

## Sistema distribuido aplicado a un control de inventario

CASTAÑEDA-PIREZ, Melina\*†, QUIÑONEZ-GARCÍA, Silverio, CUEVAS-VALENCIA, René Edmundo y VIVAS-VILLASANA, Eloisa Mercedes

*Facultad de Ingeniería. Universidad Americana de Acapulco. Facultad de Ingeniería. Universidad Autónoma de Guerrero*

Recibido Octubre 12, 2017; Aceptado Diciembre 4, 2017

### Resumen

En este trabajo se presenta el resultado de un proyecto llevado a cabo en la Facultad de Ingeniería de la Universidad Americana de Acapulco, Guerrero, México, el cual tiene como finalidad mejorar los procesos administrativos realizados en un sistema de control de inventario, que se encuentra conformado por una aplicación Web realizada con la tecnología .Net y una aplicación Móvil creada para dispositivos con Sistema Operativo Android. A través de un código QR impreso en los activos de una empresa (el cual también es generado por este sistema), se puede consultar, actualizar y eliminar la información almacenada en una Base de Datos manejada por SQL Server. Esta información puede ser dada de alta desde la aplicación móvil o la aplicación Web, permitiendo obtener información al instante en donde quiera que el usuario se encuentre. El sistema genera reportes en PDF. Con este proyecto se pretende mejorar los procesos realizados en un control de inventario, omitiendo una gran inversión en aparatos costosos y poder realizar operaciones básicas a través de un dispositivo móvil que, en la actualidad, es accesible para cualquier persona.

### Aplicación Móvil, Sistema Distribuido, QR

**Citación:** CASTAÑEDA-PIREZ, Melina, QUIÑONEZ-GARCÍA, Silverio, CUEVAS-VALENCIA, René Edmundo y VIVAS-VILLASANA, Eloisa Mercedes. Sistema distribuido aplicado a un control de inventario. Revista de Tecnología Informática 2017, 1-3: 16-25

### Abstract

This paper presents the results of a project conducted at the Faculty of Engineering of the Universidad Americana de Acapulco, Guerrero, Mexico, which purpose is to improve the administrative processes carried out in an inventory control system, which is made up of a Web application made using technology .Net and a Mobile application created for devices with Android Operating System. Through a QR code printed on the assets of a company (which is also generated by this system), you can read, update and delete information stored in a database managed by SQL Server. This information can be registered from the mobile application or the Web application, allowing to obtain information instantly wherever the user is. The system generates reports in PDF. This project aims to improve the processes carried out in an inventory control, omitting a large investment in expensive equipment and to be able to perform basic operations through a mobile device that, at present, is accessible to anyone.

### Mobile Application, Distributed System, QR

\* Correspondencia al Autor (Correo Electrónico: reneecuevas@uagro.mx)

† Investigador contribuyendo como primer autor.

**Introducción**

El principal objetivo de la ingeniería es resolver problemas y los sistemas de información han resuelto una innumerable cantidad de estos inmediatamente después del surgimiento de la primera computadora. Durante la estancia de prácticas profesionales dentro de una empresa de energía eléctrica, ellos presentaron el problema de mejorar su sistema de control de inventario; era importante resolver esto ya que un error en esta actividad podría ocasionar altas pérdidas económicas, aunado a que no tenían organizada su información de manera que pudieran acceder rápidamente a consultar el estado de cualquier activo. Esto se resolverá utilizando plataformas Web y Móviles. Con la aplicación Web se desea que la interfaz sea rápida, amigable e intuitiva, asegurándonos de esto a través de métricas de interacción humano – computadora; Con la aplicación móvil el resultado es que el usuario, a través del escaneo de códigos QR generados para sus activos, obtenga información de sus equipos de manera rápida, así como poder actualizar el registro o dar de alta estos. Utilizando el dispositivo móvil, se aprovecha la practicidad, portabilidad y recursos de hardware que nos ofrecen con la lectura de códigos de identificación, a diferencia de otros medios digitales disponibles (Rico Moreno, Arroyo Almaguer, Rodríguez Vargas, & Nito Frías, 2016).

