



DESARROLLO DEL SISTEMA DE REQUISICIONES PARA LA EMPRESA HIDROELECTRICA ABANICO S.A. APLICANDO EL ENTORNO DE PROGRAMACION NODE.JS

Autores: ¹Diana Elizabeth Gómez García.

danalizgg713@gmail.com

² Ángel Patricio Flores Orozco.

aflores@esPOCH.edu.ec

³ Ureta Valdez Rogelio Estalin.

royel_02@hotmail.com

Docentes de la Escuela Superior Politécnica de Chimborazo Ext. Morona Santiago.

Para citar este artículo puede utilizar el siguiente formato:

Diana Elizabeth Gómez García, Ángel Patricio Flores Orozco y Ureta Valdez Rogelio Estalin (2018): "Desarrollo del sistema de requisiciones para la empresa hidroeléctrica Abanico S.A. aplicando el entorno de programación Node.JS", Revista Caribeña de Ciencias Sociales (mayo 2018). En línea:
[//www.eumed.net/rev/caribe/2018/05/empresa-hidroelectrica-abanicosa.html](http://www.eumed.net/rev/caribe/2018/05/empresa-hidroelectrica-abanicosa.html)

RESUMEN

La presente investigación tuvo como finalidad el desarrollo de un sistema de requisiciones para la empresa Hidroeléctrica Abanico. S.A. aplicando el entorno de programación NodeJS. El desarrollo del sistema se lo hizo a través de Microsoft Solutions Framework, esta metodología fue útil para la gestión de riesgos, ayuda con el soporte y mantenimiento, además se enfocó en la gestión y administración del proyecto para lograr el impacto deseado. El sistema se diseñó para automatizar el proceso de pedidos de requisición a través de una aplicación web que se encuentre activa en un browser en las 3 principales instalaciones de la empresa (oficina, casa de máquinas, represa). En el desarrollo se integró Django para el módulo de administración, este framework ayudó en el desarrollo rápido y pragmático. El servidor de notificaciones fue realizado en NodeJS, utilizando websockets, permitiendo la comunicación bidireccional en tiempo real entre cliente y el servidor en un mismo socket, utilizando una conexión TCP. La medición de resultados se lo hizo por medio del sistema de escalas de usabilidad (SUS) lo que permitió medir la usabilidad por medio de encuestas a los usuarios obteniendo una aceptación satisfactoria del sistema. De esta forma se pudo concluir que la utilización del sistema ayuda en el seguimiento y control de requisiciones. Al ser una aplicación escalable, se recomienda la implementación del módulo de proveedores para complementar el pedido de la requisición.

¹ Ingeniero en Sistemas Informáticos.

² Ingeniero en Sistemas Informáticos, Magister en Informática Educativa. Docente de la ESPOCH Ext. Morona Santiago

³ Ingeniero en Industrias Pecuarias, Magister en Gestión de la Producción. Docente de la ESPOCH Ext. Morona Santiago

ABSTRACT & KEYWORDS

The purpose of the present investigation was to develop a requisition system for the hydroelectric company Abanico S.A. flushing the NodeJS programming environment. The development of the system was done through Microsoft Solutions Framework, this methodology was useful for risk management, help with support and maintenance, also focused on the management and administration of the project to achieve the desired impact. The system was designed to automate the requisition request process through a web application that is active in a browser in the three main facilities of the company (office, powerhouse, and dam). In the development Django was integrated for the administration module, this framework helped in the rapid and pragmatic development. The notification server was realized in NodeJS, using websockets, allowing bidirectional communication in real time between the client and the server in the same socket, using a TCP connection. The measurement of the results was done through the system of scales for usability (SUS), which allowed to measure the usability by means of user surveys, obtaining a satisfactory acceptance of the system. In this way it was possible to conclude that the use of the system helps in the follow-up and control of requisitions. Being a scalable application, it is recommended to implement the suppliers module to complement the requisition request.

Palabras claves:

tecnología y ciencias de la ingeniería-ingeniería de software-programación de aplicación web-optimización de procesos industriales-nodejs (framework)-javascript (software)-django (framework)-requisiciones.

Key words:

technology and engineering sciences, software engineering, web application programming, industrial process optimization, nodejs (framework), javascript (software), django (framework), requirements

1. INTRODUCCIÓN:

La tecnología está cambiando al mundo que habitamos, excluyendo barreras de tiempo y distancia, permitiendo compartir información. En la actualidad permite generar herramientas de apoyo en la solución de problemas o necesidades que requiera el ser humano. El internet junto con la tecnología se ha convertido en el gran aliado de los usuarios. El internet se insertó en la sociedad cambiando el concepto de comunicaciones, permitiendo estar conectados desde cualquier lugar del planeta; basándose en un computador, un browser y conexión a internet. Los usuarios pueden utilizar aplicaciones web subidas a la red, para solucionar alguna necesidad o problema existente.

