC do subsistema de Transporte. Os demais mapeamentos estão no Apêndice B.

*Tabela 3 - Mapeamento OPC do Subsistema de Transporte*

| **Nome do item** | **Tipo** | **Acesso** |
|---|---|---|
| S7:[S7Teleoperacao]Req_Inspecao_Est2 | bool | RW |
| S7:[S7Teleoperacao]Saida_autorizada_Est2 | bool | RW |
| S7:[S7Teleoperacao]Cor_pedido_Est2_Prata | uint16 | RW |
| S7:[S7Teleoperacao]Cor_pedido_Est2_Preta | uint16 | RW |
| S7:[S7Teleoperacao]Cor_pedido_Est2_Rosa | uint16 | RW |
| S7:[S7Teleoperacao]Cor_pedido_Est2 | uint16 | RW |
| S7:[S7Teleoperacao]Resp_req_car | bool | RW |
| S7:[S7Teleoperacao]Carro_travado_St3 | bool | RW |
| S7:[S7Teleoperacao]Req_peca_st3 | bool | RW |
| S7:[S7Teleoperacao]Req_produto_montado_st3 | bool | RW |
| S7:[S7Teleoperacao]Saida_Autorizada_St3 | bool | RW |
| S7:[S7Teleoperacao]Saida_Autorizada_St3_2 | bool | RW |

### 4.3.3. Descrição dos web services

A seguir, os serviços dos WS são apresentados nas Figura 31 a Figura 56 os respectivos diagramas de seqüência.

| **WS ALIMENTAÇÃO** |
| --- |
| **void: Pedido_Peca()**<br>Requisita “corpo” do subsistema de alimentação para o subsistema de inspeção. |

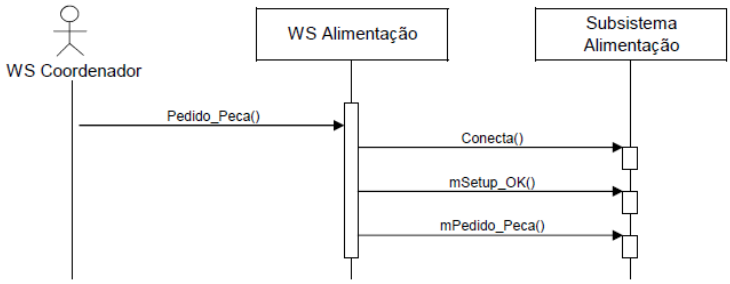



*Figura 31 – Diagrama de sequência do serviço Pedido_Peca() do WS Alimentação.*

| **WS INSPEÇÃO** |
| --- |
| **int: Pedido_Inspecao()**<br>Requisita inspeção do “corpo” para o subsistema de inspeção.<br><br>**Retorna: *0***, se a cor do “corpo” inspecionado é rosa<br>***1***, se a cor do “corpo” inspecionado é preta<br>***2***, se a cor do “corpo” inspecionado é prata |

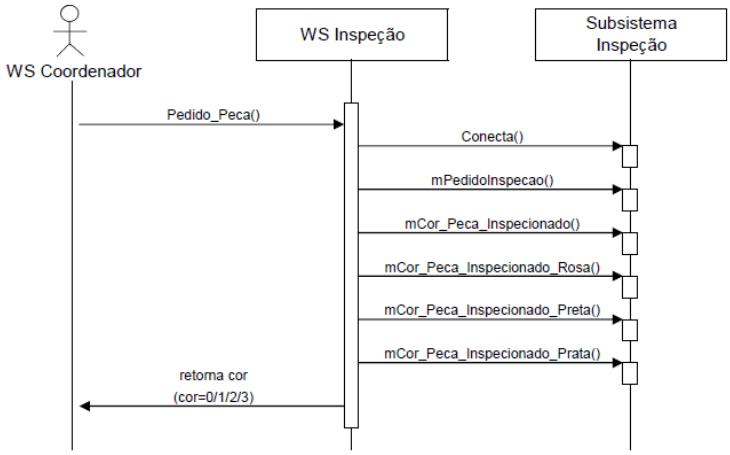



*Figura 32 - Diagrama de sequência do serviço Pedido_Inspecao() do WS Inspeção.*

**void: Envio_Peca()**

Indica que a cor inspecionada corresponde à cor de produto do pedido, e requisita liberação do “corpo” inspecionado para o subsistema de transporte.

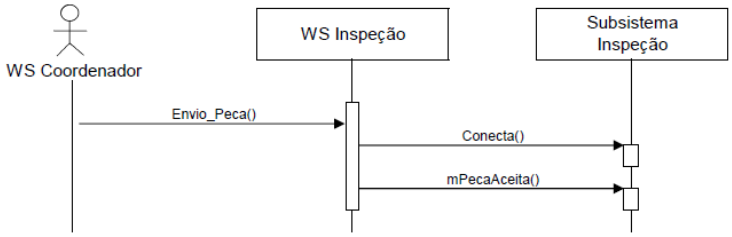



*Figura 33 - Diagrama de sequência do serviço Envio_Peca() do WS Inspeção.*

**void: Rejeita_Peca()**

Indica que a cor inspecionada não corresponde à cor de produto do pedido, e requisita o descarte do “corpo” inspecionado.

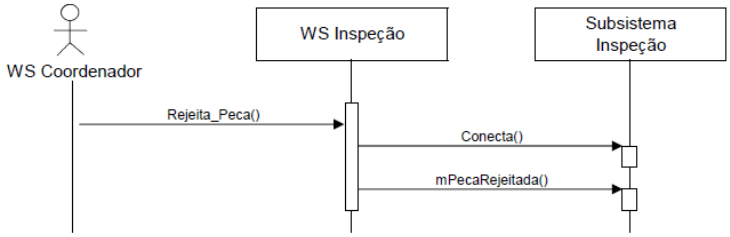



*Figura 34 - Diagrama de sequência do serviço Rejeita_Peca() do WS Inspeção.*

**void: Expulsar_peca_para_carro()**

Indica que existe um *pallet* travado na estação 2 (subsistema de inspeção), e requisita a liberação do “corpo” para o *pallet*.

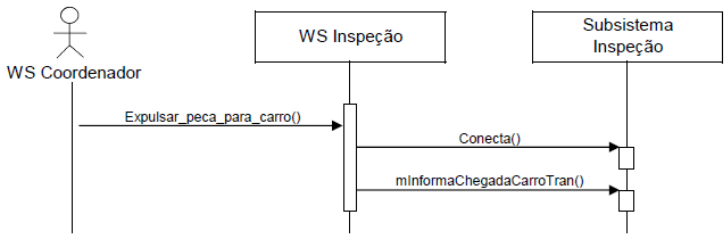



*Figura 35 - Diagrama de sequência do serviço Expulsar_peca_para_carro() do WS Inspeção.*

**void: Notifica_inspecao()**

Informa que o subsistema de inspeção concluiu seu serviço.

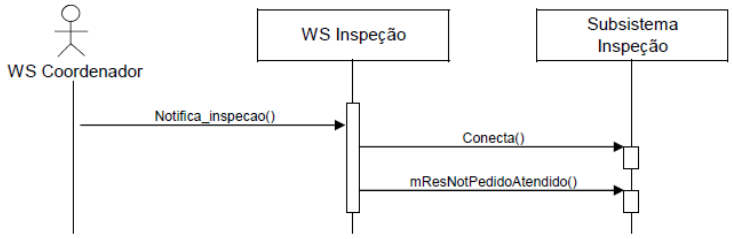



*Figura 36 - Diagrama de sequência do serviço Notifica_inspecao() do WS Inspeção.*

**int: Verificar_cor_inspecionada()**

Requisita ao subsistema de inspeção a cor do “corpo” que foi inspecionada.

**Retorna: *0***, se a cor do “corpo” inspecionado é rosa

***1***, se a cor do “corpo” inspecionado é preta

***2***, se a cor do “corpo” inspecionado é prata

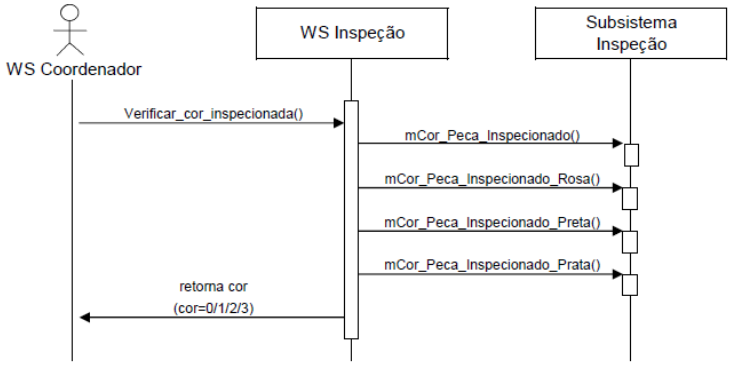



*Figura 37 - Diagrama de sequência do serviço Verificar_cor_inspecionada() do WS Inspeção.*

| **WS TRANSPORTE** |
| --- |
| **void: Inspeção_solicita_carro()**<br>Requisita um pallet vazio na estação 2 (subsistema de inspeção). |

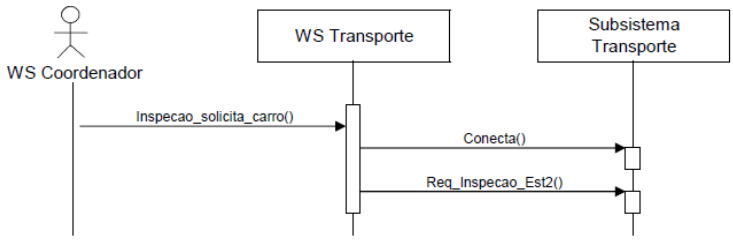



*Figura 38 - Diagrama de sequência do serviço Inspecao_solicita_carro() do WS Inspeção.*

| **void: Informa_Rosa()**<br>Indica que o “corpo” enviado ao *pallet* travado na estação 2 é rosa. |
| --- |

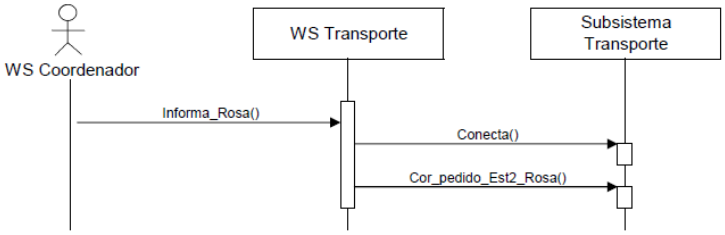



*Figura 39 - Diagrama de sequência do serviço Informa_Rosa() do WS Inspeção.*

| **void: Informa_Preta()**<br>Indica que o “corpo” enviado ao *pallet* travado na estação 2 é preta. |
| --- |

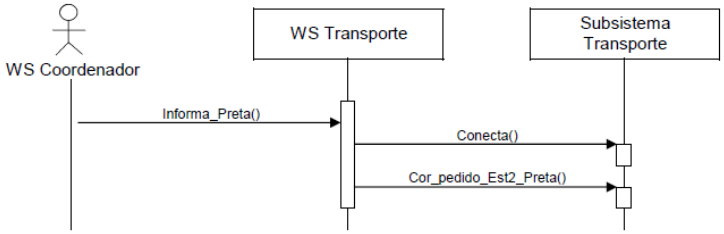



*Figura 40 - Diagrama de sequência do serviço Informa_Preta() do WS Inspeção.*

| **void: Informa_Prata()** |
| --- |
| Indica que o “corpo” enviado ao *pallet* travado na estação 2 é prata. |

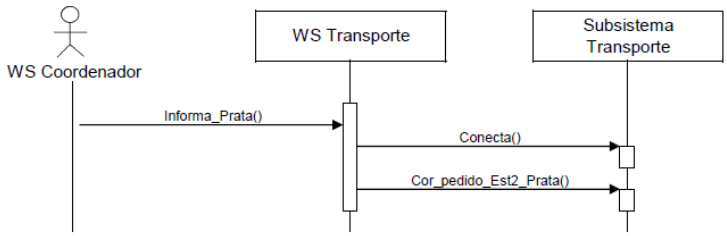



*Figura 41 - Diagrama de sequência do serviço Informa_Prata() do WS Inspeção.*

| **void: Montagem_solicita_carro_para_montagem()** |
| --- |
| Requisita um *pallet* com “corpo” na estação 3 (subsistema de montagem). |

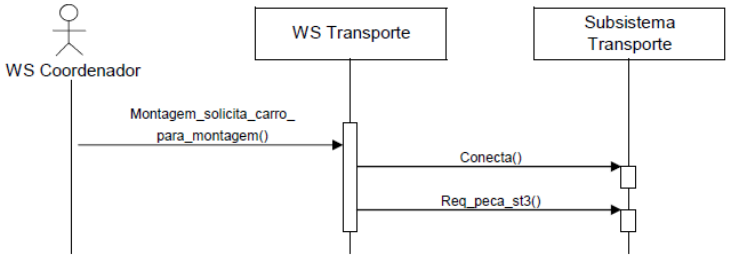



*Figura 42 - Diagrama de sequência do serviço Montagem_solicita_carro_para_montagem() do WS Inspeção.*

| **void: Montagem_solicita_carro_para_produto()** |
| --- |
| Requisita um *pallet* vazio na estação 3 (subsistema de montagem) para enviar produto montado. |

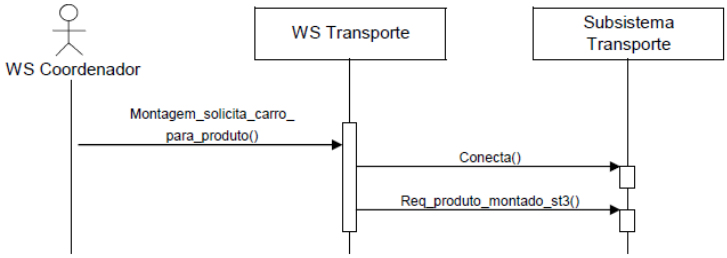



*Figura 43 - Diagrama de sequência do serviço Montagem_solicita_carro_para_produto() do WS Inspeção.*

| **WS MONTAGEM** |
| --- |
| **void: Pegar_Peca()**<br>Informa ao subsistema de montagem que existe um *pallet* travado com “corpo” na estação 3, e requisita que o “corpo” seja retirado pelo braço mecânico. |

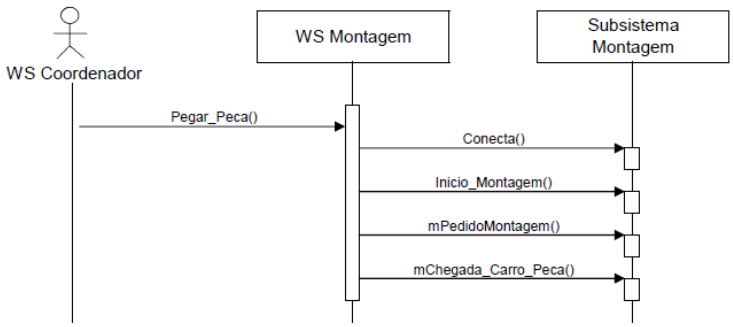



*Figura 44 - Diagrama de sequência do serviço Pegar_Peca() do WS Montagem.*

| **void: Montar_rosa()**<br>Informa ao subsistema de montagem que o “corpo” é rosa e deve ser montado com o “pino” preto, “mola” e “tampa”. |
| --- |

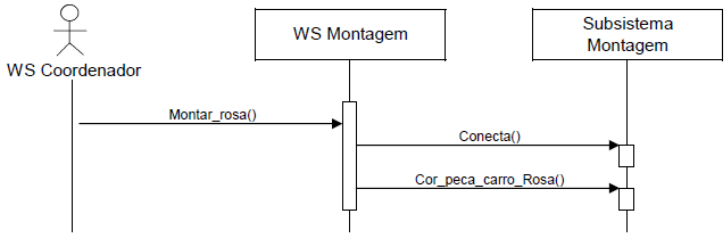



*Figura 45 - Diagrama de sequência do serviço Montar_rosa() do WS Montagem.*

| **void: Montar_preta()** |
| --- |
| Informa ao subsistema de montagem que o “corpo” é preto e deve ser montado com o “pino” prata, “mola” e “tampa”. |

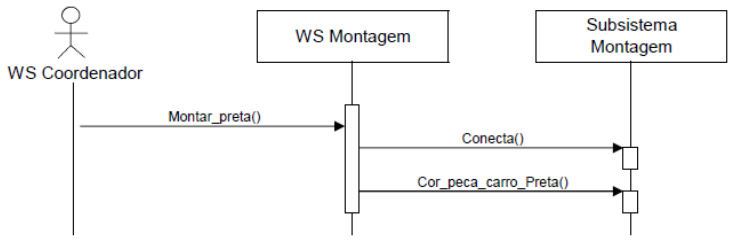



*Figura 46 - Diagrama de sequência do serviço Montar_preta() do WS Montagem.*

| **void: Montar_prata()** |
| --- |
| Informa ao subsistema de montagem que o “corpo” é prato e deve ser montado com o “pino” preto, “mola” e “tampa”. |

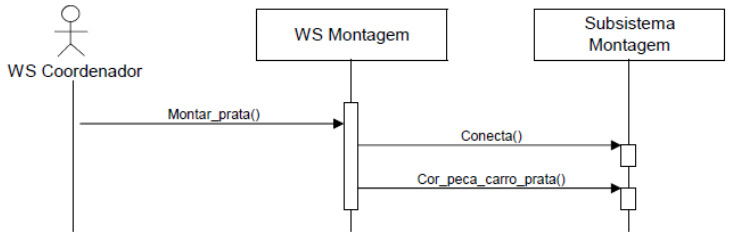



*Figura 47 - Diagrama de sequência do serviço Montar_prata() do WS Montagem.*

| **WS COORDENADOR** |
| --- |
| **void: atualizar_disponibilidade_0()** |
| Informa que o sistema produtivo está ocupado com um produto e, por isso, não está disponível. |

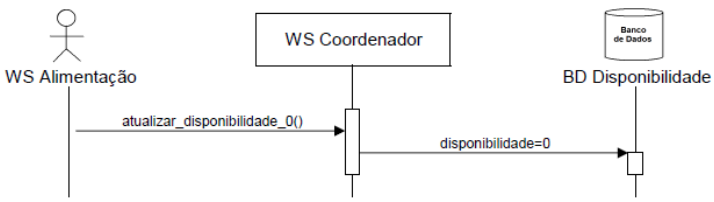



*Figura 48 - Diagrama de sequência do serviço atualizar_disponibilidade_0() do WS Coordenador.*

| **void: atualizar_disponibilidade_1()** |
| --- |
| Informa que o sistema produtivo está não ocupado e, por isso, está disponível para fazer um produto. |

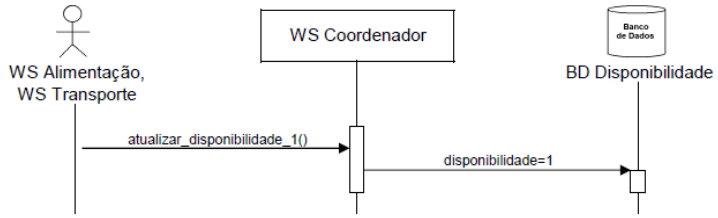



*Figura 49 - Diagrama de sequência do serviço atualizar_disponibilidade_1() do WS Coordenador.*

| **int: obter_disponibilidade()** |
| --- |
| Informa se o sistema produtivo está disponível ou não para fazer um produto. |

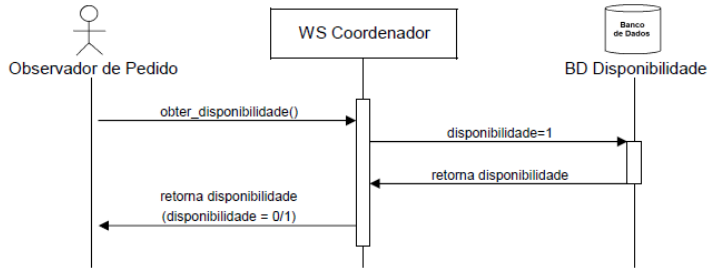



*Figura 50 - Diagrama de sequência do serviço obter_disponibilidade () do WS Coordenador.*

| **void: Chama_Alimentacao_Peca()** |
| --- |
| Requisita um “corpo” ao subsistema de alimentação. |

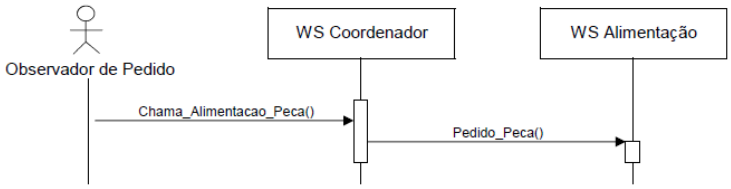



*Figura 51 - Diagrama de sequência do serviço Chama_Alimentacao_Peca() do WS Coordenador.*

**void: Resposta_Alimentacao()**

Subsistema de alimentação indica que finalizou o envio do “corpo” para o subsistema de inspeção e verifica se o “corpo” inspecionado equivale ao pedido feito pelo cliente. Caso seja equivalente ao pedido, aceita o “corpo” e requisita um *pallet* vazio para a estação 2. Caso contrário, rejeita o “corpo”.

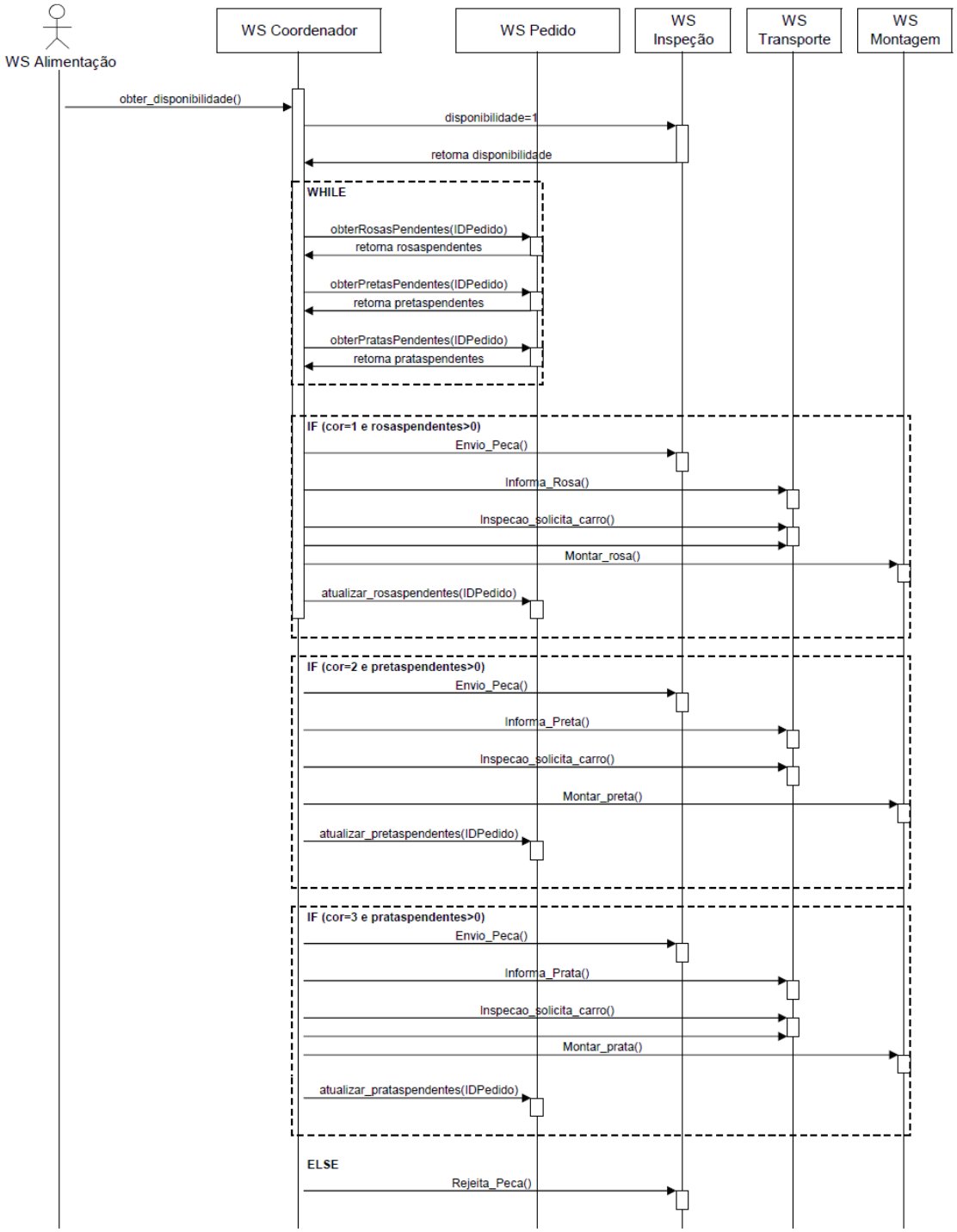



*Figura 52 - Diagrama de sequência do serviço Resposta_Alimentacao() do WS Coordenador.*

**void: Resposta_Inspecao()**

Subsistema de inspeção indica que finalizou a inspeção e requisita ao subsistema de transporte um *pallet* com “corpo” na estação 3 (subsistema de montagem).

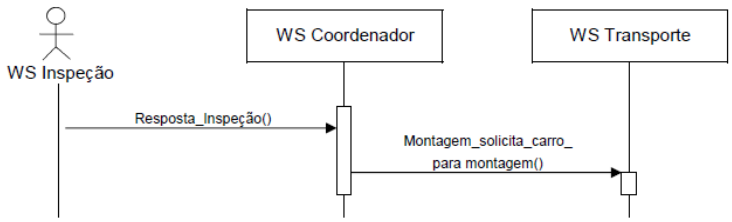



*Figura 53 - Diagrama de sequência do serviço Resposta_Inspecao() do WS Coordenador.*

**void: Resposta_Carrinho()**

Subsistema de Transporte informa que existe um *pallet* vazio na estação 2 (subsistema de inspeção) e requisita o envio do “corpo” para o *pallet*.

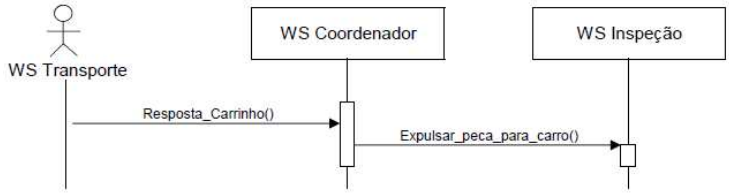



*Figura 54 - Diagrama de sequência do serviço Resposta_Carrinho() do WS Coordenador.*

**void: Resposta_Carrinho_inicio_montagem()**

Subsistema de Transporte informa que existe um *pallet* com “corpo” na estação 3 (subsistema de montagem) e requisita o envio do “corpo” para o *pallet*.

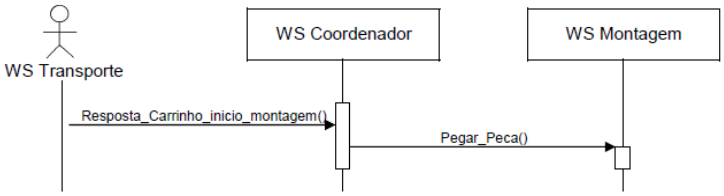



*Figura 55 - Diagrama de sequência do serviço Resposta_Carrinho_inicio_montagem() do WS Coordenador.*

**void: Resposta_Telecomando_de_inspecao()**

Subsistema de inspeção indica que finalizou a inspeção do “corpo” e verifica se e cor equivale ao pedido feito pelo cliente. Caso seja equivalente ao pedido, aceita o “corpo” e requisita um *pallet* vazio para a estação 2. Caso contrário, rejeita o “corpo”.

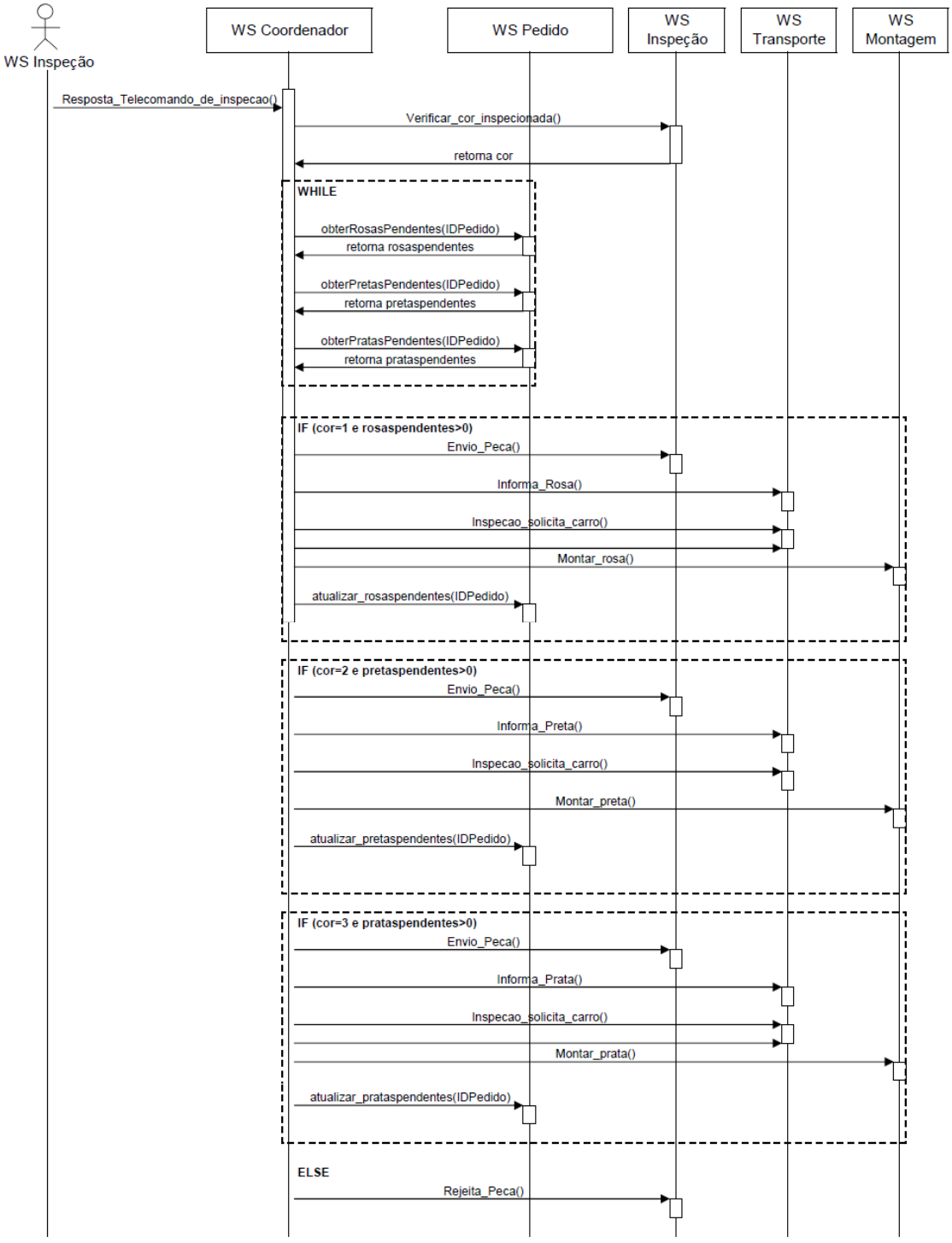



*Figura 56 - Diagrama de sequência do serviço Resposta_Telecomando_de_inspecao() do WS Coordenador.*

| **void: Requisita_Carrinho_fim_montagem()**<br>Subsistema de Montagem informa que existe um produto pronto e requisita um *pallet* vazio na estação 3 para a liberação do produto. |
|---|

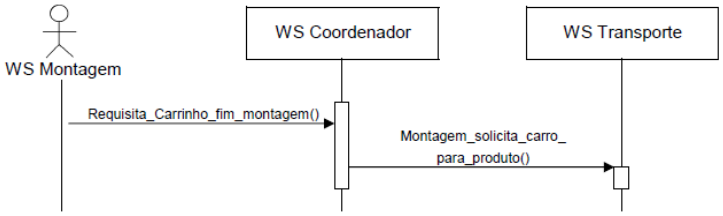



*Figura 57 - Diagrama de sequência do serviço Resposta_Carrinho_fim_montagem() do WS Coordenador.*

### 4.3.4. WS Pedido

O Banco de Dados Pedido é composto por “IDPedido”, “IDCliente”, “número de peças rosas do pedido”, “número de peças pretas do pedido”, “número de peças pratas do pedido”, “status do pedido”, “número de peças rosas pendentes”, “número de peças pretas pendentes” e “número de peças pratas pendentes”. O WS Pedido possui serviços que manipulam as informações do Banco de Dados Pedido. O WS Pedido possui serviços para a inclusão, autenticação (verifica se está cadastrado no banco de dados) e consulta dos pedidos.

| void: incluir (string id, string rosaspedido, string pretaspedido, string prataspedido) |
|---|
| int: autenticar (string id, string rosaspedido, string pretaspedido, string prataspedido) |
| int: obterRosasPedido (int id) |
| int: obterPretasPedido (int id) |
| int: obterPratasPedido (int id) |

São definidos três status para os pedidos: “pendente” (caso exista peças a serem montadas), “executando” (peças do pedido não completamente montadas) e “finalizado” (todas as peças do pedido montadas).

| object: obterStatus (int id) |
| --- |
| int obterRosasPendentes (int id) |
| int obterPretasPendentes (int id) |
| int obterPratasPendentes (int id) |

Implementou-se também, outros cinco serviços que permitem a atualização do status dos pedidos e a atualização do número de peças pendentes.

| void: atualizarStatusExe (int id) |
| --- |
| void: atualizarStatusCon (int id) |
| void: atualizar_rosaspendentes (int id) |
| void: atualizar_pretaspendentes (int id) |
| void: atualizar_prataspendentes (int id) |

### 4.3.5. WS Cliente

O Banco de Dados Cliente é composto por dois tipos de informações dos clientes: “login” e “senha”. O WS Cliente possui serviços que manipulam as informações do Banco de Dados Cliente. O WS Cliente possui serviços para a inclusão, autenticação (verifica se está cadastrado no banco de dados) e consulta dos dados cadastrados.

| void: incluir (string login, string senha) |
| --- |
| void: autenticar (string login, string senha) |
| dataset: obter(int id) |

### 4.3.6. Observador do Pedido

Os *web services* são módulos funcionais que disponibilizam os serviços na internet. O uso de suas funcionalidades depende de um cliente que consuma seus serviços, caso contrário, o WS fica inativo. Na integração do sistema flexível fez-se necessário a criação de um “Observador do Pedido”, um aplicativo que irá ficar ativo enquanto o sistema estiver operando.

