



CARRERA DE ANÁLISIS DE SISTEMAS

TEMA:

SISTEMA PARA LA GESTIÓN DE PROCESOS EN EL ÁREA DE HOSPITALIZACIÓN DE LA CLÍNICA VETERINARIA FERNÁNDEZ DE CÓRDOVA.

AUTOR:

JUAN FERNANDO FERNÁNDEZ DE CÓRDOVA AMBROSI

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

TECNÓLOGO EN ANÁLISIS DE SISTEMAS

TUTOR:

MGS. JUAN PABLO HURTADO ORTIZ

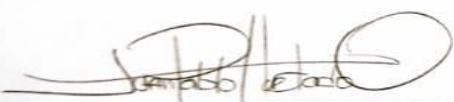
CUENCA – ECUADOR, 2019

CARRERA DE ANALISIS DE SISTEMAS
COMITÉ TÉCNICO MULTIDISCIPLINARIO
Certificación de Aprobación del Trabajo de Titulación

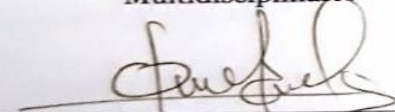
Damos fe que el trabajo desarrollado por el/la estudiante: **FERNÁNDEZ DE CÓRDOVA AMBROSI JUAN FERNANDO** con el título: “**SISTEMA PARA LA GESTIÓN DE PROCESOS EN EL ÁREA DE HOSPITALIZACIÓN DE LA CLÍNICA VETERINARIA FERNÁNDEZ DE CÓRDOVA.**” cumple con las exigencias metodológicas y técnicas.

Por lo antes mencionado, los TUTORES asignados del **COMITÉ TÉCNICO MULTIDISCIPLINARIO** resuelve **APROBAR** el Trabajo de Titulación.

Atentamente,



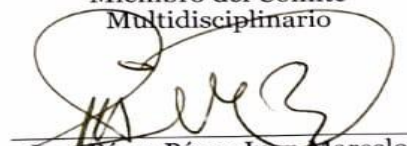
Msg. Hurtado Ortiz Juan Pablo
Miembro del Comité
Multidisciplinario



Ing. Marco Aurelio Guamán
Buestan
Miembro del Comité
Multidisciplinario



Ing. Zúñiga López Max Renato
Miembro del Comité
Multidisciplinario



Ing. Pérez Pérez Juan Marcelo
Miembro del Comité
Multidisciplinario

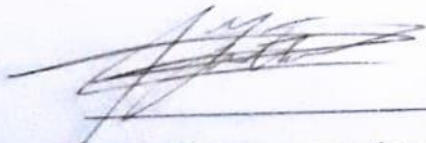


DECLARACIÓN DE AUTORÍA DEL TRABAJO

Yo, **FERNÁNDEZ DE CÓRDOVA AMBROSI JUAN FERNANDO**, estudiante del **Instituto Tecnológico Superior Particular Sudamericano** de la ciudad de Cuenca - Ecuador, que cursó la Tecnología en **ANÁLISIS DE SISTEMAS**, declaro en forma libre y voluntaria que la presente investigación que versa sobre **“SISTEMA PARA LA GESTIÓN DE PROCESOS EN EL ÁREA DE HOSPITALIZACIÓN DE LA CLÍNICA VETERINARIA FERNÁNDEZ DE CÓRDOVA.”** así como las expresiones vertidas en la misma, son autoría de la compareciente, quien ha realizado en base a recopilación bibliográfica, consultas de internet y consultas de campo.

En consecuencia, asumo la responsabilidad de la originalidad de la misma y el cuidado al remitirme a las fuentes bibliográficas respectivas para fundamentar el contenido expuesto.

Atentamente,



FERNÁNDEZ DE CÓRDOVA AMBROSI JUAN FERNANDO

Cédula: 0105230957



DERECHOS DE AUTOR

Los derechos de esta obra son irrenunciables y corresponden a su **AUTOR**, incluido sus derechos patrimoniales. El **Instituto Tecnológico Superior Particular Sudamericano** tiene licencia gratuita e intransferible sobre esta obra para uso no comercial, de necesitar uso comercial requiere autorización de su titular.

RESUMEN

La Clínica Veterinaria Fernández de Córdova de Cuenca Ecuador, maneja entre varias áreas médicas, el área de hospitalización brindando así un servicio previo y posterior a cualquier circunstancia de salud que puedan tener las mascotas. Pero a su vez implica el manejo de una gran cantidad de información debido a los requerimientos para una correcta hospitalización y manejo de los pacientes que se encuentren en esta área de la clínica, por lo que existe la necesidad de automatizar dichas funciones.

