

Relatório Técnico: Protocolo EnLayCe

Jessica de Souza, Layssa Pacheco e Schaiana Sonaglio

Outubro de 2018

1 Introdução

Este trabalho consiste no desenvolvimento do Protocolo EnLayCe: um protocolo de comunicação que possui os componentes enquadramento, ARQ e sessão.

O Protocolo EnLayCe se destina a links com baixa taxa de transmissão, de mensagens curtas, abaixo de 256 bytes, e com controle de erros. Além disso, o EnLayCe é especificado para a transmissão de forma confiável.

Para isso, especificou-se as características do protocolo EnLayCe construído e a implementação foi feita em formato de API, utilizando para o protótipo as aplicações cliente 1 e cliente 2, sendo ambas para realização de testes.

Em síntese, este relatório contém as especificações do Protocolo EnLayCe, a documentação da API e o manual de usuário, sendo os dois primeiros tópicos divididos com base nos componentes do Protocolo EnLayCe.

2 Especificação

O Protocolo EnLayCe tem por finalidade e serviço a transferência ponto-a-ponto de mensagens limitadas entre 8 a 256 bytes através da camada física do tipo UART, de forma confiável, com estabelecimento e término de sessão, garantia de entrega, controle de sequência e tratamento de erros. A codificação utilizada é do tipo binária.

O pacote usado na transferência de dados e estabelecimento de sessão entre os clientes é descrito a seguir:

- **Controle:** Este modelo usa garantia de entrega do tipo pare-e-espere, correspondendo a um byte que, em dois dois 8 *bits*, identifica:
 - bit 3:** o número de sequência, que caracteriza o modelo pare-e-espere, pois, por exemplo, se envia um dado (0) espera receber um ACK (0).
 - bit 7:** o tipo de quadro, se é referente a dados (seta *bit* 0) ou a ACK (seta *bit* 1).
- **ID Sessão:** identifica através de um byte a sessão negociada pelo gerenciamento de sessão e escolhida pelos usuários. Na garantia de entrega os quadros com ID Sessão diferentes do negociado são ignorados e descartados.

- **ID Proto:** identifica através de um byte o tipo de conteúdo transportado pelo quadro, dividindo-se entre:
 - 0:** para quadros de dados, ACK e o primeiro dado a ser mandado após o estabelecimento da sessão;
 - 255:** para quadros de gerenciamento de sessão.
- **Dados:** pode corresponder a:
 - Dado:** ao *payload* a ser transmitido pelo cliente.
 - Gerenciamento de sessão:** mensagem das transições do gerenciamento de sessão, as quais podem ser:
 - CR (Connect Request):** solicitação de estabelecimento de sessão representado pelo número 0.
 - CC (Connect Confirm):** confirmação de estabelecimento de sessão representado pelo número 1.
 - DR (Disconnect Request):** solicitação de desconexão de sessão representado pelo número 4.
 - DC (Disconnect Confirm):** confirmação de desconexão de sessão representado pelo número 5.
- **FCS:** este campo tem 2 bytes de tamanho e contém o valor do código CRC, onde se subdivide em LSB e MSB.
- **7E:** este campo tem 1 byte de tamanho e contém o delimitador de quadros, o número hexadecimal 7E, para mostrar onde começa e onde termina um quadro recebido.

O quadro tem o formato apresentado a seguir:

Quadro do Protocolo EnLayCe						
7E	Controle	ID Sessão	Proto	Dados	FCS	7E

A comunicação é iniciada quando um cliente remetente solicita o envio de dados para o cliente destinatário. Caso a conexão entre os clientes ainda não esteja estabelecida, é feito o estabelecimento de sessão, através de trocas de mensagens da classe Sessão (2.3); o primeiro quadro desta classe a ser enviado é o CR (*Connect request*, que é representado por 0). O quadro enviado é como o mostrado a seguir:

Quadro de Solicitação de Estabelecimento de Sessão						
7E	Controle	ID Sessão	255	0	FCS	7E

