

**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**  
**FACULTAD DE INGENIERÍA**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**



**PROTOTIPO DE APLICACIÓN MÓVIL PARA LA GESTIÓN DE  
COTIZACIONES ENTRE CLIENTES Y PROVEEDORES DEL ESTADO  
DE CHIHUAHUA**

**POR:**

**JESUS FRANCISCO PONCE DOMINGUEZ**

**TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE  
MAESTRO EN REDES MÓVILES**

**CHIHUAHUA, CHIH., MÉXICO**

**SEPTIEMBRE DE 2020**



Prototipo de aplicación móvil para la gestión de cotizaciones entre clientes y proveedores del estado de chihuahua. Tesis presentada por Jesús Francisco Ponce Domínguez como requisito parcial para obtener el grado de Maestro en Ingeniería de redes móviles, ha sido aprobada y aceptada por:

**M.I. Javier González Cantú**  
Director de la Facultad de Ingeniería

**Dr. Alejandro Villalobos Aragón**  
Secretario de Investigación y Posgrado

**M.S.I. Karina Rocío Requena Yáñez**  
Coordinador(a) Académico

**M.I. Oscar Daniel Pérez Piñón**  
Director(a) de Tesis

**Octubre 2020**  
Fecha

Comité:  
**M.I. Oscar Daniel Pérez Piñón**  
**M.I. David Maloof Flores**  
**Dr. Luis Raúl Lujan Vega**  
**M.A.T.I. Longinos Castillo Fernández**

© Derechos Reservados

**Jesús Francisco Ponce  
Dominguez**

Circuito No.1, Campus  
Universitario 2, Chihuahua, Chih.  
Septiembre 2020



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**

**ING. JESÚS FRANCISCO PONCE DOMÍNGUEZ.**

**Presente:**

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestría en Ingeniería en Computación, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director el **M.I. Oscar Daniel Pérez Piñón**, para que lo desarrolle como tesis, con el título: **“Prototipo de aplicación móvil para la gestión de cotizaciones entre clientes y proveedores del estado de Chihuahua”**.

### **Índice de Contenido**

Introducción

Marco Teórico

Metodología

Desarrollo

Resultados

Resultados

Conclusión

Lista de referencias

Anexo A Procedimientos Almacenados

Anexo B Archivos PHP

Anexo C Código Android

Curriculum vitae

Objetivo

Preparación Académica

Experiencia Laboral

**FACULTAD DE INGENIERÍA**  
Circuito No.1, Campus Universitario 2  
Chihuahua, Chih., México. C.P. 31125  
Tel. (614) 442-95-00  
[www.fing.uach.mx](http://www.fing.uach.mx)



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**  
Idiomas

Aptitudes

Habilidades personales y profesionales

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las tesis.

**ATENTAMENTE**

*“Naturam subiecit aliis”*

**EL DIRECTOR**

**EL SECRETARIO DE  
INVESTIGACIÓN Y POSGRADO**

FACULTAD DE  
INGENIERÍA  
U.A.CH.



**M.I. JAVIER GONZÁLEZ CANTÚ**

**DR. ALEJANDRO MILLALOBOS ARAGÓN**

DIRECCIÓN

**FACULTAD DE INGENIERÍA**  
Circuito No.1, Campus Universitario 2  
Chihuahua, Chih., México. C.P. 31125  
Tel. (614) 442-95-00  
[www.fing.uach.mx](http://www.fing.uach.mx)

## Resumen

Las aplicaciones móviles ofrecen nuevas perspectivas ya que son una herramienta prometedora diseñadas para realizar funciones concretas, como presentar la información del tiempo, comparar precios de un producto (Joyce; Lilley, 2014) o lograr unión entre dos entidades económicas que permita su interacción.

Los sucesivos avances tecnológicos, han dado lugar a nuevas denominaciones, a veces redundantes, que amplían el concepto de comercio, como el u-comercio o comercio ubicuo (Watson, 2002) definido «como comprar y vender a través de cualquier medio tecnológico, en cualquier lugar y cualquier hora, y compuesto por comercio electrónico, comercio por televisión, comercio por radio, comercio por voz, comercio por móvil y comercio silencioso» (Fernández y Peral, 2006).

El propósito de una aplicación es para ayudar en la consecución de un objetivo específico o en la ayuda de actividades diarias. Desde el punto de vista empresarial, las corporaciones buscan medios que les permitan hacer esta comunicación sencilla y flexible, de forma que pueda ser usada para mejorar la interacción entre la organización, proveedores y clientes (Zenel, 2005)

La implantación vertiginosa de las aplicaciones en el entorno móvil obedece a:

- razones de economía funcional: aportan valor en la relación tiempo/eficacia, algo especialmente relevante en la pequeña pantalla ubicua (Scolari; Aguado; Feijóo, 2012);
- criterios estratégicos de mercado



## Tabla de Contenido

ii

Capítulo 1 Introducción .....	1
1.1 Situación Problemática. ....	1
1.2 Planteamiento del Problema. ....	2
1.3 Limitaciones.....	3
1.4 Objetivo General.....	4
1.5 Objetivos Específicos.....	4
1.6 Hipótesis. ....	4
1.7 Justificación. ....	5
Capítulo 2 Marco Teórico.....	7
Web Services .....	7
PHP .....	7
Metodo POST .....	8
Metodo GET .....	8
MySql.....	8
Procedimientos Almacenados.....	8
Android Studio.....	9
JSON .....	9
GIT .....	10
GITHUB .....	11
Programación en N capas.....	11
Autenticación .....	12
Nombre de Marca .....	13
Logotipo.....	14
Isotipo Figurativo.....	15
Capítulo 3 Metodología. ....	15
3.1 Diseño Lógico.....	15
3.2 Diseño de la Base de datos.....	18
3.2.1 TB01_Usuario.....	18
3.2.2 TB02_Rol.....	19
3.2.3 TB03_Ciudad.....	20
3.2.4 TB04_Requerimiento.....	20
3.2.5 TB05_Unidad.....	21
3.2.6 TB06_Estatus.....	21
3.2.7 TB07_Cotizacion.....	22
3.2.8 TB08_Solicitud.....	22
3.2.9 TB09_Empresa.....	23
3.2.10 TB10_Concepto.....	23
3.2.11 TB11_Tipo_Venta.....	24
3.2.12 TB12_Moneda.....	24
3.2.13 TB13_Categoria.....	25
3.2.14 Procedimientos Almacenados.....	25
3.3 Desarrollo.....	29
3.3.1 Diseño general de la aplicación.....	29
3.3.2 Diagrama jerárquico.....	31
3.3.3Capa de Presentación.....	33



3.3.4Capa de Negocio.....	47iii
3.3.5Capa de Datos.....	54
3.3.6 Gestor de versiones.....	55
Capítulo 4 Resultados.....	56
4.1 Alta de Usuario.....	58
4.2 Acceso de usuario.....	59
4.3 Nueva Solicitud.....	60
4.3 Agregar con productos.....	61
4.4 Ingresar a proveedor.....	62
4.5 Buscar Solicitud.....	63
4.5 Buscar Producto.....	64
4.6 Cotizar Productos.....	65
4.7 Envió de correo.....	65
Capítulo 5 Conclusión.....	67
Lista de referencias.....	69
Anexo A Procedimientos Almacenados.....	71
Anexo B Archivos PHP.....	75
Anexo C Código Android.....	84
Curriculum vitae.....	89
Objetivo.....	89
Preparación Académica.....	89
Experiencia Laboral.....	89
Idiomas.....	91
Aptitudes.....	91
Habilidades personales y profesionales.....	91



## Tabla de figuras

iv

Figura 1 Principales comandos de Git. ....	10
Figura 2 Actores involucrados en la autenticación .....	13
Figura 3 Ejemplos de logotipos .....	15
Figura 4 Diagrama de caso de uso .....	17
Figura 5 Tabla de Usuario.....	19
Figura 6 Rol del Usuario.....	19
Figura 7 Tabla de Usuario.....	20
Figura 8 Tabla de Requerimiento .....	21
Figura 9 Tabla de Unidad .....	21
Figura 10 Tabla de Estatus.....	21
Figura 11 Tabla de Cotización.....	22
Figura 12 Tabla de Solicitud.....	23
Figura 13 Tabla de Empresa .....	23
Figura 14 Tabla de Concepto .....	24
Figura 15 Tabla Tipo de venta .....	24
Figura 16 Tabla de moneda.....	24
Figura 17 Tabla categoría .....	25
Figura 18 Diagrama de Base de Datos u901308218_CotApp.....	28
Figura 19 Diagrama de Capas.....	29
Figura 20 Diagrama de Flujo del Proceso General de la aplicación.....	30
Figura 21 Diagrama de Aplicación .....	32
Figura 22 Diseño de Logotipo .....	35
Figura 23 Diseñador de logotipo.....	35
Figura 24 Diseño base para el desarrollo de la interfaz grafica.....	36
Figura 25 Colores Base de la aplicación.....	37
Figura 26 Pantalla Login.....	38
Figura 27 Pantalla Registro de usuarios.....	39
Figura 28 Pantalla Inicio de Usuario. ....	40
Figura 29 Pantalla Nueva solicitud. ....	41
Figura 30 Pantalla Agregar productos .....	42
Figura 31 Pantalla inicio proveedor .....	43
Figura 32 Pantalla Buscar solicitud. ....	44
Figura 33 Pantalla Buscar Productos .....	45
Figura 34 Pantalla Cotizar Producto .....	47
Figura 35 Diagrama de acceso de usuario .....	48
Figura 36 Spinner de Roles.....	49
Figura 37 Pantalla de productos.....	50
Figura 38 Spinner de categorías.....	51
Figura 39 Spinner de ciudad .....	51
Figura 40 Listado de credito .....	52
Figura 41 Listado de unidad .....	52
Figura 42 Mensaje de solicitud agregada.....	53
Figura 43 Mensaje insertar usuario.....	53
Figura 44 Ejemplos de commits generados .....	55





Figura 45 Alta de usuario.....	58v
Figura 46 Acceso a la aplicación .....	59
Figura 47 Crear una solicitud nueva .....	60
Figura 48 Agregar producto a la solicitud .....	61
Figura 49 Pantalla de inicio de proveedor .....	62
Figura 50 Búsqueda de solicitudes .....	63
Figura 51 Búsqueda de productos.....	64
Figura 52 Búsqueda de productos.....	65
Figura 53 Finalizar cotizacion .....	66



## Capítulo 1

### Introducción

#### 1.1 Situación Problemática.

Actualmente, las compañías son muy dependientes de sus proveedores inmediatos, y todos ellos del enlace inmediato superior de la cadena de suministros, por lo que el éxito de la empresa depende de buenos proveedores, de su comunicación y de que puedan mantener costos bajos para hacer que el último de la cadena de suministros pueda ser competitivo.

[1]

Existen muchos procesos dentro de una cadena de suministro, por lo que las empresas tienen necesidades y retos que los obligan a contar con sistemas de información que les ayuden a controlar y ejecutar estos procesos. [1] Para un proveedor obtener la máxima cantidad de información de los clientes es fundamental para aumentar las ventas, mejorar las relaciones con los clientes y tratar a cada cliente del modo más eficaz y exitoso posible.

[2] Una correcta comunicación entre el proveedor y el cliente permiten un desarrollo en la relación comercial de estos actores.

Estamos ante una revolución tecnológica como lo son robotización, automatización, la inteligencia artificial, el internet de las cosas lo que hace una oportunidad de generar nuevas soluciones digitales desafortunadamente si en un tiempo de corto a mediano plazo no cambiamos el entorno tecnológico México se puede quedar en desventaja competitiva.



Actualmente existen diversas soluciones tecnológicas y de sistemas como los ERP (del inglés Enterprise Resource Planning) que se encargan de administrar los recursos internos de la empresa así como de los proveedores y sistemas. Otros sistemas como los CRM (del inglés Customer Resource Management), Intranets y Extranets que se encargan de la administración y operación de las transacciones con los clientes. [1]

Las pymes tienen una gran ventaja típicamente son más ágiles que una empresa grande eso significa nuevos servicios, nuevos productos desafortunadamente este sector del mercado que ocupa el 90% de la economía en México cree que la adopción de las tecnologías es para las grandes empresas. Según el INEGI al 2018 menos del 10% de las empresas mexicanas cuentan con algún tipo de herramienta que les ayude a eficientar sus procesos.

### **1.2 Planteamiento del Problema.**

El proceso de compra dentro de una compañía no se podría realizar sin el apoyo de proveedores que suministraran las necesidades que existen dentro de una organización sin embargo la adquisición de refacciones, piezas especializadas o algún insumo crítico se vuelve difícil a falta de una herramienta donde se pueda localizar rápidamente las empresas que ofrezcan cubrir las necesidades de dichas empresas.

Mientras que el control por parte de los proveedores se pudiera lograr mediante CRM móvil que permiten conocer las expectativas de los cliente, sus necesidades y al mismo tiempo responder rápidamente a los requisitos que dichos clientes tengan pero el



desconocimiento, el costo elevado o un paradigma al rechazo de la tecnología no permite que las empresas como las pymes puedan implementar un control de manejo de cliente para lograr ser competitivos en el mercado.

Aunque es difícil predecir la intensidad y la importancia del cambio que se va a producir en los próximos años en el comercio (algunos auguran una auténtica revolución), lo que supone ya una realidad es el hecho de que cualquier empresa debe tener en cuenta, con respecto a la venta, la comunicación y su relación con los consumidores, el reto que supone Internet, el móvil, la combinación de ambos, la web 2.0 y 3.0 y otros muchos elementos que surgen cada día y que configuran la sociedad digital.[4]

### **1.3 Limitaciones.**

El ritmo de adopción de las pequeñas empresas a las nuevas tecnologías es lento ya que hay una falsa percepción que la adopción de la tecnología es muy cara. Gracias a la aparición de la nube y de nuevas formas de almacenamiento de datos, el precio de sistemas y tecnología se ha disminuido.

Por otro lado aunque se está avanzando en las tecnologías de cifrado y certificados SSL (Secure Sockets Layer) que permiten la autenticación de dominio, nombre de la empresa, y ganar confianza a través de las llamadas marcas de confianza, todavía existe un gran temor a los «robos de identidad» y gran preocupación por la seguridad de los datos de la tarjeta de crédito y otras informaciones confidenciales. Podemos afirmar que las tecnologías para crear la tienda del futuro, para comprar online, para almacenar, procesar



y clasificar inmensas bases de datos, que nos permitan conocer y crear nuevas formas de comunicación con los consumidores, ya existen. El problema no es tecnológico, sino de cómo se resuelva el debate sobre la identificación de personas, del respeto a la privacidad y a la diferencia. [4]

#### **1.4 Objetivo General.**

Desarrollo de un prototipo de aplicación móvil para que eficiente el proceso de cotización de productos MRO (Mantenimiento, Reparación y Operaciones) entre los proveedores y el cliente en el estado de Chihuahua.

#### **1.5 Objetivos Específicos.**

- Diseñar interfaces útiles, intuitivas, sencillas y de fácil funcionamiento.
- Integrar tecnología Web Services en la aplicación.
- Utilizar un gestor de versiones en el desarrollo de la aplicación.
- Lograr obtener el costo de un producto mediante una cotización previamente solicitada.

#### **1.6 Hipótesis.**

Una aplicación móvil permite agilizar el proceso de compras en el sector industrial mediante la correcta administración de la información entre los usuarios participantes.



### **1.7 Justificación.**

Para una gran parte de las empresas, su forma de comercialización será el denominado comercio multicanal, que consiste según el consejero delegado de eBay: En la posibilidad de poder comprar y vender en cualquier momento y en cualquier lugar, bien a través de un ordenador, un teléfono, una tableta, una televisión, una tienda o un quiosco. Esta será la forma en que los consumidores conectarán con las cosas que necesiten o que les gusten. No habrá diferencias entre hacerlo online ni offline [4].

Algunos autores, como Deza Pulido (2007), ya hablaban hace años de la nueva comunicación en un entorno digital como alternativa a la comunicación unidireccional clásica. La publicidad en Internet y especialmente a través del móvil, definirá lo que se denomina el siglo del mobile marketing, donde el segmento de consumidores nómada tendrá un crecimiento espectacular y se dispondrá de móviles mind-reading, con capacidad para aprender y adaptarse a las preferencias y necesidades de los usuarios[4].

Las bases de datos en línea, permitirán una selección de productos automática y con mejor información. Se evitará manejar y guardar físicamente catálogos, que serán sustituidos por lo que podríamos llamar catálogos virtuales.

Los productos defectuosos o de mala calidad se darán a conocer rápidamente entre todos los usuarios de la red y compradores potenciales, acortando así a días el ciclo vital de los productos inferiores.



El comercio electrónico es una importante apuesta de futuro, pero sobre todo es un valioso canal para incrementar las ventas y reducir los costes estructurales inherentes al comercio tradicional.



## Capítulo 2

### Marco Teórico.

#### Web Services

Es una interfaz, accesible por protocolos (estándar o no) usados en internet, que permite acceder a las funcionalidades de un objeto concreto, sin importar las tecnologías ni plataformas implicadas en la petición.

Es una parte lógica de negocio, capaz de procesar y accesible desde cualquier lugar, por cualquier persona, a través de cualquier medio.

Si bien es cierto que los web services pueden ser accedidos usando múltiples protocolos, quizá sea el HTTP el más usado, por la facilidad de implementación, por ser interactivo y por el hecho de estar ampliamente extendido. [5]

#### PHP

Es un acrónimo recursivo para Hypertext Preprocessor, originalmente Personal Home Page, es un lenguaje interpretado libre, usado originalmente solamente para el desarrollo de aplicaciones presentes y que actúan en el lado del servidor, capaces de generar contenido dinámico en la World Wide Web.

El código es interpretado en el lado del servidor por el módulo PHP, que también genera la página web para ser visualizada en el lado del cliente. [6]





### **Metodo POST**

Es una variable súper global de PHP que se utiliza para recopilar datos de formulario después de enviar un formulario HTML con `method = "post"`. `$_POST` también se usa ampliamente para pasar variables.

### **Metodo GET**

Es una variable súper global de PHP que se utiliza para recopilar datos de formulario después de enviar un formulario HTML con `method = "get"`. `$_GET` también puede recopilar datos enviados en la URL.

### **MySql**

Es una base de datos relacional que utiliza el lenguaje SQL. Se trata de un SDB de código abierto. [7]

### **Procedimientos Almacenados**

La interacción con la base de datos es mediante los procedimientos almacenados. Los procedimientos almacenados son un conjunto de instrucciones SQL que trabajan como una unidad y que se ejecutan utilizando solo el nombre que le hemos asignado. El uso de procedimientos almacenados tiene las ventajas siguientes:

- Se ejecutan en el servidor.
- Las instrucciones se ejecutan más rápido.
- Ayudan en la programación orientada a objetos, cuando se trabaja con las capas en general ya sea con aplicaciones windows, o web [12].



## **Android Studio**

Es la plataforma de desarrollo de Android, siendo una plataforma muy completa, teniendo una estructura excelente, una visualización muy buena, usa plantillas, y es una manera fácil de crear aplicaciones. Tenemos el inconveniente de que no es multiplataforma, ya que solo lo soporta Android.

## **JSON**

Es un formato ligero basado en de texto, con definición de lenguaje completamente independiente, cuya sintaxis está basada en un subconjunto de funcionalidades y palabras reservadas del lenguaje de script Web: JavaScript. Debido a su sencillez es fácil generar y procesar un documento con este formato.

JSON usa las convenciones para el manejo de datos, que son comunes a la familia de lenguajes: C, C++, C#, Java, Perl, Python, etc; esta característica hace de JSON un lenguaje ideal para el intercambio de datos entre aplicaciones cliente servidor o aplicaciones que se ejecuten en diferentes escenarios. Un mensaje JSON está constituido por dos estructuras básicas: la primera es una colección de parejas o estructuras nombre-valor, las cuales en varios lenguajes son conocidas como: registros, diccionarios, tablas, hash, listas de claves o arreglos asociativos; y la segunda es una lista ordenada de valores; en la mayoría de los lenguajes, esta se suele representar en forma de: arreglos, vectores, listas, etc.

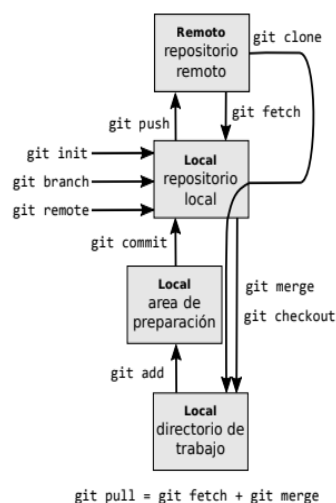
Las anteriores estructuras son usadas para conformar los mensajes, en los cuales se define un protocolo interno con la representación: palabra clave - valor o un conjunto de valores



ordenado con significado; estos mensajes son interpretados por el cliente y servidor y decodificados de acuerdo al protocolo interno definido, para lo cual se requiere el uso de librerías que permitan interpretar la sintaxis propia del lenguaje JSON. [10]

## GIT

Git es un SCV distribuido diseñado para la gestión eficiente de flujos de trabajo distribuido no lineales. Git fue diseñado y desarrollado inicialmente por Linus Torvalds en 2005 para el desarrollo del kernel de Linux. La licencia de Git es libre y hay distribuciones oficiales para los sistemas operativos Mac OS X, Windows, Linux y Solaris. La distribución de Git incluye herramientas de línea de comando y de escritorio. Además, hay disponibles herramientas proporcionadas por terceros que permiten una mayor integración con el escritorio o con entornos de desarrollo [11]



*Figura 1 Principales comandos de Git.*



## **GITHUB**

GitHub es un servicio comercial de alojamiento de repositorios Git remotos creado en el año 2008. GitHub proporciona una interfaz Web que permite al usuario registrado crear repositorios vacíos o por clonación de otro repositorio hospedado en GitHub (fork en la terminología de GitHub), enviar solicitudes de cambio entre repositorios hospedados (pull request en la terminología de GitHub), y gestionar dichas solicitudes. Además, los repositorios hospedados en GitHub pueden actuar como repositorios remotos de repositorios locales. Los repositorios que se crean en GitHub son por defecto de acceso público.

Además del alojamiento, GitHub proporciona a cada repositorio una wiki, un gestor de tareas (issues), un completo sistema de gestión de comentarios, un cuadro de control con grafos sociales e, incluso, una página web propia. Finalmente, todo este contenido puede ser accedido y manipulado mediante un API Web2 [11].

### **Programación en N capas**

El estilo arquitectural en n capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan. Cuanto más se aumenta el proceso operativo de la empresa, las necesidades de proceso crecen hasta desbordar las máquinas. Es por ello que se separa la estructura de un programa en varias capas [12].



Actualmente, la más utilizada en cuanto a este estilo es la programación en tres capas que se divide en: 1) Capa de presentación, la cual interactúa con el usuario; 2) La capa de negocios, donde se establecen las reglas del negocio que deben cumplirse y; 3) La capa de datos en la cual se define la conexión con el servidor y la base de datos, es en esta capa en donde se invoca a los procedimientos almacenados o comandos SQL, a fin de que se realicen las operaciones correspondientes en la base de datos; sin embargo, si es conveniente podemos dividir la estructura del código fuente en más capas [12].

### **Autenticación**

Es la propiedad que controla el acceso de los usuarios a un sistema y del mismo modo también controla los recursos e información a los que un usuario tiene acceso. En un sistema informático esto se consigue mediante un control de acceso con usuarios y contraseñas.

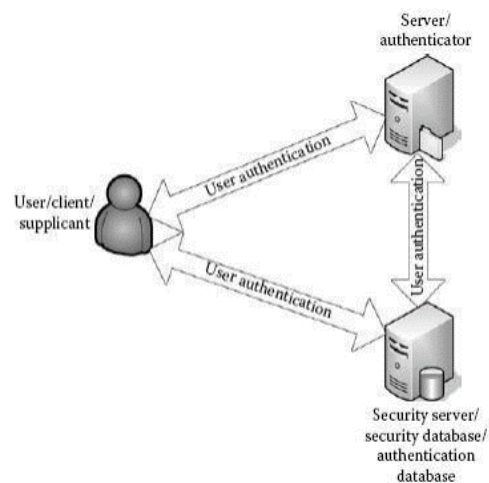
La autenticación de un usuario involucra 3 actores:

- Solicitante: es el propio usuario, el cual proporciona la información necesaria para demostrar que es quien dice ser. Esta información se emplea a modo de método de identificación y, por ejemplo, puede ser un nombre o un número identificativo. Estudio de mecanismos de autenticación basados en contraseñas visuales en dispositivos móviles Android 8.



- Autenticador: es la entidad encargada de verificar que la información proporcionada por el solicitante es correcta. Dicho de otro modo, el autenticador está asociado al método o conjunto de métodos mediante los cuales el usuario prueba que es quien dice ser.[15]

- Autoridad de seguridad: está encargada de custodiar y verificar las credenciales de autenticación. Esto es, la autoridad de seguridad se encarga de controlar el acceso del solicitante a los recursos mediante la validación de la información aportada tanto por el identificador como por el autenticador.[13]



*Figura 2 Actores involucrados en la autenticación*

### **Nombre de Marca**

Según Bennett (1988) el BN es uno de los componentes más importantes de la marca y se podría definir como el elemento que es verbalizado por los consumidores. Cualquier nombre de marca puede ser descriptivo, sugerente y/o distintivo, y puede construir

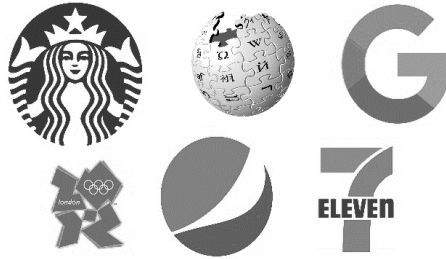


asociaciones con el producto y/o empresa. De acuerdo con el modelo de memoria basado en redes asociativas (Anderson, 1993) se ha sugerido que BN es un puente entre lo que la empresa quiere que se entienda y las asociaciones que han construido los individuos en su memoria (Urde, 1999). De esta manera, la marca se puede definir como palabra(s), número(s) y/u otro símbolo(s) vinculados a las percepciones, expectativas y sentimientos acerca de un producto, que se reflejan en las asociaciones que los consumidores construyen en su cerebro.[15]

### **Logotipo**

El término 'logotipo' o 'logo' se refiere nombres propios, completos o abreviados, y, por consiguiente, a letras, cifras y palabras escritas con cierta clase de fuente, tipografía o caligrafía, sobre todo, estilizada o con algún diseño que las destaque. Así podemos decir: "logotipo = nombre + diseño".

Desde el punto de vista etimológico el término 'logotipo' proviene del griego logos (λόγος), que significa, en este caso, palabra y typos, golpe, marca o señal, en forma de impronta (Ortega Pedraza, 1980). Así que un logotipo es un identificador visual o gráfico (o un signo distintivo) formado (únicamente) por letras, cifras o palabras.[16]



*Figura 3 Ejemplos de logotipos*

### **Isotipo Figurativo**

Son aquellos en los que no existe una relación evidente con algún aspecto de la realidad visible, pues lo que se trata de sugerir o provocar con ellos son ideas, conceptos, valores o emociones.

## **Capítulo 3**

### **Metodología.**

#### **3.1 Diseño Lógico.**

El diseño de un sistema de información produce los elementos que establecen cómo el sistema cumplirá los requerimientos identificados durante el análisis del sistema. Es un proceso que convierte el modelo conceptual en un modelo más técnico.

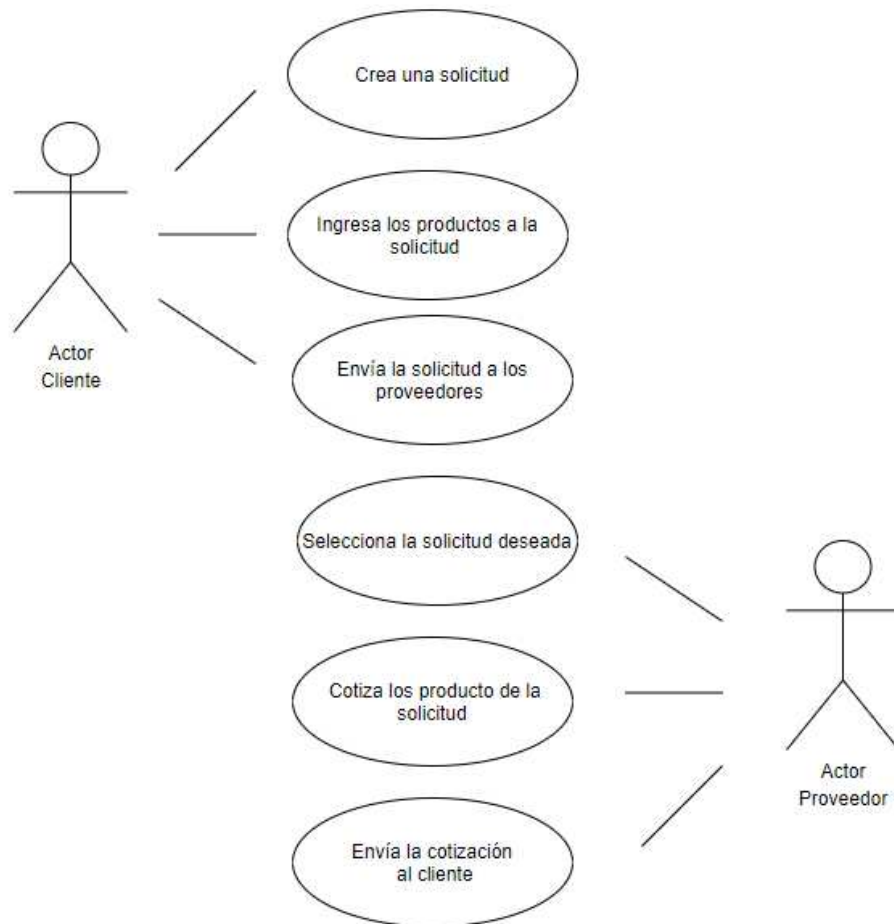
El primer paso en el diseño de sistemas es identificar los informes y las salidas que el sistema producirá. Partiendo que se tendrán dos roles de usuarios (cliente- proveedor) se contempla que cada uno de ellos pueda administrar de formas diferentes la información para así lograr una correcta comunicación entre cada uno de los actores.





El funcionamiento de la aplicación se enfoca en que el cliente obtenga el costo de algún artículo partiendo de una solicitud previa hacia el mercado de proveedores que ofertan.

La actividad de los proveedores se resume en que puedan dar el costo más exacto hacia un artículo deseado por el cliente, así como el tiempo de entrega y condiciones comerciales de venta. La aplicación contempla presentar la información más completa al cliente en forma de cotización.



