app **RUData** CommonData - RU está em greve? : bool nome : string Pessoas no RU : int - matrícula : ulong creditsMobile: float - creditsCard : float + getPessoas() : int + getRUState() : bool + setPessoas(int) : void LocalData +setRUState(bool): void senha : int cardNumber : string? - monthDeadline : byte - yearDeadline : byte - safeCard : int Login Credits **LED** matriculaDigitada: ulong - creditsMobile : float · isAceso: bool - senhaDigitada : int creditsCard: float + syncData : bool + toggle(bool) : bool + lerMatricula(ulong) : bool + getCredito() : float + lerSenha(int) : bool + addCredito(): bool + autenticaLogin () : bool + ultimaConexao(): tempo **RefectoryAccess Synchronization** - codigo: char[8] · dia : byte - mes : byte - hora: byte + tentaEntrar(char[8], ulong) : bool #conseguiu enviar - minuto : byte pedido p/ Atlys? + conseguiuEntrar(bool) : bool #sequencia correta? + getLastAcess() : void + setLastAcess(byte, byte, byte, byte): void + sendData(bool) : void É uma lista. Classes como o Banco de dados CommonData LED

RIP.Leon

- nome : string - matrícula : ulong - creditsMobile : float - creditsCard : float + getNome() : string + getMatricula() : ulong + chargeMobile(??) : void + chargeCard(??) : void

MobileCom

- · userCode: ??
- userInput : byte
- machineCode : AccessGenerator
- + decodeUser(&ulong) : void
- + setUserInput() : byte
- + getUserInput() : byte
- + compareCode() : bool

CardCom

- userCode: ??
- + decodeUser(usercode?) : ulong

ISR

+ getRand(): byte

- isAceso: bool

rand: byte

+ toggle(bool) : bool

accessGenerator

- + accessMobile: bool
- + accessCard: bool
- + ISR_Wifi(void) : void + ISR_Serial(void) : void

Leon 4

- + data: Estrutura_dados
- + leituraCartao(): int
- + comandoSmartphone(): int
- + autentificacaoSmartphone(int): int
- + liberarCatraca(): void
- + registraSaida(): void

Catraca

- isOpen : bool (?)
- + liberaCatraca(bool) : void

banco de dados

CommonData

- nome : string
- matrícula : ulong
- creditsMobile : float - creditsCard : float

PersonalData

- passwordBU : int
- + PersonalData(): void + PersonalData(string, ulong, int, string) : bool
- + ~PersonalData(): bool
- + getNome() : string
- + getMatrícula() : ulong
- + getCreditsMobile() : float + getCreditsCard() : float
- + setNome(string) : bool
- + setMatrícula(ulong) : bool
- + addCreditsMobile(float, string?, byte, byte, int) : bool
- + addCreditsCard(float, string?, byte, byte, int) : bool
- + reduceOnePassMobile (): bool
- + reduceOnePassCard() : bool

TransactionData

- dia : byte - mes : byte
- ano: byte
- valor : float + next : TransactionData*
- + getDia(): byte + getMes() : byte
- + getAno() : byte + getValor(): float
- + getNext() : TransactionData*
- + setDia(byte) : void
- + setMes(byte) : void
- + setAno(byte) : void
- + setValor(float) : void + setNext(TransactionData*) : void

Node

- data : PersonalData
- firstTransaction : TransactionData*
- currentTransaction : int
- + nodeParent : Node*
- + nodeLeft : Node*
- + Node()
- + Node(string, ulong, int, string)
- + ~Node()
- + nodeParent() : Node*
- + nodeLeft() : Node*
- + nodeRight(): Node *
- + EFolha(): bool
- + addTransacao () : bool
- + listAllTransacoes (): void
- + clearAllTransacoes (): void

List

- raiz : Nodo*
- · auxApaga(Nodo*) : void
- auxInserir(Nodo *&, ulong) : void
- auxRemoverNodo(Nodo *&, ulong) : void
- auxProcurar(Nodo *&, ulong) : Node*
- auxListAll(Nodo *&, ulong) : void
- + List()
- + ~List()
- + inserirNodo(string, ulong, int, string) : void
- + removerNovo(string) : void
- + procurarNodo(ulong) : Node *
- + listAll (): void