Con el diseño e implementación de una aplicación móvil mejoramos la accesibilidad a la información de la Clínica Veterinaria Fernández de Córdova, consiguiendo ahorro de tiempo de respuesta, un medio de verificación confiable de los medicamentos, y mayor facilidad al momento de consultar los datos del paciente hospitalizado.

ABSTRACT

The Fernández de Córdova Veterinary Clinic in Cuenca Ecuador, manages the hospitalization area between several medical areas, thus providing a service before and after any health circumstance that pets may have. But this in turn implies the management of a large amount of information due to the requirements for proper hospitalization and management of patients in this area of the clinic, so there is a need to automate these functions.

With the design and implementation of a mobile application, we improve the accessibility to the information of the Fernández de Córdova Veterinary Clinic, saving response time, a reliable means of verification of medications, and greater ease when consulting patient data hospitalized.

PALABRAS CLAVE

Diagnóstico, Medicación, Aplicaciones Híbridas, web, hospitalización, veterinaria

KEY WORDS

Diagnosis, Medication, Hybrid Applications, web, hospitalization, veterinary

DEDICATORIA

Este trabajo está dedicado a toda mi familia por su completo apoyo, especialmente dedicada a mis abuelos pues su asombro por la tecnología, me llevan a la búsqueda de nuevos conocimientos, a mi abuelo Eduardo quien, a pesar de hoy ya no estar con nosotros, siempre se preocupó por mis estudios.

AGRADECIMIENTO:

A mis padres por todo su apoyo a mis abuelos por su preocupación y a mi novia Estefanía por su cariño y ayuda, por último, agradezco profundamente a mi tutor Juan Pablo Hurtado por todo su apoyo y paciencia.

ÍNDICE DE CONTENIDO

RESUMEN.....	V
ABSTRACT.....	VI
PALABRAS CLAVE.....	VII
KEY WORDS	VII
DEDICATORIA	VIII
ÍNDICE DE CONTENIDO.....	IX
ÍNDICE DE ILUSTRACIONES.....	XII
INTRODUCCIÓN	1
OBJETIVOS	2
Objetivo general.....	2
Objetivos específicos	2
Preguntas de investigación.....	3
Justificación.....	4
CAPÍTULO I.....	5
1.1 Problema: necesidades	5
CAPÍTULO II	6
1.1 Marco Referencial	6
1.1.1 Marco Teórico.....	6
1.1.2 Marco Conceptual	10
CAPÍTULO III.....	13

	X
2.1 Investigación Previa al Desarrollo	13
2.2 Investigación	13
2.3 Recursos Financieros	15
2.4 Diagramación	15
CAPÍTULO IV	19
3.1 Desarrollo de la aplicación.....	19
3.1.1 Instalaciones Necesarias.....	19
3.1.2 Creación de Proyecto	19
3.1.3 Conexión y configuración de firebase.....	20
3.2 Creación de Páginas y rutas	21
CAPÍTULO V	28
4.1 Resultados	28
Socialización	28
Instalación	29
Capacitación.....	30
Pruebas de Uso.....	31
CAPÍTULO VI.....	32
5.1 Cronograma de actividades	32
5.2 Conclusiones	33
5.3 Recomendaciones.....	34
BIBLIOGRAFÍA.....	35

5.5 Glosario 37

6.1 Anexos..... 38

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Sala de Hospitalización	14
Ilustración 2 Diagrama de Flujo General, funcionamiento del sistema	16
Ilustración 3 Login, Home Y Nuevo, primeras interfaces	17
Ilustración 4 Interfaces Home y Editar	17
Ilustración 5 Alta de un paciente.....	18
Ilustración 6 configuración de firebase para nuestra app.....	20
Ilustración 7 Importaciones en app.module.ts.....	21
Ilustración 8 inicialización de nuestras importaciones.....	21
Ilustración 9 rutas creadas en app.routing.....	22
Ilustración 10 Importaciones requeridas para nuestro servicio.....	22
Ilustración 11 Creación de los servicios requeridos para el CRUD.....	23
Ilustración 12 Aplicación de HTML5 para la interfaz gráfica.	23
Ilustración 13 Interfaces ya creadas según diseños previos	24
Ilustración 14 Ventana de Medicación general de los pacientes.....	25
Ilustración 15 Ventana Lista medicación diaria de los pacientes	26
Ilustración 16 Ventanas de Login y Registro	27
Ilustración 17 Socialización de la aplicación	28
Ilustración 18 Instalación de la aplicación en dispositivos del personal.....	29
Ilustración 19 Capacitación.....	30

