

**PRACTICA EMPRESARIAL EN IBM DE COLOMBIA: IMPLEMENTACIÓN DE
SOLUCIÓN MOVIL PARA LA GESTIÓN Y EL CUIDADO DE UN PACIENTE
MEDIANTE EL USO DEL MONITOREO REMOTO DE SIGNOS VITALES Y LA
ADMINISTRACIÓN DEL PLAN DE TRATAMIENTO.**

KAREN LINETH CALDERÓN RODRÍGUEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2017

**PRACTICA EMPRESARIAL EN IBM DE COLOMBIA: IMPLEMENTACIÓN DE
SOLUCIÓN MOVIL PARA LA GESTIÓN Y EL CUIDADO DE UN PACIENTE
MEDIANTE EL USO DEL MONITOREO REMOTO DE SIGNOS VITALES Y LA
ADMINISTRACIÓN DEL PLAN DE TRATAMIENTO.**

KAREN LINETH CALDERÓN RODRÍGUEZ

**Trabajo de grado para optar al título de
Ingeniero de Sistemas**

Director:

**SERGIO CASTILLO CASTELBLANCO
Ingeniero de sistemas Ph.D**

Tutor:

**MARIO ALONSO MATEUS GOMEZ
Ingeniero de Sistemas**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2017

AGRADECIMIENTOS

Primeramente, a Dios por ser el pilar fundamental en mi vida, porque me permitió llegar hasta donde lo he logrado.

A mi madre Lina María Rodríguez, por su guía, amor incondicional, por ser en todo momento esa voz de aliento, esa persona que me enseñó a esforzarme y a ser valiente, por soñar a mi lado.

A mi padre Álvaro Calderón, mis hermanas Laura Calderón y Nathalia Calderón por todo su apoyo, preocupación y amor brindado.

A mis Abuelos maternos por ser como unos segundos padres, A mi familia en general por siempre sentirse orgullosos de mis logros.

A mis amigos y compañeros de clase Jhon Meneses y Lizeth García por haberme brindado su ayuda y hacer de este pasó por la universidad algo más agradable.

A mi director de tesis, el profesor Sergio Castillo por sus enseñanzas, su guía y su paciencia en el desarrollo de mi trabajo de grado.

A mis jefes en IBM y a mis compañeros del Centro de innovación quienes hicieron de este proceso una experiencia enriquecedora y valiosa, con la cual crecí profesionalmente.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. GENERALIDADES DEL PROYECTO	18
1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA.....	18
2. OBJETIVOS.....	20
2.1 OBJETIVO GENERAL	20
2.2 OBJETIVOS ESPECÍFICOS.....	20
3. ALCANCE	22
4. MARCO DE REFERENCIA.....	23
4.1 IBM	23
4.2 GLOBAL BUSINESS SERVICES	24
4.3 CENTRO DE INNOVACION IBM.....	25
5. MONITOREO REMOTO	27
5.1 OBJETIVO ESPECÍFICO 1.....	27
5.2 MARCO TEÓRICO	28
5.2.1 Aplicaciones Híbridas	28
5.2.2 Servicio WEB REST	29
5.2.3 Apache Cordova	30
5.3 DESARROLLO DE LA PRÁCTICA.....	31
5.3.1 Prerrequisitos.....	32
5.3.2 Creación de la aplicación	32
5.3.3 Añadir Plataformas	32
5.3.4 Construcción de la aplicación	33
5.3.5 Visualización de la aplicación	36
5.4 RESULTADOS.....	36
6. IBM HEALTHCARE	39

6.1 OBJETIVO ESPECIFICO 2:.....	39
6.2 MARCO TEÓRICO	40
6.2.1 Aplicaciones Web	40
6.2.2 Prueba de concepto.....	40
6.2.3 Bluemix	41
6.3 DESARROLLO DE LA PRÁCTICA	42
6.4 RESULTADOS.....	44
6.4.1 hAppy Doc	45
6.4.2 Happy Health	49
7. IBM CARE MANAGEMENT	56
7.1 OBJETIVO ESPECÍFICO 3.....	56
7.2 MARCO TEÓRICO	56
7.2.1 Cúram Software	56
7.2.2 IBM Care Management	58
7.2.3 IBM Rational Software Architect for WebSphere Software	59
7.3 DESARROLLO DE LA PRÁCTICA	60
7.3.1 HU Creación de persona.....	63
7.3.2 HU Búsqueda de persona.....	64
7.3.3 HU Configuración de proveedor.....	65
7.3.4 Cargue Batch de proveedor	67
7.3.5 Resultados	69
7.3.6 HU Creación de persona.....	69
7.3.7 HU Búsqueda de persona.....	71
7.3.8 HU Configuración de proveedor.....	72
7.3.9 Cargue Batch de proveedor	73
8. MANTENIMIENTO Y SOPORTE	75
8.1 OBJETIVO ESPECÍFICO 4.....	75
8.2 MARCO TEÓRICO	75
8.2.1 Mantenimiento	75
8.2.2 Solicitudes.....	76

8.3 DESARROLLO DE LA PRÁCTICA Y RESULTADOS	77
9. CAPACITACIONES Y CUMPLIMIENTO DE OBJETIVOS	80
9.1 ADQUISICIÓN DE CONOCIMIENTO	80
9.2 CUMPLIMIENTO DE OBJETIVOS	82
10. CONCLUSIONES	83
11. RECOMENDACIONES.....	84
BIBLIOGRAFIA.....	85

LISTA DE TABLAS

	Pág.
Tabla 1. Indicadores de estado.....	43
Tabla 2. Etapas de la prueba de concepto hAppy Doc	45
Tabla 3. Etapas de la prueba de concepto hAppy Health	49
Tabla 4. Proceso estándar para el desarrollo en ICM.....	62
Tabla 5. Comandos para la generación Backend y Frontend	62
Tabla 6. Descripción del mantenimiento y soporte de la operación	77
Tabla 7. Cumplimiento de objetivos	82

LISTA DE FIGURAS

	Pág.
Figura 1. Logotipo IBM.....	23
Figura 2. Ubicación del centro de innovación IBM	26
Figura 3. Arquitectura de la aplicación móvil	27
Figura 4. Arquitectura de un servicio Web REST.....	29
Figura 5. Arquitectura Detallada aplicación móvil	31
Figura 6. Código fuente de la petición al servidor.....	33
Figura 7. Respuesta del servidor	34
Figura 8. Arreglo de datos del servicio.....	34
Figura 9. Código fuente del uso de HighCharts	35
Figura 10. Inicio de la aplicación móvil	37
Figura 11. Visualización en tiempo real sensor de pulso	37
Figura 12. Visualización en tiempo real sensor temperatura	38
Figura 13. IBM Bluemix Dashboard	42
Figura 14. Estado de las pruebas de concepto.....	44
Figura 15. Registro hAppy Doc.....	46
Figura 16. Inicio de sesion hAppy Doc.....	46
Figura 17. Mi perfil hAppy Doc.....	47
Figura 18. Búsqueda de paciente hAppy Doc.....	47
Figura 19. Perfil del paciente	48
Figura 20. Historial de signos vitales	48
Figura 21. Agenda hAppy Doc.....	49
Figura 22. Registro de hAppy Health	50
Figura 23. Inicio de sesión hAppy Health.....	51
Figura 24. Perfil del paciente hAppy Health	51
Figura 25. Inicio hAppy Health	52
Figura 26. Signos vitales hAppy Health	52

Figura 27. Citas Programadas hAppy Health.....	53
Figura 28. Registros Médicos hAppy Health.....	53
Figura 29. Agendar Citas hAppy Health.....	54
Figura 30. Agendar Citas hAppy Health.....	54
Figura 31. Recomendaciones hAppy Health.....	55
Figura 32. Estructura de Cúram.....	57
Figura 33. Descripción de la instalación del producto.....	58
Figura 34. Aplicaciones de IBM Care Management.....	59
Figura 35. Ejemplo sintaxis archivo de carga masiva	68
Figura 36. Configuración del proceso Batch	68
Figura 37. Comprobación de persona.....	70
Figura 38. Inscripción de Persona	70
Figura 39. Búsqueda por tomador de la póliza	71
Figura 40. Búsqueda por asegurado.....	71
Figura 41. Inscripción de proveedor.....	72
Figura 42. Búsqueda de proveedor.....	72
Figura 43. Resultados consola de carga masiva	74

GLOSARIO

AFILIADO o PACIENTE: Es un beneficiario de los servicios ofrecidos por la organización y por tanto es sujeto de seguimiento dentro de un caso. Un afiliado o paciente, puede pertenecer a múltiples programas y por tanto tener asociados múltiples casos.

BPM: (Beats per minute) unidad empleada para medir el ritmo.

CASO: Es una instancia que crea el sistema cuando a un afiliado o paciente se le asigna un programa y un gestor responsable del seguimiento de su plan de cuidado.

CLOUDANT: Es un servicio de base de datos distribuido y no relacional de código abierto que ofrece un motor integrado de administración, búsqueda y análisis de datos diseñado para aplicaciones web.

GESTOR DE CASOS: Es un rol dentro de la solución IBM Care Management, al cual se le asignan casos para ejercer la labor de seguimiento del plan de tratamiento y los resultados en términos de un paciente.

INTERNET OF THINGS (IoT): El internet de las cosas se trata de métodos de recolección de datos a través de sensores, dispositivos, máquinas y personas que pueden permitir mejores operaciones, proceso de negocio, cadena de suministro y gestión de activos, los datos se utilizan para el análisis de la información en tiempo real.

PLAN DE CUIDADO / FLUJO DE TRABAJO: Es un conjunto de actividades, eventos y recursos, organizados para realizar el seguimiento de cada asegurado y de cada caso asociado a un asegurado.

PLAN DE TRATAMIENTO o RESULTADOS: Es un conjunto de actividades, eventos y recursos, organizados para realizar el seguimiento de cada paciente.

PROGRAMA: Es una definición de negocio que agrupa una población caracterizada, sobre la cual se busca lograr resultados.

PULSO: Palpitación provocada por la expansión de las arterias como consecuencia de la circulación de sangre bombeada por el corazón.

SENSOR: Dispositivo que detecta o mide una propiedad física y luego registra o responde a esa propiedad.

SIGNOS VITALES: Son indicadores que reflejan el estado fisiológico de los órganos fundamentales como: cerebro, corazón, pulmones.

TEMPERATURA: Magnitud física que expresa el nivel de calor que ostenta un cuerpo determinado, un objeto, un ambiente, entre otros.

VISTA 360: Ofrece una visualización integral de la información del paciente, la cual permite la identificación oportuna de factores de riesgo y evaluar el progreso del paciente en el plan de cuidado.

RESUMEN

TITULO: PRACTICA EMPRESARIAL EN IBM DE COLOMBIA: IMPLEMENTACIÓN DE SOLUCIÓN MOVIL PARA LA GESTIÓN Y EL CUIDADO DE UN PACIENTE MEDIANTE EL USO DEL MONITOREO REMOTO DE SIGNOS VITALES Y LA ADMINISTRACIÓN DEL PLAN DE TRATAMIENTO. *

AUTOR: KAREN LINETH CALDERÓN RODRÍGUEZ **

PALABRAS CLAVE: IBM, Bluemix, Care Management, Telemedicina, Servicio Web Rest

DESCRIPCIÓN:

El presente proyecto está basado en la práctica empresarial realizada en IBM de Colombia, en el Centro de Innovación en Santander para la industria de salud, el cual brinda apoyo en: el delivery de proyectos de Care Management, el desarrollo de activos en áreas como Telemedicina (monitoreo remoto) y la realización de pruebas de conceptos en tecnologías como Bluemix entre otras. Durante la práctica, se adquieren conocimientos en varios lenguajes de programación y frameworks para el desarrollo en tecnologías móviles y plataformas propias de IBM.

Durante el proceso de formación y adquisición de nuevos conocimientos se estudia: las nociones básicas sobre el desarrollo móvil multiplataforma, el consumo de datos provenientes de un servicio Web Rest y el uso de la plataforma en la nube IBM Bluemix, en donde inicialmente se tiene la tarea de desarrollar una aplicación móvil a través de la cual se puedan visualizar los datos provenientes de un sensor en tiempo real, los cuales están asociados a un paciente. Como un siguiente paso se establece la realización de las pruebas de concepto enfocadas al paciente y al médico gestor, con las cuales se quiere garantizar un manejo adecuado de los datos y el plan de tratamiento del paciente. En el transcurso de la práctica se adquirieren habilidades para: llevar a cabo tareas en equipo, aprender de manera autodidacta, hacer una resolución eficiente de los incidentes que se presenten en cualquiera de las implementaciones, así como también destrezas para dar soporte a las aplicaciones que actualmente se encuentran disponibles en los clientes.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de sistemas e Informática.
Director: Sergio Fernando Castillo Castelblanco.

ABSTRACT

TITLE: INTERNSHIP AT IBM COLOMBIA: IMPLEMENTATION OF A MOBILE SOLUTION FOR THE CARE MANAGEMENT OF A PATIENT THROUGH THE REMOTE MONITORING OF VITAL SIGNS AND THE ADMINISTRATION OF THE TREATMENT PLAN*

AUTHOR: KAREN LINETH CALDERÓN RODRÍGUEZ**

KEYWORDS: IBM, Bluemix, Care Management, Telemedicine, Web Rest Service

DESCRIPTION:

The present project is based on the business practice carried out in IBM of Colombia, in the center of innovation in Santander for the health industry, which provides support in: the delivery of projects of Care Management, the development of assets in areas such as telemedicine (remote monitoring) and the testing of concepts in technologies such as Bluemix, among others. During the practice, knowledge is acquired in a variety of programming languages and frameworks for development in mobile technologies and proprietary platforms of IBM.

Throughout the process of training and acquisition of new knowledge the following are studied: the basic notions on the multi-platform mobile development, consumption of data from a Web Rest service and the use of the platform in the IBM Bluemix cloud, where initially there's the task of developing a mobile application through which you can visualize the data from a sensor in real time, which is associated with a patient. As a next step, the tests of concept focused on the patient and care manager are executed, tests with which you want to guarantee a proper handling of the data and of the treatment plan for the patient.

In the course of the practice, certain skills are acquired in order to: Perform tasks as a team, learn in a self-taught way, make an efficient resolution of the incidents that may present themselves in any of the implementations, as well as obtaining skills to support the applications that are currently available in the customers.

