|  |  |
| --- | --- |
| logoutnba_2 | **Universidad Tecnológica Nacional**  **Facultad Regional Buenos Aires**  **Ingeniería en Sistemas de Información** |

**AÑO 2018**

***SGE - Sistema de Gestión Energética***

***ENTREGA 0***

**Materia: Diseño de Sistemas**

**Docente:**

**Pablo Sabatino**

**Fecha prevista de entrega: 24/04/2018**

|  |  |
| --- | --- |
| **Nombre** | **Legajo** |
| Angel Dario | 144.506-6 |
| Mendez Angelica | 144.095-0 |
| Nicora Facundo | 163.093-3 |
| Otero Alejandro | 146.820-0 |
| Peralta Liliana | 143.242-4 |

**INDICE**

[**REQUERIMIENTOS DE LA ENTREGA 0**](#_3447tzggueer) **3**

[**ENTREGABLES**](#_fqrgmrlwmwh) **4**

[**IMPLEMENTACIÓN**](#_hdpydqy4f3nn) **8**

[**HERRAMIENTAS ADICIONALES**](#_re16oh1kftoy) **9**

[**TABLAS DE DECISIONES DE DISEÑO**](#_fsq46b7810sz) **10**

# **REQUERIMIENTOS DE LA ENTREGA 0**

● Capturar el comportamiento esencial del sistema a través de casos de uso, de forma tal de facilitar la comprensión, validar arquitectura y verificarlo a medida que evoluciona a lo largo de su desarrollo.

● Modelar el dominio con el paradigma orientado a objetos con el propósito de abstraer las entidades y sus características más relevantes para el sistema.

● Investigar las tecnologías recomendadas y decidir por la que mejor se adapte a los requerimientos del sistema (contemplando experticia de los integrantes del equipo).

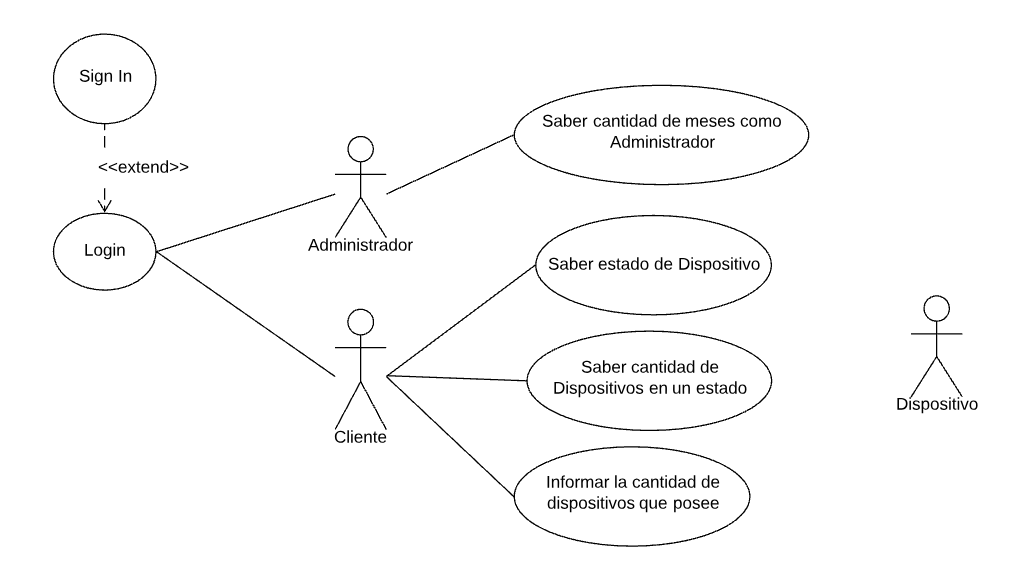
● Establecer un ambiente de desarrollo donde compilar y ejecutar una versión inicial del sistema.

● Establecer un repositorio para el proyecto donde se concentrará toda la producción del proyecto.

● Seleccionar y utilizar herramientas colaborativas de diseño UML.

