

# ***Sistema de Generación y Seguimiento de Solicitudes de Asistencia a los Ciudadanos***

Antonio J. Correa López

Maestría en Ingeniería en Ingeniería de Computadoras

Dr. Nelliud Torres

Departamento de Ingeniería Eléctrica y Computadoras y Ciencia de Computadoras

Universidad Politécnica de Puerto Rico

---

**Abstracto** — *El 24 de septiembre de 2017, el huracán María azotó nuestra Isla de Puerto Rico y causó daños extremos y la necesidad de ayudar a muchas personas en condiciones difíciles. La necesidad de ser eficientes en los esfuerzos para ayudar a las personas requiere buena información y una mejor respuesta de parte de las agencias gubernamentales. En mi ciudad natal, Cayey, el alcalde organizó el personal y los recursos de la ciudad para ayudar a los ciudadanos en dificultades. Una herramienta que se utilizó para organizar este proyecto fue un sistema de reportes de problemas. Este sistema no fue optimizado ni en opciones ni en plataforma. Entendí que proporcionar al gobierno municipal una mejor herramienta para organizar las necesidades de nuestros ciudadanos sería de gran ayuda e incluso podría salvar vidas.*

**Términos Claves** — *Base de datos, Fácil acceso, Manejo de querellas, Paquetes de código.*

## **INTRODUCCIÓN**

En septiembre 24 el huracán María impactó la Isla de Puerto Rico tumbando gran parte de la estructura del país dejando toda la población sin las necesidades esenciales y comunicación. Cada municipio se dedicó a ayudar a sus residentes con diferentes tipos de asistencias desde repartir comida hasta recogido de escombros [1].

En Cayey se eligió el Coliseo Municipal, localizado en el pueblo, como centro de emergencia para que los residentes pudieran ir a solicitar ayuda. En el centro había facilidades que la gente podía utilizar y servicios que eran ofrecidos como comida, agua y electricidad. Además de ofrecer servicios, el centro también tenía información sobre

el estatus de las facilidades de agua y energía eléctrica para los diferentes barrios y sectores.

El servicio desarrollado bajo este proyecto va a estar enfocado en el manejo de querellas del centro de atención al ciudadano de Cayey, que fue ejecutado eficientemente, pero que puede ser mejorado en varias áreas. Al observar el proceso de manejar las solicitudes de los ciudadanos, uno se percata de que las solicitudes son manejadas mediante el uso de una computadora utilizando el programa de *Microsoft Access*, cuya versión era antigua.

La estructura en la que se guardan las solicitudes de las personas se puede considerar ineficiente ya que hay una moderada probabilidad de que se pierda la información. Por esa razón, y otras no mencionadas, se decidió crear un programa que provea una interfaz para manejar solicitudes y que provea herramientas para aumentar la eficiencia de la operación. De esta manera ayudar a más ciudadanos y aumentar el sentido de responsabilidad de los recursos a cargo de trabajar cada querella.

El programa que se va a estar creando es un sistema de seguimiento de incidentes (*support ticket system*) el cual maneja los incidentes que son reportados por clientes para que los empleados del municipio puedan atenderlos eficientemente.

El proyecto consiste en la creación de un programa o aplicación que tenga las funciones del sistema de manejo de incidentes existente, pero mejorado, para que sea más eficiente y esté más al día con programación más moderna. Esta nueva aplicación va a necesitar la capacidad de operar sin tener acceso al Internet considerando lo ocurrido en la devastación causada por el huracán María que interrumpió todo acceso al Internet ya que afectó los sistemas terrestres e inalámbricos (excepto

satélite) por la destrucción de la infraestructura de telecomunicaciones y la falta de energía. La operación normal de esta nueva aplicación requerirá el almacenar las bases de datos utilizadas por el mismo en la “nube” (*cloud*) por seguridad, resguardo y acceso por varios usuarios. Cuando el acceso al Internet no está disponible, la aplicación utilizaría la base de datos que fue bajada la última vez que fue conectado a la base de datos en la nube. Esto es porque cuando se conecta a la nube, se bajará la base de datos automáticamente en la computadora del usuario, en el espacio que esté la aplicación de la base de datos.

También va a poder generar reportes para que los encargados tengan suficiente información para poder actuar adecuadamente. Además de las funciones anteriores, esta aplicación tendrá que ser fácil de acceder y manejar para que los empleados no tengan dificultad en su uso y sobre todo para que no se pierda ninguna querella en el sistema.

