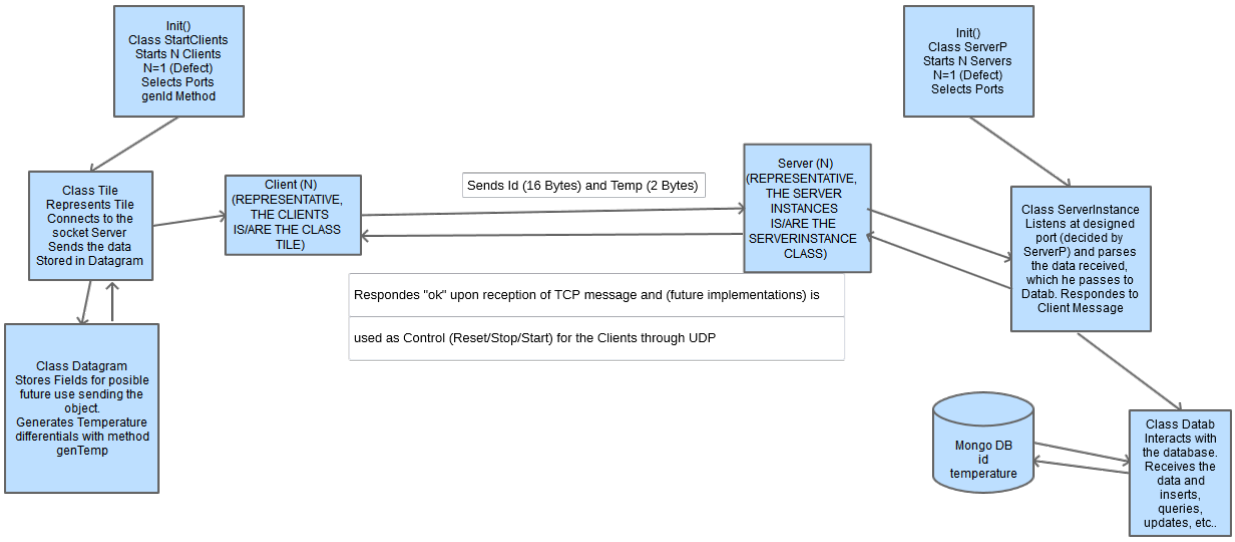
Decisiones Test Plactherm.

Arquitecture:



Protocol:

In this case we decided to choose TCP because of the definition of the test, but, under my perspective, the implementation in UDP (suposing triggered events, and low data datagrams) would be more efficient for the following reasons:

1.- The advantages that TCP provides (Flow of bytes, congestion and flow control, etc..) dont give enough advantages over UDP in this practical case, since we have very small datagrams, and a local network, so, unless we were streaming datagrams at a huge rate, (and even in that case, UDP has a smaller header by 12 bytes, therefore each packet will be considerably smaller).

2.- Because of the small header, if we consider a panel system with lots of small pannels that send information simultániously and with a certain frequency, there would be less congestion in the system.

3.- Supposing "triggered" (executed by a reaction to a sensor, or state) it is not necesary to have an active TCP connection like we do.

4.- The biggest advantage that TCP can provide in this scenario over udp is the use of the URG flag (urgent pointer) since it will pass the data to the application as soon as posible but, since we work with sistems in which time is not a critical factor (as long as it is smaller than a few ms) we can guarantee that UDP will work fine.

In any case, the implementation in UDP of this service or similar ones is a simple task.

Aditional Implementations:

- Server in Amazon EC2 to respond to client petitions outside local area.
- Posibility of Instantiation of N independant clients (and N server socket instances).
- Random Generator of temperature differentials to express change in the temperature.

Implementations of interest (Did not have time to do them):

- MD5 encription of the data to asure security and that no one can have access to the presure sensors of the floor (for example to detect where everyone in the house is).
- UDP Control System Server->Client. Sends to the client a command indicating if the tile should sleep, reset, or turn itself of (as an example, could deactivate sensors, switch the tile off in case of a flood so the hardware doesn´t break, etc..).
- Central control system, with the local server sending the data from the database every X time to generate statistics, or see if there is a malfunction from a remote location without having to connect to the local server.