INTRODUCCIÓN

La Clínica Veterinaria Fernández de Córdova realiza múltiples actividades seguidas de diversos procesos que deben cumplirse a cabalidad con el fin de brindar cada día un mejor servicio a la ciudadanía cuencana. Esto implica la atención a sus clientes y el manejo de los pacientes en el área de hospitalización, por lo que existe la necesidad de automatizar dichas funciones.

La Veterinaria Fernández de Córdova ha brindado sus servicios desde 1993 y su fundador el Dr. Fernando Fernández de Córdova Vásquez con más de 27 años de experiencia y ejerciendo su profesión en el manejo de pequeñas especies está en constante actualización para el manejo de su Clínica, por ello con el diseño e implementación de un sistema que asista en el área de hospitalización mejorara la accesibilidad a la información y el estado del paciente, consiguiendo ahorro de tiempo y mayor facilidad al momento de almacenar y recuperar datos. Analizando detenidamente cómo funciona la Clínica Veterinaria se elabora una aplicación móvil capaz de facilitar el su trabajo.

Es importante recordar que esta aplicación no es una ficha medica como tal por el contrario es únicamente un asistente para el personal, un medio por el cual los médicos van a tener más control con los pacientes hospitalizados.

OBJETIVOS

Objetivo general

Desarrollar de un sistema que gestione los procesos en el área de hospitalización de la Clínica Veterinaria Fernández de Córdoba.

Objetivos específicos

- Automatizar el aviso para la aplicación de medicamentos de pacientes hospitalizados.
- Llevar un control digital de los pacientes hospitalizados.
- Mejorar la búsqueda de los pacientes por medio de una base de datos.

Preguntas de investigación

- ¿Cuáles son los requerimientos para poder implementar una aplicación móvil en la clínica veterinaria Fernández de Córdova?
- ¿Qué ventajas tiene una app móvil híbrida para manejar información?
- ¿De qué manera se puede obtener un mejor manejo del área de hospitalización?

Justificación

El presente proyecto permite generar ayuda a los médicos veterinarios para un mayor control al momento de realizar su atención previa durante y posterior en el área de hospitalización, mejorando la calidad de servicio de estos centros veterinarios, digitalizando información y gestión de sus procesos, para reducir el tiempo al verificar el estado de un paciente al momento de la hospitalización.

Al desarrollar una aplicación móvil e implementarla, se propone administrar las diferentes medicaciones, con el fin de obtener un mejor manejo de los pacientes y control de medicación.

Esta aplicación servirá como un asistente para los médicos en turno, muy diferente a un historial clínico el cual es permanente, los datos ingresados a esta aplicación simplemente pretenden ayudar al médico al momento de hospitalizar un paciente y durante su tratamiento, terminando sus funciones en el momento que el paciente es dado de alta.

CAPÍTULO I

1.1 Problema: necesidades

En el Cantón Cuenca, el avance tecnológico orientado hacia esta rama de la medicina veterinaria se ha descontinuado dado que los sistemas existentes tienen un alto costo, además de no cumplir con todas las necesidades que las Clínicas Veterinarias requieren.

Esto conlleva a que las clínicas veterinarias realicen sus procesos manualmente, malgastando tiempo y recursos. Al momento de realizar la hospitalización el registro de la medicación aplicada a los pacientes es muy básico y general, dejando a que los médicos dependan únicamente de su memoria para poder recordar que tiene que medicar al paciente a diferentes horas a lo largo del día.

En ocasiones el cambio de turno se presta para confusiones, ya que el medico que sale del turno ciertas veces no informa completamente al que ingresa sobre los pacientes hospitalizados, por esto, tener un medio en el cual los médicos puedan verificar o ser informados de la próxima medicación es de gran utilidad.

CAPÍTULO II

1.1 Marco Referencial

1.1.1 Marco Teórico

Ionic 4

Ionic es un equipo de desarrollo de software que fusiona HTML5, CSS3, Javascript o Typescript, para crear aplicaciones móviles híbridas todo esto siendo una herramienta gratuita y de código abierto.

