

Documentação

Estrutura

Os projeto foi desenvolvido utilizando a linguagem JavaScript e demais tecnologias voltadas para a Web. Segue abaixo as principais pastas e arquivos do projeto:

- `fontes/`: contém os arquivos fontes do projeto.
- `fontes/css`
- `fonte/js`
- `fonte/html`
- `documentacao/`: contém o manual, a documentação, imagens do sistema e diagramas.
- `manual.pdf`
- `documentacao.pdf`
- `bibliotecas/`: contém bibliotecas de suporte utilizadas no projeto.

Módulos

O código fonte foi dividido em módulos para facilitar a separação. Assim, na nossa pasta `fonte/js` cada arquivo corresponde a um módulo independente, sendo eles:

- `aleatorio.js`: contém as classes estatísticas de suporte para a realização das operações de modelagem. Contém classes como `Gna`, `MonteCarlo`, `Normal`, `Triangular` e outras.
- `modelo.js`: Responsável por criar as distribuições utilizadas na simulação e gerenciar as configurações.
- `simulacao.js`: Realiza a simulação. Nesse módulo estão os centros de recepção, de serviço e de despacho, além de classes necessárias na simulação como `Mensagem` e `EventoDeSimulacao`.
- `controle.js`: Realiza o controle gráfico da página e faz a intermediação entre a interface gráfica com o módulo `modelo.js` e com o módulo `simulacao.js`. Responsável também por controlar os botões e ações do usuário.

Módulo aleatório

Classes do módulo:

- `CongruenteMisto`
- `CongruenteMultiplicativo`
- `Gna`
- `ClasseMonteCarlo`
- `MonteCarlo`
- `Uniforme`
- `Exponencial`
- `Normal`
- `Triangular`

Módulo modelo

Fachada do módulo: `SceModelo`

Enumerações do módulo:

- `TipoDeMensagem`
- `TipoDeTrafegoDeMensagem`

Módulo simulação

Fachada do módulo: `SceSimulacao`

Enumerações do módulo:

- `TipoDeEventoDeSimulacao`

Classes do módulo:

- `EventoDeSimulacao`
- `CentroDeRecepcao`
- `CentroDeServico`
- `CentroDeDespacho`
- `Mensagem`
- `Fotografia`
- `Estatistico`

Módulo controle

Fachada do módulo: `SceControle`

Simulação

O funcionamento da simulação consiste em uma tabela de eventos futuros. Cada vez que um evento é disparado, esse evento é executado e pode colocar novos eventos na tabela de eventos futuros. O disparo dos eventos é realizado através do módulo **controle** que por sua vez é responsável por conversar com o usuário.

Antes de um novo evento ser disparado a tabela de eventos futuros é ordenada para que seja escolhido o próximo evento. A ordenação ocorre de acordo com o tempo no relógio do disparo do evento. Além disso ao disparar um evento, o tempo no relógio da simulação passa a ser o tempo no relógio do disparo do evento.

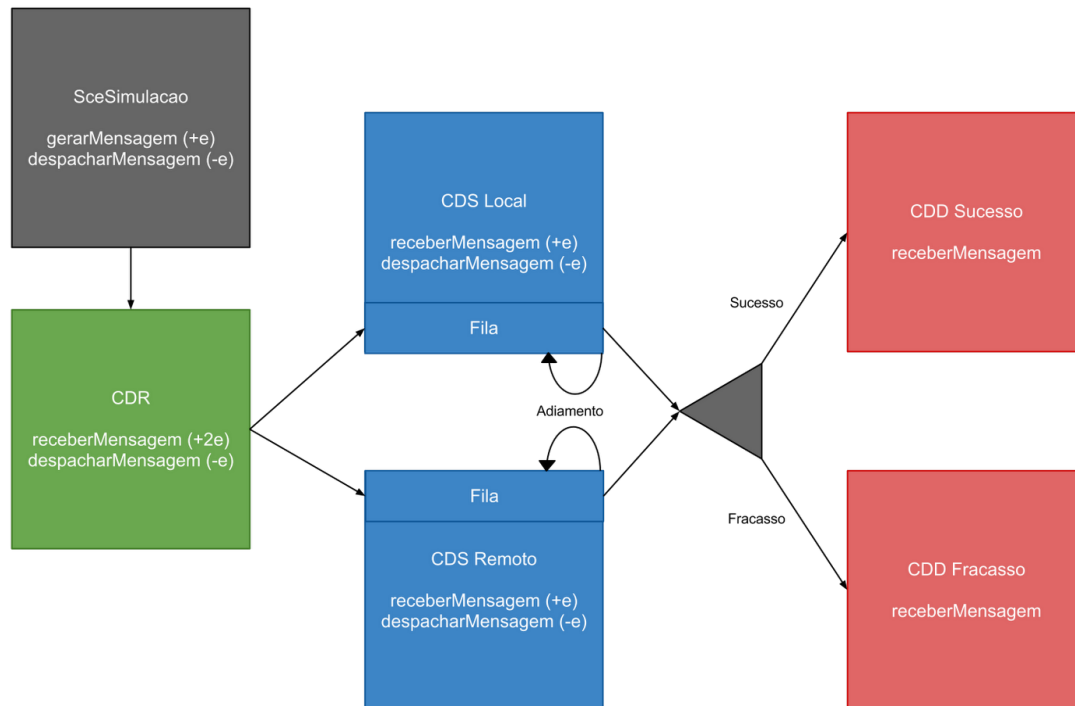
No caso do nosso simulador, temos um evento basilar que é o evento **INICIO**. Este evento é responsável por gerar a primeira mensagem no sistema e é responsável por adicionar um novo evento na tabela de eventos futuros. O novo evento será a entrada da mensagem no CDR (centro de recepção).

Sempre que uma nova mensagem chega no CDR, este por sua vez adiciona dois novos eventos na tabela de eventos futuros, o primeiro é o evento de uma nova entrada de mensagem no CDR, e o segundo é um evento que despachará a mensagem que está no CDR para o CDS (centro de serviço).

Quando o evento que realiza a entrada de uma mensagem no CDS é disparado, o processamento referente a mensagem é realizado e o CDS adiciona um novo evento na tabela de eventos futuros. Esse novo evento é o despacho da mensagem. O despacho da mensagem depende do tipo de mensagem. Se a mensagem for um sucesso ou um fracasso ela é despachada para um CDD (centro de despacho). Se a mensagem for um adiamento, então ela entra novamente no CDS. Vale lembrar que o CDS pode possuir filas. Sendo assim, a mensagem pode precisar esperar até existirem servidores livres para poder processá-la e despachá-la.

Finalmente, quando o evento que encaminha a mensagem para o CDD é disparado, a mensagem chega ao fim do seu percurso. É importante mencionar que o CDD não existe fisicamente (assim como existe o CDR e os CDSs), porém ele foi criado apenas para facilitar a coleta das estatísticas das mensagens que já saíram do sistema. De qualquer forma, para todos os efeitos, quando uma mensagem chega no CDD é como se ela estivesse saído do sistema.

O simulador também possui um outro evento básico que é o **FIM**. Assim que o usuário finaliza a simulação esse evento é disparado e mais nenhum evento poderá ser adicionado. O caminho que uma mensagem segue no sistema pode ser visto no diagrama abaixo:



(+xe) significa adição de **x** novos eventos na tabela de eventos futuros.

(-xe) significa consumo de **x** eventos da tabela de eventos futuros.