*Figura 4 Diagrama de caso de uso*



### **3.2 Diseño de la Base de datos.**

Uno de los aportes que ha traído el desarrollo de los campos de informática y electrónica ha sido el concepto de Base de Datos (BD) con formatos electrónicos, por cuanto aporta a la planificación y organización (3) que se resumen como colecciones de información (datos) que se relacionan para crear un sentido y dar más eficiencia a la estructura de datos de una empresa. (7)

El sistema se gestiona en un sistema de base de datos (SBD) MySQL V10.2 alojada en la nube con el nombre u901308218\_CotApp y se basa en el modelo relacional en la cual se contemplan doce tablas normalizadas lo que va a permitir una armonización coordinada de los datos.

#### **3.2.1 TB01\_Usuario.**

La tabla uno se enfoca al almacenamiento de los datos del usuario en la cual se guardan los datos como Nombre, apellido, teléfono y correo del usuario, campos que permitirán identificar al usuario al cual se dirigirán las cotizaciones del sistema o el remitente que las envía.

En tanto el correo y la contraseña también almacenados en esta tabla contemplaran el acceso al sistema. Se relaciona el usuario con la tabla empresa para identificar de donde se está realizando la solicitud.



El usuario tiene un Rol (proveedor o cliente) así como un estatus activo o desactivado, este para saber cuáles usuarios se encuentran bloqueados por ingresar mal su contraseña tres veces.

Column Name	Data Type
TB01_id_Usuario	int(5)
TB01_Nombre_Usuario	varchar(100)
TB01_Apellido_Usuario	varchar(100)
TB01_Telefono_Usuario	int(10)
TB01_Correo_Usuario	varchar(150)
TB01_Password	varchar(100)
TB02_id_Rol	int(5)
TB03_id_Ciudad	int(5)
TB06_id_Estatus	varchar(5)
TB09_id_Empresa	int(5)
TB01_Fecha_Alta	date

*Figura 5 Tabla de Usuario*

### 3.2.2 TB02\_Rol.

Esta tabla se agrega ya que manejará los diferentes tipos de roles que se asignan a los usuarios registrados, al momento se cuenta con tres: Usuario, Proveedor y administrador.

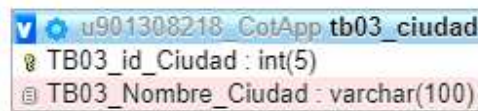
Column Name	Data Type
TB02_id_Rol	int(5)
TB02_Nombre_Rol	varchar(50)

*Figura 6 Rol del Usuario*



### 3.2.3 TB03\_Ciudad.

La tabla ciudad es una tabla que se encuentra previamente llena con los nombres de las 25 ciudades de chihuahua, las cuales se relacionan con la tabla de usuario y de empresa, ya que es necesario conocer el lugar de donde se trabaja con la aplicación para la solicitud de emoción y recepción de cotizaciones.



*Figura 7 Tabla de Usuario*

### 3.2.4 TB04\_Requerimiento.

Una cotización así como una solicitud se componen de diferentes conceptos, es en esta tabla donde se guardarán esos diversos conceptos, podríamos decir que a esta tabla se le conoce como cuerpo mientras que su cabecera se encuentra en la tabla solicitud; En esta tabla se guardan los datos generales como lo son nombre del producto a solicitar, una descripción, cantidad, unidad también se manejan datos más específicos como lo son marca del producto, modelo, número de serie siendo estos últimos no necesarios para ser guardados ya que al ser datos específicos es probable que el cliente no cuente con ello, a su vez todos los datos se relacionan a la tabla ocho.



u901308218_CotApp tb04_requerimiento	
PK	TB04_id_Requerimiento : int(5)
	TB04_Nombre_Producto : varchar(100)
	TB04_Descripcion_Producto : varchar(250)
	TB04_Marca_Producto : varchar(100)
	TB04_Modelo_Producto : varchar(100)
	TB04_Numero_Serie : varchar(150)
	# TB04_Cantidad_Producto : int(5)
	# TB08_id_Solicitud : int(5)
	# TB05_id_Unidad : int(5)

*Figura 8 Tabla de Requerimiento*

### 3.2.5 TB05\_Unidad.

La tabla unidad se encuentra previamente llena con los tipos de unidades que se manejan un ejemplo de estos conceptos son piezas.

u901308218_CotApp tb05_unidad	
PK	TB05_id_Unidad : int(5)
	TB05_Nombre_Corto_Unidad : varchar(20)
	TB05_Nombre_Largo_Unidad : varchar(50)

*Figura 9 Tabla de Unidad*

### 3.2.6 TB06\_Estatus.

Esta tabla almacenara los diferentes estatus que se manejan en las tablas, como lo son abrir, cerrar, activo o desactivado.

u901308218_CotApp tb06_estatus	
PK	TB06_id_Estatus : int(5)
	TB06_Nombre_Estatus : varchar(50)

*Figura 10 Tabla de Estatus*



### 3.2.7 TB07\_Cotizacion.

Esta tabla es la unión de todo por lo cual se realiza el sistema, ya que la finalidad del sistema es lograr obtener el costo de un producto mediante una cotización previamente solicitada. Por lo que se relaciona con diversas tablas como lo son solicitud, usuario, estatus. La tabla se normaliza de manera que todos los datos del producto se encuentran en la tabla requerimiento que a su vez se relaciona con la tabla solicitud, esto permite que la cotización pueda relacionarse con cada uno de los productos y se pueda asignar el costo a estos.

Column Name	Data Type
TB07_id_Cotizacion	int(5)
TB08_id_Solicitud	int(5)
TB01_id_Usuario	int(5)
TB06_id_Estatus	int(5)
TB07_Terminos_Venta	int(5)
TB07_Fecha_Fin_Cotizacion	date
TB07_Fecha_Cotizacion	date

*Figura 11 Tabla de Cotización*

### 3.2.8 TB08\_Solicitud.

La solicitud es de donde parte el todo el sistema por lo que se agrega la tabla de solicitud, esta tabla es el cabecero ya que el cuerpo se encuentra en la tabla cuatro, esta tabla guarda datos como la fecha de solicitud, fecha de cuando finaliza la solicitud y de igual forma está relacionada con la tabla uno y seis.



Column Name	Data Type
TB08_id_Solicitud	int(5)
TB01_id_Usuario	int(5)
TB06_id_Estatus	int(5)
TB13_id_Categoria	int(5)
TB11_id_Tipo_Venta	int(5)
TB08_Nombre_Solicitud	varchar(250)
TB08_Numero_Cotizacion	int(5)
TB08_Fecha_Solicitud	date
TB08_Fecha_Fin_Solicitud	date

*Figura 12 Tabla de Solicitud*

### **3.2.9 TB09\_Empresa.**

Esta tabla como su nombre lo indica guardara los datos de la empresa, como lo son ciudad, RFC, razón social esto con el fin de saber a qué empresa está dirigida la cotización.

Column Name	Data Type
TB09_id_Empresa	int(5)
TB09_Nombre_Empresa	varchar(150)

*Figura 13 Tabla de Empresa*

### **3.2.10 TB10\_Concepto.**

La tabla concepto, es el cuerpo de la tabla cotización ya que aquí se conservan los conceptos cotizados por la empresa.





Column Name	Data Type
TB10_id_Concepto	int(5)
TB07_id_Cotizacion	int(5)
TB05_id_Unidad	int(5)
TB12_id_Moneda	int(5)
TB10_Nombre_Producto	varchar(150)
TB10_Descripcion_Producto	varchar(250)
TB10_Costo_Producto	decimal(11,2)
TB10_Fecha_Concepto	date

*Figura 14 Tabla de Concepto*

### 3.2.11 TB11\_Tipo\_Venta.

Se crea esta tabla para conocer qué tipo de venta se va a realizar ya sea a crédito o de contado. Esta tabla esta precargada.

Column Name	Data Type
TB11_id_Tipo_Venta	int(5)
TB11_Nombre_Tipo_Venta	varchar(50)

*Figura 15 Tabla Tipo de venta*

### 3.2.12 TB12\_Moneda.

Debido a los diferentes costos que se manejan, la volatilidad del peso mexicano, así como la cercanía con la frontera existen proveedores que sus costos están manejados solamente en dólares por lo que se genera una tabla para cargada con los tipos de monedas MXN y USD para que puedan elegir en qué tipo de moneda se va a cotizar.

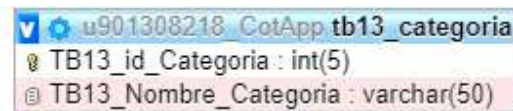
Column Name	Data Type
TB12_id_Moneda	int(5)
TB12_Nombre_Corto_Moneda	varchar(5)
TB12_Nombre_Largo_Moneda	varchar(20)

*Figura 16 Tabla de moneda*



### 3.2.13 TB13\_Categoría.

Se identifica de esta manera ya que es la que contiene los diferentes ramos de compra venta que pueden existir, un ejemplo puede ser Metales, papelería, construcción entre otros.



*Figura 17 Tabla categoría*

### 3.2.14 Procedimientos Almacenados.

#### *Nuevos Usuarios.*

Se encarga de insertar en la base de datos a nuevos usuarios; Dentro de este se realiza una validación mediante el campo correo, ya que este campo se utiliza como valor de usuario a la hora de realizar el acceso en la aplicación por lo que si el correo se encuentra en la base de datos ya no es posible dar de alta a este nuevo usuario y regresa un mensaje a la aplicación “El usuario ya existe” de lo contrario el usuario se da de alta en la tabla TB01\_Usuario.

#### *Login.*

Este procedimiento se crea para poder acceder al sistema, se encarga de realizar la selección de los datos del usuario mediante las credenciales ingresadas previamente en la aplicación, si estos no se encuentran regresa un mensaje “El usuario no se encuentra”



### ***Busca Rol.***

Se encarga de buscar los datos que hay en la tabla TB02\_Rol con el fin de llenar el spinner del Activity Registro de Usuario.

### ***Busca Cabecera de solicitud.***

Se realiza la búsqueda del nombre de la solicitud y de su número de control para ser utilizado posteriormente en la búsqueda de los detalles de la solicitud.

### ***Busca categoría.***

Realiza la búsqueda de las diferentes categorías de venta que existen, para llenar los spinner que servirán como filtros para el llenado o filtros de pantallas en la aplicación.

### ***Busca Ciudad.***

Permite llenar los spinner que mostraran las diferentes ciudades que tiene chihuahua y que permitirá registrar la ciudad del usuario así como poder filtrar las solicitudes por ciudad.

### ***Busca Crédito.***

Realiza la búsqueda de los diferentes días de pago que puede ofrecer el cliente, ya que en la empresa es común que las empresas realizan su pago a 45 60 o 90 días según la política de cada uno.

### ***Busca Unidad.***

Los productos se manejan con diferentes tipos de medidas desde un metro de cable de uso rudo para energía hasta unas cuantos cientos de pizas de tornillos por lo cual la aplicación tiene que saber unidades bajo las cuales el cliente le interesa comprar o las cuales el proveedor puede vender ejemplo el cliente quiere piezas pero el proveedor vende cajas.



### ***Cerrar Detalle.***

El manejo de estatus permite conocer cuál es la situación en la cual se encuentra una solicitud, al ingresar la solicitud se le asigna el estatus abierto. Una vez que el cliente termina de ingresar los productos que se requieren comprar este procedimiento cambia el estatus a finalizado para que pueda ser cotizado por los proveedores.

### ***Nueva Solicitud.***

Este procedimiento se encarga de realizar la inserción de los datos de la solicitud a la tabla tb08\_solicitud.

### ***Nuevos Requerimientos.***

Este procedimiento nos permite agregar el detalle de la solicitud, los productos que se necesitan comprar.

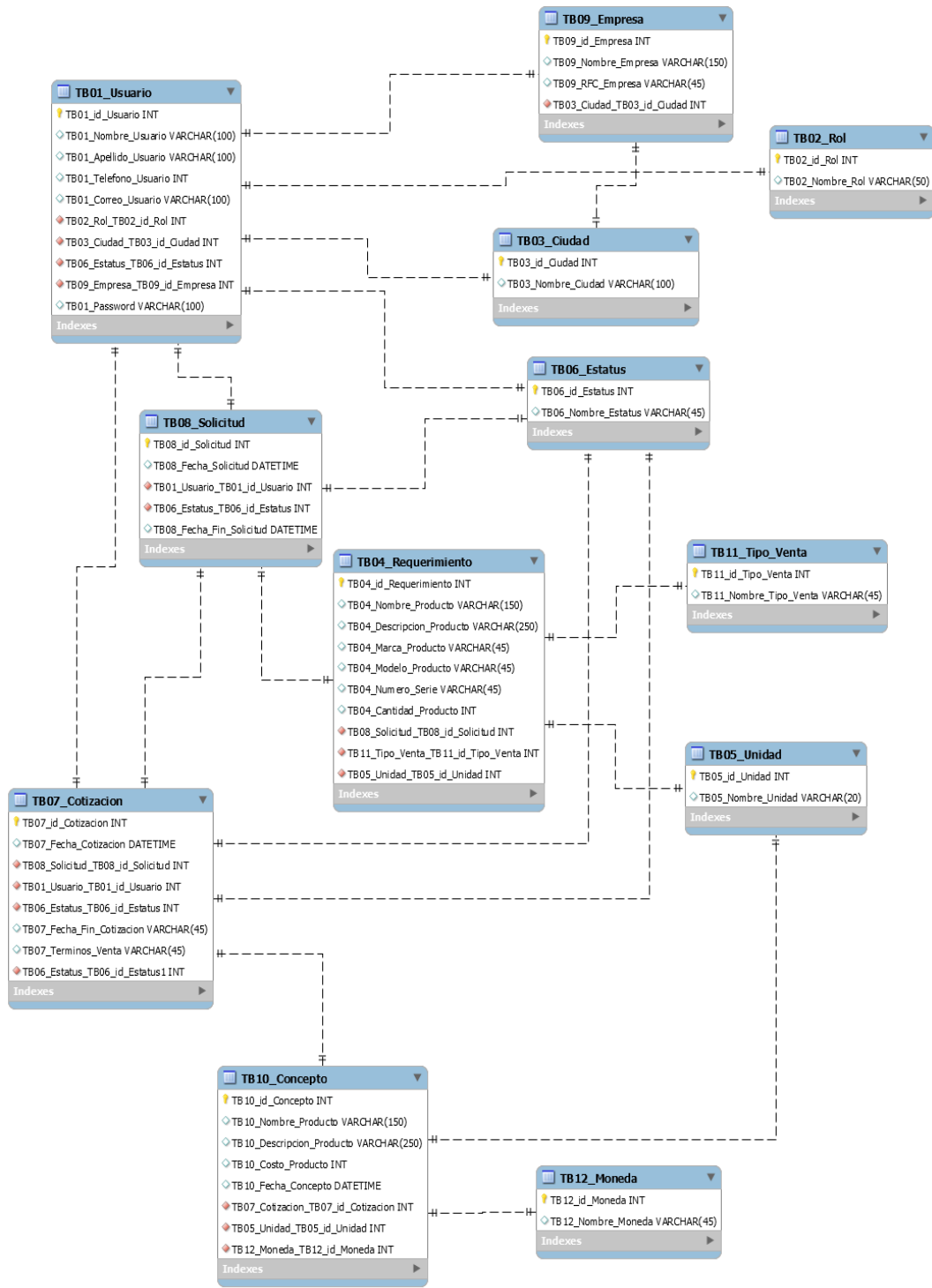
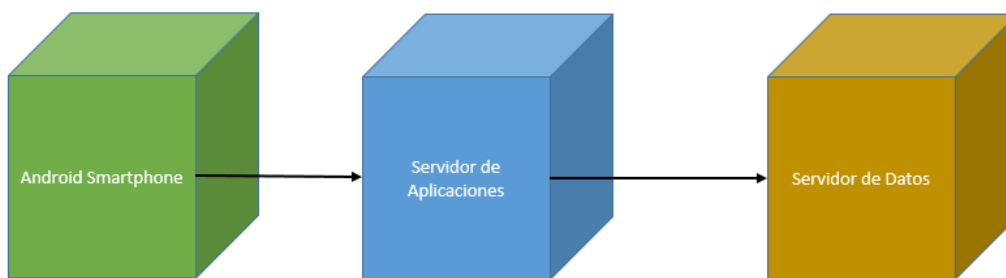


Figura 18 Diagrama de Base de Datos u901308218\_CotApp

### 3.3 Desarrollo.

#### 3.3.1 Diseño general de la aplicación.

La aplicación está diseñada en un estilo de tres capas: presentación, negocios y de datos; cual permite separar el código fuente según el rol, responsabilidad y funcionalidad. [12]



*Figura 19 Diagrama de Capas*

El flujo general de la aplicación inicia cuando el cliente tiene una nueva solicitud, con esto nos referimos a que el cliente requiere de un martillo, desarmador, herramienta o refacción para algún equipo.

