

**INSTITUTO TECNOLÓGICO DEL VALLE DE OAXACA**

## APLICACIÓN WEB PARA LA CENTRALIZACIÓN DE INFORMACIÓN DEL CENTRO DE ASESORÍA PROFESIONAL PAULO FREIRE

REPORTE FINAL DE RESIDENCIA PROFESIONAL QUE PRESENTAN:

Luis Enrique Hernández Hernández

Reyes Iván Gonzáles Paredes

# Como requisito para acreditar la Residencia Profesional de la Licenciatura en:

**INGENIERÍA INFORMÁTICA**

Ex Hacienda de Nazareno, Xoxocotlán, Oaxaca.

Agosto 2021

**AGRADECIMIENTOS**

## RESUMEN

El desarrollo de aplicaciones móviles ha tenido un importante crecimiento en los últimos años, ya que brindan una amplia utilidad a las personas tales como, medios de información, acceder a diferentes servicios como compras en línea, pagos de servicios entre otros.

En el ámbito móvil existen tres tipos de aplicaciones móviles: nativas, web e hibridas, todas presentan una amplia variedad de ventajas e inconvenientes en función del tipo de proyecto a desarrollar. Las aplicaciones nativas se enfocan en un solo sistema operativo móvil, estas acceden a la totalidad de los componentes del dispositivo, generalmente requieren mayores recursos para su desarrollo. Las aplicaciones web logran que se asemejen a las nativas, pero estas no acceden a todos los componentes del dispositivo y esto resulta menos fiables en algunos entornos. Por ultimo las aplicaciones hibridas toma las funcionalidades más relevantes de las aplicaciones nativas y web, haciendo de esto una aplicación mucho más funcional.

Esta aplicación web se desarrolló usando un marco de trabajo ( framework) que brinda una interfaz de ayuda para mejorar la eficiencia, optimización y fiabilidad del sistema.

Por lo expuesto anteriormente el centro de asesorías PAULO FREIRE propuso el proyecto de una aplicación web el cual está enfocado a los docentes, administradores y estudiantes del CAP ; en el presente documento se presenta los objetivos a lograr, metodología aplicada, resultados, conclusiones, recomendaciones, experiencia adquirida y competencias alcanzadas.

**INDICE GENERAL**

[CAPITULO I. GENERALIDADES DEL PROYECTO 24](#_bookmark0)

* 1. [Introducción](#_bookmark1) [24](#_bookmark1)
  2. [Descripción de la empresa](#_bookmark2) [26](#_bookmark2)
     1. [Objetivos](#_bookmark3) [26](#_bookmark3)
     2. [Misión](#_bookmark4) [27](#_bookmark4)
     3. [Valores](#_bookmark5) [27](#_bookmark5)
     4. [Lugar](#_bookmark6) [28](#_bookmark6)
  3. [Problema a resolver](#_bookmark7) [29](#_bookmark7)
  4. [Objetivos](#_bookmark8) [30](#_bookmark8)
     1. [Objetivo general](#_bookmark9) [30](#_bookmark9)
     2. [Objetivos específicos.](#_bookmark10) [30](#_bookmark10)
  5. [Justificación](#_bookmark11) [31](#_bookmark11)

[CAPITULO II. MARCO TEÓRICO 32](#_bookmark12)

* 1. [Fundamentación Teórica](#_bookmark13) [32](#_bookmark13)
     1. [Tecnologías de la información y comunicación](#_bookmark14) [32](#_bookmark14)
     2. [Internet](#_bookmark15) [33](#_bookmark15)
     3. [Tecnología móvil](#_bookmark16) [34](#_bookmark16)
     4. [Sistemas operativos](#_bookmark17) [35](#_bookmark17)
        1. [Sistemas operativos de dispositivos móviles](#_bookmark18) [37](#_bookmark18)
        2. [Sistemas operativo Android](#_bookmark19) [38](#_bookmark19)
        3. [Sistemas operativo iOS](#_bookmark20) [39](#_bookmark20)
     5. [Base de datos](#_bookmark21) [41](#_bookmark21)
        1. [Aplicaciones de los sistemas de bases de datos](#_bookmark22) [41](#_bookmark22)
        2. [Tipos de bases de datos](#_bookmark23) [42](#_bookmark23)
           1. [Modelo relacional](#_bookmark24) [42](#_bookmark24)

[Ventajas y desventajas de las bases de datos relacionales](#_bookmark25) [44](#_bookmark25)

* + - * 1. [Modelo no relacional](#_bookmark26) [45](#_bookmark26)

[2.1.5.2.2.2 Ventajas y desventajas de las bases de datos no relacionales](#_bookmark27) [46](#_bookmark27)

* 1. [Desarrollo de aplicaciones móviles](#_bookmark29) [50](#_bookmark29)
     1. [Aplicaciones móviles multiplataforma](#_bookmark30) [51](#_bookmark30)
        1. [Aplicaciones web](#_bookmark31) [52](#_bookmark31)
        2. [Aplicaciones nativas](#_bookmark32) [53](#_bookmark32)
        3. [Aplicaciones hibridas](#_bookmark33) [55](#_bookmark33)
  2. [Framework para el desarrollo móvil](#_bookmark34) [56](#_bookmark34)
     1. [Frameworks](#_bookmark35) [58](#_bookmark35)
        1. [React Native](#_bookmark36) [58](#_bookmark36)
        2. [Ionic](#_bookmark37) [59](#_bookmark37)
        3. [Xamarin](#_bookmark38) [60](#_bookmark38)
  3. [Tecnología implementada](#_bookmark39) [63](#_bookmark39)
     1. [Android Studio](#_bookmark40) [63](#_bookmark40)
     2. [Dart](#_bookmark41) [64](#_bookmark41)
     3. [Flutter](#_bookmark42) [65](#_bookmark42)
        1. [Manejadores de estado flutter](#_bookmark43) [67](#_bookmark43)
           1. [BLoc](#_bookmark44) [67](#_bookmark44)
           2. [Provider](#_bookmark45) [69](#_bookmark45)
     4. [Visual Studio Code](#_bookmark46) [70](#_bookmark46)
     5. [Material Design](#_bookmark47) [71](#_bookmark47)
     6. [Postman](#_bookmark48) [72](#_bookmark48)
     7. [Xcode](#_bookmark49) [73](#_bookmark49)
     8. [Sistemas gestores de bases de datos](#_bookmark50) [73](#_bookmark50)
        1. [Gestores de bases de datos más utilizados actualmente](#_bookmark51) [74](#_bookmark51)
           1. [MySQL](#_bookmark52) [74](#_bookmark52)

[Características MySQL](#_bookmark53) [75](#_bookmark53)

* + - * 1. [MongoDB](#_bookmark54) [75](#_bookmark54)

[2.4.8.1.2.2 Características de MongoDB](#_bookmark55) [76](#_bookmark55)

* + - 1. [Diferencias MySQL y MongoDB](#_bookmark56) [76](#_bookmark56)
      2. [Descripción del sistema de base de datos seleccionado](#_bookmark57) [81](#_bookmark57)
         1. [MongoDB Atlas](#_bookmark58) [82](#_bookmark58)
         2. [MongoDB Compass](#_bookmark59) [83](#_bookmark59)
    1. [Heroku](#_bookmark60) [83](#_bookmark60)
    2. [Node JS](#_bookmark61) [84](#_bookmark61)
       1. [Paquetes Node JS](#_bookmark62) [85](#_bookmark62)
    3. [Rest Server](#_bookmark64) [87](#_bookmark64)
  1. [Metodología](#_bookmark66) [89](#_bookmark66)
     1. [Metodología Scrum](#_bookmark67) [90](#_bookmark67)
  2. [Trabajos relacionados](#_bookmark68) [96](#_bookmark68)
     1. [Information System for Providing Food Services Based on Mobile Application Using Flutter Framework](#_bookmark69) [96](#_bookmark69)
     2. [Desarrollo de aplicaciones móviles multiplataforma con Flutter](#_bookmark70) [96](#_bookmark70)
     3. [Conceptualización y prototipado de una aplicación móvil para padres de niños con síndrome de Down](#_bookmark71) [97](#_bookmark71)

[CAPITULO III. DESARROLLO 98](#_bookmark72)

* 1. [Fase de planificación](#_bookmark73) [98](#_bookmark73)
     1. [Formación del equipo](#_bookmark74) [98](#_bookmark74)
     2. [Definición de los requerimientos funcionales](#_bookmark76) [102](#_bookmark76)
     3. [Definición de los requerimientos no funcionales](#_bookmark77) [103](#_bookmark77)
     4. [Restricciones](#_bookmark78) [105](#_bookmark78)
     5. [Definición de los objetos del producto (Product Backlog)](#_bookmark79) [106](#_bookmark79)
  2. [Fase de diseño](#_bookmark81) [109](#_bookmark81)
     1. [Prototipos](#_bookmark82) [109](#_bookmark82)
     2. [Diagrama entidad-relación](#_bookmark83) [114](#_bookmark83)
     3. [Planificación y estimación](#_bookmark84) [130](#_bookmark84)
        1. [Historias de usuarios](#_bookmark85) [131](#_bookmark85)
           1. [Historias de usuarios inicio de sesión](#_bookmark86) [131](#_bookmark86)
           2. [Historias de usuarios perfil de usuario](#_bookmark89) [133](#_bookmark89)
           3. [Historias de usuarios pantalla principal](#_bookmark91) [134](#_bookmark91)
        2. [Desarrollo de casos de usos](#_bookmark95) [137](#_bookmark95)
           1. [Casos de usos inicio de sesión](#_bookmark97) [137](#_bookmark97)
           2. [Casos de usos perfil de usuario](#_bookmark104) [143](#_bookmark104)
           3. [Casos de usos pantalla principal](#_bookmark111) [149](#_bookmark111)
        3. [Diagramas de secuencias](#_bookmark118) [155](#_bookmark118)
           1. [Diagrama de secuencia inicio de sesión joven](#_bookmark119) [156](#_bookmark119)
           2. [Diagrama de secuencia inicio de sesión empresa](#_bookmark121) [157](#_bookmark121)
           3. [Diagrama de secuencia inicio de sesión administrador](#_bookmark123) [158](#_bookmark123)
           4. [Diagrama de secuencia perfil de usuario joven](#_bookmark125) [159](#_bookmark125)
           5. [Diagrama de secuencia perfil de usuario empresa](#_bookmark127) [160](#_bookmark127)
           6. [Diagrama de secuencia perfil de usuario administrador](#_bookmark129) [161](#_bookmark129)
           7. [Diagrama de secuencia pantalla principal joven](#_bookmark131) [162](#_bookmark131)
           8. [Diagrama de secuencia pantalla principal empresa](#_bookmark133) [163](#_bookmark133)
           9. [Diagrama de secuencia pantalla principal administrador](#_bookmark135) [164](#_bookmark135)
  3. [Fase de desarrollo](#_bookmark137) [165](#_bookmark137)
     1. [Instalación y configuración de la tecnología implementada en Windows](#_bookmark138) [165](#_bookmark138)
        1. [Instalación y configuración de Node JS](#_bookmark139) [165](#_bookmark139)
        2. [Instalación y configuración de Mongo Compass](#_bookmark140) [168](#_bookmark140)
        3. [Instalación y configuración de Postman](#_bookmark141) [170](#_bookmark141)
        4. [Instalación y configuración de Git Bash](#_bookmark142) [171](#_bookmark142)
        5. [Instalación y configuración de Android Studio](#_bookmark143) [172](#_bookmark143)
        6. [Instalación y configuración de Visual Studio Code](#_bookmark144) [174](#_bookmark144)
        7. [Instalación y configuración de Flutter](#_bookmark145) [176](#_bookmark145)
     2. [Instalación y configuración de la tecnología implementada en macOS](#_bookmark146) [179](#_bookmark146)
        1. [Instalación y configuración de Flutter en macOS](#_bookmark147) [179](#_bookmark147)
        2. [Instalación y configuración de Xcode](#_bookmark148) [181](#_bookmark148)
        3. [Instalación y configuración de Android Studio](#_bookmark149) [183](#_bookmark149)
        4. [Instalación y configuración de Postman](#_bookmark150) [184](#_bookmark150)
        5. [Instalación y configuración de Visual Studio Code](#_bookmark151) [185](#_bookmark151)
  4. [Desarrollo de la aplicación móvil](#_bookmark152) [186](#_bookmark152)
     1. [Creación del restfull server](#_bookmark153) [186](#_bookmark153)
        1. [Paqueterías utilizadas](#_bookmark154) [186](#_bookmark154)
        2. [Database](#_bookmark155) [187](#_bookmark155)
        3. [Models](#_bookmark156) [188](#_bookmark156)
        4. [Controllers](#_bookmark157) [191](#_bookmark157)
        5. [Middleware](#_bookmark158) [196](#_bookmark158)
        6. [Helpers 198](#_bookmark159)
        7. [Routes](#_bookmark160) [199](#_bookmark160)
     2. [Pruebas del Restfull Server](#_bookmark161) [203](#_bookmark161)
        1. [CRUD](#_bookmark162) [204](#_bookmark162)

[3.4.2.2 GET](#_bookmark163) [204](#_bookmark163)

[3.4.2.3 POST](#_bookmark164) [205](#_bookmark164)

[3.4.2.4 GET](#_bookmark165) [207](#_bookmark165)

[3.4.2.5 PUT](#_bookmark166) [208](#_bookmark166)

[3.4.2.6 DELETE](#_bookmark167) [209](#_bookmark167)

* + 1. [Desarrollo de la aplicación móvil](#_bookmark168) [211](#_bookmark168)
       1. [Creación de los modelos](#_bookmark169) [211](#_bookmark169)
       2. [Provider](#_bookmark170) [213](#_bookmark170)
       3. [Bloc](#_bookmark171) [217](#_bookmark171)
       4. [Widget Inheritedwidget](#_bookmark172) [222](#_bookmark172)
       5. [Pages](#_bookmark173) [223](#_bookmark173)
       6. [Routes](#_bookmark174) [226](#_bookmark174)

[CAPITULO IV. RESULTADOS 227](#_bookmark175)

* 1. [Vistas de la aplicación móvil](#_bookmark176) [227](#_bookmark176)
     1. [Pantalla de inicio de sesión](#_bookmark177) [228](#_bookmark177)
     2. [Pantalla de inicio se sesión joven y empresa](#_bookmark178) [228](#_bookmark178)
     3. [Pantalla de registro](#_bookmark179) [229](#_bookmark179)
     4. [Menú principal](#_bookmark180) [230](#_bookmark180)
     5. [Noticias](#_bookmark181) [231](#_bookmark181)
     6. [Talleres.](#_bookmark182) [232](#_bookmark182)
     7. [Convocatorias](#_bookmark183) [233](#_bookmark183)
     8. [Oferta educativa](#_bookmark184) [233](#_bookmark184)
        1. [Estructura de escuelas.](#_bookmark185) [234](#_bookmark185)
        2. [Estructura de carreras](#_bookmark186) [234](#_bookmark186)
        3. [Estructura de becas](#_bookmark187) [235](#_bookmark187)
        4. [Estructura de apoyos](#_bookmark188) [235](#_bookmark188)
     9. [Bolsa de trabajo](#_bookmark189) [236](#_bookmark189)
     10. [Foros](#_bookmark190) [237](#_bookmark190)
     11. [Webinar](#_bookmark191) [238](#_bookmark191)

[CAPITULO V. CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA ADQUIRIDA. 239](#_bookmark192)

* 1. [Conclusiones](#_bookmark193) [239](#_bookmark193)
  2. [Recomendaciones.](#_bookmark194) [240](#_bookmark194)
  3. [Glosario de términos](#_bookmark195) [242](#_bookmark195)
  4. [Experiencia adquirida](#_bookmark196) [244](#_bookmark196)

[CAPITULO VI. COMPETENCIAS DESARROLLADAS Y/O APLICADAS 245](#_bookmark197)

* 1. [Competencias generales.](#_bookmark198) [245](#_bookmark198)
  2. [Competencias estratégicas.](#_bookmark199) [245](#_bookmark199)

[CAPITULO VII. FUENTES DE INFORMACIÓN 246](#_bookmark200)

* 1. [Referencias bibliográficas](#_bookmark201) [246](#_bookmark201)
  2. [Bibliográficas virtuales](#_bookmark202) [249](#_bookmark202)

[CAPITULO VIII. ANEXOS. 253](#_bookmark203)

* 1. [Manual de usuario](#_bookmark204) [253](#_bookmark204)
     1. [Configuración del Restfull server](#_bookmark205) [253](#_bookmark205)
     2. [Agregar administradores](#_bookmark206) [259](#_bookmark206)
     3. [Agregar datos](#_bookmark207) [262](#_bookmark207)
     4. [Editar datos](#_bookmark208) [264](#_bookmark208)
     5. [Eliminar datos](#_bookmark209) [266](#_bookmark209)
     6. [Administrar galería](#_bookmark210) [266](#_bookmark210)
     7. [Administrar participantes](#_bookmark211) [270](#_bookmark211)

**INDICE DE FIGURAS**

Figura 1 Ubicación del Instituto de la Juventud del Estado de Oaxaca 28

Figura 2 Sistemas operativos móviles más usados en el mundo 38

Figura 3 Características de aplicaciones web, nativa e hibrida 51

Figura 4 Logo de React Native 58

Figura 5 Logo de Ionic. Fuente 59

Figura 6 Logo de Xamarin. Fuente 60

Figura 7 Logo de Android Studio 63

Figura 8 Logo de Dart 64

Figura 9 Logo de Flutter 65

Figura 10 Estructura de flutter 67

Figura 11 Ejemplo de patrón BLoC 69

Figura 12 Logo de Visual Studio Code 70

Figura 13 Logo de Material Design 71

Figura 14 Logo de Postman 72

Figura 15 Logo de Xcode 73

Figura 16 Logo de MySQL 74

Figura 17 Logo de MongoDB 75

Figura 18 Logo de Heroku 83

Figura 19 Logo de Node JS 84

Figura 20 Prototipo de inicio de sesión con sus credenciales de usuario 109

Figura 21Prototiopo de inicio de sesión seleccionar usuario 109

Figura 22 Prototipo de perfil de usuario 110

Figura 23 Prototipo de pantalla principal "menús" 111

Figura 24 Prototipo de los diferentes módulos de la aplicación 1 112

Figura 25 Prototipos de los diferentes módulos de la aplicación 2 113

Figura 26 Diagrama general de la base de datos de la aplicación móvil 114

Figura 27 Tabla usuario 115

Figura 28 Tabla persona 116

Figura 29 Tabla roles 117

Figura 30 Tabla externo 117

Figura 31 Tabla inscripciones 118

Figura 32 Tabla escuela 118

Figura 33 Tabla carrera 119

Figura 34 Tabla oferta 119

Figura 35 Tabla bolsa de trabajo 120

Figura 36 Tabla apoyo 120

Figura 37 Tabla beca 121

Figura 38 Tabla foro 122

Figura 39 Tabla usuario-foro 122

Figura 40 Tabla convocatoria 123

Figura 41 Tabla convocatoria-imagen 123

Figura 42 Tabla taller 124

Figura 43 Tabla taller-img 124

Figura 44 Tabla imagen 125

Figura 45 Tabla noticia 126

Figura 46 Tabla noticia-img 126

Figura 47 Tabla participante 127

Figura 48 Tabla participante-noticia 127

Figura 49 Tabla instructor 128

Figura 50 Tabla instructor-taller 128

Figura 51 Tabla red-social 129

Figura 52 Tabla res-instructor 129

Figura 53 Tabla red-participante 129

[Figura 54 Caso de uso "Inicio de sesión joven" 137](#_bookmark98)

[Figura 55 Caso de uso "Inicio de sesión empresa" 139](#_bookmark100)

[Figura 56 Caso de uso "Inicio de sesión administrador" 141](#_bookmark102)

[Figura 57 Caso de uso “Perfil de usuario joven” 143](#_bookmark105)

[Figura 58 Caso de uso "Perfil de usuario empresa" 145](#_bookmark107)

[Figura 59 Caso de uso "Perfil de usuario administrador” 147](#_bookmark109)

[Figura 60 Caso de uso "Pantalla principal joven" 149](#_bookmark112)

[Figura 61 Caso de uso "Pantalla principal empresa" 151](#_bookmark114)

[Figura 62 Caso de uso "Pantalla principal administrador" 153](#_bookmark116)

[Figura 63 Diagrama de secuencia inicio de sesión joven 156](#_bookmark120)

[Figura 64 Diagrama de secuencias inicio de sesión empresa 157](#_bookmark122)

[Figura 65 Diagrama de secuencias inicio de sesión administrador 158](#_bookmark124)

[Figura 66 Diagrama de secuencia perfil de usuario joven 159](#_bookmark126)

[Figura 67 Diagrama de secuencia perfil de usuario empresa 160](#_bookmark128)

[Figura 68 Diagrama de secuencia perfil de usuario administrador 161](#_bookmark130)

[Figura 69 Diagrama de secuencia pantalla de principal joven 162](#_bookmark132)

[Figura 70 Diagrama de secuencia pantalla principal empresa 163](#_bookmark134)

[Figura 71 Diagrama de secuencia pantalla principal administrador 164](#_bookmark136)

Figura 72 Descarga de Node JS 165

Figura 73 Instalación de Node JS 166

Figura 74 Configuración de Node JS 166

Figura 75 Comprobación del Node JS 167

Figura 76 Instalación de librerías de Node JS 167

Figura 77 Instalación del nodemon en Node JS 167

Figura 78 Descarga de Mongo Compass 168

Figura 79 Instalación de Mongo Compass 168

Figura 80 Área de trabajo de Mongo Compass 169

Figura 81 Descarga de postman 170

Figura 82 Área de trabajo de postman 170

Figura 83 Descarga de git bash 171

Figura 84 Instalación del git bash 171

Figura 85 Descarga de Android Studio 172

Figura 86 Instalación de Android Studio 172

Figura 87 Configuración de los SDK 173

Figura 88 Descarga de los paquetes SDK 173

Figura 89 Descarga de Visual Studio Code 174

Figura 90 Instalación de Visual Code Studio 174

Figura 91 Configuración datos de Visual Studio Code 175

Figura 92 Configuración del PATH en Visual Studio Code 175

Figura 93 Descarga de Fluttter 176

Figura 94 Carpeta src 176

Figura 95 Carpeta flutter 176

Figura 96 Verificación de flutter 177

Figura 97 Descarga de paquetes de flutter 177

Figura 98 Configuración de las variables de entorno 178

Figura 99 Configuración de Flutter y Dark 178

Figura 100 Configuración de flutter en el equipo 179

Figura 101 Configuración del PATH “Creación del archivo .zshrc” 180

Figura 102 Configuración del PATH 180

Figura 103 Configuración del archivo .zshrc 180

Figura 104 Actualización de componentes de flutter 181

Figura 105 Instalación de Xcode 181

Figura 106 Instalación de componentes adicionales Xcode 182

Figura 107 Configuración del dispositivo físico en iOS 182

Figura 108 Instalación de Android Studio en macOS 183

Figura 109 Configuración de Android Studio en macOS 183

Figura 110 Aplicación postman en macOS 184

Figura 111 Aplicación Visual Studio Code en macOS 185

Figura 112 Paqueterías utilizadas para el restfull server 186

Figura 113 Conexiones del restfull server 187

Figura 114 Conexión a la base de datos 187

Figura 115 Llamado de la base de datos. 188

Figura 116 Upload de archivos. 188

Figura 117 Creación del constructor PATH 188

Figura 118 Configuración de las rutas PATH 189

Figura 119 Creación de los modelos 189

Figura 120 Modelos creados para el restfull server 190

Figura 121 Llamado de diferentes bloques. 191

Figura 122 Validación de correo electrónico. 191

Figura 123 Validación de usuario activo. 192

Figura 124 Validación de contraseña 192

Figura 125 Validación de token de usuario. 192

Figura 126 Obtener todos los registros. 193

Figura 127 Obtener un solo registro. 193

Figura 128 Creación de un registro. 194

Figura 129 Actualización de registros 194

Figura 130 Eliminación de registros 194

Figura 131 Controladores creados para el restfull server 195

Figura 132 Validación del token. 196

Figura 133 Validación de token incorrecto 196

Figura 134 Validación de token de usuario eliminado 197

Figura 135 Validación de token por error del usuario. 197

Figura 136 Middlewares creados para el restfull server 197

Figura 137 Declaración de archivos 198

Figura 138 Extracción de archivos 198

Figura 139 Validación del tipo de extensión 198

Figura 140 Validaciones creadas para el restfull server 199

Figura 141 Rutas de los middlewares. 199

Figura 142 Rutas de los helpers. 200

Figura 143 Rutas de los controladores. 200

Figura 144 Obtención de los registros 200

Figura 145 Obtener un solo registro. 201

Figura 146 Creas y guardar un registro. 201

Figura 147 Actualizar registros 201

Figura 148 Eliminar registro 202

Figura 149 Diferentes routes creadas para el restfull server 202

Figura 150 Módulos creados en Postman. 203

Figura 151 Estructura de los módulos. 203

Figura 152 Método get. 204

Figura 153 Resultado de la consulta 204

Figura 154 Método post 205

Figura 155 Registro de datos. 205

Figura 156 Error de token de usuario. 205

Figura 157 Obtener token de usuario. 206

Figura 158 Token de usuario 206

Figura 159 Menú headers. 206

Figura 160 Resultado de la consulta 207

Figura 161 Método get 207

Figura 162 Resultado de la consulta 208

Figura 163 Método put 208

Figura 164 Registros a modificar 208

Figura 165 Resultado de la consulta 209

Figura 166 Método get 209

Figura 167 Resultado de la consulta 209

Figura 168 Estado de la creación de las consultas. 210

Figura 169 Constructor 211

Figura 170 Variables de entorno 211

Figura 171 Conversión a instancias. 212

Figura 172 Conversión a json 212

Figura 173 Creación de los diferentes modelos creados para la app móvil 212

Figura 174 Librerías utilizadas. 213

Figura 175 Dirección a Postman y token 213

Figura 176 Método fromJson 214

Figura 177 Cache de errores. 214

Figura 178 Eliminar registro 214

Figura 179 Obtener un solo registro. 215

Figura 180 Registrar datos. 215

Figura 181 Actualizar datos. 216

Figura 182 Actualizar imagen 216

Figura 183 Librerías utilizadas. 217

Figura 184 Envió de errores al oyente 217

Figura 185 Creación de un objeto y secuencia de eventos asíncronos. 218

Figura 186 Función try catch. 218

Figura 187 Función para obtener un registro 219

Figura 188 Función para eliminar un registro 219

Figura 189 Función para registro de datos. 219

Figura 190 Función para actualización de datos 220

Figura 191 Función para actualización de imagen 220

Figura 192 Método dispose. 220

Figura 193 Blocs creados para la aplicación móvil 221

Figura 194 Creación de los blocs. 222

Figura 195 Librerías utilizadas. 223

Figura 196 Widge para la creación de la interfaz gráfica. 223

Figura 197 Declaración de variables de entorno 224

Figura 198 Widget \_builder 224

Figura 199 Validación de datos 225

Figura 200 Pages creadas para el desarrollo de la aplicación móvil 225

Figura 201 Llamado de los diferentes módulos pages. 226

Figura 202 Creación de las rutas. 226

Figura 203 Pantalla de inicio de sesión 228

Figura 204 Pantalla de joven y empresa 228

Figura 205 Pantalla de registro 1 229

Figura 206 Pantalla de registro 2 229

Figura 207 Pantalla de menú principal 230

Figura 208 Pantalla de noticias 231

Figura 209 Pantalla de participantes 232

Figura 210 Pantalla de talleres 232

Figura 211 Pantalla de convocatorias 233

Figura 212 Estructura de la pantalla de oferta educativa 233

Figura 213 Estructura de la pantalla de escuela 234

Figura 214 Estructura de la pantalla de carreras 234

Figura 215 Estructura de la pantalla becas 235

Figura 216 Estructura de la pantalla de apoyos 235

Figura 217 Estructura de la pantalla de bolsa de trabajo 236

Figura 218 Estructura de la pantalla de foros 237

Figura 219 Estructura de la pantalla webinar 238

Figura 220 Creación de las carpetas para el restfull server 253

Figura 221 Configuración del restfull server 254

Figura 222 Configuración de npm 254

Figura 223 Creación del archivo .env 255

Figura 224 Inicialización del servidor 255

Figura 225 Comprobación que el servidor este corriendo correctamente 256

Figura 226 Instalación de heroku 256

Figura 227 Inicio de sesión en Heroku 2 257

Figura 228 Inicio de sesión en Heroku 2 257

Figura 229 Creación del restfull server en Heroku 257

Figura 230 Configuración de las variables del archivo .env 258

Figura 231 Comprobación del proyecto en heroku 258

Figura 232 Importación de los CRUD 259

Figura 233 Registro de un usuario nuevo. 259

Figura 234 Datos a registrar de un usuario. 260

Figura 235 Datos registrados de un usuario nuevo mostrado en la terminal 260

Figura 236 Finalizar registro de datos de un usuario nuevo 261

Figura 237 Interfaz gráfica del módulo de noticias 262

Figura 238 Interfaz gráfica para registro nuevo de una noticia 263

Figura 239 Menús de la aplicacion 264

Figura 240 Interfaz gráfica para editar 265

Figura 241 Menú eliminar 266

Figura 242 Galería de imágenes. 266

Figura 243 Interfaz gráfica para agregar imágenes. 267

Figura 244 Interfaz gráfica para seleccionar imágenes. 268

Figura 245 Interfaz gráfica para eliminar imágenes. 269

Figura 246 Interfaz gráfica para administrar los contribuyentes. 270

Figura 247 Interfaz gráfica para agregar nuevos contribuyentes 271

Figura 248 Interfaz gráfica para editar los contribuyentes 271

**INDICE DE CUADROS**

[Tabla 1 Cuadro comparativo de las bases de datos relacionales y no relacionales 49](#_bookmark28)

[Tabla 2 Lista de paquetes Node JS 86](#_bookmark63)

[Tabla 3 CRUD 88](#_bookmark65)

[Tabla 4 Formación del equipo SCRUM para el proyecto 102](#_bookmark75)

[Tabla 5 Definición de los objetivos del producto (Product Backlog) 108](#_bookmark80)

[Tabla 6 Historia de usuario "Inicio de sesión para usuario joven y empresa" 131](#_bookmark87)

[Tabla 7 Historia de usuario "Inicio de sesión para administrador" 132](#_bookmark88)

[Tabla 8 Historia de usuario para” Perfil de usuario” 133](#_bookmark90)

[Tabla 9 Historia de usuario "Módulos de la pantalla principal Joven" 134](#_bookmark92)

[Tabla 10 Historia de usuario "Módulos de la pantalla principal Empresa" 135](#_bookmark93)

[Tabla 11 Historia de usuario "Módulos de la pantalla principal Administrador" 136](#_bookmark94)

[Tabla 12 Actores de los casos de usos 137](#_bookmark96)

[Tabla 13 Especificación de caso de uso inicio de sesión joven 138](#_bookmark99)

[Tabla 14 Especificación de caso de uso inicio de sesión empresa 140](#_bookmark101)

[Tabla 15 Especificación de caso de uso inicio de sesión administrador 142](#_bookmark103)

[Tabla 16 Especificaciones de caso de uso perfil de usuario joven 144](#_bookmark106)

[Tabla 17 Especificaciones de caso de usuario perfil de usuario empresa 146](#_bookmark108)

[Tabla 18 Especificación de caso de uso perfil de usuario administrador 148](#_bookmark110)

[Tabla 19 Caso de uso pantalla principal joven 150](#_bookmark113)

[Tabla 20 Especificación de caso de uso pantalla principal empresa 152](#_bookmark115)

[Tabla 21 Especificación de caso de uso pantalla principal administrador 154](#_bookmark117)

CAPITULO I. GENERALIDADES DEL PROYECTO

* 1. Introducción

Hoy en día el uso de la tecnología es prácticamente una parte de nuestra vida diaria ya que la mayoría de las personas optan por esta nueva modalidad ya sea para pagos, tramites o simplemente para estar comunicados, una necesidad para poder realizar las actividades cotidianas, con la llegada de la pandemia (COVID-19) la cual afecto a toda la población mundial, las empresas tuvieron que buscar una forma de llegar a sus clientes, de tal manera que esto provoco un aumento en el uso de aplicaciones web puesto que estas proporcionan un sinfín de características tales como bridar ayuda, entretenimiento, soporte etcétera.

Estas aplicaciones han permitido que se mantenga un contacto más cercano con los usuarios, por lo que en la actualidad muchas empresas y personas desean desarrollar sus propias aplicaciones y ofrecer algún servicio. Actualmente diversas empresas han creado departamentos informáticos en los cuales se está creando software, estas no solo están siendo enfocadas para teléfonos celulares, sino que también para la web.

Teniendo en cuenta el gran impacto y crecimiento que está teniendo en el mercado las aplicaciones web, el proyecto está enfocado en desarrollar una aplicación web que permita visualizarse en cualquier multiplataforma de navegación como lo es Google Chrome, Firefox, Microsoft Edge y muchos más sin importar en el dispositivo que se utilice para su visualización , evidentemente esto nos obliga como profesionales del sector tecnológico a conocer los retos y posibilidades de estos entornos, para lo cual se realizara un estudio minucioso, así como de las diferentes herramientas para su desarrollo, subsiguientemente seleccionar la metodología más adecuada para el desarrollo, con el fin de empezar el desarrollo de la aplicación cumpliendo con cada una de sus fases.

* 1. Descripción de la empresa

Surge en el 2017 en el mes de marzo ofreciendo la preparatoria a jóvenes para acreditar el nivel medio superior, y a los 6 meses surge la primera generación lo que permitió seguir invitando a gente para concluir sus estudios de media superior. A finales del año 2017 se establece el acuerdo con Universidades INNOVA, RUBENSTEIN, y grupo época para intercambiar y establecer a través de sus RVOES, registro ante la Secretaria de Educación Pública: certificar, preparar y titular; en diferentes carreras desde licenciatura, maestría y doctorado.

Esto trajo un beneficio para conformar un consorcio e intercambiar sus ofertas educativas. Cuando miran la respuesta que se tuvo con la población de egresos por parte del Centro de Asesoría Profesional “Paulo Freire”, brindan todas las facilidades, trámites administrativos y títulos para conformas la maestría en ciencias de la educación y su doctorado en educación en beneficio para los docentes que se encuentran laborando en el sistema federal y estatal.

Actualmente el Centro de Asesoría Profesional “Paulo Freire” ofrece desde bachillerato hasta doctorado bajo un lema “Esfuerzo, Perseverancia y Superación”, porque cada uno de los egresados asume un gran compromiso a la sociedad con un pensamiento crítico y con la responsabilidad social que le corresponde.

Y por último se ofrecen diplomados con certificación por parte de la secretaria de Trabajo y previsión Social, para desarrollar competencias y habilidades laborales con la finalidad de actualizar, capacitar y certificar en temas como Liderazgo a Directivos, Uso y Manejo de Pruebas psicométricas, equidad e igualdad de género, y capacitaciones para empresas y escuelas tanto en sector público y privado.

