



UNIVERSIDAD DE MONTEMORELOS

FACULTAD DE INGENIERÍA Y TECNOLOGÍA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

“INTERFAZ DE USUARIO PARA CONFIGURACIÓN DE SISTEMAS DOMÓTICOS”

TESIS

JACINTO ABESO MOLONGUA NCHAMA

ASESORES

**ING. IGNACIO CRUZ DOMÍNGUEZ
ING. ALEJANDRO GARRIDO SOTO**

Interfaz de usuario para la configuración del recinto domótico.

Jacinto Abeso Molongua Nchama, Ignacio Cruz, Alejandro Garrido.

Facultad de Ingeniería y Tecnología, Universidad de Morelos, Morelos, Nuevo León, México.

Resumen—Las interfaces han sido un elemento importante para el uso de cualquier tecnología, ya que una interface amigable permite lograr la facilidad de uso que el ser humano busca. Una interfaz conecta diferentes elementos y permite llevar a cabo la interacción entre ellos. Las interfaces gráficas específicamente permiten que la interacción humano-dispositivo pueda llevarse a cabo de una manera mas transparente y sencilla. En este proyecto se desarrolla una interfaz gráfica para la representación del recinto domótico y sus elementos. Mediante esta interfaz se pueden agregar o eliminar áreas del recinto domótico, y agregar o eliminar elementos a cada área del recinto. El objetivo es lograr la especificación y configuración de los elementos pertenecientes al recinto de una forma sencilla, amigable y escalable. Ésta interface es implementada en Web y accesible desde cualquier dispositivo. La interface fue implementada en Java utilizando el marco de trabajo proporcionado por Appfuse. En la capa de datos se utilizó Hibernate y para la inyección de dependencias se utilizó Spring. El software se ejecuta sobre un servidor Apache y utiliza la base de datos de MySQL, ambos fueron instalados mediante el paquete XAMP el cual facilita la puesta en marcha de los servidores.

Palabras claves— domótica, escalabilidad, interfaz, web.

I. INTRODUCCIÓN

La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de la comunicación entre el usuario y el sistema con el fin de aumentar el confort [1].

En este proyecto, se desarrolló una interfaz de usuario para la configuración del recinto domótico. Ésta interfaz permite construir de una manera gráfica una configuración personalizada de dispositivos en un recinto domótico. La interfaz a desarrollar permitirá, de manera sencilla, la manipulación de diferentes aspectos de los dispositivos tales como la cantidad de dispositivos, su ubicación, estado e información, permitiendo la escalabilidad y cambios según se requieran.

A. Antecedentes

Gracias a la tecnología se han realizado diferentes tipos de interfaces de usuario. De acuerdo a la forma como el usuario puede interactuar con una interfaz, estos tipos pueden ser los siguientes:

Interfaces alfanuméricas (intérpretes de comandos), que solo presentan texto.

Interfaces gráficas de usuario (GUI, Graphics User Interfaces), las que permiten comunicarse con el ordenador de una forma muy rápida e intuitiva representando gráficamente los elementos de control y medida.

Interfaces táctiles, que presentan gráficamente un “panel de control” en una pantalla sensible que permite interactuar con el dedo de forma similar a si se accionara un control físico [2].

Autores como Barbacil [1], Villalba [3] y Domínguez [4] estos últimos han hecho interfaces domóticas centradas sobre una cuestión específica, interfaces que solo tienen acceso a una tarea, y se manejan con un solo tipo de dispositivos para que el usuario pueda interactuar.

En este proyecto, se extenderá la funcionalidad de tal manera que la interfaz además de ser amigable, permitirá la manipulación de los dispositivos dentro del recinto domótico. La interfaz para realizar esta manipulación podrá ser accesible desde dispositivos Android y iPad utilizando una conexión a Internet.

B. Definición del problema

Una de las debilidades que se encuentran generalmente en las interfaces para domótica es que son limitadas a un escenario o ambiente específico, resuelve una sola situación o un solo problema y no son escalables ni responsivas. Con estas limitaciones no se puede lograr el verdadero objetivo de la domótica que es hacer las cosas mas sencillas y confortables para el ser humano. También varias interfaces para domótica solo están hechas para dispositivos específicos de esta forma el usuario se encuentra limitado a utilizar solo tal plataforma específica.

Este proyecto persigue resolver estas limitaciones al desarrollar una interfaz configurable y que pueda utilizarse en diferentes plataformas.

C. Justificación del problema

El sector de la domótica ha evolucionado considerablemente en los últimos años, y en la actualidad ofrece una oferta más consolidada. Hoy en día, la domótica aporta soluciones dirigidas a todo tipo de viviendas, incluidas las construcciones de vivienda oficial protegida. Además, se ofrecen más funcionalidades por menos dinero, más variedad de producto, que gracias a la evolución tecnológica, son más fáciles de usar y de instalar.

En definitiva, la oferta es mejor y de mayor calidad, y su utilización es ahora más intuitiva y perfectamente manejable por cualquier usuario. Paralelamente, los instaladores de domótica han incrementado su nivel de formación y los modelos de implantación se han perfeccionado. Asimismo, los servicios posventa garantizan el perfecto mantenimiento de todos los sistemas.

En general, la domótica de hoy contribuye a aumentar la calidad de vida, hace más versátil la distribución de la casa, cambia las condiciones ambientales creando diferentes escenas predefinidas, y consigue que la vivienda sea más funcional al permitir desarrollar facetas domésticas, profesionales, y de ocio bajo un mismo techo. En la domótica se utiliza aparatos como son los sensores y actuadores, estos últimos son los que se enlazan en la programación de una casa inteligente y con estos se establece una configuración a través de una interface [5].