A aplicação consulta freqüentemente o Banco de Dados Pedido e verifica se existem pedidos pendentes cadastrados. Caso exista pedido pendente, verifica a disponibilidade do sistema produtivo. Se disponível, a aplicação requisita o serviço Chama_Alimentacao_Peca() do WS Coordenador. A seguir, o diagrama de seqüência do Observador de Pedido:

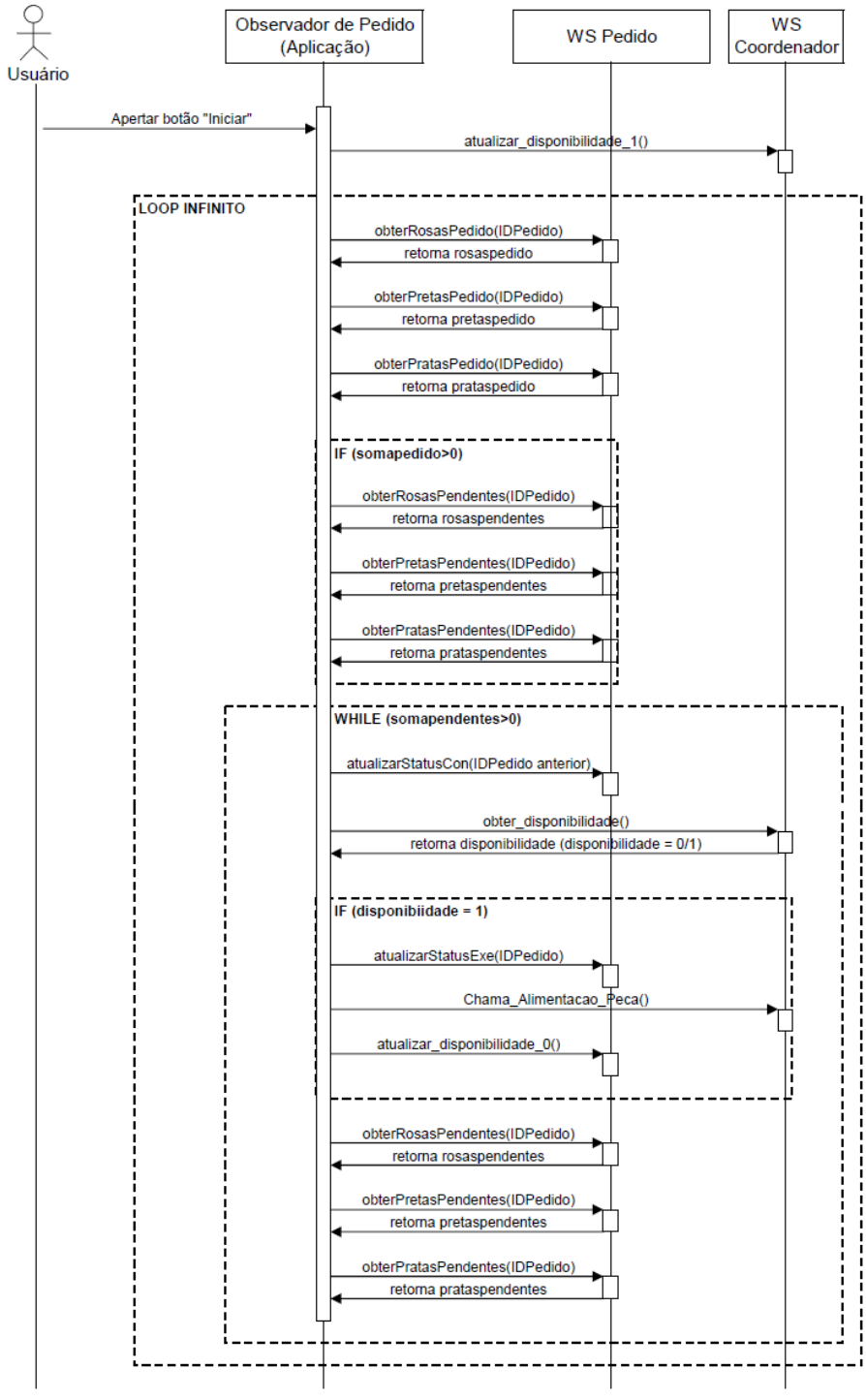



*Figura 58 - Diagrama de sequência do Observador de Pedido.*

O Aplicativo Windows (*Windows Application*) foi desenvolvido em Visual Studio e em Microsoft .NET Framework 4.0. Uma interface foi criada para o Observador de Pedido (Figura 59). Para iniciar o aplicativo, o usuário deve selecionar a opção “Iniciar”.

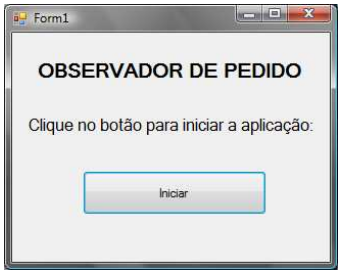



*Figura 59 - Observador de Pedido.*

## 4.4. IMPLEMENTAÇÃO

A estrutura do software de controle do subsistema de transporte (Figura 60) pode ser vista como um conjunto de camadas. Na camada inferior estão os programas que processam os sinais de entrada e saída (I/O) gerados pelos dispositivos de atuação e detecção. Logo acima desta camada está o programa que executa as funções de controle local, que é realizado pelo controlador (SIMATIC S7-300). Na camada acima está o programa supervisório local que gerencia as tarefas do subsistema de transporte e do subsistema de montagem. Este programa supervisório é executado em um PC ligado a internet. Acima do supervisório temos os WSs do sistema produtivo e do teleoperador que são acessados através da internet.

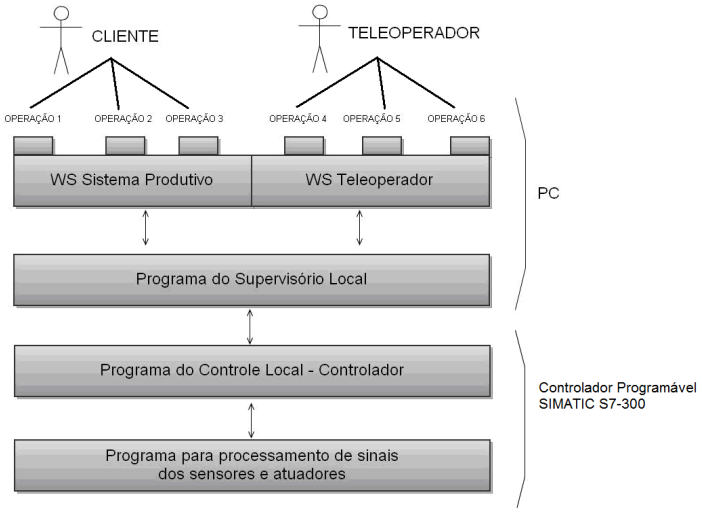



*Figura 60 - Estrutura do Software de controle do sistema produtivo*

### 4.4.1. Descrição das funções de controle

A interface homem-máquina (IHM) se refere a uma implementação de dispositivos de monitoração e dispositivos de comando disponibilizados pelo SP para o usuário (cliente ou teleoperador). Em outras palavras, esta interface é implementada num computador de modo que o usuário tem disponibilizado na tela do computador, ícones que indicam o estado de elementos como lâmpadas e botões *switches,* sendo que o teclado do computador é usado para os comandos.

Esta interface no contexto do SP disperso deve estar disponível tanto para "operadores" que podem estar em locais remotos e distintos do local onde estão as máquinas do SP, como para "clientes" e "supervisores" que solicitam a manufatura de produtos e desejam acompanhar o processo produtivo de seus pedidos. Estes "clientes", em geral, estão em locais distintos das instalações do SP e podem apenas acompanhar os processos produtivos. Os "supervisores", que também estão, em geral, em locais remotos, além de acompanhar os processos produtivos devem, em situações específicas de ocorrência de imprevistos, terem condições de interferir no processo produtivo.

Os requisitos do sistema envolvem a definição das condições ou capacidades que devem ser atendidas por um sistema para implementar funcionalidades e processos requeridos pelos usuários. Estes requisitos podem ser classificados em: (a) requisitos funcionais, baseados nas funcionalidades a serem implementadas no sistema; (b) requisitos operacionais, relacionadas aos processamentos a serem executados, desempenho e localização do sistema; (c) requisitos de contingência, como tarefas alternativas para o caso de indisponibilidade do sistema; (d) requisitos técnicos, baseados nas restrições quanto à arquitetura do sistema e às ferramentas e linguagens implementadas; e (e) requisitos não-funcionais (FERNANDES, 2005). Quanto aos requisitos do sistema considerado neste projeto, a arquitetura, conhecida como SOA, deve garantir as funcionalidades desejadas. Na Tabela 4, listam-se alguns requisitos não funcionais que devem ser considerados:

*Tabela 4 - Descrição de alguns requisitos não-funcionais (adaptado de FERNANDES, 2005)*

| | |
|---|---|
| **Reusabilidade** | Uso de módulos na implementação do sistema, que permite a reutilização destes em outras aplicações. |
| **Multimodalidade** | Uso de diversos canais de comunicação, como visual, tátil, auditiva e motora, para apresentação de informação ou interação com o usuário. |

| | |
|---|---|
| **Usabilidade** | Refere-se ao grau de facilidade oferecido para que um usuário aprenda a operar, fornecer entradas e interpretar saídas de um componente ou do sistema como um todo. |
| **Extensibilidade** | Capacidade de ampliar o sistema, pela incorporação de novas funcionalidades, pelo aumento da capacidade de armazenamento, etc. |

A fim de especificar os requisitos do sistema, considera-se a utilização do PFS e da UML. A seguir, descreve-se a modelagem conceitual e funcional do subsistema de transporte.

Para especificar o subsistema de transporte, fez-se necessária uma descrição explícita dos processos produtivos deste subsistema. O subsistema de transporte possui quatro posições distintas para a parada dos *pallets* (estação 1, estação 2, estação 3 e estação 4), e com relação a cada uma das estações, o *pallet* pode ser visto como se estivesse em três situações distintas: “antes da estação” (A.E.), “na estação” (N.E.) e “depois da estação” (D.E.). Estas situações são ilustradas na Figura 61.

No estado inicial do subsistema, assume-se que os *pallets* ficam acumulados em fila antes da estação 1. Após uma peça “corpo” sair do subsistema de alimentação e ser aprovada pelo subsistema de inspeção, o subsistema de inspeção envia um sinal para o supervisor geral do SP disperso que, de acordo com seu plano de produção, pode enviar um pedido de serviço para o subsistema de transporte, requisitando o transporte do “corpo” para o subsistema de montagem. Dessa forma, um *pallet* deve atravessar a estação 1, recolher o “corpo” na estação 2 e o transportar para a estação 3, onde o *pallet* deve aguardar a retirada do “corpo”. Quando o subsistema de montagem envia um sinal para o supervisor geral indicando que já completou a montagem de um produto, este pode enviar um pedido de serviço ao subsistema de transporte para que um *pallet* vazio seja enviado à estação 3. Quando o *pallet* já está na estação 3, o subsistema de montagem pode enviar o produto para a estação 4 (de armazenagem).

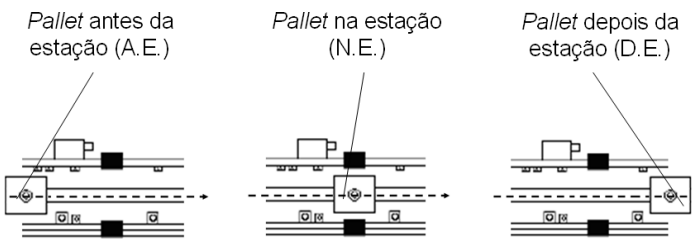



*Figura 61 - As três situações de um pallet em relação a cada estação de parada*

Dessa forma considera-se que o subsistema de transporte é responsável pela execução de dois serviços: (a) transportar a peça “corpo” do subsistema de inspeção até o subsistema de montagem e (b) retirada do produto montado do subsistema de montagem.

No primeiro caso, como já mencionado, o responsável pelo envio da ordem de serviço é um supervisor que depois de comprovar a aprovação do “corpo” pelo subsistema de inspeção, envia uma ordem de serviço para o subsistema de transporte. Em seguida, o subsistema de transporte verifica a disponibilidade de *pallets* antes da estação 1, e caso essa disponibilidade se confirme e não exista um *pallet* na estação 1, um dos *pallets* disponíveis é enviado para a estação 2. Caso já exista um *pallet na estação 1*, nenhum *pallet* é transportado até a estação 1. Quando existe um *pallet* na estação 1, o subsistema verifica se a estação 2 está ocupada com um *pallet* ou se está livre. Caso a estação 2 esteja ocupada, o *pallet na estação 1* deverá ser mantido nesta estação. Caso contrário, o *pallet* é enviado para a estação 2. Quando o *pallet* está na estação 2, o subsistema verifica se o *pallet* já recebeu a peça tipo “corpo”. Depois dessa situação ser confirmada, o subsistema verifica se a estação 3 está ocupada com um *pallet* ou se está livre. Caso a estação 3 esteja ocupada, o *pallet* na estação2 deverá ser mantido nesta estação. Caso contrário, o *pallet* é enviado para a estação 3. Na estação 3, o *pallet* aguarda a retirada do “corpo” pelo subsistema de montagem, e agora em estado vazio espera pelo novo comando do supervisor.

No segundo caso, o subsistema de montagem, requisita ao supervisor a retirada do produto (montado) final. Se já existe um pallet vazio na estação 3, a este *pallet* deve ser atribuído o serviço de retirada do produto (montado) final. Verifica-se assim se o produto final já está carregado no *pallet* que está a estação 3. Caso isso se confirme, o subsistema de transporte leva o *pallet* até a estação 4, espera que o produto final seja retirado e vai para a fila de *pallets* antes da estação 1.

### 4.4.2. Production Flow Schema

Na modelagem conceitual, considera-se o PFS (*Production Flow Schema*). A Figura 62 ilustra o PFS do subsistema de transporte. Nesse grafo, o supervisor é responsável pelo controle da movimentação dos *pallets* pelas estações 1, 2, 3 e 4.

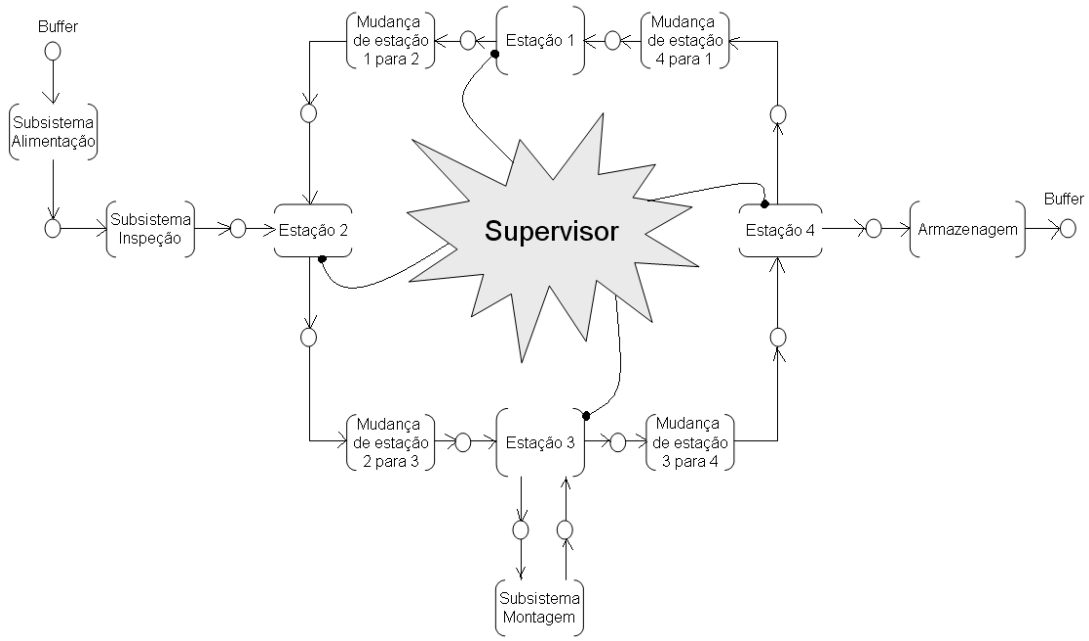



*Figura 62 - PFS do subsistema de transporte*

O PFS indica os estados e as atividades executadas sobre as peças no sistema produtivo. Os “corpos” estão armazenados em um buffer inicial. Após a alimentação, passam por um processo de inspeção de altura e cor. Em seguida, esses “corpos” são colocados sobre os *pallets* e atravessam as estações do subsistema de transporte (na ordem, estação 2, estação 3, estação 4 e estação 5), até ser requisitado pelo subsistema de montagem, na estação 3. Por fim, após a montagem, o produto é de novo colocado sobre um *pallet* e enviado para armazenagem, na estação 4.

### 4.4.3. Configuração no SIMATIC Manager

O ambiente de desenvolvimento para a lógica de controle é o SIMATIC Manager. Após a criação de um projeto, deve-se configurar o hardware, software e o protocolo de comunicação entre o PC e o CLP. A Figura 63 mostra os componentes configurados no projeto “Transporte”:

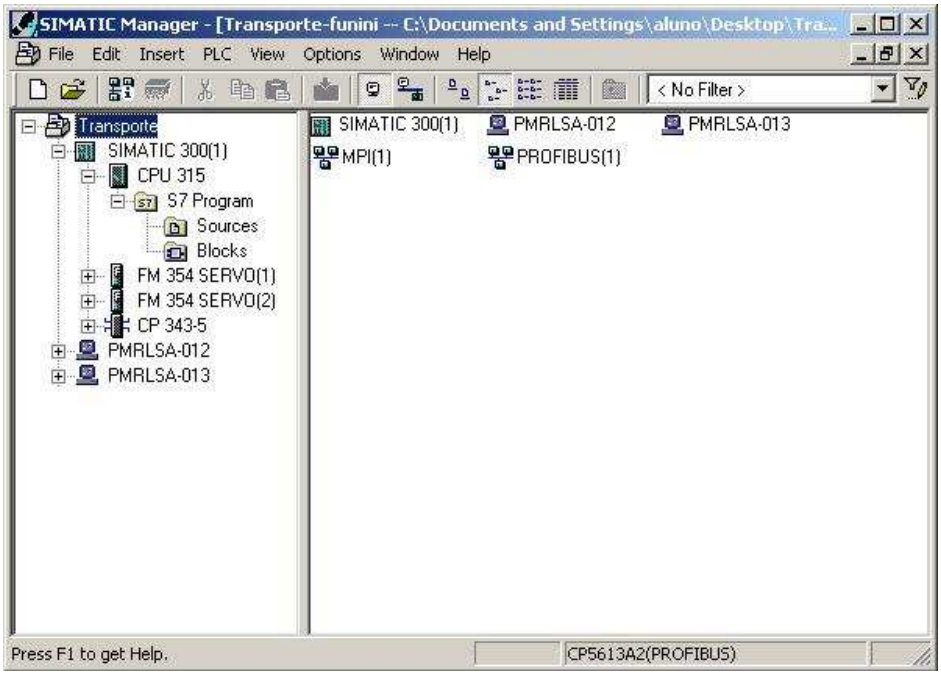



*Figura 63 - Configuração do SIMATIC Manager.*

Faz-se necessário destacar que dois servos motores são utilizados no subsistema de montagem para movimentação de um braço mecânico. No sistema produtivo, o subsistema de transporte e o subsistema de montagem estão conectados a uma mesma rede PROFIBUS. Com o intuito de tornar o projeto flexível para a programação dos dois subsistemas, configuraram-se os módulos FM 354 (drives dos servos motores) no projeto do subsistema de transporte, além da CPU 315-2 DP e o módulo CP353.

Uma vez adicionado os componentes hardwares ao projeto é necessário a configuração deles. A Figura 64 mostra o mapeamento dos módulos de hardware do projeto:

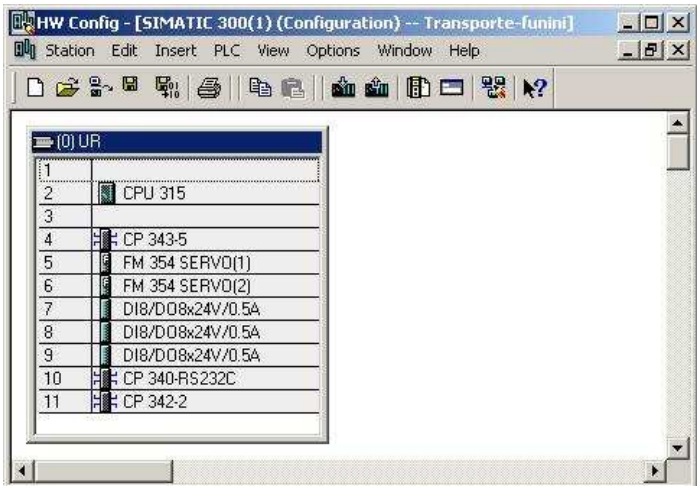



*Figura 64 - Configuração de hardware.*

A configuração de cada módulo pode ser feita individualmente. A Figura 65 ilustra a janela de configuração da CPU 315-2 DP. Diferentes propriedades são configuradas, tais como interrupções de hardware, sincronização de *clock* e tempo de monitoramento de *scan cycle*:

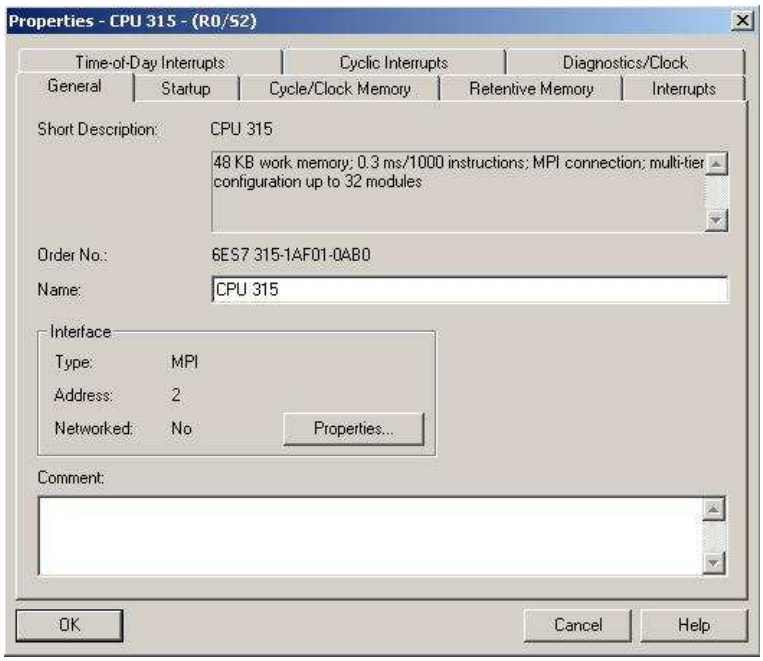



*Figura 65 - Janela de configuração da CPU 315-2 DP*

Adicionalmente, configura-se a rede de comunicação entre o PC e o CLP. O protocolo utilizado é o PROFIBUS-DP. A Figura 66 indica a configuração:

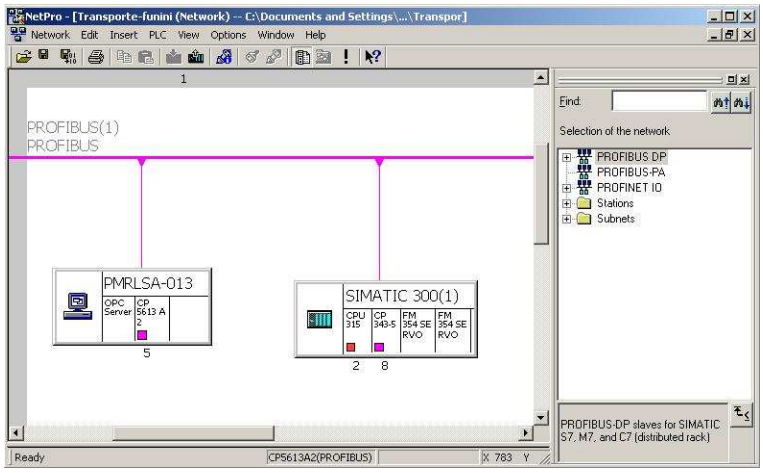



*Figura 66 - Rede de comunicação entre o PC e o CLP*

A Figura mostra o PC, nomeado como “PMRLSA-013” e o CLP Simatic S7-300 na rede PROFIBUS. Na configuração do PC considerou-se o servidor OPC e o cartão CP5613-A2, responsável pela comunicação do computador com a rede PROFIBUS. Na configuração do CLP considerou-se a CPU315, os módulos CP 343-5 e FM354, descritos anteriormente.

### 4.4.4. Lógica de controle no SIMATIC Manager

A programação das funções de controle no controlador (SIMATIC S7-300) é realizada através do pacote de softwares STEP 7. No caso foi adotada a linguagem Ladder, uma linguagem gráfica baseada na lógica de relês.

Resumidamente, o projeto do Step 7 é dividido em “blocos”. Os “blocos” são parte do programa e se distinguem pela sua função, estrutura e propósito. O STEP7 permite a criação dos seguintes tipos de “Blocos”: (a) blocos lógicos (FB, FC, OB, SFB, SFC) e (b) bloco de dados (DB, SDB). A Figura 67 ilustra os blocos lógicos e blocos de dados no *SIMATIC Manager*.


*Figura 67 - Blocos do STEP7*

Como exemplo, para a identificação dos *pallets*, utilizou-se um DB (ou “*Data Base*”) que é um tipo de bloco de dados. Na Figura 68, identifica-se o endereço da memória, ID dos *pallets* e o registro em hexadecimal lido pelo sensor.

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | ID_CAR1 | DWORD | DW#16#A955550 | |
| +4.0 | ID_CAR2 | DWORD | DW#16#A955718 | |
| +8.0 | ID_CAR3 | DWORD | DW#16#B4E0892 | |
| +12.0 | ID_CAR4 | DWORD | DW#16#B257549 | |
| +16.0 | ID_CAR5 | DWORD | DW#16#B2F6875 | |
| =20.0 | | END_STRUCT | | |

*Figura 68 - Data Base para identificação do pallet*

Como dito anteriormente, utilizou-se a linguagem Ladder para o desenvolvimento do programa de controle local. O programa completo encontra-se em anexo.

### 4.4.5. Teste do programa através do SIMATIC Manager

O aplicativo Simatic Manager possui uma funcionalidade chamada Tabela de Variáveis (VAT – do inglês *VAriable Table*). Na VAT é possível mapear as variáveis do programa e monitorar seu estado, sendo possível modificar o estado de variáveis de entrada. Durante a realização dos testes a VAT foi utilizada para verificar a consistência das saídas de acordo com as entradas e o estado do subsistema.

No subsistema de transporte são utilizados sensores que identificam a posição dos *pallets* em cada estação, que se comunicam através da rede ASI. Os sensores de posição representam saídas que devem ser lidas pelo sistema e serão importantes para o controle dos pistões que regulam a lógica de entrada e saída das estações.

Os testes iniciais foram realizados individualmente em casa estação, de forma a regular a seqüência de acionamentos dos pistões para a entrada e saída de cada estação. Partindo da lógica de recirculação dos carrinhos o próximo passo foi testar o funcionamento nesta situação variando as solicitações nas diferentes estações.

ESTAÇÃO 2 -- JOSE_REIN_MINICIM_TRANSP\SIMATIC 300(1)\CPU 315\S7 Program(2)

| | Address | Symbol | Display format | Status value | Modify value |
|---|---|---|---|---|---|
| 1 | M 80.0 | "Req_Inspecao_Est2" | BOOL | | false |
| 2 | MW 180 | "COR_PECA" | DEC | | |
| 3 | MW 150 | "Cor_atualizacao_Est2" | DEC | | |
| 4 | M 80.2 | "Saida_autorizada_Est2" | BOOL | | |
| 5 | M 3.1 | "Hab_Saida_Est2" | BOOL | | |
| 6 | M 1.1 | "Hab_Trav_Est2" | BOOL | | |
| 7 | M 2.1 | "Hab_Destrav_Est2" | BOOL | | |
| 8 | M 80.3 | "Resp_req_car" | BOOL | | |
| 9 | M 80.4 | "Status_carrinho_travado" | BOOL | | |
| 10 | M 80.5 | "Status_car_destravado" | BOOL | | |
| 11 | M 80.6 | "Status_car_saindo" | BOOL | | |
| 12 | DB5.DBD 0 | "ID_ESTACAO".ID_EST1 | HEX | | |
| 13 | DB5.DBD 4 | "ID_ESTACAO".ID_EST2 | HEX | | |
| 14 | DB5.DBD 8 | "ID_ESTACAO".ID_EST3 | HEX | | |
| 15 | DB5.DBD 12 | "ID_ESTACAO".ID_EST4 | HEX | | |
| 16 | DB6.DBD 0 | "ID_CARROS".ID_CAR1 | HEX | | |
| 17 | DB6.DBD 4 | "ID_CARROS".ID_CAR2 | HEX | | |
| 18 | DB6.DBD 8 | "ID_CARROS".ID_CAR3 | HEX | | |
| 19 | DB6.DBD 12 | "ID_CARROS".ID_CAR4 | HEX | | |
| 20 | DB7.DBW 0 | "ID_COR".COR_ID1 | HEX | | |
| 21 | DB7.DBW 2 | "ID_COR".COR_ID2 | HEX | | |
| 22 | DB7.DBW 4 | "ID_COR".COR_ID3 | HEX | | |
| 23 | DB7.DBW 6 | "ID_COR".COR_ID4 | HEX | | |
| 24 | DB7.DBW 8 | "ID_COR".COR_ID5 | HEX | | |
| 25 | M 11.0 | "S_Cil_Estend_2" | BOOL | | |
| 26 | | | | | |

*Figura 69 - Tabela de Variáveis para a estação 3*

ESTAÇÃO 4 -- JOSE_REIN_MINICIM_TRANSP\SIMATIC 300(1)\CPU 315\S7 Program(2)

| | Address | Symbol | Display format | Status value | Modify value |
|---|---|---|---|---|---|
| 1 | // IDENTIFICAÇÃO DE CARRO NA ESTAÇÃO 4 | | | | |
| 2 | M 13.4 | "S_Est4" | BOOL | | |
| 3 | DB5.DBD 12 | "ID_ESTACAO".ID_EST4 | HEX | | DW#16#0B4E0892 |
| 4 | M 91.0 | "id_carro1_st4" | BOOL | | |
| 5 | M 91.1 | "id_carro2_st4" | BOOL | | |
| 6 | M 91.2 | "id_carro3_st4" | BOOL | | |
| 7 | M 91.3 | "id_carro4_st4" | BOOL | | |
| 8 | M 91.4 | "id_carro5_st4" | BOOL | | |
| 9 | // BANCO DE DADOS DE PRODUTO ACABADO | | | | |
| 10 | DB8.DBX 0.0 | "ID_PRODUTO".ID1_PRODUTO_PRONTO | BOOL | | |
| 11 | DB8.DBX 0.1 | "ID_PRODUTO".ID2_PRODUTO_PRONTO | BOOL | | |
| 12 | DB8.DBX 0.2 | "ID_PRODUTO".ID3_PRODUTO_PRONTO | BOOL | | |
| 13 | DB8.DBX 0.3 | "ID_PRODUTO".ID4_PRODUTO_PRONTO | BOOL | | |
| 14 | DB8.DBX 0.4 | "ID_PRODUTO".ID5_PRODUTO_PRONTO | BOOL | | |
| 15 | // SENSORES DE TRAVA | | | | |
| 16 | M 13.1 | "S_Cil_Recuado_4" | BOOL | | |
| 17 | M 13.0 | "S_Cil_Estend_4" | BOOL | | |
| 18 | // COMANDOS DE SAÍDA | | | | |
| 19 | M 1.3 | "Hab_Trav_Est4" | BOOL | | |
| 20 | M 2.3 | "Hab_Destrav_Est4" | BOOL | | |
| 21 | M 3.3 | "Hab_Saida_Est4" | BOOL | | |
| 22 | // STATUS DA ESTAÇÃO 4 | | | | |
| 23 | M 13.5 | "S_Saida-Est4" | BOOL | | |
| 24 | M 82.0 | "Carro_Travado_St4" | BOOL | | |
| 25 | M 82.1 | "Carro_Destravado_St4" | BOOL | | |
| 26 | M 82.2 | "Carro_Saindo_St4" | BOOL | | |
| 27 | | | | | |

*Figura 70 - Tabela de Variáveis para a estação 4*

### 4.4.6. Teste do programa através do SP

Para o teste do programa através do sistema produtivo disperso é necessário a habilitação do OPC Server e o cartão de comunicação CP5613-A2 no modo “RUN”. A Figura 71 mostra o Station Configuration Editor que possibilita essa configuração:

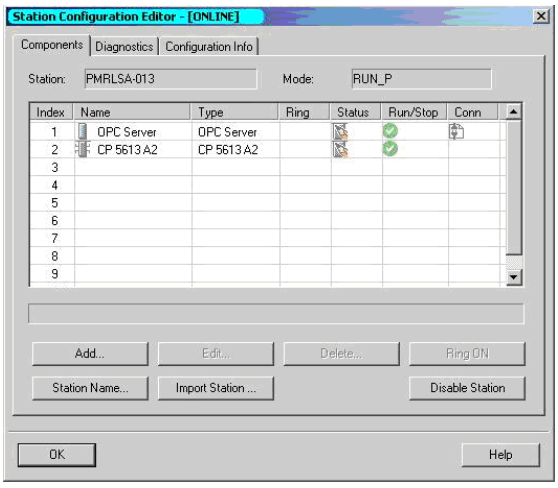



Figura 71 - Station Configuration Editor

### 4.4.7. Mapeamento no OPC Server

Após a conclusão dos testes do subsistema de transporte e a habilitação para modo “RUN” do OPC Server no Station Configuration Editor, é feito o mapeamento das funções de controle. Para isso, utiliza-se o OPC Scout, um aplicativo da Siemens. Terminado o mapeamento das funções de controle no OPC Server, foi possível a criação da WS que se utiliza dessas lógicas em seus serviços. A WS criada para o subsistema de transporte é descrita no item 50.

## 4.5. INTERFACES DO CLIENTE

Feito a integração dos subsistemas, desenvolveu-se a tela de interface do cliente. Define-se como tela de interface, o *web site* que permite o acesso do cliente, inclusão e acompanhamento dos pedidos de produtos. O fluxograma dos estágios de acesso do cliente no *web site* foi modelado como ilustra a Figura 72.