* + 1. Objetivos
       - busca la formación, la identidad, la investigación, bajo un fundamento epistemológico, ontológico y humanista; hacia una teoría critica desde una carácter interdisciplinario; para seguir preparando a los futuros docentes, profesionales de la educación.
       - Procurar que las y los jóvenes cumplan con sus deberes éticos y jurídicos frente a la sociedad.
       - Promover la congruencia de los programas y políticas en materia de juventud del orden federal, estatal y municipal.
       - Fomentar la participación del sector público, privado y social, en actividades que incidan en el desarrollo de las y los jóvenes en materia educativa, laboral, sexual, cultural, deportiva y su incorporación a los procesos productivos, así como en la prevención de adicciones, enfermedades y situaciones que pongan en riesgo el desarrollo integral de la juventud.
    2. Misión

Somos un centro de asesoría profesional que ofrece programas de maestría y doctorado para fortalecer la práctica docente de profesionales que laboran en el sector educativo público y/ o privado que desean fortalecer su ejercicio laboral a través de los principios de la pedagogía crítica como la libertad, el amor y la fraternidad, generando actitud crítica y competitiva para enfrentar los desafíos que exigen los cambios en nuestra sociedad; así como también, brinda educación continua por medio de certificaciones, diplomados, talleres y conferencias para contribuir al fortalecimiento de conocimientos y habilidades de carácter personal y profesional.

* + 1. Valores
       - Corresponsabilidad: La que existe entre el Estado, y la sociedad oaxaqueña en la atención a la juventud.
       - Igualdad: Que garantizara el acceso de las y los jóvenes oaxaqueños a las mismas oportunidades de desarrollo, tomando en cuenta, sus necesidades propias.
       - Respeto: Para garantizar la no discriminación a las personas jóvenes, por razones de sexo, edad, origen étnico, preferencia sexual, religión, embarazo, aspecto físico o cualquier otra reconocida por las leyes correspondientes.
       - Transparencia y rendición de cuentas: Las que se evidenciaran en los procesos de toma de decisiones que afecten la vida y el entorno de la juventud oaxaqueña.
    2. Lugar

Instituto de la Juventud del Estado de Oaxaca (INJEO). Calle Emilio Carranza No. 604 Planta Baja, Reforma, 68050 Oaxaca de Juárez.



Figura 1 Ubicación del Instituto de la Juventud del Estado de Oaxaca. Fuente: (https://www.google.com/maps/@17.0756989,- 96.7152431,16z)

* 1. Problema a resolver

En Oaxaca ha existido una enorme brecha digital en las TICS ya que la baja capacidad adquisitiva en conjuntos con otros factores, constituyen la pobreza multidimensional. Esta consiste en la insatisfacción de al menos uno de los derechos para el desarrollo social, que son: educación, servicios de salud, seguridad social, calidad y espacios de la vivienda, servicios básicos en la vivienda, y alimentación, (Coria, Pérez & Mendoza, 2019).

Aunque la mayoría de los Oaxaqueños sufren de este tipo de necesidades, datos arrojados por la Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares (ENDUTIH), arroja que el 62.6 % de personas cuentan con una Smartphone, donde la mayoría son jóvenes, (Excélsior, 2020).

Estos jóvenes de acuerdo con una encuesta realizada por Motorola: 9 de cada 10 adolescentes mexicanos revisan el celular tan seguido que nada cambió desde la última vez que lo vieron. Además, reveló que la mayoría de ellos mantiene encendido el teléfono las 24 horas del día, (Staff, 2019).

Por tal motivo el Instituto de la Juventud del Estado de Oaxaca (INJEO) planea llegar a los jóvenes por medio de una aplicación móvil, la cual brindara a los jóvenes una herramienta digital en donde podrán acceder a múltiples beneficios tales como convocatorias lanzadas por el gobierno del estado, así como cursos variados que el INJEO publica, todo esto con el fin de poder llevar a estos jóvenes por el buen camino, y que contribuyan a su formación y desarrollo de competencias profesionales y humanas.

# Objetivos

* + 1. Objetivo general.

Desarrollar una aplicación móvil que notifique a los jóvenes Oaxaqueños información sobre talleres, cursos y convocatorias que realiza el Instituto de la Juventud del Estado de Oaxaca (INJEO).

* + 1. Objetivos específicos.
       1. Establecer una línea de comunicación con los jóvenes Oaxaqueños para conocer sus inquietudes y necesidades.
       2. Brindar herramientas de soluciones en innovación tecnológica que coadyuven a el cumplimiento de los objetivos del Instituto de la Juventud del Estado de Oaxaca.
       3. Agilizar los procesos de registro para los usuarios beneficiarios.
       4. Indagar entre las diferentes metodologías de gestión de procesos para lograr que la aplicación sea eficiente y cumpla con los requerimientos planteados.
       5. Desarrollar un rest server que suministre de información a la aplicación.

# Justificación

El desarrollo de aplicaciones móviles es una herramienta tecnológica de gran utilidad, hoy en día la mayoría de las empresas las utilizan como un medio de comunicación rápida, de esta manera los usuarios acceden a la información desde cualquier lugar de manera efectiva y rápida.

Con lo antes mencionado surge el propósito de brindar una aplicación móvil para los jóvenes Oaxaqueños, que se enfoque no solo en brindar información del estado de Oaxaca, sino que además briden herramientas de aprendizaje enfocado al desarrollo cognitivo, social y educativo, pues hasta el día de hoy no hay ninguna aplicación que se centre en brindar ese tipo de información para los jóvenes Oaxaqueños.

Con esta aplicación se pretende que más jóvenes Oaxaqueños puedan estar informados de acontecimientos relevantes que suceden en el estado de Oaxaca ya sean talleres, convocatorias, apoyos monetarios, etcétera puesto que la aplicación solo se enfocara en el estado de Oaxaca.

CAPITULO II. MARCO TEÓRICO

El presente capitulo nos servirá para poner en claro algunos conceptos teóricos y prácticos, que son indispensables para el desarrollo de la aplicación, así como de las herramientas y tecnologías aplicadas, así como la metodología aplicada en el proyecto.

2.1 Fundamentación Teórica

* + 1. Tecnologías de la información y comunicación.

Las Tecnologías de la Información y Comunicación (TIC) se puede definir como las tecnologías que gira en torno a cuatro medios básicos: la informática, la microelectrónica, los multimedia y las telecomunicaciones; las cuales de manera conjunta permiten conseguir nuevas realidades comunicativas (Cabero, 2005).

Por otro lado, Tello (2011) menciona que las Tecnologías de la Información y de la Comunicación es un término que explora toda forma de tecnología usada para crear, almacenar, intercambiar y procesar información en sus varias formas, tales

como datos, conversaciones de voz, imágenes fijas o en movimiento, presentaciones multimedia y otras formas.

Con la aplicación de estas tecnologías se logra entrar a un mundo nuevo lleno de información de fácil acceso, de tal modo que existen múltiples ejemplos que engloban el concepto de TIC, la televisión, las computadoras, los smartphones, pero sin lugar a duda, lo medios más representativos de la sociedad actual son los ordenadores que nos permiten utilizar diferentes aplicaciones informáticas y más específicamente las redes de comunicación, en concreto internet, (Belloch, 2008).

* + 1. Internet

Se podría definir Internet como la red de redes, también conocida como red global, el cual es un sistema mundial de comunicaciones que permite a las personas acceder a información disponible en cualquier servidor mundial (Belloch 2008).

Algunas de las características de la información de Internet han sido analizadas por Cabero (2005) como representativas de las TIC son:

* + - * Inmaterialidad, su materia prima es la información en múltiples códigos y formas.
      * Interconexión, aunque se presentan de forma independiente, pueden combinarse y ampliar sus conexiones.
      * Interactividad, permite la interacción del sujeto con la máquina y, así, la adaptación de éstas a diversas áreas educativas y cognoscitivas de las personas.
      * Instantaneidad, facilita la rapidez de acceso e intercambio de la información.
      * Calidad de imagen y/o sonido, lo que da fiabilidad y fidelidad a la información transferida.
      * Penetración en diversos sectores de la sociedad: salud, educación, economía, etcétera.

Como se mencionó anteriormente el internet no es una sola red, está unida por diversas redes internacionales a un núcleo central, la original Arpanet (Advanced Projects Agency Net). Esta tecnología nació en los años 60 por el ministerio de defensa de Estados Unidos, ya que solicitaban que la defensa del país dependiera de la misma red y compartiera los recursos de ésta que posteriormente serviría para la creación de lo que hoy se conoce como internet, (Cuadra, 1996).

* + 1. Tecnología móvil

La tecnología móvil, en sus inicios era desarrollada para utilizarse en las computadoras de escritorio y portátiles, lo común en ese tiempo; sin embargo, los dispositivos han cambiado y con ese cambio se ha presentado la evolución de la tecnología web. Ahora bajo esta tecnología los recursos son creados para ejecutarse mediante dispositivos móviles inteligentes (Garita, 2013).

Garita (2013) señala que se podría denominar dispositivo móvil a todo aparato electrónico que cumple con unas características básicas:

* + - * Reducción de tamaño
      * Presentación actualmente en tamaño
      * Conexión inalámbrica

Por otra parte, Velastegui y López (2018) se refieren a los dispositivos móviles como un grupo sumamente heterogéneo y pueden incorporar casi cualquier componente tales como hardware y software que amplíe y diversifique su función inicial.

El avance de la tecnología móvil ha tenido un gran impacto en la sociedad ya que gracias a ella las personas pueden hacer diversas tareas todo desde la palma de su mano así mismo permitiéndoles estar comunicados en cualquier parte del mundo, cabe resaltar que el mayor factor para que los dispositivos móviles tuvieran un mayor alcance en la sociedad fue la posibilidad de acceso a internet (internet móvil).

Cajilima (2016), menciona que gracias a las aplicaciones disponibles para teléfonos inteligentes le dan un valor agregado para quienes buscan entrenamiento, así mismo todas estas aplicaciones se ejecutan dentro de un ecosistema existiendo varios factores como son: la infraestructura de la aplicación, el sistema operativo, los métodos de entrada de información, los canales de distribución de la aplicación, permitiendo que estas aplicaciones puedan ejecutarse en diferentes tipos de dispositivos.

* + 1. Sistemas operativos

Para Villareal (2017) un sistema operativo es un programa que controla la ejecución de los programas de aplicación y que actúa como interfaz entre el usuario de un computador y el hardware de la misma.

Por otro lado, Brookshear (2007) define que un sistema operativo es el software que controla la operación general de una computadora, proporciona los medios por los que un usuario puede almacenar y recuperar archivos, provee la interfaz por la que un usuario puede solicitar la ejecución de programas y provee el ambiente necesario para que los programas solicitados se ejecuten.

Existen diferentes sistemas operativos en el mercado, pero para EAE (2021) los que más han destacado en la actualidad son:

* + - * Microsoft Windows: Es quizá, el más genérico de todos los sistemas operativos actuales. Creado en 1985, ofrece la mayor gama de aplicaciones para software. Al mismo tiempo, es el que ofrece mayor flexibilidad para la introducción de actualizaciones, es funcional para cualquier tipo de usuario.
      * GNU/Linux: Es un sistema operativo libre. Lleva varios años desarrollándose a la par de las grandes marcas, destaca por una mayor potencia, estabilidad, seguridad ante amenazas externas y la posibilidad de modificar el sistema según las preferencias individuales; ideal para aquellos usuarios a los que les guste experimentar constantemente. Una de sus variantes más conocidas es el sistema Android, empleado para teléfonos móviles de alta gama.
      * OSX: Sistema operativo exclusivo para los productos de la casa Apple, en los últimos años ha ido ganando presencia entre los consumidores, sobre todo tras el empleo masivo de dispositivos IPod, IPhone e IPad.
      * Chrome OS: Es un sistema operativo que ofrece la empresa Google ha sido uno de los últimos en llegar al mercado. Su principal característica es que todo el trabajo se realiza desde el navegador, con lo cual los dispositivos no requieren de un disco local grande ni de un hardware

demasiado sofisticado, se destaca del resto por la velocidad de navegación, la sencillez y el precio asequible.

Así como existen sistemas operativos para ordenadores existen para los dispositivos móviles que les permiten la interacción con los usuarios, de este modo es que logra realizar las tareas cotidianas los smartphones, tales como la apertura de aplicaciones, encendido y apagado del dispositivo no obstante para realizar estas tareas es necesario contar con su respectivo hardware para poder realizar dichas tareas.

* + - 1. Sistemas operativos de dispositivos móviles

Villota (2015) se refiere a los Sistemas Operativos para Móviles “Como programas que se encargan de manejar los procesos básicos de un dispositivo, permitiendo el uso de sus diferentes recursos; inicialmente fueron desarrollados para computadoras, pero actualmente son utilizados para dispositivos móviles como celulares, tabletas electrónicas, para tener la misma interlocución entre el hardware y los programas que el usuario desea utilizar “.

Por otro lado, para Cajilima (2016) un Sistema Operativo Móvil realiza la interacción real con lo que podemos hacer a partir de las capacidades del hardware que conforman un equipo. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice y cada vez lo ejecuta con mayor inteligencia y rapidez con la que en general de desempeña.

Actualmente datos arrojados por StatCounter el Sistema Operativo para Móviles más usados por los usuarios alrededor del mundo es Android, en la figura 2 se

logra apreciar con mayor detalle el porcentaje de usuarios, así como los diferentes sistemas operativos que existen en la actualidad.

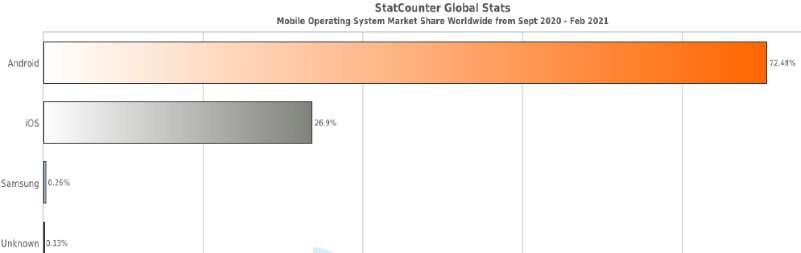


Figura 2 Sistemas operativos móviles más usados en el mundo. Fuente: (https://gs.statcounter.com/os-market- share/mobile/worldwide/#monthly-202009-202102-bar)

Android ha tenido un gran impacto en el mercado, durante los últimos años, se ha convertido en el líder de los Sistemas Operativos Móviles ya que su software de código libre permite integración con soluciones de código de propietario, por otro lado, se encuentra iOS el cual es un S.O. Móvil el cual se caracteriza por tener una interfaz más fluida y una mejor seguridad.

* + - 1. Sistemas operativo Android

Android es un sistema operativo con una plataforma abierta para dispositivos móviles adquirido por Google y la Open Handset Allience, su finalidad es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos, además de fomentar el desarrollo de aplicaciones, cualidad que ningún otro sistema operativo incluye en sus conceptos (Maleve & Beauperthu, 2011).

Una de sus principales características de Android es que cuenta con un código abierto el cual permite que cualquier persona pueda tener acceso al código fuente y modificar este software, por otro lado, cuenta con una gran variedad de aplicaciones cabe recalcar que no llevan un largo proceso de evaluación para ser publicadas, lo cual es el caso contrario de iOS.

Otra característica del sistema operativo Android es que se encuentra instalado en varias marcas de celulares, entre ellas encontramos a Huawei, Samsung, Motorola, entre otros grandes fabricantes de teléfonos móvil, por ende, cuenta con una amplia usabilidad y un gran apartado de temas para su personalización.

Por otro lado, algunas de las desventajas que se encuentra en este sistema operativo es un mayor riesgo de vulnerabilidad ya que, al tener un código abierto, los teléfonos móviles son más susceptibles a ser hackeados. Otro punto débil de Android son las actualizaciones ya que esto depende del fabricante del dispositivo móvil, lo que hace que una respuesta pueda tardar semanas e incluso meses en adaptar la nueva versión al móvil del usuario.

* + - 1. Sistemas operativo iOS

iOS (iPhone Operating System) es un sistema operativo propiedad de Apple, utilizado en dispositivos propios de la marca entre ellos iPhone y iPad. En esencia es muy similar a Android, permitiendo instalar aplicaciones básicas de un teléfono tales como aplicaciones ofimáticas, de productividad, almacenamiento en la nube, educativas, juegos y muchas más (Filgueira, 2019).

Sus diferencias con Android, es que el uso de su sistema operativo se restringe a dispositivos de fabricación propia, mientras que Android se puede instalar en infinidad de dispositivos. Por otro lado, el nivel de seguridad es mucho más alto, las aplicaciones pasan por un proceso de validación mucho más estricto y además el firmware del sistema es más robusto.

Las ventajas que este sistema operativo ofrece es una mayor seguridad frente a las ciberamenazas, cuenta con un mayor filtro y exclusividad dentro del mercado de aplicaciones ya que estas atraviesan un proceso arduo de validación para ser publicadas en su tienda de iOS. Por otra parte, nos encontramos con un sistema operativo con interfaz limpia y muy sencilla de utilizar, así mismo la mayoría de las aplicaciones son lanzadas para iOS antes que para Android.

Los puntos débiles de iOS son las pocas opciones que da para su personalización del sistema operativo ya que todas las interfaces del sistema suelen ser muy similares. Cabe recalcar que una de su desventaja más hablada por sus usuarios es el bajo rendimiento de la batería.

Otro aspecto de iOS es que no permite instalar aplicaciones de terceros como en Android solo se pueden descargar desde su tienda oficial, además es un sistema de código cerrado el cual es solo exclusivo de Apple y por lo tanto no se dispone de otros dispositivos móviles como en Android.

* + 1. Base de datos

Para Silberschatz, Korth & Sudarshan (2002) un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Por otra parte, para Ramos, Montero & Ramos (2008) un sistema gestor de bases de datos o SGBD, también llamado DBMS (Data Base Management System) es una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan los datos. La colección de esos datos se denomina Base de Datos o BD (DB Data Base).

* + - 1. Aplicaciones de los sistemas de bases de datos

Las bases de datos son ampliamente usadas, Silberschatz, Korth & Sudarshan (2002) mencionan algunas de sus aplicaciones más representativas:

* + - * + Banca. Para información de los clientes, cuentas y préstamos, y transacciones bancarias.
        + Universidades. Para información de los estudiantes, matriculas de las asignaturas y cursos.
        + Telecomunicaciones. Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
        + Finanzas. Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.
        + Ventas. Para información de clientes, productos y compras.
        + Producción. Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.

Estas son algunas de las empresas que utilizan las bases de datos cabe mencionar que en la actualidad existen aún más empresas que utilizan esta tecnología.

* + - 1. Tipos de bases de datos

En la actualidad existen diferentes tipos de bases de datos, las cuales son orientadas a diferentes necesidades de los usuarios, en este caso se hablará solamente de las bases de datos relacionales y las bases de datos no relacionales que también son conocidas como NoSQL.

* + - * 1. Modelo relacional

Una base de dato relacional es un grupo de tablas que representan los datos y están relacionados entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. En el modelo relacional de base de datos, las tablas comparten algún campo entre ellas, estos campos compartidos van a servir para establecer relaciones entre las tablas que permitan consultas complejas.

La idea básica de las bases de datos relacionales es la existencia de entidades (filas en una tabla) caracterizadas por atributos (columnas en la tabla). Cada una de las tablas almacena entidades del mismo tipo y entre entidades de distinto tipo se establecen relaciones. Las tablas comparten algún campo entre ellas y estos campos compartidos van a servir para establecer relaciones entre las tablas.

A continuación, Gómez (2013) cita los cuatro objetos básicos de una base de datos relacional:

Las relaciones o tablas almacenan los datos en filas y columnas. Todas las bases de datos contienen una o más tablas.

Las consultas recuperan y procesan los datos. Pueden combinar datos de distintas tablas, actualizar los datos y realizar cálculos con éstos.

Los formularios controlan la entrada de datos y las vistas de datos, proporcionan indicaciones visuales que simplifican el trabajo con los datos.

Los informes resumen e imprimen los datos. Convierten los datos de las tablas y consultas en documentos que comunican ideas.

Lafuente (2018), indica que en el ámbito informático se habla mucho de ACID, cuyas siglas significan: atomicidad, consistencia, aislamiento y durabilidad, las cuales son propiedades que las bases de datos aportan a los sistemas y permiten ser más robustos y menos vulnerables ante fallos.

Ventajas y desventajas de las bases de datos relacionales

Desde el punto de vista de Robles (2019) menciona las siguientes ventajas de las bases de datos relacionales:

Al ser una tecnología bastante madura, cuenta con una documentación muy extensa y una comunidad bastante activa. Cualquier duda puede ser resuelta con un poco de investigación.

Los estándares SQL se encuentran bien definidos y son ampliamente aceptados.

Una gran cantidad de desarrolladores cuentan con amplia experiencia en esta tecnología.

Toda base de datos relacional debe cumplir con los principios ACID, por lo cual los datos son confiables.

La escritura es simple ya que gran parte de la aceptación depende de la sencillez de su método de escritura, este es muy parecido al lenguaje que utilizamos los humanos, facilitando para nosotros la comprensión de las operaciones.

Por otra parte, Cutipa (2020) menciona las desventajas que se encuentran en este tipo de bases de datos:

Dificultades de crecimiento: Cuando estas bases de datos comienzan a crecer en volumen, el almacenamiento y el costo de mantenimiento se convierten en un problema de alto costo.

Cambios en la estructura: el entorno empresarial es altamente dinámico. Esto exige que se realicen cambios de forma eventual en los registros de datos. Si ejecutamos cambios, la Base de Datos debe ser modificada en su estructura para admitir las modificaciones. Si las modificaciones no se realizan esta se verá afectada y sus procesos interrumpidos.

Complejidad en la instalación: Algunas bases de datos SQL se ven condicionadas por el sistema operativo en el cual van a funcionar y los requisitos mínimos de funcionamiento de los servidores u ordenadores.

Dificultad en la interfaz: La interfaz de una base de datos SQL son más complejas que agregar algunas líneas de código.

Más características implementadas de forma patentada: Aunque las bases de datos SQL se ajustan a los estándares ANSI e ISO, algunas bases de datos implementan extensiones propietarias al SQL estándar para garantizar el bloqueo del proveedor.

* + - * 1. Modelo no relacional

Las bases de datos no relacionales también conocidas como NoSQL son sistemas de almacenamientos de información que no cumplen con los esquemas entidad/relación es decir que no se imponen una estructura de datos en forma de tablas y relaciones entre ellas, por lo tanto, estos sistemas son más flexibles, ya que nos permiten almacenar información.

El término NoSQL fue inicialmente utilizado en el año 1998, y fue para denominar una base de datos relacional que no utilizaba el lenguaje SQL para funcionar. A partir de aquí, el término fue rescatado en 2009 en unas charlas por defensores de las bases de datos no relacionales (Herranz, 2014).

Este tipo de base de datos cuenta con una gran escalabilidad y están diseñadas para la gestión de grandes volúmenes de datos. Por otro lado, a diferencia de las bases de datos relacionales no cumple con el estándar ACID de atomicidad, consistencia, aislamiento y durabilidad. Cabe mencionar que las bases de datos

no relacionales son más actuales que las relacionales y su creación se basó en la necesidad de crear sistemas de gestión capaces de trabajar con datos no estructurados o semi estructurados (Tablado, 2020).

Es importante destacar algunas características y funciones que cuentan este tipo de base de datos Tablado (2020).

* La información no se almacena en tablas sino a través de documentos.
* Son bases de datos muy útiles para organizar y gestionar información no estructurada, o cuando no se tiene una noción clara de los datos a almacenar.
* Son bases de datos con alto grado de escalabilidad y están diseñadas para soportar grandes volúmenes de datos.
* No utilizan el lenguaje SQL para consultas, aunque sí lo pueden usar como herramienta de apoyo.
* Es un sistema de almacenamiento de datos relativamente nuevo, y como tal, todavía no posee un sistema estandarizado.
* A diferencia de las no relacionales, no garantizan el cumplimiento de las cualidades ACID, esto es, atomicidad, consistencia, integridad y durabilidad.

Ventajas y desventajas de las bases de datos no relacionales

Cutipa (2020) hace referencia a las ventajas que proporcionan las bases de datos no relacionales las cuales son las siguientes:

Aplicaciones de big data: grandes volúmenes son manejados fácilmente por las bases de datos NoSQL.

Administración de la base de datos: Requieren menos administración práctica, cuenta con capacidades de distribución de datos y reparación automática, modelos de datos simplificados y menos requisitos de ajuste y administración.

Versatilidad: Las posibilidades de crecimiento en el volumen de datos o la posibilidad de incluir cambios sobre la forma en la que ingresan los datos sin necesidad de alterar la estructura, permite adaptarse de forma rápida a un entorno de alto dinamismo.

Crecimiento Horizontal: Son altamente escalables, si se requiere instalar mayor cantidad de nodos para ampliar la capacidad, se puede hacer sin problemas. Esto no interrumpe la usabilidad o consultas dentro de la base de datos.

Economía: No se necesitan servidores con gran cantidad de recursos para operar. La adaptabilidad y flexibilidad permiten empezar con bajos niveles de inversión en equipos e ir ampliando la capacidad a medida de las necesidades.

Posteriormente nos encontramos con sus desventajas:

Menos maduro: la mayoría de las alternativas de bases de datos NoSQL apenas han salido de las etapas de preproducción, y hay muchas características importantes que aún no se han implementado.

Atomicidad: Algunas de estas bases de datos no incorporan la atomicidad de información. Esto puede derivar en que la información no sea consistente entre nodos.

Software poco documentado: Al ser tan relativamente nuevo, el NoSQL puede adolecer de que algunas operaciones sean limitadas por la falta de información sobre las herramientas y sus características.

Herramientas GUI: la mayoría de las bases de datos NoSQL no contienen una interfaz gráfica. Requiere conocimiento especial para poder ejecutar algunas de ellas.

En la tabla 1 se muestran las diferencias que existe entre las bases de datos relacionales y no relacionales.

|  |  |  |
| --- | --- | --- |
|  | **Bases de datos relacionales** | **Base de datos NoSQL** |
| Cargas de trabajo optimas | Las bases de datos relacionales están diseñadas para aplicaciones de procesamiento de transacciones online (OLTP) altamente coherentes y transaccionales, y son buenas para el procesamiento analítico online (OLAP). | Las bases de datos NoSQL están diseñadas para varios patrones de acceso a datos que incluyen aplicaciones de baja latencia. Las bases de datos de búsqueda NoSQL están diseñadas para hacer  análisis sobre datos semiestructurados. |
| Modelo de datos | El modelo relacional normaliza los datos en tablas conformadas por filas y columnas. Un esquema define estrictamente las tablas, las filas, las columnas, los índices, las relaciones entre las tablas y otros elementos de las bases de datos. La base de datos impone la integridad referencial en las relaciones entre  tablas. | Las bases de datos NoSQL proporcionan una variedad de modelos de datos, como clave-valor, documentos y gráficos, que están optimizados para el rendimiento y la escala. |
| Propiedades ACID | Las bases de datos relacionales ofrecen propiedades de atomicidad, coherencia, aislamiento y durabilidad (ACID):   * La atomicidad requiere que una transacción se ejecute por completo o no se ejecute en absoluto. * La coherencia requiere que, una vez confirmada una transacción, los datos deban acoplarse al esquema de la base de datos. * El aislamiento requiere que las transacciones | Las bases de datos NoSQL a menudo hacen concesiones al flexibilizar algunas de las propiedades ACID de las bases de datos relacionales para un modelo de datos más flexible que puede escalar horizontalmente. Esto hace que las bases de datos NoSQL sean una excelente opción para casos de uso de baja latencia y alto rendimiento que necesitan escalar horizontalmente más allá de las limitaciones de una sola instancia. |

|  |  |  |
| --- | --- | --- |
|  | simultáneas se ejecuten por separado.   * La durabilidad requiere la capacidad de recuperarse de un error inesperado del sistema o de un corte de energía y volver al último   estado conocido. |  |
| Rendimiento | Normalmente, el rendimiento depende del subsistema de disco. Se necesita la optimización de consultas, índices y estructura de tabla para lograr el máximo  rendimiento. | El rendimiento es, por lo general, depende del tamaño del clúster de hardware subyacente, la latencia de red y la aplicación que efectúa la  llamada. |
| Escalado | Las bases de datos relacionales generalmente escalan en forma ascendente las capacidades de computación del hardware o la ampliación mediante la adición de réplicas para cargas de trabajo de solo lectura. | Las bases de datos NoSQL normalmente se pueden particionar porque los patrones de acceso son escalables mediante el uso de arquitectura distribuida para aumentar el rendimiento que proporciona un rendimiento constante a una escala casi  ilimitada. |
| Api | Solicita almacenar y recuperar datos que están comunicados mediante consultas que se ajustan a un lenguaje de consulta estructurado (SQL). Estas consultas son analizadas y ejecutadas por la base de datos relacional. | Las API basadas en objetos permiten a los desarrolladores almacenar y recuperar fácilmente estructuras de datos. Las claves de partición permiten que las aplicaciones busquen pares de clave-valor, conjuntos de columnas o documentos semiestructurados que contengan atributos y objetos de aplicación  serializados. |

Tabla 1 Cuadro comparativo de las bases de datos relacionales y no relacionales.

Fuente: (https://aws.amazon.com/es/nosql/)

* 1. Desarrollo de aplicaciones móviles

Con la llegada de la quinta generación 5G de telefonía móvil es necesario incorporar nuevos requerimientos a las aplicaciones dependiendo del perfil del usuario, gracias a esto la programación móvil *“ha abierto la posibilidad de poner en juego la creatividad, el ingenio, el conocimiento y sobre todo el deseo de satisfacer los requerimientos sociales de esta nueva era, de muchos desarrolladores de software de incontables equipos móviles que ofertan el mercado tecnológico existente.”* (Villota, 2015)

En la actualidad, la mayoría de los servicios móviles están desarrollados en diferentes lenguajes de programación: HTML 5, WAP, Java, C#, Silverlight, Dark, Kotlin, Swift entre otros, por otra parte, existen aplicaciones nativas para un especifico Sistema Operativo Móvil como: Android y iOS (Gasca, Camargo & Medina, 2013).

Para maximizar su presencia en el mercado, una aplicación móvil debe correr en la mayor cantidad de dispositivos posible, así mismo trata con diversas plataformas, estándares, protocolos y tecnologías de red. Una solución consiste en el desarrollo nativo de la aplicación en cada una de las plataformas existentes utilizando el entorno de desarrollo integrado (IDE por sus siglas en inglés), el lenguaje y las herramientas propias de cada plataforma (Spyros & Stelios, 2013).

* + 1. Aplicaciones móviles multiplataforma

El desarrollo multiplataforma procura optimizar la relación costo/beneficio compartiendo la misma codificación entre las versiones para las distintas plataformas. Entre otras ventajas sobresalen: menor tiempo y costo de desarrollo; prestaciones cercanas a las nativas con acceso al hardware del dispositivo y disponibilidad de entornos potentes de desarrollo (Delphi, Visual Studio) o, en su lugar, utilización de tecnologías bien conocidas por los desarrolladores web (HTML5, Javascript y CSS) que pueden trasladar sus conocimientos y experiencias al paradigma móvil. Las aplicaciones multiplataforma pueden clasificarse en: aplicaciones web móviles, nativas e híbridas en la figura 3 se logra apreciar algunas características.



Figura 3 Características de aplicaciones web, nativa e hibrida. Fuente: (https://cmapspublic2.ihmc.us/rid=1NTQ9NMKD-R1SKBP- 24M4/Aplicaciones%20moviles%20hibridas-

%20lo%20mejor%20de%20dos%20mundos.pdf)

* + - 1. Aplicaciones web

Para Leandro (2013), una aplicación móvil web es *“cualquier aplicación que es accedida vía web por una red como internet o intranet. Una de las ventajas de las aplicaciones web cargadas desde internet u otra es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes, también la posibilidad de ser ejecutadas en múltiples plataformas”*

Por otra parte, para Delia, Galdamez, Thomas, Corbalan & Pesado (2015), este tipo de aplicaciones, están diseñadas para correr dentro de un navegador, se desarrollan con tecnología web (HTML, CSS y JavaScript) y cuentan con una serie de características favorables: no necesitan adecuarse a ningún entorno operativo, son independientes de la plataforma y su puesta en marcha es rápida y sencilla.

Las aplicaciones web tienen varios usos, tanto para los usuarios como para los desarrolladores, entre las que se cita Adobe (2021):

* + - * + Permitir a los usuarios localizar información de forma rápida y sencilla en un sitio web en el que se almacena gran cantidad de contenido.
        + Recoger, guardar y analizar datos suministrados por los visitantes de los sitios.
        + Actualizar sitios web cuyo contenido cambia constantemente.

Leandro (2013) menciona algunas características principales de Aplicaciones Web:

* + - * + El usuario puede acceder fácilmente a estas aplicaciones empleando un navegador web (cliente) o similar.
        + Si es por internet el usuario puede entrar desde cualquier lugar del mundo donde tenga un acceso a internet.
        + Pueden existir miles de usuarios, pero una única aplicación instalada en un servidor, por lo tanto, se puede actualizar y mantener una única aplicación y todos los usuarios verán los resultados inmediatamente.
        + Emplean tecnología como Java, JavaFx, JavaScript, DHTML, Flash, Ajax, otros, que dan gran potencia a la interfaz de usuarios.
        + Emplean tecnologías que permiten una gran portabilidad entre diferentes plataformas. Por ejemplo, una aplicación web flash podría ejecutarse en un dispositivo móvil, en una computadora con Windows, Linux u otro sistema, en una consola de video juegos, etc.
      1. Aplicaciones nativas

Las aplicaciones nativas son aquellas que se conciben para ejecutarse en una plataforma específica, es decir, se debe considerar el tipo de dispositivo, el sistema operativo a utilizar y su versión (Delia, Galdamez, Thomas, Corbalan & Pesado, 2015).

Este tipo de aplicaciones requieren un desarrollo para cada sistema operativo, tienen mayor rendimiento, y su interfaz concuerda con el aspecto del sistema en la mayoría de los casos; mientras que las aplicaciones web funcionan en todas las plataformas, para lo cual debe solo considerarse la compatibilidad con el navegador (Villota, 2015).

Entre las ventajas que estas aplicaciones nativas LanceTalent (2014) menciona:

* + - * + Acceso completo al dispositivo.
        + Mejor experiencia del usuario.
        + Visibilidad en APP Store.
        + Envío de notificaciones o avisos a los usuarios.
        + La actualización de la app es constante.

Claramente estas ventajas se pagan con un mayor costo de desarrollo, pues se debe utilizar un lenguaje de programación diferente según la plataforma. Por ende, si se desea cubrir varias plataformas, se deberá generar una aplicación para cada una de ellas. Esto conlleva a mayores costos de actualización y distribución de nuevas versiones (Delia, Galdamez, Thomas, Corbalan & Pesado, 2014).

* + - 1. Aplicaciones hibridas

Las aplicaciones híbridas combinan lo mejor de los dos tipos de aplicaciones anteriores. Se utilizan tecnologías multiplataforma como HTML, JavaScript y CSS, pero se puede acceder a buena parte de las capacidades específicas de los dispositivos, son desarrolladas utilizando tecnología web y son ejecutadas dentro de un contenedor web sobre el dispositivo móvil (Delia, Galdamez, Thomas, Corbalan & Pesado, 2014).

Delgado (2013) menciona que las aplicaciones hibridas permiten:

* + - * + Crear un núcleo de código común utilizando tecnologías web.
        + Complementarlo de forma opcional con elementos desarrollados con tecnologías nativas de cada plataforma.
        + Encapsularlo en un contenedor específico para cada plataforma, de manera que la aplicación pueda interaccionar con el sistema operativo subyacente, acceder a las capacidades del hardware y puede distribuirse a través de los marketplaces oficiales de cada fabricante como Apple Store, Google Play.

Entre las principales ventajas Delia, Galdamez, Thomas, Corbalan & Pesado (2014) nos cita:

* + - * + La posibilidad de distribución de la aplicación a través de las tiendas de aplicaciones.
        + La reutilización de código para múltiples plataformas.
        + La posibilidad de utilizar las características de hardware del dispositivo.

Desde la perspectiva de Delgado (2013) alude que el desarrollo hibrido frente a un 100% nativo, son:

* + - * + Misma experiencia de usuario y mismo rendimiento.
        + Menor time-to-market y menores riesgos asociados con el lanzamiento de nuevas aplicaciones.
        + Aprovechamiento del conocimiento y los recursos acumulados en el desarrollo con tecnologías web.
        + Un núcleo de código común a todas las plataformas
  1. Framework para el desarrollo móvil

Para Gutiérrez (2014), el termino de framework se refiere a una estructura de software compuesta de componentes personalizados e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Por otra parte, para Acens Technologies (2014) menciona que un framework es un conjunto de archivos y directorios que facilitan la creación de aplicaciones, ya que incorporan funcionalidades ya desarrolladas y probadas, implementadas en un determinado lenguaje de programación.

El objetivo principal de los framework es acelerar los procesos de desarrollo, reutilizar código, así como promover buenas prácticas de desarrollo como el uso de patrones.

Actualmente existen muchos framework enfocado al desarrollo móvil multiplataforma, los más utilizados son React Native, Ionic y Xamarin. A este conjunto se ha unido recientemente Flutter, que soluciona los principales problemas del resto de framework de desarrollo móvil multiplataforma, (Zazo, 2019).

* + 1. Frameworks

La aplicación móvil se desarrollará usando Flutter, no obstante, existen diferentes framework en el mercado para el desarrollo de aplicaciones multiplataformas, en el siguiente apartado se menciona los framework más utilizados actualmente, para tener un panorama de sus diferentes características.

* + - 1. React Native



Figura 4 Logo de React Native. Fuente: (https://reactnative.dev/docs/getting-started)

Creado por Facebook, React Native es un framework utilizado principalmente para el desarrollo de aplicaciones móviles, aunque también permite trabajar para la plataforma universal de Windows. La idea principal es permitir el uso de React, una famosa librería para la construcción de interfaces de usuario mantenida por Facebook, en las plataformas nativas, (Evkoski, 2018).

Una aplicación construida con esta herramienta usa componentes React para la interfaz y código JavaScript para la lógica, permite utilizar las mismas librerías usadas para el desarrollo de aplicaciones web, pero para plataformas móviles.

Esto resulta que esta herramienta sea especialmente atractiva para desarrolladores u organizaciones que ya tengan aplicaciones web implementadas con estas tecnologías. Sin embargo, la librería React tiene una curva de aprendizaje pronunciada, lo que dificulta su adopción por parte de usuarios nuevos.

Las aplicaciones desarrolladas con este framework no se compilan a las plataformas de destino, sino que su código es interpretado durante la ejecución en el dispositivo. Esto hace que sea más difícil detectar errores en el código durante el desarrollo, además de suponer un peor rendimiento frente a alternativas que sí que compilan el código, como es el caso de Flutter (Vázquez, 2019).

* + - 1. Ionic



Figura 5 Logo de Ionic. Fuente: (https://ionicframework.com)

Ionic es un framework de código abierto basado en la tecnología de Apache Cordova que permite crear aplicaciones móviles híbridas. A diferencia de las aplicaciones nativas, las aplicaciones híbridas son implementadas usando tecnologías web (HTML, CSS y JavaScript) y ejecutadas en el dispositivo de destino aprovechando las capacidades de su motor de navegador, (Ravularu, 2017).

El HTML usado para definir la interfaz es renderizado durante la ejecución, al igual que ocurre con el código JavaScript, el cual se interpreta sobre la marcha en el dispositivo. Como ocurría con React Native, el principal inconveniente de este acercamiento es la pérdida de rendimiento. Sin embargo, si comparamos estas dos herramientas, React Native sale favorecida por el hecho de apoyarse en un único lenguaje, JavaScript, en vez del conglomerado de tecnologías que usa Ionic (Vázquez, 2019).

* + - 1. Xamarin



Figura 6 Logo de Xamarin. Fuente: (https://dotnet.microsoft.com/apps/xamarin

En sus comienzos, Xamarin era una empresa que desarrolló implementaciones de la plataforma .NET de Microsoft para plataformas móviles. Microsoft acabó comprando esta empresa y usando su nombre para identificar al conjunto de herramientas que permiten utilizar una base de código C# para el desarrollo de aplicaciones en diversas plataformas, permitiendo la compartición de código entre estas. Con Xamarin, se puede usar C# no sólo para la implementación de la lógica de negocio, sino que también provee de controles de interfaz personalizados para cada plataforma, (Vázquez, 2019).

Sin embargo, esto hace que la implementación de estas interfaces con C# no se pueda compartir entre plataformas, haciendo que el porcentaje de código

compartido sea mucho menor que en otras soluciones, como Flutter. Por último, otra de las desventajas de Xamarin es el gran tamaño de sus aplicaciones. Las aplicaciones desarrolladas con esta herramienta se compilan para cada plataforma, pero siguen teniendo un tamaño que, en ocasiones, es notablemente mayor que el de aplicaciones nativas.

* + 1. Tecnología framework a utilizar

Después de analizar las diferentes tecnologías presentadas en el apartado 2.3.1, permite construir un panorama del Framework que mayor se adapte al proyecto. React Native tiene una simplicidad y un uso único de tecnología, la cual permite construir aplicaciones mucho menos pesadas que Xamarin, así mismo la madurez de la herramienta y el gran catálogo de documentación que cuenta React Native.

Sin embargo, el punto débil de este Framework es su código el cual es interpretado durante la ejecución, lo que provoca que el acceso a las funcionalidades nativas de la plataforma se realice a través de un puente, causando problemas de rendimiento, por otro lado, Ionic cuenta con este tipo de problema aparte de que sus aplicaciones son demasiadas pesadas para los dispositivos.

Por otra parte, nos encontramos con Flutter, el cual no sufre con ninguna de estas limitaciones, este framework usa un diferente lenguaje de programación el cual tiene por nombre Dart, una de las principales características que nos encontramos con este framework es que su código se compila en hot realoading, al hacer algún cambio al código se podrán ver los efectos reflejados inmediatamente, sin tener que compilar la aplicación de nuevo.

Flutter al ser un framework multiplataforma nos ofrece un código único para aplicaciones de Android y de iOS al mismo tiempo. Asimismo, otorga acceso a funciones nativas que la aplicación necesite para su funcionamiento, sin necesidad de llamar las funciones, ya que en otros framework requieren que se implementen mediante lenguajes nativos.

# Tecnología implementada

* + 1. Android Studio



Figura 7 Logo de Android Studio. Fuente: (https://developer.android.com/studio)

Android Studio es el entorno de desarrollo oficial para la creación de aplicaciones para Android. Está basado en otro entorno llamado IntelliJ IDEA, con el que comparte el editor de texto y otras herramientas para desarrolladores. A estas herramientas se añaden otras funciones específicas para el desarrollo de aplicaciones Android (Android, 2021).

Algunas de estas funciones que aporta Android Studio para facilitar el desarrollo son:

* + - * Un sistema de compilación flexible basado en Gradle.
      * Un rápido emulador con varias funciones.
      * Un entorno unificado en el que se pueden realizar desarrollos para todos los dispositivos Android (móviles, wearables, TV, …).
      * Visualización de los cambios realizados a la aplicación sin la necesidad de compilar una nueva APK.
      * Gran cantidad de herramientas y framework de prueba.
      * Herramientas de inspección del código para detectar problemas de rendimiento, usabilidad, compatibilidad, etc.
      * Soporte integrado para Google Cloud Platform, la plataforma de computación en la nube de Google.

Además de todo esto, en la instalación de Android Studio se incorpora también el SDK de Android y sus herramientas. Esto es necesario para poder compilar una APK que se pueda ejecutar en un dispositivo Android.

* + 1. Dart



Figura 8 Logo de Dart. Fuente:(https://dart.dev/)

Dart es un lenguaje de programación de código abierto, orientado a objetos, basado en clases y con recolector de basura (garbage collector), desarrollado por Google y presentado por primera vez en 2011, aunque su primera versión estable no llegó hasta 2013, su principal objetivo era ofrecer una alternativa más moderna a JavaScript como principal lenguaje de programación web, (Zazo, 2019).

A diferencia de muchos lenguajes, Dart se diseñó con el objetivo de hacer el proceso de desarrollo lo más cómodo y rápido posible para los desarrolladores, cuenta con un conjunto bastante extenso de herramientas integrado, como su

propio gestor de paquetes, varios compiladores/transpiladores, un analizador y formateador. Además, la máquina virtual de Dart y la compilación Just-in-Time hacen que los cambios realizados en el código se puedan ejecutar inmediatamente, en cuanto a la sintaxis de Dart es muy similar a lenguajes como JavaScript, Java y C++ (inLab, 2020).

* + 1. Flutter



Figura 9 Logo de Flutter. Fuente: (https://flutter.dev/)

Flutter es un framework de desarrollo de aplicaciones móviles multiplataforma creado por Google. Es de código abierto y permite construir aplicaciones tanto para Android como para iOS. Su versión 1.0 fue lanzada al mundo el 4 de diciembre de 2018, por lo que es una tecnología muy nueva. A pesar de su corta edad, se trata de una tecnología muy madura debido a que es utilizada en Google para crear sus herramientas internas, (Vázquez, 2019).

Por otra parte, Flutter utiliza el lenguaje de programación Dart, también desarrollado por Google, tanto para la interfaz de usuario como para la lógica de la aplicación. Todo el código escrito en Dart se compila utilizando AOT a código máquina en ambas plataformas para así obtener el mayor rendimiento posible (Zazo, 2019).

Vázquez (2019) menciona que el objetivo de esta herramienta es permitir a los desarrolladores construir aplicaciones multiplataforma a partir de una única base de código, la cual es compilada a código nativo para cada una de las plataformas objetivo. Además, se aprovecha de la flexibilidad de Dart en cuanto a su compilación y ejecución para obtener ciclos de desarrollo más rápidos y tiempos de ejecución más bajos.

Actualmente Flutter permite compilar la aplicación a Android y iOS, aunque en el futuro también permitirá compilar aplicaciones de escritorio (Windows, macOS, Linux), aplicaciones web, aplicaciones para dispositivos embebidos (Raspberry Pi), e incluso en el Google I/O de 2019 se mostró una aplicación desarrollada en Flutter ejecutándose en un televisor (Android TV) y en un reloj (Wear OS) (Beyonce Mobile, 2019).

En la figura 10 se puede observar los componentes que permiten el funcionamiento de Flutter, en los que se observan dos bloques principales: Framework y Engine. El primero, escrito en Dart, es el que permite al programador desarrollar la aplicación utilizando Flutter y sus componentes. El segundo bloque es el motor que permite ejecutar estas aplicaciones, escrito en C++ y de código abierto, con el que el programador no necesita interactuar directamente dado que Flutter gestiona automáticamente la interacción con dicho motor del framework (Vázquez, 2019).

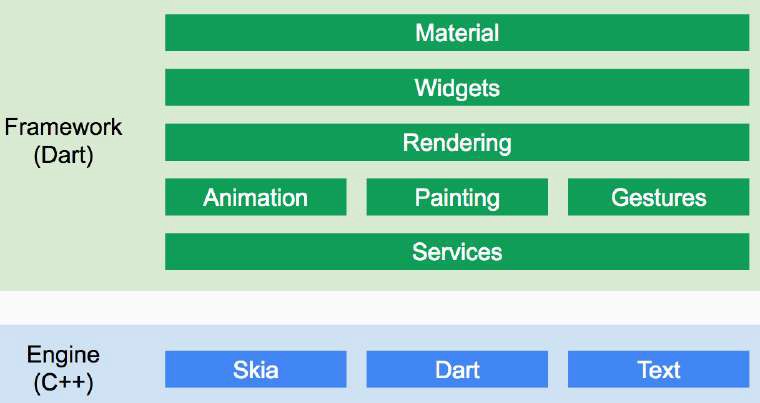


Figura 10 Estructura de flutter. Fuente: (https://medium.com/maybeoscar/flutter-i-introducci%C3%B3n-al- desarrollo-8c2250d60186)

* + - 1. Manejadores de estado flutter

Los manejadores de estado en Flutter se refiere a la forma de administrar los datos importantes de una aplicación, estos datos tienen que estar en una estructura que nos facilite poder mostrar la información en pantalla y actualizarla de forma fácil y eficiente, en esta aplicación se implementara el patrón BLoC y provider.

* + - * 1. BLoc

El patrón BLoC (Business Logic Component) fue diseñado por Paolo Soares y Cong Hu de Google y fue presentado en la Dart Coference 2018. El objetivo de este patrón, es la reutilización de código entre plataformas (móvil y web), utilizando Flutter con Dart y web utilizando Angular Dart cabe mencionar que el concepto principal es que exista una capa intermedia entre las vistas y el modelo. Esta capa gestionará los estados y manejará los datos dependiendo de los eventos que se reciban de la vista (Sánchez, 2020).

Como ya se mencionó anteriormente un BLoC es un componente intermediario entre las vistas y el modelo, este pude estar presente cuando se utiliza el MVP (Model–View–Presenter) o MVVM (Model–View–ViewModel), el cual cuenta con los siguientes puntos:

Centralizar la lógica de negocio: Este se centra en la parte de un sistema que se encarga de codificar las reglas de negocio del mundo real que determinan como la información puede ser creada, almacenada y cambiada.

Centralizar cambios de estado: Estas clases BLoC van a centralizar los cambios de estados que se producen en la aplicación, de esta forma las clases BLoC se encargaran de recibir las acciones o eventos que se producen en la aplicación móvil y que modifican el estado, así mismo son las responsables de comunicar a todas las partes implicadas (componentes o widgets).

Mapear al formato que necesita la vista: BLoC se encarga de formatear los datos a como lo necesitan las vistas, de esta forma esta lógica de presentación de formateo es también reutilizable

Un ejemplo de la utilización de BLoC seria en la aplicación predeterminada que flutter nos genera al crear un proyecto nuevo, aquí solo hay un evento, y ese es el incrementCouter el cual genera el nuevo estado, que se encarga de mostrar el valor incrementado del contador, en la figura 11 se muestra la estructura de la creación del BLoC.



Figura 11 Ejemplo de patrón BLoC. Fuente: (https://medium.com/comunidad- flutter/simplificando-la-gesti%C3%B3n-de-estado-con-bloc-en-flutter- 9d5670cd97d8)

BLoC es visto como una caja negra que le dirá que operación debe realizar en que evento y luego que estado debe generar después de que complete la operación. Este puede ser utilizado tanto en framework imperativos como declarativos, pero por su capacidad reactiva, encaja mucho mejores entornos declarativos donde además hay una necesidad de compartir estado entre diferentes vistas.

* + - * 1. Provider

Provider, tiene como función principal, permitir manejar el estado de una aplicación. Como se sabe, en Flutter para poder construir una aplicación, y alterar el aspecto visual de la misma, se debe trabajar con el “Estado de la App”. La forma más sencilla de hacer esto, es mediante el uso de “StateFul Widgets” y la función de “setState”. Eso puede funcionar para alguna aplicación muy pequeña, sin embargo, cuando se está construyendo una app que contiene varias pantallas y hay que acceder a esas propiedades, el proceso se vuelve muy complejo, pero existe una alternativa como el patrón BLoC, (Ram, 2019)

Por otro lado, Waldmana (2022), menciona que Provider, es encantador ya que permite tener toda la lógica de negocio en un archivo independiente, sin necesidad de tanto código. Es fácil de entender, pero tiene una fuerte desventaja, un cambio de estado, de datos, etcétera, renderiza nuevamente todas las vistas, ocasionando consumos y bajas de rendimiento, pero al juntar Provider y el rendimiento del patrón BLoC crean una aplicación mucho más completa ya que los dos manejadores de datos se complementan.

* + 1. Visual Studio Code



Figura 12 Logo de Visual Studio Code. Fuente: (https://code.visualstudio.com/docs)

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) (VSC, 2021).

El verdadero potencial de este editor se encuentra en la personalización. Más allá de la infinidad de temas y fuentes de entre los que elegir, podemos instalar extensiones de su mercado propio, la mayoría de ellas mantenidas por la comunidad. Estas extensiones añaden soporte para lenguajes específicos o proporcionan herramientas adicionales para la realización de algunas tareas (Vázquez, 2019).

* + 1. Material Design



Figura 13 Logo de Material Design. Fuente: (https://material.i o/design)

Material Design es un lenguaje visual creado por Google para la creación de interfaces de usuario. Trata de juntar los principios básicos del buen diseño junto con las innovaciones en tecnología y ciencia, (Vázquez, 2019).

Los componentes de material son bloques de construcción interactivos para crear una interfaz de usuario e incluyen un sistema de estados integrado para comunicar los estados de enfoque, selección, activación, error, desplazamiento, presión, arrastre y deshabilitación. Las bibliotecas de componentes están disponibles para Android, iOS, Flutter y la web, (Google, 2021).

* + 1. Postman



Figura 14 Logo de Postman. Fuente: (https://[www.postman.com/)](http://www.postman.com/))

Postman surgió originariamente como una extensión para el navegador Google Chrome. A día de hoy dispone de aplicaciones nativas para MAC y Windows y están trabajando en una aplicación nativa para Linux, (Redondo, 2021).

Postman es una herramienta que se utiliza para crear peticiones sobre API de una forma muy sencilla y de esta manera, probar las API, por otra parte, se pude monitorear, escribir pruebas automatizadas para luego documentarlas, (Cuervo, 2019).

Es importante destacar también que, aunque no sea una de las herramientas más famosas para documentar API REST, genera una documentación bastante interesante y bastante atractiva, con ejemplos y snippets de código, de forma que hace que sea muy fácil de entender cómo funciona una API determinada, (Lopez, 2021).

* + 1. Xcode



Figura 15 Logo de Xcode.

Fuente: (https://developer.appl e.com/xcode/)

Xcode es un IDE completo, con conjunto de herramientas para desarrolladores que permite crear apps para Mac, iPhone, iPad, Apple Watch y Apple TV. Xcode combina las funcionalidades de diseño de interfaz de usuario, programación, pruebas, depuración y envío a App Store en un flujo de trabajo unificado. Este entorno de desarrollo integrado, es compatible con variados lenguajes de programación (Apple, 2021).

* + 1. Sistemas gestores de bases de datos

Tradicionalmente, para manipular grandes cantidades de datos aparecieron en primer lugar las bases de datos o "bancos de datos", cumpliendo de manera aproximada con la definición presentada en el apartado anterior. Se trataba de almacenar ordenadamente datos en un juego de ficheros, y mediante unas aplicaciones informáticas y un sistema de índices gestionarlas adecuadamente, (Quintas, 2008).

Al aumentar la complejidad de estos bancos de datos se producen problemas de eficiencia, así como de integridad y accesibilidad de los datos.

Para solucionar estos problemas salen al mercado los sistemas de gestión de bases de datos (SGBD), los cuales convierten el acceso a los datos y su gestión en una aplicación cerrada, interponiéndose entre los usuarios y los ficheros, haciéndose cargo de todos los problemas de explotación, mantenimiento y comprobación de los datos.

* + - 1. Gestores de bases de datos más utilizados actualmente
         1. MySQL



Figura 16 Logo de MySQL. Fuente: (https://[www.mysql.com/)](http://www.mysql.com/))

MySQL es un sistema de gestión de base de datos (SGBD) de código abierto. El SGBD MySQL pertenece actualmente a Oracle. Funciona con un modelo cliente- servidor. Eso quiere decir que los ordenadores que instalan y ejecutan el software de gestión de base de datos se denominan clientes. Cada vez que necesitan acceder a los datos, los clientes se conectan al servidor del sistema de gestión

de base de datos y le solicitan la información que necesitan, el servidor se la brinda siempre y cuando tenga los derechos de acceso (Carrera, 2021).

Características MySQL

Viviana (2018) menciona que una de las características que distingue a MySQL de otros gestores de bases de datos es su amplia comunidad, así mismo las consultas que se manejan con un lenguaje estructurado como lo es SQL, por otra parte, la información será consistente entre todos los nodos mientras los mismos estén disponibles, cabe mencionar que cuentan con una conexión entre el server y el cliente de tipo SSL lo cual lo hace más seguro.

* + - * 1. MongoDB



Figura 17 Logo de MongoDB. Fuente: (https://[www.mongodb.com/)](http://www.mongodb.com/))

MongoDB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. Ninguna otra ofrece un nivel de productividad de uso tan alto, (MongoDB, 2021).

Como ya se mencionó MongoDB es una base de datos documental, lo que significa que almacena datos en forma de documentos tipos JSON, por ende, facilita la lectura de los datos; frente al tradicional modelo de filas y columnas.

2.4.8.1.2.2 Características de MongoDB

A continuación, Gómez (2020) menciona que la principal característica de MongoDB es la potente sintaxis de cara a las consultas, lo que permite hacer consultas, desde las más sencillas hasta las más complejas, obteniendo todo tipo de información, por otro lado cuenta con un amplio soporte para SQL, aunque normalmente este tipos de bases de datos NoSQL están enfocadas para usar otro tipo de lenguaje de consulta, para todos los que tienen experiencia con las mismas, el uso de SQL está muy arraigado, así que MongoDB ofrece este soporte para que lo puedan seguir utilizando en los proyectos.

Así mismo cuenta con escalabilidad vertical la cual es posible de aumentar los recursos relacionado con la memoria o la CPU del servidor en el que esta MongoDB. La escalabilidad horizontal es la posibilidad de crear diferentes nodos, que permiten aumentar la disponibilidad de la aplicación conforme el volumen de los datos o el número de accesos a dicha base datos aumenta.

* + - 1. Diferencias MySQL y MongoDB

Como ya se mencionó anteriormente MySQL es un sistema de administración de base de datos relacionales, de uso gratuito y código abierto desarrollado por Oracle. Al igual que otros sistemas de bases de datos MySQL almacena datos usando tablas y filas, refuerza la integridad referencial y usa lenguaje de consulta estructurado el cual se conoce como SQL.

Los esquemas de la base de datos y los modelos de datos deben definirse con anticipación, y los datos deben coincidir con este esquema para poder almacenarse en la base de datos. Este enfoque rígido para almacenar datos ofrece cierto grado de seguridad, pero lo cambia por flexibilidad. Si es necesario almacenar un nuevo tipo o formato de datos en la base de datos, debe ocurrir la migración del esquema, lo que puede volverse complejo y costoso a medida que aumenta el tamaño de la base de datos, (MongoDB, 2021).

Por otro lado, MongoDB también es gratuito y de código abierto, sin embargo, este sistema cambio por completo el esquema de los sistemas relacionales tradicionales ya que se basa en un diseño de sistema no relacional también conocido NoSQL. MongoDB se enfoca en almacenar datos, en JSON los cual hace más fácil su consulta a diferencia de la tabla y formato de fila de sistemas relacionales.

A continuación, muestran las diferentes diferencias que existen entre MongoDB y MySQL:

* + - * 1. Facilidad de Uso:

Una de las diferencias que se encuentra entre MongoDB y MySQL es la facilidad de uso de MongoDB ya que su filosofía de almacenamiento de datos es simple e inmediatamente compresible para cualquier persona con experiencia en programación, por otra parte, MongoDB almacena datos en colecciones sin un esquema obligatorio este enfoque flexible para almacenar datos lo hace particularmente adecuado para desarrolladores que pueden no ser expertos en bases de datos, pero quieren usar una base de datos para respaldar el desarrollo de sus aplicaciones.

MongoDB proporciona una interfaz de desarrollador flexible para equipos que están creando aplicaciones que no necesitan todas las características de seguridad que ofrecen los sistemas relacionales. Un ejemplo común de una aplicación de este tipo es una aplicación web que no depende de esquemas estructurados; puede servir fácilmente datos no estructurados, semiestructurados o estructurados, (MongoDB, 2021).

Cabe recalcar que MongoDB admite propiedades ACID de transacciones (atomicidad, consistencia, aislamiento y durabilidad). Esto permite una mayor flexibilidad en la creación de un modelo de datos transaccionales que se puede escalar horizontalmente en un entorno distribuido y no tiene ningún impacto en el rendimiento de las transacciones de varios documentos.

En comparación con MySQL es una opción común para los usuarios que tienen una amplia experiencia en el uso de secuencias de comandos SQL tradicionales, el diseño de soluciones para bases de datos relacionales o que están modificando o actualizando aplicaciones existentes que ya funcionan con un sistema relacional.

* + - * 1. Escalabilidad:

Un gran beneficio que ofrece MangoDB es la base de datos es extremadamente fácil de escalar. La cual por medio de un clúster fragmentado permite que una parte de la base de datos, denominada fragmento, también se configure como un conjunto de réplicas. En un clúster fragmentado, los datos se distribuyen entre muchos servidores. Este enfoque altamente flexible permite a MongoDB escalar horizontalmente el rendimiento de lectura y escritura para adaptarse a aplicaciones de cualquier escala.

En cambio, con un sistema con MySQL las opciones de escalabilidad son mucho más limitadas. Normalmente, tiene dos opciones: escalabilidad vertical o agregar réplicas de lectura. Escalar verticalmente implica agregar más recursos al servidor de base de datos existente, pero esto tiene un límite superior inherente.

* + - * 1. Rendimiento

Comparar el rendimiento de dos sistemas de bases de datos completamente diferentes es complicado, ya que ambos sistemas de gestión abordan la tarea de almacenamiento y recuperación de datos de formas completamente diferentes. Si bien es posible comparar directamente dos bases de datos SQL con un conjunto de puntos de referencia SQL estándar, lograr lo mismo en bases de datos relacionales y no relacionales es mucho más difícil y subjetivo.

Un ejemplo de comparación de rendimiento de las bases de dato que nos brinda MongoDB (2021) es:

*MySQL está optimizado para uniones de alto rendimiento en varias tablas que se han indexado adecuadamente. En MongoDB, las uniones son compatibles con la operación de búsqueda “$”, pero son menos necesarias debido a la forma en que tienden a usarse los documentos de MongoDB; siguen un modelo de datos jerárquico y mantienen la mayoría de los datos en un documento, por lo que eliminan la necesidad de uniones en varios documentos.*

*MongoDB también está optimizado para el rendimiento de escritura y cuenta con una API insertMany () específica para insertar datos rápidamente, priorizando la velocidad sobre la seguridad de las transacciones, donde los datos MySQL deben insertarse fila por fila.*

*Al observar algunos de los comportamientos de consulta de alto nivel de los dos sistemas, podemos ver que MySQL es más rápido para seleccionar una gran cantidad de registros, mientras que MongoDB es significativamente más rápido para insertar o actualizar una gran cantidad de registros.*

* + - * 1. Flexibilidad

MongoDB logra que sea bastante fácil crear y mejorar aplicaciones a lo largo del tiempo, sin necesidad de ejecutar procesos de migración de esquemas complejos y costosos como lo haría con una base de datos relacional, así mismo se cuenta con opciones más dinámicas para actualizar el esquema de una colección, como crear nuevos campos basados en una canalización de agregación o actualizar campos de matriz anidados. Este beneficio es particularmente importante a medida que las bases de datos aumentan de tamaño.

Por otro lado, en MySQL entre más grandes sean las bases de datos, estas son más lentas para migrar esquemas y procedimientos almacenados que pueden depender de los esquemas actualizados.

* + - * 1. Seguridad

MongoDB aprovecha el popular modelo de control de acceso basado en roles con un conjunto flexible de permisos. A los usuarios se les asigna un rol, y ese rol les otorga permisos específicos sobre conjuntos de datos y operaciones de bases de datos. Toda la comunicación está encriptada con TLS, y es posible escribir documentos encriptados en colecciones de datos de MongoDB usando una clave maestra que nunca está disponible para MongoDB, logrando el cifrado de datos en reposo.

MySQL admite las mismas funciones de cifrado que MongoDB; su modelo de autenticación también es similar. A los usuarios se les pueden otorgar roles, pero también privilegios, otorgándoles permisos sobre operaciones de bases de datos particulares y contra conjuntos de datos particulares.

* + - 1. Descripción del sistema de base de datos seleccionado

El sistema de base de datos que se selección para este proyecto fue MongoDB ya que cuenta con un enfoque de implementación hibrida que brinda un beneficio de ambos mundos (modelo relacional y no relacional), así mismo por la implementación de la base de datos en la nube, ya que permite crear aplicaciones más rápidas, manejar tipos de datos muy diversos y administrar aplicaciones de manera más eficiente a escala.

En el desarrollo se simplifica ya que los documentos de MongoDB se asignan de forma natural a los lenguajes de programación modernos orientados a objetos. El uso de MongoDB elimina la capa compleja de mapeo relacional de objetos (ORM) que traduce los objetos en código a tablas relacionales. El modelo de datos flexible de MongoDB también significa que el esquema de su base de datos puede evolucionar con los requisitos comerciales. La rígida estructura relacional de MySQL agrega sobrecarga a las aplicaciones y ralentiza a los desarrolladores, ya que deben adaptar los objetos en el código a una estructura relacional.

MongoDB también se puede escalar dentro y a través de múltiples centros de datos distribuidos, proporcionando nuevos niveles de disponibilidad y escalabilidad que antes eran inalcanzables con bases de datos relacionales como MySQL. A medida que sus implementaciones crecen en términos de volumen de datos y rendimiento, MongoDB se escala fácilmente sin tiempo de

inactividad y sin cambiar su aplicación. Por el contrario, lograr escalar con MySQL a menudo requiere un importante trabajo de ingeniería personalizado.

* + - * 1. MongoDB Atlas

Como ya se mencionó anteriormente el sistema de base de datos a utilizar es MongoDB, pero la base de datos que nos ofrece este sistema se llama MongoDB Atlas la cual está basada en la nube y se utilizara para este proyecto.

MongoDB Atlas cuenta con el servicio más innovador en la nube, la cual se enfoca para aplicaciones modernas, está completamente administrado en AWS, Azure y Google Cloud, con la mejor automatización integrada para la optimización de recursos y cargas de trabajo, así mismo garantiza la disponibilidad, escalabilidad y cumplimiento de los estándares de privacidad y seguridad de datos más exigentes.

Algunas de las características que MongoDB Atlas ofrece son:

Distribución de datos en múltiples nubes

Seguridad sofisticados y funcionabilidad para satisfacer las medidas de cumplimiento y privacidad de datos.

Diseñado para crear aplicaciones más rápido con controladores, integraciones y herramientas nativas para manipular, visualizar y analizar datos.

Altamente disponible con tolerancia a fallas distribuidas y opciones de respaldo para cumplir con sus objetivos de recuperación de datos.

Escalado bajo demanda, herramientas de optimización de recursos e información en tiempo real sobre el rendimiento de su base de datos.

Aprovisionamiento de infraestructura, configuración de base de datos, mantenimiento y actualizaciones de versión totalmente automatizados.

* + - * 1. MongoDB Compass

Al igual que MongoDB Atlas se utilizará MongoDB Compass el cual es la GUI (Interfaz Gráfica del Usuario) de MongoDB. Compass permite analizar y comprender el contenido de los datos sin un conocimiento formal de la sintaxis de consulta de MongoDB. Además de explorar los datos en un entorno visual, también Compass optimiza el rendimiento de las consultas, administrar índices e implementar la validación de documentos (MongoDB, 2021).

* + 1. Heroku



Figura 18 Logo de Heroku. Fuente: (https://[www.heroku.com/)](http://www.heroku.com/))

Heroku es una plataforma en la nube como servicio (PaaS) basada en conectores que permite a las empresas construir, entregar, monitorear y escalar aplicaciones, ofrece una enorme rapidez con la que se puede publicar una aplicación y en consecuencia que al momento de publicarla está se encuentre activa y preparada para recibir peticiones sin tener que realizar configuraciones adicionales.

Heroku está completamente administrado, lo que permite a los desarrolladores la libertad de concentrarse en su producto principal sin la distracción de mantener servidores, hardware o infraestructura, brindando servicios, herramientas, flujos de trabajo y soporte políglota, todo diseñado para mejorar la productividad de los desarrolladores.

También cabe destacar la amplia variedad de lenguajes que soporta por lo que la elección de sobre que tecnología se quiere realizar una aplicación web, un proceso de gestión de colas, etcétera.

* + 1. Node JS



Figura 19 Logo de Node JS. Fuente: (https://nodejs.org/es/)

Node.js es un entorno de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript). Este entorno se ejecuta en tiempo real incluyendo todo lo que se necesita para ejecutar un programa escrito en JavaScript.

Node.js fue creado por los desarrolladores originales de JavaScript, lo transformaron de algo que solo podía ejecutarse en el navegador en algo que se

podría ejecutar en los ordenadores como si de aplicaciones independientes se tratara. Gracias a Node.js se puede ir un paso más allá en la programación con JavaScript no solo creando sitios web interactivos, sino teniendo la capacidad de hacer cosas que otros lenguajes de secuencia de comandos como Python pueden crear (Lucas, 2019).

2.4.10.1 Paquetes Node JS

En el siguiente apartado se listará todos los paquetes que se utilizaran en el desarrollo de la aplicación.

|  |  |
| --- | --- |
| **Paquetes** | **Función** |
| bcryptjs | Es una función para almacenar contraseñas evitando que dos  passwords iguales se generen. |
| cloudinary | Es una solución de gestión de imágenes y videos de un extremo a otro para sitios web y aplicaciones móviles, que cubre todo, desde la carga de imágenes y videos, el almacenamiento, las manipulaciones y  las optimizaciones hasta la entrega. |
| cors | Es utilizado para proporcionar permisos a los usuarios para acceder a recursos seleccionados desde un servidor, en un origen distinto  (dominio) al que pertenece. |
| dotenv | Es un módulo para acceder a las  variables de entorno de una aplicación. |

|  |  |
| --- | --- |
| express | Es un framework web que permite estructurar una aplicación web para manejar múltiples solicitudes http diferentes en una dirección url  especifica. |
| express-fileupload | Es un módulo que provee múltiples opciones para la carga y administración de archivos dentro de  una aplicación Node JS. |
| express-validator | Se encarga de validar solicitudes y devuelve una respuesta con errores si es que falla alguna regla de validación  configurada en la aplicación. |
| json web token | Es un token de acceso estandarizado que permite el intercambio seguro de datos entre dos partes, contiene toda la información importante de una  entidad. |
| mongoose | Permite definir esquemas con datos fuertemente tipados, una vez que se define un esquema, Mongoose, permite crear un modelo basado en un  esquema específico. |
| uuid | Es utilizado para generar identificadores únicos en el sistema. Con ella crearemos un uuid que le asignaremos a cada petición de  manera única. |

Tabla 2 Lista de paquetes Node JS. Fuente: Elaboración propia

* + 1. Rest Server

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP que sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON. El formato más utilizado actualmente es el formato JSON, ya que es más ligero y legible en comparación al formato XML (Rosa, 2018).

Actualmente muchas empresas como Google, Facebook, Netflix, Twitter y otros miles de empresas usan REST. Una de sus principales ventajas que ofrece es la escalabilidad ya que separa los conceptos de cliente y servidor, lo que genera que se centren únicamente en la parte del servidor.

El diseño arquitectónico del Rest está conformado por:

* + - * Un sistema de cliente-servidor.
      * Tiene que ser sin estado, esto se refiere a que los servicios guarden las sesiones de usuarios (cada petición al servicio tiene que ser independiente de las demás).
      * Tiene que ser escalable.
      * Debe ser un sistema uniformemente accesible, cada recurso debe tener una única dirección y un punto válido de acceso.
      * La estructura tiene que soportar un sistema de caches.
      * Debe admitir mensajes autodescriptivos, los cuales deben ser accedidos en una gran variedad de formatos como por ejemplo XML, HTML, TEXTO PLANO, PDF, JPEG, JSON, etcétera.

Los servicios REST suelen asignar los cuatro métodos HTTP principales a las operaciones que realizar: CREATE, RETRIEVE, UPDATE y DELETE conocido como CRUD. En la siguiente tabla se muestra la correspondencia entre las acciones CRUD sobre los datos y los métodos HTTP correspondientes:

|  |  |  |
| --- | --- | --- |
| **Acción sobre los**  **datos** | **Protocolo HTTP**  **equivalente** | **Operación que realiza** |
| CREATE | POST | Crea datos y otras operaciones, ya que no  tiene semántica definida |
| RETRIEVE | GET | Obtiene datos |
| UPDATE | PUT | Crea o actualiza datos |
| DELETE | DELETE | Elimina datos |

Tabla 3 CRUD. Fuente: Elaboración propia

# Metodología

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.”, (Avison y Fitzgerald, 2006).

Los aportes que una metodología aporta a los proyectos de desarrollo son:

* Facilidad de planificación.
* Facilidad de control y seguimiento adecuado de proyectos.
* Mejora de uso en los recursos.
* Permiten evaluar de forma más fácil los resultados obtenidos y valorar los objetivos conseguidos.
* Mejora la comunicación entre el cliente y las personas que van a llevar a cabo el proyecto.
* Garantiza que el proyecto final tendrá la calidad esperada.
* Se tendrá presentes unos plazos para el desarrollo del producto.
* Permitirá definir el ciclo de vida adecuado al proyecto.

Los factores a considerar para seleccionar una metodología adecuada a un proyecto dependerán del tamaño de la organización, experiencia profesional y la

capacidad de innova, las herramientas técnicas de las que se dispone y el tipo de proyecto a desarrollar (Trigas, 2012).

* + 1. Metodología Scrum

Dentro de las distintas metodologías ágiles que existen, SCRUM es un conjunto de prácticas para trabajar de forma colaborativa, en equipo y obtener el mejor resultado posible de un proyecto, en SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al cliente del proyecto, (Villota, 2015).

SCRUM es un marco de trabajo iterativo e incremental para el desarrollo de proyectos y se estructura en ciclos de trabajo llamados Sprint. Éstos son iteraciones de 1 a 4 semanas, y se suceden una detrás de otra. Al comienzo de cada Sprint, el equipo multifuncional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. Al final del Sprint, el equipo lo revisa con los interesados en el proyecto, y les enseña lo que han construido, (Schwaber y Sutherland, 2017)

Las practicas empleadas por SCRUM para mantener un control ágil en el proyecto son:

* + - * Revisión de las iteraciones.
      * Desarrollo incremental.
      * Auto organización del equipo.
      * Colaboración.

A continuación, se describe las fases que conforman la metodología Scrum.

* + - * Planeación: En esta fase se establecen los objetivos generales del proyecto y el diseño de la arquitectura de software.
      * Valoración: Se asignan prioridades y riesgos. El cliente interviene estrechamente en este proceso y al comienzo de cada sprint puede introducir nuevos requerimientos o tareas.
      * Selección: El incluye a todo el equipo del proyecto que trabaja con el cliente, con la finalidad de seleccionar las características y la funcionalidad a desarrollar durante el sprint.
      * Desarrollo: El equipo se organiza para desarrollar el software con el objetivo de revisar el progreso y, si es necesario, volver a asignar prioridades al trabajo.
      * Cierre: Esta fase del proyecto es en donde se concluye el proyecto, completa la documentación requerida, como los marcos de ayuda del sistema y los manuales del usuario, y valora las lecciones aprendidas en el proyecto (Navarro, Fernández y Morales, 2013).

Scrum tiene cinco eventos esenciales para asegurar un desarrollo óptimo del producto. Cada evento ayudará al rendimiento y hará que las cosas sean más eficientes desde el principio hasta el final del proyecto. Los eventos empleados por Scrum son:

* + - * Sprint: El corazón de Scrum es el Sprint, es un bloque de tiempo time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado” utilizable y potencialmente desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.
      * Planificación de sprint: El trabajo a realizar durante el Sprint se planifica en la Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo. La planificación de Sprint tiene un límite de tiempo de hasta ocho horas para un Sprint de un mes.
      * Reunión diaria: El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para el Equipo de Desarrollo. En él, el Equipo de Desarrollo planea el trabajo para las siguientes 24 horas. Esto optimiza la colaboración y el desempeño del equipo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección del trabajo del Sprint a realizar a continuación.
      * Revisión de sprint: Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.
      * Retrospectiva de Sprint: La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Planificación de Sprint. Se trata de una reunión de una a tres horas para Sprint de un mes. Para Sprint más cortos el evento es usualmente más corto.

El propósito de la retrospectiva de Sprint es:

* + - * Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas
      * Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
      * Crear un plan para implementar las mejoras a la forma en la que el quipo Scrum desempeña su trabajo.

Scrum usa el término artefacto para referirse a las principales herramientas utilizadas dentro del marco. Los dos artefactos principales de Scrum son la lista de productos, y el incremento del producto, que operan con el mismo objetivo en mente, que es aumentar la transparencia y garantizar siempre una comprensión compartida del trabajo. A continuación, se describen los artefactos.

* + - * Lista de producto: La Lista de Producto es una lista ordenada de todo lo que se conoce que es necesario en el producto. Es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación. Una Lista de Producto nunca está completa.
      * Incremento: El Incremento es la suma de todos los elementos de la Pila del Producto completados durante un Sprint y el valor de los incrementos de todos los Sprint anteriores. Al final de un Sprint, el nuevo Incremento debe estar "Listo", lo que significa que debe estar en condiciones de uso y cumplir con la definición del Equipo Scrum de "Terminado", (Schwaber y Sutherland, 2017).

La metodología ágil Scrum implica un equipo autónomo y autogestionado, en donde hay varios roles que desempeñar en el equipo Scrum, cada uno con deberes y responsabilidades específicas:

* + - * El dueño del producto: El papel del dueño del producto está en el corazón del marco de Scrum. Es una sola persona y representa a los interesados, es el responsable de maximizar el valor del producto y ayuda a tomar decisiones precisas que dan vida al producto
      * Scrum Master: El deber principal de un Scrum Master es garantizar que todos los miembros del equipo se mantienen enfocados en sus tareas. El rol de un Scrum Master radica entre ser un líder de equipo y un rol de gestión, lo que hace definir sus rasgos o cualidades.
      * Equipo de desarrollo: El equipo de desarrollo, por su parte, tiene como responsabilidad convertir lo que el cliente quiere, en iteraciones funcionales del producto; el equipo de desarrollo no tiene jerarquías, todos sus miembros tienen el mismo nivel y cargo. Una vez que el equipo se compromete a entregar ciertos resultados, tiene la responsabilidad de cumplir su promesa, asegurándose de cumplir con resultados de calidad en el tiempo acordado.
      * Partes interesadas: Colaboran frecuentemente con el dueño del producto, Scrum Master, y el equipo de desarrollo. Su trabajo es proponer ideas y ayudar a iniciar la creación del servicio o producto del proyecto y proporcionar influencia durante todo el desarrollo del proyecto.

Por lo tanto, en Scrum, los roles de las partes interesadas incluyen:

* + - * Comunicar sus inquietudes y deseos al propietario del producto, particularmente en lo que respecta a la duración y calidad del proyecto.
      * Ayuda al propietario del producto a priorizar efectivamente algunas tareas para asegurarse de que el proyecto continúe sin problemas.
      * Continuamente tomar y proporcionar actualizaciones sobre cualquier necesidad de cambios a los planes, (Villota, 2015).

# Trabajos relacionados

En el siguiente apartado se presentan algunos artículos relacionados sobre el desarrollo de aplicaciones móviles en flutter.

* + 1. Information System for Providing Food Services Based on Mobile Application Using Flutter Framework

El siguiente artículo se basa en el desarrollo de una aplicación móvil en flutter la cual ayuda a la prestación de servicios de alimentos en la construcción, así mismo lleva un control en el proceso de transacción de las ventas de catering de alimentos en el AI Iman Boga.

Esta aplicación está construida con el método waterfall o también conocido de cascada, el cual consiste en varias etapas: análisis, diseño, codificación y pruebas. Los resultados de este estudio mostraron que la aplicación agiliza los procesos de transacción, así como de pedidos de alimentos.

* + 1. Desarrollo de aplicaciones móviles multiplataforma con Flutter

En la tesis “Desarrollo de aplicaciones móviles multiplataforma con Flutter”, menciona el funcionamiento de dicha aplicación, la cual está enfocada al tratamiento de pacientes de cirrosis hepática.

Esta aplicación permite llevar un mejor control de consultas, así como acortar los tiempos de estos tratamientos al realizar el seguimiento de la dosis y su aumento

a través de esta, sin necesidad de ir a consulta y siempre bajo el control del paciente.

Esta aplicación tiene un parecido con el proyecto que se piensa desarrollar, ya que utiliza las mismas tecnologías para su desarrollo, las cuales son Flutter, Dark, Android Studio entre otras.

* + 1. Conceptualización y prototipado de una aplicación móvil para padres de niños con síndrome de Down

En esta tesis se desarrolló una aplicación móvil la cual provea información de calidad al entorno cercano de personas con síndrome de Down, el enfoque de la aplicación estará en las personas y sus inquietudes sobre el cuidado de las personas con esta discapacidad.

El diseño de esta aplicación se desarrolló en Sketch la cual consta de los siguientes módulos: splash screen, login, registro, home, artículos, navigation drawer menu, ajustes, cuadro de búsqueda, mensajes, comunidad, 5 contactos y solicitudes de contacto. Por otra parte, esta aplicación es de tipo hibrida ya que hace uso de un framework el cual es NativeScript, así mismo implementa herramientas tecnológicas que se utilizaran a lo largo de este proyecto tales son: Material Desing, Android Studio con sus respectivos SDK, entre otros.

CAPITULO III. DESARROLLO

En este capítulo se hablará sobre las diferentes fases que se llevaron a cabo para la creación de la aplicación móvil desde el levantamiento de datos, hasta el desarrollo de ella.

# Fase de planificación

La primera fase del desarrollo de la aplicación móvil fue el análisis, en el cual se definieron todos los requerimientos solicitados por la empresa, para obtener una visión más específica de lo que se planeaba hacer.

* + 1. Formación del equipo

En la tabla 4 se muestra la estructura para el equipo de trabajo, con sus respectivos roles y actividades, todo esto enfocado a la metodología Scrum.

|  |  |  |
| --- | --- | --- |
| **Responsable** | **Área** | **Actividades** |
| Ing. Gustavo Arremilla Niño de Rivera | El dueño de Producto (Product Owner) | * Responsable de gestionar la lista del producto (Product Backlog). * Expresa claramente los elementos de la lista del producto. * Responsable de tomar las decisiones del proyecto y conoce el objetivo y visión del producto. * Se encarga de autorizar las ideas del producto, las ordena por prioridad y las coloca en el Product   Backlog. |
| Germain Agudo Juarez Neftalí López Antonio (Residentes Profesionales) | Equipo de Desarrollo (Development Team) | * Realizan el diseño y prototipos de la aplicación. * Realizan las pruebas necesarias para que la aplicación no tenga fallos. * Realizan la entrega del producto final en cada Spring. * Los miembros individuales tienen habilidades y tareas   especializadas en las |

|  |  |  |
| --- | --- | --- |
|  |  | que están as  enfocados. |
| Ing. Gustavo Arremilla Niño de Rivera  Ing. Jorge Ramírez García | Scrum Master | * Eliminar impedimentos para el progreso del Equipo de Desarrollo. * Se asegura que los objetivos, el alcance y el dominio del   proyecto sean entendidos por todos de la mejor manera.   * Ayudar a los empleados e   interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto.   * Ayudar al Equipo de Desarrollo a crear   productos de alto valor |
| L.I. Hiber Yair Ambrocio López | Grupo de interés (Stakeholders) | * Hace que la experiencia y el conocimiento esté disponible. * Ayudan a eliminar impedimentos que identifican el equipo de desarrollo. * Retroalimenta al   equipo de desarrollo. |

|  |  |  |
| --- | --- | --- |
|  |  | * Dan seguimiento al   proceso de cada iteración. |

Tabla 4 Formación del equipo SCRUM para el proyecto. Fuente: Elaboración propia

* + 1. Definición de los requerimientos funcionales

La aplicación estará compuesta de los siguientes requerimientos funcionales los cuales se detallan a continuación

* + - * El software debe correr en cualquier Smartphone con sistema operativo Android e iOS.
      * La aplicación contará con una pantalla de inicio de sesión la cual será amigable y accesible para los usuarios.
      * Desde la pantalla de inicio se contará con dos apartados para inicio de sesión, los cuales serán para usuarios y otras para empresas.
      * La aplicación móvil contara con diferentes apartados los cuales son:
        + Noticias: Se mostrará información relevante sobre el estado de Oaxaca, enfocado a lo cultura, tecnológico y sociales.
        + Talleres: En este apartado se mostrarán actividades en la cual los jóvenes aprenderán a desarrollar nuevas competencias para aplicarlas en la vida cotidiana.
        + Convocatorias: Se publicarán diferentes convocatorias emitidas por el gobierno federal como estatal, las cuales los jóvenes podrán registrarse y participar.
        + Webinar: Se trasmitirán temas enfocados sobre diferentes áreas de aprendizaje la cual nutra a los jóvenes en su desarrollo educativo, social y cultural.
        + Foros: Se discutirán diferentes temas de interés público, en el cual los jóvenes publicarán mensajes alrededor de un tema, creando de esta forma un hilo de conversación jerárquico.
        + Universidades: Se visualizará las diferentes universidades que existen en el Estado de Oaxaca.
        + Carreras: Mostrara las diferentes carreras que ofrecen las universidades en el Estado de Oaxaca.
        + Becas y apoyos: Apartado para apoyos monetarios para los jóvenes.
        + Bolsa de trabajo: Se mostrarán trabajos disponibles en el estado de Oaxaca para los jóvenes
    1. Definición de los requerimientos no funcionales
       - Rendimiento de la aplicación:
* La aplicación correrá en cualquier dispositivo Android e iOS siempre y cuando el dispositivo tenga espacio suficiente de almacenamiento.
* Proporcionará respuesta (datos) al usuario en tiempo real.
* El tiempo de respuesta promedio de la aplicación no debe superar los 15 segundos.
  + - * Seguridad
* Los datos de los usuarios siempre estarán cifrados para que ningún tercero pueda acceder a ellos.
* Al ingresar los usuarios sus credenciales la aplicación lo consultara directamente con la base de datos para que pueda acceder a la aplicación.
  + - * Disponibilidad
* La aplicación estará disponible en cualquier momento que el usuario lo requiera.
* La aplicación contara con conexión a internet o plan de datos, para el envío y recepción de datos. En caso de no tener conexión a internet la misma se guardará en la aplicación (localmente) para posteriormente reenviar o recibir los datos.
* La aplicación será cargada en las tiendas virtuales de iOS y Android para su respectiva descargar por los usuarios.
  + - * Mantenibilidad

o El sistema contara con su mantenimiento ya que se podría agregar nuevas funcionalidades o realizar modificaciones o correcciones.

* + - * Portabilidad

o La aplicación tendrá compatibilidad con versiones superiores o iguales a 5.0 de Android y 5.0 para iOS, ya sea en una Tablet o un smartphone.

* + - * Operabilidad

o La aplicación solo podrá ser operada por una persona con el rol de administrador.

* + 1. Restricciones

El dispositivo móvil en donde se ejecutará la aplicación para su correcto funcionamiento deberá tener los siguientes requisitos mínimos.

* + - * Procesador: 1.2 GHz.
      * Memoria RAM: 2 GB.
      * Almacenamiento: 40 MB disponibles.
      * Pantalla de 7” pulgadas en adelante.
      * Sistema operativo. Android 8 o superior y iOS 6.0

Para poder ingresar a la aplicación será necesario contar con una conexión a internet o un plan de datos, cabe recalcar que todas las restricciones descritas anteriormente son una sugerencia para el óptimo funcionamiento de la aplicación móvil, es decir que tenga un rendimiento óptimo.

* + 1. Definición de los objetos del producto (Product Backlog)

En la siguiente lista de producto se muestra los módulos que se desarrollaron a lo largo del proyecto, esta lista contiene un orden especifico, conforme a las historias de usuario que revelan el flujo de interacción del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Modulo** | **Nombre de tarea** | **Prioridad (1-3)** |
| 1 | 1 | ***Inicio de sesión*** | 3 |
| Seleccionar usuario: Joven o Empresa |
| Inicio de sesión. |
| Formulario para registros de usuarios  nuevos. |
| 2 | 2 | ***Perfil de usuario joven*** | 2 |
| Actualización de datos. |
| Cambiar foto de perfil. |
| Ayuda |
| Términos y condiciones |
| 3 | 3 | ***Perfil de usuario empresa*** | 2 |
| Actualización de datos. |
| Cambiar foto de perfil. |
| Ayuda. |
| Términos y condiciones. |
| 4 | 4 | ***Perfil de usuario administrador*** | 2 |
| Actualización de datos. |
| Cambiar foto de perfil. |
| Ayuda. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Términos y condiciones. |  |
| 5 | 5 | ***Pantalla principal*** | 3 |
| Noticias: Visualizar noticias sobre lo más  relevante en el estado de Oaxaca enfocado al marco cultural, social y educativo. |
| Talleres: Visualizar y aprender nuevas habilidades por medio de los talleres  impartidos por diferentes personalidades. |
| Convocatorias: Estar informado sobre las diferentes convocatorias que otorga el  gobierno federal como estatal. |
| Webinar: Visualizar transmisiones en vivo  sobre diferentes temas. |
| Foros: Visualizar transmisiones en vivo sobre diferentes temas en la cual podrán participar  los jóvenes y dar su punto de vista. |
| Escuelas: Visualizar todas las universidades  que se encuentran en el estado de Oaxaca. |
| Carreras: Visualizar todas las carreras ofertadas por las universidades del estado de  Oaxaca. |
| Becas y apoyos: Visualizar becas y apoyos  ofertadas por el gobierno estatal y federal. |
|  |  | Bolsa de trabajo: Visualizar ofertas de trabajo  ofertadas en el estado de Oaxaca. |  |

Tabla 5 Definición de los objetivos del producto (Product Backlog). Fuente: Elaboración propia

# Fase de diseño

* + 1. Prototipos

Con la finalidad de crear una aplicación móvil en términos de funcionalidad, se crearon los prototipos, para llevar un mejor control en su desarrollo, así mismo para que el cliente se diera una idea del producto final que obtendría.

* + - * Modulo 1: Página de inicio principal
        + El módulo 1 está conformado por la página de inicio principal (interfaz), la cual mostrará primeramente una ventana en donde se seleccionará el tipo de usuario el cual se divide en joven y empresa, posteriormente se iniciará sesión con sus credenciales de usuarios, en caso de no estar registrados se contará con un apartado para su respectivo registro.

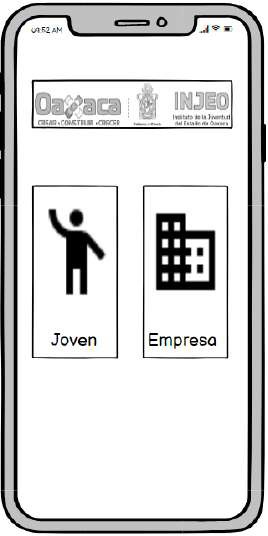
 

Figura 21Prototiopo de inicio de sesión seleccionar usuario

Figura 20 Prototipo de inicio de sesión con sus credenciales de usuario

* + - * Modulo 2: Perfil de usuario.
        + En este módulo el usuario podrá modificar su información personal, contará con un apartado de ayuda, así como los términos y condiciones de la aplicación.

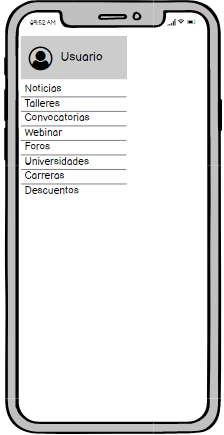


Figura 22 Prototipo de perfil de usuario

* + - * Modulo 3: Pantalla principal.
        + Este módulo estará conformado por la pantalla principal la cual nos mostrara diferentes menús de la aplicación los cuales son noticias, talleres, convocatorias, carreras, escuelas, becas, apoyos, bolsa de trabajo, foros y webinar. Posteriormente nos mandara a otras

ventanas diferentes con su respectiva información.



Figura 23 Prototipo de pantalla principal "menús"

El diseño que tienen los módulos de noticias, talleres, convocatorias, carreras, escuelas, becas, apoyos, bolsa de trabajo, foros y webinar, se muestra a continuación:



Figura 24 Prototipo de los diferentes módulos de la aplicación 1

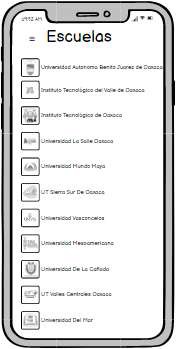
 



Figura 25 Prototipos de los diferentes módulos de la aplicación 2

* + 1. Diagrama entidad-relación

La estructura general de la base de datos, modelo entidad relación quedo de la siguiente manera:

Usuario\_Foro



CP id\_UsuarioForo comentarioUsuarioF estadoUsuarioF

fecha\_registroUsuarioF

fecha\_eliminacionUsuarioF

id\_Foro (PK) id\_Usuario (PK)

Foro

CP id\_Foro

tituloForo descripcionForo estadoForo

Convocatoria

CP id\_Convocatoria

tituloConvocatoria subtituloConvocatoria imgConvocatoria

descripcionConvocatoria enlaceConvocatoria

estadoConvocatoria

fecha\_registroConvocatoria

id\_Usuario (PK)

Convocatoria\_Image

CP id\_Convocatoria\_Img id\_Img (PK)

id\_Covocatoria (PK) estadoConvImg

fecha\_registroConv Img

fecha\_eliminacionConvImg

id\_Usuario (PK)

Taller

CP id\_Taller

tituloTaller subitituloTaller imgTaller

descripcionTaller enlaceTaller

estadoTaller

fecha\_registroTaller fecha\_eliminacionTaller

id\_Usuario (PK)

Usuario

Taller\_Img

CP id\_TallerImg estadoTaImg

fecha\_registroTaImg fecha\_eliminacionTaImg

id\_Img (PK) id\_Taller (PK) id\_Usuario (PK)

Imagen

CP id\_Img img

estadoImg

fecha\_registroImg fecha\_eliminacionImg

id\_Usuario (PK)

Noticia

CP id\_Noticia

tituloNoticia subtituloNoticia imgNoticia

descripcionNoticia estadoNoticia

fecha\_registroNoticia id\_Usuario (PK)

Noticia\_Img

CP id\_Noticia\_Img estadoNotImg

fecha\_registroNotImg fecha\_eliminacionNotImg

id\_Img (PK) id\_Noticia (PK) id\_Usuario (PK)

Participante

Participante\_Noticia

CP id\_Participante\_Noticia estadoPartiN

fecha\_registroPartiN fecha\_eliminacionPartiN

id\_Noticia (PK) id\_Participante (PK) id\_Usuario (PK)

Red\_Participante

Instructor\_Taller

CP id\_Instructor\_Taller estadoInstructorT fecha\_registroInstruT

fecha\_eliminacionInstruT id\_Taller (PK)

id\_Instructor (PK) id\_Usuario (PK)

fecha\_registroForo fecha\_eliminacionForo

id\_Usuario (PK)

Apoyo

CP id\_Apoyo id\_Usuario (PK)

tituloApoyo descripcionApoyo requsitosApoyo enlaceApoyo

imgApoyo

BolsaTrabajo

CP id\_BolsaT id\_Usuario (PK)

tituloBolsaT descripcionBolsaT requisitosBolsaT imgBolsaT enlaceBolsaT

estadoBolsaT

Externo

CP id\_Externo id\_Usuario (PK)

nombreExterno rfcExterno direccionExterno estadoExterno

fecha\_registroExterno

CP idUsuario

user\_nameUsuario

numero\_telefo nicoUsuario correoUsuario

passwordUsuario imgUsuario

rolUsuario

Oferta

CP id\_Oferta id\_Usuario (PK)

descripcionOferta estadoOferta

fecha\_registroOferta

fecha\_eliminacionOferta

Red\_Social

CP id\_Participante

nombreParticipante imgParticipante estadoParticipante

fecha\_registroParticipante

fecha\_eliminacionParticipan te

id\_Usuario (PK)

CP id\_Red\_Participante

id\_Participante (PK) urlRedSocial estadoRedSocial

fecha\_registroRedSocial

fecha\_eliminacionRedSocial

id\_RedSocial (PK) id\_Usuario (PK)

Instructor

CP id\_Instructor nombreInstructor

descripcionIntructor imgIntructor

estadoInstructor

fecha\_registroInstructor

fecha\_eliminacionInstructor

id\_Usuario (PK)

Beca

estadoApoyo

fecha\_registroApoyo fecha\_eliminacionApoyo

fecha\_registroBolsaT fecha\_eliminacionBolsaT

fecha\_eliminacionExterno

Inscripciones

CP id\_Inscripciones id\_Usuario (PK)

descripcionInscripciones

estadoUsuario

fecha\_registroUsuario fecha\_eliminacionUsuario datos\_completosUsuario

id\_Escuela (PK) id\_Carrera (PK)

Carrera

CP id\_Carrera

CP id\_Redsocial redRedSocial

id\_Usuario (PK)

imgRedSocial estadoRedSocial

Red\_lnstructor

CP id\_RedSocialInstructor id\_Instructor (PK)

urlRedInstruc

CP id\_Beca id\_Usuario (PK)

tituloBeca descripcionBeca requisitosBeca enlaceBeca

imgBeca estadoBeca

fecha\_registroBeca fecha\_eliminacionBeca

estadoInscripciones

fecha\_registroInscripciones responsable\_Registro (FK)

id\_Taller (PK)

Persona

CP id\_Persona id\_Usuario (PK)

nombrePersona apellido\_paternoPersona apellido\_maternoPersona edadPersona sexoPersona

curpPersona

fecha\_nacimientoPersona municipio Persona

Roles

CP id\_Rol rol

id\_Usuario (PK)

nombreCarrera imgCarrera estadoCarrera

fecha\_registroCarrera

Escuela

CP id\_Escuela id\_Usuario (PK)

nombreEscuela imgEscuela estadoEscuela

fecha\_registroEscuela

fecha\_registroRedSocial

fecha\_eliminacionRedSocial

estadoRedInstruc

fecha\_registroRedInstruc

fecha\_eliminacionRedInstru

id\_RedSocial (PK) id\_Usuario (PK)

regionPersona

estadoPersona

fecha\_registroPersona fecha\_eliminacionPersona

Figura 26 Diagrama general de la base de datos de la aplicación

móvil.

En la figura 26 se muestra la estructura completa del modelo entidad-relación a continuación se desglosará cada una de las tablas para tener una mejor apreciación:

Tabla Usuario: Está conformada por idUsuario, user\_nameUsuario, numero\_telefonicoUsuario, correoUsuario, passwordUsuario, imgUsuario, rolUsuario, estado Usuario, fecha\_registroUsuario, fecha\_eliminacionUsuario y datos\_completosUsuario

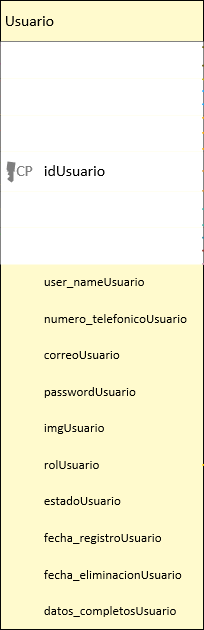


Figura 27 Tabla usuario

Tabla Persona: Está conformada por id\_Persona, por primary key id\_Usuario, nombrePersona, apellido\_paternoPersona, apellido\_maternoPersona, edadPersona, sexoPersona, curpPersona, fecha\_nacimientoPersona, municipioPersona, regionPersona, estadoPersona, fecha\_registroPersona y fecha\_eliminaciónPersona.

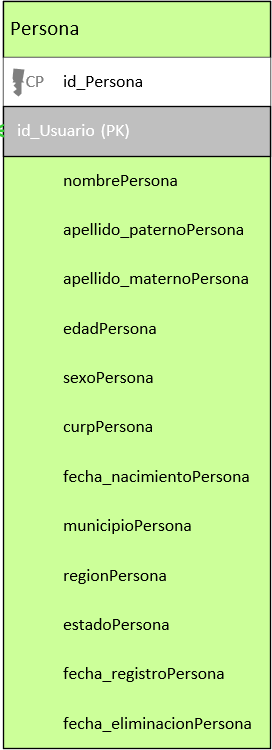


Figura 28 Tabla persona

Tabla roles: Está conformada por id\_Rol y rolUsuario.

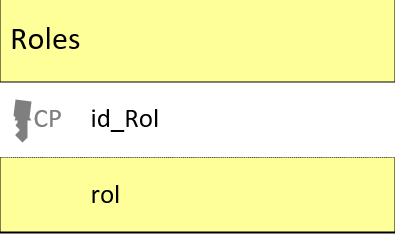
Figura 29 Tabla roles



Figura 30 Tabla externo

Tabla Externo: Está conformada por id\_Externo, una primary key de id\_Usuario, nombreExterno, rfcExterno, direccionExterno, estadoExterno, fecha\_registroExterno y fecha\_eliminaciónExterno.

Tabla Inscripciones: Está conformada por id\_Inscripciones, descripcionInscripciones, estadoInscripciones, fecha\_registroInscripciones, y dos primary key: id\_Usuario e id\_Taller, así mismo una foreign key: responsable\_Registro.

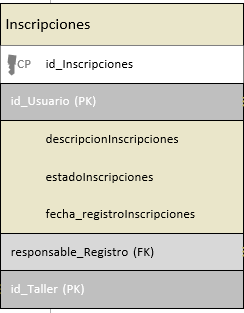


Figura 31 Tabla inscripciones



Figura 32 Tabla escuela

Tabla Escuela: Está conformada por id\_Escuela, nombreEscuela, imgEscuela, estadoEscuela fecharegistroEscuela y una primary key: id\_Usuario.

Tabla Carrera: Está conformada por id\_Carrera, nombreCarrera, imgCarrera, estadoCarrera, fecha\_registroCarrera y una primary key: id\_Usuario.

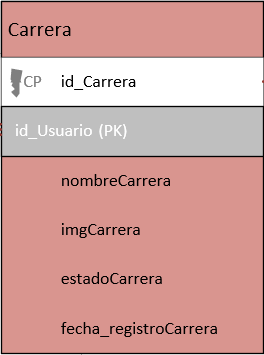


Figura 33 Tabla carrera

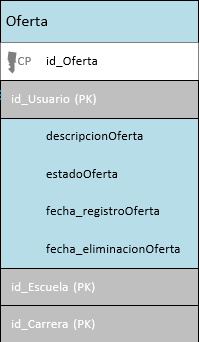


Figura 34 Tabla oferta

Tabla Oferta: Está conformada por id\_Oferta, descripcionOferta, estadoOferta, fecha\_registroOferta, fecha\_eliminaciónOferta y de tres primary key las cuales son: id\_Usuario, id\_Escuela e id\_Carrera.

Tabla Bolsa de trabajo: Está conformada por id\_BolsaT, tituloBolsaT, descripcionBolsaT, requisitosBolsaT, imgBolsaT, enlaceBolsaT, estado BolsaT, fecha\_registroBolsaT, fecha\_eliminaciónBolsaT y una primary key: id\_Usuario.



Figura 35 Tabla bolsa de trabajo

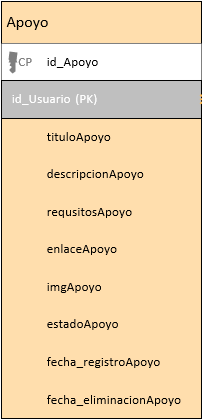


Figura 36 Tabla apoyo

Tabla Apoyo: Está conformada por id\_Apoyo, tituloApoyo, descripciónApoyo, requisitosApoyo, enlaceApoyo, fecha\_registroApoyo, fecha\_eliminaciónApoyo y una primary key: id\_Usuario.

Tabla Beca: Está conformada por id\_Beca, tituloBeca, descripciónBeca, requsitosBeca, enlaceBeca, imgBeca, estadoBeca, fecha\_registroBeca, fecha\_eliminaciónBeca y una primary key: id\_Usuario.

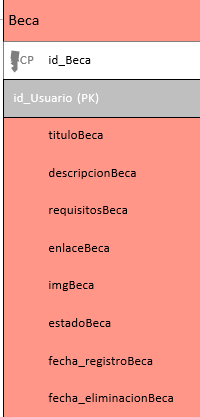


Figura 37 Tabla beca

Tabla Foro: Está compuesta por id\_Foro, tituloForo, descripcionForo, estadoForo, fecha\_registroForo, fecha\_eliminaciónForo y una primary key: id\_Usuario.



Figura 38 Tabla foro



Figura 39 Tabla usuario-foro

Tabla Usuario\_Foro: Está conformada por id\_UsuarioForo, comentarioUsuarioF,

estadoUsuarioF, fecha\_registroUsuarioF, fecha\_eliminaciónnUsuarioF y dos primary key: id\_Foro, id\_Usuario.

Tabla Convocatoria: Está conformada por id\_Convocatoria, tituloConvocatoria,

subtituloConvocatoria, imgConvocatoria, descripcionConvocatoria, enlaceConvocatoria, estadoConvocatoria, fecha\_registroConvocatoria y una primary key: id\_Usuario.

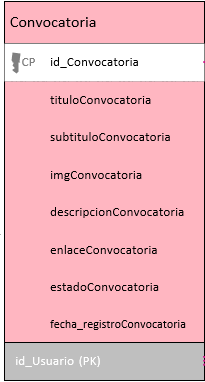


Figura 40 Tabla convocatoria

Figura 41 Tabla convocatoria-imagen

Tabla Convocatoria\_Image: Está conformada por id\_Convocatoria\_img, estadoConvImg, fecha\_registroConvImg, fecha\_eliminaciónConvImg y tres primary key: id\_Img, id\_Convocatoria e id\_Usuario.

Tabla Taller: Está conformada por id\_Taller, tituloTaller, subtituloTaller, imgTaller, descripcionTaller, enlaceTaller, estadoTaller, fecha\_registroTaller, fecha\_eliminaciónTaller y una primary key: id\_Usuario.

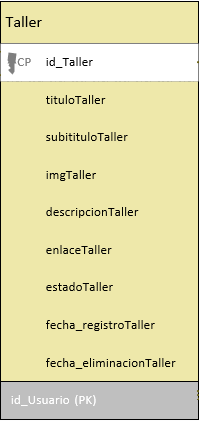


Figura 42 Tabla taller



Figura 43 Tabla taller-img

Tabla Taller\_Img: Está conformada por id\_TallerImg, estadoTaImg, fecha\_registroTaImg, fecha\_eliminaciónTaImg y tres primary key: id\_Img, id\_Taller e id\_Usuario.

Tabla Imagen: Está conformada por id\_Img, img, estadoImg, fecha\_registroImg, fecha\_eliminacionImg y una primary key: id\_Usuario.

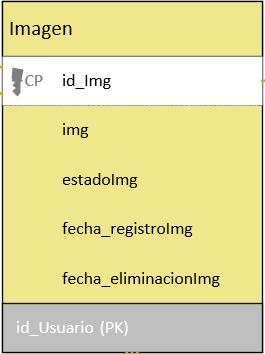


Figura 44 Tabla imagen

Tabla Noticia: Está conformada por id\_Noticia, tituloNoticia, subtituloNoticia, imgNoticia, descripcionNoticia, estadoNoticia, fecha\_registroNoticia y una primary key id\_Usuario.

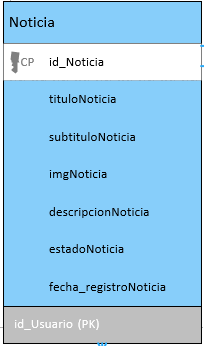


Figura 45 Tabla noticia

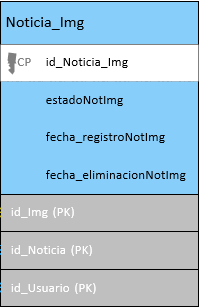


Figura 46 Tabla noticia- img

Tabla Noticia\_Img: Está conformada por id\_Noticia\_Img, estadoNotImg, fecha\_registroNoImg, fecha\_eliminaciónNotImg y por tres primary key: id\_Img, id\_Noticia, id\_Usuario.

Tabla Participante: Está conformada por id\_Participante, nombreParticipante, imgParticipante, estadoParticipante, fecha\_registroParticipante, fecha\_eliminaciónParticipante y una primary key: id\_Usuario

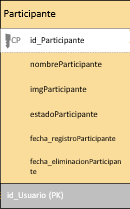


Figura 47 Tabla participante

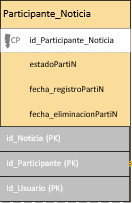


Figura 48 Tabla participante- noticia

Tabla Participante\_Noticia: Está conformada por id\_Participante\_Noticia, estadoPartiN, fecha\_registrosPartiN, fecha\_eliminaciónPartiN y tres primary key id\_Noticia, id\_Participante, id\_Usuario.

Tabla Instructor: Está compuesta por id\_Instructor, nombreInstrutor, cargoInstructor, descripcionInstructor, imgInstructor, estadoInstructor, fecha\_registroInstructor, fecha\_eliminaciónInstructor y una primary key id\_Usuario.

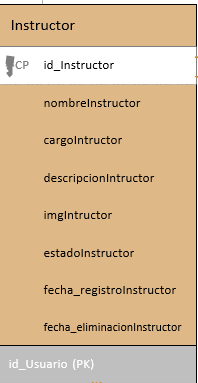


Figura 49 Tabla instructor



Figura 50 Tabla instructor- taller

Tabla Instructor\_Taller: Está conformado por id\_Instructor\_Taller, estadoInstrutorT, fecha\_registroInstructorT, fecha\_eliminación\_InstructorT y tres primary key:

id\_Taller, id\_Instructor, id\_Usuario.

Tabla Red Participante: Está conformada por id\_RedSocial, redRedSocial, imgRedSocial, estadoRedSocial, fecha\_registroRedSocial, fecha\_eliminaciónRedSocial y una primary key id\_Usuario.

Tabla Red\_Instructor: Está conformada por id\_RedSocialInstructor, urlRedIntruc, estadoRedInstruc, fecha\_registroRedInstruc, fechaeliminaciónRedInstruc y tres primary key id\_Instructor, id\_RedSocil, id\_Usuario.

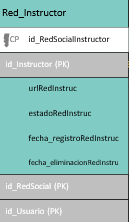


Figura 51 Tabla red-

social

Figura 52 Tabla res- instructor

Tabla Red\_Participante: Está conformada por id\_Red\_Participante, urlRedSocial, estadoRedSocial, fecha\_registroRedSocial, fecha\_eliminaciónRedSocial y tres primary key: id\_Usuario, id\_ReSocial, id\_Usuario.

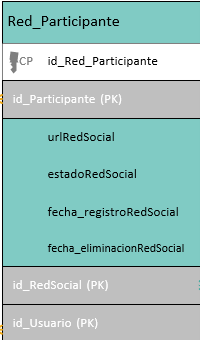


Figura 53 Tabla red-

participante

* + 1. Planificación y estimación

En este apartado se desglosarán las historias de usuarios, se diseñaron diagramas de casos de uso para identificar los distintitos escenarios por consiguiente se desarrollaron diagramas UML para la creación de un modelo visual del software.

En las historias de usuario se define:

* + - * Numero de historia: Numero identificador.
      * Nombre: Titulo de la historia.
      * Prioridad: Importancia en el desarrollo. Los valores son: alta (100), media

(50) y baja (20).

* + - * Riesgo de desarrollo: Dificultad estimada que se pueda tener con el desarrollo de la historia, los valores son: alto, medio y bajo.
      * Puntos estimados: Estimación de tiempo para alcanzar la finalización de una historia de usuario los valores asignados para estos puntos son los siguientes
        + 1 = 1 Semana
        + 2 = 2 Semanas
        + 3 = 3 Semanas
      * Iteración asignada: Es el sprint en el cual se decide realizar la historia.
      * Descripción: Se relata, desde el punto de vista del interesado, la función que se pide en la historia.
      * Validación: Criterios de aceptación de la historia de usuarios, definidos por parte del interesado en su realización.
      1. Historias de usuarios

Como se mencionó anteriormente en este apartado se desglosarán las historias de usuarios las cuales corresponde a un Sprint definido anteriormente en el punto

3.1.6 del producto Backlog.

* + - * 1. Historias de usuarios inicio de sesión

Historia de usuario 1: Esta historia de usuario corresponde al Sprint 1 el cual comprende el acceso a la aplicación móvil para su posterior uso, esta historia de usuario se enfoca en el inicio de sesión para el usuario joven de la aplicación móvil.

|  |  |
| --- | --- |
| Historia de usuario N° 1 | |
| Nombre de historia: Inicio de sesión para usuario joven y empresa | |
| Prioridad: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: La aplicación debe tener dos apartados en los cuales se podrá  ingresar por dos tipos de usuarios: joven y empresa. | |
| Validación:   * Ingresar correctamente sus datos. * Validar el correo electrónico en la base de datos mediante el restfull   server y si este existe mandara una clave codificada para dar acceso al usuario (token). | |

Tabla 6 Historia de usuario "Inicio de sesión para usuario joven y empresa"

Historia de usuario 2: Esta historia de usuario corresponde al Sprint 1 el cual comprende el acceso a la aplicación móvil para su posterior uso, esta historia de usuario se enfoca en el inicio de sesión para el usuario administrador de la aplicación móvil.

|  |  |
| --- | --- |
| Historia de usuario N° 2 | |
| Nombre de historia: Inicio de sesión para usuario administrador | |
| Prioridad: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: La aplicación contara con un usuario administrador el cual no  necesita rellenar ningún registro para su ingreso. | |
| Validación:   * Ingresar correctamente sus datos. * Datos del administrador predefinido por los desarrolladores de la aplicación móvil. * Validar el correo electrónico en la base de datos mediante el restfull server y si existe mandara una clave codificada para dar acceso al usuario (token). | |

Tabla 7 Historia de usuario "Inicio de sesión para administrador"

* + - * 1. Historias de usuarios perfil de usuario

Historia de usuario 3: Esta historia de usuario corresponde al Sprint 2, 3 y 4 el cual comprende al perfil de usuario de la aplicación móvil, esta historia de usuario se enfocado para los tres diferentes tipos de usuarios (Joven, Empresa y Administrador), cabe recalcar que el perfil de usuario realizara las mismas acciones para los tres tipos de usuarios.

|  |  |
| --- | --- |
| Historia de usuario N° 3 | |
| Nombre de historia: Perfil de usuario | |
| Prioridad: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 2 | Iteración asignada: 2, 3, 4. |
| Descripción: La aplicación debe contar con un menú lateral el cual contendrá el perfil de usuario, ayuda, términos y condiciones, así como todos los módulos de la aplicación los cuales son: Noticias, Talleres, Convocatorias, Webinar,  Foros, Escuelas, Carreras, Becas y Apoyos. | |
| Validación:  No aplica | |

Tabla 8 Historia de usuario para” Perfil de usuario”

* + - * 1. Historias de usuarios pantalla principal

Historia de usuario 4: Esta historia de usuario corresponde al Sprint 5 el cual comprende a los diferentes módulos que tiene la aplicación móvil, cabe recalcar que esta historia de usuario está enfocada al usuario Joven.

|  |  |
| --- | --- |
| Historia de usuario N° 4 | |
| Nombre de historia: Módulos de la pantalla principal (menús). | |
| Prioridad: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 3 | Iteración asignada: 5 |
| Descripción: La aplicación debe contar con diferentes módulos los cuales son:  Noticias, Talleres, Convocatorias, Webinar, Foros, Escuelas, Carreras, Becas y Apoyos, en el cual el usuario joven podrá visualizar toda la información. | |
| Validación:  - Registro en talleres | |

Tabla 9 Historia de usuario "Módulos de la pantalla principal Joven"

Historia de usuario 5: Esta historia de usuario corresponde al Sprint 3 el cual comprende a los diferentes módulos que tiene la aplicación móvil, cabe recalcar que esta historia de usuario está enfocada al usuario empresa.

|  |  |
| --- | --- |
| Historia de usuario N° 5 | |
| Nombre de historia: Módulos de la pantalla principal (menús). | |
| Prioridad: Media | Riesgo en desarrollo: Alto |
| Puntos estimados: 3 | Iteración asignada: 5 |
| Descripción: Al igual que el usuario “Joven” el usuario “Empresa” contara con los mismos módulos los cuales son: Noticias, Talleres, Convocatorias, Webinar, Foros, Escuelas, Carreras, Becas y Apoyos. El usuario “Empresa” aparte de poder visualizar la información de todos los módulos de la app tendrá la opción de poder registrar Talleres, Convocatorias, Becas y Apoyos, para que  los jóvenes puedan visualizarlas. | |
| Validación:   * Visualizar información. * Subir convocatorias, talleres, becas y apoyos. | |

Tabla 10 Historia de usuario "Módulos de la pantalla principal Empresa"

Historia de usuario 6: Esta historia de usuario corresponde al Sprint 3 el cual comprende a los diferentes módulos que tiene la aplicación móvil, cabe recalcar que esta historia de usuario está enfocada al usuario administrador.

|  |  |
| --- | --- |
| Historia de usuario N° 6 | |
| Nombre de historia: Módulos de la pantalla principal | |
| Prioridad: Media | Riesgo en desarrollo: Alto |
| Puntos estimados: 3 | Iteración asignada: 5 |
| Descripción: El usuario administrador se encargará de agregar, modificar y eliminar información de todos los módulos de la aplicación móvil los cuales son: Noticias, Talleres, Convocatorias, Webinar, Foros, Escuelas, Carreras, Becas  y Apoyos. | |
| Validación:   * Agregar información a los diferentes módulos (texto, imágenes). * Visualizar quien agrega, modifica y elimina la información. | |

Tabla 11 Historia de usuario "Módulos de la pantalla principal Administrador"

* + - 1. Desarrollo de casos de usos

Identificación de los actores en la aplicación móvil.

|  |  |
| --- | --- |
| Actores de Aplicación Joven | |
| Joven | Ingresar a la aplicación, ingresar a todos los menús de la  aplicación, así como modificar su información personal. |
| Administrador | Ingresar a la aplicación, navegar entre los diferentes menús d  la aplicación, podrá modificar, eliminar y agregar datos. |
| Empresa | Ingresar a la aplicación, navegar entre los diferentes menús de la aplicación, podrá agregar datos en algunos módulos  específicos. |

Tabla 12 Actores de los casos de usos

* + - * 1. Casos de usos inicio de sesión

Caso de uso 1: En el siguiente caso de uso se muestra la interacción del Usuario

“Joven” en el escenario de inicio de sesión de la aplicación móvil.



Ingresar a la aplicación

Busca datos del

usuario

<<incluir>>

<<incluir>>

Inicio de sesión

<<extender>> Mostar información

Joven

Registro de usuario nuevo

<<incluir>>

Registrar en base de

Base de datos

datos

Figura 54 Caso de uso "Inicio de sesión joven"

Especificaciones de Caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Inicio de sesión joven”** | | |
| Nombre | Inicio de sesión para el usuario joven | |
| Descripción | Identificar al usuario para el uso de la aplicación | |
| Actores | Joven | |
| Precondición | Que la aplicación se haya ejecutado. Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Tener al usuario registrado. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | La aplicación despliega la interfaz de  inicio de sesión. |
|  | 2 | El usuario “joven” ingresa su correo y contraseña, en caso de no estar registrado existe un apartado para su registro, en donde se da la opción de elegir el usuario a registrar ya sea joven  o empresa. |
|  | 3 | La aplicación valida los datos en la base  de datos. |
|  | 4 | Da acceso a la aplicación. |
| Post condición | Si el usuario no cuenta con una conexión a internet  o un plan de datos no podrá acceder a la aplicación. | |

Tabla 13 Especificación de caso de uso inicio de sesión joven

Caso de uso 2: En el siguiente caso de uso se muestra la interacción del Usuario

“Empresa” en el escenario de inicio de sesión de la aplicación móvil.



Ingresar a la aplicación

Busca datos de

empresa

<<incluir>>

<<incluir>>

Inicio de sesión

<<extender>>

Mostar información

Base de datos

Empresa

Registro de usuario nuevo

<<incluir>>

Registrar en base de

datos

Figura 55 Caso de uso "Inicio de sesión empresa"

Especificaciones de Caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Inicio de sesión empresa”** | | |
| Nombre | Inicio de sesión para el usuario empresa | |
| Descripción | Identificar al usuario para el uso de la aplicación | |
| Actores | Empresa | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Tener al usuario registrado. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | La aplicación despliega la interfaz de  inicio de sesión. |
|  | 2 | El usuario “Empresa” ingresa su correo y contraseña, en caso de no estar registrado existe un apartado para su  registro. |
|  | 3 | La aplicación valida los datos en la  base de datos. |
|  | 4 | Da acceso a la aplicación. |
| Post condición | Si el usuario no cuenta con una conexión a internet o un plan de datos no podrá acceder a la  aplicación. | |

Tabla 14 Especificación de caso de uso inicio de sesión empresa

Caso de uso 3: En el siguiente caso de uso se muestra la interacción del Usuario

“Administrador” en el escenario de inicio de sesión de la aplicación móvil.



Ingresar a la aplicación

Busca datos del

usuario

<<incluir>>

<<incluir>>

Mostar información

<<extender>>

Base de datos

Administrador

Inicio de sesión

Figura 56 Caso de uso "Inicio de sesión administrador"

Especificaciones de Caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Inicio de sesión administrador”** | | |
| Nombre | Inicio de sesión para el usuario administrador | |
| Descripción | Identificar al usuario para el uso de la aplicación | |
| Actores | Administrador | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Tener al usuario registrado. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | La aplicación despliega la interfaz de  inicio de sesión. |
|  | 2 | El usuario “Administrador” ingresa su correo y contraseña, el cual fue proporcionado por los desarrolladores  de la aplicación móvil. |
|  | 3 | La aplicación valida los datos en la  base de datos. |
|  | 4 | Da acceso a la aplicación. |
| Post condición | Si el usuario no cuenta con una conexión a internet o un plan de datos no podrá acceder a la  aplicación. | |

Tabla 15 Especificación de caso de uso inicio de sesión administrador

* + - * 1. Casos de usos perfil de usuario

Caso de uso 4: En el siguiente caso de uso se muestra la interacción del usuario

“Joven” en el escenario de perfil de usuario de la aplicación móvil.



Actualización de datos <<incluir>> Modificar información

<<extender>>

Ayuda

Visualizar información

Joven

Base de datos

Menús de la aplicación

movil

<<incluir>>

Visualizar información

Terminos y

condiciones

Figura 57 Caso de uso “Perfil de usuario joven”

Especificaciones de Caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Perfil de usuario joven”** | | |
| Nombre | Perfil de usuario joven | |
| Descripción | Navegación en el menú lateral de la aplicación  móvil. | |
| Actores | Joven | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Haber ingresado a la aplicación móvil. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | La aplicación despliega un menú lateral con todos los módulos de la aplicación móvil (Noticias, Talleres, Convocatorias, Webinar, Foros, Escuelas, Carreras, Becas y Apoyos) también se encuentra las opciones de perfil de usuario, ayuda y términos y  condiciones. |
|  | 2 | El usuario “Joven” accederá a todas las  opciones de los menús. |
| Post condición | Si el usuario no cuenta con una conexión a internet  o un plan de datos no podrá acceder a la aplicación. | |

Tabla 16 Especificaciones de caso de uso perfil de usuario joven

Caso de uso 5: En el siguiente caso de uso se muestra la interacción del usuario

“Empresa” en el escenario de perfil de usuario de la aplicación móvil.



Actualización de datos <<incluir>> Modificar información

<<extender>>

Ayuda

Visualizar información

Joven

Base de datos

Menús de la aplicación

movil

<<incluir>>

Visualizar información

Terminos y

condiciones

Figura 58 Caso de uso "Perfil de usuario empresa"

Especificaciones de Caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Perfil de usuario empresa”** | | |
| Nombre | Perfil de usuario empresa | |
| Descripción | Navegación en el menú lateral de la aplicación  móvil. | |
| Actores | Empresa | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Haber ingresado a la aplicación móvil. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | La aplicación despliega un menú lateral con todos los módulos de la aplicación móvil (Noticias, Talleres, Convocatorias, Webinar, Foros, Escuelas, Carreras, Becas y Apoyos) también se encuentra las opciones de perfil de usuario, ayuda y términos y  condiciones. |
|  | 2 | El usuario “Empresa” accederá a todas  las opciones de los menús. |
| Post condición | Si el usuario no cuenta con una conexión a internet o un plan de datos no podrá acceder a la  aplicación. | |

Tabla 17 Especificaciones de caso de usuario perfil de usuario empresa

Caso de uso 6: En el siguiente caso de uso se muestra la interacción del usuario

“Administrador” en el escenario de perfil de usuario de la aplicación móvil.



Actualización de datos <<incluir>> Modificar información

<<extender>>

Ayuda

Visualizar información

Joven

Base de datos

Menús de la aplicación

movil

<<incluir>>

Visualizar información

Terminos y

condiciones

Figura 59 Caso de uso "Perfil de usuario administrador”

Especificaciones de caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Perfil de usuario administrador”** | | |
| Nombre | Perfil de usuario administrador | |
| Descripción | Navegación en el menú lateral de la aplicación móvil. | |
| Actores | Administrador | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente. Haber ingresado a la aplicación móvil. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | La aplicación despliega un menú lateral con todos los módulos de la aplicación móvil (Noticias, Talleres, Convocatorias, Webinar, Foros, Escuelas, Carreras, Becas y Apoyos) también se encuentra las opciones de  perfil de usuario, ayuda y términos y condiciones. |
|  | 2 | El usuario “Administrador” accederá a  todas las opciones de los menús. |
| Post condición | Si el usuario no cuenta con una conexión a internet o un plan de datos no podrá acceder a la aplicación. | |

Tabla 18 Especificación de caso de uso perfil de usuario administrador

* + - * 1. Casos de usos pantalla principal

Caso de uso 7: En el siguiente caso de uso se muestra la interacción del Usuario

“Joven” en el escenario de pantalla principal de la aplicación móvil.



Joven

Base

de datos



Noticias

<<incluir>>

Visualizar información

Talleres

<<incluir>>

Registrarse en talleres

<<incluir>>

Visualizar información

Convocatorias

<<incluir>>

Visualizar información

Webinar

<<incluir>>

Visualizar información

Foros

<<incluir>>

Visualizar información

Universidades

<<incluir>>

Visualizar información

Carreras

<<incluir>>

Visualizar información

Becas

<<incluir>>

Visualizar información

Apoyos

<<incluir>>

Visualizar información

Bolsa de trabajo

<<incluir>>

Visualizar información

Figura 60 Caso de uso "Pantalla principal joven"

Especificaciones de caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Pantalla principal joven”** | | |
| Nombre | Pantalla principal usuario joven | |
| Descripción | Navegación en los menús de la pantalla principal. | |
| Actores | Joven | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Haber ingresado a la aplicación móvil. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | Se mostrarán los módulos de la aplicación móvil. (Noticias, Talleres, Convocatorias, Webinar, Foros,  Escuelas, Carreras, Becas y Apoyos). |
|  | 2 | El usuario al seleccionar algún módulo podrá acceder, para visualizar la información la cual se encuentra en la  base de datos. |
|  | 3 | El usuario joven en el módulo de talleres contara con un apartado en el cual podrá registrarse a los diferentes  talleres que se oferten. |
| Post condición | Si el usuario no cuenta con una conexión a internet  o un plan de datos no podrá acceder a la aplicación. | |

Tabla 19 Caso de uso pantalla principal joven

Caso de uso 8: En el siguiente caso de uso se muestra la interacción del usuario

“Empresa” en el escenario de pantalla principal de la aplicación móvil.

Noticias

<<incluir>>

Visualizar información

Visualizar información

<<incluir>>

Talleres Agregar

<<incluir>>

Modificar

<<incluir>>

Convocatorias <<incluir>>

Eliminar

<<incluir>>

Visualizar información

Webinar <<incluir>> Visualizar información

Foros



<<incluir>>

Visualizar información



Empresa



Universidades

<<incluir>>

Visualizar información

Base de datos

Carreras

<<incluir>>

Visualizar información

Becas <<incluir>> Visualizar información

Apoyos <<incluir>> Visualizar información

Bolsa de trabajo

<<incluir>>

Visualizar información

Figura 61 Caso de uso "Pantalla principal empresa"

Especificaciones de caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Pantalla principal empresa”** | | |
| Nombre | Pantalla principal usuario empresa | |
| Descripción | Navegación en los menús de la pantalla principal. | |
| Actores | Empresa | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Haber ingresado a la aplicación móvil. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | Se mostrarán los módulos de la aplicación móvil. (Noticias, Talleres, Convocatorias, Webinar, Foros,  Escuelas, Carreras, Becas y Apoyos). |
|  | 2 | El usuario al seleccionar algún módulo podrá acceder, para visualizar la información la cual se encuentra en la  base de datos. |
|  | 3 | En algunos módulos se contará con tres botones los cuales servirán para agregar, modificar y eliminar  información. |
| Post condición | Si el usuario no cuenta con una conexión a internet  o un plan de datos no podrá acceder a la aplicación. | |

Tabla 20 Especificación de caso de uso pantalla principal empresa

Caso de uso 9: En el siguiente caso de uso se muestra la interacción del usuario

“Administrador” en el escenario de pantalla principal de la aplicación móvil.



Agregar

<<incluir>>

Noticias

<<incluir>>

<<incluir>>

Modificar

Eliminar Agregar

<<incluir>>

Talleres <<incluir>> Modificar

<<incluir>>

Eliminar

Agregar

<<incluir>>

Convocatorias

<<incluir>>

<<incluir>>

Modificar

Eliminar

Agregar

<<incluir>>

Webinar

<<incluir>>

<<incluir>>

Modificar

Eliminar

Foros

<<incluir>>

<<incluir>>

Agregar

Eliminar

Agregar

<<incluir>>

Administrador

Universidades <<incluir>> Modificar

<<incluir>>

Base

de datos

Eliminar

Agregar

<<incluir>>

Carreras <<incluir>> Modificar

<<incluir>>

Eliminar

Agregar

<<incluir>>

Becas

<<incluir>>

<<incluir>>

Modificar

Eliminar Agregar

<<incluir>>

Apoyos <<incluir>> Modificar

<<incluir>>

Eliminar

Agregar

Bolsa de trabajo

<<incluir>>

<<incluir>>

<<incluir>>

Modificar

Eliminar

Figura 62 Caso de uso "Pantalla principal administrador"

Especificaciones de caso de uso:

|  |  |  |
| --- | --- | --- |
| **Caso de uso “Pantalla principal administrador”** | | |
| Nombre | Pantalla principal administrador | |
| Descripción | Navegación en los menús de la pantalla principal. | |
| Actores | Administrador | |
| Precondición | Que la aplicación se haya ejecutado Conexión a internet.  Conexión a la base de datos satisfactoriamente.  Haber ingresado a la aplicación móvil. | |
| Flujo del sistema | Paso | Acción |
|  | 1 | Se mostrarán los módulos de la aplicación móvil. (Noticias, Talleres, Convocatorias, Webinar, Foros,  Escuelas, Carreras, Becas y Apoyos). |
|  | 2 | El usuario al seleccionar algún módulo podrá acceder, para visualizar la información la cual se encuentra en la  base de datos. |
|  | 3 | En cada módulo se contará con tres botones los cuales servirán para agregar, modificar y eliminar  información. |
| Post condición | Si el usuario no cuenta con una conexión a internet o un plan de datos no podrá acceder a la  aplicación. | |

Tabla 21 Especificación de caso de uso pantalla principal administrador

* + - 1. Diagramas de secuencias

En este apartado se desarrollaron los diagramas de secuencias los cuales sirvieron para tener una noción más clara del funcionamiento de la aplicación móvil.

* + - * 1. Diagrama de secuencia inicio de sesión joven

**Joven**

**Interfaz con la aplicación**

**Base de datos**

1. Ingresar a la aplicación



1. Seleccionar Usuario
2. Ingresar correo
3. Ingresar contraseña
4. Enviar correo
5. Enviar contraseña
   1. Consultar datos
   2. Comparar datos
   3. Iniciar sesión
   4. Registrarse como un

usuario nuevo

* 1. Envia formulario para registro
  2. Rellena formulario
  3. Envia datos de usuario
  4. Guarda los datos
  5. Inicia sesión

Figura 63 Diagrama de secuencia inicio de sesión joven

* + - * 1. Diagrama de secuencia inicio de sesión empresa

**Empresa**

**Interfaz con la aplicación**

**Base de datos**

1. Ingresar a la aplicación



1. Seleccionar Usuario
2. Ingresar correo
3. Ingresar contraseña
4. Enviar correo
5. Enviar contraseña
   1. Consultar datos
   2. Comparar datos
   3. Iniciar sesión
   4. Registrarse como un

usuario nuevo

* 1. Envia formulario para registro
  2. Rellena formulario
  3. Envia datos de usuario
  4. Guarda los datos
  5. Inicia sesión

Figura 64 Diagrama de secuencias inicio de sesión empresa

* + - * 1. Diagrama de secuencia inicio de sesión administrador

**Administrador**

**Interfaz con la aplicación**

**Base de datos**

1. Ingresar a la aplicación



* 1. Ingresar correo
  2. Ingresar contraseña
  3. Enviar correo
  4. Enviar contraseña
     1. Consultar datos
     2. Comparar datos
     3. Iniciar sesión

Figura 65 Diagrama de secuencias inicio de sesión administrador

* + - * 1. Diagrama de secuencia perfil de usuario joven

**Joven**

**Aplicación**

**Internet**

**WebServer**

**Base de datos**

1. Modificar información



1. Establece conexión
2. Envia parámetros. *set()*
3. Arma Consulta. (Insert*)*
4. Valida autenticación

y realiza consulta

1. (Update) Tablas

10. Muestra información

9. Carga información

8. Recibe parámetros. *get ()*

1. Devuelve consulta
   1. Muestra información (ayuda, terminos

y condiciones)

* 1. Establece conexión
  2. Envia parámetros. *set()*
  3. Arma consulta (select)
  4. Realiza consulta

19. Muestra información

18. Carga información 17. Recibe parámetros. get()

* 1. Devuelve información
     1. Mostrar información (menús)
     2. Establece conexión
     3. Envia parámetros. *set()*
     4. Arma Consulta. (Insert*)*
     5. Realiza consulta

28. Muestra información

27. Carga información

26. Recibe parámetros. *get ()*

25. Devuelve consulta

Figura 66 Diagrama de secuencia perfil de usuario joven

* + - * 1. Diagrama de secuencia perfil de usuario empresa

**Aplicación WebServer**



**Base de datos**

**Internet**

**Empresa**

1. Modificar información
2. Establece conexión
3. Envia parámetros. *set()*
4. Arma Consulta. (Insert*)*
5. Valida autenticación

y realiza consulta

1. (Update) Tablas

10. Muestra información

9. Carga información

8. Recibe parámetros. *get ()*

1. Devuelve consulta
   1. Muestra información (ayuda, terminos

y condiciones)

* 1. Establece conexión
  2. Envia parámetros. *set()*
  3. Arma consulta (select)
  4. Realiza consulta

19. Muestra información

18. Carga información

* 1. Recibe parámetros. get()

1. Devuelve información
   1. Mostrar información (menús)
   2. Establece conexión
   3. Envia parámetros. *set()*
   4. Arma Consulta. (Insert*)*
   5. Realiza consulta

28. Muestra información

27. Carga información

1. Recibe parámetros. *get ()*
   1. Devuelve consulta

Figura 67 Diagrama de secuencia perfil de usuario empresa

**Administrador**

**Aplicación**

**Internet**

**WebServer**

**Base de datos**

* + 1. Modificar información
    2. Establece conexión
    3. Envia parámetros. *set()*
    4. Arma Consulta. (Insert*)*
    5. Valida autenticación

y realiza consulta

* + 1. (Update) Tablas

10. Muestra información

9. Carga información

8. Recibe parámetros. *get ()*

* + 1. Devuelve consulta
       1. Muestra información (ayuda, terminos

y condiciones)

* + - 1. Establece conexión
      2. Envia parámetros. *set()*
      3. Arma consulta (select)
      4. Realiza consulta

19. Muestra información

18. Carga información 17. Recibe parámetros. get()

* + - 1. Devuelve información
         1. Mostrar información (menús)
         2. Establece conexión
         3. Envia parámetros. *set()*
         4. Arma Consulta. (Insert*)*
         5. Realiza consulta

28. Muestra información

27. Carga información

1. Recibe parámetros. *get ()*
   1. Devuelve consulta

Figura 68 Diagrama de secuencia perfil de usuario administrador

* + - * 1. Diagrama de secuencia pantalla principal joven

**Joven**

**Aplicación**

**Internet**

**WebServer**

**Base de datos**

1 Registro (talleres)

* + 1. Establece conexión
    2. Envia parámetros. *set()*
    3. Arma Consulta. (Select*)*
    4. Valida autenticación

y realiza consulta

* + 1. (Update) Tablas

10. Muestra información

9. Carga información

8. Recibe parámetros. *get ()*

1. Devuelve consulta
   1. Visualizar información

(menús)

* 1. Establece conexión
  2. Envia parámetros. *set()*
  3. Arma Consulta. (Select*)*
  4. Valida autenticación

y realiza consulta

19. Carga información

* + - 1. Recibe parámetros. *get ()*

16. Devuelve consulta

20. Muestra información

Figura 69 Diagrama de secuencia pantalla de principal joven

* + - * 1. Diagrama de secuencia pantalla principal empresa

**Empresa**

**Aplicación**

**Internet**

**WebServer**

**Base de datos**

Agregar información



Establece conexión

Envia parámetros. *set()*

Arma Consulta. (Insert*)*



Valida autenticación

y realiza consulta

(Update) Tablas

10. Muestra información

9. Carga información

1. Recibe parámetros. *get ()*
   1. Devuelve consulta
      1. Modificar información
      2. Establece conexión
      3. Envia parámetros. *set()*
      4. Arma Consulta. (Insert*)*
      5. Valida autenticación

y realiza consulta

* + 1. (Update) Tablas

20. Muestra información

19. Carga información

1. Recibe parámetros. *get ()*
   1. Devuelve consulta
      1. Elimnar información
      2. Establece conexión
      3. Envia parámetros. *set()*
      4. Arma Consulta. (Insert*)*
      5. Valida autenticación

y realiza consulta

* + 1. (Update) Tablas

30. Muestra información

29. Carga información

1. Recibe parámetros. *get ()*
   1. Devuelve consulta
      1. Visualizar información
      2. Establece conexión
      3. Envia parámetros. *set()*
      4. Arma Consulta. (Select*)*
      5. Valida autenticación

y realiza consulta

40. Muestra información

39. Carga información

1. Recibe parámetros. *get ()*

36. Devuelve consulta

Figura 70 Diagrama de secuencia pantalla principal empresa

* + - * 1. Diagrama de secuencia pantalla principal administrador

**Aplicación WebServer**



**Base de datos**

**Internet**

**Administrador**

Visualizar información

Establece conexión

Envia parámetros. *set()*

Arma Consulta. (Insert*)*

Realiza consulta

9. Muestra información

8. Carga información

7. Recibe parámetros.

*get ()*

1. Devuelve consulta
   1. Agregar información
   2. Establece conexión
   3. Envia parámetros. *set()*
   4. Arma Consulta. (Insert*)*
   5. Valida autenticación

y realiza consulta

* 1. (Update) Tablas

19. Muestra información

17. Carga información

* 1. Recibe parámetros. *get ()*

1. Devuelve consulta
   1. Modificar información
   2. Establece conexión
   3. Envia parámetros. *set()*
   4. Arma Consulta. (Insert*)*
   5. Valida autenticación y realiza consulta
   6. (Update) Tablas
2. Muestra información
3. Elimnar información

28. Carga información

* 1. Recibe parámetros. *get ()*
     1. Devuelve consulta
        1. Establece conexión
        2. Envia parámetros. *set()*
        3. Arma Consulta. (Insert*)*
        4. Valida autenticación

y realiza consulta

39. Muestra información

38. Carga información

1. Recibe parámetros. *get ()*
   1. (Update) Tablas
   2. Devuelve consulta

Figura 71 Diagrama de secuencia pantalla principal administrador

* 1. Fase de desarrollo

En este apartado se mostrará la instalación y configuración de las tecnologías utilizadas para el desarrollo de la aplicación para posteriormente pasar a mostrar el código y el resultado obtenido del desarrollo de la aplicación móvil.