En este proyecto se desarrolló una interfaz para domótica con un lenguaje de programación que sea accesible a cualquier tipo de dispositivo móvil, escalable y amigable, de esta forma el usuario podría tener más control de las instalaciones del hogar y ser más consciente de las tareas que tiene que realizar desde cualquier lugar que se encuentre.

Hacer una interfaz para el sistema domótica es algo que casi la mayoría de los sistemas tienen pero hacer una interfaz domótica que sea escalable y amigable a todo tipo de dispositivo móviles y en cualquier lugar, significa un gran beneficio para el usuario.

D. Objetivos

1) General:

Desarrollar una interfaz web para asignar elementos domóticos al recinto de tal manera que el administrador de viviendas inteligentes pueda establecer los componentes del recinto antes de instalarlos físicamente tales componentes.

2) Específicos:

Investigar acerca de la tecnología que se va a realizar.
Creación de la interacción con el sistema.
Seleccionar la herramienta de desarrollo apropiado.
Crear el software.
Creación del software.

E. Hipótesis

Es posible integrar tecnologías de desarrollo Web como Java, Bootstrap, JavaScript, XML y el framework Spring para crear una interfaz de usuario para la configuración de un recinto domótico

II. FUNDAMENTOS TEÓRICOS

A. Marco teórico

El termino domótica proveniente del idioma francés “domotique” que quiere decir conjunto de servicios proporcionados por sistemas tecnológicos integrados para satisfacer las necesidades básicas de seguridad, bienestar, comunicación, gestión y el confort del usuario que es el más cercano al sistema [6]. En este sentido podemos definir la domótica como disciplina que estudia el desarrollo de infraestructuras inteligentes, así como las tecnologías de la información en edificios.

Los primeros que comenzaron a utilizar el término inteligente para distinguir aquellos terminales con capacidad de procesamiento de datos de aquellos sin esa capacidad, fue el sector informático, o también llamados no inteligentes. A comienzos de los 80, algunas revistas especializadas del tema, comenzaron a referirse como domótica a edificios con características particulares en cuanto a prestaciones de confort en sintonía con su entorno tales como “edificios inteligentes”, “edificios listos”, “edificios automatizados”, o inclusive “edificios electrónicamente optimizados”. Sin embargo, no existe una definición muy clara que nos pueda informar lo que es un edificio inteligente.

Una definición acordada en el Simposio Internacional de la especialidad llevado a cabo en 1985 en Toronto es: “Un edificio inteligente combina innovaciones, tecnológicas o no, con una gestión competente a efectos de maximizar el retorno de la inversión” [7]. Una definición aportada por el AIBI (American Intelligent Building Institute) es: “Un edificio que usa los cuatro elementos fundamentales de la construcción: estructura, sistemas, servicios y gestión, como así también su interacción optimizada a efectos de maximizar el retorno en la inversión y proveer un ambiente de trabajo altamente eficiente, confortable y conveniente”.

Otra definición del EIBG (European Intelligent Building Group) consiste en: “Un edificio que le permita a los usuarios desarrollar su máxima eficiencia al menor costo de operación posible y que usa eficientemente sus recursos. Esto significa que el edificio debe proveer respuesta rápida, altamente eficiente y de generación de un ambiente de soporte al usuario para la consecución de sus objetivos de negocio” [8]. Otra definición aportada por el DEGW (An international building design consultancy company) dice de esta manera: “un edificio inteligente es uno que brinda adecuadas respuestas al usuario y tiene la

habilidad de adaptarse a las nuevas tecnologías o cambios en las estructuras organizacionales” [9].

Sin embargo estos conceptos no deben confundirse con la idea de un recinto automatizado.

a) Recinto Automatizado:

Es un término utilizado para referirse a cualquier edificio o vivienda que posea algún tipo de automatismo con sistemas no integrados entre sí, de tal manera que presente una respuesta adecuada ante una solicitud prevista y que se encuentre dentro de un rango específico y ordenado para que actúe consecuentemente. Una vivienda es recinto domótico cuando incluye una infraestructura propia de cableado y equipos necesarios para brindar servicios avanzados, optimizando a la vez las funciones dentro del hogar, planteándose el objetivo de permitir una mayor calidad de vida a través de la tecnología, ofreciendo una reducción del trabajo doméstico, un aumento del bienestar y de la seguridad de sus habitantes y un mayor control en el uso de la energía. También en la domótica es muy importante los dispositivos a utilizar, podemos encontrar variedades de dispositivos les presento algunos mas utilizados.

Controlador: Los controladores son los dispositivos que administran el sistema según la programación y la información que reciben.

Actuadores: Los actuadores son dispositivos capaces de recibir una orden procedente de un sistema de control y realizar una acción que modifique el estado de un determinado equipo o instalación: encendido o apagado, subida o bajada, apertura o cierre [4].

Sensores: Los sensores o detectores son dispositivos capaces de recoger la información de los distintos parámetros que controlan (el nivel de presión de una tubería, la temperatura ambiente, el suministro de gas natural.) y de transmitir esta información para su procesamiento [4].