Una vez que el cliente guardo la solicitud en la aplicación se hace pública esta solicitud, por lo que aparecerá en un listado en la pantalla del proveedor para que este pueda generar una cotización, esta se creara automáticamente, el proveedor solo deberá colocar el costo del producto que se solicita.

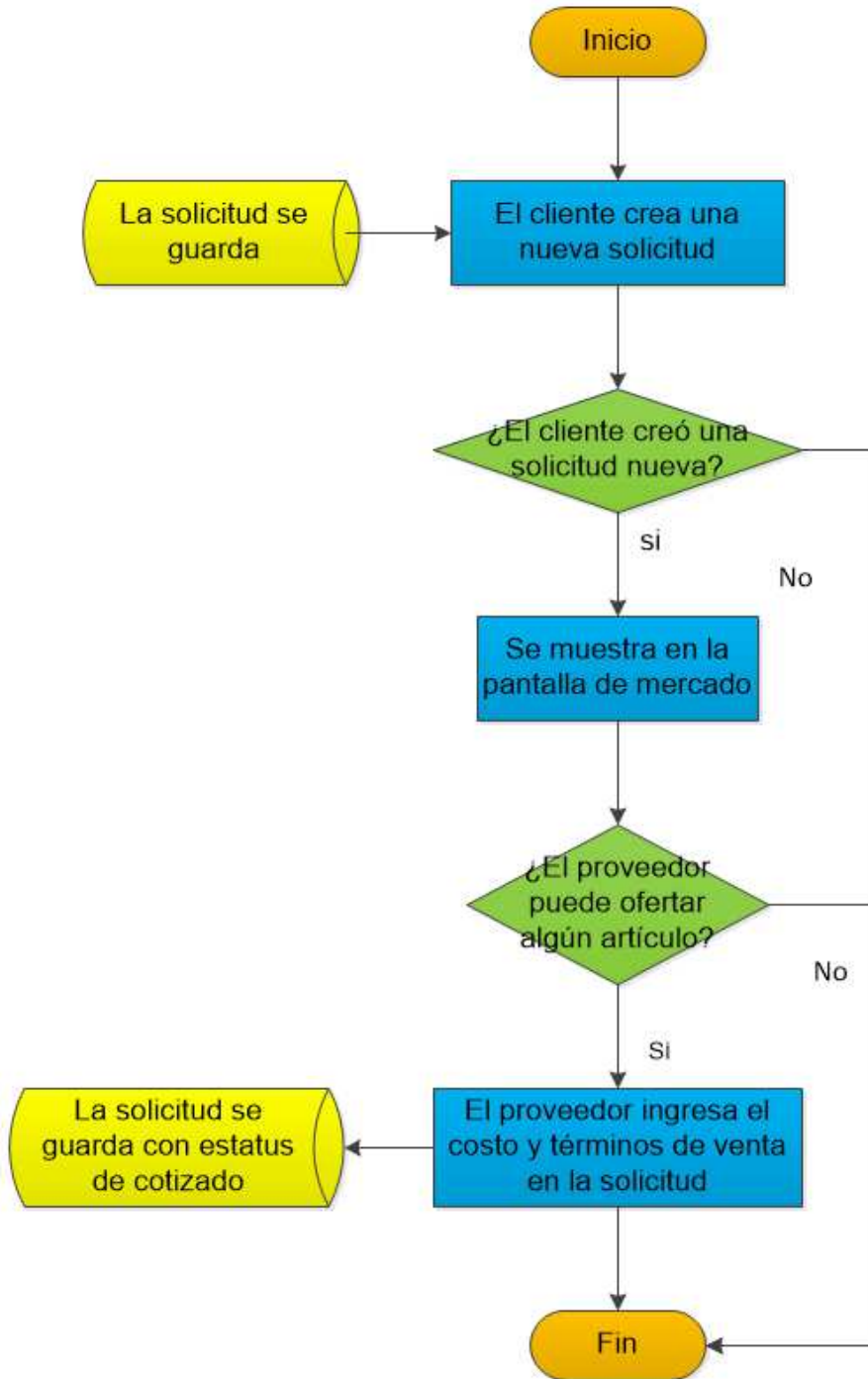


Figura 20 Diagrama de Flujo del Proceso General de la aplicación



### 3.3.2 Diagrama jerárquico

El diagrama jerárquico identifica las partes compuestas por la aplicación móvil en la cual se pueden percibir nueve diferentes pantallas que permite administrar conceptualmente la información que se planea administrar.

Este diagrama se divide en dos roles, el rol del cliente, el cual está compuesto por tres pantallas (2, 3,4) que actúan como pantalla de inicio (pantalla 2) la cual presenta un botón para crear una nueva solicitud (pantalla 3) en la cual se solicitan datos que van a permitir englobar los productos que se ingresaran en la siguiente pantalla (pantalla 4) la cual permitirá guardar los datos con los que cuenta el cliente para solicitar la cotización.

El rol número dos se identifica como el proveedor, en este rol se permite el acceso a cuatro pantallas (5,6,7,8) que servirán para realizar la búsqueda de solicitudes y posterior poderle cotizar.

Una vez ingresamos que se ingresa al perfil del proveedor logramos realizar la búsqueda de solicitudes (pantalla 6) mediante la ciudad y la categoría de producto que se puede ofertar una vez ingresados estos datos se listan las solicitudes de las cuales se puede elegir una la cual detallara los productos que incluye la solicitud (pantalla7) la cual describe cada uno de los productos cantidades y detalles del producto para poder darnos una idea de lo que se requiere cotizar por lo que cada seleccionando el producto se nos despliega una pantalla para poder capturar un los detalles exactos del articulo cotizado.





Figura 21 Diagrama de Aplicación



### **3.3.3Capa de Presentación.**

Es la responsable de la presentación visual de la aplicación. La capa de presentación enviará mensajes a los objetos PHP de la capa de negocios, la cual responderá entonces directamente o mantendrá un diálogo con la capa de la base de datos, la cual proporcionará los datos que se mandarían como respuesta a la capa de presentación [12].

Para esta capa se presenta mediante Android 8.0 (Oreo) en la API 26, en la cual se manejan lenguajes como Java, JavaScript, JSON y HTML5.

#### ***Nombre de la aplicación.***

Las marcas y los nombres de marca juegan un papel importante en nuestra vida diaria, transmitiendo confianza y facilitando el sentimiento de pertenencia a un grupo. Se admite también que el nombre de marca es clave para identificar el producto, diferenciarlo de la competencia y crear asociaciones con la marca. Esto hace que el BN tenga una gran influencia en la toma de decisiones de compra.[15]

ASO es uno de los pilares más importantes en cualquier estrategia de App Marketing que influye a que se realicen mejores prácticas para optimizar la aplicación móvil. El nombre es una de las partes más importantes del App Store Optimization, por eso debemos prestar atención, ya que será lo primero que leerá el usuario, también ayuda a construir la marca.

Desde el punto de vista del consumidor, un nombre de marca puede proporcionar un significado simbólico que agiliza el proceso de toma de decisiones y puede ayudar a simplificar la forma en que los consumidores toman las decisiones y realizan las compras.[15]



Debe ser un nombre único, fácil de recordar, preferiblemente corto, fácil de recordar, que se refiera a la marca o a la función principal de la app, pero que al mismo tiempo destaque de la competencia. Se toma en cuenta cómo quedará el nombre de la App en la pantalla del teléfono, que es como lo verán tus usuarios.

En este caso después del nombre de la App se colocan unas palabras clave que describen la funcionalidad de ésta.

Para el desarrollo del nombre de la aplicación se toma como referencia la palabra “Cotización” que proviene del latín *quotus*, que significa cuánto, que en el sector empresarial hace referencia a un documento informativo que el departamento de compras de una empresa o institución utiliza para entablar una negociación.

Ya que la palabra es bien conocida por el cliente y por el proveedor se toma como referencia para crear el nombre de la aplicación que se recorta para que así sea fácil de recordar y se le bautiza como “COT”.

### ***Diseño del Logotipo.***

Lo más importante es crear un icono de app original al cien por cien. Algunas recomendaciones básicas son utilizar una única forma o símbolo en el diseño de apps y sobretodo hacerlo lo más simple y reconocible posible. Una técnica para que destaque sobre las demás apps de la store es usar colores vibrantes.

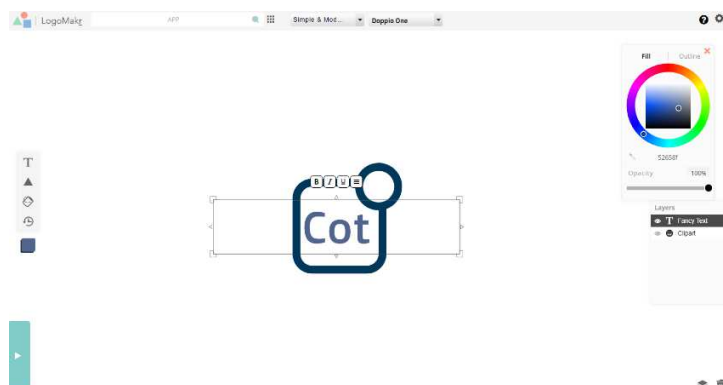


El diseño se basa en un isotipos abstractos que hace referencia a un concepto estadístico. El cuadro grande en el logotipo referencia al universo, el cual en palabras más simples sería el universo de proveedores que están dentro de la app, posterior se distingue un círculo que hace referencia al concepto de conjunto para hacer referencia que se puede encontrar el conjunto de proveedores que se busca.



*Figura 22 Diseño de Logotipo*

El diseño se basó en los colores que se manejan en el diseño del logotipo son tres, Vermillion, Fresh , Prussian Blue, mientras que la letras que aparecen dentro del cuadro son Doppio One, un estilo de letra que hace parecer al logotipo como si fuera una tecla de PC.



*Figura 23 Diseñador de logotipo*



### ***Diseño de la interfaz gráfica.***

El diseño de la interfaz gráfica se basa en la app de una cadena de café, ya que muestra una simplicidad con el uso de sus botones, una ayuda visual sencilla con respecto a los colores por lo que no es compleja su uso.



*Figura 24 Diseño base para el desarrollo de la interfaz grafica*

### ***Colores de la interfaz.***

El color es uno de los factores más determinantes en la primera impresión que tienen los clientes de tu negocio. Es clave para la identidad de tu marca y es, de hecho, mucho más que una simple ayuda visual, ya que el color transmite emociones, sentimientos y experiencias. En definitiva, el 93% de los consumidores se ven influenciados por los colores y la apariencia visual.



Los tonos ideales para la industria son el azul ya que el Color se relaciona con la limpieza, la tecnología, ideal para productos relacionados con la informática y es Ideal para ejecutivos, relacionado con servicios financieros, inversiones, banca, grandes empresas.

El diseño de la aplicación está basada en tres colores Vermillion, Fresh y Prussian Blue colores recomendados por la página [canva.com](https://www.canva.com) y los conceptualiza como colores corporativo y elegantes en el cual indican que el color distintivo azul marino intenso se combina a la perfección con el fondo blanquecino semimate para crear un diseño simple y estilizado.



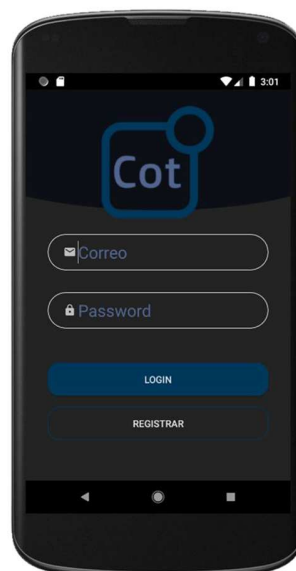
*Figura 25 Colores Base de la aplicación*



### ***Pantalla Login.***

Todo sistema de seguridad de la información debe cumplir con una serie de requisitos mínimos de confidencialidad, integridad y disponibilidad para que pueda considerarse realmente seguro y fiable [13].

Por lo que se diseña la pantalla de autenticación, esta se realiza colocando el correo y el password del usuario previamente registrado, estos datos se manejan internamente en el archivo mainactivity, java el cual valida que ninguno de los dos campos solicitados estén vacíos, en caso de que este un campo vacío se mostrara un mensaje "No se permiten campos vacíos". Si ambos campos están llenos se envían los campos al servidor en cual si la sentencia se cumple se permite el acceso al usuario, si no se encuentra el usuario se mostrara el mensaje "usuario o contraseña incorrectos"



*Figura 26 Pantalla Login*

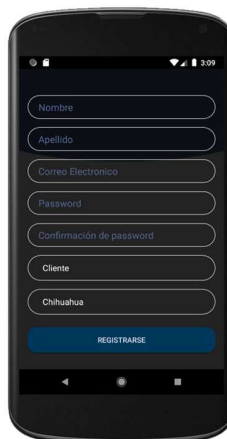


### ***Pantalla Registro de Usuarios.***

Aun cuando la tabla de usuario tiene once campos por llenas no son necesarios todos ya que lo que se busca con esta pantalla es un simple registro de usuario, sencillo y nada complicado por lo que se decide tomar como base el nombre, el apellido, el correo electrónico que sirve como identificador para que no se registre el mismo usuario dos veces y por supuesto su contraseña

El ultimo campo es un spinner que permite seleccionar el tipo de Rol que será el usuario, este último campo es muy importante ya que definirá cual será la pantalla de inicio del usuario registrado, este es definido si es cliente o proveedor.

Para lograr el registro se envían los datos mediante el método POST que posterior son recibidos en la capa de negocio, esto se puede observar en el anexo de registro de usuario de este documento.



*Figura 27 Pantalla Registro de usuarios*





### ***Pantalla Inicio de Usuarios.***

La pantalla de inicio muestra una interfaz sencilla con cuatro botones que permiten el acceso a diferentes pantallas, una de las más relevantes es la de “nueva solicitud “en la cual como su nombre lo indica se permite al usuario realizar una nueva solicitud para que posterior el proveedor pueda examinarla y cotizarla.

Después aparecen dos botones más que permitirán ver el historial de solicitudes y cotizaciones que se han realizado. Por último se nos muestra el botón de datos, que este botón permite que se muestren los datos del usuario, también podrá modificarlos.



*Figura 28 Pantalla Inicio de Usuario.*



### ***Pantalla Nueva Solicitud.***

Para crear una solicitud se requiere colocar el nombre del proyecto con el cual se va a trabajar y el tipo de material que se va a solicitar, además de esto se agrega la fecha fin de envío de cotizaciones, en otro campo se solicita el número de cotizaciones que se van a requerir para así no saturar de información al cliente y lograr que el proveedor envíe su cotización tan pronto como sea posible, por último se ingresan los términos de pago por ya que es muy común que las empresas paguen posterior a la compra.



