

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

**Interfaz Web para Laboratorios Remotos en
dispositivos móviles**

Autor: Edel Rodríguez Alvarez

Tutor: Dr. Iván Santana Ching

Santa Clara

2015

"Año 57 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

Interfaz Web para Laboratorios Remotos en dispositivos móviles

Autor: Edel Rodríguez Alvarez

E-mail: eralvarez@uclv.edu.cu

Tutor: Dr. Iván Santana Ching

E-mail: Ching@uclv.edu.cu

Santa Clara

2015

"Año 57 de la Revolución"

PENSAMIENTO

Mirada de cerca, la vida parece una tragedia; vista de lejos, parece una comedia. Nunca te olvides de sonreír, porque el día en que no sonrías será un día perdido.

Charles Chaplin

DEDICATORIA

Dedicado a mi mama y mi papa, sin ustedes esto no fuera posible.

AGRADECIMIENTOS

Gracias a mi familia, principalmente a mis padres, por su apoyo

Gracias a mis abuelos que sin ellos por su esfuerzo

Gracias a mi tía Caridad por su apoyo semanal

Gracias a mis compañeros de aula, por todos los desafíos, problemas y alegrías que compartimos

Gracias a mi tutor por su ayuda

Gracias al Javier alias el Chavi por su ayuda

RESUMEN

Un laboratorio remoto es un sistema de software y hardware que permite a los estudiantes utilizar experimentos reales ubicados físicamente en una universidad. Estos laboratorios suelen ser costosos por su equipamiento, problemas de seguridad, el tiempo y el mantenimiento. Acceder a un recurso real a través de una red, le permite al usuario controlar de forma remota, mediante una interfaz experimental que se ejecuta en un ordenador, a sistemas reales que no pueden acceder físicamente. Por eso el uso de las oportunidades que ofrecen estas nuevas tecnologías reduce estas limitaciones grandemente. Con el aumento del desarrollo de las nuevas plataformas móviles avanzadas, entre los estudiantes, se genera un nuevo desafío, que es la explotación de estos como un recurso más de aprendizaje. Se pretende diseñar una interfaz Web para un laboratorio remoto, a la cual puedan acceder los usuarios desde sus terminales móviles, mediante una revisión bibliográfica del objeto de estudio, exponer los principales *frameworks* de desarrollo de aplicaciones Web. Como resultado obtenido, la creación de la interfaz para móviles del Sistema de Laboratorio a Distancia, donde se comprobó las ventajas aportadas al SLD, mediante la realización de prácticas desde terminales móviles.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
RESUMEN	iv
INTRODUCCIÓN	1
Organización del informe	5
CAPÍTULO 1. ANÁLISIS DE LOS LABORATORIOS REMOTOS	6
1.1 Definición de Laboratorios Remotos	6
1.2 Niveles que conforman un Sistema de Laboratorios Remotos	8
1.2.1 Interfaz de usuario	8
Selección de la tecnología para el desarrollo de la interfaz de usuario	8
1.2.2 Gestión de prácticas	9
Selección de la tecnología para la gestión de prácticas	9
1.2.3 Procesamiento de las prácticas	9
Selección de la tecnología para el procesamiento de las prácticas	10
1.3 Herramientas de administración y gestión.....	10
1.4 Análisis de laboratorios remotos que brindan servicio a móviles	14
1.5 Apreciaciones sobre el capítulo	24
CAPÍTULO 2. SELECCIÓN DEL FRAMEWORK	26
2.1 Marco Conceptual.....	26
2.2 Web App o Native App	28
2.3 Frameworks de lenguaje <i>HTML5</i>	30

2.3.1	Twitter BootStrap	31
2.3.2	Foundation 4	32
2.4	Frameworks del lenguaje CSS	33
2.4.1	960 Grid System	33
2.4.2	Blueprint CSS	34
2.5	Frameworks del lenguaje JavaScript	35
2.5.1	Backbone.js	35
2.5.2	Ember.js	36
2.5.3	Angular.js	37
2.6	Selección del framework para la construcción de la interfaz Web para el SLD...38	
2.7	Consejos a seguir para el diseño de una página Web para móviles.....41	
2.8	Re-direccionar la nueva interfaz para dispositivos móviles	42
2.9	Consideraciones finales del capítulo.....43	
CAPÍTULO 3. RESULTADOS OBTENIDOS		45
3.1	Arquitectura General del Sistema	45
3.2	Estructura de la página Web	47
3.3	Estructura de las páginas de prácticas.....51	
3.4	Comprobación de los resultados	52
3.5	Revisión de los resultados.....54	
3.6	Análisis económico.....55	
CONCLUSIONES Y RECOMENDACIONES		56
Conclusiones		56
Recomendaciones		56
REFERENCIAS BIBLIOGRÁFICAS		57

INTRODUCCIÓN

Estudios de psicología cognitiva han demostrado que las personas reciben el conocimiento mucho mejor al realizar cosas y reflexionar sobre los resultados de sus acciones que mirar o escuchar a alguien que les dice lo que deben conocer (Dormido, 2004a).

La práctica de laboratorio es el espacio de enseñanza en el cual el estudiante se desarrolla y obtiene destrezas prácticas, que le permiten establecer criterios de ingeniería, comprobar y entender los conceptos teóricos y, además, relacionar conocimientos previos que ya posee.

Los laboratorios realizan diversas acciones y crean diferentes objetivos como: la práctica, comprobación, validación, verificación, generación, construcción, aplicación e integración. Además, de fomentar el trabajo en equipo, la interdisciplinariedad, promover la participación, la innovación y desarrollar competencias (E. Lindsay, 2007).

Dentro de los tipos de laboratorios, los laboratorios remotos obtienen importancia por los beneficios que brindan, permite a los estudiantes utilizar experimentos reales ubicados físicamente en una universidad. Estos laboratorios suelen ser costosos por su equipamiento, problemas de seguridad, el tiempo y el mantenimiento, pero acceder a un recurso real a través de una red, le permite al usuario controlar de forma remota, mediante una interfaz experimental que se ejecuta en un ordenador, a sistemas reales que no pueden acceder físicamente.

Con el uso de las oportunidades que ofrecen estas nuevas tecnologías reduce estas limitaciones grandemente. Con el aumento del desarrollo de las nuevas plataformas móviles avanzadas, como *iPhone* o *Android* entre los estudiantes, se genera un nuevo desafío, es la explotación de estos como un recurso más de aprendizaje. La adaptación de laboratorios remotos para dispositivos móviles se ha estudiado en la literatura (D. López-de-Ipiña, 2006)

para los dispositivos más antiguos. Por lo cual se busca poder incrementar las potencialidades de los laboratorios remotos a través de los beneficios que brindan los dispositivos móviles.

La integración de la tecnología móvil con sistemas de laboratorios remotos es un trabajo que tiene mucha fuerza en la actualidad por los beneficios que presentan. Muchos son los artículos en los que se abordan el uso de la tecnología móvil con los laboratorios a distancia, que sirven como antecedentes.

En el acceso remoto a un recurso real, el usuario maneja y controla de forma remota sistemas reales mediante una interfaz de experimentación que se ejecuta en un ordenador conectado a una red. El sistema real puede o no estar conectado a un ordenador, aunque lo común es que lo esté.

Los laboratorios remotos han demostrado ser de gran calidad a la hora de llevarse a cabo los experimentos y pueden mejorarse la calidad de los experimentos con la integración de otras herramientas digitales para el aprendizaje. Experimentos complejos como VISIR1 han sido construidos (Gustavsson., 2007), al ser tan exitoso, el VISIR ha sido empleado en 6 universidades europeas. Nuevos enfoques se han centrado en la facilidad de implementar experimentos (García-Zubia, 2010) e incluso haciéndolos móviles (Y. Lyalina, 2011). La idea es que una universidad proveedora puede compartir experiencias, que se utiliza con sus alumnos, así estudiantes con dificultades, de universidades consumidoras pueden utilizarlos. El team3MIT iLabs es pionero en este campo, tiene experimentos por lotes compartidos entre diferentes universidades desde el año 2004 (J.Harward, June 2008) y experimentos interactivos desde 2008.

Con el desarrollo en los últimos años de los dispositivos móviles varios de los laboratorios remotos se han ido adaptando para permitir el acceso a través de ellos. Ejemplo de esto es, la experiencia de un Laboratorio Remoto real WebLab-Deusto (P.Orduna, 2011) donde los experimentos manejados están compuestos por dos componentes: el cliente, que corre desde un navegador Web, mientras son accedidos desde computadoras personales, celulares y PDA. El servidor que trabaja junto al hardware que va a hacer uso por el estudiante. Estrategia similar se llevó a cabo en la universidad Simón Bolívar donde los estudiantes que usan los dispositivos móviles con WiFi, tales como teléfonos inteligentes, tabletas o PDA pueden

interactuar con el hardware físico de laboratorio. A través de una interfaz de dispositivo móvil, en tiempo real, el usuario puede verificar lo que sucede en el hardware de laboratorio cuando cambian los parámetros del sistema (E. Granado, June 2013).

También se cuenta con la experiencia llevada a cabo por el proyecto piloto desarrollado por el Laboratorio de Experimentación Remota (RExLab), de la Universidad Federal de Santa Catarina y ejecutado por la Escuela de Educación Básica María García Pessi en Santa Catarina, Brasil. Donde la integración entre dispositivos móviles, entornos virtuales de aprendizaje, y experimentos de acceso remoto proporcionan a los estudiantes una nueva forma de interactuar con la disciplina de la física de una manera sencilla y agradable (J. Bento da Silva, 2014).

El PIONEER 3-AT es un robot de experimentación remota, en el cual se adaptaron dos ambientes Web de laboratorio remoto para dispositivos móviles como PDA y teléfonos inteligentes. Mediante redes de comunicación inalámbricas (WLAN). Este es otro ejemplo de la vinculación entre estas dos tecnologías.

Otro ejemplo es el sistema de laboratorios remotos y virtuales de PLC (A. Safavi, 2013) vía Smartphone le permite al usuario realizar experimentos en PLC reales. Esto le proporciona experiencias tales como: presenciar la dinámica del proceso, aplicar hipótesis, cometer errores y aprender de estos.

En todos los trabajos anteriores se muestra el uso de la tecnología celular como un medio más de aprendizaje y una herramienta que posibilita un mayor aprovechamiento de los laboratorios remotos, por lo que es posible la creación de una interfaz para la tecnología móvil.

Este trabajo se centrará en el desarrollo de una interfaz para dispositivos móviles, donde les permita a los usuarios un fácil acceso al Sistema de Laboratorios a Distancia.

Situación del problema:

Con el déficit que tienen las universidades cubanas de implementación de laboratorios reales, por el elevado precio que representa, la de implementarlos en laboratorios remotos sería una buena solución. Este recurso, le proporciona al estudiantado una forma de práctica real, que es otra de las necesidades de los escolares.

La utilización de laboratorios remotos accedidos desde dispositivos móviles sería una buena estrategia. Esto posibilita un mayor aprovechamiento de los laboratorios remotos, debido a la cada vez más estrecha relación de los estudiantes con el uso de estos dispositivos. Tecnología que cada vez va desarrollándose y haciéndose más frecuente su uso, en el mundo estudiantil.

Objetivos:**Objetivo general:**

Diseñar una interfaz de usuario para el Sistema de Laboratorios a Distancia que permita el desarrollo de actividades prácticas desde dispositivos móviles.

Objetivos específicos:

- Identificar las diferentes tecnologías de programación para dispositivos móviles.
- Implementar el acceso al Sistema de Laboratorios a Distancia a través de dispositivos móviles.
- Valorar el funcionamiento del sistema mediante el acceso a través de dispositivos móviles.

Posible impacto:

Se presenta una herramienta educativa con un alto valor formativo que permite a los estudiantes de las asignaturas de Automática la adquisición de competencias específicas en temas de diseño de reguladores en tiempo real. Los laboratorios remotos constituyen una herramienta muy útil para adquirir con menos restricciones de tiempo y horario, las habilidades de ajuste y diseño de reguladores, identificación de sistemas y otros contenidos afines a la automática, desde equipos reales. Con el empleo de la tecnología móvil para su desarrollo, el impacto, de los laboratorios remotos, hacia los alumnos es mucho mayor por la gran tendencia que existe en la actualidad del uso de estos dispositivos en la vida cotidiana. De modo que se puedan realizar desde, sus lugares de estudio, como bibliotecas, áreas de estudio, junto a otros estudiantes y se fomenta el trabajo en equipo.

Organización del informe

El presente trabajo contará con introducción, tres capítulos y las conclusiones. En los cuales: en el capítulo 1 se realizará un análisis de los laboratorios remotos con énfasis en los que son accedidos mediante dispositivos móviles. El capítulo 2 se realizará un análisis de las tecnologías de desarrollo de aplicaciones para móviles en los cuales se verán características, ventajas, desventajas, limitantes y el alcance de ellas. Se efectuará la selección de la tecnología a utilizar. En el capítulo 3 se plasmará el desarrollo de la aplicación y pruebas de desempeño.

CAPÍTULO 1. ANÁLISIS DE LOS LABORATORIOS REMOTOS

El diseño e implementación de un Sistema de Laboratorios a Distancia (SLD) conlleva un estudio detallado de una amplia muestra de laboratorios virtuales y remotos, definir las principales partes que lo componen: interacción de la plataforma con el usuario, gestión de los pedidos de prácticas y procesamiento de las mismas. Ahora se pretende lograr que este sea accedido no solo desde las computadoras de un laboratorio, sino que presente la oportunidad a los usuarios de acceder desde móviles, para hacer más amplio el aprovechamiento de este sistema.

Para el desarrollo de este ambiente se hará un análisis de laboratorios remotos que presentan y no presentan, la característica de la posibilidad de ser accedidos vía dispositivos móviles, con el fin de lograr este objetivo en la plataforma Sistema de Laboratorios a Distancia ya desarrollada. Mediante esta forma se puede lograr un mayor uso de los laboratorios en la enseñanza de asignaturas de control automático, lo que constituye una actividad relevante en la preparación de los estudiantes.

1.1 Definición de Laboratorios Remotos

Por su necesidad los laboratorios remotos representan una buena inversión para los estudiantes en el curso de las asignaturas. Estos laboratorios proveen a estos de habilidades y situaciones, las cuales le serán necesarias para un buen uso de los conocimientos. Los laboratorios deben ser planteados para proporcionar una formación efectiva de los alumnos (Cooper, 2005).

Varias son las definiciones sobre los tipos de laboratorio. Una de las más acertadas es la mostrada por el profesor Dormido de la UNED (Dormido, 2004b) que basa su clasificación en dos criterios: la forma de acceder a los recursos sobre los que se experimenta y la

naturaleza del sistema sobre el que se opera. Al tener en cuenta el primer criterio, se puede distinguir entre acceso remoto a través de una red y acceso local, que no implica la utilización de una conexión a Internet para poder operar con los componentes. En lo relativo al tipo de recurso, estaría entre los modelos simulados o trabajar con plantas reales. De la composición de estos criterios se logran cuatro clases de entornos muy diferentes, pero que incluyen todas las formas de experimentación posibles (Tabla 1.1).

Tabla 1.1 Clasificación de los entornos experimentales.

		Naturaleza del recurso	
		Real	Simulado
Acceso al Recurso	Local	Laboratorio tradicional	Laboratorio virtual monousuario
	Remoto	laboratorio remoto	Laboratorio virtual multiusuario

Acceder localmente a un recurso real, es lo que simboliza la práctica de laboratorio tradicional, donde un alumno es capaz de realizar una práctica proporcionada mediante un sistema real frente a él. Mencionar que el sistema puede estar conectado o no a una computadora.

Para acceder localmente a un recurso simulado, este se presentará mediante un software. La interfaz de experimentación trabaja sobre un sistema que se encuentra en el mismo ordenador. Es utilizado por un alumno a la vez.