* Degree Work

** Faculty of Physico-Mechanical Engineering. Department of Systems Engineering and Computing Science. Supervisor: Sergio Castillo Castelblanco

INTRODUCCIÓN

Actualmente los proveedores de servicios de salud deben afrontar grandes retos que les exigen aumentar la calidad de sus productos y servicios, reducir los costos de operación y cumplir con marcos regulatorios cada vez más exigentes.

Así mismo se enfrentan a un entorno que crece a un ritmo acelerado, la interconexión de dispositivos y sensores generan una cantidad de datos valiosos que deberían ser aprovechados para la toma de decisiones y con los cuales se podría evitar diagnósticos erróneos, “IBM estima que el 90% de todos los datos generados por dispositivos tales como los teléfonos inteligentes, las tabletas o los vehículos conectados nunca se analiza ni se tiene en cuenta. El 60% de estos datos pierden valor unos milisegundos después de haberse generado”¹. Debido a esto la industria de la salud requiere de una transformación con la cual se tengan nuevas perspectivas que permitan sortear los retos de salud que afligen a la sociedad. Al mismo tiempo que se presentan soluciones innovadoras.

A través del uso de las tecnologías móviles en el campo de la salud, más precisamente en el área denominada telemedicina se puede dar un mejor manejo, en el tratamiento de estados de salud cada vez más complejos.

IBM ha centrado sus esfuerzos en el desarrollo de estrategias y soluciones para el sector salud, a través de las cuales se pueda facilitar la transformación de los entornos clínicos para proveer accesos inmediatos y seguros a la información vital del paciente, así como también tener un control eficiente de los recursos del hospital, tanto de personas como de instrumentos, equipos, materiales y medicamentos.

¹ IBM. IBM crea una nueva unidad de Internet de las Cosas. Notas de prensa. [en línea]. [Citado 29 Enero, 2017]. Disponible en: <https://www-03.ibm.com/press/es/es/pressrelease/46459.wss>

El desarrollo de este proyecto pretende aportar en soluciones que ayuden a contrarrestar el impacto negativo en la atención y el cuidado de la salud del paciente, haciendo de esto una ventaja competitiva.

Este documento presenta una descripción de las características principales de algunas herramientas de IBM y el uso de estas para el desarrollo del proyecto.

1. GENERALIDADES DEL PROYECTO

1.1 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

La OMS, en su definición de salud, establece que esta es “el estado de perfecto bienestar físico, psíquico y social, y no solo la ausencia de lesión o enfermedad”². El término hace referencia a un estado de bienestar o equilibrio en donde un ser humano asume como aceptable el estado general en el cual se encuentra o hace constancia de la ausencia de enfermedades o de factores perjudiciales en su organismo. El sistema de salud en Colombia busca garantizar a los usuarios la prestación de servicios de alta calidad con el mejoramiento continuo en la atención, de modo que este ayude a contrarrestar el impacto sobre las condiciones de salud de toda la población.

Dicho objetivo se ha convertido en ineficiente debido a la baja calidad de los servicios prestados, a la negación de los servicios a los usuarios y a la confusión entre las distintas patologías, actualmente existe una coyuntura en el sistema de salud que nace de la incapacidad de gestionar adecuadamente los recursos económicos y no solo esto, sino también la información vital relacionada con los usuarios.

A través del uso de las tecnologías móviles en el campo de la salud, más precisamente en el área denominada telemedicina, se puede dar un mejor manejo a la crisis actual por la que atraviesa el sistema de salud en Colombia, en temas que van desde la prevención, el diagnóstico hasta el tratamiento de enfermedades crónicas.

² PARDO, A. ¿Qué es salud? Rev. Medicina de la Universidad de Navarra, 41(2), 4-9. 1997.

Dado que los dispositivos móviles (Smartphone, tablets, PDA's entre otros) han avanzado a la par con la tecnología, y estos nos pueden brindar capacidades similares o incluso superiores a las antiguas herramientas.

IBM pretende sacar provecho de esto y jugar un rol importante en el cuidado de la salud. Ha colaborado en mejorar la atención de los pacientes, a reducir los errores médicos, a controlar los costos innecesarios para las entidades prestadores de los servicios de salud y a implementar la integración de los sistemas de información. A través de IBM Care Management se ofrecen las capacidades claves y necesarias para gestionar adecuadamente la atención en los pacientes, evaluando sus necesidades para de esta forma establecer un plan de asistencia adecuada y de este modo responder a sus necesidades de manera eficiente.

A través de la presente práctica empresarial se pretende aportar en soluciones móviles para los distintos sistemas operativos que ayuden a contrarrestar el impacto negativo en la atención y el cuidado de la salud del paciente y hacer un uso más apropiado de la información vital de los usuarios, haciendo de esto una ventaja competitiva para el diagnóstico oportuno y el tratamiento a través de la telemedicina.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar solución móvil que permita hacer eficiente la gestión, el cuidado, y el seguimiento de los pacientes a través de planes de tratamiento, haciendo uso para ello: de la solución IBM Care Management y de los datos generados por sensores, con la cual se pueda aportar también en la preventa de soluciones IBM, donde se empleen las últimas tecnologías del mercado, para de esta manera ofrecer un valor agregado a los clientes.

2.2 OBJETIVOS ESPECÍFICOS

- Implementar aplicación móvil en los distintos sistemas operativos: Android, iOS. Para el consumo de datos que son vitales para un paciente (pulso cardiaco, temperatura) provenientes de un sensor en tiempo real a través de un servicio web REST.
- Realizar pruebas de concepto en Bluemix con “Responsive Design” las cuales estarán enfocadas: al paciente y al médico (gestor) respectivamente, donde estos podrán acceder a sus datos en tiempo real, realizar o verificar recomendaciones médicas dependiendo del tipo de usuario que sea y tener un control eficiente de sus citas médicas.
- Programar el código fuente requerido para personalizar la solución IBM Care Management a los diferentes usuarios que lo soliciten, utilizando el lenguaje de programación Java para el back-end y XML para el front- end.

- Brindar mantenimiento correctivo a los componentes de la solución IBM Care Management, los cuales implican la corrección de defectos, la actualización o adaptación de los modelos de acuerdo a las solicitudes de mejora reportadas en el sistema.

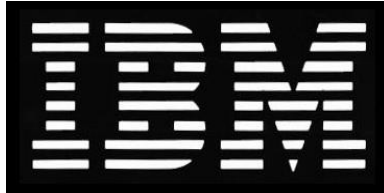
3. ALCANCE

Como resultado del presente proyecto, se tendrán varios desarrollos en diferentes herramientas propias de IBM como lo son: Bluemix, IBM Care Management. Con los cuales se quiere hacer eficiente la gestión, el cuidado, y el seguimiento de los pacientes, además de aportar en la preventa de soluciones IBM.

4. MARCO DE REFERENCIA

4.1 IBM

Figura 1. Logotipo IBM



Fuente: ABADIADIGITAL. Evolución del logotipo de IBM (1889 – 2008) Disponible en: <http://www.abadiadigital.com/evolucion-del-logotipo-de-ibm-1889-2008/>

International Business Machines Corp. (IBM) es una compañía estadounidense privada que provee soluciones de hardware, entre los que se incluyen computadoras portátiles y de escritorio, y sistemas de mainframes, servidores y almacenamiento, así como software, servicios financieros y una amplia gama de servicios de tecnología de información. Las principales operaciones de la compañía comprenden cinco segmentos: servicios globales de tecnología, servicios de negocios mundiales, software, sistemas y tecnología, y financiamiento global. Además de los servicios de TI, provee servicios de consultoría de negocios, aplicaciones y outsourcing, entre otros. IBM tiene presencia global y opera en más de 170 países, incluidas oficinas en Latinoamérica y el Caribe.³

IBM se estableció en Colombia en 1937, siendo la primera empresa en el país en iniciar la integración de la tecnología al desarrollo. Hoy, cumple 80 años de presencia activa en el mercado colombiano.

³ BNAMERICAS. International Business Machines Corp. [en línea]. [Citado 29 Enero, 2017]. Disponible en Internet: <http://www.bnamericas.com/company-profile/es/international-business-machines-corp-ibm>

Desde su establecimiento en Colombia, IBM ha ofrecido el soporte tecnológico más eficiente para las industrias colombiana, apoyando sus acciones por medio de unidades especializadas de negocios, integrando una amplia gama de productos y servicios en las áreas de producción, distribución, comercio, banca, educación, salud y telecomunicaciones.

La innovación es el núcleo de la estrategia de IBM. Hoy en día, IBM se centra en cuatro iniciativas de crecimiento: analítica empresarial, cloud computing, mercados en crecimiento y Smarter Planet.

4.2 GLOBAL BUSINESS SERVICES

Global Business Services es la división de consultoría de IBM que brinda servicios de integración de sistemas, desarrollo, gestión y mantenimiento de aplicaciones, analítica de la información y estrategia en recursos humanos, logística, finanzas y marketing. Todas las industrias se sienten presionadas a innovar, a responder a un cambio veloz con respecto a las expectativas de los clientes y aprovechar los avances tecnológicos que se encuentran disponibles, desde la información analítica a la conectividad móvil, desde las redes sociales a Cloud Computing. Como la mayor organización consultora del mundo, *IBM Global Business Services* ayuda a enfrentar los cambios actuales, y a capturar nuevas oportunidades de negocio⁴.

Tecnología basada en la vista integral del cliente y la coordinación del equipo humano, recursos y actividades para la gestión de casos, permitiendo hacer seguimiento a los resultados e indicadores de gestión y definir actividades proactivas en los servicios de atención.

⁴ IBM. Industrias. [en línea]. [Citado 29 Enero, 2017]. Disponible en: <http://www-935.ibm.com/services/co/gbs/consulting/>

4.3 CENTRO DE INNOVACION IBM

El centro de innovación IBM en Santander para la industria de salud es una iniciativa de IBM Colombia que busca posicionar la compañía en el sector salud a través de la entrega de soluciones innovadoras con las que se pueda impactar positivamente la región y el país.

El centro de innovación se fundó en el mes abril del 2016 y se encuentra ubicado en zona franca Santander, cuenta 18 puestos de trabajo y una sala de reuniones.

Actualmente están vinculados al centro de innovación IBM: 12 estudiantes en prácticas de diferentes universidades y carreras (Universidad industrial de Santander, Universidad Santo Tomas y Universidad Autónoma de Bucaramanga), además de esto el centro está dirigido por dos ingenieros (Líder del centro, Ingeniero de desarrollo de software).

Algunos de sus principales objetivos están enfocados a:

- Apoyar el delivery de proyectos en la solución IBM Care Management y otras tecnologías para la industria salud.
- Desarrollar activos y pruebas de concepto para la industria salud.
- Posicionar el centro de innovación a nivel local y nacional.
- Fortalecer el conocimiento de los profesionales en tecnologías IBM aplicadas a la industria salud.

Figura 1. Ubicación del centro de innovación IBM



5. MONITOREO REMOTO

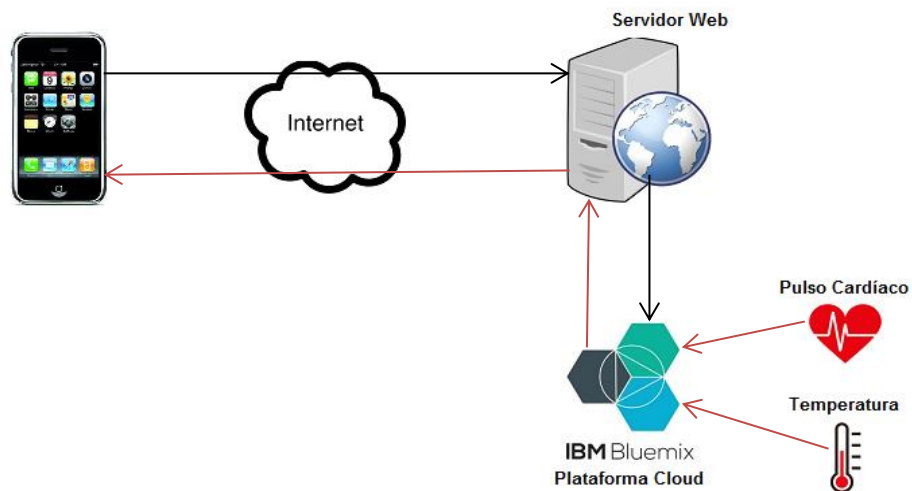
5.1 OBJETIVO ESPECÍFICO 1

El objetivo específico 1 de la práctica empresarial se definió de la siguiente manera:

“Implementar aplicación móvil en los distintos sistemas operativos: Android, iOS. Para el consumo de datos que son vitales para un paciente (pulso cardiaco, temperatura) provenientes de un sensor en tiempo real a través de un servicio web REST”.

La empresa IBM buscaba el desarrollo de un aplicativo móvil a través del cual se pudiera monitorear en tiempo real el pulso y la temperatura de un paciente, los datos de los sensores están almacenados en la plataforma en la nube *IBM Bluemix* y disponibles mediante un servicio Web Rest como lo muestra la figura 3. La aplicación debe leer tales datos y mostrarlos en tiempo real al usuario final.

Figura 2. Arquitectura de la aplicación móvil



En el presente capítulo se explica lo referente al desarrollo móvil híbrido, las tecnologías que se emplearon para llevar a cabo esto, así como también el consumo de datos a través de un servicio *Web Rest*, que dan como resultado una aplicación móvil multiplataforma.

5.2 MARCO TEÓRICO

5.2.1 Aplicaciones Híbridas. Las aplicaciones Híbridas son una combinación de las: aplicaciones nativas y aplicaciones web.

