

DISEÑO DE SISTEMAS

Trabajo Práctico Anual

“Sistema de Gestión Energética”

Grupo: 8

Integrantes:

- Alejo Scotti - alejoscott@gmail.com - 1528142
- Ezequiel González - egonzalez01@gmail.com - 1256970
- Juan Pablo Ferreira - juanpabloferreira88@gmail.com - 1275902
- Nicolas Hovassapian - Nichova.95@gmail.com - 1530318

Fecha de entrega: 22/05/2018

Profesor: Martín Agüero

Ayudante a cargo: Alejandro Ezequiel Leoz - Nicolas Contreras

Repositorio: <https://github.com/JuanPabloFerreira/TP-DDS2018-Grupo8>

Branch: master

Commit ID: ba3641e0fa9df0e5cdb762444227c56f28ae08ac

Registro de cambios

Fecha	Modificaciones
12/05/2018	Patron State para el estado de los dispositivos inteligentes
12/05/2018	Patron decorador para adaptadores
13/05/2018	Patron Bridge para los Actuadores
16-05-2018	Diagrama de Clases preliminar con los patrones seleccionados.

Tabla de Requerimientos no funcionales TP - Entrega 1

Fueron identificados los siguientes requerimientos no funcionales. Se tuvieron en cuenta aquellos que afectarían a la arquitectura del sistema.

Permitir la instalación de sensores.

Diagrama de Clases TP - Entrega 1



Diagrama de
clases.svg

Tabla de decisiones de diseño:

Fecha	Decisión	Ventaja	Desventaja	Alternativa
12/05/2018	La idea sería usar un Estate para los Estados de los Dispositivos Inteligentes .	Permitir que un objeto altere su comportamiento cuando su estado interno cambia. Permite modelar las transiciones entre estados.		
12/05/2018	Para los Adaptadores de los Dispositivos Estadar usaríamos un Decorador .	Agregar dinámicamente responsabilidades (funcionalidad) extra a un objeto. Es una forma flexible que sirve de alternativa a subclassing para extender funcionalidad. Mas flexibilidad que la	Un decorador y su componente no son idénticos.	Estrategias

		herencia estática.		
13/05/2018	Para el Actuador se usaría el patrón Bridge via la interface Implementador para no depender de implementación que tiene cada electrodomestico según su fabricante como indica el enunciado.	Desacoplar una abstracción de su implementación, de modo que ambas puedan variar de forma independiente.		Composite
13/05/2018	Se agregó el patrón Observer , para avisar cada vez que un Sensor realiza una medición de la magnitud que corresponda	Definir dependencias one-to-many entre objetos, de forma tal que cuando un objeto cambia su estado todos los objetos dependientes son notificados y actualizados inmediatamente		
16/05/2018	Una Regla tendría un listado de Condiciones y Acciones, además conoce a un dispositivo, sobre el cual comprobaría el cumplimiento de las Condiciones, en caso de cumplirse todas las condiciones, ejecutaría las acciones. A su vez el cliente definiría las reglas a aplicar sobre sus dispositivos.			