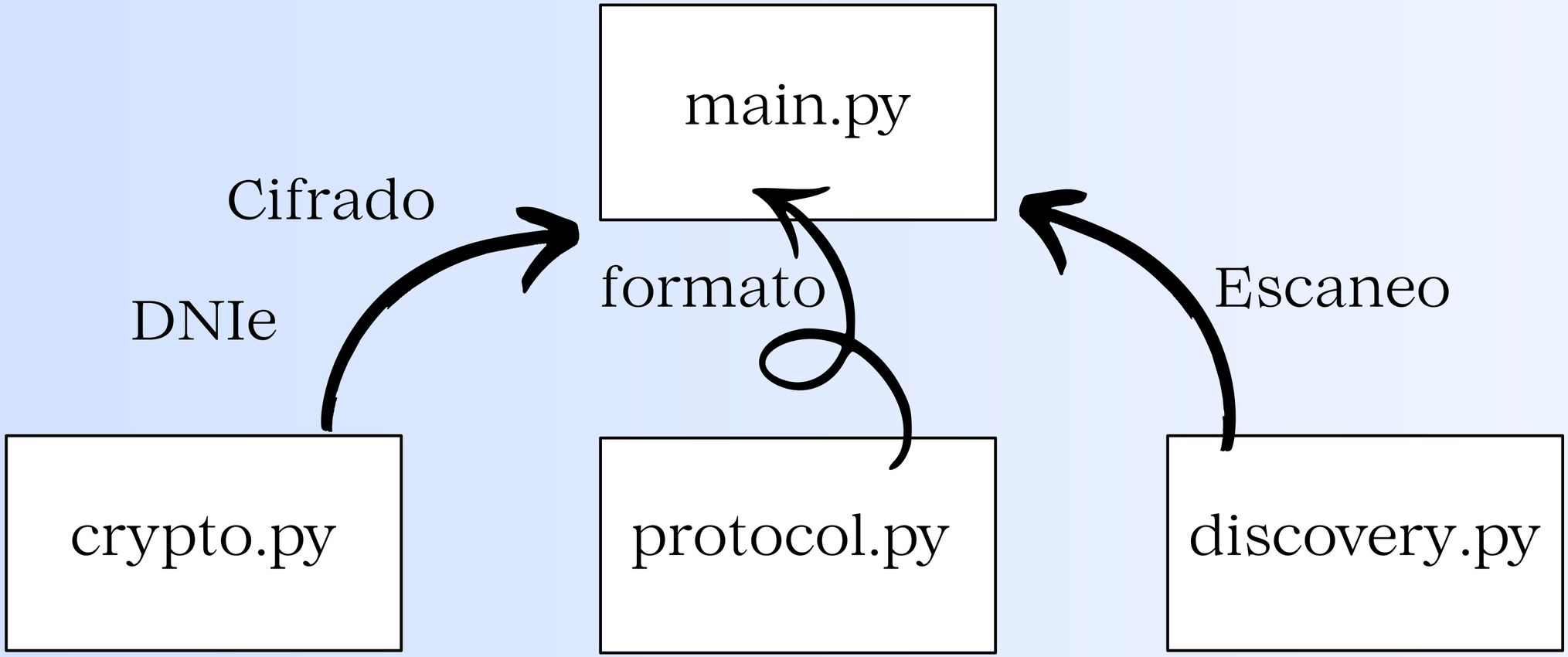
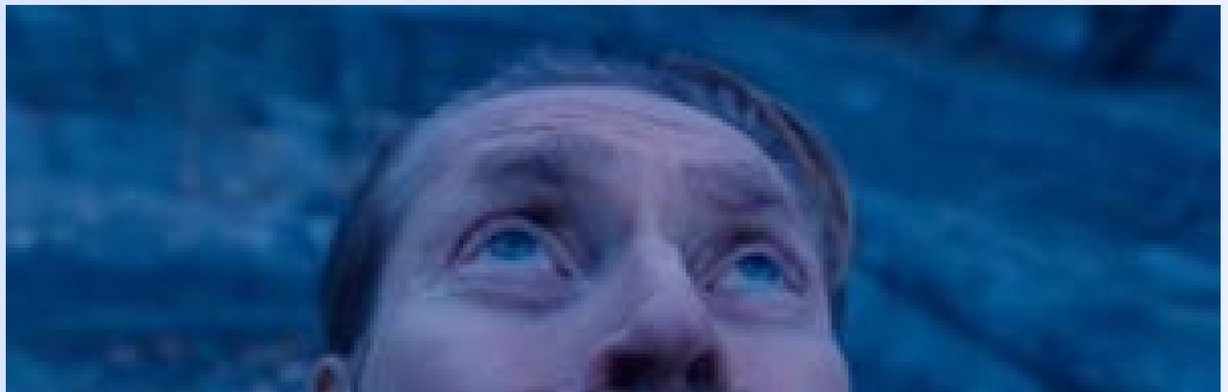
Mensajería con DNIe

