

# **Documentação de Projeto – Parte 1**

## **CONOPS, Domínio do Problema, Especificação**

---

**Projeto:** Controlador de simulação de 3 elevadores

**Autores:** Mariana Bittencourt e Henrique Mazzuchetti

## Parte 1a – CONOPS

---

### 1 Introdução

---

O controlador de simulação de 3 elevadores é um software que irá se comunicar através de porta serial com uma aplicação de simulação já existente. É capaz de enviar, receber e interpretar comandos que controlam três elevadores de um prédio de 15 andares mais o térreo.

Esse sistema e sua lógica podem ser aproveitados para um sistema real de elevadores, visto que o funcionamento do simulador é bem próximo de uma situação da vida real, guardadas as devidas proporções.

## 2 Descrição do Sistema

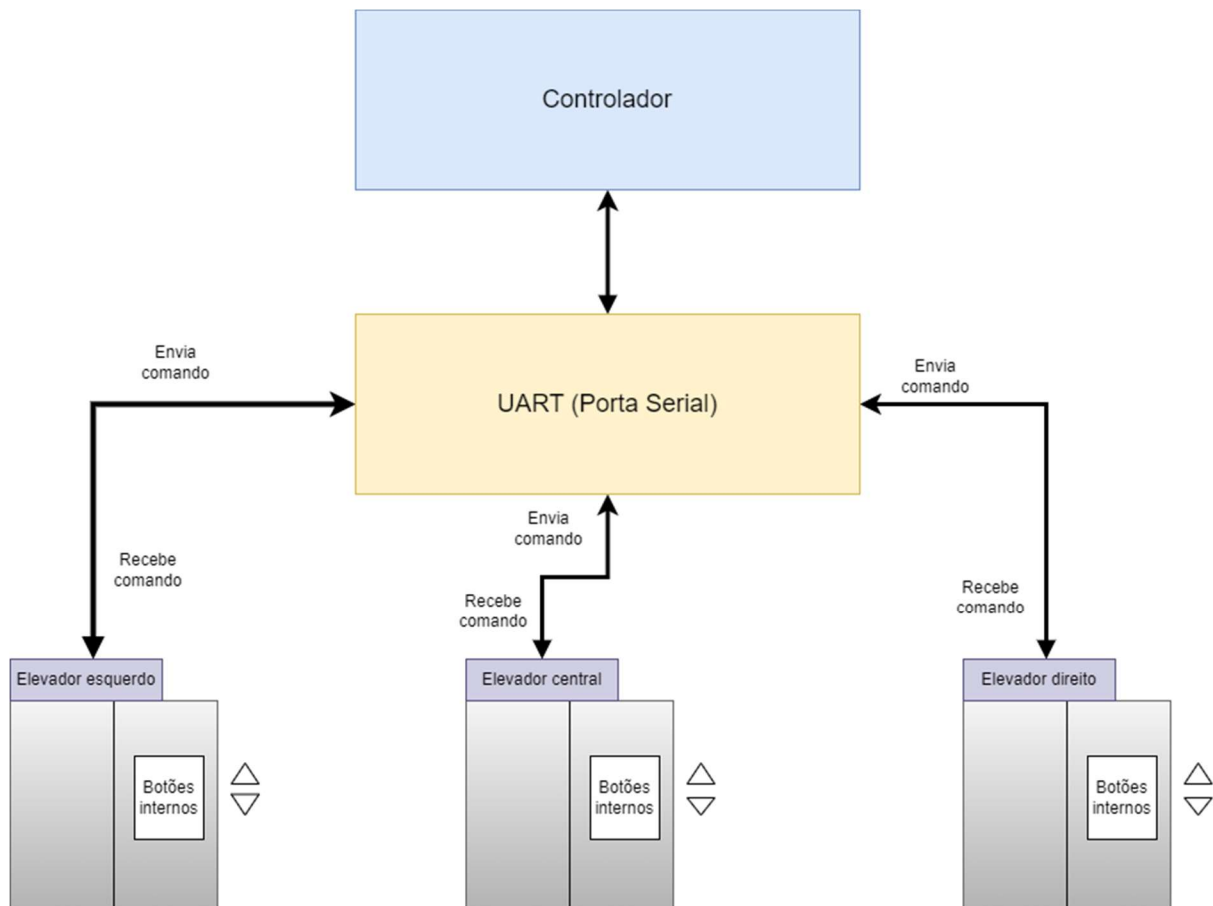


Figura 1 – Diagrama de descrição do sistema

O controlador de simulação de 3 elevadores (em azul na figura) é um software que irá se comunicar através de porta serial (em amarelo) com uma aplicação de simulação já existente. Essa aplicação de simulação é representada pelos três elevadores na figura.