(IonicFramework, 2019)

Angular

Framework JavaScript de código abierto que fue desarrollado por Google. Su trabajo es dar todos los medios técnicos necesarios para el desarrollo de este tipo de aplicaciones y proporcionar una base que permita desarrollar una aplicación organizada.

(OLLOVIER & GURY, 2016)

TypeScript

Es un superconjunto de JavaScript que está destinado u orientado en generar código seguro y predecible que tenga la capacidad de ser compilado por cualquier tiempo de ejecución JavaScript.

(Freeman, 2019)

Firebase

Proporciona diversas funciones tales como estadísticas, bases de datos, informes de fallas y mensajería, de manera que puedas ser más eficiente y enfocarte en tus usuarios. Siendo una herramienta gratuita, es muy eficiente gracias a sus servicios de bases de datos en tiempo real.

(Firebase, s.f.)

Apache Cordova

Transforma su aplicación HTML / JavaScript en un contenedor nativo que puede ingresar a diversas funciones de un Teléfono inteligente de distintas plataformas. Estas funciones se exponen por medio de una API JavaScript unificada, lo que le posibilita escribir de manera sencilla un grupo de códigos para señalar a casi todos los dispositivos móviles en el mercado hoy.

(Apache.Cordova, s.f.)

Apache Cordova es perfecto para desarrollar aplicaciones híbridas multiplataforma con HTML5, CSS3 y JavaScript, y da a los desarrolladores libertad ante interfaces de usuario específicas de la plataforma.

(Nachimuthu, 2017)

Metodología SCRUM

Scrum es un grupo de técnicas para crear y mantener productos complicados. Es un marco el cual los individuos pueden embarcarse en problemas adaptativos complicados, mientras distribuyen productiva y creativamente productos con mejor calidad. Scrum es simple de comprender y no es difícil de irlo mejorando.

(Altman, 2018)

Nodejs

Nació como un entorno de ejecución de JavaScript dirigido a casos asíncronos, está desarrollado para elaborar aplicaciones en red escalables, y en constante evolución, además de estar fuertemente orientada al desarrollo de aplicaciones móviles.

(nodejs, s.f.)

Visual Studio Code

Es un editor de texto multiplataforma. Es decir, que hay diferentes versiones para los distintos sistemas operativos existentes como lo son Windows, macOS y Linux, ofrece una gama de herramientas que facilitan su uso para los desarrolladores al momento de trabajar por medio de instalación de plugins externos que mejoran su interfaz.

(Johnson, 2019)

HTML

Hyper Text Markup Language.

Por medio del ordenamiento separado por tags, nos permite diferenciar cada componente de una pagina web para que de manera óptima y diferenciada sea visualizada del resto.

(Luna, Peña Millahual, & Lacono, PROGRAMACION WEB Full Stack 2 - HTML5: Desarrollo frontend y backend - Curso visual y práctico, Volumen 2 de PROGRAMACION WEB Full Stack - Curso visual y práctico , 2017)

HTML5

HTML tuvo distintas versiones con el tiempo a partir de su creación y HTML5 es su quinta versión, siendo así la más efectiva en cuanto a evolución desde el nacimiento de la web.

(Luna, Peña Millahual, & Lacono, PROGRAMACION WEB Full Stack 2 - HTML5: Desarrollo frontend y backend - Curso visual y práctico, Volumen 2 de PROGRAMACION WEB Full Stack - Curso visual y práctico , 2017)

CSS

Permite generar cientos de combinaciones, que logran la mejora del diseño base y la organización adecuada de los diversos elementos en los dispositivos.

(Luna, Peña Millahual, & Lacono, PROGRAMACION WEB Full Stack 10 - CSS avanzado: Desarrollo frontend y backend - Curso visual y práctico, 2018)

1.1.2 Marco Conceptual

Aplicaciones Móviles Nativas

Una aplicación móvil nativa se denomina a toda aplicación que allí sido desarrollada para ser instalada y usada únicamente en un sistema operativo determinado, es decir en un lenguaje o framework normalmente recomendado por el mismo fabricante del SO.

(Luna, Desarrollo web para dispositivos móviles: Herramientas para diseñar y programar WebApps, 2016)

Aplicaciones móviles Híbridas

Se define una aplicación Híbrida a una aplicación web diseñada y elaborada bajo estándares como HTML, CSS, JavaScript, etc., que empaqueta con un grupo de reglas y parámetros que posteriormente autorizara la instalación en cualquier dispositivo simulando una aplicación nativa.

