**Especificación de requisitos de software y documentación**

**Proyecto**: CiclApp

**Revisión**: 1.14

**Tabla de Contenido**

[**Introducción**](#_joko1li8m0ft) **5**

[Propósito](#_54pcnsx5b4yl) 5

[Alcance](#_12wx4oe39bo4) 5

[Definiciones, acrónimos y abreviaturas](#_vu0wia7wm79x) 5

[Referencias](#_pm7m3yt5nmce) 7

[Descripción general](#_48xtou6gl8so) 7

[Perspectivas del producto](#_fm8alwcvp6x9) 7

[Funciones del producto](#_7oqi2lwykq0d) 7

[Restricciones](#_krzsjzp3ga9j) 7

[Políticas Reguladoras](#_e7nuenkavwiz) 7

[Limitaciones de hardware](#_nh03xqykuitz) 8

[Interfaces con otras aplicaciones](#_rmdyd1h8ulpt) 8

[Requisitos del lenguaje](#_sdkqfptx5t90) 8

[Protocolos señalados](#_jp2r785vcp50) 8

[Requisitos funcionales](#_76ydnelf5nti) 8

[**Servidor EC2**](#_4hchuodu3jfw) **9**

[Objetivo](#_uyzfpn3d0d2) 9

[Funcionamiento](#_ycichff36hbk) 9

[Instalación](#_o81vsctcai9i) 11

[Servidor Apache](#_go0tmwjfels2) 11

[Base de Datos MySQL](#_a7jaz77p9f88) 11

[Administrador phpMyAdmin](#_h6kps2r0melf) 11

[Webmin](#_rs2k13xe1ln) 11

[Python & Libreria MySQL](#_oypqnh1p7xcv) 12

[Scripts Back-End](#_o85mxeltx2mw) 13

[Servidor CiclApp](#_43525zioxxlv) 13

[Base de datos 1](#_l8xoeraxufan) 14

[Base de datos 2](#_gwo2hu8rcmr1) 16

[**Aplicación móvil**](#_qww90old9qhl) **19**

[Objetivo](#_fkype08zqbng) 19

[Lenguaje](#_tzxg7pl5xgra) 19

[Estructura y funcionamiento](#_rgme8zo8zujy) 19

[Código](#_kclt2cjkym34) 20

[Main\_Activity](#_wqhh402qdk8a) 20

[Versions\_Activity](#_t3el086reycz) 21

[**Aplicación WEB**](#_esy5jbuxu2xl) **22**

[Objetivos y Funciones](#_x8ln90ar9y3y) 22

[Lenguajes Utilizados](#_7xd0wi5b6i11) 22

[PHP y MySQL](#_i3m2d2m6ghei) 22

[HTML, CSS y JS](#_dgzqvpqehtht) 26

[LogIn](#_yyacj61kadq3) 26

[Registro](#_51ghj08l06gr) 27

[Index - Inicio](#_wrnh7xit0rsy) 27

[Registros](#_z9kqvzu12vp2) 27

[Match](#_42365o3cfgzb) 28

[Tareas](#_acxc98u68ylj) 28

[Check](#_4e53hzh5vec) 28

[Tabla de ventas](#_h2jdnzk4z76x) 28

[**Bases de datos**](#_a6ap2wvdh09f) **29**

[Lenguaje y funcionamiento](#_31tpowuxz3gc) 29

# 

# 

# 

# 

# Introducción

## Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación móvil y WEB que permitirá administrar y consultar la información de los diversos dispositivos en inventario y vendidos de la empresa **CICLAT** (a partir de ahora, **LA EMPRESA**), utilizada por los diversos empleados de dicha organización.

## Alcance

* Registrar diversos teléfonos móviles
* Comunicarse mediante el protocolo TCP/IP con el servidor
* Consultar los diversos teléfonos móviles registrados en la base de datos
* Eliminar los diversos teléfonos móviles registrados en la base de datos
* Gestionar inventario de repuestos
* Informar estadísticas diversas con el fin de mejorar el rendimiento de LA EMPRESA

CICLAPP permitirá mediante el uso de una aplicación móvil el acceso y registro de los diversos teléfonos que se encuentren en stock o se hayan vendido alguna vez, manteniendo información acerca de todos los cambios que han tenido los mismos, pudiéndose consultar desde la misma aplicación o desde un cliente WEB.

## Definiciones, acrónimos y abreviaturas

**Servidor**: Un servidor es una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.

**Cliente**: El cliente es una aplicación informática o un ordenador que consume un servicio remoto en otro ordenador conocido como servidor, normalmente a través de una red de telecomunicaciones.

**Protocolo de Socket:** Mediante este, dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

**Amazon Web Services (AWS):** Grupo de servicios en la nube (web services) ofrecidos por la compañía Amazon.

**Python**: Lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

**Kotlin**: Kotlin es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de Java y que también puede ser compilado a código fuente de JavaScript.

**Dominio**: Nombre base que agrupa a un conjunto de equipos o dispositivos y que permite proporcionar nombres de equipo más fácilmente recordables en lugar de una dirección IP numérica, Ej: sis-web.com.

**Servidor WEB** : Es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

**URL** : Significa Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

**HTTP** : El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).

**Enlaces** : una referencia en un documento de hipertexto a otro documento o recurso

**Sistema de gestión de Base de Datos:** Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**Base de Datos:** Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

**Aplicación**: Es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.

**MySQL**: Sistema de gestión de base de datos relacional. Comúnmente utilizados en aplicaciones Web.

**PHP (PHP Hypertext Pre-processor):** Lenguaje de programación interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

**Apache**: Servidor de WEB de código abierto

**Dirección IP (Internet Protocol) :** Número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

**Protocolo IP** : El Protocolo de Internet (IP, de sus siglas en inglés Internet Protocol) es un protocolo no orientado a la conexión, usado tanto por el origen como por el destino para la comunicación de estos a través de una red (Internet).

**Browser o Motor de Navegación:** Es una aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet

**DNS**: Es una base de datos distribuida y jerárquica que almacena información asociada de una dirección IP a nombres de dominio en redes como Internet.

**HTML (HyperText Markup Language):** Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

**Javascript**: Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

**Internet**: Es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP

**Encriptar**: Ocultar el contenido de un mensaje mediante un proceso de codificación de manera tal que el acceso al contenido del mensaje sea restringido.

## Referencias

|  |  |
| --- | --- |
| **Título del documento** | **Referencia** |
| Documentos técnicos y guías de AWS | aws.amazon.com/es/whitepapers |
| Estándar IEEE | IEEE |
| Modelo de arquitectura del protocolo TCP/IP | docs.oracle.com/cd/E19957-01/820-2981/ipov-10/ |
| Documentación de Kotlin | kotlinlang.org/docs/reference/ |
| Python 3.8.2rc1 documentation | docs.python.org/3/ |

## Descripción general

### Perspectivas del producto

### Funciones del producto

### Restricciones

#### Políticas Reguladoras

La aplicación se desarrollará mediante software de licencia abierta por lo tanto no se deberá pagar por el uso de : servidor WEB (Amazon LINUX 2), Sistema de Gestión de base de datos (MySQL) y el lenguaje de programación (PHP), por lo tanto, la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

A su vez, el hosteo de la aplicación web y el servidor para la aplicación móvil se desarrollaran utilizando los servicios de máquinas virtuales EC2 que provee **Amazon Web Services**, con un costo aproximado de ARS $10.000 anuales.

#### Limitaciones de hardware

Para esta aplicación móvil y web será necesario un computador servidor en el cual se instalará el servidor WEB LINUX, MySQL, PHP y el servidor Python..

#### Interfaces con otras aplicaciones

Se deberá establecer una comunicación mediante mensajes con la **aplicación móvil**, que permitirá, mediante el uso de un **scanner QR**, agregar. modificar y eliminar teléfonos de la base de datos, llevando un historial de versiones de los mismos. Se debe poder acceder a toda esta información mediante la **aplicación web**.

#### Requisitos del lenguaje

Todas las aplicaciones deben estar presentadas en el lenguaje Español

#### Protocolos señalados

Se utilizarán los protocolos Socket TCP/IP y HTTP.

## Requisitos funcionales

El sistema permitirá:

* **Administrar los datos de los teléfonos\* y versiones desde una aplicación móvil**
  + Registrar teléfonos
  + Consultar teléfonos
  + Registrar versiones
  + Consultar versiones

Datos de los teléfonos: Todo lo especificado en el apartado “Bases de Datos”

* **Administrar teléfonos vendidos**
  + Registrar teléfonos vendidos
  + Consultar teléfonos vendidos
  + Modificar el estado de un teléfono sin vender a vendido
  + Dar de baja teléfonos vendidos
* **Administrar los datos de los teléfonos\* y versiones desde una aplicación WEB**
  + Registrar teléfonos
  + Contabilizar teléfonos por su estado, modelo, marca, etc
  + Consultar teléfonos
  + Emparejar teléfonos según su estado
  + Dar de baja teléfonos
  + Registrar versiones
  + Consultar versiones
  + Dar de baja versiones

Datos de los teléfonos: Todo lo especificado en el apartado “Bases de Datos”

# Servidor EC2

## Objetivo

Registrar datos de los teléfonos brindando soporte a la aplicación móvil y a la aplicación web, pudiendo estas acceder a la base de datos mediante el protocolo TCP/IP.

## Funcionamiento

El servidor va a estar basado en las máquinas virtuales EC2 que provee Amazon Web Services (AWS).

La instancia en cuestión se instaló con el sistema operativo **Amazon Linux 2 AMI (HVM), SSD Volume Type**, y con las siguientes especificaciones:

**Family**: General purpose

**Type**: t2.micro

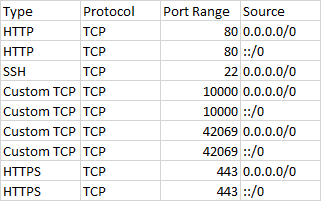
**vCPUs**: 1, 2.5 GHz, Intel Xeon Family

**Memory (GiB)**: 1

**Instance Storage (GB)**: EBS only

**IPv6 Support**: Yes

A su vez, pertenece a un grupo de seguridad con las siguientes características entrantes



Cada vez que se inicia el sistema, junto con el mismo comienzan los procesos del servidor apache y el servidor en Python, para comenzar a recibir las peticiones de la aplicación móvil.

Se puede acceder a la consola de la máquina virtual mediante PuTTY y la llave encriptada dedicada a dicha tarea. El usuario para acceder es ec2-user y la direccion sera:

**ec2-user@ec2-18-216-97-211.us-east-2.compute.amazonaws.com**

## Instalación

Los siguientes comandos se ejecutaron en orden en la consola de la máquina virtual, accediendo desde PuTTY:

### Servidor Apache

[ec2-user ~]$ **sudo yum update -y**

**[ec2-user ~]$ sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd**

**[ec2-user ~]$ sudo service httpd start**

**[ec2-user ~]$ sudo chkconfig httpd on**

### Base de Datos MySQL

[ec2-user ~]$ **sudo service mysqld start**

**[ec2-user ~]$ sudo mysql\_secure\_installation**

### Administrador phpMyAdmin

[ec2-user ~]$ **sudo yum install php72-mbstring.x86\_64 -y**

**[ec2-user ~]$ sudo service httpd restart**

**[ec2-user ~]$ cd /var/www/html**

**[ec2-user html]$ wget** *https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz*

*[ec2-user html]$* ***mkdir phpMyAdmin && tar -xvzf*** *phpMyAdmin-latest-all-languages.tar.gz* ***-C phpMyAdmin --strip-components 1***

***[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz***

***[ec2-user ~]$ sudo service mysqld start***

Una vez hecho todo esto, queda configurado el servidor Apache, MySQL y phpMyAdmin

### Webmin

[ec2-user ~]$ **vim /etc/yum.repos.d/webmin.repo**

[ec2-user ~]$ **wget http://www.webmin.com/jcameron-key.asc**

[ec2-user ~]$ **rpm -import jcameron-key.asc**

[ec2-user ~]$ **yum install webmin**

### Python & Libreria MySQL

[ec2-user ~]$ **sudo yum install python36**

[ec2-user ~]$ **pip install mysql**

## 

## Scripts Back-End

### Servidor CiclApp

El Script está alojado en la ruta /home/ec2-user/server/main.py. Su objetivo es escuchar las peticiones y recibir mensajes mediante el protocolo TCP/IP, respondiendo con diversos mensajes según la entrada recibida.

1. import socket
2. from threading import \*
3. import database
4. import time
5. class client(Thread):
6. def \_\_init\_\_(self, socket, address):
7. Thread.\_\_init\_\_(self)
8. self.sock = socket
9. self.addr = address
10. self.start()
11. def run(self):
12. msg = self.sock.recv(1024).decode()
13. if msg == '':
14. print('Cliente cerrado')
15. else:
16. print('Mensaje recibido:', msg)
17. dividido = msg.split(",")
18. if len(dividido) == 1:
19. a = database.buscar\_imei\_ultimo(("".join(dividido)))
20. if a is False:
21. mensaje = str.encode("NewPhone")
22. else:
23. mensaje = str.encode(",".join(a))
24. self.sock.send(mensaje)
25. self.sock.close()
26. elif len(dividido) == 2:
27. print(0)
28. lista = database.buscar\_imei(dividido[1])
29. for a in range(len(lista)):
30. lista[a] = ",".join(lista[a])
31. lista = "&".join(lista)
32. a = str.encode(lista)
33. self.sock.send(a)
34. self.sock.close()
35. print(a)
36. else:
37. database.data\_entry(dividido)
38. a = str.encode("Ok")
39. self.sock.send(a)
40. self.sock.close()
41. serversocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)
42. host = "0.0.0.0"
43. port = 42069
44. serversocket.bind((host, port))
45. serversocket.listen(5)
46. print('Servidor iniciado, esperando...')
47. while 1:
48. clientsocket, address = serversocket.accept()
49. print('Anadido')
50. client(clientsocket, address)

### Base de datos 1

1. import sqlite3
2. import csv
3. import os.path
4. import mysql.connector
5. import database2
6. from datetime import datetime

9. def conector():
10. mydb = mysql.connector.connect(
11. host = "localhost",
12. user = "root",
13. passwd = "administrador",
14. database = "database"
15. )
16. mycursor = mydb.cursor()
18. return mydb, mycursor

21. def obtener\_ultima\_id():
22. conexion, cursor = conector()
23. cursor.execute("SELECT \* FROM CELULAR")
24. celulares = cursor.fetchall()
25. ultima = 0
26. for a in celulares:
27. if int(a[0]) > ultima:
28. ultima = int(a[0])
30. conexion.close()
31. return ultima

34. def data\_entry(telefono):
35. conexion, cursor = conector()
36. imei = telefono[0]
37. if buscar\_imei\_ultimo(imei) is False:
38. version = 1
39. idtel = str(obtener\_ultima\_id()+1)
40. else:
41. version = str(int(buscar\_imei\_ultimo(imei)[-1]) + 1)
42. idtel = buscar\_imei\_ultimo(imei)[0]
43. telefono.insert(0, idtel)
44. telefono.append(version)
45. telefono[4] = str(datetime.now()).split(" ")[0]
47. sentencia = '''INSERT INTO CELULAR
48. (ID, IMEI,Marca,Modelo,Fecha,Revisor,GB,Color,Estetica,Carcasa,CamaraTrasera,CamaraDelantera,PinCarga,Auriculares,ParlanteFrontal,ParlanteTrasero,SensorProx,Bateria,BateriaPorcentaje,Wifi,Bluetooth,Vidrio,Modulo,Tras$
49. VALUES
50. (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'''
51. cursor.execute(sentencia, telefono)
52. conexion.commit()
53. conexion.close()
55. database2.ultima\_version(telefono)

58. def buscar\_imei(imei):
59. #Busca todas las versiones de un telefono por su IMEI
60. conexion, cursor = conector()
61. cursor.execute("SELECT \* FROM CELULAR")
62. celulares = cursor.fetchall()
63. lista = []
64. for a in celulares:
65. if a[1] == str(imei):
66. lista.append(a)
67. conexion.close()
68. return lista

71. def buscar\_imei\_ultimo(imei):
72. #Busca la ultima version de un telefono segun su imei
73. todos = buscar\_imei("".join(imei))
74. if len(todos) == 0:
75. return False
76. retornar = todos[0]
77. for a in todos:
78. if int(a[-1]) > int(retornar[-1]):
79. retornar = a
80. return retornar

### Base de datos 2

1. import sqlite3
2. import csv
3. import os.path
4. import mysql.connector
6. def conector():
7. mydb = mysql.connector.connect(
8. host = "localhost",
9. user = "root",
10. passwd = "administrador",
11. database = "ultimavers\_db"
12. )
13. mycursor = mydb.cursor()
15. return mydb, mycursor

18. def ultima\_version(telefono):
19. conexion, cursor = conector()
20. eliminar(telefono[1])
22. sentencia = '''INSERT INTO CELULAR
23. (ID, IMEI,Marca,Modelo,Fecha,Revisor,GB,Color,Estetica,Carcasa,CamaraTrasera,CamaraDelantera,PinCarga,Auriculares,ParlanteFrontal,ParlanteTrasero,SensorProx,Bateria,BateriaPorcentaje,Wifi,Bluetooth,Vidrio,Modulo,Tras$
24. VALUES
25. (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'''
26. cursor.execute(sentencia, telefono)
27. conexion.commit()
28. conexion.close()

31. def eliminar(imei):
32. conexion, cursor = conector()
33. cursor.execute("DELETE FROM CELULAR WHERE IMEI = '" + imei + "'")
34. conexion.commit()
35. conexion.close()

38. def mostrar\_tabla():
39. conexion, cursor = conector()
40. cursor.execute("SELECT \* FROM CELULAR")
41. celulares = cursor.fetchall()
42. for a in celulares:
43. print(a)
44. conexion.close()
46. def vaciar\_tabla():
47. conexion, cursor = conector()
48. cursor.execute("TRUNCATE TABLE CELULAR")
49. conexion.close

52. def obtener\_ultimas\_versiones():
53. mydb = mysql.connector.connect(
54. host="localhost",
55. user="root",
56. passwd="administrador",
57. database="database"
58. )
59. mycursor = mydb.cursor()
61. mycursor.execute("SELECT \* FROM CELULAR")
62. celulares = mycursor.fetchall()
64. ultimas\_versiones = []
66. # Primero obtenemos la id mas grande
67. mayor = "0"
68. for a in celulares:
69. if int(a[0]) > int(mayor):
70. mayor = a[0]
72. # Luego recorremos id por id buscando el mayor
73. for a in range(int(mayor)):
75. # Por cada id va a hacer esto (agarrar y devolver en el arreglo todas las versiones
76. arreglo = []
77. for b in celulares:
78. if b[0] == str(a):
79. arreglo.append(b)
81. # Aca agarro todas las versiones y guardo la ultima
82. if len(arreglo) != 0:
83. telefono\_appendear = arreglo[0]
84. for a in arreglo:
85. if int(a[-1]) > int(telefono\_appendear[-1]):
86. telefono\_appendear = a
87. ultimas\_versiones.append(telefono\_appendear)
89. mydb.close()
90. return ultimas\_versiones

93. def actualizador():
94. vaciar\_tabla()
95. telefonos\_ultimos = obtener\_ultimas\_versiones()
96. for a in telefonos\_ultimos:
97. ultima\_version(a)


101. if \_\_name\_\_ == "\_\_main\_\_":
102. print("1- Mostrar tabla\n2- Vaciar tabla\n3- Actualizar tabla")
104. opc = input("Seleccione una opcion: ")
105. if opc == "1":
106. mostrar\_tabla()
107. if opc == "2":
108. vaciar\_tabla()
109. if opc == "3":
110. actualizador()

## 

# Aplicación móvil

## Objetivo

Lograr optimizar la administración de información de los registros de celulares de forma más eficaz y eficiente

## Lenguaje

En esta aplicación hemos utilizado un lenguaje de programación fuertemente tipado llamado Kotlin desarrollado por JetBrains (los creadores de IntelliJ IDEA).

Ha sido fuertemente influenciado por lenguajes como Groovy, Scala o C#.

Permite generar código para la JVM (máquina virtual de Java 6) Javascript y en las últimas versiones también ejecutables nativos.

Kotlin es ideal para desarrollos basados en JVM y para hacer aplicaciones de Android.

Kotlin es un lenguaje moderno y una buena elección en general, ameno de escribir y operar con el, acompañado de un conjunto de herramientas muy maduro y todo el apoyo de JetBrains e IntelliJ, Gradle etc.

## Estructura y funcionamiento

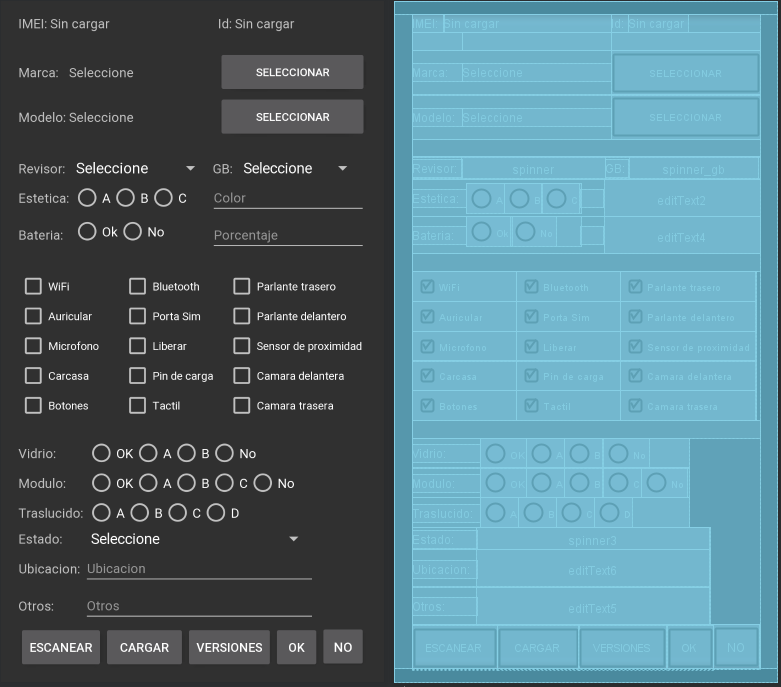
Para la realización de la aplicación se han escrito **más de 2000 líneas de código** utilizando el lenguaje **Kotlin,** y las diversas librerías que este y android ofrecen.

## 

## 

## Código

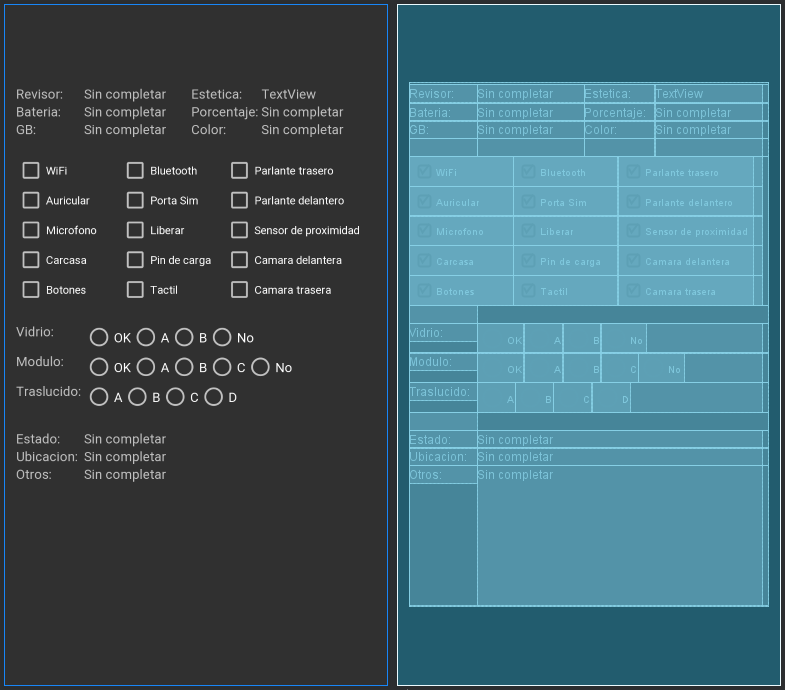
### Main\_Activity



### 

### 

### Versions\_Activity



# Aplicación WEB

## Objetivos y Funciones

La principal función de la Parte WEB de este sistema es cumplir con:

* Facil acceso a los datos
* Facil manejo de datos e información
* Gestionar los telefonos
* Busquedas rapidas y precisas para facilitar el trabajo
* Buen diseño visial para los datos de forma sencilla
* Llevar registro
* Otras funciones como Tabla de tareas pendientes, etc

Con el cumplimiento de estos puntos y junto a la implementación de la APP, estas se complementan perfectamente logrando el objetivo deseado.

## Lenguajes Utilizados

Para la implementación y uso de las bases de datos se utilizo lenguaje PHP y MySQL para generar las Querys en la aplicación web.

Tambien se manejaron sintaxis de CSS, junto con implementaciones HTML y JS(JavaScript)

### PHP y MySQL

La forma de manejo de datos fue totalmente hecha con querys generadas mediante lineas de codigo hechas en PHP. Con estas, se pudo generar consultas de forma dinamica, modificar, agregar, eliminar y contabilizar los datos de las bases de respaldo y la de ultima version de forma segura y eficiente. Esto se realiza mediante Query’s de MySQL con datos especificos que se detallan a continuacion.

* ***Selección y ordenamiento en tabla para visualizar los datos***

1. <?php
2. session\_start();
4. include('conexionmysql/conectar.php');
6. $query="SELECT \* FROM CELULAR";
7. $resultado=$conexion->query($query);
8. while($row=$resultado->fetch\_assoc()){
10. ?>
11. <table class="table table-hover table-bordered table-dark" id="dataTable" width="100%" cellspacing="0">
12. <thead >
13. <tr>
14. <th class="m-0 font-weight-bold text-white">ID</th>
15. <th class="m-0 font-weight-bold text-white"><center>IMEI</center></th>
16. <th class="m-0 font-weight-bold text-white">Marca</th>
17. <th class="m-0 font-weight-bold text-white">Modelo</th>
18. <th class="m-0 font-weight-bold text-white">Fecha</th>
19. <th class="m-0 font-weight-bold text-white">Estado</th>
20. <th class="m-0 font-weight-bold text-white">Revisor</th>
21. <th class="m-0 font-weight-bold text-white">GB</th>
22. <th class="m-0 font-weight-bold text-white">Color</th>
23. <th class="m-0 font-weight-bold text-white">Estetica</th>
24. <th class="m-0 font-weight-bold text-white">Carcasa</th>
25. <th class="m-0 font-weight-bold text-white">CamaraTrasera</th>
26. <th class="m-0 font-weight-bold text-white">CamaraDelantera</th>
27. <th class="m-0 font-weight-bold text-white">PinCarga</th>
28. <th class="m-0 font-weight-bold text-white">Auriculares</th>
29. <th class="m-0 font-weight-bold text-white">ParlanteFrontal</th>
30. <th class="m-0 font-weight-bold text-white">SensorProx</th>
31. <th class="m-0 font-weight-bold text-white">Bateria</th>
32. <th class="m-0 font-weight-bold text-white">BateriaPorcentaje</th>
33. <th class="m-0 font-weight-bold text-white">Wifi</th>
34. <th class="m-0 font-weight-bold text-white">Bluetooth</th>
35. <th class="m-0 font-weight-bold text-white">Vidrio</th>
36. <th class="m-0 font-weight-bold text-white">Modulo</th>
37. <th class="m-0 font-weight-bold text-white">Traslucido</th>
38. <th class="m-0 font-weight-bold text-white" width="60px">Otros</th>
39. <th class="m-0 font-weight-bold text-white">Lugar</th>
40. <th class="m-0 font-weight-bold text-white">Liberar</th>
41. <th class="m-0 font-weight-bold text-white">PortaSim</th>
42. <th class="m-0 font-weight-bold text-white">Microfono</th>
43. <th class="m-0 font-weight-bold text-white">Botones</th>
44. <th class="m-0 font-weight-bold text-white">Tactil</th>
45. <th class="m-0 font-weight-bold text-white">version</th>
47. </tr>
48. </thead>
49. <tfoot>
50. <tr>
51. <th>ID</th>
52. <th><center>IMEI</center></th>
53. <th>Marca</th>
54. <th>Modelo</th>
55. <th>Fecha</th>
56. <th>Estado</th>
57. <th>Revisor</th>
58. <th>GB</th>
59. <th>Color</th>
60. <th>Estetica</th>
61. <th>Carcasa</th>
62. <th>CamaraTrasera</th>
63. <th>CamaraDelantera</th>
64. <th>PinCarga</th>
65. <th>Auriculares</th>
66. <th>ParlanteFrontal</th>
67. <th>SensorProx</th>
68. <th>Bateria</th>
69. <th>BateriaPorcentaje</th>
70. <th>Wifi</th>
71. <th>Bluetooth</th>
72. <th>Vidrio</th>
73. <th>Modulo</th>
74. <th>Traslucido</th>
75. <th>Otros</th>
76. <th>Lugar</th>
77. <th>Liberar</th>
78. <th>PortaSim</th>
79. <th>Microfono</th>
80. <th>Botones</th>
81. <th>Tactil</th>
82. <th>version</th>

85. </tr>
86. </tfoot>
87. <tbody>
88. <?php
89. include('conexionmysql/conectar.php');
91. $query="SELECT \* FROM CELULAR";
92. $resultado=$conexion->query($query);
93. while($row=$resultado->fetch\_assoc()){
94. ?>
95. <tr>
96. <td><center><?php echo $row['ID'] ?></center></td>
97. <td><center><?php echo $row['IMEI'] ?></center></td>
98. <td><center><?php echo $row['Marca'] ?></center></td>
99. <td><center><?php echo $row['Modelo'] ?></center></td>
100. <td><center><?php echo $row['Fecha'] ?></center></td>
101. <td><center><strong><?php echo $row['Estado'] ?></strong></center></td>
102. <td><center><?php echo $row['Revisor'] ?></center></td>
103. <td><center><?php echo $row['GB'] ?></center></td>
104. <td><center><?php echo $row['Color'] ?></center></td>
105. <td><center><?php echo $row['Estetica'] ?></center></td>
106. <td><center><?php echo $row['Carcasa'] ?></center></td>
107. <td><center><?php echo $row['CamaraTrasera'] ?></center></td>
108. <td><center><?php echo $row['CamaraDelantera'] ?></center></td>
109. <td><center><?php echo $row['PinCarga'] ?></center></td>
110. <td><center><?php echo $row['Auriculares'] ?></center></td>
111. <td><center><?php echo $row['ParlanteFrontal'] ?></center></td>
112. <td><center><?php echo $row['SensorProx'] ?></center></td>
113. <td><center><?php echo $row['Bateria'] ?></center></td>
114. <td><center><?php echo $row['BateriaPorcentaje'] ?></center></td>
115. <td><center><?php echo $row['Wifi'] ?></center></td>
116. <td><center><?php echo $row['Bluetooth'] ?></center></td>
117. <td><center><?php echo $row['Vidrio'] ?></center></td>
118. <td><center><?php echo $row['Modulo'] ?></center></td>
119. <td><center><?php echo $row['Traslucido'] ?></center></td>
120. <td><center><?php echo $row['Otros'] ?></center></td>
121. <td><center><?php echo $row['Lugar'] ?></center></td>
122. <td><center><?php echo $row['Liberar'] ?></center></td>
123. <td><center><?php echo $row['PortaSim'] ?></center></td>
124. <td><center><?php echo $row['Microfono'] ?></center></td>
125. <td><center><?php echo $row['Botones'] ?></center></td>
126. <td><center><?php echo $row['Tactil'] ?></center></td>
127. <td><center><?php echo $row['version'] ?></center></td>

130. <?php
131. }
132. ?>
133. </tr>
134. </tbody>
135. </table>

Eventualmente, se le agregaron los botones de Modificar, Eliminar, y Marcado como Vendido, usando PHP y otros archivos con distintas Querys

* ***Actualizacion de los datos***

1. <?php
2. $sql = "UPDATE `CELULAR` SET ID='$id',IMEI='$imei',Marca='$Marca',Modelo='$Modelo',Fecha='$Fecha',Revisor='$Revisor',GB='$GB',Color='$Color',Estetica='$Estetica',Carcasa='$Carcasa',CamaraTrasera='$CamaraTrasera',CamaraDelantera='$CamaraDelantera',PinCarga='$PinCarga',Auriculares='$Auriculares',ParlanteFrontal='$ParlanteFrontal',ParlanteTrasero='$ParlanteTrasero',SensorProx='$SensorProx',Bateria='$Bateria',BateriaPorcentaje='$BateriaPorcentaje',Wifi='$Wifi',Bluetooth='$Bluetooth',Vidrio='$Vidrio',Modulo='$Modulo',Traslucido='$Traslucido',Otros='$Otros',Estado='$Estado',Lugar='$Lugar',Liberar='$Liberar',PortaSim='$PortaSim',Microfono='$Microfono',Botones='$Botones',Tactil='$Tactil',version='$version' WHERE id='$idmod' AND imei='$imeimod' AND version='$versionmod'";
3. if ($conn->query($sql) === TRUE) {
4. header('Location: http://18.216.97.211/tables2.php');
5. } else {
6. echo "Error: " . $sql . "<br>" . $conn->error;
7. }
8. $conn->close();
9. ?>

* ***Otros manejos de datos***

Para eliminar y agregar otros datos se utilizaron métodos muy similares en cuanto a querys, apuntando a las mismas variables.

**Entre ellas:**

* DELETE
* SELECT
* INSERT
* Otros…

### HTML, CSS y JS

Para la implementación del FrontEnd, se uso una plantilla base de Bootstap 4, llamada SB Admin 2, y sus archivos CSS y JS correspondientes para sus diseños y animaciones, ademas de la estructura base HTML. Al igual que la estructura HTML, los archivos CSS y JS en gran medida fueron modificados para la prefencia de este trabajo, facilitando el despliegue de los datos sobre tablas creadas desde el principio y a su vez trabajando con el BackEnd de PHP y MySQL para complementarse y generar un resultado vistoso, funcional y optimo.

Se utilizo framework propio de Google para generar las graficas visuales del Dashboard o Inicio en la pagina, que se conectan y actualizan con la base de datos en tiempo real

#### 

#### LogIn

El ingreso a la pagina WEB y sus funciones comienza en una pantalla de LOG IN, donde se debe tener un usuario y contraseña que se haya verificado anteriormente y cargado en la base de USUARIOS.

De no ser el caso se rechaza la solicitud de ingreso al sitio como medida de seguridad basica.

Si ya dispone de un usuario validado, se genera una sesión durante el periodo que se utilize la plataforma. Esta sesión se valida en cada sección de la plataforma como medida de seguridad para evitar entradas sin validación de usuario previa, como ya se aclaró en la sección de PHP y MySQL.

#### Registro

En caso de no tener un usuario validado, puede hacer una solicitud para generar un nuevo usuario. Se llena un formulario y se envia a la tabla de USUARIOS-NUEVOS. Donde debe ser Revisado por alguna autoridad de la empresa desde phpMyAdmin (sujeto a modificacion para mayor comodidad) y cargado en la tabla de Usuarios, con lo que quedaria hanilitado para el acceso a la plataforma y sus funciones.

#### Index - Inicio

En la pagina de inicio o DashBoard, se muestra toda la informacion relevante, graficas de datos y avisos de importancia para el usuario. Este inicio se encuentra personalizado para el usuario que ingrese con su imagen, nombre y datos correspondientes a la base de datos de USUARIO.

En las graficas se puede ver la subdivicion de toda la base de datos por “Marcas” y dentro de las opciones de la grafuca otros filtros de “Modelos” por “Marca”, con porcentajes de representacion en la base de datos para un mayor control del stock general.

A su izquierda, la grafica de ventas que muestra en una grafica lineal de area, por mes, la cantidad de ventas registradas en la base de datos de VENTAS.

#### Registros

En esta Pantalla se encuentra uno de los objetivos principales del programa que es ,a gestion de los datos. En una tabla bajo a un buscador, esta la tabla de registros donde podemos ver todos los telefonos registrados hasta el momento.

Cada registro tiene sus 3 botones de funcion: Editar, Eliminar y Vendido. Con estos el usuario que quiera realizar alguna tarea sobre un regitro en particular, puede usar los botones de funcion. Nuevamente se genera una solicitud de sesion para evitar accesos no deseados a la base sobre los datos sensibles del servidor.

Al final de la tabla se encuentra un formulario de carga manual en caso de ser necesario. Este ya cuenta con el nombre de usuario en el campo de Revisor para facilitar la carga. También se genera una validacion de seguridad.

Se dispone de un botón de descarga de toda la base de datos en formato excel para usarse con otros programas si así se desea, se debe revisar el formado de las tablas dentro del excel para la columna de IMEI por asuntos de compatibilidad.

#### Match

Esta pagina es encargada de generar 2 tablas con un filtrado especifico sobre una busqueda realizada en la misma pagina, y asi ver que telefonos pueden ser otenciales parejas para ser arreglados y conseguir un teléfono funcional. Estas tablas brindan todos los datos necesarios para discernir si un telefono vale la pena ser arreglado o no. En caso de serlo, se lo ubica por su registro y si este es reparado o usado para repuestos, se cambia su estado para que se registre

#### Tareas

La finalidad de esta pagina es poder generar tareas o solicitudes que todos los usuarios pueden ver y cargar. En la carga segun el usuario que ingreso sera el nombre que se registrara en la base de datos de TAREAS.

Cuenta con un boton de confirmacion cuando una tarea ya a sido cumplida o se quiere eliminar directamente.

#### Check

A esta pantalla se accede desde el Index, en la parte de Telefonos mal cargados. El boton “Revisar” genera una consulta especifica sobre toda la base de datos para verificar que registros estan mal cargados y puedan generar conflictos sobre la plataforma y sus funciones. Estos registros mal cargados se encuentran desplegados en una tabla para su acceso y los datos importantes para solucionar resaltados para ser resueltos, tales comp: Estado, Revisor, Ubicacion, Marca y Modelo.

Con estos datos se puede saber todo lo posible para resolver un registro mal cargado.

#### Tabla de ventas

Como su nombre lo dice, se muestran todos los registros donde su estado sea “Vendido” o “Listo para vender”, quedando registrado en esta tabla y sumado en la gráfica del index para visualizar mes a mes la cantidad de ventas en base a los registros de esta tabla.

# Bases de datos

## Lenguaje y funcionamiento

Para la implementación y uso de las bases de datos se utilizo lenguaje PHP y MySQL para generar las Querys en la aplicación web.

* ***Archivo de conexion:***  conectar.php

1. <?php
2. session\_start();
3. $user="root";
4. $pass="administrador";
5. $db="ultimavers\_db";
6. $conexion=new mysqli("localhost",$user,$pass,$db);
8. ?>

Con este archivo la conexión queda establecida entre la página y la base de datos con las últimas versiones de los teléfonos.

Se usa este archivo en el 80% de los demas archivos para gestión de datos, además de contar con el sistema de seguridad por sesiones con validación inicial.