Acceder remotamente a un recurso real es la posibilidad de comunicar un sistema real, mediante una red. Los usuarios operan y controlan los sistemas por medio de una interfaz del experimento, que se encuentra ejecutándose en un ordenador conectado a una red, de forma remota. Los sistemas reales pueden o no estar conectados a ordenadores.

Para acceder remotamente a un recurso simulado es igual al acceso remoto de un recurso real, la diferencia es que el equipamiento real es sustituido por un modelado, lo que enuncia que el estudiante realiza su labor en una interfaz de experimentación que trabaja sobre un

sistema virtual accesible. Esto permite la posibilidad de que trabajen varios usuarios sobre el mismo sistema virtual.

1.2 Niveles que conforman un Sistema de Laboratorios Remotos

1.2.1 Interfaz de usuario

La interfaz de usuario es una de las más importantes dentro de un sistema de laboratorios remoto, le da a conocer al usuario el entorno de trabajo, esta tiene que tener por característica la flexibilidad, ser amigable y permitir un fácil uso del sistema. Además tiene que ser realizada de forma que sea fácil de mantener y ampliar.

El objetivo primero de la interfaz de usuario es mostrar las prácticas disponibles en el sistema, atender el pedido de las prácticas con todos sus datos y enviarlo a la parte encargada de gestionarlos, además de presentar los resultados. En este nivel se encuentra la administración y gestión del sistema que, a su vez, se interrelaciona con los demás niveles.

Con las nuevas tecnologías que aparecen, la interfaz de usuario, que se basa Web principalmente, ya no solo tiene que presentar sus servicios hacia pc de escritorio o portátiles, sino que aparecen una nueva gama de dispositivos móviles como los *smartphone*, las *tablet*, que por sus características, muy variadas y principalmente el tamaño de su pantalla, presentan la necesidad de crear interfaces de usuario más acorde con sus necesidades de tamaño y espacio de estos dispositivos, lo que corresponde con el objetivo principal de este trabajo.

Selección de la tecnología para el desarrollo de la interfaz de usuario

Al tener en cuenta el estudio realizado (Garcia-Zubia et al., 2009) se eligieron seis características para comparar las diferentes tecnologías más utilizadas para el desarrollo de la interfaz de usuario. Estas características son:

- Paradigma: coincidencia con los modelos actuales de las nuevas tecnologías.
- Acceso entre plataformas: las aplicaciones se ejecutarán entre diferentes Sistemas Operativos.
- Compatibilidad con los navegadores Web: es factible para diferentes navegadores Web.
- Instalación necesaria: requiere de un software de instalación para su funcionamiento.

- Audio y vídeo: que tan bueno es la capacidad de procesar audio y video.
- Flexibilidad: desarrollo de aplicaciones en otros contextos.

1.2.2 Gestión de prácticas

En el nivel de gestión de prácticas, el sistema manda los pedidos de prácticas hechos por los usuarios mediante la interfaz, a la capa de procesamiento de prácticas para entonces devolver la respuesta de la ejecución. La gestión de prácticas funciona como enlace entre la interfaz de usuario y la capa de procesamiento de prácticas.

Por el momento se utilizan tres variantes principalmente, *MATLAB Web Server*, *LabVIEW Internet Tool-kit* y en otros trabajos se implementan mecanismos propios de comunicación ya sea con *MATLAB* o con otro software de procesamiento de prácticas.

Selección de la tecnología para la gestión de prácticas

Se realiza un análisis de las tecnologías utilizadas más comúnmente en la gestión de prácticas.

Para el análisis se tomaron las seis características siguientes:

- Acceso entre plataformas: las aplicaciones se ejecutarán entre diferentes Sistemas Operativos.
- Herramientas de desarrollo: la existencia de herramientas necesarias para el desarrollo de la aplicación.
- Velocidad de desarrollo: rapidez de desarrollo de la aplicación de gestión de prácticas.
- Robustez: durabilidad de la aplicación desarrollada.
- Flexibilidad: desarrollo de aplicaciones en otros contextos.
- Precio: costo de los software necesarios para el desarrollo de la aplicación.

1.2.3 Procesamiento de las prácticas

Este nivel realiza la ejecución de las prácticas seleccionadas por los usuarios, los datos y parámetros son definidos en la interfaz de usuario. El procesamiento de las prácticas depende del software utilizado. Las variantes más recurridas son software específico, *LabVIEW* y *MATLAB/Simulink*.

LabVIEW y *MATLAB/Simulink*, a pesar de ser software propietario, son los más utilizados (Gravier et al., 2008).

Selección de la tecnología para el procesamiento de las prácticas

Se realiza un análisis de las tecnologías para el procesamiento de prácticas. Para esto se toman en cuenta las siguientes características:

- Acceso entre plataformas: las aplicaciones se ejecutarán entre diferentes Sistemas Operativos.
- Herramientas de desarrollo: la existencia de herramientas necesarias para el desarrollo de la aplicación.
- Velocidad de desarrollo: rapidez de desarrollo de la aplicación de gestión de prácticas.
- Robustez: durabilidad de la aplicación desarrollada.
- Compatibilidad con el hardware: compatibilidad entre la tarjeta de adquisición de datos y los otros elementos.
- Precio: costo de los software necesarios para el desarrollo de la aplicación.

1.3 Herramientas de administración y gestión

Dentro de las herramientas de administración más comunes en los laboratorios remotos, es el registro de usuarios uno de ellas, la cual puede ser de dos formas: registro en línea donde el usuario introduce sus datos en un formulario, que incluye por supuesto el nombre de usuario o *login* que quiere utilizar y su contraseña, y automáticamente se le introduce en la base de datos del sistema, o fuera de línea donde un administrador o usuario privilegiado introduce los nuevos usuarios en la base de datos de forma manual.

Existen una serie de sistemas Laboratorios Virtuales y a Distancia que implementan el registro en línea para lograr una mayor utilización de estos por parte de diferentes usuarios interesados.

Un ejemplo de esta variante de registro en línea es el Laboratorio Remoto de Automática (Figura 1.1), permitiéndole a los usuarios el acceso a equipos industriales, como una planta piloto para la realización de experiencias de control de operación y supervisión remota, maquetas de procesos de control sobre variables de nivel, caudal, temperatura, y otros equipos de automatización industrial. Los usuarios registrados pueden realizar prácticas de control sobre las plantas antes mencionadas, comunicarse con otros usuarios, entre otras opciones que no posee el que entra como invitado. Los interesados tienen que llenar un

formulario y colocar una dirección de correo electrónico válida mediante la cual obtienen la clave de acceso. Es de mencionar que debido a esto el acceso al sistema no es inmediato sino solo después de recibir la contraseña vía correo electrónico.

Home > User account

User account [Create new account](#) [Log in](#) [Request new password](#)

Account information

Username: *

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address: *

A valid e-mail address. All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail.

Bienvenidos al proceso de registro de la web del Laboratorio de Automática y Control.

Si es alumno de la Universidad de León utilice como nombre de usuario (*username*) la primera parte del correo electrónico de la universidad. Por ejemplo, si su dirección es *tinaov02@unileon.es*, debería utilizar como nombre de usuario *tinaov02*.

Personal Information

Name: *

The content of this field is kept private and will not be shown publicly.

First surname: *

The content of this field is kept private and will not be shown publicly.

Second surname:

The content of this field is kept private and will not be shown publicly.

DNI: *

The content of this field is kept private and will not be shown publicly.

Figura 1.1: Formulario de registro de usuarios de Laboratorio Remoto de Automática.

El registro fuera de línea, a pesar de restringir el acceso al sistema a pocos usuarios, es la más usada, quizás con el objetivo de lograr una mayor seguridad.

Un ejemplo de esta forma de registro de usuarios es *JavaLab*: Laboratorio virtual y Remoto para prácticas de Regulación Automática, donde para poder ejecutar los laboratorios en tiempo real el usuario debe estar validado. Aunque en el sistema existe un lugar para registrarse, es solo para usuarios agregados por los administradores.

El proyecto *AutomatL@bs* solo permite el acceso a los usuarios previamente introducidos en su base de datos. Con las claves de acceso (*login* y *password*) que les han sido enviadas a los alumnos en una carta.

El registro de usuarios, es una herramienta administrativa muy útil, pero se hace más eficiente si se adiciona una personalización de acceso de los usuarios. Cada usuario tendrá un rol, y en dependencia del mismo, permiso de acceso a diferentes niveles del sitio.

El objetivo principal de los laboratorios a distancia es la docencia, cuando una institución quiere hacer una práctica sobre una planta física tiene que implementar mecanismos para evitar que usuarios externos entren, ya que estas solo pueden ser accedidas por un usuario a la vez.

Por ejemplo en *JavaLab* los usuarios, estudiantes y grupos del sistema pueden reservar horarios de laboratorio (Figura 1.2).



Figura 1.2: Ventana de Gestión de Horarios del sistema *JavaLab*.

La reserva horaria se puede hacer en el cuadro de diálogo o sobre el calendario, donde se reserva la semana en la cual se trabajará y el tiempo de trabajo en intervalos de 15 min con un máximo de 1 día. El administrador puede modificar o borrar reservas de los usuarios.

Otra herramienta administrativa usada actualmente en los Laboratorios Remotos es la Gestión de Asignaturas y Grupos. Cada asignatura que se introduzca en el laboratorio puede tener varios grupos, de ahí la necesidad de tener un control de los grupos matriculados.

Un ejemplo de ello se muestra en el laboratorio de la Universidad Politécnica de Catalunya (UPC) (Figura 1.3).

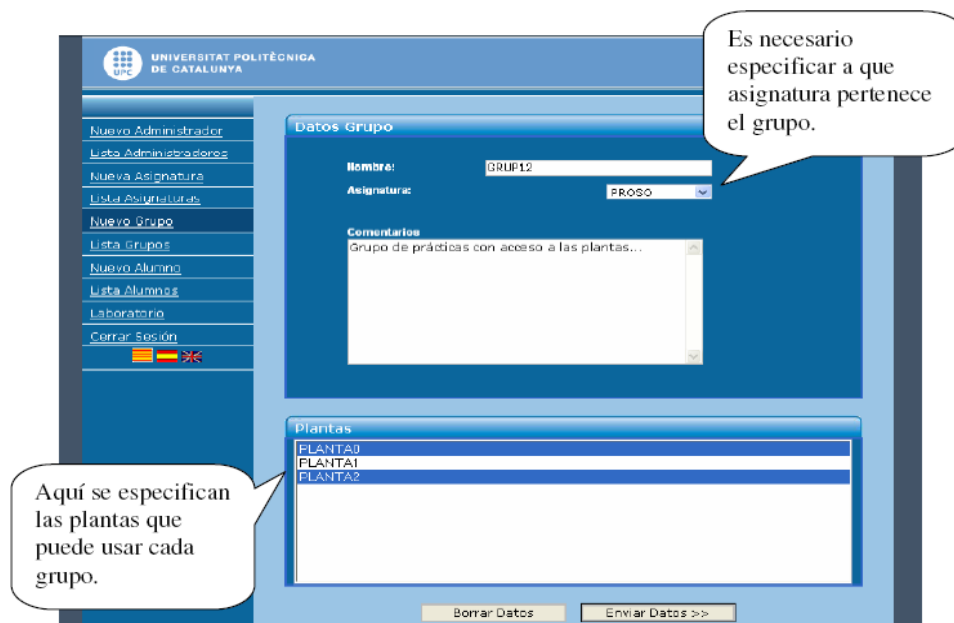


Figura 1.3: Nuevo Grupo, UPC.

Los posibles tipos de asignatura están especificados en la base de datos. Al eliminar una asignatura, se eliminan todos sus grupos y se borran las relaciones entre los grupos borrados y los alumnos. Otra herramienta es la gestión de plantas, donde se ven y modifican los estados en las que ellas se encuentran, así como introducir y configurar nuevas plantas (Figura 1.4).

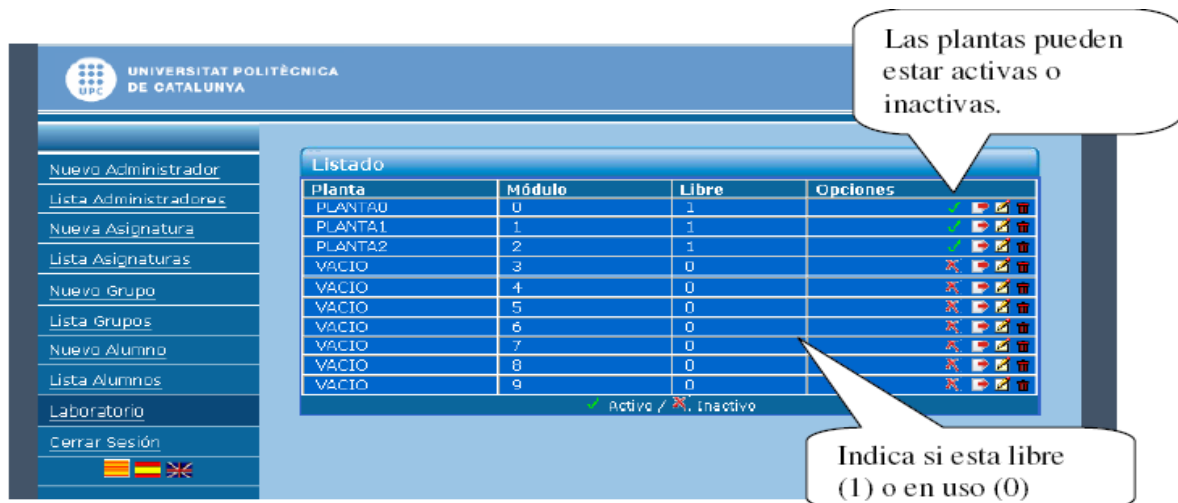


Figura 1.4: Listado de Plantas, Laboratorio UPC.

1.4 Análisis de laboratorios remotos que brindan servicio a móviles

Los cursos impartidos en las universidades tienen dos componentes mutuamente complementarios: la teoría y la práctica. Las experiencias de laboratorio constituyen una poderosa herramienta de aprendizaje ya que le permiten al estudiante tener contacto con un equipo real y poner en práctica todos los conocimientos adquiridos en la teoría. Los equipos de laboratorio, que imitan la complejidad de los procesos reales, son elementos esenciales en la enseñanza de ingeniería. En las experiencias prácticas, el estudiante tiene la oportunidad de “aprender-haciendo”.

El adecuado equipamiento de un laboratorio requiere la adquisición de equipos especializados y costosos. Generalmente, estos costos son prohibitivos para la mayoría de las instituciones, por lo que es común encontrar un número reducido de equipos los cuales deben ser compartidos entre investigadores y una gran cantidad de estudiantes que cursan diferentes carreras de ingeniería. El tiempo que disponen los estudiantes para el uso de los laboratorios es ajustado y en ocasiones insuficiente para la culminación exitosa de la experiencia práctica. Esto trae como consecuencia una disminución en la calidad de la enseñanza universitaria.

De ahí la gran necesidad de su aprovechamiento, principalmente en el uso de laboratorios remotos para la educación, se han hecho estudios por ejemplo, sobre la efectividad del uso de laboratorios virtuales y remotos en la educación y sugieren su introducción de una forma gradual en los estudiantes (Nickerson et al., 2007). Otro ejemplo es que consideran que para los cursos a distancia de ingeniería, el acceso remoto constituye el único método realista de llevar a cabo muchos experimentos. Expresan que se debe permitir el acceso remoto a los estudiantes, y no debe estar limitada la conexión por tiempo o por consideraciones geográficas, para completar las tareas de laboratorio. Por lo que los autores concluyen que las prácticas remotas permiten facilitar el desarrollo de habilidades en el uso de los sistemas reales y la instrumentación (Callaghan et al., 2008). Son varios los ejemplos de laboratorios virtuales y remotos que se muestran en la actualidad.

