



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**TECNOLÓGICO
NACIONAL DE MÉXICO**

Instituto Tecnológico de Pabellón de Arteaga
Departamento de Ciencias Económico Administrativas

REPORTE FINAL PARA ACREDITAR RESIDENCIA PROFESIONAL DE LA CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN.

[SOFTV - APP]



SOFTV Sistemas Administrativos de Televisión Restringida

M.A.T.I Jorge Norberto Mondragón Reyes
Ing. Miguel Vázquez Martín del Campo

04 de Diciembre del 2019

CAPITULO 1: PRELIMINARES

2.- AGRADECIMIENTOS

De antemano le doy gracias a mis profesores por los conocimientos adquiridos en mi periodo estudiantil, por cada enseñanza y consejo dado que serviría de gran ayuda en el momento de realizar mis residencias profesionales, por cada asesoramiento en algún conflicto que se pudo presentar al momento de la realización de este proyecto.

A todos y cada uno de mis compañeros de clases así como de otras ingenierías por los conocimientos en otras áreas que de igual manera me sirvieron de mucho, por cada apoyo y consejo.

Agradezco infinitamente a mis padres por ser el pilar de que yo continuará con mis estudios en nivel superior, por cada consejo y apoyo de todo tipo.

Se refiere a un conjunto de software que las organizaciones utilizan para administrar las actividades comerciales diarias, como contabilidad, adquisición, administración de proyectos, administración de riesgo y cumplimiento y operaciones de la cadena de suministro.

3.-RESUMEN

El proyecto realizado consta de una aplicación móvil basada en un ERP. Un conjunto de software que las organizaciones utilizan para administrar las actividades comerciales diarias, como contabilidad, adquisición, administración de proyectos, administración de riesgo y cumplimiento y suministro. Un sistema ya implementado en las empresas de telecomunicaciones, empresas que proveen de servicios de internet y telefonía en distintos sectores de la república mexicana así como en latino américa.

En base a este sistema en la manera en que los técnicos acudían al domicilio asignado para llevar alguna instalación o algún reporte solicitado al cliente, este notificaba por medio de documentación a la mesas de control ya que ellos se encargaban de registrar en sistema las ordenes de cada técnico así como los aparatos y material que usaban. La aplicación se implementó de manera que el técnico pueda ejecutar las órdenes y reportes desde su propio teléfono móvil de una manera más sencilla, óptima, y eficaz, ahorrando tiempo y haciéndolo de una manera más segura. El técnico puede ver que material se le ha asignado (calle, conectores, aparatos, etc.) así como podrá observar todas las ordenes y reportes que se le asignen en su día laboral.

ÍNDICE

Contenido

CAPITULO 1: PRELIMINARES	2
2.- AGRADECIMIENTOS	2
3.-RESUMEN	3
ÍNDICE	4
CAPITULO 2: GENERALIDADES DEL PROYECTO	6
5.- INTRODUCCIÓN	6
6.- DESCRIPCIÓN DE LA EMPRESA U ORGANIZACIÓN Y DEL PUESTO O AREA DE TRABAJO DEL RESIDENTE.....	7
7.-PROBLEMAS A RESOLVER	8
7.1 Implementar nuevas tecnologías en la empresa	8
7.2 Implementar componentes de una aplicación Android	8
7.4 Manejo de pantallas (Aplicación responsiva)	8
7.5 Manejo de versiones.....	8
8.- JUSTIFICACIÓN	9
9.- OBJETIVOS	10
9.1 Objetivo general	10
9.2 Objetivo específicos	10
CAPÍTULO 3: MARCO TEÓRICO	11
10.- MARCO TEÓRICO (Fundamentos teóricos)	11
10.1 Internet.....	11
10.2 Internet en México (Historia)	11
10.3 Demanda de internet en México	11
10.4 Uso de internet en México.....	12
10.5 Telefonía	12
10.6 Aplicaciones Móviles.....	13
10.6.1 Historia	13
10.6.2 Evolución.....	13
10.7 Estructura y modelos de programación en Android	14
10.7.1 Lenguaje Java	14
10.7.3 El modelo Orientado a Objetos.....	14
10.7.4 Objetos.....	14
10.7.5 Las Clases.....	15
10.7.6 Herencia.....	16

10.7.7 Abstracción.....	17
10.7.8 Encapsulamiento	18
10.7.9 Ocultamiento	18
10.8 Lenguajes de Programación Orientado a Objetos	18
10.8.1 Análisis y diseño Orientado a Objetos	19
10.9 Modelo vista controlador.....	19
10.10 Servicio web	20
10.11 Web Token	21
10.12 JSON	21
CAPITULO 4. DESARROLLO	22
11.- PROCEDIMIENTO Y DESCRIPCION DE LAS ACTIVIDADES REALIZADAS	22
11.1 Cronograma de actividades	22
11.2 Creación y uso del login (Autenticación).....	22
11.3 Pantallas de secciones	24
11.3.1 Tabs.....	25
11.4 Diseño de listas	27
11.5 Implementación de fotos de instalación	28
CAPITULO 5: RESULTADOS	31
CAPITULO 6: CONCLUSIONES	32
CAPITULO 7: COMPETENCIAS DESARROLLADAS	33
CAPITULO 8: FUENTES DE INFORMACIÓN	34
Bibliografía	34

LISTA DE ILUSTRACIONES

Ilustración 1 ejemplo de clase con atributos	15
Ilustración 2 Ejemplo de instancia de un objeto a partir de una clase	16
Ilustración 3 ejemplo de herencia entre clases	17
Ilustración 4 ejemplo de superclases.....	17
Ilustración 5 ejemplo de Web Token.....	21
Ilustración 6 Cronograma de actividades.....	22
Ilustración 7 Pantalla de Login.....	23
Ilustración 8 Configuración de pantalla responsiva.....	23
Ilustración 9 programación de botón inicio de sesión.....	24
Ilustración 10 Programación de servicio web por medio de Retrofit.....	24
Ilustración 11 Graficas de uso	31

CAPITULO 2: GENERALIDADES DEL PROYECTO

5.- INTRODUCCIÓN

En el mundo actual el desarrollo de las aplicaciones móviles crece día a día, implementando nuevos sistemas domésticos, de entretenimiento o empresariales que nos ayudan a mejorar u optimizar alguna actividad o proceso que realicemos.

Día a día podemos hacer una infinidad de cosas desde nuestro teléfono móvil desde jugar un videojuego con conexión internet, con otros usuarios, en diferentes países, así como realizar nuestros pedidos ya sea de comida, transporte o pagar algún servicio como televisión o internet.

La comunicación entre usuarios es cada vez más estable debido a que las telecomunicaciones van evolucionando y ya hay más cobertura en más áreas del planeta, en este caso México.

En este nuevo proyecto se realizó una aplicación móvil a nivel empresarial para la empresa SOFTV, una empresa que brinda diversas tecnologías a las empresas de telecomunicaciones que proveen de servicios de telefonía e internet entre otros. La Aplicación móvil se diseñó de la mano de un sistema ERP que usan diferentes empresas a lo largo de la república mexicana y latino américa.

Esta aplicación móvil será de gran ayuda ya que el técnico ejecutar las ordenes y reportes en el momento actual y desde el mismo domicilio, optimizando el sistema ya antes implementado ya que desde mesa de control podrá corroborar que en verdad se realizó la orden con éxito o reporte según sea el caso.

La aplicación se montó sobre el sistema operativo Android ya que en su mayoría los dispositivos móviles, es intuitiva para que cualquier usuario pueda usarla con exactitud y sin problema, así como responsiva para adaptar a la múltiple variedad de dispositivos que funcionan con sistema operativo Android.

En este reporte se explicara una a una las funciones de la aplicación móvil como cada funcionalidad técnica, la creación de servicios web, nuevos procedimientos para la base de datos, así como modificación de los mismos.

6.- DESCRIPCIÓN DE LA EMPRESA U ORGANIZACIÓN Y DEL PUESTO O AREA DE TRABAJO DEL RESIDENTE.

SOFTV es una empresa que como principales actividades son la creación de sistemas Web y de escritorio ERP para empresas de telefonía e internet, brinda sistemas de gestión de la información de la empresa que integra todas las áreas y las intercomunica de forma que mediante la explotación de estos datos puede hacer crecer la empresa aplicando los procesos de negocio necesarios.

Mi puesto en la realización de este proyecto fue programador Android en lenguaje java, que de la mano con los programadores que realizaban y modificaban el ERP me fueros orientando y ayudando en su realización.

7.-PROBLEMAS A RESOLVER

Debido a que los técnicos al momento de realizar sus instalaciones o atender alguna queja notificaban a mesa de control acerca de las órdenes y reportes que ellos realizaban, se topaban con problemas de comunicación o llenar mal los formatos ya que todo era sobre papel.

7.1 Implementar nuevas tecnologías en la empresa

Como era el primer proyecto que la empresa desarrollaba bajo plataforma Android se tuvo que indagar en aprender lenguaje java y estar a prueba y error con los resultados obtenidos.

7.2 Implementar componentes de una aplicación Android

En cuanto a las peticiones y requerimientos de la empresa, en cuestión de diseño se optaba por manejar componentes nuevos como listas personalizadas, manejo de la cámara el dispositivo móvil, uso de la firma electrónica, entre otros.

Se tenía que ver librerías nuevas y que estas a su vez fueran compatibles con la versión de Android para la cual estábamos desarrollando la aplicación.

7.4 Manejo de pantallas (Aplicación responsiva)

Debido a que el sistema operativo Android está diseñado para múltiples pantallas en diferentes dispositivos, de un principio los componentes no se veían en su totalidad o muy grandes o muy chicos por lo que en base a la investigación se fueron añadiendo diferentes tamaños de pantallas para así fuera adaptable para cualquier dispositivo.

7.5 Manejo de versiones

En el desarrollo de la aplicación móvil estuvimos tres personas a cargo del proyecto por lo cual se nos asignaban varias tareas, a veces por separado o en conjunto, en la misma parte, por lo que al momento de añadir nuestro nuevo código se empalmaba con el de otro compañero y teníamos problemas al momento de compilar y sacar la aplicación final.

Con el tiempo el manejo de versiones mejoro debido a que las tareas fueron asignadas por diferente parte y así ya no teníamos problemas de este tipo.

8.- JUSTIFICACIÓN

En la actualidad todos los trabajos de instalación de telefonía, televisión e internet se realizan por un técnico seleccionado en el domicilio para instalar dichos servicios, estas órdenes se llevan a cabo por mesa de control que usan un software para agendar las ordenes y reportes a los domicilios indicados.

Las ordenes que se le agendan al técnico son en base a una bitácora en papel y este en base a sus trabajos realizaba las actividades según la dirección de domicilio.

Con este proyecto se pretende crear una aplicación móvil en plataforma Android para que el técnico día a día sepa que actividades a realizar y ejecutarlas desde la misma app sin necesidad de informar a mesa de control. La app registrara la hora de ejecución así como el lugar sin necesidad de llenar ningún documento, será intuitiva para tener un proceso más exacto y las órdenes se ejecuten con exactitud sin margen de error.

La aplicación móvil le dará a conocer las órdenes y reportes de diversos clientes de telefonía, internet o televisión que tenga contratada la empresa.

Dentro de la misma app se podrán ejecutar varios trabajos como una instalación de aparato que esta se realiza después de haber contratado un servicio de internet, telefonía o televisión según sea el caso y es la principal prioridad de la aplicación, se podrán hacer retiros de aparato para cuando el cliente cancele el servicio que tiene contratado, cambios de aparato por si tiene algún aparato dañado así como cambios de domicilio para cuando el cliente decida cambiar su estancia a otra parte.

9.- OBJETIVOS

9.1 Objetivo general

Crear una aplicación móvil basada de un ERP de telecomunicaciones que sea de uso para los técnicos que llevan sus tareas de instalación o algún otro trabajo que se le asigne

9.2 Objetivo específicos

- Lograr la implementación de la aplicación en el periodo establecido
- Tener buenas referencias de la aplicación en cuanto a funcionamiento de las empresas que la utilicen.
- Entregar la aplicación totalmente funcional.
- Evaluar el correcto funcionamiento de la aplicación móvil mediante pruebas.
- Hacer modificaciones de la aplicación según la empresa que la solicite.
- Implementar nuevas funcionalidades y este más completa según requerimientos de los clientes.

CAPÍTULO 3: MARCO TEÓRICO

10.- MARCO TEÓRICO (Fundamentos teóricos)

10.1 Internet

El Internet, es un sistema mundial de redes de computadoras, un conjunto integrado por las diferentes redes de cada país del mundo, por medio del cual un usuario en cualquier computadora puede, en caso de contar con los permisos apropiados, acceder a información de otra computadora y poder tener inclusive comunicación directa con otros usuarios en otras computadoras.

10.2 Internet en México (Historia)

La historia de Internet en México propiamente da inicio en 1989, cuando el Instituto Tecnológico y de Estudios Superiores de Monterrey estableció el primer enlace dedicado a la red de la National Science Foundation (NSF), a través de la Escuela de Medicina de la Universidad de Texas, en la ciudad de San Antonio (UTSA), utilizando los protocolos de conexión propios de Internet. Entonces se conecta el primer equipo a Internet bajo el dominio .mx: dns.mty.itesm.mx con la dirección 131.178.1.1 El 28 febrero de 1989, la NSFnet reconoció oficialmente la conexión de México.

Poco después fue establecido un segundo nodo entre el Instituto de Astronomía de la Universidad Nacional Autónoma de México, y el Centro Nacional de Investigación Atmosférica (NCAR) de Boulder, Colorado, en Estados Unidos. Posteriormente el ITESM Campus Estado de México se conectó a la red de la NSF a través del NCAR, por medio de un enlace digital vía satélite.

De acuerdo con un estudio realizado por NIC México, en 1991 los servicios que con mayor frecuencia utilizaban los académicos e investigadores eran los siguientes:

- Acceso remoto (Telnet)
- Transferencia de Archivos (FTP)
- Correo Electrónico (E-mail)

10.3 Demanda de internet en México

En México hay 74.3 millones de usuarios de Internet de seis años o más, que representan el 65.8% de la población en ese rango de edad. El 51.5% de los internautas son mujeres y 48.5% son hombres. Se observa un crecimiento de 4.2 puntos porcentuales respecto

a lo reportado en 2017, cuando se registraron 71.3 millones de usuarios. Del total de la población usuaria de internet de seis años o más, el grupo de entre 25 y 34 años es el que registra la mayor proporción de usuarios de internet, las mujeres en este rango de edad representan 10.4% y los hombres 9.8%. Por otro lado, la población de 55 años o más es la que menos usa internet, registrando cifras del 4.1% para las mujeres y 4.0% para los hombres. Las tres principales actividades de los usuarios de Internet en 2018 fueron: entretenimiento (90.5%), comunicación (90.3%) y obtención de información (86.9 por ciento).

10.4 Uso de internet en México.

En México hay 74.3 millones de usuarios de Internet de seis años o más, que representan el 65.8% de la población en ese rango de edad. El 51.5% de los internautas son mujeres y 48.5% son hombres. Se observa un crecimiento de 4.2 puntos porcentuales respecto a lo reportado en 2017, cuando se registraron 71.3 millones de usuarios. Del total de la población usuaria de internet de seis años o más, el grupo de entre 25 y 34 años es el que registra la mayor proporción de usuarios de internet, las mujeres en este rango de edad representan 10.4% y los hombres 9.8%. Por otro lado, la población de 55 años o más es la que menos usa internet, registrando cifras del 4.1% para las mujeres y 4.0% para los hombres. Las tres principales actividades de los usuarios de Internet en 2018 fueron: entretenimiento (90.5%), comunicación (90.3%) y obtención de información (86.9 por ciento).

10.5 Telefonía

Durante el año pasado, 73.5% de la población de seis años o más utilizó el teléfono celular. De éstos, ocho de cada diez usuarios, contaban con un celular inteligente (Smartphone), que les permitía conectarse a Internet. El número total de usuarios que disponen de celular inteligente (Smartphone) creció de 64.7 millones de personas en 2017 a 69.6 millones en 2018. Además, en 2018 hay un aumento de los usuarios que se conectan a internet desde un celular inteligente (Smartphone), pasando del 92.0% en 2017 a 93.4% en 2018; con una diferencia de 5.5 millones de personas. La conexión móvil a internet (conexión de datos) es la más utilizada por el 89.0% de los usuarios, mientras que el restante 11.0% se conecta a internet desde un celular inteligente (Smartphone) mediante WiFi. De los usuarios de celular inteligente (Smartphone), 45.5 millones instalaron aplicaciones en sus teléfonos: 89.5% de mensajería instantánea, 81.2% herramientas para acceso a redes sociales, 71.9% aplicaciones de contenidos de audio y video, y 18.1% alguna aplicación para acceder a banca móvil.

10.6 Aplicaciones Móviles

Una aplicación móvil la podemos definir como cualquier programa informático que ejecuta tu teléfono móvil para realizar una tarea, mostrar medios de información, facilitar la comunicación, entretener o brindar un servicio.

10.6.1 Historia

Cualquier software que esté diseñado para ser usado en un dispositivo móvil y que cumpla con una función: ESO ES UNA APP, así que aunque antes podíamos pensar que el término se reservaba para funciones sofisticadas, la realidad es que las primeras apps fueron simples y sencillas. Fue a principios de los 80s cuando las principales empresas productoras de calculadoras lanzaron a la venta las “Hand Held Computers” computadoras que se sostenían en la mano. Entre sus principales usuarios se encontró PepsiCo siendo sus repartidores quienes ocuparon las funciones del hand held computer para levantar pedidos y registrar sus distribuciones. Sí, a través de una calculadora con una programación ligeramente más compleja.

A inicios de los noventas, cuando el uso de teléfonos móviles se disparó, pudimos encontrar en ellos funciones como calculadoras, juegos arcade, sms o ringtones, esas sencillas características que el día de hoy pasan desapercibidas en nuestros móviles fueron el cimiento para el complejo mundo de aplicaciones del que hoy un 61.75% de la población mundial es usuario.

Muchas personas tienden a pensar que las apps no tienen más de una década de existencia, esto es porque su boom (tal como las conocemos) se dio en 2008 cuando de la mano del primer iPhone emergió un mercado que marcaría la historia y el uso de los móviles. Casi de inmediato conocimos un nuevo sistema operativo que competiría directamente con Apple: Android. Estos dos gigantes cuentan con más de 3M de distintas aplicaciones móviles en sus mercados.

10.6.2 Evolución

La evolución que está sufriendo el mercado de las apps no sigue una línea continua, ya que el consumidor ha evolucionado de la misma manera que el mercado. Durante los años 2012 y 2013 el número de descargas de aplicaciones era desmedido. El consumidor quería conocer nuevas aplicaciones y usarlas.

En esta línea parece que tendría más sentido diseñar aplicaciones móviles enfocadas de un modo todo lo particular que se pueda e intentando no moverse en la generalidad, ya que es más difícil mantener la fidelidad con el público y su uso de las apps. Todavía

existen diferentes ámbitos de mercado por explorar en donde veremos cómo se desarrollan las futuras aplicaciones.

Existe un gran mercado en desarrollo de las apps, casi se podría decir que hay un app para cada ámbito de nuestras vidas. Muchas marcas están aprovechando este nicho de mercado para crecer. Sin embargo, esta estrategia debe estar bien planteada, ya que el usuario de telefonía móvil es más consciente del poder que tiene en sus manos.

En este sentido, un buen servicio hacia el usuario final es una atención fundamental, ya que de esa manera se va construyendo una nueva relación que permitirá a la marca y al cliente establecer un canal de comunicación en el que puedan expresar sus necesidades. Ese servicio que se busca dar es determinante a la hora de que un usuario continúe usando o no la aplicación.

10.7 Estructura y modelos de programación en Android

10.7.1 Lenguaje Java

La tecnología Java se usa para desarrollar aplicaciones para un amplio alcance de entornos, desde dispositivos del consumidor hasta sistemas empresariales heterogéneos. En esta sección, obtenga una vista de alto nivel de la plataforma Java y sus componentes.

10.7.3 El modelo Orientado a Objetos

Para entender este modelo vamos a revisar 4 conceptos básicos:

- Objetos
- Clases
- Herencia
- Envío de mensajes

10.7.4 Objetos

Entender que es un objeto es la clave para entender cualquier lenguaje orientado a objetos.

Existen muchas definiciones que se le ha dado al Objeto. Primero empecemos entendiendo que es un objeto del mundo real. Un objeto del mundo real es cualquier cosa que vemos a nuestro alrededor. Digamos que para leer este artículo lo hacemos a través del monitor y una computadora, ambos son objetos, al igual que nuestro teléfono celular, un árbol o un automóvil.

Analicemos un poco más a un objeto del mundo real, como la computadora. No necesitamos ser expertos en hardware para saber que una computadora está compuesta internamente por varios componentes: la tarjeta madre, el chip del procesador, un disco duro, una tarjeta de video, y otras partes más. El trabajo en conjunto de todos estos componentes hace operar a una computadora.

Internamente, cada uno de estos componentes puede ser sumamente complicado y puede ser fabricado por diversas compañías con diversos métodos de diseño. Pero nosotros no necesitamos saber cómo trabajan cada uno de estos componentes, como saber qué hace cada uno de los chips de la tarjeta madre, o cómo funciona internamente el procesador. Cada componente es una unidad autónoma, y todo lo que necesitamos saber de adentro es cómo interactúan entre sí los componentes, saber por ejemplo si el procesador y las memorias son compatibles con la tarjeta madre, o conocer donde se coloca la tarjeta de video. Cuando conocemos como interaccionan los componentes entre sí, podremos armar fácilmente una computadora.

Los Objetos de Software, al igual que los objetos del mundo real, también tienen características y comportamientos. Un objeto de software mantiene sus características en una o más "variables", e implementa su comportamiento con "métodos". Un método es una función o subrutina asociada a un objeto.

10.7.5 Las Clases

En mundo real, normalmente tenemos muchos objetos del mismo tipo. Por ejemplo, nuestro teléfono celular es sólo uno de los miles que hay en el mundo. Si hablamos en términos de la programación orientada a objetos, podemos decir que nuestro objeto celular es una instancia de una clase conocida como "celular". Los celulares tienen características (marca, modelo, sistema operativo, pantalla, teclado, etc.) y comportamientos (hacer y recibir llamadas, enviar mensajes multimedia, transmisión de datos, etc.).

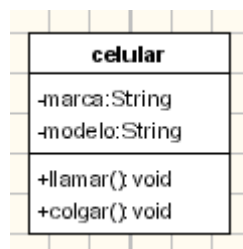


Ilustración 1 ejemplo de clase con atributos

Cuando se fabrican los celulares, los fabricantes aprovechan el hecho de que los celulares comparten esas características comunes y construyen modelos o plantillas comunes, para que a partir de esas se puedan crear muchos equipos celulares del mismo

modelo. A ese modelo o plantilla le llamamos CLASE, y a los equipos que sacamos a partir de ella la llamamos OBJETOS.

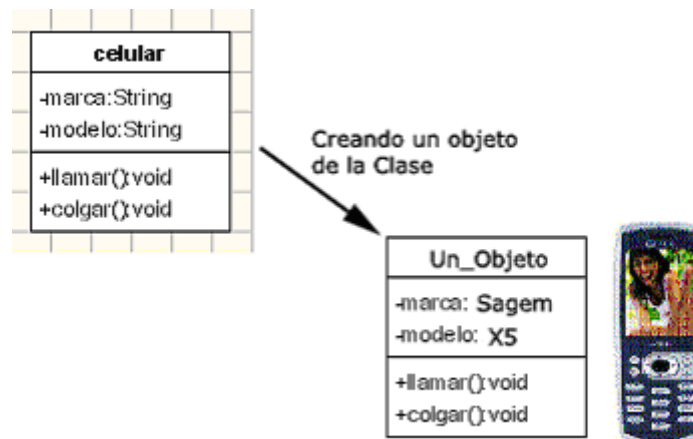


Ilustración 2 Ejemplo de instancia de un objeto a partir de una clase

Esto mismo se aplica a los objetos de software, se puede tener muchos objetos del mismo tipo y mismas características.

Definición teórica: La clase es un modelo o prototipo que define las variables y métodos comunes a todos los objetos de cierta clase. También se puede decir que una clase es una plantilla genérica para un conjunto de objetos de similares características.

Por otro lado, una instancia de una clase es otra forma de llamar a un objeto. En realidad no existe diferencia entre un objeto y una instancia. Sólo que el objeto es un término más general, pero los objetos y las instancias son ambas representación de una clase.

Definición Teórica: Una instancia es un objeto de una clase en particular.

10.7.6 Herencia

La herencia es uno de los conceptos más cruciales en la POO. La herencia básicamente consiste en que una clase puede heredar sus variables y métodos a varias subclases (la clase que hereda es llamada superclase o clase padre). Esto significa que una subclase, aparte de los atributos y métodos propios, tiene incorporados los atributos y métodos heredados de la superclase. De esta manera se crea una jerarquía de herencia.

Por ejemplo, imaginemos que estamos haciendo el análisis de un Sistema para una tienda que vende y repara equipos celulares.

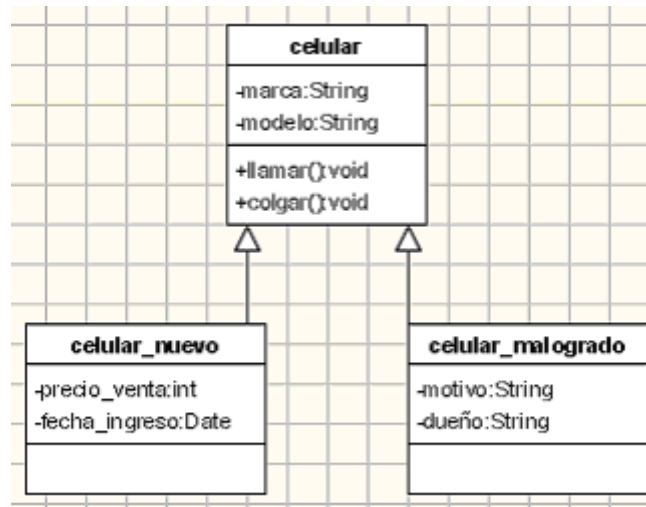


Ilustración 3 ejemplo de herencia entre clases

En el gráfico vemos 2 Clases más que posiblemente necesitemos para crear nuestro Sistema. Esas 2 Clases nuevas se construirán a partir de la Clase Celular existente. De esa forma utilizamos el comportamiento de la súper clase.

En general, podemos tener una gran jerarquía de Clases tal y como vemos en el siguiente gráfico:

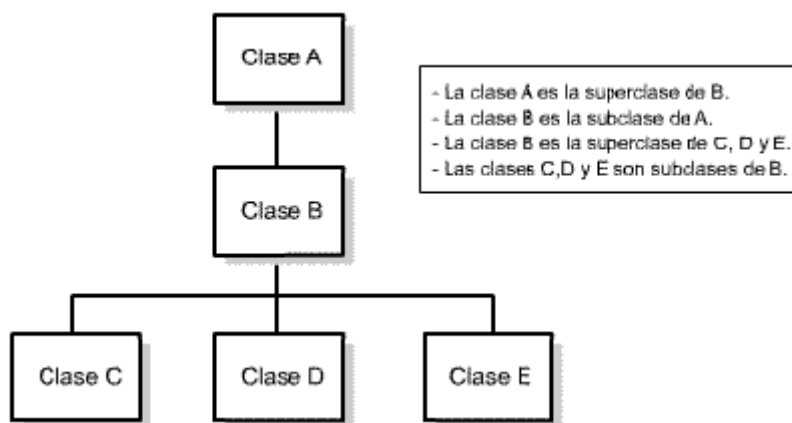


Ilustración 4 ejemplo de superclases

10.7.7 Abstracción

La abstracción consiste en captar las características esenciales de un objeto, así como su comportamiento. Por ejemplo, volvamos al ejemplo de los automóviles, ¿Qué características podemos abstraer de los automóviles? O lo que es lo mismo ¿Qué características semejantes tienen todos los automóviles? Todos tendrán una marca, un

modelo, número de chasis, peso, llantas, puertas, ventanas, etc. Y en cuanto a su comportamiento todos los automóviles podrán acelerar, frenar, retroceder, etc.

En los lenguajes de programación orientada a objetos, el concepto de Clase es la representación y el mecanismo por el cual se gestionan las abstracciones.

10.7.8 Encapsulamiento

El encapsulamiento consiste en unir en la Clase las características y comportamientos, esto es, las variables y métodos. Es tener todo esto en una sola entidad. En los lenguajes estructurados esto era imposible. Es evidente que el encapsulamiento se logra gracias a la abstracción y el ocultamiento que veremos a continuación.

La utilidad del encapsulamiento va por la facilidad para manejar la complejidad, ya que tendremos a las Clases como cajas negras donde sólo se conoce el comportamiento pero no los detalles internos, y esto es conveniente porque nos interesará saber qué hace la Clase pero no será necesario saber cómo lo hace.

10.7.9 Ocultamiento

Es la capacidad de ocultar los detalles internos del comportamiento de una Clase y exponer sólo los detalles que sean necesarios para el resto del sistema.

El ocultamiento permite 2 cosas: restringir y controlar el uso de la Clase. Restringir porque habrá cierto comportamiento privado de la Clase que no podrá ser accedido por otras Clases. Y controlar porque daremos ciertos mecanismos para modificar el estado de nuestra Clase y es en estos mecanismos donde se validarán que algunas condiciones se cumplan. En Java el ocultamiento se logra usando las palabras reservadas: public, private y protected delante de las variables y métodos.

10.8 Lenguajes de Programación Orientado a Objetos

En 1985, E. Stroustrup extendió el lenguaje de programación C a C++, es decir C con conceptos de clases y objetos, también por esas fechas se creó desde sus bases el lenguaje Eiffel.

En 1995 apareció el más reciente lenguaje OO, Java desarrollado por SUN, que hereda conceptos de C++.

El lenguaje de desarrollo más extendido para aplicaciones Web, el PHP 5, trae todas las características necesarias para desarrollar software orientado a objetos.

Además de otros lenguajes que fueron evolucionando, como el Pascal a Delphi.

Finalmente también otros lenguajes script como el ActionScript que si bien no es totalmente orientado a objetos pero sí posee las características.

10.8.1 Análisis y diseño Orientado a Objetos

Para el desarrollo de software orientado a objetos no basta usar un lenguaje orientado a objetos. También se necesitará realizar un análisis y diseño orientado a objetos.

El modelamiento visual es la clave para realizar el análisis OO. Desde los inicios del desarrollo de software OO han existido diferentes metodologías para hacer esto del modelamiento, pero sin lugar a duda, el Lenguaje de Modelamiento Unificado (UML) puso fin a la guerra de metodologías.

Según los mismos diseñadores del lenguaje UML, éste tiene como fin modelar cualquier tipo de sistemas (no solamente de software) usando los conceptos de la orientación a objetos. Y además, este lenguaje debe ser entendible para los humanos y máquinas.

10.9 Modelo vista controlador

En el patrón de arquitectura MVC (Modelo-Vista-Controlador) todos los objetos de nuestra aplicación deben caer en una de las 3 categorías: Ser un objeto del modelo (de datos) o de vista o un controlador.

Un objeto del modelo guarda tanto los datos de la aplicación como la denominada “Lógica del Negocio”.

En aplicaciones android, las clases del modelo suelen ser creadas desde cero por el desarrollador. Todos los objetos del modelo conforman la Capa de Modelo de Datos. Una regla simple dice que cualquier cosa que veamos en pantalla es una vista. El conjunto de objetos vista de una aplicación componen la Capa de Presentación o Visualización.

Los controladores se diseñan para que puedan responder a los distintos eventos originados por los objetos de vista y gestionar el flujo de datos entre los objetos del modelo de datos y la capa de visualización.

La separación del código fuente en “Clases” nos facilita el diseño y comprensión de la aplicación como un todo, puesto que de este modo podemos pensar en términos de clases y no de variables y métodos individuales. De forma parecida la separación de clases en las “Capas modelo de datos, vista y controladores” nos permite dar un paso

más a la hora de diseñar y entender la aplicación completa, ya que podemos pensar en términos de capas y no de clases individuales.

MVC también facilita la reutilización de las clases. “Una clase con responsabilidades limitadas es más reutilizable que una con la que se cuente en todas partes”.

Si una app se ejecuta en un dispositivo que tenga una densidad en la pantalla no incluida en ninguna de las carpetas previstas (mdpi/hdpi/xhdpi/xxhdpi/xxxhdpi), Android realizará automáticamente el escalado de la imagen disponible al tamaño adecuado para el dispositivo. Gracias a estas características, no es imprescindible proporcionar imágenes para todas las densidades de píxeles posibles. Podemos centrarnos en una o varias resoluciones más altas, reduciendo el tamaño de la app.

10.10 Servicio web

El World Wide Web Consortium (W3C) define un servicio web como un sistema de software designado para dar soporte a la interacción de máquina a máquina interpretativa a través de una red.

Un servicio web realiza una tarea específica o un conjunto de tareas, y se describe mediante una descripción de servicio en una notación XML estándar llamada WSDL (Web Services Description Language). La descripción de servicio proporciona todos los detalles necesarios para interactuar con el servicio, incluidos los formatos de mensaje (que detallan las operaciones), los protocolos de transporte y la ubicación.

Otros sistemas utilizan mensajes SOAP para interactuar con el servicio web, normalmente utilizando HTTP con una serialización XML conjuntamente con otros estándares relacionados con la web.

La interfaz WSDL oculta los detalles de cómo se implementa el servicio, y el servicio se puede utilizar independientemente de la plataforma de hardware o software en la que se implementa e independientemente del lenguaje de programación en el que está escrito.

Las aplicaciones basadas en servicios web son implementaciones en todas las tecnologías, con acoplamientos flexibles y orientados a componentes. Los servicios web se pueden utilizar individualmente o junto con otros servicios web, para llevar a cabo una agregación completa o una transacción empresarial.

10.11 Web Token

Si hay algo característico de las API, REST o cualquier otra arquitectura, es que no manejan estado y por lo tanto no tienen sesiones. Lo destacable ahora es que, como consecuencia de no tener estado y no tener sesiones en el servidor para cada usuario, el backend es incapaz de recordar al usuario entre llamada y llamada al API.

Para evitar que en cada llamada el usuario deba entregar su clave y password se usa un token, que no es más que una cadena de texto larga, encriptada con una clave. Según se hace el login, el servidor devuelve al cliente un token y el cliente en futuros accesos entrega ese token para certificar que está autenticado.

```
2019-11-04 09:56:41.031 2050-2084/system_process I/GnssLocationProvider: WakeLock released by handleMessage(REPORT_SV_STATUS, 0, com.android.server.location.GnssLocationProvider$SVSt
2019-11-04 09:56:41.032 2050-2213/system_process I/GnssLocationProvider: WakeLock acquired by handleMessage(REPORT_SV_STATUS, 0, com.android.server.location.GnssLocationProvider$SVSt
2019-11-04 09:56:41.034 2050-2084/system_process I/GnssLocationProvider: WakeLock released by handleMessage(REPORT_SV_STATUS, 0, com.android.server.location.GnssLocationProvider$SVSt
2019-11-04 09:56:41.870 26611-26707/com.Softv.SoftvApp.SoftvApp D/token: eyJhbGciOiJIUWU1NiIsInR5cCI6IkpXVCJ9.eyJ1bmI6dGVmZjZiZmZ3IiwiaWF0IjoiMTU3ODQ1MjY0In0.eyJhbGciOiJIUWU1NiIsInR5cCI6IkpXVCJ9.eyJ1bmI6dGVmZjZiZmZ3IiwiaWF0IjoiMTU3ODQ1MjY0In0
2019-11-04 09:56:42.030 2050-2213/system_process I/GnssLocationProvider: WakeLock acquired by handleMessage(REPORT_SV_STATUS, 0, com.android.server.location.GnssLocationProvider$SVSt
2019-11-04 09:56:42.031 2050-2084/system_process I/GnssLocationProvider: WakeLock released by handleMessage(REPORT_SV_STATUS, 0, com.android.server.location.GnssLocationProvider$SVSt
```

Ilustración 5 ejemplo de Web Token

10.12 JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

CAPITULO 4. DESARROLLO

11.- PROCEDIMIENTO Y DESCRIPCION DE LAS ACTIVIDADES REALIZADAS

11.1 Cronograma de actividades

Esta aplicación móvil como ya se dijo se realizó en lenguaje Java y se utilizó el software de Android Studio para su creación.

Dado a que la aplicación iba a realizar varias tareas e iba a hacer llamada a varios procedimientos nos dimos a la tarea de dividir las partes del trabajo entre las 3 personas encargadas de la realización de este proyecto.

FASES	ACTIVIDADES	SEMANAS	AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Creación del login	Diseño de pantalla principal		■															
	Responsividad entre componentes de pantalla			■														
	Generar token				■													
	Programación de botones					■												
Pantalla de secciones	Diseño de pantalla con Tabs de movimiento					■												
	Diseño de pantalla de cada sección de las Tabs						■											
	Consumo de WEB API de información del cliente							■										
	Consumo de WEB API de trabajos de técnico								■									
	Programación de ejecución de ordenes ejecutadas									■								
Diseño de listas	Programación de ordenes ejecutadas como visita									■								
	Diseño de listas de ordenes agendadas										■							
	Diseño de lista de reportes agendados											■						
	Consumo de WEB API para mostrar listas												■					
	Pruebas de ordenes y reportes agendados													■				
Implementación de fotos de instalación	Implementación de componentes en listas personalizadas													■				
	Programación de servicios de camara														■			
	creacion de procedimientos almacenados para codificar imágenes															■		
	creacion de servicio web para guardar imágenes dentro del servidor																■	
Filtros de busqueda	pruebas con nuevo servicio implementado																	■
	Implementación de filtros de busqueda ordenes																	
Publicación	Implementación de filtros de busqueda reportes																	
	Modificación de datos y componentes a petición de la empresa																	
	publicacion de servicios web																	
	Implementación de tablas y procedimientos de base de datos																	

Ilustración 6 Cronograma de actividades

11.2 Creación y uso del login (Autenticación)

Para dar pie con la creación de la aplicación como primera tarea nos dimos a la creación del login, la autenticación de un usuario por medio de una contraseña, para esto se tuvo que generar un token que como ya se mencionó va ligado con la autenticación.

La pantalla principal con el logo de la empresa únicamente pidiendo usuario y contraseña con la imagen de la principal empresa.



Ilustración 7 Pantalla de Login.

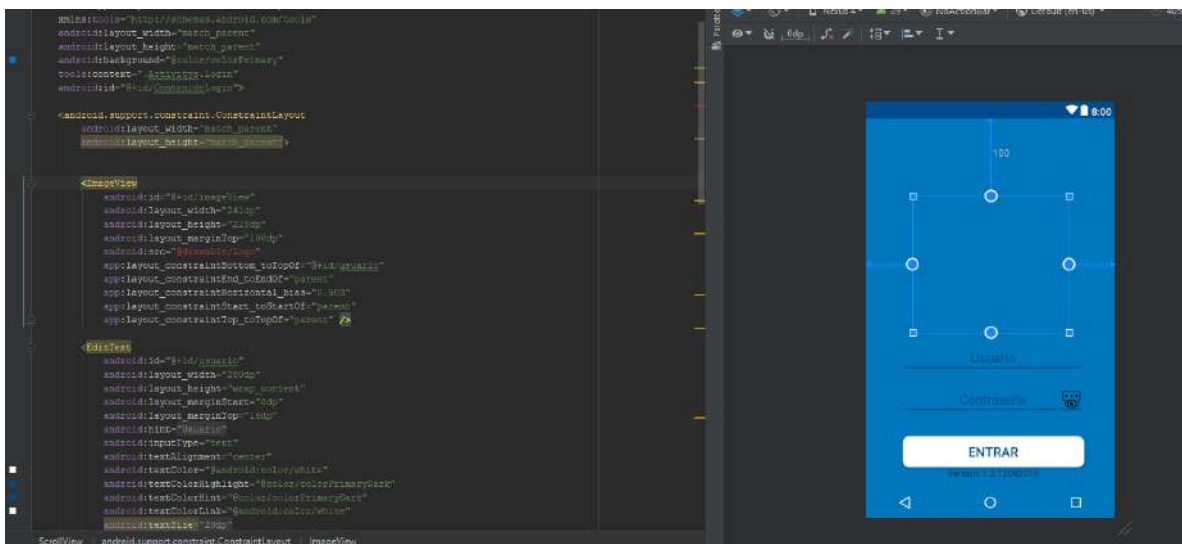


Ilustración 8 Configuración de pantalla responsiva

De esta manera nos encargábamos de que la aplicación fuera responsiva por medio de la herramienta de componentes que nos ofrecía Android Studio especificamos márgenes y tamaños de todos los elementos.

```

//Boton de inicio de sesion
entrar.setOnClickListener(v -> {
    //verifica que el campo usuario este lleno
    if (usuario.getText().toString().length() == 0) {
        Toast.makeText(getApplicationContext(), text "Introduzca usuario", Toast.LENGTH_LONG).show();
    } else {
        //verifica que el campo contraseña este lleno
        if (contraseña.getText().toString().length() == 0) {
            Toast.makeText(getApplicationContext(), text "Introduzca contraseña", Toast.LENGTH_LONG).show();
        } else {
            //verifica que el usuario tenga internet
            if (!isOnline()) {
                Toast.makeText(getApplicationContext(), text "No cuenta con conexión a Internet", Toast.LENGTH_LONG).show();
            } else {
                //se encadena el usuario y contraseña
                user = usuario.getText().toString() + ":" + contraseña.getText().toString();
                //se codifica la cadena de texto en base64
                enco = (android.util.Base64.encodeToString(user.getBytes(), android.util.Base64.NO_WRAP));
                //se manda el campo usuario y enco para guardarlos en el SharedPreferences
                guardarPre(usuario.getText().toString(), enco);
                //manda request del login
                request.getReviews(context Login.this, dialogLogin, view);
                //inicia el dialog de sesion
                dialogLogin.show();
                ////////////
            }
        }
    }
}
}

```

Ilustración 9 programación de botón inicio de sesión

Por medio de funciones ya establecidas teníamos que instanciar cada elemento de cada pantalla para darles sus funciones correspondientes y programar características propias del botón por medio de funciones en su respectiva clase con sus validaciones correspondientes.

```

//Proxima cita//
public void getProximaCita(final Context context, final JSONObject jsonObject, final View view, final ProgressDialog dialogInicio,
    final Activity activity) {
    Call<JSONObject> call = services.RequestPost(context, jsonObject).getDataProx();
    call.enqueue(new Callback<JSONObject>() {
        @Override
        public void onResponse(Call<JSONObject> call, Response<JSONObject> response) {
            if (response.code() == 200) {
                TextView tipoTrabajo, contratoTrabajo, horaTrabajo, calleDireccion, numeroDireccion, coloniaDireccion;
                JSONObject userJson = response.body().getAsJsonObject("GetDameSiguienteCitaResult");
                String jsonToString = String.valueOf(userJson);
                ProximaCitaModel proximaCitaModel = new ProximaCitaModel();
                Gson gson = new Gson();
                proximaCitaModel = gson.fromJson(jsonToString, ProximaCitaModel.class);
                tipoTrabajo = view.findViewById(R.id.tipoDeTrabajo);
                contratoTrabajo = view.findViewById(R.id.contrato);
                horaTrabajo = view.findViewById(R.id.hora);
                calleDireccion = view.findViewById(R.id.calle);
                numeroDireccion = view.findViewById(R.id.numero);
                coloniaDireccion = view.findViewById(R.id.colonia);
                tipoTrabajo.setText(proximaCitaModel.Tipo);
                contratoTrabajo.setText(proximaCitaModel.Contrato);
                horaTrabajo.setText(proximaCitaModel.Hora);
                calleDireccion.setText(proximaCitaModel.Calle);
                numeroDireccion.setText(proximaCitaModel.NUMERO);
                coloniaDireccion.setText(proximaCitaModel.Colonia);
            } else {
                ErrorInicioDeSesion(context, dialogInicio, activity);
                ErrorMessage(context, error "Error al conseguir datos de inicio");
            }
        }
    }
}

```

Ilustración 10 Programación de servicio web por medio de Retrofit

En esta función como en muchas otras se programó en base a la pantalla correspondiente diferentes funciones que por medio de la librería Retrofit se llamaba un servicio web y se obtenían los datos correspondientes para hacer mención a otro elemento o mostrar información del usuario o de algún otro servicio que se necesitara.

11.3 Pantallas de secciones

En la aplicación para que el técnico pudiese ejecutar una orden de instalación, consultar los datos del cliente, ver que trabajos se le tienen asignados, ver que aparatos se le van a instalar según los servicios que contrato entre otras cosas.

11.3.1 Tabs

La app está dividida por secciones, si el técnico desea ejecutar una orden tiene que pasar por varios filtros los cuales son:

Estatus:

En esta pantalla el usuario puede ver la información del cliente, se puede ver la latitud y longitud para dar la localización donde se ejecutara esa orden, tiene observaciones previas del cliente antes de solicitar su servicio u observaciones que se le hacen al momento de agendar la orden.

En este caso es una orden y puede marcar la orden como visita que es en dado caso que el cliente no se encuentre en el domicilio o deje alguna razón.

Al marcar la orden como ejecutar da paso continuar con la siguiente sección que es trabajos.

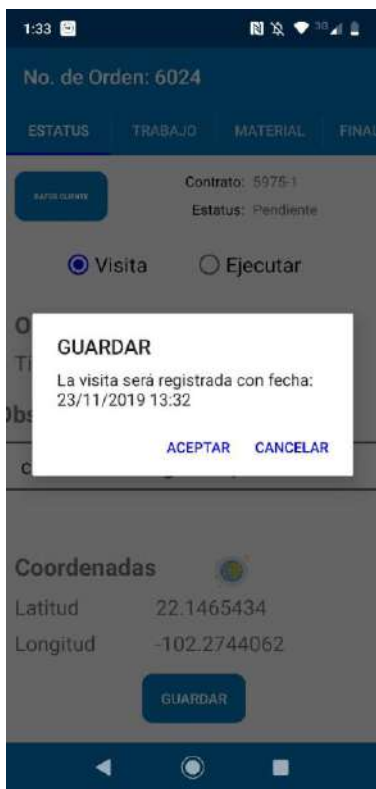


Ilustración 11 Pantalla de sección de Estatus



Ilustración 12 Confirmar marcar orden como visita con fecha y hora



Ilustración 13 Información del cliente



Ilustración 14 Pantalla de sección de trabajos

En esta sección de trabajos el técnico puede ver los trabajos que tiene asignados según la orden establecida según por la clave el trabajo se sabe qué tipo de trabajo realizara ya sea una instalación, un cambio de aparato o domicilio, entre otros trabajos más.

Se puede observar el estatus de la orden y el número de contrato, así como el botón de datos del cliente para cualquier consulta que se pueda hacer.



Ilustración 15 Ejemplo pantalla de trabajo, cambio de aparato



Ilustración 16 Ejemplo pantalla instalación de aparatos

11.4 Diseño de listas



Ilustración 17 Ejemplo de List View, lista de ordenes agendadas

Como ya se mencionó el técnico se le agendan en su día a día varias órdenes y trabajos que estos son mostrados por listas personalizadas en la aplicación móvil con un componente llamado list view siendo una vista que visualiza una lista deslizable verticalmente de varios elementos.

Por medio de este componente y con el uso de la librería retrofit consumimos un servicio web para órdenes como para trabajos para de esta forma personalizarlos y mostrar la lista en una pantalla en la aplicación.