*Figura 29 Pantalla Nueva solicitud.*



### ***Pantalla Agregar Productos.***

En esta pantalla se guarda la información del detalle de la solicitud, la cual permitirá capturar n número de detalles en una solicitud, guardamos el nombre, la descripción. Número de serie, modelo, cantidad y unidad.

Muestra el botón agregar que permite que se agreguen más datos a la solicitud abierta y el botón finalizar que cerrara la solicitud para mostrarla al proveedor.



*Figura 30 Pantalla Agregar productos*



### ***Pantalla Inicio de Proveedor.***

Esta es la primera pantalla que el usuario proveedor puede ver al ingresar al sistema y cuenta con cuatro botones, el primero le permite realizar la búsqueda de solicitudes realizadas por los clientes.

El segundo botón permite ingresar a sus datos el cual permite actualizar los datos y cerrar la sesión. Mientras que el tercero y cuarto se pretende en una segunda fase del sistema poder mostrar un reporte de las cotizaciones enviadas a los clientes y así poder darle seguimiento.



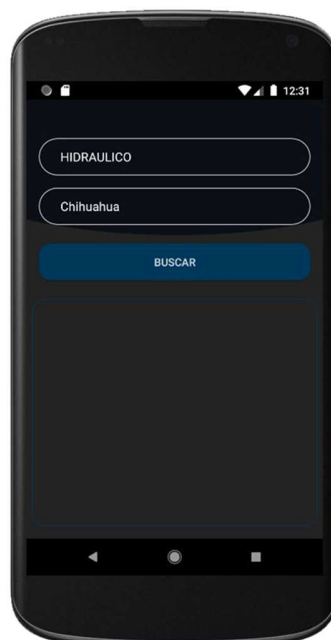
*Figura 31 Pantalla inicio proveedor*



### ***Pantalla Buscar Solicitudes.***

Esta pantalla permite al proveedor buscar las solicitudes generadas por el cliente mediante dos filtros, el primero sería la categoría del producto que le permite al proveedor elegir el tipo de mercado que desea ofertar, el segundo filtro es la ciudad que le permite buscar la ciudad en donde se requiere el producto.

El segundo filtro permitirá aumentar la competencia entre proveedores porque si un proveedor es de chihuahua capital y desea vender en la ciudad de Delicias solamente necesitara buscar que solicitudes que existen para esta ciudad y competir con los proveedores locales de Delicias lo que le permite al cliente tener una gama más amplia de proveedores.



*Figura 32 Pantalla Buscar solicitud.*



### ***Pantalla Buscar Productos.***

Para la búsqueda de productos se realiza una vez que se elige la solicitud, la aplicación carga en una lista de los productos que se requiere cotizar, donde se detallan el nombre, piezas unidades del producto así como la fecha límite para enviar la cotización.



*Figura 33 Pantalla Buscar Productos*



### ***Pantalla cotizar Producto.***

Esta pantalla ayuda al proveedor a manejar los datos de la cotización de una manera óptima ya que permite cotizar por producto.

El proveedor puede guardar de nuevo el nombre de producto, ya que muchas veces el cliente no tienen el nombre técnico del producto, un ejemplo es si el cliente solicita unas pinzas pero el proveedor identifica que lo que en realidad el cliente quiere es una pieza de pinzas eléctricas de corte diagonal.

Por otra parte se agrega la cantidad y las unidades ya que muchos productos se pueden vender en diferente tipo de unidad a la que el cliente necesita continuando con el ejemplo de las pinzas imaginemos que el cliente pide diez piezas pero el proveedor solo las vende por cajas de cinco este pondrá en su cotización dos cajas ya que como lo mencionamos cada caja contiene cinco piezas.

El proveedor puede colocar los términos de venta de cada producto ya que las políticas de venta del proveedor no podría coincidir con las políticas de compra del cliente por lo que el cliente tiene que estar informado los términos, de la misma manera el proveedor coloca la fecha final de la propuesta que realizo.



*Figura 34 Pantalla Cotizar Producto*

### **3.3.4Capa de Negocio.**

Es la responsable del procesamiento que tiene lugar en la aplicación, Contendrá objetos definidos por clases reutilizables que se pueden utilizar una y otra vez en otras aplicaciones.

Estos objetos se suelen llamar objetos de negocios y son los que contienen la gama normal de constructores, métodos para establecer y obtener variables, métodos que llevan a cabo cálculos. [11].

Es en esta capa donde se reciben los requerimientos del usuario y se envían las respuestas tras el proceso, a requerimiento de la capa de presentación. Se denomina capa de negocio o lógica del negocio, es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se escribe en PHP V7.4 en el servidor de aplicaciones hospedado en la nube.

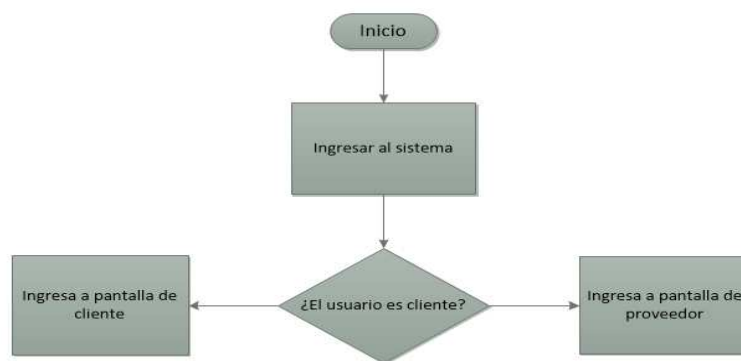




### *Login.php*

Se encarga de realizar la petición a la base de datos mediante las variables recibidas de la aplicación (usuario y contraseña), se realiza mediante la ejecución de un procedimiento almacenado. Este procedimiento almacenado nos regresa una lista de datos con los valores del usuario, los cuales podrán servir la aplicación.

Una vez que se tenga el acceso a la aplicación, el sistema definirá si el rol es de proveedor o de cliente, esto se realiza con los valores que regresa el archivo php.



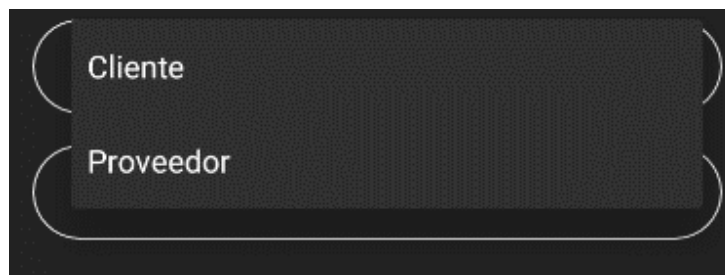
*Figura 35 Diagrama de acceso de usuario*



### ***Fill\_Spinner\_Rol.php***

Se realiza un llamado a la base de datos para regresar a la aplicación un arreglo mediante JSON que posterior se cargara en la aplicación y se muestran dos resultados el rol del cliente y el del proveedor.

Esta lista se carga cuando se registra el usuario para identificar posteriormente las actividades que va a realizar.



*Figura 36 Spinner de Roles*

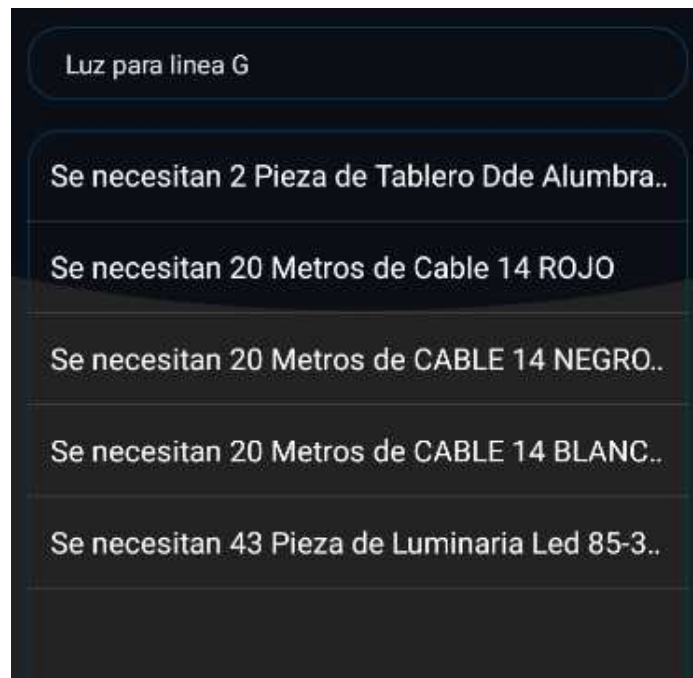
### ***Fill\_List\_Detail.php***

Este archivo se encarga de realizar un llamado a la base de datos que permite llenar una lista con las solicitudes activas, las cuales solicitan las variables de ciudad y categoría para poder ser filtradas.



### ***Fill\_List\_Product.php***

Esta pantalla se encarga de buscar todos los productos que se encuentran dentro de una solicitud, se genera a partir del llamado de un procedimiento almacenado en la base de datos el cual nos regresa dos variables una el id del producto el cual será utilizado posteriormente y la descripción que es la que se muestra al usuario.

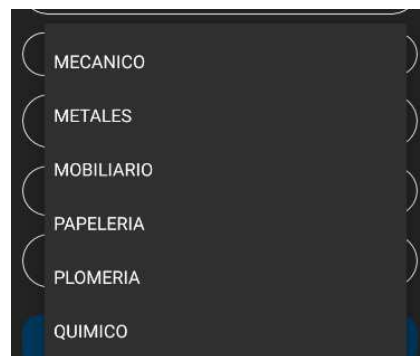


*Figura 37 Pantalla de productos.*



### ***Fill\_Spinner\_Categoria.php***

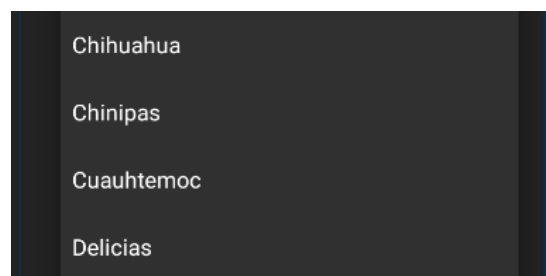
Dentro de la aplicación en diversas ocasiones se necesita del listado de la categoría de solicitudes por lo que se hace el llamado de este archivo php el cual nos regresara un listado de todas las categorías que existen en la base de datos.



*Figura 38 Spinner de categorias*

### ***Fill\_Spinner\_Ciudad.php***

Este archivo logra entregar un listado de las ciudades que se encuentran en la base de datos mediante el llamado de un procedimiento almacenado y el cual se guarda en un arreglo.

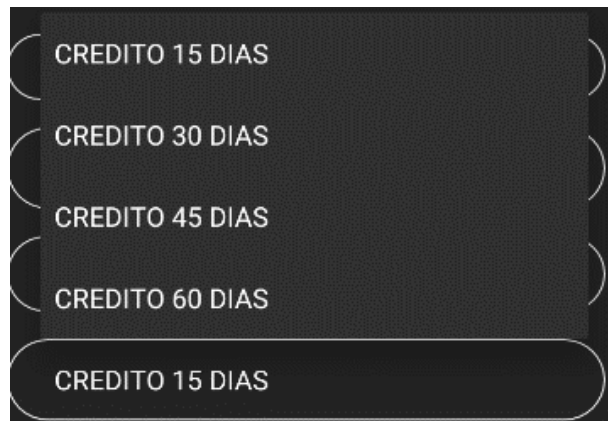


*Figura 39 Spinner de ciudad*



### ***Fill\_Spinner\_Credito.php***

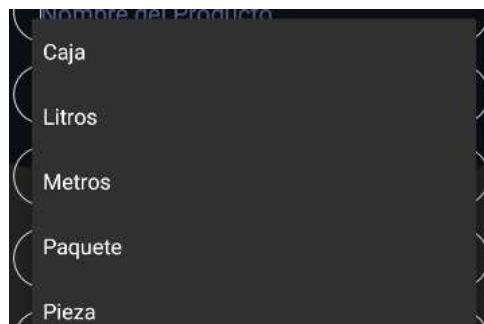
Este archivo se encarga de hacer un llamado a un procedimiento almacenado que regresa un listado de los créditos.



*Figura 40 Listado de credito*

### ***Fill\_Spinner\_Unidad.php***

Este archivo nos permite obtener en un arreglo el listado de unidades en donde se guardan la descripción y el id de cada uno de estos datos que servirán posteriormente para ser guardados en las diferentes tablas que lo necesiten.

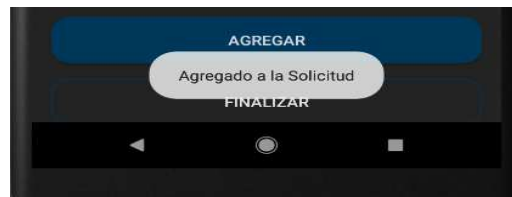


*Figura 41 Listado de unidad*



### ***Insert\_request.php***

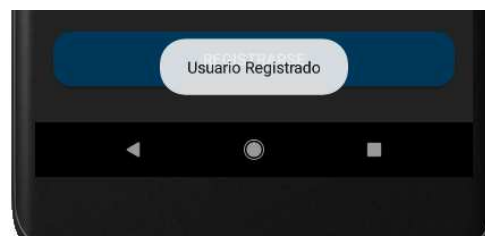
Archivo que es llamado cuando se agrega una solicitud nueva, el cual llama al proceso almacenado para insertar la solicitud en la BD y posterior regresa el id asignado para poder ser usado en la aplicación mediante una preferencia



*Figura 42 Mensaje de solicitud agregada*

### ***Insert\_User.php***

Los datos que se reciben en este archivo son nombre, apellido, correo del usuario, su contraseña y el rol del usuario, todos mediante el método POST. Antes de ser enviado a la base de datos el registro de password es encriptado para mantener la seguridad de la información, se encripta en sha1. Posterior al recibir los datos se manda llamar el procedimiento almacenado CALL Nuevos Usuarios el cual permite el registro del usuario en la base de datos.



*Figura 43 Mensaje insertar usuario*



### ***Validar.php***

Uno de los archivos más necesario que permite identificar que las credenciales que el usuario ingresa en la aplicación son válidos o no existen la base de datos, si existen el procedimiento almacenado envía una respuesta con un arreglo de datos necesarios para trabajar.

### **3.3.5Capa de Datos.**

Esta capa se encarga de acceder a los datos, se debe usar la capa de datos para almacenar y recuperar toda la información de sincronización del Sistema. Es aquí donde se implementa las conexiones al servidor y la base de datos propiamente dicha, se invoca a los procedimientos almacenados los cuales reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Todas estas capas pueden residir en un único ordenador (no debería ser lo usual), pero es lo más frecuente. En sistemas complejos se llega a tener varios ordenadores sobre los cuales reside la capa de datos, y otra serie de ordenadores sobre los cuales reside la base de datos. [11]



### 3.3.6 Gestor de versiones.



*Figura 44 Ejemplos de commits generados*

En lo largo de todo el desarrollo un apoyo importante para el respaldo del código fue Github que me apoyo en el manejo de las versiones y donde pude apoyarme al momento de revertir cambios ya que lo que escribía posterior al último cambio no era de mi agrado o simplemente no se adaptaba a lo que necesitaba.

Logrando así cumplir uno de los objetivos de la tesis poder trabajar con un gestor de versiones, el repositorio se encuentra en <https://github.com/PacoChuy>





## **Capítulo 4**

### **Resultados.**

El ser el encargado del departamento de compras durante algunos años de mi carrera me dejó el aprendizaje de que por la puerta del departamento pueden llegar distintas necesidades muy variadas ya que se convive con diversos departamentos y en cada uno de ellos las necesidades se vuelven prioridad.