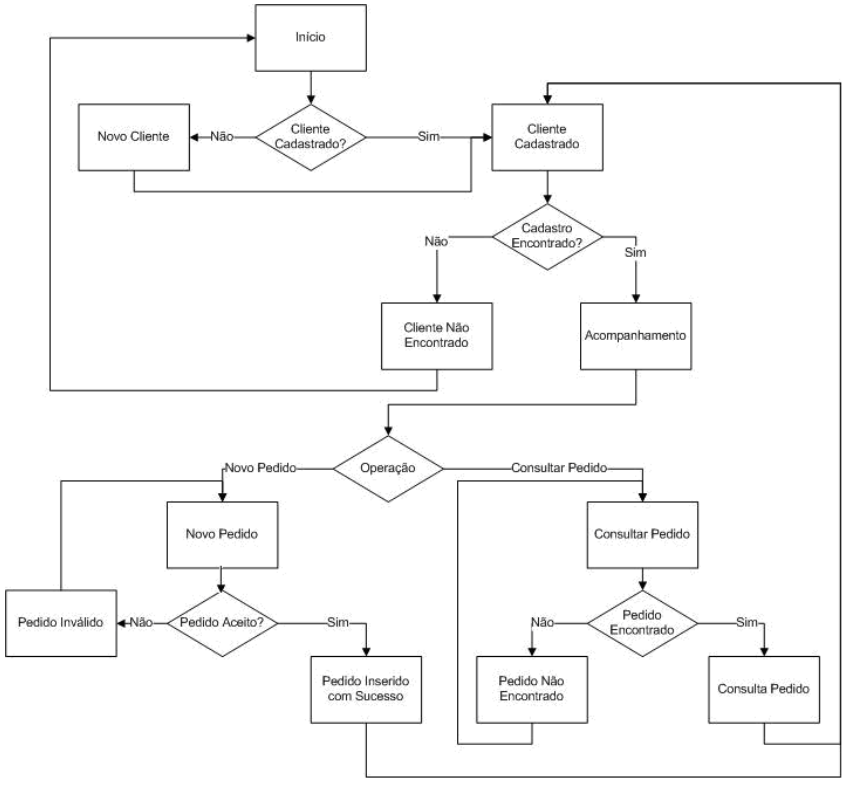



*Figura 72 Fluxograma de acesso*

O *web site* foi desenvolvido na linguagem C# e optou-se pela plataforma .NET do software Microsoft Visual Studio 2010. Um banco de dados foi criado para a manipulação

dos dados do cliente e dos seus respectivos pedidos no Microsoft Office Access 2003. A seguir, descreve-se o funcionamento do *web site*.

### 4.5.1. Página principal

Na página principal do *web site* (Figura 73), o cliente insere o seu *login* e senha e confirma os dados clicando no botão “Entrar”.

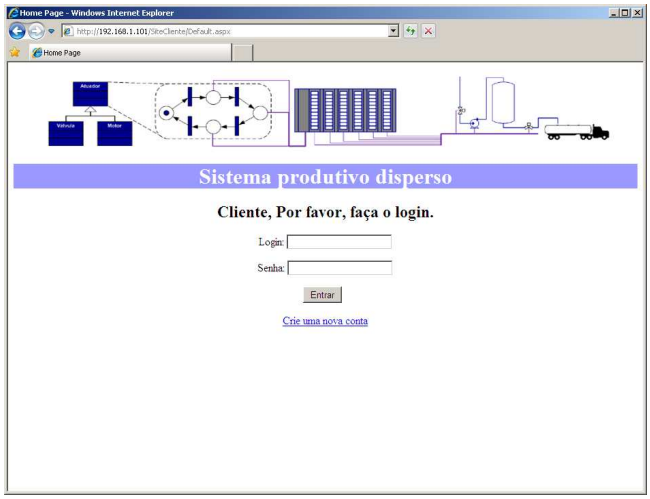



*Figura 73 - Página principal*

Caso o cliente tenha colocado alguma informação de acesso errada ou não possua cadastro no sistema, surgirá a mensagem: “Cliente não encontrado. Tente novamente ou realize seu cadastro” (na Figura 74). Para realizar um novo cadastro, o cliente deve selecionar a opção “Crie uma nova conta”.


*Figura 74 - Página Principal – Cliente não encontrado.*

### 4.5.2. Novo cadastro

Na página destinada à realização de novos cadastros (Figura 75), o cliente deve definir um *login* e uma senha. O cadastro é confirmado com a opção “Inserir”.

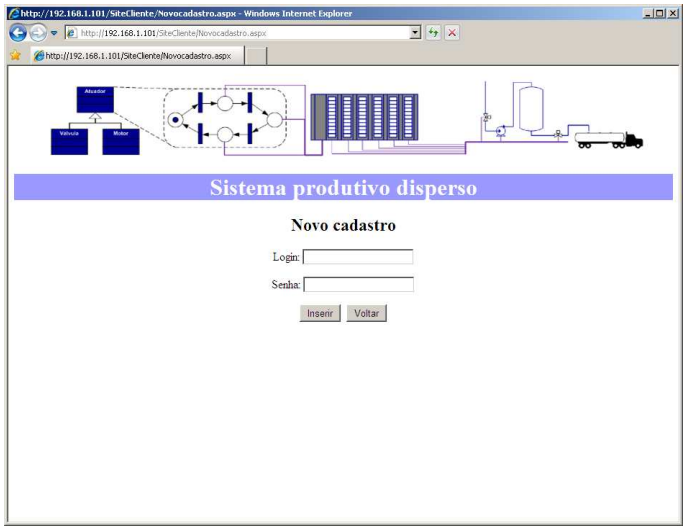



*Figura 75 - Novo Cadastro.*

Após a inserção do novo cadastro, o Cliente recebe o número de seu cadastro (Figura 76). Este número será utilizado para o cadastro do pedido. Em seguida, o cliente pode retornar à página principal com a opção “Voltar”.


*Figura 76 - Novo Cadastro – Cadastro realizado.*

### 4.5.3. Página do cliente

Com a confirmação do *login* e a senha, o cliente é direcionado para a página na qual deverá fazer a opção pela inclusão de um novo pedido ou a consulta dos pedidos já cadastrados, através dos botões “Novo Pedido” e “Consultar Pedido” (Figura 77).


*Figura 77 - Página do cliente.*

### 4.5.4. Cadastro de pedido

Ao clicar em “Novo Pedido”, o cliente é direcionado para à pagina de cadastro de pedidos (Figura 78). O cliente deverá escrever o número de produtos rosa, preta e/ou prata que deseja e, adicionalmente, informar o número do cadastro (fornecido no cadastro do *login* e senha).

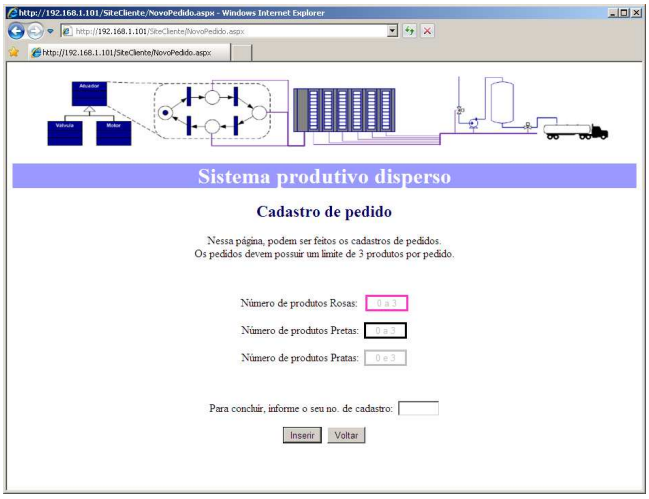



*Figura 78 - Cadastro de pedido.*

Após a inserção do novo pedido (Figura 79), uma mensagem informará o número do pedido. Este número será utilizado para o acompanhamento do pedido.

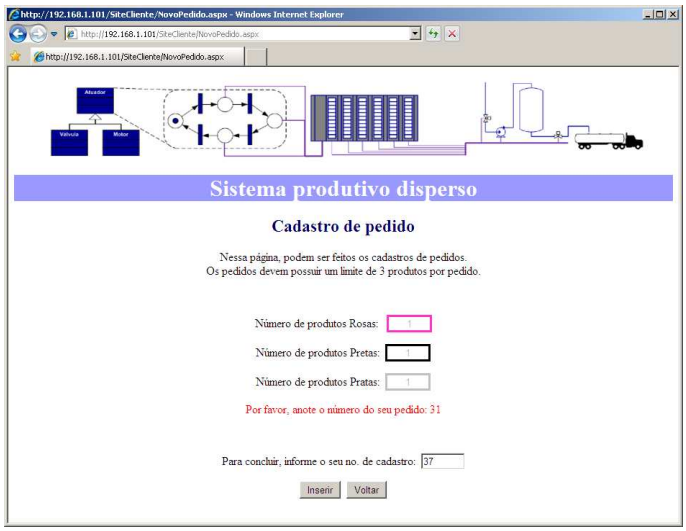



*Figura 79 - Cadastro de pedido – Cadastro realizado.*

### 4.5.5. Acompanhamento de pedido

Adicionalmente, caso o cliente após fazer o *login*, clique em “Consultar Pedido”, o cliente será direcionado a pagina de acompanhamento de pedido (Figura 80). O cliente deverá escrever o número do pedido e selecionar “Ir”.

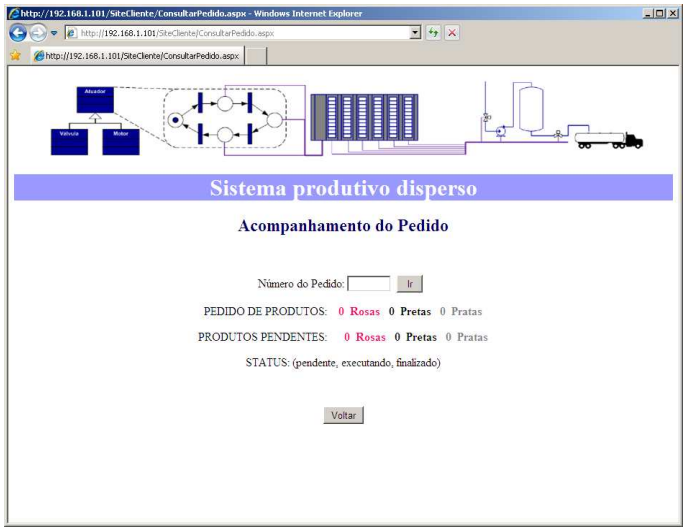



*Figura 80 - Acompanhamento do pedido.*

No caso do número do pedido indicado pelo cliente não se refira a nenhum pedido cadastrado no banco de dados, uma mensagem surgirá: “Pedido não encontrado, tente novamente”.


*Figura 81 - Acompanhamento do pedido – Pedido inexistente.*

Caso o número de pedido se refira a um pedido cadastrado, inicialmente, antes de iniciar a produção nos subsistemas, o status do pedido indicará “pendente” (Figura 82). Com o início da produção, o status recebe “executando”; e após a produção de todos os produtos, indicará “concluído”.

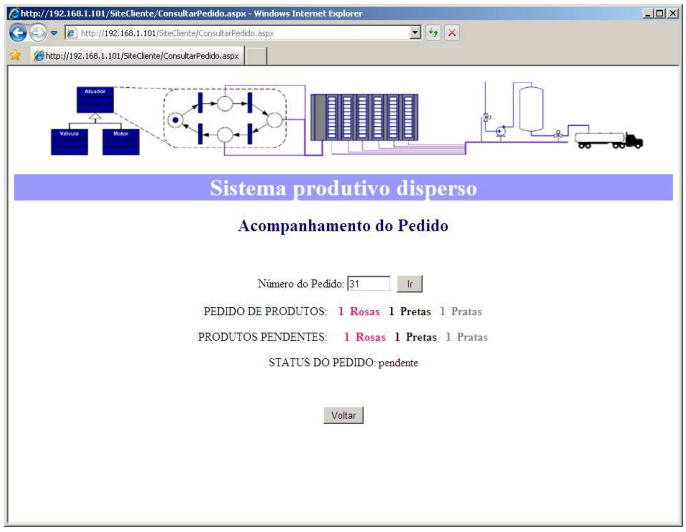



*Figura 82 - Acompanhamento do pedido – Status: pendente.*

# 5. CONCLUSÃO

O trabalho consistiu na implementação da interface homem-computador para as funções de monitoração e teleoperação do subsistema de transporte e na integração desse subsistema com os outros 3 subsistemas do sistema de manufatura automatizado instalado na EPUSP. A integração dos subsistemas foi feito através da criação de WSs.

A lógica do subsistema de transporte apresenta grande grau de complexidade devido à necessidade de coordenar o funcionamento do sistema em função de todas as estações. Por sua vez a aplicação de WSs diminui a complexidade de integrar este aos demais subsistemas devido à sua característica de modularidade, permitindo testes isolados e correções de forma mais simples.

A criação de camadas permite a integração entre as diferentes subestações, que representam plantas dispersas geograficamente. A primeira camada está relacionada ao CP e é responsável por garantir o funcionamento das funcionalidades de cada fábrica ou célula. A segunda camada é formada pelos WSs e permite a integração destas funcionalidades bem como sua disponibilização na internet sob a forma de serviços. Por fim a terceira camada é responsável por garantir a interface com o cliente e o teleoperador, através de páginas Web.

Enquanto as camadas inferiores garantem o funcionamento das funcionalidades de cada fábrica a camada superior, dos WSs, permite a integração destas funcionalidades através da publicação sob a forma de serviços.

Dentre as vantagens da implementação em camadas, temos: a padronização do código, a facilidade para encontrar problemas e a facilidade para dividir a programação em partes e ser trabalhada simultaneamente.

Devido ao enfoque do trabalho as partes voltadas ao cliente foram pouco desenvolvidas, porém o tipo de sistema estudado levanta uma gama de questões relevantes a serem estudadas, podendo-se destacar a interface do cliente e a segurança do sistema na colocação.

Os WSs se mostram um ótimo modo de criar e integrar serviços a través da internet. Sua aplicação é ampla, podendo ser aplicada a diferentes tipos de serviços disponibilizados ou não na rede, criando um servidor local. É possível verificar a facilidade de integração entre os diferentes subsistemas, suportando a hipótese de que os WS são uma ferramenta eficaz para aplicação em SPs dispersos, bem como diferentes sistemas que exijam a integração de diferentes serviços.

Do ponto de vista da aplicabilidade é necessário levar em consideração a tecnologia de internet disponível no local das plantas, sobretudo no que tange a confiabilidade. A fim de garantir bons resultados a implementação deste projeto se insere no contexto do programa TIDIA-Kyatera, que oferece a infra-estrutura mais adequada, baseada em uma rede de fibra-ótica de alta velocidade. Devido à variabilidade na qualidade das redes ao redor do mundo seria necessário um estudo prévio de acordo com as áreas abrangidas.

Para a criação do coordenador foi estudada a aplicação do *Windows Workflow Foundation* (WWF), um ambiente da plataforma .NET que possibilita a programação através de fluxogramas no software Microsoft Visual Studio. Seu funcionamento se baseia em fluxogramas que podem coordenar as chamadas aos WSs e representa outra forma de implementar o projeto aqui proposto. Dentre as vantagens do uso dessa aplicação, citam-se a facilidade do entendimento sobre o processo através de uma representação visual, e conseqüentemente, uma maior possibilidade de otimização do processo.

# 6. REFERÊNCIAS BIBLIOGRAFICAS

ALTUS. Nota de aplicação P35. Disponível em: <http://www.altus.com.br/ftp/Public/Portugues/Notas%20de%20Aplicacao/NAP035%20-%20Exemplo%20Utilizacao%20Gateway%20PROFIBUS-DP_AS-I%20e%20Modulo%20Ponto%20AS-I%20IP67/NAP035.PDF>. Acesso em Setembro de 2008.

BARROS, Ettore Apolônio de. Sistemas Dinâmicos para Mecatrônica. São Paulo: EPUSP-PMR, 2008. Notas de aula para disciplina de graduação do Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, PMR2320 – Sistemas Dinâmicos para Mecatrônica.

CHENG, H. K.; TANG, Q. C.; ZHAO, J. L. Web services and service-oriented application provisioning: an analytical study of application service strategies. IEEE: Transactions on engineering management, n. 53, n. 4, p. 520-533, nov. 2006.

CHWIF, L. Utilizando simulação de eventos discretos em projetos de sistemas automatizados de manufatura. In: CONGRESSO INTERNACIONAL DE AUTOMAÇÃO, 10. 2002, São Paulo. Anais: CONAI, 2002

CURY, J. E. R. Teoria de Controle Supervisório de Sistemas Discretos. Apresentado no V Simpósio Brasileiro de Automação Inteligente, Canela, RS, Brasil, Novembro de 2001. Dispoível em: <http://www.faatesp.edu.br/publicacoes/controle.pdf>

FAPESP. Linha de Fomento à Pesquisa para Inovação Tecnológica. Disponível em: <http://www.fapesp.br/materia/52/pesquisa-para-inovacao/linha-de-fomento-a-pesquisa-para-inovacao-tecnologica.htm>. Acesso em: 19 fev. 2010.

FERNANDES, D. B. Análise de sistemas orientada ao sucesso: por que os projetos atrasam. São Paulo: Editora Ciência Moderna, 2005.

FIELDBUS Foundation – History. Disponível em: <http://www.fieldbus.org/index.php?option=com_content&task=view&id=136&Itemid=307>. Acesso em: 20 nov. 2010.

KANEKO, A. M. Desenvolvimento de uma interface gráfica para supervisão de um sistema produtivo teleoperado. Trabalho de formatura. Escola Politécnica da Universidade de São Paulo. 2008.

KANO, C.H. et al. Framework para sistema colaborativo de tele-operação de sistemas produtivos. In: SIMPÓSIO BRASIL-JAPÃO 2009 – Encontro anual da SBPN, 18, 2009, São Paulo. Anais: SBPN, 2009. p. 96 – 97.

KYATERA. Fiber to the lab – Viabilizando a pesquisa colaborativa. Disponível em: <http://kyatera.incubadora.fapesp.br/portal/kyatera/view?set_language=pt-br>. Acesso em: 19 fev. 2010.

LING, Z., CHEN, W., YU, J., Research and Implementation of OPC Server Based on Data Access Specification, Apresentado no 5º Congresso Mundial de Controle e Automação Inteligente, Hangzhou, P.R. China, Junho de 2004. Disponível em <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1340887>

LSA. Laboratório de pesquisa do PMR – Laboratório de Sistemas de Automação. Disponível em: <http://www.pmr.poli.usp.br/lsa/>. Acesso em: 23 fev. 2010.

MAIBASHI, G. Y.; SAITO, M. M. Sistema de teleoperação de manufatura via internet. Trabalho de formatura. Escola Politécnica da Universidade de São Paulo, 2009.

MARCHENTA, M. M. L. Controle de manufatura via internet. Trabalho de formatura. Escola Politécnica da Universidade de São Paulo, 2009.

MELO, J. I. G.; Junqueira, F.; Morales, R.; Miyagi, P.E. A procedure for modeling and analysis of service-oriented and distributed productive system. In: CASE, IEEE, Washington, 2008

MIYAGI, P. E., Controle Programável - Fundamentos do Controle de Sistemas a Eventos Discretos. São Paulo: Editora Edgard Blücher, 1996.

MIYAGI, P. E. ; JUNQUEIRA, F. ; GARCIA MELO, J. I. ; SANTOS FILHO, D. J. Internet based manufacturing and disperse productive systems. In: Brazilian Conference on Dynamics, Control and Applications, 2009, Bauru, SP.

OGATA, K. Engenharia de controle moderno. 4ª edição. São Paulo: Pearson Prentice Hall do Brasil, 2006.

OPC Foundation – About OPC – What's OPC?. Disponível em: <http://www.kepware.com/Menu_items/industry_OPC_Foundation.asp>. Acesso em 09 out. 2010.

PAMPLONA, V. F. Web Services – Construindo, disponibilizando e acessando webservices via J2SE e J2ME. Disponível em: <http://javafree.uol.com.br/artigo/871485>. Acesso em: 24 mar. 2010.

PINHEIRO, Bruno de Lima. Implementação de um ambiente para simulação distribuída de sistemas produtivos. 2006. Trabalho apresentado à Escola Politécnica da USP para obtenção do título de Engenheiro.

PROFIBUS. Site oficial da organização nacional de usuários de comunicação Profibus, 2006. Disponível em: <www.profibus.org.br>. Acesso em: 04 abr. 2010.

SCHEISS, C. Emulation: Debug it in the lab – not on the floor. In: PROCEEDINGS OF THE WINTER SIMULATION CONFERENCE, 2001. Anais: WSC, 2001. p. 1463-1465.

SIXTH FRAMEWORK PROGRAMME. Socrades Brochure. Disponível em: <http://www.socrades.eu>. Acesso em 03 jan. 2010.

SOUIT, S. Ambiente distribuído para monitoração e operação remota de sistemas produtivos. Trabalho de formatura. Escola Politécnica da Universidade de São Paulo, 2009.

TALLARD GROUP. B2B. Disponível em: <http://www.itec.com.br/semanal/NewSA400B2B.htm>. Acesso em: 24 mar. 2010.

TOVAR, E.; VASQUES, F. Real-time fieldbus communications using profibus networks. IEEE: Transactions on industrial eletronics, v. 46, n. 6, dez. 1999.

VILLANI, E., MIYAGI, P.E., VALETTE, R.: Modelling and analysis of hybrid supervisory systems. London, UK: Springer, 2007.

# APÊNDICE A – Metodologia de projeto adotada

Em projetos de qualquer natureza podem-se identificar tarefas específicas, organizando-as em termos de natureza, precedência, dependência, entre outros. Diferentes métodos podem ser utilizados na realização dessas tarefas que podem necessitar do uso de ferramentas diferentes. Pode-se entender por método o conjunto dos meios dispostos convenientemente para alcançar um fim e especialmente para chegar a um conhecimento científico e ou comunicá-lo aos outros (MICHAELIS, 1998). Em um projeto pode-se dizer que o método é o conjunto de passos que se tem de tomar visando atingir um determinado objetivo de uma tarefa dentro do projeto. Desta forma a metodologia é aqui entendida como um conjunto devidamente organizado de métodos. Dessa forma, no presente projeto, adotou-se a metodologia de projeto de sistemas de controle apresentada em (MIYAGI, 1996).

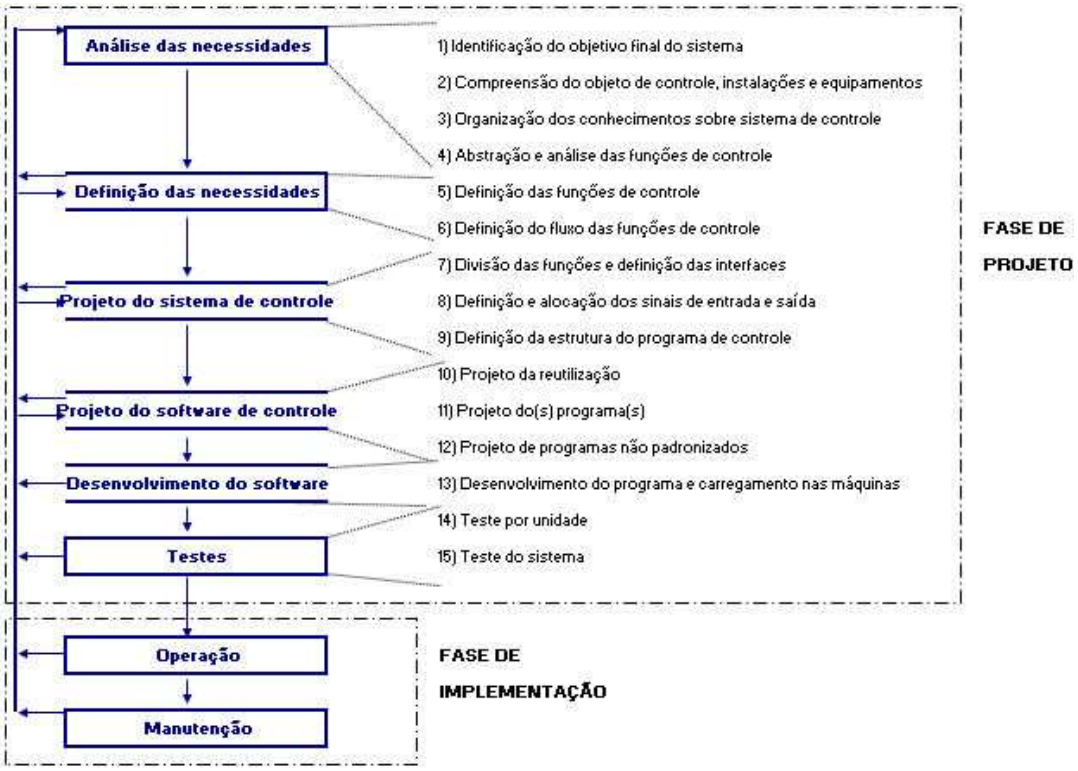



*Figura 83 - Ciclo de vida (adaptado de MIYAGI, 1996)*

A metodologia adotada considera o conceito de “ciclo de vida” no projeto de sistemas de controle. Assim, o sistema pode ser dividido em duas fases: fase de projeto (ou

desenvolvimento) e fase de implementação (MIYAGI, 1996). Um esquema do ciclo de vida é apresentado na Figura 83, com os procedimentos necessários para cada uma das etapas do ciclo de vida.

O presente projeto segue as etapas propostas para o ciclo de vida do sistema de controle:

1) **Identificação do objetivo final do sistema -** Diferentes abordagens podem ser adotadas para a identificação do objetivo final: abordagem no nível de especificação do sistema, de recursos humanos, específico de um certo domínio da produção, específico de um certo orçamento de desenvolvimento ou específico de um certo cronograma de desenvolvimento e implantação (MIYAGI, 1996). No caso o objetivo final desse projeto é o desenvolvimento de uma interface homem-computador para o subsistema de transporte, bem como sua integração aos demais subsistemas que fazem parte de um sistema de manufatura automatizado instalado na Escola Politécnica da Universidade de São Paulo (EPUSP)

2) **Compreensão do objeto de controle, instalações e equipamentos -** Estudo das funções e características de cada elemento e as inter-relações entre eles (MIYAGI,1996). Para isso, devem ser gerados o diagrama estrutural (esquemático) do objeto de controle, a lista dos atuadores, detectores e intertravamentos, e o diagrama da infraestrutura necessária. O objeto de controle é a bancada de emulação, constituída de sensores e atuadores;

3) **Organização dos conhecimentos sobre o sistema de controle** – Levantamento das técnicas de programação e especificações dos dispositivos de controle e os equipamentos periféricos (número de pontos de entrada e saída, capacidade de memória de dados e de programa, velocidade de processamento) (MIYAGI, 1996);

4) **Abstração e análise das funções de controle** – Análise das funções desejadas pelo objeto de controle, requisitos da intervenção do homem (como os modos de operação e monitoração das instalações e equipamentos) (MIYAGI, 1996). Deve-se gerar aqui um diagrama estrutural de inter-relacionamento, onde é possível verificar as funções e sua abrangência, tal como correções a serem feitas;

5) **Definição das funções de controle** – Primeiramente, faz-se um levantamento das especificações dos dispositivos de atuação, detecção, comando e monitoração. Estas funções podem ser classificadas em: função de inicialização da operação, de seleção do modo de operação, de seleção do local de operação, de sinalização, de comando, de medição, de sinalização de falha ou alarme (MIYAGI, 1996). Deve-se gerar aqui o diagrama das funções de controle, a lista dos dispositivos de atuação, de detecção e de comando/monitoração;

6) **Definição do fluxo das funções de controle** – Definição dos procedimentos que ativam as funções de controle anteriormente definidas (MIYAGI, 1996). Deve-se gerar aqui as representações em PFS (Production Flow Schema) e MFG (Mark Flow Graph), que são versões da rede de Petri próprias para a aplicação em diferentes níveis de modelagem, análise e controle do SP;

7) **Divisão das funções e definição das interfaces** – Definição do tipo e quantidade de dispositivos de controle a serem apresentados na interface do teleoperador (para os dois modos de operação: teleoperação e monitoramento) e na interface do cliente que fará os pedidos de produtos (peças montadas);

8) **Definição e alocação dos sinais de entrada e saída** – Levantamento e especificação das entradas e saídas (por exemplo, velocidade de resposta, imunidade a ruídos, confiabilidade dos contatos, capacidade de carga) (MIYAGI, 1996). No caso temos como entradas e saídas os sinais respectivamente dos instrumentos de atuação e de detecção;

9) **Definição da estrutura do programa de controle** - O controle de sistemas é resultado da interconexão de componentes numa configuração que fornece um desempenho desejado, considerando os fatores limitantes, como a complexidade do controle, distúrbio e erros de modelagem (MARUYAMA, 2007). Dessa forma, o software de controle é responsável pelo controle em tempo real do sistema, através de dispositivos como os controladores, com a execução de ações a partir de uma seqüência pré-programada e a especificação de intertravamentos (SANTOS FILHO, 2006);

10) **Projeto da reutilização** – Deve-se adotar uma organização da programação em módulos funcionais para atender a futuras modificações dos projetos de softwares para equipamentos similares (MIYAGI, 1996);

11) **Projeto dos programas** – Desenvolvimento de módulos funcionais “macro” compostos por vários módulos funcionais e integrados num programa principal que controla estes módulos funcionais (MIYAGI, 1996). A função do supervisor é realizar o controle desses módulos;

12) **Projeto de programas não padronizados** – Deve-se implementar a modularização das funções criadas, mesmo que utilizada apenas uma única vez. Essa modularização pode ser vista na implementação de funções básicas e WSs que são requisitados pelo supervisor;

13) **Desenvolvimento do programa e seu carregamento nas máquinas** – Quando é realizado um projeto iterativo através de ferramentas de apoio ao projeto, a conversão do programa-fonte (em diagrama de relés, PFS/MFG, etc) para o programa-objeto (em linguagem própria de cada CLP) é automática (MIYAGI, 1996). No presente projeto a programação é realizada no SIMATIC Manager através da linguagem Ladder, que é transcrita para código através do próprio programa;

14) **Teste por unidade e do sistema** – Execução do programa de controle de cada módulo, e em seguida, do sistema.

# APÊNDICE B – Mapeamento OPC dos Subsistemas de Alimentação, Inspeção e Montagem

Os WSs criados utilizam-se de funções definidas no OPC Server. A seguir, a listagem dos OPCs do subsistema de Alimentação, Inspeção e Montagem, respectivamente.