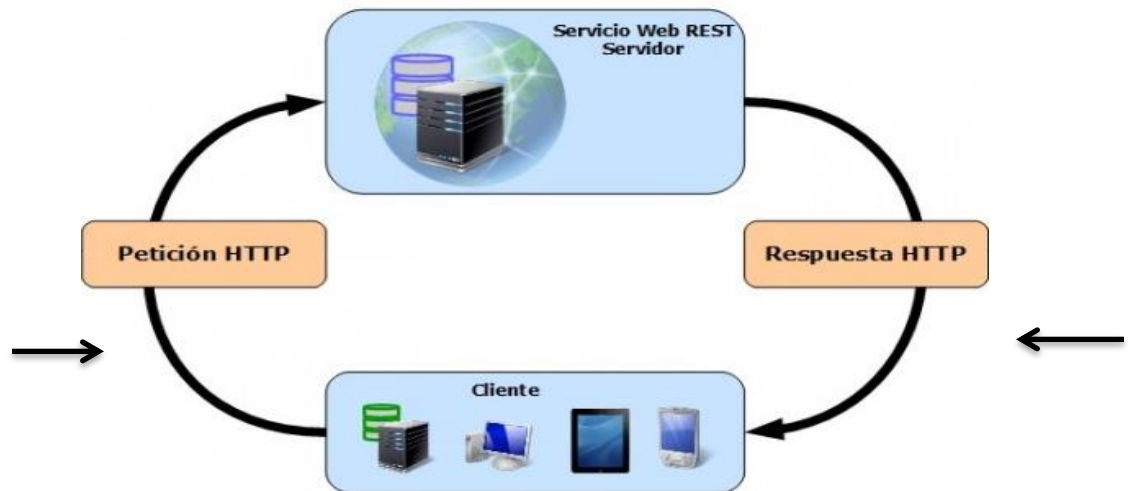
El concepto de aplicación híbrida hace referencia a una página web basada en HTML en la cual se usan las herramientas estándar HTML5, CSS3 y JavaScript. La cual posteriormente es envuelta en código nativo, de modo que el paquete resultante puede ser distribuido, al igual que en las aplicaciones nativas.

Al ser una combinación de dos tipos de aplicaciones, el desarrollo híbrido reúne lo mejor de ambos modelos. Como se ha hecho mención anteriormente en este tipo de aplicaciones se permite el uso de tecnologías multiplataforma, pero a su vez permite acceder a las utilidades y sensores del dispositivo. La mayor parte de la infraestructura es de tipo Web y la comunicación con las herramientas del terminal se lleva a cabo a través de comunicadores como *Apache Cordova* o también llamado *PhoneGap*.

Con este tipo de aplicaciones no es necesario utilizar un lenguaje específico para la construcción de la aplicación completa, como si es el caso de las aplicaciones nativas. Solo es indispensable que la plataforma contenedora está construida mediante una tecnología específica y lo restante puede ser construido con un lenguaje de los que ya se ha hecho mención.

5.2.2 Servicio WEB REST. REST (*Representational State Transfer*) es un estilo de arquitectura de software que define un conjunto de principios acerca de cómo acceder a los recursos del sistema, además de cómo transferir por el protocolo de transporte HTTP hacia clientes escritos en diversos lenguajes.

Figura 3. Arquitectura de un servicio Web REST



Fuente: ACADEMIA ANDROID. Servicios Web: arquitectura REST. Disponible en: <http://academiaandroid.com/servicios-web-arquitectura-rest/>

La arquitectura REST se basa en una serie de principios que permiten abstraer los elementos de dicha arquitectura dentro de un sistema hipermedia distribuido.

Los principios en los que se basa la arquitectura REST:

- **Métodos HTTP de forma explícita:** es una de las principales características de un servicio *Web REST*, ya que proporcionan una serie de métodos que permiten a la aplicación cliente realizar operaciones sobre los recursos del servidor web. Los métodos HTTP más importantes son: *PUT*, *GET*, *POST*, *DELETE* y *HEAD*.

- **Arquitectura cliente/servidor que no mantiene el estado:** al tratarse de una tecnología que está en constante crecimiento, es necesario mantener el control del servidor web cuando este trata de responder las peticiones que recibe. Debido a esta premisa, el servidor no guarda el estado o el contexto de ninguna petición previa del cliente al servidor.
- **Uso de URIs a modo de directorios:** permite definir un sistema de sintaxis universal, el cual otorga a cada recurso de manera individual un identificador único (*Uniform Resource Identifier*) asociado.
- **Uso de los formatos XML y JSON para la transferencia de datos:** es un principio que posibilita el uso del servidor web desde diferentes clientes, escritos en diferentes lenguajes de programación y en diferentes plataformas.

5.2.3 Apache Cordova. Apache Cordova es un marco de desarrollo móvil de código abierto utilizado por IBM para el desarrollo multiplataforma. Permite utilizar las tecnologías estándares web como: HTML5, CSS3 y JavaScript para el desarrollo, evitando el lenguaje de desarrollo nativo para cada plataforma móvil. Las aplicaciones se ejecutan dentro de envolturas propias de cada plataforma y dependen de enlaces estándares API para acceder a los sensores, datos y estado de la red en cada dispositivo.⁵

En Apache Cordova las aplicaciones se basan en un común *config.xml* archivo que proporciona información acerca de la aplicación y especifica los parámetros que afectan su funcionamiento, La misma aplicación se implementa como una página web, en un archivo local llamado *index.html* que hace referencia a cualquier CSS,

⁵ APACHE CORDOVA. Perspectiva general. [en línea]. [Citado 29 Enero, 2017]. Disponible en: <https://cordova.apache.org/docs/es/3.1.0/guide/overview/>

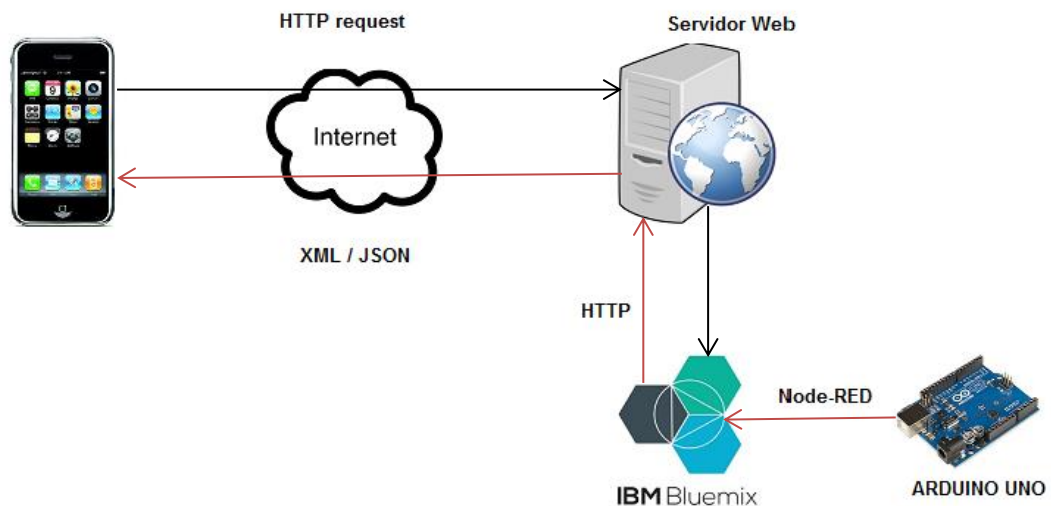
JavaScript, imágenes, archivos multimedia u otros recursos. La aplicación se ejecuta como un *WebView* dentro de la envoltura de la aplicación nativa.

5.3 DESARROLLO DE LA PRÁCTICA

Para dar claridad en cómo fue el desarrollo de la aplicación móvil que permite la visualización en tiempo real de datos vitales de un paciente, se presenta la figura 5, la cual representa el escenario de la aplicación móvil desarrollada en este objetivo.

En la figura se ilustra como la aplicación móvil está consumiendo datos de un servicio *Web Rest* alojado en la plataforma *cloud IBM Bluemix*, la cual a su vez se alimenta de datos tomados de un sensor.

Figura 4. Arquitectura Detallada aplicación móvil



Como se puede observar en la figura, la aplicación se conecta a la plataforma *cloud IBM Bluemix* a través de un servicio *web Rest*, el cual está basado en el formato de

intercambio de datos: JSON (*JavaScript Object Notation*). A través de la petición GET el sistema construye la respuesta que luego será visualizada en el dispositivo móvil.

A continuación, se detallan los requerimientos técnicos y los pasos que se tuvieron en cuenta para el desarrollo de la aplicación móvil:

5.3.1 Prerrequisitos. Antes de ejecutar cualquiera de las herramientas de línea de comandos, se requirió de la instalación del SDK (*software development kit*) para cada plataforma de destino. Para instalar el Cordova de línea de comandos es necesario:

- Descargar e instalar Node.js, con el fin de comprobar si la instalación se realizó de manera exitosa después de la instalación se debe poder invocar nodo o npm en la línea de comandos.
- Instalar la utilidad de Cordova a través del comando ***npm install -g cordova***.

5.3.2 Creación de la aplicación. Inicialmente se creó la aplicación haciendo uso del comando: ***cordova create, hello com.example.hello HelloWorld***, el anterior comando sírvase de ejemplo para clarificar la sintaxis que debe mantenerse en dicha creación.

El primer argumento especifica un directorio que se genera para el proyecto. Los otros argumentos son opcionales y corresponden a un identificador de dominio, al texto que se proporciona en pantalla de la aplicación.

5.3.3 Añadir Plataformas. Después de creada la aplicación y dentro del directorio del proyecto, se ejecutaron los comandos que especifican el conjunto de

plataformas destino (ios, Android, BlackBerry, Firefox) por ejemplo de la siguiente forma:

Cordova Platform add Android

5.3.4 Construcción de la aplicación. De forma predeterminada, el cordova **create** genera la esquelética de una aplicación basada en la web cuya portada es el proyecto **www/index.html**. El cual podrá ser modificado de múltiples maneras de modo de que se aprovechen las tecnologías web estándar.

Para el consumo del servicio Web Rest es fundamental realizar la petición a un servidor, el cual aloja en la nube los datos tomados desde el sensor de pulso y temperatura respectivamente en la plataforma cloud IBM Bluemix.

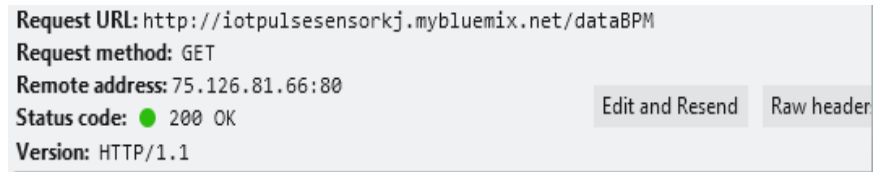
Para la codificación de la petición GET con la cual obtiene un recurso del servidor se utilizó *jQuery* mediante el método `getJSON ()` de la siguiente manera:

Figura 5. Código fuente de la petición al servidor

```
$(function () {  
    var count=0; //contador  
    var globaldata=null;  
    $(document).ready(function () {  
        $.getJSON("http://iotpulsesensorkj.mybluemix.net/dataBPM", function(data) {  
            globaldata=data.rows; // Variable que contiene los objetos recuperados  
        });  
    });  
});
```

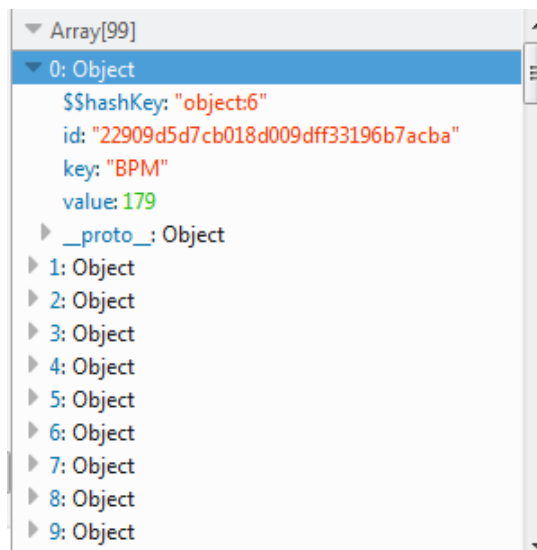
Si la petición se realiza de manera exitosa, el servidor responde con un código de estado de HTTP: 200 OK (respuesta estándar para peticiones correctas).

Figura 6. Respuesta del servidor



Después de realizada la petición al servidor los datos que provienen como respuesta se pudieron visualizar en la consola del navegador como un Array. El objeto contiene: *Key*, *Valué*, *id*. Los cuales representan el nombre del signo vital que se obtuvo (pulso o temperatura), el valor del dato, y el id único que se generó cuando los datos fueron almacenados.

Figura 7. Arreglo de datos del servicio



Habiendo realizado la petición, se hizo uso de la librería *HighCharts*⁶ para la visualización del gráfico en tiempo real. Gráfico en el cual el “eje x” está representado por el tiempo, el cual es tomado directamente desde el dispositivo de donde se lleva a cabo la medición, el “eje y” estará alimentado con los datos (Value) provenientes del servicio Web Rest y que hace referencia a los valores de los signos vitales del paciente.

Figura 8. Código fuente del uso de HighCharts

```
$('#container').highcharts({
  chart: {
    type: 'spline',
    animation: Highcharts.svg,
    marginRight: 10,
    events: {
      load: function () {

        // Actualizacion de la vista cada segundo
        var series = this.series[0];
        setInterval(function () {
          var x = (new Date()).getTime(), // Hora actual
              y = obtenerY();
          series.addPoint([x, y], true, true);
        }, 2000);
      }
    },
    title: {
      text: 'Gráfica BPM'
    },
    xAxis: {
      type: 'datetime',
      tickPixelInterval: 150
    },
    yAxis: {
      title: {
        text: 'Pulsaciones'
      },
      plotlines: [{
        value: 0,
        width: 1,
        color: '#808080'
      }]
    }
  }
});
```

⁶ HighCharts es una biblioteca de gráficos escrita en JavaScript, que ofrece una manera fácil de agregar gráficos interactivos en un sitio web o aplicación web.

5.3.5 Visualización de la aplicación. En el SDK para plataformas móviles vienen a menudo incluidos emuladores que ejecutan una imagen de la aplicación en el dispositivo, para permitir de esta manera al desarrollador visualizar la forma en que interactúa la aplicación con características de la plataforma.

De igual manera a través del acceso a la CMD (símbolo del sistema) se hizo uso de los comandos de Cordova Apache: ***cordova prepare*** y ***cordova build Android*** para generar en el caso particular de la plataforma Android un archivo con extensión ***.apk*** (*Android Application Package*), el cual fue usado para la instalación de la aplicación como tal en el dispositivo móvil.