## INICIO DEL PROYECTO

Al comienzo del proyecto, el primer paso fue recopilar información sobre los sistemas de querellas que se están usando al presente por varias organizaciones y observar todo el procedimiento que se ejecuta para manejar una querella.

El proceso es el siguiente: 1) un cliente entra a una página de Internet de la compañía u organización de la cual será la querella; 2) se inicia una sesión y se autentica al usuario en la página para que se pueda identificar el autor de la querella y se obtenga información sobre la persona para que la solución que se desarrolle tome en consideración la condición de la persona; 3) el cliente llena un formulario con la información requerida para después ser enviada a un departamento de la compañía u organización que se encarga de manejar las querellas; 4) finalmente la querella es ejecutada por un empleado para atender a los clientes. Toda querella completada o no, es guardada en los servidores de la organización para referencia y para hacer reportes sobre los niveles de servicios ofrecidos.

Antes de comenzar el desarrollo del programa, había que analizar el sistema utilizado para el manejo de querellas en el municipio para observar el proceso. Esto requirió de una visita al coliseo y a las oficinas de la alcaldía luego de que se recuperó la normalidad en el municipio.

El alcalde de Cayey autorizó y permitió el acceso al sistema y a la información para observar cómo se manejaban las querellas en el municipio. Lo primero que se observó fue el uso de un sistema para guardar la data recopilada bajo el programa *Microsoft Access*, el cual es operado desde una computadora portátil. La versión de *Access* utilizada no era una reciente.

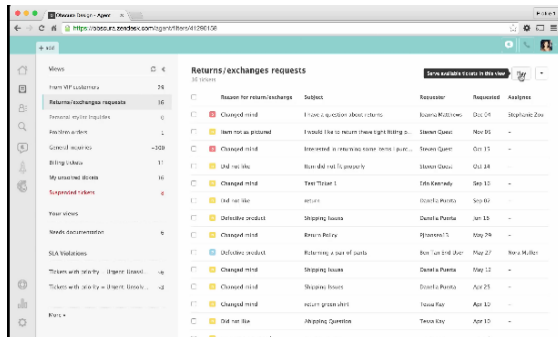
La programación utilizada es una en donde el personal se ve obligado a entrar toda la data campo por campo sin tener reglas específicas que limiten la entrada de datos. Esto permitiendo la entrada de información inconsistente y en ocasiones errónea. Al analizar la base de datos generada por este programa se pudo visualizar la cantidad de errores en las direcciones, especialmente en los nombres de los barrios y sectores, que se dejaban a la interpretación de la persona que los escribía. Sin duda alguna esta es un área en donde un nuevo programa con lista desplegable (*drop-down-list*) eliminaría este tipo de error.

Luego de observar el procedimiento que se lleva a cabo para el manejo de las querellas por el Municipio de Cayey, se puede analizar los aspectos que se deben mejorar para facilitar tal procedimiento.

Basándose en la información obtenida, ahora se procede a buscar información de varios modelos de sistemas de manejo de querellas para tener referencias que se podrán utilizar para el desarrollo de la aplicación.

El primer sistema que se evaluó fue *Zendesk* (<https://www.zendesk.com/>) que es un sistema que tiene popularidad entre varias compañías como Airbnb, TESCO, y UBER. El licenciamiento de este programa tiene varios planes de pago dependiendo de los servicios que uno escoja como manejo de querellas, un guía para los servicios, charla (*chat*) en vivo, y apoyo telefónico. El

sistema también incluye un paquete el cual es designado para uso de desarrolladores de programación que incluye el API del sistema para poder hacer cambios que puedan ser adaptados a los requerimientos del cliente.



**Figura 1**

**Página de Querellas de la Aplicación de Zendesk**

El siguiente sistema fue *freshdesk* (<https://freshdesk.com/>), que tiene los servicios similares al primer producto mencionado como manejo de querellas y conversación electrónica. Una de las bondades de este sistema resulta ser que evita la generación de solicitudes duplicadas. Otra de las características, que resulta interesante, es la automatización de ayuda al cliente. Esta disminuye el tiempo que se tardaría un cliente en esperar que alguien atienda su querella.

El programa de *Jitbit Helpdesk* (<https://www.jitbit.com/>) es otro sistema de manejo de querellas que se analizó para considerar nuevas ideas para el desarrollo del nuevo programa. Un ejemplo de las opciones de este programa es el de poder integrarse con otros programas mediante el uso de conector (*plug-in*) o interfase.