La metodología seleccionada para desarrollar este sistema es conocida como “Sistemas Distribuidos”, debido a que cada una de las etapas se adecuaba a la forma de trabajo en la empresa donde se implementó este sistema, también para ocupar una metodología diferente a las que tradicionalmente se utiliza. A lo largo del artículo se desarrollan cada una de las etapas de ésta, para entregar como producto final un sistema compuesto por una aplicación Web y una aplicación Móvil.

**Metodología de Sistemas Distribuidos**

Un sistema distribuido consiste en un conjunto de computadoras autónomas conectadas por una red y con soporte de software distribuido. Permite que las computadoras coordinen sus actividades y compartan recursos de hardware, software y datos, de manera tal que el usuario percibe una única facilidad de cómputo integrada, aunque ésta pueda estar implementada por varias máquinas en distintas ubicaciones (Pesado, y otros, 2016).

Debido a esta definición, se precisó en ocupar una metodología que logre incorporar esta definición en sus fases y la mejor decisión fue ocupar la propia del sistema.

Las fases fundamentales para el desarrollo de un sistema distribuido (De Dios Gómez, Hernández Camacho, Martínez Avíles, & Hernández Granados, 2009) son:

1. Especificación del Alto Nivel,
2. Estándares,
3. Herramientas de Diseño,
4. Documentación.

Esta metodología ocupa un caso particular de la Arquitectura Cliente – Servidor, que es una tendencia que actualmente está surgiendo en los Sistemas Web.

**Especificaciones de Alto Nivel**

Esta parte trata lo relacionado a las especificaciones de aplicaciones distribuidas, que comúnmente tienen un gran número de requerimientos de desempeño que los hace complejo de explicar.

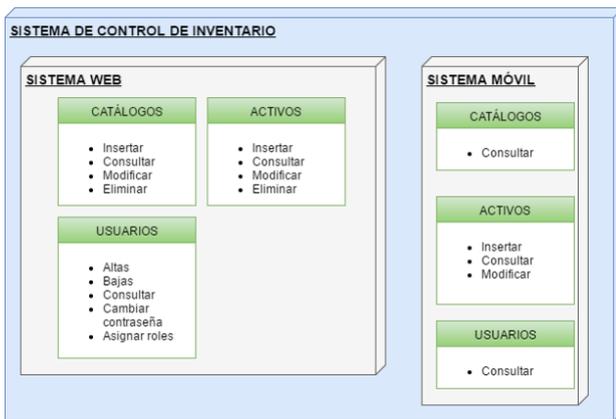


Figura 1 Especificaciones del sistema

Fuente: Elaboración propia.

En la Figura 1 se describe las operaciones que se realizarán en cada una de las aplicaciones que conformarán este sistema, específicamente el módulo móvil ya que sólo operaciones esenciales se podrán hacer a través de éste para no saturar de procesos el dispositivo. Este diagrama fue el resultado de un levantamiento de requerimientos acerca del usuario final de este sistema. El Sistema Web será el elemento activo en el proyecto, debido a que se pueden realizar todas las operaciones disponibles. El elemento pasivo será el Sistema Móvil ya que solamente ciertas operaciones podrán ser realizadas a través de la aplicación.

Resulta relativamente fácil describir un sistema distribuido dando una explicación detallada de su implementación. Debido a esto, en la Figura 2 se explica el funcionamiento de este sistema.