Bus de datos: Los buses definen su capacidad de acuerdo a la frecuencia máxima de envío y al ancho de los datos. En medio de transmisión y de información, interconexión y control, entre los distintos dispositivos de los sistemas de domótica puede ser de varios tipos. Los principales medios de transmisión son:

- **Cableado Propio:** La transmisión de cable propio es el medio mas común para los sistemas demóticos, principalmente son de tipo par trenzado, coaxial o fibra óptica.
- **Cableado Compartido:** Varios soluciones utilizan cables compartidos y/o redes existentes para la transmisión de su información, por ejemplo la red eléctrica (corrientes portadoras), la red telefónica o la red de datos.
- **Inalámbrica:** Muchos sistemas de domótica utilizan soluciones de transmisión inalámbrica entre los distintos dispositivos, principalmente tecnologías de radiofrecuencia o infrarrojo. En este ultimo se pretende trabajar con una interfaz adecuada para el manejo de la casa domótica.

Interface: Los interfaces se refiere a los dispositivos (pantallas, móvil, Internet, conectores) y los formatos (binario, audio) en que se muestra la información del sistema para los usuarios (u otros sistemas) y donde los mismos pueden interactuar con el sistema. Es preciso destacar que todos los dispositivos del sistema de domótica no tienen que estar físicamente separados, sino varias funcionalidades pueden estar combinadas en un equipo. Por ejemplo un equipo de central de domótica puede ser compuesto por un controlador, actuadores, sensores y varios interfaces.

A través de la interfaz de usuario convertimos un sistema domótico en una herramienta de convivencia. En este sentido, para que una interfaz de usuario pueda ser considerada como buena, su diseño debe cumplir cuatro criterios [10].

- Debe ser natural.
- Debe ser fácil de aprender.
- Debe ser fácil de usar.
- Debe ser consistente.

Una interfaz natural es aquella que se maneja intuitivamente, de forma similar a aquella a la que el usuario está habituado, y que por ello no provoca en el sentimiento de rechazo. Adicionalmente, una buena interfaz de usuario debe ser fácil de aprender: ha de proporcionar ayuda suficiente para usuarios inexpertos (sin que esta facilidad en el aprendizaje se convierta en un obstáculo para aquellos usuarios más habituados a su manejo). Finalmente, una buena interfaz debe ser consistente. Una interfaz es consistente cuando mantiene un estilo, asistencia y pautas de interacción uniformes, con independencia de la tarea a realizar [10].

b) Aplicaciones:

Algunas de las aplicaciones y servicios que se utiliza en el ámbito demótico son:

Ahorro energético Climatización: Según los sistemas instalados, es posible diferenciar zonas de la vivienda con regulación independiente de temperatura. De esta manera se puede optimizar el consumo de aire acondicionado o calefacción. En este mismo campo situaremos el control de ventanas y persianas automáticas, sobre las que se actúa en función de las condiciones climatológicas, la hora del día y lo programado por los usuarios.

Gestión eléctrica: En la gestión eléctrica es un aspecto importante que se puede clasificar en diferentes categorías como.

- **Racionalización de cargas eléctricas:** desconexión de equipos de uso no prioritario.
- **Reducción de la potencia contratada.**
- **Uso de energías renovables**

Nivel de confort: Nos referimos a aquellas aplicaciones y servicios que permiten mejorar la calidad de vida de los usuarios al aportar soluciones que facilitan la realización de tareas domésticas rutinarias, que suponen una comodidad

añadida y que simultáneamente optimizan el consumo energético.

III. ESTADO DE ARTE

El concepto que se entiende por interfaz es de carácter general ya que en la domótica se realiza muchos tipos de interfaces que han realizado trabajos diferentes para el confort del ser humano por ejemplo:

Barrera Durang, y Nelson [11]. Hicieron un sistema de interfaz básica de bajo costo para aplicaciones didácticas para estudiantes de la universidad Antioquía utilizando el protocolo zigBee, para la conexión de la interfaz de, esta interfaz fue aplicado al lenguaje Java, en el cual, una tarjeta embebida se le conecta a todos los dispositivos del campus, y este estado lo puedes visualizar desde una computadora que a su vez funciona como un servidor web.

También Carolina Arboleda, diseñó una interfaz de cerebro-computador para facilitar la comunicación de personas con discapacidad motora, esta interfaz fue diseñada con MATLAB 7.0, utilizando la función "load_bcidat" que viene incluida en herramientas de análisis off-line del BCI2000, para cargar los datos adquiridos [12].

Interfaz de Usuario, por Juan C. Cuevas, Jesús Martínez interfaz, estos realizaron una interfaz domótica utilizando el protocolo X10, se puede observar en la fig.1, esta interfaz, esta interfaz tenía el control de todos los dispositivos del hogar ya sea dentro de una casa o fuera de la misma [13].



Fig.1 ejemplo de interfaz domótica de Usuario.

En junio de 2010, M.Álvarez y Á. Arquero, realizaron una interfaz de control de edificio, este interfaz se encarga de ayudar al usuario a realizar ciertas tareas de control domótico de las instalaciones urbanas y edificios del Campus de la universidad evaluando, monitorizando y gestionando datos procedentes de sensores situados en el Campus utilizando la tecnología como SIG, y desarrollado en el lenguaje de programación de alto nivel C# incluido entre otros lenguajes de entorno Visual Studio.NET [14].

S. Dormido, J. Sánchez y F. Morill [15]. Desarrollaron una interfaz de Cliente-Servidor para la orientación domótica, esta interfaz se compone de dos partes, en la parte del servidor, da la configuración y gestión local,

dando así una visión global del estado de la red y la configuración. La otra parte es la del cliente, que a través de una conexión TCP/IP mantiene el diálogo con el servidor, haciendo las tareas como activación de servicios de la red o variación de configuración del estado domótico, a fin de poder controlar mediante una computadora.