O controlador é capaz de enviar, receber e interpretar comandos que controlam os três elevadores, localizados em um prédio de 15 andares mais o térreo.

Os comandos enviados pelo simulador ao controlador podem ser originados de botões internos nos elevadores, ou de botões externos em cada andar.

O simulador também envia algumas informações ao controlador, como a posição atual de cada elevador e se as portas estão fechadas ou abertas.

Em resposta aos comandos do simulador e baseado nas informações recebidas, o controlador é capaz de definir a lógica de controle dos 3 elevadores, enviando comandos para o simulador, que podem ser comandos de abrir e fechar portas e comandos de movimentação dos elevadores.

### 3 Interface com o Usuário

---

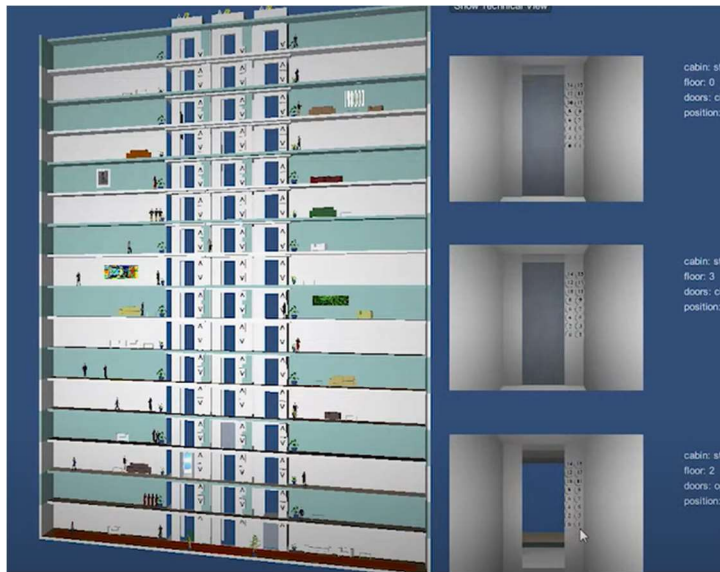


Figura 2 – Simulador dos elevadores

A interface do usuário é o simulador dos 3 elevadores, como pode ser visto na imagem acima.

O usuário pode clicar em qualquer dos botões internos dos três elevadores, e nos botões de subir e descer de cada andar.

Além disso, também é possível para o usuário visualizar os elevadores se movimentando em tempo real.

### 4 Identificação dos Stakeholders

---

- Usuário (moradores do prédio)
- Equipe de manutenção dos elevadores
- Engenheiros que projetaram os elevadores
- Síndicos e administradores do prédio

### 5 Requisitos de Stakeholders

---

- O sistema deverá operar os 3 elevadores simultaneamente
- O sistema deve ser seguro
- Os elevadores devem ter a velocidade ideal, não muito rápido e não muito lento.
- Os elevadores devem requerer pouca manutenção, ao funcionar de uma maneira eficiente, economizando recursos (principalmente energia).

## 6 Cenários de Operação

---

### Situações comuns:

- 1- Apenas uma pessoa solicita um dos elevadores. O elevador então vai até o andar da pessoa e a leva ao andar correto.
- 2- Três pessoas solicitam cada um dos elevadores ao mesmo tempo. O sistema deve ser capaz de processar esses três pedidos simultaneamente e levar cada pessoa ao andar solicitado.
- 3- Enquanto uma pessoa está dentro do elevador, alguém solicita que o elevador pare em um determinado andar. A lógica deve ser capaz de determinar qual pedido atender primeiro, a fim de economizar energia.
- 4- Múltiplas pessoas solicitam o elevador ao mesmo tempo. Utilizando a mesma lógica citada acima, o elevador deve atender a todos os pedidos de forma a economizar energia e tempo.

