



PRACTICA EMPRESARIAL EN LA EMPRESA IBM COLOMBIA

JULIAN DAVID MANTILLA BLANCO

KATHERIN GUTIÉRREZ ÁVILA

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

FACULTAD DE INGENIERÍA FISICOMECHANICAS

PROGRAMA DE INGENIERÍA MECATRÓNICA

BUCARAMANGA

2016

PRACTICA EMPRESARIAL EN LA EMPRESA IBM COLOMBIA

JULIAN DAVID MANTILLA BLANCO

KATHERIN GUTIÉRREZ ÁVILA

LÍNEA DE INVESTIGACIÓN: MODELADO Y SIMULACIÓN

DIRECTOR:

DARIO JOSE HERNANDEZ BOLIVAR

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

FACULTAD DE INGENIERÍA FISICOMECAÑICAS

PROGRAMA DE INGENIERÍA MECATRÓNICA

BUCARAMANGA

2016

### Nota de aceptación

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos para las practicas académicas por la Universidad Autónoma de Bucaramanga para optar el título de Ingeniero Mecatrónico.

---

DARIO JOSE HERNANDEZ BOLIVAR

DIRECTOR

## CONTENIDO

1. INTRODUCCION .....	11
1.1. IBM DE COLOMBIA & CIA S.C.A.....	11
1.2. DATOS GENERALES DE LA EMPRESA.....	14
1.2.1. Razón social.....	14
1.2.2. Localización .....	14
1.2.3. Teléfono .....	14
1.2.4. Dependencia adscrita .....	14
1.2.5. Función .....	14
1.2.6. Misión.....	15
1.2.7. Visión .....	15
1.2.8. Nuestros Valores.....	15
1.3. DATOS DEL JEFE INMEDIATO.....	16
1.3.1. Nombre .....	16
1.3.2. Cargo .....	16
1.3.3. Correo electrónico.....	16
1.3.4. Línea de mando .....	16
2. JUSTIFICACIÓN .....	18

3. OBJETIVOS .....	20
3.1. OBJETIVO GENERAL.....	20
3.2. OBJETIVOS ESPECÍFICOS .....	20
4. MARCO TEORICO.....	21
4.1. JavaScript.....	21
4.2. Bluemix.....	23
4.3. Cloudant .....	25
4.4. PostgreSQL .....	26
4.5. Arduino .....	28
4.6. Node-red.....	31
4.7. HTML 5.....	32
4.8. AngularJS .....	33
4.8.1. MVC .....	35
4.9. CSS .....	36
4.10. SASS.....	37
4.11. PHP .....	38
4.12. Curam IBM .....	39
5. CUADRO DE RESULTADOS.....	40
6. ACTIVIDADES REALIZADAS .....	43

CONCLUSIONES .....	46
BIBIOGRAFIA.....	47
ANEXOS.....	49

## LISTA DE FIGURAS

Figura 1. Logo oficial de IBM .....	11
Figura 2. Logo oficial de IBM Bluemix.....	23
Figura 3. Página oficial de IBM Bluemix. ....	24
Figura 4. Catalogo de IBM Bluemix . ....	24
Figura 5. IBM Cloudant .....	25
Figura 6. Logo PostgreSQL. ....	26
Figura 7. Logo Oficial Arduino. ....	28
Figura 8. Logo Oficial HTML. ....	33
Figura 9. Logo Oficial Angular JS .....	34
Figura 10. Esquema MVC.....	35
Figura 11. Logo Oficial PHP. ....	38

## LISTA DE TABLAS

Tabla 1. Cuadro de resultados.....	40
Tabla 2, Actividades realizadas. ....	43

## LISTA DE ANEXOS

Anexo A. Informe técnico del servicio Web.....	49
Anexo B. Desarrollo de la Pagina web "hAppy Health".....	69
Anexo C. Desarrollo de Pagina web "hAppy Doc".....	76
Anexo D. Manual de presentacion "hAppy Health".....	82
Anexo E. Manual de presentación "hAppy Doc".....	89
Anexo F. Implementación del Laboratorio 1 de IBM Curam. ....	92
Anexo G. Implementación del Laboratorio 2 de IBM Curam.....	103

## RESUMEN

En este informe se detallan las actividades realizadas durante el periodo de practica académica iniciado el 01 de marzo de 2016 hasta el 31 de agosto de 2016. Aquí se encuentra una descripción detallada de las labores realizadas durante el transcurso de la misma y los resultados obtenidos. De igual manera se presenta información de la empresa IBM Colombia y de los recursos tecnológicos utilizados.

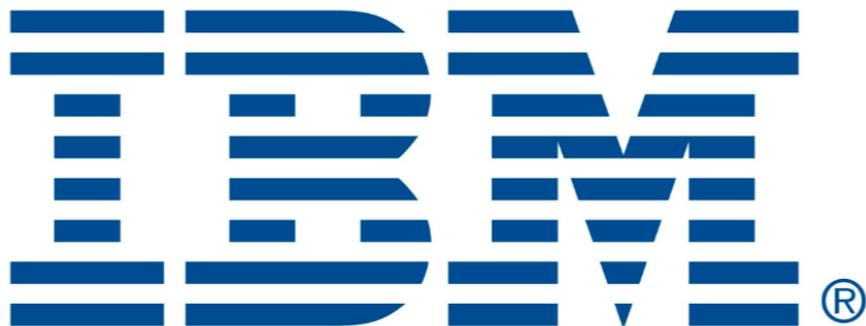
## ABSTRACT

This report presents the activities undertaken during the period of professional internship which started on March 1 2016 up to August 31 2016. Here is a detailed description of the work done during the course and the results obtained of it. Similarly, IBM Colombia, business information Colombia and technological resources used are presented.

## 1. INTRODUCCION

### 1.1. IBM DE COLOMBIA & CIA S.C.A.

Figura 1. Logo oficial de IBM



IBM cuyas siglas corresponden a la abreviación de International Business Machines, es una multinacional estadounidense de tecnología informática y consultoría, cuyos orígenes se remontan a partir del siglo XIX. Es una empresa que fabrica y comercializa hardware y software, ofrece servicios de infraestructura, servicios de TI, outsourcing, integración de sistemas, financiamiento, alojamiento de internet y consultoría informática. Es catalogada como una de las empresas más grandes y rentables del mundo, opera en 170 países y cuenta con más de 386.000 empleados. Está dedicada a proporcionar soluciones para la mejora de procesos de negocio, facilitando a sus clientes métodos para hacer frente a los problemas empresariales. [1]

IBM es la compañía de tecnologías de información que más invierte en investigación y desarrollo del mundo, cuenta con 3,000 científicos y 8 centros de investigación por todo el mundo. Además, posee más de 24 laboratorios de desarrollo y emplea más de 125,000 técnicos, con más de 37,000 patentes en todo el mundo, encabezando la lista de patentes registradas, Sus empleados han recibido cinco Premios Nobel, cuatro Premios Turing, nueve National Medals of Technology y cinco National Medals of Science. Las invenciones famosas de IBM incluyen el cajero automático, el disquete, el disco duro, la banda magnética, el modelo relacional, el Universal Product Code, el "financiamiento swap", el sistema de reservas aéreas SABRE, DRAM y el sistema de inteligencia artificial Watson [2].

El IBM Student Program es uno de los programas corporativos por excelencia en materia de selección. Este se implementa en todas las regiones con el fin de incorporar jóvenes con ganas de formar parte de una gran empresa, en complemento de su formación académica.

Aprovechando lo anterior y gracias a los conocimientos adquiridos en nuestra formación académica, la práctica se desarrolló en el Center of Excellence de IBM Colombia, el cual se ubica en Bucaramanga, Santander y está enfocado a dar solución a temas relacionados con la salud.

En este documento se encuentran las actividades realizadas y los resultados obtenidos durante el periodo en el que se desarrolló el programa, comprendiendo las diferentes líneas en las que se trabajó y que son de interés para la compañía.

## 1.2. DATOS GENERALES DE LA EMPRESA

### 1.2.1. Razón social

IBM de Colombia & Cia S.C.A.

### 1.2.2. Localización

Carrera 53 # 100-25 Bogota D.C. , Colombia

### 1.2.3. Teléfono

+ 57-6-395151

### 1.2.4. Dependencia adscrita

Sales and Distribution

### 1.2.5. Función

Desarrollar demos, pruebas de concepto e implementación de soluciones basadas en IBM Care Management, Bluemix y Watson Explorer para el sector salud

#### 1.2.6. Misión

Ayudar a nuestros clientes a alcanzar sus metas de negocio proveyéndoles servicios y soluciones innovadoras

#### 1.2.7. Visión

Ser la compañía elegida por nuestra innovación, soluciones, productos y servicios. Ser reconocida por la calidad humana y profesional de nuestra gente y por nuestra contribución a la comunidad.

#### 1.2.8. Nuestros Valores

- Dedicación al éxito de nuestros clientes
- Innovaciones que tienen importancia
- Confianza y responsabilidad en cada relación

### 1.3. DATOS DEL JEFE INMEDIATO

#### 1.3.1. Nombre

Mario Alonso Mateus

#### 1.3.2. Cargo

IBM Technical Consultant

#### 1.3.3. Correo electrónico

[mmateusg@co.ibm.com](mailto:mmateusg@co.ibm.com)

#### 1.3.4. Línea de mando

Presidenta y directora ejecutiva de IBM:  E-mail:	Virginia Marie Rometty  grometty@us.ibm.com
Vicepresidente, Sales & Distribution:  E-mail	Bruno Di Leo  bdileo@us.ibm.com
Gerente General de Latino América:  E-mail:	Rodrigo Kede De Freitas Lima  rodrigok@br.ibm.com
Gerente General de Sur América:	Jorge Goulu

E-mail:	goulu@ar.ibm.com
Gerente General de Colombia:  E-mail:	Federico Martinez Gonzalez  fmartine@co.ibm.com
Gerente de área:  E-mail:	Olga Lucia Baquero Gonzalez  (Manager)  obaquero@co.ibm.com
Jefe Directo:  E-mail:	Maurisabed Hernandez  Lopez (Coacher)  (maurisab@co.ibm.com)

## 2. JUSTIFICACIÓN

Como parte del programa y buscando contribuir con la formación integral del estudiante, la práctica académica permite validar los conocimientos adquiridos en las aulas, en el ambiente laboral. Busca brindarle al estudiante un espacio académico para permitirle un excelente desempeño en el campo profesional. Esto le permite al estudiante fortalecer sus habilidades profesionales y a desarrollar nuevas competencias laborales. Preparándolo así para su nueva etapa como Ingeniero Mecatrónico.

IBM es una multinacional con varios años de experiencia en el sector tecnológico. Desde el desarrollo de hardware con la línea de computadores desde la década de los 60, pasando por el middleware y el software, siendo los últimos dos, el enfoque principal de la empresa durante los últimos años. Durante todo este tiempo, los ejecutivos de IBM afirman que la clave del éxito de la compañía es su gente. Reconocidos por otras empresas, como gente talentosa y con gran flexibilidad a la hora de adoptar nuevas estrategias, sus empleados han sido el soporte durante décadas de éxitos, que han hecho de IBM una de las empresas más importantes del panorama mundial.

Es por eso que recursos humanos cuenta con uno de los programas corporativos más exitosos en materia de selección, El Student Program, el cual se implementa en todas las regiones con el fin de incorporar jóvenes

entusiastas y capaces, con ganas de formar parte de una gran empresa, en complemento de su formación académica. Teniendo en cuenta lo anterior, y contando con las aptitudes que adquiere un ingeniero mecatrónico UNAB a lo largo de su formación, IBM Colombia, busca estudiantes multidisciplinarios, que les ayuden a seguir siendo la gran empresa que han sido durante años.

### 3. OBJETIVOS

#### 3.1. OBJETIVO GENERAL

Desarrollar soluciones para la industria de la salud que contribuyan a un mejoramiento en la calidad de servicio.

#### 3.2. OBJETIVOS ESPECÍFICOS

- Diseñar y desarrollar un portal web adaptable, para un consumidor de servicios de salud en una entidad médica.
- Diseñar y desarrollar un portal web adaptable, para un prestador de servicios de salud en una entidad médica.
- Desarrollar un servidor web que permita consumir en línea los datos enviados por uno o más sensores.
- Desarrollar los conocimientos necesarios para el manejo y mantenimiento del software IBM Curam Social Program Management.

## 4. MARCO TEORICO

### 4.1. JavaScript

JavaScript es un lenguaje de programación de alto nivel comúnmente usado en el desarrollo de páginas web, basado en acciones que poseen menos restricciones. Gran parte de su programación está orientada a describir objetos, funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otras [3]. Ha sido estandarizado en la especificación de lenguaje ECMAScript. Fue desarrollado originalmente por Netscape, que comenzó a ofrecerlo como un medio para dinamizar las páginas web.

Proporciona los medios para controlar las ventanas del navegador y el contenido que muestran, programar páginas dinámicas simples, evitar depender del servidor Web para cálculos sencillos, capturar los eventos generados por el usuario y responder a ellos sin salir a Internet, comprobar los datos que el usuario introduce en un formulario antes de enviarlos, comunicarse con el usuario mediante diversos métodos, etc. [4]

JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. El núcleo de JavaScript

puede extenderse para varios propósitos, complementándolo con objetos adicionales, por ejemplo:

- Client-side JavaScript extiende el núcleo del lenguaje proporcionando objetos para controlar un navegador y su modelo de objetos (o DOM, por las iniciales de Document Object Model). Por ejemplo, las extensiones del lado del cliente permiten que una aplicación coloque elementos en un formulario HTML y responda a eventos del usuario, tales como clicks del ratón, ingreso de datos al formulario y navegación de páginas.
- Server-side JavaScript extiende el núcleo del lenguaje proporcionando objetos relevantes a la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comunique con una base de datos, proporcionar continuidad de la información de una invocación de la aplicación a otra, o efectuar manipulación de archivos en un servidor.