*Tabela 5 - Mapeamento OPC do Subsistema de Alimentação*

| **Nome do item** | **Tipo** | **Acesso** |
|---|---|---|
| S7:[S7TeleOpAlimentacao]mReq_Desligar_Ventosa | bool | RW |
| S7:[S7TeleOpAlimentacao]mReq_Estender_Cilindro | bool | RW |
| S7:[S7TeleOpAlimentacao]mReq_Ligar_Ventosa | bool | RW |
| S7:[S7TeleOpAlimentacao]mReq_MovAntiHorarioBraço | bool | RW |
| S7:[S7TeleOpAlimentacao]mReq_MovHorarioBraço | bool | RW |
| S7:[S7TeleOpAlimentacao]mReq_Recuar_Cilindro | bool | RW |
| S7:[S7TeleOpAlimentacao]mReset | bool | RW |
| S7:[S7TeleOpAlimentacao]mTelDesliga_Ventosa | bool | RW |
| S7:[S7TeleOpAlimentacao]mTelEstende_Cilindro | bool | RW |
| S7:[S7TeleOpAlimentacao]mTelLiga_Ventosa | bool | RW |
| S7:[S7TeleOpAlimentacao]mTelMovAntiHorario_Braço | bool | RW |
| S7:[S7TeleOpAlimentacao]mTelMovHorario_Braço | bool | RW |
| S7:[S7TeleOpAlimentacao]mTelRecuar_Cilindro | bool | RW |
| S7:[S7TeleOpAlimentacao]Modo_Remoto | bool | RW |
| S7:[S7TeleOpAlimentacao]mSetup_Ok | bool | RW |
| S7:[S7TeleOpAlimentacao]mNotPedido_Peça_Atendido | bool | RW |
| S7:[S7TeleOpAlimentacao]mRespNotPedido_Peça_Aten | bool | RW |

*Tabela 6 - Mapeamento OPC do Subsistema de Inspeção*

| **Nome do item** | **Tipo** | **Acesso** |
|---|---|---|
| S7:[S7Inspecao]mCor_Pedido | uint16 | RW |
| S7:[S7Inspecao]mInformaChegadaCarroTran | bool | RW |
| S7:[S7Inspecao]mPedidoInspecao | bool | RW |
| S7:[S7Inspecao]mResNotPedidoAtendido | bool | RW |
| S7:[S7Inspecao]mCor_Peca_Inspecionado 1 | uint16 | RW |
| S7:[S7Inspecao]mNotPedidoAtendido | bool | RW |
| S7:[S7Inspecao]mSetup_Ok | bool | RW |

| | | |
|---|---|---|
| :[S7Inspecao]mCor_Peca_Inspecionado_Prata | uint16 | RW |
| S7:[S7Inspecao]mCor_Peca_Inspecionado_Preta | uint16 | RW |
| S7:[S7Inspecao]mCor_Peca_Inspecionado_Rosa | uint16 | RW |
| S7:[S7Inspecao]mCor_Pedido_Prata | uint16 | RW |
| S7:[S7Inspecao]mCor_Pedido_Preta | uint16 | RW |
| S7:[S7Inspecao]mCor_Pedido_Rosa | uint16 | RW |
| S7:[S7Inspecao]mReqcarroparatransporte | bool | RW |
| S7:[S7Inspecao]mPecaAceita | bool | RW |
| S7:[S7Inspecao]mPecaRejeitada | bool | RW |
| S7:[S7Inspecao]mInfInspecaopeca | bool | RW |
| S7:[S7Inspecao]mModo_Remoto | bool | RW |
| S7:[S7Inspecao]mReqBaixarPlataforma | bool | RW |
| S7:[S7Inspecao]mReqcarroParaTransporte | bool | RW |
| S7:[S7Inspecao]mReqEstCilAlturaPeca | bool | RW |
| S7:[S7Inspecao]mReqEstCilEnvioPeca | bool | RW |
| S7:[S7Inspecao]mReqInspecaoPeca | bool | RW |
| S7:[S7Inspecao]mReqRecCilAlturaPeca | bool | RW |
| S7:[S7Inspecao]mReqSubirPlataforma | bool | RW |
| S7:[S7Inspecao]mReset | bool | RW |
| S7:[S7Inspecao]mTelBaixaPlataforma | bool | RW |
| S7:[S7Inspecao]mTelEstCilAlturaPeca | bool | RW |
| S7:[S7Inspecao]mTelEstCilEnvioPeca | bool | RW |
| S7:[S7Inspecao]mTelInspecaoPeca | bool | RW |
| S7:[S7Inspecao]mTelRecCilAlturaPeca | bool | RW |
| S7:[S7Inspecao]mTelSubirPlataforma | bool | RW |
| S7:[S7Inspecao]mReqRecCilEnvioPeca | bool | RW |
| S7:[S7Inspecao]mTelRecCilEnvioPeca | bool | RW |
| S7:[S7Inspecao]mResNotPedidoAtendido2 | bool | RW |

*Tabela 7 - Mapeamento OPC do Subsistema de Montagem*

| **Nome do item** | **Tipo** | **Acesso** |
|---|---|---|
| S7:[S7Montagem]mReq_Pegar_Peca | bool | RW |
| S7:[S7Montagem]mReq_Mon_Peca | bool | RW |
| S7:[S7Montagem]MReq_Pegar_Pino | bool | RW |
| S7:[S7Montagem]MReq_Mon_Pino | bool | RW |
| S7:[S7Montagem]MReq_Pegar_Mola | bool | RW |
| S7:[S7Montagem]MReq_Mon_Mola | bool | RW |
| S7:[S7Montagem]MReq_Pegar_Tampa | bool | RW |

| | | |
|---|---|---|
| S7:[S7Montagem]MReq_Mon_Tampa | bool | RW |
| S7:[S7Montagem]MReq_Trans_Prod | bool | RW |
| S7:[S7Montagem]mReq_Carro_Peca | bool | RW |
| S7:[S7Montagem]mReq_Carro_Prod | bool | RW |
| S7:[S7Montagem]mInfPed_Atendido | bool | RW |
| S7:[S7Montagem]Cor_Produco_carro_Rosa | bool | RW |
| S7:[S7Montagem]Cor_Produco_carro_Prata | bool | RW |
| S7:[S7Montagem]Cor_Produco_carro_Preta | bool | RW |
| S7:[S7Montagem]mSetup_Ok | bool | RW |
| S7:[S7Montagem]mTel_Pegar_Peca | bool | RW |
| S7:[S7Montagem]mTel_Mon_Peca | bool | RW |
| S7:[S7Montagem]MTel_Pegar_Pino | bool | RW |
| S7:[S7Montagem]MTel_Mon_Pino | bool | RW |
| S7:[S7Montagem]MTel_Pegar_Mola | bool | RW |
| S7:[S7Montagem]MTel_Mon_Mola | bool | RW |
| S7:[S7Montagem]MTel_Pegar_Tampa | bool | RW |
| S7:[S7Montagem]MTel_Mon_Tampa | bool | RW |
| S7:[S7Montagem]MTel_Trans_Prod | bool | RW |
| S7:[S7Montagem]mTele_Operacao | bool | RW |
| S7:[S7Montagem]mReset | bool | RW |
| S7:[S7Montagem]Inicio_Montagem | bool | RW |
| S7:[S7Montagem]mPedido_Montagem | bool | RW |
| S7:[S7Montagem]mChegada_Carro_Peca | bool | RW |
| S7:[S7Montagem]mChegada_Carro_Produto | bool | RW |
| S7:[S7Montagem]MNot_Res_Ped_Atendido | bool | RW |
| S7:[S7Montagem]Cor_Peca_carro | uint16 | RW |
| S7:[S7Montagem]Cor_Peca_carro_Prata | uint16 | RW |
| S7:[S7Montagem]Cor_Peca_carro_Preta | uint16 | RW |
| S7:[S7Montagem]Cor_Peca_carro_Rosa | uint16 | RW |
| S7:[S7Montagem]mMonitoracao | bool | RW |
| S7:[S7Montagem]mChegada_Carro_Produto_2 | bool | RW |

# APÊNDICE C – Integração: Web Site do Cliente

O Web Site do Cliente foi desenvolvido em linguagem C#, no Microsoft Visual Studio 2010, com .NET Framework 2.0. As Web References criadas foram:

| | |
|---|---|
| **WSCliente** | http://(IP)/WSCliente/WSCliente/WebService.asmx |
| **WSPedido** | http://(IP)/WSPedido/Service1.asmx |

O código do Web Site do Cliente utiliza-se das Web References acima. A seguir, os códigos completos:

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void ButtonEntrar_Click(object sender, EventArgs e)
    {
        //Pega valor da caixa de texto
        string login = TextBoxLogin.Text;
        string senha = TextBoxSenha.Text;
        //Cria parametro "CallWebService" da classe WSCliente
        WSCliente.WebService CallWebService = new WSCliente.WebService();
        int id = CallWebService.autenticar(login, senha);
        // está cadastrado
        if (id!=0)
        {
            Response.Redirect("PaginaDoCliente.aspx");
        }
        else { Label1.Text = "Cliente não encontrado. Tente novamente ou realize seu cadastro."; }
    }
    protected void LinkButton_Click(object sender, EventArgs e)
    {
        Response.Redirect("Novocadastro.aspx");
    }
}
```

NovoCadastro.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
public partial class NovoCadastro : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
```

```
    protected void Button_Click(object sender, EventArgs e)
    {
        //Pega valor da caixa de texto
        string login = TextBoxLogin.Text;
        string senha = TextBoxSenha.Text;
        //Cria parametro "CallWebService" da classe WSCliente
        WSCliente.WebService CallWebService = new WSCliente.WebService();
        CallWebService.incluir(login, senha);
        int id = CallWebService.autenticar(login, senha);
        Label1.Text = "Por favor, anote seu número de cadastro: " + Convert.ToString(id);
    }
    protected void ButtonVoltar_Click(object sender, EventArgs e)
    {
        Response.Redirect("Default.aspx");
    }
}
```

PaginaDoCliente.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
public partial class PaginaDoCliente : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Redirect("NovoPedido.aspx");
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("ConsultarPedido.aspx");
    }

    protected void LinkButton1_Click(object sender, EventArgs e)
    {
        Response.Redirect("Default.aspx");
    }
}
```

NovoPedido.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
public partial class NovoPedido : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void ButtonInserir_Click(object sender, EventArgs e)
    {
        int rosas = Convert.ToInt16(TextBoxRosa.Text);
        int pretas = Convert.ToInt16(TextBoxPreta.Text);
        int pratas = Convert.ToInt16(TextBoxPrata.Text);
        int soma = rosas + pretas + pratas;
        if (soma == 0)
        {
            Label2.Text = "Pedido sem produtos.";
        }
        else
        {
            if (soma > 3)
                Label2.Text = "Pedido inválido, pois possui mais de 3 produtos.";
            else
```

```
            {
                //Pega valor da caixa de texto
                string idcliente2 = TextBox3.Text;
                string rosaspedido2 = TextBoxRosa.Text;
                string pretaspedido2 = TextBoxPreta.Text;
                string prataspedido2 = TextBoxPrata.Text;
                //Cria parametro "CallWebService" da classe WSCliente
                WSPedido.Service1 CallWebService = new WSPedido.Service1();
                CallWebService.incluir(idcliente2, rosaspedido2, pretaspedido2, prataspedido2);
                int id = CallWebService.autenticar(idcliente2, rosaspedido2, pretaspedido2,
prataspedido2);

                Label2.Text = "Por favor, anote o número do seu pedido: " + Convert.ToString(id);
            }
        }
    }
    protected void ButtonVoltar_Click(object sender, EventArgs e)
    {
        Response.Redirect("PaginaDoCliente.aspx");
    }
}
```

ConsultarPedido.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
public partial class ConsultarPedido : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click1(object sender, EventArgs e)
    {
        //Pega valor da caixa de texto
        string id2 = TextBox1.Text;
        int id = Convert.ToInt16(id2);
        //Cria parametro "CallWebService" da classe WSCliente
        WSPedido.Service1 CallWebService = new WSPedido.Service1();
        int idcliente = CallWebService.obterIdCliente(id);
        int rosaspedido = CallWebService.obterRosasPedido(id);
        int pretapedido = CallWebService.obterPretasPedido(id);
        int pratapedido = CallWebService.obterPratasPedido(id);
        object status = CallWebService.obterStatus(id);
        int rosaspendentes = CallWebService.obterRosasPendentes(id);
        int pretaspendentes = CallWebService.obterPretasPendentes(id);
        int prataspendentes = CallWebService.obterPratasPendentes(id);
        LabelPedidoRosa.Text = Convert.ToString(rosaspedido);
        LabelPedidoPreta.Text = Convert.ToString(pretapedido);
        LabelPedidoPrata.Text = Convert.ToString(pratapedido);
        LabelProntoRosa.Text = Convert.ToString(rosaspendentes);
        LabelProntoPreta.Text = Convert.ToString(pretaspendentes);
        LabelProntoPrata.Text = Convert.ToString(prataspendentes);
        LabelStatus.Text = "STATUS DO PEDIDO: "+ Convert.ToString(status);
        Label7.Text = "";
        if (idcliente == 0)
        {
            Label7.Text = "Pedido não encontrado, tente novamente.";
        }
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("PaginaDoCliente.aspx");
    }
}
```

# APÊNDICE D – Integração: Web Service do Pedido

O Web Service do Pedido foi desenvolvido em linguagem C#, no Microsoft Visual Studio 2010, com .NET Framework 2.0. O Banco de Dados (BDPedido) foi criado no Microsoft Office Access 2003 e possui as seguintes informações:

| id |
|---|
| idcliente |
| rosaspedido |
| pretaspedido |
| prataspedido |
| status |
| rosaspendentes |
| pretaspendentes |
| prataspendentes |

O código do Web Service do Pedido utiliza-se do Banco de Dados acima. A seguir, o código completo:

Service1.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.Services;
using System.Text;
using System.Data;
using System.Data.OleDb;

namespace WSPedido
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]

    public class Service1 : System.Web.Services.WebService
    {
        [WebMethod]
        //Método de inclusão do pedido
        public void incluir(string idcliente2, string rosaspedido2, string pretaspedido2, string
prataspedido2)
        {
            int idcliente = Convert.ToInt16(idcliente2);
            int rosaspedido = Convert.ToInt16(rosaspedido2);
            int pretaspedido = Convert.ToInt16(pretaspedido2);
            int prataspedido = Convert.ToInt16(prataspedido2);

            // Inicia Pedido com todas as peças pendentes.
```

```
            string status = "pendente";
            int rosaspendentes = rosaspedido;
            int pretaspendentes = pretaspedido;
            int prataspendentes = prataspedido;
            //int rosasparcial = rosaspedido;
            //int pretasparcial = pretaspedido;
            //int pratasparcial = prataspedido;

            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string sql = "INSERT INTO Pedidos(idcliente, rosaspedido, pretaspedido, prataspedido,
status, rosaspendentes, pretaspendentes, prataspendentes) VALUES (@idcliente, @rosaspedido,
@pretaspedido, @prataspedido, @status, @rosaspendentes, @pretaspendentes, @prataspendentes)";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter[] param = new OleDbParameter[8];
            param[0] = new OleDbParameter("@idcliente", idcliente);
            param[1] = new OleDbParameter("@rosaspedido", rosaspedido);
            param[2] = new OleDbParameter("@pretaspedido", pretaspedido);
            param[3] = new OleDbParameter("@prataspedido", prataspedido);
            param[4] = new OleDbParameter("@status", status);
            param[5] = new OleDbParameter("@rosaspendentes", rosaspendentes);
            param[6] = new OleDbParameter("@pretaspendentes", pretaspendentes);
            param[7] = new OleDbParameter("@prataspendentes", prataspendentes);
            //param[8] = new OleDbParameter("@rosasparcial", rosasparcial);
            //param[9] = new OleDbParameter("@pretasparcial", pretasparcial);
            //param[10] = new OleDbParameter("@pratasparcial", pratasparcial);

            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);
            cmd.Parameters.Add(param[1]);
            cmd.Parameters.Add(param[2]);
            cmd.Parameters.Add(param[3]);
            cmd.Parameters.Add(param[4]);
            cmd.Parameters.Add(param[5]);
            cmd.Parameters.Add(param[6]);
            cmd.Parameters.Add(param[7]);
            //cmd.Parameters.Add(param[8]);
            //cmd.Parameters.Add(param[9]);
            //cmd.Parameters.Add(param[10]);

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        //Este método obtem o id do pedido recem inserido
        public int autenticar(string idcliente2, string rosaspedido2, string pretaspedido2, string
prataspedido2)
```

```
        {
            //TextBox libera um valor do tipo String
            //Estes valores devem ser convertidos pois estão como int no DB
            int idcliente = Convert.ToInt16(idcliente2);
            int rosaspedido = Convert.ToInt16(rosaspedido2);
            int pretaspedido = Convert.ToInt16(pretaspedido2);
            int prataspedido = Convert.ToInt16(prataspedido2);
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";
            //definição do comando sql
            string strSql = "SELECT MAX(id) FROM Pedidos WHERE idcliente = @idcliente AND rosaspedido =
@rosaspedido AND pretaspedido = @pretaspedido AND prataspedido = @prataspedido;";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[4];
            param[0] = new OleDbParameter("@idcliente", idcliente);
            param[1] = new OleDbParameter("@rosaspedido", rosaspedido);
            param[2] = new OleDbParameter("@pretaspedido", pretaspedido);
            param[3] = new OleDbParameter("@prataspedido", prataspedido);

            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);
            cmd.Parameters.Add(param[1]);
            cmd.Parameters.Add(param[2]);
            cmd.Parameters.Add(param[3]);

            int id = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                id = Convert.ToInt16(cmd.ExecuteScalar());
                return id;
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                cmd.Dispose();
                conn.Close();
                conn.Dispose();
            }
        }

        [WebMethod]
        public DataSet obter(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string strSql = "SELECT * from Pedidos WHERE id = @id";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            try
            {
```

```
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandText = strSql.ToString();
                cmd.CommandType = CommandType.Text;

                OleDbDataAdapter da = new OleDbDataAdapter(cmd);
                DataSet ds = new DataSet();
                da.Fill(ds);
                return ds;
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterIdCliente(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";
            //definição do comando sql
            string strSql = "SELECT idcliente from Pedidos WHERE id = @id";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            int idcliente = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                idcliente = Convert.ToInt16(cmd.ExecuteScalar());
                return idcliente;
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterRosasPedido(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";
```

```
            //definição do comando sql
            string strSql = "SELECT rosaspedido from Pedidos WHERE id = @id";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            int rosaspedido = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                rosaspedido = Convert.ToInt16(cmd.ExecuteScalar());

                return rosaspedido;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterPretasPedido(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string strSql = "SELECT pretaspedido from Pedidos WHERE id = @id";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            int pretaspedido = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                pretaspedido = Convert.ToInt16(cmd.ExecuteScalar());

                return pretaspedido;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
```

```
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterPratasPedido(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string strSql = "SELECT prataspedido from Pedidos WHERE id = @id";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);
            int prataspedido = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                prataspedido = Convert.ToInt16(cmd.ExecuteScalar());

                return prataspedido;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public Object obterStatus(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string strSql = "SELECT status from Pedidos WHERE id = @id";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            object status = "";

            try
            {
                conn.Open();
```

```
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                status = cmd.ExecuteScalar();
                return status;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterRosasPendentes(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string strSql = "SELECT rosaspendentes from Pedidos WHERE id = @id";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            int rosaspendentes = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                rosaspendentes = Convert.ToInt16(cmd.ExecuteScalar());

                return rosaspendentes;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterPretasPendentes(int id)
        {
            //define conexao
```

```
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";
            //definição do comando sql
            string strSql = "SELECT pretaspendentes from Pedidos WHERE id = @id";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            int pretaspendentes = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                pretaspendentes = Convert.ToInt16(cmd.ExecuteScalar());
                return pretaspendentes;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public int obterPratasPendentes(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string strSql = "SELECT prataspendentes from Pedidos WHERE id = @id";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter param = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);
            int prataspendentes = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                prataspendentes = Convert.ToInt16(cmd.ExecuteScalar());

                return prataspendentes;

            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
```

```
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public void atualizarStatusExe(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string sql = "UPDATE Pedidos SET status = 'executando' WHERE id = @id;";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);


            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public void atualizarStatusCon(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";
            //definição do comando sql
            string sql = "UPDATE Pedidos SET status = 'concluido' WHERE id = @id;";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
```

```
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public void atualizar_rosaspendentes(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string sql = "UPDATE Pedidos SET rosaspendentes = rosaspendentes-1 WHERE id = @id;";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);


            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }

        }

        [WebMethod]
        public void atualizar_pretaspendentes(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";
            //definição do comando sql
            string sql = "UPDATE Pedidos SET pretaspendentes = pretaspendentes-1 WHERE id = @id;";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", id);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);

            try
            {
```

```
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public void atualizar_prataspendentes(int id)
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSPedido/BDPedido.mdb";

            //definição do comando sql
            string sql = "UPDATE Pedidos SET prataspendentes = prataspendentes-1 WHERE id = @id;";

            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);

            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", id);

            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

    }
}
```

# APÊNDICE E – Integração: Aplicação Windows

O Observador de Pedido (Aplicação Windows) foi desenvolvido em linguagem C#, no Microsoft Visual Studio 2010, com .NET Framework 4.0. As Web References criadas foram:

| | |
|---|---|
| **WSCoord** | http://(IP)/WSCoord/Service1.asmx |
| **WSPedido** | http://(IP)/WSPedido/Service1.asmx |

O código da Aplicação Windows utiliza-se das Web References acima. A seguir, o código completo:

Service1.asmx.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;

namespace Aplicacao
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

            int IDPedido = 19;
            int i=1;

            //loop infinito
            while (i != 0){

                //verificar se pedido existe
                WSPedido.Service1 CallWebService = new WSPedido.Service1();
                int rosaspedido = CallWebService.obterRosasPedido(IDPedido);
                int pretaspedido = CallWebService.obterPretasPedido(IDPedido);
                int prataspedido = CallWebService.obterPratasPedido(IDPedido);
                int somapedido = rosaspedido + pretaspedido + prataspedido;

                int rosas_pend;
                int pretas_pend;
                int pratas_pend;
                int soma_pend;

                //pedido existe
                if (somapedido > 0)
                {
                    rosas_pend = CallWebService.obterRosasPendentes(IDPedido);
                    pretas_pend = CallWebService.obterPretasPendentes(IDPedido);
```

```
                    pratas_pend = CallWebService.obterPratasPendentes(IDPedido);

                    soma_pend = rosas_pend + pretas_pend + pratas_pend;

                    //tem que chamar alimentacao
                    while (soma_pend > 0)
                    {
                        WSCoord.Service1 CallWS = new WSCoord.Service1();
                        int disp = CallWS.obter_disponibilidade();

                        //está disponivel e alimenta
                        if (disp == 1)
                        CallWS.Chama_Alimentacao_Peca();

                        Thread.Sleep(5 * 1000);

                        //Reseta disponibilidade
                        CallWS.atualizar_disponibilidade_0();

                        // Checa se há mais peças pendentes.
                        rosas_pend = CallWebService.obterRosasPendentes(IDPedido);
                        pretas_pend = CallWebService.obterPretasPendentes(IDPedido);
                        pratas_pend = CallWebService.obterPratasPendentes(IDPedido);
                        soma_pend = rosas_pend + pretas_pend + pratas_pend;
                    }

                IDPedido = IDPedido + 1;
                }

            }


        }
    }
}
```

# APÊNDICE F – Integração: Web Service do Coordenador

O Web Service do Coordenador foi desenvolvido em linguagem C#, no Microsoft Visual Studio 2010, com .NET Framework 2.0. As Web References criadas foram:

| | |
|---|---|
| **WSCoord** | http://(IP)/WSPedido/Service1.asmx |
| **WSAlimentacao** | http://(IP)/WSAlimentacao/Service1.asmx |
| **WSInspecao** | http://(IP)/WSInspeciona/Service1.asmx |
| **WSTranporte** | http://\(IP)/WSTransporta/Service1.asmx |
| **WSMontagem** | http://(IP)/WSMontagem/Service1.asmx |

O código do Web Service do Coordenador utiliza-se das Web References acima. A seguir, o código completo:

Service1.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.Services;
using System.Data;
using System.Data.OleDb;
using System.Threading;

namespace WSCoordenador
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class Service1 : System.Web.Services.WebService
    {

        [WebMethod]
        public void Resposta_Alimentacao()
        {
            WSInspecao.Service1 CallWebService = new WSInspecao.Service1();
            int cor = CallWebService.Pedido_Inspecao();

            WSPedido.Service1 CallWSPedido = new WSPedido.Service1();
            int IDPedido = 19;
            int rosaspend = CallWSPedido.obterRosasPendentes(IDPedido);
            int pretaspend = CallWSPedido.obterPretasPendentes(IDPedido);
            int prataspend = CallWSPedido.obterPratasPendentes(IDPedido);
            int somapend = rosaspend + pretaspend + prataspend;

            while (somapend == 0)
            {
                IDPedido = IDPedido + 1;

                rosaspend = CallWSPedido.obterRosasPendentes(IDPedido);
                pretaspend = CallWSPedido.obterPretasPendentes(IDPedido);
                prataspend = CallWSPedido.obterPratasPendentes(IDPedido);
```

```
            somapend = rosaspend + pretaspend + prataspend;
        }

        if (cor == 1 && rosaspend > 0)
        {
            CallWebService.Envio_Peca();

            WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
            CallWSTransporte.Informa_Rosa();
            Thread.Sleep(2 * 1000);
            CallWSTransporte.Inspecao_solicita_carro();

            //informa cor para a montagem tb
            WSMontagem.Service1 CallWSMontagem = new WSMontagem.Service1();
            CallWSMontagem.Montar_rosa();

            //decrementa rosas pendentes
            CallWSPedido.atualizar_rosaspendentes(IDPedido);
        }

        else if (cor == 2 && pretaspend > 0)
        {
            CallWebService.Envio_Peca();

            WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
            CallWSTransporte.Informa_Preta();
            CallWSTransporte.Inspecao_solicita_carro();
            //informa cor para a montagem tb
            WSMontagem.Service1 CallWSMontagem = new WSMontagem.Service1();
            CallWSMontagem.Montar_preta();
            //decrementa pretas pendentes
            CallWSPedido.atualizar_pretaspendentes(IDPedido);
        }

        else if (cor == 3 && prataspend > 0)
        {
            CallWebService.Envio_Peca();

            WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
            CallWSTransporte.Informa_Prata();
            CallWSTransporte.Inspecao_solicita_carro();

            //informa cor para a montagem tb
            WSMontagem.Service1 CallWSMontagem = new WSMontagem.Service1();
            CallWSMontagem.Montar_prata();
            //decrementa pratas pendentes
            CallWSPedido.atualizar_prataspendentes(IDPedido);
        }

        else
        {
            CallWebService.Rejeita_Peca();
        }

    }

    [WebMethod]
    public void Resposta_Telecomando_de_inspecao()
    {
        WSInspecao.Service1 CallWebService = new WSInspecao.Service1();
        int cor = CallWebService.Verificar_cor_inspecionada();
        WSPedido.Service1 CallWSPedido = new WSPedido.Service1();
        int IDPedido = 19;

        int rosaspend = CallWSPedido.obterRosasPendentes(IDPedido);
        int pretaspend = CallWSPedido.obterPretasPendentes(IDPedido);
        int prataspend = CallWSPedido.obterPratasPendentes(IDPedido);
        int somapend = rosaspend + pretaspend + prataspend;

        while (somapend == 0)
        {
            IDPedido = IDPedido + 1;

            rosaspend = CallWSPedido.obterRosasPendentes(IDPedido);
```

```
            pretaspend = CallWSPedido.obterPretasPendentes(IDPedido);
            prataspend = CallWSPedido.obterPratasPendentes(IDPedido);
            somapend = rosaspend + pretaspend + prataspend;
        }

        if (cor == 1 && rosaspend > 0)
        {
            CallWebService.Envio_Peca();

            WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
            CallWSTransporte.Informa_Rosa();
            Thread.Sleep(2 * 1000);
            CallWSTransporte.Inspecao_solicita_carro();

            //informa cor para a montagem tb
            WSMontagem.Service1 CallWSMontagem = new WSMontagem.Service1();
            CallWSMontagem.Montar_rosa();

            //decrementa rosas pendentes
            CallWSPedido.atualizar_rosaspendentes(IDPedido);
        }

        else if (cor == 2 && pretaspend > 0)
        {
            CallWebService.Envio_Peca();

            WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
            CallWSTransporte.Informa_Preta();
            CallWSTransporte.Inspecao_solicita_carro();
            //informa cor para a montagem tb
            WSMontagem.Service1 CallWSMontagem = new WSMontagem.Service1();
            CallWSMontagem.Montar_preta();
            //decrementa pretas pendentes
            CallWSPedido.atualizar_pretaspendentes(IDPedido);
        }

        else if (cor == 3 && prataspend > 0)
        {
            CallWebService.Envio_Peca();

            WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
            CallWSTransporte.Informa_Prata();
            CallWSTransporte.Inspecao_solicita_carro();
            //informa cor para a montagem tb
            WSMontagem.Service1 CallWSMontagem = new WSMontagem.Service1();
            CallWSMontagem.Montar_prata();
            //decrementa pratas pendentes
            CallWSPedido.atualizar_prataspendentes(IDPedido);
        }

        else
        {
            CallWebService.Rejeita_Peca();
        }

    }

    [WebMethod]
    public void Resposta_Carrinho()
    {
        WSInspecao.Service1 CallWebService = new WSInspecao.Service1();
        CallWebService.Expulsar_peca_para_carro();
    }

    [WebMethod]
    public void Resposta_Inspecao()
    {
        WSTransporte.Service1 CallWSTransporte = new WSTransporte.Service1();
        CallWSTransporte.Montagem_solicita_carro_para_montagem();
    }

    [WebMethod]
    public void atualizar_disponibilidade_0()
    {
```

```
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSCoord/DBdisp.mdb";
            //definição do comando sql
            string sql = "UPDATE Disponibilidade SET disponibilidade = 0 WHERE id = @id;";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", 22);

            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }

        [WebMethod]
        public void atualizar_disponibilidade_1()
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSCoord/DBdisp.mdb";

            //definição do comando sql
            string sql = "UPDATE Disponibilidade SET disponibilidade = 1 WHERE id = @id;";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter[] param = new OleDbParameter[1];
            param[0] = new OleDbParameter("@id", 22);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param[0]);

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = sql.ToString();
                cmd.ExecuteNonQuery();
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
```

```
            }
        }

        [WebMethod]
        public int obter_disponibilidade()
        {
            //define conexao
            string strConnection = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/Inetpub/wwwroot/WSCoord/DBdisp.mdb";
            //definição do comando sql
            string strSql = "SELECT disponibilidade from Disponibilidade WHERE id = @id";
            //cria a conexao com o banco de dados
            OleDbConnection conn = new OleDbConnection(strConnection);
            OleDbParameter param = new OleDbParameter("@id", 22);
            OleDbCommand cmd = new OleDbCommand();
            cmd.Parameters.Add(param);

            int idcliente = 0;

            try
            {
                conn.Open();
                cmd.Connection = conn;
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = strSql.ToString();
                idcliente = Convert.ToInt16(cmd.ExecuteScalar());
                return idcliente;
            }
            catch (OleDbException ex)
            {
                throw new Exception("ERRO BANCO DE DADOS: " + ex.Message.ToString());
            }
            catch (Exception ex)
            {
                throw new Exception("ERRO RUNTIME: " + ex.Message.ToString());
            }
            finally
            {
                conn.Close();
                conn.Dispose();
                cmd.Dispose();
            }
        }


        // Chama alimentação a partir do coordenador
        [WebMethod]
        public void Chama_Alimentacao_Peca ()
        {
            WSAlimentacao.Service1 CallWebService = new WSAlimentacao.Service1();
            CallWebService.Pedido_Peca();
        }

        [WebMethod]
        public void Resposta_Carrinho_inicio_montagem()
        {
            WSMontagem.Service1 CallWebService = new WSMontagem.Service1();
            CallWebService.Pegar_Peca();
        }

        [WebMethod]
        public void Requisita_Carrinho_fim_montagem()
        {
            WSTransporte.Service1 CallWebService = new WSTransporte.Service1();
            CallWebService.Montagem_solicita_carro_para_produto();
        }
    }
}
```

# APÊNDICE G– Subsistema de transporte: Web Service

O Web Service do subsistema de transporte foi desenvolvido em linguagem C#, no Microsoft Visual Studio 2010, com .NET Framework 2.0. A Web Reference criada foi:

| | |
|---|---|
| **WSCoordenador** | http://(IP)/WSCoord/Service1.asmx |

O código do Web Service utiliza-se da Web Reference acima. A seguir, o código completo:

TranspTeleOpWS.cs

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.Services;
using System.Xml;
using System.Threading;
using opcconn;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

public class TranspTeleOpWS : System.Web.Services.WebService {
    private int mySleep = 1000;

    public TranspTeleOpWS () {
        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

    [WebMethod]
    public string Conecta() {
        string retorno = "OK";
        XmlDocument xmlDoc = new XmlDocument();
        XmlDocument xmlDocTela = new XmlDocument();
        opcClient opc;

        if (Application["opcClient"] == null) {
            try {
                // Trava o application para apenas apenas uma aplicação alocar o objeto
                Application.Lock();

                // Le o arquivo com os parâmetros para configurar o objeto OPC
                xmlDoc.Load(HttpContext.Current.Request.PhysicalApplicationPath +
"./xml/opcconfig.xml");
                XmlNode xmlServer = xmlDoc.GetElementsByTagName("server")[0];

                // Cria o objeto OPC
                opc = new opcClient(xmlServer.Attributes.GetNamedItem("name").Value);

                if (opc.ErrorMessage.Length > 0) {
                    opc.Dispose();
                    // Não foi possível criar o objeto OPC
                    retorno = "NOK";
                }
                else {
                    // Cria o grupo
                    foreach (XmlNode xmlGroup in xmlServer.SelectNodes("group")) {
```

```
                        opc.AddGroup(xmlGroup.Attributes.GetNamedItem("name").Value);
                        if (opc.ErrorMessage.Length == 0) {
                            // Cria os itens dentro do grupo
                            foreach (XmlNode xmlItem in xmlGroup.SelectNodes("item")) {

opc.GetGroupByPosition(0).AddItem(xmlItem.Attributes.GetNamedItem("id").Value,
xmlItem.Attributes.GetNamedItem("memory").Value);
                                if (opc.GetGroupByPosition(0).ErrorMessage.Length > 0) {
                                    // Não foi possível criar ao menos um item
                                    retorno = "NOK";
                                    break;
                                }
                            }
                        }
                        else {
                            // Não foi possível criar o grupo
                            retorno = "NOK";
                        }
                    }
                }
                if (retorno != "NOK") {
                    // Guarda o objeto OPC no application para que seja a única instância
                    Application["opcClient"] = opc;

                    // Limpa o XML para montar a tela
                    // - retira atributos que não seja ID de DESC
                    // - retira itens cujo atributo ID não comece com W (são apenas de leitura)
                    xmlDocTela.LoadXml(xmlDoc.InnerXml);

                    XmlNodeList arrayItens = xmlDocTela.GetElementsByTagName("item");
                    for (int i = arrayItens.Count - 1; i >= 0; i--) {
                        // Se for de escrita, deixo e limpo os demais atributos
                        if
(String.Compare(arrayItens[i].Attributes.GetNamedItem("id").Value.Substring(0, 1), "W") == 0) {
                            //arrayItens[i].Attributes.RemoveNamedItem("memory");
                            arrayItens[i].Attributes.RemoveNamedItem("value");
                        }
                        // Do contrário, removo o nó inteiro
                        else {
                            xmlDocTela.GetElementsByTagName("group")[0].RemoveChild(arrayItens[i]);
                        }
                    }
                    Application["xmlDocTela"] = xmlDocTela.InnerXml;

                    // Limpa o XML para atualizar variáveis de tela
                    arrayItens = xmlDoc.GetElementsByTagName("item");
                    for (int i = arrayItens.Count - 1; i >= 0; i--) {
                        // Se for de escrita, deixo e limpo os demais atributos
                        if
(String.Compare(arrayItens[i].Attributes.GetNamedItem("id").Value.Substring(0, 1), "W") == 0) {
                            arrayItens[i].Attributes.RemoveNamedItem("memory");
                            arrayItens[i].Attributes.RemoveNamedItem("desc");
                        }
                        // Do contrário, removo o nó inteiro
                        else {
                            xmlDoc.GetElementsByTagName("group")[0].RemoveChild(arrayItens[i]);
                        }
                    }
                    Application["xmlDoc"] = xmlDoc.InnerXml;
                }
            }
            catch (Exception e) {
                // Apenas em caso de algum exceção, mas não estou preocupado com qual
                Application["xmlDoc"] = null;
                Application["xmlDocTela"] = null;
                Application["opcClient"] = null;
                retorno = "NOK";
            }
            finally {
                xmlDoc = null;
                xmlDocTela = null;
                // Destrava o application
                Application.UnLock();
            }
```

```
    }
    if (retorno == "NOK") {
        Application["xmlDoc"] = null;
        Application["xmlDocTela"] = null;
        Application["opcClient"] = null;
    }
    return retorno;
}

[WebMethod]
public XmlDocument MontaComandos() {
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.LoadXml(Application["xmlDocTela"].ToString());
    return xmlDoc;
}

[WebMethod]
public string Desconecta() {
    try {
        Application.Lock();
        opcClient opc = (opcClient)Application["opcClient"];
        opc.Dispose();
        opc = null;
    }
    catch (Exception e) {
    }
    finally {
        Application["opcClient"] = null;
        Application["xmlDocTela"] = null;
        Application["xmlDoc"] = null;
        Application.UnLock();
    }
    return "OK";
}

[WebMethod]
public XmlDocument AtualizaComandos() {
    opcClient opc;
    opcClientGroup group;
    XmlDocument xmlDoc = new XmlDocument();

    bool R000, R001;
    bool W000, W001, W006, W007, W008, W009;
    int W002, W003, W004, W005;

    try {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);
        xmlDoc.LoadXml(Application["xmlDoc"].ToString());

        R000 = (Int32.Parse(group.GetItemById("R000").Read()) == 0) ? false : true;
        R001 = (Int32.Parse(group.GetItemById("R001").Read()) == 0) ? false : true;

        W000 = (Int32.Parse(group.GetItemById("W000").Read()) == 0) ? false : true;
        W001 = (Int32.Parse(group.GetItemById("W001").Read()) == 0) ? false : true;
        W005 = W002 = W003 = W004 = Int32.Parse(group.GetItemById("W005").Read());