Sistema de Mensajería con DNIe

Presentado por Marcos Fraile y Arturo Francés

Programas usados





Formato de tramas

Primer byte

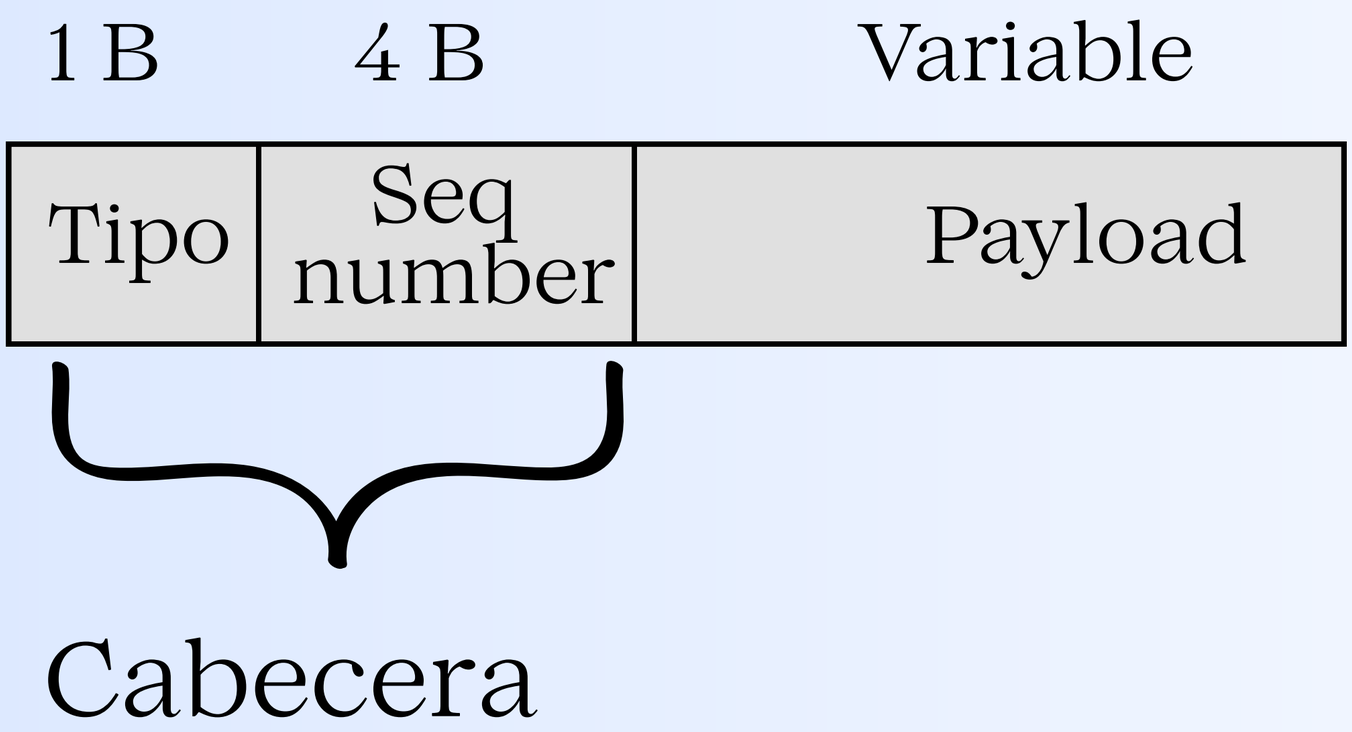
MSG_HELLO = 1

MSG_DATA = 2

MSG_AUTH = 3

MSG_ACK = 4

MSG_BYE = 5



Protocolo

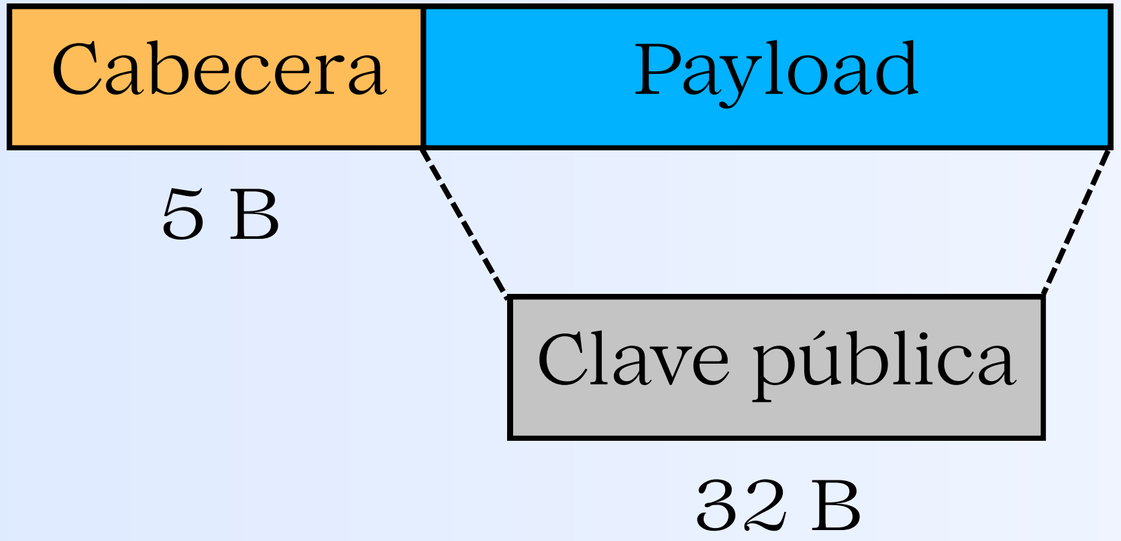
```python
# Cada vez que entran bytes por la red los procesamos para construir el mensaje.
def datagram_received(self, data, addr):
    try:
        if len(data) < 5: return
        header = data[:5]
        msg_type, seq = struct.unpack("!BI", header)
        payload = data[5:]

        # Creamos el paquete con los valores sacados de los bytes recibidos.
        self.on_packet(ChatPacket(msg_type, seq, payload), addr)
    except: pass


# Esta función se llama cuando queremos enviar un mensaje a una dirección ip:puerto.
def send_packet(self, ip, port, msg_type, seq, payload):
    if self.transport:
        header = struct.pack("!BI", msg_type, seq)
        if isinstance(payload, str): payload = payload.encode('utf-8')
        try: self.transport.sendto(header + payload, (ip, port))
        except: pass
```

Encriptado y desencriptado

```
def encrypt(self, txt):
    if not self.cipher: raise Exception("No Key")
    n = os.urandom(12)
    return n + self.cipher.encrypt(n, txt.encode(), None)


def decrypt(self, data):
    if not self.cipher: raise Exception("No Key")
    return self.cipher.decrypt(data[:12], data[12:], None).decode()
```

