DECISIONES DE DISEÑO TP SISTEMA DE GESTIÓN ENERGÉTICA

ENTREGA 1

Contenido

[Diagramas 3](#_Toc512192019)

[Diagrama de Arquitectura 3](#_Toc512192020)

[Diagrama de Casos de Uso 4](#_Toc512192021)

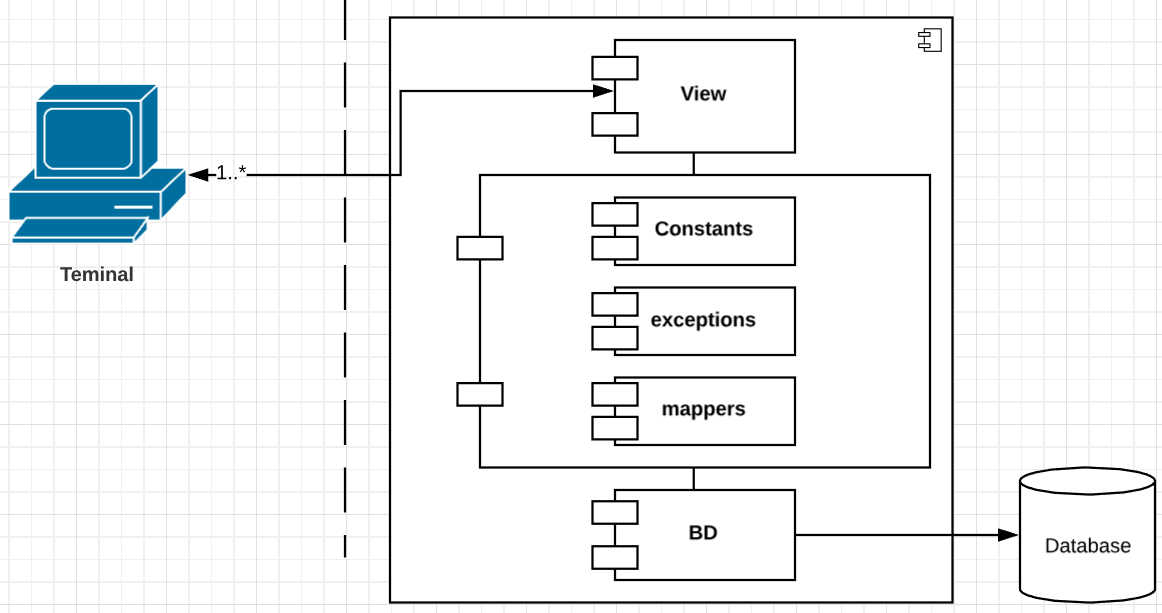
[Diagrama de clases 5](#_Toc512192022)

[Requerimientos no funcionales 7](#_Toc512192023)

[Decisiones de Diseño 8](#_Toc512192024)

# Diagramas

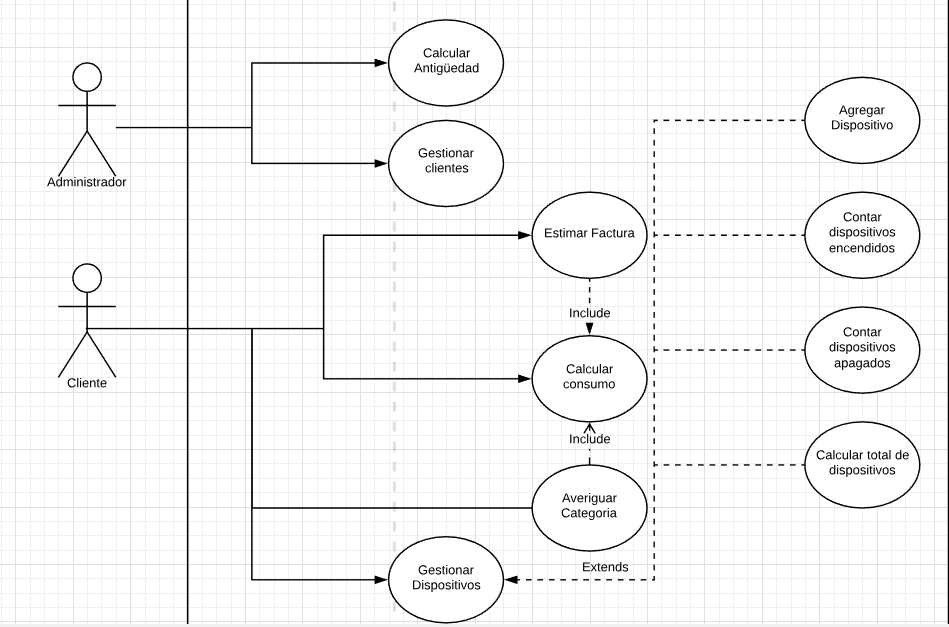
## Diagrama de Arquitectura



En la arquitectura se detalla que los usuarios podrán acceder al sistema por medio de un terminal. Se accederá a una interfaz de presentación “View”. El sistema tiene una capa de con la lógica de negocio que incluye Constants, exceptions y mappers. Se toma en cuenta que en entregas futuras se buscará persistir los datos transaccionales en una base de datos.