El departamento puede convivir con recursos humanos y su necesidad urgente pueden ser una prueba antidoping para la selección de empleados hasta un pastel para el festejo de los mismos y la reacción del departamento tiene que ser inmediata, desafortunadamente las diversas necesidades que llegan al departamento obligan a un comprador en convertirse en experto de materias que ni siquiera puede comprender o que en su educación universitaria no se le dio, es así como recuerdo que en una ocasión me pidieron un proveedor que integrara equipos, la integración de equipos consta del diseño, creación y ejecución de una máquina adaptada a las necesidades de la empresa, por lo que se requería que este equipo manejara presión de aire, al momento de entrevistarme con el proveedor él mencionó unas siglas que PSI mi primer encuentro con las libras de fuerza por pulgada que es la unidad de presión, en este momento comprendí que debía aprender a usar el lenguaje de cualquier usuario que solicitara una cotización.



En este último capítulo se muestra el funcionamiento de la aplicación mediante el planteamiento de una situación de cotización en un entorno controlado.

La empresa N es una empresa Aero espacial que tiene planeado aumentar su producción en el mes de Agosto del 2020 por lo que es necesario habilitar una línea de producción, entre lo que se requiere es la reparación de dos máquinas, adaptar las líneas de desagüe a la línea principal y la instalación eléctrica, esta última actividad se realizara por parte de los técnicos de mantenimiento por lo cual no existe la necesidad de solicitar a un proveedor de instalaciones eléctricas sin embargo si es necesario la compra del material por lo que el departamento de mantenimiento solicita al comprador la cotización del material necesario para la instalación eléctrica de la línea G.

El comprador entiende que por políticas de la empresa y por qué el flujo de efectivo en la empresa los pagos a los proveedores es a 45 días por lo que debe encontrar a proveedores que se ajusten a esta forma de pago.

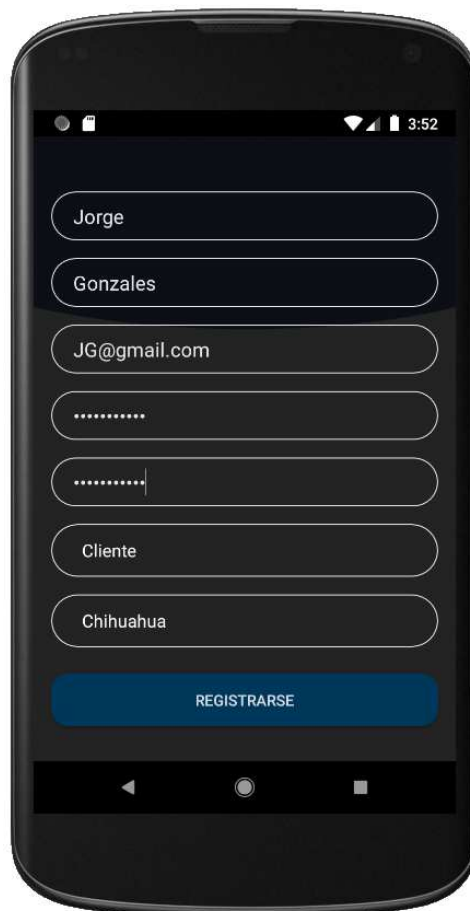
El comprador encuentra la herramienta COT una aplicación móvil que permite solicitar cotizaciones a un universo de proveedores dedicados a su necesidad por lo que la instala en su celular Android.



#### 4.1 Alta de Usuario

Lo primero que debe realizar el comprador es registrarse ya que le permitirá acceder a la aplicación y por supuesto recibir la cotización a su correo.

El comprador indica que es de la ciudad de Chihuahua ya que así el proveedor podrá localizar su solicitud.



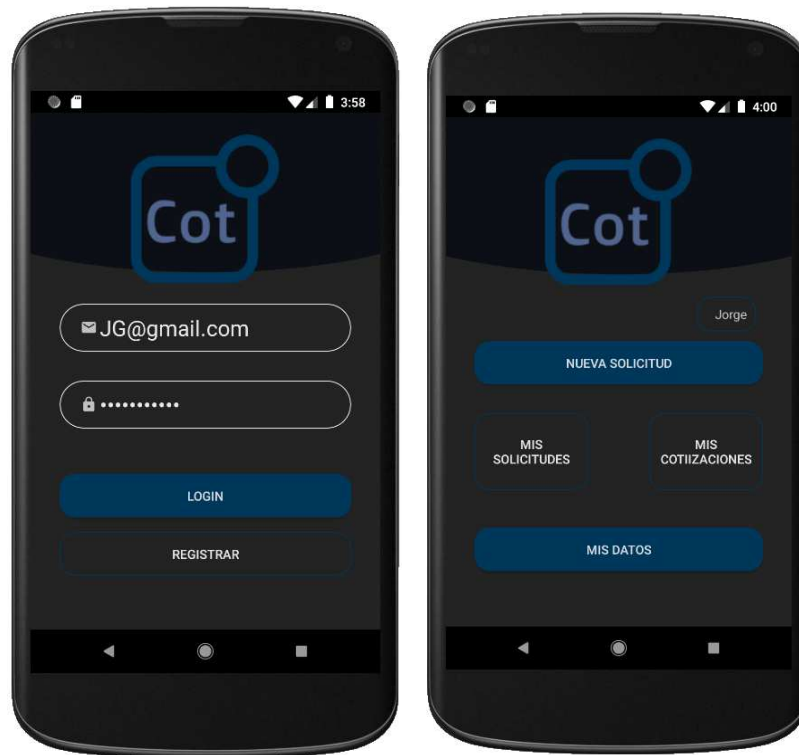
*Figura 45 Alta de usuario*



#### 4.2 Acceso de usuario

Una vez que el comprador está registrado tendrá que ingresar sus credenciales, las cuales son válidas, si estas son correctas se permite el acceso a la pantalla de inicio en la cual se mostrarán cuatro botones y el nombre del usuario que acceso.

Si el comprador sale de la aplicación no tiene que preocuparse por ingresar de nuevo las credenciales ya que la aplicación las guardo y es la aplicación quien interpreta como un usuario registrado así que la aplicación se abrirá directamente en la pantalla de inicio.



*Figura 46 Acceso a la aplicación*



### 4.3 Nueva Solicitud

Para lograr una búsqueda más rápida por parte del proveedor el comprador debe ingresar el nombre del proyecto así como el tipo de material que requiere y los días de crédito que necesita para pagar, estos son los datos que requiere el proveedor pero los datos que requiere el sistema son dos, el número de cotizaciones que se pueden registrar ya que la aplicación tiene un contador cada vez que se envía una cotización, lo que lograra que el comprador no sea saturado de información y una fecha límite, esta fecha limite permitirá que se desactive la solicitud y ya no se muestre al proveedor permitiendo así al proveedor enfocarse en las solicitudes que aún se requieren.

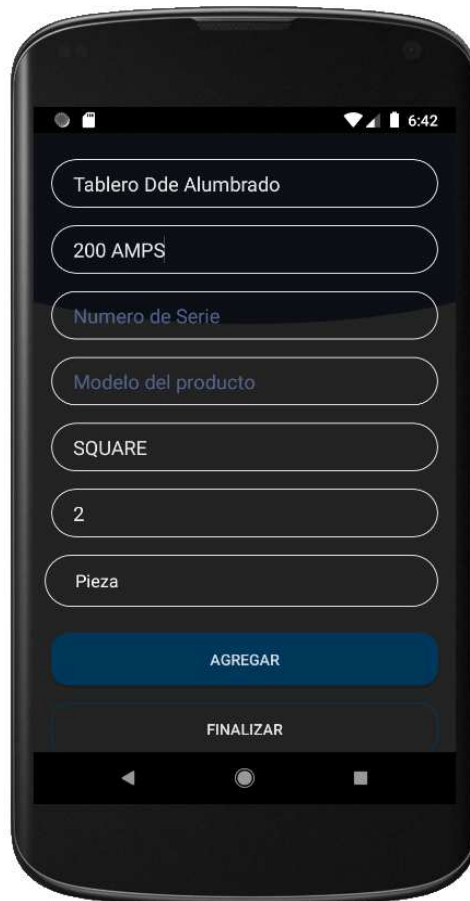


*Figura 47 Crear una solicitud nueva*



### 4.3 Agregar con productos

Una vez que se registra la solicitud se pueden agregar los productos a esta, los cuales se agregan de manera independiente para proporcionar la mayor información posible, de los cuales son dos datos los obligados: el nombre del producto y la cantidad necesaria.



*Figura 48 Agregar producto a la solicitud*



#### 4.4 Ingresar a proveedor

Al confirmar que las credenciales ingresadas por el proveedor son correctas se muestran la pantalla de inicio la cual presenta cuatro botones y el nombre del usuario.



*Figura 49 Pantalla de inicio de proveedor*



#### 4.5 Buscar Solicitud

Continuando con el ejercicio de cotización se tiene que realizar la búsqueda de las solicitudes en este caso el proveedor realiza la búsqueda por materiales eléctricos en la ciudad de chihuahua.

Por lo cual el proveedor obtendrá un listado de todas las solicitudes eléctricas que se hicieron en la ciudad de chihuahua y como se observa en la figura 50 la solicitud que realizamos previamente se encuentra al final de esta lista.



*Figura 50 Búsqueda de solicitudes*





#### 4.5 Buscar Producto

Posterior a seleccionar la solicitud el sistema muestra un listado simple con un texto que puede ser fácilmente digerible para el usuario lo que le permite identificar fácilmente lo que el cliente necesita.



*Figura 51 Búsqueda de productos*



#### 4.6 Cotizar Productos

El proveedor ingresara el costo de cada uno de los productos, el crédito que le puede dar al cliente y la fecha de valides de su propuesta, también se agrega un campo de descripción general donde él puede agregar comentarios adicionales.



*Figura 52 Búsqueda de productos*

#### 4.7 Envió de correo

Una vez que se guardan todos los costos de los productos el proveedor finaliza la cotización lo que sucede a continuación no lo ve el proveedor pero se ejecuta lo siguiente:

La solicitud reduce en uno las cotizaciones requeridas de tal manera que si el cliente requería tres cotizaciones y el proveedor finalizo en este momento su cotización se reduce a dos, esto permite que el cliente no reciba una cantidad mayor de cotizaciones solicitadas.



La cotización es guardada en la base de datos con el estatus de activa y se desactiva solamente si la fecha fin de la propuesta es excedida de este modo el cliente ya no podrá usarla y por ultimo pero no menos importante esta cotización se envía mediante un correo al comprador donde mediante un link puede verla en PDF, este PDF es creado por el servicio web y que le permite darle formalidad a la información capturada



*Figura 53 Finalizar cotizacion*



## **Capítulo 5**

### **Conclusión.**

Eloy Fustero Director de Qualcomm en España en su conferencia Camino al 5G menciona “Hasta la fecha hablando de comunicación móvil siempre hemos desarrollado algo para hablar persona con persona, llevamos treinta años en mejorar la comunicación entre personas”

Este proyecto propone una aplicación que busca la comunicación entre dos actores, en la cual ambos protagonistas pueden realizar el contacto de una manera más sencilla y se demuestra que la información puede ser administrada de manera correcta.

El manejo de la información para el cliente es más dinámico por las pantallas intuitivas logrando la captura de las solicitudes y el detalle de las mismas de manera ordenada y secuencial.

Se observa que el tiempo para la recepción de una cotización se reduce por dos factores, el primer factor es porque se incluye a todos los proveedores para que pueden presentar su propuesta y el segundo es la fecha límite de envío de propuesta asignada por el cliente provocando apresurar al proveedor para él envío de esta.



Se logra que el cliente no obtenga información extra de la solicitada ya que el sistema se limita a capturar los productos que el cliente requiere y a que el sistema solo mantiene activo el envío de cotizaciones hasta que estas alcanzan el número que el cliente pide.

El desarrollo de la aplicación fue un reto poder crear un proceso de actividades ya que se debía contar con total coherencia con los pasos que seguiría cada uno de los participantes sin embargo el objetivo de poder administrar la información se logró.

Al finalizar este proyecto puedo decir que existen demasiados conceptos y temas que integran una aplicación móvil no se puede justificar solo con el hecho de escribir el código ya que se trabaja con temas de un área totalmente administrativa donde se maneja flujo de efectivo, tipo de moneda, costos y llevarlo así de la mano hasta un tema técnico como lo es el diseño de una interfaz gráfica o un logotipo para la aplicación pasando por la administración de la información mediante una base de datos y procedimientos almacenados.



### Lista de referencias

- Víctor R. Tella Medina / Jorge López Bazá (2005). La Tecnología: cómo vencer los retos y necesidades de la cadena de suministro. Recuperado Enero 2020: <http://www.redalyc.org/pdf/402/40280102.pdf>
- Yetnalezi Quintas Ruiz (2007). Elementos a considerar para la implementación de CRM móviles en grandes empresas de monterrey y su área metropolitana. Recuperado Enero 2020: [https://repositorio.tec.mx/bitstream/handle/11285/568147/DocsTec\\_5774.pdf?sequence=1](https://repositorio.tec.mx/bitstream/handle/11285/568147/DocsTec_5774.pdf?sequence=1)
- Rondón, Y., Domínguez, L., & Berenguer, A. (2011). Diseño de la base de datos para sistemas de digitalización y gestión de medias. *Revista de Informática Educativa y Medios Audiovisuales*, 8(15), 17-25. Recuperado Enero 2020: <http://laboratorios.fi.uba.ar/lie/Revista/Articulos/080815/A3mar2011.pdf>
- Acebrón, L. B., & Pedreira, A. M. R. (2012). Importancia del comercio electrónico B2C e impacto de la innovación en los comportamientos de compra. *Abaco Revista de cultura y ciencias sociales*, 21-30. <http://www.revistas culturales.com/xrevistas/PDF/72/1605.pdf>
- Lequerica, J. R. (2003). *Web Services (edición especial)*. Joan Ribas Lequerica. Recuperado Enero 2020
- Rodriguez Jimenez, A. E. (2015). Análisis y planeamiento estratégico para integrar módulo de productos de un sistema local a internet utilizando firewall en Linux y web service en PHP/SQL Server para almacenes Estuardo Sánchez-Alessa. Recuperado Enero 2020 <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/31070>
- Arias, M. Á. (2017). *Aprende Programación Web con PHP y MySQL: 2ª Edición*. IT campus Academy. Recuperado Enero 2020
- Arias, Á. (2014). *Bases de Datos con MySQL: 2ª Edición*. IT Campus Academy. Recuperado Enero 2020
- Rodriguez Jimenez, A. E. (2015). Análisis y planeamiento estratégico para integrar módulo de productos de un sistema local a internet utilizando firewall en Linux y web service en PHP/SQL Server para almacenes Estuardo Sánchez-Alessa. Recuperado Enero 2020 <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/31070>



