

SIMULADOR DE CIRCUITOS DIGITAIS – INTERFACE
PROFESSOR: ADELARDO ADELINO DANTAS DE MEDEIROS

AS CLASSES Qt

Para representação das 4 janelas que compõem a interface do programa Circuito (a principal e as 3 janelas que aparecem quando necessário), devem ser criadas 4 classes Qt:

- **MainCircuito**: janela principal do programa, sempre em exibição
- **ModificarPorta**: janela exibida quando for modificar as características de uma porta.
- **ModificarSaída**: janela exibida quando for modificar uma saída do circuito.
- **NovoCircuito**: janela exibida quando for criar um novo circuito.

OS DADOS DA CLASSE `MainCircuito`

Os objetos da classe `MainCircuito` têm os seguintes dados membro, todos privados:

- Um objeto `C` do tipo `Circuito`.
- Um ponteiro `novoCircuito` do tipo `NovoCircuito*`.
- Um ponteiro `modificarPorta` do tipo `ModificarPorta*`.
- Um ponteiro `modificarSaída` do tipo `ModificarSaída*`.
- Ponteiros `numIn`, `numOut` e `numPortas`, do tipo `QLabel*`, para exibir na barra de status.

COMUNICAÇÃO ENTRE AS CLASSES:

Para cada sinal da classe remetente, deve haver um slot correspondente na classe destinatária, sendo feita a conexão (`connect`) entre essas duas entidades.

Sinais enviados pela classe `MainCircuito`:

- Após ação “Novo” (menu Circuito):
 - Envia sinal para a classe `NovoCircuito`.
 - O slot correspondente:
 - Limpa (retorna ao valor inicial 1) os `spinBox`’s da janela.
 - Exibe (`show`) objeto `novoCircuito`.
- Após redefinir o circuito (seja por criação de novo circuito, após receber o sinal da classe `NovoCircuito`, seja por leitura de arquivo):
 - Envia sinal para a classe `ModificarPorta` e para a classe `ModificarSaída` (o mesmo sinal pode ser conectado a 2 slots, um de cada classe) com os novos limites dos `spinBox`’s que são utilizados para indicar a origem das entradas das portas e das saídas do circuito:
 - Limite inferior (-nº de entradas) e limite superior (nº de portas).
 - Os slots correspondentes:
 - Alteram os limites (`setRange`) dos slots da janela.
- Após pedir para modificar (ENTER, clique duplo, etc.) uma porta:
 - Envia sinal para classe `ModificarPorta` com os dados atuais da porta a ser modificada:
 - id da porta, tipo da porta, número de entradas da porta, id’s das 4 entradas da porta.
 - O slot correspondente:
 - Altera os valores (`setNum`, `setCurrentText`, `setValue`) dos widgets de exibição e de entrada de dados.
 - Exibe (`show`) objeto `modificarPorta`.
- Após pedir para modificar (ENTER, clique duplo, etc.) uma saída:
 - Envia sinal para classe `ModificarSaída` com os dados da saída a ser modificada:
 - id da saída, id da origem da saída.

- O slot correspondente:
 - Altera os valores (`setNum`, `setValue`) dos widgets de exibição e de entrada de dados.
 - Exibe (`show`) objeto `modificarSaida`.

Sinais enviados pela classe `NovoCircuito`:

- Após aceitar criação (pressionar OK) de um novo circuito:
 - Envia sinal para classe `MainCircuito` com os dados do novo circuito:
 - Número de entradas, número de saídas, número de portas.

Sinais enviados pela classe `ModificarPorta`:

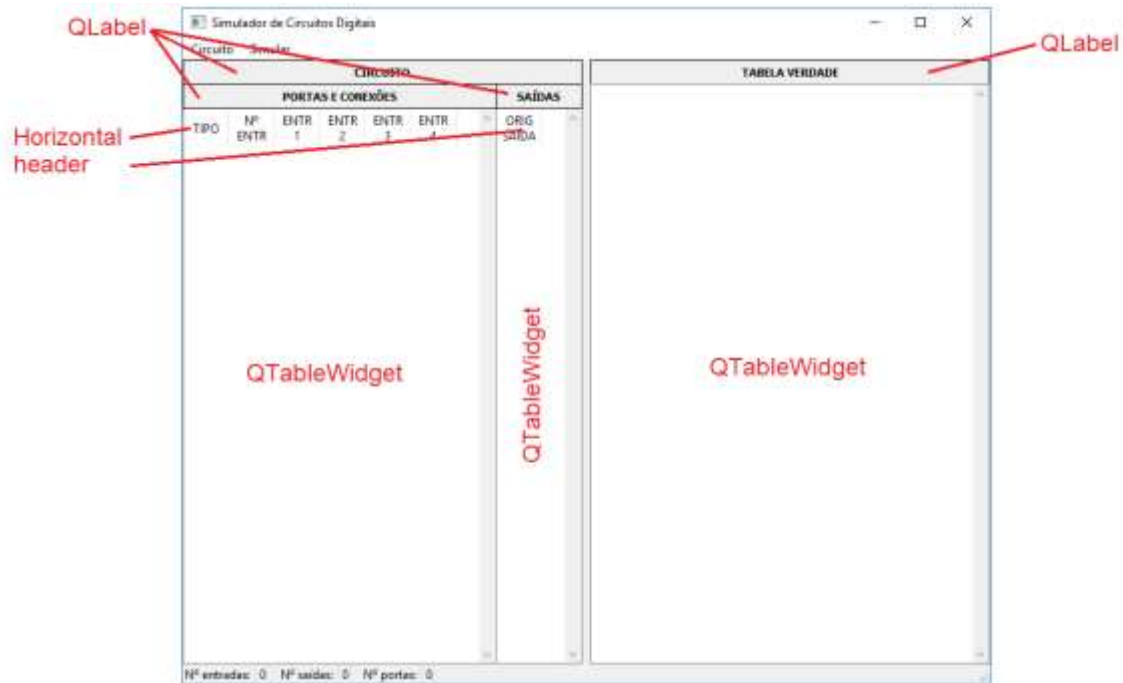
- Após aceitar (pressionar OK) novos parâmetros da porta:
 - Envia sinal para classe `MainCircuito` com os novos parâmetros da porta:
 - id da porta, tipo da porta, número de entradas da porta, id's das 4 entradas da porta.