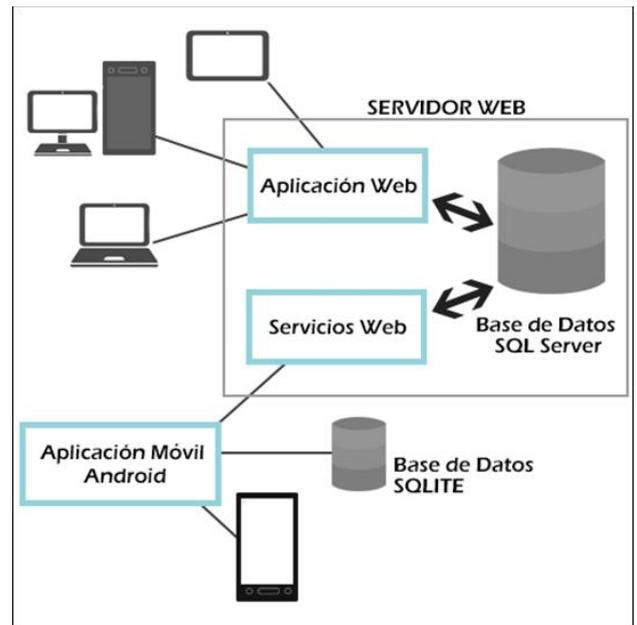


Figura 2 Implementación del sistema

Fuente: Elaboración propia.

La base de datos creada en SQL Server contiene la información almacenada por el usuario. Dentro del dispositivo móvil existe un duplicado de las tablas que funcionarán como catálogos para poder agregar información a las tablas padres o entidades fuertes; mismas que mantendrán la organización en la información y se guardarán en el dispositivo sin ocupar gran cantidad de espacio de almacenamiento, se utilizará el servicio de SQLite, siendo esto la capa conocida como datos o almacenamiento que corresponde al primer nivel del sistema.

Para poder acceder a sus registros, se usarán Servicios Web, que serán los encargados de suplir las peticiones enviadas por la aplicación Web y Móvil. Al utilizar este tipo de arquitectura se aplica seguridad a los datos puesto que no existe una comunicación directa de las aplicaciones con la base de datos de SQL Server; por lo tanto, el servicio Web figura como un intermediario para gestionar las peticiones. Esta capa corresponde al procesamiento de información o de negocios correspondiente al segundo nivel de la arquitectura.

Continuando con la arquitectura, en su tercer nivel está la capa de presentación o de interfaz de usuario. Esta se implementa bajo la aplicación Web y móvil, ya que presentará la información obtenida del Servicio Web al usuario.

### Estándares

Los estándares son importantes dentro de un sistema distribuido, aunado a que este proyecto fue realizado en trabajo en equipo, a fin de poner garantizar la uniformidad en todo el sistema.

Para determinar el procedimiento de desarrollo de código, se opta por utilizar estándares de codificación pertenecientes a los lenguajes C# para la aplicación Web y JAVA para la aplicación Móvil.

Partiendo de la existencia de tres tipos de nomenclatura: Camel Case y Pascal Case (Gordillo Arranz, 2015), se especificó cuales se utilizarían en los nombres de los archivos, funciones, métodos y variables entre otros elementos.

En el ámbito del diseño responsivo, se utilizó el patrón Mostly Fluid (Azaustre, 2015), ya que la interfaz la acomoda como una grilla y de este modo, puede acomodar en base a columnas la información sin tener pérdida de contenido.

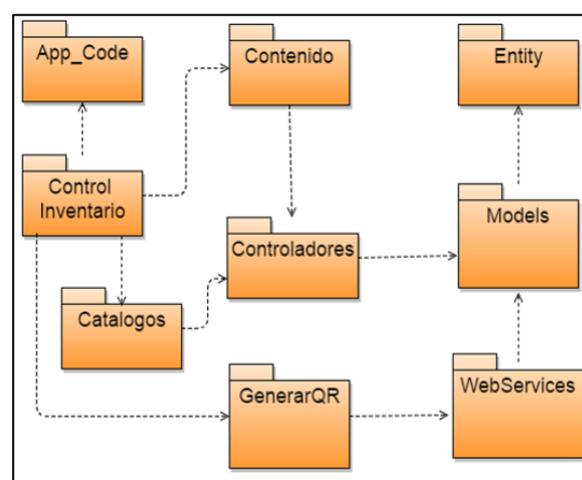
Se utilizaron iconos para la aplicación en su mayoría de licencia gratuita y los demás de elaboración propia. También se propuso una paleta de colores avalada por la institución en donde está implementado el sistema.