La investigación aquí desarrollada tiene como temática el desarrollo de un sistema de requisiciones para la empresa hidroeléctrica abanico S.A. aplicando el entorno de programación Node.JS. Al ser identificado el problema se pudo levantar información técnica, con esto se permitió diseñar una aplicación web en tiempo real. La requisición es el pedido de compra de algún bien o servicio de un departamento para la liberación de recursos y la adquisición del mismo. El entorno NodeJS permitirá la comunicación en tiempo real a través del lenguaje Javascript para la creación de notificaciones entre el personal administrativo y operativo. Las aplicaciones en tiempo real son aplicaciones bidireccionales, lo cual quiere decir que a diferencia de tecnologías como Ajax, que únicamente envían datos del cliente al servidor y reciben una respuesta, la tecnología bidireccional puede enviar datos del cliente al servidor y de igual manera del servidor al cliente, mostrando la información en vivo que se encuentre en el servidor; todo esto es posible con node.js utilizando Socket.IO. Se integrará el framework

Django a través del lenguaje Python, para la administración de la aplicación, este framework también comprende las interacciones cliente-servidor en los sitios web y la programación del lado del servidor. La aplicación se la realizará en base a la metodología MSF por sus siglas en inglés *Microsoft Solutions Framework* cuyas fases comprenden visión y alcance, planificación, desarrollo, estabilización, instalación y soporte. La aplicación tendrá como principal funcionalidad la comunicación en tiempo real a través de notificaciones entre el operador y secretaria mediante una interfaz web. Dicha aplicación ayudará con la creación de notificaciones de un pedido de requisición entre el personal operativo (operador) y el personal administrativo (secretaria). Al tratarse de una aplicación web orientada al uso continuo por parte de los usuarios, el análisis de resultados se obtendrá en base al test SUS por sus siglas en inglés *System Usability Scale*, que evalúa la usabilidad del producto final.

2. METODOLOGÍA:

Esta aplicación web está basado en la metodología MSF (*Microsoft Solutions Framework*), al ser una metodología tradicional, flexible puede adaptarse a diferentes proyectos sin importar las elecciones de herramientas tecnológicas, por ende, se aborda las fases de desarrollo de MSF, tales como visión y alcance donde está dada por la visión clara y conjunta del equipo para lograr satisfacer al cliente y motivar al equipo, a continuación la fase planificación donde se define y especifica requerimientos, posteriormente la fase de desarrollo donde se construye los componentes de la solución, finalmente la fase estabilización donde culmina con una revisión del sistema y para concluir con la fase Instalación realizando manuales de guía para el usuario.

2.1. Visión y Alcance

2.1.1. Definición del Problema

Una requisición de compra (documento) es originada por un empleado en la empresa (planta 1 o planta 2) y enviada a través del transporte hacia la oficina principal ubicada en la ciudad de Macas. Cuando el documento se encuentra en la oficina recibe la autorización por parte del gerente. Una vez autorizada, la secretaria procede al despacho hacia el lugar donde se originó la requisición con el fin de abastecer bienes o servicios requeridos por los trabajadores. Cabe recalcar que el proceso de notificar las requisiciones toma de 1 a 2 días, tampoco existe ninguna revisión de lo solicitado antes de la autorización por lo cual consume tiempo y recursos ocasionando deficiencia en el control de las requisiciones realizadas.

2.1.2. Visión del proyecto

El principal objetivo es entregar una aplicación web con el propósito de notificar de manera inmediata el pedido de requisición, de esta manera se podría incrementar la comunicación entre el personal administrativo y operativo.

2.1.3. Perfiles de Usuario

Administrador. - Persona que posea un perfil de: Ingeniero en sistema con conocimientos básicos de redes y conocimiento en lenguaje de programación Javascript y Python. Funciones: Gestionar permisos, usuarios, categorías, productos, requisiciones, notificaciones, reportes.

Administrativo. - Perfil de usuario Profesional con permiso total en la realización de las requisiciones ejemplo: secretaria, Gerente de procesos. Tareas: Gestionar inventario, requisiciones, visualizar e Imprimir reportes.

Operador. - Perfil de usuario: Empleado en las centrales con conocimientos básicos en la utilización de equipos de cómputo. Tareas: Gestionar requisiciones.

2.1.4. **Ámbito del Proyecto**

El sistema web que se desarrolla para la empresa Hidroabánico S.A. denominado sistema de requisiciones, el cual ayuda en el control y seguimiento por parte de los administrativos, para el proceso de compra de bienes.

El Sistema de requisiciones brinda un portal de comunicación entre la planta central y la oficina, donde los trabajadores registren las necesidades de compra para su posterior revisión por parte de los administrativos.

2.1.5. **Herramientas a utilizar**

En la tabla N° 1 se detalla las herramientas a utilizar en el desarrollo del sistema web de requisiciones.