Sinais enviados pela classe `ModificarSaida`:

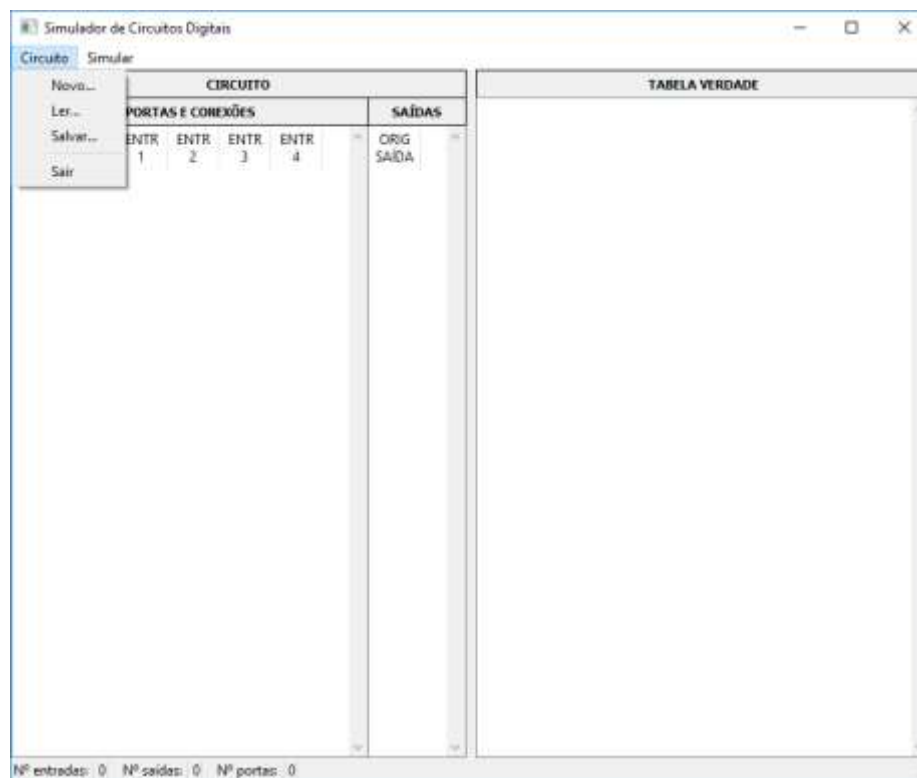
- Após aceitar (pressionar OK) novos parâmetros da saída:
 - Envia sinal para classe `MainCircuito` com os novos parâmetros da saída:
 - id da saída, id da origem da saída.

APARÊNCIA DAS JANELAS:

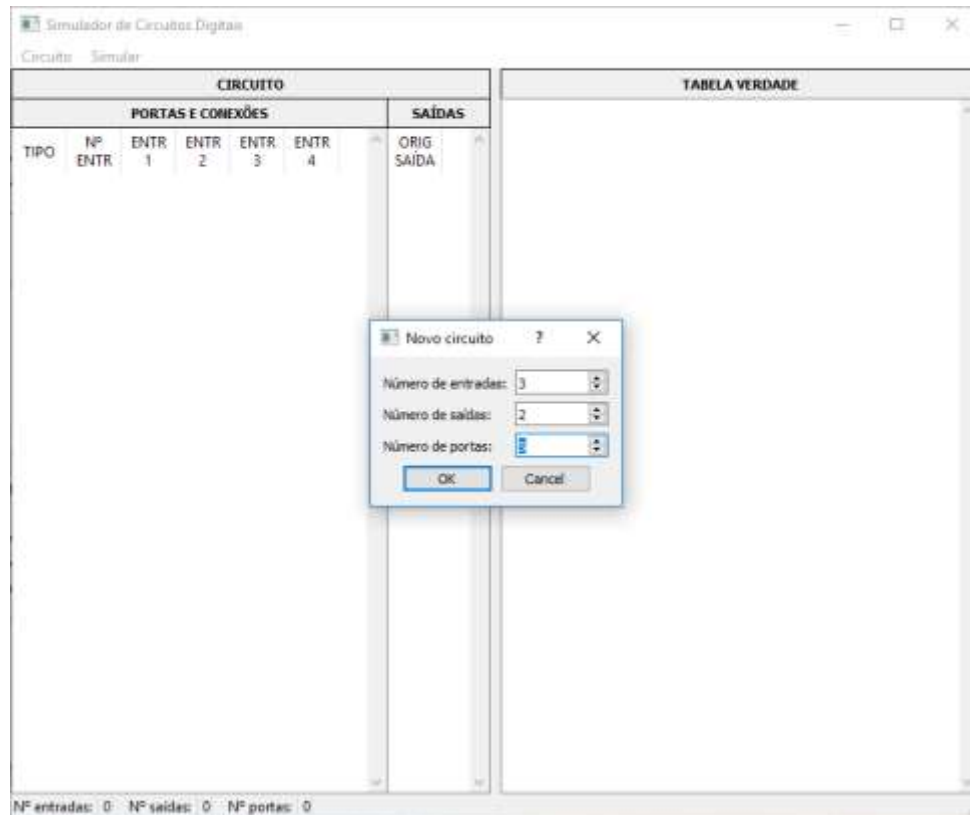
1. Tela inicial



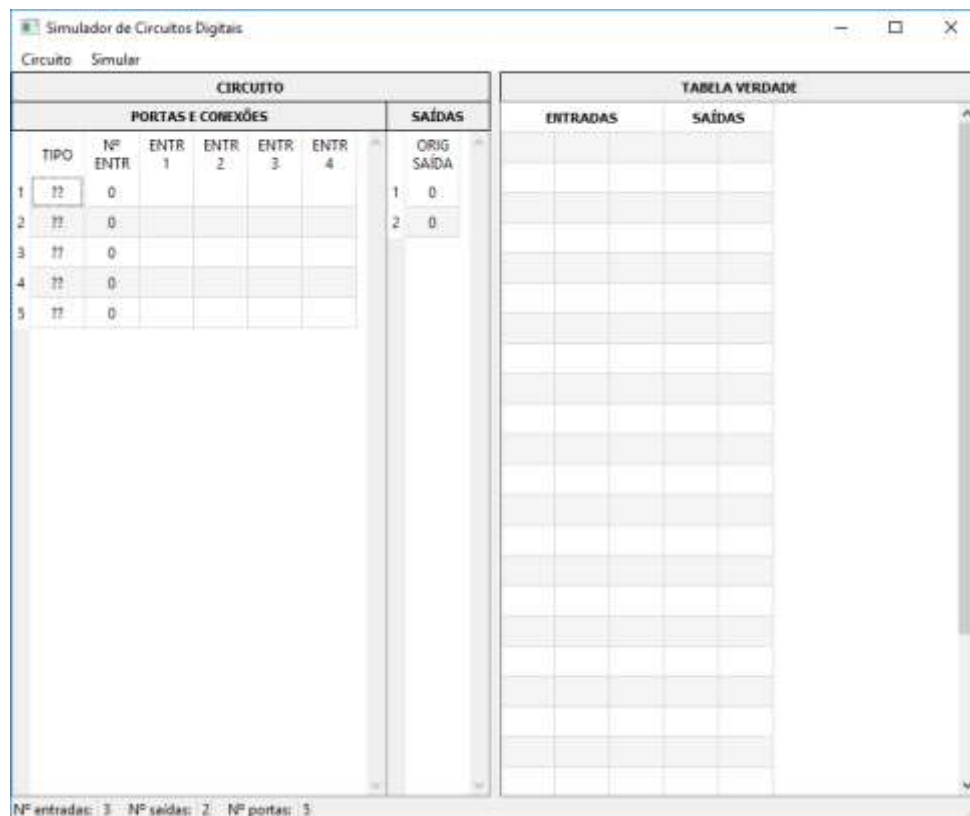
2. Menu circuito



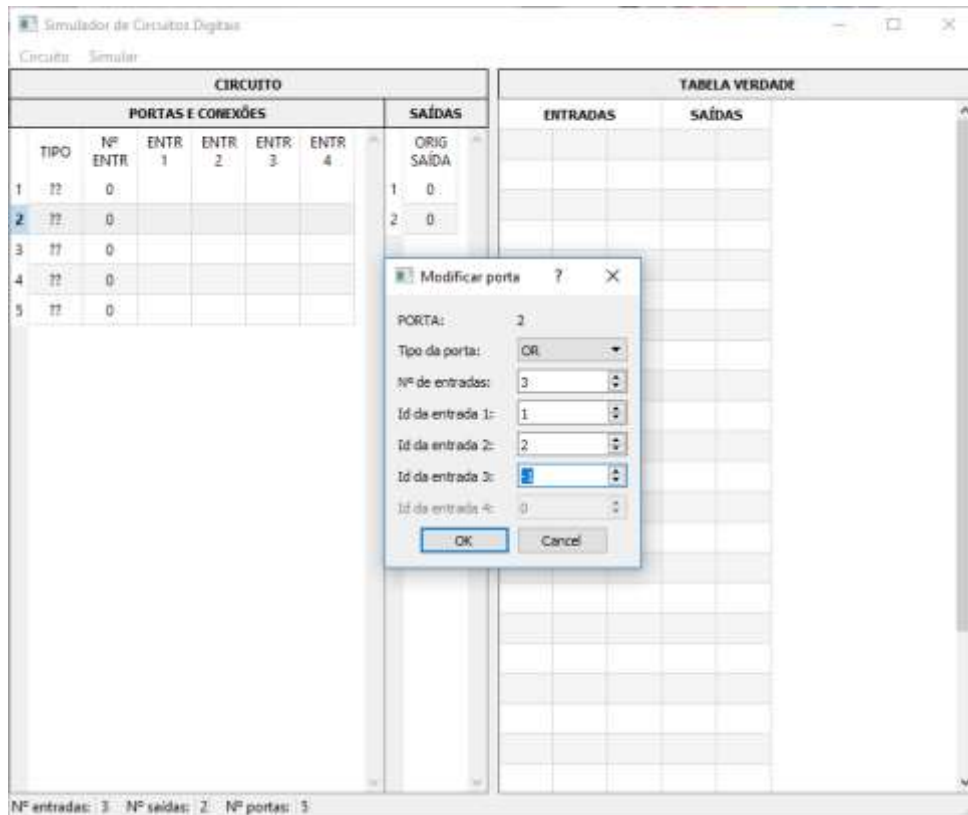
2.1. Ação Novo



2.1.1. Tela após Ação Novo



2.1.2. Tela após clicar para modificar uma porta



Limites e comportamento dos widgets:

- 1) Quando o tipo de porta é modificado no comboBox, modificar os limites do spinBox com o número de entradas daquela porta (se NOT, de 1 a 1; se outro tipo, de 2 a 4):

```
void ModificarPorta::on_comboTipoPorta_currentIndexChanged(const QString &arg1)
{