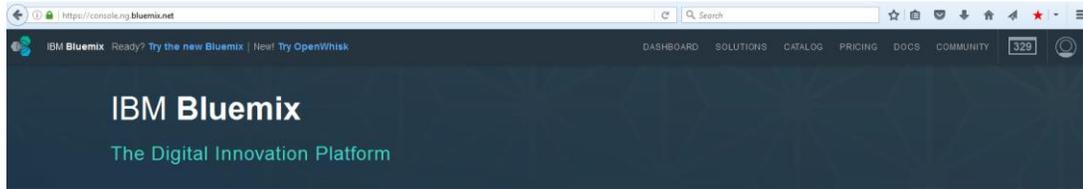
## 4.2. Bluemix

Figura 2. Logo oficial de IBM Bluemix



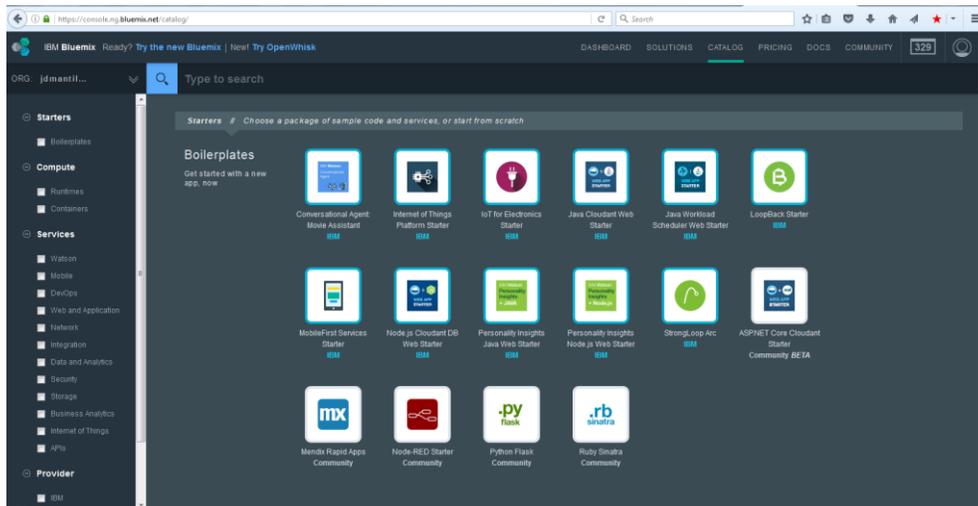
Es una plataforma cloud, basada en open- standards, desarrollado por IBM, para construir, correr, desplegar y administrar aplicaciones de todos los tipos (web, mobile, big data, dispositivos inteligentes). Bluemix e un entorno plataforma como servicio (Paas – Platform as a Service) está basado en Cloud Foundry (estándar para el desarrollo en la nube) y corre en una infraestructura softlayer. De igual manera soporta varios lenguajes de programación como Java, Node.js, Go, PHP, Python, Ruby Sinatra, Ruby on Rails y puede extenderse para que soporte otros lenguajes a través de buildpacks. [5]

Figura 3. Página oficial de IBM Bluemix [6].



Dentro de sus ventajas principales, se encuentra la reducción del tiempo para el desarrollo de la aplicación y para los costes totales de esta, ayuda a hacer frente a la falta de recursos internos de tecnología, está basado en una plataforma en la nube lo que facilita su desarrollo y ejecución, ofrece la infraestructura base para cualquier desarrollo y ofrece un alojamiento fiable para que tus clientes las usen y se pueden ejecutar aplicaciones en múltiples lenguajes. [7]

Figura 4. Catálogo de IBM Bluemix [6].



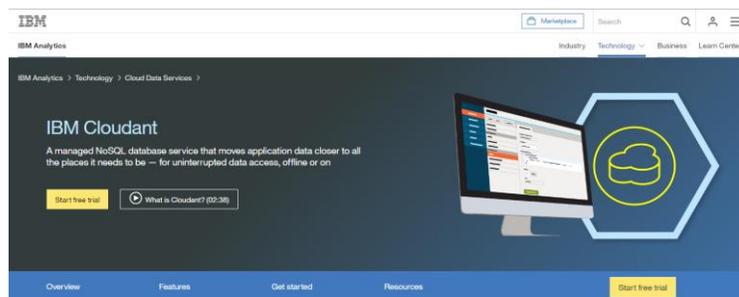
Las empresas y sus desarrolladores pueden construir, gestionar, desplegar y manejar aplicaciones personalizadas desde la nube, usando los sets de software y servicios presentes en la plataforma.

### 4.3. Cloudant

Cloudant es un producto de software de IBM, que se presenta como un servicio soportado en la nube dentro de Bluemix IBM. Es un servicio de base de datos no relacional y de fuente abierta que requiere una mínima configuración. [8]

El servicio de cloudant provee una administración, búsqueda y análisis de datos integrados para el diseño de aplicaciones web. El producto se entrega como una base de datos como servicio (DBaaS) con muchos beneficios como escalabilidad, manejo de datos multi-estructurales y una movilidad de los datos.

Figura 5. IBM Cloudant [9].



#### 4.4. PostgreSQL

Es un sistema de base de datos relacional y de fuente abierta. Tiene más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una fuerte reputación por su confiabilidad, integridad de la información y exactitud. Como base de datos, su función primaria es almacenar la información de manera segura y permitir la recuperación de la misma a solicitud de otras aplicaciones de software. Tiene algunas similitudes con MySQL, en su línea de comandos, pero con una ligera diferencia que hace que se enfoque en la integridad de los datos y no en optimización de queries simples. [10]

Figura 6. Logo PostgreSQL.



Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, algunas de sus principales características son:

- Es una base de datos 100% ACID.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP ...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado via SSL
- SE-postgres
- Completa documentación
- Licencia BSD

- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

#### 4.5. Arduino

Arduino es una compañía de hardware libre y comunidad tecnológica, que diseña y manufactura placas electrónicas para el desarrollo de hardware y software flexibles. Está enfocado principalmente en facilitar el uso de la electrónica y la programación de sistemas embebidos que involucren múltiples disciplinas.

Figura 7. Logo Oficial Arduino.



Arduino consta de una placa principal de componentes eléctricos, donde se encuentran conectados los controladores principales que gestionan los demás complementos y circuitos ensamblados en la misma y un entorno de desarrollo (IDE) basado en el lenguaje de programación Processing [11].

Sus funciones básicamente pueden resumirse en tres, una interfaz de entrada, que es la encargada de unir los periféricos, llevando la información de estos al microcontrolador quien es el encargado de procesar los datos, el microcontrolador, que varía dependiendo de los requerimientos del sistema y la interfaz de salida, que lleva la información procesada a los periféricos encargados de hacer uso de esos datos.

Hardware: está construido por un microcontrolador principal llamado Atmel, que está presente en la mayoría de modelos, es el encargado de realizar los procesos lógicos y matemáticos dentro de la placa , controla y gestiona los recursos de los componentes externos, esta placa cuenta con entradas análogas y digitales que le permiten integrar otros componentes fácilmente, dentro de sus componentes principales, se encuentran también los puestos seriales de entrada y salida permitiendo la comunicación USB [12].

Las características generales de las placas de control Arduino son:

- El microprocesador ATmega328
- 32 kbytes de memoria Flash
- 1 kbyte de memoria RAM

- 16 MHz
- 13 pins para entradas/salidas digitales (programables)
- 5 pins para entradas analógicas
- 6 pins para salidas analógicas (salidas PWM)
- Completamente autónomo: Una vez programado no necesita estar conectado al PC
- Microcontrolador ATmega328
- Voltaje de operación 5V
- Voltaje de entrada (recomendado) 7-12 V
- Voltaje de entrada (limite) 6-20 V
- Digital I/O Pins 14 (con 6 salidas PWM)
- Entradas analógicas Pins 6
- DC corriente I/O Pin 40 mA
- DC corriente 3.3V Pin 50 mA
- Memoria Flash 32 KB (2 KB para el bootloader)
- SRAM 1 KB
- EEPROM 512 byte
- Velocidad de reloj 16 MHz

Software: Su lenguaje de programación se llama wiring y está basado en la plataforma Processing y principalmente en el lenguaje de programación

C/C++, soportando diferentes lenguajes de programación derivados de C como java, Basic, Php, Phytom, JavaScript, entre otros.

#### 4.6. Node-red

Es una herramienta de visualización open-source creada por el equipo de tecnologías emergentes IBM (IBM emerging Technology), que nos permite interconectar elementos como, dispositivos de hardware, los APIs de web, servicios en línea, servicios de internet de las cosas, entre otros.

Es un editor visual de código abierto, basado en nodos que se pueden arrastrar, soltar y unir que nos permite ensamblar rápidamente flujos de servicios, cada nodo ofrece una funcionalidad diferente permitiendo unificar diferentes nodos, para obtener finalmente un sistema operable, dependiendo de la necesidad [13].

Nodo-RED está incluido en la solicitud de arranque de Bluemix, Internet de las cosas (IoT), pero también se puede instalar como aplicación Node.js por separado, el servicio de la IoT permite registrar y conectar diferentes tipos de dispositivos.

Comúnmente se utiliza para definir los flujos de cualquiera de los datos de un sensor de entrada, por ejemplo, almacenar en bases de datos, o cuando se envían comandos a los dispositivos.

Esta construido sobre node.js aprovechando al máximo su modelo orientado a eventos, lo que lo hace ideal para funcionar en el borde de la red del hardware de bajo costo, consta con más de 250000 módulos del repositorio node.js

#### 4.7. HTML 5

Es un lenguaje de marcado usado para estructurar y presentar contenido en la World Wide Web; es la quinta y actual versión del estándar HTML. HTML5 incluye modelos de procesamiento detallados en una búsqueda de implementaciones interoperables. Extiende, mejora y racionaliza el marcado disponible para los documentos e introduce las interfaces de programación de aplicaciones (API) para aplicaciones web más complejas [14].

Figura 8. Logo Oficial HTML.



El marcado HTML consiste en ciertos componentes claves, incluyendo los tags (y sus atributos), caracteres de referencia y entidades de referencia entre otros.

Se trata de un sistema para formatear el layout de nuestras páginas, así como hacer algunos ajustes a su aspecto. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto [15].

#### 4.8. AngularJS

Es un framework JavaScript para desarrollos front end de aplicaciones web de fuente abierta que se puede agregar a un documento HTML mediante el

uso del tag <script>. Angular extiende el uso del HTML añadiéndole directivas y enlaza la información usando expresiones, útil tanto para la vista del cliente como para el servidor [16].

Figura 9. Logo Oficial Angular JS



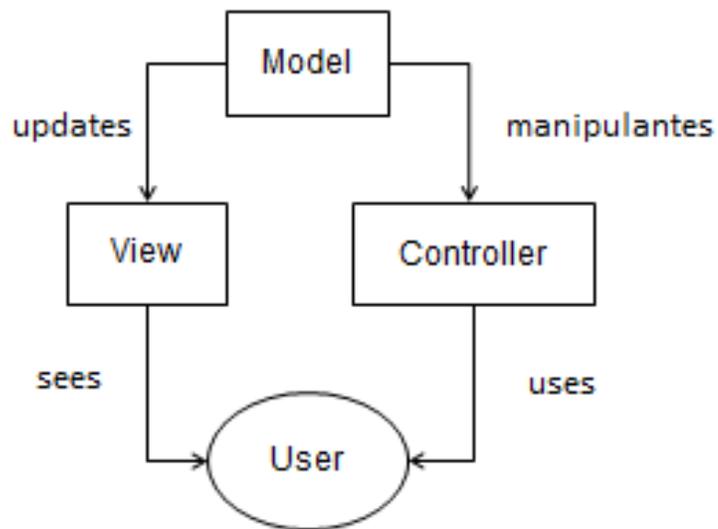
AngularJS permite extender el vocabulario HTML con directivas y atributos, manteniendo la semántica y sin necesidad de emplear librerías externas como jQuery o Underscore.js para que funcione.

Angular sigue el patrón MVC (Model-View-Controller) de ingeniería de software y alienta la articulación flexible entre la presentación, datos y componentes lógicos. Con el uso de la inyección de dependencias, Angular lleva servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente. En consecuencia, gran parte de la carga en el backend se reduce, lo que conlleva a aplicaciones web mucho más ligeras.

#### 4.8.1. MVC

Es una arquitectura de software para implementar interfaces de usuario en computación. Divide la aplicación de software en tres partes interconectadas, de tal manera que se pueda separar las representaciones internas de información, de las formas en que se presenta la información o es aceptada por el usuario [17].

Figura 10. Esquema MVC.



El componente central del MVC, el modelo, captura el comportamiento de la aplicación en términos de su problema de dominio, independiente de la interfaz de usuario.

- El modelo maneja directamente la información, la lógica y las reglas de la aplicación.
- La vista puede ser cualquier representación de una salida de información.
- El controlador, acepta las entradas y las convierte en comandos para el modelo o la vista.

#### 4.9. CSS

Cascading Style Sheets o Hoja de estilo en cascada, es un lenguaje usado para describir la presentación de un documento escrito en lenguaje de marcado (HTML/XML). Este lenguaje se desarrolló con la idea de separar la estructura de un documento y su presentación. Esta separación de formato y contenido, permite presentar la misma página de marcado, en diferentes estilos de diferentes métodos de renderizado, lo que permite lo que se conoce como un diseño web adaptable, independiente del dispositivo que se use [18].

Las especificaciones CSS son mantenidas por el consorcio del world wide web (W3C).

Las ventajas de su uso son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Optimización del ancho de banda de la conexión, pues pueden definirse los mismos estilos para muchos elementos con un sólo selector; o porque un mismo archivo CSS puede servir para una multitud de documentos.
- Mejora en la accesibilidad del documento, pues con el uso del CSS se evitan antiguas prácticas necesarias para el control del diseño, y que iban en perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales.

#### 4.10. SASS

Syntactically Awesome StyleSheets es un lenguaje script que es interpretado en CSS. La implementación oficial de Sass es de fuente abierta y escrita en Ruby. Este se desarrolló teniendo en cuenta que las páginas web hoy en día tienen documentos de estilo más largos, complejos y difíciles de mantener. Es aquí donde un preprocesador como Sass cumple su función, permitiéndole al programador usar características que no se encuentran en CSS, como variables, anidaciones, herencia entre otras [19].

#### 4.11. PHP

Pre Hypertext processor, es un lenguaje script enfocado en el lado del servidor, diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes direccionados al lado del servidor que se podían incorporar directamente en el documento HTML. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. Es considerado uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy [20].

Figura 11. Logo Oficial PHP.



PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Cuando el cliente hace una petición al servidor para que le envíe una página web, el

servidor ejecuta el intérprete de PHP. Este procesa el script solicitado que generara el contenido de manera dinámica (por ejemplo, obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Este permite la conexión a diferentes tipos de servidores de bases de datos SQL como NoSQL.

#### 4.12. Curam IBM

El concepto de Curam parte del idioma irlandés y se traduce como cuidado o protección. Es una solución de IBM diseñada para iniciativas sociales como salud, seguridad social, servicios humanos, entre otros. Esta plataforma está pensada para proveer soluciones adaptables a casos particulares y esta soportada en tecnologías abiertas.

Cúram Solutions ofrece soporte a todo el ciclo de vida en la gestión de los programas sociales y de salud, desde las necesidades hasta los resultados. Con un diseño que gira en torno al cliente, las soluciones Cúram permiten a las organizaciones colaborar para responder a las necesidades de los clientes, proporcionando a los ciudadanos un acceso a los servicios gubernamentales más sencillo y consiguiendo mayor efectividad a la hora de acercar los objetivos sociales y de salud.

## 5. CUADRO DE RESULTADOS

Tabla 1. Cuadro de resultados.