Tipos de Payload

MSG_HELLO = 1

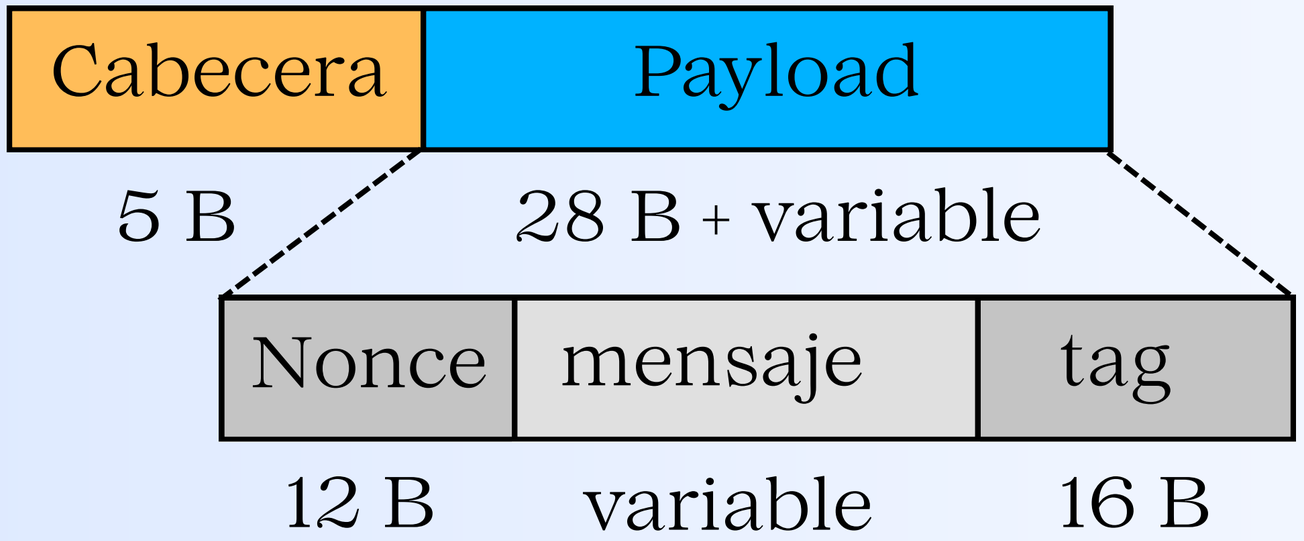


192.168.61.211	192.168.61.46	UDP	79 9999 → 8888 Len=37
192.168.61.46	192.168.61.211	UDP	79 8888 → 9999 Len=37

```
Data (37 bytes)
    Data: 01000000008f26874831fe315f5f06e45c4fedb4bd2388bd2e71e1acaa7873f1be45995a40
    [Length: 37]
```

Tipos de Payload

MSG_DATA = 2

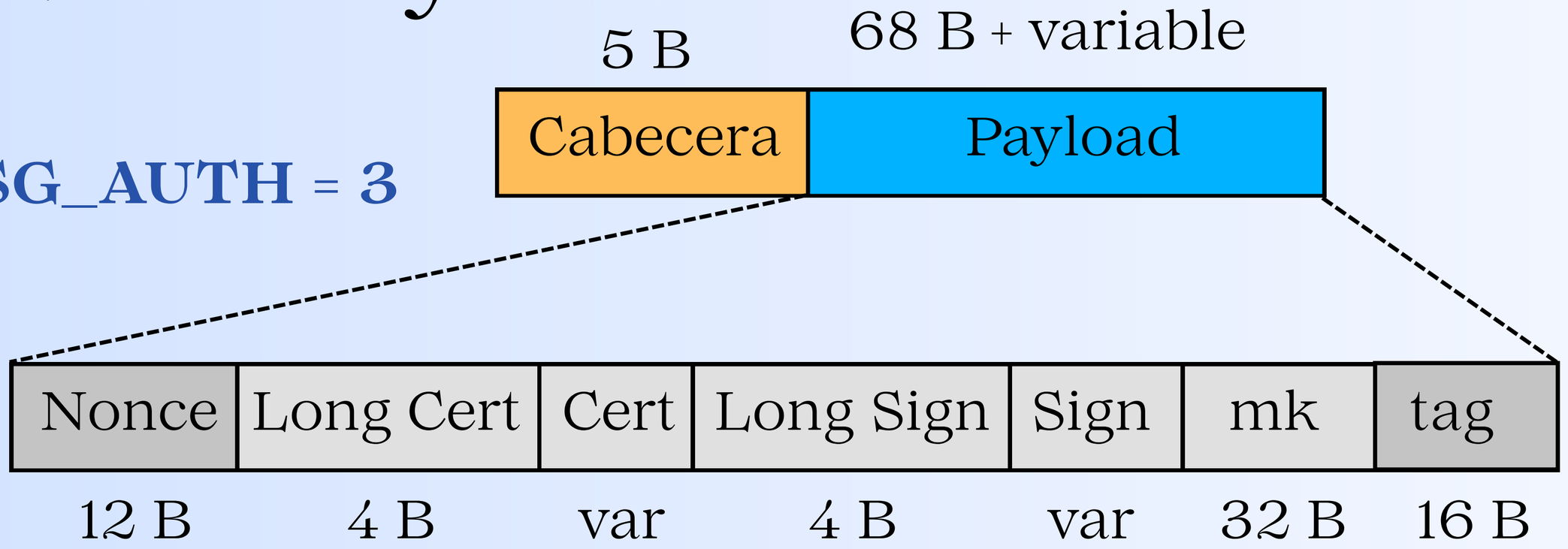



```
192.168.61.211      192.168.61.46       UDP      115 9999 → 8888 Len=73
```

```
Data (73 bytes)
    Data: 020000000001aae710b58924639d9394554a669e65d87ce94437e543b5
    [Length: 73]
```