### Situações incomuns:

- 1- Apenas uma pessoa solicita um dos elevadores. O elevador vai até o andar da pessoa, porém não a leva ao andar solicitado, cometendo um erro.
- 2- Três pessoas solicitam cada um dos elevadores ao mesmo tempo. O sistema não é capaz de processar os três pedidos ao mesmo tempo e acaba atendendo a apenas um ou dois deles.
- 3- Enquanto o elevador está em movimento, alguém solicita que ele pare em algum andar, porém o elevador não está preparado para atender solicitações enquanto em movimento e ignora o pedido.
- 4- Múltiplas pessoas solicitam o elevador ao mesmo tempo, porém como o elevador não está preparado para atender múltiplas solicitações, apenas um pedido é atendido.
- 5- Mesmo estando preparado para atender múltiplos pedidos, o elevador faz isso de forma ineficiente, gastando mais energia e tempo que o necessário.

## Parte 1b – Domínio do Problema

---

### 1 Comunicação Serial e UART

---

Existem duas formas principais de um dispositivo se comunicar com outro: Paralela e Serial.

Na forma de comunicação paralela, existem diversos fios, um para cada bit. E todos os bits da palavra são enviados simultaneamente, de forma paralela.

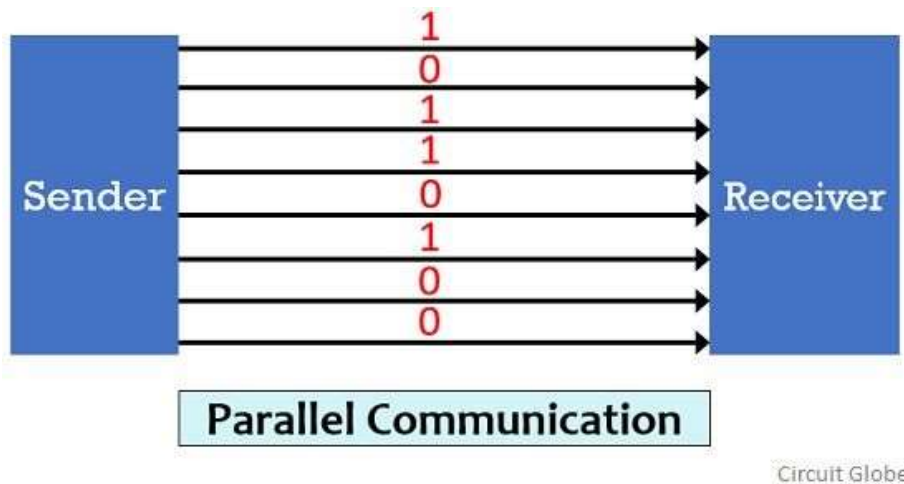


Figura 3 – Comunicação paralela

A desvantagem principal da comunicação paralela é o custo, pois o número de fios necessários aumenta conforme o tamanho da palavra.

Já na forma de comunicação serial, apenas um fio é necessário para enviar toda a mensagem.

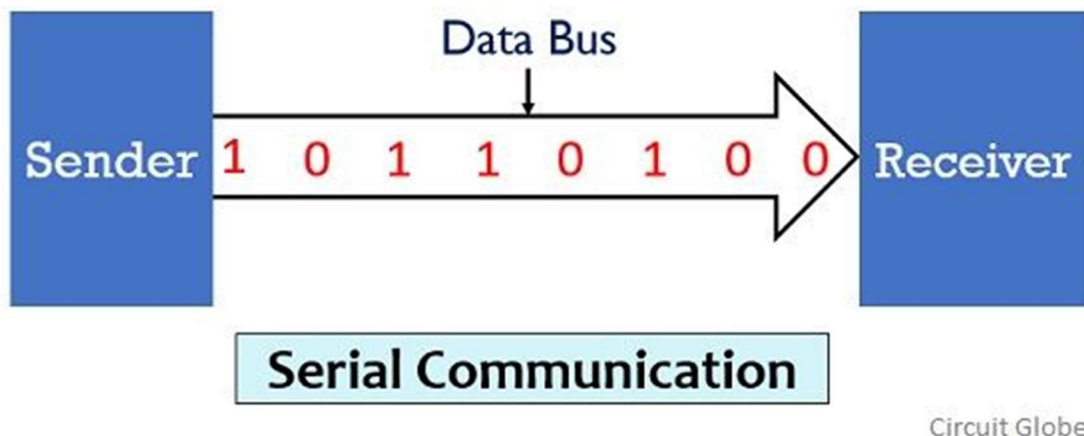


Figura 4 – Comunicação serial

Essa forma de comunicação traz muitas vantagens. Dentre elas:

- Baixo custo
- Permite comunicação de longa distância
- Eficiente para altas frequências

Normalmente é necessário um TRANSCEPTOR, que é o elemento que irá controlar a comunicação pelo barramento serial e converter os bits recebidos em uma palavra e vice-versa.