El m-laboratorio (E. Granado, June 2013) se implementó con el propósito de poner en práctica todo el conocimiento que el estudiante ha adquirido en las clases de teoría relacionados con diferentes técnicas de diseño de controladores. El objetivo principal del m-laboratorio es incorporar el uso de los dispositivos móviles de comunicación para apoyar el proceso de

enseñanza/aprendizaje en la Universidad. El m-laboratorio tiene la típica arquitectura cliente/servidor que se ilustra en la figura 1.5.



Figura 1.5. Arquitectura del sistema.

El servidor, es un computador estándar que está conectado a una planta experimental a través de un sistema de adquisición de datos (DAQ), también posee un adaptador *WiFi* para comunicarse con el dispositivo móvil. La composición del m laboratorio está formada por aplicaciones que han sido desarrolladas con LabVIEW 8.2. En la figura 1.6 se muestran las distintas acciones que realizan cada una de las aplicaciones que forman el sistema.

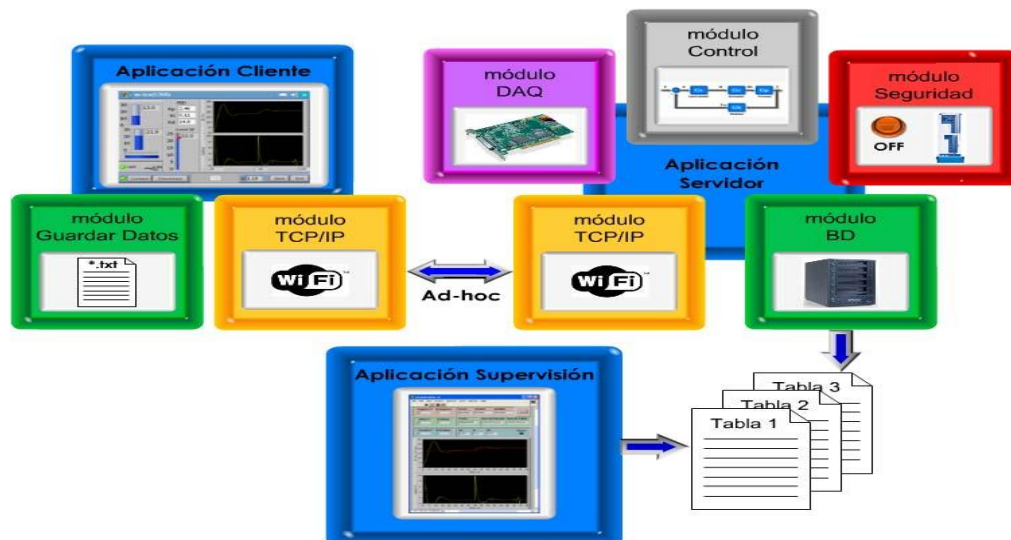


Figura 1.6. Componentes y tareas del sistema.

El servidor realiza cinco tareas: el módulo TCP/IP, establece la comunicación entre el cliente y el procesador, el módulo DAQ comunica la planta a través del sistema de adquisición de datos.

Estas tareas se realizan a un período constante de 50 ms. La tarea BD se encarga de la gestión de la base de datos (registros de usuario, resultados, operaciones). El módulo de control representa a un controlador que es un PID. Mediante el módulo de seguridad se previene el uso incorrecto de los equipos de laboratorio por parte del usuario.

La aplicación del cliente está desarrollada mediante el módulo de desarrollo de LabVIEW mobile 8.2. El cliente realiza dos tareas: TCP/IP (gestiona la comunicación entre móvil y servidor) y guardar datos (archiva los resultados).

La interfaz de usuario se muestra en la figura 1.7 muestra la interfaz de usuario del m-laboratorio. El objetivo es mostrar la información de una manera aceptable para el tamaño pequeño de la pantalla del dispositivo móvil.

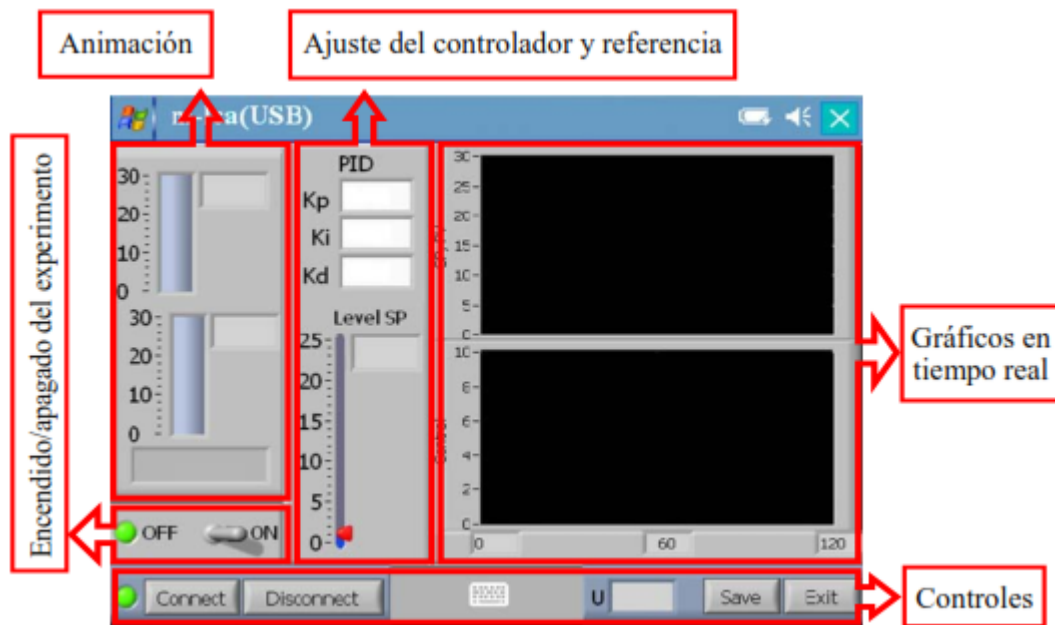


Figura 1.7. Interfaz de usuario.

En la parte derecha de la pantalla del dispositivo móvil está el área de graficación. En esta hay una ventana donde se puede visualizar en tiempo real y sobre un mismo gráfico, el comportamiento de la señal de salida o variable de proceso y la señal de referencia. En otro graficador, se muestra la correspondiente señal de control. En la parte central de la interfaz

se encuentran los controles principales que le permiten al usuario interactuar con el sistema. Tanto los iconos que representan el sistema como las cajas que reflejan numéricamente el valor de las variables del proceso, se actualizan en tiempo real según los valores obtenidos del equipo.

La aplicación de supervisión le permite revisar al profesor las actividades del estudiante desde cualquier ordenador (Figura 1.8).



Figura 1.8. Aplicación de supervisión de la actividad del estudiante.

La planta seleccionada para la implementación del m-laboratorio consiste en el sistema de tanques interconectados de la compañía *Quanser* que se ilustra en la figura 1.9.



Figura 1.9. Sistema de tanques.

El PIONEER 3-AT es un robot de experimentación remota, en el cual se adaptaron dos ambientes Web de laboratorio remoto para dispositivos móviles como PDA y teléfonos inteligentes. Mediante redes de comunicación inalámbricas (WLAN). La necesidad estuvo dada en el hecho de que la resolución mínima de los monitores estándar de PC, VGA es de 640x480 pixel, mientras que la de los PDA es solamente de 320x240 pixel y la resolución típica de los teléfonos inteligentes es de 176x220 pixel todavía peor. Normalmente los buscadores Web de los móviles (*Pocket IE, Opera, Netfront y Minimo*) ofrecen la posibilidad de adoptar los sitios a la vista del móvil (escalar imágenes y textos, evitar el desplazamiento horizontal y el modo pantalla completa). No obstante la optimización de interfaces móviles para laboratorios remotos es requerida, con una comprobación de la aplicación desarrollada. Estos ambientes Web adoptados para este experimento son:

- Una plataforma de robot móvil omnidireccional: la tarea de los estudiantes es adaptar los parámetros fuera del control del motor subyacente. La AWT GUI interfaz de experimentación basada en entorno PC puede ser reutilizado en PDA y teléfonos inteligentes. El robot móvil se mueve por caminos pre-planificados (Figura 1.10).
- Una Web controladora del robot móvil *Pioneer 3 AT*: una aplicación de control remoto para el robot que demuestran las características del robot (Figura 1.11).



Figura 1.10: GUI experimental del robot móvil para Smartphone (640x480 pixel)



Figura 1.11: Web y Java AWT basada en una GUI omnidireccional

El PIONEER 3-AT está equipado con varios tipos de sensores como ultrasónico, escáner laser, sistema de cámara CCD junto con una interfaz de programación (*Aria* y *Saphira-API*) para el comportamiento autónomo. El movimiento es controlado por un microcontrolador empotrado y su cliente PC a bordo corre bajo un sistema operativo *Linux*. Además posee un desarrollador de cliente/servidor (*Java application/Java applet*) como base de solución para el control remoto del robot. La aplicación de servidor de control remoto es de origen *Java* la cual usa el *Aria-API* (una librería binaria de *Linux* y código fuente *C++*).

El RexLab (*Remote Experimentation Lab*) de la Universidad Federal de Santa Catarina (UFSC) (J.Bento da Silva, 2014) creó con el objetivo de estimular a los estudiantes de Educación Elemental en la profesión de ingeniería e investigación científico tecnológica, un proyecto que da acceso a contenido educacional desde dispositivos móviles (experimentos remotos). El experimento remoto le permite al usuario la realización de estos experimentos mediante un simple navegador Web y una conexión inalámbrica a internet. El sitio fue desarrollado en *PHP* y utiliza *JavaScript* para la interface. La estructura del sitio esta presentada en la figura 1.12.

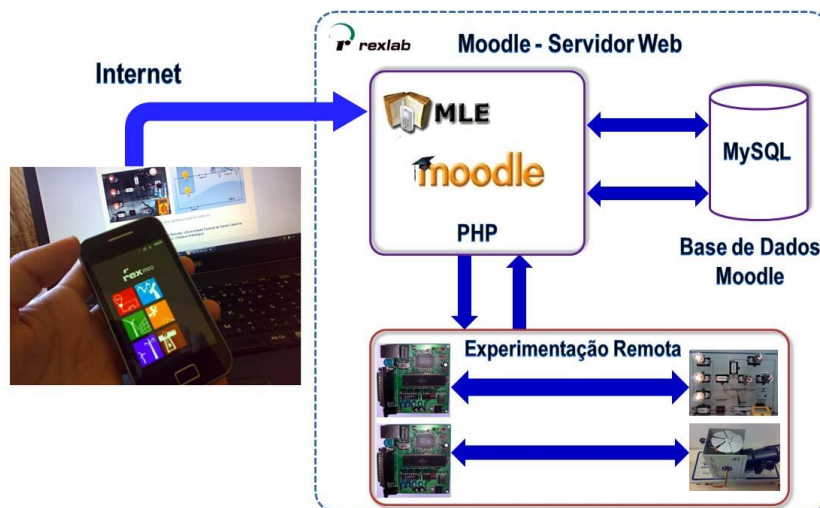


Fig. 1.12. Arquitectura propuesta.

El hardware utilizado es un micro servidor WEB (MSW) desarrollado en *RexLab*, descrito como un *standalone device*, con la capacidad de conectar dispositivos eléctricos con una red Ethernet. Utiliza el protocolo de comunicación TCP/IP y presenta funcionalidades como la extensión y adición de aplicaciones que dan la posibilidad de monitoreo y/o control de

dispositivos. Permite el uso de *JavaScript* y *Java applets*, con *HTML*, rutinas *CGI* e imágenes.

El acceso vía celular recae en la aplicación *RedMobile* (Figura 1.13), esta usa lenguaje *HTML* (*Hypertext Markup Language* versión 5) integrado con *CSS 3* (*Cascading Style Sheets*, versión 3) que es una variante de *XML* (*Extensible Markup Language*), estos son integrados por el *jQuery Mobile framework*, que emplea un sistema unificado adecuado para aplicaciones Web en dispositivos móviles y la aplicación de alto nivel de *JavaScript* que genera códigos compatibles con *iOS*, *Android*, *Windows Phone*, *Symbian*, *BlackBerry* y otros sistemas operativos móviles más importantes.



Fig. 1.13. Presentación de la aplicación *RexMobile*.

El sistema de laboratorios remotos y virtuales de PLC (A. Safavi, 2013) vía Smartphone le permite al usuario realizar experimentos en PLC reales. Esto le proporciona experiencias tales como: presenciar la dinámica del proceso, aplicar hipótesis, cometer errores y aprender de estos, para esto es necesario desarrollar una herramienta *HMI* (*Human Machine Interface*) en los celulares para los dos lenguajes de PLC simbólicos (i.e. *Ladder diagrams*) y lenguaje base de texto simple (i.e. *STL*). El método propuesto está basado en *Android* y programación *Java* por la parte de los celulares y el software *LabVIEW* en el lado de las computadoras. La estructura presentada está en el la siguiente figura 1.14.

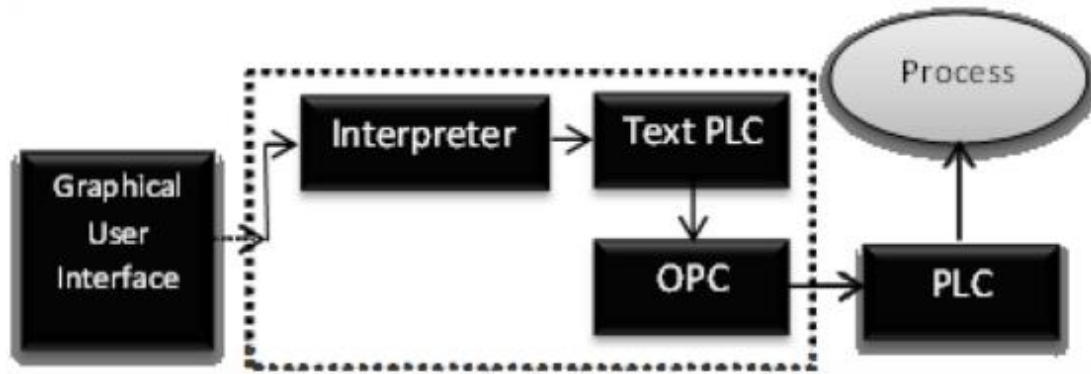


Fig. 1.14. Interfaz gráfica para conectar al PLC.

Se crea una interfaz gráfica de usuario para la programación y el aprendizaje en una plataforma móvil (Figura 1.15). En el entorno de LabVIEW (instalado en el ordenador conectado al PLC), una solicitud de interpretación de los datos obtenidos de la interfaz y su transformación a una forma adecuada para crear la aplicación *TEXTPLC*. *TEXTPLC* es una aplicación creada en LabVIEW que procesa la información recibida en una forma STL y calcula la salida adecuada para el servidor NI OPC. El servidor OPC se aplica a la salida del PLC. En esta investigación un *DELTA 2DVP20SX* se ha puesto en uso. Este proceso de control se ha aplicado en el laboratorio de *PLC* de la Universidad de Shiraz.

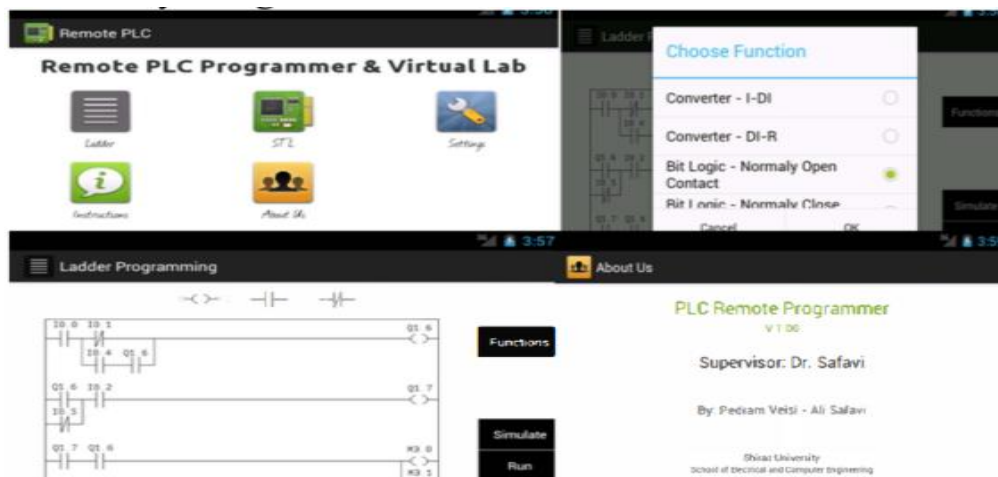


Fig. 1.15. Presentaciones de la aplicación Androide.