        W006 = (Int32.Parse(group.GetItemById("W006").Read()) == 0) ? false : true;
        W007 = (Int32.Parse(group.GetItemById("W007").Read()) == 0) ? false : true;
        W008 = (Int32.Parse(group.GetItemById("W008").Read()) == 0) ? false : true;
        W009 = (Int32.Parse(group.GetItemById("W009").Read()) == 0) ? false : true;


        // Leitura das variáveis - fim

        // Atualiza saída - início

        foreach (XmlNode item in xmlDoc.GetElementsByTagName("item")) {
            if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W000") == 0){
                item.Attributes.GetNamedItem("value").Value = "ok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W001") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = (R000) ? "ok" : "nok";
```

```
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W002") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = "ok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W003") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = "ok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W004") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = "ok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W005") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = "nok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W006") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = "ok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W007") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = "ok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W008") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = (R001) ? "ok" : "nok";
            }
            else if (String.Compare(item.Attributes.GetNamedItem("id").Value, "W009") == 0)
            {
                item.Attributes.GetNamedItem("value").Value = (R001) ? "ok" : "nok";
            }
        }

        // Atualiza saída - fim
    }
    catch (Exception e) {
        xmlDoc.LoadXml("<root><erro>NOK</erro></root>");
    }
    finally {
        group = null;
        opc = null;
    }
    return xmlDoc;
}

[WebMethod]
public string Req_Inspecao_Est2()
{
    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try
    {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);

        group.GetItemById("W000").Write("1");
        //Thread.Sleep(mySleep);
        //group.GetItemById("W000").Write("0");
    }
    catch (Exception e)
    {
        retorno = "NOK";
    }
    finally
    {
        group = null;
        opc = null;
    }
    return retorno;
}
```

```
[WebMethod]
public string Saida_autorizada_Est2()
{
    WSCoordenador.Service1 CallWebService = new WSCoordenador.Service1();
    CallWebService.Resposta_Carrinho();
    Thread.Sleep(7 * 1000);

    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);

        group.GetItemById("W001").Write("1");
        Thread.Sleep(mySleep);
        group.GetItemById("W001").Write("0");

        Thread.Sleep(mySleep);
        //libera carro e tira a solicitação de carrinho para a estação 2
        group.GetItemById("W000").Write("0");
    }
    catch (Exception e) {
        retorno = "NOK";
    }
    finally {
        group = null;
        opc = null;
    }

    return retorno;
}

[WebMethod]
public string Cor_pedido_Est2()
{
    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try
    {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);

        group.GetItemById("W005").Write("0");
        //Thread.Sleep(mySleep);
        //group.GetItemById("W019").Write("0");
    }
    catch (Exception e)
    {
        retorno = "NOK";
    }
    finally
    {
        group = null;
        opc = null;
    }
    return retorno;
}

[WebMethod]
public string Cor_pedido_Est2_Rosa()
{
    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try
    {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);
```

```
            group.GetItemById("W005").Write("1");
            //Thread.Sleep(mySleep);
            //group.GetItemById("W019").Write("0");
        }
        catch (Exception e)
        {
            retorno = "NOK";
        }
        finally
        {
            group = null;
            opc = null;
        }
        return retorno;
    }

    [WebMethod]
    public string Cor_pedido_Est2_Preta()
    {
        string retorno = "OK";
        opcClient opc;
        opcClientGroup group;

        try
        {
            opc = (opcClient)Application["opcClient"];
            group = opc.GetGroupByPosition(0);

            group.GetItemById("W005").Write("2");
            //Thread.Sleep(mySleep);
            //group.GetItemById("W019").Write("0");
        }
        catch (Exception e)
        {
            retorno = "NOK";
        }
        finally
        {
            group = null;
            opc = null;
        }
        return retorno;
    }

    [WebMethod]
    public string Cor_pedido_Est2_Prata()
    {
        string retorno = "OK";
        opcClient opc;
        opcClientGroup group;

        try
        {
            opc = (opcClient)Application["opcClient"];
            group = opc.GetGroupByPosition(0);

            group.GetItemById("W005").Write("3");
            //Thread.Sleep(mySleep);
            //group.GetItemById("W019").Write("0");
        }
        catch (Exception e)
        {
            retorno = "NOK";
        }
        finally
        {
            group = null;
            opc = null;
        }
        return retorno;
    }

    [WebMethod]
```

```
public string Req_peca_st3()
{
    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try
    {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);

        group.GetItemById("W006").Write("1");
        //Thread.Sleep(mySleep);
        //group.GetItemById("W000").Write("0");
    }
    catch (Exception e)
    {
        retorno = "NOK";
    }
    finally
    {
        group = null;
        opc = null;
    }
    return retorno;
}

[WebMethod]
public string Req_produto_montado_st3()
{
    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try
    {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);

        group.GetItemById("W007").Write("1");
        //Thread.Sleep(mySleep);
        //group.GetItemById("W000").Write("0");
    }
    catch (Exception e)
    {
        retorno = "NOK";
    }
    finally
    {
        group = null;
        opc = null;
    }
    return retorno;
}

[WebMethod]
public string Saida_Autorizada_St3()
{
    WSCoordenador.Service1 CallWebService = new WSCoordenador.Service1();
    CallWebService.Resposta_Carrinho_inicio_montagem();

    Thread.Sleep(20 * 1000);

    string retorno = "OK";
    opcClient opc;
    opcClientGroup group;

    try
    {
        opc = (opcClient)Application["opcClient"];
        group = opc.GetGroupByPosition(0);

        //group.GetItemById("W008").Write("1");
        //Thread.Sleep(mySleep);
```

```
            group.GetItemById("W008").Write("0");

            //libera carro e tira a solicitação de carrinho para a estação 3
            //group.GetItemById("W006").Write("0");
            //group.GetItemById("W007").Write("0");
        }
        catch (Exception e)
        {
            retorno = "NOK";
        }
        finally
        {
            group = null;
            opc = null;
        }
        return retorno;


    }

    [WebMethod]
    public string Saida_Autorizada_St3_2()
    {
        string retorno = "OK";
        opcClient opc;
        opcClientGroup group;

        try
        {
            opc = (opcClient)Application["opcClient"];
            group = opc.GetGroupByPosition(0);

            group.GetItemById("W009").Write("1");
            Thread.Sleep(mySleep);
            group.GetItemById("W009").Write("0");

            //libera carro e tira a solicitação de carrinho para a estação 3
            group.GetItemById("W006").Write("0");
            group.GetItemById("W007").Write("0");
        }
        catch (Exception e)
        {
            retorno = "NOK";
        }
        finally
        {
            group = null;
            opc = null;
        }

        WSCoordenador.Service1 CallWebService = new WSCoordenador.Service1();
        CallWebService.atualizar_disponibilidade_1();

        return retorno;
    }

}
```

# ANEXO A – Subsistema de transporte: Web Site do Teleoperador

O código a seguir foi desenvolvido em linguagem C# no Visual Studio. Este código utiliza-se da Web Service criada para o subsistema de transporte (Apêndice G).

estacao.js

```
// Variáveis Globais - início ------------------------------------------------------
var url = "../TranspTeleOpWS/TranspTeleOpWS.asmx/";
// Variáveis Globais - fim ---------------------------------------------------------
```

jframework.io.js

```
_classes.registerClass("jfAjax", "xbObject");
function jfAjax () {
    _classes.defineClass("jfAjax", _prototype_func);
    this.init();
    function _prototype_func() {
        jfAjax.prototype.init = init;
        function init() {
            this.parentMethod("init");
            this._xmlHttp = null;
            this._errorCode = null;
            this._return = null;
            // Parâmetros de entrada
            this.url = "";
            this.postData = "";
            this.returnFunction = null;
            try {
                // Firefox, Opera 8.0+, Safari, IE7
                this._xmlHttp = new XMLHttpRequest();
                if (this._xmlHttp.overrideMimeType) {
                    this._xmlHttp.overrideMimeType("text/html");
                }
            }
            catch (e) {
                // Internet Explorer
                var XmlHttpVersions = new Array("MSXML2.XMLHTTP.6.0", "MSXML2.XMLHTTP.5.0",
"MSXML2.XMLHTTP.4.0", "MSXML2.XMLHTTP.3.0", "MSXML2.XMLHTTP", "Microsoft.XMLHTTP");
                for(var i = 0; i < XmlHttpVersions.length && !this._xmlHttp; i++) {
                    try {
                        this._xmlHttp = new ActiveXObject(XmlHttpVersions[i]);
                    }
                    catch(e){}
                }
                if (i == XmlHttpVersions.length) {
                    this._errorCode = "IO6001";
                }
            }
        };

        jfAjax.prototype.getReturn = getReturn;
        function getReturn() {
            return this._return;
        };

        jfAjax.prototype.getErrorCode = getErrorCode;
        function getErrorCode() {
            return this._errorCode;
        };
        jfAjax.prototype.postSynchronouslyCall = postSynchronouslyCall;
        function postSynchronouslyCall() {
            try {
                this._xmlHttp.open('POST', this.url, false);
                this._xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
                this._xmlHttp.setRequestHeader("Cache-Control", "no-store, no-cache, must-revalidate");
```

```
                this._xmlHttp.setRequestHeader("Pragma", "no-cache");
                this._xmlHttp.setRequestHeader("Connection", "Keep-Alive");
                this._xmlHttp.setRequestHeader("Content-length", this.postData.length);
                this._xmlHttp.send(this.postData);
            }
            catch(e) {
                this._errorCode = "IO6002";
            }
            if (this._xmlHttp.status == 200) {
                this._return = this._xmlHttp.responseText;
            }
        };
        jfAjax.prototype.postAsynchronouslyCall = postAsynchronouslyCall;
        function postAsynchronouslyCall() {
            try {
                this._xmlHttp.open('POST', this.url, true);
                this._xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
                this._xmlHttp.setRequestHeader("Cache-Control", "no-store, no-cache, must-revalidate");
                this._xmlHttp.setRequestHeader("Pragma", "no-cache");
                this._xmlHttp.setRequestHeader("Content-length", this.postData.length);
                this._xmlHttp.setRequestHeader("Connection", "Keep-Alive");
                this._xmlHttp.onreadystatechange = function() {
                    if (this.readyState == 4) {
                        if (this.status == 200) {
                            this.returnFunction(this.responseText);
                        }
                        else {
                            this.returnFunction(null);
                        }
                    }
                }
                this._xmlHttp.send(this.postData);
            }
            catch(e) {
                this._errorCode = "IO6002";
            }
        };
        jfAjax.prototype.getSynchronouslyCall = getSynchronouslyCall;
        function getSynchronouslyCall() {
            try {
                this._xmlHttp.open('GET', this.url + "?" + this.postData, false);
                this._xmlHttp.send(null);
            }
            catch(e) {
                this._errorCode = "IO6002";
            }
            if (this._xmlHttp.status == 200) {
                this._return = this._xmlHttp.responseText;
            }
        };
        jfAjax.prototype.getAsynchronouslyCall = getAsynchronouslyCall;
        function getAsynchronouslyCall() {
            try {
                this._xmlHttp.open('GET', this.url + "?" + this.postData, true);
                this._xmlHttp.onreadystatechange = function() {
                    if (this.readyState == 4) {
                        if (this.status == 200) {
                            this.returnFunction(this.responseText);
                        }
                        else {
                            this.returnFunction(null);
                        }
                    }
                }
                this._xmlHttp.send(null);
            }
            catch(e) {
                this._errorCode = "IO6002";
            }
        };
    };
};

_classes.registerClass("jfLoadXML", "xbObject");
```

```
function jfLoadXML () {
    _classes.defineClass("jfLoadXML", _prototype_func);
    this.init();
    function _prototype_func() {
        jfLoadXML.prototype.init = init;
        function init() {
            this.parentMethod("init");
            this._xmlDoc = null;
            this._errorCode = null;
            // code for IE
            if (window.ActiveXObject) {
                this._xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
            }
            // code for Mozilla, Firefox, Opera, etc.
            else if (document.implementation && document.implementation.createDocument) {
                this._xmlDoc = document.implementation.createDocument("", "", null);
            }
            else {
                this._errorCode = "IO6003";
            }
        };
        jfLoadXML.prototype.getXML = getXML;
        function getXML(xmlDoc) {
            if (this._errorCode == null) {
                this._xmlDoc.async = false;
                try {
                    this._xmlDoc.load(xmlDoc);
                }
                catch(e) {
                    this._errorCode = "IO6004";
                }
                return this._xmlDoc;
            }
        };
        jfLoadXML.prototype.getErrorCode = getErrorCode;
        function getErrorCode() {
            return this._errorCode;
        };
    };
};
_classes.registerClass("jfParseXML", "xbObject");
function jfParseXML () {
    _classes.defineClass("jfParseXML", _prototype_func);
    this.init();
    function _prototype_func() {
        jfParseXML.prototype.init = init;
        function init() {
            this.parentMethod("init");
            this._xmlDoc = null;
            this._errorCode = null;
        };
        jfParseXML.prototype.getXML = getXML;
        function getXML(xmlDoc) {
            // code for IE
            if (window.ActiveXObject) {
                this._xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
                this._xmlDoc.async = false;
                this._xmlDoc.loadXML(xmlDoc);
            }
            // code for Mozilla, Firefox, Opera, etc.
            else if (document.implementation && document.implementation.createDocument) {
                var parser = new DOMParser();
                this._xmlDoc = parser.parseFromString(xmlDoc,"text/xml");
                parser = null;
            }
            else {
                this._errorCode = "IO6004";
            }
            return this._xmlDoc;
        };

        jfParseXML.prototype.getErrorCode = getErrorCode;
        function getErrorCode() {
            return this._errorCode;
```

```
        };
    };
};
```

jframework.menu.js

```
loadCSS("jframework.menu.css");

_classes.registerClass("jfMenuH", "xbObject");
function jfMenuH (id, parentId) {
    _classes.defineClass("jfMenuH", _prototype_func);
    this.init(id, parentId);

    function _prototype_func() {

        jfMenuH.prototype.init = init;
        function init(id, parentId) {
            this.parentMethod("init");
            this._div = document.createElement("div");
            this._div.style.visibility = "hidden";
            if (id != null) {
                this._div.id = id;
            }
            this._div.className = "jfMenuH";
            this._div.parentId = parentId;
            this._div.cont = 0;
            this._div.menuSelected = null;
            this._menuWidth = 0;
            // Insere o elemento no parent
            if (parentId != null) document.getElementById(parentId).appendChild(this._div);
        };
        jfMenuH.prototype.addMenuItem = addMenuItem;
        function addMenuItem(text, action) {
            var _tagUL = document.createElement("ul");
            var _tagLI = document.createElement("li");
            var _tagA = document.createElement("a");
                        _tagA.href = "#";
            _tagA.innerHTML = text;
            _tagLI.className = "jfMenuHItem";
            this._div.menuSelected = _tagLI.id = "mItem" + this._div.cont++;
            _tagLI.onclick = action;
            _tagLI.appendChild(_tagA);
            _tagUL.appendChild(_tagLI);
            this._div.appendChild(_tagUL);
            this._menuWidth += _tagUL.clientWidth + 6;
        };
        jfMenuH.prototype.addSubMenuItem = addSubMenuItem;
        function addSubMenuItem(text, action) {
            var _tagLI = document.getElementById(this._div.menuSelected);
            var _tagUL = _tagLI.getElementsByTagName("ul");
            if (_tagUL.length == 0) {
                _tagUL = document.createElement("ul");
                _tagUL.className = "jfMenuHDD";
                _tagLI.target = _tagUL.id = "mItem" + this._div.cont++;
                _tagLI.onmouseover = function() {
                    document.getElementById(this.target).style.visibility = "visible";
                };
                _tagLI.onmouseout = function() {
                    document.getElementById(this.target).style.visibility = "hidden";
                };
                _tagLI.getElementsByTagName("a")[0].innerHTML += " <span>&#9660;</span>";
                _tagLI.appendChild(_tagUL);
            }
            else {
                _tagUL = _tagUL[0];
            }
            _tagLI = document.createElement("li");
            var _tagA = document.createElement("a");
            _tagA.href = "#";
            _tagA.innerHTML = text;
            _tagLI.className = "jfMenuHDDItem";
            _tagLI.id = "mItem" + this._div.cont++;
            _tagLI.onclick = action;
            _tagLI.appendChild(_tagA);
```

```
            _tagUL.appendChild(_tagLI);
        };
        jfMenuH.prototype.show = show;
        function show() {
            this._div.style.width = this._menuWidth + "px";
            this._div.style.visibility = "visible";
        };
        jfMenuH.prototype.getObj = getObj;
        function getObj() {
            return this._div;
        };
    };
};
```

jframework.messages.js

```
// Variáveis globais - início ------------------------------------------------------------
var globalErrorMessagesXml = null;
// Variáveis globais - fim ---------------------------------------------------------------

// Carrega as mensagens de erro utilizadas na aplicação
function loadMessagesXml() {
    var _loadMessageXml = new loadXML();
    globalErrorMessagesXml = _loadMessageXml.getXML("./xml/messages.xml");
    _loadMessageXml = null;
};
alertFunction = function(codMessage, parameters) {
    if (globalErrorMessagesXml == null) {
        loadMessagesXml();
    }
    var _message = "";
    try {
        _message = globalErrorMessagesXml.getElementsByTagName(codMessage)[0].getAttribute("texto");
    }
    catch(e) {
        _message = "Mensagem " + codMessage + " não cadastrada!";
    }

    if (parameters != "") {
        var arrayParameters = parameters.split("#");
        for (var i = 0; i < arrayParameters.length; i++) {
            _message = _message.replace("{" + i + "}", arrayParameters[i]);
        }
    }
    var _cursor = document.body.style.cursor;
    document.body.style.cursor = "default";
    alert(_message);
    document.body.style.cursor = _cursor;
    _cursor = null;
};
confirmFunction = function(codMessage, parameters) {
    if (globalErrorMessagesXml == null) {
        loadMessagesXml();
    }
    var _message = globalErrorMessagesXml.getElementsByTagName(codMessage)[0].getAttribute("texto");
    if (_message == null) {
        _message = "Mensagem " + codMessage + " não cadastrada!";
        alert(_message);
        return false;
    }
    else {
        if (arrayParameters != "") {
            var arrayParameters = parameters.split("#");
            for (var i = 0; i < arrayParameters.length; i++) {
                _message = _message.replace("{" + i + "}", arrayParameters[i]);
            }
        }
        var _cursor = document.body.style.cursor;
        document.body.style.cursor = "default";
        var _returnOption = confirm(_message);
        document.body.style.cursor = _cursor;
        _cursor = null;
        return _returnOption;
    }
```

```
};
```

jframework.util.js

```
// Função que carrega o CSS específico de uma classe - início -------------------------
loadCSS = function(cssName) {
    if (document.getElementById(cssName) == null) {
        var _css = document.createElement("link");
        _css.type = "text/css";
        _css.rel = "stylesheet";
        _css.href = _css.id = "./css/" + cssName;
        document.getElementsByTagName("head")[0].appendChild(_css);
        _css = null;
    }
};
// Função que carrega o CSS específico de uma classe - fim ----------------------------
// Função que apresenta uma mensagem de debug na tela - início ------------------------
// Esta é apenas uma assinatura que não faz nada
debugFunction = function(message) {
};
// Função que apresenta uma mensagem de debug na tela - fim ---------------------------
```

jframework.window.js

```
// Objeto que representa um botão - início ---------------------------------------------
_classes.registerClass("jfButtom", "xbObject");
function jfButtom (id, className, text, parentId) {
    _classes.defineClass("jfButtom", _prototype_func);
    this.init(id, className, text, parentId);
    function _prototype_func() {

        jfButtom.prototype.init = init;
        function init(id, className, text, parentId) {
            this.parentMethod("init");
            // Atributos
            this._a = document.createElement("a");
            //this._a.style.visibility = "hidden";
            this._a.style.display = "none";
            this._a.id = id;
            this._a.className = (className != null) ? className : "jfButtom";
            this._a.innerHTML = text;
            this._a.parentId = parentId;
            this._a.href = "#";
            this._a.enabled = true;
            this._a.validate = function () {
                return true;
            };
            this._a.action = function () {
                return true;
            };

            // Eventos
            this._a.onclick = function () {
                if (this.enabled == true) {
                    this.enabled = false;

                    var _cursor = document.body.style.cursor;
                    document.body.style.cursor = "wait";
                    if (this.validate()) {
                        this.action();
                    }
                    document.body.style.cursor = _cursor;
                    this.enabled = true
                }
            };
        };
        jfButtom.prototype.setValidation = setValidation;
        function setValidation(validationFunction) {
            this._a.validate = validationFunction;
        };
        jfButtom.prototype.setAction = setAction;
        function setAction(actionFunction) {
            this._a.action = actionFunction;
        };
```

```
        jfButtom.prototype.show = show;
        function show() {
            //this._a.style.visibility = "visible";
            this._a.style.display = "inline";
        };
        jfButtom.prototype.getObj = getObj;
        function getObj() {
            return this._a;
        };
    };
};
// Objeto que representa um botão - fim ------------------------------------------------
// Objeto que representa uma janela - início -------------------------------------------
_classes.registerClass("jfWindow", "xbObject");
function jfWindow (id, className, parentId) {
    _classes.defineClass("jfWindow", _prototype_func);
    this.init(id, className, parentId);
    function _prototype_func() {
        jfWindow.prototype.init = init;
        function init(id, className, parentId) {
            this.parentMethod("init");
            // Atributos
            this._div = document.createElement("div");
            //this._div.style.visibility = "hidden";
            this._div.style.display = "none";
            this._div.id = id;
            this._div.className = (className != null) ? className : "jfWindow";
            this._div.parentId = parentId;
            // Cria uma região para inserir título
            this._div.tit = null;
            // Cria uma região para inserir conteúdo
            this._div.cont = document.createElement("div");
            this._div.cont.className = this._div.className + "Cont";
            this._div.appendChild(this._div.cont);
            // Cria uma região para inserir botões
            this._div.botoes = null;
            // Insere o elemento no parent
            if (parentId != null) document.getElementById(parentId).appendChild(this._div);
        };
        jfWindow.prototype.setTitle = setTitle;
        function setTitle(text) {
            this._div.tit = document.createElement("div");
            this._div.tit.className = this._div.className + "Tit";
            this._div.tit.innerHTML = text;
            this._div.insertBefore(this._div.tit, this._div.cont);
        };
        jfWindow.prototype.setTitleAction = setTitleAction;
        function setTitleAction(action) {
            var _tagA = document.createElement("a");
            _tagA.href = "#";
            _tagA.onclick = action;
            _tagA.innerHTML = "clique aqui.";
            this._div.tit.innerHTML += ", ";
            this._div.tit.appendChild(_tagA);
            _tagA = null;
        };
        jfWindow.prototype.addCont = addCont;
        function addCont(newCont) {
            this._div.cont.appendChild(newCont);
        };
        jfWindow.prototype.addButtom = addButtom;
        function addButtom(className, text, validationFunction, actionFunction) {
            if (this._div.botoes == null) {
                this._div.botoes = document.createElement("div");
                this._div.botoes.className = this._div.className + "Buttoms";
                this._div.botoes.cont = 1;
                this._div.appendChild(this._div.botoes);
            }
            var _botao = new jfButtom(this._div.id + this._div.botoes.cont++, className, text,
this._div.id);
            this._div.botoes.appendChild(_botao.getObj());
            if (validationFunction != null) _botao.setValidation(validationFunction);
            if (actionFunction != null) _botao.setAction(actionFunction);
            _botao.show();
```

```
            _botao = null;
        };
        jfWindow.prototype.show = show;
        function show() {
            //this._div.style.visibility = "visible";
            this._div.style.display = "block";
        };
        jfWindow.prototype.hide = hide;
        function hide() {
            //this._div.style.visibility = "hidden";
            this._div.style.display = "none";
        };
        jfWindow.prototype.clearCont = clearCont;
        function clearCont() {
            var _arrayCont = this._div.cont.childNodes;
            for (var i = _arrayCont.length - 1; i >= 0; i--) {
                this._div.cont.removeChild(_arrayCont[i]);
            }
        };
        jfWindow.prototype.clearAll = clearAll;
        function clearAll() {
            //this._div.style.visibility = "hidden";
            this._div.style.display = "none";
            this.clearCont();
            if (this._div.tit != null) {
                this._div.removeChild(this._div.tit);
                this._div.tit = null;
            }
            if (this._div.botoes != null) {
                this._div.removeChild(this._div.botoes);
                this._div.botoes = null;
            }
        };
        jfWindow.prototype.getObj = getObj;
        function getObj() {
            return this._div;
        };
    };
};
// Objeto que representa uma janela - fim ----------------------------------------------
```

main.js

```
// Variáveis Globais - início ----------------------------------------------------------
var flagAtualiza = false;
var timeOut1 = null;
var timeOut2 = null;
// Variáveis Globais - fim -------------------------------------------------------------
window.onload = function() {
    // Monta o Menu
    var menu = new jfMenuH("jfMenuH", "menu");
    menu.addMenuItem("Câmeras", null);
    menu.addSubMenuItem("Câmera 1", menuCamera1);
    menu.addSubMenuItem("Câmera 2", menuCamera2);
    menu.addSubMenuItem("Câmera 3", menuCamera3);
    menu.addMenuItem("Conectar", menuConectar);
    menu.addMenuItem("Desconectar", menuDesconectar);
    menu.show();
    menu = null;
    // Define as dimensões da área de trabalho
    funcaoDimensionaAreaDeTrabalho();
};
window.onresize = function() {
    funcaoDimensionaAreaDeTrabalho();
};
window.onunload = function() {
    //alert("Volte sempre!");
};
funcaoDimensionaAreaDeTrabalho = function() {
    var intContainerHeight = document.getElementById("container").clientHeight;
    var intLogoHeight = document.getElementById("logo").clientHeight;
    var intMenuLineHeight = document.getElementById("menu").clientHeight;
    var intRodapeHeight = document.getElementById("rodape").clientHeight;
```

```
    var intAreaTrabalho = intContainerHeight - intLogoHeight - intMenuLineHeight - intRodapeHeight -
30;
    if (intAreaTrabalho > 0) {
        document.getElementById("areatrabalho").style.height = intAreaTrabalho + "px";
    }
    else {
        document.getElementById("areatrabalho").style.height = "0px";
    }
};
// Funções para apresentar máscara de dados - início ----------------------------------
funcaoAguarde = function(mostra) {
    if (mostra) {
        document.getElementById("aguarde").style.display = "block";
    }
    else {
        document.getElementById("aguarde").style.display = "none";
    }
};
// Funções para apresentar máscara de dados - fim -------------------------------------
// Menu - início ----------------------------------------------------------------------
menuCamera1 = function() {
    document.getElementById("cam2").style.display = "none";
    document.getElementById("cam3").style.display = "none";
    document.getElementById("cam1").style.display = "block";
    document.getElementById("cabcameras").innerHTML = "Câmera 1";
};
menuCamera2 = function() {
    document.getElementById("cam1").style.display = "none";
    document.getElementById("cam3").style.display = "none";
    document.getElementById("cam2").style.display = "block";
    document.getElementById("cabcameras").innerHTML = "Câmera 2";
};
menuCamera3 = function() {
    document.getElementById("cam1").style.display = "none";
    document.getElementById("cam2").style.display = "none";
    document.getElementById("cam3").style.display = "block";
    document.getElementById("cabcameras").innerHTML = "Câmera 3";
};
menuConectar = function() {
    var id, desc, memory, indice, tagDIV, tagIMG, tagA;
    id = desc = memory = indice = tagDIV = tagIMG = tagA = null;
    var retorno = document.getElementById("retorno");
    var objAjax = new jfAjax();
    objAjax.url = url + "Conecta";
    objAjax.postSynchronouslyCall();
    // Verificar se o retorno do webservice foi null
    if (objAjax.getReturn() == null) {
        retorno.innerHTML = "Problema na conexão com a estação!!!";
        retorno.className = "nok";
    }
    else if (objAjax.getReturn().indexOf("NOK") > 0) {
        retorno.innerHTML = "Problema na conexão com a estação!!!";
        retorno.className = "nok";
    }
    else {
        // Cria um objeto para recuperar o xml
        objAjax.url = url + "MontaComandos";
        objAjax.postSynchronouslyCall();
        var objIO = new jfParseXML();
        var comandos = objIO.getXML(objAjax.getReturn());
        // Gera um array de Itens
        var arrayItem = comandos.getElementsByTagName("item");
        // Para cada item do array, recupera o ID, a DESC, e o MEMORY
        document.getElementById("listacomandos").innerHTML = "";
        for (var i = 0; i < arrayItem.length; i++) {
            // Para o memory, tem que pegar apenas a parte do nome que ficar após o ']'
            id = arrayItem[i].getAttribute("id");
            desc = arrayItem[i].getAttribute("desc");
            memory = arrayItem[i].getAttribute("memory");
            indice = memory.indexOf("]") + 1;
            memory = memory.substr(indice);
            // Cria div e faz className = "ok" e id = ID
            tagDIV = document.createElement("div");
            tagDIV.id = id;
```

```
            tagDIV.className = "nok";
            // Cria img e faz src = "./images/Ok.png"
            tagIMG = document.createElement("img");
            tagIMG.src = "./images/nok.png";
            // Cria a e faz href = "#", innerHTML = DESC, funcao = MEMORY, e onclick = funcao a definir
            tagA = document.createElement("a");
            tagA.href = "#";
            tagA.innerHTML = desc;
            tagA.funcao = memory;
            tagA.onclick = null;
            // div <- img e a
            tagDIV.appendChild(tagIMG);
            tagDIV.appendChild(tagA);
            // div 'listacomandos' <- div
            document.getElementById("listacomandos").appendChild(tagDIV);
        }
        objIO = id = desc = memory = tagDIV = tagIMG = tagA = null;

        tagDIV = document.createElement("div");
        tagDIV.funcao = "Setup";
        tagDIV.onclick = executaComando;
        tagDIV.onclick();
        tagDIV = null;
        retorno.innerHTML = "Conectado com sucesso!!!";
        retorno.className = "ok";
    }
    retorno.style.display = "block";
    timeOut1 = setTimeout("escondeMensagem()", 2000);

    // Faz flag de atualização = true
    flagAtualiza = true;
    // Chama função de atualização
    atualizaComandos();
};
menuDesconectar = function() {
    flagAtualiza = false;
    clearTimeout(timeOut1);
    document.getElementById("listacomandos").innerHTML = "";
    document.getElementById("retorno").style.display = "none";
};
atualizaComandos = function() {
    var id, value;
    id = value = null;
    var retorno = document.getElementById("retorno");
    var objAjax = new jfAjax();
    objAjax.url = url + "AtualizaComandos";
    objAjax.postSynchronouslyCall();
    // Verificar se o retorno do webservice foi null
    if (objAjax.getReturn() == null) {
        retorno.innerHTML = "Problema na conexão com a estação!!!";
        retorno.className = "nok";
    }
    else if (objAjax.getReturn().indexOf("NOK") > 0) {
        retorno.innerHTML = "Problema na conexão com a estação!!!";
        retorno.className = "nok";
    }
    else {
        var objIO = new jfParseXML();
        var comandos = objIO.getXML(objAjax.getReturn());
        // Gera um array de Itens
        var arrayItem = comandos.getElementsByTagName("item");
        // Para cada item do array, recupera o ID, e VALUE
        for (var i = 0; i < arrayItem.length; i++) {
            id = arrayItem[i].getAttribute("id");
            value = arrayItem[i].getAttribute("value");
            // Percorre os elementos e atualiza os IDs com OK e NOK, trocando as imagens, quando
necessário
            document.getElementById(id).className = value;
            document.getElementById(id).getElementsByTagName("img")[0].src = "./images/" + value +
".png";
            if (value == "ok") {
                document.getElementById(id).getElementsByTagName("a")[0].onclick = executaComando;
            }
            else {
```

```
                document.getElementById(id).getElementsByTagName("a")[0].onclick = null;
            }
        }
        objIO = id = value = null;
        retorno.style.display = "none";
    }

    if (flagAtualiza) {
        timeOut1 = setTimeout("atualizaComandos()", 1000);
    }
};
executaComando = function() {
    var retorno = document.getElementById("retorno");
    retorno.style.display = "none";
    retorno.className = "ok";
    retorno.innerHTML = "Comando executado com sucesso!"
    var objAjax = new jfAjax();
    objAjax.url = url + this.funcao;
    objAjax.postSynchronouslyCall();
    // Verificar se o retorno do webservice foi null
    if (objAjax.getReturn() == null) {
        retorno.innerHTML = "Problema no acionamento do webservice!!!";
        retorno.className = "nok";
    }
    else if (objAjax.getReturn().indexOf("NOK") > 0) {
        retorno.innerHTML = "Problema na execução do comando!!!";
        retorno.className = "nok";
    }
    retorno.style.display = "block";
    retorno = null;
    timeOut1 = setTimeout("escondeMensagem()", 2000);
};
escondeMensagem = function() {
    document.getElementById("retorno").style.display = "none";
}
// Menu - fim ------------------------------------------------------------------------
```

main.js

```
function click(event) {
    var evento = event;
    if (!evento) {
        evento = window.event;
    }
    if ((evento.button == 2) || (evento.button==3)) {
        oncontextmenu = "return false";
    }
}
document.onmousedown = click;
document.oncontextmenu = new Function("return false;");
```

web.config