### Desarrollo del Proyecto

El siguiente paso es buscar que plataformas serían útiles para implementar los requerimientos de este proyecto. Se decidió inicialmente escoger *Visual Studio 2017* [2], ya que gran parte de los sistemas o programas que son utilizados en Puerto Rico son de *Microsoft*, sería conveniente tener una aplicación que pueda ser compatible con otros sistemas. Como referencia se utilizó *StatCounter Global Stats* [3] quienes indican que el 62.71% de

las computadoras de las empresas utilizan *Windows*.

Lo próximo es identificar qué tipo de infraestructura (*framework*) se utilizaría para facilitar el manejo del código del programa. Una característica que es muy importante para la selección de la plataforma es el apoyo o servicios que ofrecen, especialmente cuando hay programas en donde puedes instalar diferentes librerías o paquetes que ayuden a acortar el ciclo de desarrollo. Ya las librerías pueden tener código hecho para ciertas funciones que van a ser requeridas en el programa que se está construyendo.

Lo primero que se intentó hacer era una aplicación que corra utilizando el navegador con la ayuda de *ASP.NET* con el formato de *Web Forms*. Pero luego de considerar el tener que utilizar un navegador para iniciar la aplicación, el concepto que se quería implementar era evitar que se tuviese que activar un navegador para iniciar la aplicación. Por esa razón se decidió utilizar *Windows Form*, este sería esencial para crear una aplicación que no se utilizara para iniciarse un navegador.

Desafortunadamente hubo varios problemas con el uso del formato de *Windows Form* que evitaron que se usara el sistema. En especial que surgía un error que causaba que la interface del programa se borrara. Luego se descubrió que era un error en *Visual Studio* que ocurría varias veces en el uso de *Windows Form*, que al presente no ha sido resuelto.

Finalmente se encontró un *framework* que es parte de *Node.js* [4], un paquete de *JavaScript* [5] que maneja las conexiones de servidores. Este se podía utilizar para la creación del programa y se podía usar sin la necesidad de un navegador. El mismo se llama *Electron* [6] y fue utilizado para desarrollar aplicaciones como *Skype*, *GitHub Desktop*, *Atom*, y hasta ayudo a desarrollar la aplicación de *Visual Studio Code* [7] (fue el editor de texto que se utilizó para desarrollar esta aplicación). Este *framework* utiliza *Chromium* y *Node.js* para construir una aplicación con el uso de lenguajes de programación para Web como *HTML*,

CSS, y JavaScript, lo cual es acceso abierto, y es compatible con las plataformas de Mac, Windows, y Linux. Lo más crucial de Electron es que puedes poder crear aplicaciones que normalmente requieren el uso de un navegador como aplicaciones independientes.

## ANÁLISIS DE REQUERIMIENTOS

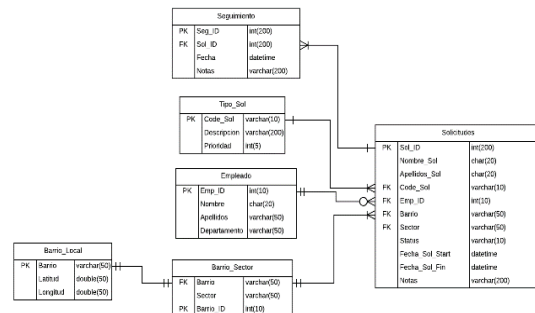
Al encontrar los utensilios para poder desarrollar la aplicación, ahora era tiempo de analizar los requerimientos para poder implementar funciones en el diseño. Lo primero es entender el tipo de información que se va a estar manejando y en donde se va a guardar esta información que luego se usaría para la aplicación. Desde el principio ya se entendía que era necesario el uso de una base de datos para almacenar la información. Habría que diseñar la estructura de esa base de datos de manera que la información que se va a estar almacenando es: información del cliente que se recopila en la creación de la querella, del empleado que va a cumplir con la querella y la información sobre la querella.

El tipo de servicio implementado fue de cliente a servidor, porque los clientes o usuarios estarían utilizando la base de datos para pedir información que tienen almacenada. El modelo de base de datos que se usaría es el modelo relacional que es el más común entre todos. El diagrama de la base de datos fue creado utilizando el sitio web de Lucidchart que es muy útil para la creación de estos tipos de diagramas. También tiene la función de exportar los diagramas a diferentes programas de base de datos para que el programa desarrolle la base de datos con todas sus relaciones y restricciones que son añadidos al diagrama.