Tipos de Payload

MSG_AUTH = 3

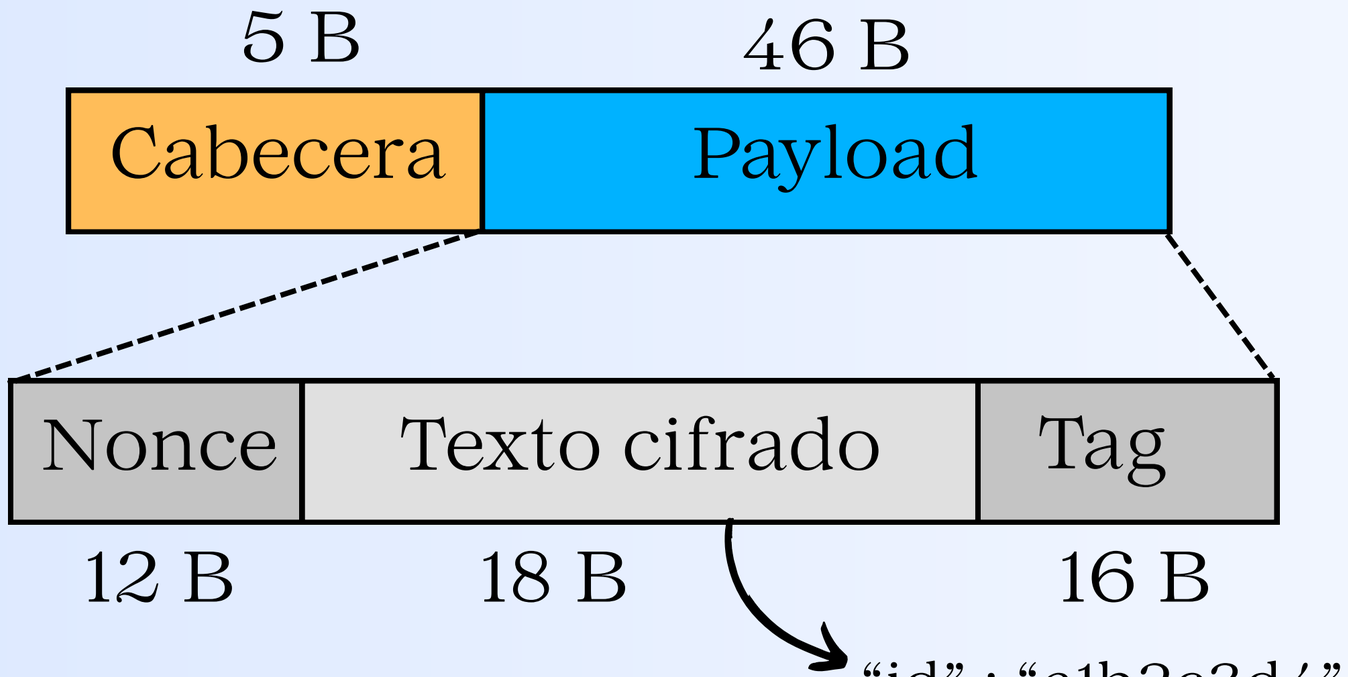


192.168.61.46	192.168.61.211	UDP	1402	8888 → 9999 Len=2840
192.168.61.211	192.168.61.46	UDP	1429	9999 → 8888 Len=2867

```
Data (2840 bytes)
    Data [truncated]: 030000000000b0d2f322d32af7571e30e6e72bd8e55d46c1f7ac47b5a6450e1829164b6
    [Length: 2840]
```