Objetivos	Resultados esperados	Resultados Obtenidos	Anexo Soporte
Adquirir conocimientos de las herramientas IBM y la industria salud.	-Obtener un manejo de la herramienta Bluemix, SPSS y IBM MessageBroker. -Obtener un conocimiento del sistema de salud nacional.	-Desarrollo de talleres de capacitación de las plataformas. -Presentación conceptual sobre interoperabilidad. -Conocimiento del sistema de salud y sus conceptos legales.	Anexo B.
Pruebas de concepto sobre el internet de las cosas (IoT)	-Conectar sensores biométricos y	Desarrollo de un servidor web, que permite el consumo de datos médicos de	Anexo C, Anexo E.

	<p>médicos a una placa integrada.</p> <p>-Enviar la información obtenida a un servicio de base de datos en la nube.</p> <p>-Desarrollar un servidor web para el consumo de los datos.</p>	<p>un sensor, mediante solicitudes HTTP.</p>	
<p>Desarrollo del proyecto “IBM hAppy Health.”</p>	<p>-Desarrollar un backend para el servicio de hAppy Health y hAppy Doc.</p> <p>- Desarrollar un frontend para el servicio de</p>	<p>Despliegue de las plataformas web autoajustables, hAppy Health y hAppy Doc, en su versión demo.</p>	<p>Anexo F, Anexo D.</p>

	hAppy Health y hAppy Doc.		
Pruebas de conceptos, IBM Curam Social Program Management.	Adquirir conocimiento en el manejo de la plataforma Curam.	Implementación del laboratorio de conceptos básicos de Curam.	Anexo G, Anexo H.

## 6. ACTIVIDADES REALIZADAS

Tabla 2. Actividades realizadas.

Actividades	Logros	Anexo Soporte
Desarrollo de un servicio web para consumir datos de los sensores.	<ul style="list-style-type: none"><li>-Conexión de los sensores a la red con ayuda de la placa ethernet.</li><li>-Conexión de los sensores a la base de datos en la nube.</li><li>-Web Server para consumir los datos de los sensores.</li></ul>	Anexo B
Informe Técnico del servicio web	<ul style="list-style-type: none"><li>-Redacción del informe técnico.</li></ul>	Anexo B
Desarrollo Pagina web "hAppy Health"	<ul style="list-style-type: none"><li>-Programación del backend y el frontend autoajutable de la plataforma web.</li><li>-Consumo de los datos obtenidos del servidor web generado previamente.</li></ul>	Anexo C

<p>Desarrollo Pagina web "hAppy Doc"</p>	<p>-Programación del backend y el frontend autoajutable de la plataforma web.</p> <p>-Consumo de los datos obtenidos del servidor web generado previamente.</p>	<p>Anexo D</p>
<p>Manual de presentación: "hAppy Health"</p>	<p>-Redacción del manual.</p> <p>-Pruebas de presentación del instructivo.</p>	<p>Anexo E</p>
<p>Manual de presentación: "hAppy Doc"</p>	<p>-Redacción del manual.</p> <p>-Pruebas de presentación del instructivo.</p>	<p>Anexo F</p>
<p>Implementación del Laboratorio 1 de IBM Curam.</p>	<p>-Entendimiento del programa Rational Software Architect.</p> <p>-Manejo de los servicios de CURAM atraves de Java Eclipse.</p>	<p>Anexo G</p>

	<ul style="list-style-type: none"><li>-Modificación de la base de datos según la necesidad.</li><li>-Desarrollo de frontends Curam usando UML.</li></ul>	
Implementación del Laboratorio 2 de IBM Curam.	<ul style="list-style-type: none"><li>- Creación de un proceso Batch para el medico con Curam.</li></ul>	Anexo H

## CONCLUSIONES

La realización de las practicas académicas nos permitió adquirir nuevos conocimientos y habilidades en el manejo de plataformas y servicios específicos de IBM como lo son IBM Bluemix, IBM Rational Architect Software, IBM Cloudant, IBM WebSphere Message Broker y principalmente IBM curam. Al mismo tiempo les permitió desarrollar habilidades para la realización del backend y frontend de aplicaciones web, con la integración de servicios en la nube, manejo de base de datos y tecnologías electrónicas para el monitoreo remoto.

Permitió a los estudiantes ampliar sus conocimientos sobre temas paralelos a la ingeniería como lo es el sector Salud a nivel Nacional, debido a que IBM "Center of Excellence" en Bucaramanga es un centro de investigación enfocado en desarrollos para este sector.

Como experiencia laboral, contribuyo a los estudiantes a asumir responsabilidades, cumpliendo horarios de entrega, horario laboral y desarrollando actividades específicas dentro de tiempos determinados. Además fortaleció sus habilidades para el trabajo en equipo y el manejo de una comunicación hacia sus compañeros, jefes, gerentes y colaboradores.

## BIBIOGRAFIA

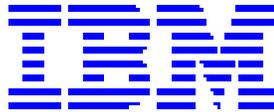
- [1] IBM, «IBM, Colombia,» [En línea]. Available: <https://www.ibm.com/ibm/co/es/>. [Último acceso: 10 6 2016 ].
- [2] IBM, «Conozca IBM, productos y soluciones.,» [En línea]. Available: [http://www.ibm.com/expressadvantage/mx/pdf/Folleto\\_Conozca\\_IBM.pdf](http://www.ibm.com/expressadvantage/mx/pdf/Folleto_Conozca_IBM.pdf). [Último acceso: 2016 22 05].
- [3] D. P. Valdés, «JavaScript,» 03 5 2007. [En línea]. Available: <http://www.maestrosdelweb.com/que-es-javascript/>. [Último acceso: 09 6 2016].
- [4] E. Informatica, «Tutorial JavaScript,» Dpto.L.C.C. U.M.A, [En línea]. Available: <http://www.lcc.uma.es/~eat/services/html-js/manual14.html>. [Último acceso: 27 7 2016].
- [5] IBM developerWorks , «Comience a utilizar IBM Bluemix,» IBM, [En línea]. Available: <http://www.ibm.com/developerworks/ssa/bluemix/index.html>. [Último acceso: 18 6 2016].
- [6] IBM , «IBM Bluemix,» IBM , [En línea]. Available: <https://console.ng.bluemix.net/>. [Último acceso: 21 07 2016].
- [7] I. developerWorks, «Bluemix España, Introduccion.,» IBM , 24 6 2014. [En línea]. Available: [https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/We0d917403ade\\_46b2\\_8991\\_d1eabb8126f6](https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/We0d917403ade_46b2_8991_d1eabb8126f6). [Último acceso: 04 6 2016].
- [8] IBM, «IBM Cloudant,» IBM, [En línea]. Available: <https://www.ibm.com/marketplace/cloud/database-management/us/en-us>. [Último acceso: 13 6 2016].
- [9] IBM, «IBM Cloudant,» [En línea]. Available: <http://www.ibm.com/analytics/us/en/technology/cloud-data-services/cloudant/>. [Último acceso: 19 7 2016].
- [10] M. G. Ginesta, Base de datos en PostgreSQL, UOC edu, 2010.

- [11] Arduino , «Arduino Genuino,» Arduino, [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction#>. [Último acceso: 25 7 2016].
- [12] Weebly, «Arduino: tecnología para todos,» [En línea]. Available: <http://arduinodhtics.weebly.com/iquestqueacute-es.html>. . [Último acceso: 13 6 2016].
- [13] Node-Red, «IBM,» [En línea]. Available: <http://nodered.org/>. [Último acceso: 10 6 2016].
- [14] A. C. Cantón, «Manual de HTML5 en español,» thePROC, [En línea]. Available: [http://php.openwifizone.info/manuales/manual\\_de\\_html\\_5.pdf](http://php.openwifizone.info/manuales/manual_de_html_5.pdf). [Último acceso: 08 7 2016].
- [15] A. Pastorini, «HTML5 y CSS3,» 2010. [En línea]. Available: [http://php.openwifizone.info/manuales/manual\\_de\\_html\\_5.pdf](http://php.openwifizone.info/manuales/manual_de_html_5.pdf). [Último acceso: 10 7 2016].
- [16] Tutorialspoint, «Learn angularjs,» 2014 . [En línea]. Available: [https://www.tutorialspoint.com/angularjs/angularjs\\_tutorial.pdf](https://www.tutorialspoint.com/angularjs/angularjs_tutorial.pdf). [Último acceso: 2016].
- [17] J. P. Mestras, «El patrón MVC,» 2008. [En línea]. Available: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>. [Último acceso: 08 7 2016].
- [18] J. E. Pérez, «CSS avanzado,» 3 2010. [En línea]. Available: [http://www.jesusda.com/docs/ebooks/css\\_avanzado.pdf](http://www.jesusda.com/docs/ebooks/css_avanzado.pdf). [Último acceso: 7 7 2016].
- [19] B. Thesis, Tutorial Sass, Bratislava : Faculty of Mathematics, Physics and Informatics , 2013.
- [20] PHP, «Manual PHP,» the PHP group , 2001. [En línea]. Available: <http://php.net/manual/es/book.pdf.php>. [Último acceso: 14 8 2016 ].

## ANEXOS

Anexo A. Informe técnico del servicio Web.

# Creación de una base de datos Cloudant, utilizando el servicio de Internet of Things en Bluemix.



**Ultimo Cambio al documento:**

18/05/16

**Autor:** Julián David Mantilla Blanco

Katherin Gutiérrez Ávila

DOCUMENTO DE DESARROLLO

Se planteó la conexión de un sensor de pulsos a la plataforma de Watson Internet of Things, para desplegar la información obtenida en la nube para su posterior consumo por parte de servicios móviles, servicios web o Curam. Se utilizó un arduino uno y su ethernet shield para conectarlo a la plataforma bluemix para el manejo de la información obtenida. En el caso de la obtención de datos se utilizó un sensor óptico que permite determinar los beats por minuto del paciente. De igual manera la información obtenida se muestra en un display LCD para que el usuario tenga acceso a ella.

### **16X2 LCD**

Es un módulo de display electrónico con un amplio rango de aplicaciones. Un LCD 16X2, significa que puede mostrar 16 caracteres por línea y cuenta con dos de estas líneas. En este LCD, cada carácter es desplegado en una matriz de pixeles de 5x7. Cuenta con dos registros, Command y Data.

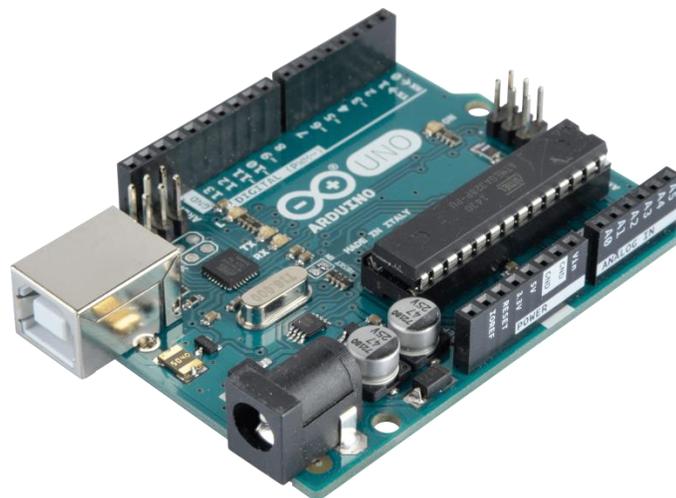
El registro command almacena las instrucciones dadas al LCD. Un command es una instrucción dada al LCD para realizar una tarea predefinida como inicializar, limpiar la pantalla, determinar la posición del cursor, controlar el display etc. El registro data es el valor ASCII del carácter a desplegar en el LCD.

Arduino cuenta con la librería LiquidCrystal que permite controlar LCD's compatibles.



## Arduino UNO

El arduino UNO es un microcontrolador basado en el ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usadas como salidas de PWM), 6 entradas análogas, un cristal de 16MHz, una conexión USB, un cabezal ICSP y un botón de reset. Este se programa utilizando el IDE de Arduino.



## Arduino Ethernet Shield

Este es un shield que permite conectar nuestro Arduino a internet. Se conecta mediante un cable RJ45. Está basado en un chip ethernet Wiznet W5100. El Wiznet provee una red (IP) capaz de manejar TCP y UDP. El Arduino IDE, cuenta con su librería para manejar el shield. Cuenta de igual manera un espacio para manejar una micro-SD.



## Pulse Sensor

El sensor, es un sensor óptico de ritmo cardiaco, que utiliza un método denominado fotopletismografía para medir en este caso, los pulsos por minuto (beats per minute - BPM). La pletismografía es un método que se utiliza para medir cambios de volumen en diferentes partes del cuerpo. En el caso de la fotopletismografía, se hace por medio de un diodo que emite haces

de luz, y un foto-sensor que, según el haz de luz acentuado, determina la expansión en la parte del organismo examinado, expansión que se presenta por las contracciones cardiacas.



### **IBM® Cloudant® NoSQL Database for Bluemix®**

Es una base de datos NoSQL como servicio (DBaaS). Esta maneja una gran variedad de tipos de datos como JSON, texto completo y geoespaciales. Es un almacén de datos operativo optimizado para manejar lecturas simultáneas y registro, y proporcionar una alta disponibilidad y durabilidad de los datos.

### **Node-RED**

Es un editor de flujo que facilita la conexión de dispositivos, API's y servicios online, mediante la conexión de una gran variedad de nodos. Estos flujos se pueden ejecutar en Node.js con un click. Esta versión, adaptada para trabajar

con Bluemix, lo que permite agregarle al flujo una gran variedad de servicios como el “IBM Watson Internet of Things Platform.”

## **Bluemix**

Es uno de los últimos servicios en la nube de IBM. Permite que organizaciones y desarrolladores, crear, desplegar y manipular rápidamente aplicaciones en la nube. Es una plataforma como servicio (PaaS), que entrega al usuario, servicios que pueden integrarse fácilmente a sus aplicaciones, sin necesidad de saber cómo instalarlos o configurarlos.

Empezamos creando una app en Bluemix con el boilerplate de Internet de las cosas (Internet of Things Platform Starter):



Internet of Things  
Platform Starter

IBM

Agregamos el nombre de nuestra aplicación y le damos crear, cuando se despliegue el servicio, volvemos al Dashboard de nuestro boilerplate y

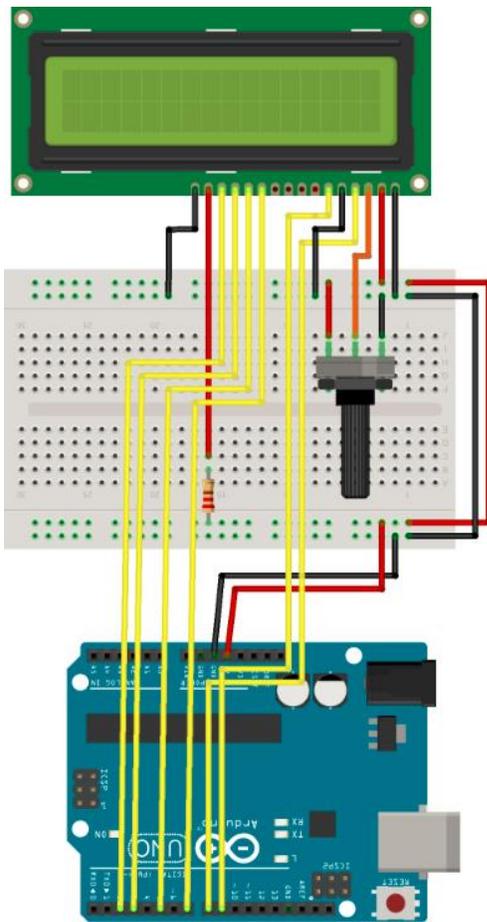
agregamos un servicio de internet de las cosas (Internet of Things Platform),  
le asignamos nombre y le damos crear.