Gonzalo Zabala de la Universidad Abierta Interamericana, desarrolló una interfaz con varios Kit electrónicos con un diseño especial, utilizadas para aplicación electrónicos de y automatización eléctrica de baja potencia, esta interfaz se puede interactúan en PC. El Kit se conecta a una PC, por medio del programa se puede observar en la fig 2 como se interactúa el Cliente-Servidor a través de control de dispositivos como lamparitas, motores de corriente continua o relés. Se destaca más en manejar las entradas y salidas digitales controladas por software a través de un Puerto paralelo. [16].



Fig.2 se muestra la interfaz Cliente –Servidor en ámbito de recepción de señal.

Interfaz local Una interfaz local también fue desarrollada para Agente virtuales 3D para el control de entornos inteligentes de una sala domótica vía web, como lo podemos observar en la fig.3, esta interfaz lleva un control de dispositivos físicos que se realizo mediante el bus domótica EIB (European Installation Bus), en la actualidad conocido como KNX, también se implementaron Ethernet para realizar información multimedia basando también del lenguaje XML. Esta interfaz se caracterizo de un agente virtual 3D que permite la interacción vía voz en lenguaje natural con un sistema remoto. A través de este interfaz el usuario no solo obtiene información sobre el estado de los diferentes elementos (puertas, luces, televisor,...) si también quiere puede cambiarlo, así se logro desarrollar esta interfaz local para gente discapacitada. [15].



Fig.3 una interfaz local establecida en una vivienda.

Interfaz de voz Montano, Saz y Lleida, realizaron sistema de interfaz hombre – maquina de una silla de ruedas con reconocimiento de voz, se puede observar en la fig.4 en este interfaz el usuario introduce los sistemas en unas ondas de forma verbales y el sistema reconoce la voz y la interpreta en señales o ondas, y este la transforma las señales en voz a través de un micrófono en unos parámetros que representa la señales y sus variaciones, para realizar esto se utilizo los parámetros MFCC (Mel-Frequency Cepstrum Coefficients), por su capacidad de reducir al menor número de parámetros la información más relevante de la señal de voz. [17].



Fig.4 interfaz de voz a traves de una silla de ruedas.

Interfaz web Mauricio Esteban Pardo, Guillermo Enrique Strack y Diego C. Martínez [18]. Implementaron un sistema domótico basado en una interfaz de servicio web, la utilización del sistema desde la Web es intuitivo al hacer uso de la técnica AJAX, que permitió brindar una apariencia similar al uso de la aplicación ubicada en la casa, en esta aplicación, con la aplicación web los usuarios del sistema podrían iniciar una sesión y realizar las mismas tareas que puede ejecutar otras aplicaciones anteriores se puede visualizar la fig.5 Esto es: como el encendido y apagado de los distintos dispositivos también ingreso, modificación y eliminación de las rutinas pre programadas. También se tomara en cuenta que dicha sesión va a poder ser iniciada únicamente cuando el programa anterior (Control De Dispositivos) esté conectado al servidor, es decir que la computadora in-situ disponga conexión a

Internet activa y la aplicación se esté ejecutando. Esta restricción es naturalmente necesaria para la interacción entre el hogar y el servidor. De lo contrario, podría ocurrir que remotamente se intente cambiar el estado de un dispositivo hogareño cuando es imposible que la orden llegue a los mismos. Un aspecto importante a remarcar es que las

Acciones indicadas remotamente por medio de esta aplicación web no son comunicadas a la computadora del hogar, sino que es ésta quien indaga al servidor sobre los cambios requeridos. Esta comunicación unidireccional es explicada en la siguiente sección.



Fig.5 interfaz web con diferentes funciones en la aplicación.

IV. METODOLOGIA

En primer lugar se eligió la herramienta para el desarrollo de la interfaz. Existen diferentes opciones de herramientas y lenguajes de programación tales como Ruby, C++, C#, Java, Python y Html. Cada uno de estos lenguajes tiene diferentes características y ventajas. La selección de la herramienta se realizó por conveniencia de acuerdo a la experiencia que ya se tenía.

Para el acceso a datos se tienen también diferentes tecnologías, entre ellas están MongoDB, MySQL, Postgresql, etc. que están disponibles de forma gratuita.

MongoDB, tiene unas ventajas como realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente en las bases de datos para ya después poder ejecutarlas, también cuenta con un sistema de archivos que cuenta con la capacidad de balancear la carga y recopilar datos utilizando múltiples servidores para almacenar archivos. Y una de las desventajas que tiene MongoDB, es que bloquea la base de datos cada vez que se realiza una escritura, retorna la información del espacio del almacenamiento, y esto puede ocasionar perdida de información y tiene problemas de rendimiento cuando el volumen supera los 100GB.

MySQL es un software libre, su velocidad es muy buena, esto le cataloga como uno de los gestores con mas rendimiento. Tiene un costo muy bajo, es fácil de configurar e instalar soporta una variedad de Sistemas Operativos, tiene una baja probabilidad de corromper datos,

incluso si los errores no se producen en el propio gestor y su conectividad, velocidad y seguridad son altamente apropiado para acceder a bases de datos en internet.

Para iniciar con el desarrollo se instalaron las siguientes herramientas: XAMPP, Postgresql, Netbeans 8.1, PgAdmin III, IntelliJ Idea 15, Sublime Text, Python y el SDK de Java.

XAMPP: es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, es servidor Apache y es intérpretes para lenguajes de script.

Postgresql: es un sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL.

NetBeans es un ambiente integrado de desarrollo de código abierto para desarrolladores de software que intentan construir aplicaciones usando mayormente Java.

PgAdmin III es una aplicación grafica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source.

