**A picture containing shape

Description automatically generated**

Industry 4.0 Enabling Technologies

Prof. Tiago Manuel Fernández Caramés

Prof. Paula Fraga Lamas

SMART CONTRACT DOCUMENTATION

**BY**

**NWAFOR OGECHUKWU**

**MEHMET ANIL GÜRLEK**

**LUIZ FELIPE BAUZER DE OLIVEIRA**

**ENES SANCAK**

**SUSTAINABLE SHIP AND SHIPPING 4.0**

**HIGHER POLYTECHNIC UNIVERSITY COLLEGE**

**UNIVERSITY OF A CORUÑA**

**A close-up of a logo

Description automatically generated with medium confidence**

Documentation

The smart contract named **HVACContract.sol** deployed is a **Temperature Verification and Validation System** such that the temperature sensor data is collected and if the data recorded is within the acceptable range it accepts the data collected from the temperature sensor, but if the data collected from the temperature sensor is not within the acceptable range the system reverts and prints the statement “Temperature is not within the acceptable range”.

For our case study in validation with our Arduino project, the temperature limit should not be lower than 18 degrees Celsius nor exceed 25 degrees Celsius, therefore temperature not within this level stated reverts and the system records the information that the data collected is not within the acceptable range.

Key Points

* Contract is created using Solidity version >=0.5.0 and 0.8.0
* The keccak256 hash of abi.encodePacked(\_str) is used to generate a pseudo-random hexadecimal, typecast it as a uint, and finally store the result in a uint called rand.
* Function Value set as public view takes a string called \_Parameter which will be inputted, in our case the parameter to be considered is the temperature, and a uint \_degree which is our sensor data that will retrieved from our sensor.
* Function validationTemp set to private view takes a uint \_valueId and checks that the temperature data received by the system does not exceed 25 degrees C.
* Uint mapping named degree is created to store the value of the temperature data.
* Modifier is used inorder to apply our conditions with a require that checks if the temperature is within acceptable range otherwise it reverts if the transaction if the check fails.