El *WebLab-DEUSTO* (P. Orduña, October 2011) es una plataforma de laboratorio remoto de fuente abierta diseñado e implementado por la Facultad de Ingeniería de la Universidad de Deusto. Desde el punto de vista tecnológico el laboratorio ha sido implementado en su última

versión como un *Web Services*¹ a través de *SOAP*², *AJAX*³ y *Python*⁴ (Garcia-Zubia et al., 2009).

El desarrollo de los experimentos puede ser de dos formas diferentes. Experimentos manejados que están compuestos por dos componentes: el cliente, que corre desde un navegador Web, mientras son accedidos desde computadoras personales, celulares y PDA. El servidor que trabaja junto al hardware que va a hacer usado por el estudiante. La otra forma es los experimentos no manejados donde las comunicaciones no son manejadas directamente por la plataforma, mientras que la autenticación, autorización y seguimiento sí lo son. La arquitectura de estos experimentos se puede ver en la figura 1.16.

Se le han agregados varios rasgos nuevos a la hora de realizar los experimentos como una *Database*, *LDAP* (*Lightweight Directory Access Protocol*), *OpenID* para compartir los experimentos con otras universidades.

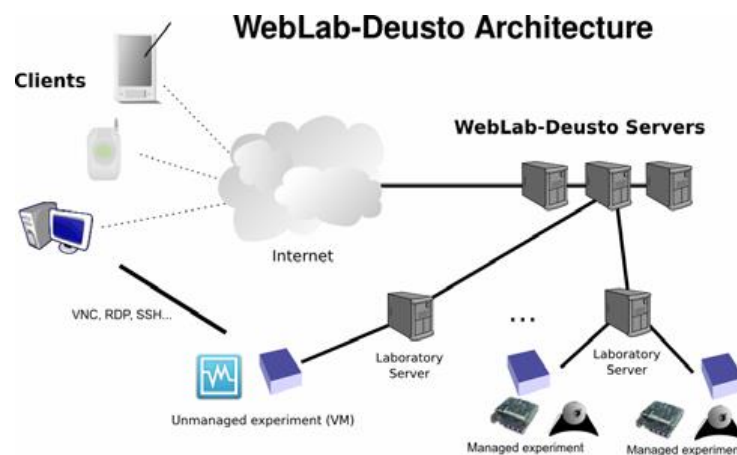


Figure 1.16. Experimentos manejados y no manejados en la plataforma *WebLab-Deusto*.

Las universidades de Tiaret y Mostaganam han realizado estudios (A. Adda Benattia, 2014) sobre los laboratorios reales y la importancia del uso de la tecnología móvil para su

¹ *Web service* es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

² *SOAP* (siglas de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

³ *AJAX*, acrónimo de *Asynchronous JavaScript And XML*, es una técnica de desarrollo Web para crear aplicaciones interactivas.

⁴ *Python* es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

realización por el amplio desarrollo que esta ha alcanzado en la última década. Donde se propuso la creación de un sistema de laboratorio remoto donde se permite que el cliente acceda al servidor con el fin de realizar diversas manipulaciones tales como: el control de un generador de señal, el ajuste de una pantalla de osciloscopio, cambiar una configuración de circuito, la realización de una medición, o cualquier otra operación puesta a disposición por la interfaz.

La tarjeta de conmutación le permite al sistema hacer múltiples configuraciones de circuito PW de acuerdo a los instrumentos usados. La arquitectura del software ha sido escrita en *HTML5, Java, JavaScript, PHP y Android*. Además la aplicación androide TG2511-GBF es usada para controlar el generador de señal. La arquitectura presentada se muestra en la siguiente figura 1.17.

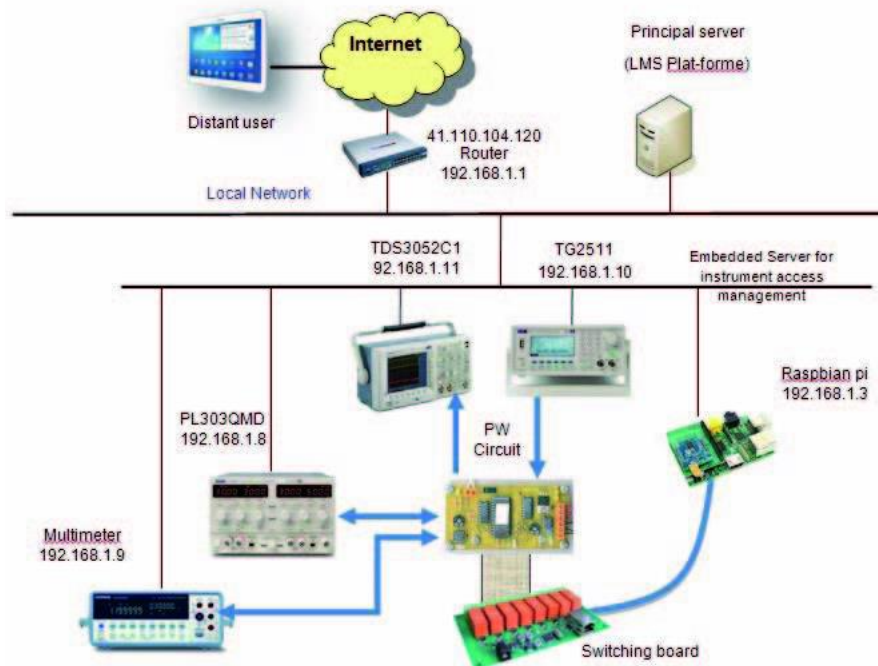


Fig.1.17. Arquitectura de hardware.

1.5 Apreciaciones sobre el capítulo

El uso de los laboratorios a distancia está bien difundido en la actualidad en Internet. En el área del control automático también están presentes estas potentes y versátiles herramientas principalmente debido a la necesidad de compartir recursos costosos como son robots, motores, y de poder hacer un uso efectivo de los mismos.

Con la aparición de las tecnologías móviles, que en la última década se han desarrollado ampliamente, aparece una forma de utilizar las ventajas que nos brindan estos. Para conseguir un uso más efectivo de los laboratorios remotos, se propone la realización de una nueva interfaz de usuario, la cual les permita a usuarios acceder desde sus teléfonos celulares, sobre la base del ya creado con anterioridad. Esto le brindará al estudiante que lo desee, la posibilidad de realizar la actividad práctica en el horario de mejor provecho para él. A su vez, esto posibilita reducir el número de estudiantes que realizan sus ensayos de manera presencial y que dispongan de mayor tiempo para finalizar sus experimentos. También el acceso a los laboratorios remotos desde cualquier área de la universidad como bibliotecas, zona de estudio.

CAPÍTULO 2. SELECCIÓN DEL FRAMEWORK

La palabra inglesa "**framework**" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En este capítulo se abordarán los principales *frameworks* de desarrollo de aplicaciones Web actuales, a los cuales se le hará una determinación de cuál es la mejor solución para el desarrollo de la interfaz del SLD, basándonos en las características, alcance, ventajas, desventajas, limitaciones que estos presentan. Se darán a conocer los conceptos de los principales términos de los cuales se abordarán en el capítulo, así como consejos para el diseño de la interfaz Web.

2.1 Marco Conceptual

Framework:

El término *framework*, hace referencia a una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones, en la actualidad existen frameworks para lenguajes como *html5*, *css* y *JavaScript*.

Framework Web:

Un *framework Web*, se puede definir como un conjunto de componentes (por ejemplo clases en *java* y descriptores y archivos de configuración en *XML*) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

HTML:

HTML es una implementación del *standard SGML (Standard Generalized Markup Language)*, estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

Es un metalenguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones.

HTML5:

Es una colección de estándares para el diseño y desarrollo de páginas Web. Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella.

HTML 5 permite una mayor interacción de las páginas Web con contenidos de reproducción digital (Videos, audio, juegos, entre otros) y facilidad a la hora de maquetar una página Web.

CSS:

Hojas de Estilo en Cascada, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos *HTML* y *XML*, que separa el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos *HTML* y *XML*. *CSS* permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la *CSS* afectará a todas las páginas vinculadas a esa *CSS* en las que aparezca ese elemento.

JavaScript:

JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento *HTML*.

Hasta entonces ya se usaba *HTML* y *JAVA*, pero la aparición del *JavaScript* produjo una importante revolución, ya que dio al usuario la posibilidad de crear aplicaciones *on-line*.

Aplicación Web:

En la Ingeniería de software se denomina aplicación Web a aquellas aplicaciones que los usuarios pueden utilizar al acceder a un Servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador.

Las aplicaciones Web son populares debido a lo práctico del navegador Web, a la independencia del Sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software a miles de usuarios potenciales.

2.2 Web App o Native App

En el mercado de las aplicaciones móviles se cuenta con dos grupos claramente diferenciados: Las *Web Apps* y las aplicaciones nativas, o lo que es lo mismo, las que necesitan ser instaladas previamente en nuestros dispositivos. No está claro cuál de ellas es mejor, esto depende del campo en el que nos centremos, para unos serán mejores las *Web Apps* y para otros las nativas. Basándose en la estrategia y necesidades, debemos decidir, ya que para algunos campos son mejores las *Web Apps* y para otros las *nativas*.

Antes de decidir por una de ellas, debemos tener muy en cuenta tres aspectos fundamentales:

- No vamos a tener conexión siempre: Este punto es muy sencillo de explicar. Si es necesario utilizar o mostrar ciertos productos sin la necesidad de estar conectados a la red, la elección es clara, una aplicación nativa. Estas, aunque necesitan una instalación previa en nuestro dispositivo, no necesitan disponer de conexión a internet posteriormente cada vez que las usemos.

- Coste: el objetivo es que la aplicación Web sea lo más económica posible. La elección es compleja, ya que depende de los recursos que se necesitan, puede ser más rentable una aplicación nativa, aunque por lo general no suele ocurrir. Por otra parte, si tenemos en cuenta el tiempo empleado para la creación de una u otra, claramente es mejor una *Web App*, ya que un experto desarrollador puede hacer que en muy pocos días dispongas de dicha aplicación (el tiempo de creación de una nativa puede llegar a ser de varios meses).
- Cliente: El conocimiento del tipo de cliente al que se dirige es imprescindible, ya que este puede poseer distintos dispositivos.

Para mantener esa forma de pensar, serán mejores las *Web Apps*, ya que una aplicación nativa no se comporta de la misma manera en un sistema operativo que en otro (*Android*, *iPhone*).

Una vez tenidos en cuenta estos apartados, vamos a detallar brevemente las ventajas e inconvenientes de las *Web Apps*, al tratarse de nuestra elección final.

Ventajas

- Una vez cargada la aplicación, todos los datos están a disposición del cliente.
- El cliente siempre tiene acceso a la última versión de la *Web App*. Siempre que la cargue dispondrá de la última versión de la misma.
- Compatibilidad con todos los sistemas operativos.
- No ocupan espacio en el disco duro del cliente.
- Son compatibles con todos los sistemas operativos.
- Mayor funcionalidad con la continua actualización de los exploradores.
- Son móviles, podemos tener acceso a ellas desde cualquier dispositivo conectado a la red.

Inconvenientes

- Se necesita disponer de conexión a la red en todo momento, para acceder a ellas y ejecutarlas. Si no estamos conectados o si se interrumpe la conexión al ejecutarlas, no podremos utilizar la *Web App*.
- El usuario solo dispone de la última versión, no puede elegir entre cada una de las versiones usadas anteriormente.

- En el momento en que el desarrollador decida hacer desaparecer la aplicación, el cliente dejará automáticamente de poder usarla. En el caso de las nativas, el cliente siempre podrá usarla mientras la tenga instalada en su dispositivo.
- A pesar del gran avance que ha supuesto *HTML5* para la funcionalidad de las *Web Apps*, las aplicaciones nativas están un paso por delante al no verse limitadas por el navegador.

Las App Nativas tienen como ventajas que las interfaces son más intuitivas, con un mejor rendimiento y mayor ciclo de vida de la aplicación. Tienen presencia en la *AppStore* y acceso a *iAd* (y otros proveedores de *Ads*). Tiene como inconvenientes una implementación por cada plataforma, necesidad de personal más cualificado y una implementación más costosa.

2.3 Frameworks de lenguaje *HTML5*

En la actualidad existen gran variedad de *frameworks HTML5*, algunos con enfoque a diseñadores y otros a programadores, se seleccionaron los *frameworks* más utilizados con el fin de evaluar las características, ventajas, desventajas, alcances y limitaciones. Basados en esta información se comparan los *frameworks*, con el objetivo de seleccionar el que más ventajas aporte a los usuarios y se estructura un ejemplo básico de su funcionamiento.

Para seleccionar los *frameworks* se tomó como referencia la tabla 2.1

Tabla 2.1: Comparativa de *Frameworks HTML5*. Tomado de (Gube, 2013)

Frameworks	Licencia	Tamaño (MB)	Contribuidores	Inicio
Twitter Bootstrap	Apache License v2.0	3.15	235	2011
Foundation 4	MIT License	2.20	237	2011
HTML5 Boilerplate	Múltiples licencias de open source	0.09	157	2010

La *frameworks* seleccionados para este estudio, al tener en cuenta su facilidad de manejo y su popularidad de uso son: Twitter Bootstrap y Foundation 4.

2.3.1 Twitter Bootstrap

Twitter Bootstrap es una colección de herramientas de software libre para la creación de sitios y aplicaciones Web. Contiene plantillas de diseño basadas en: *HTML* y *CSS* con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de *JavaScript*, además permite la adaptación de la interfaz, en dependencia del tamaño del dispositivo en el que se visualice sin que el usuario tenga que hacer nada, esto se denomina *Responsive Web Design*.

El *framework* trae varios elementos con estilos predefinidos fáciles de configurar e integración *jQuery* para ofrecer ventanas y *tooltips* dinámicos. Es el proyecto más popular en *GitHub*(Fontela, 2013).

Las características de Twitter Bootstrap son: ofrece *grids* fijados y líquidos 724px, 940px, 1170px, un sistema de parrilla de 12 columnas para maquetar la *Web app*, se gestiona a través de *GitHub*, es compatible con *Chrome*, *Firefox*, *Safari*, *IE 7* y *Opera* en *tablet* y teléfonos inteligentes, usa *HTML5* y *CSS3*. Además de integrar librerías *JavaScript* y *Less*. Es *open source*, su descarga ocupa pocos *Kilobytes* (Templatemonster, 2013).

El alcance es que este *framework* se puede utilizar en cualquier tipo de aplicación pero se recomienda en estos tipos:

- Diseño de Panel de Control (*Dashboard*) de una aplicación Web: Básicamente el *Dashboard* es el punto de entrada a la administración de una aplicación Web, generalmente de acceso restringido.
- Diseño de Aplicaciones Móviles: Twitter Bootstrap se apoya en *HTML5* y *CSS3*, que son soportados por los dispositivos móviles (Patricio, 2014).