Internet of Things  
Platform

IBM

Circuito de conexión con pantalla LCD y arduino:



Código arduino para comprobar el funcionamiento de la pantalla LCD

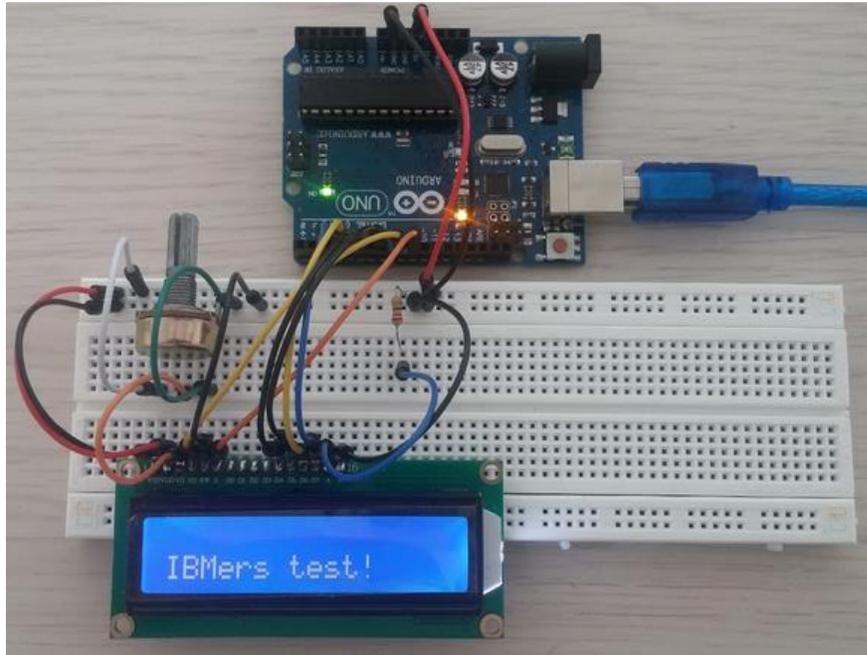
Código arduino: pantalla\_LCD

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 8, 9, 10, 11 , 12);

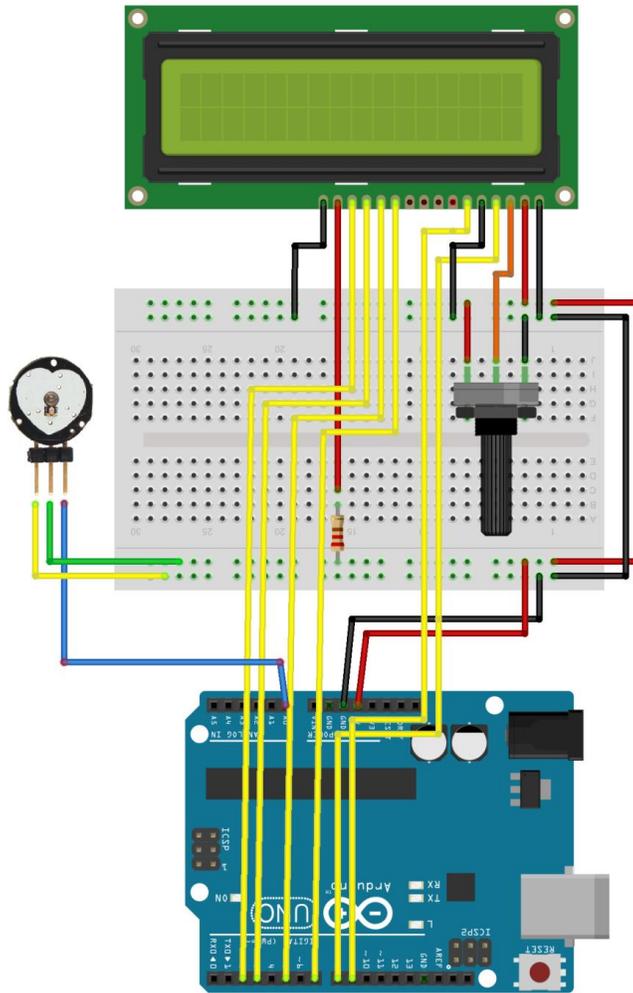
void setup() {
  lcd.begin(16, 2);
  lcd.setCursor(0,1);
  lcd.write("IBMers test!"); // Mensaje de la pantalla
}

void loop() {}
```

Después de compilar el código (código arduino: pantalla\_LCD) y validar la correcta conexión del sistema, se debe observar en la pantalla el siguiente mensaje:



Circuito de conexión con arduino y el sensor de pulso:



Código arduino para comprobar el funcionamiento del sensor

```
Código arduino: sensor
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 5, 7, 3, 2);

// VARIABLES
int pulsePin = 0;           // Señal del sensor de pulso en el pin A0
int blinkPin = 13;         // parpadea el led 13 en cada larido
```

```

int fadePin = 8;
int fadeRate = 0;

// variables utilizadas en el servicio de interrupcion
volatile int BPM;           // mantener la frecuencia del pulso
volatile int Signal;       // mantiene los datos entrantes
volatile int IBI = 600;    // determina el tiempo entre cada pulsacion
volatile boolean Pulse = false;
volatile boolean QS = false;

byte heart[8] =
{
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000
};

void setup(){
  lcd.createChar(1, heart);
  lcd.begin(16, 2);

  pinMode(blinkPin,OUTPUT); // pin que parpadea con los latidos del corazón
  pinMode(fadePin,OUTPUT);  // pin que se apaga con los latidos del corazón
  Serial.begin(115200);     // velocidad de transmisión
  interruptSetup();        // leer la señal de pulso cada 2mS

  pinMode(0,INPUT);
}

void loop(){
  lcd.clear();
  lcd.setCursor(0,0);

  Signal = analogRead(0); //inicialización de la pantalla LCD
  lcd.write("Heart Rate: "); // mensaje de la pantalla
  lcd.setCursor(0,1);
  lcd.print(BPM);
  lcd.print(" Per minute");
  lcd.write(1);

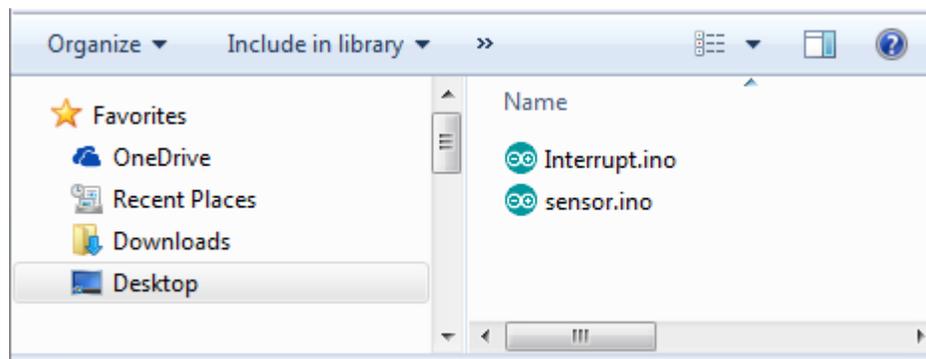
  if (QS == true){
    fadeRate = 255;
  }
}

```

```
    QS = false;
  }
  ledFadeToBeat();
  delay(20);
}

void ledFadeToBeat(){
  fadeRate -= 15;           // enviar al Led Fade el valor
  fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into negative
  numbers!
  analogWrite(fadePin,fadeRate);    // fade LED
}
```

Para que el código compile es necesario agregar en la misma carpeta donde se ubica el código del sensor, un archivo.ino de interrupción como se muestra en la siguiente imagen:



Y dentro de este archivo debe ir el siguiente código:

```
Codigo arduino: interrupt
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 512;
volatile int T = 512;
volatile int thresh = 525;
volatile int amp = 100;
```

```

volatile boolean firstBeat = true;
volatile boolean secondBeat = false;
void interruptSetup(){
TCCR2A = 0x02;
TCCR2B = 0x06;
OCR2A = 0X7C;
TIMSK2 = 0x02;
sei();
}
ISR(TIMER2_COMPA_vect){
cli();
Signal = analogRead(pulsePin);
sampleCounter += 2;
int N = sampleCounter - lastBeatTime;

if(Signal < thresh && N > (IBI/5)*3){
if (Signal < T){
T = Signal;
}
}

if(Signal > thresh && Signal > P){
P = Signal;
}

if (N > 250){
if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
Pulse = true;
digitalWrite(blinkPin,HIGH);
IBI = sampleCounter - lastBeatTime;
lastBeatTime = sampleCounter;

if(secondBeat){
secondBeat = false;

for(int i=0; i<=9; i++){
rate[i] = IBI;
}
}

if(firstBeat){
firstBeat = false;
secondBeat = true;
sei();
return;
}
}
}

```

```

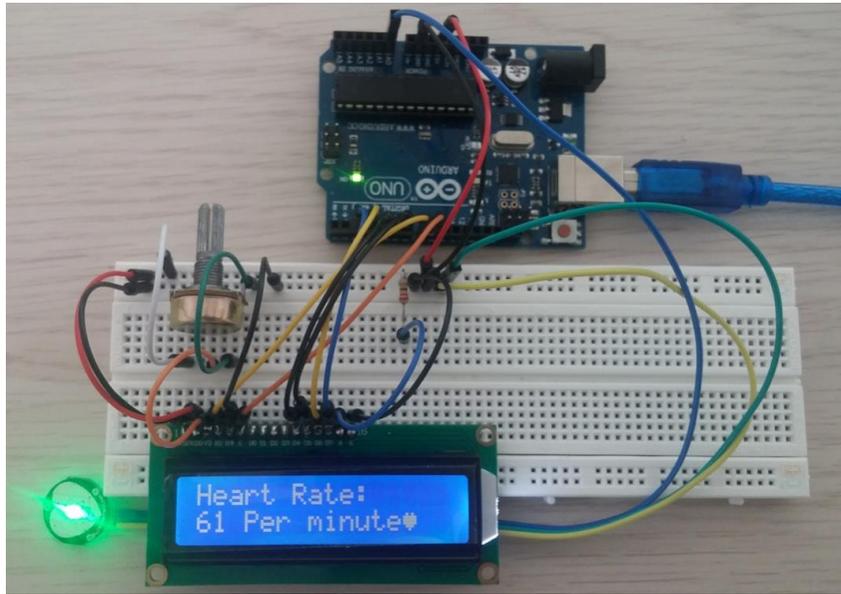
}
word runningTotal = 0;
for(int i=0; i<=8; i++){
rate[i] = rate[i+1];
runningTotal += rate[i];
}
rate[9] = IBI;
runningTotal += rate[9];
runningTotal /= 10;
BPM = 60000/runningTotal;
QS = true;
}
}

if (Signal < thresh && Pulse == true){
digitalWrite(blinkPin,LOW);
Pulse = false;
amp = P - T;
thresh = amp/2 + T;
P = thresh;
T = thresh;
}

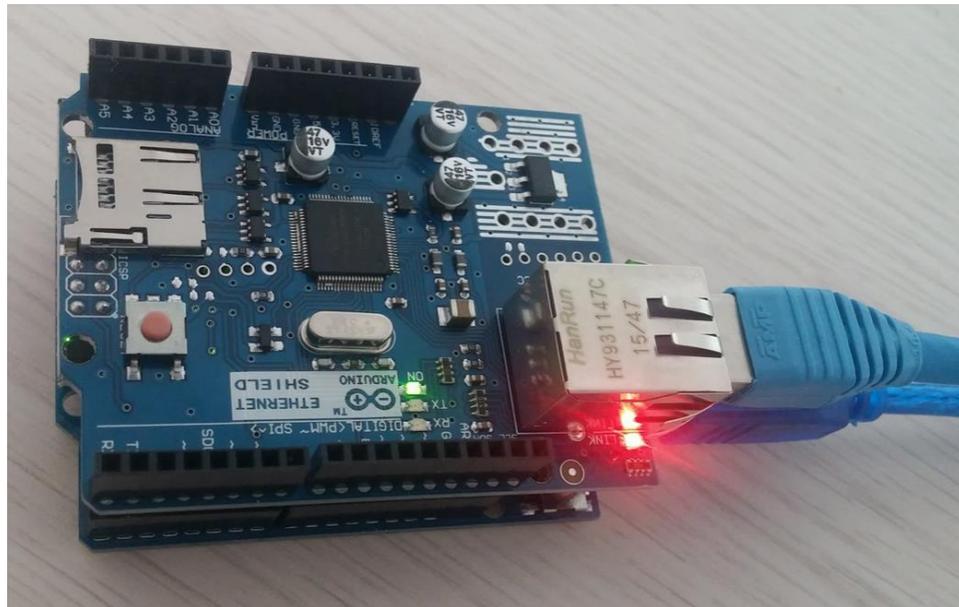
if (N > 2500){
thresh = 512;
P = 512;
T = 512;
lastBeatTime = sampleCounter;
firstBeat = true;
secondBeat = false;
}
sei();
}

```

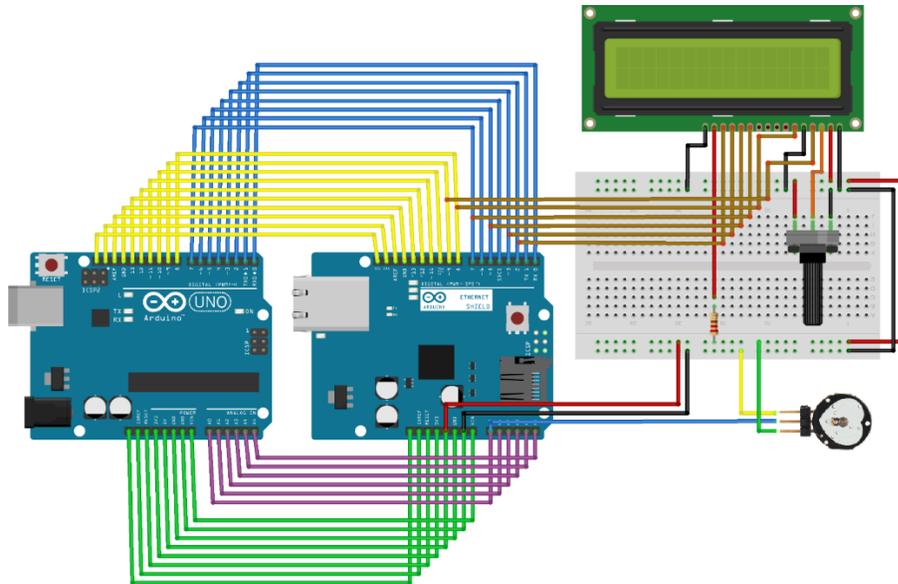
Después de compilar el código inicial (código arduino: sensor) y validar la correcta conexión del sistema, se debe observar en la pantalla LCD el valor ofrecido por el sensor:



Conexión con arduino y el shield ethernet:



Circuito general:



#### Codigo arduino: CodigoCompleto

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include <Ethernet.h>
#include <IPStack.h>
#include <Countdown.h>
#include <MQTTClient.h>
#include <ctype.h>
#define MS_PROXY "quickstart.messaging.internetofthings.ibmcloud.com"
#define MQTT_PORT 1883
#define MQTT_MAX_PACKET_SIZE 100

byte mac[] = { 0x00, 0xFF, 0xAA, 0xCC, 0xDE, 0x02 }; // mac del dispositivo

byte heart[8] =
{
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000
};
#define MQTT_CLIENTID "d:quickstart:iotsample-arduino:00ffaaccde02"
```

```

#define MQTT_TOPIC "iot-2/evt/status/fmt/json"
#define AUTHMETHOD "use-token-auth"
#define SUBSCRIBE_TOPIC "iot-2/cmd/+/fmt/json"
#define AUTHTOKEN "TOKEN" //se configure en token del dispositivo

LiquidCrystal lcd(8, 9, 5, 7, 3, 2);
EthernetClient c;
IPStack ipstack(c);

MQTT::Client<IPStack, Countdown, 100, 1> client = MQTT::Client<IPStack, Countdown,
100, 1>(ipstack);

// VARIABLES
String deviceEvent;
int pulsePin = 0;
int blinkPin = 13;
int fadePin = 8;
int fadeRate = 0;
// these variables are volatile because they are used during the interrupt service routine!
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;

//void messageArrived(MQTT::MessageData& md);

void setup(){
  lcd.createChar(1, heart);
  lcd.begin(16, 2);

  pinMode(blinkPin,OUTPUT);
  pinMode(fadePin,OUTPUT);
  Serial.begin(115200);
  Ethernet.begin(mac);
  delay(2000);
  interruptSetup();
  pinMode(0,INPUT);
}

void loop(){
  lcd.clear();
  lcd.setCursor(0,0);

  Signal = analogRead(0);
  lcd.write("Heart Rate: ");
  lcd.setCursor(0,1);
  lcd.print(BPM);
  lcd.print(" ");
}