En esta lista personalizada se muestran las ordenes que tiene agendadas el técnico mostrando el contrato, numero de orden y el estatus de la orden, esta puede ser en visita o pendiente, así mismo se observan los datos del cliente, la fecha de agenda de la orden, el primer trabajo a realizar.

```

@Override
public void onBindViewHolder(ordensrcViewHolder viewHolder, final int position) {

    viewHolder.noOrden.setText("No. de Orden");
    viewHolder.nombre.setText(Array.nombresrc.get(position));
    viewHolder.orden.setText(Array.ordensrc.get(position));
    viewHolder.contrato1.setText(Array.contratosrc.get(position));
    viewHolder.status.setText(Array.statussrc.get(position));
    viewHolder.direccionOrd.setText(Array.direccionsrc.get(position));

    viewHolder.trabajoordenes.setText(Array.trabajosrc.get(position));

    if(Array.napsrc.get(position).equals("") || Array.napsrc.get(position).equals(null)) {
        viewHolder.txtnap.setVisibility(View.GONE);
        viewHolder.napordenes.setVisibility(View.GONE);
    }else{
        viewHolder.txtnap.setVisibility(View.VISIBLE);
        viewHolder.napordenes.setVisibility(View.VISIBLE);
        viewHolder.napordenes.setText(Array.napsrc.get(position));
    }

    if(Array.tapsrc.get(position).equals("") || Array.fapsrc.get(position).equals(null)) {
        viewHolder.txttap.setVisibility(View.GONE);
        viewHolder.tapordenes.setVisibility(View.GONE);
    }else{
        viewHolder.txttap.setVisibility(View.VISIBLE);
        viewHolder.tapordenes.setVisibility(View.VISIBLE);
        viewHolder.tapordenes.setText(Array.fapsrc.get(position));
    }

    viewHolder.control.setOnClickListener((v) -> {
        Util.preferences = mContext.getSharedPreferences("credenciales", Context.MODE_PRIVATE);
        Util.editor = Util.preferences.edit();
        Util.editor.putInt("c1vOrden", Integer.valueOf(ordensrc.get(position)));
        Util.editor.commit();

        request.getDeepCons(mContext);
        request.getValidaFirma(mContext);
    });
}

```

Ilustración 18 Programación de adaptador para list view

En la parte de programar el adaptador para hacer llamada a la lista se necesita de instanciar cada componente que se hace visible dentro de esta, hacer llamar un array (una lista de información) creada a partir del consumo del servicio Web y asignar a cada componente un parámetro de la lista.

11.5 Implementación de fotos de instalación

Después de haber realizado una orden de instalación el técnico debe capturar como mínimo 5 fotos de evidencia.

Para realizar este proceso se utilizó una librería externa llamada magic cámara para tomar fotos y seleccionar imágenes en Android de manera simple y también guardar las imágenes en el dispositivo y el reconocimiento facial.

Para esto también del lado de la vista se diseñó de tal forma que el usuario pueda ir tomando las fotos y asignado a una imagen pueda verla en tiempo real y así contabilizar las fotos que lleva tomadas.

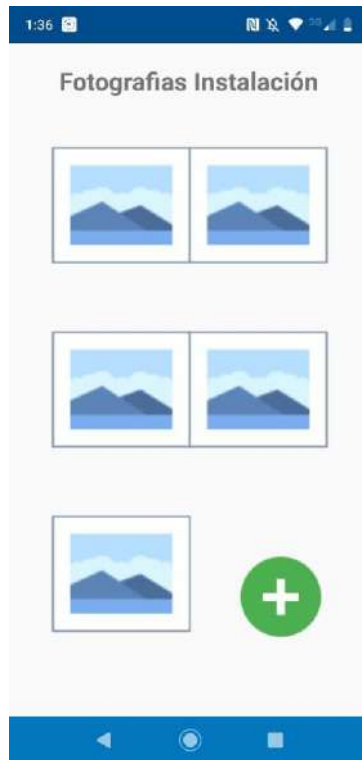


Ilustración 18 Pantalla fotos de instalación

```

magicalPermissions = new MagicalPermissions(this,permissions);
magicalCamera = new MagicalCamera(this,50,magicalPermissions);
seleccionaImg.setOnClickListener((v) -> { magicalCamera.takePhoto(); });

enviarImg.setOnClickListener((v) -> {
    dialogFotos.show();
    try{
        JSONArray jsonArray = new JSONArray();
        JSONObject jsonObject2 = new JSONObject();
        for(int a=0; a<ruta.size();a++){
            Bitmap imagen = BitmapFactory.decodeFile(ruta.get(a));
            JSONObject jsonObject1 = new JSONObject();
            jsonObject1.put( name: "imagen", ConvertirImgString(imagen));
            jsonObject1.put( name: "Nombre", nombre.get(a));
            jsonObject1.put( name: "clv_tecnologia", Util.getClvTec(Util.preferences));
            jsonObject1.put( name: "clv_orden", Util.getClvOrden(Util.preferences));
            jsonArray.put(a,jsonObject1);
        }
        jsonObject2.put( name: "imagenes",jsonArray);
        request.imagen(getApplicationContext(),jsonObject2,activity,dialogFotos);
    }catch (Exception e){
        dialogFotos.dismiss();
        Log.d( tag: "error", e.getMessage());
    }
});

```

Ilustración 19 Programación de uso de la librería Magic cámara

En esta función de igual manera instanciamos cada elemento de la pantalla de las fotografías, creamos un array con las 5 fotos y creamos un Json Array y de esta manera las fotos de esa instalación se envían al servidor donde se encuentra la base de datos.

☰ Orden

Orden No. de contrato

_____ 🔍 _____ 🔍

Contrato	No. de Orden	Estatus
5975-1	6024	P
MARIA VIVIAN GUERRA TRUEBA		
Dirección: FLOR DE AZAFRAN, -M56-L6-N2 C,...		
Trabajo: CLI - Cambio de lugar dentro del D...		
Fecha: 15/10/2019 12:00:00 a. m.		
Turno: MAÑANA		

Los filtros de orden y contrato dada la redundancia filtran por el número de orden o número de contrato, filtra una orden pendiente o una en visita, si la orden ya está marcada con status E (ejecutada) únicamente se verán los datos del cliente más el usuario no podrá ejecutarla nuevamente.

Ilustración 20 Filtros de búsqueda

CAPITULO 5: RESULTADOS

La aplicación móvil se hizo con el fin de dar mejora al sistema web que la empresa ya tenía, de la mano de esta y teniendo grandes mejoras al sistema ya echo.

Al finalizar la realización de la aplicación se montó en un servidor de pruebas y al probar su ejecución con resultados correctos, óptimos y útiles, se pasó a un sistema web de una empresa de telefonía e internet cliente de la empresa SOFTV.

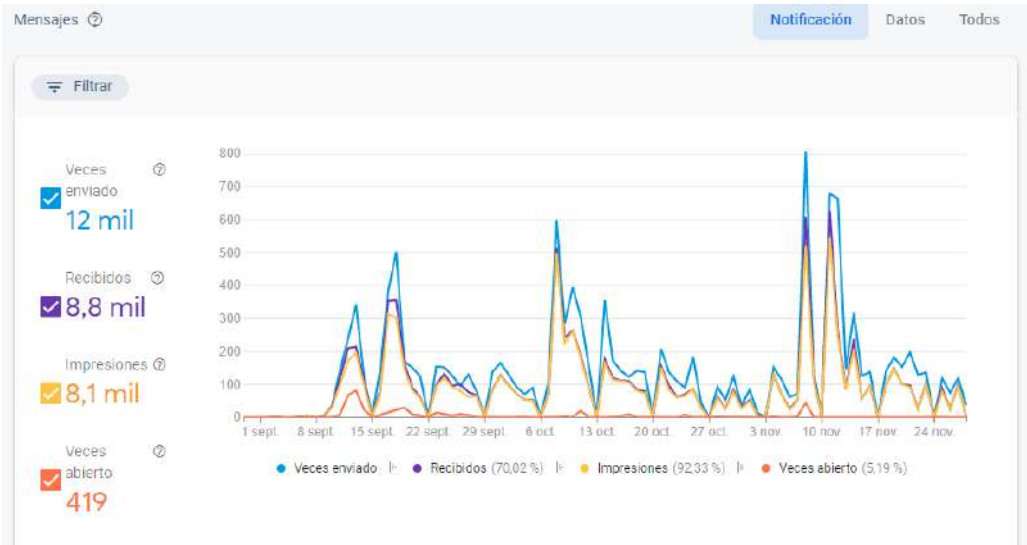


Ilustración 11 Graficas de uso

Actualmente la aplicación móvil opera en más de 5 empresas de telefonía e internet alrededor de la república mexicana así como en empresas dentro de Perú y Colombia.