Tipos de Payload

MSG_ACK = 4



192.168.61.46	192.168.61.211	UDP	93	8888 → 9999 Len=51
192.168.61.46	192.168.61.211	UDP	93	8888 → 9999 Len=51
192.168.61.46	192.168.61.211	UDP	93	8888 → 9999 Len=51

```
Data (51 bytes)
    Data: 0400000000820dece185f3f9985fa017df5138cf77a2d89d8452f82139de80438
    [Length: 51]
```

Tipos de Payload

MSG_BYE = 5

5 B

Cabecera (no hay payload)

192.168.61.46	192.168.61.211	UDP	47 8888 → 9999 Len=5
192.168.61.46	192.168.61.211	UDP	47 8888 → 9999 Len=5
192.168.61.46	192.168.61.211	UDP	47 8888 → 9999 Len=5
192.168.61.46	192.168.61.211	UDP	47 8888 → 9999 Len=5
192.168.61.46	192.168.61.211	UDP	47 8888 → 9999 Len=5

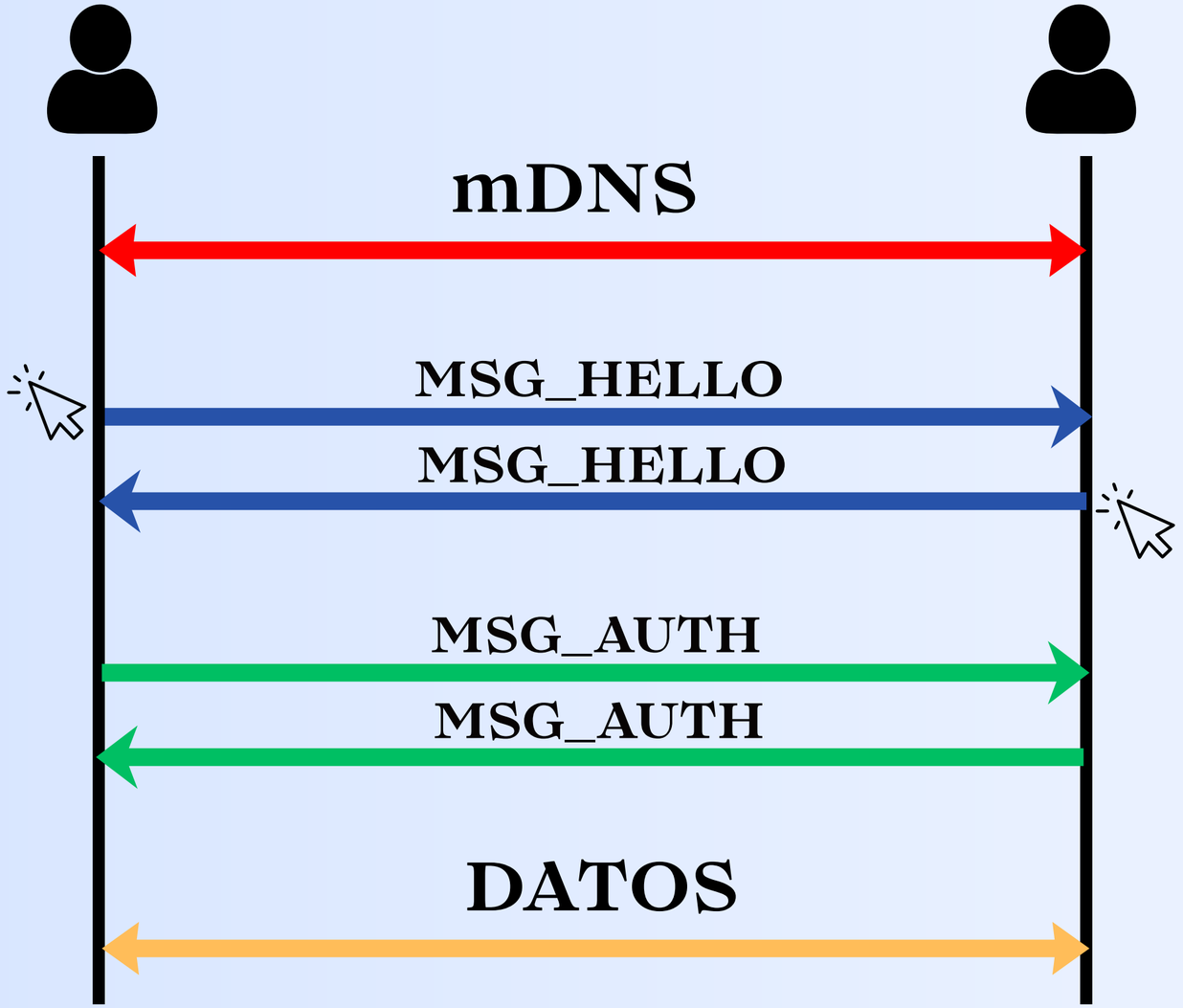
```
Data (5 bytes)
    Data: 0500000000
    [Length: 5]
```

