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

AÑO 2018

SGE - Sistema de Gestión Energética

ENTREGA 1

Materia: Diseño de Sistemas

Docente:

Pablo Sabatino

Ayudante:

César Cappetto

Fecha prevista de entrega: 06/06/2018

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

REQUERIMIENTOS DE LA ENTREGA 1

Concepción y Comunicación del Diseño

● Diagrama/s de clases actualizado (en el caso de que se utilice algún patrón de diseño, marcarlo en el diagrama)

● Proponer y documentar de qué manera puede darse la comunicación entre el sistema y los dispositivos (a alto nivel, sin detalles de implementación). Evaluar su impacto sobre el modelo de objetos.

● Cualquier otro diagrama o explicación que consideren necesaria Implementación

Implementar todos los requerimientos y las pruebas que consideren necesarias, abstrayéndose (para esta entrega) de la implementación de las comunicaciones entre el sistema y los dispositivos.

Será obligatorio que mínimamente se realice:

● Modelado de los dispositivos inteligentes

● Modelado de los dispositivos estándares

● Modelado de conversión estándar-inteligente

● Modelado de los actuadores

● Modelado de los sensores

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.

El modelado de los dispositivos **estándar/inteligente** está dado por una simple herencia, donde cada subclase define las acciones que puede realizar cada dispositivo. La adaptación de un dispositivo estándar a inteligente no se ve reflejada ya que consideramos que dicho proceso no será parte del sistema, en el momento de la conversión se desechara el existente y se creará un nuevo dispositivo. En el caso de los **actuadores y sensores** utilizamos el patrón “Observer”, ya que ante las mediciones enviadas por el sensor, el actuador ejecutará alguna regla según la acción que quiera realizar. En el caso de los fabricantes consideramos el patrón “Adapter” para convertir el archivo de origen proporcionado por las diferentes marcas tal que sea entendible por nuestro sistema.

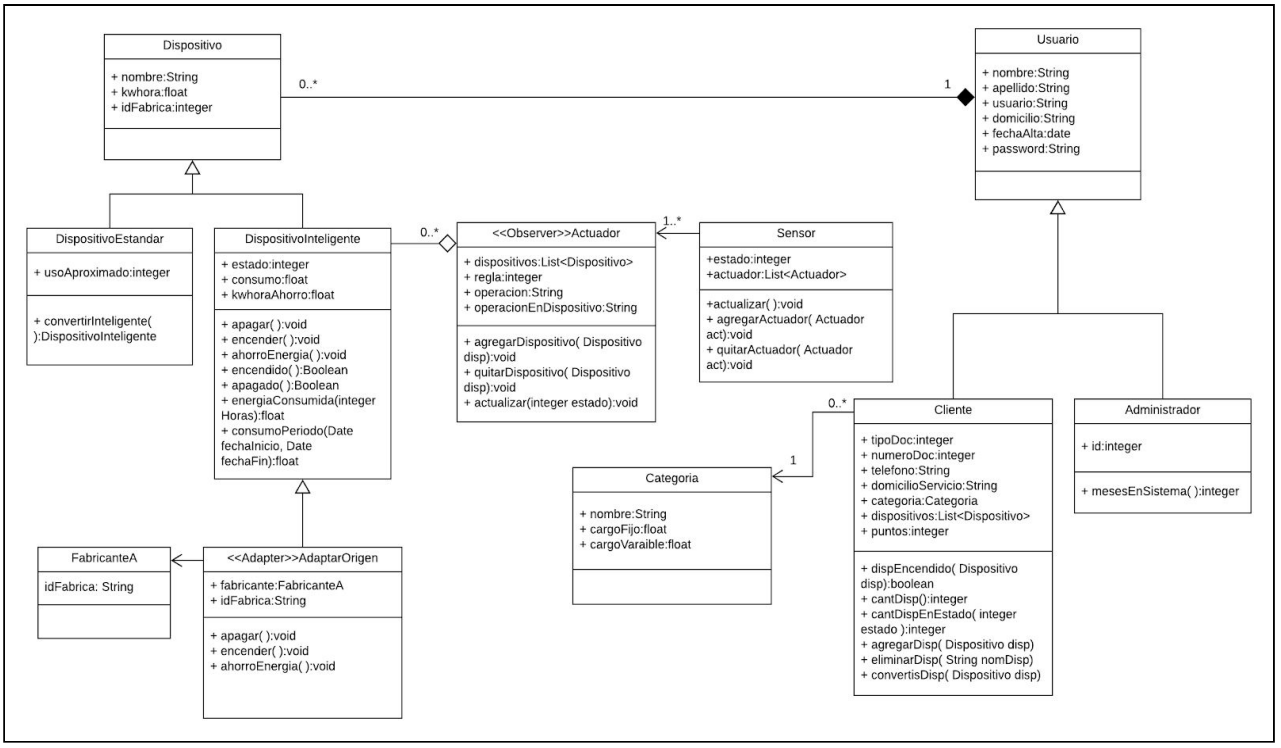
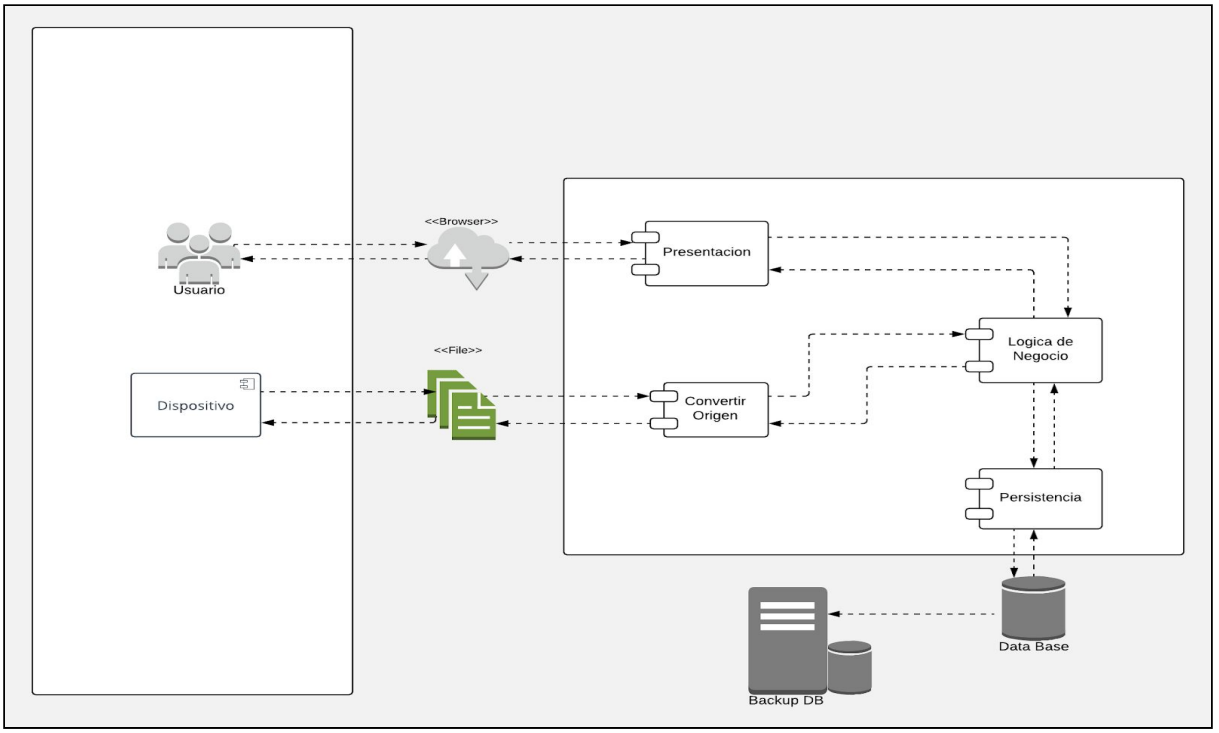