    if (arg1=="NT") ui->spinNumInputs->setRange(1,1);
    else ui->spinNumInputs->setRange(2,4);
}
```

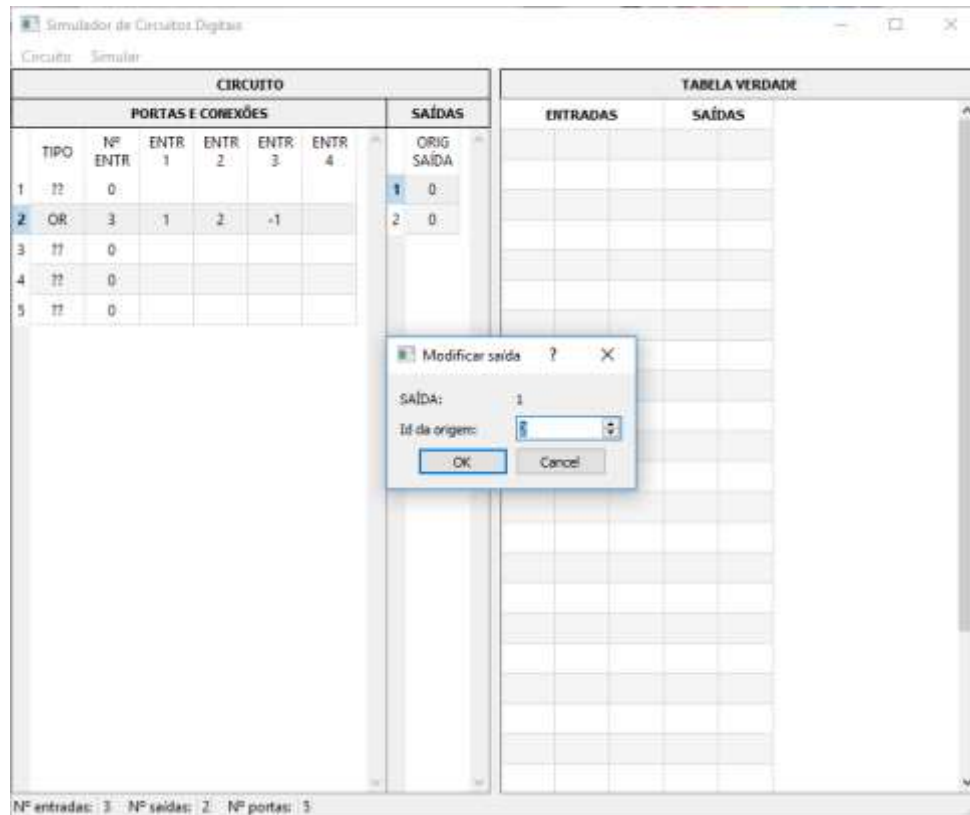
- 2) Quando altera o número de entradas da porta, habilita/desabilita os widgets correspondentes:

```
void ModificarPorta::on_spinNumInputs_valueChanged(int arg1)
{
    ui->labelInput2->setEnabled(arg1 >= 2);
    ui->spinInput2->setEnabled(arg1 >= 2);
    ui->labelInput3->setEnabled(arg1 >= 3);
    ui->spinInput3->setEnabled(arg1 >= 3);
    ui->labelInput4->setEnabled(arg1 >= 4);
    ui->spinInput4->setEnabled(arg1 >= 4);
}
```