### Herramientas de Diseño

Los principios de diseño deben aumentar la productividad, mejorar la reusabilidad del software y facilitar la generación automática de software, a partir de sentencias declarativas de requerimientos. El uso de los principios de diseño reduce también, el tramo que separa la interconexión de sistemas separados.

Existen programas que simplifican el proceso de diseño de aplicaciones distribuidas como CORBA, RMI o DCOM (Quintana Rodríguez, 2013). Sin embargo, utilizando diagramas UML y algunas herramientas proporcionadas por el IDE Visual Studio se podrá diseñar de manera óptima los procesos realizados por el Sistema de Control de Inventario.

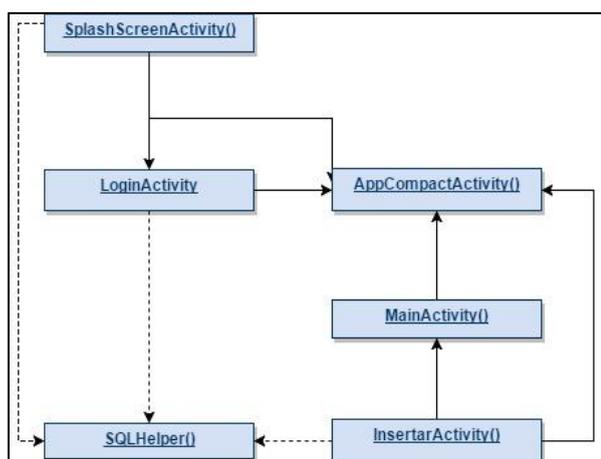
En la Figura 3 se modela la estructura interna de la aplicación Web. Dentro de estas carpetas se tienen todas las clases necesarias para el funcionamiento del sistema. De igual manera se pueden ver las relaciones entre estas.



**Figura 3** Estructura de la aplicación Web

Fuente: *Elaboración propia.*

De igual manera, se esquematiza el diagrama de clases de la aplicación Móvil. En este se pueden ver todos los métodos a utilizar para el correcto funcionamiento, incluyendo las clases para tareas asíncronas que se encuentran dentro de *SplashScreenActivity*.