Na maioria dos circuitos de comunicação serial, esse transceptor é o UART (Universal Asynchronous Receiver Transmitter).

Como a transmissão em UART é assíncrona, torna-se necessário o envio de bits especiais para a sincronização. Existem três principais: O bit Start (para anunciar o início da transmissão), o bit Stop (para sinalizar o final da transmissão) e o bit de paridade (para checar eventuais erros na transmissão).

A velocidade de transmissão e tamanho da palavra a ser transmitida são parâmetros configuráveis e devem ser iguais no transmissor e no receptor.

## 2 Algoritmos de Scheduling - SCAN

---

Para implementar a lógica de controle dos 3 elevadores, alguma forma de algoritmo de scheduling deverá ser utilizada.

Scheduling é uma palavra em inglês que representa a ação de escolher quais ações serão feitas e em qual ordem.

Esses algoritmos são utilizados, por exemplo, por Hard Disks para buscar informações armazenadas, e por Sistemas Operacionais para determinar quais tarefas serão executadas pelo processador.

O algoritmo escolhido para esse sistema é o algoritmo SCAN, também conhecido como o algoritmo do elevador. Ele é chamado assim pois prioriza o atendimento de requisições na mesma direção em que já está se movimentando.

Por exemplo, se o elevador está no 4º andar e subindo, e recebe uma requisição para o 6º andar e para o 3º andar, irá priorizar o 6º andar, pois já está se movimentando para cima.

Existem outros algoritmos de scheduling que também poderiam ser utilizados nesse projeto, como o FCFS (First Come First Served) e o Shortest Seek First. Porém, por simplicidade e eficiência, o escolhido foi o SCAN.

## Parte 1c – Especificação

### 1 Introdução

Nessa parte do documento serão apresentados os requisitos do sistema (funcionais e não funcionais), assim como as restrições.