- 3) Cada vez que altera o valor de um dos spinBox da ID de uma entrada da porta, ou depois que altera o número de entradas da porta, chama uma função que testa se todas as entradas da porta são válidas (diferentes de 0), cabendo a essa função habilitar/desabilitar o botão OK.

```
void ModificarPorta::testa_entradas_validas(void)
{
    bool in_ok1 = (ui->spinInput1->value() != 0);
    bool in_ok2 = (ui->spinInput2->value() != 0 || !ui->spinInput2->isEnabled());
    ...
    // Recupera um ponteiro para o botao OK
    QPushButton *botao_ok = ui->buttonBox->button(QDialogButtonBox::Ok);
    // Habilita/desabilita o botao OK
    botao_ok->setEnabled(in_ok1 && in_ok2 && in_ok3 && in_ok4);
}
```

2.1.3. Tela após clicar para modificar uma saída

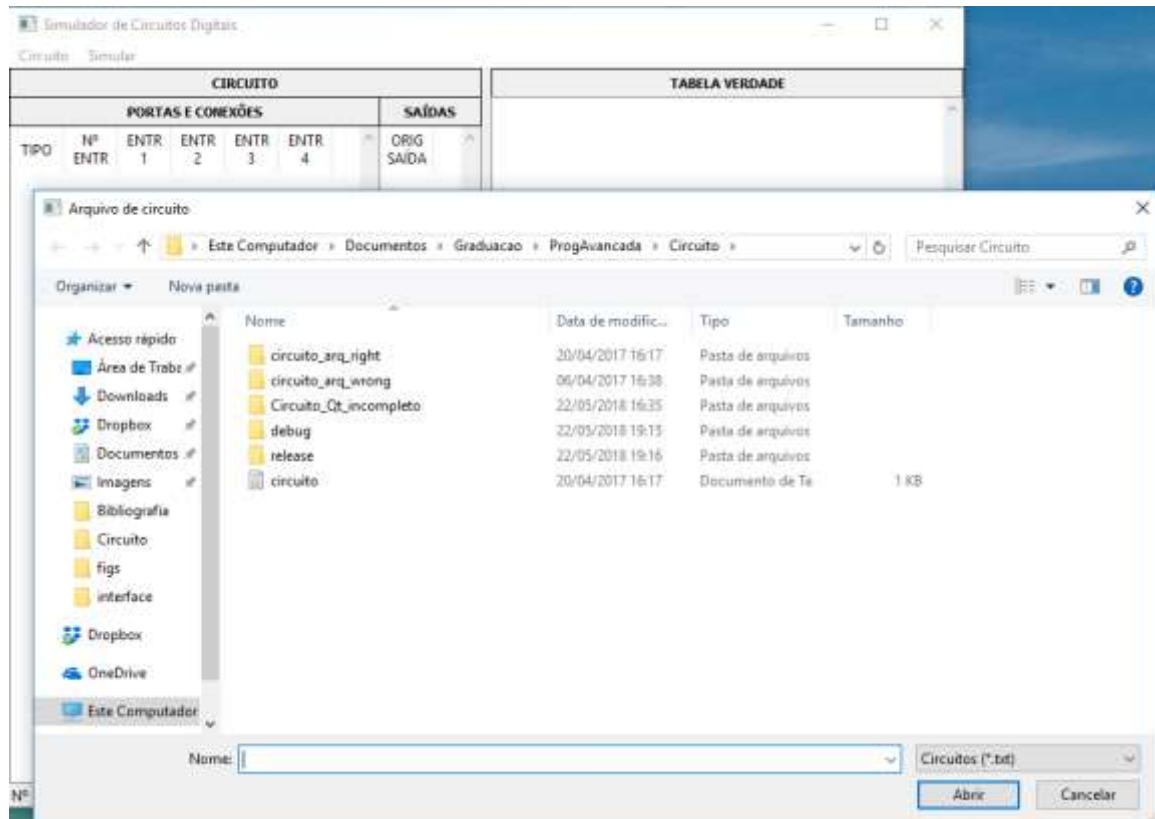


Limites e comportamento dos widgets:

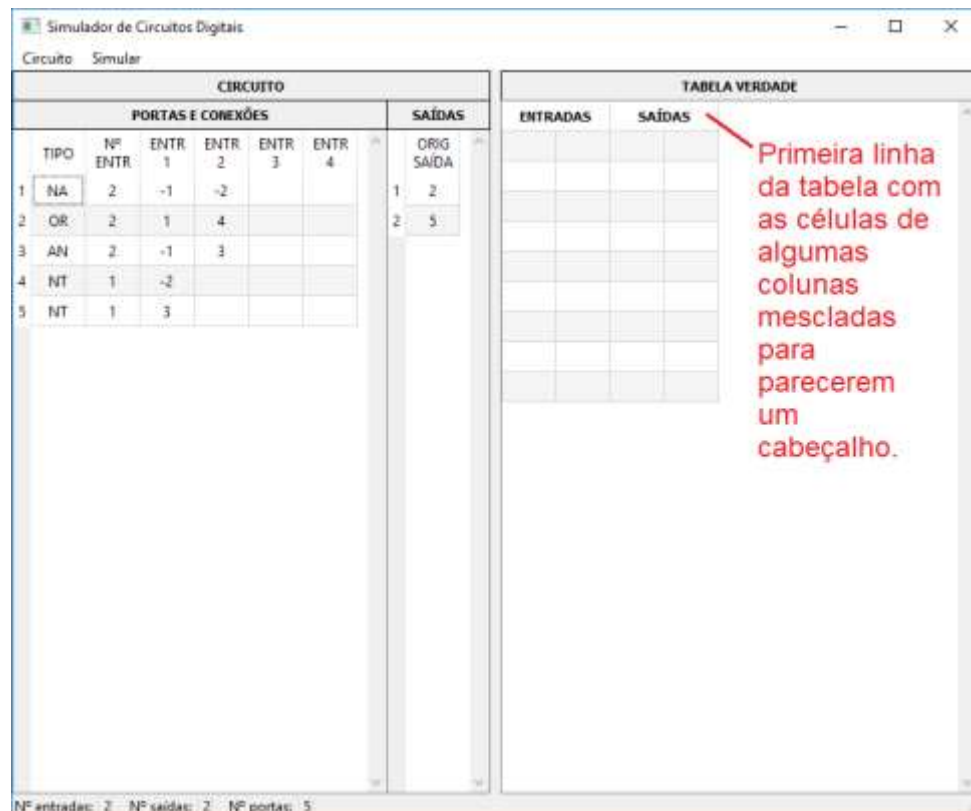
- 1) Quando modifica o valor da origem da saída, testa se o valor é inválido (zero); se for o caso, desabilita o botão OK.

```
void ModificarSaida::on_spinOrigemSaida_valueChanged(int arg1)
{
    bool saida_valida = (arg1 != 0);
    // Recupera um ponteiro para o botao OK
    QPushButton *botao_ok = ui->buttonBox->button(QDialogButtonBox::Ok);
    // Habilita ou nao o botao OK
    botao_ok->setEnabled(saida_valida);
}
```