```
<?xml version="1.0"?>
<configuration>
        <appSettings>
                <clear/>
                <add key="sigla" value="PMRLSA"/>
                <add key="nome" value="Laboratório de Sistemas de Automação"/>
    <add key="estacao" value="Transporte - TELEOPERADOR"/>
    <add key="camera1" value="192.168.0.195"/>
    <add key="camera2" value="192.168.0.194"/>
    <add key="camera3" value="192.168.0.193"/>
        </appSettings>
        <connectionStrings/>
        <system.web>
                <compilation debug="true"></compilation>
                <authentication mode="None"/>
        </system.web>
        <system.codedom></system.codedom>
        <system.webServer></system.webServer>
</configuration>
```

# ANEXO B – Subsistema de transporte: Programa em Ladder

O código a seguir foi desenvolvido no Simatic Manager, software da Siemens, em linguagem Ladder. O código implementa as funções de controle do controlador programável, CLP SIMATIC S7-300.

DB1: Instance data block for FB1

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR1_ENT | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR2_ENT | BOOL | FALSE | FALSE | |
| 0.2 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 2.0 | in | Temporizador | TIMER | T 0 | T 0 | |
| 4.0 | out | CIL_ENTR | BOOL | FALSE | FALSE | |

DB2: Instance data block for FB2

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | CIL_EST | BOOL | FALSE | FALSE | |
| 0.2 | in | CIL_REC | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB3: Instance data block for FB3

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | S_CIL_REC | BOOL | FALSE | FALSE | |
| 0.2 | in | S_CIL_EST | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB4: Instance data block for FB4

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR_SAIDA | BOOL | FALSE | FALSE | |
| 2.0 | out | A_CIL_SAIDA | BOOL | FALSE | FALSE | |

DB5: “ID_ESTACAO”

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | ID_EST1 | DWORD | DW#16#0 | ID_CARRO_ESTAÇÃO 1 |
| +4.0 | ID_EST2 | DWORD | DW#16#0 | ID_CARRO_ESTAÇÃO 2 |
| +8.0 | ID_EST3 | DWORD | DW#16#0 | ID_CARRO_ESTAÇÃO 3 |
| +12.0 | ID_EST4 | DWORD | DW#16#0 | ID_CARRO_ESTAÇÃO 4 |
| =16.0 | | END_STRUCT | | |

DB6: “ID_CARROS”

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | ID_CAR1 | DWORD | DW#16#A955550 | |
| +4.0 | ID_CAR2 | DWORD | DW#16#A955718 | |
| +8.0 | ID_CAR3 | DWORD | DW#16#B4E0892 | |
| +12.0 | ID_CAR4 | DWORD | DW#16#B257549 | |
| +16.0 | ID_CAR5 | DWORD | DW#16#B2F6875 | |
| =20.0 | | END_STRUCT | | |

DB7: “ID_COR”

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | COR_ID1 | INT | 0 | COR DO CARRO ID1 |
| +2.0 | COR_ID2 | INT | 0 | COR DO CARRO ID2 |
| +4.0 | COR_ID3 | INT | 0 | COR DO CARRO ID3 |
| +6.0 | COR_ID4 | INT | 0 | COR DO CARRO ID4 |
| +8.0 | COR_ID5 | INT | 0 | COR DO CARRO ID5 |
| =10.0 | | END_STRUCT | | |

DB8: “ID_PRODUTO”

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | ID1_PRODUTO_PRONTO | BOOL | FALSE | PRODUTO PRONTO NO CARRO ID1 |
| +0.1 | ID2_PRODUTO_PRONTO | BOOL | FALSE | PRODUTO PRONTO NO CARRO ID2 |
| +0.2 | ID3_PRODUTO_PRONTO | BOOL | FALSE | PRODUTO PRONTO NO CARRO ID3 |
| +0.3 | ID4_PRODUTO_PRONTO | BOOL | FALSE | PRODUTO PRONTO NO CARRO ID4 |
| +0.4 | ID5_PRODUTO_PRONTO | BOOL | FALSE | PRODUTO PRONTO NO CARRO ID5 |
| =2.0 | | END_STRUCT | | |

DB10: Global data block

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | res_SL1 | BYTE | B#16#0 | |
| +1.0 | SL2_SL3 | BYTE | B#16#0 | |
| +2.0 | SL4_SL5 | BYTE | B#16#0 | |
| +3.0 | SL6_SL7 | BYTE | B#16#0 | |
| +4.0 | SL8_SL9 | BYTE | B#16#0 | |
| +5.0 | SL10_SL11 | BYTE | B#16#0 | |
| +6.0 | SL12_SL13 | BYTE | B#16#0 | |
| +7.0 | SL14_SL15 | BYTE | B#16#0 | |
| +8.0 | SL16_SL17 | BYTE | B#16#0 | |
| +9.0 | SL18_SL19 | BYTE | B#16#0 | |
| +10.0 | SL20_SL21 | BYTE | B#16#0 | |
| +11.0 | SL22_SL23 | BYTE | B#16#0 | |
| +12.0 | SL24_SL25 | BYTE | B#16#0 | |
| +13.0 | SL26_SL27 | BYTE | B#16#0 | |
| +14.0 | SL28_SL29 | BYTE | B#16#0 | |
| +15.0 | SL30_SL31 | BYTE | B#16#0 | |
| =16.0 | | END_STRUCT | | |

DB20: Global data block

| Address | Name | Type | Initial value | Comment |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | res_SL1 | BYTE | B#16#0 | |
| +1.0 | SL2_SL3 | BYTE | B#16#0 | |
| +2.0 | SL4_SL5 | BYTE | B#16#0 | |
| +3.0 | SL6_SL7 | BYTE | B#16#0 | |
| +4.0 | SL8_SL9 | BYTE | B#16#0 | |
| +5.0 | SL10_SL11 | BYTE | B#16#0 | |
| +6.0 | SL12_SL13 | BYTE | B#16#0 | |
| +7.0 | SL14_SL15 | BYTE | B#16#0 | |
| +8.0 | SL16_SL17 | BYTE | B#16#0 | |
| +9.0 | SL18_SL19 | BYTE | B#16#0 | |
| +10.0 | SL20_SL21 | BYTE | B#16#0 | |
| +11.0 | SL22_SL23 | BYTE | B#16#0 | |
| +12.0 | SL24_SL25 | BYTE | B#16#0 | |
| +13.0 | SL26_SL27 | BYTE | B#16#0 | |
| +14.0 | SL28_SL29 | BYTE | B#16#0 | |
| +15.0 | SL30_SL31 | BYTE | B#16#0 | |
| =16.0 | | END_STRUCT | | |

DB21: Instance data block for FB1

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR1_ENT | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR2_ENT | BOOL | FALSE | FALSE | |
| 0.2 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 2.0 | in | Temporizador | TIMER | T 0 | T 0 | |
| 4.0 | out | CIL_ENTR | BOOL | FALSE | FALSE | |

DB22: Instance data block for FB2

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | CIL_EST | BOOL | FALSE | FALSE | |
| 0.2 | in | CIL_REC | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB23: Instance data block for FB3

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | S_CIL_REC | BOOL | FALSE | FALSE | |
| 0.2 | in | S_CIL_EST | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB24: Instance data block for FB4

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR_SAIDA | BOOL | FALSE | FALSE | |
| 2.0 | out | A_CIL_SAIDA | BOOL | FALSE | FALSE | |

DB31: Instance data block for FB1

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR1_ENT | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR2_ENT | BOOL | FALSE | FALSE | |
| 0.2 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 2.0 | in | Temporizador | TIMER | T 0 | T 0 | |
| 4.0 | out | CIL_ENTR | BOOL | FALSE | FALSE | |

DB32: Instance data block for FB2

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | CIL_EST | BOOL | FALSE | FALSE | |
| 0.2 | in | CIL_REC | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB33: Instance data block for FB3

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | S_CIL_REC | BOOL | FALSE | FALSE | |
| 0.2 | in | S_CIL_EST | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB34: Instance data block for FB4

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR_SAIDA | BOOL | FALSE | FALSE | |
| 2.0 | out | A_CIL_SAIDA | BOOL | FALSE | FALSE | |

DB41: Instance data block for FB1

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR1_ENT | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR2_ENT | BOOL | FALSE | FALSE | |
| 0.2 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 2.0 | in | Temporizador | TIMER | T 0 | T 0 | |
| 4.0 | out | CIL_ENTR | BOOL | FALSE | FALSE | |

DB42: Instance data block for FB2

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | CIL_EST | BOOL | FALSE | FALSE | |
| 0.2 | in | CIL_REC | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

DB43: Instance data block for FB3

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | S_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | S_CIL_REC | BOOL | FALSE | FALSE | |
| 0.2 | in | S_CIL_EST | BOOL | FALSE | FALSE | |
| 2.0 | out | CIL_TRAVA | BOOL | FALSE | FALSE | |

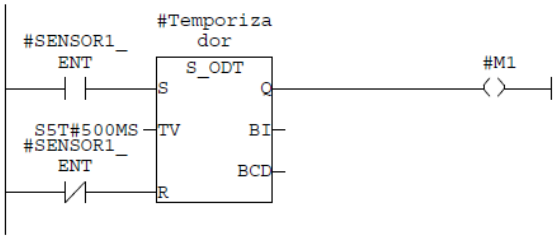
DB44: Instance data block for FB4

| Address | Declaration | Name | Type | Initial value | Actual value | Comment |
|---|---|---|---|---|---|---|
| 0.0 | in | SENSOR_EST | BOOL | FALSE | FALSE | |
| 0.1 | in | SENSOR_SAIDA | BOOL | FALSE | FALSE | |
| 2.0 | out | A_CIL_SAIDA | BOOL | FALSE | FALSE | |

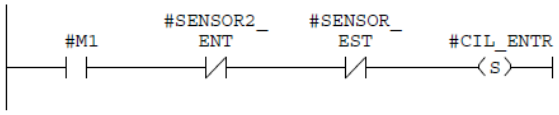
## FB1: “FUNCAO_ENTRADA”

Network: 1 ABAIXA O CILINDRO DE ENTRADA

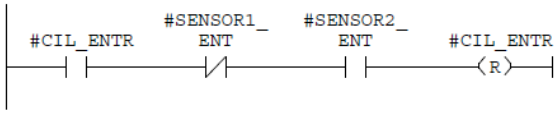
QUANDO EXISTIR UM CARRO NA ENTRADA DA ESTAÇÃO, ABAIXAR O CILINDRO DE ENTRADA
PARA PERMITIR A ENTRADA DO CARRINHO




Network: 2




Network: 3 LEVANTA O CILINDRO DE ENTRADA



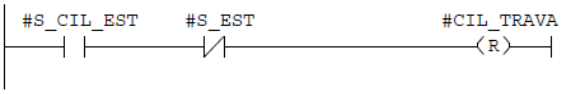

## FB2: “FUNCAO_TRAVA”

Network: 1 TRAVA DO CARRINHO NA ESTAÇÃO

ACIONAR O CILINDRO DE TRAVA DO CARRINHO QUANDO:
O SENSOR DA ESTAÇÃO ESTIVER ATUADO & O SENSOR DE CILINDRO RECUADO ESTIVER ATUADO
& O SENSOR DE CILINDRO ESTENTIDO NÃO ESTIVER ATUADO

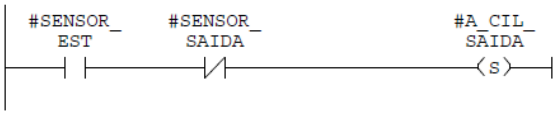

## FB3: “FUNCAO_DESTRAVA”

Network: 1

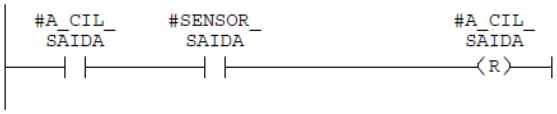


FB4: “FUNCAO_DE_SAIDA”

Network: 1



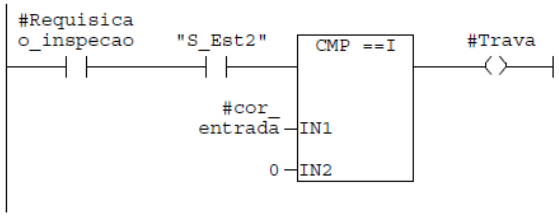

Network: 2




FC1: “Funcionamento_Estacao 2”

Network: 1 FIXAÇÃO DO CARRINHO VAZIO NA ESTAÇÃO 2

Caso seja requisitado um carrinho e o carrinho esteja vazio, então trava o carrinho na estação 2

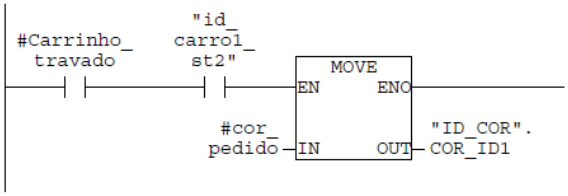



Network: 2

SINALIZAÇÃO PARA O WEB SERVER QUE O CARRINHO ESTÁ TRAVADO NA EST.2

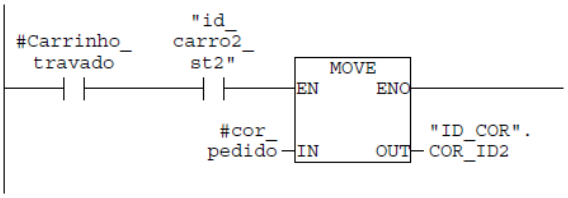

Network: 3

ESTANDO O CARRINHO ID1 TRAVADO, ATUALIZA A COR DO CORPO NO CARRINHO ID1



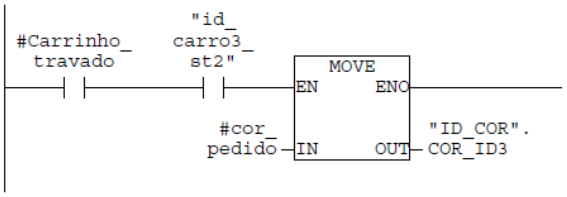
Network: 4

ESTANDO O CARRINHO ID2 TRAVADO, ATUALIZA A COR DO CORPO NO CARRINHO ID2



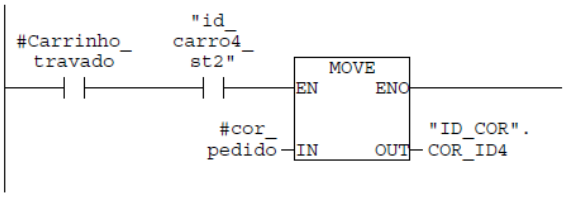

Network: 5

ESTANDO O CARRINHO ID3 TRAVADO, ATUALIZA A COR DO CORPO NO CARRINHO ID3



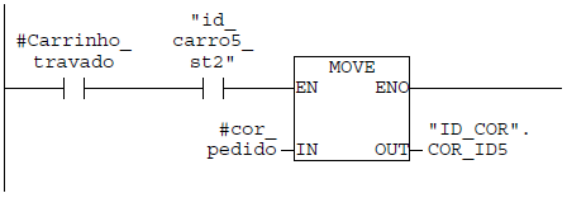

Network: 6

ESTANDO O CARRINHO ID4 TRAVADO, ATUALIZA A COR DO CORPO NO CARRINHO ID4




Network: 7

ESTANDO O CARRINHO ID5 TRAVADO, ATUALIZA A COR DO CORPO NO CARRINHO ID5




Network: 8

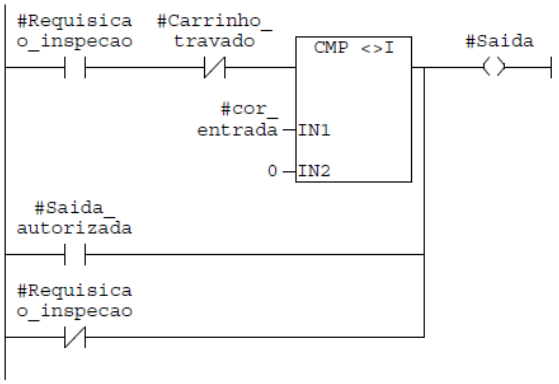
Quando for enviado um comando de saída, destrava o carrinho da estação 2


Network: 9

SINALIZA PARA O WS DE TRANSPORTE QUE O CARRINHO ESTÁ DESTRAVADO

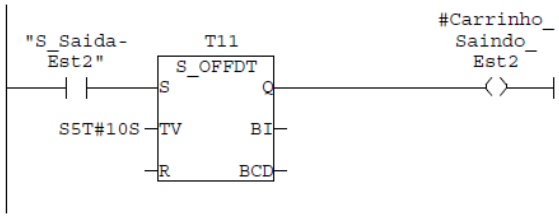
Network: 10 LIBERAÇÃO DO CARRINHO CARREGADO DA ESTAÇÃO 2

Quando Saida_autorizada =1 --> Destrava e libera a saída do carrinho;
Quando não tem requisição --> Libera a saída de qualquer carrinho da estação 2




Network: 11

SINALIZA PARA O WS DE TRANSPORTE QUE O CARRINHO ESTÁ SAINDO DA ESTAÇÃO 2




Network: 12 informa que o carrinho esta na estacao 2

Retorna status de requisição de carrinho para o WS de Transporte

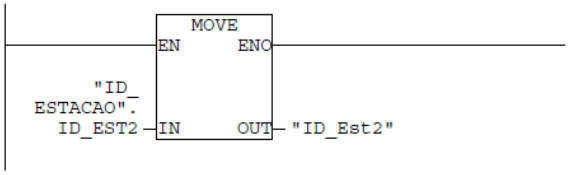

FC2: “Pesquisa_Estacao 2”

Identificação do ID do carrinho que está chegando na estação 2 (inspeção)

---

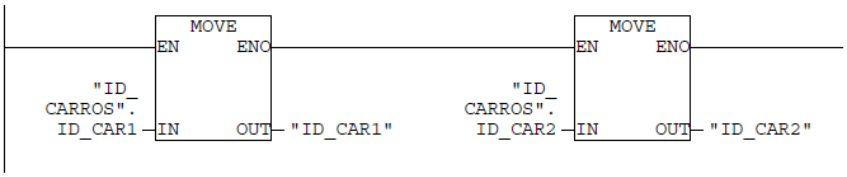
Network: 1

CONVERSÃO DO ID DA ESTAÇÃO 2 DE DWORD PARA INTEIRO DE 32 BITS
(preparando sinal para comparação)



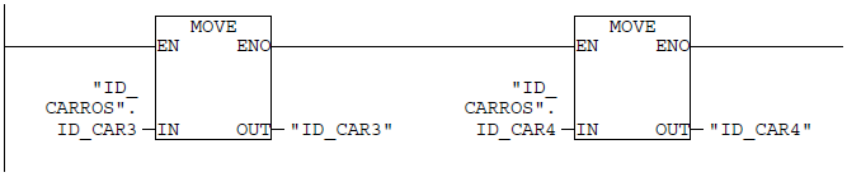
Network: 2

CONVERSÃO DOS ID_FIXOS DOS CARROS PARA INTEIRO DE 32 BITS, COM PROPÓSITO DE COMPARAÇÃO.



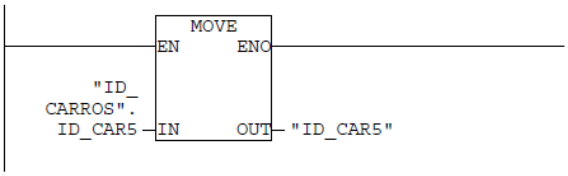

Network: 3

CONVERSÃO DOS ID_FIXOS DOS CARROS PARA INTEIRO DE 32 BITS, COM PROPÓSITO DE COMPARAÇÃO.



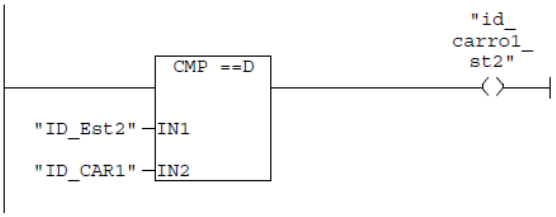

Network: 4

CONVERSÃO DOS ID_FIXOS DOS CARROS PARA INTEIRO DE 32 BITS, COM PROPÓSITO DE COMPARAÇÃO.



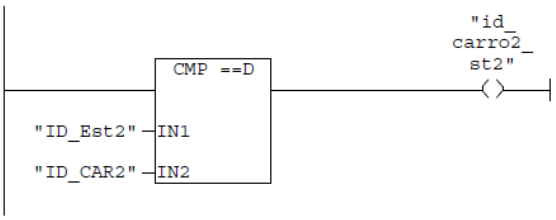

Network: 5

IDENTIFICA QUE O CARRINHO ID1 ESTÁ NA ESTAÇÃO 2



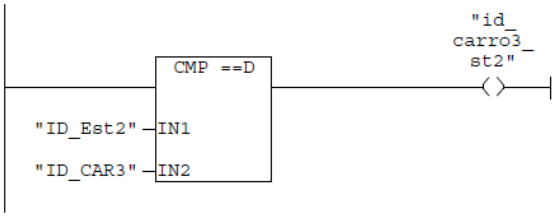

Network: 6

IDENTIFICA QUE O CARRINHO ID2 ESTÁ NA ESTAÇÃO 2



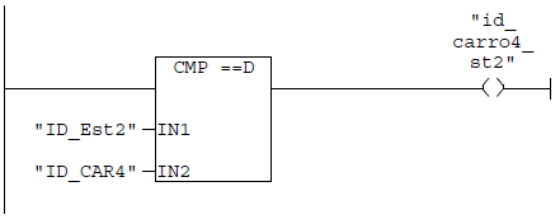
Network: 7

IDENTIFICA QUE O CARRINHO ID3 ESTÁ NA ESTAÇÃO 2



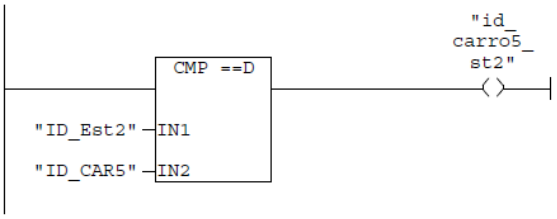

Network: 8

IDENTIFICA QUE O CARRINHO ID4 ESTÁ NA ESTAÇÃO 2



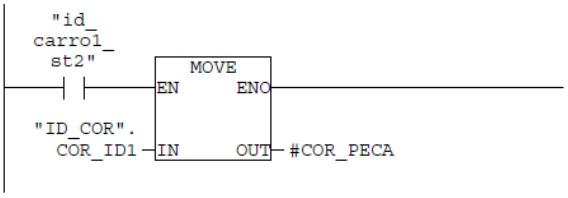

Network: 9

IDENTIFICA QUE O CARRINHO ID5 ESTÁ NA ESTAÇÃO 2



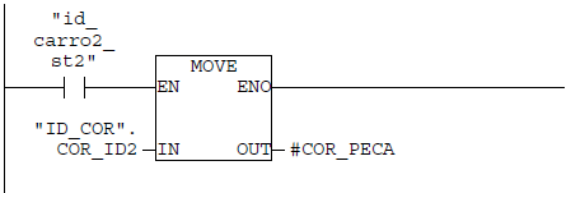

Network: 10

ALTERA A VARIÁVEL COR_ENTRADA COM A COR DA PEÇA ARMAZENADA NO BANCO DE DADOS



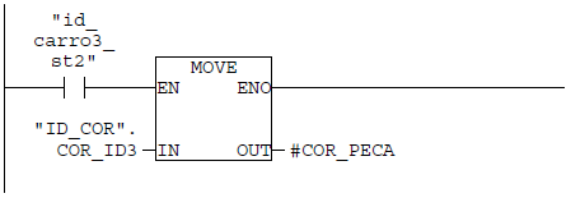

Network: 11

ALTERA A VARIÁVEL COR_ENTRADA COM A COR DA PEÇA ARMAZENADA NO BANCO DE DADOS



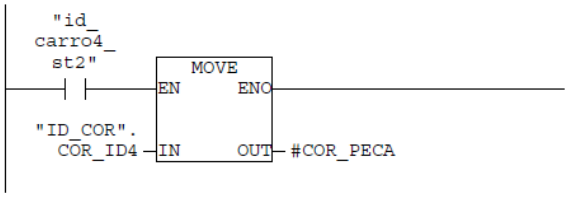
Network: 12

ALTERA A VARIÁVEL COR_ENTRADA COM A COR DA PEÇA ARMAZENADA NO BANCO DE DADOS



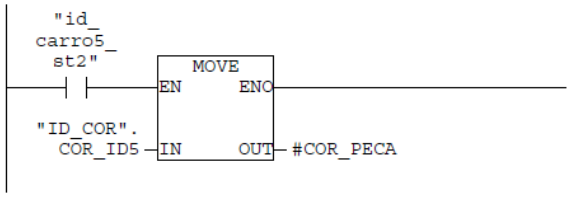

Network: 13

ALTERA A VARIÁVEL COR_ENTRADA COM A COR DA PEÇA ARMAZENADA NO BANCO DE DADOS




Network: 14

ALTERA A VARIÁVEL COR_ENTRADA COM A COR DA PEÇA ARMAZENADA NO BANCO DE DADOS



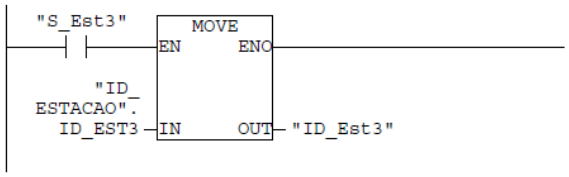

FC3: “Funcionamento_Estação 3”

Funcionamento da estação 3

---

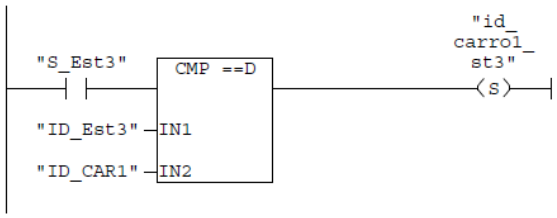
Network: 1

QUANDO UM CARRO CHEGA NA ESTAÇÃO 3, O CONTROLADOR LOCAL RECEBE DO WS TRANSPORTE
O ENDEREÇO DO CARRO DA ESTAÇÃO E CONVERTE ESTE ENDEREÇO PARA DOUBLE INTEGER.



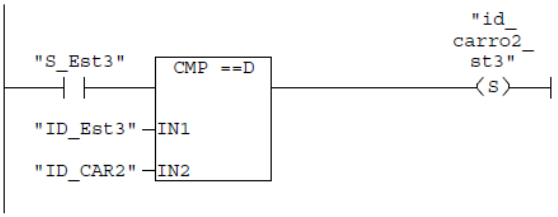

Network: 2

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 3 E O ID1



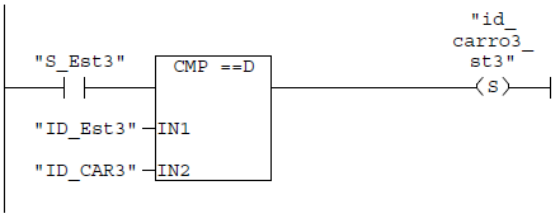
Network: 3

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 3 E O ID2



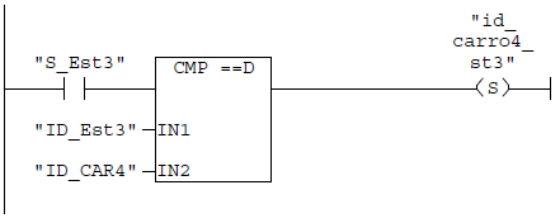

Network: 4

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 3 E O ID3



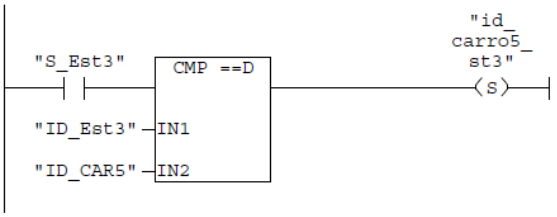

Network: 5

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 3 E O ID4

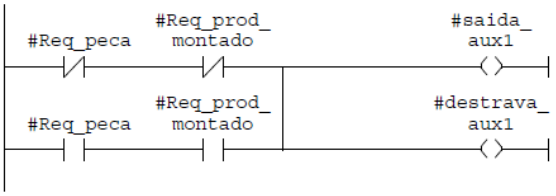



Network: 6

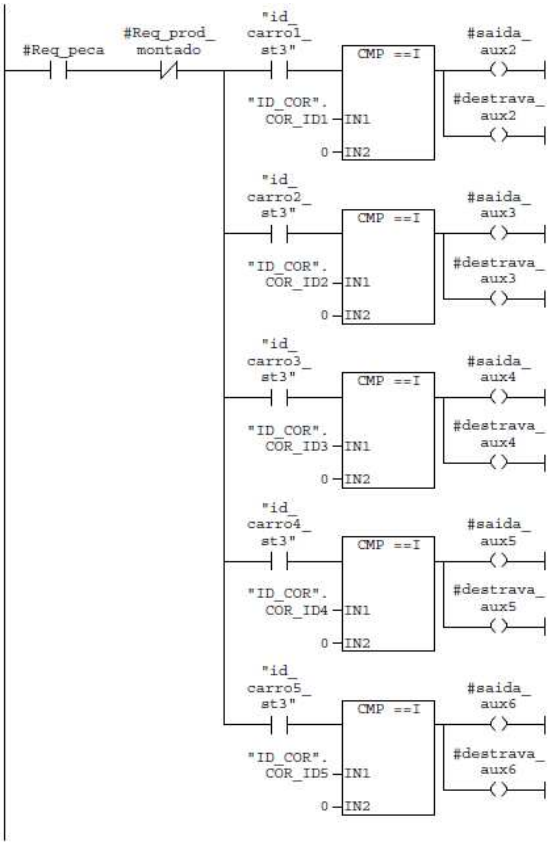
IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 3 E O ID5



| Network: 7 |
| --- |
| Se NÃO existirem requisições vindo do WS MONTAGEM --> libera saída da ESTAÇÃO 3<br>Se existirem 02 requisições simultâneas vindas do WS -> libera saída da ESTAÇÃO 3 |

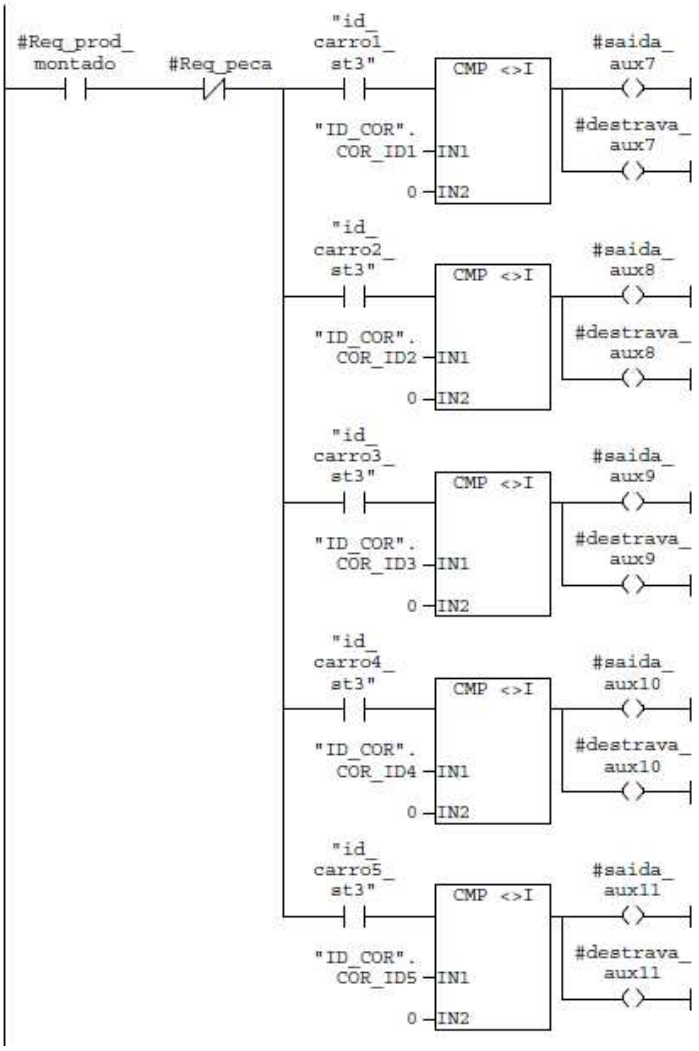



| Network: 8 ID_CARRO_ESTAÇÃO 3 |
| --- |
| SE REQ_PECA ==1 E CARRO ESTIVER VAZIO --> LIBERA SAÍDA DA ESTAÇÃO 3 |

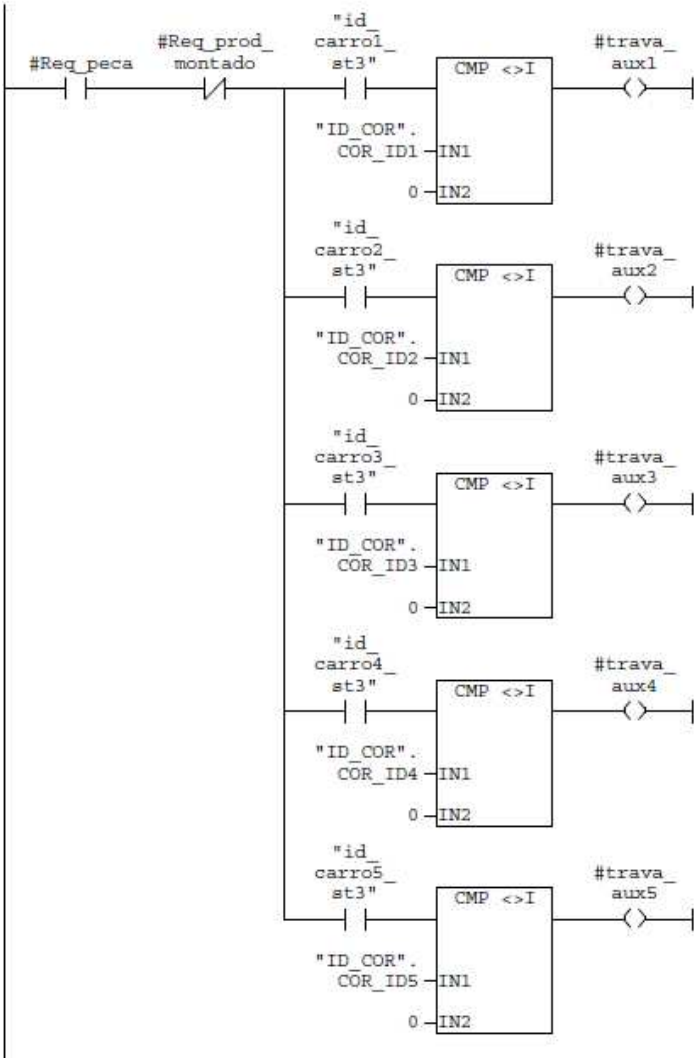


Network: 9 ID_CARRO_ESTAÇÃO 3

SE REQ_PRODUTO_MONTADO ==1 E CARRO CONTÉM PEÇA --> LIBERA SAÍDA DA ESTAÇÃO 3

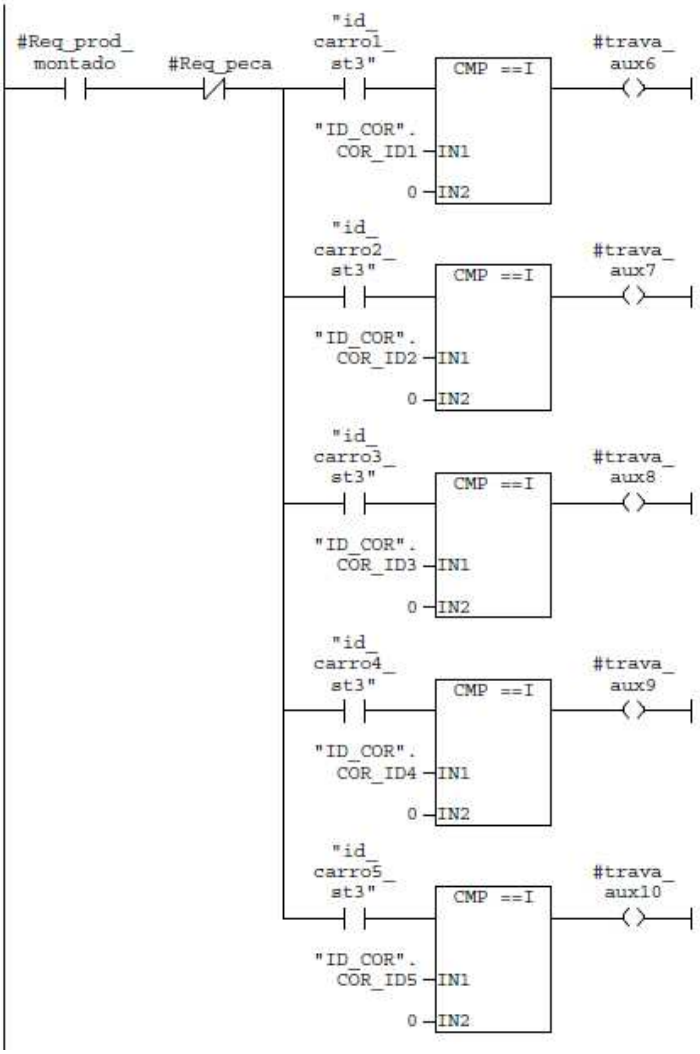


| Network: 10 informa que o carrinho esta saindo da estacao 3 |
| --- |
| SE REQ_PEÇA ==1 E CARRINHO TEM PEÇA --> TRAVA CARRINHO NA ESTAÇÃO 3 |



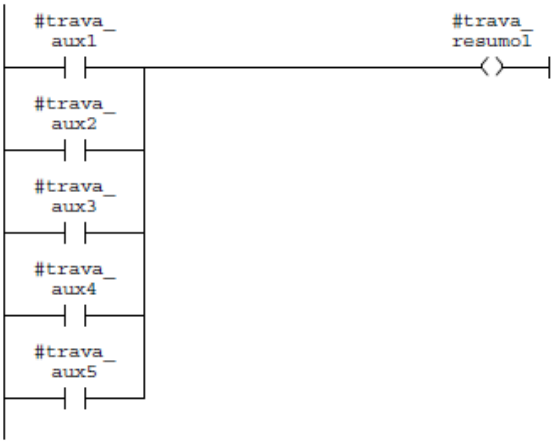
Network: 11

SE REQ_PROD_MONTADO ==1 E CARRINHO ESTÁ VAZIO --> TRAVA CARRINHO NA ESTAÇÃO 3



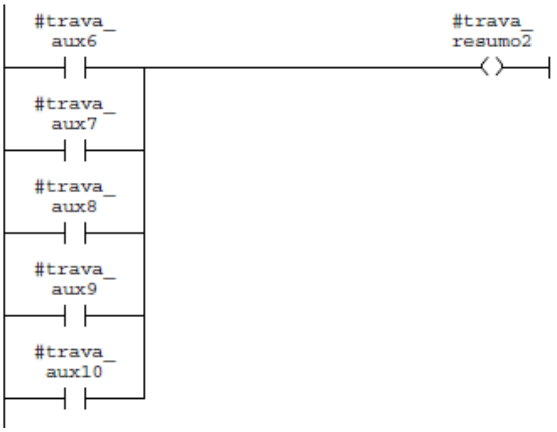
Network: 12

TRAVA RESUMO 1



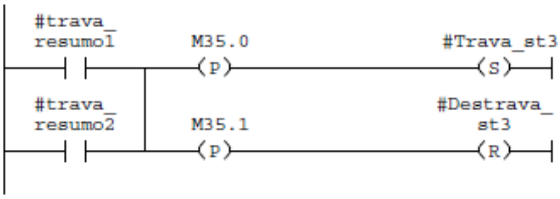

Network: 13

TRAVA RESUMO 2

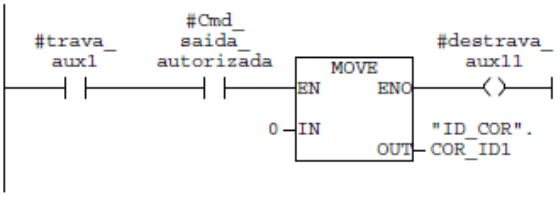



Network: 14

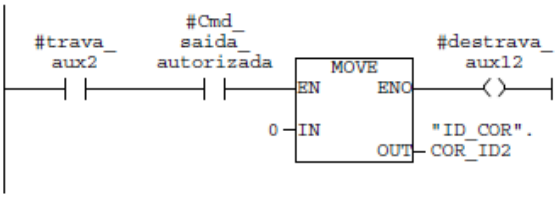
TRAVA ESTAÇÃO 3



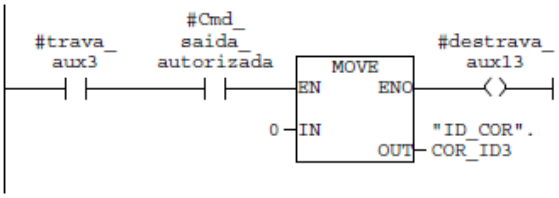

Network: 15 Informa que o cilindro da estação 3 esta recuado



Network: 16



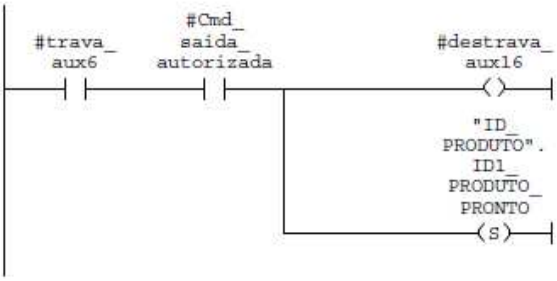

Network: 17



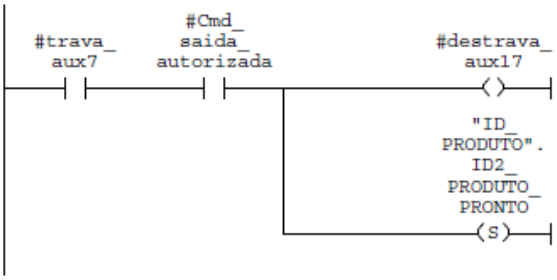

Network: 18


Network: 19

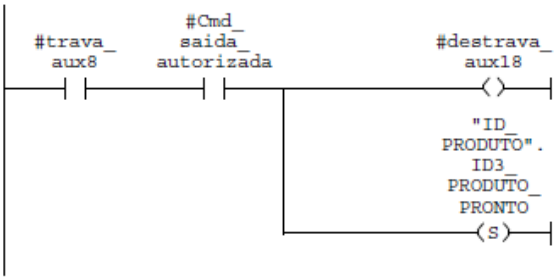

Network: 20



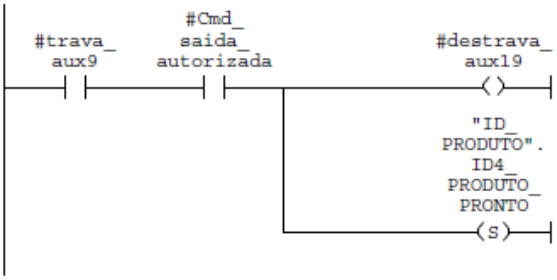
Network: 21



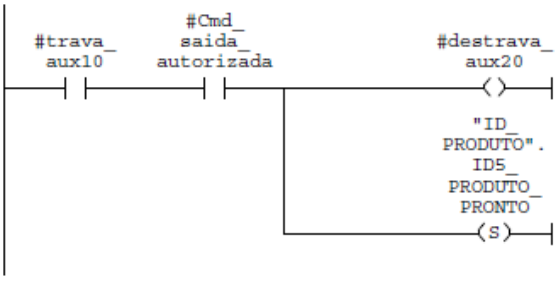

Network: 22




Network: 23

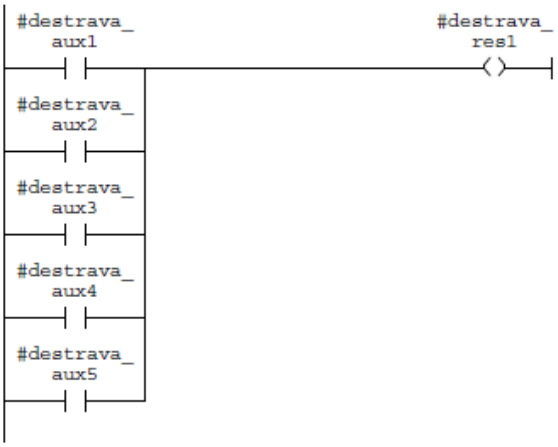



Network: 24



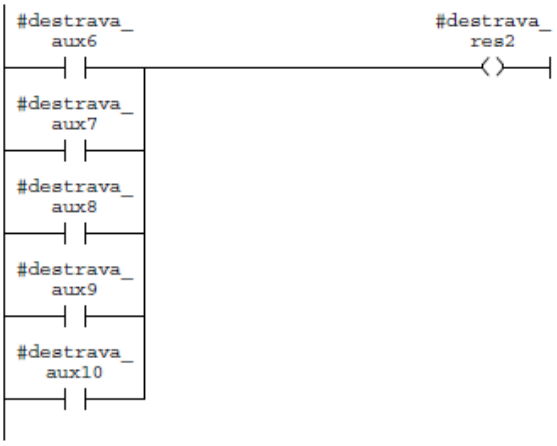
Network: 25

RESUMO 1 DE DESTRAVA



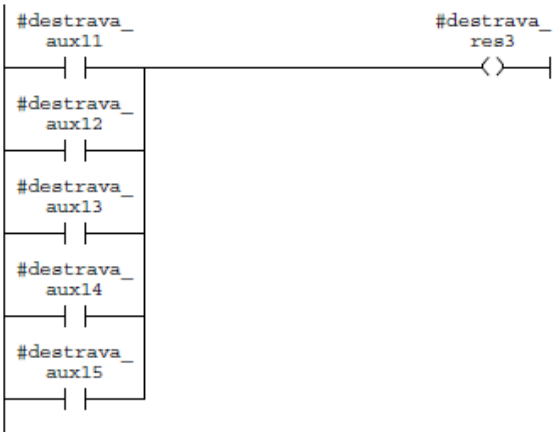

Network: 26

RESUMO 2 DE DESTRAVA



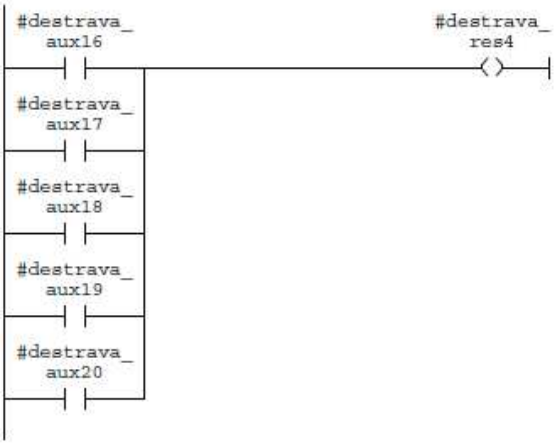

Network: 27

RESUMO 3 DE DESTRAVA



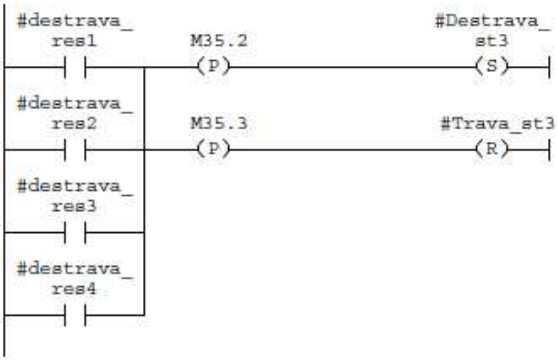
Network: 28

RESUMO 4 DE DESTRAVA



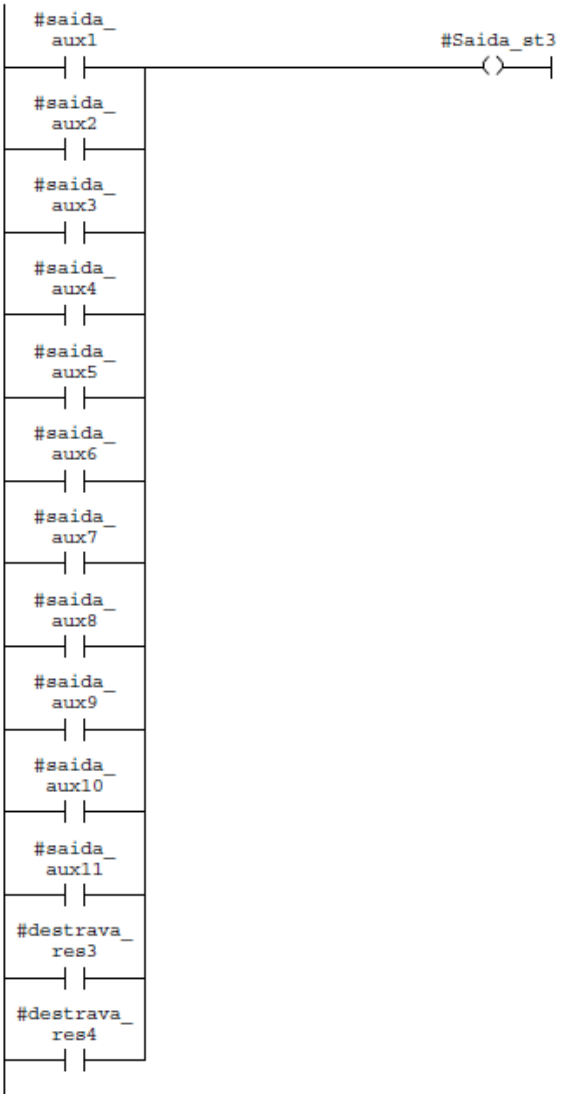

Network: 29 informa que o carrinho esta saindo da estacao 3

DESTRAVA ESTAÇÃO 3



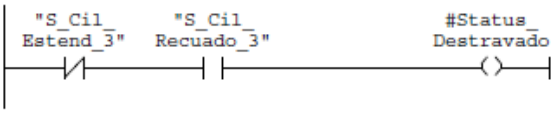
Network: 30 ID_CARRO_ESTAÇÃO 3

LIBERA SAÍDA DA ESTAÇÃO 3

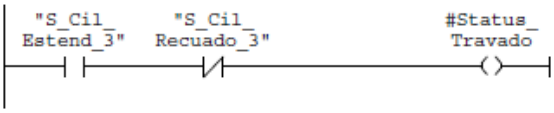



Network: 31

SINALIZA QUE A ESTAÇÃO 3 ESTÁ DESTRAVADA

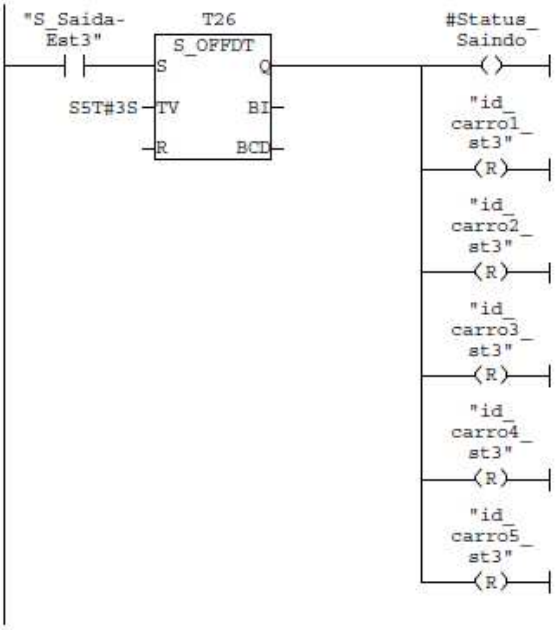



Network: 32



Network: 33

SINALIZA QUE O CARRO ESTÁ SAÍNDO DA ESTAÇÃO 3




FC4: “Funcionamento_Estação 4”

Funcionamento da estação 4

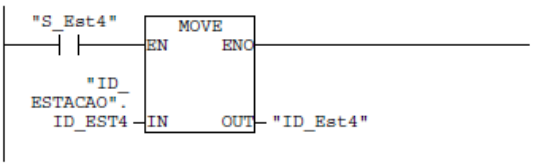
---

**Block: FC4**

FUNCIONAMENTO DA ESTAÇÃO 4

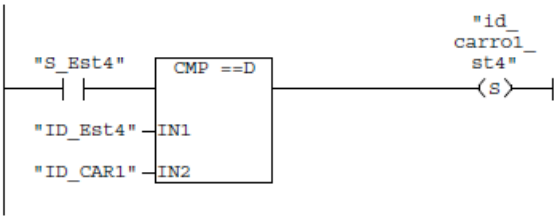
Network: 1

QUANDO UM CARRO CHEGA NA ESTAÇAO 4, O CONTROLADOR LOCAL RECEBE DO WS TRANSPORTE
O ENDEREÇO DO CARRO DA ESTAÇÃO E CONVERTE PARA DOUBLE
INTEGER



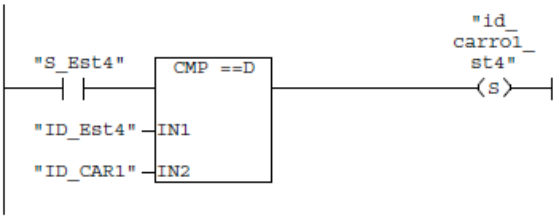

Network: 2

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 4 É O ID1



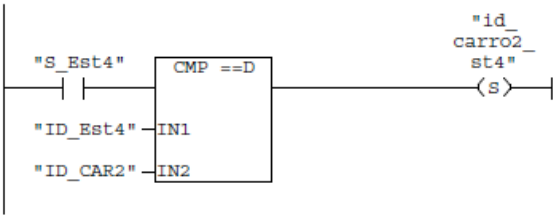
Network: 2

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 4 É O ID1



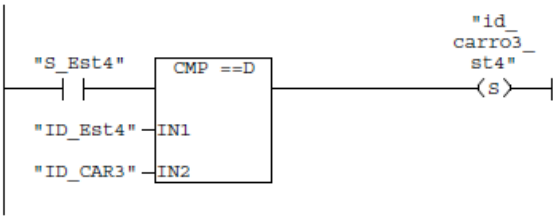

Network: 3

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 4 É O ID2



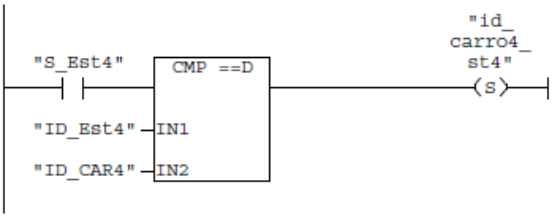

Network: 4

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 4 É O ID3



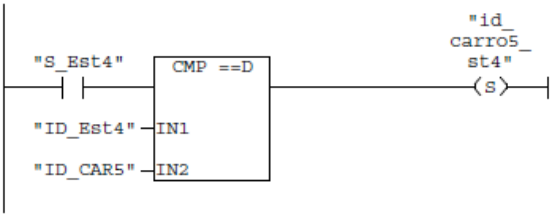

Network: 5

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 4 É O ID4



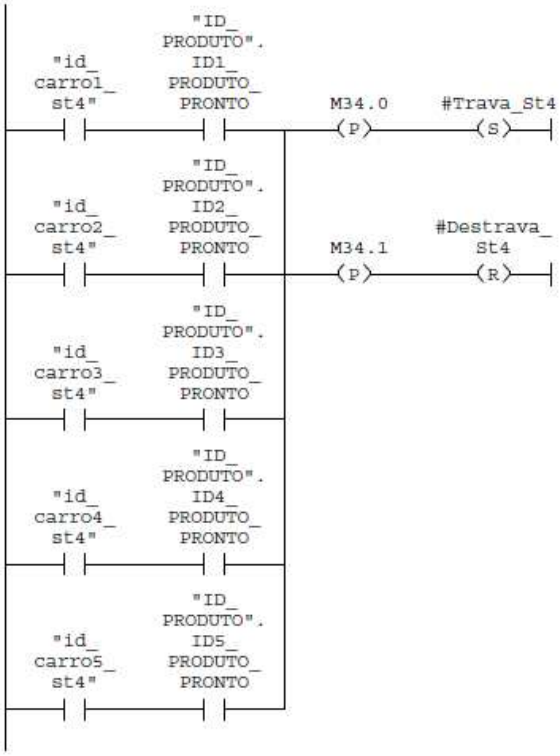

Network: 6

IDENTIFICA SE O CARRO QUE ESTÁ NA ESTAÇÃO 4 É O ID5



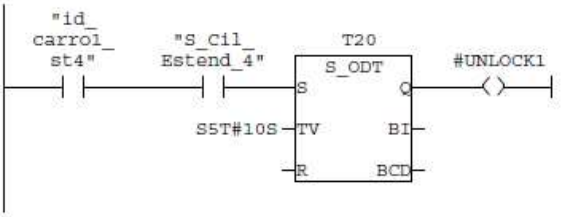
Network: 7 informa que o carrinho esta na estacao 4

CARRO NA ESTAÇÃO 4 COM PRODUTO ACABADO
TRAVA O CARRO NA ESTAÇÃO



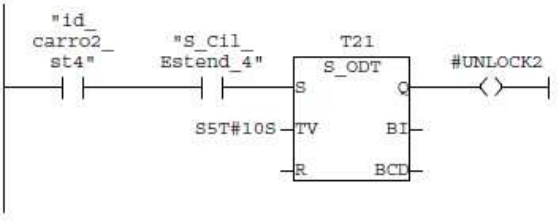

Network: 8

CARRO ID1 NA ESTAÇÃO 4 COM PRODUTO ACABADO
TEMPORIZA 10s e DESTRAVA o CARRO DA ESTAÇÃO 4



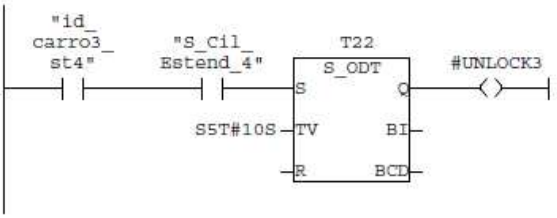

Network: 9

CARRO ID2 NA ESTAÇÃO 4 COM PRODUTO ACABADO
TEMPORIZA 10s e DESTRAVA o CARRO DA ESTAÇÃO 4



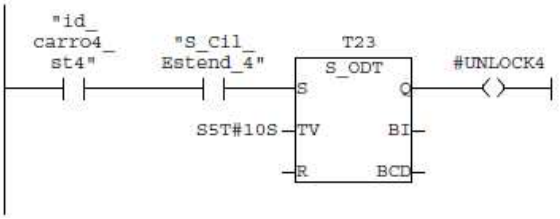
Network: 10

CARRO ID3 NA ESTAÇÃO 4 COM PRODUTO ACABADO
TEMPORIZA 10s e DESTRAVA o CARRO DA ESTAÇÃO 4



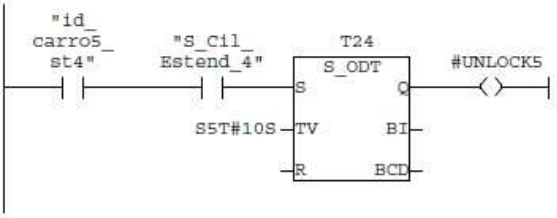

Network: 11

CARRO ID4 NA ESTAÇÃO 4 COM PRODUTO ACABADO
TEMPORIZA 10s e DESTRAVA o CARRO DA ESTAÇÃO 4



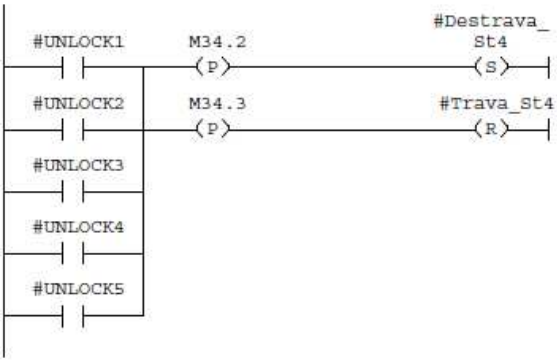

Network: 12

CARRO ID4 NA ESTAÇÃO 4 COM PRODUTO ACABADO
TEMPORIZA 10s e DESTRAVA o CARRO DA ESTAÇÃO 4



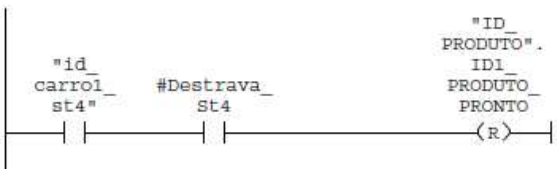

Network: 13

DESTRAVA ESTAÇÃO 4 PARA PERMITIR SAÍDA DO CARRO DA ESTAÇÃO

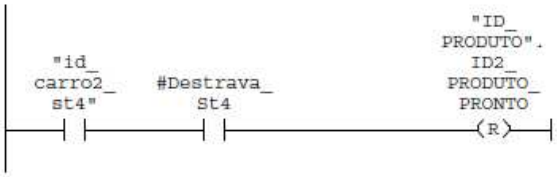



Network: 14 PRODUTO PRONTO NO CARRO ID1

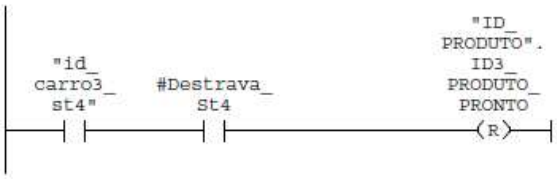
ZERA BANCO DE DADOS DE PRODUTO PRONTO DO CARRO ID1



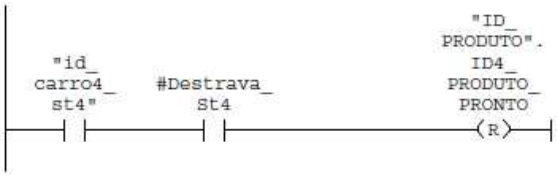
| Network: 15 PRODUTO PRONTO NO CARRO ID1 |
| --- |
| ZERA BANCO DE DADOS DE PRODUTO PRONTO DO CARRO ID2 |



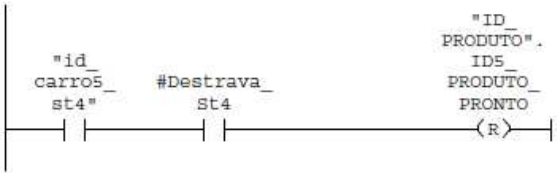

| Network: 16 PRODUTO PRONTO NO CARRO ID1 |
| --- |
| ZERA BANCO DE DADOS DE PRODUTO PRONTO DO CARRO ID3 |




| Network: 17 PRODUTO PRONTO NO CARRO ID1 |
| --- |
| ZERA BANCO DE DADOS DE PRODUTO PRONTO DO CARRO ID4 |

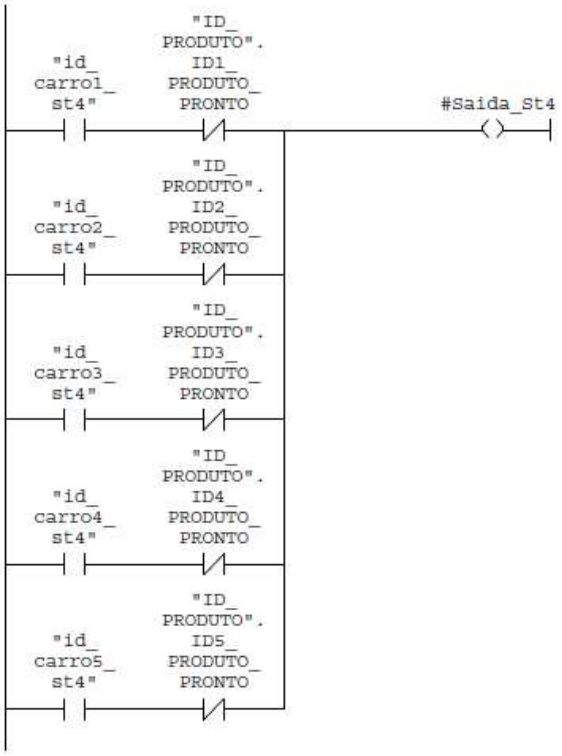



| Network: 18 PRODUTO PRONTO NO CARRO ID1 |
| --- |
| ZERA BANCO DE DADOS DE PRODUTO PRONTO DO CARRO ID5 |



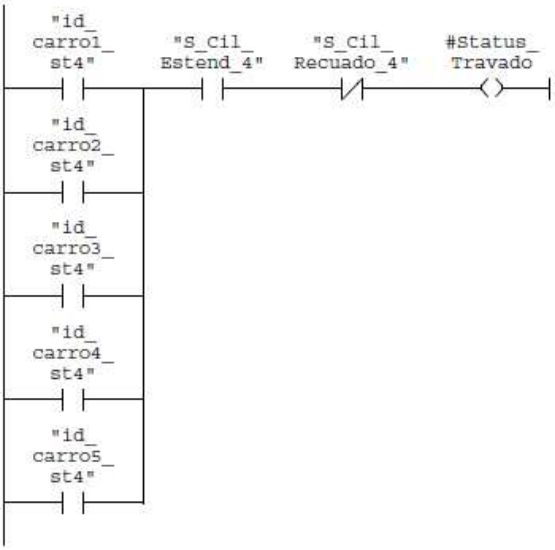
Network: 19

ZERA BANCO DE DADOS DE PRODUTO PRONTO DO CARRO ID5



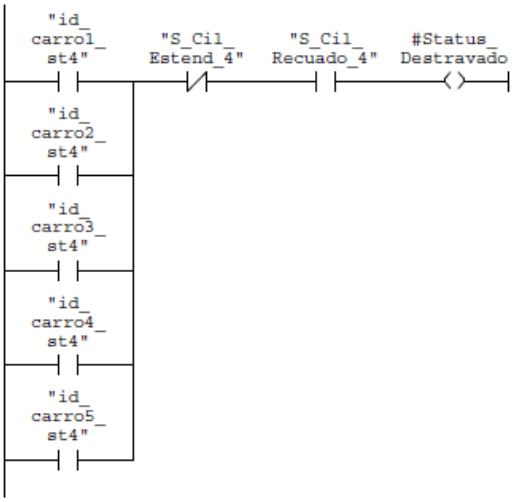

Network: 20

SINALIZA QUE O CARRO ESTÁ TRAVADO NA ESTAÇÃO 4



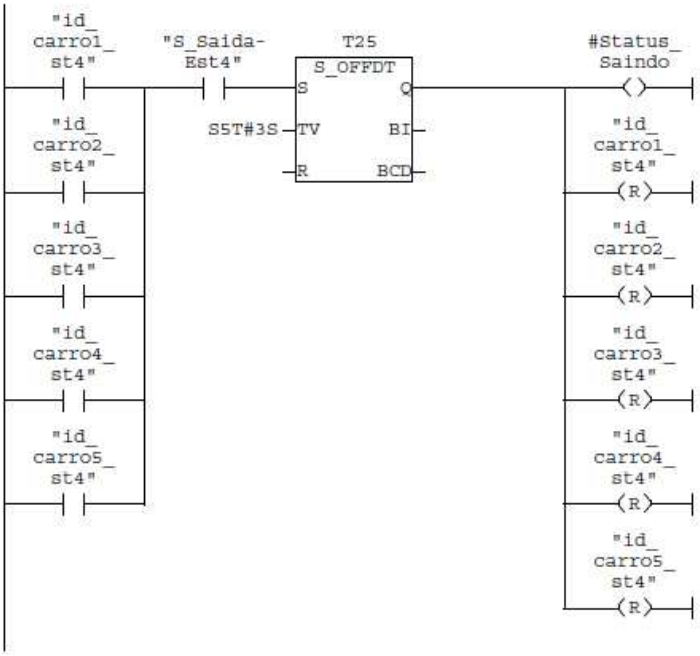
Network: 21

SINALIZA QUE O CARRO ESTÁ DESTRAVADO NA ESTAÇÃO 4




Network: 22

SINALIZA QUE O CARRO ESTÁ SAINDO DA ESTAÇÃO 4




FC10: “fcLeituraASI”

Leitura de todos os sensores na rede ASI

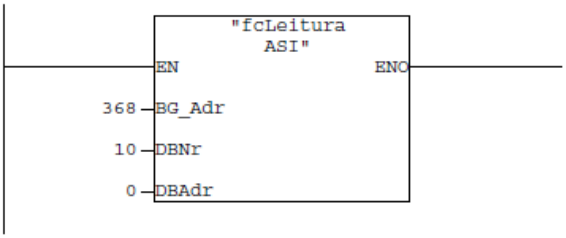
```
Network: 1