- Chanchí, G. E., Campo, W. Y., Amaya, J. P., & Arciniegas, J. L. (2011). Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON. *Cadernos de Informática*, 6(1), 233-240.  
Recuperado Enero 2020  
<https://www.seer.ufrgs.br/cadernosdeinformatica/article/view/v6n1p233-240>
- Lopez-Pellicer, F. J., Béjar, R., Latre, M. A., Nogueras-Iso, J., & Zarazaga-Soria, F. J. (2015, July). GitHub como herramienta docente. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática* (pp. 66-73). Universitat Oberta La Salle.  
Recuperado Enero 2020  
<https://upcommons.upc.edu/handle/2117/76761>
- Henríquez, S. D. M., Huerta, H. V., & Grados, L. G. (2010). Programación en N capas. *RISI*, 7(2).  
Recuperado Enero 2020  
<https://scholar.google.es/citations?user=CDt8mKsAAAAJ&hl=es&oi=sra>
- González Nahón, M. (2014). Estudio de mecanismos de autenticación basados en contraseñas visuales en dispositivos móviles android (Bachelor's thesis).  
Recuperado Enero 2020  
<https://repositorio.uam.es/handle/10486/660995>
- Heredia Torres, J. G. (2011). Estudio sobre la implementación de encriptación md5 en sitios web durante el flujo y almacenamiento de contraseñas (Bachelor's thesis, Quito: Universidad Israel, 2011).  
Recuperado Enero 2020  
<http://repositorio.uisrael.edu.ec/handle/47000/192>
- Ostrovskaya, L., & Sarabia-Sánchez, F. J. (2013). Efecto del etnocentrismo sobre la tendencia a usar el nombre de marca en las decisiones de compra. Una aplicación al caso del calzado en España y Rusia. *INNOVAR. Revista de Ciencias Administrativas y Sociales*, 23(49), 51-66.  
Recuperado Enero 2020  
<https://www.redalyc.org/pdf/818/81828691011.pdf>
- Olivares, E. H. (2014). Logotipos, isotipos, imagotipos e isologos: una aclaración terminológica. *IX*, 46, 47.  
Recuperado Enero 2020  
[http://www.paginaspersonales.unam.mx/app/webroot/files/157/Logotipos,\\_isotipos,\\_imagotipos\\_e\\_isologos\\_\(e.pdf](http://www.paginaspersonales.unam.mx/app/webroot/files/157/Logotipos,_isotipos,_imagotipos_e_isologos_(e.pdf)
- Lister, C., West, J. H., Cannon, B., Sax, T., & Brodegard, D. (2014). Just a fad? Gamification in health and fitness apps. *JMIR serious games*, 2(2), e9.  
<https://games.jmir.org/2014/2/e9/>



## Anexo A

### Procedimientos Almacenados

#### *Nuevos Usuarios.*

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Nuevo_Usuarios` (
  IN _Nombre_Usuario VARCHAR (250),
  IN _Apellido_Usuario VARCHAR (250),
  IN _Correo_Usuario VARCHAR (250),
  IN _Password VARCHAR (250),
  IN _id_Rol INT
)
BEGIN
INSERT INTO
tb01_usuario (
  TB01_Nombre_Usuario,
  TB01_Apellido_Usuario,
  TB01_Correo_Usuario,
  TB01_Password,
  TB02_id_Rol,
  TB06_id_Estatus,
  TB01_Fecha_Alta
)
SELECT
  _Nombre_Usuario, _Apellido_Usuario, _Correo_Usuario, _Password, _id_Rol, '3', NOW ()
FROM
  dual
WHERE
  NOT EXISTS (
    SELECT
      TB01_Correo_Usuario
    FROM
      tb01_usuario
    WHERE
      TB01_Correo_Usuario = _Correo_Usuario
    LIMIT
      1
  );
END$$
DELIMITER;
```





### ***Login.***

```
DELIMITER $$
CREATE PROCEDURE `login` (
  IN _Correo_Usuario INT,
  IN _Password INT
)
BEGIN
SELECT TB01_id_Usuario, TB01_Nombre_Usuario, TB01_Apellido_Usuario,
TB01_Telefono_Usuario, TB01_Correo_Usuario, TB02_id_Rol, TB03_id_Ciudad,
TB06_id_Estatus, TB09_id_Empresa FROM tb01_usuario WHERE TB01_Password=_Password
AND TB01_Correo_Usuario=_Correo_Usuario;
END$$
DELIMITER;
```

### ***Busca Rol.***

```
DELIMITER $$
CREATE PROCEDURE `Busca_Rol`()
BEGIN
SELECT tb02_rol.TB02_id_Rol AS id, tb02_rol.TB02_Nombre_Rol AS Rol FROM tb02_rol
WHERE tb02_rol.TB02_id_Rol IN (2, 3) ORDER BY tb02_rol.TB02_Nombre_Rol ASC;
END$$
DELIMITER;
```

### ***Busca cabecera de solicitud.***

```
DELIMITER $$
CREATE PROCEDURE `Busca_Cab_Solicitud`()
IN _Nombre_Categoria VARCHAR (250),
IN _Nombre_Ciudad VARCHAR (250)
)
BEGIN
SELECT tb13_categoria.TB13_id_Categoria INTO @idcat FROM tb13_categoria WHERE
tb13_categoria.TB13_Nombre_Categoria=_Nombre_Categoria;
SELECT tb03_ciudad.TB03_id_Ciudad INTO @idCiu FROM tb03_ciudad WHERE
tb03_ciudad.TB03_Nombre_Ciudad=_Nombre_Ciudad;

SELECT S.TB08_id_Solicitud AS solicitud,
concat_ws("','Solicitud de cotizacion ',S.TB08_Nombre_Solicitud,' con un
',V.TB11_Nombre_Tipo_Venta ,' con una fecha limite el',
DATE_FORMAT(S.TB08_Fecha_Fin_Solicitud, '%d/%m/%y'),") as descripcion
FROM tb08_solicitud AS S INNER JOIN tb11_tipo_venta AS V ON
v.TB11_id_Tipo_Venta=S.TB11_id_Tipo_Venta INNER JOIN tb01_usuario AS U ON
U.TB01_id_Usuario= S.TB01_id_Usuario INNER JOIN tb03_ciudad AS C ON
C.TB03_id_Ciudad=U.TB03_id_Ciudad WHERE S.TB06_id_Estatus = '3' AND
S.TB13_id_Categoria=@idcat AND C.TB03_id_Ciudad=@idCiu ORDER BY
S.TB08_Fecha_Fin_Solicitud,S.TB08_id_Solicitud ;
END$$
DELIMITER ;
```



### ***Busca categoría.***

```
DELIMITER $$
CREATE PROCEDURE `Busca_Categoria`()
BEGIN
SELECT TB13_Nombre_Categoria AS Nombre FROM tb13_categoria ORDER BY Nombre;
END$$
DELIMITER ;
```

### ***Busca Ciudad.***

```
DELIMITER $$
CREATE PROCEDURE `Busca_Ciudad`()
BEGIN
SELECT tb03_ciudad.TB03_Nombre_Ciudad AS Nombre FROM tb03_ciudad ORDER BY
tb03_ciudad.TB03_Nombre_Ciudad ASC;
END$$
DELIMITER ;
```

### ***Busca Crédito.***

```
DELIMITER $$
CREATE PROCEDURE `Busca_Credito`()
BEGIN
SELECT TB11_Nombre_Tipo_Venta AS Nombre FROM tb11_tipo_venta ORDER BY Nombre;
END$$
DELIMITER ;
```

### ***Busca Lista de Producto.***

```
DELIMITER $$
CREATE PROCEDURE `Busca_Lista_Producto`(
IN _id_Solicitud VARCHAR (250)
)
BEGIN
SELECT R.TB04_id_Requerimiento AS producto,concat_ws(", ' Se necesitan
',R.TB04_Cantidad_Producto,' ',U.TB05_Nombre_Largo_Unidad,' de
',R.TB04_Nombre_Producto,' ',R.TB04_Descripcion_Producto,' ',R.TB04_Marca_Producto,'
',R.TB04_Modelo_Producto,' ',R.TB04_Numero_Serie ) As descripcion FROM
tb04_requerimiento AS R INNER JOIN tb05_unidad AS U ON
R.TB05_id_Unidad=U.TB05_id_Unidad WHERE R.TB08_id_Solicitud=_id_Solicitud ORDER BY
R.TB04_Cantidad_Producto;
END$$
DELIMITER ;
```

### ***Busca Unidad.***

```
DELIMITER $$
CREATE ` PROCEDURE `Busca_Unidad`()
BEGIN
SELECT TB05_Nombre_Largo_Unidad AS Nombre FROM tb05_unidad ORDER BY Nombre
ASC;
END$$
DELIMITER ;
```



### ***Cerrar Detalle.***

```
DELIMITER $$
CREATE PROCEDURE `Cerrar_Detalle`(
  IN _id_Solicitud VARCHAR (250)
)
BEGIN
UPDATE tb08_solicitud SET TB06_id_Estatus='3' WHERE TB08_id_Solicitud=_id_Solicitud;
END$$
DELIMITER ;
```

### ***Nueva Solicitud.***

```
DELIMITER $$
CREATE PROCEDURE `Nueva_Solicitud`(
  IN _id_Usuario VARCHAR(5),
  IN _Nombre_Solicitud VARCHAR (250),
  IN _Fecha_Fin_Solicitud DATE,
  IN _Nombre_Categoria VARCHAR (250),
  IN _Numero_Cotizacion VARCHAR(5),
  IN _Nombre_Tipo_Venta VARCHAR (250)
)
BEGIN
  SELECT TB13_id_Categoria INTO @idC FROM tb13_categoria WHERE
  TB13_Nombre_Categoria = _Nombre_Categoria;
  SELECT TB11_id_Tipo_Venta INTO @idV FROM tb11_tipo_venta WHERE
  TB11_Nombre_Tipo_Venta = _Nombre_Tipo_Venta;

  INSERT INTO
  tb08_solicitud(TB01_id_Usuario,TB06_id_Estatus,TB08_Nombre_Solicitud,TB08_Numero_Cotiza
  cion,TB08_Fecha_Solicitud,TB08_Fecha_Fin_Solicitud,TB13_id_Categoria,
  TB11_id_Tipo_Venta)
  SELECT
  _id_Usuario,'1',_Nombre_Solicitud,_Numero_Cotizacion,NOW(),_Fecha_Fin_Solicitud,@idC,@id
  V

  FROM dual WHERE NOT EXISTS (SELECT TB08_Nombre_Solicitud FROM tb08_solicitud
  WHERE TB08_Nombre_Solicitud = _Nombre_Solicitud AND TB01_id_Usuario=_id_Usuario AND
  TB06_id_Estatus=3 LIMIT 1 );

  SELECT max( tb08_solicitud.TB08_id_Solicitud) AS solicitud FROM tb08_solicitud WHERE
  tb08_solicitud .TB01_id_Usuario=_id_Usuario AND tb08_solicitud .TB06_id_Estatus ='1' ORDER
  by TB08_id_Solicitud DESC;

END$$
DELIMITER;
```

### ***Nuevo Requerimiento.***

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Nuevos_Requerimientos`(
  IN _Nombre_Producto VARCHAR (250),
  IN _Descripcion_Producto VARCHAR (250),
```



```
IN _Marca_Producto VARCHAR (250),
IN _Modelo_Producto VARCHAR (250),
IN _Numero_Serie VARCHAR(250),
IN _Cantidad_Producto VARCHAR (250),
IN _id_Solicitud VARCHAR (250),
IN _Nombre_Unidad VARCHAR (250)

)
BEGIN
SELECT tb05_unidad.TB05_id_Unidad INTO @idU FROM tb05_unidad WHERE
tb05_unidad.TB05_Nombre_Largo_Unidad = _Nombre_Unidad;

INSERT INTO tb04_requerimiento
(TB04_Nombre_Producto,TB04_Descripcion_Producto,TB04_Marca_Producto,TB04_
Modelo_Producto,TB04_Numero_Serie,TB04_Cantidad_Producto,TB08_id_Solicitud,T
B05_id_Unidad)

VALUES
(_Nombre_Producto,_Descripcion_Producto,_Marca_Producto,_Modelo_Producto,_Nu
mero_Serie,_Cantidad_Producto,_id_Solicitud,@idU);

END$$
DELIMITER;
```

## Anexo B

### Archivos PHP

#### *connection.php*

```
<?php
$hostname='localhost';
$databse='coti';
$username='root';
$password="";

$conexion=new mysqli($hostname,$username,$password,$databse);
if($conexion->connect_errno){
    echo "El sitio web está experimentado problemas";
}
?>
```



### ***close\_request.php***

```
<?php
include 'connection.php';
$reg_solicitud=$_POST['solicitud'];
$insertar=("CALL Cerrar_Detalle ('$reg_solicitud')");
mysqli_query($conexion,$insertar) or die(mysqli_error());
mysqli_close($conexion);
?>
```

### ***Fill\_List\_Detail.php***

```
<?php
switch($_GET['f']){
    case 'datos':
        $metodos = new Metodos();
        $objDatos = $metodos->obtenerCabecera();
        break;
    }
    class Metodos{
public function obtenerCabecera(){
include 'connection.php';
if(isset($_GET["ci"]) && isset($_GET["ca"])){
$reg_Categoria=$_GET["ca"];
$reg_Ciudad=$_GET["ci"];
$resultado=mysqli_query($conexion,"CALL
Busca_Cab_Solicitud('$reg_Categoria','$reg_Ciudad')")or die(mysqli_error());
$datos=Array();
if (mysqli_num_rows($resultado) > 0) {
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
}
else
{
                $results["solicitud"]='0';
                $results["descripcion"]='0';
                $datos[]=$results;
}
echo json_encode($datos);
$conexion->close();
}
}
?>
```

### ***Fill\_List\_Product.php***

```
<?php
include 'connection.php';
if (isset($_POST['solicitud']));
$reg_solicitud=$_POST['solicitud'];
$resultado=mysqli_query($conexion,"CALL Busca_Lista_Producto('$reg_solicitud')")or
die(mysqli_error());
```



```
$datos=Array();
if (mysqli_num_rows($resultado) > 0) {
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
}
else
{
$results["producto"]="0";
$results["descripcion"]="No hay productos en esta solicitud";
$datos[]=$results;
}
echo json_encode($datos);
$conexion->close();
?>
```

### ***Fill\_Spinner\_Categoria.php***

```
<?php
switch($_GET['funcion']){
case 'C':
$metodos = new Metodos();
$objDatos = $metodos->obtenerCategoria();
break;
}
class Metodos{
public function obtenerCategoria(){
include 'connection.php';
$resultado=mysqli_query($conexion,"CALL Busca_Categoria")or die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
}
}
?>
```

### ***Fill\_Spinner\_Moneda.php***

```
<?php
switch($_GET['funcion']){
case 'U':
$metodos = new Metodos();
$objDatos = $metodos->obtenerUnidad();
break;
}
class Metodos{
public function obtenerUnidad(){
```



```
include 'connection.php';
$resultado=mysqli_query($conexion,"CALL Busca_Moneda")or die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
    $datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
}
?>
```

### ***Fill\_Spinner\_Ciudad.php***

```
<?php

switch($_GET['funcion']){
    case 'cd':
        $metodos = new Metodos();
        $objDatos = $metodos->obtenerCiudad();
        break;
}

class Metodos{

    public function obtenerCiudad(){
include 'connection.php';
$resultado=mysqli_query($conexion,"CALL Busca_Ciudad")or die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
    $datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
}
}
?>
```

### ***Fill\_Spinner\_Credito.php***

```
<?php

switch($_GET['funcion']){
    case 'C':
        $metodos = new Metodos();
        $objDatos = $metodos->obtenerCredito();
        break;
}

class Metodos{
public function obtenerCredito(){
include 'connection.php';
$resultado=mysqli_query($conexion,"CALL Busca_Credito")or die(mysqli_error());
```



```
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
    $datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
}
?>
```

### ***Fill\_Spinner\_Unidad.php***

```
<?php
switch($_GET['funcion']){
    case 'C':
        $metodos = new Metodos();
        $objDatos = $metodos->obtenerCredito();
        break;
}
class Metodos{
public function obtenerCredito(){
include 'connection.php';
$resultado=mysqli_query($conexion,"CALL Busca_Credito")or die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
}
}
?>
```