```

```

lcd.write(1);

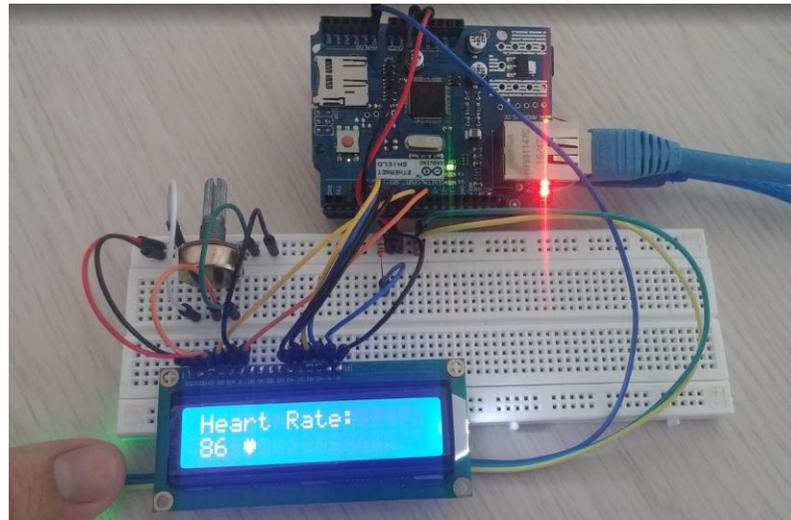
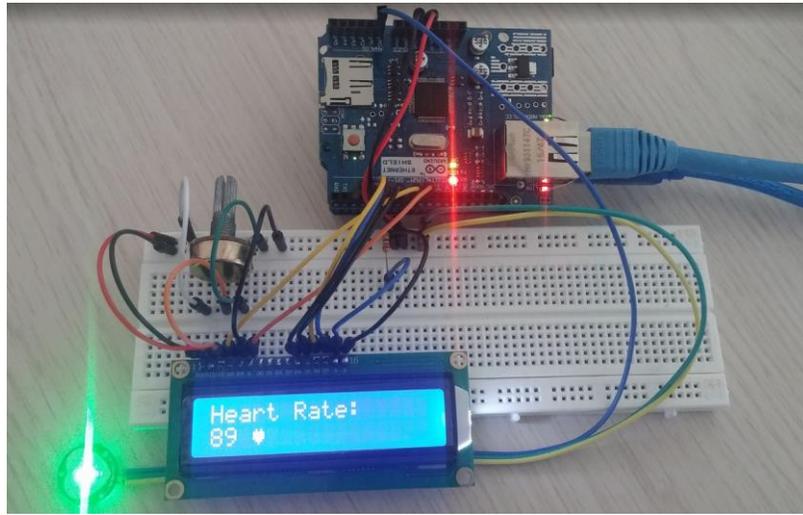
int rc = -1;
if (!client.isConnected()) {
  Serial.println("Connecting to IoT Foundation for publishing Beats Per Minute");
  while (rc != 0) {
    rc = ipstack.connect(MS_PROXY, MQTT_PORT);
  }
  MQTTPacket_connectData data = MQTTPacket_connectData_initializer;
  data.MQTTVersion = 3;
  data.clientID.cstring = MQTT_CLIENTID;

  rc = -1;
  while ((rc = client.connect(data)) != 0);
  Serial.println("Temperature(in C)\tDevice Event (JSON)");
  Serial.println("_____ }

if (QS == true){
  MQTT::Message message;
  message.qos = MQTT::QOS0;
  message.retained = false;
  deviceEvent = String("{\"d\":{\"myName\":\"Arduino Uno\",\"BPM\":"); // salida en
format JSON
  char buffer[60];
  dtostrf(BPM,5,1, buffer);
  deviceEvent += buffer;
  deviceEvent += "}}";
  Serial.print("\t");
  Serial.print(buffer);
  Serial.print("\t\t");
  deviceEvent.toCharArray(buffer, 60);
  Serial.println(buffer);
  message.payload = buffer;
  message.payloadlen = strlen(buffer);
  rc = client.publish(MQTT_TOPIC, message); // envia el mensaje

  if (rc != 0) {
    Serial.print("return code from publish was ");
    Serial.println(rc);
  }
  QS = false;
}
}
delay(20);
}

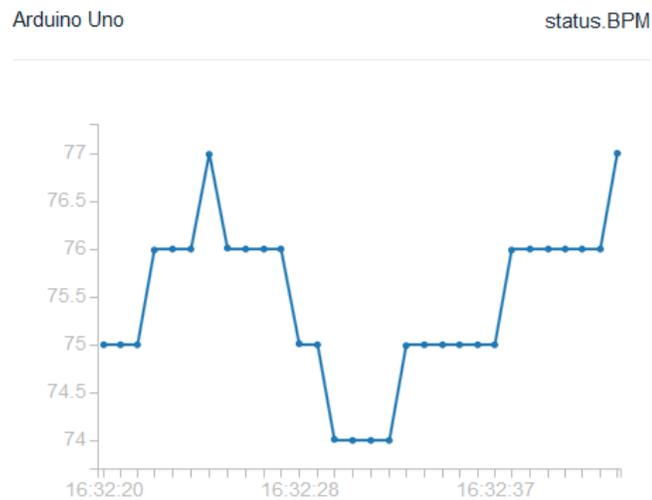
```



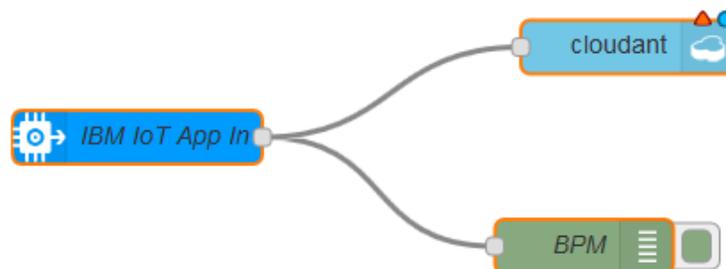
Ingresamos a <http://quickstart.internetofthings.ibmcloud.com/#/> y agregamos el MAC del código del arduino (código arduino:CodigoCompleto).



Damos click en go y debe aparecer la información del sensor graficada:



Luego ingresamos a Node-Red y configuramos los nodos que necesitamos para que reciba la información y la almacene en la nube:



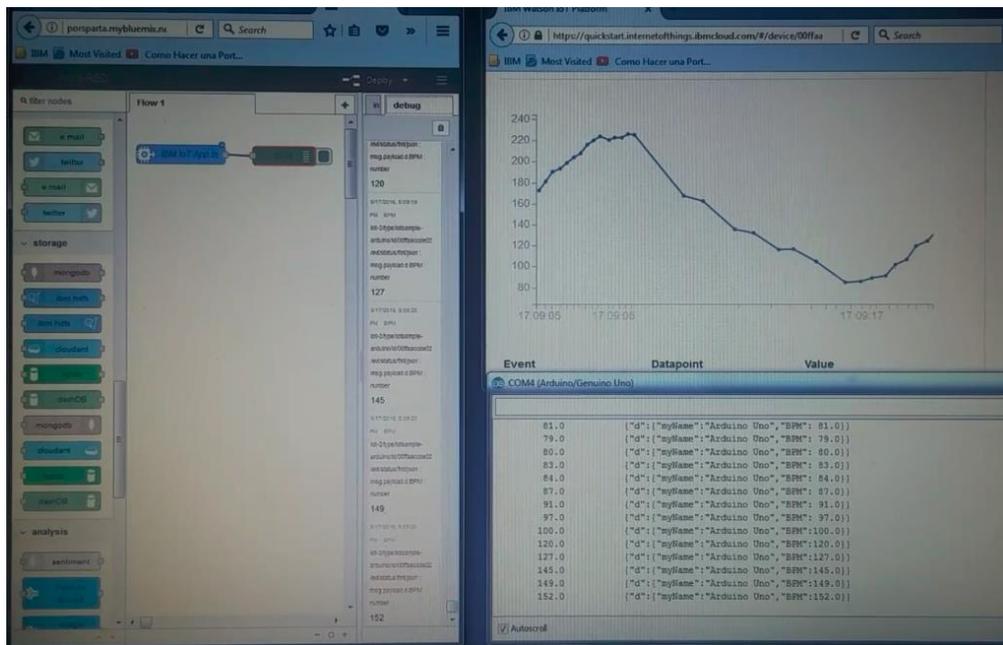
Mientras que en el Debug podremos ir viendo de igual manera la información mostrada por el sensor y que esta siendo registrada en la base de datos.

```
5/18/2016, 4:38:19 PM BPM
iot-2/type/iotsample-arduino/id/00ffaacode02/evt/status/fmt/json : msg.payload.d.BPM : number
80

5/18/2016, 4:38:20 PM BPM
iot-2/type/iotsample-arduino/id/00ffaacode02/evt/status/fmt/json : msg.payload.d.BPM : number
80

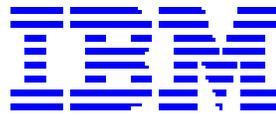
5/18/2016, 4:38:21 PM BPM
iot-2/type/iotsample-arduino/id/00ffaacode02/evt/status/fmt/json : msg.payload.d.BPM : number
80
```

Simultáneamente en el monitor serial del Arduino IDE, se despliega la información tomada y la entrega en el formato que se le dio en el código.



Anexo B. Desarrollo de la Pagina web "hAppy Health".

# Creación de una Pagina Web “hAppy Health”.



**Ultimo Cambio al documento:** 24/08/16

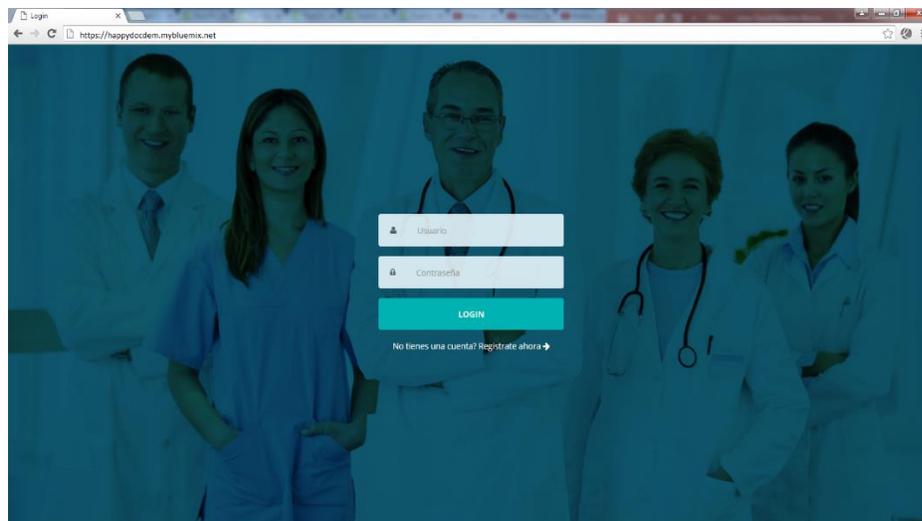
**Autor:** Julián David Mantilla Blanco

Katherin Gutiérrez Ávila

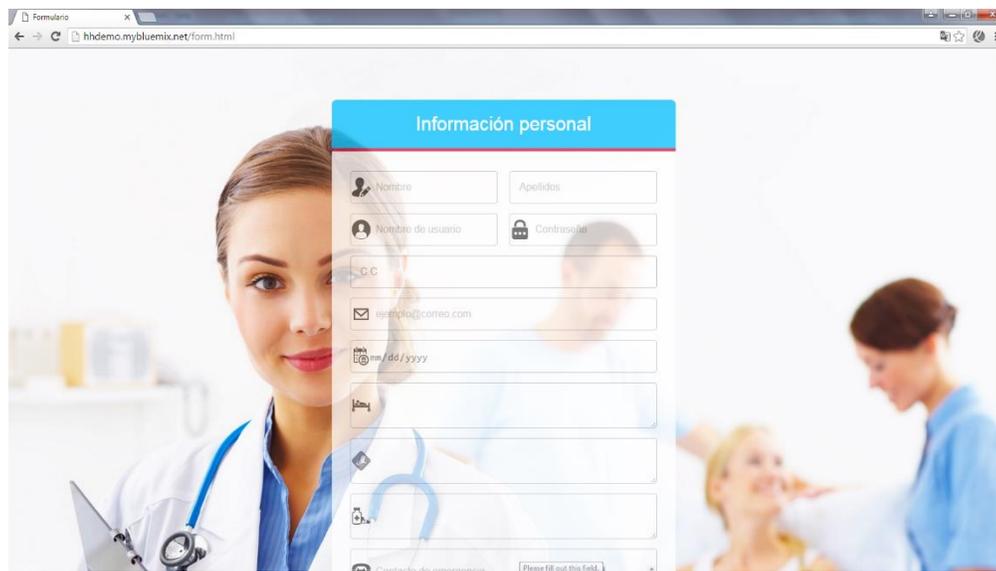
DOCUMENTO DE DESARROLLO

El objetivo de esta iniciativa dentro de IBM es aprovechar el auge que ha tenido la telemedicina en los últimos años, para la cual se proyecta una gran inversión por parte de los servicios de salud, todo con el propósito de mejorar la experiencia para los pacientes. Teniendo en cuenta la petición de uno de sus clientes, se planteó la presentación del proyecto, el cual se puede encontrar en línea en: [hhdemo.mybluemix.net](https://hhdemo.mybluemix.net)