Las ventajas de Twitter Bootstrap son: utiliza componentes y servicios creados por la comunidad por ejemplo, *HTML5 shim*, *Normalize.css*, *OOCSS*, *jQuery UI* (implementa 12 *plugins*), *LESS*, *GitHub*. Es un compendio de buenas prácticas porque implementa los nuevos estándares *HTML5 + CSS3*, *cross-browser*, *IE 7/8/9*, *Firefox*, *Chrome*, *Safari* y *Opera 11*, *Grid system*, 12 columnas por defecto, fijas (px) o fluidas (%) a la hora de la maquetación. También aprovecha la comunidad *GitHub* de *Twitter Bootstrap*. Es una herramienta ágil para construir interfaces con un tema por defecto (Echegaray, 2014).

Las desventajas que presenta es que aunque la curva de aprendizaje es liviana, se debe familiarizar con su estructura y nomenclatura. El diseño gráfico debe estar adaptado a las 12 columnas. Por defecto ya tiene anchos, márgenes y altos de línea. Es complicado cambiar de versión al realizarse modificaciones profundas sobre el core. A la necesidad de añadir componentes que no existan, se debe personalizar para que mantenga la coherencia con el diseño (Echegaray, 2014).

Las limitaciones de Twitter Bootstrap son:

- Framework orientado a desarrolladores.
- Se debe tener conocimientos en CSS para utilizarlo, el no tenerlos generaría que todas las páginas se vean igual y problemas al intentar modificarlas (Chávez, 2014).

2.3.2 Foundation 4

Es un *framework front-end* que consta de herramientas útiles para la toma de respuesta, primero los sitios Web móviles, construido con *HTML*, *CSS* y *jQuery*, utiliza tecnologías y prácticas modernas.

Foundation 4 cuenta con clases de marcador de posición y un nuevo marcado semántico, la principal diferencia con la última versión es un enfoque en ser móvil primero, lo que significa que los contenidos móviles se priorizan sobre contenido habitual, permite a los usuarios móviles tener tiempos de carga tan cortos como sea posible (Serrano, 2013).

Las características Foundation 4 son: tiene *grids* fijados en varias plataformas, ofrece 1-16 columnas con personalizador, tiene un conjunto de herramientas de interfaz de usuario que no se adhieren a cualquier estilo de conjunto. También se gestiona a través de *GitHub*, compatible con navegadores *Chrome*, *Firefox*, *IE* +, *Safari* con *OS*, *iPad*, *Android* 2 al 4, *tablet* y teléfono y *Windows Phone* 7 + (Templatemonster, 2013). Es *open source*.

El alcance que tiene el *framework*, es que este se puede utilizar para aplicaciones móviles, Foundation 4 proporciona tiempos de carga mínimos para aquellos usuarios que naveguen desde sus móviles.

Las ventajas que permite son: la compatibilidad con cualquier dispositivo y ofrece ordenamiento, *nesting*, *offsets* con soporte de presentación de dispositivo para facilitar la creación de diseños. Función de navegación que permite crear el estilo de la barra y con la

navegación desplegable de 3 niveles para barras simples, barras laterales y soporte de *subnav pills*. Da soporte de *UI Elements* para incluir elementos útiles tales como información sobre herramientas, etiquetas, clases de visibilidad para que se pueda ocultar o mostrar las cosas de acuerdo con la orientación de la pantalla o el tamaño, alertas, paneles y muchos más (Templatemonster, 2013).

Las Desventajas de Foundation 4 es que no se cuenta con una comunidad oficial y es orientado a aplicaciones móviles. La limitación es que se debe tener conocimiento en los estándares de aplicaciones móviles.

2.4 Frameworks del lenguaje CSS

Debido a que las estructuras CSS son complicadas surgen los *frameworks* que ayudan a facilitar el desarrollo de las aplicaciones.

En la actualidad existen gran variedad de *frameworks* de CSS. Se seleccionaron los *frameworks* más utilizados con el fin de evaluar las características, ventajas, desventajas, alcances y limitaciones.

Los *frameworks* seleccionados para este estudio al tener en cuenta los artículos encontrados son: 960 Grid System y Blueprint CSS.

2.4.1 960 Grid System

960 Grid es un *framework* de CSS que se creó por la necesidad de tener un “estándar” en el ancho de los sitios Web, el cual facilita la maquetación a un estándar de 960px de ancho, tiene configuraciones de 12 y 16 columnas para poder combinar entre sí y así crear *layout* de una forma simple y rápida (Bernardino, 2014).

Presenta como características un diseño basado en retícula, utiliza columnas de tamaño fijo para establecer una retícula que sirva de base en la creación de los sitios Web. Puede ser una combinación de 12 o 16 columnas. Compatibilidad, se creó desde el inicio para que fuera compatible con los diferentes navegadores de Internet, utiliza etiquetas estándar de *HTML* y *CSS*. Administración independiente de las tipografías, mediante un archivo externo llamado *text.css*. Tipografía definida en pixeles, facilita el diseño ya que se pueden manejar tamaños absolutos en el diseño.

Estructura básica sin diseño (Smith, 2014). Aplicaciones Web de estructura orientada a columnas (Rodríguez, 2014).

Las ventajas de 960 Grid System es una estructura previa, el sistema ofrece una estructura para la diagramación de los sitios Web. Experiencia visual, ofrece una retícula que mejora el balance, la alineación y el espacio para lograr una mejor experiencia visual. Reemplaza el uso de tablas. Compatibilidad. El 960 Grid System es compatible con la mayoría de los navegadores más utilizados como el *Mozilla Firefox*, *Google Chrome*, *Safari*, *Internet Explorer* y *Opera*. Mayor velocidad de desarrollo. El uso de un sistema como el 960 Grid System agiliza el proceso de creación de sitios Web (Sánchez, 2014).

Las desventajas que presenta es una estructura rígida, el sistema ofrece una retícula rígida para la diagramación de los sitios Web que no permite variantes para lograr otras formas de mostrar el contenido. Limita la creatividad, el uso continuo de una retícula, limita la creatividad en el diseño y la diagramación de sitios Web. Nuevas dimensiones de pantalla, funciona muy bien con resoluciones de hasta 1024 píxeles; sin embargo, cada vez las pantallas permiten mayores resoluciones y el sistema no se adapta satisfactoriamente a resoluciones superiores. Difícil implementación (Sánchez, 2014).

Tiene como limitación lo difícil que es de implementar por cualquier diseñador Web, se deben tener conocimientos de *HTML* y *CSS* (Sánchez, 2014).

2.4.2 Blueprint CSS

Blueprint es un marco *CSS* que sirve de apoyo en el diseño y maquetación de Webs 2.0, que apunta a reducir tiempo de desarrollo, lo que ofrece una base sólida para construcción de un proyecto, con una tabla fácil de usar, tipografía sensible, *plugins* útiles y una hoja de estilo para la impresión (Montoya, 2014).

Las características de Blueprint CSS son: una rejilla que permite crear cualquier estructura de página Web, se trata de un conjunto de clases *CSS* para posicionar cualquier elemento en un espacio de 950 píxeles de anchura, dividido en 24 columnas. Brinda una definición de tipografía predeterminada, de manera precisa para todos los elementos de la página que pueden tener texto. Estilos para formularios, con los que mejorar las interfaces de usuario. Estilos para impresión, con las definiciones *CSS* más útiles para que las páginas se lean bien

al imprimirse en papel. Estilos específicos para IE, con los que resolver algunos de los problemas más comunes de Internet Explorer (Álvarez, 2013).

El alcance de Blueprint CSS es que cualquier aplicación se puede desarrollar al utilizar el *framework*.

Las Ventajas aportados por el *framework* son: la curva de aprendizaje es corta, muy robusto, se integra bien con otros *frameworks*, su nivel de *Cross-browser* es del 99%, no es restrictivo. Cuenta con una hoja de estilos para impresión, clases fáciles de recordar. Puede crear tu propia rejilla, cuenta con una gran comunidad (Bayona., 2014).

Las desventajas de Blueprint CSS es que en un principio se puede percibir como un sistema rígido. Tiene un énfasis en la estructura por encima del contenido de los objetos (Ganarce, 2014). La limitación es que hay que tener conocimientos previos en CSS.

2.5 Frameworks del lenguaje JavaScript

Con la llegada del Web 2.0 y lo que se conoce como *AJAX*, *JavaScript* está presente en los navegadores, los *frameworks* para JavaScript han permitido que el desarrollo sea más dinámico pues basta con llamar un conjunto de elementos objetos como controles, *widgets*, entre otros para crear una aplicación Web del tipo *RIA* (*Rich Internet applications*).

En la actualidad existen gran variedad de *frameworks* de *JavaScript*, se seleccionaron los *frameworks* más utilizados con el fin de evaluar las características, ventajas, desventajas, alcances y limitaciones. Estos son: el Backbone.js, el Ember.js y el Angular.js.

2.5.1 Backbone.js

Backbone.js es un *framework* que permite al desarrollador estructurar la aplicación, al delegar ciertas labores a elementos específicos con el fin de hacer aplicaciones escalables, robustas y que brinden al usuario una gran experiencia de uso (Baquero, 2014).

Qué características tiene: permite la programación, mientras atiende al paradigma MVC (Modelo Vista Controlador). Ayuda a crear estructuras bien definidas para los datos de la aplicación y facilita la creación de eventos al cambiar los datos. Simplifica el uso de vistas, ayuda a pintar interfaces de usuario en la página. Permite el uso de colecciones de modelos, en los que se cuenta con acceso a diferentes operaciones sobre los mismos. Simplifica y

ayuda a ordenar el código de las peticiones *AJAX* para realizar solicitudes al servidor (Chavez, 2013).

El alcance es: las aplicaciones que se pueden crear al utilizar Backbone.js

- Para aplicaciones Web de lado cliente y de una sola página cuya complejidad exija el apoyo de una estructura definida para un proceso de desarrollo ordenado (Dimaggio., 2014).
- Aplicación Web del tipo *RIA* (*Rich Internet applications*).

Las ventajas que tiene el *framework* es que la comunidad es fuerte y crece, además de una comunidad *Github* (Roldán and 2013). Carga la funcionalidad por defecto, la librería se encarga de todo lo que hay por debajo (Thode, 2012). No hay necesidad de aprender una sintaxis nueva, si se tiene experiencia con *jQuery* y *Underscore.js*. Se gana control ya que se puede programar y cambiar el comportamiento de prácticamente cualquier (Nitsnets, 2014b). La desventaja que permite es que delega la responsabilidad al programador a la hora de estructurar el código, con *app* de tamaño mediano/grande, se debe tener cuidado y refactorizar frecuentemente el código (Nitsnets, 2014b).

La limitación de Backbone.js es que la curva de aprendizaje es lenta, si no hay conocimientos previos en *jQuery*.

2.5.2 Ember.js

Ember.js es un *framework JavaScript* para crear aplicaciones Web del lado del cliente. Está basado en la arquitectura modelo-vista-controlador (MVC).

Está catalogado como unos de los principales *framework* en el mundo de *JavaScript* ya que permite a los desarrolladores crear aplicaciones de una sola página (*single-page*) escalables (Picca, 2014).

Las características del Ember.js es que representa un cambio de paradigma con respecto a otros frameworks ya que para funcionar correctamente exige que se respeten totalmente las convenciones de nomenclatura para nombrar los objetos de la *App*. El framework tiene la posibilidad de definir rutas anidadas, que son rutas que comparten un mismo recurso. Una única página puede tener varios controladores activos a la vez. Ember.js está muy integrado

con el motor de plantillas *Handlebars*. Tiene una capa de persistencia datos llamada *Ember Data* (Nitsnets, 2013).

El alcance es que permite desarrollar aplicaciones del tipo *SPA (Aplicaciones Single Page)* (Suarez, 2014).

Cuenta con ventajas como: convención sobre configuración, estandariza el desarrollo de aplicaciones del lado del cliente, utiliza reglas muy estrictas para el desarrollo de las aplicaciones. Enlace de datos. Ember.js está muy unido a *Handlebars.js*, es una librería que permite definir *templates* o plantillas semánticas con las cuales podemos representar data en nuestras vistas de manera sencilla. La automatización de cambios, es decir, un objeto *JavaScript* que este enlazado cambia automáticamente, se actualiza *DOM (Document Object Model)* de la *App* y viceversa, o sea mientras que el *DOM* cambia, se actualiza automáticamente el objeto de *JavaScript*. Mayor enfoque en la creación. Ember.js permite enfocarse en el desarrollo de la aplicación (Picca, 2014).

Las desventajas son la poca información, recursos para aprender y practicar que estén actualizados de forma gratuita. Curva de aprendizaje un tanto abrupta ya que es indispensable que respetemos todas las convenciones y el flujo de eventos del *framework* para aprovechar toda su potencia (Nitsnets, 2013).

Tiene como limitación que el *framework* Ember.js no tiene buenas herramientas de aprendizaje.

2.5.3 Angular.js

Angular.js es un *framework JavaScript Open Source* desarrollado por *Google*, sirve para crear *Webapps* en lenguaje cliente con *JavaScript* ejecutándose con el conocido *single-page applications* (aplicación de una sola página) que extiende el tradicional *HTML* con etiquetas propias. Es totalmente extensible y funciona bien con otras bibliotecas. Cada función puede ser modificada o reemplazada para satisfacer las necesidades de flujo de trabajo de desarrollo y características únicas (Nitsnets, 2014a).

Presenta características tales como: no oculta el *HTML* y el *CSS*, toma sus fortalezas y las extiende volviéndolas adecuadas para la descripción de vistas dinámicas. El resultado es un flujo de trabajo que resulta bastante familiar para cualquier desarrollador Web y con un estilo

de programación de *JavaScript* sorprendentemente conciso, claro y enfocado. Fácil comprensión para los que comienzan a usarlo pues ofrece características sofisticadas para desarrolladores con necesidades complejas.

El código de aplicaciones creadas con Angular.js siempre está organizado Modelos, Vistas, Controladores y opcionalmente Servicios (Activ, 2014).

El alcance de Angular.js es que es ideal para declarar documentos estáticos.

Las ventajas que tiene es un potente sistema de *templating* incluido en el mismo *framework*. Sincronización entre vistas y modelos para crear páginas *one-page*. Uso de *directives* para la creación de nuevos atributos o nuevas *etiquetas html*. Utilización de filtros para alterar la presentación de datos. Utiliza servicios que se encargan de la comunicación con el servidor para la consulta de datos (Serrano, 2014).

Las desventajas de Angular.js son: funciona bien para aplicaciones *REST* en donde la mayor carga del desarrollo está en el lado del cliente, pero no para aplicaciones donde el manejo de datos es muy grande y se deba atribuir un tanto de esa carga al servidor.

Sintaxis de *templating* común por muchos frameworks de este tipo incluso en el lado del servidor. Se corre el riesgo de crear conflictos, pero afortunadamente esto se puede alterar en la configuración de Angular.js. La directiva *ng-cloak* permite ocultar elementos hasta que Angular.js los tiene listos, sin embargo en algunas ocasiones tus hojas de estilo pueden interferir. Alta segmentación de nuestro scripting y estructura de archivos, se deben tener separados los *views*, los *models*, los *modules*, los *filters* y los *directives* (Serrano, 2014).

La limitante de Angular.js es que hay que tener conocimientos previos en *HTML*.

2.6 Selección del framework para la construcción de la interfaz Web para el SLD

A continuación se presentan tablas comparativas entre los *framework* por el tipo de lenguaje en que están contruidos. Se atienden características como, si son multi-navegador o sea si son soportados por la gran diversidad de navegadores Web que existen en los principales dispositivos móviles. Si presentan un diseño Web adaptable, que se adapten a las dimensiones de los dispositivos. Características como *open source*, tamaño, integración con otros *frameworks*, importantes para la selección. También se verá si cuentan con comunidad de

desarrolladores y si presentan documentación, importantes para un buen desempeño ya que se verá más fácil el trabajo.

La siguiente tabla muestra una comparación entre los *frameworks HTML5*.

Tabla 2.2: Comparación entre los *Frameworks HTML 5*

Características	Twitter Bootstrap 2.3.1	Foundation 4
Multi-Navegador	Firefox, Chrome, Safari, IE 7+ y Opera	IE 7+, Firefox, Chrome, Safari y Opera 11
Diseño Web adaptable	✓	✓
<i>Open Source</i>	✓	✓
Tamaño	140 KB	249 KB
Integración con otros <i>frameworks</i>	✓	x Limitada algunos <i>frameworks</i>
Comunidad	✓	x No cuenta con una comunidad oficial
Documentación	✓	x Muy poca documentación

La siguiente tabla muestra una comparación entre los *frameworks CSS*.

Tabla 2.3: Comparación ente los *Frameworks CSS*.

Características	960 Grid System	Blueprint CSS
Diseño Web adaptable	✓	✓
Multi-Navegador	✓	✓
Documentación	✓	✓
Comunidad	x No cuenta con una comunidad oficial	✓
Integración con otros <i>frameworks</i>	✓	✓

La siguiente tabla muestra una comparación entre los *frameworks JavaScript*.

Tabla 2.4: Comparación ente los *Frameworks JavaScript*.

Características	Backbone.js	Ember.js	Angularjs
Multi-Navegador	IE 6+, Firefox 2+, Chrome, Opera, Safari	IE 9+, FF 11+, Chrome 17+, Safari 5+	IE8 +, Chrome, Safari, FireFox
Diseño Web adaptable	✓	✓	✓
Open Source	✓	✓	✓
Integración con otras frameworks	✓	✓	
Tamaño	51 KB -120 KB	308 KB	81 KB
Comunidad	✓	X No tiene una comunidad oficial	✓
Documentación	✓	X Muy poca documentación	✓

Una vez estudiado las características de los *frameworks* presentados para crear la nueva interfaz del SLD ha llegado a la conclusión que los *frameworks* que utilizan lenguaje JavaScript no se deben utilizar. El *framework* Angular.js no funciona bien para manejar gran volumen de datos y esta es una de las características del SLD, su curva de aprendizaje es abrupta. El *framework* Ember.js tiene pocas herramientas de aprendizaje disponibles, actualizadas y gratuitas. El *framework* Backbone.js el aprendizaje es lento, hay que tener conocimientos previos en *jQuery*. Todos estos *framework* tienen también como limitante la necesidad de saber lenguaje *JavaScript*, por lo cual quedan descartados.

Una vez descartados los *framework* de lenguaje JavaScript, queda revisar los de lenguaje CSS y *HTML5*. El *framework* 960 Grid System al igual que el Blueprints CSS presentan las desventajas que hay que tener conocimientos previos en CSS, por lo cual no son de ventaja para el trabajo, por lo que se decidió trabajar con los *frameworks* de *HTML5* ya que presentan todas las características necesarias para implementar la interfaz, con un capacidad de multi-navegador, presentan un diseño Web adaptable, son *open source*, pequeño tamaño de almacenamiento, integración con otros *frameworks* y presentan documentación. Entre los dos se decidió trabajar con Twitter Bootstrap ya que fue más fácil acceder a él para la realización del trabajo.

2.7 Consejos a seguir para el diseño de una página Web para móviles

Con la gran cantidad de dispositivos móviles que se encuentran a disposición, en la actualidad, se puede llegar a la conclusión que los usuarios a los que se quiere alcanzar poseen cientos de dispositivos diferentes, y la mala noticia es que es imposible realizar un único diseño que funcione en todos y cada uno de ellos. Por tanto se verán obligados a limitarse a los terminales que sean más habituales para el objetivo perseguido.

Dentro de los dispositivos móviles existen cientos de móviles diferentes que varían en cuanto a tamaño, orientación y resoluciones de pantalla, soporte de *css* y *javascript*, detección de gestos y posición, botones de hardware, etc. Sobre todo a la hora de realizar una página Web en su versión móvil se tiene que asegurar que sea accesible para el mayor número de dispositivos.

En una Web habrá menos espacio de pantalla disponible pues se ha de reservar el que ocupan las barras del navegador, pero sin embargo no hará falta buscar un sitio para poner un botón de atrás/volver. **¿Qué va a estar visible en la pantalla?** Presencia de botones de hardware o teclado, ejemplo: botones de atrás/volver o de desplegar un menú que tienen los dispositivos con *Android*, *Nokia* y *Blackberry*. Soporte de *javascript*, ejemplo: ¿tengo soporte suficiente para desplegar/plegar listas y capas?

¿Para qué resoluciones de pantalla? Existen muchas resoluciones de pantalla diferentes, tanto en tamaño como en orientación. Si se quieren soportar varias estarán obligados a generar los gráficos a 2 o 3 tamaños distintos. Para páginas Web es mejor pensar en un diseño fluido, que pueda adaptarse a pequeñas variaciones en las dimensiones de la pantalla e incluso al cambio de orientación de los dispositivos.

Para esto es muy aprovechado la característica que ofrece Twitter Bootstrap, *responsive desing*. En general el usuario no espera encontrar los mismos contenidos en un móvil, de forma que no basta con mostrar lo mismo pero con diferente distribución. Es una buena opción para Webs pequeñas, la Web del SLD. Se optó por esta solución, pero debe hacerse bajo la filosofía del *progressive enhancement* dado que las *media queries* de CSS en las que se basa este método no funcionan en todos los terminales. La versión base debe ser la que

tenga el *css* básico, las imágenes pequeñas y sin *javascript*, y no la de PC, que eso es lo que se quiere lograr con la nueva interfaz Web para el SLD.

El *responsive desing* de Twitter Bootstrap contiene media *queries* que permiten a un CSS personalizado basado en una serie de condiciones de *ratios*, anchos, tipo de pantalla, pero por lo general se centra en torno *min-width* y *max-width*: modificar el ancho de columna en nuestra red, apilar los elementos en lugar de flotación donde sea necesario y cambiar el tamaño de los títulos y el texto a ser más apropiado para dispositivos.

2.8 Re-direccionar la nueva interfaz para dispositivos móviles

Ahora el SLD posee dos interfaces una para clientes Web, que acceden mediante un ordenador o para clientes móviles, que acceden mediante un dispositivo móvil. Para direccionar ahora las interfaces del SLD hay varias variantes que se utilizan en la actualidad. Están los *URL* independientes donde en esta configuración, cada *URL* de escritorio tiene otra *URL* equivalente que proporciona contenido optimizado para móviles. Un ejemplo de configuración habitual sería el siguiente: las páginas de *www.example.com* se muestran a los usuarios de ordenador y las páginas equivalentes de *m.example.com* se muestran a los usuarios de móvil. Para esto en la página para ordenadores, se añade una etiqueta de enlace especial y en la página para móviles, se añade una etiqueta de enlace que dirija a la *URL* para ordenadores correspondiente. Para hacerlo hay que mantener una relación de 1 a 1 entre la página para móviles y la página para ordenadores correspondiente. En especial, evitar la anotación de muchas páginas para ordenadores que hagan referencia a una sola página para móviles (o viceversa).

Otra vía es el re-direccionamiento automático, que se tiene dos formas de hacerlo. Una es el re-direccionamiento *HTTP*, que es muy común para re-direccionar a los clientes a *URL* específicas para el dispositivo. Normalmente, el re-direccionamiento se lleva a cabo al utilizar el agente de usuario que hay en las cabeceras de solicitud *HTTP*. Es importante que el re-direccionamiento sea coherente con respecto a la *URL* alternativa especificada en la etiqueta de enlace o en el *sitemap* de la página. Si es difícil implementar el re-direccionamiento *HTTP*, se puede utilizar JavaScript para re-direccionar a los usuarios a las

URL que señala la etiqueta de enlace. Si se opta por utilizar esta técnica, hay que tener en cuenta la latencia provocada por el lado del cliente del re-direccionamiento debido a la necesidad de descargar primero la página y después analizar y ejecutar el *JavaScript* antes de activar el re-direccionamiento.

Hay muchas más formas de implementar un re-direccionamiento basado en *JavaScript*. Por ejemplo, se puede usar *JavaScript* para ejecutar las consultas de medios que el sitio ya utiliza en las anotaciones de enlace de la página. Para esto se debe tener en cuenta que *JavaScript* puede deshabilitarse en todos los navegadores por acción del usuario, por lo que el mejor método sería redirigirlo mediante *PHP*.

La detección del tipo de dispositivo que está utilizando el usuario puede realizarse en el servidor mediante *PHP* a través de un proyecto de código abierto mantenido y distribuido de forma gratuita en GitHub llamado *PHP Mobile Detector*.

PHP Mobile Detector es de uso sencillo e incluye todos los dispositivos móviles actuales más significativos, a la vez que se actualiza con asiduidad con los nuevos dispositivos que surgen. Solo es necesario descargar el archivo *Mobile-Detect.php* desde GitHub. El archivo se coloca en una carpeta en el servidor. Este archivo se incluye en un *script php* donde se utiliza *require* o *include*, según las necesidades y puede detectar cualquier “*smartphone* o *tablet*”:

Los diferentes sitios Web implementan diferentes políticas de re-direccionamiento. Algunos sitios Web solo re-direccionan a los usuarios de móvil que visitan una página para ordenadores a la página para móviles (re-direccionamientos "unidireccionales") y otros sitios Web re-direccionan a los usuarios de móvil y a los de ordenador si visitan, respectivamente, páginas para ordenadores o para móviles (re-direccionamientos "bidireccionales").

2.9 Consideraciones finales del capítulo

Como consideraciones finales del capítulo, se han expuesto los principales *frameworks* tanto de lenguaje *HTML5* como *CSS* y *JavaScript*, con un análisis de sus principales características, ventajas y desventajas.

Se realizó una selección del mejor *framework* para el desarrollo de la interfaz Web para el Sistema de Laboratorio a Distancia, el cual fue el *framework* Twitter Bootstrap, que cumple con las principales características para un buen desempeño del trabajo.

Se vieron aspectos para hacer un diseño Web para móvil, que están dados a la necesidad de conocer los móviles más utilizados por los estudiantes, que es el principal público al cual está dirigido este trabajo. También las principales preguntas a la hora de llevar a cabo el diseño. ¿Qué va a estar visible en la pantalla? ¿Para qué resoluciones de pantalla?

CAPÍTULO 3. RESULTADOS OBTENIDOS

En este capítulo se mostrará, cómo quedó el nuevo diseño de la interfaz Web para el acceso mediante dispositivos móviles al Sistema de Laboratorio Remoto a Distancia. Se mostraran los nuevos cambios a las páginas de las prácticas y la respuesta de estas. Se verán también los resultados obtenidos en la nueva interfaz del SLD.

3.1 Arquitectura General del Sistema

La arquitectura del Sistema de Laboratorios a Distancia (SLD) es mostrada en la figura 3.1.

La interfaz de usuario está formada por páginas *HTML Y PHP*, para el registro de usuarios y la administración y gestión del sitio Web. El sistema puede ser accedido desde cualquier computadora con conexión a la red usando cualquier navegador para Web, al igual que desde cualquier dispositivo móvil usando un navegador para Web conectado a una red *WiFi*.

El Servidor de Administración de Prácticas (SAP) está formado por páginas *PHP*, lo cual hace al sistema más portable y seguro, al permitir ejecutarse este nivel en sistemas operativos *Windows o Linux*. La comunicación del SAP con el Cliente de Administración de Prácticas (CAP), se hace vía *HTTP (Hypertext Transfer Protocol)* por el puerto 80 estándar, a diferencia del uso de *socket* que necesitaban puertos potencialmente bloqueables por los *firewall*.

Otro elemento que conforma el sistema es el Cliente de Administración de Prácticas (CAP), el cual está implementado con *Web Services*. El CAP se encarga de comunicarse con el *Matlab/Simulink* el cual ejecuta las prácticas tanto reales como simuladas.

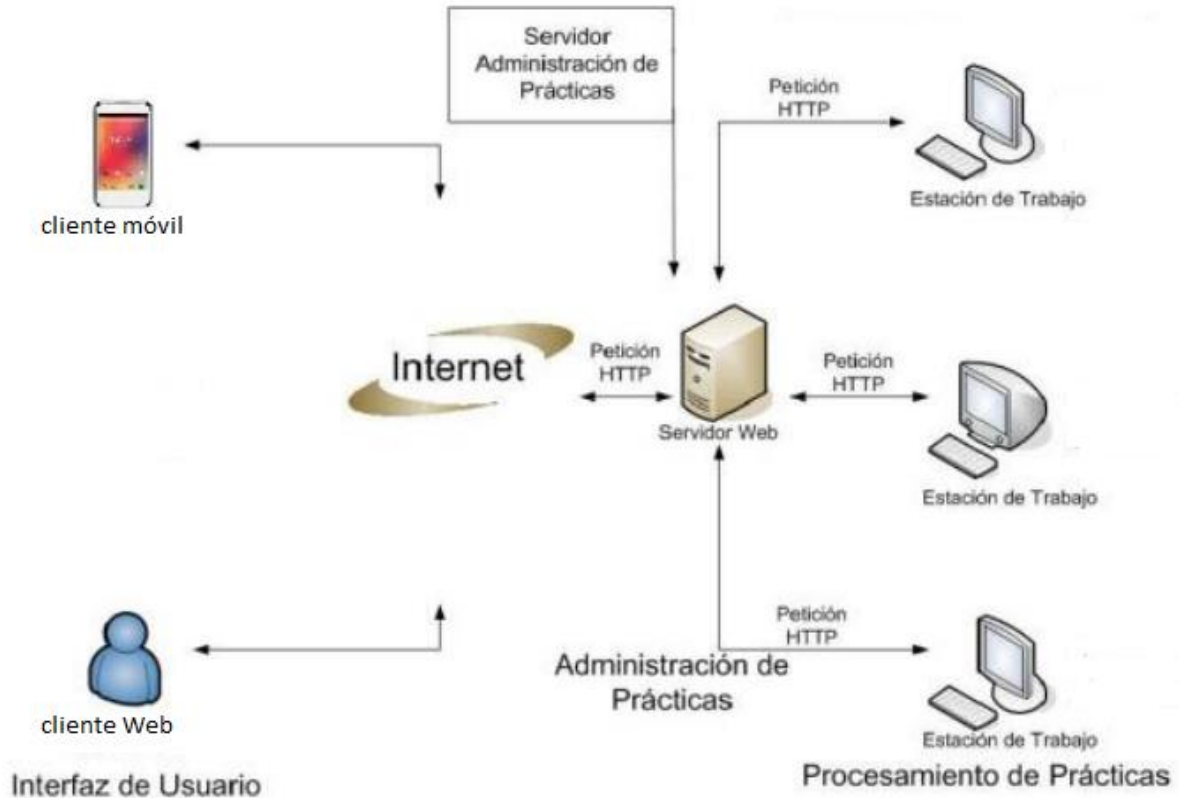


Figura 3.1: Arquitectura del SLD.

El sistema realiza las prácticas con *Matlab/Simulink* y el *Toolbox Real Time Windows Target* debido a la facilidad de uso y potentes capacidades de este software. *Real Time Windows Target* permite la ejecución del esquema *Simulink* en tiempo real sobre *Windows*. El hecho de ejecutar directamente un esquema *Simulink* brinda ventajas, ya que el tiempo y la complejidad de crear nuevos algoritmos de control se reducen drásticamente y permite una fácil creación y modificación de esquemas de control. En la Tabla 3.1 se relacionan cada una de las partes de la plataforma, así como su implementación.

Tabla 3.1: Partes e implementación de la plataforma.

Partes de la Plataforma	Implementación en la plataforma del SLD
Interacción del Usuario con la Plataforma	Sitio Web con <i>Web Services</i> , páginas HTML y PHP.
Gestión de Pedidos de Prácticas	Aplicaciones propias (Servidor y Estación), <i>Web Services</i> .
Procesamiento de las Prácticas	Matlab con Real Time <i>toolbox</i> de Humusoft.