### ***Find\_Cab\_Solicitud.php***

```
<?php
include 'connection.php';
if (isset($_POST['Categoria']) && !empty($_POST['Categoria'])
    && isset($_POST['Ciudad']) && !empty($_POST['Ciudad']));
$reg_Categoria=$_POST['Categoria'];
$reg_Ciudad=$_POST['Ciudad'];
$resultado=mysqli_query($conexion,"CALL
Busca_Cab_Solicitud('$reg_Categoria','$reg_Ciudad')")or die(mysqli_error());
$datos=Array();
if (mysqli_num_rows($resultado) > 0) {
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
}
else
{
    $results["solicitud"]='0';
    $results["descripcion"]='No hay requerimientos para esta ciudad';
}
```





```
                $datos[]=$results;
    }
    echo json_encode($datos);
    $conexion->close();
?>
```

### ***Fill\_Spinner\_Rol.php***

```
<?php
switch($_GET['funcion']){case 'obtenerRol':
                        $metodos = new Metodos();
                        $objDatos = $metodos->obtenerRol();
                        break; }

class Metodos{
    public function obtenerRol(){
include 'connection.php';
$resultado=mysqli_query($conexion,"CALL Busca_Rol")or die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {$datos[]=$respuesta;}
echo json_encode($datos);
$conexion->close();
    }
}
?>
```

### ***Insert\_request.php***

```
<?php
include 'connection.php';
$reg_usuario=$_POST['usuario'];
$reg_nombre=$_POST['Nombre_Solicitud'];
$reg_fecha=$_POST['Fecha_Fin'];
$reg_categoria=$_POST['categoria'];
$reg_cotizacion=$_POST['num_cotizacion'];
$reg_credito=$_POST['credito'];
$resultado=mysqli_query($conexion,"CALL
Nueva_Solicitud('$reg_usuario','$reg_nombre','$reg_fecha','$reg_categoria','$reg_cotizaci
on','$reg_credito')")or die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
?>
```



### ***Insert\_request\_detail.php***

```
<?php
include 'connection.php';
$reg_Nombre_p=$_POST['Nombre_producto'];
$reg_Descripcion_p=$_POST['Descripcion_Producto'];
$reg_Serie_p=$_POST['Serie_Producto'];
$reg_Modelo_p=$_POST['Modelo_Producto'];
$reg_Marca_p=$_POST['Marca_Producto'];
$reg_Cantidad_p=$_POST['Cantidad_Producto'];
$reg_Unidad_p=$_POST['Unidad_Producto'];
$reg_solicitud=$_POST['solicitud'];
$insertar=("CALL
Nuevos_Requerimientos('$reg_Nombre_p','$reg_Descripcion_p','$reg_Marca_p','$reg_M
odelo_p','$reg_Serie_p','$reg_Cantidad_p','$reg_solicitud','$reg_Unidad_p')");
mysqli_query($conexion,$insertar) or die(mysqli_error());
mysqli_close($conexion);
?>
```

### ***Insert\_User.php***

```
<?php
include 'connection.php';
$reg_Nombre_u=$_POST['Nombre_usuario'];
$reg_Apellido_u=$_POST['Apellido_usuario'];
$reg_Correo_u=$_POST['Correo_usuario'];
$reg_Password_u=sha1($_POST['Password_usuario']);
$reg_Rol_u=$_POST['reg_Rol_u'];
$insertar=("CALL
Nuevos_Usuarios('$reg_Nombre_u','$reg_Apellido_u','$reg_Correo_u','$reg_Password_u
','$reg_Rol_u')");
mysqli_query($conexion,$insertar) or die(mysqli_error());
mysqli_close($conexion);
?>
```

### ***Insert\_cotizacion.php***

```
<?php
include 'connection.php';
$reg_Producto_u=$_POST['Producto'];
$reg_Nombre_u=$_POST['Nombre_Producto'];
$reg_Costo_u=$_POST['Costo_Producto'];
$reg_Credito_u=$_POST['Credito'];
$reg_Moneda_u=$_POST['Moneda'];
$reg_Fecha_u=$_POST['Fecha_Fin'];
```



```
$reg_Unidad_u=$_POST['Unidad'];
$reg_Usuario_u=$_POST['Usuario'];
$reg_Pieza_u=$_POST['Pieza_Producto'];
$reg_Descripcion_u=$_POST['Descripcion'];

$resultado=mysqli_query($conexion,"CALL
Nueva_Cotizacion('$reg_Producto_u','$reg_Nombre_u','$reg_Costo_u','$reg_Credito_u',
$reg_Moneda_u','$reg_Fecha_u','$reg_Unidad_u','$reg_Usuario_u','$reg_Pieza_u')")or
die(mysqli_error());
mysqli_close($conexion);
?>
```

### ***login.php***

```
<?php
switch($_GET['f']){
    case 'datos':
        $metodos = new Metodos();
        $objDatos = $metodos->obtenerdatos();
        break;
    }
    class Metodos{
public function obtenerdatos(){
include 'connection.php';
if(isset($_GET["u"]) && isset($_GET["p"])){
    $usu_usuario=$_GET['u'];
    $usu_password=sha1($_GET['p']);
$resultado=mysqli_query($conexion,"CALL login ('$usu_usuario','$usu_password')")or
die(mysqli_error());
$datos=Array();
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
echo json_encode($datos);
$conexion->close();
    }
}
?>
```

### ***validar.php***

```
<?php
include 'connection.php';
if(isset($_POST['usuario']) && isset($_POST['clave'])){
    $usu_usuario=$_POST['usuario'];
```



```
$usu_password=sha1($_POST['clave']);
$resultado=mysqli_query($conexion,"CALL login ('$usu_usuario','$usu_password')")or
die(mysqli_error());
$datos=Array();
if (mysqli_num_rows($resultado) > 0) {
while ($respuesta= mysqli_fetch_object($resultado)) {
$datos[]=$respuesta;
}
}
else
{
    $results["usuario"]='0';
    $results["nombre"]='0';
    $results["apellido"]='0';
    $results["telefono"]='0';
    $results["correo"]='0';
    $results["rol"]='0';
    $results["ciudad"]='0';
    $results["empresa"]='0';
    $datos[]=$results;
}
echo json_encode($datos);
$conexion->close();
?>
```



## Anexo C

### Código Android

#### *MainActivy.java*

```
package com.example.cotapp;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {
    EditText txt_Usuario, txt_Password;
    Button btnLogin;
    String usuario, password;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txt_Usuario=findViewById(R.id.txt_Usuario);
        txt_Password=findViewById(R.id.txt_password);
        btnLogin= findViewById(R.id.btn_Login);
        recuperarPreferencias();

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                usuario=txt_Usuario.getText().toString();
                password=txt_Password.getText().toString();
                if(!usuario.isEmpty() && !password.isEmpty()){
                    validarUsuario(
                        "http://192.168.0.5/coti/controller/login.php");
                }
                else
                {
```



```
        Toast.makeText(MainActivity.this, "No se permiten
campos vacios", Toast.LENGTH_SHORT).show();
    }
}

private void validarUsuario (String URL ){
    StringRequest StringRequest= new StringRequest (Request.Method.POST,
URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if (!response.isEmpty()){
                guardarPreferencias();
                Intent intent= new Intent (getApplicationContext(),
Home_activity.class);
                startActivity(intent);
                finish();
            }else{
                Toast.makeText( MainActivity.this, "usuario o contraseña
incorrectos", Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText (MainActivity.this,
error.toString(), Toast.LENGTH_SHORT).show();
        }
    }){
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map <String,String> parametros =new HashMap<String, String>();
            parametros.put ("usuario", usuario);
            parametros.put ("password", password);
            return parametros;
        }
    };
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    requestQueue.add(StringRequest);
}

private void guardarPreferencias(){
    SharedPreferences
preferences=getSharedPreferences("preferenciasLogin", Context.MODE_PRIVATE );
    SharedPreferences.Editor editor = preferences.edit();
    editor.putString("usuario", usuario);
    editor.putString("password", password);
    editor.putBoolean("sesion", true);
    editor.commit();
}

private void recuperarPreferencias(){
    SharedPreferences preferences =
getSharedPreferences ("preferenciasLogin", Context.MODE_PRIVATE);
    txt_Usuario.setText (preferences.getString("usuario", "Correo"));
    txt_Password.setText (preferences.getString("password", "Password"));
}
}
```



```
public void registro_usuario (View view ){
    Intent datos = new Intent(this, Registro_Usuario.class);
    startActivity(datos);
}

}
```

## *Registro\_Usuario.java*

```
public class Registro_Usuario extends AppCompatActivity {

    List<String>listaIdSucursales; //guarda el id de los Roles
    List<String>listaRoles; // guarda el nombre de la sucursal
    EditText nombre, apellido, correo, password, confirm_password;
    Button btnAgregar;
    Spinner spinnerRol;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registro_usuario);
        nombre=(EditText) findViewById(R.id.txt_Nombre);
        apellido=(EditText) findViewById(R.id.txt_Apellido);
        correo=(EditText) findViewById(R.id.txt_Correo);
        password=(EditText) findViewById(R.id.txt_password);
        confirm_password=(EditText) findViewById(R.id.txt_con_pass);
        btnAgregar=(Button) findViewById(R.id.btn_registar);
        spinnerRol=(Spinner) findViewById(R.id.sp_Rol);
        cargarRoles();

        btnAgregar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (nombre.length()==0
                || apellido.length()==0 || correo.length()==0 || password.length()==0 || confirm_passw
                ord.length()==0)
                {
                    Toast.makeText(Registro_Usuario.this, "Informacion
                pendiente por llenar ", Toast.LENGTH_LONG).show();
                }
                else {
                    ejecutarServicio("Insert_User.php");
                }
            }
        });
    }

    public void cargarRoles(){
        RequestQueue queue = Volley.newRequestQueue(this);
        String url ="Fill_Spinner_Rol.php?funcion=obtenerRol";
        StringRequest stringRequest = new StringRequest(Request.Method.GET,
        conexion.URL_WEB_SERVICES+url, new Response.Listener<String>() {
```



```
@Override
public void onResponse(String response) {
    if (response.length()>0){
        try {
            JSONArray jsonArray = new JSONArray(response);
            obtenerRoles(jsonArray);

        }catch (JSONException jsnex1){

Toast.makeText(getApplicationContext(), jsnex1.toString(), Toast.LENGTH_LONG).show();
        }

    }

}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {

Toast.makeText(getApplicationContext(), error.toString(), Toast.LENGTH_LONG).show();
    }

});
queue.add(stringRequest);
}

public void obtenerRoles(JSONArray jsonArray){
    listaIdSucursales=new ArrayList<String>();
    listaRoles = new ArrayList<String>();
    for(int i=0; i<jsonArray.length(); i++){
        try{
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            String id = jsonObject.getString("id");
            String Rol =jsonObject.getString("Rol");
            listaIdSucursales.add(id);
            listaRoles.add(Rol);
        }catch (JSONException jsnEx2){

Toast.makeText(getApplicationContext(), jsnEx2.toString(), Toast.LENGTH_LONG).show();
        }

    }

}

ArrayAdapter<String>adapterRoles = new
ArrayAdapter<String>(this,R.layout.support_simple_spinner_dropdown_item,listaRoles);
spinnerRol.setAdapter(adapterRoles);
}

private void ejecutarServicio(String URL)
{
    StringRequest stringRequest=new StringRequest(Request.Method.POST,
conexion.URL_WEB_SERVICES +URL, new Response.Listener<String>() {
```





```
        @Override
        public void onResponse(String response) {
            if (!response.isEmpty()) {
                Toast.makeText(getApplicationContext(), "Este usuario ya
esta registrado", Toast.LENGTH_SHORT).show();
            }
            else{
                nombre.setText("");
                apellido.setText("");
                correo.setText("");
                password.setText("");
                confirm_password.setText("");
                Toast.makeText(getApplicationContext(), "Usuario
Registrado", Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

Toast.makeText(getApplicationContext(), error.toString(), Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> parametros=new HashMap<>();
            parametros.put("Nombre_usuario", nombre.getText().toString());

parametros.put("Apellido_usuario", apellido.getText().toString());
            parametros.put("Correo_usuario", correo.getText().toString());

parametros.put("Password_usuario", password.getText().toString());

parametros.put("$reg_Rol_u", spinnerRol.getSelectedItem().toString());
            return parametros;
        }
    };
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    requestQueue.add(stringRequest);
}
}
```



## **Curriculum vitae.**

### **JESÚS FRANCISCO PONCE DOMÍNGUEZ**

Lic. En Informática

Correo: Ponce.Jesuf@gmail.com

#### **Objetivo.**

Contribuir con la organización en el cumplimiento de las metas y políticas establecidas considerando tiempos y fechas requeridas, desarrollarme profesional y personalmente logrando en conjunto un beneficio para los fines de la empresa y la sociedad.

#### **Preparación Académica.**

Licenciado en Informática

Instituto Tecnológico de Delicias

Especialidad |Tecnologías de la Información y Gestión. 2008 – 2013 Estatus| Titulado

#### **Experiencia Laboral.**

##### ***Ingeniero de Software.***

Operadora Futurama | Diciembre 2019 – Actualmente.

Desarrollo de software interno para la administración de la información interna en la cadena de sucursales.

##### ***Desarrollador Analista.***

Sistemas y Soluciones Administrativas | Mayo 2018 – Diciembre 2019.



Desarrollo de software para cliente externo con compatibilidad a sistemas Conpaqi en C# y SQLServer. Mediante el desarrollo de un software interno (PHP, JavaScript, MySQL.) se logró establecer un procedimiento para la captura de los servicios a los clientes para su posterior facturación.

### ***Comprador de Proyectos.***

Bio Pappel | Abril 2017 –Mayo2018.

Realizar órdenes de compra en ERP SAP. Búsqueda y desarrollo de proveedores.

Realizar licitaciones dirigidas a la compra. Negociar precios y acuerdos con proveedores.

Gestión de contratos con departamento legal.

### ***Comprador.***

Grupo American Industries| Febrero 2016 – Enero 2017.

Compras nacionales y en el extranjero. Realizar órdenes de compra en Oracle. Solicitud de cotizaciones. Negociar precios y acuerdos con proveedores. Desarrollo de proveedores. Seguimiento al pedido hasta asegurar la entrega.

### ***Soporte IT.***

Televisora Delicias. | Agosto 2015 – Febrero 2016

Diseño, creación y administración la página Web de la Institución. Apoyar a usuarios en operaciones de servicios informáticos. Control de la red.

### ***Soporte IT.***

Tristone Flowtech Mexico | Abril 2012 – Agosto 2015

Administrar la página Web de la Institución. Implementación de sistema de monitoreo.

Creación de sistema interno para disposición de desperdicios.



### ***Comprador.***

Administración y Coordinación de las actividades necesarias para la adquisición y suministros de los bienes y servicios que se requerían en la Institución con las mejores condiciones de compra.

### **Idiomas**

Inglés: Nivel Intermedio.

Diplomado de Ingles

Universidad Autónoma de Chihuahua | Agosto 2018 –Octubre 2019.

### **Aptitudes**

Creatividad, Organización, Sociabilidad, Empatía, Iniciativa, Trabajo en equipo.

### **Habilidades personales y profesionales**

Desarrollo en los lenguajes C#, PHP, HTML5, JavaScript. Administracion en los motores de Base de datos: MYSQL, SQL SERVER. Control de versiones en GIT (Github – GitLab).

Certificaciones en Conpaqi Facturación Electrónica, Contabilidad y Nominas.

Certificaciones en Inttelmex IT como Programador orientado a objetos y desarrollador front-end.