# **ENTREGABLES**

**DIAGRAMA DE CASOS DE USO**

Presentamos el Diagrama de Uso que realizamos en base a los requerimientos del sistema para esta Entrega.**DIAGRAMA DE ARQUITECTURA**

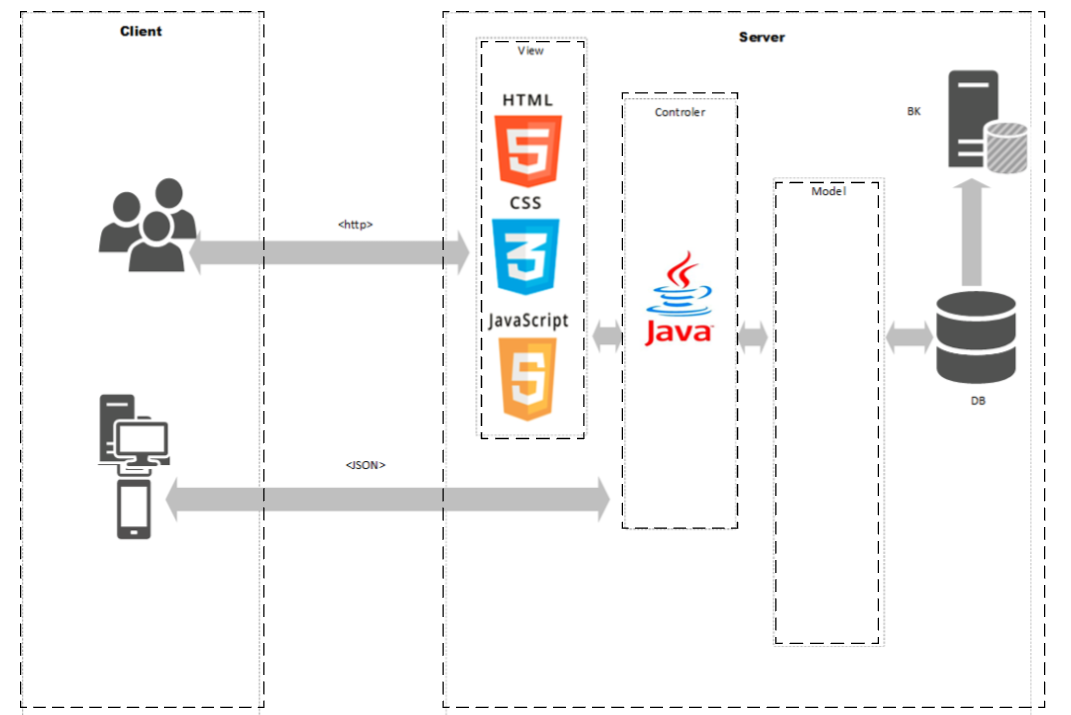
Elegimos la arquitectura basada en el tipo Modelo-Vista-Controlador (MVC).

El modelo será responsable de: acceder a la capa de almacenamiento de datos independiente de la base de datos.

La vista será responsable de: recibir datos del modelo y mostrar al usuario, tener registro del controlador.

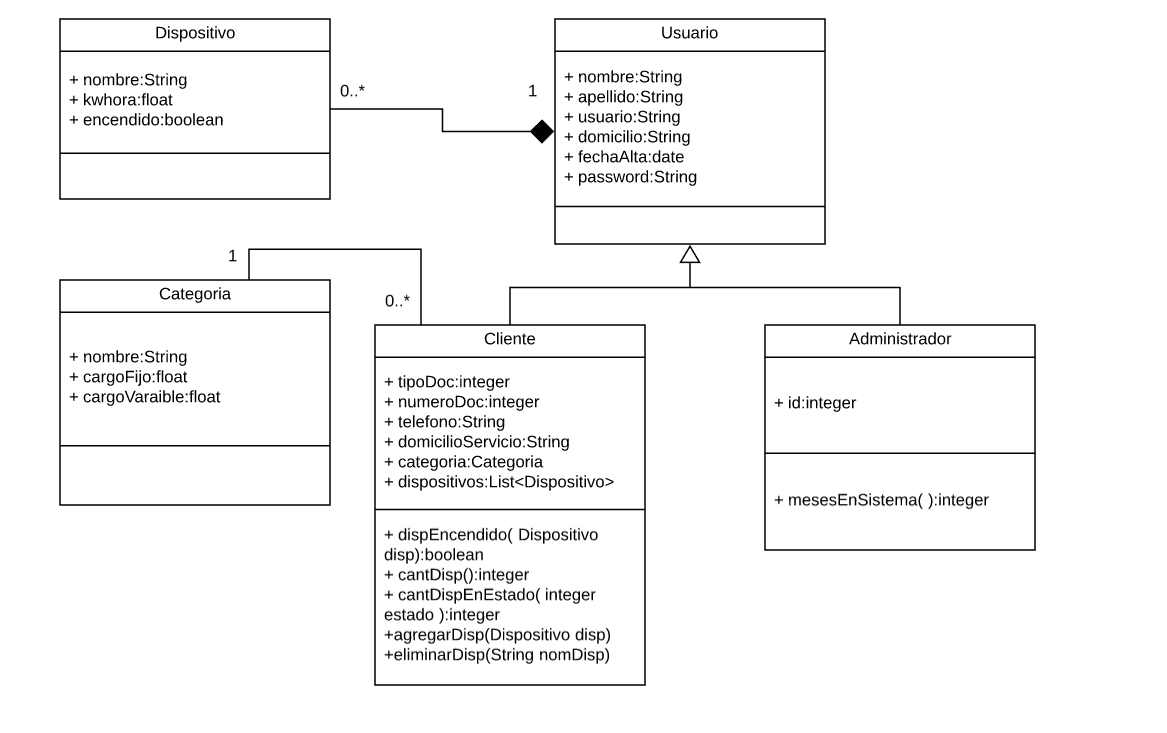
El controlador será responsable de: recibir los eventos de entrada(consultas, click,modificaciones).

En el diagrama de la arquitectura identificamos las tecnologías a utilizar como decisión de diseño, ya que son las que conocemos, y están sujetos a modificación ante la necesidad de adaptarlo a los requerimientos.



**DIAGRAMA DE CLASES INICIAL**

El diagrama de Clases, contiene la representación del modelo. 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.



**TABLA DE REQUERIMIENTOS NO FUNCIONALES**

Los Requerimientos No Funcionales que detectamos en esta etapa son:

|  |  |
| --- | --- |
| **REQUERIMIENTO** | **DESCRIPCION** |
| Rendimiento | El sistema debe soportar el manejo de una gran cantidad de información: modificada o accedida por 500000 de usuarios |
| Mantenibilidad | La aplicación debe poder ser mantenida, en caso de que falte algún integrante del desarrollo inicial, se debe documentar dicho desarrollo y mejoras aplicadas |
| Escalabilidad | La aplicación debe poder ser escalable, en el sentido de poder migrar de un servidor a otro y de agregar una base de datos |
| Usabilidad | Debe ser fácil de utilizar por los usuarios, es decir definir una vista de la aplicación que se intuitiva |
| Performance | Velocidad de proceso de la información, procesos que no deben tardar más de x segundos |
| Seguridad | Políticas de contraseñas seguras para el ingreso a la aplicación, extensión de 8 caracteres(letras y/o números) que ante 3 intentos fallidos se deba esperar 30 min. para blanquear la password a través del email |
| Precisión | La información debe ser correcta en el tiempo correcto; se debe corresponder en tiempo real |

# **IMPLEMENTACIÓN**

**DESARROLLO DE LA IMPLEMENTACIÓN**

**Lenguaje de Programación :** Java

**Entorno de trabajo:** Eclipse

**URL Repositorio:**  <https://github.com/AleOtero93/diseno/tree/entrega0>

**Branch:** entrega0

**MODELADO DE DISPOSITIVOS**

Se encuentra modelado en la clase Dispositivo del package Entidades

**MODELADO DE USUARIOS**

Modelado en Entidades>> Usuario, cuyas clases dependientes son : Administrador y Cliente.

**MODELADO DE CATEGORIAS**

Modelado en Entidades>>Categoria, según la especificación solicitada.

# **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.

# 

# **TABLAS DE DECISIONES DE DISEÑO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Decisión** | **Ventaja** | **Desventaja** | **Alternativa** |
| 18/04/2018 | Réplica de base de datos | Mayor disponibilidad y rápida recuperación ante fallas de BD | Mayor costo | Contar con sólo una BD |
| 14/04/2018 | Tecnología Java | Mayor soporte de ayudantes. Lenguaje más completo (Junit y dependencias) | Inexpertos por parte de los integrantes del grupo | Tecnología PHP |
| 14/04/2018 | Herencia | Utilizamos herencia para la clase Cliente y Administrador, creando una clase padre: Usuario, ya que los mismos comparten varios atributos |  |  |
| 14/04/2018 | Modelo MVC | Alta cohesión. Bajo acoplamiento. Elimina el problema de código Spaghetti | Mayor dedicación de tiempo en las etapas iniciales del proyecto | No usar este modelo |