Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Ingeniería en Sistemas de Información

AÑO 2018

SGE - Sistema de Gestión Energética

ENTREGA 2

Materia:

Diseño de Sistemas

Docente:

Pablo Sabatino

Ayudante:

César Cappetto

Fecha prevista de entrega: 22/8/2018

Nombre	Legajo
Angel Dario	144.506-6
Méndez Angélica	144.095-0
Nicora Facundo	163.093-3
Otero Alejandro	146.820-0
Peralta Liliana	143.242-4

ÍNDICE

OBJETIVOS DE LA ENTREGA 2

Esta entrega tiene por objetivo que el alumno sea capaz de:

● Evaluar y reutilizar software de terceros, probando su validez y conveniencia para el caso.

● Diseñar una solución flexible que permita escalar el sistema sin requerir nuevos cambios de diseño.

Implementación:

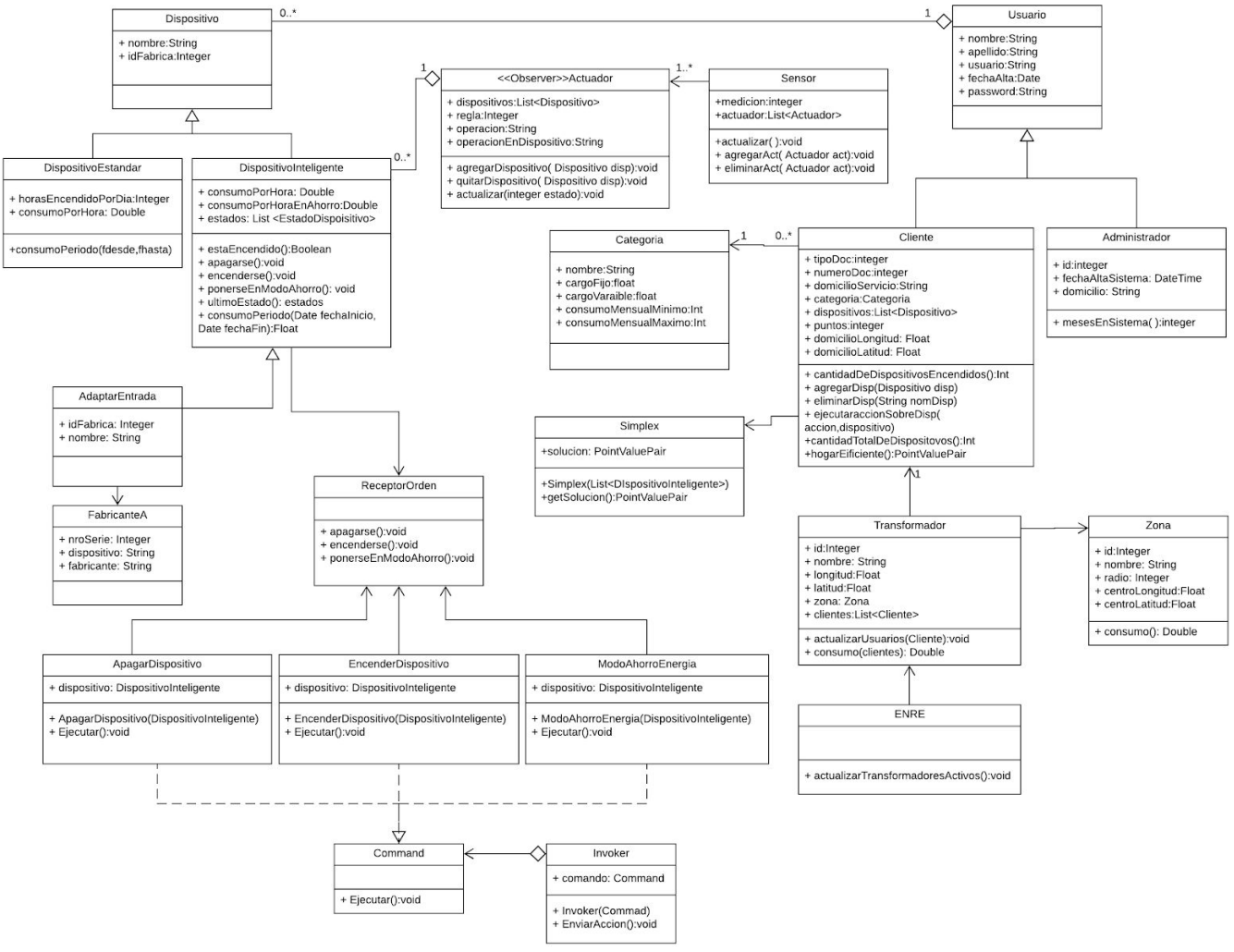
Modelado de la mejor combinación de dispositivos

Modelado de transformadores y geoposicionamientos

ENTREGABLES

DIAGRAMA DE CLASES ACTUALIZADO

El siguiente diagrama de Clases, contiene la representación del modelo actualizado según los requerimientos de la presente entrega. Entendemos que cada atributo de las clases tiene sus setters y getters, pero decidimos no colocarlos en el diagrama y mostrar solo los métodos esenciales para la entrega.



