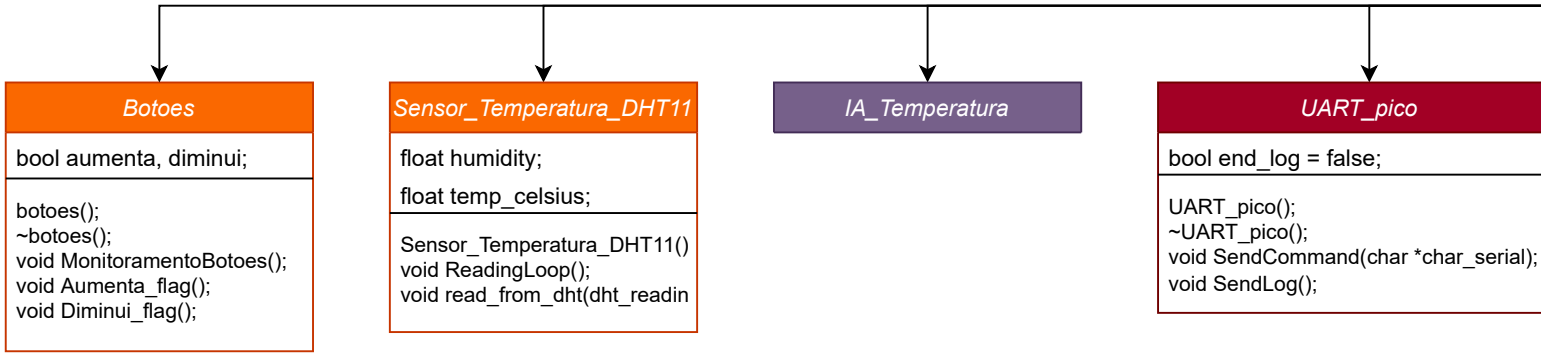
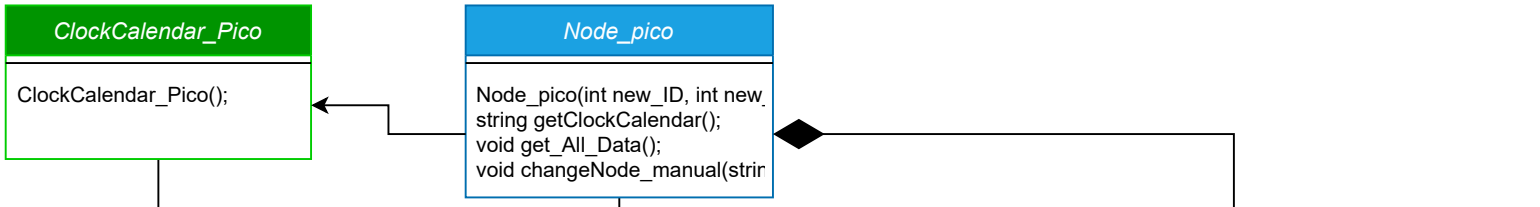
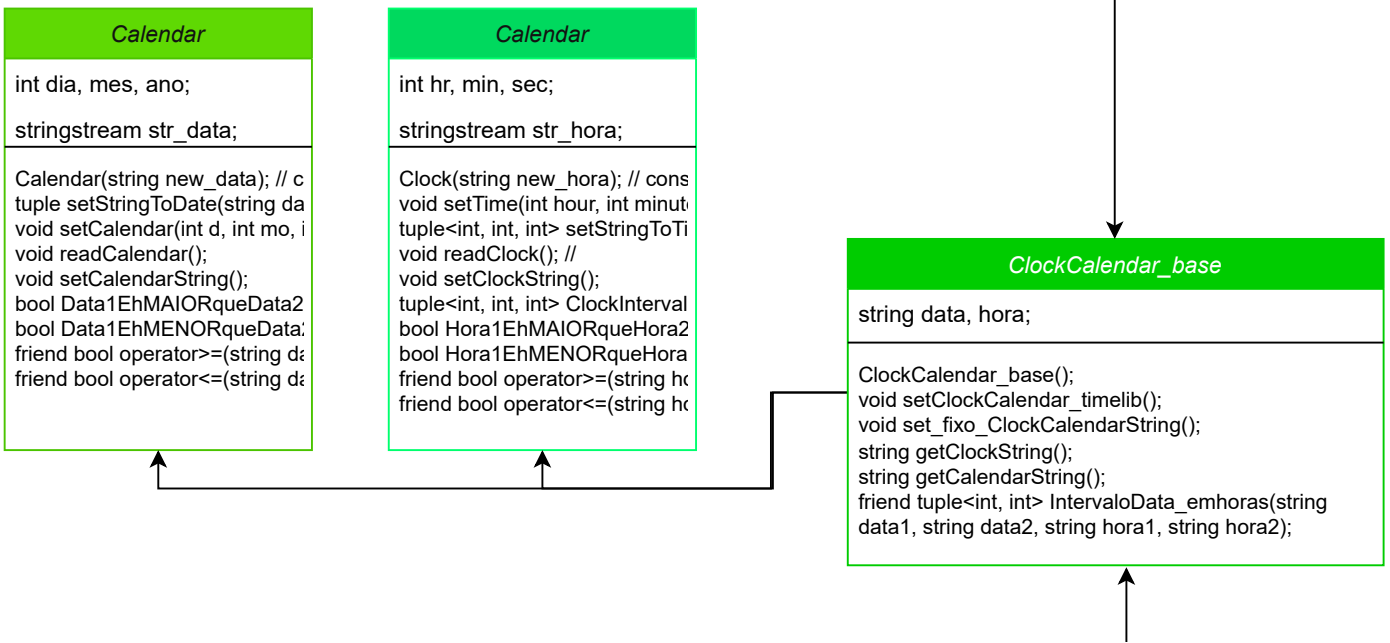
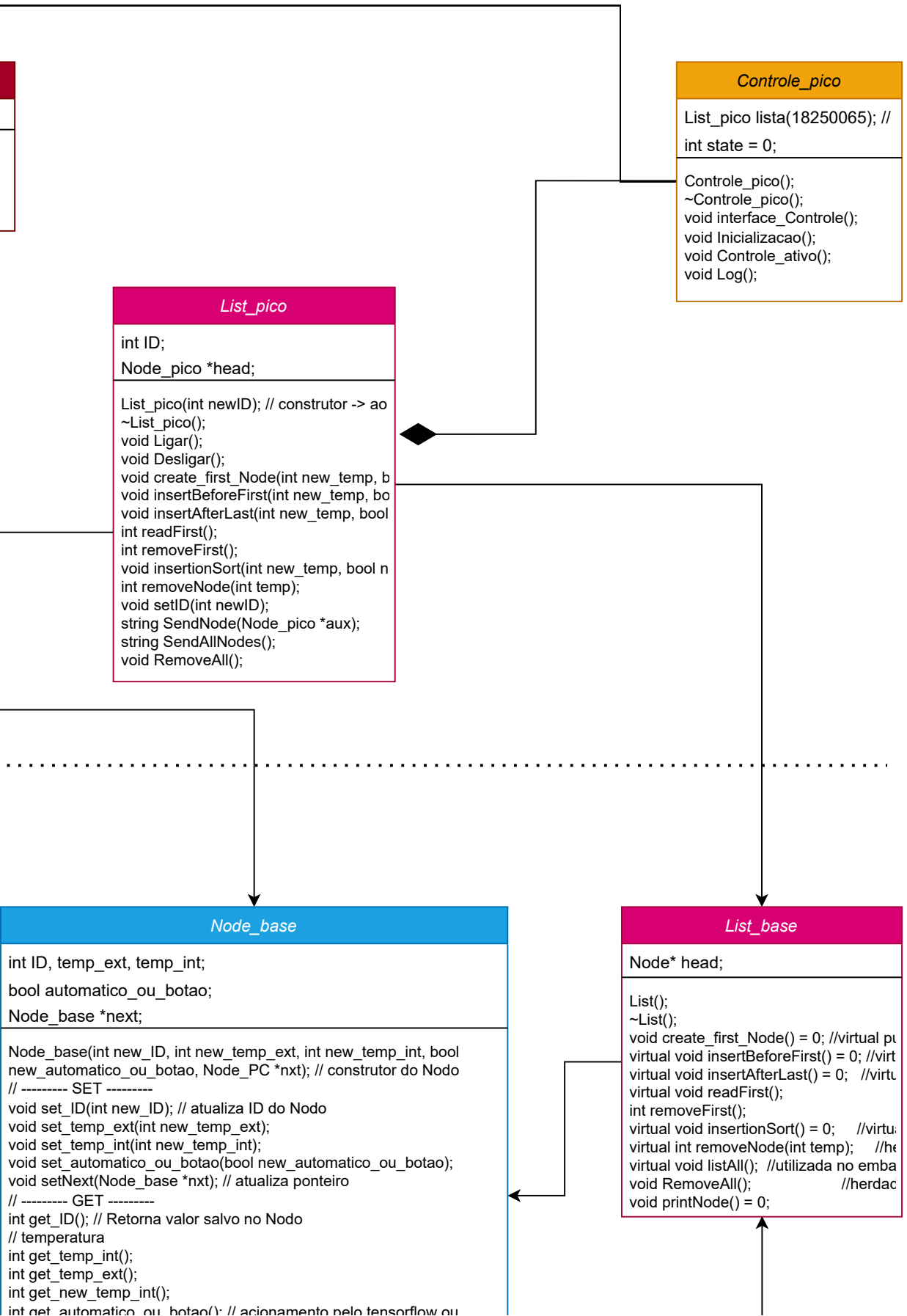
# Diagrama de classes Raspberry



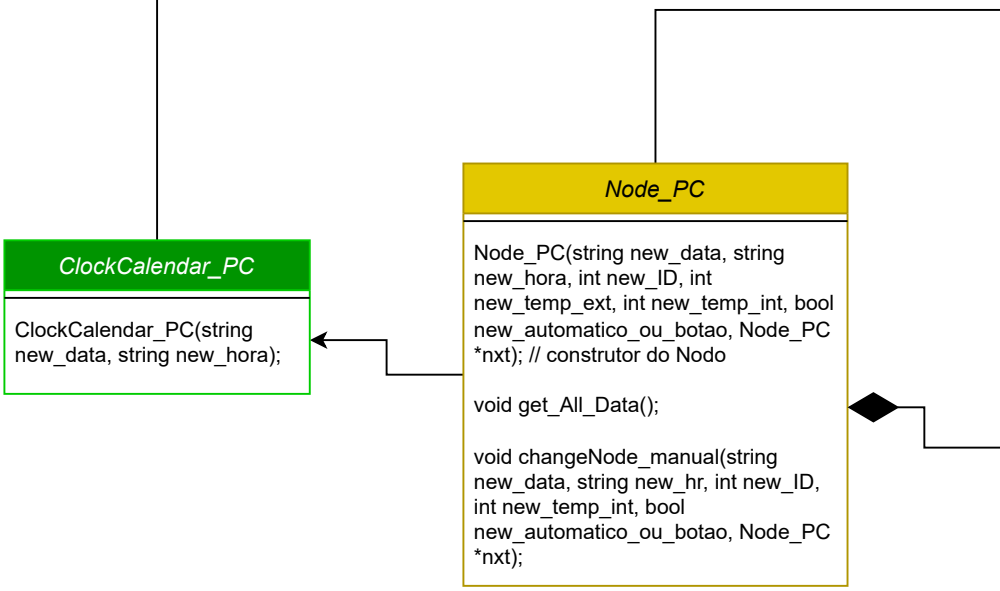



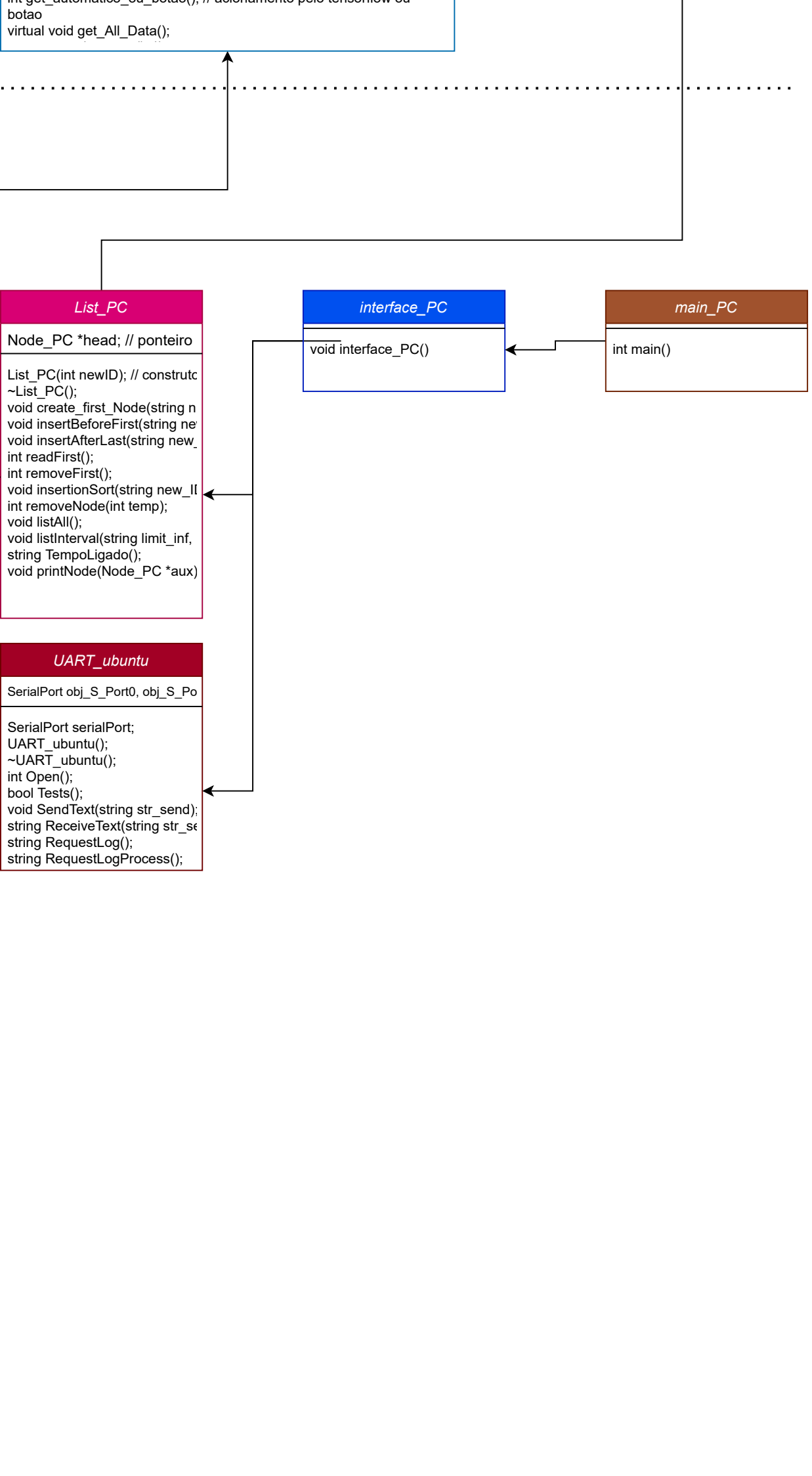

# Classes Base




Controle_pico
List_pico lista(18250065); //
int state = 0;
Controle_pico();
~Controle_pico();
void interface_Controle();
void Inicializacao();
void Controle_ativo();
void Log();
List_pico
int ID;
Node_pico *head;
List_pico(int newID); // construtor -> ao
~List_pico();
void Ligar();
void Desligar();
void create_first_Node(int new_temp, b
void insertBeforeFirst(int new_temp, bo
