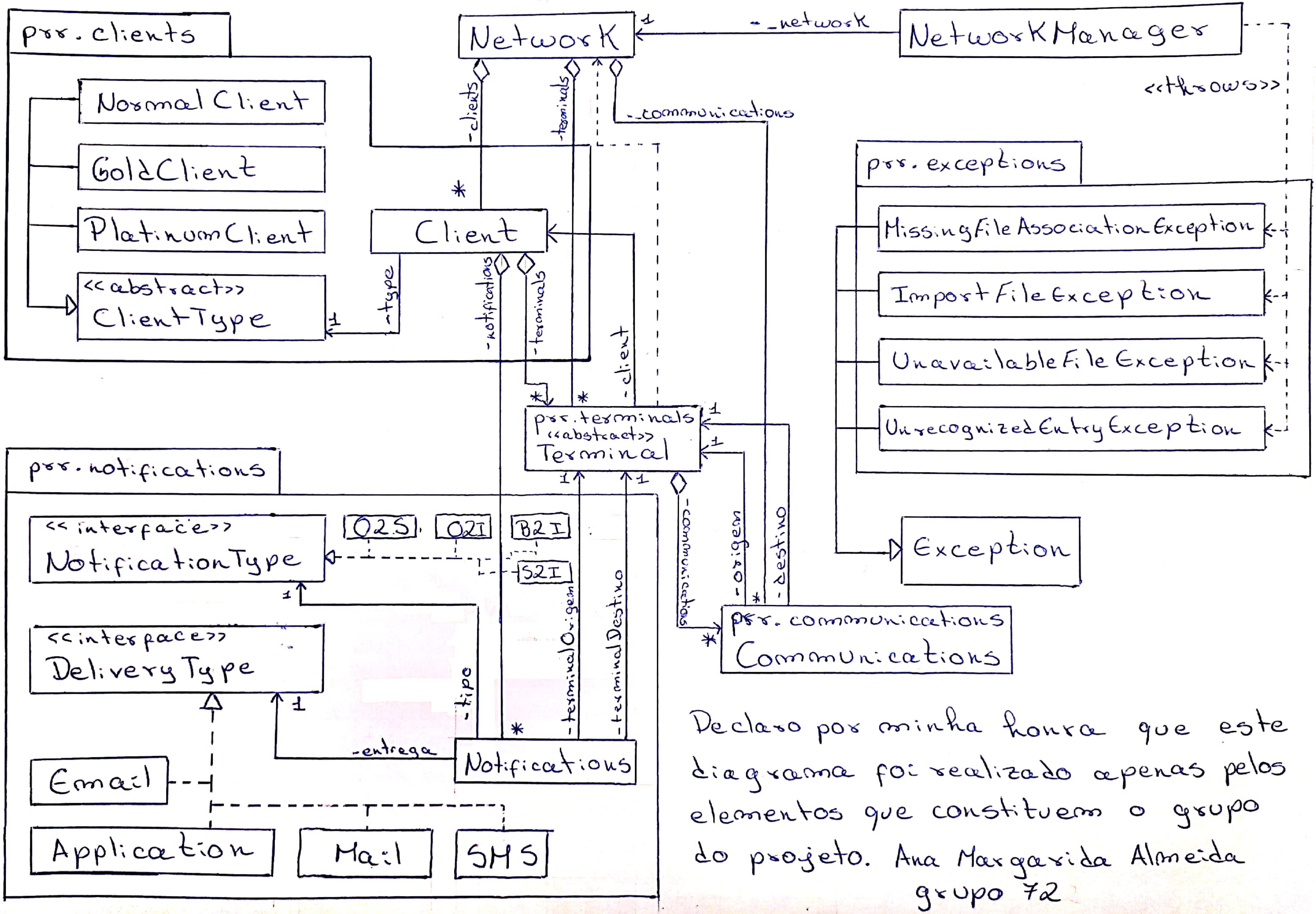