Para confirmar o estabelecimento da conexão, o quadro enviado pelo destinatário tem a mesma característica, diferenciando-se apenas pela mensagem de dados adicionada pela classe Sessão do Protocolo EnLayCe, a qual é CC (*Connect confirm*, que é representado por 1). O quadro é como mostrado a seguir:

Quadro de Confirmação de Estabelecimento de Sessão						
7E	Controle	ID Sessão	255	1	FCS	7E

Como o estabelecimento de conexão padronizado para o protocolo foi o do tipo *3-way handshake*, o remetente envia um quadro vazio, representando que recebeu a confirmação de conexão do destinatário. Este quadro é mostrado a seguir:

Quadro de Confirmação de Estabelecimento de Sessão com conteúdo zerado

7E	Controle	ID Sessão	0	0	FCS	7E
----	----------	-----------	----------	----------	-----	----

Agora ambos os lados estão conectados. Depois de estabelecida a sessão, ocorre o envio contínuo de dados e seus respectivos ACKs, os quais são representados de acordo com os seguintes quadros:

Quadro de Dados

7E	000X0000	ID Sessão	0	Dados	FCS	7E
----	----------	-----------	----------	-------	-----	----

O quadro de Ack não possui o campo "dados", pois não envia nenhuma mensagem (*payload*).

Quadro de ACK

7E	000X0001	ID Sessão	0	FCS	7E
----	----------	-----------	----------	-----	----

Já para o encerramento da sessão, o remetente ou o destinatário faz o pedido de desconexão, enviando um quadro com uma mensagem DR (*Disconnect request*, que é representado por 4). O quadro é como mostrado a seguir:

Quadro de Solicitação de Encerramento de Sessão

7E	Controle	ID Sessão	255	4	FCS	7E
----	----------	-----------	------------	----------	-----	----

Neste momento, a conexão está parcialmente fechada. O lado que recebeu o quadro com o DR retorna um quadro com outra mensagem DR. Neste momento, ambos os lados estão cientes da desconexão. O lado que enviou o primeiro DR envia então uma mensagem DC (*Disconnect confirm*, que é representado por 5). O quadro é como mostrado a seguir.

Quadro de Confirmação de Encerramento de Sessão

7E	Controle	ID Sessão	255	5	FCS	7E
----	----------	-----------	------------	----------	-----	----

Entre o segundo DR e o primeiro DC, o lado que possui envio de dados pendente faz o envio destes dados. Por fim, quando o DC é recebido, é encerrado o envio de dados e o último DC é enviado e após isto a sessão é encerrada.

A seguir cada subcamada usada no Protocolo EnLayCe é explicada detalhadamente.

2.1 Enquadramento

A subcamada Enquadramento é responsável por realizar a delimitação dos quadros na interface com a camada física, através da técnica Sentinela, utilizada pelo protocolo PPP, e por realizar o controle de erros.

O Enquadramento utiliza como delimitador o valor hexadecimal 7E, no início e no fim do quadro. Caso o quadro possua em sua mensagem algum valor 7E, é realizado um "mascaramento" deste valor através de uma operação XOR(7E,20), tendo como resultado o valor 5E. Para sinalizar este byte modificado (5E), usa-se como byte de escape o valor 7D para preenchimento de octeto, transmitindo primeiramente o byte de sinalização, 7D para, posteriormente, enviar o 5E. Caso na mensagem exista o valor 7D, também é feito o mascaramento através do XOR20, resultando num byte 5D; idêntico ao byte 7E, o 5D também é enviado depois do byte de sinalização 7D.

Além dos delimitadores de quadro, o Enquadramento evita a transmissão de quadros corrompidos. Para isto, no lado transmissor, são adicionados 2 bytes de redundância, cujos valores se baseiam no conteúdo da mensagem, que é utilizada para realizar o cálculo dos bytes, através do método CRC-16; este método é uma verificação de redundância do tipo cíclica e utiliza álgebra polinomial para o cálculo. No lado receptor, é realizado o mesmo cálculo do CRC e depois se compara o CRC transmitido com o calculado no receptor, caso haja diferença entre eles, o pacote é descartado.