IntelliJ IDEA es un entorno desarrollo integrado de Java (IDE), para desarrollar programas informáticos con licencia Apache.

Python es un lenguaje de programación de uso general, orientado a objetos e interpretado.

Java es un software que provee herramientas de desarrollo para la creación de programas java.

El desarrollo se llevó a cabo en el lenguaje Java debido a las ventajas que nos proporciona su sistema de ejecución en tiempo real y su semántica sencilla. Con Java puedes desarrollar aplicaciones web dinámicas y así crear programas modulares y códigos reutilizables. El manejo de la base de datos se llevó a cabo utilizando PostgreSQL y el cliente de postgresql llamado PgAdmin que proporciona una interface gráfica para la administración de la base de datos. La base de datos se utilizó para guardar los cambios generados en las especificaciones del recinto doméstico en cualquier instante.

Una de las principales herramientas utilizadas fue Spring. Spring es un framework que proporciona los objetos necesarios de los que la aplicación depende. Éste framework fue de gran utilidad debido a que puede interactuar con otros frameworks tales como Hibernate que se encarga de la capa de datos del modelo MVC. En la figura 6, podemos ver que si cada framework llegara a generar sus propios objetos esto generaría problemas ya que cada framework tiene su propio ciclo de vida de los objetos. Al utilizar Spring, los objetos que requeridos por los otros frameworks fueron inyectados y administrados por éste a través de anotaciones y archivos de configuración XML.

El código utilizado puede ser reutilizado por otros proyectos que no usan Spring, en caso que sea necesario ya que la mayor parte de éste fue desarrollado utilizando POJOS, los cuales son objetos simples de Java que promueven la separación de características.

El Spring se utilizó en la inversión del control. La configuración de Spring esta basada principalmente en archivos XML o con anotaciones que describen las dependencias entre los componentes. En este proyecto el

uso de Spring proporcionó una delimitación clara de las capas del modelo MVC, lo cual es una buena práctica de programación. Otra de las facilidades proporcionadas por Spring, es la inyección de dependencias, lo cual fue un factor muy importante para la programación de esta aplicación.

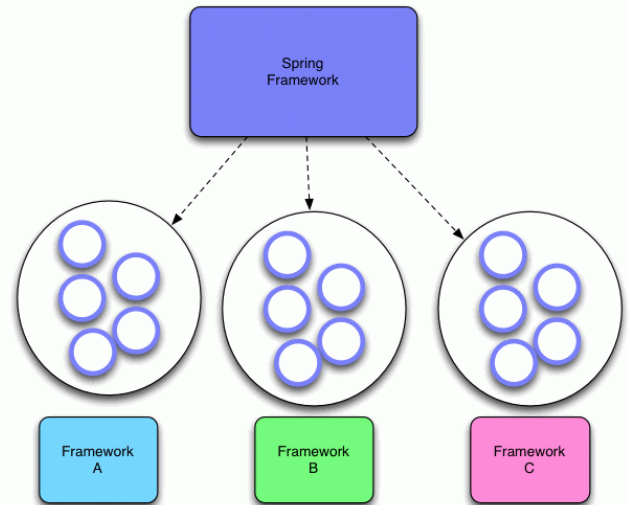


Fig. 6 Spring con diferentes framework

A. PASOS

- a) Primero se crearon los pojos, los pojos son módulos o mas bien modelos que contiene la interfaz del recinto doméstico que son el modelo casas, como se puede observar en la figura 7. Este modelo casa debe contar con las identidades que debe contener una casa como se visualiza en la imagen siguiente.


```

package com.thio.myinterfaz.model;

import ...

/**
 * Created by thio on 10/03/16.
 */
@Entity
@Table(name = "casas")
public class Casa {
    public Casa() { this.status = "A"; }

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Column(length = 70, nullable = false, unique = true)
    @NotEmpty
    private String nombre;
    @Column(nullable = false)
    @NotEmpty
    private String direccion;
    @Column(length = 2, nullable = false)
    private String status;

    public Long getId() { return id; }

    public void setId(Long id) { this.id = id; }

    public String getNombre() { return nombre; }

    public void setNombre(String nombre) { this.nombre = nombre; }

    public String getDireccion() { return direccion; }

    public void setDireccion(String direccion) { this.direccion = direccion; }

    public String getStatus() { return status; }

    public void setStatus(String status) { this.status = status; }
}

```

Fig.7 modelo casa con su identidad

- b) Segundo paso se crearon los servicios CasaManager, los servicios CasaManager se una extensión del modelo de la lista de las casas que se va a manejar y llevar el control de los datos de la casa como podemos ver en la figura 8 el modelo servicios CasaManager.

```

package com.thio.myinterfaz.service;

import ...

/**
 * Created by thio on 10/03/16.
 */
public interface CasaManager extends GenericManager<Casa, Long> {
    public List<Casa> getCasaList(final Casa casa);

    public Casa getCasa(final Long id);
}

```

Fig.8 servicios de extensión de CasaManager.

- c) Tercer paso creamos los DAO (del inglés Data Access Objects) de la casa, que podemos ver en la figura 9, estos nos ayudan a suministrar la interfaz y otros dispositivos de almacenamiento de datos y archivos que vayamos a generar en la casa

```

package com.thio.myinterfaz.dao;

import ...

/**
 * Created by qsofo on 10/03/16.
 */
public interface CasaDao extends GenericDao<Casa, Long> {
    public List<Casa> getCasaList(final Casa casa);

    public Casa getCasa(final Long id);
}

```

Fig.9 creación del modulo CasaDao.