El primer diseño del diagrama de la base de datos incluía seis tablas de datos:

- La primera era la tabla que ocupaba la información de las solicitudes con todos los datos necesarios como el numero de la solicitud, nombre del solicitante y el tipo de solicitud.

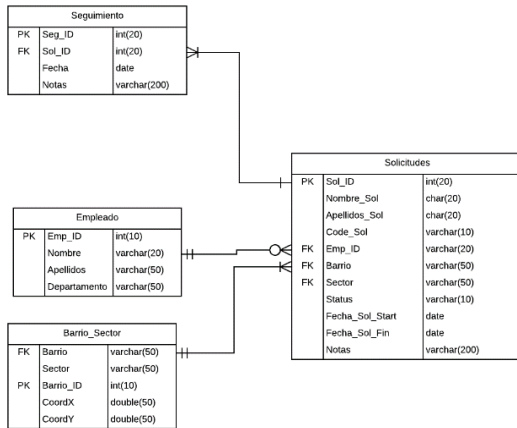
- La segunda tabla era la de seguimiento que tendría la información añadida sobre solicitudes específicas que describe el proceso de la solicitud.
- La tercera tabla sería la de tipo de solicitud que tendría información de los diferentes tipos de solicitudes que puede haber con sus respectivos números de identificación y el nivel de prioridad que lleva asignado.
- La cuarta tabla es la de empleado que llevaría la información privada de los empleados del municipio como nombre y departamento que pertenece.
- La próxima tabla era la de barrio y sectores que llevaría los datos sobre los diferentes sectores de cada barrio.
- La última tabla era la localización de cada barrio a base de coordenadas para añadir mapas en la aplicación que puede ayudar al empleado a localizar al cliente.



**Figura 2**  
Diagrama del Primer Diseño Desarrollado para la Aplicación

Luego en el desarrollo del programa se simplificó la base de datos al remover algunas de las tablas mencionadas porque no había suficiente tiempo para poder añadir servicios que utilizarían esas tablas eliminadas. Se eliminó la tabla de las coordenadas de los barrios porque no se pudo implementar la idea de tener un mapa que demostrará las direcciones de la persona que sometió la solicitud para facilitar su asistencia, pero el servicio puede ser añadido en un desarrollo futuro de la aplicación.

La tabla de tipo de solicitud fue eliminada porque los empleados que las manejan no tienen una lista de solicitudes fijas. La solicitud varía por cliente y la prioridad fue añadida a la tabla de solicitudes, lo cual puede ser editado en la aplicación.



**Figura 3**  
**Diagrama del Diseño Final después del Análisis del Primer Diseño**

Durante la fase del diseño de la aplicación, se tenía planificado utilizar *Microsoft SQL Management Studio* [8] porque se iba a desarrollar la aplicación en la plataforma de *Windows* usando *Visual Studio* que es de *Microsoft* para que haya más compatibilidad entre los componentes de la aplicación. Después de varias pruebas con las aplicaciones desarrolladas con *Visual Studio* hubo varios problemas con la ejecución del programa y se tuvo que cambiar de *Visual Studio 2017* para otra aplicación que codifica con el formato que se eligió después. Además de cambiar de programa de desarrollo, también se cambió la base de datos a *phpMyAdmin* [9] porque es la más utilizada para programas que son basados en diseño de Web, por lo cual se emplea *XAMPP* [10] para iniciar la base de datos para el uso en la aplicación.

La aplicación utiliza varias librerías o paquetes de código que son bajadas desde el Internet utilizando un *NPM (Node Package Manager)* [11] que se encarga de manejar las librerías o paquetes que se instalan para el uso de la aplicación. Estos paquetes se instalan escribiendo cuál se quiere instalar en la consola de la computadora, utilizando

la dirección de la localización del programa. Los paquetes que se instalaron fueron los de *Electron*, *Node.js*, *Materialize.css*, *Bootstrap*, *Chart.js*, y *Mysql*.

*Electron* es un sistema que se utiliza para crear aplicaciones usando lenguajes de programación que son para crear páginas en la Web. Este sistema ayuda a crear aplicaciones nativas que utilicen funciones y herramientas que son exclusivas a las plataformas evitando el problema de tener que codificar diferente dependiendo de la plataforma que va a utilizar la aplicación. Esto es posible por la utilización de programación de Web la cual es la misma para todas las plataformas.