Para adaptar los nuevos requerimientos de esta entrega, tuvimos que agregar algunos atributos y métodos a las clases existentes y creamos otros nuevos.

Clase Cliente:

Agregamos los atributos de domicilioLongitud y domicilioLatitud, para que se pueda referenciar al cliente con el punto más cercano de un transformador. Agregamos un método para que consulte por su hogar eficiente.

Clase Transformador, Zona y Enre:

Son clases agregadas al modelo para adaptar el requerimiento.

Transformador: contiene una lista de clientes, el cual están en un punto cercano al mismo.

Zona, cada transformador pertenece a una Zona, el cual contiene los datos de radio, centroLatitud y centroLongitud.

Enre, es el que contendrá la información de los transformadores activos.

Clase Simplex

Creamos una clase Simplex, la cual recibiendo una lista de dispositivos inteligentes, devuelve la solución, es decir, el uso que se recomienda para cada uno de ellos.

IMPLEMENTACIÓN

DESARROLLO DE LA IMPLEMENTACIÓN

Lenguaje de Programación **:** Java

Entorno de trabajo **:** Eclipse

URL Repositorio **:** https://github.com/AleOtero93/diseno/tree/entrega2

Branch **:** entrega2

TABLA DE DECISIONES DE DISEÑO

Fecha	Decisión	Ventaja	Desventaja	Alternativa
08/08/2018	Utilización del Patrón Observer para los actuadores y sensores	Resuelve la dependencia que existe entre los dispositivos, actuadores y sensores, de tal forma que cuando ocurra una alerta (reglas) los afectados sean avisados.		Aplicar otro patrón o ninguno.
08/08/2018	Utilización de los patrones mencionados en el modelado de clases Patrón Command en las acciones de los dispositivos. Patrón State en los estados de los dispositivos y no Strategy	Al aplicar patrones de diseño hace que en la implementación se tenga un desarrollo más legible, con mayor cohesión y bajo nivel de acoplamiento entre las entidades el cual resulta más sencillo de mantener. Encapsula un mensaje como un objeto.Permite solicitar una operación a un objeto sin conocer el contenido ni el receptor real de la misma. Emplear una clase que represente el estado del dispositivo brindará mayor extensibilidad ya que, en caso de ser necesario en el futuro, se pueden modelar distintos tipos de estado. Se aplica el patrón mencionado en lugar del Strategy ya que la transitividad de los estados es lineal, a diferencia del Strategy que permite que sean intercambiables, y el comportamiento del objeto depende del estado.	Desarrollo de más clases y mayor prueba.	No utilizar patrones de diseño.
06/06/2018	Comunicación cliente/sistema a nivel componentes	Mostrar la interacción de los usuarios y dispositivos con el sistema en forma general, observando las partes principales de comunicación.	No se conoce el detalle de cada componente.	Aplicar otro diagrama de comunicación.

06/06/2018	No modelado de la CONVERSIÓN ESTÁNDAR-INTELIGENTE	Evitar la incorporación de un módulo compleja que realice la conversión de dispositivos ya que el mismo será algo externo al sistema y no una funcionalidad del mismo.		Aplicar patrón creacional.
19/06/2018	Inclusión de motor Javascript para uso de función eval	Posibilidad de evaluar un condicional modificable según la acción que se reciba.	Utilización de un elemento externo a Java y Eclipse.	Utilizar algún componente de Java o un switch con cada acción posible.

HERRAMIENTAS ADICIONALES

Para esta Etapa trabajamos con las siguientes herramientas :

- **Slack : DdS - SGE - dds-sge.slack.com**

Utilizamos esta herramienta como servicio de mensajería, para identificar las actividades realizadas como: commits en github, actualizaciones de tareas en el trello.

- **Trello**

Antes de iniciar con el desarrollo de la entrega dividimos las tareas según los requerimientos, creandolos en 3 estados: Lista de tareas, En proceso y Hecho.

- **Lucidchart**

Optamos por utilizar esta herramienta online para confeccionar los diagramas requeridos, compartidos para visualizar el estado y realizar modificaciones sobre los mismos, evitando inconvenientes de versiones en caso de utilizar herramientas de escritorio.