* + 1. Instalación y configuración de la tecnología implementada en Windows

En este punto se mostrará las instalaciones y configuraciones del software que se utilizó para el desarrollo de la aplicación móvil en el sistema operativo Windows.

* + - 1. Instalación y configuración de Node JS

Para la creación del Rest Full Server se utilizó Node JS, este software se descargó de la página oficial <https://nodejs.org/es/>. En la página oficial existen diferentes versiones, en este caso se utilizó la versión 14.17.4 LTS, a continuación, se muestran los pasos de su instalación:

* + - * 1. Descargar Node JS <https://nodejs.org/es/>.



Figura 72 Descarga de Node JS

* + - * 1. Se ejecuta el archivo.

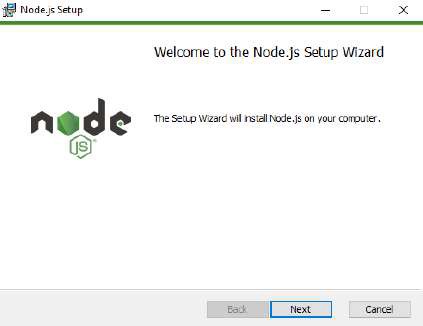


Figura 73 Instalación de Node JS

* + - * 1. En la siguiente ventana la configuración queda por defecto.

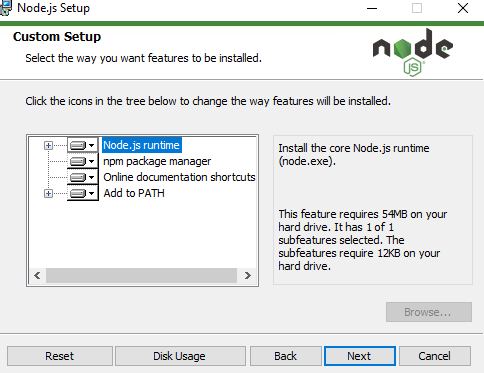


Figura 74 Configuración de Node JS

* + - * 1. Al término de la instalación se ejecuta una terminal en la cual se ingresa la siguiente línea de comando “node -v” este comando ayuda a verificar que Node JS se encuentre instalado en el equipo.

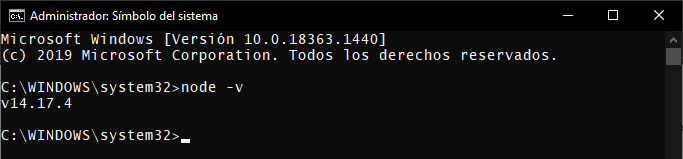


Figura 75 Comprobación del Node JS

* + - * 1. En la misma terminal instalaran unas librerías de node js para ello se

ingresará el siguiente comando “npm install -g npm”.



Figura 76 Instalación de librerías de Node JS

* + - * 1. Después de haber terminado la instalación del comando “npm install – g nom” se continuará con la instalación de nodemon el cual que el anterior son librerías de node js, para esto se tecleara el comando “npm install -g nodemon” y con esto terminamos la instalación y configuración de Node JS.



Figura 77 Instalación del nodemon en Node JS

* + - 1. Instalación y configuración de Mongo Compass

Mongo Compass fue el entorno de trabajo en el cual se realizaron las consultas de la base de datos a continuación se muestra su instalación y configuración:

* + - * 1. Descargar Mongo Compass<https://www.mongodb.com/try/download/compass>.

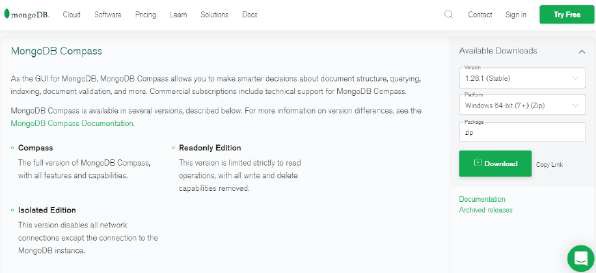


Figura 78 Descarga de Mongo Compass

* + - * 1. Una vez descargado el archivo, se procede a descomprimir, este creara una carpeta con el nombre de mongodb\_compass, en donde contendra el software a utilizar.



Figura 79 Instalación de Mongo Compass

* + - * 1. Dentro de la carpeta creada anteriormente se encuentra un archivo con el nombre de MongoDBCompass.exe. Al ejecutar el archivo abrirá la app mostrando un apartado para insertar un link el cual será obtenido de Mongo DB este servirá para poder administrar nuestra base de datos, con esto queda instalado y configurado Mongo Compass.

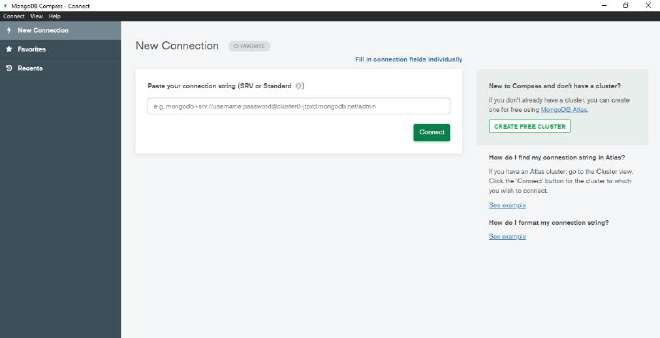


Figura 80 Área de trabajo de Mongo Compass

* + - 1. Instalación y configuración de Postman

Postman fue utilizado para testear el Rest Server, este fue de gran utilidad para el desarrollo de la aplicación, a continuación, se muestra su instalación y configuración.

* + - * 1. Descargar Postman <https://www.postman.com/downloads/>



Figura 81 Descarga de postman

* + - * 1. Una vez descargado el archivo, se ejecuta y abrira el software para poder desarrollar los CRUD, con esto se finaliza la instalacion y configuracion de Postman.

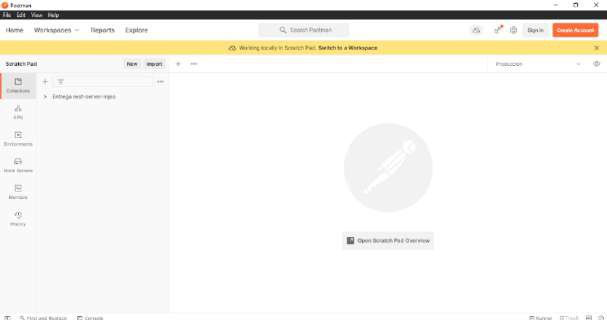


Figura 82 Área de trabajo de postman

* + - 1. Instalación y configuración de Git Bash

El git bash es una aplicación el cual ofrece una capa de emulación para una mejor experiencia de líneas de comandos a continuación se muestra su instalación y configuración.

* + - * 1. Descargar Git Bash <https://git-scm.com/downloads>.

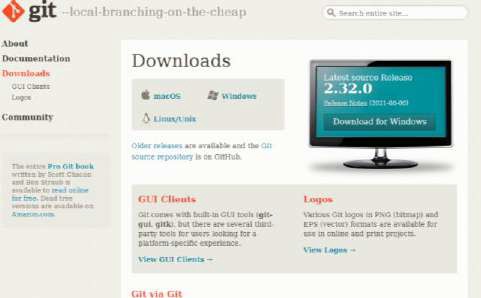


Figura 83 Descarga de git bash

* + - * 1. Descargado el archivo se ejecuta y se prosigue con la instalación cabe recalcar que la configuración queda por defecto solo se proseguirá a dar siguiente hasta terminar la instalación.

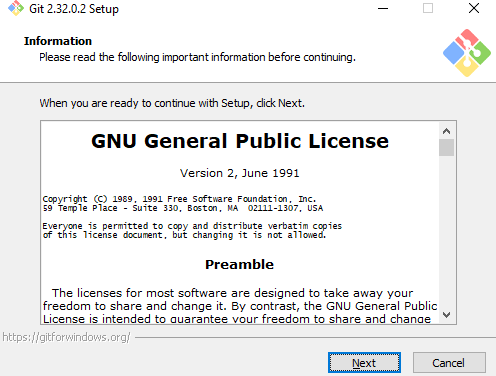


Figura 84 Instalación del git bash

* + - 1. Instalación y configuración de Android Studio

Android Studio se utilizó para montar la aplicación móvil en su emulador ya que gracias su amplia variedad de sdk permitió probar la aplicación en diferentes versiones de Android, a continuación, se muestra su instalación y configuración.

* + - * 1. Descargar Android Studio <https://developer.android.com/studio>.

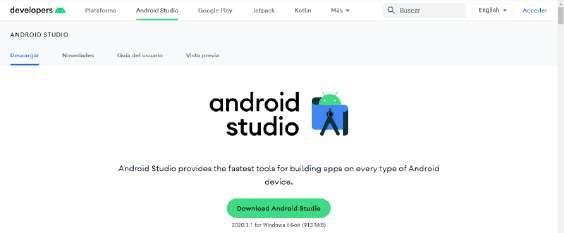


Figura 85 Descarga de Android Studio

* + - * 1. Se ejecuta el archivo que se descargó anteriormente y la configuración queda por defecto, solo daremos siguiente hasta que termine la instalación.



Figura 86 Instalación de Android Studio

* + - * 1. Una vez instalado Android Studio se pasará a descargar los SDK a utilizar. En la barra de menús se localiza la opción de Tool en la cual se buscará SDK Manager.

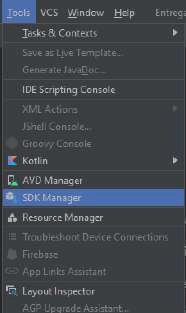


Figura 87

Configurac ión de los SDK

* + - * 1. Estando dentro del SDK Manager seleccionamos la versión de Android a utiliza, con esto se terminar la instalación y configuración de Android Studio.

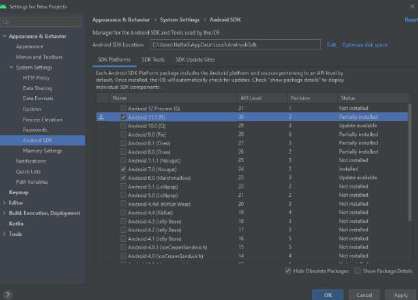


Figura 88 Descarga de los paquetes SDK

* + - 1. Instalación y configuración de Visual Studio Code

Visual Studio Code fue el editor de código en donde se desarrolló la aplicación móvil, ya que es compatible con varios lenguajes de programación en este caso Dart, a continuación, se muestra su instalación y configuración:

* + - * 1. Descargar Visual Studio Code <https://code.visualstudio.com/download>.

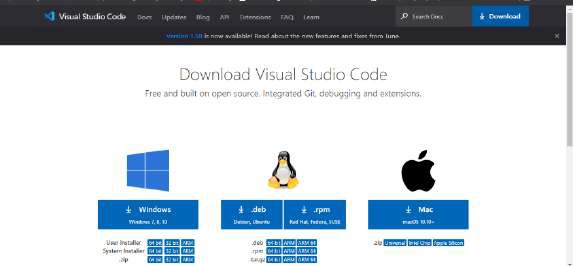


Figura 89 Descarga de Visual Studio Code

* + - * 1. Se ejecuta el archivo descargado el cual lanzara una ventana para la instalación de Visual Code Studio.



Figura 90 Instalación de Visual Code Studio

* + - * 1. Se procede a elegir el lugar en donde se desea guardar los datos del software, en este caso se queda por defecto.



Figura 91 Configuración datos de Visual Studio Code

* + - * 1. En caso de no estar seleccionado el PATH se marca la casilla y se prosigue con la instalación y con esto se concluye la instalación y configuración de Visual Code Studio.

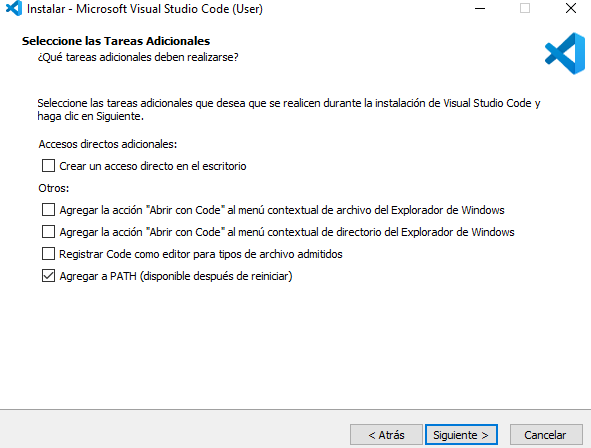


Figura 92 Configuración del PATH en Visual Studio Code

* + - 1. Instalación y configuración de Flutter

Como ya se mencionó anteriormente en el punto “*2.3.2 Tecnología framework a utilizar”* Flutter es el framework en donde se desarrolló la aplicación móvil, a continuación, se muestra su instalación y configuración.

* + - * 1. Descargar flutter <https://flutter.dev/>.

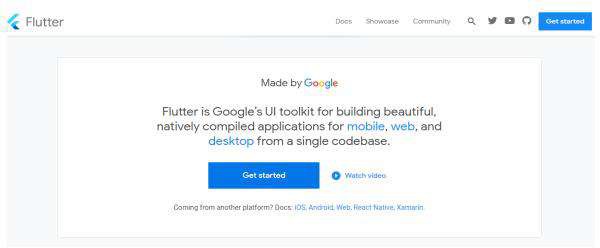


Figura 93 Descarga de Fluttter

* + - * 1. Se descargará un archivo zip el cual se descomprimirá, pero antes de eso es necesario crear una carpeta con el nombre de “src” en el disco local C de nuestro equipo, para posteriormente migrar los datos descomprimidos a la carpeta src. Ver figura 94 y 95.



Figura 94 Carpeta src



Figura 95 Carpeta flutter

* + - * 1. Dentro de la carpeta “src” se abrirá una terminal con Git Bash en la cual se ingresa el código “flutter -versión” este verificará que se encuentre correctamente instalado Flutter.

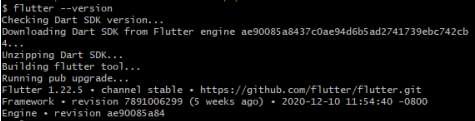


Figura 96 Verificación de flutter

* + - * 1. En la misma terminal se ingresa el comando “flutter doctor” este se encargará de verificar que todos los paquetes que utiliza flutter se encuentren actualizados o instalados correctamente en caso de que un paquete no se encuentre en la pantalla se visualizara con un signo de admiración (!). Para instalar el paquete es volver a poner el comando de flutter doctor con el nombre del paquete faltante.

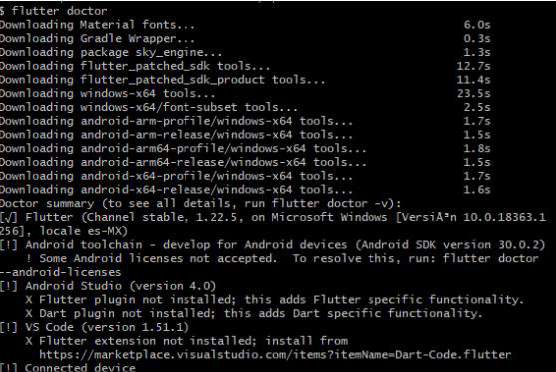


Figura 97 Descarga de paquetes de flutter

* + - * 1. En el buscador de Windows se busca variables de entorno y se agrega las variables para flutter.

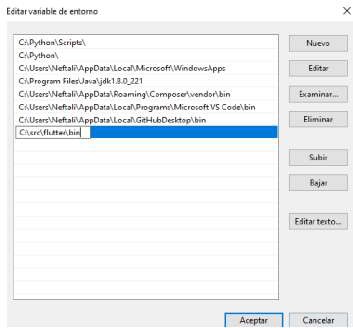


Figura 98 Configuración de las variables de entorno

* + - * 1. Se ejecuta Visual Studio Code para instalar las extensiones de “Flutter” y “Dark”, con esto se finaliza la instalación y configuración de flutter.

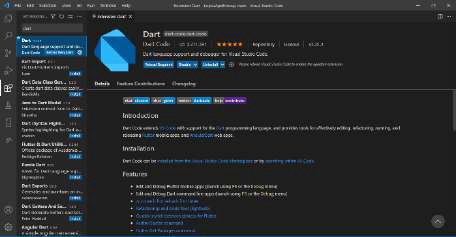
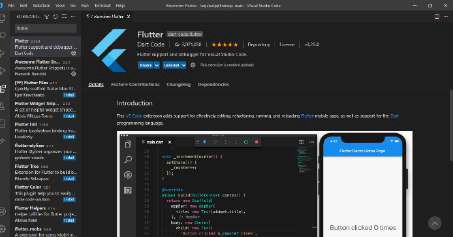
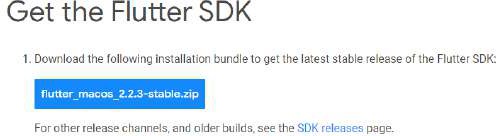


Figura 99 Configuración de Flutter y Dark

* + 1. Instalación y configuración de la tecnología implementada en macOS

En este punto se mostrará las instalaciones y configuraciones del software que se utilizó para el desarrollo de la aplicación móvil en el sistema operativo MacOS.

* + - 1. Instalación y configuración de Flutter en macOS
         1. Descargar flutter <https://flutter.dev/>.



* + - * 1. En el directorio principal del equipo se crea una carpeta con el nombre

“development”, dentro de esa carpeta de moverán los datos de flutter.

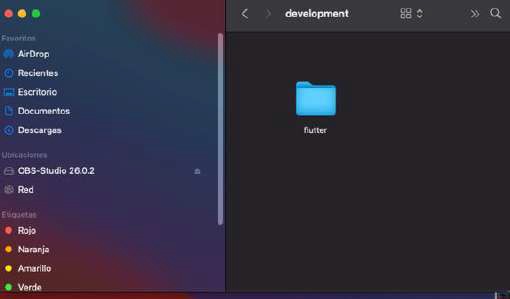


Figura 100 Configuración de flutter en el equipo

* + - * 1. Ahora se configurará el PATH para esto se abrirá una terminal y se ingresará el

siguiente comando “touch .zshrc”.



Figura 101 Configuración del PATH

“Creación del archivo .zshrc”

* + - * 1. Nos dirigiremos a la carpeta en donde se creó el archivo .zshrc y pegaremos el siguiente comando

PATH="$PATH:[PATH\_OF\_FLUTTER\_GIT\_DIRECTORY]/bin”, pero en el cual

se le agregará la dirección en donde se encuentra la carpeta de flutter.



Figura 102 Configuración del PATH

* + - * 1. En una nueva terminal se ejecutará el comando “source $HOME/.zshrc”

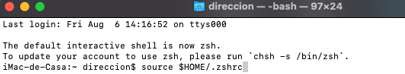


Figura 103 Configuración del archivo .zshrc

* + - * 1. En la misma terminal se ejecutará el código “flutter doctor” para completar la

instalación de flutter.



Figura 104 Actualización de componentes de flutter

* + - 1. Instalación y configuración de Xcode

Xcode fue utilizado para emular el dispositivo virtual iOS, a continuación, se muestra su instalación y configuración.

* + - * 1. Se ingresa a la tienda oficial de macOS, en la cual se busca la aplicación.



Figura 105 Instalación de Xcode

* + - * 1. En una terminal se instalarán componentes adicionales de Xcode se

ingresa el siguiente comando en la terminal “sudo xcode-select --switch

/Applications/Xcode.app/Contents/Developer” y “sudo xcodebuild -

runFirstLaunch”



Figura 106 Instalación de componentes adicionales Xcode

* + - * 1. En una nueva terminal se ingresar el siguiente comando el cual sirve para implementar una aplicación desarrollada en flutter en un dispositivo iOS físico “sudo gem install cocoapods” y con eso queda configurado Xcode.



Figura 107 Configuración del dispositivo físico en iOS

* + - 1. Instalación y configuración de Android Studio
         1. Para la instalación de Android Studio en macOS, se ingresa a la página oficial de Android Studio en donde descargara un archivo de extensión zip, al término de la descarga automáticamente se descomprime y se prosigue con la instalación.



Figura 108 Instalación de Android Studio en macOS

* + - * 1. La configuración de Android Studio queda por defecto, solo esperamos a que termine la instalación, al término de la instalación al igual que en Windows se procede a instalar sus sdk y con esto se termina la instalación y configuración de Android Studio.

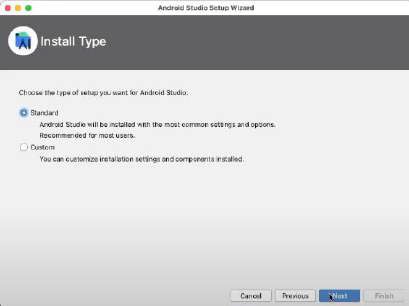


Figura 109 Configuración de Android Studio en macOS

* + - 1. Instalación y configuración de Postman
         1. Para la instalación de Postman se descargó el archivo de su página oficial, pero en este caso para macOS, se descargará un archivo zip, al descomprimirlo creará un ejecutable listo para su uso.



Figura 110

Aplicac ión postma n en macO S

* + - 1. Instalación y configuración de Visual Studio Code
         1. Para la instalación de Visual Studio Code se descargó el archivo de su página oficial, pero en este caso para macOS, se descargará un archivo zip, al descomprimirlo creará un ejecutable listo para su uso, al igual que Windows es necesario instalar sus extensiones y con esto se concluye la instalación y configuración.



Figura 111

Aplicac ión Visual Studio Code en macO S

* 1. Desarrollo de la aplicación móvil
     1. Creación del restfull server

Para la creación de la aplicación fue necesario crear un restfull server el cual se encargará de administrar de información a nuestra aplicación móvil, este restfull server se empezó desde cero el cual cuenta con diferentes módulos los cuales son, *controllers, database, helpers, middlewares, models y routes*, se tomara un ejemplo de cada uno y se explicara su funcionamiento, cabe destacar que solo se tomara un ejemplo ya que si se toman más ejemplos el documento se volverá muy extenso y repetitivo.

* + - 1. Paqueterías utilizadas

A continuación, se muestran todas las paqueterías utilizadas para el desarrollo del restfull server. Ver figura 112

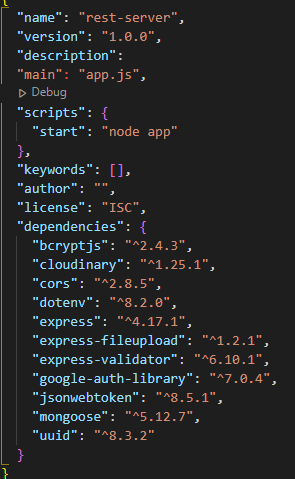


Figura 112 Paqueterías utilizadas para el restfull server.

* + - 1. Conexiones del Restfull server Conexiones del restfull server. Ver figura 113



Figura 113 Conexiones del restfull server

* + - 1. Database

En el módulo de database (base de datos) existe un archivo con el nombre de *config.js* el cual es el encargado de hacer la conexión de la base de datos. Ver figura 114

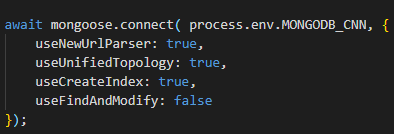


Figura 114 Conexión a la base de datos

* + - 1. Models

Los modelos sirvieron para la representación de los datos que se manejaran en la aplicación móvil para ello se creó un archivo con el nombre de server.js en el cual se creó un método constructor para posteriormente hacer las respectivas conexiones a los diferentes módulos de la aplicación.

Conexión de la base de datos. Ver figura 115



Figura 115 Llamado de la base de datos.

Función para subir archivos en este caso imágenes. Ver figura 116



Figura 116 Upload de archivos.

Creación del método constructor y creación de las rutas de ficheros (PATH). Ver figura 117

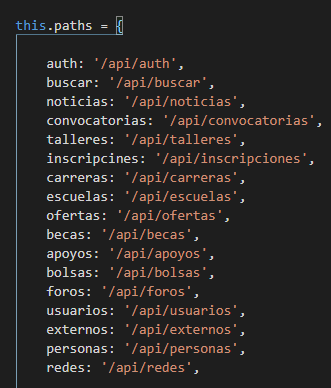


Figura 117 Creación del constructor PATH.

Configuración de las rutas (PATH). Ver figura 118

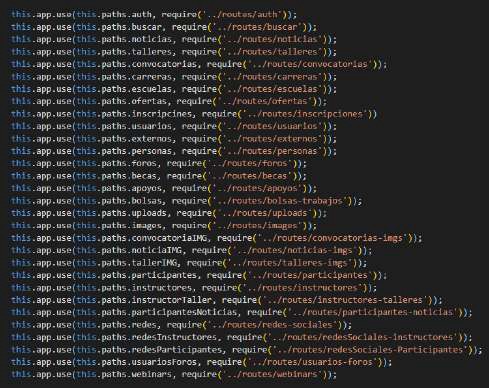


Figura 118 Configuración de las rutas PATH.

Posteriormente se crearon los diferentes modelos de la aplicación, cabe mencionar que la estructura de los modelos es similar. Estos fueron creados conforme están las tablas de la base de datos en este caso nos encontramos con el *nombre, descripción, requisitos, enlace, img, usuario\_id, estado, fecha de registro y fecha de eliminación.* Ver figura 119

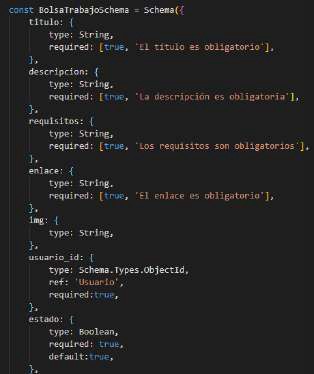


Figura 119 Creación de los modelos.

Los otros modelos creados fueron los siguientes. Ver figura 120

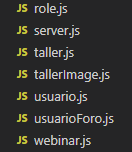
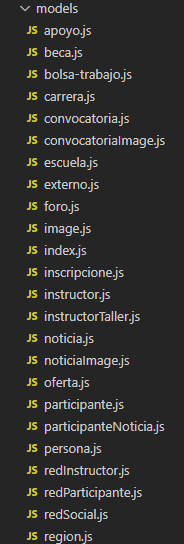


Figura 120 Modelos creados para el restfull server.

* + - 1. Controllers

Los controladores fueron los encargados de responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una búsqueda, agregar información, etc. En este caso se tomaron dos ejemplos ya que uno corresponde al inicio de sesión el cual es totalmente diferente a los demás controladores.

* + - * 1. Controller Auth

Este controlador se enfoca en el inicio de sesión de la aplicación móvil.

La variable const presentan un ámbito de bloque (block scope) tal y como lo hacen las variables definidas en este caso se están mandado a llamar los bloques que aparecen en color rojo. Ver figura 121

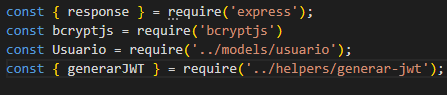


Figura 121 Llamado de diferentes bloques.

Por medio del siguiente try catch se valida el correo electrónico del usuario. Ver figura 122

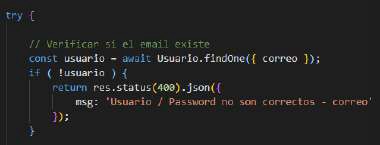


Figura 122 Validación de correo electrónico.

Validación si el usuario esta activo. Ver figura 123

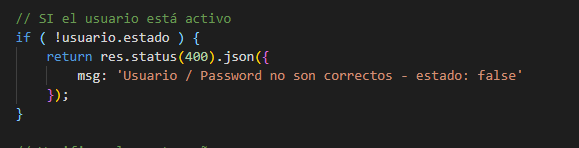


Figura 123 Validación de usuario activo.

Verifica si la contraseña es la correcta. Ver figura 124

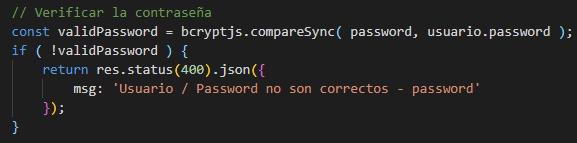


Figura 124 Validación de contraseña.

Se genera un token de validación para el usuario. Ver figura 125

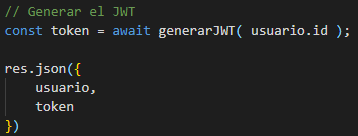


Figura 125 Validación de token de usuario.

* + - * 1. Controller ejemplo

En este punto se tomará de ejemplo un controlador en este caso el contraller apoyo para mostrar su estructura.

Por medio del siguiente código se logra obtener todos los apoyos registrados. Ver figura 126

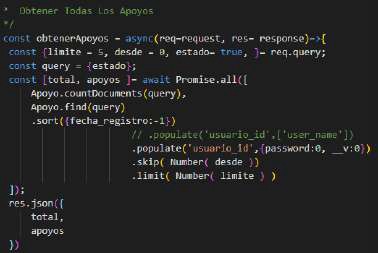


Figura 126 Obtener todos los registros.

Obtener un solo registro. Ver figura127



Figura 127 Obtener un solo registro.

Creación de un registro. Ver figura128

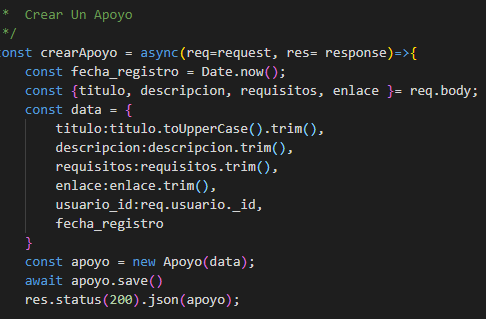


Figura 128 Creación de un registro.

Actualizar los registros. Ver figura 129

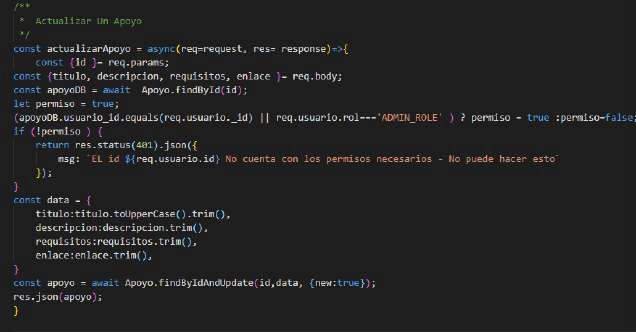


Figura 129 Actualización de registros

Eliminar registros. Ver figura 130

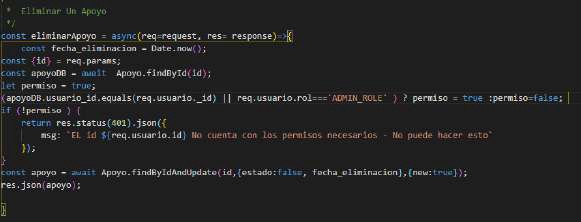


Figura 130 Eliminación de registros.

Los otros controladores que se crearon son los siguientes:

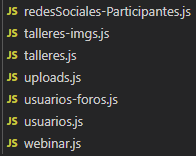
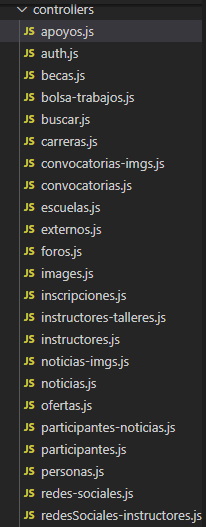


Figura 131 Controladores creados para el restfull server.

* + - 1. Middleware

Las funciones middleware se utilizaron como mecanismo para verificar niveles de acceso antes de entrar en una ruta, manejo de errores, validación de datos, etc. Al igual que lo demás modelos se tomará de ejemplo un solo modulo ya que las validaciones se asemejan.

Validación del token en caso de contar con un token. Ver figura 132

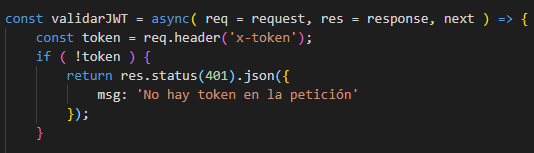


Figura 132 Validación del token.

Validación de token en caso de ser incorrecto o usuario no existente en la base de datos. Ver figura 133

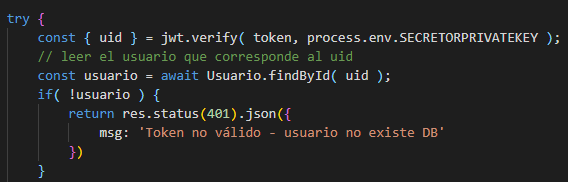


Figura 133 Validación de token incorrecto.

Validación de token con usuario eliminado de la base de datos. Ver figura 134

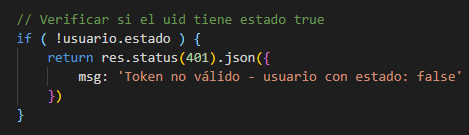


Figura 134 Validación de token de usuario eliminado.

En caso de ingresar el token mal por ejemplo no haberlo copiado correctamente. Ver figura135

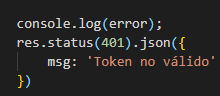


Figura 135 Validación de token por error del usuario.

En el ejemplo anterior se mostró algunas validaciones que se crearon para la aplicación móvil a continuación se muestran los demás módulos creados.

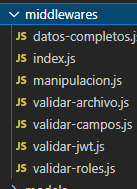


Figura 136

Middleware s creados para el restfull server.

* + - 1. Helpers

Los helpers se encargan en de resolver los problemas de incompatibilidades.

Declaración de tipos de archivos. Ver figura 137



Figura 137 Declaración de archivos.

Se extrae el nombre del archivo. Ver figura 138

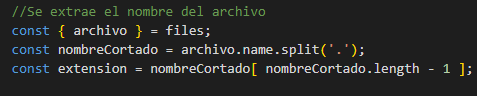


Figura 138 Extracción de archivos.