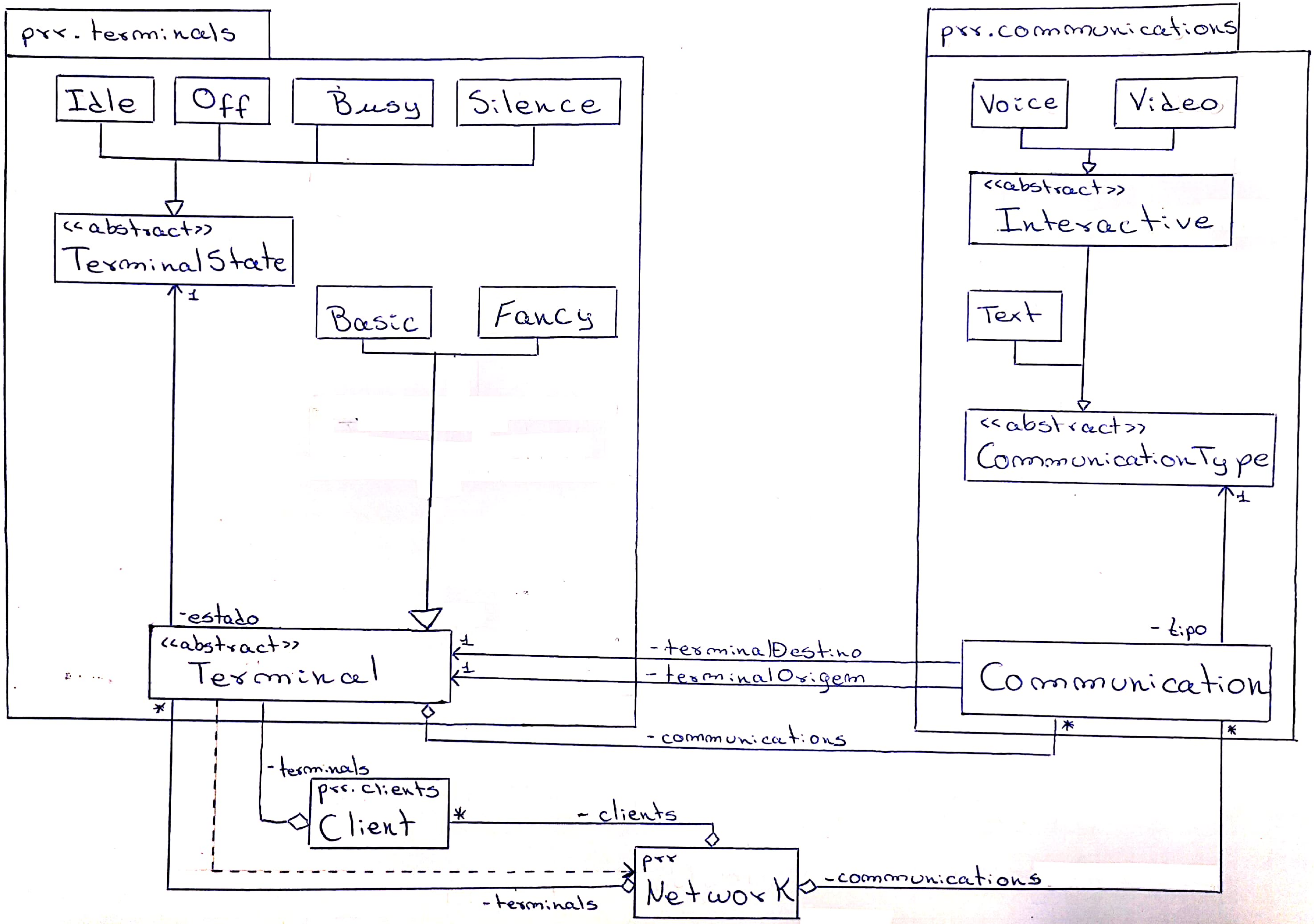