Para el caso en particular de la plataforma Android el archivo con extensión ***.apk*** se ubica en la ruta ***nombre_proyecto/platforms/android/build/outputs/apk***, cabe aclarar que la carpeta build se construye cuando se ejecuta el comando que genera la aplicación (***cordova build***)

5.4 RESULTADOS

Una vez concluido el desarrollo se obtuvo una aplicación móvil multiplataforma de la cual se puede evidenciar a continuación.

Las figuras que se presentan en esta parte de resultados fueron tomadas de un dispositivo móvil con sistema operativo android (lado izquierdo de la figura) real versus un emulador de un dispositivo con sistema operativo iOS (lado derecho de la figura).

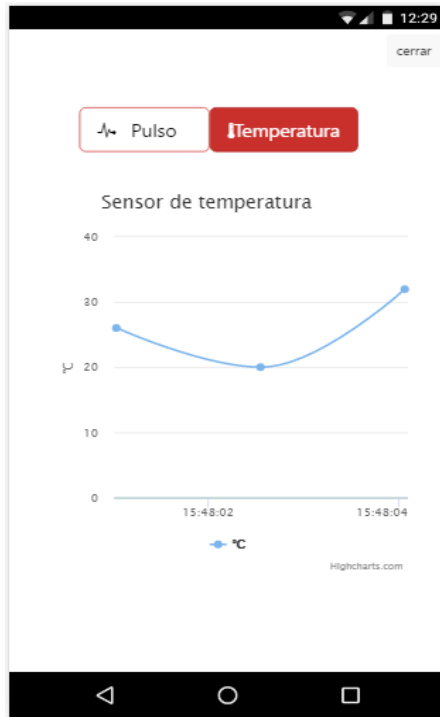
Figura 9. Inicio de la aplicación móvil



Figura 10. Visualización en tiempo real sensor de pulso



Figura 11. Visualización en tiempo real sensor temperatura



6. IBM HEALTHCARE

6.1 OBJETIVO ESPECIFICO 2:

El objetivo específico 2 de la práctica empresarial se definió de la siguiente manera:

“Realizar pruebas de concepto en Bluemix con “*Responsive Design*” las cuales estarán enfocadas: al paciente y al médico (gestor) respectivamente, donde estos podrán acceder a sus datos en tiempo real, realizar o verificar recomendaciones médicas dependiendo del tipo de usuario que sea y tener un control eficiente de sus citas médicas”.

Las pruebas de concepto son, en esencia, desarrollos informáticos rápidos que permiten verificar si una aplicación es de interés para los posibles usuarios. En la presente práctica, la empresa solicitó realizar dos de estos desarrollos software: **hAppy Health** y **hAppy Doc**.

hAppy Health es una prueba de concepto orientada al paciente, con la que se quiere brindar a este las facilidades para gestionar adecuadamente su información médica, además de brindarle una experiencia de cuidado personalizada a través de una serie de funcionalidades como son: Signos vitales, Registros médicos, Recomendaciones médicas entre otras que serán mencionadas en el desarrollo del capítulo.

Desde la perspectiva del médico está la prueba de concepto titulada **hAppy Doc** la cual pretende principalmente apoyar al médico en el seguimiento de sus pacientes, por medio del control y el manejo eficiente de la información, para de este modo aumentar la calidad de la atención.

El ambiente de trabajo para estas dos pruebas de concepto es la plataforma en la nube *IBM Bluemix*, los requisitos de funcionamiento de estas pruebas de concepto exigen que sean aplicaciones web y que además cumplan con el diseño *Responsive design*.

6.2 MARCO TEÓRICO

6.2.1 Aplicaciones Web. Se denomina aplicación Web o *Web App*, a las aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de internet mediante el uso de un navegador.

La arquitectura de una aplicación web se basa en una estructura constituida por tres capas. La primera capa la proporciona el navegador web, la segunda capa la constituye un motor que tiene la capacidad de utilizar tecnologías web dinámicas (como es el caso de PHP) y la tercera capa la constituye una base de datos. De este modo el navegador envía peticiones a la capa dos, la cual proporciona información valiéndose de las consultas que realiza a la base de datos.

Las *Web App* son un tipo de aplicación cliente/ servidor en donde tanto el cliente (navegador web), como el servidor (servidor web) y el protocolo mediante el cual se comunican, son estandarizados y no deben ser creados por programador.

6.2.2 Prueba de concepto. Las Pruebas de Concepto (o *PoC*, por sus siglas en inglés) son desarrollos rápidos, y normalmente incompletos, realizados con el propósito de verificar que un concepto o una teoría es susceptible de ser llevada a la práctica.⁷

⁷ UTÓPICA INFORMÁTICA. (01 Julio de 2011). Pruebas de concepto [Mensaje en un blog]. Recuperado de: <http://blog.utopicainformatica.com/2011/07/pruebas-de-concepto.html>

El objetivo de la prueba de concepto es valorar el concepto de producto antes de comenzar su desarrollo a nivel técnico, se considera habitualmente un paso importante para crear un prototipo realmente operativo. En este sentido, la prueba de concepto se utiliza como demostración de una aplicación o servicio.

6.2.3 Bluemix. *IBM Bluemix* es una plataforma informática de nube basada en estándares abiertos desarrollada por IBM, soporta diversos lenguajes de programación (*PHP, Node.js Ruby* etc...) y servicios, así como también la metodología de desarrollo DevOps (*Development and Operations*) de forma integrada para crear, ejecutar, desplegar y gestionar aplicaciones en la nube de todos los tipos (*Web, Mobile, Big Data, etc...*).

Debido a que Bluemix está basada en la tecnología de *Cloud Foundry* (plataforma de código abierto en la que se pueden implementar y ejecutar aplicaciones en la nube) puede aprovechar un creciente ecosistema de *frameworks* y servicios de tiempo de ejecución.

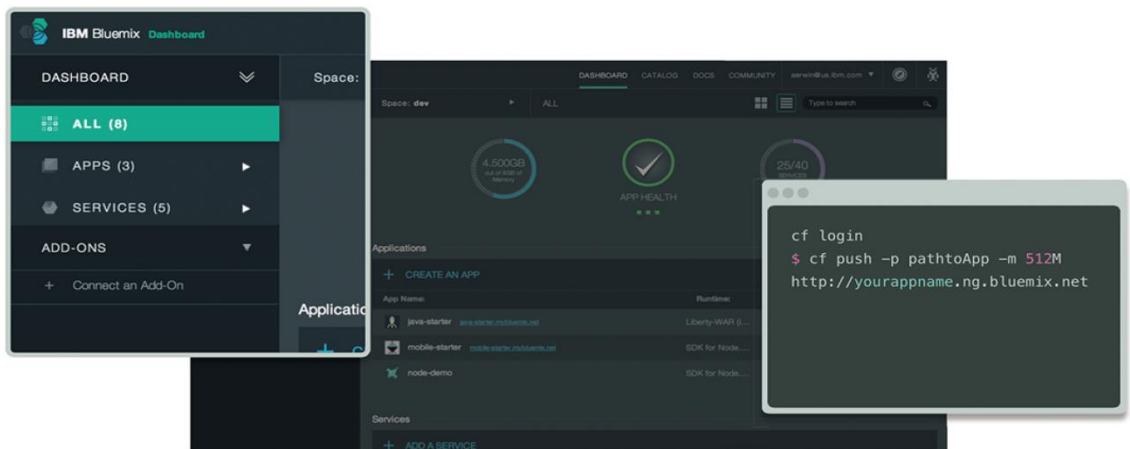
IBM Bluemix proporciona un panel de control en el que se pueden crear, ver y administrar aplicaciones y servicios, así como supervisar el uso de los recursos de la aplicación. El tablero de Bluemix ofrece la capacidad de administrar organizaciones, espacios y acceso de usuarios. Algunos de los servicios se entregan a través de *Cloud Foundry*, otros a través de IBM y de terceros proveedores.

6.3 DESARROLLO DE LA PRÁCTICA

Para dar claridad de la temática desarrollada en este capítulo, es necesario introducir en dos conceptos propios de Bluemix y son los siguientes:

- Apps: en el panel de control se proporciona todo lo necesario para ejecutar aplicaciones y gestionar estas mientras se ejecutan. Bluemix proporciona varios contenedores modelo y tiempos de ejecución.
- Servicios: en el panel de control se puede acceder a los servicios de Bluemix disponibles desde IBM y proveedores de terceros. A través de estos servicios el desarrollador puede fácilmente implementar, alojar y escalar sus aplicaciones. Algunos de los servicios que se incluyen son: *Watson*, *Internet of Things*, *Analytics*, *Mobile* y *DevOps*.

Figura 12. IBM Bluemix Dashboard



Fuente: IBM. Disponible en: <http://www.ibm.com/cloud-computing/pe/es/platform/>



Inicialmente se crearon las apps en la interfaz de usuario de Bluemix, seguido de esto se añadieron a cada una de ellas los servicios requeridos para la implementación de las pruebas de concepto:



- **ElephantSQL:** es un servicio de alojamiento de bases de datos *PostgreSQL*, que ofrece bases de datos que van desde servidores compartidos hasta configuraciones de servidores multi-nivel.
- **Internet of Things Platform:** es un servicio a través del cual se pueden construir aplicaciones y conectar dispositivos para el análisis de datos.
- **Conversation:** es un servicio que permite construir, probar y desplegar rápidamente un agente virtual que puede ser usado en las aplicaciones.

En lo referente a la codificación se emplearon varios lenguajes como: HTML5, CSS, JavaScript y el framework AngularJS, basándose en las funcionalidades que iban a tener las dos pruebas de concepto.

Luego de codificada complemente la aplicación, se hizo la gestión de la app la cual hace referencia al despliegue como tal de la prueba de concepto. Esto se realizó mediante el panel de control de la interfaz de usuario de IBM Bluemix la cual puede dar como resultado varios estados que se detallan en la siguiente tabla:

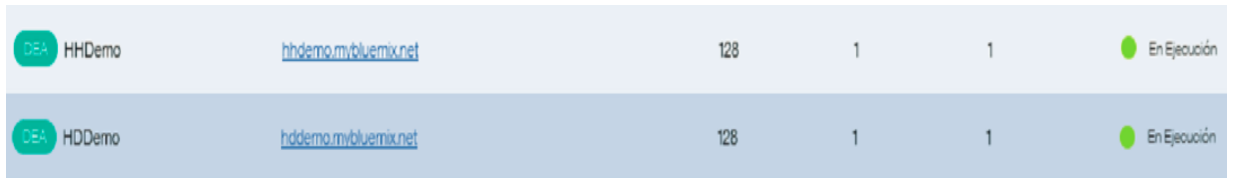
Tabla 1. Indicadores de estado





Estado	Color Asociado	Significado
Detenido o desconocido		La aplicación está detenida o el estado de esta es desconocido.
En Ejecución		La aplicación se ha iniciado y se están ejecutando todas las instancias.

Estado	Color Asociado	Significado
Numero en ejecución		La aplicación se ha iniciado, pero no se están ejecutando todas las instancias.
No se está ejecutando		La aplicación se ha iniciado, pero no hay ninguna instancia en ejecución.

Las dos pruebas de concepto que se han mencionado en el desarrollo de esta capítulo fueron desplegadas exitosamente con todas sus instancias, lo que indica que están disponibles para la visualización y uso de los usuarios (ver resultados).

Figura 13. Estado de las pruebas de concepto



 hHDemo	hhdemo.mybluemix.net	128	1	1	 En Ejecución
 hDDemo	hddemo.mybluemix.net	128	1	1	 En Ejecución

6.4 RESULTADOS

Una vez concluido el desarrollo y el despliegue se obtuvieron las dos pruebas de concepto en la plataforma *cloud IBM Bluemix* de las cuales se puede visualizar a continuación, para tener una mejor comprensión de la funcionalidad de cada vista de las pruebas de concepto se presentan las siguientes tablas acompañadas de su respectiva captura de pantalla.

Para la prueba de concepto **hAppy Health** se crearon: 11 archivos .html, 3 archivos .js, y un archivo .css (aportan el estilo) por cada .html para la prueba de concepto hAppy Health.

Para **hAppy Doc** se crearon: 8 archivos .html, 5 archivos .js y un archivo .css (aportan el estilo) por cada .html creado.