- d) Cuarto paso se crea el modulo CasaDaoHibernate, que se puede ver en la siguiente figura 10.

```

package com.thio.myinterfaz.dao.hibernate;

import ...

/**
 * Created by thio on 10/03/16.
 */
@Repository("casaDao")
public class CasaDaoHibernate extends GenericDaoHibernate<Casa, Long> implements CasaDao {
    public CasaDaoHibernate() { super(Casa.class); }

    @Override
    public List<Casa> getCasaList(Casa casa) {
        return getSession().createCriteria(Casa.class)
            .add(Restrictions.eq("casa", casa))
            .list();
    }

    @Override
    public Casa getCasa(Long id) {
        return getSession().createCriteria(Casa.class)
            .add(Restrictions.idEq(id))
            .uniqueResult();
    }
}

```

Fig.9 creación del modelo Híbrante de CasaDaoH.

- e) En este paso creamos el modulo que se dedicara al control de la casa, como sea tanto grabar como eliminar y editar la casa como podemos visualizar en la figura 11.

```

package com.thio.myinterfaz.webapp.controller;

import ...

/**
 * Created by thio on 10/03/16.
 */
@Controller
@RequestMapping("/casa")
public class CasaController extends BaseController {
    private Logger log = LoggerFactory.getLogger(this.getClass());
    private CasaManager mgr;

    @Autowired
    public void setCasaManager(CasaManager mgr) { this.mgr = mgr; }

    @RequestMapping(value = "/lista", method = RequestMethod.GET)
    public String lista(Model model) {
        log.debug("Entrando al metodo 'lista'");

        List<Casa> casas = mgr.getCasaList(null);
        log.debug("Se recuperaron las casas {} ", casas);

        model.addAttribute("casaList", casas);
        return "casaList";
    }

    @RequestMapping(value = "/id", method = RequestMethod.GET)
    public String getCasa(@PathVariable Long id, Model model) {
        log.debug("Entrando al metodo 'getCasa'");

        Casa casa = mgr.getCasa(id);
        log.debug("Casa {} ", casa);

        model.addAttribute("casaForm", casa);
        return "casaForm";
    }

    @RequestMapping(value = "/grabar", method = RequestMethod.POST)
    public String saveCasa(@Validated Casa casa, Model model, BindingResult errors, HttpServletRequest request) {
        log.debug("Entrando al metodo 'saveCasa'");
    }
}

```

Fig.11 el modulo Controller de la casa.

- f) En este paso se crea un test los test son pruebas que se hace antes de programar estas pruebas nos ayudan asegurar que nuestro código programado o a programar hace lo que necesitamos y en la figura 12, podemos ver como se obtiene una casa.

```

package com.thio.myinterfaz.dao;

import ...

/*
 * Created by thio on 10/03/16.
 */
public class CasaDaoTest extends BaseDaoTestCase {
    @Autowired
    private CasaDao dao;

    @Test
    public void testGetCasaList() throws Exception {
        log.debug("Entrando al metodo 'testGetCasaList' ");
        List<Casa> casas = dao.getCasaList(null);
        assertEquals(2, casas.size());
    }

    @Test
    public void testGetCasa() throws Exception {
        log.debug("Entrando al metodo 'testGetCasa' ");
        Casa casa = dao.getCasa(-1L);
        assertNotNull(casa);
    }
}

```

Fig.12 código de prueba de creación de casa.

- g) En el paso siete implementamos CasaManagerTest, que este nos ayuda a la prueba de implementación de la casa, para asegurar que los requisitos que contiene la casa están ejecutando de una forma correcta como podemos apreciar en la figura 13.

```

import ...

/*
 * Created by osoto on 10/03/16.
 */
public class CasaManagerTest extends BaseManagerTestCase {
    @InjectMocks
    private CasaManagerImpl mgr;

    @Mock
    private CasaDao dao;

    @Test
    public void testGetCasaList() {
        log.debug("Entrando al metodo 'testGetCasaList' ");

        //given
        final List casas = new ArrayList();
        log.debug("dao "+ dao);
        given(dao.getCasaList(new Casa())).willReturn(casas);

        //when
        List result = mgr.getCasaList(null);

        //then
        assertEquals(casas, result);
    }

    public void testGetCasa() {
        log.debug("Entrando al metodo 'testGetCasa' ");

        //given
        final Long id = -1L;
        final Casa casa = new Casa();

        //when
        Casa result = mgr.getCasa(id);

        //then
        assertEquals(casa, result);
    }
}

```

Fig.13 pruebas de la casa.

- h) También se crea un modulo de internacionalización ya que una interfaz debe ser abierta a diferentes idiomas, esto ayuda a facilitar a los usuarios a interactuar siendo de cualquier país o hablando cualquier idioma, se puede ver en la figura 14 un pequeño código de la internacionalización de la interfaz.

```

package com.thio.myinterfaz.util;

import ...

public class DateConverterTest {
    private final DateConverter converter = new DateConverter();

    @Test
    public void testInternationalization() throws Exception {
        final List<Locale> locales = new ArrayList<Locale>() {
            private static final long serialVersionUID = 1L;
            {
                add(Locale.US);
                add(Locale.GERMANY);
                add(Locale.FRANCE);
                add(Locale.CHINA);
                add(Locale.ITALY);
            }
        };

        for (final Locale locale : locales) {
            LocaleContextHolder.setLocale(locale);
            testConvertStringToDate();
            testConvertDateToString();
            testConvertStringToTimestamp();
            testConvertTimestampToString();
        }
    }
}

```

Fig.14 código de internacionalización de interfaz.