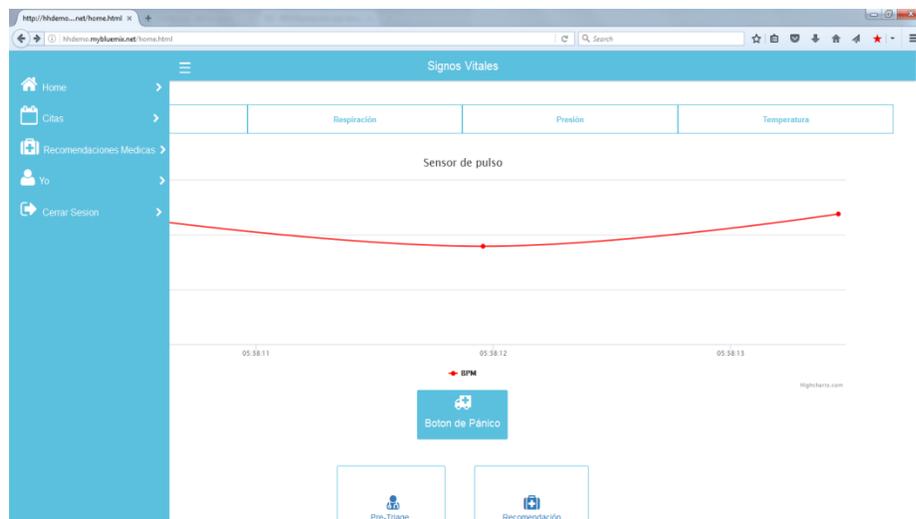
Inicialmente se desarrolló un Login para la autenticación del usuario (paciente) en la página web.



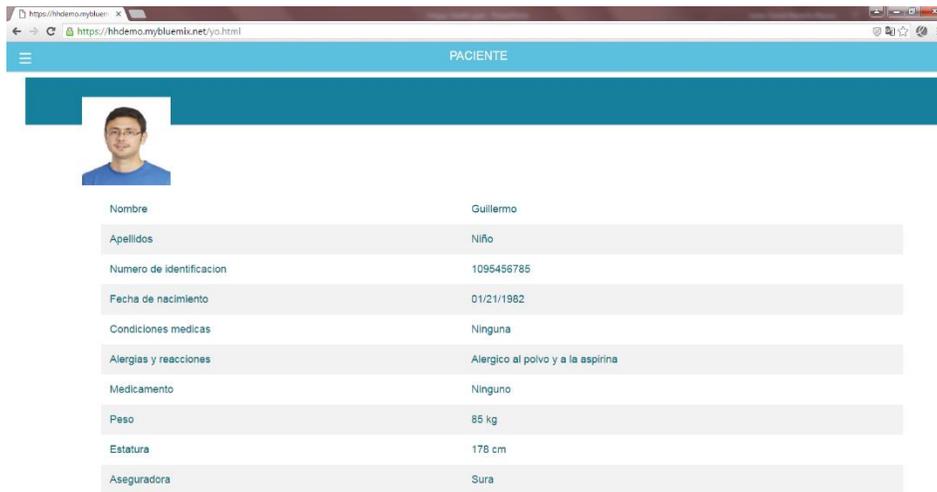
La información del usuario es previamente almacenada en la base de datos por medio del formulario de registro, como se observa a continuación:



La vista principal muestra los datos de los sensores adquiridos previamente por la placa integrada y los sensores biométricos y médicos que se encuentren a disposición, esta vista permite, además, realizar una llamada a emergencias (123) y el re direccionamiento a otras vistas dentro de la aplicación.



El menú desplegable permite navegar por la aplicación de manera intuitiva y amigable, podemos observar la información personal diligenciada en el registro.



Podemos realizar un Pre triage de urticaria,



Hola soy tu Asistente happyHealth, te puedo ayudar revisando los síntomas que tienes para priorizar tu atención.



Escibe sólo uno de los cuatro estados ¿Qué tipo de lesión cutánea tienes?--Lesión Crítica--Lesión Grave--Lesión media--Lesión Leve--

Escribe una respuesta y presiona enter.

Tabla valoración de la actividad y la gravedad de la urticaria.

Suma el valor de la lesión urticarial más la intensidad de la molestia durante las primeras 24 horas, lo máximo es 6.

Valor	Lesiones urticariales	Intensidad de la molestia	Valor
0	Ninguna	Nada	0
1	Menos de 20	Leve	1
2	Entre 21 y 50.	Moderado	2
3	Más de 50 o lesiones grandes confluentes.	Intenso	3

Pre-triage

dialogue-trial.mybluemix.net

Pre Triage Urticaria



# happyHealth

Hola soy tu Asistente happyHealth, te puedo ayudar revisando los síntomas que tienes para priorizar tu atención.

Escibe solo uno de los cuatro estados ¿Qué tipo de lesión cutánea tienes?--Lesión Crítica--Lesión Grave--Lesión media--Lesión Leve--

media

¿Cuáles son los síntomas que presenta? Són 4 síntomas y los más relevantes.

hinchazon, ardor, fiebre, picazon

Tabla valoración de la actividad y la gravedad de la urticaria.

Sume el valor de la lesión urticarial más la intensidad de la molestia durante las primeras 24 horas, lo máximo es 6.

ValorLesiones urticariales	Intensidad de la molestia	Valor
0 Ninguna	Nada	0
1 Menos de 20	Leve	1
2 Entre 21 y 50.	Moderado	2
3 Más de 50 o lesiones grandes confluentes.	Intenso	3

Puedo observar las recomendaciones médicas otorgadas por los médicos

https://hhdemo.mybluemix.net/recomendacion.html

## Recomendaciones Medicas



Dr. Sebastian Ruiz Ayala

Realizar actividades físicas mínimo tres veces en la semana, consumir mucho líquido (2 lt diarios) y una buena cantidad de frutas y verduras durante el día, realizar 5 o 6 ingestas de comida en el día.



Dr. Alonso Rueda

Tomar suficiente líquido durante el día, realizar actividades deportivas terminando siempre con el estiramiento, realizar 20 minutos de caminatas diarias.

Finalmente se puede agendar una cita con un doctor, se realiza la búsqueda del doctor,

Busqueda de paciente x

https://hhdemo.mybluemix.net/busqueda.html

BUSQUEDA DE PACIENTE

Encuentre un paciente

Alberto Castañeda

Sebastian Ruiz Ayala

María Isabel Rueda

Ana Silvia Ayala

Nicolas Agudelo

y se agenda la cita en la fecha y hora deseada.

hhdemo.mybluemix.net/citas.html

Citas

Dr. Sebastian Ruiz Ayala

Nació en Bogotá - Colombia el 16 de Diciembre de 1984 es un médico neurofisiólogo. Se graduó como médico cirujano de la Pontificia Universidad Javeriana. Actualmente es catedrático de neurociencia en la escuela de medicina de la Universidad de Nueva York en la que es además director del departamento de Physiology & Neuroscience en la escuela de medicina de la Universidad.

Seleccione el centro Médico

Clinica Chicamocha

Seleccione la fecha de la cita

Martes Agosto 16, 2016

2016 Agosto

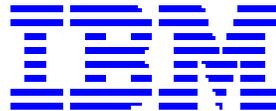
Dom	Lun	Mar	Mié	Jue	Vie	Sáb
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

10:00

- 10:00 am + am pm

Anexo C. Desarrollo de Pagina web "hAppy Doc".

# Creación de una Pagina Web “hAppy Doc”.



**Ultimo Cambio al documento:** 24/08/16

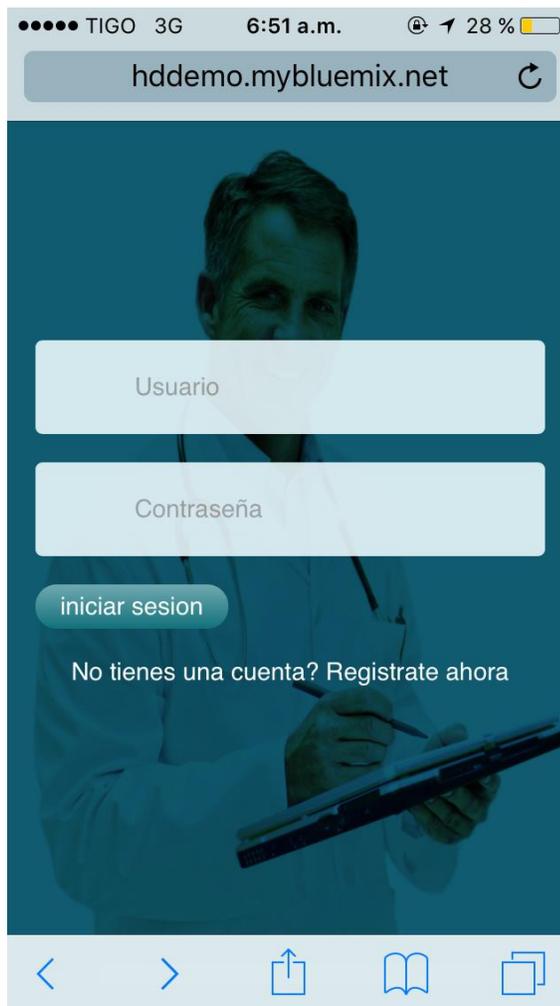
**Autor:** Julián David Mantilla Blanco

Katherin Gutiérrez Ávila

DOCUMENTO DE DESARROLLO

En el caso del doctor también se cuenta con un servicio para el, y se puede encontrar en: [hddemo.mybluemix.net](http://hddemo.mybluemix.net)

Inicialmente se desarrolló un Login para la autenticación del usuario (Doctor) en la página web.



La información del usuario es previamente almacenada en la base de datos por medio del formulario de registro.

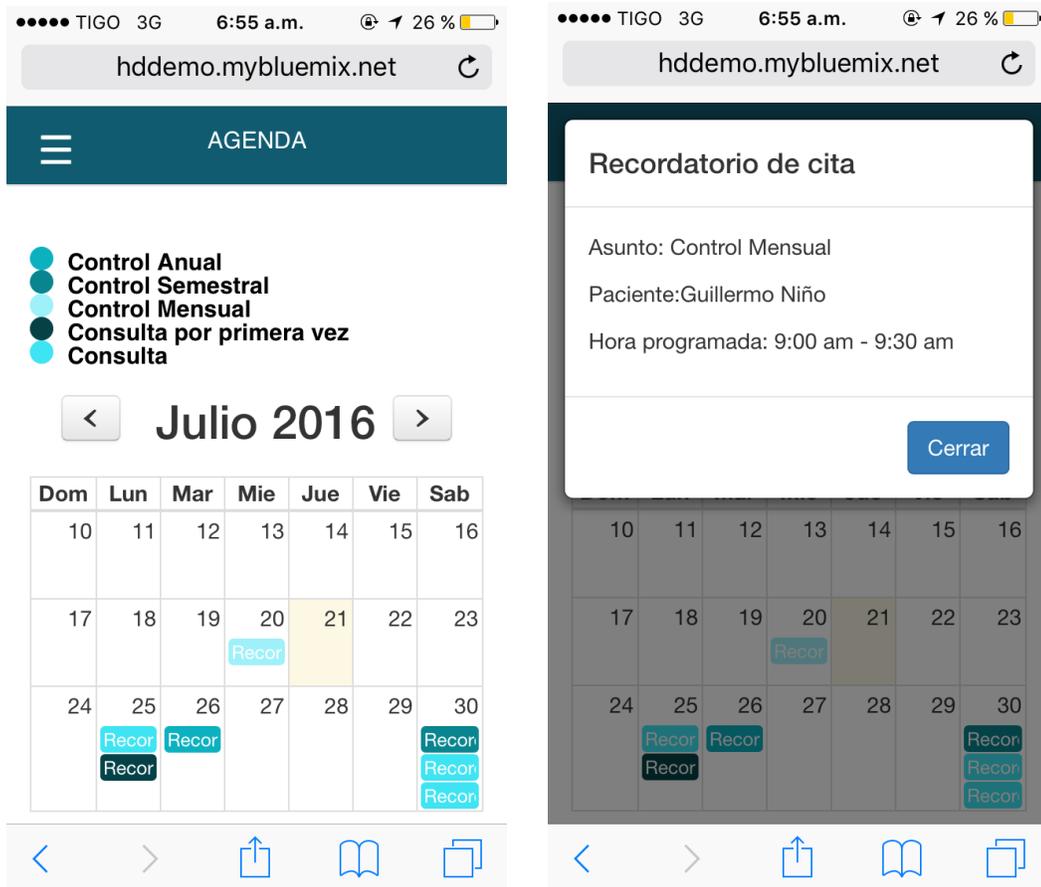
La vista principal muestra la información personal



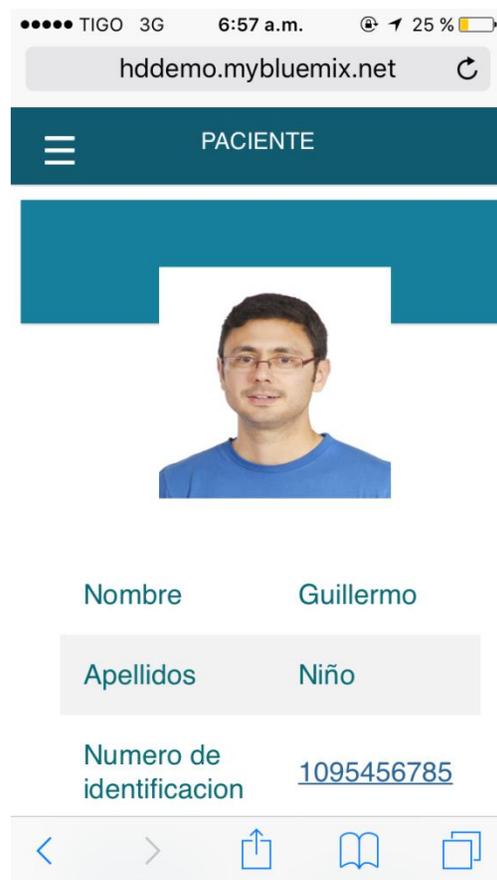
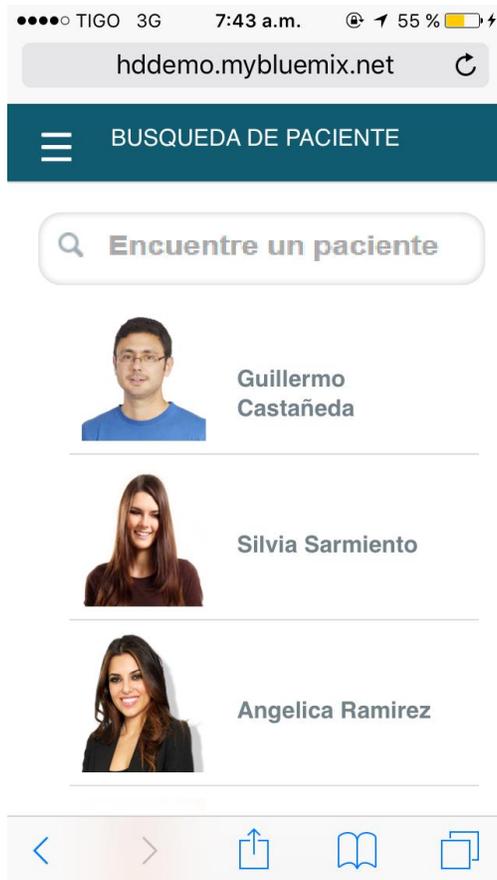
## Biografía

Nació en Bogotá - Colombia el 16 de Diciembre de 1984 es un médico neurofisiólogo. Se graduó como médico

El menú desplegable permite navegar por la aplicación de manera intuitiva y amigable, podemos observar la agenda del médico, con sus citas previamente agendadas.