### 2 Estrutura do Sistema

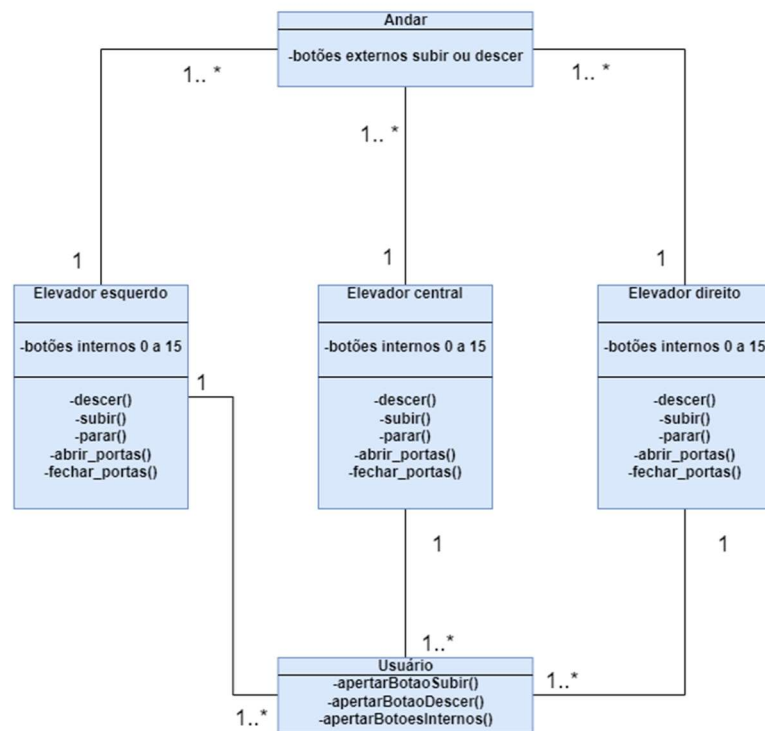


Figura 5 – Diagrama estrutural UML



### 3 Arquitetura funcional

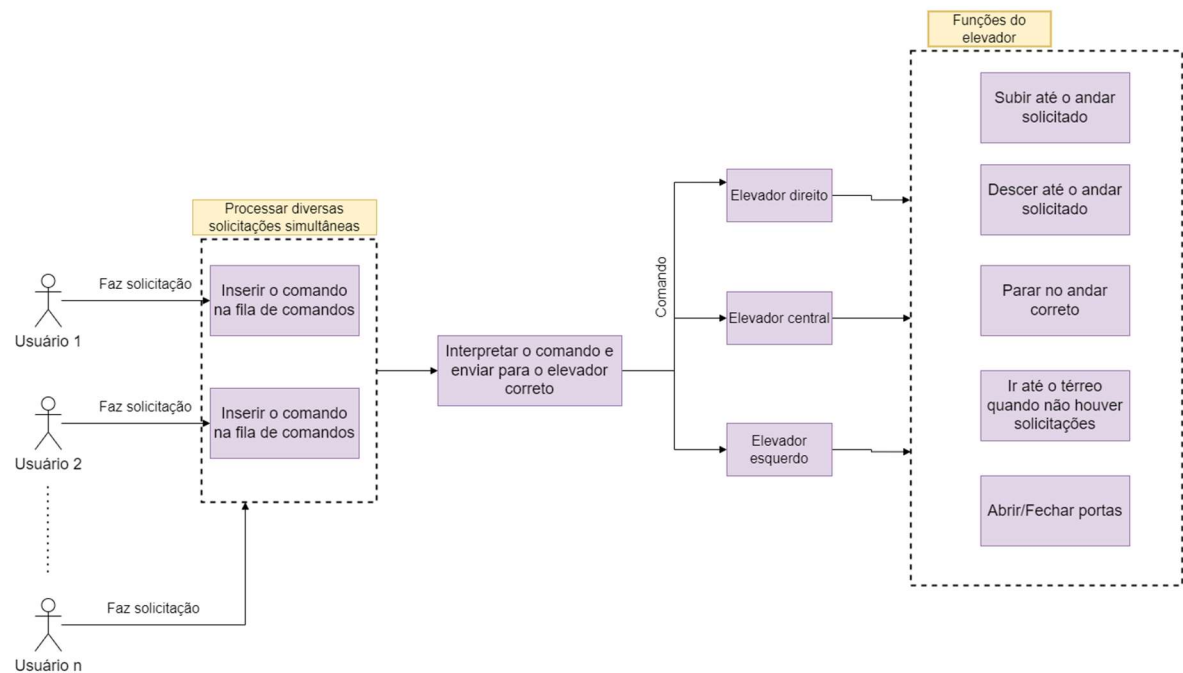


Figura 6 – Diagrama funcional

### 4 Especificação Funcional

RF-01: O sistema deverá ser capaz de processar diversas solicitações simultâneas (diversos usuários).

RF-02: O sistema deverá inserir cada comando em uma fila de comandos principal do controlador.

RF-03: O controlador deverá ser capaz de interpretar o comando da fila principal e enviar para o elevador correto.

RF-04: Cada elevador deverá possuir uma fila de comandos associada.

RF-05: Cada elevador deverá ser capaz de subir/descer até o andar solicitado, conforme determinar o controlador.

RF-06: Cada elevador deverá ser capaz de parar na posição correta em cada andar.

RF-07: Cada elevador deverá ser capaz de abrir ou fechar as portas corretamente.

RF-08: Cada elevador deverá ir até o andar térreo se não houver solicitações a serem atendidas.

### 5 Especificação não Funcional

RNF-01: O sistema deverá ser eficiente, de forma a economizar energia e tempo.

RNF-02: O sistema deverá utilizar o algoritmo de scheduling "SCAN" para atender às solicitações dos usuários.

RNF-03: O sistema deverá ter 4 threads: 1 principal, e 1 para cada elevador.

RNF-04: O sistema deverá ter uma função que envia os comandos para o simulador protegida por um Mutex.

## **6 Restrições**

---

O sistema deverá ser desenvolvido totalmente em linguagem C.

O sistema deverá usar a placa de desenvolvimento Tiva.

O sistema deverá usar o CMSIS-RTOS como Sistema Operacional de Tempo Real.