*Electron* utiliza los módulos de *Node.js* y *Chromium* para correr las aplicaciones, pero también puede correr con otros módulos independientes. El paquete o librería de *Materialize.css* [12] se utiliza para la apariencia de la aplicación, para darle una forma más moderna.

*Bootstrap* [13] se utilizó para darle función a varias secciones de la aplicación y se necesitaba para operar el paquete de *Chart.js* [14] que se utilizó para crear gráficas de los datos que se recopilan de la base de datos para poder hacer toda información más presentable a quien lo analice.

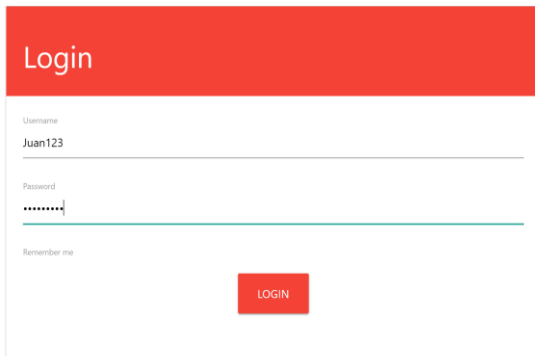
El último paquete que se utilizó fue el de *Mysql*, este es el más importante porque maneja todos los intercambios de la base de datos con la aplicación.

Esta aplicación utilizaría unas bases de datos locales y la nube para almacenar dicha información a modo de resguardo y de bases de datos compartidas con diferentes usuarios. Las bases de datos serán sincronizadas en la nube para asegurarse de que los usuarios compartan la misma información. Esta función es posible mediante el uso de un programa llamado *SQLBackupFTP* [15] que se conecta a la base de datos local y envía los datos a la nube que esté disponible y sincronizada.

## OPERACIÓN DE LA APLICACIÓN

La aplicación comienza con una página de iniciar sesión para que el empleado pueda ingresar

su nombre de usuario y contraseña para empezar una sesión del programa. Toda información de los empleados será buscada en la base de datos o en el servidor que se estará usando en el municipio y cualquier intento de entrar contraseñas incorrectas le bloqueará el acceso a la aplicación por un tiempo. Un nuevo empleado tendrá que ser añadido al servidor o base de datos directamente porque la aplicación no tendrá esa función, pero puede ser considerado en instalaciones futuras.



**Figura 4**  
**Página de Iniciar Sesión**

Ahora el empleado está observando la página principal de la aplicación en la cual puede notar una lista de querellas y organizarla por prioridades, dependiendo de la preferencia del usuario. Además puede ver: las diferentes opciones para manejar las querellas, como añadir una querella, editar la querella seleccionada, y exporta la tabla a *Excel* o *PDF*.

Otra opción que está en la esquina de abajo a la derecha es un botón para ver los cambios más recientes en la base de datos. El usuario puede ver más información de la querella si selecciona el botón al final de la fila en donde la querella se está presentando, además de ver información también puedes añadir notas a la querella y editar algunos datos como su prioridad y estatus.

La tabla de querellas tiene las funciones de organizar las filas en el orden que el usuario quiera, y el usuario tiene la opción de poder buscar una querella específica con escribir un dato específico en la barra de búsqueda.

La tabla en la página principal demuestra las querellas dependiendo de la prioridad que el

usuario escoja que se vea y puede escoger las de prioridad alta, media o baja. El botón al final de las filas en la tabla al seleccionarlo los lleva a otra página que se utilizaría para cambiar el estatus de la querella, su prioridad, y añadir una nota sobre el proceso de la querella.

ID	Nombre	Tipo de Solicitud	Empleado	Estatus	Fecha de Inicio	Nota
1	Pedro	Recogido de basura	Juan Veloz	INC	Wed Aug 08 2018 00:00:00 GMT-0400 (SA Western Standard Time)	Ver Nota
2	Maria	Recogido de escombros	Luis Berrios	Comp	Wed Aug 08 2018 00:00:00 GMT-0400 (SA Western Standard Time)	Ver Nota
3	Juan	Limpias	Martin	Prec	2000-00-00	Ver Nota
4	Juan	Basura	Maria	INC	2000-00-00	Ver Nota
5	Diego	Reparacion	Martin	INC	2000-00-00	Ver Nota
6	Diego	Limpias	Martin	INC	2000-00-00	Ver Nota