      L     #BG_Adr          // Baugruppenadresse lokal speichern
      SLD   3                // Auf Pointerformat anpassen
      T     #L_BGAdr

      L     #DBNr            // Datenbausteinnummer lokal speichern
      T     #L_DBNr

      L     #DBAdr           // Datenbaustein Adresse lokal speichern
      SLD   3                // Auf Pointerformat anpassen
      T     #L_DBAdr

      L     3                // Schleifenzähler laden
next: T     #L_Zael
      L     PID [#L_BGAdr]   // Peripherie laden

      OPN   DB [#L_DBNr]     // DB aufrufen
      T     DBD [#L_DBAdr]   // In DB ablegen


      L     #L_BGAdr         // Baugruppenadresse laden und
      SRD   3                // auf Zahlenformat anpassen
      L     4
      +I                     // um 4 erhöhen da Doppelwort
      SLD   3                // Auf Pointerformat anpassen
      T     #L_BGAdr

      L     #L_DBAdr         // Datenbausteinadresse laden und
      SRD   3                // auf Zahlenformat anpassen
      L     4
      +I                     // um 4 erhöhen
      SLD   3                // Auf Pointerformat anpassen
      T     #L_DBAdr

      L     #L_Zael
      LOOP  next             // Sprung zu Sprungmarke next
```

FC11: “fcLeitura”

Mapeamento dos sensores na memória interna

---

```
Network: 1      Lê os sensores na rede ASI
```



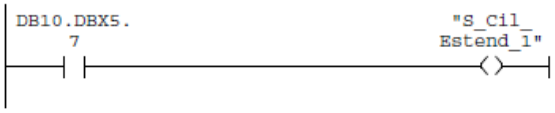