* El usuario del sistema se conectará con el sistema por medio de un dispositivo (celular, ordenador, etc), a una capa de presentación (GUI) donde verá los elementos necesarios para él. Detrás de la capa gráfica, estará el detalle de las reglas de negocio, donde se realizará la operativa del sistema. Luego se buscará que los datos del sistema se guarden en el tiempo, en una base de datos, que puede ser o no relacional.
* Se planea que la interacción entre los dispositivos y el sistema sea mediante un protocolo web.

## Diagrama de Casos de Uso



El Caso de Uso “Gestionar Clientes”, no se implementa en esta entrega, se deja el caso de uso para que se entienda el rol del administrador.

## Diagrama de clases

En el diagrama se omitieron agregar los métodos de getters y setters de los atributos de cada clase.

Se decidió crear una clase abstracta Usuario, de la cual heredan Cliente y Administrador.

Se decide hacer la clase Dispositivo como clase abstracta

Se cambia el tipo de Estado de los dispositivos por String

# Requerimientos no funcionales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Código | Requerimiento | Descripción | Atributo de calidad afectado | Métricas |
| R1 | El software es tolerante a fallos. | El software es capaz de recuperarse ante fallas. Ofrece las funcionalidades con la capacidad de brindar confiabilidad al usuario en caso de fallas y excepciones. | Confiabilidad | Aceptable: Sistema captura el 100% de las excepciones mostrando mensaje amigable  No Aceptable: Sistema no captura alguna excepción |
| R2 | El software realiza todos los requerimientos funcionales | El software tiene las características y funcionalidades que se especifican en los requerimientos funcionales de la entrega 0 y con un 100% de testings unitarios en verde (sin fallas). | Funcionalidad | Aceptable: Sistema cumple con el 100% de lo solicitado en el Doc de la entrega 0 y si hay un porcentaje de incumplimiento debe estar justificado y corroborado por el ayudante  No aceptable: AL sistema le falta una funcionalidad que solicita el Doc de la entrega 0 |
| R3 | El software es mantenible. | El software puede ser ampliado y adaptado con facilidad. Además sus módulos presentan bajo acoplamiento, lo que permite que sea sencilla su modificación sin afectar a los demás y pueden ser probados. | Flexibilidad, Mantenibilidad |  |
| R4 | El sistema es seguro. | Los permisos de acceso al sistema se manejarán con un usuario y contraseña. Los administradores sólo podrán cambiar información de configuración.  Las claves deben tener un mínimo de 5 caracteres que incluyan letras y números. | Seguridad | Aceptable: El sistema debe resistir el 80% de los ataques de diccionarios(brute force) , más adelante se realizara un hardening al ingreso de usuarios. |
| R5 | El sistema es escalable | Se espera que el sistema tenga el alcance de clientes residenciales, teniendo en cuenta que existen otros tipos. | Escalabilidad | Aceptable: el sistema soporta la cantidad de clientes residenciales de la ciudad, pero se espera y debería soportar clientes comerciales y fabriles |

En la tabla se describieron los requerimientos que consideramos que aplican para la entrega 0.

No se tienen en cuenta los requerimientos relacionados con interfaz gráfica, usabilidad, eficiencia, ya que no están en el alcance de la entrega.

# Decisiones de Diseño

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Decisión | Ventaja | Desventaja | Alternativa |
| 29/5/2018 | Se decide utilizar el patrón adapter para el manejo de dispositivos estándares que se convierten en inteligentes | Permite adaptar fácilmente los dispositivos | Se genera un acoplamiento entre el dispositivo adaptado y el dispositivo inteligente | Hacerlo sin aplicar el patrón |
| 29/5/2018 | Se decide utilizar el patrón Observer para el manejo de los sensores de los dispositivos | Se utiliza un patrón que resuelve el problema |  | Hacerlo sin aplicar el patrón |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |