

# Universidad de Buenos Aires

## Facultad de Ingeniería

### 75.10 Técnicas de Diseño

#### Trabajo Práctico 3

**GRUPO N° 20**

**Integrantes:**

CHELOTTI, ADRIANA GRETTEL	83513	<a href="mailto:adrichelo84@yahoo.com.ar">adrichelo84@yahoo.com.ar</a>
PEREZ STALTARI, DARIO MARTIN	83514	<a href="mailto:dmpstaltari@yahoo.com.ar">dmpstaltari@yahoo.com.ar</a>

**Repositorio:** <https://tecnicasdiseniogrupo20.googlecode.com/svn/trunk>

**Etiqueta:** <https://tecnicasdiseniogrupo20.googlecode.com/svn/tags/TP3>

## Índice

Enunciado .....	3
Explicación.....	4
Documentación .....	6
Arquitectura Utilizada. ....	6
Diagrama de Casos de Uso .....	7
Diagrama de Paquetes .....	8
Diagrama de Clases .....	9
Diagrama de Componentes.....	10
Diagrama de Despliegue .....	12
Interfaz Gráfica.....	13

## Enunciado

### Objetivo

Se pide diseñar y construir una aplicación que permita administrar los dispositivos de un edificio.

El edificio cuenta con sensores (de temperatura, de humedad, de humo, de ruido, de luz, etc), temporizadores (calendarios, relojes), detectores de movimiento y también cuenta con dispositivos como calefacciones, aires acondicionados, ventiladores, regaderas contra incendios, regadores de pasto, ventanas, puertas, luces, radios, televisiones, etc.

Se desea contar con una aplicación que permita al usuario ver el estado de los sensores y dispositivos, y administrar reglas sobre los mismos de acuerdo a sus necesidades y preferencias.

Un usuario podría configurar una regla que diga que si la temperatura baja de los 18 grados, encienda la calefacción, y que si la temperatura sube de los 25 grados la apague, a combinar reglas complejas como varios sucesos en tal orden, etc, etc.

Los dispositivos y sensores cuentan con unos drivers implementados en java y .net que permiten interactuar con ellos, que son provistos por el proveedor del dispositivo.

El sistema debe poder soportar incorporar nuevos sensores o dispositivos que se agreguen en el edificio, sin necesidad de recompilar la aplicación, solamente configurándola y agregando, los nuevos drivers necesarios, para que el usuario ya pueda definir reglas sobre los mismos.

Al implementar el sistema, será necesario construir mockups para los dispositivos y sensores, que puedan utilizarse a modo de prueba durante el desarrollo. Queda a libertad de ustedes definir las interfaces y que es capaz de hacer cada dispositivo. Los mismos deberán contar con la interfaz de su driver, y una implementación con cierta lógica del dispositivo o sensor.

Se está pensando incorporar toda una gama de nuevos dispositivos para manejo de seguridad del edificio, donde alguien encargado de la seguridad pueda definir reglas de monitoreo, activación de alarmas, luces, etc. Dado el sistema genérico que se está planteando debería ser sencilla la incorporación de estos nuevos equipos en un futuro cercano.

La aplicación deberá contar con una interfaz donde poder configurar los dispositivos existentes y activos, otra donde poder ver el estado de los mismos y poder realizar cambios en sus propiedades para interactuar con los dispositivos, por ejemplo abrir una ventana, y otra interfaz bastante amigable donde el usuario pueda crear y configurar las reglas dinámicamente sobre dichos dispositivos.

Se desea también contar con alguna forma de acceso remoto, aun no está bien definido por qué medio, pero podría ser a través de internet, por PC o celular por ejemplo, que permita ver mínimamente las reglas disponibles y cambiarlas su estado de activas / inactivas. Sería ideal poder contar con un manejo visual, sobre imágenes de las plantas de los edificios, de los dispositivos existentes, para poder interactuar con ellos y ver el estado de los mismos.

## Explicación

El Manejador de Suceso es una API que permite configurar sucesos ocurridos en un dominio cualquiera. Esta API permite que objetos puedan suscribir implicaciones a las mismas. Con implicaciones nos referimos a un conjunto de sucesos que tienen que ocurrir para que se produzca una acción de interés. Los objetos que quieran utilizar la API, deben crear dichos sucesos diferenciados por un identificador que los caracterice y pueden tener un identificador de un suceso que lo cancele.

Además deben crear una clase que implemente la interfaz Acción. Dentro de esta clase se debe implementar la acción a efectuar por el objeto cliente.

El cliente tendrá la posibilidad de configurar la API, según el comportamiento que desee.

Existen 4 configuraciones posibles:

- Sucesos Continuos.
- Sucesos Discontinuos.
- Secuencia Continua.
- Secuencia Discontinua.

### Sucesos Continuos

En esta configuración, la comparación se da si el conjunto de sucesos esperados por la implicación está incluido en el conjunto de sucesos ocurridos sin importar el orden.

### Sucesos Discontinuos

En esta configuración, la comparación se da si el conjunto de sucesos esperados por la implicación ocurrieron en su totalidad sin importar el orden pero sí que los mismos sean continuos.

### Secuencia Continua

En la configuración Secuencia Continua, se comparan los sucesos de las implicaciones suscriptas a la API con los sucesos ocurridos. Para que se ejecute la acción, la secuencia de sucesos de la implicación debe ocurrir en forma continua y en orden.

### Secuencia Discontinua

En la configuración Secuencia Discontinua, se comparan los sucesos de las implicaciones suscriptas a la API con los sucesos ocurridos. Para que se ejecute la

acción, la secuencia de sucesos de la implicación debe ocurrir en forma discontinua y en orden.

Otra posibilidad que ofrece la API, es permitir habilitar o deshabilitar las cancelaciones entre sucesos. La cancelación de sucesos cancelables entre si se producen uno a uno. Se considera que dos sucesos son cancelables cuando el identificador del suceso cancelador del último de los sucesos ocurridos concuerda con el identificador del primero de los sucesos ocurridos.

El objeto cliente puede agregar un suceso o un conjunto de sucesos a la API, los cuales serán notificados a las implicaciones que fueron suscriptas. Los sucesos que ocurrieron antes de la inscripción de una implicación no serán visibles por esta. Teniendo en cuenta esto último, la API mirará las implicaciones inscriptas y mediante un evaluador corroborara si los sucesos que espera ocurrieron o no. En caso de cumplir con los sucesos de la implicación según la configuración establecida procederá a ejecutar la acción de interés para el objeto cliente.

El edificio inteligente consta de pisos en cada uno de los cuales se tiene un manejador de sucesos. De esta forma cada piso tendrá dispositivos sobre los cuales podrá encenderlos o apagarlos y contara con sensores. Los sensores son los responsables de obtener mediciones sobre el ambiente y notificar lo sucesos al manejador de sucesos. Las acciones clientes que se utilizaron son las de prender dispositivo y apagar dispositivo. El edificio a su vez tiene dos listas de drivers, una para los drivers de los dispositivos y otra para los drivers de los sensores. Estos drivers pueden cargarse además en run time.

La aplicación también provee un servicio web con el que puede accederse en forma remota, se podrán consultar las reglas que tiene el edificio, como así también habilitar o deshabilitar algunas de ellas.

## Documentación

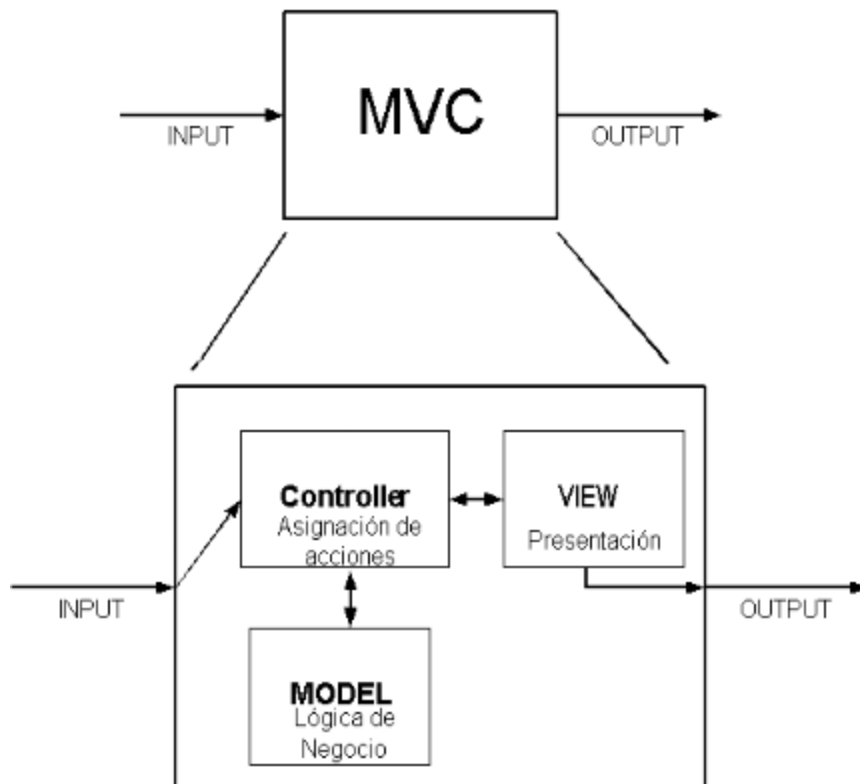
### Arquitectura Utilizada.

Se realizó una aplicación web con Struts 2, la arquitectura del sistema está basada en el patrón MVC (Modelo-Vista-Controlador), una arquitectura que busca reducir el acoplamiento dividiendo las responsabilidades en 3 capas claramente diferenciadas:

**Modelo:** En esta parte de la arquitectura, se encuentran las clases que manejan las reglas de negocio, como son las que usa la API de manejo de sucesos. Aquellas que son para modelar el edificio, y las interfaces de los drivers.

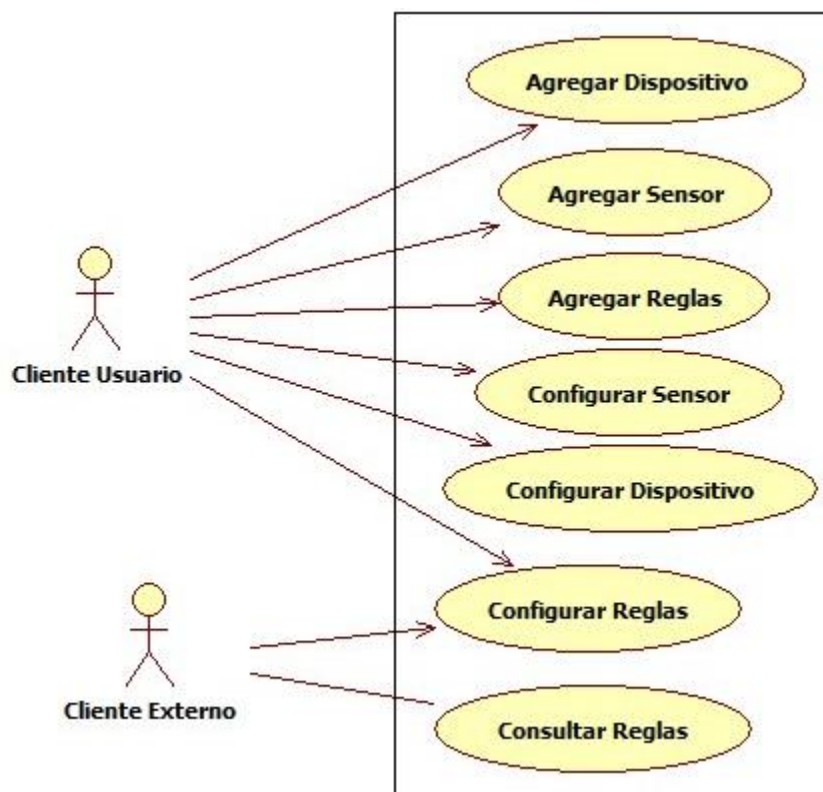
**Controlador:** En esta parte encontramos las clases que extienden Actions que comunican la vista y el modelo respondiendo a eventos generados por el usuario en la vista, efectuando cambios en el modelo, y devolviendo a la vista la información del modelo, necesaria para que se pueda generar la respuesta adecuada para el usuario.

**Vista:** En esta parte se tienen las vistas que es la interfaz con la que interactúa el usuario.

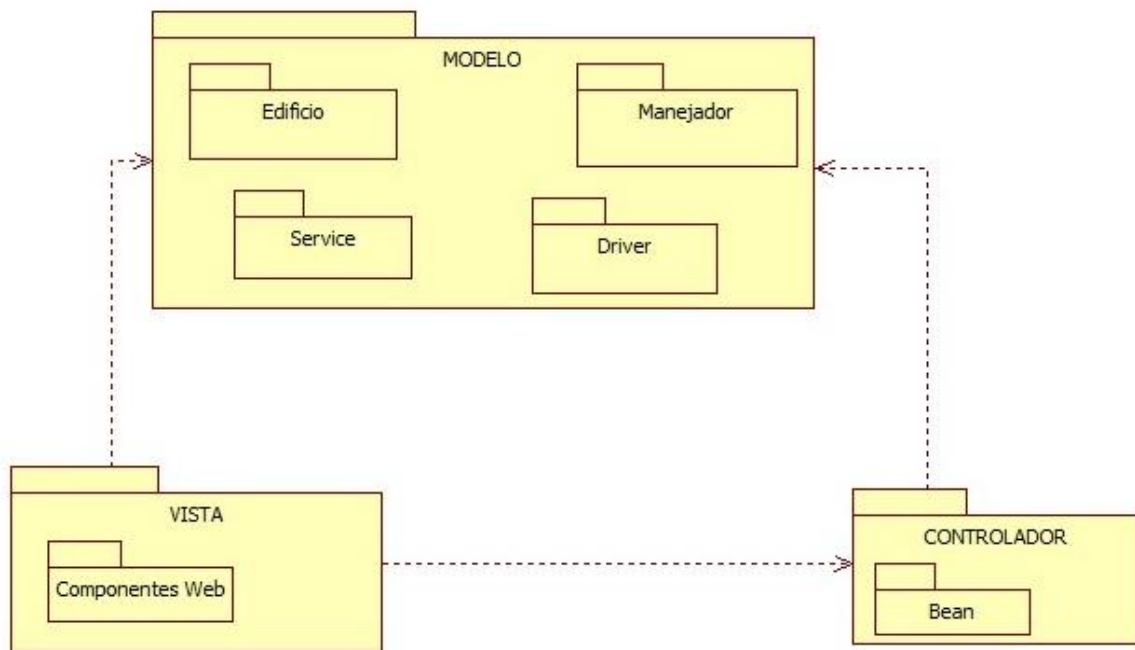


## Diagrama de Casos de Uso

En el mismo se ven los Casos de Uso extraídos de los requerimientos. Se muestran los correspondientes al agregado y configuración de reglas, dispositivos y sensores. Existen también aquellos que corresponden a las consultas, en este caso solo se representó el correspondiente a la consulta de reglas, dado que interactúa con distintos actores, se omitieron siendo válidas las consultas restantes. Dentro de los actores distinguimos al Cliente Usuario como aquel que interactúa con la aplicación mediante su interfaz Web. Por su parte el cliente Externo es aquel que interactúa con el servicio Web que se brinda, pudiendo ser este una aplicación cliente.



## Diagrama de Paquetes



Modelo: En este paquete se encuentran las clases del modelo de negocio. El mismo está compuesto por otros paquetes:

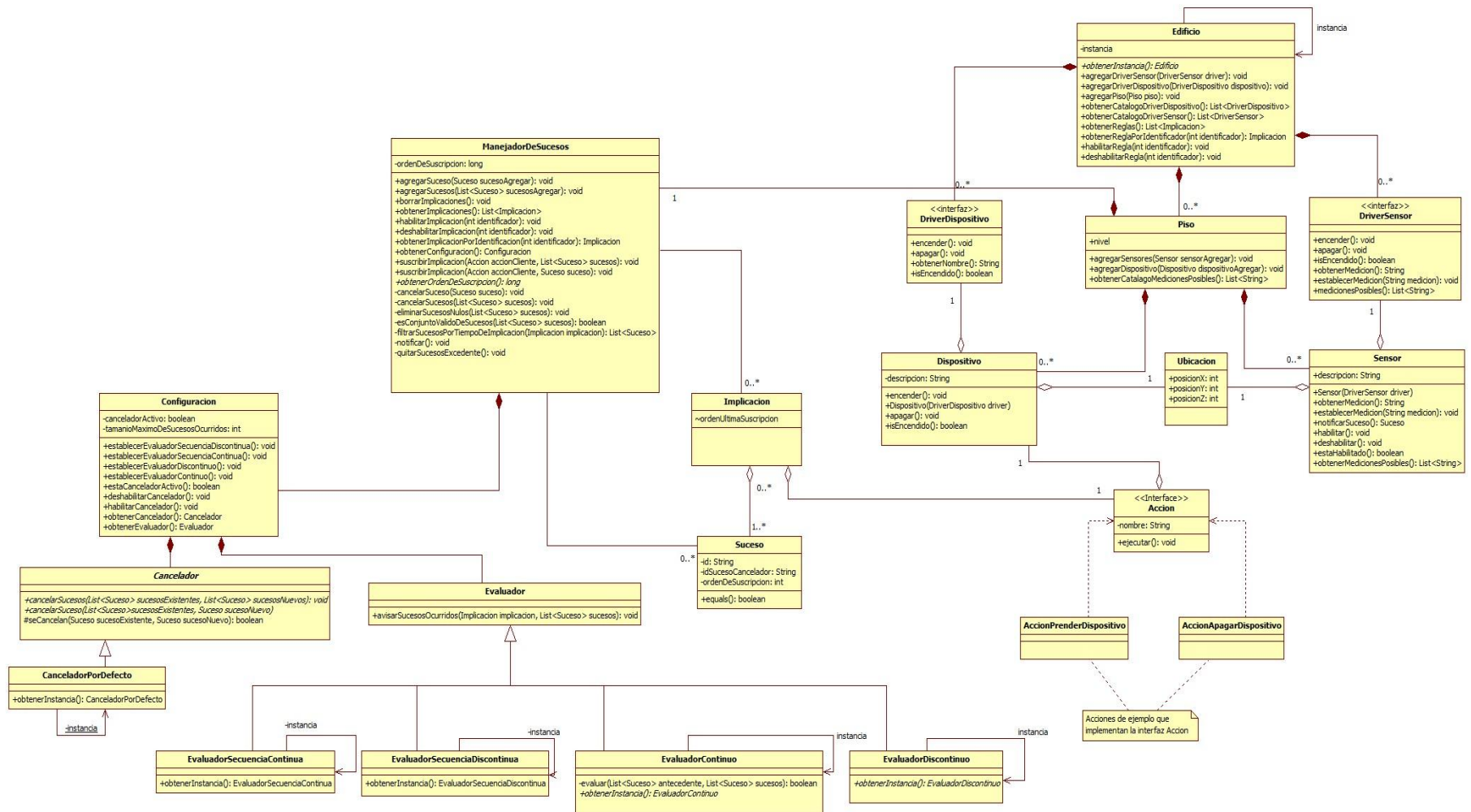
- Edificio: Se encuentran las clases que modelan al edificio con sus pisos, sensores y dispositivos.
- Manejador: Representa la API para el manejo de sucesos.
- Driver: Contiene las interfaces que deben implementar los drivers de los dispositivos y sensores que se deseen agregar.
- Service: Contiene una clase que modela el servicio web de Reglas.

Vista: En este paquete se encuentra los .jsp que representan las vistas de la aplicación.

Controlador: En este paquete se encuentran los Actions que responde los eventos del usuario e invoca peticiones al modelo.