```
Network: 2      informa que o carrinho esta na entrada da estacao 1
```


```
Network: 3      informa que o carrinho esta entrando na estacao 1
```

Network: 4 informa que o carrinho esta na estacao 1


Network: 5 Informa que o cilindro da estação 1 esta recuado

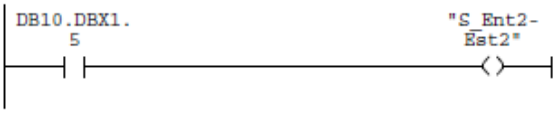

Network: 6 Informa que o cilindro da estação 1 esta estendido




Network: 7 informa que o carrinho esta saindo da estacao 1


Network: 8 informa que o carrinho esta na entrada da estacao 2


Network: 9 informa que o carrinho esta entrando na estacao 2




Network: 10 informa que o carrinho esta na estacao 2


Network: 11 Informa que o cilindro da estação 2 esta estendido

Network: 12 Informa que o cilindro da estação 2 esta recuado


Network: 13 informa que o carrinho esta saindo da estacao 2


Network: 14 informa que o carrinho esta na entrada da estacao 3

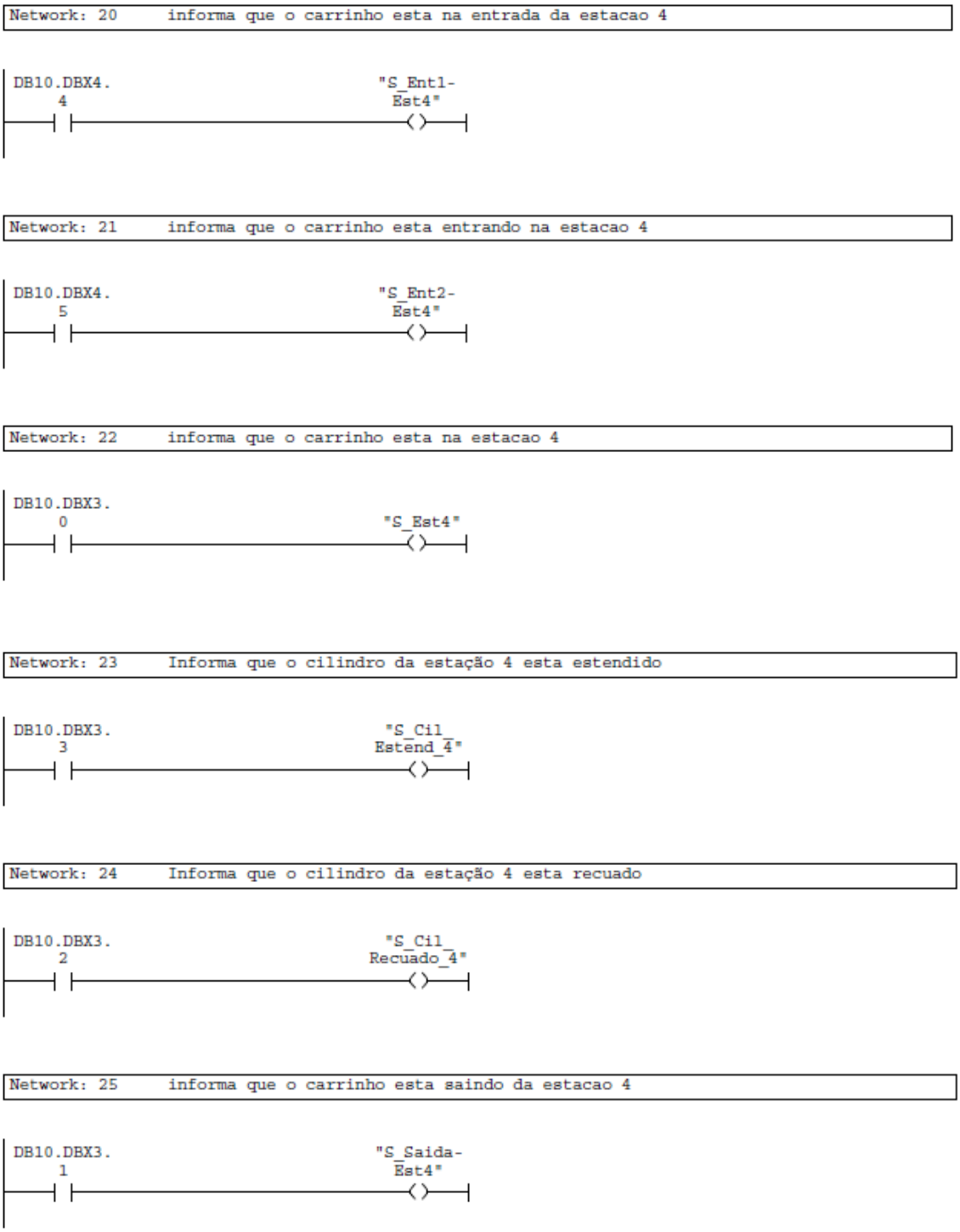

Network: 15 informa que o carrinho esta entrando na estacao 3


Network: 16 informa que o carrinho esta na estacao 3


Network: 17 Informa que o cilindro da estação 3 esta recuado


Network: 18 Informa que o cilindro da estação 3 esta estendido


Network: 19 informa que o carrinho esta saindo da estacao 3




FC20: “fcEscritaASI”

Escrita nos atuadores na rede ASI

```
Block: FC20
Baustein zum schreiben der Slavedaten zum ASI CP 342-2
```

```
Network: 1
```

```
      L     #DBadr          // Datenbaustein Adresse lokal speichern
      SLD   3               // Auf Pointerformat anpassen
      T     #L_DBAdr

      L     #DBNr           // Datenbausteinnummer lokal speichern
      T     #L_DBNr

      L     #BG_Adr         // Baugruppenadresse lokal speichern
      SLD   3               // Auf Pointerformat anpassen
      T     #L_BGAdr

      L     4
next: T     #L_Zael         // Schleifenzähler laden


      OPN   DB [#L_DBNr]    // DB aufrufen
      L     DBD [#L_DBAdr]  // Daten aus DBladen


      T     PQD [#L_BGAdr]  // Zur Peripherie schreiben

      L     #L_DBAdr        // Datenbausteinadresse laden und
      SRD   3               // Auf Zahlenformat anpassen
      L     4
      +I                    // um 4 erhöhen
      SLD   3               // Auf Pointerformat anpassen
      T     #L_DBAdr

      L     #L_BGAdr        // Baugruppenadresse laden und
      SRD   3               // Auf Zahlenformat anpassen
      L     4
      +I                    // um 4 erhöhen, da Doppelwort
      SLD   3               // Auf Pointerformat anpassen
      T     #L_BGAdr

      L     #L_Zael
      LOOP  next            // Sprung zu Sprungmarke next
```

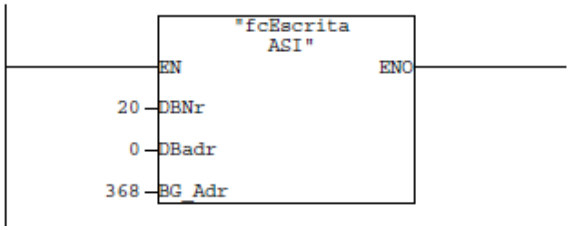
FC21: “fcEscrita”

Mapeamento dos atuadores na memória interna

---

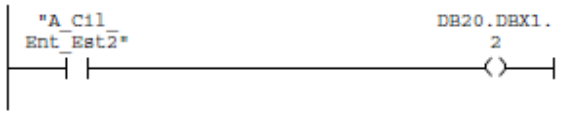
```
Block: FC21
```

```
Network: 1      Escrebe em tudos os atuadores na rede ASI
```

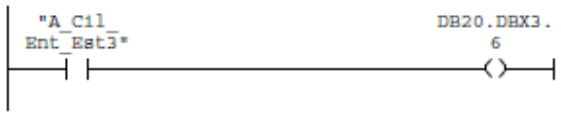


Network: 2 Mapea o cilindro que permite a entrada do carro na estação 1


Network: 3 Mapea o cilindro que permite a entrada do carro na estação 2

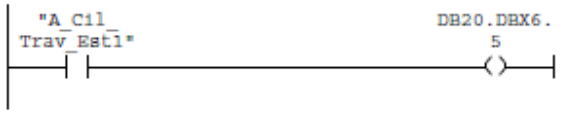



Network: 4 Mapea o cilindro que permite a entrada do carro na estação 3

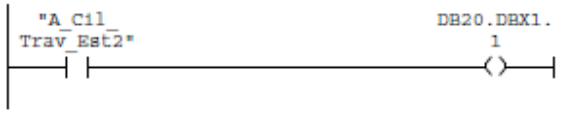



Network: 5 Mapea o cilindro que permite a entrada do carro na estação 4

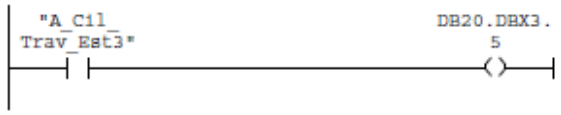

Network: 6 Mapea a trava do carro na estação 1



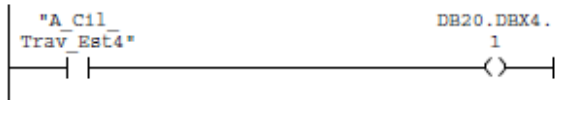

Network: 7 Mapea a trava do carro na estação 2




Network: 8 Mapea a trava do carro na estação 3




Network: 9 Mapea a trava do carro na estação 4



Network: 10 Mapea o cilindro que permite a saida do carro da estação 1


Network: 11 Mapea o cilindro que permite a saida do carro da estação 2


Network: 12 Mapea o cilindro que permite a saida do carro da estação 3

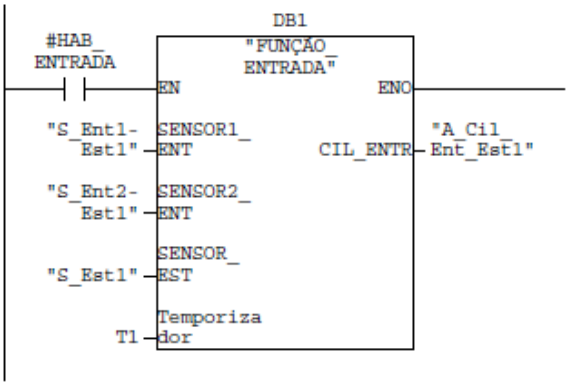

Network: 13 Mapea o cilindro que permite a saida do carro da estação 4


FC50: “Módulo básico Est_1”

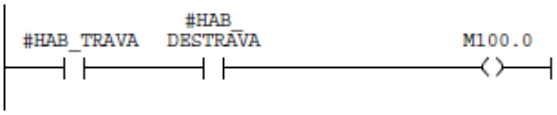
Módulo básico da estação 1

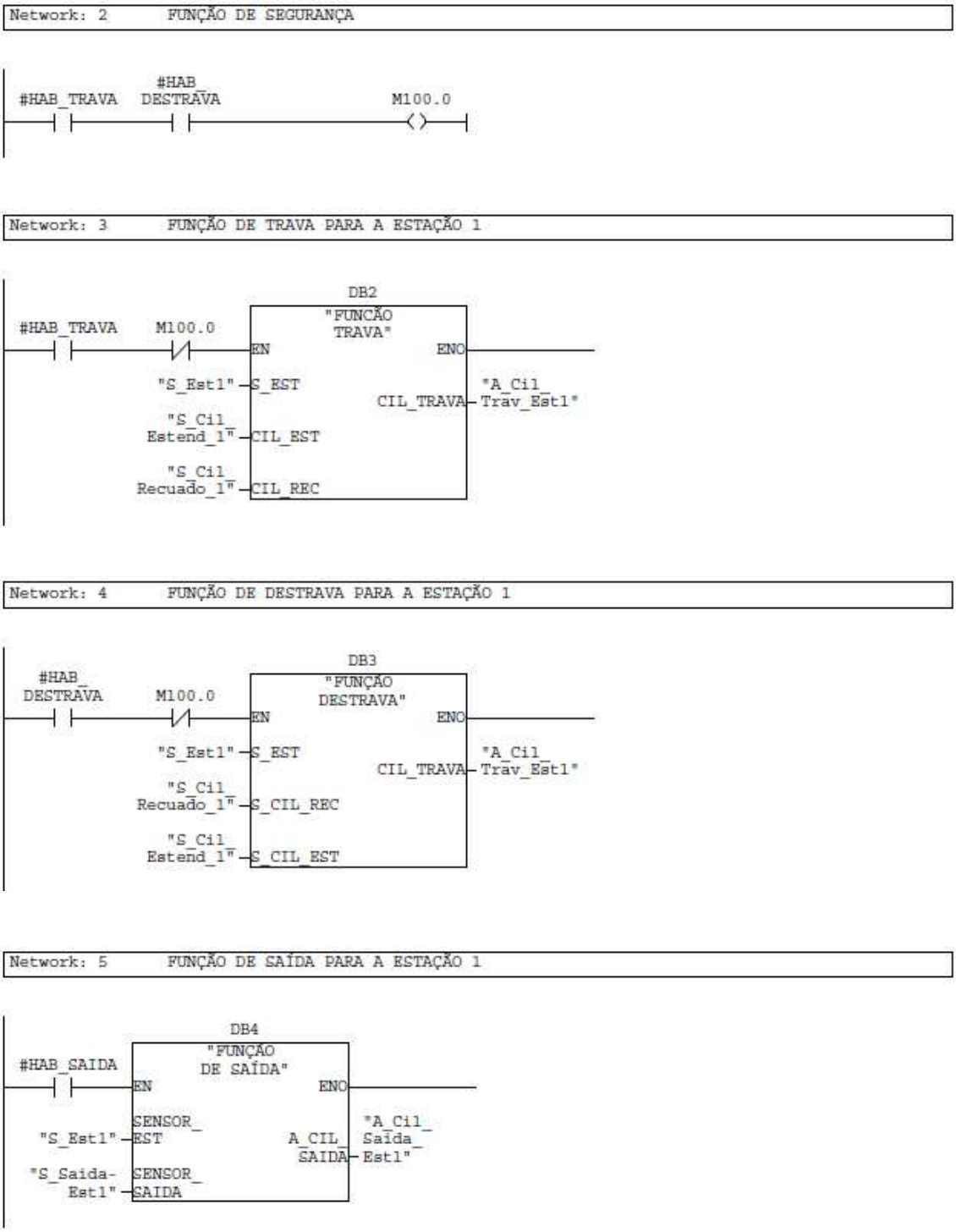
---

Network: 1 FUNÇÃO ENTRADA PARA A ESTAÇÃO 1



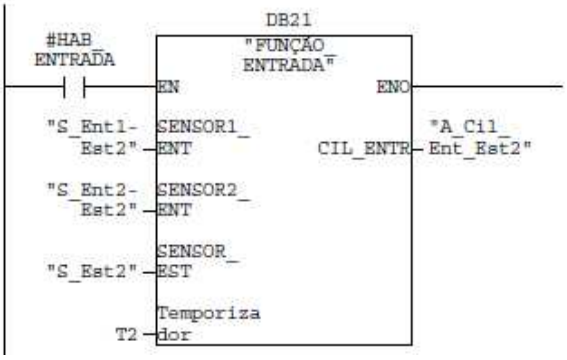

Network: 2 FUNÇÃO DE SEGURANÇA



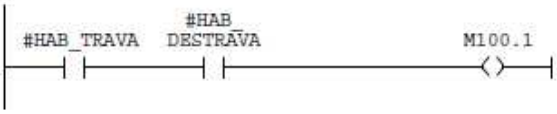



FC51: “Modulo básico Est_2”

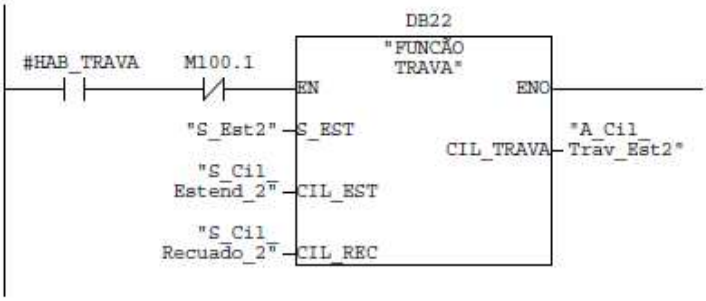
Network: 1 FUNÇÃO ENTRADA PARA A ESTAÇÃO 2



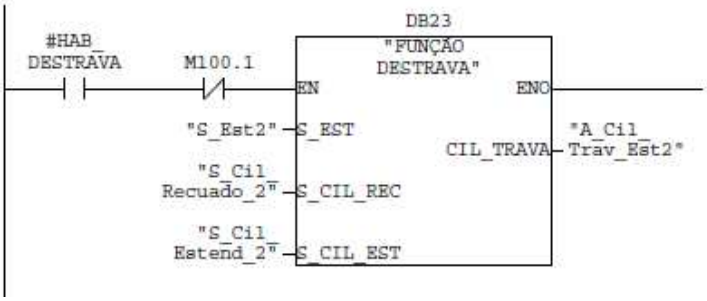

Network: 2 FUNÇÃO DE SEGURANÇA



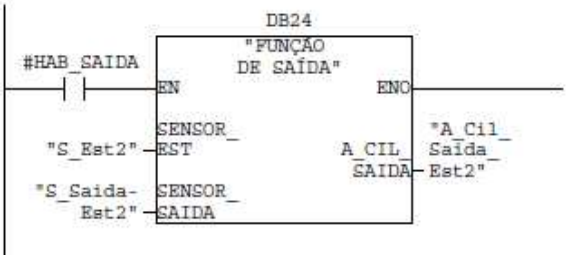

Network: 3 FUNÇÃO DE TRAVA PARA A ESTAÇÃO 2




Network: 4 FUNÇÃO DE DESTRAVA PARA A ESTAÇÃO 2

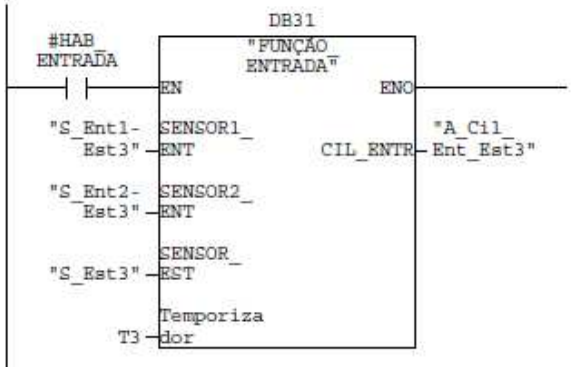



Network: 5 FUNÇÃO DE SAÍDA PARA A ESTAÇÃO 2

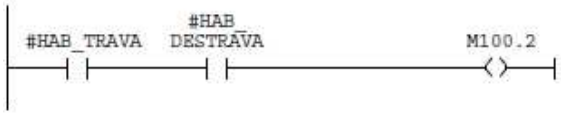


FC52: “Modulo básico Est_3”

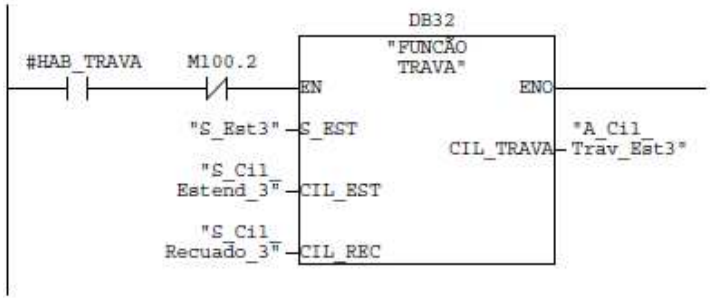
Network: 1 FUNÇÃO ENTRADA PARA A ESTAÇÃO 3



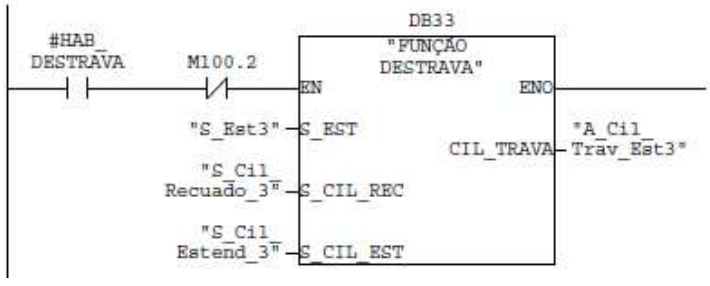

Network: 2 FUNÇÃO DE SEGURANÇA




Network: 3 FUNÇÃO DE TRAVA PARA A ESTAÇÃO 3



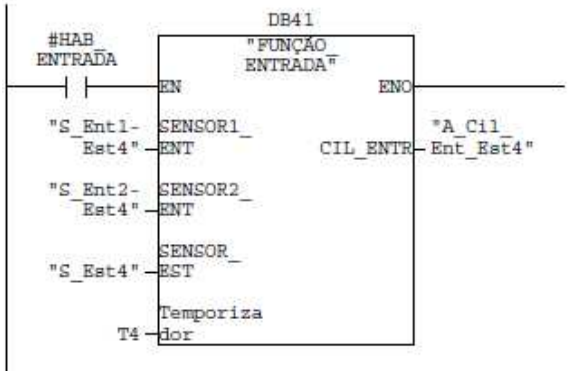

Network: 4 FUNÇÃO DE DESTRAVA PARA A ESTAÇÃO 3



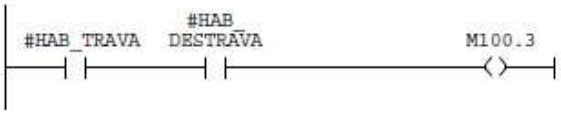
Network: 5 FUNÇÃO DE SAÍDA PARA A ESTAÇÃO 3


FC53: “Modulo básico Est_4”

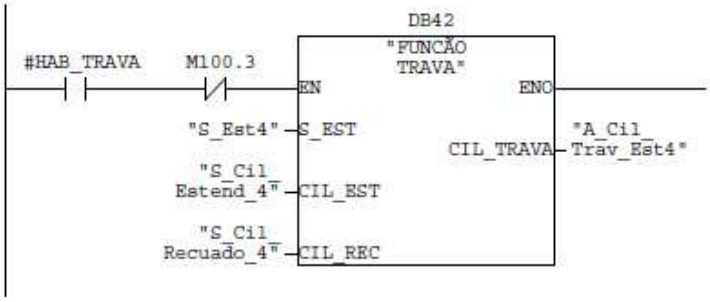
Network: 1 FUNÇÃO ENTRADA PARA A ESTAÇÃO 4

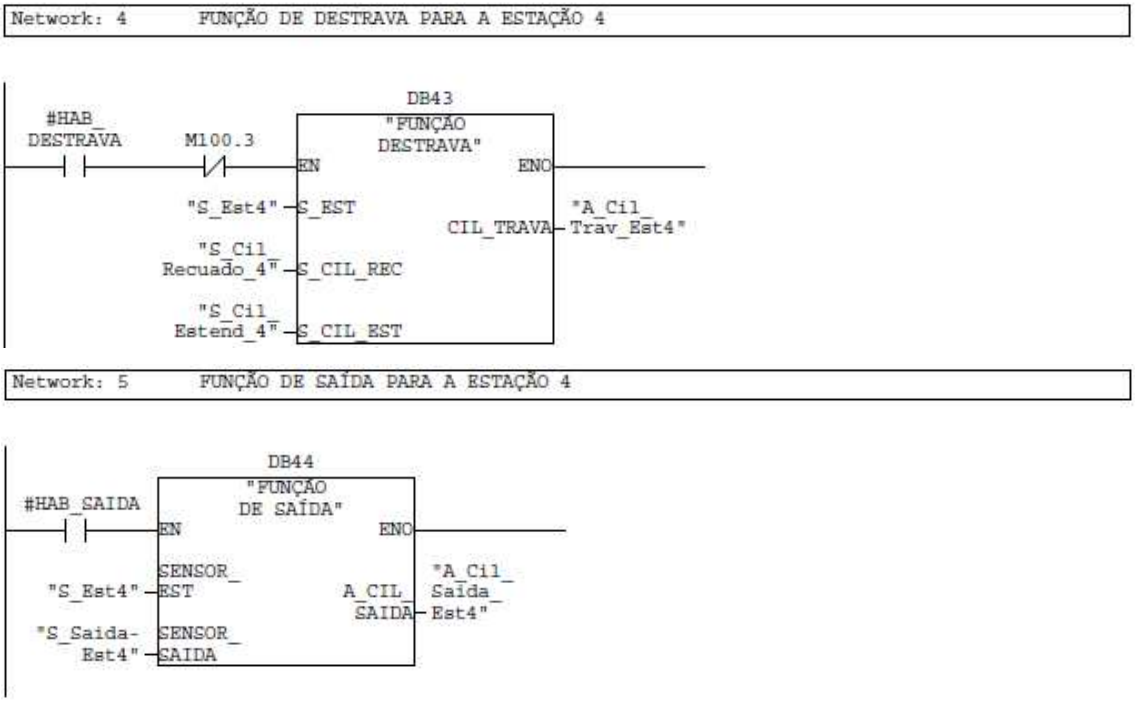



Network: 2 FUNÇÃO DE SEGURANÇA




Network: 3 FUNÇÃO DE TRAVA PARA A ESTAÇÃO 4

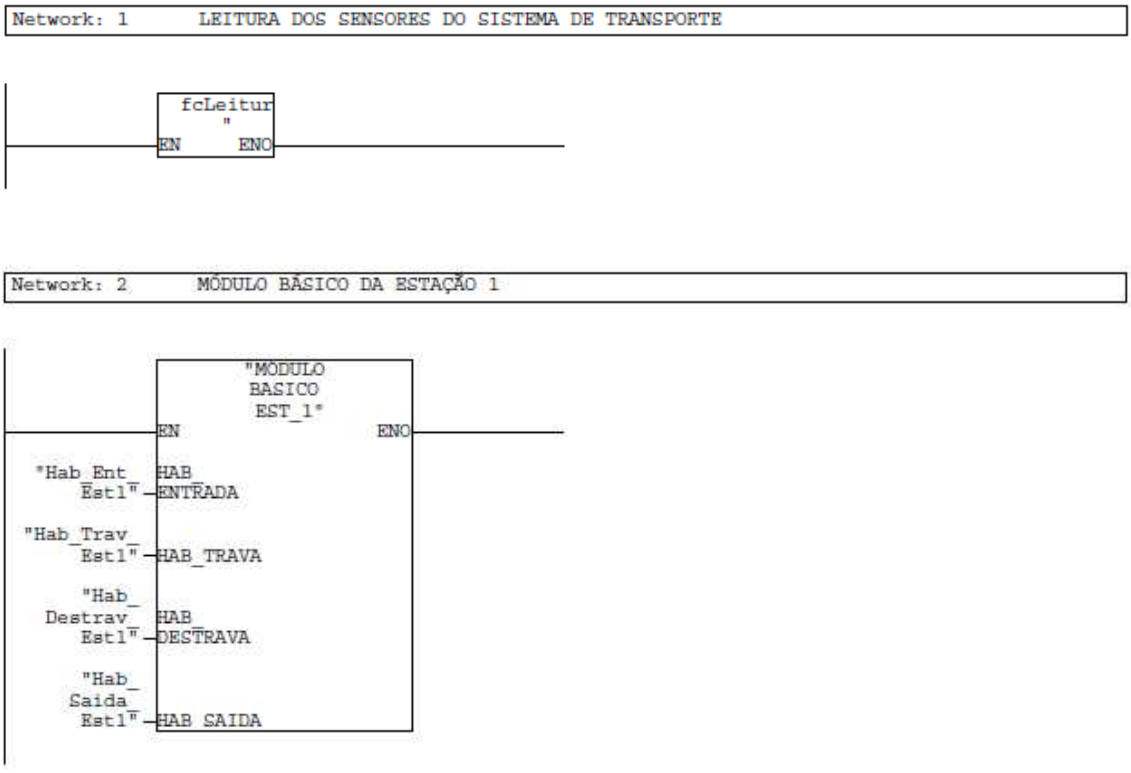





OB1: “Principal”

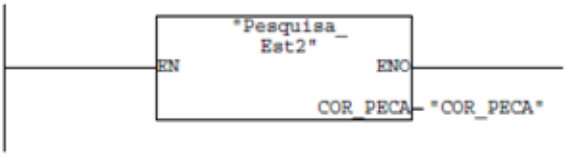
Programa principal

---



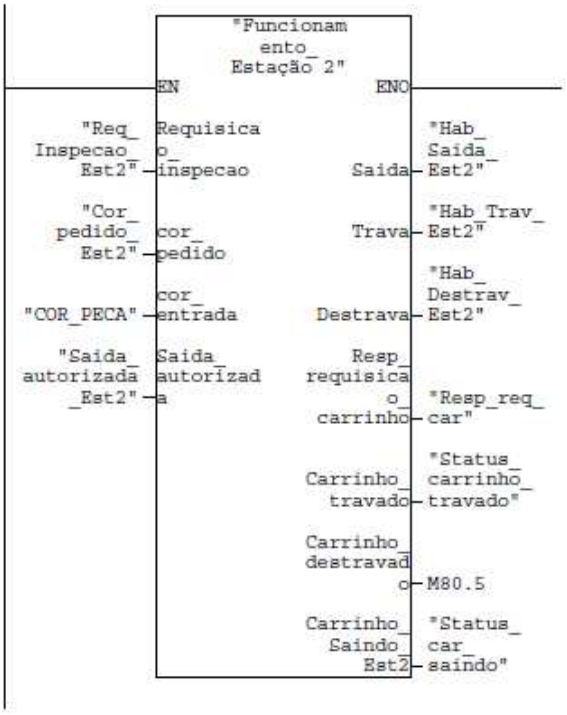
Network: 3

PESQUISA DO ID DO CARRO ESTACIONADO NA ESTAÇÃO 2

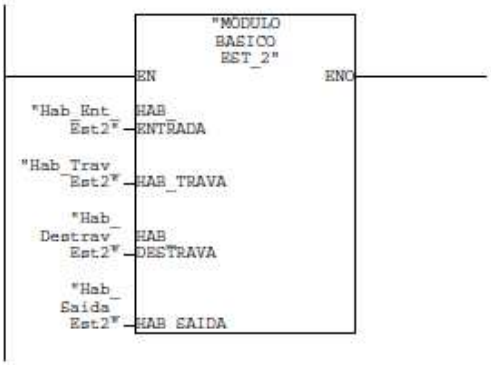



Network: 4

BLOCO DE FUNCIONAMENTO DA ESTAÇÃO 2 --> ESTAÇÃO DE INSPEÇÃO

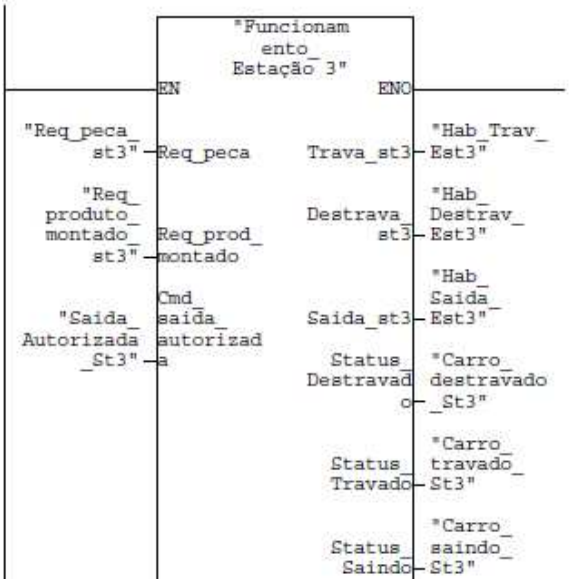



Network: 5 MÓDULO BÁSICO ESTAÇÃO 2

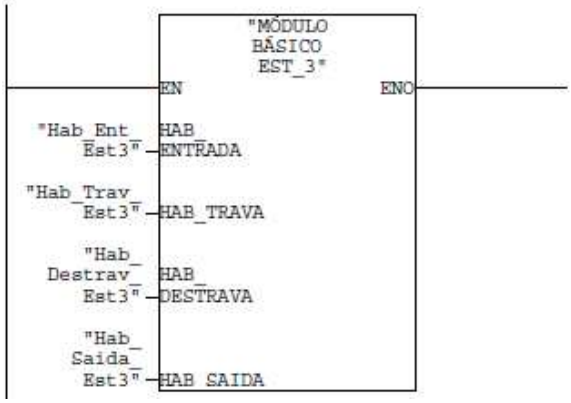


Network: 6

BLOCO DE FUNCIONAMENTO DA ESTAÇÃO 3 --> ESTAÇÃO DE MONTAGEM

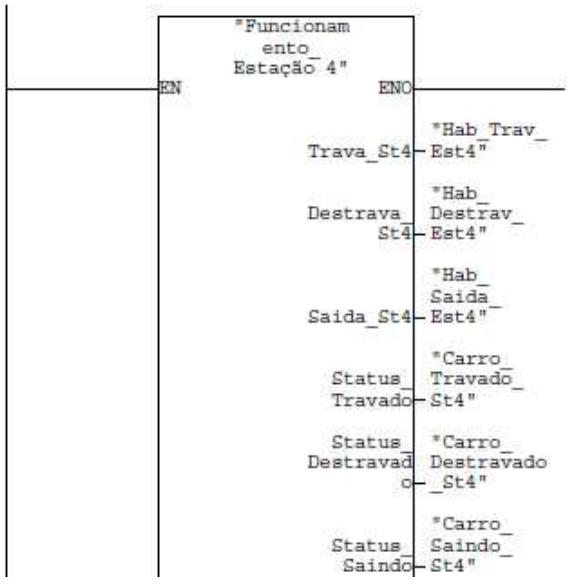



Network: 7 MÓDULO BÁSICO ESTAÇÃO 3

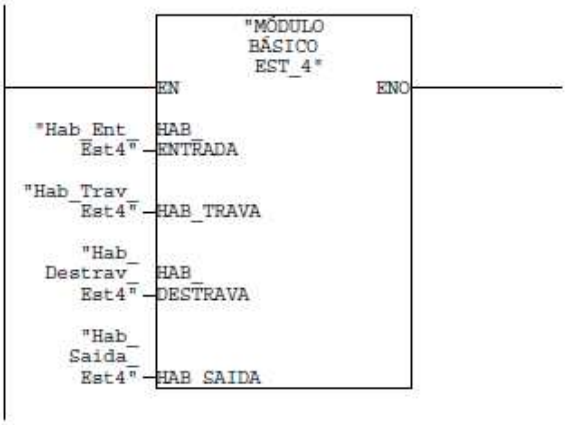



Network: 8

BLOCO DE FUNCIONAMENTO DA ESTAÇÃO 4 --> ESTAÇÃO DE ARMAZENAGEM



Network: 9 MÓDULO BÁSICO ESTAÇÃO 4




Network: 10 ESCRITA NOS ATUADORES DO SISTEMA DE TRANSPORTE