Um exemplo de como é a visão do quadro pela subcamada Enquadramento é mostrado abaixo:

Quadro na subcamada Enquadramento

7E	Dados	CRC	7E
----	-------	-----	----

2.2 ARQ

A subcamada ARQ é responsável pela garantia de entrega do Protocolo EnLayCe, de forma a possibilitar que o transmissor se certifique de que a mensagem enviada foi entregue ou não ao destino.

Dentre os campos do cabeçalho do quadro trabalhado no Protocolo EnLayCe, a subcamada ARQ é responsável pelo Controle e ID Sessão. Nessa etapa da comunicação o ID Sessão é verificado, a fim de confirmar se os quadros manipulados correspondem a sessão estabelecida. Contrário o ID Sessão seja diferente, o quadro é descartado.

O Controle é responsável por identificar se o quadro é do tipo Dado ou ACK, correspondendo, respectivamente, aos valores 0 ou 1 do bit 7 do byte Controle. Assim como, a qual sequência de dados este dado ou ACK consiste, ou seja, se o dado enviado é da sequência 0, o ACK recebido precisa ser da sequência 0 e, se o dado enviado é da sequência 1, o ACK recebido precisa ser da sequência 1; respeitando assim o modelo pare-e-espere.

Um exemplo de como é a visão do quadro pela subcamada ARQ é mostrado abaixo:

Quadro na subcamada ARQ

Controle	ID Sessão	Dados
----------	-----------	-------

Após as formatações da subcamada Sessão, a subcamada ARQ completa o quadro com os campos de sua responsabilidade e o envia para a subcamada Enquadramento.

2.3 Sessão

A subcamada Sessão é responsável pelo estabelecimento e terminação da conexão entre um par de participantes. Isso porque, como o EnLayCe é um protocolo ponto-a-ponto, é necessário estabelecer um enlace entre as duas pontas participantes antes de poder transferir os dados. Com isso, evita-se que as transmissões entre um par de participantes sejam confundidas com transmissões de outros pares.

Dentre os campos do cabeçalho do quadro trabalhado no Protocolo EnLayCe, a subcamada Sessão é responsável pelo Proto e Dados. Quando o campo Dados é preenchido com os dados a serem enviados, o byte do campo Proto assume valor 0. Já quando o campo Dados é preenchido com informações sobre estabelecimento ou encerramento de sessão, o byte do campo Proto assume valor 255.

As etapas de estabelecimento ou encerramento de sessão são diferenciadas de acordo com o valor adicionado no campos Dados, os quais podem ser, conforme explicado na seção Especificação (2), CR, CC, DR e DC, com valores 0, 1, 4 e 5, respectivamente.

O Cliente Remetente estabelece uma sessão ao solicitar o envio de dados para o Cliente Destinatário. Da mesma maneira, quando deseja encerrar uma sessão, ambos os clientes podem solicitar a desconexão. Dessa forma, sempre que um dos pedidos é feito, o Protocolo EnLayce chama na subcamada Sessão o método correspondente, seja para estabelecer ou encerrar uma sessão, ou para enviar dados e ACK.

Depois de processar os campos ao qual é responsável, a subcamada Sessão envia os campos para a subcamada ARQ para continuar a construção do quadro.

Um exemplo de como é a visão do quadro pela subcamada Sessão é mostrado abaixo:

Quadro na subcamada Sessão

Proto	Dados
-------	-------

2.4 Diagramas e máquinas de estado

Abaixo, representados em algumas figuras, estão os diagramas e as máquinas de estado do Protocolo EnLayCe.

No diagrama da figura abaixo, é mostrada a arquitetura do *software* do protocolo, ou seja, as suas subcamadas e, no texto em sequência, uma breve explicação de como elas se relacionam entre si.

