***Tabla de decisiones de diseño***

| **Fecha** | **Decisión** | **Ventaja** | **Desventaja** | **Alternativa** |
| --- | --- | --- | --- | --- |
| 13/10/2018 | Se decidió usar SINGLE TABLE para solucionar el problema de la herencia de los dispositivos inteligentes y estándar. | Al unificar todo en una tabla me ahorro todo el tema de los joins de varias tablas para relacionarlas, y además es mas performante por lo antes mencionado. |  |  |
| 13/10/2018 | Se decidió usar la estrategia SINGLE TABLE para persistir los usuarios (clientes y administradores). | Al persistir todos los usuarios en una sola tabla, si se necesita recuperarlos, solo basta con llamar a llamar a una sola tabla, y los identifico por el TIPO y su ID. | En la base se va a ver campos en NULL, por ejemplo en administrador, al no tener esos atributos. | Crear una tabla distinta por cada tipo de usuario. |
| 13/10/2018 | Se creó las clases DAO de las entidades persistidas. | La comunicación con la base de datos se delega a una nueva clase. En ésta están los métodos de querys y updates. Se separa lo que tenga que ver con nuestro modelo. |  | La comunicación con la base se haga dentro de las clases persistidas, en unos métodos separados. |
| 13/10/2018 | Se utilizó la estrategia SINGLE TABLE para persistir sensores, actuadores. | Al igual que en lo demás en una sola tabla se visualiza las clases heredadas. |  |  |
| 13/10/2018 | Se creó la clase Medicion y su persistencia en la base de datos. | Permite persistir las mediciones realizadas por los sensores. |  | Agregar campos a las tablas existentes en la base. |