void insertAfterLast(int new_temp, bool
int readFirst();
int removeFirst();
void insertionSort(int new_temp, bool n
int removeNode(int temp);
void setID(int newID);
string SendNode(Node_pico *aux);
string SendAllNodes();
void RemoveAll();
Node_base
int ID, temp_ext, temp_int;
bool automatico_ou_botao;
Node_base *next;
Node_base(int new_ID, int new_temp_ext, int new_temp_int, bool
new_automatico_ou_botao, Node_PC *nxt); // construtor do Nodo
// --------- SET ---------
void set_ID(int new_ID); // atualiza ID do Nodo
void set_temp_ext(int new_temp_ext);
void set_temp_int(int new_temp_int);
void set_automatico_ou_botao(bool new_automatico_ou_botao);
void setNext(Node_base *nxt); // atualiza ponteiro
// --------- GET ---------
int get_ID(); // Retorna valor salvo no Nodo
// temperatura
int get_temp_int();
int get_temp_ext();
int get_new_temp_int();
List_base
Node* head;
List();
~List();
void create_first_Node() = 0; //virtual pu
virtual void insertBeforeFirst() = 0; //virt
virtual void insertAfterLast() = 0; //virtu
virtual void readFirst();
int removeFirst();
virtual void insertionSort() = 0; //virtu
virtual int removeNode(int temp); //he
virtual void listAll(); //utilizada no emba
void RemoveAll(); //herdac
void printNode() = 0;

# Diagrama de classes PC




botao
virtual void get_All_Data();
List_PC
Node_PC *head; // ponteiro
List_PC(int newID); // construtc
~List_PC();
void create_first_Node(string n
void insertBeforeFirst(string ne
void insertAfterLast(string new_
int readFirst();
int removeFirst();
void insertionSort(string new_II
int removeNode(int temp);
void listAll();
void listInterval(string limit_inf,
string TempoLigado();
void printNode(Node_PC *aux)
interface_PC
void interface_PC()
main_PC
int main()
UART_ubuntu
SerialPort obj_S_Port0, obj_S_Po
SerialPort serialPort;
UART_ubuntu();
~UART_ubuntu();
int Open();
bool Tests();
void SendText(string str_send);
string ReceiveText(string str_se
string RequestLog();
string RequestLogProcess();