Envío mensajes

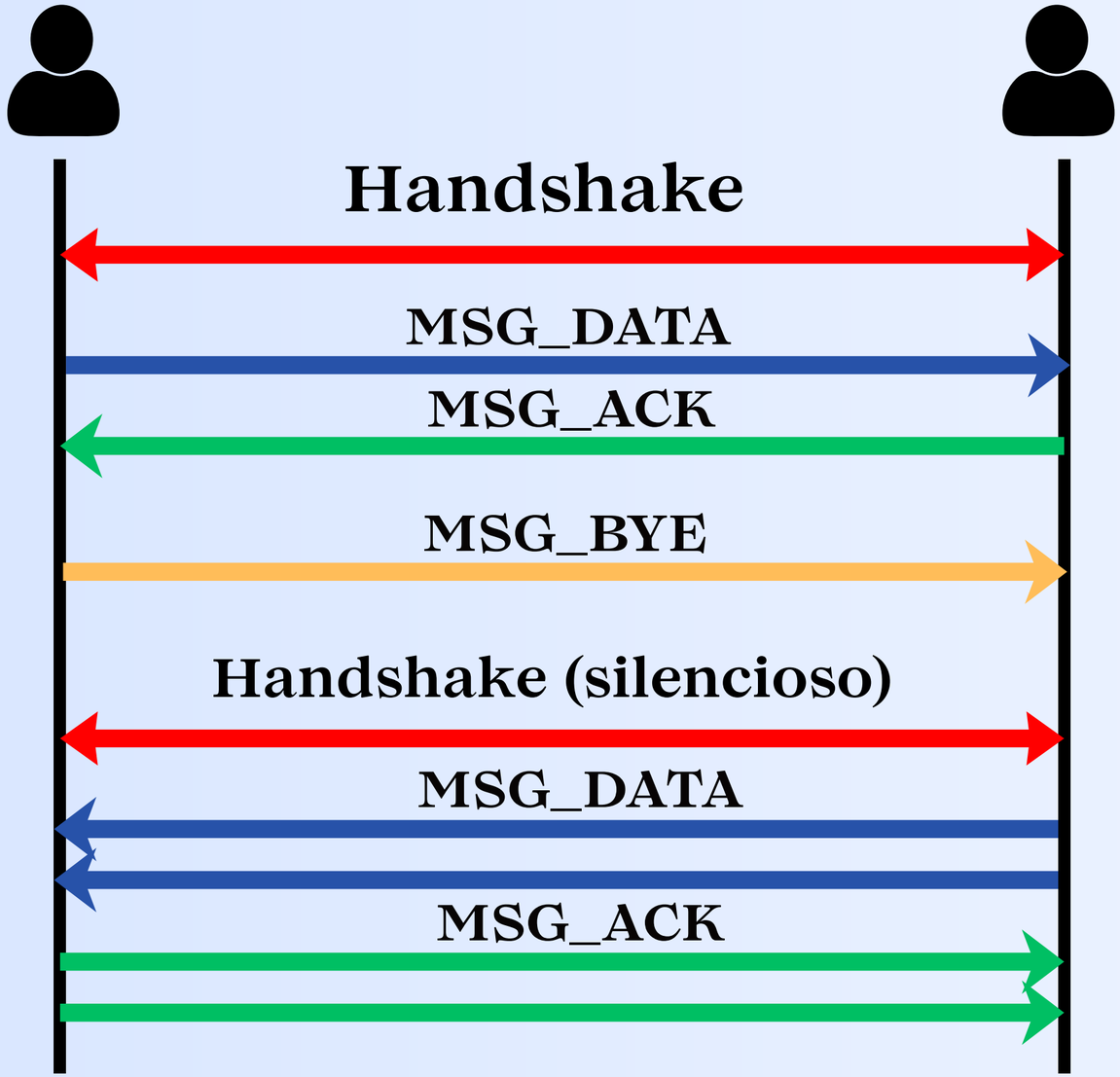
```python
# Función para enviar mensajes.
def send_msg(self, text):
    if STATE.target_info:
        ip, port = STATE.target_info
        sk = f"{ip}:{port}"

        # ID aleatorio generado para un mensaje.
        mid = str(uuid.uuid4())[:8]

        # Si está OFFLINE, lo guardamos en cola.
        if sk in self.offline_peers:
            if sk not in self.message_queue: self.message_queue[sk] = []

            msg_data = {
                'id': mid, 'sender': self.name, 'text': text,
                'is_me': True, 'is_sys': False,
                'time': datetime.now().strftime("%H:%M"), 'status': 'queued'
            }
            self.message_queue[sk].append(msg_data)

            STATE.add_message(self.name, text, True, mid=mid, status='queued')
            self.save_msg_to_history(sk, msg_data)
            self.save_sessions_securely()
            return
```

```python
# Si está ONLINE, lo enviamos.
try:
    pl = json.dumps({"id": mid, "msg": text})
    enc = self.sessions[sk].encrypt(pl)
    self.protocol.send_packet(ip, port, MSG_DATA, 1, enc)
    m_obj = STATE.add_message(self.name, text, True, mid=mid)
    self.save_msg_to_history(sk, m_obj)
except: pass
```