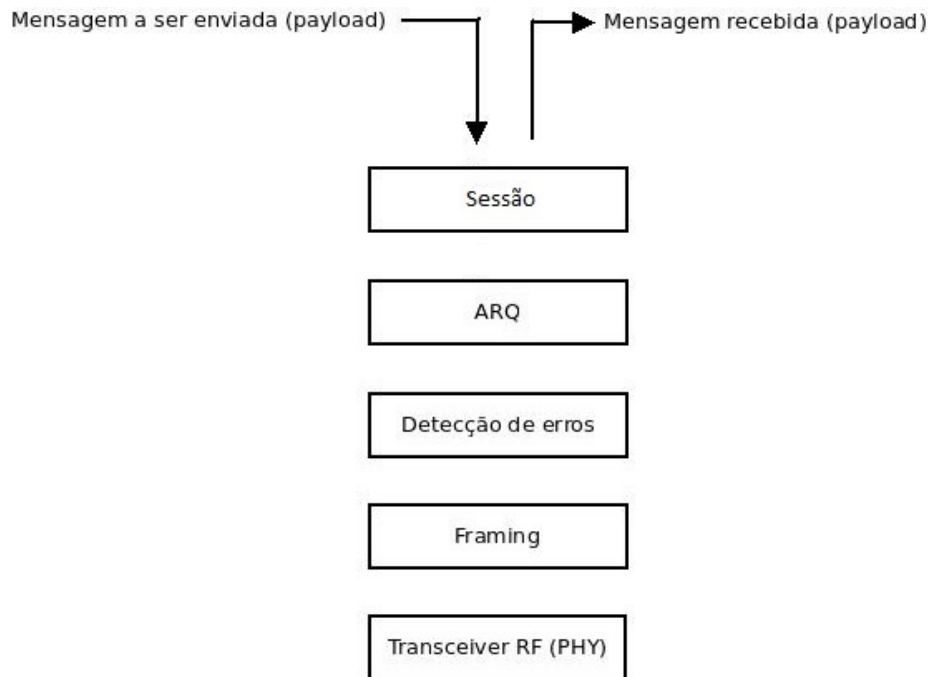


Figura 1: Arquitetura do protocolo. Fonte: Material disponibilizado pelo professor modificado.

- Cenário 1: Cliente envia *payload*: Quando a aplicação solicita o envio de um *payload*, primeiramente é estabelecida uma sessão entre este cliente e o cliente que irá receber, através de quadros da classe Sessão, onde é colocado o valor de proto. Após isso, o quadro é incrementado na classe ARQ com o byte de controle, para gerenciar se é dado ou ACK e para saber a sequência que está, além de acrescentar o ID da sessão corrente. Depois, o quadro vai para a classe Enquadramento, onde é calculado um CRC e colocado os delimitadores de início e término de quadro. Após passar por todas estas camadas, o quadro é enviado pela Classe Serial.
- Cenário 2: Cliente recebe quadro: Quando o quadro é recebido pela classe Serial do cliente remetente, a primeira ação é da classe Enquadramento, que calcula o CRC novamente e compara com o CRC do quadro recebido,

após isso, ela retira o delimitadores de quadro e passa o quadro para a Classe ARQ, que verifica se é um quadro daquela sessão corrente e de que se trata de um quadro de dados, após isso, envia o respectivo ACK e passa o quadro para a classe Sessão, que tira o campo proto e, enfim, repassa o *payload* para o cliente remetente.

No diagrama de classes, foram exemplificados os métodos e variáveis de cada subcamada.