## Diagrama de Clases

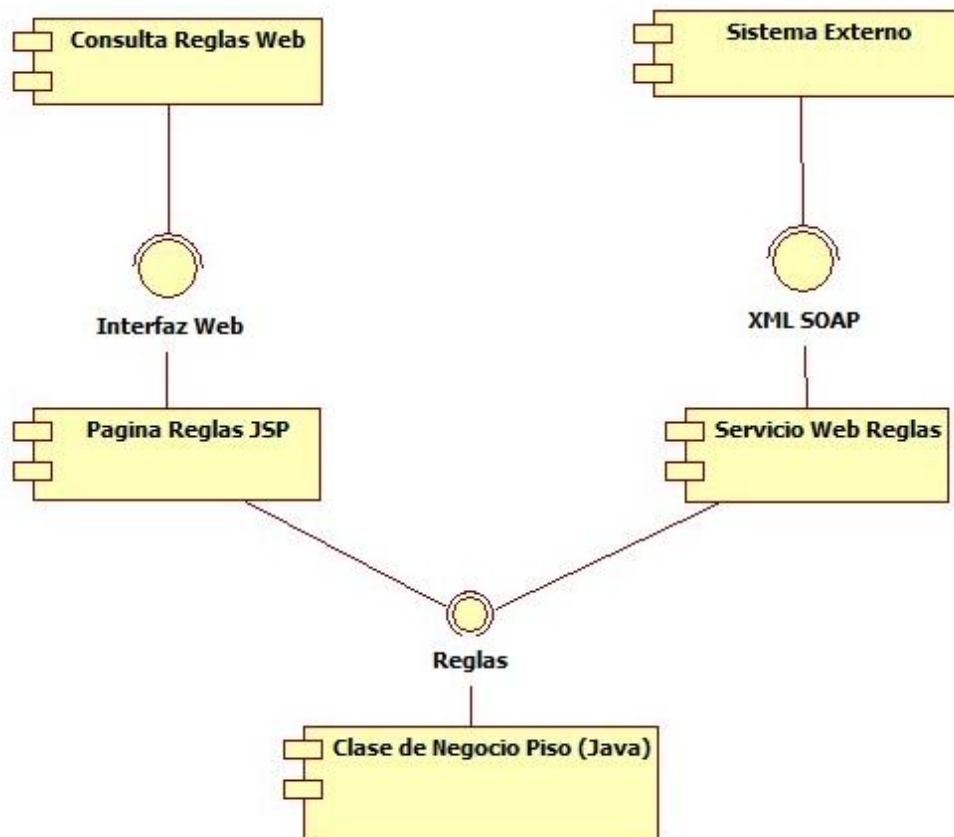


## Diagrama de Componentes

### CU: Consultar Reglas

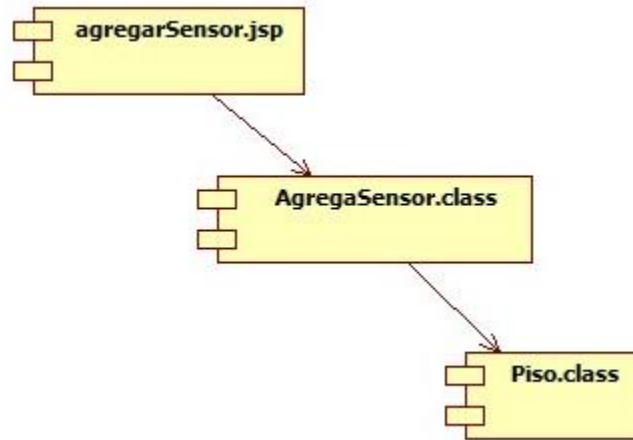
El diagrama de componentes muestra los componentes y dependencias para el CU de consultas de reglas. En el mismo se establecen los siguientes componentes:

- Consulta Reglas Web: El mismo hace referencia a un navegador que utiliza la interfaz web provista por la pagina JSP de Reglas.
- Pagina Reglas JSP: Representa a la página web que provee la interfaz y actúa como punto de acceso a la aplicación.
- Sistema Externo: Representa a cualquier sistema que consuma en forma remota el servicio web de reglas.
- Servicio Web Reglas: Representa a la publicación del servicio web, proveyendo una interfaz de acceso a la aplicación sin necesidad de ingresar a la aplicación.
- Clases de Negocio Piso: El mismo hace referencia a la clase de negocio Piso, en la misma se encuentra la interfaz para acceder a la reglas. Dado que estas últimas son propias de cada piso.



## CU: Agregar Sensores

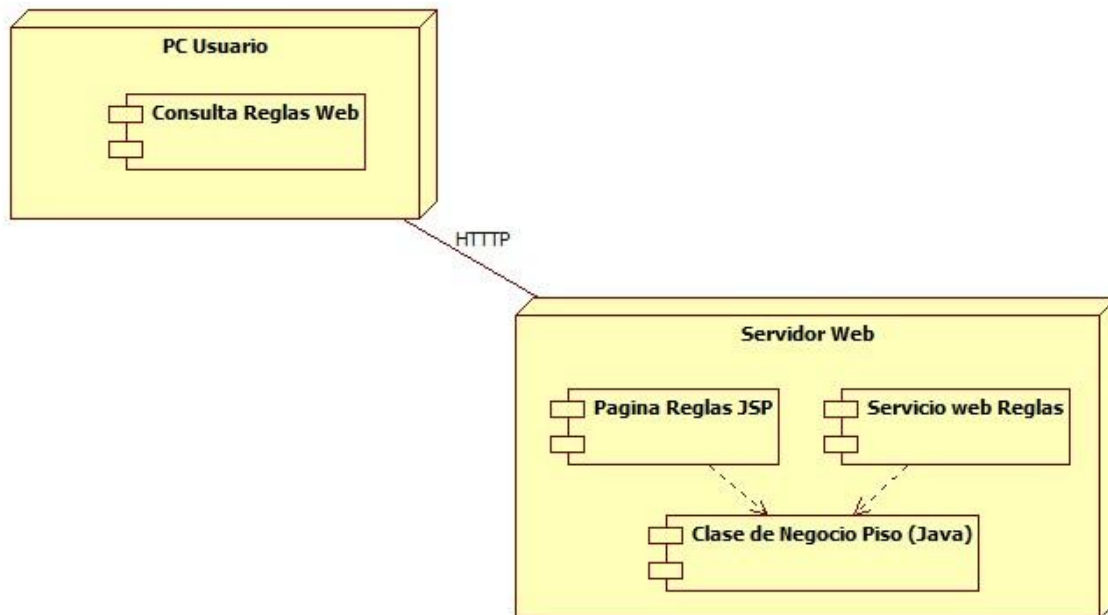
El diagrama de componentes muestra los componentes y dependencias para el CU agregar sensores. En el mismo se establecen los siguientes componentes:



- agregarSensor.jsp: Representa a la página web que provee la interfaz y permite agregar un sensor por parte del cliente.
- AgregaSensor.class: Representa a la clase de control que permite comunicar la vista con el modelo.
- Piso.class: Representa a la clase del modelo que contiene una colección de sensores en la que se agregara el sensor en cuestión.

Los diagramas de componentes de los casos de uso restantes resultan similares a los efectuados.

## Diagrama de Despliegue

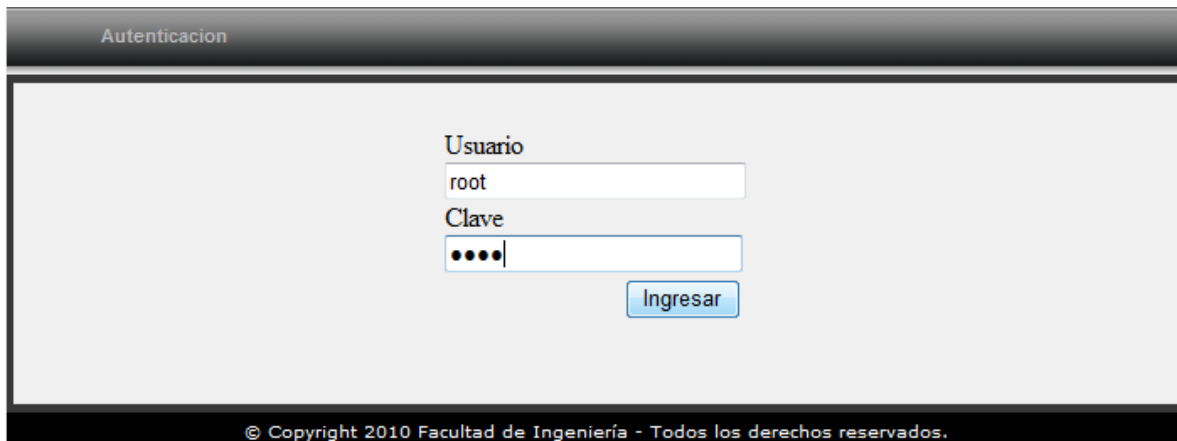


Cada cubo representa un nodo de hardware y las líneas que los unen son conexiones. Los nodos representados son:

- **PC Usuario:** Representa a la computadora desde donde el usuario interactúa con la aplicación web.
- **Servidor Web:** Representa al servidor donde se despliega la aplicación.

## Interfaz Gráfica

En primer lugar tenemos una pantalla de autenticación de usuario:



The image shows a web-based authentication interface. At the top, there is a dark header bar with the word "Autenticacion" in white. Below this, the main content area has a light gray background. It contains two input fields: the first is labeled "Usuario" and contains the text "root"; the second is labeled "Clave" and contains four black dots, indicating a password. To the right of the password field is a blue button with the text "Ingresar" in white. At the bottom of the page, there is a dark footer bar with the copyright notice "© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados."

Una vez lograda una autenticación exitosa, visualizaremos el siguiente menú:



## Edificio

El usuario podrá para un piso seleccionado agregar dispositivos y sensores. Para ellos deberá seleccionar el driver correspondiente.

The screenshot shows a web application interface with a dark navigation bar at the top containing links: Home, Edificio, Dispositivos, Sensores, Reglas, and Salir. Below this is a header section titled 'Pisos'. The main content area is titled 'Listado de Pisos' and contains a table with three columns: 'Nivel', 'Agregar Dispositivos', and 'Agregar Sensores'. The table has two rows for levels 0 and 1. Below the table, there is a copyright notice: '© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.'

Nivel	Agregar Dispositivos	Agregar Sensores
0	<a href="#">Agregar Dispositivos</a>	<a href="#">Agregar Sensores</a>
1	<a href="#">Agregar Dispositivos</a>	<a href="#">Agregar Sensores</a>

Para Agregar Dispositivos, se tiene la siguiente pantalla, donde se pide seleccionar el driver correspondiente y una breve descripción.