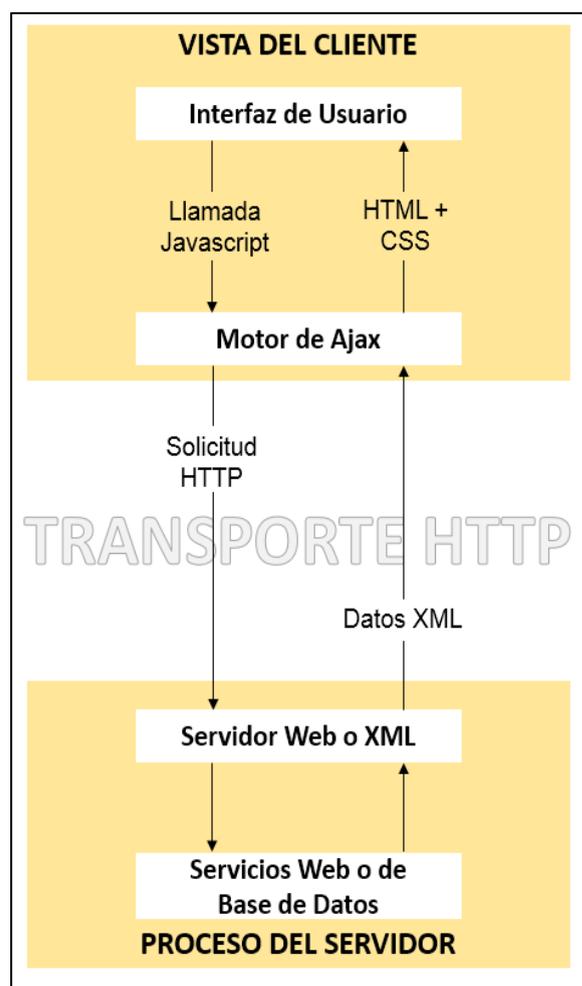


**Figura 4** Diagrama de clases aplicación Móvil

Fuente: *Elaboración propia.*

Entre las tecnologías a destacar en este sistema se encuentra AJAX, la cual permite que las páginas Web se actualicen asincrónicamente intercambiando datos con un servidor Web entre bastidores. Esto significa que es posible actualizar partes de una página web, sin recargar la página entera (W3Schools, 2017). En la Figura 5 se esquematiza el modo de funcionamiento de esta tecnología.

En primer lugar, se realiza una llamada desde el lado del cliente utilizando el lenguaje Javascript. En este proceso se prepara toda la información para que el motor de Javascript envíe la solicitud al servidor. Utilizando un tipo de transporte Web, en este caso HTTP (HyperText Transfer Protocol), se comunica la petición al servidor. En este punto comienzan los procesos del lado del servidor donde un Servicio Web, sistema de base de datos o cualquier otro tipo de procedimiento procesará la solicitud y enviará la respuesta pertinente en formato XML. Dentro del XML la información puede ir estructurada como un diccionario de datos o como una cadena JSON. El mismo motor que envía la solicitud es el que espera la respuesta de ésta, enviándola al proceso del lado del cliente utilizando los lenguajes HTML y CSS.



**Figura 5** Modelo de comunicación AJAX

Fuente: *Elaboración propia.*

En la siguiente Figura se explica la tecnología KSOAP. KSOAP es una biblioteca de peso ligero para uso en dispositivos limitados. No pretende ser un do-it-all biblioteca abarca todo. Sin embargo, está destinado a interactuar con la mayoría de los motores de SOAP populares (KObjects, 2014).

Para este caso, las llamadas de datos se realizan a través de clases en Java, enviadas a través del transporte Ksoap utilizando el protocolo HTTP. En el servidor, se procesa la solicitud correspondiente dando como respuesta la información en un archivo XML. El mismo transporte que envía la solicitud espera la respuesta del servicio Web y el resultado lo imprime en pantalla como una cadena de resultados.