3.2 Estructura de la página Web

El sistema SLD contaba con una interfaz de usuario realizada en el año 2009, la cual no tenía las características para un desempeño en teléfonos móviles. El objetivo de este trabajo fue diseñar una interfaz que les permitiera a los usuarios del Sistema de Laboratorios a Distancia, la realización de prácticas desde sus teléfonos celulares. La nueva interfaz cuenta con una misma estructura que la anterior, una página de Inicio donde los usuarios se identifican o se registran en el sitio. (Figura 3.2)

La nueva interfaz tiene la propiedad de *responsive desing* lo que le permite adaptarse a las dimensiones de los teléfonos móviles o *tablet* de los usuarios. Al igual que la anterior versión de la interfaz, la nueva, cuenta con una barra de navegación para acceder a las páginas de Teoría, Practicas, no se pude acceder a la realización de las prácticas hasta que no estés logueado como usuario, y Plataforma.

SLD WEB | Sistema de Laboratorios a Distancia Inicio Teoría Prácticas Plataforma

Acceso **Obligatorio**

Autenticación

Nombre de usuario

Contraseña

Dominio

db

Aceptar

Registrarse

Bienvenidos al Sistema de Laboratorios a Distancia SLDWEB

Desarrollado por GARP
(Grupo de Automatización Robótica y Percepción)

En colaboración con UPM (DISAM)
(Departamento de Automática, Ingeniería Electrónica e Informática Industrial)

Permite ejecutar experiencias de control tanto de forma virtual (simulando con el modelo correspondiente) como real (accionando un dispositivo en tiempo real).

©Copyright 2015 GARP - Facultad de Ingeniería Eléctrica
Universidad Central "Marta Abreu" de Las Villas.

Figura 3.2 Interfaz nueva del SLD.

Una vez logueado en el sistema el usuario podrá acceder al menú, en el cual se encuentran enlaces a Prácticas, donde se muestran las prácticas realizables presentadas por el sistema, Mis Prácticas, desde donde puedes revisar las prácticas realizadas, desde prácticas revisadas

y por revisar. Además de que puedes acceder a las otras páginas, que también se encuentran en la barra de navegación, Inicio, Teoría, Plataforma.

En la barra de navegación después de logueado aparece el botón de usuario, desde el cual puedes acceder a la opción de editar usuario para cambiar las características como el nombre de usuario, correo electrónico y contraseña. Otra opción presente en el icono de usuario es la de salir, que su función es desloguearse del SLD, e ir a la página de inicio.

Mientras se trabaja desde un teléfono móvil o una *tablet*, puedes acceder a estos menús desde un botón en la parte superior de la página, esto facilita la navegación ya que muestra el contenido primero y después puedes acceder a las opciones de navegación de esta desde este botón (Figura 3.3 a y b).

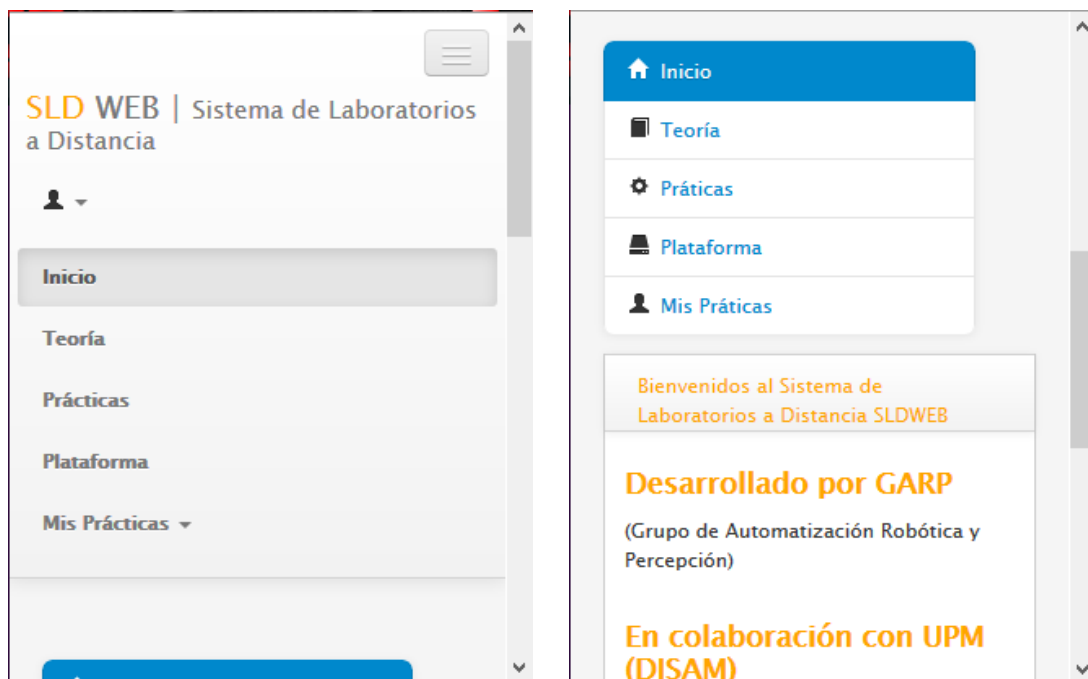


Figura 3.3 a) Barra de navegación desde la vista desde un móvil. b) Menú desde la vista desde un móvil.

Para las páginas de Teoría y Plataforma, se tomó un diseño más estructurado, que le permita ver al usuario desde su móvil todo el contenido que se mostrará en la página (Figura 3.4 y 3.5).

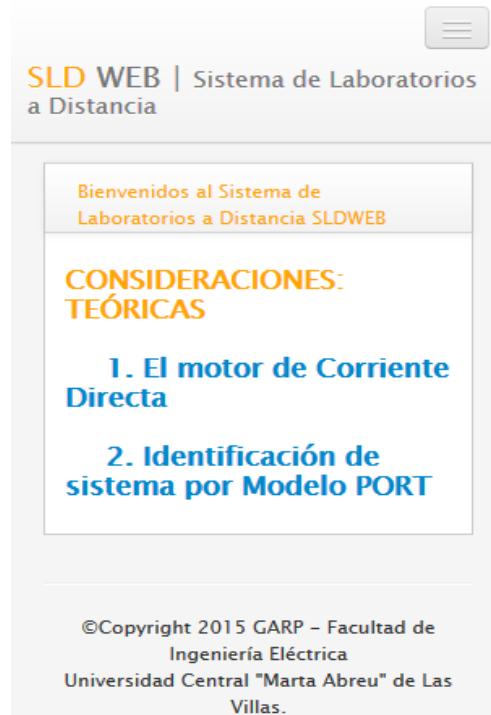


Figura 3.5 Ambiente más estructurado de teoría.

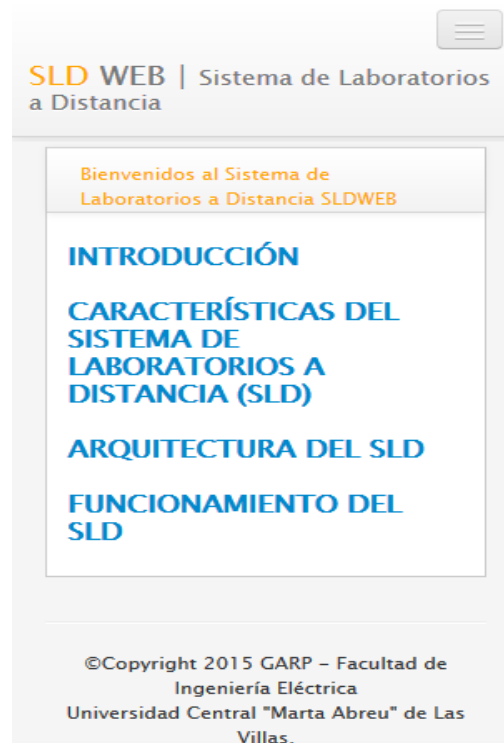


Figura 3.4 Ambiente más estructurado de plataforma.

En la página de práctica, al seleccionar uno de las opciones visibles, ya sea introducción, características del sistema de laboratorios a distancia, arquitectura o funcionamiento, se despliega la información de manera que es fácilmente apreciable desde la terminal móvil en cuestión. Si desea volver a ocultar la información, solo tiene que seleccionar la opción seleccionada y se ocultará.

En el caso de la página de teoría, cuenta con una forma similar de presentar su información, al seleccionar una de las opciones visibles, el motor de corriente directa o identificación del sistema por modelo PORT, se despliega la información, pero como en este caso era mucho más extensa, se dividió como un sistema de páginas que el usuario puede seleccionar y ver la información presentada (Figura 3.6). Para ocultar la información, selecciona la opción seleccionada y automáticamente esta se ocultará.

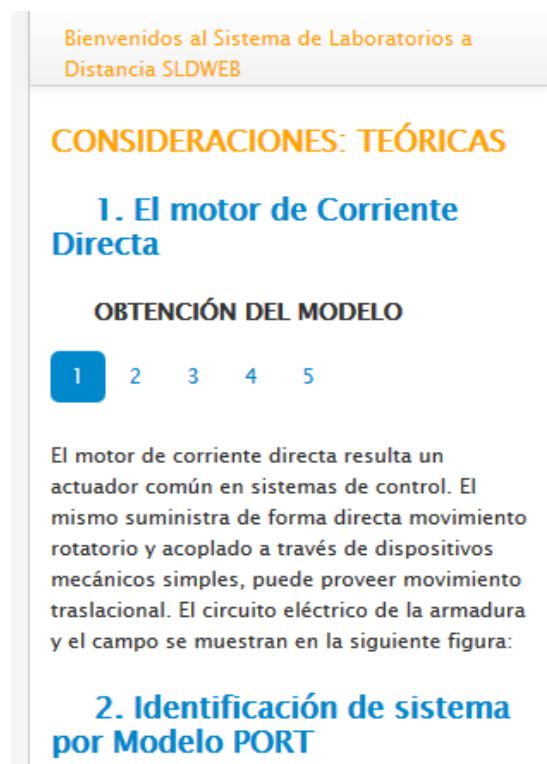


Figura 3.6 Presentación de información en la página Teoría.

Con el objetivo de las fotos de los esquemas de las prácticas se puedan apreciar correctamente se buscó una clase CSS de zoom. Esta clase le permite a las fotos que por el tamaño de las pantallas de las terminales móviles, no sean bien apreciadas por los usuarios, al dar clic sobre

estas aparezcan en mayor tamaño y puedan ser aumentadas de tamaño por si el usuario lo necesite para una mayor apreciación de la misma.

3.3 Estructura de las páginas de prácticas

Las prácticas en la anterior interfaz del SLD, mostraban toda la información directamente en la página porque está hecha para interfaces de pc de escritorio o portátiles o sea mayores 1024x720 pixeles. Esta nueva interfaz realizada para resoluciones más pequeñas, del tipo de móviles o *tablet*, se presenta con una distribución de la información estructurada de forma tal que el usuario tenga visible dos opciones o campos en los que está contenida la información y los parámetros realización de la práctica (Figura 3.7).

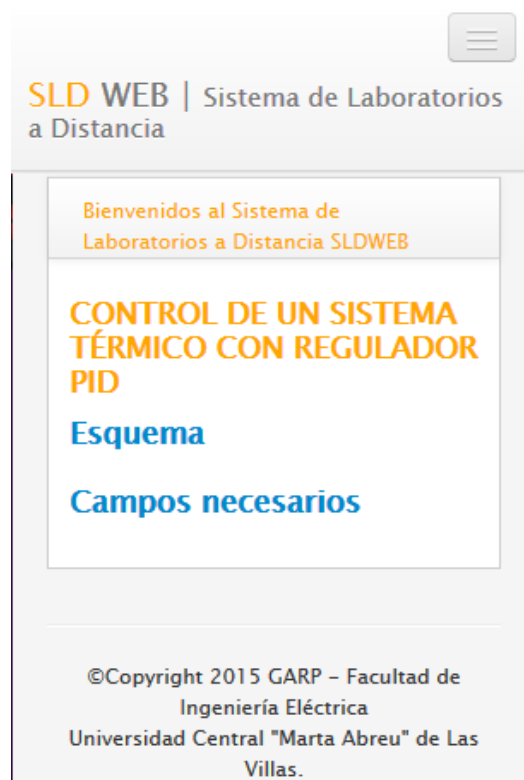
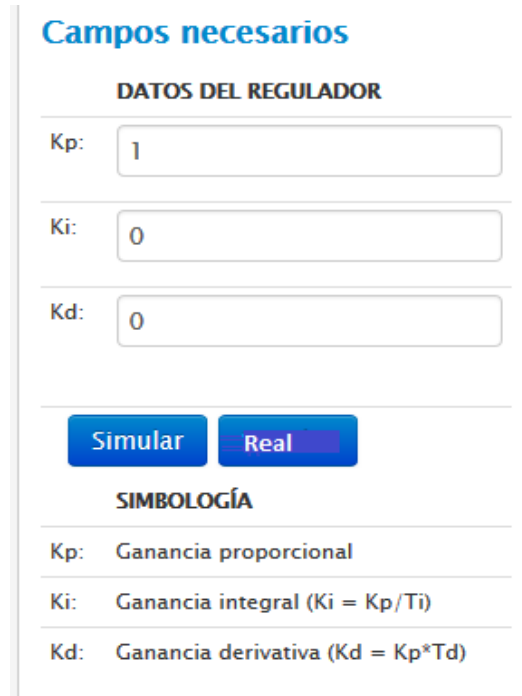


Figura 3.7 Presentación de la página de práctica en la nueva interfaz.

Esta forma de presentación de la información de la nueva interfaz le brinda dos opciones al usuario, esquema y campos necesarios. En el esquema el usuario puede ver el esquema en *Simulink* de la práctica de va a desarrollar, así como una explicación de los parámetros que debe introducir y como se desempeñará, también exhibirá si están disponibles u ocupadas

las estaciones reales de la práctica y en qué tiempo se desocuparán. En la opción de campos necesarios se muestran los parámetros que deben ser completados o seleccionados por el usuario para realizar la práctica de laboratorio ya sea simulada o real y la una tabla con la simbología utilizada (Figura 3.8).



Campos necesarios

DATOS DEL REGULADOR

Kp:

Ki:

Kd:

SIMBOLOGÍA

Kp:	Ganancia proporcional
Ki:	Ganancia integral ($K_i = K_p/T_i$)
Kd:	Ganancia derivativa ($K_d = K_p \cdot T_d$)

Figura 3.8 Vista de campos necesarios a tener en cuenta en la práctica desde un móvil.

3.4 Comprobación de los resultados

Para realizar esta tarea se tomó la práctica de laboratorio del SLD, control de un sistema térmico con regulador PID. Con esta práctica se puede realizar por parte del alumno un estudio de la respuesta del sistema en lazo cerrado ante cambios de ganancia, la figura 3.9 a y b muestra el correspondiente diagrama y datos del regulador de control visto desde una terminal móvil. Para esto el estudiante realiza variaciones de K_p , al mantener en cero los valores de K_i y K_d .

Por último, implementa un regulador PID, a través primeramente por un P y un PI. La función de transferencia teórica del sistema se muestra en (1).

$$G(s) = \frac{0.9}{(1+7s)^3} \quad (1)$$

Bienvenidos al Sistema de Laboratorios a Distancia SLDWEB

CONTROL DE UN SISTEMA TÉRMICO CON REGULADOR PID

Esquema

A continuación se muestra el esquema para el control un sistema térmico.

Introduzca los parámetros del regulador PID que se desea ensayar. La estructura implementada para el regulador es de la forma suma de ganancias proporcional (K_p), integral (K_i) y derivativa (K_d). La entrada de referencia (Temperatura deseada) es igual a

Campos necesarios

DATOS DEL REGULADOR

Kp:

Ki:

Kd:

Simular

Real

SIMBOLOGÍA

Kp: Ganancia proporcional

Ki: Ganancia integral ($K_i = K_p/T_i$)

Kd: Ganancia derivativa ($K_d = K_p \cdot T_d$)

Figura 3.9 a) Esquema de la práctica de sistema térmico con regulador PID visto desde móvil.

b) Datos del regulador de la práctica de sistema térmico con regulador PID visto desde móvil.

La respuesta simulada del sistema térmico con regulador PID, para $K_p = 1$, $K_i = 0.067$ y $K_d = 7$ se muestra en la figura 3.10. Un dato muy importante es que en cada ejecución el estudiante puede descargar los resultados de las prácticas para luego, si lo desea, trazar las gráficas en el Matlab y realizar un estudio más minucioso de lo obtenido.

Para una mayor apreciación en la terminal móvil de la respuesta, con la clase CSS de zoom, el usuario puede dar clic sobre esta y aparezca en mayor tamaño.

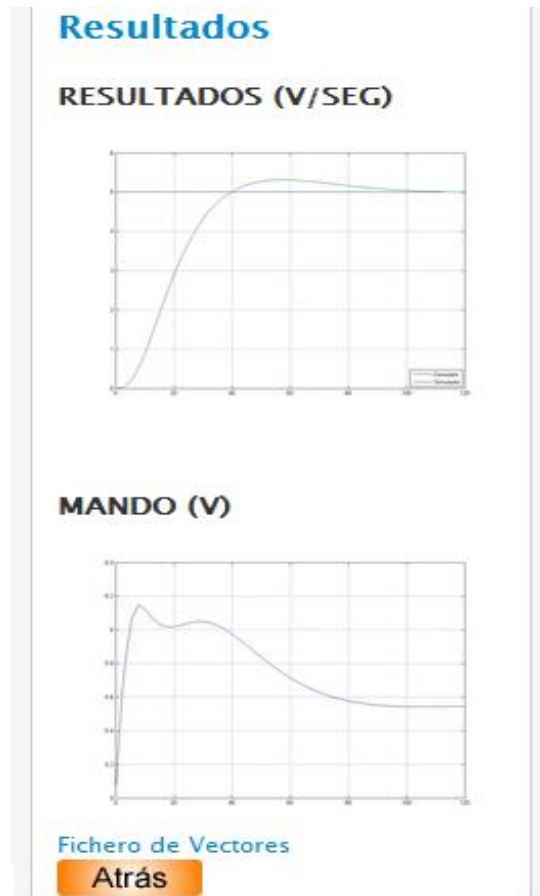


Figura 3.10. Respuesta simulada del sistema térmico con regulador PID.

3.5 Revisión de los resultados

Como resultados se obtienen la realización de la nueva interfaz para móvil del Sistema de Laboratorio a Distancia. La nueva estructura de la interfaz cuenta con una barra de navegación, menú de navegación y menú de usuario, estos están ocultos por un botón desde la visita del móvil, aportando una mejor navegación desde teléfonos. Las páginas de gran contenido de información, se estructuraron de forma que el usuario pueda observar que contenido se van a mostrar, para después acceder al necesitado.

Las páginas de las prácticas se organizaron en dos campos, en uno se observan los datos de la práctica, instrucciones y el esquema de esta, mientras que en un segundo, los campos a seleccionar y completar para el desarrollo de la misma. Lo que permite una mejor ejecución de la simulación de la práctica en Matlab, al tener mejor distribuida la información.

La página de las respuestas muestra una distribución igual a la de prácticas, un campo donde están los datos de la práctica, esquema y directivas de trabajo. Otro donde se aprecian los resultados obtenidos, se muestran las gráficas de los resultados de la simulación en Matlab, estas gráficas no son muy apreciables desde el móvil, por lo que se le brindó al usuario la herramienta de ampliarlas para una mejor apreciación. Por último está la opción de comentarios donde el usuario puede dejar los comentarios que cree necesarios y la solicitud de evaluación al profesor, logrando la posibilidad de una mayor comunicación entre el alumno y el profesor.

La comprobación de la posibilidad de realización de prácticas desde un dispositivo móvil, al realizarse una práctica de laboratorio simulada, lo que permite ver el desempeño de la interfaz en estos dispositivos, brindándole una mayor versatilidad y aplicabilidad al SLD.

3.6 Análisis económico

Los laboratorios reales requieren de equipos electrónicos y de cómputo realmente costosos, en Cuba actualmente es muy difícil contar con un equipamiento así. La implementación de laboratorios remotos ahorra una cifra notable de dinero, una maqueta para control de temperatura (*Airstream and Temperature Control Plant LTR701*) puede costar 7 420 euros, una para control de nivel (*Three-Tank-System DTS200*) puede alcanzar la cifra de 9 390 euros. Mientras que los laboratorios remotos permiten compartir estos recursos costosos y geográficamente distantes. Esto posibilita que en una estación puedan estar conectados y trabajar muchos usuarios a la vez, por lo tanto se emplea menos equipamiento y se disminuyen las inversiones por compra de los mismos, con la limitante de la cantidad de ordenadores presentes en los laboratorios de la universidad, que son equipos caros con precios alrededor de los cientos de dólares y no siempre están a la disposición de la necesidad de los alumnos.

Con la implementación de una interfaz Web para el Sistema de Laboratorio a Distancia para dispositivos móviles, les permite a los estudiantes que tengan en su poder terminales móviles, ya sean *smartphone* o *tablet*, la posibilidad de acceder a este valioso recurso, sin necesidad de estar presentes en los laboratorios. Esto disminuye las inversiones de compra de los equipos de pc para los laboratorios, la implementación de una red inalámbrica WiFi, tiene menor costo monetario que la implementación de una red LAN física.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Como resultado de la elaboración de este trabajo se han llegado a las siguientes conclusiones:

- Producto del amplio desarrollo de los dispositivos móviles, son más los Sistemas de Laboratorio Remoto que implementan el acceso a través de estos dispositivos, por eso se corrobora la factibilidad de dotar al Sistema de Laboratorio a Distancia con una interfaz Web para ser accedido de dispositivos móviles.
- De los *frameworks* estudiados el más adecuado para la implementación de la interfaz Web para móviles es Twitter Bootstrap porque este presenta ventajas significativas como el *responsive desing*, ser multi-navegador, amplia comunidad y documentación que facilitan el desarrollo del trabajo.
- La interfaz implementada para el acceso al Sistema de Laboratorio a Distancia desde dispositivos móviles, permite un acceso amigable, transparente al sistema, con un buen desempeño de su funcionamiento, lo que se demostró a través del desarrollo de una práctica.

Recomendaciones

Por la importancia que encierra la realización de este trabajo, pensamos que pueda servir de base a futuros estudios sobre este tema, por lo que proponemos como recomendación:

- Poner a funcionar el sitio Sistema de Laboratorio a Distancia con todas sus prácticas y permitir el acceso de los estudiantes.

REFERENCIAS BIBLIOGRÁFICAS

- A. ADDA BENATTIA, A. B. Remote laboratory via Interactive Mobile Technology - Application in Electronics Learning 2014 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), November 13-14 2014 Thessaloniki, Greece. IEEE, 373-375.
- A. SAFAVI, A. A. S., P. VEISI. A Remote and Virtual PLC Laboratory Via Smartphones The 4th International Conference on e-Learning and e-Teaching, ICELET 2013, 2013. IEEE, 63-68.
- ACTIV. 2014. *Comenzando con AngularJS* [Online]. Available: <http://activ.com.mx/comenzando-con-angularjs/> 2015].
- ÁLVAREZ, M. A. 2013. *Blueprint, Framework CSS* [Online]. Available: <http://www.desarrolloweb.com/manuales/blueprint-frameworks-css.html> 2015].
- BAQUERO, O. A. G. 2014. *Tutorial de Backbone.js (I)* [Online]. Available: <http://notoquesmicodigo.blogspot.com/2014/03/tutorial-de-backbonejs.html>.
- BAYONA., A. 2014. *Taller de maquetación web* [Online]. Available: <http://www.slideshare.net/webmastergranada/taller-maquetacion-web> 2015].
- BERNARDINO, J. 2014. *960 grid, un framework para CSS* [Online]. Available: <http://jepserbernardino.com/idea/960-grid-un-framework-para-css> 2015].
- CALLAGHAN, M., HARKIN, J., MCGINNITY, T. & MAGUIRE, L. 2008. Intelligent User Support in Autonomous Remote Experimentation Environments. *Industrial Electronics, IEEE Transactions on*, 55, 2355-2367.
- COOPER, M. 2005. Remote laboratories in teaching and learning – issues impinging on widespread adoption in science and engineering education. *International Journal of Online Engineering (iJOE)*, 1.
- CHÁVEZ, A. 2014. *Si eres desarrollador web, debes utilizar Bootstrap y punto* [Online]. Available: <http://alanchavez.com/si-eres-desarrollador-web-debes-utilizar-bootstrap-y-punto> 2015].
- CHAVEZ, E. R. D. 2013. *Qué es BackboneJS* [Online]. Available: <http://www.desarrolloweb.com/manuales/que-es-backbonejs.html> 2015].

- D. LÓPEZ-DE-IPÍÑA, J. G.-Z., P. ORDUÑA. Remote Control of Web 2.0-enabled Laboratories from Mobile Devices. 2nd IEEE International Conference on e-Science and Grid Computing 2006 Amsterdam, The Netherlands IEEE, 123-123.
- DIMAGGIO., M. 2014. *Backbone.js: otro enfoque para desarrollar aplicaciones en JavaScript* [Online]. Available: <http://www.4rsoluciones.com/backbone-js-otro-enfoque-paradesarrollar-aplicaciones-en-javascript/> 2015].
- DORMIDO, S. 2004a. Control Learning: Present and Future. *Annual Reviews in Control*, 28, 115-136.
- DORMIDO, S. 2004b. Control Learning: Present and Future. *Annual Reviews in Control*, 28, 115-136.
- E. LINDSAY, P. L., Y P. K. IMBRIE., Workshop - remote laboratories: Approaches for the future. Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. W1C-1-W1C-2.
- E.GRANADO, W. C., O.PEREZ, G.CATALDO June 2013. Remote Experimentation Using Mobile Technology *Latin America Transactions, IEEE (Revista IEEE America Latina)*.
- ECHEGARAY, B. G. 2014. *¿Qué es Bootstrap?* [Online]. Available: <http://bruno.garciaechegaray.com/Bootstrap.The.Melee/presentation/#more> 2015].
- FONTELA, A. 2013. *¿Qué es Bootstrap?* [Online]. Available: <http://openwebcms.es/2013/que-esbootstrap/> 2015].
- GANARCE. 2014. *Blueprint CSS framework* [Online]. Available: <http://www.slideshare.net/ganarce/blueprint-css-framework> 2015].
- GARCÍA-ZUBIA, J. Easily Integrable platform for the deployment of a Remote Laboratory for microcontrollers Education Engineering(EDUCON),2010 IEEE, 2010 Madrid. IEEE, 327-334.
- GARCIA-ZUBIA, J., ORDUNA, P., LOPEZ-DE-IPINA, D. & ALVES, G. R. 2009. Addressing Software Impact in the Design of Remote Laboratories. *Industrial Electronics, IEEE Transactions on*, 56, 4757-4767.
- GRAVIER, C., FAYOLLE, J., BAYARD, B., ATEZ, M. & J., L. 2008. State of the Art About Remote Laboratories Paradigms - Foundations of Ongoing Mutations. *International Journal of Online Engineering (iJOE)*, 4(1), 19-25.
- GUBE, J. 2013. *10 Best Responsive HTML5 Frameworks and Tools* [Online]. Available: <http://designinstruct.com/roundups/html5-frameworks> 2015].
- GUSTAVSSON., I. The VISIR project--an open source software initiative for distributed online laboratories. REV 2007 Conference, 2007 Porto, Portugal. . IEEE.
- J.BENTO DA SILVA, W. R., J.P.SCHARDOSIM SIMAO, A.VAZ DA SILVA FIDALGO 2014. Adaptation Model of Mobile Remote Experimentation for Elementary Schools *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana*. IEEE.

- J.HARWARD June 2008. The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories *Proceedings of the IEEE* 96, 931-950.
- MONTOYA, C. 2014. *Blueprint* [Online]. Available: <http://www.blueprintcss.org/> [2015].
- NICKERSON, J. V., CORTER, J. E., ESCHE, S. K. & CHASSAPIS, C. 2007. A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. *Computers & Education*, 49, 708-725.
- NITSNETS. 2013. *Introducción al framework Ember.js* [Online]. Available: <http://www.losttiemposcambian.com/blog/javascript/intro-frameworkember/> [2015].
- NITSNETS. 2014a. *AngularJS. Framework JavaScript para Webapps* [Online]. Available: <http://www.losttiemposcambian.com/blog/javascript/angularjsframework-javascript-para-webapps/> [2015].
- NITSNETS. 2014b. *Backbone vs Angular vs Ember* [Online]. Available: <http://www.losttiemposcambian.com/blog/javascript/backbone-vs-angularvs-ember/> [2015].
- P. ORDUÑA, J. I., L. RODRIGUEZ-GIL, J. GARCÍA-ZUBIA, F. GAZZOLA, D. LÓPEZ-DE-IPÍÑA October 2011. Adding New Features to New and Existing Remote Experiments Through Their Integration in WebLab-Deusto. *International Journal of Online Engineering (iJOE)*, 7, 33-39.
- P.ORDUNA, J. G.-Z., J. IRURZUN,D.LOPEZ-DE-IPINA Enabling mobile access to Remote Laboratories Global Engineering Education Conference (EDUCON), 2011 IEEE, 4-6 April 2011 2011 Amman. IEEE, 312-318.
- PATRICIO. 2014. *Desarrollando Webs Dinámicas* [Online]. Available: <http://desarrollandowebsdinamicas.blogspot.com/2014/02/que-es-twitterbootstrap.html> [2015].
- PICCA, C. 2014. *Ember.js desde Cero: Introducción e Instalación* [Online]. Available: <http://codehero.co/ember-js-desde-cero-introduccion-e-instalacion/> [2015].
- RODRÍGUEZ, N. 2014. *960 Grid System, un framework para CSS* [Online]. Available: <http://www.genbetadev.com/desarrollo-web/960-grid-system-un-frameworkpara-css> [2015].
- ROLDÁN, C. & 2013. *Los 5 mejores frameworks MVC de JavaScript* [Online]. Available: <http://www.codejobs.biz/es/blog/2013/04/11/los-5-mejores-frameworksmvc-de-javascript#sthash.5gvsIIX.dpbs> [Accessed 2015].
- SÁNCHEZ, M. 2014. *960 Grid System* [Online]. Available: http://observatoriotecedu.uned.ac.cr/media/960_grid.pdf [2015].
- SERRANO, B. 2014. *AngularJs vs. BackboneJs* [Online]. Available: <http://foreground.com.co/blog/index.php/component/content/article/79-desarrollo-web/73-angularjs-vs-backbonejs.html>.
- SERRANO, E. 2013. *Framework de Diseño Web Adaptativo Foundation 4* [Online]. Available: <http://webadaptativo.blogspot.com/2013/03/framework-de-diseno-webadaptativo.html> [2015].

- SMITH, N. 2014. *960 Grid System* [Online]. Available: http://wiki.ead.pucv.cl/index.php/960_GRID_SYSTEM 2015].
- SUAREZ, D. D. 2014. *Nuestra Primera App con Ember.js* [Online]. Available: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=PrimeraAplicacionEmber> 2015].
- TEMPLATEMONSTER, B. 2013. *¿Bootstrap o Foundation? ¿Cuál HTML5 Framework es más Sensible para el Desarrollo de Sitios Web?* [Online]. Available: <http://www.templatemonsterblog.es/2013/06/05/bootstrap-o-foundationcual-html5-framework-es-mas-sensible-para-el-desarrollo-de-sitios-web/> 2015].
- THODE, D. S. 2012. *Crear una aplicación sencilla con Backbone.js* [Online]. Available: <http://dsthode.info/2012/09/crear-una-aplicacion-sencilla-con-backbone-js/> 2015].
- Y. LYALINA, O. M., R. LANGMANN. Smart Labs for remote experimentation in communication technologies. REV 2011 Conference, 2011 Brasov, Romania. IEEE.