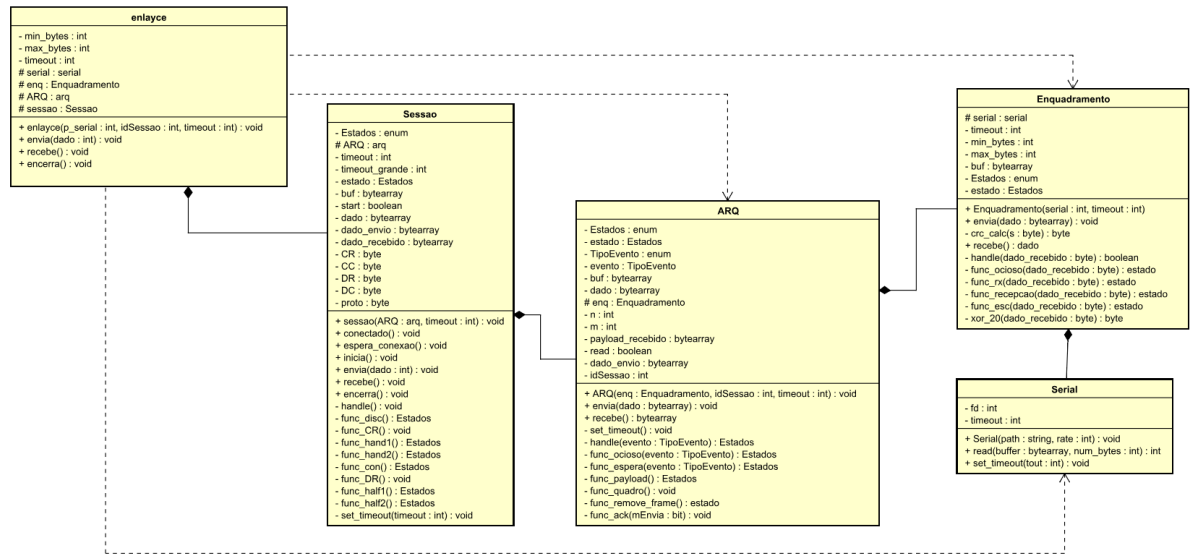


Figura 2: Diagrama de classes do protocolo. Fonte: Elaboração própria.

Já nas máquinas de estado, é mostrado o comportamento de cada subcamada. Como a implementação foi feita seguindo o material disponibilizado pelo professor, as máquinas foram retiradas de lá.

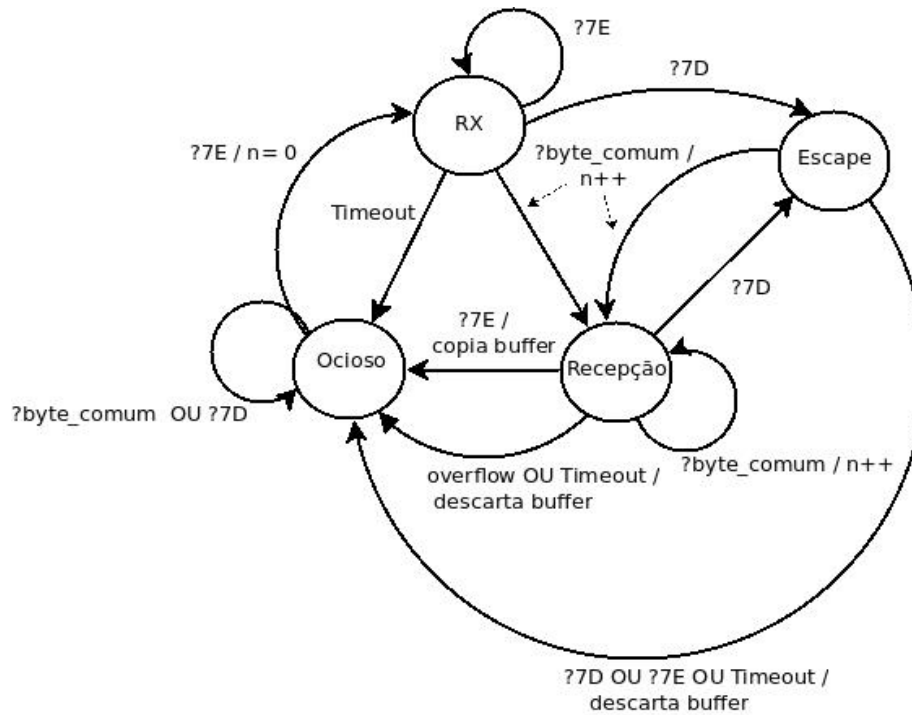


Figura 3: Máquina de estados da classe Enquadramento. Fonte: Material disponibilizado pelo professor.