6.4.1 hAppy Doc:

Tabla 2. Etapas de la prueba de concepto hAppy Doc

Nombre etapa	Funcionalidad	Pantallas
REGISTRO	<ul style="list-style-type: none"> El doctor registra en el sistema información tanto personal como profesional. 	Figura 15
INICIO SESION	<ul style="list-style-type: none"> El doctor accede desde su dispositivo móvil a la página principal de hAppy Doc, donde ingresa con los datos registros previamente. 	Figura 16
MI PERFIL	<ul style="list-style-type: none"> El doctor ingresa a la vista principal donde puede ver toda su información y verificar si es correcta y está actualizada. 	Figura 17
BÚSQUEDA DE PACIENTE	<ul style="list-style-type: none"> El doctor realizar una consulta en la base de datos de pacientes, para obtener información relacionada con este, su historial de signos vitales, así como también le permite agregar recomendaciones si lo considera pertinente. 	Figura 18
PERFIL DEL PACIENTE	<ul style="list-style-type: none"> El doctor podrá tener información básica de su paciente, antes de tener una consulta con el 	Figura 19
HISTORIAL SIGNOS VITALES	<ul style="list-style-type: none"> El doctor podrá mantenerse informado de la última toma de signos vitales de sus pacientes. 	Figura 20
AGENDA	<ul style="list-style-type: none"> El doctor ingresa a su agenda, donde podrá visualizar que pacientes han reservado una cita y verificar que tipo de consulta es. 	Figura 21

Figura 14. Registro hAppy Doc

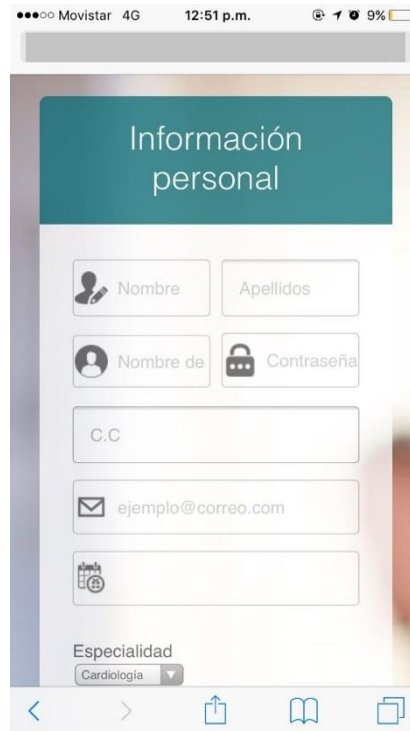


Figura 15. Inicio de sesion hAppy Doc



Figura 16. Mi perfil hAppy Doc



Figura 17. Búsqueda de paciente hAppy Doc



Figura 18. Perfil del paciente



Figura 19. Historial de signos vitales

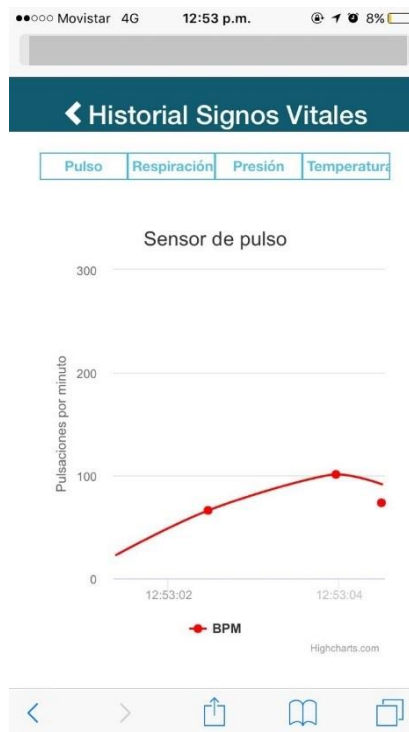


Figura 20. Agenda hAppy Doc



6.4.2 Happy Health:

Tabla 3. Etapas de la prueba de concepto hAppy Health

Nombre etapa	Funcionalidad	Pantallas
REGISTRO	<ul style="list-style-type: none"> El paciente suministra su información básica, además de registrar algunas de sus condiciones médicas o datos que pueden ser útiles para su doctor. 	Figura 22
LOGIN	<ul style="list-style-type: none"> El paciente accede desde su dispositivo móvil a la página principal de hAppy Health, donde ingresa con los datos registros previamente. 	Figura 23
MI PERFIL	<ul style="list-style-type: none"> El paciente ingresa a la vista principal donde puede ver toda su información médica más 	Figura 24

Nombre etapa	Funcionalidad	Pantallas
	relevante, que previamente fue suministrada por el	
INICIO	<ul style="list-style-type: none"> • Llamar a emergencias 	Figuras: 25, 26, 27, 28
	<ul style="list-style-type: none"> • Pre-Triage 	
	<ul style="list-style-type: none"> • Signos vitales 	
	<ul style="list-style-type: none"> • Citas programadas 	
AGENDAR CITAS	<ul style="list-style-type: none"> • El paciente puede realizar una búsqueda del doctor de su preferencia, y agendar una cita con este. 	Figura 29, 30
RECOMENDACIONES MEDICAS	<ul style="list-style-type: none"> • El paciente puede visualizar las recomendaciones que sus médicos le han realizado, de acuerdo a sus consultas, o afecciones presentadas 	Figura 31

Figura 21. Registro de hAppy Health

Información personal

Nombre Apellidos

Nombre de usuario Contraseña

C.C.

ejemplo@correo.com

Contacto de emergencia

Grupo Sanguíneo AB+ Aseguradora Seguros Bolívar S.A.

Donación de órganos No

Kg Cm

¿Ya tienes una cuenta? [¡¡¡¡¡](#)

Figura 22. Inicio de sesión hAppy Health



Figura 23. Perfil del paciente hAppy Health

☰ Mi Perfil



Nombre	Guillermo
Apellidos	Niño
Número de identificación	1095456785
Fecha de nacimiento	01/21/1982
Condiciones médicas	Ninguna
Alergias y reacciones	Alérgico al polvo y a la aspirina
Grupo Sanguíneo	A+
Medicamentos	Ninguno
Peso	85 kg
Estatura	178 cm
Aseguradora	Sura

Figura 24. Inicio hAppy Health

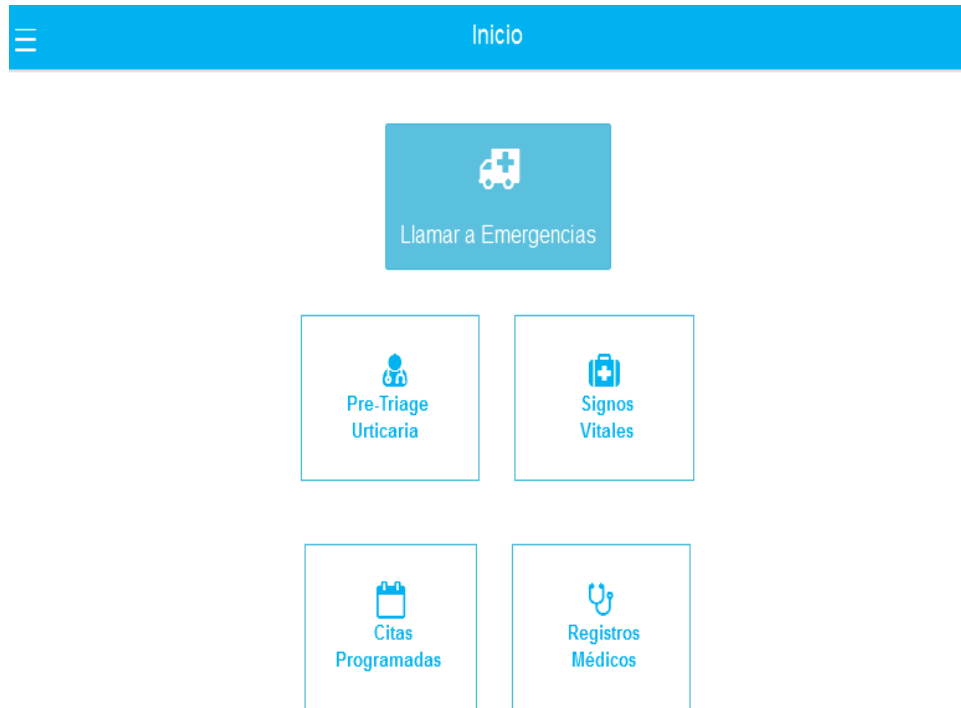


Figura 25. Signos vitales hAppy Health

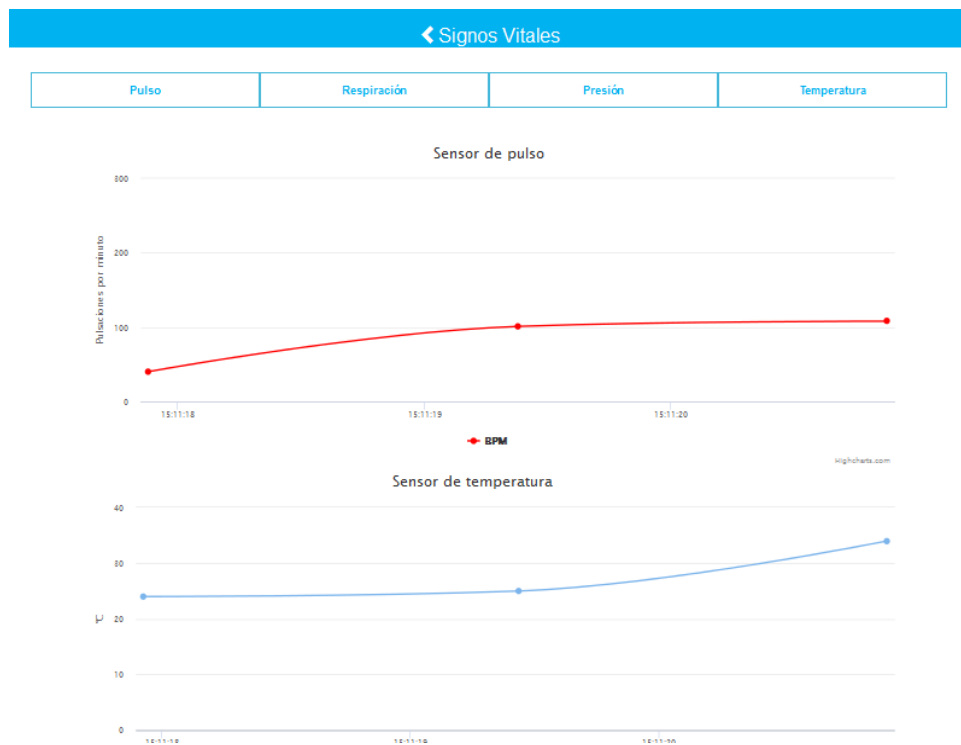


Figura 26. Citas Programadas hAppy Health

← Citas Programadas

Su Doctor:
Sebastian Ruiz Ayala

Fecha de la cita

25
Febrero

Hora	Procedimiento
2:05 pm	Control Mensual

Señor(a) paciente, por favor llegar 10 minutos antes de la hora programada. Recuerde que si no puede asistir a su cita, tiene hasta 2 horas antes de la hora programada para cancelarla.

Figura 27. Registros Médicos hAppy Health

← Registros Médicos

Información Básica

Paciente	Guillermo Niño		
Fecha de Nacimiento	01/21/1982	Género	♂ Masculino

+ Lista de problemas

Nombre	Fecha	Estado
Asma	15 de enero 2018	Activo

+ Alergias, Reacciones Adversas

Alérgeno	Reacción	Severidad Reacción
Alergia al polvo	Picor en los ojos, estornudos	Media
Acido acetilsalicílico	Inflamación de las extremidades, urticaria	Alta

+ Medicamentos

Fármaco	Instrucción	Dosis	Estado
Albuterol 0-09 MG	2 puff cada 6 horas		Inactivo

+ Exámenes

Resultado	Valor	Rango	Interpretación
CO2	27	23-29 mmol/L	normal

+ Signos Vitales

Fecha	Prueba	Resultado	Detalles
20 de Octubre 2016	Temperatura	30 °C	
13 de Noviembre 2016	Pulso	182 BPM	
17 de Noviembre 2016	Pulso	191 BPM	

+ Procedimientos

Sin información

+ Historia Social

Nunca ha fumado. Grabado 5 de Septiembre de 2018

+ Plan de Cuidado


Fecha	Nota	Instrucciones


Figura 28. Agendar Citas hAppy Health

Agendar Citas

Q Encuentre un doctor

5 Doctores encontrados

 Alberto Castañeda
Neurólogo

 Sebastian Ruiz Ayala
Cardiólogo



 María Isabel Rueda
Pediatra

Figura 29. Agendar Citas hAppy Health

Agendar Citas

 Dr. Sebastian Ruiz Ayala

Nació en Bogotá - Colombia el 16 de Diciembre de 1966 es un médico Cardiólogo. Se graduó como médico cirujano de la Pontificia Universidad Javeriana. Actualmente es catedrático de Cardiología en la escuela de medicina de la Universidad de Nueva York en la que es además director del departamento de Cardiología en la escuela de medicina de la Universidad.

Seleccione el centro Médico

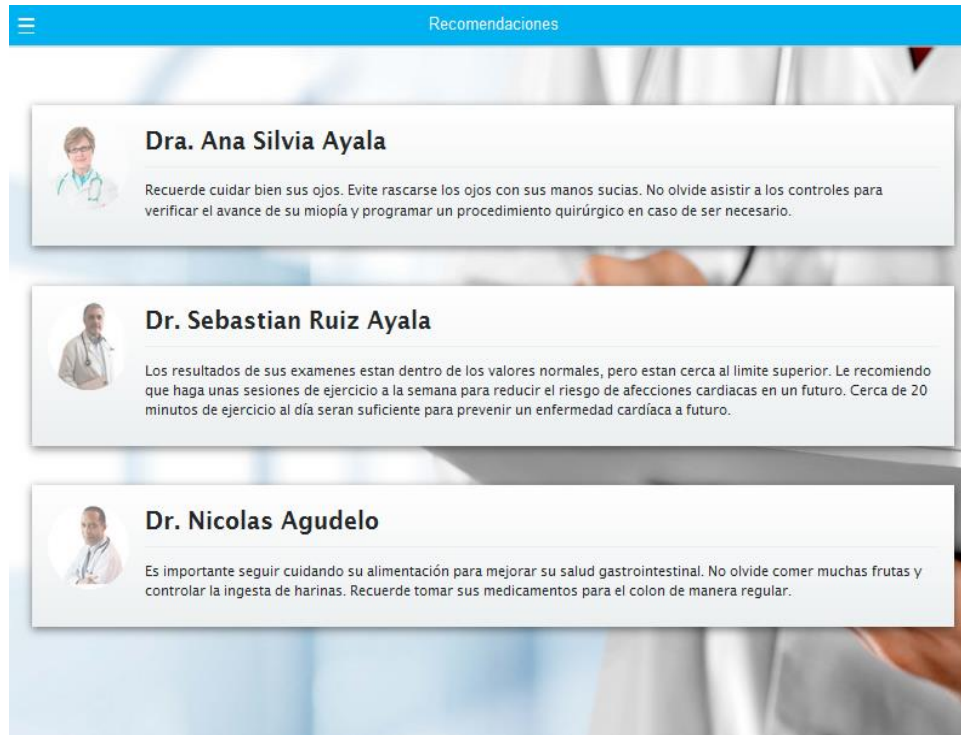
Clínica Chicamocha

Seleccione la fecha de la cita

← DIC		ENE 2017					FEB →	
Dom	Lun	Mar	Mié	Jue	Vie	Sab		
	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	28	
	29	30	31					

Hoy es Ene 15

Figura 30. Recomendaciones hAppy Health



The screenshot shows a mobile application interface with a blue header bar containing a menu icon and the text "Recomendaciones". Below the header, there are three white recommendation cards, each featuring a circular profile picture of a doctor, the doctor's name, and a short text-based recommendation.

Dra. Ana Silvia Ayala
Recuerde cuidar bien sus ojos. Evite rascarse los ojos con sus manos sucias. No olvide asistir a los controles para verificar el avance de su miopía y programar un procedimiento quirúrgico en caso de ser necesario.