Los pasos mostrados hasta éste punto se realizaron también con los módulos casa, cuarto, actuadores y sensores. Las entidades casa y cuartos son entidades diferentes, del mismo modo los actuadores y sensores tienen identidades diferentes, por ejemplo una casa tiene nombre del dueño de la casa, contiene una dirección de la casa y el estado de la casa, a diferencia que los cuartos tienen número de cuarto y cada cuarto tendría actuadores y sensores, a la vez cada sensor tendrá un actuador conectado. Por esto es muy importante identificar cada modulo y sus identidad. Para cada módulo se escribieron pruebas para verificar la operación correcta.

En el paso siguiente fue descarga y la instalación de las herramientas de bases de datos. La administración de la base de datos se realizó con postgresql utilizando la aplicación gráfica PgAdmin la cual se puede ver en la imagen de la figura 15. Al utilizar la estructura basada en Appfuse las operaciones en la base de datos se realizan de forma transparente con la ayuda de Spring y la capa de acceso a datos proporcionado por Hibernate. Integrar estas tecnologías permite obtener un software mas robusto de manera que los cambios impacten menos en el código de tal manera que requieran menos esfuerzo.



Fig.15 la base de datos que se generó.

Mediante la instalación y uso de XAMP se proporcionaron los servicios HTTP y de bases de datos, de tal manera que la base de datos y a la aplicación propiamente, fueran accesibles de manera local y remota.

V. PRUEBAS Y RESULTADOS

En las pruebas pude interactuar con Java, Xml, y Spring utilizando IntelliJ IDEA y almacenando los resultados en una base de datos con PostgreSQL, allí se realizó la parte de añadir y eliminar elementos como sea casa, cuartos y sensores y actuadores en forma general.

En esta pantalla de fig. 16 podemos visualizar en como el usuario podría acceder a la interfaz introduciendo el id usuario y seguido de una contraseña, ya que puede haber varias casas, es muy importante que el usuario tenga que identificarse y por ahora es a través de un Id de usuario y contraseña.

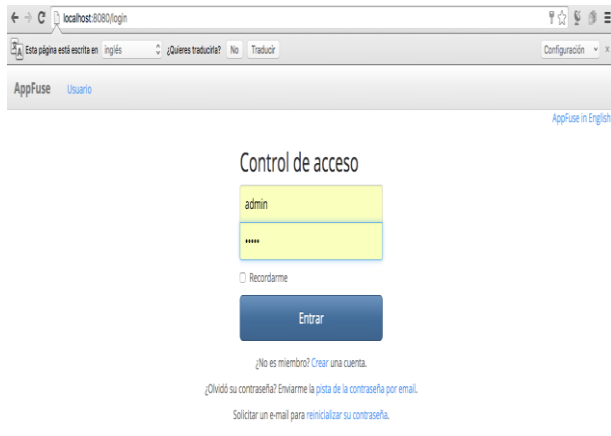


Fig.16 la interfaz con usuario y contraseña.

En la figura 17 se puede observar como el usuario puede dar de alta una casa solicitando el nombre del dueño de la casa y la dirección de esta misma, ya que las casas tendrían que tener una dirección y así ser identificados.

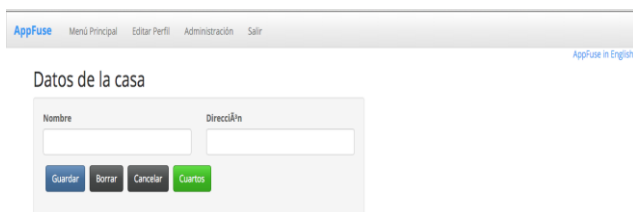


Fig.17 pantalla para dar de alta una casa.

En la imagen de figura 18 podemos ver como el usuario ya esta añadido en una casa. Allí podremos observar las dos casas creadas con sus direcciones ya que una casa debe tener un dueño y una dirección.

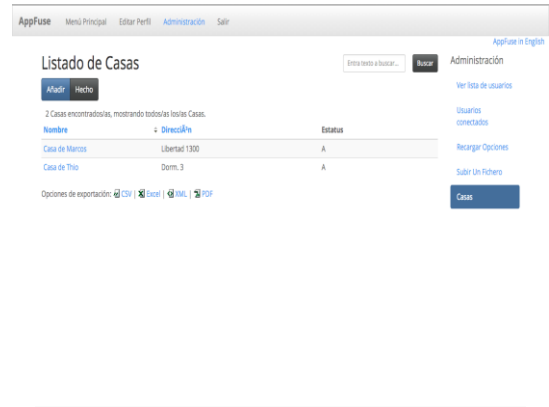


Fig.18 casa añadidas con sus respectivas direcciones

En la figura 19 se puede observar que el sistema le entrega al usuario el nombre de la casa donde quiere agregar el cuarto. Los cuartos son enumerados y el sistema guarda los cuartos en la base de datos. También se implementó la operación de eliminar.

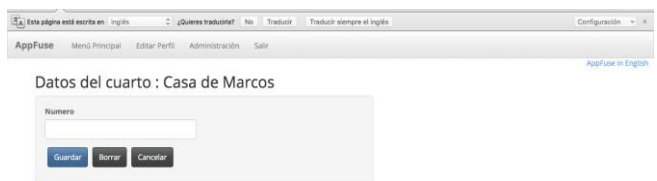


Fig.19 agregar cuarto para una casa.

En la siguiente figura 20 podemos ver como los cuartos ya están agregados en la interfaz con los nombres de los dueños de cada casa.