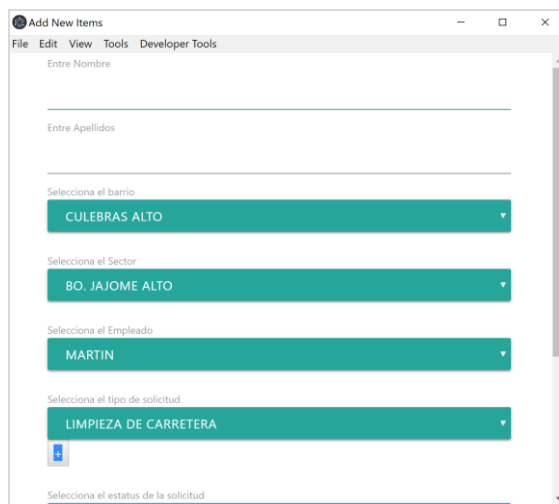
**Figura 5**  
**Página Principal de la Aplicación**

La página de editar querella va a demostrar información principal como el nombre del solicitante, empleado que está manejando la querella, el tipo de solicitud que es, la dirección del solicitante, y las notas sobre la solicitud. Las opciones que hay para manejar o editar la querella en la página son: cambiar el estatus de la querella para que se notifique de que se está procesando o se haya completado, cambiar la prioridad de la querella y añadir una nota para dar una explicación del proceso que se esté llevando a cabo. Cada una de las notas añadidas tiene la fecha de cuándo se realizó.

**Figura 6**  
**Página de Editar Querella o Añadir Notas**

Otra función que tiene la página principal es la creación de una nueva querrella. Se puede empezar al darle al botón de añadir querrella que está localizado en el área superior de la página a la derecha.

Al iniciar el proceso de añadir una querrella aparece otra página con varios campos que son necesarios para tener información como: el nombre y apellido del solicitante, barrio y sector, empleado que se va a ocupar de la querrella, y tipo de solicitud. Toda la información que es escrita y seleccionada es guardada directamente a la base de datos para luego poder verlo en la página principal de la aplicación.



**Figura 7**  
**Página de Crear Nueva Querrella**

Como fue mencionado anteriormente sobre el manejo de querrellas, la entrada de los datos era hecha manualmente por los empleados dando la posibilidad de errores. Ahora la aplicación tiene algunos campos como barrio, sector, empleado y tipo de solicitud, que utilizan campo tipo *drop-down list*. Esto ofrece las diferentes alternativas y evita que el empleado pase trabajo escribiendo información adicional y posiblemente con inconsistencia y errores.

La próxima función que la aplicación puede ejecutar es el análisis de los datos de las querrellas y hacer un reporte para el usuario para poder observar detalladamente lo que está ocurriendo con todas las querrellas. Esta función ayudaría a mejorar los

servicios que se le ofrecen a los clientes o residentes del municipio porque podrán notarse las deficiencias que pueda tener ciertos ofrecimientos de servicios. El reporte va a demostrar datos en forma de graficas para que sea más fácil de entender. También tendrá la función de generar ese reporte en formato de *PDF* para poder imprimirlo y que sea más fácil llevar esa información a otra persona que requiera datos.



**Figura 8**  
**Página de Análisis de Datos**

## CONCLUSIÓN

Los requerimientos que fueron establecidos al principio de la creación de este proyecto fueron cumplidos con las funciones de la aplicación. Se pudo crear una página donde se pueden ver las querrellas que se han creado. También tiene varias formas de organizar las querrellas como por columnas y por prioridad del tipo de solicitud. La aplicación puede crear querrellas con toda la información necesaria y añadirlas a la base de datos donde se guardan el resto de las querrellas creadas. Toda querrella tiene la opción de darle seguimiento al poder añadir notas que expliquen el procedimiento que se este haciendo para completar o solucionar la misma. Finalmente tiene la opción de ver estadísticas sobre las querrellas que se han

hecho y se puede observar para hacer un análisis y mejorar el proceso.