(Luna, Desarrollo web para dispositivos móviles: Herramientas para diseñar y programar WebApps, 2016)

Ingreso de pacientes a hospitalización

El veterinario que realiza el ingreso debe cumplir de forma obligatoria el registro del paciente en la historia clínica en caso de no tener se le abre una historia clínica nueva. Debe tener todos los datos de identificación del paciente, su peso, propietario, raza, marcas, etc. También debe especificar de forma clara la patología o problema principal por la que ingresa a esta área.

(ucm.es)

Área de hospitalización

Es la zona donde se internan a los pacientes sean perros o gatos para proceder a terapias que exijan en confinamiento d ellos animales. En esta área consistente en baterías de jaulas generalmente se alojan animales que necesitan de terapia de fluidos(sueros). Es así que se pueden administrar medicamentos a horas determinadas y además se puede controlar el estado de salud de cada animal. Muchas enfermedades eligen este sistema de terapia para poder lograr la recuperación del paciente.

(Córdoba, 2019)

Pequeñas Especies

Generalmente se encarga de caninos y felinos, por otra parte queda a decisión de cada profesional elegir la línea de estudio.

(Alzate, 2017)

Prescripción Médica Veterinaria

La prescripción ha de ser por nombre de medicamento, especificando cantidad a suministrar y el tiempo de espera según el último medicamento administrado, es decir, cada que tiempo se administrara.

La medicación podrá referirse a:

- Un único animal o
- Un grupo de animales (siempre y cuando sean de la misma especie).

La medicación podrá indicar:

- Un único medicamento o
- Un grupo de medicamentos mismos que serán necesarios para tratar una dolencia, y aplicados como un único tratamiento.

(Consejo General de Colegios Oficiales de Farmacéuticos, 2016)

Normas ISO-27001

Se encarga de manejar los riesgos que tiene la información que línea, ya que se extiende con un gran número de controles para mejores practicasen cuando a seguridad, además pueden implementarse en empresas tanto pequeñas como gigantes internacionales.

(Calder, 2017)

CAPÍTULO III

2.1 Investigación Previa al Desarrollo

Para llevar a cabo el desarrollo de este proyecto, fue necesario realizar una investigación, realizando una entrevista y verificando las instalaciones.

2.2 Investigación

Se realizó una entrevista en primera instancia al Doctor Fernando Fernández de Córdova V. mismo que dirige la Veterinaria, además para que los datos sean más precisos se realizó una segunda entrevista a los médicos de los diferentes horarios que también manejaran la aplicación.

Las preguntas fueron:

- ¿Por qué cree necesario una plataforma web o aplicaciones móviles para su clínica veterinaria?
- ¿Al momento en el mercado, existe algún medio tecnológico igual o similar que cubra estas necesidades?
- ¿Cuál cree que es el mayor problema en este tema?
- ¿Con que frecuencia recibe pacientes para hospitalización?
- ¿Cuál es la capacidad de pacientes que usted puede llegar a tener hospitalizados al mismo tiempo?

Como parte de la misma investigación, se verifico que se cuente con internet y, que el personal cuente con dispositivos móviles actuales, confirmando que en general cuentan con dispositivos Android 2019, los cuales son ideales para el buen funcionamiento de la aplicación, y en caso de cambiar de Android a iOS nos hay problema alguno dado que Ionic permite la creación de aplicaciones móviles híbridas.



Ilustración 1 Sala de Hospitalización

Fuente: Autor

2.3 Recursos Financieros

Las herramientas utilizadas como Ionic, Firebase, Angular, Visual Studio Code, los cuales son software libre y no representan gasto alguno, por otro lado, una vez concluida las pruebas necesarias de funcionamiento de la plataforma, se externalizará los servicios de hosting para el alojamiento de la misma.

La movilización para la instalación del software y realizar las pruebas, correrán por cuenta propia.

Los gastos por internet, y luz eléctrica correrán por la Clínica Veterinaria Fernández de Córdova.

2.4 Diagramación

Tras diferentes revisiones y análisis de la investigación preliminar, se tuvo los datos suficientes para iniciar el proceso de diagramación, mismo que consto de dos partes, iniciando con un diagrama de flujo como estructura principal, posteriormente desarrollando diagramas de la aplicación mismo como tal y como debería funcionar.

Al desarrollar el diagrama de flujo se tuvo una idea mas clara de lo que la aplicación debería hacer para cumplir con las necesidades.

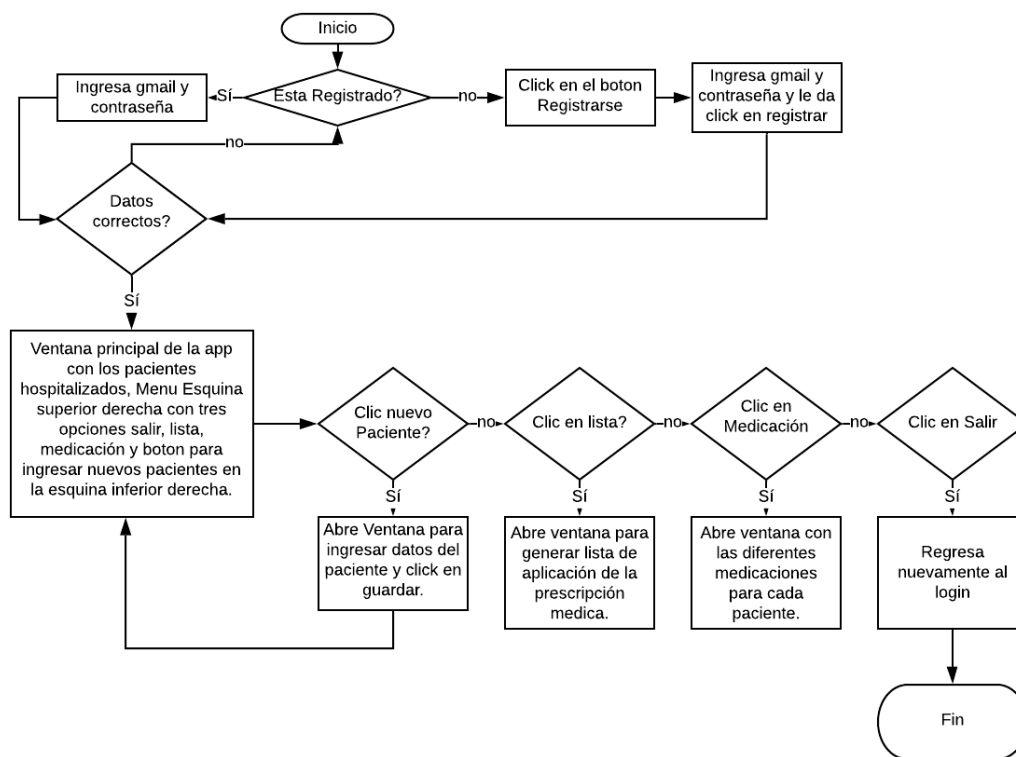


Ilustración 2 Diagrama de Flujo General, funcionamiento del sistema

Fuente: Autor

Además, sirvió como base para estructurar la aplicación, posteriormente se realizó diferentes diagramas más visuales del funcionamiento de la aplicación, estos fueron variando en cuestión de diseño y funcionamiento ya que se buscó la forma en la que esta aplicación sea lo más amigable posible para el personal de la veterinaria, llegando a un modelo final el cumple con los parámetros requeridos.