The screenshot shows the 'Agregar Dispositivos' form within the 'Pisos' section. The form has two input fields: 'Elija Driver:' with a dropdown menu showing 'Luces-Driver' and 'Descripción:' with a text input field containing 'luz 1'. Below the form are two buttons: 'Guardar' and 'Volver'. The same copyright notice is at the bottom: '© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.'

Elija Driver:	Luces-Driver ▼
Descripción:	luz 1

[Guardar](#) [Volver](#)

Para Agregar Sensores, se tiene la siguiente pantalla, donde se pide seleccionar el driver correspondiente y una breve descripción.

The screenshot shows a web application interface with a dark grey navigation bar at the top containing the links: Home, Edificio, Dispositivos, Sensores, Reglas, and Salir. Below this is a black header bar with the title 'Pisos' in white. The main content area is white and contains a form titled 'Agregar Sensor'. The form has two rows: the first row is labeled 'Elija Driver:' and has a dropdown menu with 'Sensor Humo-Driver' selected; the second row is labeled 'Descripción:' and has a text input field containing 'sensor humo 1'. Below the form are two buttons: 'Guardar' (blue) and 'Volver' (purple). At the bottom of the page is a black footer bar with the text '© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.'

## Dispositivos

En primer lugar se pide seleccionar el piso donde se quiere ver los dispositivos.

The screenshot shows a web application interface with a dark grey navigation bar at the top containing the links: Home, Edificio, Dispositivos, Sensores, Reglas, and Salir. Below this is a black header bar with the title 'Dispositivos' in white. The main content area is white and contains a form titled 'Listado de Pisos'. The form has a single row with a label 'Nivel' and a dropdown menu with '0' selected. Below the form is a button labeled 'Ver Dispositivos'. At the bottom of the page is a black footer bar with the text '© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.'

A continuación, vemos los dispositivos que tenemos. Para cada dispositivo se visualiza si el mismo esta Encendido o Apagado. También se permite cambiar el estado de los mismos.

Home
Edificio
Dispositivos
Sensores
Reglas
Salir

Dispositivos

Listado de Dispositivos

Dispositivo	Estado	Ejecutar	
Estufa	Apagado	Encender ▾	<input type="button" value="cambiar"/>
luz 1	Apagado	Encender ▾	<input type="button" value="cambiar"/>

Volver

© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.

## Sensores

De forma análoga tenemos el listado de sensores para un piso seleccionado.

En cada sensor se permite cambiar la medición del mismo así como también si el mismo se encuentra o no habilitado.

Home
Edificio
Dispositivos
Sensores
Reglas
Salir

Sensores

Listado de Sensores

Sensor	Medicion Actual	Mediciones Posibles	Estado Actual	Cambiar Estado
sensor de temp 1	TEMPERATURA_ALTA	TEMPERATURA_ALTA ▾ <input type="button" value="cambiar"/>	Habilitado	Habilitar ▾ <input type="button" value="cambiar"/>
sensor humo 1	HUMO_EXISTENTE	HUMO_EXISTENTE ▾ <input type="button" value="cambiar"/>	Habilitado	Habilitar ▾ <input type="button" value="cambiar"/>

Volver

© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.



## Reglas

Se podrán visualizar las reglas que tiene un piso en particular de la siguiente forma:

[Home](#) [Edificio](#) [Dispositivos](#) [Sensores](#) [Reglas](#) [Salir](#)

**Reglas**

Listado de Reglas

Reglas	Sucesos Esperados	Dispositivo	Accion	Estado Actual	Cambiar Estado
0	TEMPERATURA_BAJA	Estufa	Encender	Habilitada	Habilitar Cambiar
1	HUMO_EXISTENTE TEMPERATURA_ALTA	Estufa	Apagar	Habilitada	Habilitar Cambiar

[Volver](#)

© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.

También se ofrece la posibilidad de habilitar o deshabilitar una regla de interés.

Para agregar una regla, se contara con la siguiente pantalla:

[Home](#) [Edificio](#) [Dispositivos](#) [Sensores](#) [Reglas](#) [Salir](#)

**Reglas**

Configuración de Reglas

Regla	Sucesos	Suceso Esperados	Dispositivo	Acciones
NUEVA	TEMPERATURA_ALTA <input type="button" value="agregar"/>	TEMPERATURA_BAJA	luz 1	<input type="button" value="Apagar"/>

[Cancelar](#)

© Copyright 2010 Facultad de Ingeniería - Todos los derechos reservados.

Interactuando por las distintas opciones se podrá ver como los dispositivos cambian de estado.