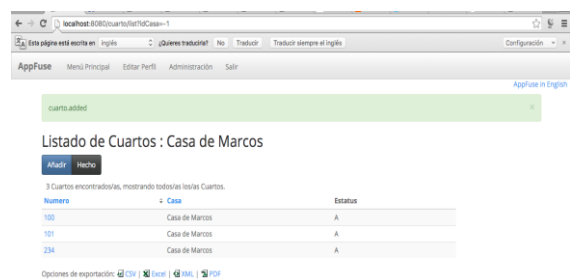


Fig.20 listado de cuartos ya guardados.

Por último, en la figura 21 podemos ver en la base de datos los registros que han guardado y que representan al recinto doméstico.

CONCLUSION

En este trabajo fue posible crear los objetos casa, cuarto y se pudo tanto agregar como eliminar casas, cuartos y guardar los resultados en una base de datos.

También como se esperaba, la interfaz está conectada vía web a través de un usuario y contraseña.

La interfaz puede ser utilizada en diferentes aspectos.

VI. TRABAJOS FUTUROS

Poder conectar la interfaz con los sensores y actuadores. Utilizar algún framework tal como interactjs que nos faltó experimentar.

id	numero	status	id_casa
1	100	A	-2
2	101	A	-1
3	102	A	-1
4	204	A	-1
5	1	A	-2

Fig.21 tabla de bases de datos de los cuartos de una casa.

Base de datos y las direcciones de las casas registradas en la interfaz se puede visualizar en la figura 22.

id	direccion	nombre	status
1	Libertad 1300	Casa de Sistemas	I
2	Dorm. 3	Casa de Thio	A
3	Libertad 1300	Casa de Marcos	A
4	Sabinos 200	Marcos	A

Fig.22 resultados de la base de datos de las casas.

C. Discusión

Después de la creación de la interfaz de usuario para recinto doméstico, se han implementado las operaciones para crear el recinto, añadir cuartos y eliminar los elementos que son manejados en la interfaz.

Este trabajo a diferencia de las interfaces citadas en [15] y [17], no depende de un único y exclusivo dispositivo para poder interactuar con ella, este tipo de interfaces se basan en un dispositivo único que se encarga de realizar todas las instrucciones del recinto.

Esta interfaz también se puede utilizar para el control de grupos, de la manera que en los grupos o equipos este ha de tener un líder, de igual forma sería útil para el registro del alumnado o personal de una organización.

D. Trabajos futuros.

Poder conectar la interfaz con los sensores y actuadores. Utilizar algún framework tal como interactjs que faltó experimentar.

REFERENCIAS

- [1] M. B. Juan M. "La domótica". Revista de Edificación, 27 (1998): 71-76 Servicio de Publicaciones de la Universidad de Navarra <http://hdl.handle.net/10171/17408>
- [2] E. interfaz de usuario 9 de marzo rticulo 155412 http://www.ecured.cu/Interfaz_de_usuario.155412
- [3] A. Villalba Tecnología, arquitectura y diseño integrados en la concepción de tiendas, experiencia de compra y nuevos espacios comerciales 10 de diciembre 2011 <http://www.tendencias21.net/domotica/>
- [4] D. M. & Sáez F, N. Sergio 2006. and L, Emilio Interfaces móviles en Dspace., 2014 . In IV
- [5] Cedom 2016-03-09 Asociacion Española de domótica e inmotica.
- [6] A. Krainier , Towards smart buildings. Architectural Assian Graduate School. Environment & Energy Studies Program (1996).
- [7] A. Harrison., Intelligent buildings in South Asia, E & F. Spon (1998).
- [8] A. Sidwell. Australia's intelligent home. Construction Industry Insititute. Australia. (1996).
- [9] So, A. Intelligent Building Systems. Editorial Kluwer Academic Press. (1999).
- [10] I. Illich capitulo 2 1974 http://tesis.uson.mx/digital/tesis/docs/22168/Capitulo_2.pdf
- [11] A. R. Al-Ali, M. Al-Rousan. "Java-based home automation system". *IEEE Transactions on Consumer Electronics*. Vol. 50. 2004. pp. 498- 504
- [12] EIA. Revista, ISSN 1794-1237 Número 11, p. 105-115. Julio 2009 Escuela de Ingeniería de Antioquia, Medellín (Colombia).Diseño y construcción de un prototipo de interfaz cerebro-computador para facilitar la comunicación de personas con discapacidad motora.
- [13] C. J. Cuevas y M. Jesús M. Pedro. El Protocolo x10: Una Solución Antigua a Problemas Actuales 93. 2002
- [14] M. Álvarez, Á. Arquero, E. Martínez, O. Río Domogis: prototype of a interface of a building control system integrated into a GIS Informes de la Construcción Vol. 62, 518, 15-24, abril-junio 2010. ISSN: 0020-0883 eISSN: 1988-3234 [doi: 10.3989/ic.08.034](https://doi.org/10.3989/ic.08.034)

- [15] E. Cerezo, y S. Baldassarri, E. Cuartero, F. Serón. Análisis e Investigación en IPO. Agentes virtuales 3D para el control de entornos inteligentes domóticos congreso de domótica mayo 2007.
- [16] IX Workshop de Investigadores en Ciencias de la Computación 978-950-763-075-0. 2007.
- [17] J.M. Alcubierre Mínguez, y M. Saz, Lleida Montesano Silla de Ruedas Inteligente Controlada por Voz (Harrison 1996) Primer Congreso Internacional de Domótica, Robótica y Teleasistencia para todos, 2005
- [18] M. E. Pardo. una implementación de un sistema de control domotico basada en servicio web. Departamento de ciencias e ingeniería de la computacion Universidad nacional del sur.