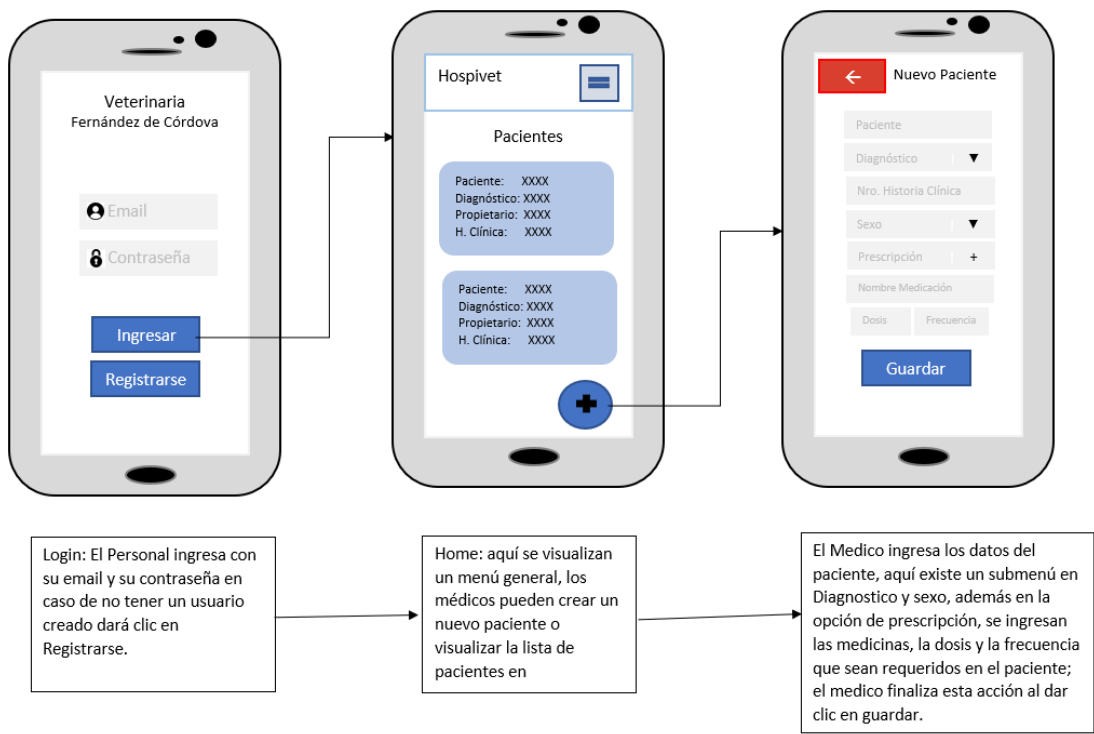


Ilustración 3 Login, Home Y Nuevo, primeras interfaces

Fuente: Autor

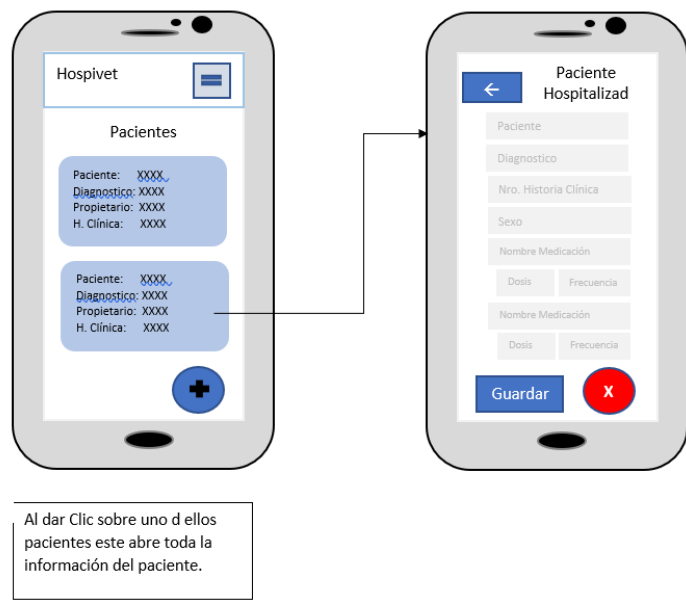


Ilustración 4 Interfaces Home y Editar

Fuente: Autor

En esta última ventana se puede editar datos del paciente, dar de alta o regresar al home.



Ilustración 5 Alta de un paciente

Fuente: Autor

Al dar de alta un paciente, nos mostrara una alerta la cual nos pedirá la confirmación de esta acción, al confirmar el paciente será dado de alta y saldrá del sistema, por otro lado, en caso de ser cancelada simplemente se cerrará la alerta y, nos mostrara la página en la que estábamos.

La aplicación fue tomando forma y ya con los diseños tanto de funcionamiento y de estética se inició el proceso de desarrollo de la aplicación.

CAPÍTULO IV

3.1 Desarrollo de la aplicación

3.1.1 Instalaciones Necesarias

Instalaciones requeridas para iniciar el desarrollo de la aplicación en orden:

1. Visual Studio Code
2. Nodejs.
3. Apache Cordova
4. Ionic
5. firebase

En el caso del Visual Studio Code, se puede usar cualquier editor de texto, pero gracias a sus diferentes ventajas se usó visual Studio Code,

Tanto Apache Cordova, Ionic y firebase, la instalación es por medio de la terminal.

3.1.2 Creación de Proyecto

Por medio de la consola se genera un nuevo proyecto usando el comando:

```
ionic start Hospivet blank
```

Donde Hospivet es el nombre de la aplicación y blank es una plantilla en este caso en blanco.

Para poder ver la aplicación por medio de nuestro navegador, escribimos el comando

```
Ionic serve
```

Es importante recalcar que debe estar en la ruta de nuestra aplicación, para ello visual studio code nos ayuda ya que incorpora una terminal; este proceso puede ser llevado a cabo en el cmd.