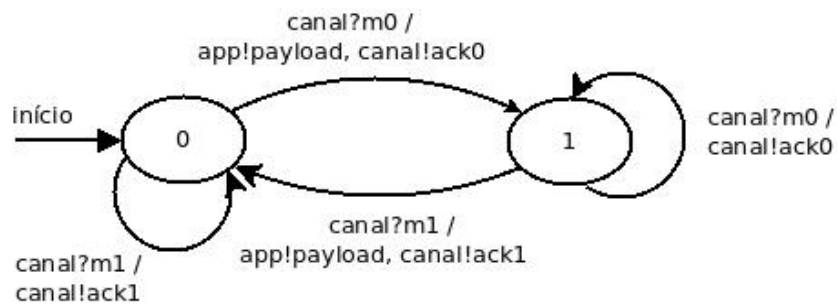


Figura 4: Máquina de estados da classe ARQ. Fonte: Material disponibilizado pelo professor.

Na máquina de estados da classe Sessão, mostrada na Figura 5, não foi implementado o estado "Check", pois este estado teria que acontecer através de uma ação do protocolo, a necessidade de fazer manutenção na conexão, e não se encaixava no modelo proposto para este projeto.

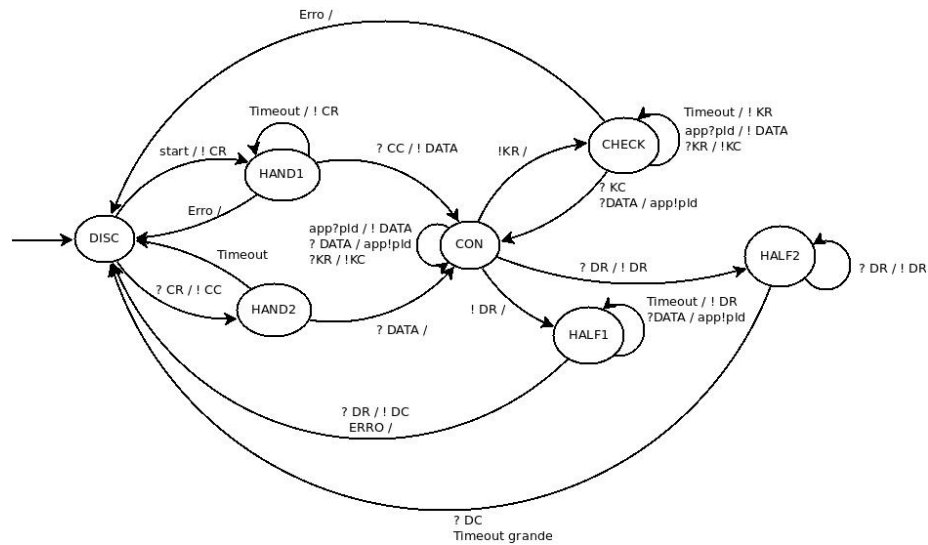


Figura 5: Máquina de estados da classe Sessão. Fonte: Material disponibilizado pelo professor.

3 Documentação da API

A Tabela 1 apresenta a utilização da API EnLayCe descrevendo suas funções e aplicação de uso.