Finalmente, el medico puede realizar una búsqueda de los pacientes, observar su información personal,



tener acceso al registro de sus sensores y realizar una recomendación partiendo de los mismos.

●●●● TIGO 3G 7:01 a.m. 23 %

hddemo.mybluemix.net

Medicamento	Ninguno
Peso	85 kg
Estatura	178 cm
Aseguradora	Sura

Historial de datos

Agregar Recomendacion

Agrega aqui una recomenda

< >   

●●●● TIGO 3G 7:00 a.m. 23 %

hddemo.mybluemix.net

SIGNOS VITALES

Pulso Respiración Presión Temperatur

Sensor de temperatura



°C	2016-07-21 07:00:09	37.00
----	---------------------	-------

< >   

Anexo D. Manual de presentacion "hAppy Health".

## **hAppy Health**

Demo Story Board

### **BACKGROUND INFORMATION**

#### **Roles:**

**El cliente**– Guillermo Niño es un hombre de 31 años que está registrado en la base de datos de hAppy Health. Últimamente ha sentido algunas molestias de salud por lo que decide ingresar y utilizar los servicios disponibles para el en hAppy Health.

**El doctor** – El Dr. Sebastian Ruiz, quien es uno de los especialistas encargados de la salud de Guillermo. Él puede constantemente verificar los datos médicos y recomendaciones que sus colegas le han hecho a Guillermo. Teniendo en cuenta esto y en su propio criterio médico, puede realizar recomendaciones de salud al paciente o verificar si tiene alguna cita cercana programada con él.

El doctor recoge toda la información disponible sobre el paciente, lo que le permite ofrecerle un mejor cuidado a su salud, que se adapte a cada paciente en particular según sus condiciones.

**Ciente nuevo** – Juan Carlos Machado conoce a Guillermo hace algún tiempo y ha visto como él tiene un mejor control sobre su salud desde su computador o su celular, por lo que decide registrarse y empezar a usar el servicio.

---

Guillermo Niño (31 años) lleva algunos días con una molestia en su brazo, acompañado de fiebre ocasional. Guillermo decide ingresar a hAppy Health para ver si puede encontrar algo que le ayude con la sintomatología que ha presentado en los últimos días.

Dentro de hAppy Health, Guillermo, entre otras cosas, encuentra una herramienta que le podría ayudar a determinar qué es lo que necesita para su salud según sus síntomas. Luego de realizar el proceso y según las recomendaciones del sistema, decide buscar en la lista de médicos disponibles y sacar una cita para seguir correctamente un tratamiento.

### Demo Steps

Step Name	Steps	Screens	Story/Notes
Login	<ul style="list-style-type: none"><li>El paciente ingresa desde su navegador web a hAppy</li></ul>	Screens: <ul style="list-style-type: none"><li>Página de acceso a hAppy Health</li></ul>	<i>Guillermo Niño, quien ha tenido fiebre en días pasados y siente</i>

	<p>health, donde ingresa sus datos registrados previamente.</p>	<p>(navegador web)</p>	<p><i>una extraña picazón en su brazo, quiere ingresar al sitio web de hAppy Health para ver sus datos como paciente y ver si puede conseguir una cita con algún doctor que pueda tratar sus síntomas.</i></p>
<p><b>Inicio</b></p>	<ul style="list-style-type: none"> <li>• El paciente ingresa al inicio donde puede observar los valores de su último registro de BPM's y</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>• Vista principal – inicio del paciente</li> </ul>	<p><i>Inicialmente Guillermo tendrá acceso a su último registro de BPM's y temperatura. Igualmente</i></p>

	<p>temperatura, al mismo tiempo cuenta con una lista de botones que le facilitaran la navegación en el sitio web.</p>		<p><i>cuenta con un botón de pánico, que hará un llamado a las líneas de emergencias en caso de ser necesario. Haciendo uso del menu, el paciente accede a los demás servicios de hAppy Health.</i></p>
<p><b>Información del paciente</b></p>	<ul style="list-style-type: none"> <li>• El paciente ingresa a la vista "Yo" donde puede ver toda su información y verificar su estado.</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>• Vista de información del paciente</li> </ul>	<p>En la sección "Yo", Guillermo puede ver su información básica como paciente,</p>

			<p>incluso la aseguradora a la cual está afiliado.</p>
<p><b>Diagnostico</b></p> <p><b>Urticaria</b></p>	<ul style="list-style-type: none"> <li>• El paciente puede ingresar y realizar un pretriage de urticaria para evaluar la gravedad de sus síntomas y cómo reaccionar ante estos.</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>• Vista de pretriage urticaria</li> </ul>	<p><i>Teniendo en cuenta sus síntomas, Guillermo decide utilizar el asistente de hAppy Health que le permite determinar si sus síntomas corresponden a un caso de urticaria y las posibles respuestas a la situación.</i></p>

<b>Recommendations</b>	<ul style="list-style-type: none"> <li>El paciente puede acceder a las últimas recomendaciones que realizaron sus médicos.</li> </ul>	Screens: <ul style="list-style-type: none"> <li>Vista recomendaciones por parte del profesional</li> </ul>	<i>Luego de ver sus resultados en su caso de urticaria, Guillermo accede a las últimas recomendaciones que le han hecho sus médicos.</i>
<b>Cita</b>	<ul style="list-style-type: none"> <li>El paciente puede agendar una cita con su médico de preferencia.</li> </ul>	Screens: <ul style="list-style-type: none"> <li>Vista búsqueda de doctor y agendamiento de la cita</li> </ul>	<i>Teniendo en cuenta las recomendaciones recibidas por su caso de urticaria, Guillermo decide sacar una cita. Aquí él podrá escoger el doctor y fecha de su cita médica.</i>

<b>Registro</b>	<ul style="list-style-type: none"><li>• Registro a la app web</li></ul>	Screens: <ul style="list-style-type: none"><li>• Formulario de registro a hAppy Health</li></ul>	<i>Por ultimo Juan Carlos Machado, amigo de Guillermo, decide empezar a usar hAppy Health, y Guillermo le ayuda en su registro.</i>

Anexo E.Manual de presentación "hAppy Doc".

## **hAppy Doc**

Demo Story Board

### **BACKGROUND INFORMATION**

#### **Roles:**

**El doctor**– El doctor Sebastián Ruiz, ingresa al portal de hAppy Doc desde su celular, para verificar que su información sea la correcta y revisar su agenda de citas para los siguientes días. Teniendo en cuenta las citas que tendrá, ingresa a ver los datos de sus próximos pacientes para tener una vista previa de estos antes de tratarlos.

**El paciente** – Guillermo Niño, quien con la ayuda de hAppy Health, pidió una cita con su médico especialista como consecuencia de algunos síntomas que ha venido presentando.

---

Sebastián Ruiz ingresa a hAppy Doc para revisar su agenda y revisar algunos datos de los pacientes que tratara en sus próximas citas de la semana.

Demo Steps

<b>Step Name</b>	<b>Steps</b>	<b>Screens</b>	<b>Story/Notes</b>
<b>Login</b>	<ul style="list-style-type: none"> <li>El doctor ingresa desde su celular a la página principal de hAppy Doc, donde ingresa con los datos registrados previamente.</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>Página de acceso a hAppy Doc (mobile).</li> </ul>	<p><i>Sebastián quien ya cuenta con un registro previo en hAppy Health accede a la página e inicia sesión.</i></p>
<b>Información del doctor</b>	<ul style="list-style-type: none"> <li>El doctor ingresa a la vista principal donde puede ver toda su información y verificar si es correcta y está actualizada.</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>Vista principal-Biografía del doctor</li> </ul>	<p><i>El doctor Ruiz puede verificar que los datos que verán sus pacientes sean los correctos.</i></p>
<b>Agenda</b>	<ul style="list-style-type: none"> <li>El doctor ingresa a la agenda y puede ver las</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>Calendario con las citas</li> </ul>	<p><i>Sebastián ingresa a ver sus citas programadas en los</i></p>

	<p>citas que sus pacientes han pedido y verificar que tipo de consulta será.</p>	<p>programadas.</p>	<p><i>días próximas y se da cuenta que su próxima cita es con Guillermo Niño por lo que decide ver los datos del paciente.</i></p>
<p><b>Búsqueda del paciente</b></p>	<ul style="list-style-type: none"> <li>• Después de verificar sus citas, ingresa a la base de datos de pacientes para obtener información sobre sus próximas citas.</li> </ul>	<p>Screens:</p> <ul style="list-style-type: none"> <li>• Vista paciente- Recomendaciones</li> <li>• Resgistros</li> </ul>	<p><i>El doctor busca al paciente por su nombre e ingresa a ver los datos de este. Ahí puede ver información básica del paciente e ingresa a ver algunos datos de su última medición de ritmo</i></p>

			<i>cardiaco y temperatura corporal.</i>
--	--	--	---

# **Laboratorio 1**

## **Configuración**

Despliegue de un servicio en Curam  
para el registro de un médico.

## Introducción

Cúram se desarrolla bajo una arquitectura de componentes. Esta básicamente consiste en la separación del diseño en componentes funcionales, lo que permite una mejor reutilización e implementación de los componentes de software. En el caso de Cúram, se separa la programación del frontend y el backend (cliente-servidor). Como parte de la capacitación se planteó un taller que permitiera cubrir ambos componentes del software y tener un mejor entendimiento del funcionamiento de la plataforma.

## Ejercicio 1.

Crear entidad medico:

- Modelar entidad “médico” con los siguientes campos:
  - *Nombres*
  - *Apellidos*
  - *Cédula de Ciudadanía*
  - *Email*
  - *Fecha de Nacimiento*
  - *Sexo*
  - *Especialidad (Codetable: Cardiología, Neurología, Pediatría, Ginecología, Dermatología, Psiquiatría)*

Crear fachada medico

- Modelar la facha con los siguientes métodos:
  - *Leer todos los médicos.*
  - *Leer médico por especialidad*
  - *Leer médico por cédula.*
  - *Crear médico.*

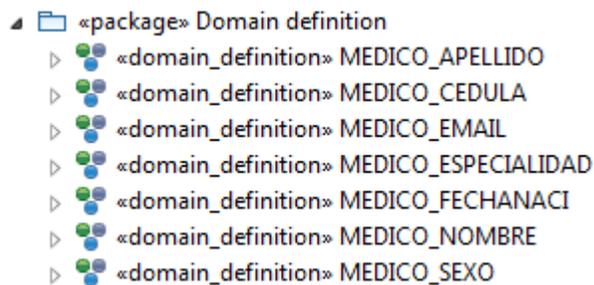
- *Modificar médico.*

## Crear UIMs

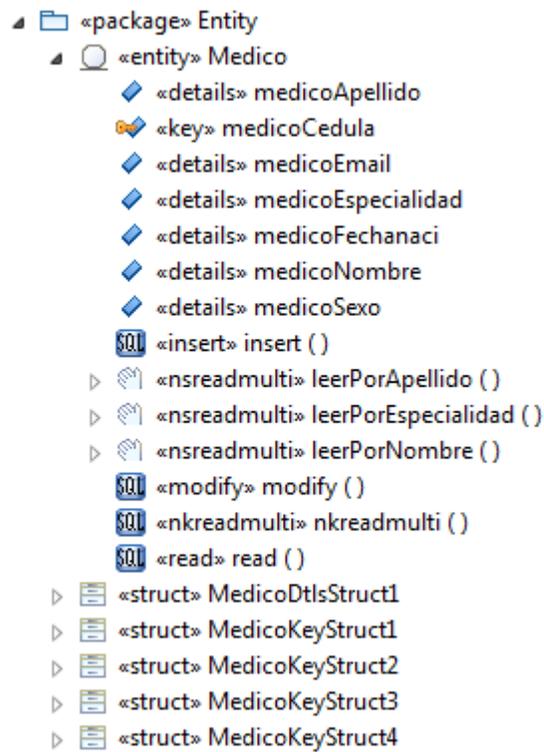
- Crear los siguientes UIM
  - *Lista de todos los médicos (Se debe hacer campo de búsqueda por especialidad,). Los datos que se deben mostrar son “Número de cédula”, Nombres, Apellidos y especialidad. Adicionalmente para cada una de las filas se debe tener la opción de “Revisar” que lo debe enviar a la vista detalles médico y la opción “Editar”, la cual le permite Editar los datos del médico. Por ultimo debe tener la opción de crear nuevo médico.*
  - *Detalles del Médico*
  - *Creación de médico.*
  - *Modificación de médico. (Todos se pueden editar excepto Número de cédula.*

## Solución

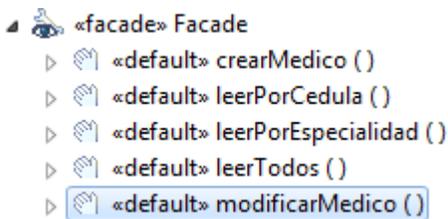
Inicialmente, por el lado del servidor, se definen los distintos artefactos a utilizar, utilizando el Rational Architect Software (RSA). Para empezar, se crean los dominios específicos. Aquí se definen los tipos de datos que se manejarán.