Tabla 1: Herramientas.

| HERRAMIENTA | DESCRIPCIÓN |
|---------------------------------|--|
| <i>Node.JS</i> | Framework para desarrollo de aplicaciones web en tiempo real |
| <i>Java Script & Python</i> | Lenguaje de programación |
| <i>Django</i> | Framework de desarrollo |
| <i>PostgreSQL</i> | Base de datos |
| <i>Windows</i> | Sistema operativo |
| <i>Ubuntu Server</i> | Servidor |
| <i>MSF</i> | Metodología de desarrollo de software. |
| <i>Microsoft Project</i> | Calendarización, cálculo estimado del tiempo |
| <i>StartUML</i> | Lenguaje gráfico para modelamiento de software |
| <i>ReportLab</i> | Constructor y diseñador de reportes |

Elaborado por: Autores.

2.2. **Planificación**

2.2.1. **Interfaces de Hardware**

En la tabla N° 2 y N° 3 se indica las características del servidor de aplicaciones para el despliegue del sistema y las características de la computadora cliente para el uso del sistema.

Tabla 2: Requisitos Servidor

| CARACTERÍSTICAS | SISTEMA OPERATIVO | MEMORIA RAM | DISCO DURO |
|------------------------|--------------------------|--------------------|-------------------|
| <i>Detalle</i> | Ubuntu | 512 MB | 20 GB |

Elaborado por: Autores.

Tabla 3: Requisitos Cliente

| DETALLE CARACTERÍSTICAS | PROCESADOR | RAM | DISCO DURO |
|------------------------------------|-------------------|------------|-----------------------|
| <i>Computador</i> | 1.8 GHZ | 1 GB | 500 GB |
| <i>Acceso Internet</i> | 512 Kbps | | |

Elaborado por: Autores.

2.2.2. Interfaces de Software

La Tabla N° 4 indica la instalación de la plataforma de software necesaria.

Tabla 4: Interfaces de Software

| TIPO | SOFTWARE |
|-----------------------------------|--|
| <i>Sistema Operativo Servidor</i> | Ubuntu |
| <i>DBMS</i> | PostgreSQL |
| <i>Framework JavaScript</i> | NodeJS |
| <i>Framework Python</i> | Django |
| <i>Plataforma de aplicaciones</i> | Mozilla Firefox / Google Chrome / Internet Explorer. |

Elaborado por: Autores.

2.2.3. Interfaces de Comunicación

Para el funcionamiento del sistema se recomienda el uso de nginx, que es un servidor ligero de alto rendimiento, de software libre y código abierto; el cual permite el despliegue de la aplicación web.

2.2.4. Requerimientos Funcionales del Sistema

Los requerimientos funcionales del sistema de requisiciones se los ha organizado en cinco módulos, donde se detalla entradas, procesos y salidas.

- **Gestión de Usuarios:** El módulo de gestión de usuarios es administrado bajo el perfil de superusuario o administrativo donde se creará usuario, contraseña y se asigna su perfil de usuario.
- **Gestión de Inventario:** El módulo de inventario es administrado bajo el perfil de superusuario o administrativo donde se ingresa, modifica o elimina productos o categorías.
- **Gestión de Requisiciones:** El módulo de requisiciones es administrado bajo el perfil operativo o administrativo donde se ingresa, modifica o elimina pedidos de requisición.

- Gestión de Notificaciones: El sistema debe generar notificaciones entre el personal operativo que radica en la planta y el personal administrativo que se encuentra en la oficina central.
- Reportes: El sistema permite la visualización de reportes.

2.2.4.1. Requerimiento principal

En la figura N° 1 y tabla N° 5 se muestra el caso de uso del requerimiento gestión de notificaciones.