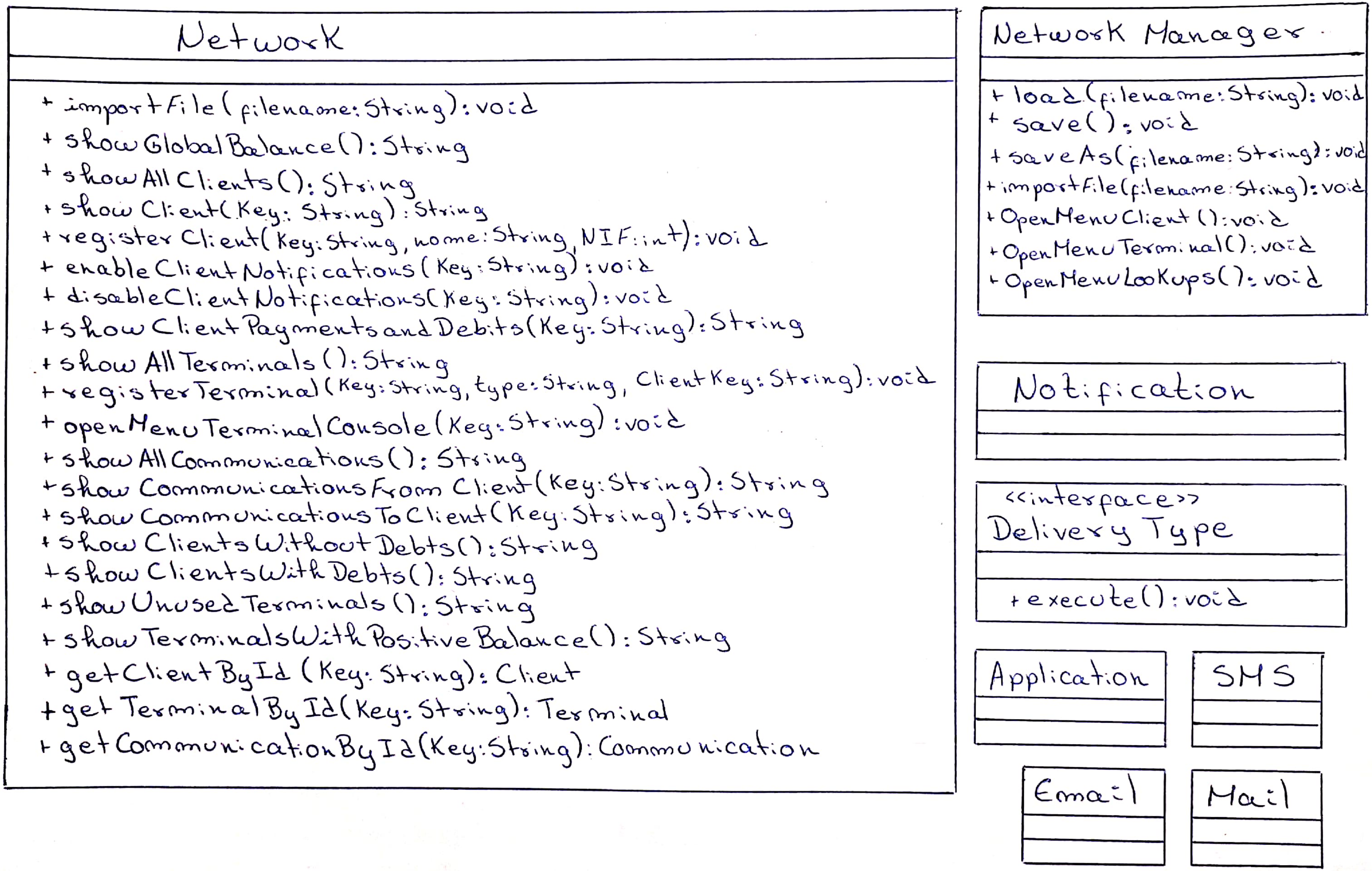


Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto. Ana Margarida Almeida

grupo 72


prr.terminals
Idle
Off
Busy
Silence
<<abstract>>
TerminalState
1
Basic
Fancy
-estado
<<abstract>>
Terminal
*
prr.communications
Voice
Video
<<abstract>>
Interactive
Text
<<abstract>>
CommunicationType
1
-tipo
1
-terminalDestino
1
-terminalOrigem
Communication
-communications
*
*
-terminals
prr.clients
Client
*
-clients
prr
NetworK
-communications
-terminals



Network

+ importFile(filename:String):void
+ showGlobalBalance():String
+ showAllClients():String
+ showClient(Key: String):String
+ registerClient(Key:String, nome:String, NIF:int):void
+ enableClientNotifications(Key:String):void
+ disableClientNotifications(Key:String):void
+ showClientPaymentsandDebits(Key:String):String
+ showAllTerminals():String
+ registerTerminal(Key:String, type:String, ClientKey:String):void
+ openMenuTerminalConsole(Key:String):void
+ showAllCommunications():String
+ showCommunicationsFromClient(Key:String):String
+ showCommunicationsToClient(Key:String):String
+ showClientsWithoutDebts():String
+ showClientsWithDebts():String
+ showUnusedTerminals():String
+ showTerminalsWithPositiveBalance():String
+ getClientById(Key:String):Client
+ getTerminalById(Key:String):Terminal
+ getCommunicationById(Key:String):Communication

Network Manager

+ load(filename:String):void
+ save():void
+ saveAs(filename:String):void
+ importFile(filename:String):void
+ OpenMenuClient():void
+ OpenMenuTerminal():void
+ OpenMenuLookups():void