Luego de definir los dominios, se crea una entidad, la cual representa un objeto persistente en la base de datos. Esta incluye los atributos y métodos correspondientes a esa entidad. De igual manera se generan las estructuras las cuales son usadas como parámetros de entrada y salida de los métodos y no son persistentes en la base de datos.

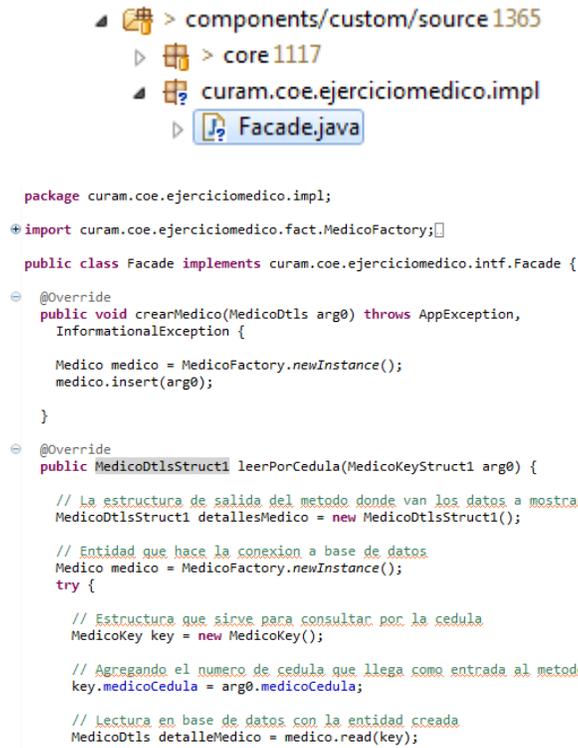


Teniendo lo anterior se crean los métodos en la fachada que serán los procesos de negocio a utilizar.



Luego de crear todos estos artefactos del servidor en el RSA, y usando el command window, vamos a la dirección del server de cúram y ejecutamos el comando build para generar lo hecho previamente. Luego de esto, vamos a eclipse, donde luego de refrescar debemos encontrar lo ejecutado en el RSA.

Ya en eclipse, buscamos el archivo del facade para empezar a codificar los métodos modelados previamente en el Facade.



```
components/custom/source 1365
├── core 1117
│   └── curam.coe.ejercicio medico.impl
│       └── Facade.java
```

```
package curam.coe.ejercicio medico.impl;

import curam.coe.ejercicio medico.fact.MedicoFactory;

public class Facade implements curam.coe.ejercicio medico.intf.Facade {

    @Override
    public void crearMedico(MedicoDtls arg0) throws ApplicationException,
        InformationalException {

        Medico medico = MedicoFactory.newInstance();
        medico.insert(arg0);

    }

    @Override
    public MedicoDtlsStruct1 leerPorCedula(MedicoKeyStruct1 arg0) {

        // La estructura de salida del metodo donde van los datos a mostrar
        MedicoDtlsStruct1 detallesMedico = new MedicoDtlsStruct1();

        // Entidad que hace la conexion a base de datos
        Medico medico = MedicoFactory.newInstance();
        try {

            // Estructura que sirve para consultar por la cedula
            MedicoKey key = new MedicoKey();

            // Agregando el numero de cedula que llega como entrada al metodo
            key.medicoCedula = arg0.medicoCedula;

            // Lectura en base de datos con la entidad creada
            MedicoDtls detalleMedico = medico.read(key);

        } catch (Exception e) {

        }

    }

}
```

Por último, se realiza la programación del frontend que va a interactuar con lo realizado en los pasos anteriores. Para empezar este paso, se realizan los archivos uim, los cuales representaran las paginas visibles en la plataforma.

```

<PAGE PAGE_ID="ListaMedicos" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd">

<PAGE_TITLE>
  <CONNECT>
    <SOURCE PROPERTY="Page.Title" NAME="TEXT" />
  </CONNECT>
</PAGE_TITLE>

  <!-- LLAMA AL METODO (OPERATION TIENE EL NOMBRE DEL METODO Y VA EN NAME DE LOS CONNECT)

<SERVER_INTERFACE NAME="leerMedicos" CLASS="Facade"
  OPERATION="leerTodos" PHASE="DISPLAY">
</SERVER_INTERFACE>

<SERVER_INTERFACE NAME="leerEspecialidad" CLASS="Facade"
  OPERATION="leerPorEspecialidad" PHASE="ACTION">
</SERVER_INTERFACE>

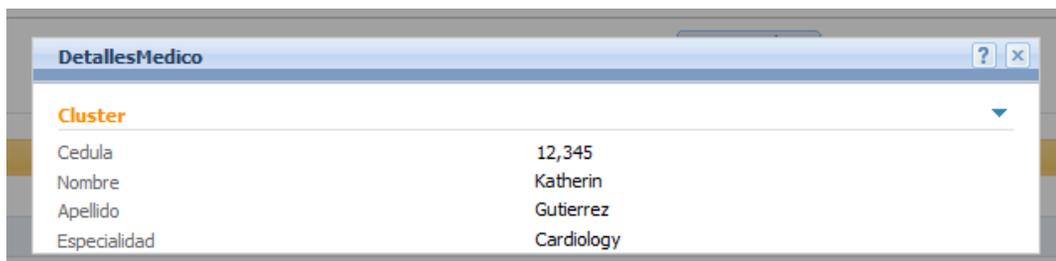
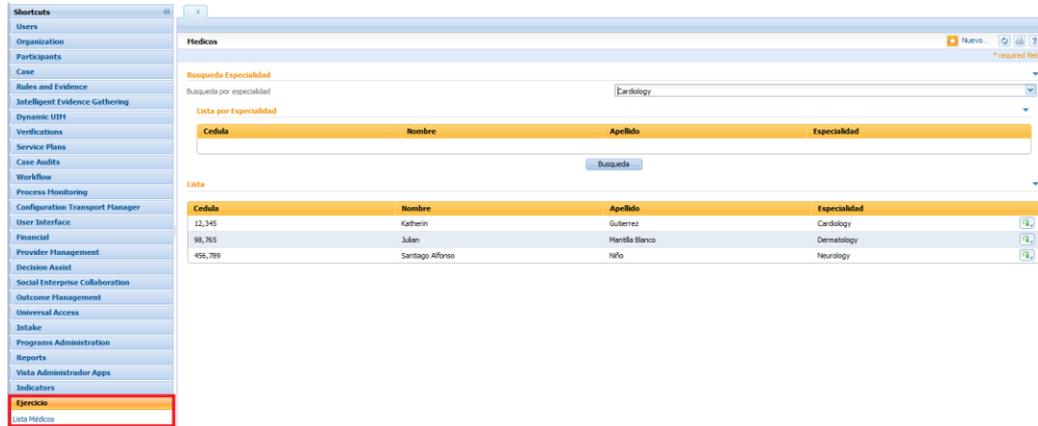
<CLUSTER TITLE="Cluster.Title.Search">
  <ACTION_SET TOP="false" ALIGNMENT="CENTER">
    <ACTION_CONTROL LABEL="ActionControl.Label.Search" TYPE="SUBMIT">
      <LINK PAGE_ID="THIS"/>
    </ACTION_CONTROL>
  </ACTION_SET>
  <FIELD LABEL="Field.Label.Bespecialidad">
    <CONNECT>
      <TARGET NAME="leerEspecialidad" PROPERTY="arg1$medicoEspecialidad"/>
    </CONNECT>
  </FIELD>
  <LIST TITLE="Lista.Title.MedicoEs">

```

Es importante tener en cuenta que cada archivo “.uim”, tendrá su correspondiente archivo de propiedades en los idiomas en los que vaya a ser implementado. En estos archivos, se tienen que definir todos los elementos que se vayan a utilizar en la página.

-  CrearMedico\_es.properties
-  CrearMedico.properties
-  CrearMedico.uim
-  DetallesMedico\_es.properties
-  DetallesMedico.properties
-  DetallesMedico.uim
-  ListaMedicos\_es.properties
-  ListaMedicos.properties
-  ListaMedicos.uim
-  ModificarMedico\_es.properties
-  ModificarMedico.properties
-  ModificarMedico.uim

Luego de definir los archivos necesarios, se realiza el build client en el command window y se inicia el servidor para poder ingresar a la plataforma, donde podremos ver los servicios agregados.



Item 1

### ModificarMedico

\* required field

**Cluster** ▼

Cedula	12,345
Nombre	<input type="text" value="Katherin"/>
Apellido	<input type="text" value="Gutierrez"/>
Email	<input type="text" value="Kathe"/>
FechaNaci	<input type="text" value="21/01/1995"/> 📅
Sexo	<input type="text" value="Femenino"/> ▼
Especialidad	<input type="text" value="Cardiology"/> ▼

**Guardar**

te

### CrearMedico

\* required field

**Cluster** ▼

Cedula	<input type="text"/>
Nombre	<input type="text"/>
Apellido	<input type="text"/>
Email	<input type="text"/>
Fecha Nacimiento	<input type="text" value="18/08/2016"/> 📅
Sexo	<input type="text" value="Masculino"/> ▼
Especialidad	<input type="text" value="Cardiology"/> ▼

**Crear**

Anexo G.Implementación del Laboratorio 2 de IBM Curam.

# **Laboratorio 2**

## **Configuración**

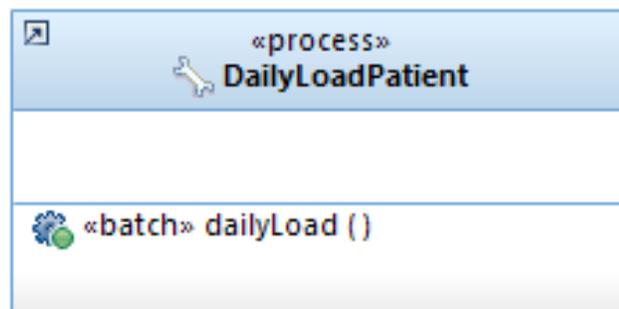
Implementación procesos Batch en  
Curam.

## Ejercicio 1.

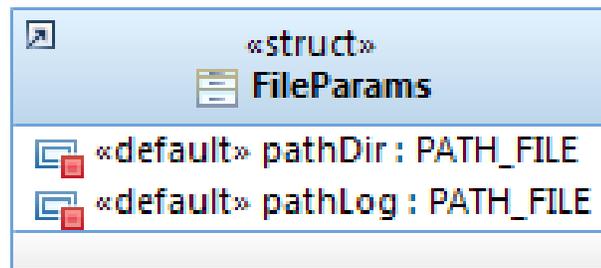
- Se debe crear un proceso batch con un archivo de entrada que cree un médico por medio de este proceso, donde se debe validar por cada campo los siguientes campos:
  - *Nombres: Máximo de 60 Caracteres. Obligatorio.*
  - *Apellidos: Máximo de 60 Caracteres. Obligatorio.*
  - *Cédula de Ciudadanía: Máximo 16 cifras. Obligatorio.*
  - *Email: Máximo de 60 Caracteres. Opcional.*
  - *Sexo. 1 carácter. Obligatorio.*
  - *Especialidad: Máximo 10 caracteres. Opcional.*
- Se debe crear la validación de estructura y de negocio. Dentro de negocio debe validar que el médico no haya sido creado, que el sexo está dentro los códigos validos al igual que la especialidad

## Solución

Inicialmente, por el lado del servidor, se definen los distintos artefactos a utilizar, haciendo uso del Rational Architect Software (RSA). Para empezar, se modela un proceso y se define un método de tipo Batch.



Se define una estructura de entrada que tiene como atributo el directorio del archivo y el directorio de log con el resultado de la operación.



Se inicia el proceso de implementación del batch creando la clase impl del proceso y agregándola a la extensión de la clase FileLoaderBatch

```

public class DailyLoadPatient extends FileLoaderBatch implements curam.cco.batchprocess.intf.DailyLoadPatient {
    /*
     * (non-Javadoc)
     * @see
     * curam.cpm.batchprocess.intf.DailyLoadPatient#dailyLoad(curam.cpm.batchprocess
     * .struct.DailyLoadPatientParams)
     */
    @Override
    public void dailyLoad(DailyLoadPatientParams info) throws ApplicationException,
        InformationalException {
        try {
            processBatch(IFileLoaderType.CPMPatient, info.pathDir, info.pathLog);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Se agrega el código único del proceso batch a IFileLoaderType

```

public interface IFileLoaderType {
    /**
     * El tipo de file loader CPMPatient
     */
    public static final String CPMPatient = "CPMPatient";

    public static final String CPMSyncInfoRisk = "CPMSyncInfoRisk";
    public static final String CPMUser = "CPMUser";
    public static final String CPMUpdateEnabledCase = "CPMUpdateEnabledCase";
    public static final String CPMCloseCase = "CPMCloseCase";
    public static final String CPMMaintainCase = "CPMMaintainCase";
    public static final String CPMCostReferenceCostCUM = "CPMCostReferenceCostCUM";
    public static final String CPMCostReferenceCostCUP = "CPMCostReferenceCostCUP";
    public static final String CPMLoadProgram = "CPMLoadProgram";
    public static final String CPMUpdateAffiliatePlan = "CPMUpdateAffiliatePlan";
}

```

Se genera la estructura del proceso Batch que inicia con la validación de la estructura proceso, la validación de negocio y finaliza con la carga del proceso.

Para la validación de la estructura, se debe validar el archivo, con el número de campos que debe tener cada uno de los atributos, el tipo de datos, la obligatoriedad y su máximo de caracteres, se crea la clase y se implementa la interfaz con el método específico

```

public class CPMPatientStructureValidation implements IStructureValidation {

    @Override
    public ValidationStructureOutDTO validate(String[] data) throws Exception {

        // Fecha de nacimiento
        dataObj[7] = ConvertUtil.convertDate(data[7], false,
            "Fecha de nacimiento", retorno);
    }
}

```

La validación del negocio proceso Batch consiste en realizar validaciones de la existencia del usuario, si el campo corresponde al indicado, etc. Se debe implementar la interfaz y su respectivo método.

```

public class CPMPatientBusinessValidation implements IBusinessValidation {

    @Override
    public ValidationOutDTO validate(Object[] data) throws Exception {
    }
}

```

Para la carga del proceso Batch, se crea la estructura necesaria para hacer la llamada a la fachada que lleva a cabo el proceso.

```

public class CPMPatientLoader implements IFileLoader {

    @Override
    public ValidationOutDTO load(Object[] data) throws Exception {
    }
}

```

Se crean las propiedades para la ejecución del batch:  
Codigo\_batch.val.structure, Codigo\_batch.val.business,  
Codigo\_batch.loader, Codigo\_batch.regex.

Finalmente se ejecuta el proceso, se crean los archivos para el médico y se valida el resultado de la operación, verificando la correcta creación del médico dentro de la base de datos.