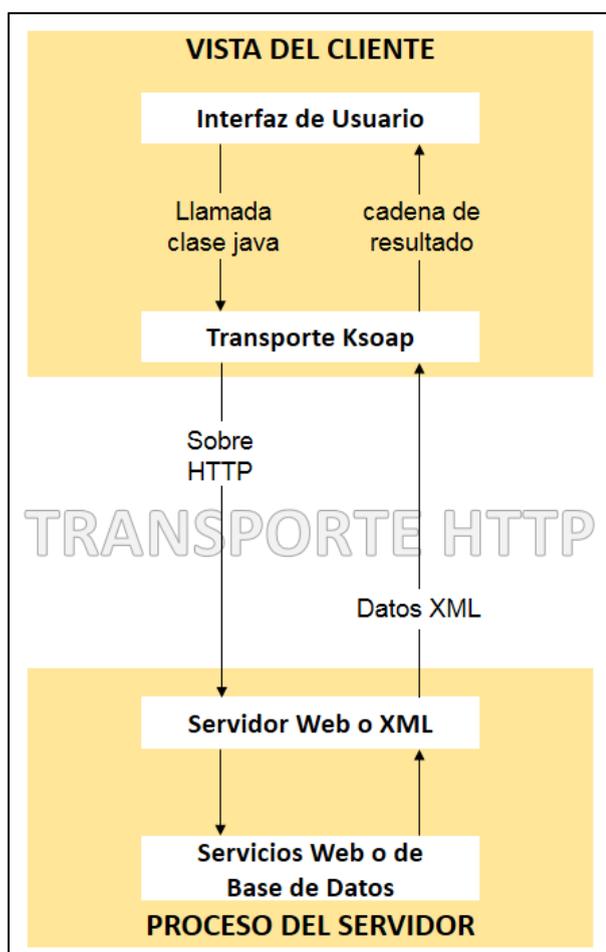


Figura 6 Modelo de comunicación KSOAP

Fuente: *Elaboración propia.*

## Documentación

El ciclo de vida del desarrollo del software contempla a la documentación como una etapa de gran importancia para el producto final. Resulta crucial guardar las especificaciones establecidas que fundamentan el funcionamiento del software, así como de los componentes a partir de los que se forma la aplicación final (Rivera Landa, 2016).

Esta etapa se ve reflejada en el sistema en el módulo de Ayuda, donde el usuario puede encontrar preguntas frecuentes y tutoriales que puede seguir para utilizar de manera correcta el sistema, así como los pasos que necesita realizar para instalar la aplicación Móvil en su dispositivo.

## Resultados

El sistema se terminó e implementó en una red interna en febrero 2017. Se obtuvieron 2 aplicaciones: La aplicación Web la cual está desarrollada en el framework .NET y es totalmente responsiva. En la Figura 7 se encuentra la página principal del sistema.



Figura 7 Página principal de la aplicación Web

Fuente: *Elaboración propia*

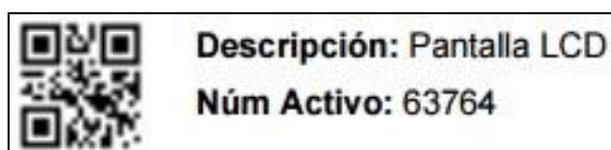
En la Figura 8 se encuentra el formulario de registro de activos a través de la plataforma Web. Todos los datos son los requeridos por el centro de trabajo donde fue implementada la aplicación.

El formulario 'Agregar Registro' está diseñado para capturar información detallada de un activo. Incluye campos de texto para 'Descripción', 'Fecha de Ingreso', 'Fecha de Recepción', 'Fecha de Garantía' y 'Fecha de Capitalización'. También cuenta con menús desplegables para 'Marca', 'Modelo', 'Tipo Activo', 'Centro de Trabajo', 'Ubicación', 'Grupo' y 'Estado de Registro'. Sección de radio buttons para 'Activo Oficial' (Sí/No) y 'Equipo en Uso' (Sí/No). Un campo de texto para 'Código de Barras' y un menú para 'Asignado a'. Botones 'Aceptar' y 'Cancelar' están ubicados en la parte inferior derecha.

Figura 8 Registro de Activo

Fuente: *Elaboración propia*

Desde la aplicación Web se generan los códigos QR que son colocados en los equipos, el Figura 9 se puede ver la plantilla de estas etiquetas.



**Figura 9** Etiqueta de código QR

*Fuente: Elaboración propia.*

La Experiencia de Usuario, o UX como se abrevia comúnmente, se refiere a los todos los aspectos en la relación de una persona con un producto, aplicación o sistema (Angulo, 2015). Para garantizar la usabilidad de este sistema se realizaron pruebas basándose en las métricas de interacción humano computadora. Se realizaron pruebas con los usuarios para calificar rubros como la sencillez de llegar a realizar una función, la cantidad de elementos que se presentan en una página y si la gama de colores parecía adecuada.

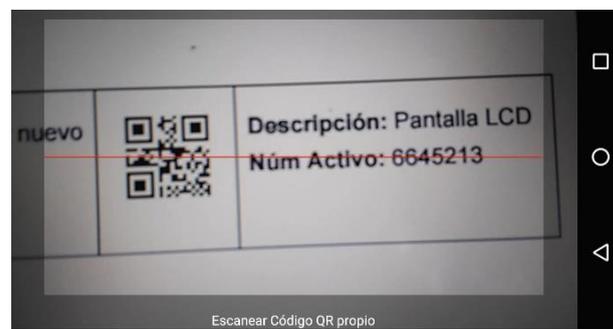


**Figura 10** Pantalla principal de la aplicación Web

*Fuente: Elaboración propia.*

En la Figura 10 se encuentra la pantalla principal de la aplicación Móvil después de que el usuario haya proporcionado las credenciales pertinentes. En esta se encuentra el menú con las distintas operaciones que se pueden realizar. Para acceder al escaneo de códigos QR se selecciona en el cuarto icono, representado por una lupa.

Activando la cámara como escáner, se enfoca hacia la etiqueta que previamente se generó desde el módulo Web y al identificar el código lo envía al servidor y éste responde una cadena con la información del equipo que ha sido escaneado. La función de la cámara como escáner se muestra en la Figura 11.

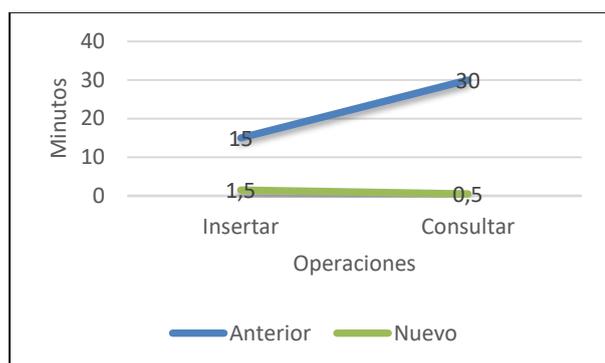


**Figura 11** Escaneo de código QR

*Fuente: Elaboración propia.*

## Conclusiones

El implementar el sistema distribuido para el control de inventario de una empresa agilizó los tiempos al realizar el proceso de alta y consulta de información de activos gracias al módulo Móvil, ya que fue reemplazado un sistema anterior que se encontraba obsoleto, así como algunas prácticas llevadas en papel.



**Gráfico 1** Resultados de pruebas de tiempo

*Fuente: Elaboración propia.*

En el gráfico 1 se puede ver los tiempos de carga en las operaciones anteriormente mencionadas. Cabe señalar que las operaciones incluyen el tiempo, en el caso del antes, de anotar en un papel las características del equipo para después insertarlo o consultarlo, cuando ahora se puede realizar desde el dispositivo móvil tratándose de un tiempo de pocos minutos.

La interfaz de usuario ha sido adecuada para el usuario ya que se ofrece una mayor experiencia a través de componentes intuitivos y una gama de colores apropiada.

Un gran beneficio que aporta este sistema es que no se necesita de herramientas externas como un lector de código de barras especializado para llevar un mejor control, si no se ocupa el propio dispositivo móvil evitando así costos extras que pueden resultar muy altos, así como ahorros en mantenimiento de dichas herramientas externas.

## Anexos

En la Tabla 1 se describe brevemente el contenido de las carpetas mostradas en la Figura 3.

Carpeta	Descripción
<b>Control Inventario</b>	Es la clase principal de todo el proyecto. En esta se encuentra el Site Master de la aplicación y configuraciones generales.
<b>App_Data</b>	En esta se encuentran las configuraciones iniciales de los encabezados del sitio.
<b>Contenido</b>	Contiene los accesos a todas las páginas principales del sitio.
<b>Catálogos</b>	Contiene todos los catálogos del proyecto.
<b>Controladores</b>	Encargados de responder las solicitudes de los usuarios a través del HTTP. Usualmente se utiliza para acceder a la información de los catálogos y tablas de la base de datos.
<b>WebServices</b>	Se utilizan para realizar las operaciones CRUD (Create, Read, Update y Delete) del proyecto Web y Móvil.
<b>GenerarQR</b>	Contiene las clases necesarias para generar QR a partir de un registro en la Base de Datos.
<b>Models</b>	Se encuentran los modelos necesarios para la autenticación. Para este proyecto se ocupa la autenticación Identity.
<b>Entity</b>	Se encuentran los modelos de Entity Frameworks que representan las tablas de la Base de Datos como clases.

**Tabla 1** Descripción de carpetas de la estructura de la aplicación Web

*Fuente: Elaboración propia.*

En la Tabla 2 se encuentra la descripción de las clases mencionadas en la Figura 4.

Nombre	Descripción
SplashScreenActivity	Es la actividad que se muestra a los usuarios como pantalla de presentación a la aplicación cada vez que se ejecuta.
LoginActivity	Actividad que gestiona el inicio de sesión de los usuarios.
MainActivity	Actividad principal donde se muestran todas las operaciones que se pueden realizar.
InsertarActivity	Actividad que muestra el formulario para poder insertar datos al control de inventario.
AppCompatActivity	Clase base para las actividades que utilizan las bibliotecas de soporte para características de la barra de acción.
SQLHelper	Clase que contiene los métodos para realizar operaciones en la Base de Datos interna.
SQLiteOpenHelper	Método que se ejecuta cada vez que se hace una llamada al SQLHelper.
View.OnClickListener	Método que gestiona todos los eventos al dar click en los elementos de la pantalla.

**Tabla 2** Descripción de las clases de la aplicación Móvil

Fuente: *Elaboración propia*

## Referencias

Angulo, M. (4 de Junio de 2015). *Diseño de métricas de usabilidad*. Obtenido de Revista Digital UX Nights: <http://revista.uxnights.com/sobre-metricas-de-usabilidad/>

Azaustre, C. (04 de Noviembre de 2015). <https://i.ytimg.com/vi/i3MUY5qtANo/maxresdefault.jpg>. Obtenido de Carlos Azaustre: <https://carlosazaustre.es/blog/los-5-patrones-del-responsive-design/>

De Dios Gómez, S., Hernández Camacho, E., Martínez Aviles, A., & Hernández Granados, J. J. (Julio de 2009). *Unidad 4: Metodología para el desarrollo de proyectos en Sistemas Distribuidos*. Obtenido de Sistemas Distribuidos: <http://sdequipo2.blogspot.mx/2009/07/unidad-4-metodologia-para-el-desarrollo.html>

Gordillo Arranz, P. (2015). *Desarrollo de frameworks con .NET*. Madrid: Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (Universidad Politécnica de Madrid).

KObjects. (18 de Septiembre de 2014). *kSOAP 2*. Obtenido de kObjects: <http://kobjects.org/ksoap2/index.html>

Pesado, P. M., Bertone, R. A., Thomas, P. J., Marrero, L., Pasini, A., Delía, L. N., . . . Pi Puig, M. (2016). Experiencias en el desarrollo de Sistemas de Software Distribuidos. *XVIII Workshop de Investigadores en Ciencias de la Computación*, 576-580.

Quintana Rodríguez, J. (2013). *Sistemas distribuidos con componentes*. Obtenido de Universidad Veracruzana: <https://www.uv.mx/iiesca/files/2013/01/distribuidos1999-2000.pdf>

Rico Moreno, J., Arroyo Almaguer, M., Rodríguez Vargas, M., & Nito Frías, F. (2016). App prototipo gestor de identificación de alumnos a través de códigos QR. *Revista de Sistemas Computacionales y TIC's*, 9 - 14.

Rivera Landa, A. (2016). Importancia de la Arquitectura de Software. En A. Rivera Landa, *Importancia de la Documentación de la Arquitectura de Software* (pág. 22). Ciudad de México: Escuela de Graduados en Ingeniería y Arquitectura.

W3Schools. (2017). *AJAX*. Obtenido de W3Schools:

[https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)