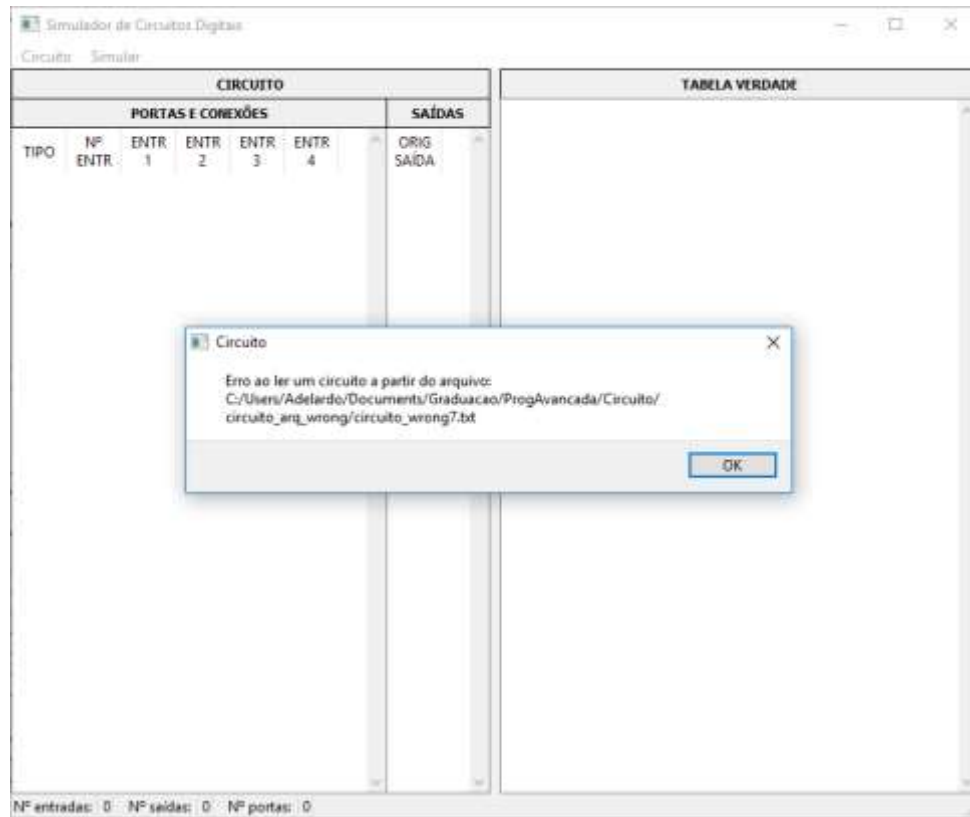
2.2 Ação Ler



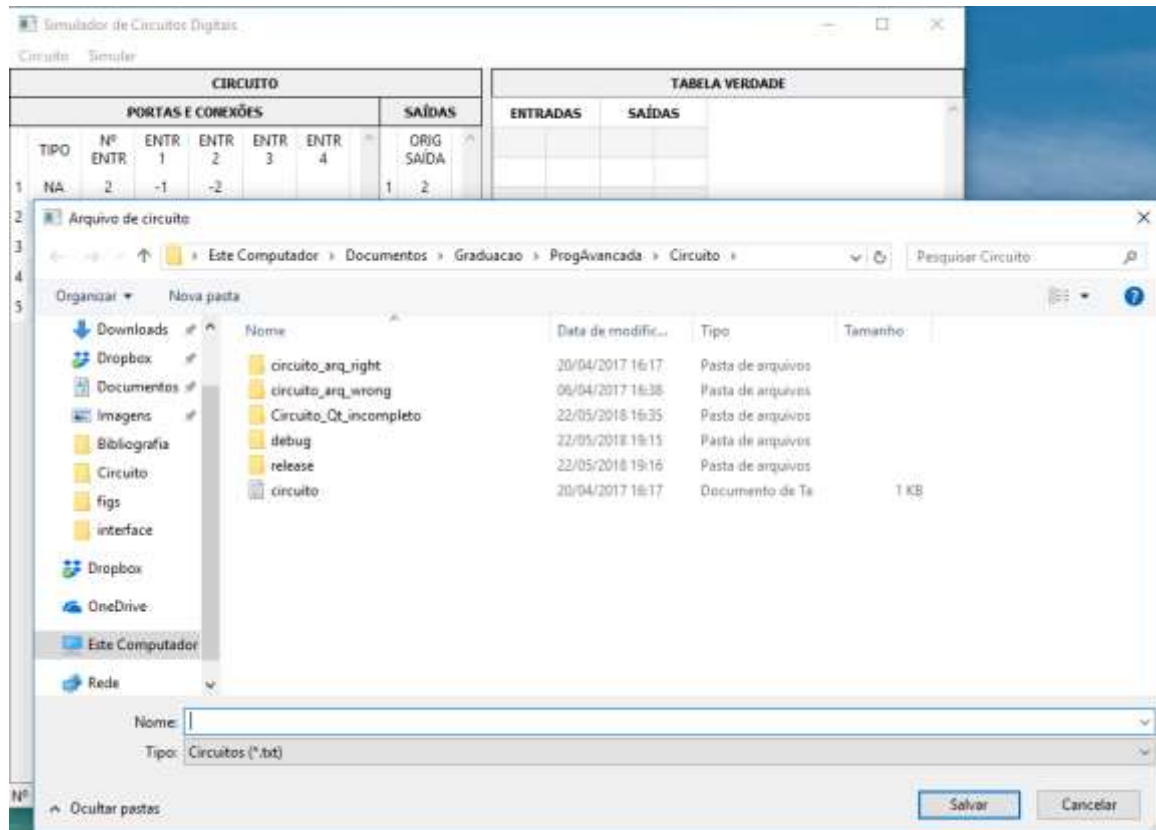
2.2.1 Tela após Ação Ler



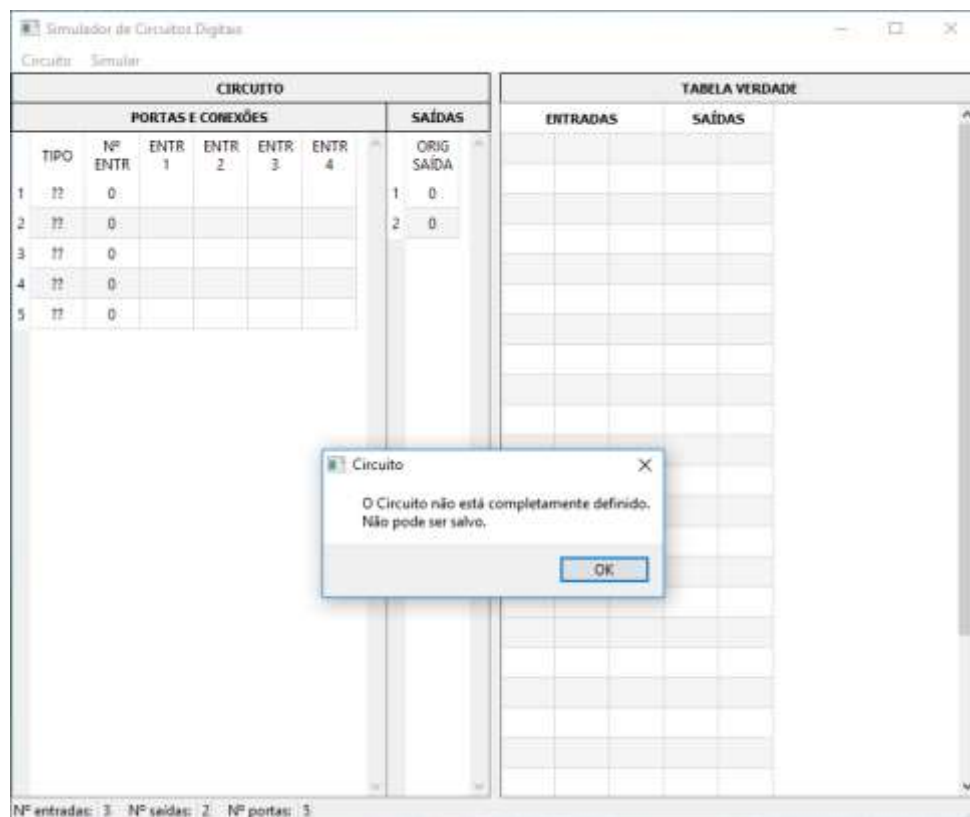
2.2.2 Erro após Ação Ler