Notification

<<interface>>
Delivery Type

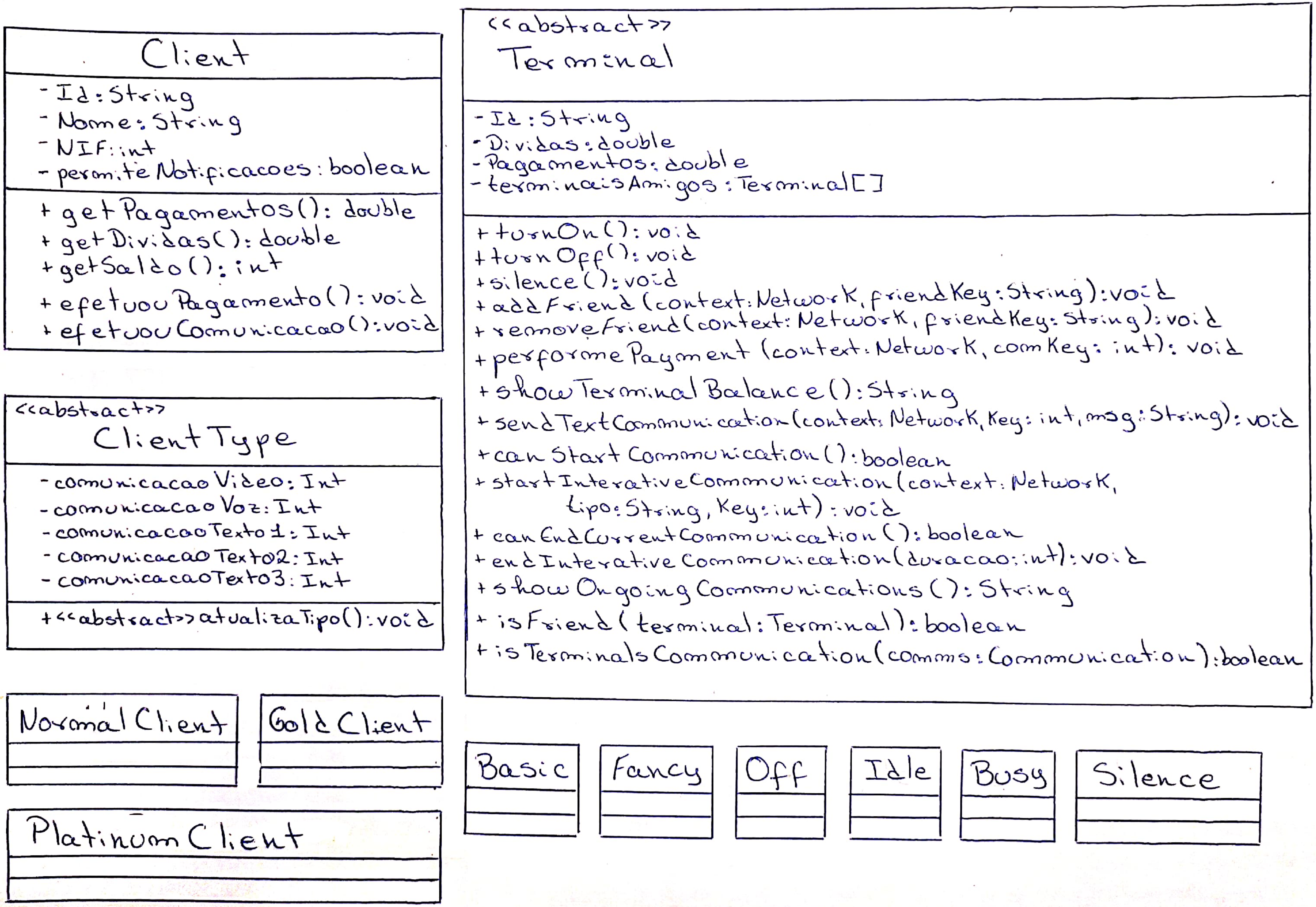
+ execute():void

Application

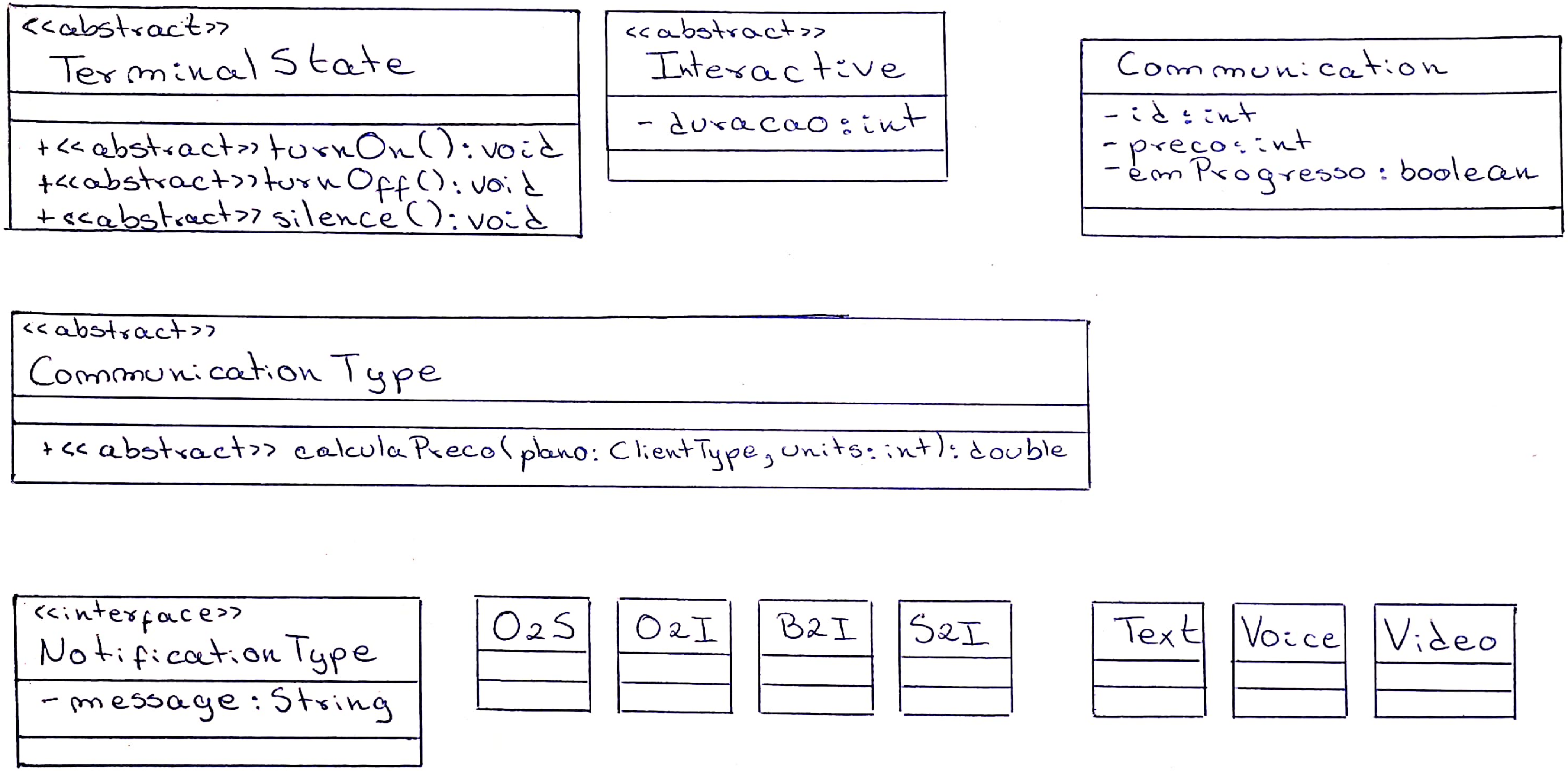
SMS

Email

Mail



```
Client
- Id: String
- Nome: String
- NIF::int
- permiteNotificacoes: boolean
+ getPagamentos(): double
+ getDividas(): double
+ getSaldo(): int
+ efetuouPagamento(): void
+ efetuouComunicacao():void

<<abstract>>
ClientType
- comunicacaoVideo: Int
- comunicacaoVoz: Int
- comunicacaoTexto1: Int
- comunicacaoTexto2: Int
- comunicacaoTexto3: Int
+<<abstract>>atualizaTipo():void

NormalClient

GoldClient

PlatinumClient

<<abstract>>
Terminal
- Id: String
- Dividas: double
- Pagamentos: double
- terminaisAmigos: Terminal[]
+ turnOn(): void
+ turnOff(): void
+ silence(): void
+ addFriend(context: Network, friendKey: String): void
+ removeFriend(context: Network, friendKey: String): void
+ performePayment(context: Network, comKey: int): void
+ showTerminalBalance(): String
+ sendTextCommunication(context: Network, Key: int, msg: String): void
+ canStartCommunication(): boolean
+ startInterativeCommunication(context: Network,
        tipo: String, Key: int): void
+ canEndCurrentCommunication(): boolean
+ endInterativeCommunication(duracao::int): void
+ showOngoingCommunications(): String
+ isFriend(terminal: Terminal): boolean
+ isTerminalsCommunication(comms: Communication): boolean

Basic

Fancy

Off

Idle

Busy

Silence
```


<<abstract>>
TerminalState
+ <<abstract>> turnOn(): void
+<<abstract>> turnOff(): void
+ <<abstract>> silence(): void
<<abstract>>
Interactive
- duracao : int
Communication
- id : int
- preco : int
- emProgresso : boolean
<<abstract>>
CommunicationType
+ <<abstract>> calculaPreco(plano: ClientType, units: int): double
<<interface>>
NotificationType
- message : String
O2S
O2I
B2I
S2I
Text
Voice
Video