Tabela 1: Funções da API EnLayCe

Função	Descrição	Aplicação
enlayce.envia(dado)	Função que envia um dado (<i>payload</i>) de uma aplicação para outra.	Cliente
enlayce.recebe()	Função que espera receber um dado de uma aplicação.	Cliente
enlayce.encerra()	Função que encerra a conexão estabelecida entre as aplicações.	Cliente

4 Manual do Usuário

Para executar as aplicações teste e verificar o funcionamento do protocolo EnLayCe, siga os passos abaixo:

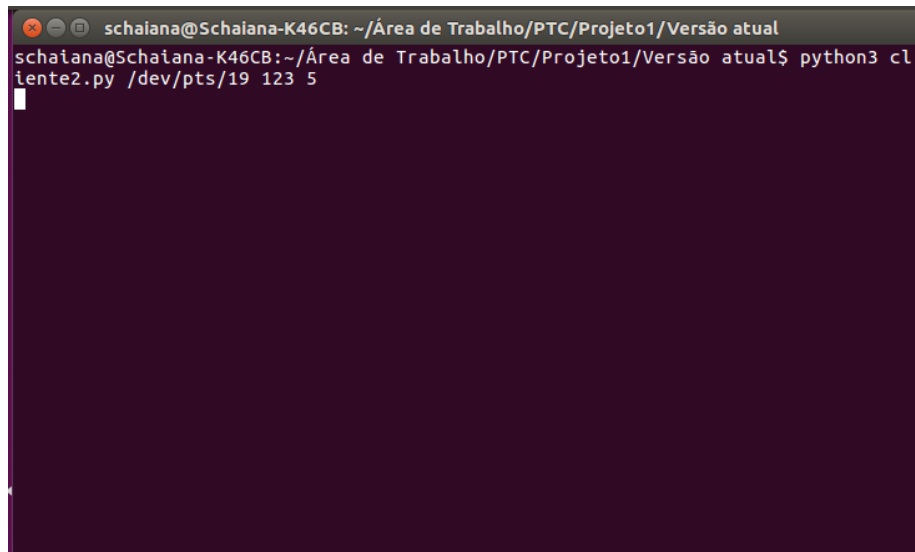
- Coloque os arquivos cliente1.py, cliente2.py, enlayce.py, sessao.py, ARQ.py e enquadramento.py numa mesma pasta.
- Execute a aplicação cliente1.py passando como argumento o número da porta serial, o ID da sessão escolhido e o tempo de timeout, conforme ilustrado na 6. Exemplo: `python3 cliente1.py /dev/pts/18 123 5`



```
schaiana@Schaiana-K46CB: ~/Área de Trabalho/PTC/Projeto1/Versão atual
schaiana@Schaiana-K46CB:~/Área de Trabalho/PTC/Projeto1/Versão atual$ python3 cliente1.py /dev/pts/18 123 5
Digite um dado para transmissão:
█
```

Figura 6: Exemplo de uso da aplicação do cliente 1. Fonte: Elaboração Própria.

- Execute a aplicação cliente2.py passando como argumento o número da porta serial, o ID da sessão escolhido e o tempo de timeout, conforme ilustrado na 7. Exemplo: `python3 cliente2.py /dev/pts/19 123 5`

A terminal window with a dark background and light text. The title bar at the top reads 'schaiana@Schalana-K46CB: ~/Área de Trabalho/PTC/Projeto1/Versão atual'. The command prompt shows the user 'schaiana@Schalana-K46CB' at the directory '~/Área de Trabalho/PTC/Projeto1/Versão atual' running the command 'python3 cliente2.py /dev/pts/19 123 5'. The rest of the terminal is empty, indicating the command has been executed successfully without output.

```
schaiana@Schalana-K46CB: ~/Área de Trabalho/PTC/Projeto1/Versão atual
schaiana@Schalana-K46CB:~/Área de Trabalho/PTC/Projeto1/Versão atual$ python3 cliente2.py /dev/pts/19 123 5
```

Figura 7: Exemplo de uso da aplicação do cliente 2. Fonte: Elaboração Própria.

Seguindo os passos acima, você poderá enviar mensagens de uma cliente para outro.