Valida el tipo de extensión. Ver figura139

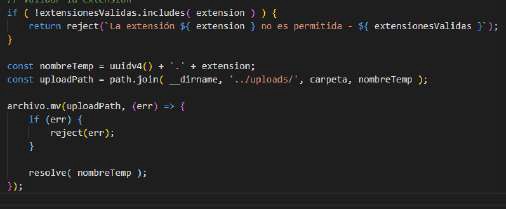


Figura 139 Validación del tipo de extensión.

A continuación, se muestra los demás módulos de Helpers desarrollados. Ver figura 140

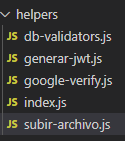


Figura 140

Validacione s creadas para el restfull server.

* + - 1. Routes

Se utilizaron para crear una vía de acceso específica a una ubicación.

Ruta a los Middlewares. Ver figura 141

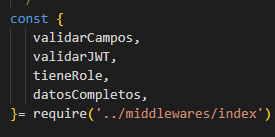


Figura 141 Rutas de los middlewares.

Ruta los Helpers. Ver figura 142

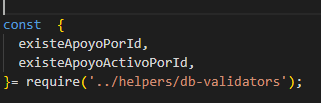


Figura 142 Rutas de los helpers.

Ruta a los controladores. Ver figura 143

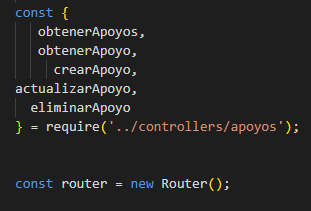


Figura 143 Rutas de los controladores.

Obtener todos los registros. Ver figura 144



Figura 144 Obtención de los registros.

Obtener un solo registro. Ver figura 145

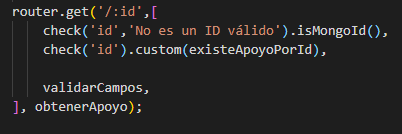


Figura 145 Obtener un solo registro.

Crear y guardar un registro. Ver figura 146

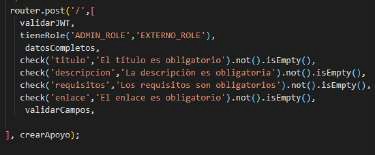


Figura 146 Creas y guardar un registro.

Actualizar registros. Ver figura 147

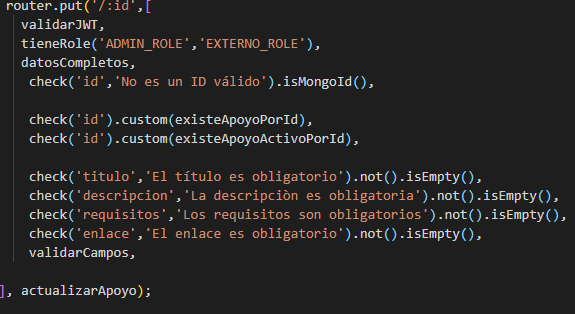


Figura 147 Actualizar registros.

Eliminar registros. Ver figura 148

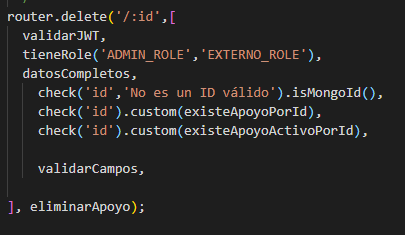


Figura 148 Eliminar registro.

A continuación, se muestra los demás módulos de routes desarrollados. Ver figura 149

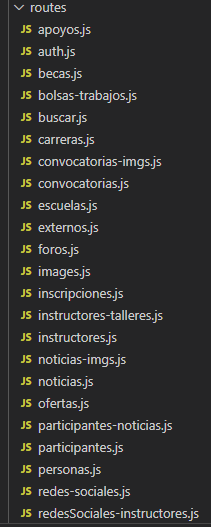


Figura 149 Diferentes routes creadas para el restfull server.

* + 1. Pruebas del Restfull Server

Después de haber terminado el restfull server se hicieron las pruebas pertinentes, todo esto se realizó en Postman, fue necesario crear los CRUD de los diferentes módulos que se programaron.

Modulos desarrollados. Ver figura 150

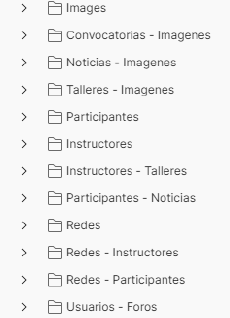
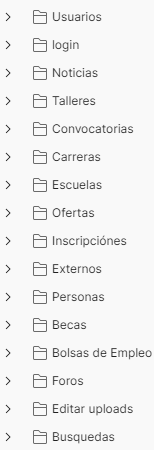


Figura 150 Módulos creados en Postman.

Dentro de cada módulo se crearon sus respectivos CRUD. Ver figura 151



Figura 151 Estructura

de los módulos.

* + - 1. CRUD

En este apartado se mostrará un ejemplo del funcionamiento de los CRUDS.

* + - 1. GET

Su función es obtener todos los registros del web server. Para inicializar la búsqueda se inserta la url a consultar. Ver figura 152



Figura 152 Método get.

Se realiza la búsqueda. Ver figura153



Figura 153 Resultado de la consulta.

* + - 1. POST

Se inserta la url. Ver figura 154



Figura 154 Método post.

Se rellenan los campos solicitados. Ver figura 155



Figura 155 Registro de datos.

En caso de no contar un token mandara una alerta. Ver figura 156

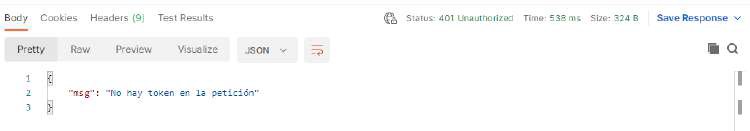


Figura 156 Error de token de usuario.

Para obtener un token es necesario iniciar sesión en el módulo de *inicio de sesión.* Ver figura 157

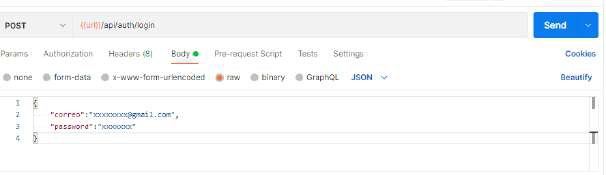


Figura 157 Obtener token de usuario.

Posteriormente mandara un token de usuario. Ver figura 158

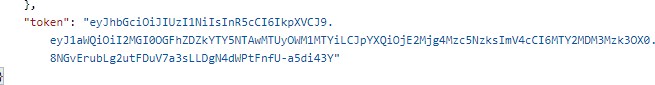


Figura 158 Token de usuario.

Ese token se ingresará en el apartado de headers. Ver figura 159



Figura 159 Menú headers.

Posteriormente en la terminal nos mostrara los datos enviados a la base de datos. Ver figura 160



Figura 160 Resultado de la consulta.

* + - 1. GET

La función de este GET es obtener detalle de un solo registro para ello se ingresa el id de la información a buscar. Ver figura 161



Figura 161 Método get.

En la terminal mostrara toda la información del registro, así como diferentes datos del usuario que registro esa información. Ver figura 162

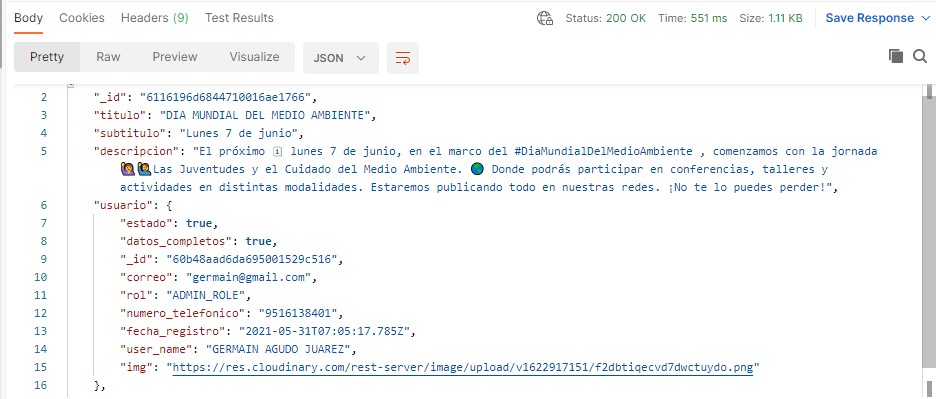


Figura 162 Resultado de la consulta.

* + - 1. PUT

La función del PUT es actualizar los datos, se ingresa el id del registro a modificar. Ver figura 163



Figura 163 Método put.

Se envía la solicitud con los datos a modificar. Ver figura 164

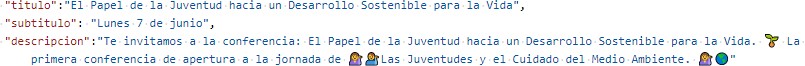


Figura 164 Registros a modificar.

La información modificada se muestra en la terminal. Ver figura 165

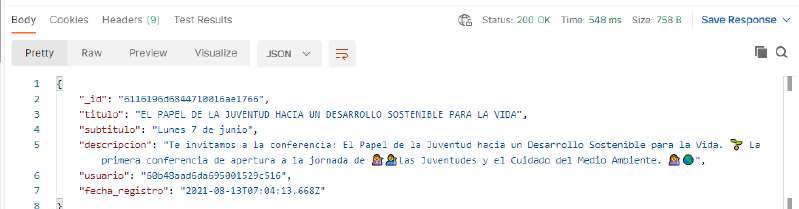


Figura 165 Resultado de la consulta.

* + - 1. DELETE

Para eliminar un registro es necesario ingresar su id en la barra de búsqueda. Ver figura 166



Figura 166 Método get.

En la terminal nos muestra los datos eliminados. Ver figura 167



Figura 167 Resultado de la consulta.

Cabe recalcar que en la terminal en la parte superior derecha aparece el estado de la creación los cuales se crearon en el Middlewares.



Figura 168 Estado de la creación de las consultas.

* + 1. Desarrollo de la aplicación móvil

En este apartado se mostrará el desarrollo de la aplicación móvil.

* + - 1. Creación de los modelos

Para el desarrollo de este proyecto se definieron los modelos de cada tabla de la base de datos estos modelos se crearon en Visual Studio Code, se tomará el ejemplo de un módulo para mostrar el código fuente ya que la estructura es la misma para todos los módulos.

Constructor del modelo. Ver figura 169

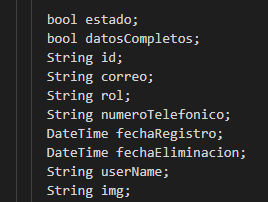


Figura 169

Constructor.

Variables del modelo. Ver figura 170

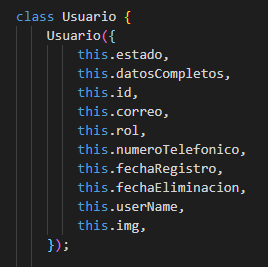


Figura 170 Variables

de entorno.

Convierte todo el cuerpo de nuestra respuesta en instancias. Ver figura 171

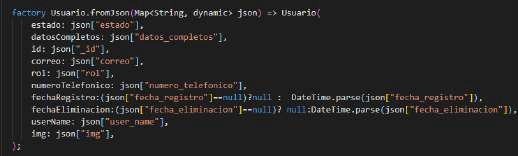


Figura 171 Conversión a instancias.

Convierte el objeto a json. Ver figura 172

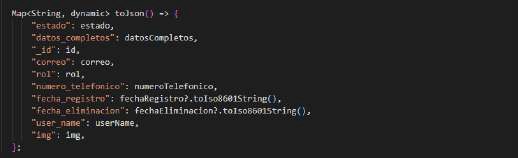


Figura 172 Conversión a json

A continuación, se muestran todos los modelos creados. Ver figura173

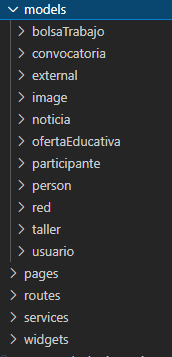


Figura 173

Creación de los diferentes modelos creados para la app móvil.

* + - 1. Provider

Se desarrollo un apartado con las funcionalidades de provider, el cual tiene como función, permitirnos manejar el estado de nuestra app para ello se creó el apartado de servicios los cuales son los encargados de hacer la petición directamente al Restfull Server transformando los datos y filtrando la información. Cabe recalcar que la estructura mostrada a continuación es la misma para los demás modelos ya se asemejan.

Librerías utilizadas para el desarrollo de la aplicación móvil en los módulos de servicios. Ver figura 174



Figura 174 Librerías utilizadas.

La primera línea de código nos indica la dirección del módulo el cual servirá para las pruebas en Postman, en la siguiente línea de código se encuentra el token de usuario el cual sirve como identificador único de un usuario. Ver figura 175



**1**

**2**

Figura 175 Dirección a Postman y token.

La siguiente línea de código es un método fromjson, el cual se encarga de transformar los Json a objetos. Ver figura 176



Figura 176 Método fromJson.

En caso de encontrar algún error durante la ejecución de la aplicación la siguiente línea de código se encargará de cacharlo y enviarlo a la función bloc la cual se explica más adelante. Ver figura 177



Figura 177 Cache de errores.

El ejemplo anterior corresponde al módulo, bolsa de trabajo cabe destacar que el procedimiento que se mostró anteriormente es para obtener todas las ofertas de trabajo en este caso; así como se obtuvieron dichas ofertas de trabajo, también se consta de las siguientes funciones:

Eliminar registro. Ver figura 178

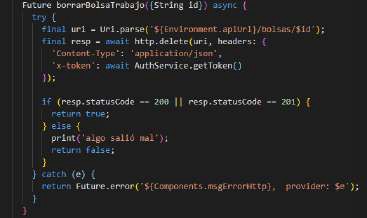


Figura 178 Eliminar registro.

Obtener un solo registro. Ver figura 179

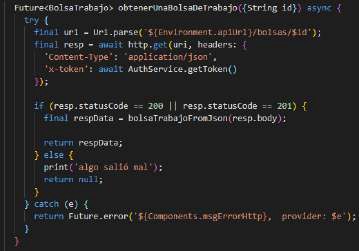


Figura 179 Obtener un solo registro.

Registrar una nueva oferta. Ver figura 180



Figura 180 Registrar datos.

Actualizar oferta de trabajo. Ver figura 181

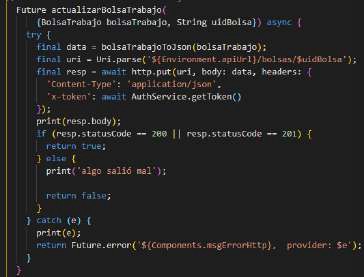


Figura 181 Actualizar datos.

Actualizar imagen de la oferta. Ver figura 182

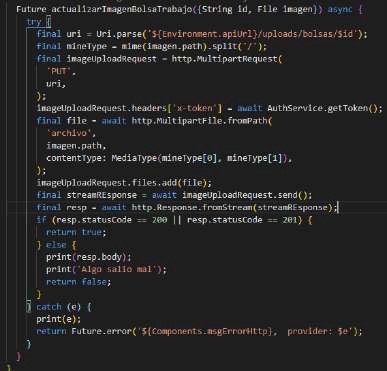


Figura 182 Actualizar imagen.

* + - 1. Bloc

Como se mencionó en el punto 2.4.3.1 Manejadores de estado Bloc se encarga de la gestión y acceso de este, consiste en tener un almacén central de estados, al cual se pude acceder desde cualquier punto de la aplicación para consultar, modificar o crear estados, donde dicho almacén es consultado por las diferentes interfaces de la aplicación, para ello se crearon diferentes módulos enfocados a la funcionalidad de bloc que a continuación se describe su funcionabilidad.

Librerías utilizadas para el desarrollo de la aplicación móvil en los módulos de bloc. Ver figura 183

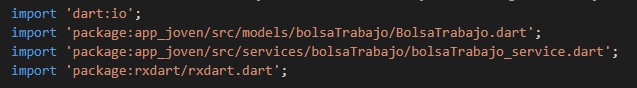


Figura 183 Librerías utilizadas.

El BehaviroSubject es un streamcontroller especial el cual permite enviar datos, errores y eventos hechos al oyente, esta toma el último elemento que se ha agregado al controlador y lo emite como nuevo para los nuevos oyentes. Ver figura 184



Figura 184 Envió de errores al oyente.

En las siguientes dos líneas de código se muestra la creación de un objeto, en este caso se tomó como referencia el módulo de bolsa de trabajo, cabe mencionar que este objecto es igual para los demás módulos el único cambio que sufriría sería el nombre. En la segunda línea de código se logra apreciar un Stream el cual es una secuencia de eventos asíncronos, la función de este es ayudar a refrescar la interfaz de usuario. Ver figura 185



**1**

**2**

Figura 185 Creación de un objeto y secuencia de eventos asíncronos.

En el siguiente try catch se manda a llamar la función *cargarBolsastrabajo,* el cual se encuentra dentro de nuestro provider en este caso definido como servicios el cual lleva el nombre de *Bolsatrabajoservice,* este solicita una petición y si la petición sale correcta agrega esos resultados a la variable

*\_bolsatrabajocontroller,* en caso de que existiera un error se agrega esa variable a bloc para que al finalizar todo ese procedimiento se obtengan los registros almacenados en la base de datos. Ver figura 186

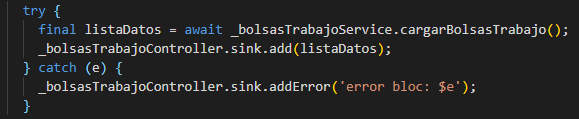


Figura 186 Función try catch.

En los siguientes try catch se muestra las diferentes funciones que se encuentran en el modelo bolsa de trabajo en la figura 186 ya se explicó su funcionamiento ahora solo se listaran los diferentes try catch que se desarrollaron.

Obtener solo un registro. Ver figura187

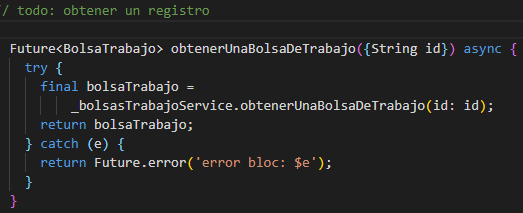


Figura 187 Función para obtener un registro.

Eliminar un registro. Ver figura 188

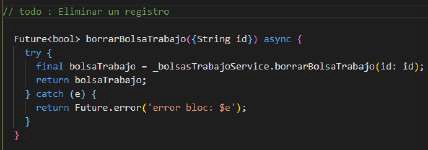


Figura 188 Función para eliminar un registro.

Registrar datos. Ver figura189

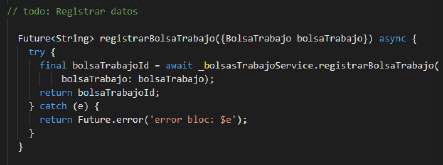


Figura 189 Función para registro de datos.

Actualizar datos. Ver figura 190

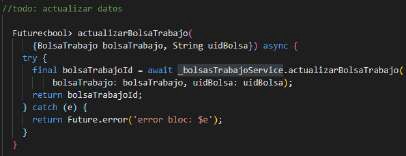


Figura 190 Función para actualización de datos.

Actualizar imagen. Ver figura 191

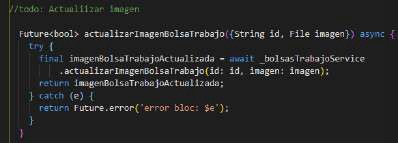


Figura 191 Función para actualización de imagen.

Por último, el método dispose se utiliza cuando un objeto se elimina del árbol de datos de forma permanente. Ver figura 192

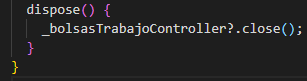


Figura 192 Método dispose.

Ahora se muestra todos los Bloc desarrollados.

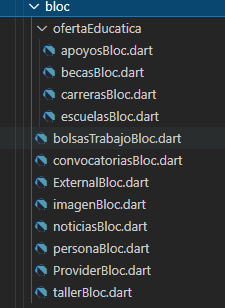


Figura 193 Blocs creados para la aplicación móvil.

* + - 1. Widget Inheritedwidget

Esta clase permite que un widget descendiente pueda acceder a los datos directamente si necesidad de ir escalando entre los widgets padres e hijas, esta clase se complementan con bloc.

Creación de los blocs.

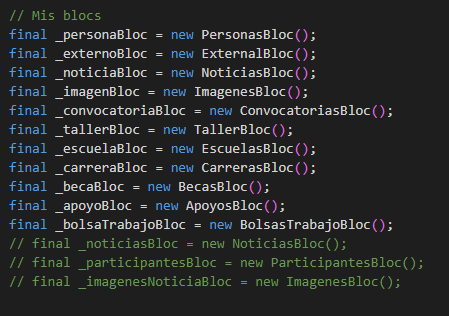


Figura 194 Creación de los blocs.

* + - 1. Pages

Se desarrollo un apartado de pages el cual es el encargado de obtener el valor de bloc este en pocas palabras seria nuestra interfaz visual con la cual el usuario interactuaría.

Librerías utilizadas. Ver figura 195

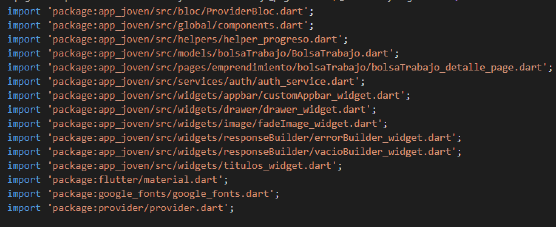


Figura 195 Librerías utilizadas.

Para la creación de las interfaces graficas se usó del StatefulWidget los cuales son widgets cuyo estado puede cambiar a través del tiempo, en pocas palabras, contiene algún estado mutable que se necesita rastrear. Un ejemplo de esto es un widget de texto que muestra la cantidad de veces que se ha hecho clic en un botón. Dado que el texto que muestra dicha interacción cambia con cada clic es un candidato perfecto a ser stateful. Ver figura 196



Figura 196 Widge para la creación de la interfaz gráfica.

Se declaran las variables de entorno que son de tipo *bolsatrabajobloc* el cual indiga la creación de un nuevo *providerbloc,* posteriormente se ejecuta la instrucción de *cargarbolsadetrabajo* de *bolsatrabajobloc.* Ver figura 197



Figura 197 Declaración de variables de entorno.

Se crearon diferentes widgets para el diseño de la interfaz, pero el que nos interesa es el *\_builderbolsas* el cual manda como parámetro el stream creado en *bolsatrabajobloc.* Ver figura 198



Figura 198 Widget \_builder.

Después se recibe un stream y se pasa en un *streambuilder* donde valida si hay datos, posteriormente verifica si está vacío el widget y si este tuviera datos retorna un widget donde se muestra la información, en caso de no tener nada mostrara un error retornando un widget de error y en lo que esta carga muestra un widget de carga. Ver figura 199

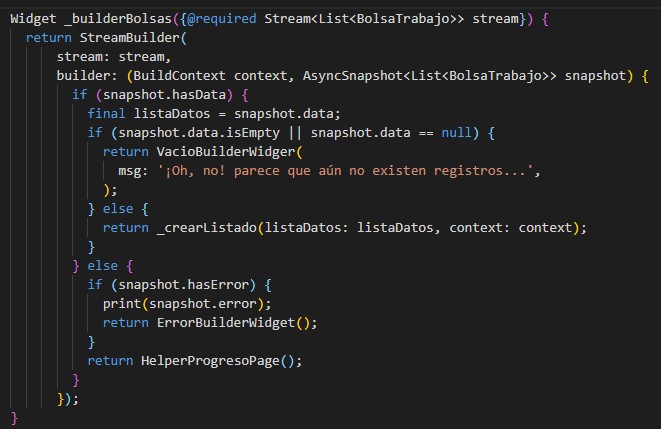


Figura 199 Validación de datos.

Ahora se muestra todas las pages creadas. Ver figura 200

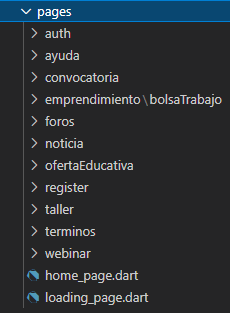


Figura 200 Pages creadas para el desarrollo de la aplicación móvil.

* + - 1. Routes

En este apartado se muestran las rutas de los diferentes módulos.

Llamado de los diferentes módulos usados. Ver figura 201

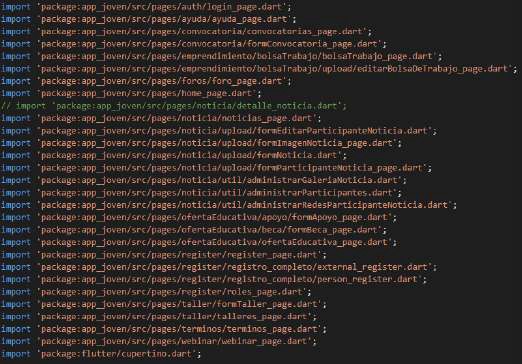


Figura 201 Llamado de los diferentes módulos pages.

Creación de las rutas. Ver figura 202

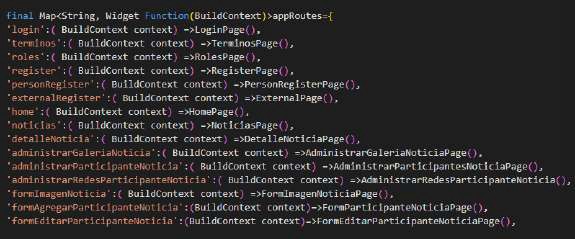


Figura 202 Creación de las rutas.

CAPITULO IV. RESULTADOS

En el presente capitulo se presentan los resultados obtenidos del desarrollo de la aplicación móvil en este caso su interfaz gráfica, todo esto se logró en base a todas las herramientas tecnológicas mencionadas en capítulos anteriores a continuación se presenta el resultado final de la aplicación móvil

4.1 Vistas de la aplicación móvil

Las vistas son las interfaces graficas o pantallas a observar por el usuario al interactuar con la aplicación. A continuación, se muestra el resultado final de la aplicación móvil con todos sus módulos corriendo correctamente. Cabe destacar que algunos módulos el diseño es similar así que solo se mostrara su pantalla principal.

* + 1. Pantalla de inicio de sesión.



Figura 203 Pantalla de inicio de sesión

* + 1. Pantalla de inicio se sesión joven y empresa Selección de usuario para su registro joven o empresa.

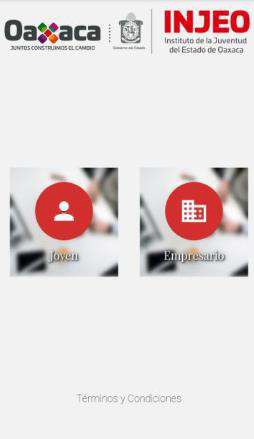


Figura 204 Pantalla de joven y empresa

* + 1. Pantalla de registro Registro de usuarios.



Figura 205 Pantalla de registro 1

Registro de datos de los usuarios.



Figura 206 Pantalla de registro 2

* + 1. Menú principal

Menú principal con todos los módulos completamente funcional.



Figura 207 Pantalla de menú principal

* + 1. Noticias

En la siguiente figura se logra apreciar la estructura principal del módulo de noticias, posteriormente al ingresar se visualizar su estructura interna la cual es similar a otros módulos de la aplicación que más adelante se mostraran.

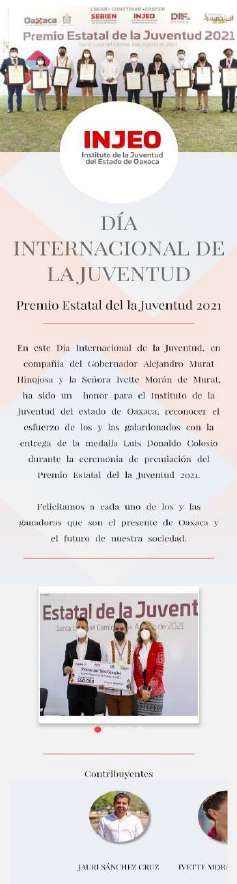


Figura 208 Pantalla de noticias

El apartado de participantes comprende a la estructura interna del módulo de noticias, en este se muestra las personalidades que asistieron al evento.

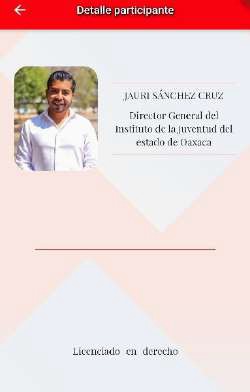


Figura 209 Pantalla de participantes.

* + 1. Talleres.

Estructura de talleres.



Figura 210 Pantalla

de talleres

* + 1. Convocatorias

Estructura principal de convocatorias.



Figura 211 Pantalla de

convocatorias.

* + 1. Oferta educativa.

La estructura de la oferta educativa está conformada por las escuelas, carreras, becas y apoyos.

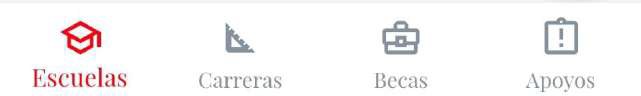


Figura 212 Estructura de la pantalla de oferta educativa

* + - 1. Estructura de escuelas.



Figura 213 Estructura

de la pantalla de escuela

* + - 1. Estructura de carreras



Figura 214 Estructura de la pantalla de carreras.

* + - 1. Estructura de becas



Figura 215 Estructura de la pantalla becas.

* + - 1. Estructura de apoyos



Figura 216 Estructura

de la pantalla de apoyos

* + 1. Bolsa de trabajo

En el siguiente apartado se muestra la estructura principal del módulo de bolsas de trabajo, posteriormente se muestra su estructura interna.

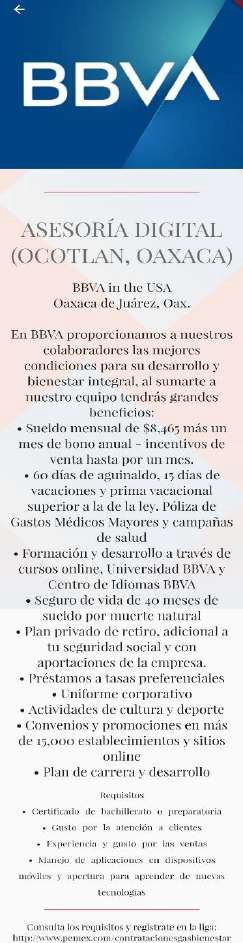


Figura 217 Estructura de la pantalla de bolsa de trabajo.

* + 1. Foros

En este apartado los jóvenes podrán dar su opinión o punto de vista en los diferentes foros que el injeo imparte.



Figura 218 Estructura de la pantalla de foros

* + 1. Webinar

En el siguiente modulo muestra el diseño de las Webinar, al ingresar alguna Webinar mandara a otra ventana en donde mostrara las Webinar.

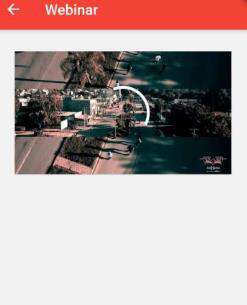


Figura 219 Estructura de la pantalla webinar

CAPITULO V. CONCLUSIONES, RECOMENDACIONES Y EXPERIENCIA ADQUIRIDA.

* 1. Conclusiones

El desarrollo de aplicaciones móviles en estos tiempos ha crecido considerablemente ya que, a mayor consumo de dispositivos móviles, mayor demanda en aplicaciones móviles por ello han aparecido nuevas herramientas de desarrollo de software, para poder cubrir esta demanda añadiendo más componentes y funciones para los desarrolladores.

Tras finalizar con la implementación del proyecto el cual se realizó para los sistemas operativo iOS y Android, se hizo una retrospectiva de la cual se concluye que se han conseguido los objetivos propuestos con la empresa.

A continuación, se detalla los principales resultados obtenidos tras el desarrollo de este proyecto

* Se ha diseñado un API REST totalmente funcional, el cual cumple con los estándares definidos para ser considerado RESTFull, todo esto utilizando los métodos HTTP de manera correcta.
* Se creo una base de datos de datos en MongoDB este proporciona una amplia colección de herramientas para los usuarios cabe recalcar que MongoDB es para tipos de bases no relacionales, pero en nuestro caso se hicieron relaciones para que la aplicación móvil funcionara con mayor fluidez.
* Se implemento el uso de Cloudinary, el cual sirvió para el almacenamiento de las imágenes, ya que esto ayuda a la base de datos a no sobrecargarse de información, haciendo que la aplicación reaccione de manera más rápida.
* En el desarrollo de la aplicación se utilizó Flutter como el framework y Dark como el lenguaje de programación, proporcionando una aplicación funcional e intuitiva para los usuarios, ya que gracias a las opciones de personalización de la interfaz de usuario que flutter ofrece se consigue una experiencia de usuario mucho mejor.

Desde nuestro punto de vista personal, sobre el desarrollo de este proyecto nos ha permitido desarrollarnos profesionalmente, se adquirieron conocimientos en lo que respeta a la gestión de proyectos y la ingeniería de software, durante el transcurso del desarrollo del proyecto se puso en práctica los diferentes conocimientos adquiridos a lo largo de la ingeniería informática.

* 1. Recomendaciones.

Se recomienda seguir modificando o mejorando esta aplicación para que sea más completa ya que al ser una versión inicial siempre existirán pequeñas fallas

Crear encuestas para los usuarios sobre el funcionamiento de la aplicación final, para tomar sus opiniones e ir agregando diferentes mejoras conforme a los criterios de los usuarios.

Se recomienda crear su página web de la aplicación para tener un mayor alcance tanto móvil como web.

Para las posteriores mejoras de la aplicación móvil se recomienda usar un dispositivo físico ya que los emuladores de iOS y Android carecen de funcionalidades, así mismo se evita consumir recursos de procesamiento del computador y poder probar la verdadera funcionalidad de la aplicación en un entorno real.

En lo relacionado a la escuela, se recomienda que dentro de la asignatura desarrollo de aplicaciones móviles, estudiar más sobre las herramientas para desarrollo de aplicaciones móviles multiplataforma.