DIAGRAMA DE COMPONENTES ACTUALIZADO

Consideramos que la comunicación entre el sistema y los dispositivos a alto nivel, se dará de la siguiente manera:



Tenemos a los dispositivos inteligentes que se conectan al sistema a través de diferentes formatos de archivo (http, json, txt, texto comprimido,etc.) definido por el fabricante de origen.

Tenemos a los usuarios quienes se conectan a través de un browser de navegación, alli se ejecutará un request/response entre las diferentes capas del sistema para retornar algún resultado, consideramos las siguientes capas:

- Convertir origen: recibe información de los dispositivos y lo transforma tal que sea entendible por el sistema.
- Presentación: interfaz visualizada por el usuario para ingresar y obtener información.
- Lógica de negocio: modelado de negocio, instrucciones para atender las peticiones ingresadas por el usuario y captura de información proveniente de los dispositivos.
- Persistencia: lógica que utilizará el sistema para la conexión a una base de datos con el fin de almacenar y recuperar datos.

IMPLEMENTACIÓN

DESARROLLO DE LA IMPLEMENTACIÓN

Lenguaje de Programación : Java

Entorno de trabajo : Eclipse

URL Repositorio : https://github.com/AleOtero93/diseno/tree/entrega1

Branch : entrega1

MODELADO DE DISPOSITIVOS INTELIGENTES

Modelado en la clase DispositivoInteligente.java

MODELADO DE LOS DISPOSITIVOS ESTÁNDARES

Modelado en la clase DispositivoEstandar.java

MODELADO DE CONVERSIÓN ESTÁNDAR-INTELIGENTE

No se encuentra modelado ya que consideramos que dicho proceso no será parte del sistema, es decir, no existirá una implementación de dicha conversión, en el momento de realizar este proceso se dejará de usar el estándar sacándolo de la lista y creando un nuevo dispositivo el cual será inteligente.

MODELADO DE LOS ACTUADORES

Modelado en la clase Actuador.java

MODELADO DE LOS SENSORES

Modelado en la clase Sensor.java

Lo modelamos en relación con la clase Actuadores, el cual contiene una lista de actuadores .

La clase Sensor, es el encargado de quitar o agregar actuadores y contiene el método de actualizar; es el elemento que cumple la función de observadores dentro del Patrón “Observer” (clase Actuadores).

TABLA DE DECISIONES DE DISEÑO

Fecha	Decisión	Ventaja	Desventaja	Alternativa
06/06/2018	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.
06/06/2018	Utilización del Patrón Adapter para los archivos origen de fábrica	Utilizamos la clase adaptadora como versatilidad a la hora de recibir información de cualquier fabricante y que nuestro sistema lo pueda interpretar		Simple herencia, pero como contamos el conocimiento de patrones utilizamos lo que ya está creado
06/06/2018	Utilización de los patrones mencionados en el modelado de clases	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.	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 de diseño Decorator, sumándole al Dispositivo Estándar las funcionalidades del Inteligente

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.