Dr. Sebastian Ruiz Ayala
Los resultados de sus exámenes están dentro de los valores normales, pero están cerca al límite superior. Le recomiendo que haga unas sesiones de ejercicio a la semana para reducir el riesgo de afecciones cardíacas en un futuro. Cerca de 20 minutos de ejercicio al día serán suficientes para prevenir una enfermedad cardíaca a futuro.

Dr. Nicolas Agudelo
Es importante seguir cuidando su alimentación para mejorar su salud gastrointestinal. No olvide comer muchas frutas y controlar la ingesta de harinas. Recuerde tomar sus medicamentos para el colon de manera regular.

7. IBM CARE MANAGEMENT

7.1 OBJETIVO ESPECÍFICO 3

El objetivo específico 3 de la práctica empresarial se definió de la siguiente manera:

“Programar el código fuente requerido para personalizar la solución IBM Care Management a los diferentes usuarios que lo soliciten, utilizando el lenguaje de programación Java para el back-end y XML para el front- end”.

Actualmente IBM dispone de la aplicación software *IBM Care Management* para atender las necesidades de sus usuarios en el área de salud. En diversas ocasiones, los clientes requieren de adaptaciones del producto de acuerdo a las necesidades propias de su empresa, adaptaciones que involucran programación. Durante la práctica empresarial, IBM solicitó la realización de 4 implementaciones también llamadas historias de usuario (HU) las cuales son:

- Creación de persona
- Búsqueda de persona
- Configuración de un proveedor
- Cargue Batch de proveedor

7.2 MARCO TEÓRICO

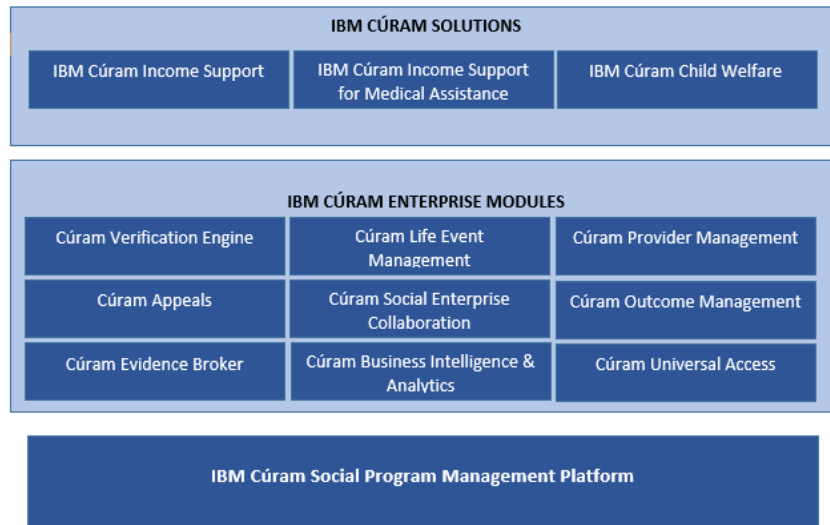
7.2.1 Cúram Software. Curám que traducido al español significa “Cuidado y protección” es un software que inicialmente pertenecía a la compañía irlandesa llamada Curám, la cual actualmente forma parte de IBM. Cúram puede entenderse

como una suite de productos con la que se apoya todo tipo de programas orientados al ciudadano, incluyendo servicios laborales, salud y servicios sociales, entre otros.

“La estructura de IBM *Cúram* está constituida por varias capas horizontales (ver figura), donde la plataforma IBM *Cúram Social Program Management* (SPM) es obligatoria. La plataforma SPM alberga la infraestructura y funcionalidad básica de IBM *Cúram social Program Management*. IBM *Cúram Enterprise Modules* proporciona características específicas que se pueden utilizar para crear soluciones personalizadas.

IBM *Cúram Solutions* son módulos de procesos de negocio diseñados para abordar los requerimientos de soluciones para sectores específicos”⁸.

Figura 31. Estructura de *Cúram*

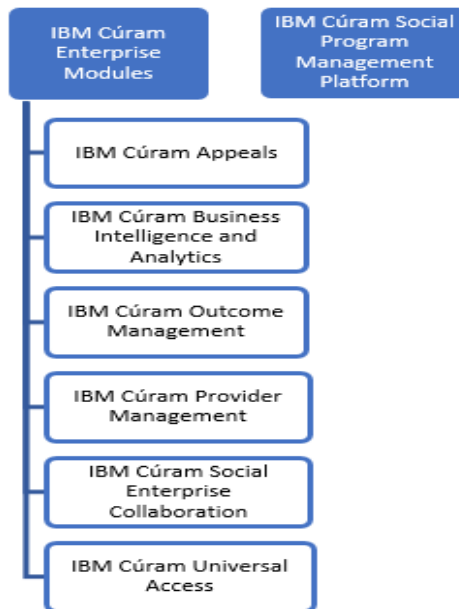


Curám se caracteriza por tener una arquitectura escalable y fácilmente adaptable debido al uso de tecnologías estándares de la industria, gracias a esto provee flexibilidad, rendimiento en la implementación de las aplicaciones.

⁸ IBM CORPORATION. Fundamentals of the IBM *Cúram* Platform for Developers 6.0.4. 2012.

Particularmente para la instalación y el desarrollo se hizo uso de dos de las capas de la estructura de Cúram como se detalla a continuación:

Figura 32. Descripción de la instalación del producto

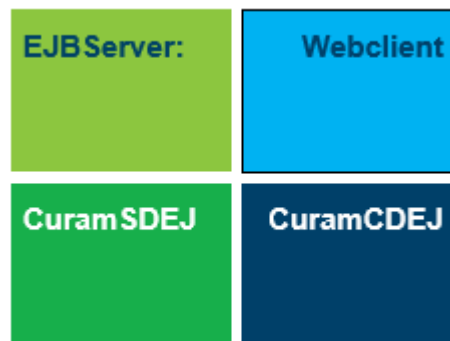


7.2.2 IBM Care Management. “Es una aplicación de software empaquetada que ofrece capacidades clave necesarias para gestionar la asistencia en la atención continuada. Permite identificar clientes que precisan asistencia, evaluar sus necesidades, establecer el plan de asistencia adecuada para responder a sus necesidades, además de gestionar la asistencia y supervisar los resultados obtenidos”⁹.

⁹ IBM. IBM Care Management. [en línea]. [Citado 29 Enero, 2017]. Disponible en: <http://www-03.ibm.com/software/products/es/ibm-care-management>.

IBM Care Management combina la integración de datos, el análisis y la coordinación de atención en una única oferta lista para usar que ofrece una visión personalizada de 360 grados de la persona para facilitar el cuidado centrado en los resultados.

Figura 33. Aplicaciones de IBM Care Management



IBM Care Management contiene las cuatro aplicaciones que se visualizaron en la figura anterior donde:

- EJBServer: corresponde al Backend
- WebClient: corresponde al Frontend
- CuramSDEJ: generación del Backend
- CuramCDEJ: generación Frontend

IBM Care Management está desarrollada bajo una arquitectura de componentes, esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas.

7.2.3 IBM Rational Software Architect for WebSphere Software. IBM Rational Software Architect es una herramienta integral de diseño, modelado y desarrollo para la entrega de software de extremo a extremo. Utiliza el Lenguaje de modelado

unificado (UML) para diseñar aplicaciones y servicios web. Rational Software Architect se basa en el marco de software de código abierto de Eclipse y es extensible con una variedad de complementos de Eclipse. También puede mejorar la funcionalidad para sus requisitos específicos con las extensiones Rational compradas por separado.

Rational Software Architect le ayuda a mantener un mejor control de la arquitectura y los resultados de entrega con estos beneficios:

- El soporte de modelado basado en UML y las herramientas de desarrollo impulsado por modelo (MDD) ayudan a agilizar la creación de aplicaciones y servicios Java y Web 2.0.
- Herramientas potentes y guía de procesos ayudan a reducir la complejidad y brindan soporte a una mayor calidad y eficiencia.
- El acceso a los servicios en la nube le permite aprovechar los servicios de infraestructura escalables.
- Una plataforma flexible y extensible le ayuda a ofrecer software de alta calidad con un retorno de la inversión (ROI) más rápido¹⁰.

7.3 DESARROLLO DE LA PRÁCTICA

Para tener una mejor comprensión del presente capítulo se debe aclarar que la personalización y el desarrollo de nuevas funcionalidades en IBM Care Management (ICM) se realizaron en el componente “Custom”. Dentro de este componente deben ir todos los desarrollos que no sean de la solución estándar.

¹⁰ IBM. Rational Software Architect Designer. [en línea]. [Citado 29 Enero, 2017]. Disponible en <http://www-03.ibm.com/software/products/es/ratsadesigner>.

Pero algunos desarrollos también se implementaron en el componente “CareCoordination” debido a reglas del negocio.

En este capítulo se detallará lo realizado en 4 Historias de usuario en particular, pero antes de esto se debe entrar en contexto con el proceso estándar para el desarrollo (ver tabla 5) y una serie de conceptos referentes al modelado en IBM Care Management:

- **Domain Definitions:** Dominio en específico para un atributo ya sea de una entidad o de una estructura. Se basa en los tipos de datos primitivos: Int, Long, Char, String.
- **Structs:** Elemento del modelado que representa un conjunto de atributos. No es persistente en base datos y no contiene métodos. Son simplemente utilizadas como parámetros de entrada y salida de los métodos.
- **Entity:** Elemento del modelado que representa un objeto persistente en base de datos. Cada entidad en Curám es una tabla en la base de datos. Sus atributos pueden ser de tipo “detalles” o de tipo “llave”.
- **Facade:** La fachada en Curám es el elemento que permite modelar los procesos de negocio. La fachada solo contiene método, los parámetros de entrada y de salida de estos métodos son las estructuras.
- **Codetable:** Son valores en Curám a los cuales se quiere estandarizar estos y se quiere restringir el valor de un campo.
- **Properties:** Funcionan como valores parametrizables que pueden cambiar en el tiempo y son utilizados en el desarrollo de nuevas funcionalidades.

Tabla 4. Proceso estándar para el desarrollo en ICM

Tipo de proceso	Procedimiento
Modelo	<ul style="list-style-type: none"> • Modelar con RSA • Guardar el modelo
Comando	<ul style="list-style-type: none"> • Build generated • Build database • Build bean-doc
Código	<ul style="list-style-type: none"> • Codificar en eclipse
Código	<ul style="list-style-type: none"> • Crear los archivos UIM's
Comando	<ul style="list-style-type: none"> • Build client

Tabla 5. Comandos para la generación Backend y Frontend

Aplicación donde se ejecuta	Comando	Función
EJBSERVER	Build server	Genera el código de la aplicación: Clases, codetables, propiedades).
EJBSERVER	Build database	Genera las nuevas tablas en la base de datos.
WEBCLIENT	Build beandoc	Actualiza y crea la versión HTML del ServerAccessBean.xml
WEBCLIENT	Build client	Construye el Frontend de Curám, donde se hace el proceso de transformación de los UIM a páginas.


7.3.1 HU Creación de persona. La HU tenía por objetivo la creación de una persona en Curám, con los atributos de información requeridos para su identificación dentro del sistema, de tal forma que permita la creación de casos asociados a dicha persona, de acuerdo a las reglas de negocio.


- Descripción del proceso:


El usuario ingresa a la función de “Registro” en el menú de atajos y elige la opción “Persona” realiza la búsqueda en Curám con base a determinados criterios, si la persona ya está creada en el sistema, se muestra la grilla de resultados con los campos que respaldan la existencia de esta persona en el sistema.

Si la persona no aparece registrada, el sistema permite acceder a una pantalla en la cual se crea el registro con la información de la persona.

El desarrollo de la HU de usuario a pesar de ser una implementación que no se encontraba en la solución estándar, se creó en el componente *CareCoordination*. A través de la herramienta *IBM Rational Software Architect For WebSphere Software* se modelaron los siguientes artefactos y se continuó con el proceso estándar para la implementación (ver tabla 4).

 **DOMAIN DEFINITIONS:** se crearon dos nuevos Domain Definitions llamados: PAIS y ENTIDAD_CLIENTE

 **FACADE:** se creó una nueva fachada llamada “CreacionPersona”, la cual contiene dos operaciones: crearPersona () y leerPersona ().

 **STRUCTS:** se creó la estructura “DatosCrearPersona” la cual contiene los parámetros de entrada para el método de crearPersona (), para el método de

LeerPersona () se reutilizaron estructuras ya disponibles en RSA que contiene los parámetros de búsqueda para la persona.

Después de la creación del modelo, se ejecutan los comandos mostrados en la tabla 4, los cuales generan una esquelética de la clase en java en la cual se codificará la lógica de la historia de usuario.

Al estar la implementación completa se realizaron pruebas dos pruebas mediante la herramienta Junit la cual hace referencia a un conjunto de bibliotecas utilizadas en programación mediante el cual se evalúa el funcionamiento correcto de cada uno de los métodos de las clases en Java.


Las pruebas unitarias mostraron el comportamiento esperado de la clase, por lo que se prosigió a iniciar Cúram a través de: Start Server, RMILoginClient, TomCat Web Server con el fin de visualizar y probar completamente la implementación (ver resultados).


7.3.2 HU Búsqueda de persona. La HU tenía por objetivo proporcionar una pantalla de búsqueda de las personas y sus riesgos asegurados, con base en la aplicación de varios criterios de búsqueda y con algunos propósitos definidos como: el acceso a la vista 360, la visualización de información básica de la persona, ingresar a la creación de un caso.

- Descripción del proceso:

El usuario ingresa a la “búsqueda de persona” y visualiza los criterios de búsqueda que fueron definidos en la creación de la HU que básicamente son dos: por asegurado o por tomador de la póliza, si la persona existe en el sistema se tendrá una lista de salida que contiene determinadas columnas.

De igual modo que la HU anterior, el desarrollo se realizó en el componente CareCoordination y a través de la herramienta *IBM Rational Software Architect For WebSphere Software* se modelaron los siguientes artefactos:

 **FACADE:** se creó una nueva fachada llamada “BusquedaPersona”, la cual contiene la operación de buscarPersonas ().

 **STRUCTS:** se creó la estructura “BusquedaDePersona” la cual contiene los parámetros de búsqueda del método buscarPersonas (), también se creó la estructura ResultadosBusquedaDePersonas la cual contiene los parámetros que visualizara el usuario una vez realice la búsqueda (grilla de resultados).