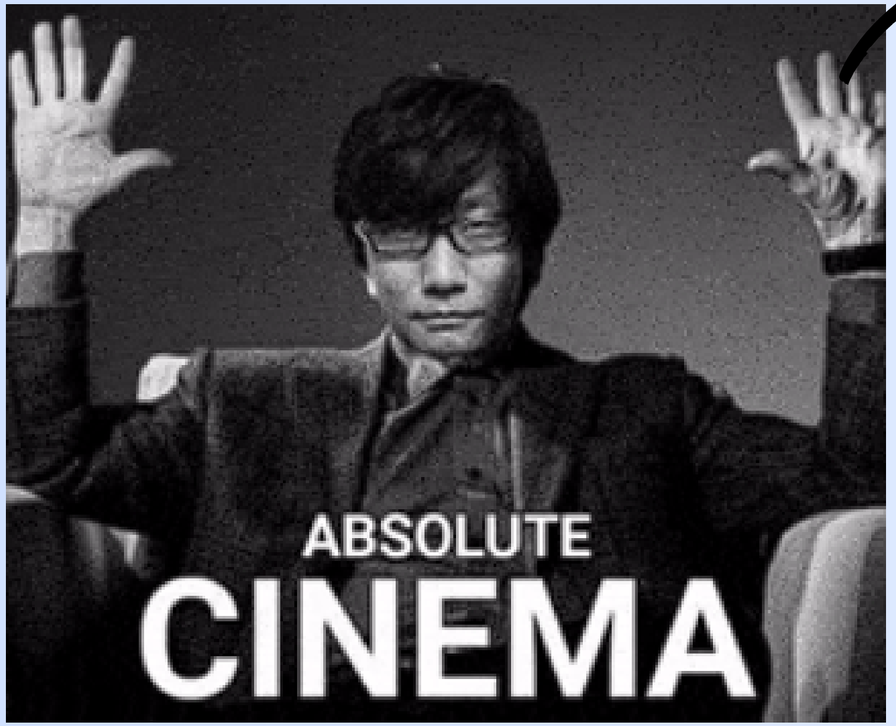
Ejemplo HANDSHAKE



Ejemplo transmisión



Interfaz


ABSOLUTE
CINEMA


CONECTADO: MARIA [OK]
140.85
MEMORY
Maria [OK]
Lucia [OK]
Carlos
Juan
Hola María
22:56 [√√]
Hola Paco, que tal estás??
22:56
Bueno...
22:56 [√√]
De exámenes
22:56 [√√]
LOL, que pringado
22:56
SI BUENAS NOTAS QUIERES TENER
VIDA SOCIAL NO DEBES HACER
EFECTIVA COMUNICACION
IMPORTANTE ES
memegenerator.es

SOMOS EL WHATSAPP 2