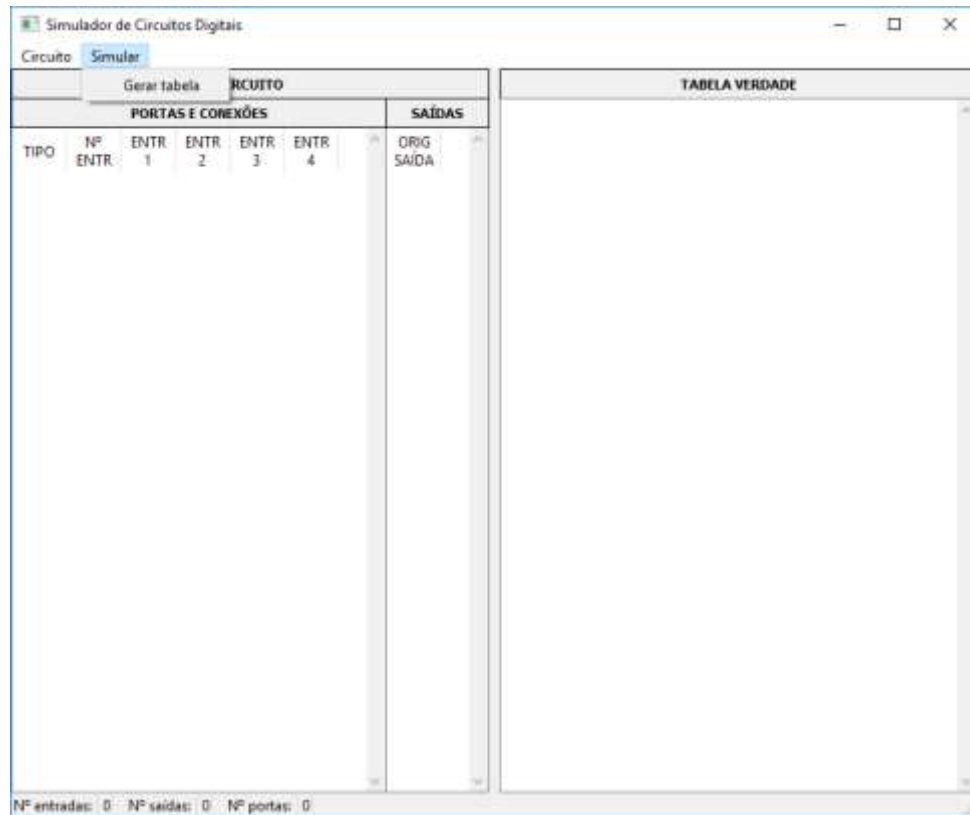
2.3 Ação Salvar



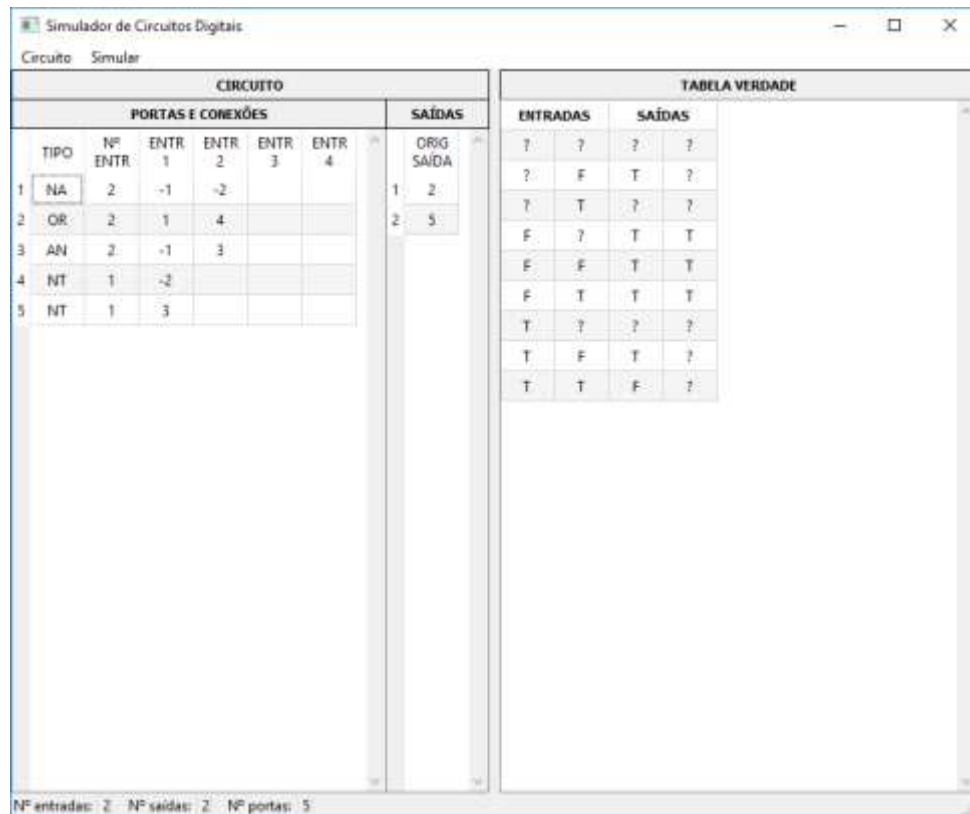
2.3.1 Erro após Ação Salvar



3. Menu simular



3.1 Ação Gerar Tabela



3.1.1 Erro após Ação Gerar Tabela