Al tratarse de un proceso estándar en el desarrollo, se ejecutan los comandos mostrados en la tabla 4, los cuales fueron también mencionados en la Historia de usuario anterior.

Como se hizo mención en la HU anterior se realizó una prueba con la herramienta Junit para verificar el buen funcionamiento de la clase en java y sus métodos asociados.


Las pruebas unitarias mostraron el comportamiento esperado de la clase, por lo que se prosiguió a iniciar Cúram a través de: Start Server, RMILoginClient, TomCat Web Server con el fin de visualizar y probar completamente la implementación (ver resultados).


7.3.3 HU Configuración de proveedor. La HU tenía por objetivo la creación y configuración de un proveedor de servicios a través del portal Cúram, indicando información administrativa, financiera y tributaria de cada uno de los proveedores.


- Descripción del proceso:


El usuario encargado de la gestión de recursos ingresa al sistema y accede al menú de atajos a la opción de Proveedores y a la opción de inscribir nuevo proveedor. El sistema muestra un wizard (Barra de menú de progreso) “inscribir proveedor” con cuatro secciones que corresponden a la información requerida para la creación del proveedor.

El desarrollo de la HU se realizó en el componente “*Custom*” en el cual idealmente se crean las implementaciones no estándar de la solución. A través de la herramienta RSA se crearon algunos Domain Definitions para ser usados en 4 nuevas entidades que se adicionaron, se creó una nueva fachada con sus respectivas operaciones, así como también se hizo uso de estructuras ya existentes con algunas modificaciones de las cuales se detallara a continuación:

 **DOMAIN DEFINITIONS:** se crearon 20 nuevos Domain Definitions, con el fin de ejemplificar se hace mención de algunos a continuación: ACCOUNTENTITY, ACCOUNTNAME, ACCOUNTNUMBER, ACCOUNTTYPE, CITY, CONTACTNAME, CONTACTROLE, CONTRIBUTORTYPE, COUNTRY, EXEMPTICA, EXPMPIVA, REGIME, WEBSITE.

 **ENTITY:** se crearon 4 nuevas entidades, que traducen a tablas en la base de datos, donde se va a almacenar información propia del proveedor.

 **FACADE:** se creó una nueva fachada llamada “Proveedor”, la cual contiene dos operaciones: detallesProveedor () y busquedaProveedor ().

 **STRUCTS:** para esta HU no se modelaron estructuras nuevas, sino que se re utilizaron estructuras que ya estaban disponibles en RSA a las cuales se le adicionaron algunos parámetros faltantes.

De igual manera y como ya se ha hecho mención anteriormente se realizaron dos pruebas unitarias con la herramienta Junit, una asociada a la creación y la otra asociada a la búsqueda del proveedor. Las pruebas se realizaron de manera exitosa por lo que el siguiente paso fue iniciar Cúram en el entorno local como ya se ha mencionado anteriormente.

7.3.4 Cargue Batch de proveedor. La HU tenía por objetivo la creación y configuración de un proveedor de servicios a través de un proceso de cargue masivo de información. En Cúram el proceso Batch hace referencia a un procesamiento por lotes en donde se realiza una ejecución sin el control o supervisión directa del usuario.

- Descripción del proceso:

La información del proveedor se almacena en un archivo de carga que tendrá ciertas características:

- **Nombre:** Cargue_inicial_proveedores_sb
- **Formato:** csv (extensión de formato propia de Cúram)
- **Separador:** “;” (símbolo que hace la distinción entre un campo y otro)

A diferencia de las otras Historias de usuario que se han mencionado en este capítulo, para la implementación de la HU Cargue Batch se modela en RSA un: “process” con un método de tipo “Batch”, este método recibe de entrada una estructura que tiene como atributos el directorio donde se encuentra ubicado el

archivo que contiene la información a cargar y el directorio de los archivos log donde se registran los errores que se presentaron al realizar la carga masiva.

Las validaciones en un proceso Batch pueden ser de dos tipos:

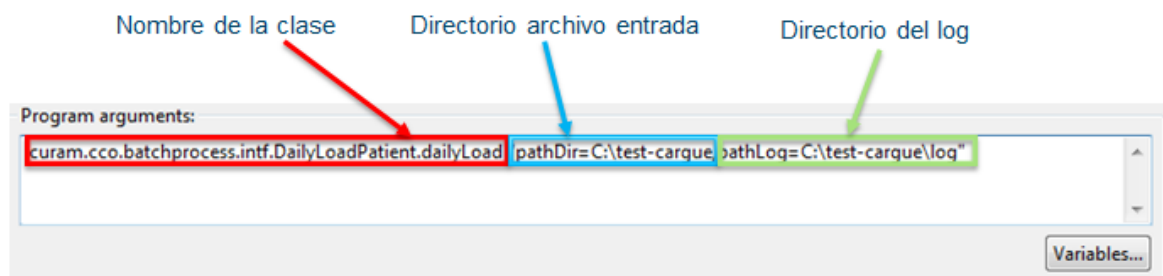
- Validación de estructura: consiste en validar la estructura del archivo, como el número de campos que deben ir en el archivo, el tipo de datos de cada campo, si es obligatorio o no, y la longitud del campo.
- Validación de negocio: consiste en hacer las validaciones de negocio, como validar si el usuario que se está creando ya existe, si los datos suministrados del usuario son válidos.

Figura 34. Ejemplo sintaxis archivo de carga masiva

```
NOMBRE_PROVEEDOR, TIPO_ID, NUMERO_ID, DIGITO_VERIFICACION, PAIS, DEPARTAMENTO, DIRECCION, CODIGO_POSTAL, COMUNICACION_PREFERIDA, PREFIJO_NACION
Proveedor;CC;1098753117;6;169;5001;CR 21B # 111 - 61;32257;PH4;5;7;6046689;111;310287111;JULIAN MANTILLA;INGENIERO;correo@co.net.co;sl
Proveedor1;CC;1098753116;;169;5001;CLL 16 # 25-60;;PH4;5;7;6046333;;;;;correo1@co.net.co;;cuenta2;9876500090;SIT Branch;BA11;1;1;1;0;
Proveedor3;CC;1098753116;;169;5001;CLL 16 # 25-60;;PH4;5;7;6046333;;;;;correo1@co.net.co;;cuenta2;9876500090;SIT Branch;BA11;1;1;1;0;
```

Luego de codificadas las respectivas validaciones en las clases, se continuó con el proceso de carga para el cual se configuro unos parámetros en la IDE Eclipse de la siguiente manera:

Figura 35. Configuración del proceso Batch



Al realizar el proceso de carga la consola indica cuantos archivos fueron registrados exitosamente y cuantos registros fueron fallidos.

Para identificar las causas por las que los registros fallaron al ser cargados. Se hizo uso del registro de logs de errores, los cuales se encontrarán almacenados en la base de datos con el fin de mantener una trazabilidad de este proceso.

A diferencia de las otras Historias de usuario que se trataron en este capítulo, en la HU Cargue Batch de proveedor no se realizó ninguna prueba unitaria debido a que en este tipo de implementaciones no se requiere de este tipo de pruebas.

7.3.5 Resultados. Las figuras presentadas en esta sección de resultados fueron obtenidas directamente desde el entorno local de desarrollo para el cliente, con el cual se prueba el correcto funcionamiento de las implementaciones para que luego estas puedan ser utilizadas por el usuario final.

7.3.6 HU Creación de persona. En la IDE Eclipse se creó la clase "CreacionPersona" la cual contiene 464 líneas de código que forman la lógica del Backend, 5 archivos XML con extensión .uim asociados al Frontend y dos nuevas Codetables. Por cada archivo .uim que se creó, se tiene un archivo de propiedades conocido dentro de Cúram como un archivo .properties en los cuales se define los campos que va a visualizar el usuario final.

Figura 36. Comprobación de persona

?
x

1 Comprobación de persona inscrita 2 Inscripción

Paso 1: Comprobación de persona inscrita
 Realice esta búsqueda para comprobar si el cliente ya está registrado como cliente o está registrado como persona.

Criterios de búsqueda

Tipo de documento Número de documento

Criterios de búsqueda adicionales

Nombre

Resultados de la búsqueda (Número de elementos: 1)

Nombres y apellidos	Tipo de documento	Número de documento
Lina Maria Rodriguez De la rosa	Cédula de Ciudadanía	63486071

Figura 37. Inscripción de Persona

?
x

1 Comprobación de persona inscrita 2 Inscripción

Paso 2: Inscripción

Tipo de documento * Número de documento *

Primer Nombre * Segundo nombre

Primer Apellido * Segundo Apellido

Fecha de nacimiento * Género *

Dirección de correo electrónico Entidad cliente *

Dirección privada

Pais * Dirección

Departamento *

Dirección postal

Especifique una dirección postal correo si es diferente de la dirección privada.

Número de teléfono

Número de teléfono Número de celular *

7.3.7 HU Búsqueda de persona. En el desarrollo de la HU Búsqueda de persona se creó la clase en la IDE Eclipse “BusquedaPersona” la cual contiene 347 líneas de código que forman la lógica al Backend, 2 archivos XML con extensión .uim asociados al Frontend.

Por cada archivo .uim que se creó, se tiene un archivo de propiedades conocido dentro de Cúram como un archivo .properties en los cuales se define los campos que va a visualizar el usuario final.

Figura 38. Búsqueda por tomador de la póliza

Formulario de búsqueda por tomador de la póliza. Incluye campos para Tipo de documento, Nombre Asegurado, Número de documento, Identificación Riesgo, Tipo de documento, Nombre Tomador, y botones de 'Buscar' y 'Restablecer'. Muestra resultados de la búsqueda (Número de elementos: 2).

Tipo de documento	Número de documento	Nombre Asegurado	Ramo	ID Riesgo	Riesgo	Nombre Tomador	Nuevo Caso
Cédula de Ciudadanía	109873116	Karen Gomez Calderon	Salud	4	Riesgo1	Laura Rodriguez Ariza	Nuevo Caso ...
Cédula de Ciudadanía	91154113	Daniel Diaz Perez	Salud	5	Riesgo2	Laura Rodriguez Ariza	Nuevo Caso ...

Figura 39. Búsqueda por asegurado

Formulario de búsqueda por asegurado. Incluye campos para Tipo de documento, Nombre Asegurado, Número de documento, Identificación Riesgo, Tipo de documento, Nombre Tomador, y botones de 'Buscar' y 'Restablecer'. Muestra resultados de la búsqueda (Número de elementos: 1).

Tipo de documento	Número de documento	Nombre Asegurado	Ramo	ID Riesgo	Riesgo	Nombre Tomador	Nuevo Caso
Cédula de Ciudadanía	109873116	Karen Gomez Calderon	Salud	4	Riesgo1	Laura Rodriguez Ariza	Nuevo Caso ...

7.3.8 HU Configuración de proveedor. En el desarrollo de la HU Configuración de Proveedor se creó la clase en la IDE Eclipse “Proveedor” la cual contiene 1156 líneas de código que forman la lógica al Backend, 3 archivos XML con extensión .uim asociados al Frontend.

Por cada archivo .uim que se creó, se tiene un archivo de propiedades conocido dentro de Cúram como un archivo .properties en los cuales se define los campos que va a visualizar el usuario final.

Figura 40. Inscripción de proveedor

Figura 41. Búsqueda de proveedor

Tipo ID	Numero ID	Nombre de proveedor	Categoria Principal	Nombre del propietario	Estado
NIT	91154113	PROVEEDOR 1	Profesional médico	Unauthenticated Test User	Abierta

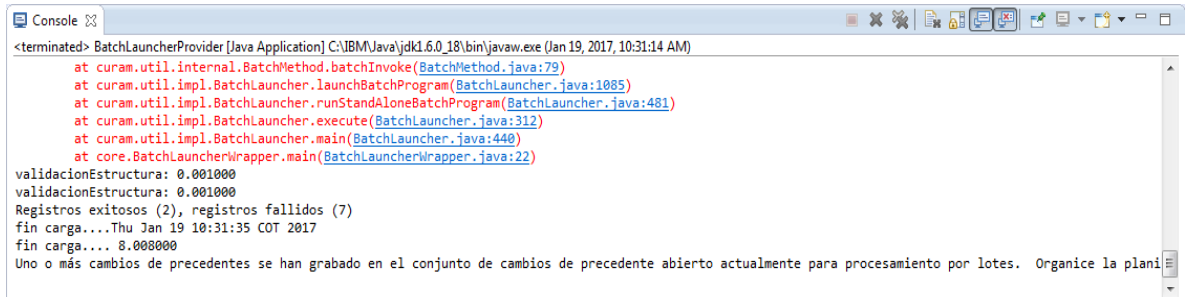
7.3.9 Cargue Batch de proveedor. En la HU Cargue Batch de proveedor no se pueden visualizar resultados como en las otras Historias de usuario debido a que esta no involucra el *Frontend*, sino solo el *Backend*, la única evidencia visual que se puede tener de este proceso son los registros que quedan almacenados en la base de datos, tanto los registros exitosos como los registros fallidos producto de la carga masiva.

En particular para la codificación de la HU Cargue Batch de proveedor se crearon 3 clases en la IDE Eclipse:

- **LoadProviderInfoStructureValidation:** clase en la que se realizan las validaciones de estructura, contiene 188 líneas de código fuente.
- **LoadProviderInfoBusinessValidation:** clase en la que se realizan las validaciones de negocio, contiene 350 líneas de código fuente.
- **ProviderLoader:** clase en la que se leen los datos del archivo de carga y se hace la inserción en las respectivas tablas de la base de datos, contiene 186 líneas de código fuente.

A diferencia de las demás, en esta HU no se crean archivos con extensión `.uim` para el *Frontend*, debido a que es un proceso que se ejecuta sin involucrar a un usuario final.