La aplicación tiene mucho espacio para mejorar con varias ideas para instalaciones futuras. Unas de las ideas es mejorar la seguridad del manejo de información de las querellas con los empleados, también se puede mejorar la actuación del procedimiento, y mejorar la conexión entre la aplicación y la base de datos. Otra mejora que se puede implementar en la aplicación es añadirle un mapa en la página de crear querellas para poder facilitar el trabajo de los empleados al servirle al cliente. Esta era una idea que se iba a implementar desde el principio, pero no se pudo por el tiempo y la experiencia que se tenía al momento del desarrollo. La aplicación se puede expandir a otros dispositivos electrónicos como el celular, lo cual haría que se puedan crear aplicaciones que pueden usar los empleados que sirven a los clientes, y esta aplicación puede notificar a los empleados sobre las querellas asignadas a ellos y su localización.

Finalmente, algo que se aprendió en este proceso fue estar abierto a evaluar diferentes productos de desarrollo de aplicaciones para poder encontrar el producto que más se acerca a cumplir con las expectativas de los usuarios.

La oportunidad de trabajar con diferentes plataformas de desarrollo permitió aprender de todas ellas, aunque en ocasiones resultaba un poco frustrante porque no se lograba desarrollar la aplicación que se necesitaba. Hay que estudiar las mismas en detalle para poder entender su operación y sus capacidades, y así alcanzar el mejor resultado posible dentro de los parámetros del proyecto que incluyen el tiempo de desarrollo.

## REFERENCIAS

- [1] Editorial El Nuevo Día, “Desolación Extrema en la zona montañosa”, El Nuevo Día, p. 28-29, Sept. 23, 2017. [Online]. Available: [epaper.elnuevodia.com/el-nuevodia/20170923](http://epaper.elnuevodia.com/el-nuevodia/20170923). [Accessed: Nov. 12, 2018].
- [2] Microsoft Corporation. (1997, Feb.). *Visual Studio IDE* [Online]. Available: <https://visualstudio.microsoft.com/vs/>. [Accessed: Nov. 15, 2018].
- [3] StatCounter. (2018, Ene.). “Desktop Operating System Market Share Puerto Rico | StatCounter Global Stats”, *StatCounter Global Stats* [Online]. Available: <http://gs.statcounter.com/os-market-share/desktop/puerto-rico>. [Accessed: Nov. 14, 2018].
- [4] Ryan Dahl. (2009, Mayo 27). *Node.js Foundation* [Online]. Available: <https://nodejs.org/en/docs/>. [Accessed: Nov. 18, 2018].
- [5] Refsnes Data. (1999). *JavaScript Tutorial* [Online]. Available: <https://www.w3schools.com/js/>. [Accessed: Nov. 17, 2018].
- [6] GitHub Community. (2013, Ago. 12). *Electron | Build cross platform desktop apps with JavaScript, HTML, and CSS* [Online]. Available: <https://electronjs.org/>. [Accessed: Nov. 15, 2018].
- [7] Visual Code. (2015, Abr. 29). *Documentation for Visual Studio Code* [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed: Nov. 15, 2018].
- [8] Microsoft Corporation. (2005). *SQL Server Management Studio (SSMS)* [Online]. Available: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>. [Accessed: Nov. 19, 2018].
- [9] phpMyAdmin Contributors. (1998, Sept. 9). *Bringing MySQL to the Web* [Online]. Available: <https://www.phpmyadmin.net/docs/>. [Accessed: Nov. 22, 2018].
- [10] Apache Software Foundation. (1995, Abr.). *XAMPP Installers and Downloads for Apache Friends* [Online]. Available: <https://www.apachefriends.org/index.html>. [Accessed: Nov. 22, 2018].
- [11] I. Z. Schlueter. (2010, Ene. 12). *npm Documentation* [Online]. Available: <https://docs.npmjs.com/>. [Accessed: Nov. 16, 2018].
- [12] A. Wang. (2014). *About – Materialize* [Online]. Available: <https://materializecss.com/about.html>. [Accessed: Nov. 19, 2018].
- [13] Refsnes Data. (1999). *Bootstrap 3 Tutorial* [Online]. Available: <https://www.w3schools.com/bootstrap/default.asp>. [Accessed: Nov. 17, 2018].
- [14] Chart.js Community. (2016). *Chart.js · Chart.js documentation* [Online]. Available: <https://www.chartjs.org/docs/latest/>. [Accessed: Nov. 24, 2018].
- [15] Pranas.NET. (2008). *SQLBackupAndFTP - SQL Server, MySQL and PostgreSQL Backups* [Online]. Available: <https://sqlbackupandftp.com/mysql-backup>. [Accessed: Dic. 6, 2018].