Objetivo propuesto	Resultado esperado
Elaborar la Aplicación móvil en tiempo y forma	Elaboración de Aplicación Móvil en tiempo de Residencias
Implementar modelo vista controlador	Implementación de modelo vista controlador en cada pantalla
Prueba de errores	Errores modificados optimizando procesos
Modificación de procesos según requerimiento de la empresa	Se modificó o agrego componentes y procesos según como la empresa lo solicitara.

Tabla 1 Objetivos y Resultados

CAPITULO 6: CONCLUSIONES

La creación de esta aplicación móvil dio un gran avance en la empresa ya que se optimizo la forma de ejecutar los trabajos que se realizaban en mesa de control de una manera más rápida e intuitiva.

Fue de gran uso para diferentes empresas dentro de la república mexicana así como en diversos países como Colombia y Perú.

Aún sigue la aplicación en constante mantenimiento según los requerimientos de cada empresa y se está optando por usarla en cada trabajo que realice un técnico en cada domicilio.

En cuanto el aprendizaje se obtuvieron nuevo conocimientos en base de datos en cuanto a manejo de tablas, creación de nuevos procedimientos, control de usuarios, así como creación y modificación de servicios web api rest para consumo de información y mejoramiento de la lógica al momento de programar y de esta manera realizar los cambios de una manera más eficiente y en menor tiempo, se profundizo el aprendizaje en lenguaje de programación java, lenguaje esencial para la realización de este proyecto se usó la programación orientada a objetos así como el lenguaje C# para la creación de servicios web, la optimización y creación de procedimientos almacenados para la base de datos en lenguaje SQL

Se espera que estos nuevos conocimientos y nuevas técnicas de programación ayuden en la creación de nuevos proyectos no solo de dispositivos móviles sino también en proyectos web y así buscar mejora continua.

CAPITULO 7: COMPETENCIAS DESARROLLADAS

1. Aplica el modelo vista controlador para cada diseño de pantalla de la aplicación.
2. Implementa nuevas librerías de localización para obtener una ubicación más exacta del dispositivo móvil.
3. Usa técnicas de responsividad para adaptar el software a más pantallas de diferentes dispositivos.
4. Profundiza más en el aprendizaje de lenguaje java para así implementar nuevas funciones.
5. Busca nuevas librerías que ayudaran en la elaboración de funciones como la cámara móvil, GPS, giroscopio del teléfono móvil e implementarlas en la aplicación.
6. Busca alternativas para mejorar los procedimientos en la base de datos
7. Implementa nuevos procedimientos en la base de datos para el consumo de información.
8. Optimiza los procesos de ejecución de órdenes y servicios en el sistema web.
9. Aplica métodos de investigación para desarrollar la lógica y buscar nuevas alternativas de implementación de nuevas funciones.
10. Utiliza componentes de sistema operativo Android para facilitar el uso aplicación humano.


CAPITULO 8: FUENTES DE INFORMACIÓN

Bibliografía

- Android. (2019). *developer.android.com*. Recuperado el 2019, de <https://developer.android.com/docs?hl=es-419>
- Gutiérrez, F. (2006). *www.fergut.com*. Recuperado el Noviembre de 2019, de <https://www.fergut.com/la-evolucion-de-internet-en-mexico-y-su-impacto-en-el-ambito-educativo-de-1986-a-2006/>
- Hermoza, F. (24 de Marzo de 2017). Recuperado el Noviembre de 2019, de <https://medium.com/@fahedhermoza/notas-del-libro-programaci%C3%B3n-con-android-the-big-nerd-ranch-cap-2-46d9071d9e7e>
- INEGI. (2019). *COMUNICADO DE PRENSA NÚM. 179/19*.
- Perry, J. S. (03 de Marzo de 2012). *ibm.com*. Recuperado el Noviembre de 2019, de <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/index.html>
- stackoverflow. (s.f.). *stackoverflow*. Obtenido de <https://es.stackoverflow.com/>.
- www.electrica.mx*. (Febrero de 2018). Recuperado el Noviembre de 2019, de Electrica: <https://electrica.mx/la-historia-de-las-telecomunicaciones-en-mexico/>

CAPITULO 9: ANEXOS

17. Anexos

	Formato para Asignación de Asesor (a) Interno (a) de Residencias Profesionales por competencias.	Código: TecNM-AC-PO-004-02
	Referencia a la Norma ISO 9001:2015 7.5.1	Revisión: 0
		Página: 1 de 69

Departamento: INGENIERÍAS
No. de Oficio: DIN 104BBB/2019

ASUNTO: **Asesor interno de Residencias Profesionales.**

C. MTRO. JORGE NORBERTO MONDRAGÓN REYES
DOCENTE DEL I.T. DE PABELLÓN DE ARTEAGA
PRESENT E.

Por este conducto informo a usted que ha sido asignado para fungir como Asesor(a) Interno (a) del Proyecto de Residencias Profesionales que a continuación se describe:

a) Nombre del (la) Residente:	MATA VELASCO VICENTE
b) Carrera:	INGENIERÍA EN TIC'S
c) Nombre del Proyecto:	Softv-App
d) Periodo de Realización	AGOSTO_DICIEMBRE 2019
e) Empresa	Sistemas Administrativos para Tv restringija S de RL MI de CV

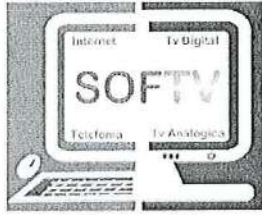
Así mismo, le solicito dar el seguimiento pertinente a la realización del proyecto aplicando los lineamientos establecidos para ello, en el procedimiento del SGC para Residencias Profesionales.

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestro estudiantado.

Atentamente.

MTRO. VICTOR MANUEL VELASCO GALLARDO

c.c.p. Coordinación de Carrera
c.c.p. Expediente



DEPARTAMENTO:
No. DE OFICIO: (4)

AGUASCALIENTES AGS, 03 DE DICIEMBRE 2019

MATI. Humberto Ambriz Delgadillo
Director Del Instituto Tecnológico
De Pabellón De Arteaga.

At'n: MA. Magdalena Cuevas Martinez
Jefe(a) del Departamento de Gestión Tecnológica y Vinculación

PRESENTE.


Por este conducto, me permito informarle que la empresa SOFTV, (Sistemas administrativos de tv restringida), aún no cuenta con sello oficial, por lo cual cada documento será firmado por el responsable correspondiente.

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial saludo.

ATENTAMENTE



JESUS OSWALDO ARCOS OBREGON
GERENTE DESARROLLO

	Formato para Asignación de Asesor (a) Interno (a) de Residencias Profesionales por competencias.	Código: TecNM-AC-PO-004-02
	Referencia a la Norma ISO 9001:2015 7.5.1	Revisión: 0 Página: 1 de 69

Departamento: INGENIERÍAS
No. de Oficio: DIN 104BBB/2019

ASUNTO: **Asesor interno de Residencias Profesionales.**

**C. MTRO. JORGE NORBERTO MONDRAGÓN REYES
DOCENTE DEL I.T. DE PABELLÓN DE ARTEAGA
P R E S E N T E.**

Por este conducto informo a usted que ha sido asignado para fungir como Asesor(a) Interno (a) del Proyecto de Residencias Profesionales que a continuación se describe:

a) Nombre del (la) Residente:	MATA VELASCO VICENTE
b) Carrera:	INGENIERÍA EN TIC'S
c) Nombre del Proyecto:	Softv-App
d) Periodo de Realización	AGOSTO_DICIEMBRE 2019
e) Empresa	Sistemas Administrativos para Tv restringija S de RL MI de CV

Así mismo, le solicito dar el seguimiento pertinente a la realización del proyecto aplicando los lineamientos establecidos para ello, en el procedimiento del SGC para Residencias Profesionales.

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestro estudiantado.

Atentamente.

MTRO. VICTOR MANUEL VELASCO GALLARDO

c.c.p. Coordinación de Carrera
c.c.p. Expediente