Figura 42. Resultados consola de carga masiva



```
<terminated> BatchLauncherProvider [Java Application] C:\IBM\Java\jdk1.6.0_18\bin\javaw.exe (Jan 19, 2017, 10:31:14 AM)
  at curam.util.internal.BatchMethod.batchInvoke(BatchMethod.java:79)
  at curam.util.impl.BatchLauncher.launchBatchProgram(BatchLauncher.java:1085)
  at curam.util.impl.BatchLauncher.runStandAloneBatchProgram(BatchLauncher.java:481)
  at curam.util.impl.BatchLauncher.execute(BatchLauncher.java:312)
  at curam.util.impl.BatchLauncher.main(BatchLauncher.java:440)
  at core.BatchLauncherWrapper.main(BatchLauncherWrapper.java:22)
validacionEstructura: 0.001000
validacionEstructura: 0.001000
Registros exitosos (2), registros fallidos (7)
fin carga...Thu Jan 19 10:31:35 COT 2017
fin carga... 8.000000
Uno o más cambios de precedentes se han grabado en el conjunto de cambios de precedente abierto actualmente para procesamiento por lotes. Organice la plani
```

8. MANTENIMIENTO Y SOPORTE

8.1 OBJETIVO ESPECÍFICO 4

El objetivo específico 4 de la práctica empresarial se definió de la siguiente manera:

“Brindar mantenimiento correctivo a los componentes de la solución *IBM Care Management*, los cuales implican la corrección de defectos, la actualización o adaptación de los modelos de acuerdo a las solicitudes de mejora reportadas en el sistema”.

El mantenimiento de software es la modificación de un producto de software después de la entrega, con el fin de corregir errores, mejorar el rendimiento u otros atributos

La temática de este capítulo está enfocada a: la resolución de los problemas existentes, y al soporte de las operaciones relacionadas con la solución *IBM Care Management*, la cual como se mencionaba en el capítulo 5 es una aplicación de negocio empaquetada que permite la coordinación de atención para diversas poblaciones.

8.2 MARCO TEÓRICO

8.2.1 Mantenimiento. El mantenimiento de software es la modificación de un producto software después de la entrega, para corregir errores, mejorar el rendimiento, u otros atributos.

El servicio de mantenimiento puede ser de tres tipos: Evolutivos (mejoras en los módulos que componen el sistema), Correctivos (resolución de problemas

existentes), Regulatorios (requerimientos normativos y de ley provenientes de entidades regulatorios).

En el mantenimiento correctivo se corrigen los defectos hallados en el software y que originan un comportamiento distinto al esperado. Estas fallas pueden ser causadas por distintos factores como: procesamiento, rendimiento, codificación, seguridad, estabilidad entre otras.

8.2.2 Solicitudes. Se designa con el término de “solicitud “a aquel documento a través del cual se solicita algo a alguien.

Cuando se trata de brindar mantenimiento correctivo y soporte a la operación, las solicitudes se pueden presentar en tres tipos de complejidades:

- Complejidad baja: requieren hasta 24 horas de esfuerzo por parte del recurso asignado, generan un impacto bajo en algunas de las operaciones. La infraestructura funciona con fallas mínimas.
- Complejidad media: requieren hasta 48 horas de esfuerzo por parte del recurso asignado, generan un impacto medio en algunas de las operaciones. La infraestructura se puede usar con algunas funcionalidades críticas.
- Complejidad alta: requieren hasta 96 horas de esfuerzo por parte del recurso asignado, afectan la operación del usuario final parcialmente, generan un impacto crítico en la operación. La infraestructura esta inoperable.

8.3 DESARROLLO DE LA PRÁCTICA Y RESULTADOS

En el desarrollo de la práctica empresarial no se definió un único periodo para el cumplimiento de este objetivo, sino que se llevó a cabo basándose en la demanda por parte del cliente.

Las solicitudes de mantenimiento correctivo fueron asignadas a través de documentos de especificaciones, donde se detallaba la solicitud que necesitaba ser solucionada, los actores involucrados, la prioridad que esta tenía y un flujo de eventos. En promedio se emplearon tres semanas para el mantenimiento y el soporte de los incidentes que equivalen 12,5 % de la totalidad de la práctica.

A continuación, se presenta la tabla 6 en la que se describe el tipo de mantenimiento y soporte a la operación que se realizó en la solución *IBM Care Management*.

Tabla 6. Descripción del mantenimiento y soporte de la operación

Descripción solicitud	complejidad	Solución Brindada
Notificación de reasignación de casos	Media	Se modificaron las clases implementadas en Java que involucran la reasignación de los casos de pacientes, agregando una nueva propiedad a través de la cual se controla las notificaciones que son enviadas al equipo, cuando se reasigna el caso de un paciente.
Notificación de carga masiva	Media	Se modificaron las clases implementadas en Java que involucran la carga de pacientes, agregando una nueva propiedad a través de la cual se controla las notificaciones que son enviadas al

Descripción solicitud	complejidad	Solución Brindada
		<p>gestor, cuando se realiza una carga masiva de pacientes.</p> <p>Si el número de pacientes cargados es inferior a 100, se envía una notificación al gestor, si el número de pacientes es superior a los 100 no se notifica esta carga.</p>
<p>Se presentan 3 clases en Java que no cumple con las normas de implementación estándar para el desarrollo, que indican que los métodos no pueden exceder un número determinado de líneas de código.</p>	<p>Baja</p>	<p>Se tiene como norma estándar para el desarrollo en IBM Care Management, que el número de líneas de código fuente permitidas por método no puede exceder las 200 líneas.</p> <p>A través de la creación de métodos privados y llamadas a estos métodos se corrigió esta mala práctica de programación que se presentó en 3 clases diferentes.</p>
<p>Ajuste de reporte gestión de casos</p>	<p>Media</p>	<p>Con el fin de hacerle seguimiento a los casos a los que llevan los determinados equipos, un gestor de casos puede generar un reporte en el que se detalla información del caso.</p> <p>Se notifica a través de un documento, que los reportes no se están realizando de manera adecuada, por ejemplo: si a un paciente se le reasigna el equipo al generar el reporte este es contado doble: en el antiguo equipo y el nuevo equipo.</p> <p>Para la solución de este incidente se estudian las clases que involucran la generación del reporte y se procede a hacer sus respectivas modificaciones.</p>
<p>Configuración de ambiente de pruebas en maquina SoftLayer</p>	<p>Media</p>	<p>Como apoyo en el soporte se realiza la instalación y configuración de un ambiente de pruebas para la solución</p>

Descripción solicitud	complejidad	Solución Brindada
		IBM Care Management en una maquina SoftLayer ¹¹
Reporte diario de máquinas SoftLayer	Baja	Como apoyo en el soporte y la operación asociada a la solución IBM Care Management se realiza un soporte a diario en el que se verifica la disponibilidad de los servicios del entorno de pruebas.

¹¹ SoftLayer es una compañía que fue adquirida por IBM que se especializa en entregar infraestructura como servicio, es un servidor dedicado, hosting administrado y proveedor de cloud computing.

9. CAPACITACIONES Y CUMPLIMIENTO DE OBJETIVOS

9.1 ADQUISICIÓN DE CONOCIMIENTO

Al inicio de la práctica empresarial recibí una serie de capacitaciones basadas en diferentes temáticas, a través de las cuales obtuve un primer acercamiento con la compañía, las capacitaciones estuvieron enfocadas a distintas áreas en particular en las que actualmente IBM centra sus esfuerzos y se detallan a continuación:

- Sector salud en Colombia:

Debido a que el centro de innovación en Santander está enfocado al sector salud, las capacitaciones iniciaron guiadas hacia: la comprensión de los principales conceptos de “salud”, cómo está constituido el sistema de salud en Colombia, cuales son los problemas que actualmente enfrenta el sistema de salud, cuales son los retos y oportunidades que tiene una compañía como IBM frente a esta situación.

- Internet of Things:

Al haber entrado en contexto con la problemática que enfrenta el sector salud en Colombia, en las capacitaciones se hace mención de un término muy importante y es el de “interoperabilidad” que no es más que la habilidad de dos o más sistemas o componentes para intercambiar y utilizar la información intercambiada. Actualmente en Colombia, en las diferentes entidades promotoras de salud e instituciones prestadoras de salud, los sistemas están totalmente desconectados por lo que la información valiosa de los usuarios se está desaprovechando.

Es ahí donde IBM encuentra una oportunidad de negocio con el “Internet de las cosas” la interconexión digital de los objetos cotidianos con internet, con los cuales se podría tener un mejor aprovechamiento de la información.

- IBM Bluemix

Una de las estrategias de IBM está enfocada a “*cloud*” donde IBM Bluemix juega un papel muy importante, por ser una plataforma como servicio que posee grandes capacidades y un amplio catálogo para el desarrollo de aplicaciones con las cuales facilita la productividad del desarrollador en la nube pública.

Se recibió una capacitación sobre el desarrollo en Bluemix por parte de un experto con el fin de adquirir las habilidades para desarrollar las pruebas de concepto que se mencionaron en el capítulo 4 del presente libro.

- Mobile

Mobile está presente en cada aspecto de nuestra vida, ha evolucionado para ser el método principal para el intercambio de información, bienes y servicios. Sacando provecho de la revolución móvil, IBM busca ofrecer soluciones móviles que unan las capacidades del *Frontend* y el *Backend* para entregar una experiencia atractiva y segura en cualquier dispositivo.

Partiendo de esto se recibe una capacitación sobre los conceptos básicos de las tecnologías móviles y el desarrollo multiplataforma, con el fin de obtener las habilidades para desarrollar la aplicación que da origen al objetivo 3 del presente proyecto.

- IBM Care Management

Inicialmente, la capacitación se llevó a cabo con una máquina virtual y una serie de manuales, con los cuales se adquirió bases para desarrollo en la solución *IBM Care Management* y el cumplimiento de los objetivos mencionados en los capítulos 5 y 6 del presente proyecto.

Las capacitaciones fueron guiadas por el consultor técnico de Cúram y tutor del presente proyecto.

9.2 CUMPLIMIENTO DE OBJETIVOS

Con el fin de ilustrar de manera más clara se presenta a continuación la tabla 7 donde se describen el cumplimiento de los objetivos

Tabla 7. Cumplimiento de objetivos

OBJETIVO	META	CUMPLIMIENTO
1. Implementar aplicación móvil en los distintos sistemas operativos (ver capítulo 3)	Implementación de una aplicación Móvil multiplataforma	100%
2. Realizar pruebas de concepto en Bluemix con Responsive Design (ver capítulo 4)	Implementación de las pruebas de concepto en la plataforma cloud IBM Bluemix	100%
3. Programar el código fuente requerido para personalizar la solución IBM Care Management (ver capítulo 5)	Desarrollo de las historias de usuario utilizando Java para el Backend y archivos XML	100%
4. Brindar mantenimiento correctivo a los componentes de la solución IBM Care Management (ver capítulo 6)	Se dio respuesta y cumplimiento a 6 tickets.	100%

10. CONCLUSIONES

La realización de la presente práctica empresarial, me permitió aplicar los conocimientos que adquirí en el transcurso de mi programa académico en proyectos de la vida real donde se emplearon nuevas tecnologías como lo son IBM Bluemix, IBM Care Management, a través de las cuales obtuve experiencia laboral y herramientas para enfrentar el mundo profesional de manera más competente.

El hacer parte de un equipo de trabajo multidisciplinario del centro de innovación de una empresa reconocida internacionalmente me permitió adquirir un conocimiento integral aplicado al sector salud con un enfoque técnico, asimismo me brindo una nueva perspectiva de cómo sacar provecho de las oportunidades y retos que enfrenta el sector salud en Colombia.

El aprovechamiento de tecnologías como Apache Cordova, me brindaron rapidez y facilidad en el momento de codificar. Además de ayudarme a hacer un uso óptimo de los recursos que tenía disponibles para la construcción de la aplicación móvil.

El autor constata que el acceso en tiempo real a datos vitales de un paciente, le permite al personal médico actuar sobre la información de manera más rápida para de este modo mejorar la calidad en el cuidado y seguridad de los pacientes.

11. RECOMENDACIONES

Realizar iniciativas académicas entre la escuela de ingeniería de sistemas e informática y compañías como IBM a través de las cuales se puedan dar a conocer temáticas nuevas que no estén incluidas en el pensum académico y que generen un valor agregado en la formación de los estudiantes.

Proponer a la universidad industrial de Santander la creación de una línea de investigación en donde el tema central sea la innovación en los distintos sectores industriales aplicada al uso de agentes cognitivos, internet de las cosas y analítica de datos.

Fomentar entre los estudiantes de ingeniería de sistemas el desarrollo de nuevas aplicaciones móviles de uso institucional con las cuales se impacte y se transforme la forma de llevar a cabo diferentes actividades dentro de la universidad.

BIBLIOGRAFIA

IBM CORPORATION. Bluemix Docs. [En línea]. [Citado 29 de Enero, 2017]. Disponible en: <https://console.ng.bluemix.net/docs/>.

IBM CORPORATION. Course Introduction. IBM Care Management 6.0 – Function Technical Overview.

IBM CORPORATION. Cloud Integration for Bluemix- User Guide. [Citado 29 de Enero, 2017]. Disponible en:
ftp://public.dhe.ibm.com/cloud/bluemix/cloudintegration/Cloud_Integration_for_Bluemix_User_Guide.pdf.

IBM CORPORATION. Fundamentals of the IBM Cúram SPM Platform for Developers 6.0.4.

IBM CORPORATION. Soluciones IBM para el Sector Salud [en línea]. [Citado 1 Julio, 2016]. Disponible en:
ftp://ftp.software.ibm.com/la/documents/gb/commons/Brochure_Soluciones_IBM_Sector_Salud.pdf.

IBM SOFTWARE SOLUTIONS. Soluciones de SWG IBM para una industria con nuevos desafíos. [En línea]. [Citado 29 de Enero, 2017]. Disponible en:
ftp://ftp.software.ibm.com/la/documents/imc/la/co/swg_bogota/salud/soluciones_de_swgibm_para_una_industria_con_nuevos_desafios.pdf

IBM SOFTWARE. Native, web or hybrid mobile-app development. [En línea]. [Citado 29 de Enero, 2017]. Disponible en:
<ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>.

M. WARGO, John. Apache Cordova 4 Programming. [En línea]. [Citado 29 de Enero, 2017]. Disponible en:
<http://ptgmedia.pearsoncmg.com/images/9780134048192/samplepages/9780134048192.pdf>.

RAFAEL NAVARRO, Marset. ELP-DSIC-UPV. Modelado, Diseño e Implementación de Servicios Web [en línea]. [Citado 29 de Enero, 2017]. Disponible en:
<http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>.