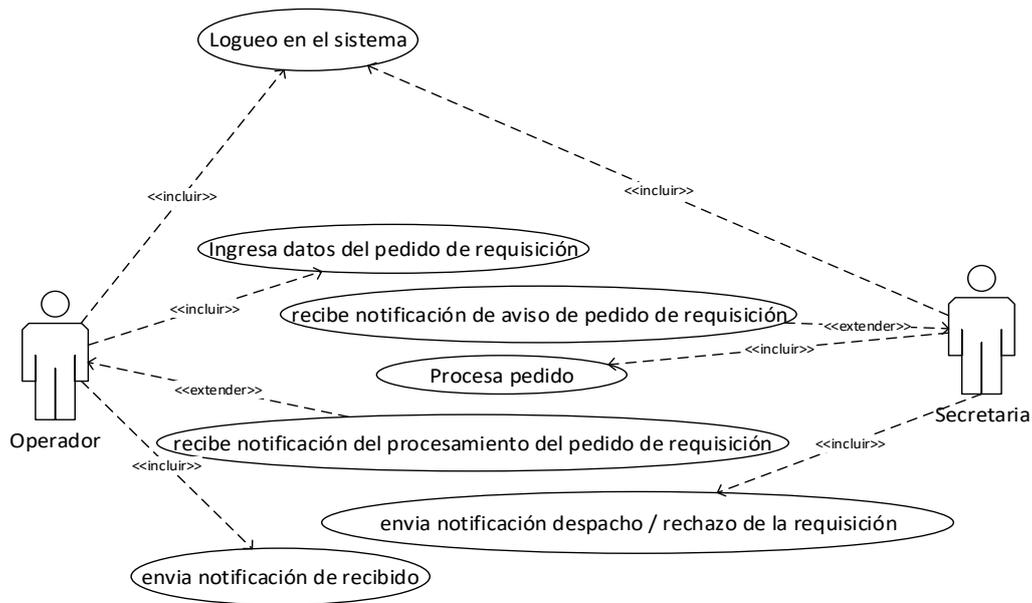


Figura 1: Caso de Uso - Requerimiento Gestión de Notificaciones

Tabla 5: Caso de uso - Gestión de Notificaciones

Número: Requerimiento 4. **Usuario:** Operador – secretaria – gerente procesos.

Nombre historia: Gestión de Notificaciones.

Responsable: Diana E. Gómez G.

Descripción: El sistema debe generar notificaciones entre el personal operativo que radica en la planta y el personal administrativo que se encuentra en la oficina central.

| ACCIONES DE ACTORES | RESPUESTA DEL SISTEMA |
|---|--|
| Ingresa al sistema | Valida perfil de usuario. |
| Operador ingresa datos del pedido de requisición | Valida, guarda información en base de datos. |
| Secretaria recibe notificación de aviso / pedido de requisición | Envía notificación a la secretaria. |
| Secretaria procesa el pedido | Valida entradas y ejecuta la acción. |
| Operador recibe notificación del | Envía notificación al operador. |

procesamiento.

Secretaria notifica despacho o rechazo de la requisición. Envía notificación al operador.

Operador notifica la recepción de la requisición. Envía notificación a la secretaria.

Observaciones: Usuario ingresa información incorrecta el sistema informa y no ejecuta la acción

Elaborado por: Autores.

2.2.5. Arquitectura para la solución

En la figura N° 2, N° 3 y N° 4 se muestra respectivamente la arquitectura interna, externa y



detallada del sistema de requisiciones.

Figura 2: Arquitectura Interna

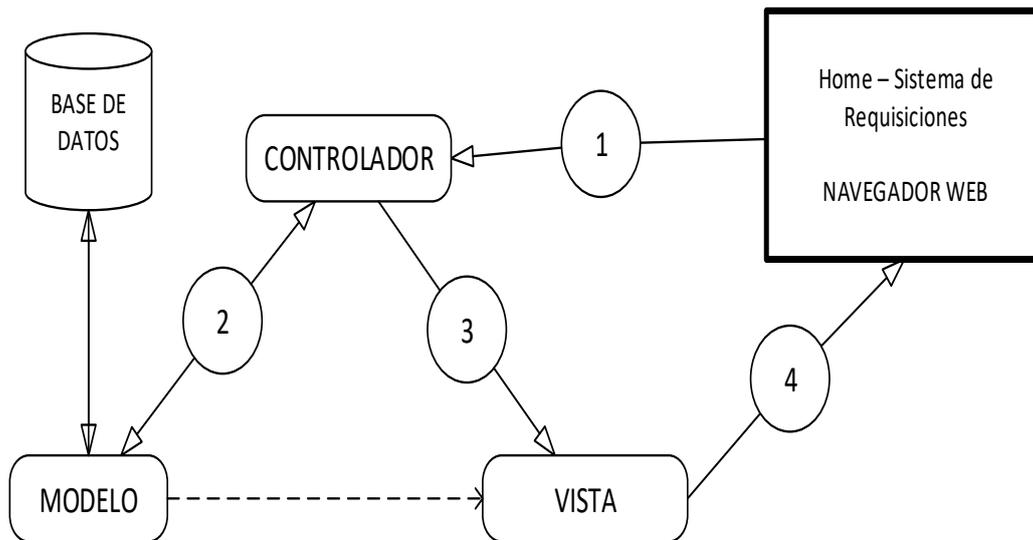


Figura 3: Arquitectura Interna

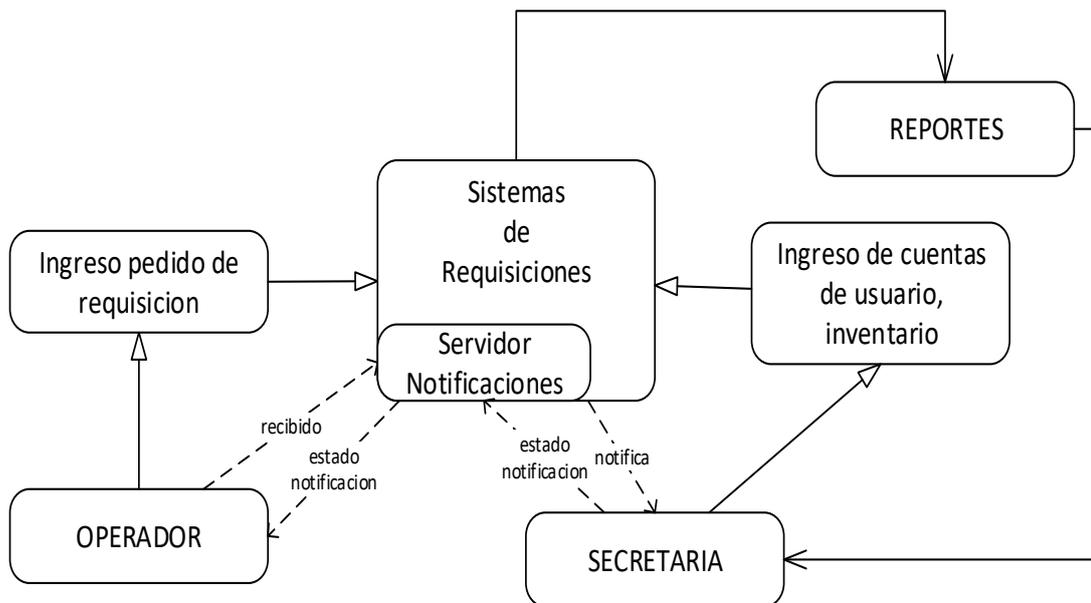


Figura 4: Arquitectura Detallada

2.3. Desarrollo

La Arquitectura es MVC; Modelo (Model) es la capa de acceso a la base de datos; Vista (Template) esta capa contiene la presentación y Controlador (Views) muestra en la plantilla la lógica que accede al modelo.

2.4. Estabilización

En la figura N° 5 se muestra el código fuente que permite la comunicación en tiempo real entre el personal administrativo y operativo cuando se realiza el pedido de requisición.

```

1  var express = require('express'),
2      app     = express();
3
4  var server = require('http').createServer(app)
5      io     = require('socket.io').listen(server);
6
7  var pg = require ('pg');
8  var pgConString = "postgres://postgres:081011@localhost:5432/requisicionesdb"
9  var client = new pg.Client(pgConString)
10 client.connect()
11 client.query("LISTEN addnotificacion");
12 client.handler = false;
13
14 var clientes = {}
15
16 io.sockets.on('connection', function (socket) {
17     if (!client.handler) {
18         client.on('notification', function(data) {
19             //socket.broadcast.to('gestionadores').emit('message', {data:data.payload});
20             var notificacion = JSON.parse(data.payload);
21             var receptor = notificacion['receptor_id'];
22             if (receptor in clientes) {
23                 clientes[receptor].socket.emit('count', notificacion)
24                 clientes[receptor].socket.emit('notificacion', notificacion)
25             }
26         });
27         client.handler=true;
28     }
29
30     socket.emit('data', {});
31
32     socket.on('data', function (data) {
33         var receptor = data['user_id']
34         this['receptor'] = receptor
35         clientes[receptor] = {
36             'socket': this,
37         }
38     });
39
40     socket.on('disconnect', function () {
41         delete clientes[this['receptor']];
42         console.log('Socket: ' + this['id'] + ' se ha desconectado.')
43     });
44 });
45
46
47 console.log('Server listen on port 8002');
48 server.listen(8002)

```

Figura 5: Código fuente – Servidor de notificaciones

3. RESULTADOS:

El marco de resultados se basa en la aplicación del estándar ISO/IEC 9126, para la evaluación del producto de origen software teniendo en cuenta las métricas de usabilidad, para esto se tomó el test SUS (*System Usability Scale*) a los usuarios directos para determinar el grado de usabilidad de la aplicación web.

3.1. Análisis Test SUS

La población para tomar el test SUS en base a la métrica de usabilidad en el sistema de requisiciones, serán los empleados de la empresa específicamente 11 personas; dispuestos por el gerente de la empresa Hidroabanico en la provincia de Morona Santiago, quienes van a utilizar la aplicación web.

Para evaluar la utilidad de la aplicación, se realizó la capacitación respecto al uso del sistema web dirigido al personal administrativo y operativo de la empresa Hidroeléctrica Abanico S.A., en la cual se puso a consideración el test SUS.

Una vez completadas las encuestas a los empleados se procede a la tabulación de las mismas. Cabe recalcar que la escala de uso de la aplicación va desde completamente de acuerdo (5) a completamente desacuerdo (1).

En la tabla N° 6 se realizó la tabulación de los datos obtenidos en las encuestas realizadas a los empleados, sobre el uso del sistema de requisiciones implementado en la empresa. Con los siguientes resultados:

Tabla 6: Tabulación datos Test SUS

| N° | PREGUNTA | N° de test | | | | | | | | | | | % |
|----|--|------------|---|---|---|---|---|---|---|---|----|----|--------|
| | | # Usuarios | | | | | | | | | | | |
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
| 1 | Creo que me gustaría utilizar este sistema frecuentemente. | 5 | 4 | 5 | 5 | 5 | 4 | 4 | 5 | 5 | 4 | 5 | 92,73 |
| 2 | El sistema me resulto innecesariamente complejo. | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 29,09 |
| 3 | Creo que el sistema es bastante fácil de utilizar. | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 100,00 |
| 4 | Creo que necesitaría el soporte de un técnico para poder utilizar este sistema. | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 23,64 |
| 5 | Creo que as diferentes funciones del sistema se encuentran muy bien integradas. | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 5 | 96,36 |
| 6 | Opino que hubo demasiada inconsistencia en el sistema. | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 23,64 |
| 7 | Imagino que la mayoría de las personas aprendería a utilizar el sistema rápidamente. | 4 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | 4 | 5 | 5 | 94,55 |
| 8 | Me sentí algo incómodo al utilizar este sistema. | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 21,82 |
| 9 | Me sentí muy seguro al utilizar este sistema. | 5 | 5 | 5 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 94,55 |
| 10 | Necesito aprender muchas otras cosas | 2 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 32,73 |

antes de poder utilizar
correctamente el
sistema.

| | | | | | | | | | | | | |
|-----------------------|----|----|-----|----|----|----|----|----|----|----|----|-------|
| TOTAL | 37 | 36 | 40 | 37 | 36 | 36 | 37 | 39 | 37 | 37 | 39 | 93,41 |
| Puntuación SUS | 93 | 90 | 100 | 93 | 90 | 90 | 93 | 98 | 93 | 93 | 98 | 93,41 |

Elaborado por: Autores.

Análisis: Se realizó la encuesta a 11 empleados de la empresa, en la cual se obtuvo los siguientes porcentajes de usabilidad; el usuario 1 obtiene un 93%, el usuario 2 obtiene un 90%, el usuario 3 obtiene un 100%, el usuario 4 obtiene un 93%, el usuario 5 obtiene un 90%, el usuario 6 obtiene un 90%, el usuario 7 obtiene un 93%, el usuario 8 obtiene un 98%, el usuario 9 obtiene un 93%, el usuario 10 obtiene un 93%, el usuario 11 obtiene un 98%. En promedio el sistema de requisiciones obtiene 93.41% de aceptación por parte de los usuarios luego de ser aplicado el test SUS (*System Usability Scale*).

En la Tabla N° 6 se obtuvo un promedio de los resultados en cada pregunta del test SUS, con lo cual se realizó un gráfico estadístico de la incidencia de cada pregunta realizada.



Figura 6: Evaluación por pregunta

Análisis: En la figura 6 se observa el porcentaje obtenido en cada pregunta. En las preguntas impares se obtuvo los siguientes resultados: pregunta 1 = 92.73%, pregunta 3 = 100%, pregunta 5 = 96.36%, pregunta 7 = 94.55%, pregunta 9 = 94.55%; cabe recalcar que según el test SUS estas preguntas son consideradas como positivas porque su cercanía al completamente de acuerdo (5). En las preguntas pares se obtuvo los siguientes resultados: pregunta 2 = 29.09%, pregunta 4 = 23.64%, pregunta 6 = 23.64%, pregunta 8 = 21.82%, pregunta 10 = 32.73%; según el test SUS estas preguntas son consideradas como negativas por su cercanía al completamente desacuerdo (1).

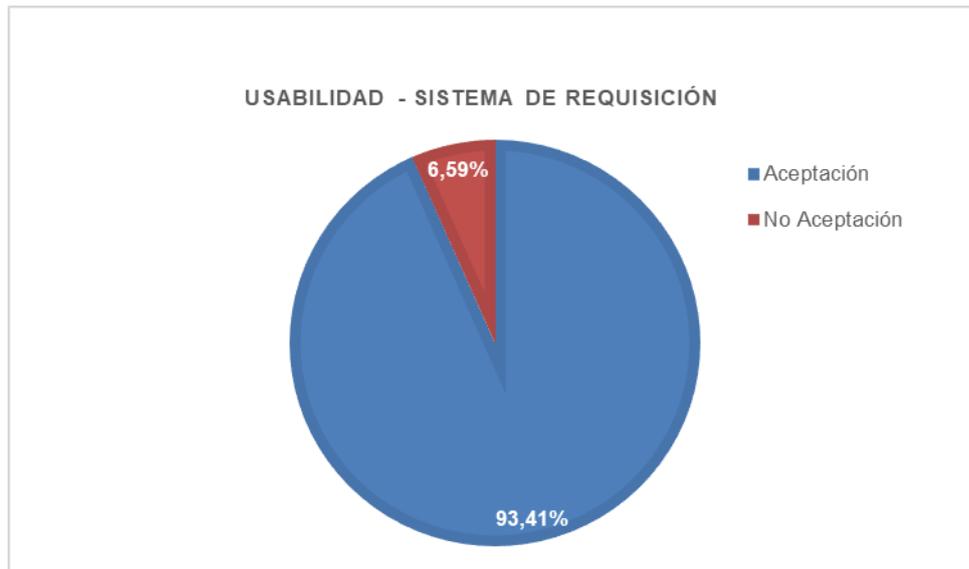


Figura 7: Usabilidad Sistema de Requisición

Análisis: En la figura 6 se observa el grado de usabilidad que obtuvo el sistema de requisiciones implementado en la empresa Hidroabánico S.A., que es del 93,41% por lo que el sistema cumple con los parámetros básicos de usabilidad para el usuario.

3.2. Análisis de resultados

Para iniciar el desarrollo de la investigación, se obtuvo información mediante la entrevista realizada al personal administrativo de la empresa. A continuación, se pasó a la etapa de análisis donde, se definió los requerimientos funcionales y no funcionales. Para el desarrollo de la aplicación se aplicó la metodología MSF (*Microsoft Solutions Framework*) que resulta ser un método estructurado, flexible y eficiente para un grupo de desarrollo pequeño. Una vez concluido el proceso de desarrollo de la aplicación, se procede a realizar el test SUS (System Usability Scale) a los usuarios reales de prueba, además de realizar pruebas reales en el sistema con los usuarios.

Al término del desarrollo de la aplicación de requisiciones, su resultado permite ver que:

- La aplicación web nos ayuda con disponibilidad en cuanto tiempo y lugar, se podrá trabajar en ella en el momento que surge el pedido de requisición y en cualquiera de sus plantas u oficina mientras tengan conexión a internet.
- Para trabajar en la aplicación de requisiciones solo necesita un computador, un navegador web y conexión a internet.
- La aplicación web sirve para mantener una comunicación adecuada continua entre el personal operativo que se encuentra en las plantas (casa de máquinas o represa) y el personal administrativo.
- La utilización de la aplicación por parte del personal administrativo y operativo, no requiere una guía constante por parte del desarrollador ya que la página web es de fácil interpretación para que el usuario, cabe recalcar que los usuarios deberán tener un mínimo de conocimientos en la utilización del computador.
- Una vez aplicado el test SUS, una encuesta estandarizada de 10 preguntas para medir la usabilidad de un producto software, la tabulación arroja el 93.41% en la escala de usabilidad.

4. CONCLUSIONES:

- La utilización del sistema de requisiciones permite incrementar la comunicación entre los empleados a través de notificaciones en tiempo real, además agiliza la toma de decisiones por parte del personal administrativo, quien se basará en el inventario o reportes que proporciona la página web.
- La utilización de javascript del lado del servidor a través de node.js permite la creación del sistema web de requisiciones en tiempo real a través de websockets, que no permite puntos muertos ni bloqueos por llamadas de entrada y salida, basada en la comunicación cliente\servidor.
- El framework de desarrollo de sitios web Django permite protección a las cuentas de usuario y contraseñas, evitando errores comunes como colocar la información de la sesión en cookies donde es vulnerable, protege la inyección SQL; además permite la reutilización de componentes sin la necesidad de empezar desde cero.
- La metodología de desarrollo Microsoft Solutions Framework permite trabajar en grupo reducido de desarrollo, además soporta adaptabilidad a cambios repentinos en el sistema sin afectar el trascurso de la ingeniería.
- Al aplicar el test SUS (System Usability Scale) basado en la Norma ISO/IEC 9126-1 para medir la usabilidad del sistema de requisiciones, se obtuvo un 93.41%, lo cual indica la aceptación por parte de los usuarios hacia la aplicación web.

5. BIBLIOGRAFÍA:

CASTRELO, Alejandro. MMO de Navegador en Tiempo Real con Node.js y WebSockets. (Tesis de Ingeniería Informática). [en línea] Universidad de Barcelona, Facultad de Matemáticas, Barcelona, 2014. [Consulta: 14 de marzo 2018]. Disponible en: <http://diposit.ub.edu/dspace/bitstream/2445/59452/1/memoria.pdf>

CATALDI, Zulma. Metodología de diseño, desarrollo y evaluación de software educativo. (Tesis de maestría). [en línea] Universidad Nacional de la Plata, Facultad de Informática, Colombia. 2000. [Consulta: 10 de enero 2018]. Disponible en: http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2737/0053L864e_anexo.pdf;jsessionid=49CEC8600FD92C87B4599BBE4FAEAD6A?sequence=2

FLORES, Ángel. Desarrollo de un Entorno Virtual Colaborativo Aplicado a la Enseñanza del Diseño Web, en la Epoch Extensión Morona Santiago para mejorar el aprendizaje significativo (Tesis de maestría). [En línea] Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica. Riobamba-Ecuador. 2016. [Consulta: 10 febrero 2018]. Disponible en: <http://dspace.epoch.edu.ec/bitstream/123456789/4687/1/20T00704.pdf>

GARCÍA, Ana. Evaluación de métricas de calidad del software sobre un programa Java (Tesis) (maestría). Universidad Complutense de Madrid, Facultad de Informática, Departamento de Sistemas Informáticos y Computación. Madrid-España. 2010. pp. 7-8 [Consulta: 15 julio 2007]. Disponible en: <http://eprints.ucm.es/11487/>

- GARCÍA, Dina. Propuesta para mejorar la respuesta a las requisiciones de compra de crear ingeniería Ltda. Mediante simulación en arena y prueba bonferroni. (Trabajo Aplicado). [en línea]. Politécnico Grancolombiano Institución Universitaria, Facultad de Ingeniería Industrial, Especialización en logística y gestión de la cadena de abastecimiento. Bogotá-Colombia. 2015. [Consulta: 15 de marzo 2018]. Disponible en: <http://repository.poligran.edu.co/bitstream/handle/10823/648/Proyecto%20de%20Grado%20Antonia%20Garcia.pdf?sequence=1&isAllowed=y>
- GARCÍA, Oscar. Northware. Software Development. [en línea]. 2014. Aplicaciones en Tiempo Real con node.js. [Consulta: 14 de marzo del 2018]. Disponible en: http://www.northware.mx/wp-content/uploads/2014/03/northware_marzo2014_aplicaciones_tiemporeal_nodejs.pdf?x69168
- GAUCHAT, Juan. EL GRAN LIBRO DE HTML5, CSS3 Y JAVASCRIPT, Primera Edición, Barcelona-España, Editorial MARCOMBO: 2012, pp. 25-70.
- GÓMEZ Andrés., & ALVARADO, Laura. Análisis de usabilidad de la página web de la Universidad Católica de Colombia. (Trabajo de Investigación). [en línea]. Universidad Católica de Colombia, Facultad de Ingeniería en Sistemas. Bogotá-Colombia. 2015. [Consulta: 15 de marzo 2018]. Disponible en: <http://repository.ucatolica.edu.co/bitstream/10983/1295/1/TRABAJO%20DE%20GRADO.pdf>
- GÓMEZ, Diana. Desarrollo del sistema de requisiciones para la empresa hidroelectrica abanico s.a. Aplicando el entorno de programacion node.js. (Trabajo de Investigación). Escuela Superior Politecnica de Chimborazo, Facultad de Informática y Electrónica. Riobamba-Ecuador. 2018.
- GUADALUPE, Eduardo., & MOROCHO, Saúl. Desarrollo de una Aplicación Web Utilizando la Norma NTE INEN-ISO/IEC 9126-1 para Automatizar el Cobro del Servicio de Agua Potable en la Provincia de Morona Santiago (Tesis de pregrado). [en línea] Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica. Riobamba-Ecuador. 2016. [Consulta: 15 agosto 2017]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/4732#sthash.iIHj1XJS.dpuf>
- LANDÁZURI, Carlos. Estudio de la metodología MSF "Microsoft Solutions Framework" aplicada al desarrollo de un sistema de facturación "software libre" para la unión de papelerías de la ciudad de Ibarra. [en línea]. Ecuador- Ibarra: Universidad Técnica del Norte. [Consulta: 14 de marzo 2018]. Disponible en: <http://repositorio.utn.edu.ec/bitstream/123456789/1830/3/ISC%20258%20Art%C3%ADculo%20Cient%C3%ADfico%20Español.pdf>

MICROSOFT SOLUTIONS FRAMEWORK (MSF). Microsoft Documentation [en línea].
Microsoft solutions Framework (MSF) Overview. Disponible en:
[https://msdn.microsoft.com/es-es/library/jj161047\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/jj161047(v=vs.120).aspx)

MOZILLA, mdn. Acerca de Javascript. [en línea]. [Consulta: 14 de marzo 2018]. Disponible en:
https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript

ONGEI, Mark. Guía Técnica sobre Evaluación de software en la Administración Pública [en línea]. Piura-Perú: Oficina Nacional de Gobierno Electrónico e Informática. [Consulta: 15 de marzo 2017]. Disponible en:
http://www.ongei.gob.pe/bancos/banco_normas/archivos/guia-evaluacion-sw.pdf

Usabilidad web, un test que te dará grandes resultados [en línea]. QuestionPro. 2018.
[Consulta: 24 de febrero 2018]. Disponible en:
<https://www.questionpro.com/blog/es/usabilidad-web-test/>