* 1. Glosario de términos
* **Android**: es un sistema operativo móvil creado por Google, su desarrollo es completamente libre.
* **API´S**: Application Programming Interface, es el conjunto de funciones y procedimientos, que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
* **APK**: es un archivo ejecutable de aplicaciones para Android.
* **BD**: abreviatura de base de dato.
* **CSS**: es un lenguaje de hojas de estilos creada para controlar el aspecto de los documentos electrónicos definidos con html y xhtml.
* **GUI**: abreviatura de interfaz gráfica de usuario.
* **Hardware**: es la parte física de un ordenador o sistema informático.
* **HTML5**: es la quinta versión de HTML, que es un lenguaje de marcado de etiquetas, con el que se puede desarrollar aplicaciones web. La quinta versión posee nuevas y funcionales herramientas y etiquetas que ayudan a generar rápidamente el código.
* **HTTP**: (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web
* **IDE**: es un entorno de desarrollo integrado, donde se puede desarrollar programas.
* **iOS**: iPhone Operating System, es un sistema operativo móvil creado por Apple.
* **Java**: lenguaje de programación.
* **JS**: abreviatura de JavaScript
* **Json**: notación de objetos de javascript (javascript object notation) es un formato ligero de intercambio de datos, el cual resulta sencillo leer para los programadores.
* **Nosql**: abreviatura de base de datos no relacionales
* **Rest**: es un estilo de arquitectura a la hora de realizar una comunicación entre cliente y servidor.
* **Scrum**: es una metodología de desarrollo ágil de software, en el que se interesa plasmar los requerimientos del usuario de manera rápida.
* **SDK**: abreviaturas de kit de desarrollo de software.
* **SGBD**: abreviatura de sistema gestor de base de datos
* **SO**: abreviación de Sistema Operativo
* **Software**: conjunto de programas de cómputo, así como datos, procedimientos y pautas que permiten realizar distintas tareas
* **SQL**: es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.
* **Web**: sistema de gestión de información más popular para la trasmisión de datos a través de internet.
* **XML**: es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.
  1. Experiencia adquirida.

Durante la residencia profesional se llevó un proceso de nuevos aprendizajes, se trabajó de manera autónoma y responsable, aplicando los conocimientos adquiridos aplicando los criterios del buen uso y manejo de las tecnologías todo esto con los principios académicos, teóricos y prácticos recibidos en el Instituto Tecnológico del Valle de Oaxaca.

Se valoró la importancia del trabajo en equipo puesto que este proyecto involucro diferentes personalidades, así mismo permitió enriquecer la formación profesional, ya que logramos desafiar la vida profesional desde una perspectiva más inherente a la práctica.

Un aspecto de gran importancia para nosotros fue la toma de decisiones, ya que estas influyen directamente al proyecto por lo que era necesario conocer el entorno de una decisión, así como sus posibles consecuencias para tomar la mejor decisión, sin no dejar fuera el compromiso y la responsabilidad con la empresa, la universidad y nosotros mismos, ya que son unos elementos muy importantes en el desarrollo de cualquier proyecto.

CAPITULO VI. COMPETENCIAS DESARROLLADAS Y/O APLICADAS

* 1. Competencias generales.

1. Aplicación de lógica de programación.
2. Aprendizaje de técnicas de investigación.
3. Aplicación de conocimiento de base de datos.
4. Aplicación de conocimientos de lenguajes de programación Dark y JavaScript.
   1. Competencias estratégicas.
5. Trabajar de forma autónoma y en equipo.
6. Capacidad de analizar, resumir y sintetizar las fuentes de información.
7. Ser responsable en la ejecución del proyecto.
8. Tener iniciativa, creatividad y liderazgo.
9. Capacidad de organizar, planificar y tomar decisiones.

CAPITULO VII. FUENTES DE INFORMACIÓN.

* 1. Referencias bibliográficas

Brookshear, J. (2007). Computer Science, an Overview. 9th ed. New York: Pearson Addison Wesley [https://computarium.lcd.lu/library/PDF/BROOKSHEAR\_COMPUTER\_SCI](https://computarium.lcd.lu/library/PDF/BROOKSHEAR_COMPUTER_SCIENCE_9th_edition_2007.pdf) [ENCE\_9th\_edition\_2007.pdf](https://computarium.lcd.lu/library/PDF/BROOKSHEAR_COMPUTER_SCIENCE_9th_edition_2007.pdf)

Cabero, J. (2005) Cibersociedad y juventud: la cara oculta (buena) de la Luna, en AGUIAR, M.V. y FARRAY, J.I. (2005): Un nuevo sujeto para la sociedad de la información. A Coruña, Netbjblo, 13-42. Disponible en <http://tecnologiaedu.us.es/bibliovir/pdf/ciberjuve.pdf>

Cajilima, J. (2016). Desarrollo de una aplicación, para dispositivos móviles que permita administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa: “Almacenes Juan ELJURI CIA. LTDA.”. Tesis de licenciatura. Universidad Politécnica Salesiana Ecuador [https://dspace.ups.edu.ec/bitstream/123456789/7951/1/UPS-](https://dspace.ups.edu.ec/bitstream/123456789/7951/1/UPS-CT004811.pdf) [CT004811.pdf](https://dspace.ups.edu.ec/bitstream/123456789/7951/1/UPS-CT004811.pdf)

Coria, R., Pérez M., Mendoza E., y Martinez R. (2017). Brecha digital y pobreza digital en el estado de Oaxaca. Conciencia Tecnológica No. 42.

Cuadra, E. (1996). Internet: Conceptos Básicos: Cuadernos de Documentación Multimedia. Recuperado el 28 de marzo de 2021, de [http://www.dirinfo.unsl.edu.ar/profesorado/INfyEduc/teorias/Clase1Interne](http://www.dirinfo.unsl.edu.ar/profesorado/INfyEduc/teorias/Clase1Internet2008.pdf) [t2008.pdf](http://www.dirinfo.unsl.edu.ar/profesorado/INfyEduc/teorias/Clase1Internet2008.pdf)

D. E. Avison y G. Fitzgerald. (2006). Information system development. Maidenhead: McGraw-Hill Education.

Delia, L., Galdamez, N., Thomas, P., Corbalan, L. & Pesado, P. (2015). Análisis Experimental de desarrollo de Aplicaciones Móviles Multiplataforma. Recuperado el 1 de abril de 2021 de [http://sedici.unlp.edu.ar/bitstream/handle/10915/42355/Documento\_comp](http://sedici.unlp.edu.ar/bitstream/handle/10915/42355/Documento_completo.pdf?sequence=1) [leto.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/42355/Documento_completo.pdf?sequence=1)

Garita, R. (2013). Tecnología Móvil: desarrollo de sistemas y aplicaciones para las Unidades de Información. E-Ciencias de la Información. Recuperado el 29 de marzo de 2021, de <https://core.ac.uk/download/pdf/16292287.pdf>

Gasca, M., Camargo, L. & Medina, R. (2013). Metodología para el desarrollo de aplicaciones móviles. Tecnura, vol. 18, núm. 40, abril-junio 2014, pp.20-35

Universidad Distrital Francisco José de Caldas Bogotá, Colombia. Recuperado el 1 de abril de 2021 de <https://www.redalyc.org/pdf/2570/257030546003.pdf>

Gómez, M., (2013). Notas del Curso “Bases de datos “. (1° ed.) Publidisa Mexicana S.A de C.V. Recuperado el 30 de abril de 2021 de [http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas\_del\_curso\_Bases\_](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_del_curso_Bases_de_Datos.pdf) [de\_Datos.pdf](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_del_curso_Bases_de_Datos.pdf)

Gutiérrez, J. (2014). ¿Qué es un framework web? Recuperado el 3 de abril de 2021 de

<http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf>

Herranz, R. (2014). Bases de datos NoSQL: Arquitectura y Ejemplos de Aplicación. Trabajo de Tesis. Recuperado el 2 de mayo de 2021 de <https://core.ac.uk/download/pdf/44310803.pdf>

Larman, C. (2003). Agile & iterative development: a manager´s guide. Addison- Wesley, Boston.

Navarro, A. Fernández, J. y Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software PROSPECTIVA, vol. 11, núm. 2, julio- diciembre, 2013, pp. 30-39 Universidad Autónoma del Caribe Recuperado el 3 de abril de <https://www.redalyc.org/pdf/4962/496250736004.pdf>

Quintas, L. (2008). Sistemas de Información Geográfica y Teledetección. Recuperado el 2 de mayo de 2021 de <https://static.eoi.es/savia/documents/componente45499.pdf>

Ramos, J., Ramos, A. & Montero, F. (2008). Sistemas gestores de bases de datos. (1° ed.). McGraw Hill. Recuperado el 29 de abril de 2021 de <https://www.mheducation.es/bcv/guide/capitulo/8448148797.pdf>

Ravulavaru, A. (2017). Learning Ionic 2. 2nd ed. Packt Publishing.

Riyadi, A. (2020). Information System for Providing Food Services Based on Mobile Application Using Flutter Framework. [https://www.atlantis-](https://www.atlantis-press.com/article/125952565.pdf) [press.com/article/125952565.pdf](https://www.atlantis-press.com/article/125952565.pdf)

Sarasola, J. (2019). Design and Implementation of a Mobile Solution for Searching Video Game Offers. Tesis de Licenciatura. Universidad Complutense de Madrid. Recuperado de <https://eprints.ucm.es/id/eprint/57084/1/JoseMiguelSarasola.pdf>

Schwab, K., y Sutherland, J. (2017). La Guía de Scrum (Scrum ORG, Ed.). Recuperado el 3 de abril de 2021 de [https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-](https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf) [Spanish-SouthAmerican.pdf](https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf)

Silberschatz, A., Korth, H. & Sudarshan, S. (2002). Fundamentos de base de datos. (4° ed.) Madrid, España: McGraw Hill Book Co. Recuperado el 29 de abril de 2021 de <http://mirrors.sandino.net/pelogo/FundaBD/FundaBD_Silverschatz.pdf>

Trigas, M. (2012). Metodología Scrum. Recuperado el 4 de abril de 2021 de: [http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigas](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf) [TFC0612memoria.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf)

Villota, V. (2015). Estudio De La Herramienta Ibm Worklight Studio Para El Desarrollo De Aplicaciones Móviles Multiplataforma Con Aplicativo De Consulta De Notas, Libros Y Sílabos En El Instituto Tecnológico Superior “José Chiriboga Grijalva”. Tesis de licenciatura. Universidad Técnica del Norte. Recuperado el 31 de marzo de 2021 de [http://repositorio.utn.edu.ec/bitstream/123456789/4653/1/04%20ISC%20](http://repositorio.utn.edu.ec/bitstream/123456789/4653/1/04%20ISC%20326%20TESIS.pdf) [326%20TESIS.pdf](http://repositorio.utn.edu.ec/bitstream/123456789/4653/1/04%20ISC%20326%20TESIS.pdf)

Zazo, C. (2019). Migración de aplicaciones Android hacia Flutter, un framework para desarrollo de apps multiplataforma. Tesis de Licenciatura. Universidad Politécnica de València. Recuperado el 3 de abril de 2021 de [https://riunet.upv.es/bitstream/handle/10251/128486/Zazo%20-](https://riunet.upv.es/bitstream/handle/10251/128486/Zazo%20-%20Migraci%C3%B3n%20de%20aplicaciones%20Android%20hacia%20Flutter%2C%20un%20framework%20para%20desarrollo%20de%20apps%20mult....pdf?sequence=1&isAllowed=y)

[%20Migraci%C3%B3n%20de%20aplicaciones%20Android%20hacia%20](https://riunet.upv.es/bitstream/handle/10251/128486/Zazo%20-%20Migraci%C3%B3n%20de%20aplicaciones%20Android%20hacia%20Flutter%2C%20un%20framework%20para%20desarrollo%20de%20apps%20mult....pdf?sequence=1&isAllowed=y) [Flutter%2C%20un%20framework%20para%20desarrollo%20de%20apps](https://riunet.upv.es/bitstream/handle/10251/128486/Zazo%20-%20Migraci%C3%B3n%20de%20aplicaciones%20Android%20hacia%20Flutter%2C%20un%20framework%20para%20desarrollo%20de%20apps%20mult....pdf?sequence=1&isAllowed=y)

[%20mult. pdf?sequence=1&isAllowed=y](https://riunet.upv.es/bitstream/handle/10251/128486/Zazo%20-%20Migraci%C3%B3n%20de%20aplicaciones%20Android%20hacia%20Flutter%2C%20un%20framework%20para%20desarrollo%20de%20apps%20mult....pdf?sequence=1&isAllowed=y)

* 1. Bibliográficas virtuales

Adobe. (2014). Adobe. Obtenido de Adobe: [https://helpx.adobe.com/es/dreamweaver/user-](https://helpx.adobe.com/es/dreamweaver/user-guide.html/es/dreamweaver/using/web-applications.ug.html) [guide.html/es/dreamweaver/using/web-applications.ug.html](https://helpx.adobe.com/es/dreamweaver/user-guide.html/es/dreamweaver/using/web-applications.ug.html)

Acens Technologies (2014). Framework para el desarrollo ágil de aplicaciones. Recuperado el 3 de abril de 2021 de [https://www.acens.com/wp-](https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf) [content/images/2014/03/frameworks-white-paper-acens-.pdf](https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf)

Belloch, C. (2008). Las Tecnologías de la Información y Comunicación en el aprendizaje: Recuperado el 24 de marzo de 2021, de [04 LAS TIC.pdf](http://eduteka.icesi.edu.co/gp/upload/04%20LAS%20TIC.pdf) [(icesi.edu.co)](http://eduteka.icesi.edu.co/gp/upload/04%20LAS%20TIC.pdf)

Beyond Mobile: Material Design, Adaptable UIs, and Flutter (Google I/O'19). (consultado el 3 de abril de 2021). Obtenido de YouTube: <https://www.youtube.com/watch?v=YSULAJf6R6M>

Carrera, L. (2021, 12 abril). MySQL: ¿cómo funciona este sistema de gestión de base de datos? TIC Portal. Recuperado el 2 de mayo de 2021 <https://www.ticportal.es/glosario-tic/mysql>

Cuervo, P. V. (2019, 14 febrero). ¿Qué es Postman? Arquitecto IT. Recuperado el 2 de mayo de 2021 de [https://www.arquitectoit.com/postman/que-es-](https://www.arquitectoit.com/postman/que-es-postman/) [postman/](https://www.arquitectoit.com/postman/que-es-postman/)

Cutipa, G. (2020, 20 abril). Las 5 principales ventajas y desventajas de bases de datos relacionales y no relacionales: NoSQL vs SQL. Recuperado el 2 de mayo de 2021 de [https://guidocutipa.blog.bo/principales-ventajas-](https://guidocutipa.blog.bo/principales-ventajas-desventajas-bases-de-datos-relacionales-no-relacionales-nosql-vs-sql/) [desventajas-bases-de-datos-relacionales-no-relacionales-nosql-vs-sql/](https://guidocutipa.blog.bo/principales-ventajas-desventajas-bases-de-datos-relacionales-no-relacionales-nosql-vs-sql/)

Delgado, M. (2013). Aplicaciones híbridas: presente y futuro del desarrollo Móvil. Obtenido de [https://docplayer.es/781287-Aplicaciones-hibridas-presente-](https://docplayer.es/781287-Aplicaciones-hibridas-presente-y-futuro-del-desarrollo-movil.html) [y-futuro-del-desarrollo-movil.html](https://docplayer.es/781287-Aplicaciones-hibridas-presente-y-futuro-del-desarrollo-movil.html)

EAE Business School. (2021). Cuatro sistemas operativos que debes conocer. EAE Programas.

[https://www.eaeprogramas.es/blog/negocio/tecnologia/cuatro-sistemas-](https://www.eaeprogramas.es/blog/negocio/tecnologia/cuatro-sistemas-operativos-que-debes-conocer) [operativos-que-debes-conocer](https://www.eaeprogramas.es/blog/negocio/tecnologia/cuatro-sistemas-operativos-que-debes-conocer)

En Oaxaca rompen brecha tecnológica. (2020, 10 octubre). Excélsior. [https://www.excelsior.com.mx/nacional/en-oaxaca-rompen-brecha-](https://www.excelsior.com.mx/nacional/en-oaxaca-rompen-brecha-tecnologica/1410431) [tecnologica/1410431](https://www.excelsior.com.mx/nacional/en-oaxaca-rompen-brecha-tecnologica/1410431)

Evkoski, B. (2018, 9 octubre). *React Native: What it is and how it works - We Talk IT*. Medium. [https://medium.com/we-talk-it/react-native-what-it-is-and-](https://medium.com/we-talk-it/react-native-what-it-is-and-how-it-works-e2182d008f5e) [how-it-works-e2182d008f5e](https://medium.com/we-talk-it/react-native-what-it-is-and-how-it-works-e2182d008f5e)

Fernández, J. S. (2020, 18 junio). *Introducción al patrón BLoC*. Xurxodev | Desarrollador Android y Flutter Freelance. <http://xurxodev.com/introduccion-al-patron-bloc/>

Filgueira, J. M. (2019). *¿Qué es iOS? | Gabit*. J.M. Filgueira. http://www.gabit.org/gabit/?q=es/que-es-ios

Flutter documentation, Introduction to widgets [Online]. Recuperado el 3 de abril de 2021 de: <https://flutter.dev/docs/development/ui/widgets-intro>

Gómez, P. M. (2020, 10 agosto). Características de MongoDB. OpenWebinars.net. Recuperado el 2 de mayo de 2021 de <https://openwebinars.net/blog/caracteristicas-de-mongodb/>

[https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202009-](https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202009-202102-bar) [202102-bar](https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202009-202102-bar)

inLab (2020). ¿Qué es el lenguaje de programación Dart? (2020, 30 octubre). Obtenido de: [https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-](https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart) [programacion-dart](https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart)

Lafuente, A. (2018, 12 septiembre). Bases de datos relacionales vs. no relacionales: ¿qué es mejor? Aukera. Recuperado el 1 de mayo de 2021 de <https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/>

LanceTalent. (20 de febrero de 2014). LanceTalent. Obtenido de LanceTalent: [http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-](http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajasinconvenientes/) [ventajasinconvenientes/](http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajasinconvenientes/)

Leandro, A. (2013). DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA.

Recuperado el 1 de abril de 2021 de <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>

López, A. (2021, 26 febrero). Qué es Postman y para qué sirve. OpenWebinars.net. Recuperado el 2 de mayo de 2021 <https://openwebinars.net/blog/que-es-postman/>

Lucas, J. (2020, 1 junio). Qué es NodeJS y para qué sirve. OpenWebinars.net. <https://openwebinars.net/blog/que-es-nodejs/>

Maleve, K. y Beauperthuy, J (2011). “Android” Google´s Operating System For Moibile Devices. Recuperado el 31 de marzo de 2021 de <https://dialnet.unirioja.es/servlet/articulo?codigo=7165367>

Moncayo, J. M. R. (2020, 25 junio). ¿Qué es REST? Conoce su potencia. OpenWebinars.net. [https://openwebinars.net/blog/que-es-rest-conoce-su-](https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/#%3A%7E%3Atext%3DREST%20es%20una%20interfaz%20para%2Cespec%C3%ADficos%2C%20como%20XML%20y%20JSON) [potencia/#:%7E:text=REST%20es%20una%20interfaz%20para,espec%](https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/#%3A%7E%3Atext%3DREST%20es%20una%20interfaz%20para%2Cespec%C3%ADficos%2C%20como%20XML%20y%20JSON) [C3%ADficos%2C%20como%20XML%20y%20JSON](https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/#%3A%7E%3Atext%3DREST%20es%20una%20interfaz%20para%2Cespec%C3%ADficos%2C%20como%20XML%20y%20JSON).

MongoDB. (2021). La base de datos líder del mercado para aplicaciones modernas. Recuperado el 2 de mayo de 2021 de <https://www.mongodb.com/es>

Ram, S. (2019, 28 mayo). *FLUTTER: PROVIDER PARA PRINCIPIANTES -GUIA*

*DE INICIO*. Medium. [https://medium.com/@sarlmnt.soul/flutter-provider-](https://medium.com/%40sarlmnt.soul/flutter-provider-para-principiantes-guia-de-inicio-f304c1318d5e) [para-principiantes-guia-de-inicio-f304c1318d5e](https://medium.com/%40sarlmnt.soul/flutter-provider-para-principiantes-guia-de-inicio-f304c1318d5e)

Redondo, F. (2021, 15 marzo). Postman: gestiona y construye tus APIs rápidamente. Paradigma. Recuperado el 2 de mayo de 2021 de [https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-](https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/) [apis-rapidamente/](https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/)

Robles, C. C. (2019b, octubre 8). Comparación bases de datos relacionales y no relacionales. Medium. Recuperado el 2 de mayo de 2021 de [https://medium.com/200-response/comparaci%C3%B3n-bases-de-datos-](https://medium.com/200-response/comparaci%C3%B3n-bases-de-datos-relacionales-y-no-relacionales-424dd83dd693) [relacionales-y-no-relacionales-424dd83dd693](https://medium.com/200-response/comparaci%C3%B3n-bases-de-datos-relacionales-y-no-relacionales-424dd83dd693)

Spyros, X., Stelios, X. (2013). A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. Recuperado el 1 de abril de 2021 de [https://www.researchgate.net/publication/258010031\_A\_Comparative\_An](https://www.researchgate.net/publication/258010031_A_Comparative_Analysis_of_Cross-platform_Development_Approaches_for_Mobile_Applications) [alysis\_of\_Cross-](https://www.researchgate.net/publication/258010031_A_Comparative_Analysis_of_Cross-platform_Development_Approaches_for_Mobile_Applications) [platform\_Development\_Approaches\_for\_Mobile\_Applications](https://www.researchgate.net/publication/258010031_A_Comparative_Analysis_of_Cross-platform_Development_Approaches_for_Mobile_Applications)

Staff, F. (2019, 16 enero). Los adolescentes mexicanos pasan 12 horas al día en sus celulares. Forbes México. [https://www.forbes.com.mx/los-](https://www.forbes.com.mx/los-adolescentes-mexicanos-pasan-12-horas-al-dia-en-sus-celulares/) [adolescentes-mexicanos-pasan-12-horas-al-dia-en-sus-celulares/](https://www.forbes.com.mx/los-adolescentes-mexicanos-pasan-12-horas-al-dia-en-sus-celulares/)

Tablado, F. (2020, 10 septiembre). Base de datos no relacional. ¿Qué es? Características y ejemplos. Ayuda Ley Protección Datos (LOPDGDD). Recuperado el 2 de mayo de 2021 de <https://ayudaleyprotecciondatos.es/bases-de-datos/no-relacional/>

Tello, E. (2011). Las tecnologías de la información y comunicaciones (TIC) y la brecha digital: su impacto en la sociedad de México. Rev. RUSC, 4(2). Recuperado el 28 de marzo de 2021, de [https://rusc.uoc.edu/rusc/es/index.php/rusc/article/view/v4n2-tello/v4n2-](https://rusc.uoc.edu/rusc/es/index.php/rusc/article/view/v4n2-tello/v4n2-tello.html) [tello.html](https://rusc.uoc.edu/rusc/es/index.php/rusc/article/view/v4n2-tello/v4n2-tello.html)

Vazquez, V. (2019). Desarrollo de aplicaciones móviles multiplataforma con Flutter. Tesis de Licenciatura. Universidad de Almeria. Recuperado el 3 de abril de 2021 de [http://repositorio.ual.es/bitstream/handle/10835/8010/TFG\_VAZQUEZ%2](http://repositorio.ual.es/bitstream/handle/10835/8010/TFG_VAZQUEZ%20RODRIGUEZ%2C%20VICTOR.pdf?sequence=1&isAllowed=y) [0RODRIGUEZ%2C%20VICTOR.pdf?sequence=1&isAllowed=y](http://repositorio.ual.es/bitstream/handle/10835/8010/TFG_VAZQUEZ%20RODRIGUEZ%2C%20VICTOR.pdf?sequence=1&isAllowed=y)

Velasteguí, E. y López, B. (2018). El avance en la tecnología móvil y su impacto en la sociedad. Explorador Digital. Recuperado el 29 de marzo de 2021 de <https://doi.org/10.33262/exploradordigital.v2i4.337>

Vico, A (2012). *Arquitectura de Android*. La columna 80. <https://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>

Villareal, V. (2017). Sistemas Operativos. Recuperado el 30 de marzo de 2021 de [https://ridda2.utp.ac.pa/bitstream/handle/123456789/5074/folleto\_sistema](https://ridda2.utp.ac.pa/bitstream/handle/123456789/5074/folleto_sistemas_operativos.pdf?sequence=3&isAllowed=y) [s\_operativos.pdf?sequence=3&isAllowed=y](https://ridda2.utp.ac.pa/bitstream/handle/123456789/5074/folleto_sistemas_operativos.pdf?sequence=3&isAllowed=y)

Viviana, V. (2018, 30 octubre). MySQL vs MongoDB –. Guiadev. Recuperado el

2 de mayo de 2021 de [https://guiadev.com/mysql-vs- mongodb/#Caracteristicas\_de\_MySQL](https://guiadev.com/mysql-vs-mongodb/#Caracteristicas_de_MySQL)

[https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/OtrTemEcon/](https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/OtrTemEcon/ENDUTIH_2020.pdf) [ENDUTIH\_2020.pdf](https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/OtrTemEcon/ENDUTIH_2020.pdf)

<https://developer.android.com/studio/intro?hl=es-419> <https://material.io/design/introduction#components> <https://developer.apple.com/es/support/xcode/>

CAPITULO VIII. ANEXOS.

8.1 Manual de usuario

* + 1. Configuración del Restfull server
       1. Para la configuración del Restfull en una maquina nueva es necesario crear una carpeta nueva, esta se creará en cualquier parte del equipo.



Figura 220 Creación de las carpetas para el restfull server

* + - 1. Esa carpeta se abrirá con Visual Studio Code.

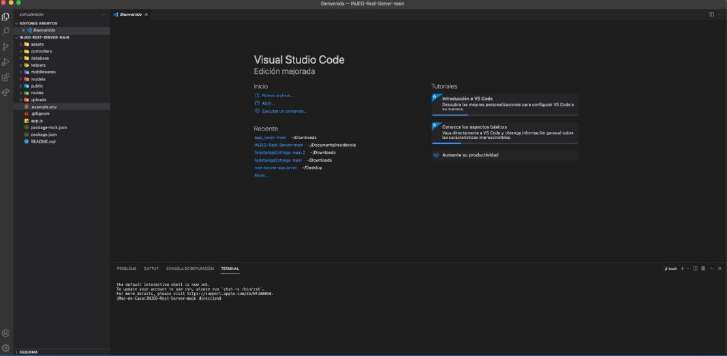


Figura 221 Configuración del restfull server

* + - 1. Se procederá abrir una terminal en Visual Studio Code en la cual se ingresará el comando *npm install* o bien abrir el archivo README.md.

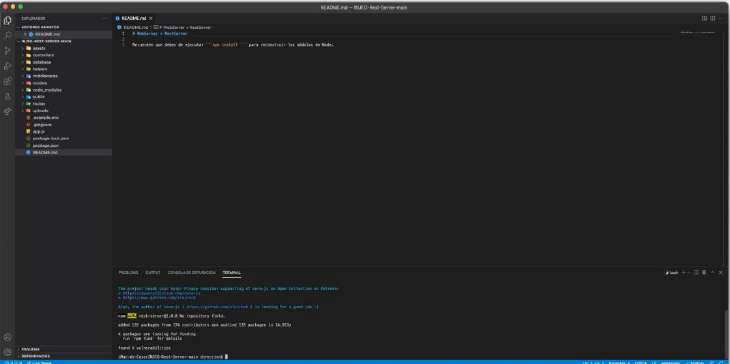


Figura 222 Configuración de npm

* + - 1. Se crea un archivo. env en el cual se pondrá el puerto, la dirección de la base de datos, el tipo de encriptación, así como la dirección de cloduinary.

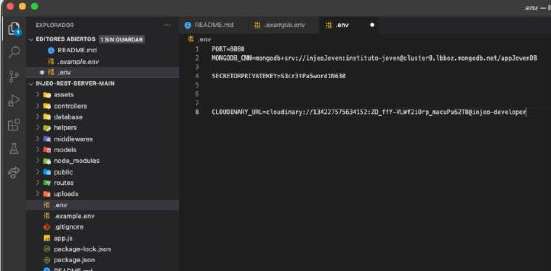


Figura 223 Creación del archivo .env

* + - 1. En la terminal de Visual Studio Code se ingresa el comando nodemon app para inicializar el servidor.



Figura 224 Inicialización del servidor.

* + - 1. Para verificar que este corriendo correctamente se abre el navegador en este caso Chrome y se ingresa a la siguiente dirección *localhost:8080/api/usuarios.*

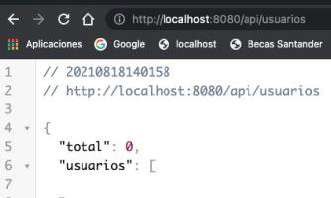


Figura 225 Comprobación que el servidor este corriendo correctamente

* + - 1. Se ingresa a la página de Heroku<https://devcenter.heroku.com/articles/heroku-cli>en la cual se encuentran las instrucciones para su respectiva instalación en este caso la instalación se hizo en mac, para esto se abre una terminal y se copia el comando, con esto quedara instalado Heroku en nuestro equipo.



Figura 226 Instalación de heroku.

* + - 1. En la misma terminal se iniciará Heroku con el comando *heroku login.*



Figura 227 Inicio de sesión en Heroku 2

* + - 1. Se abrirá una ventana del navegador en el cual nos pedirá el correo y contraseña.



Figura 228 Inicio de sesión en Heroku 2

* + - 1. En la terminal de Visual Studio Code se ingresa el comando de *Heroku git:remote -a example-injeo.*



Figura 229 Creación del restfull server en Heroku

* + - 1. En la misma terminal se agregarán las configuraciones el archivo .env con el comando *Heroku config:set “nombre”.* En la figura se muestra el procedimiento.

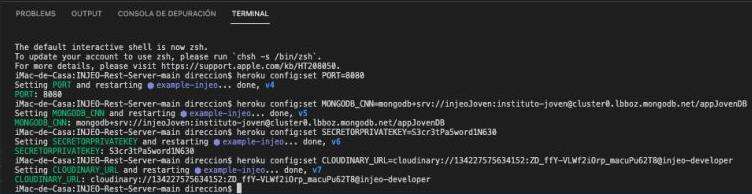


Figura 230 Configuración de las variables del archivo .env

* + - 1. Para verificar que todo salió correcto en el navegador se ingresa la dirección “*nombredelproyecto*”.herokuapp.com/api/usuarios.

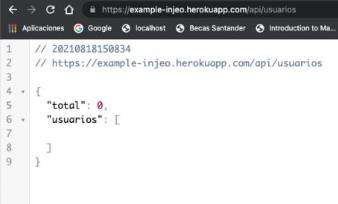


Figura 231 Comprobación del proyecto en heroku.

* + 1. Agregar administradores
       1. Ahora se exportará el CRUD de los diferentes módulos para ello se inicializará Postman y posteriormente se importará el archivo.

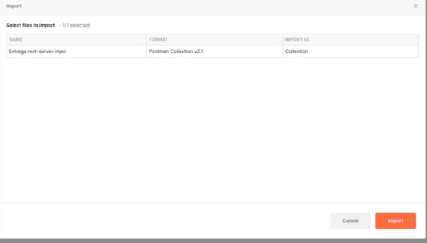


Figura 232 Importación de los CRUD.

* + - 1. Una vez abierto los diferentes CRUD nos enfocaremos en el módulo de usuarios el cual servirá para agregar nuevos administradores



Figura 233

Registro de un

usuario nuevo.

* + - 1. En el área de trabajo mostrara los datos a rellenar cabe destacar que en tipo de rol existen tres: ADMIN\_ROLE, USER\_ROLE y EXTERNO\_ROLE

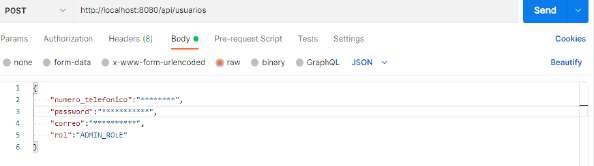


Figura 234 Datos a registrar de un usuario.

* + - 1. Al a completar los datos en la terminal mostrara los datos del registro.



Figura 235 Datos registrados de un usuario nuevo mostrado en la terminal.

* + - 1. Posteriormente es necesario a completar el registro del usuario para esto se ingresará al CRUD de persona en el cual con el id dado anteriormente se completará los datos de usuario y con esto se termina la configuración de un usuario administrador nuevo.



Figura 236 Finalizar registro de datos de un usuario nuevo.

* + 1. Agregar datos
       1. Para agregar datos a la aplicación en todos los módulos aparecerá un menú flotante en la parte inferior derecha de la pantalla, la función de este es agregar la información.



Figura 237 Interfaz gráfica del módulo de noticias

* + - 1. Mandara a otra pantalla en donde mostrara un formulario en el cual se ingresarán los datos, en este caso se tomó de ejemplo el módulo de noticias, cabe mencionar que este formulario se utilizó para los demás módulos solo se cambiaron los datos a rellenar.



Figura 238 Interfaz gráfica para registro nuevo de una noticia

* + 1. Editar datos
       1. Para la edición de datos es necesario ingresar a la noticia en este caso, en la parte superior derecha aparecerá un menú el cual nos mostrará diferentes opciones.



Figura 239 Menús de la aplicacion.

* + - 1. Se ingresa a la opción de editar y mostrará una pantalla en la cual se podrá cambiar los datos, así como su imagen.



Figura 240 Interfaz gráfica para editar

* + 1. Eliminar datos

Para la eliminación de datos es muy sencillo solo se dará clic en la opción de eliminar y la información se eliminará.



Figura 241 Menú

eliminar.

* + 1. Administrar galería
       1. Esta opción ayudar a subir y eliminar todas las imágenes que se suban, en este caso en el módulo de noticas se subieron diferentes imágenes.



Figura 242 Galería de imágenes.

* + - 1. Mostrará todas las imágenes que se subieron, dentro de este apartado se pueden agregar más imágenes, así como eliminar.



Figura 243 Interfaz gráfica para agregar imágenes.

* + - 1. Para agregar más imágenes solo se da clic en el icono rojo y lanzará una ventana en la cual se podrá agregar más imágenes o tomar fotografías desde su dispositivo.

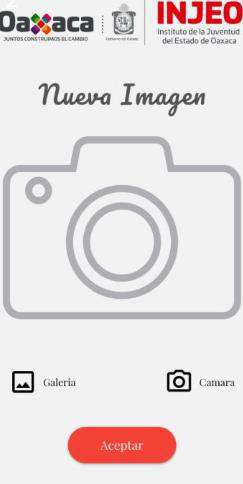


Figura 244 Interfaz gráfica para seleccionar imágenes.

* + - 1. Para eliminar las imágenes no deseadas solo se deslizarán a la parte derecha de la pantalla y estas se eliminarán.



Figura 245 Interfaz gráfica para eliminar imágenes.

* + 1. Administrar participantes
       1. En este apartado se agregan los participantes en algún evento realizado. Se mostrará una pantalla con todos los participantes en la cual se podrán agregar y eliminar al igual que las imágenes.



Figura 246 Interfaz gráfica para administrar los

contribuyentes.

* + - 1. Apartado para agregar nuevos participantes



Figura 247 Interfaz gráfica para agregar nuevos contribuyentes

* + - 1. Para editar los participantes se dará clic sobre el participante a editar en el cual nos aparecerán tres opciones editar, eliminar y administrar redes este último sirve para agregar sus redes sociales de los participantes.



Figura 248 Interfaz gráfica para editar los contribuyentes

Cabe mencionar que la estructura que se tomó de ejemplo de agregar, editar y eliminar es la mismas para todos los módulos de la aplicación móvil, así que ya no es necesario explicar el mismo procedimiento.