3.1.3 Conexión y configuración de firebase

1. Primero nos dirigimos a la página de firebase, y creamos un nuevo proyecto.
2. Para crear el nuevo proyecto solo hay que seguir las indicaciones por defecto de firebase.
3. Ya creado el proyecto, nos dirigimos a la opción de database y creamos una nueva base de datos
4. Nos dirigimos a configuración del proyecto, ahí tendremos todos los datos de acceso para nuestra aplicación.
5. En el apartado de web nos dará la apiKey, URL, projectId, etc y todo lo que necesitamos indicar en el proyecto.
6. Ahora ya dentro de nuestro proyecto se configuro firebase, para ello en el fichero environment pegamos todos los datos antes mencionados de firebase.



Ilustración 6 configuración de firebase para nuestra app

Fuente: Autor

En el fichero app.module.ts creamos algunas importaciones.

```
import { AngularFireModule } from 'angularfire2';
import { environment } from '../environments/environment';
import { AngularFireFirestoreModule } from 'angularfire2/firestore';
```

Ilustración 7 Importaciones en app.module.ts

Fuente: Autor

Y añadimos el AngularFireModule en el NgModule:

```
@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule, IonicModule.forRoot(), AppRoutingModule,
    AngularFireModule.initializeApp(environment.firebaseConfig),
    AngularFireFirestoreModule
  ]
})
```

Ilustración 8 inicialización de nuestras importaciones

Fuente: Autor

Hecho esto, pasamos a crear paginas servicios y la interface.

3.2 Creación de Páginas y rutas

En la carpeta app, en nuestra aplicación, creamos una nueva página llamada interface la cual contendrá los datos de nuestro paciente.

Posteriormente creamos la página de paciente, donde podremos crear, editar, actualizar, y eliminar los registros, mismos que serán almacenados en firebase, además debemos crear las rutas en `app-routing.module.ts`.

```
const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', loadChildren: './home/home.module#HomePageModule' },
  { path: 'register', loadChildren: './register/register.module#RegisterPageModule' },
  { path: 'login', loadChildren: './login/login.module#LoginPageModule' },
  { path: 'admin', loadChildren: './admin/admin.module#AdminPageModule' },
  { path: 'PresHora/id', loadChildren: './PresHora/PresHora.module#PresHoraPageModule' },
  // { path: 'admin', loadChildren: './admin/admin.module#AdminPageModule', canActivate:
  { path: 'details/id', loadChildren:
    './pages/todo-details/todo-details.module#TodoDetailsPageModule' },
  { path: 'details', loadChildren:
    './pages/todo-details/todo-details.module#TodoDetailsPageModule' },
  { path: 'PresHora', loadChildren: './PresHora/PresHora.module#PresHoraPageModule' },
  { path: 'prescripciones', loadChildren: './prescripciones/prescripciones.module#Prescri
```

Ilustración 9 rutas creadas en app.routing

Fuente: Autor

Ya con la ruta creada, creamos un servicio, ahí se crearon los métodos que se conectan con firebase, para lo que es necesario generar algunas importaciones.

```
2 import { AngularFireStore, AngularFireStoreCollection } from 'angularfire2/firestore';
3 import { Observable } from 'rxjs';
4 import { map } from 'rxjs/operators';
```

Ilustración 10 Importaciones requeridas para nuestro servicio.

Fuente: Autor

Por último, en esta parte creamos nuestros servicios.

```

getTodo(id: string) {
  return this.todosCollection.doc<TaskI>(id).valueChanges();
}

updateTodo(todo: TaskI, id: string) {
  return this.todosCollection.doc(id).update(todo);
}

addTodo(todo: TaskI) {
  return this.todosCollection.add(todo);
}

removeTodo(id: string){
  return this.todosCollection.doc(id).delete();
}

```

Ilustración 11 Creación de los servicios requeridos para el CRUD

Fuente: Autor

Finalizada la creación de nuestro servicio crearemos la página como tal, esto se realizó con la ayuda de HTML5 como se muestra en el ejemplo:

```

2 | <ion-toolbar color="primary">
3 |   <ion-title>
4 |     Ionic CRUD-TODOS
5 |   </ion-title>
6 | </ion-toolbar>
7 </ion-header>
8
9 <ion-content padding>
10 <ion-list>
11   <ion-item-sliding *ngFor="let todo of todos">
12     <ion-item>
13       <ion-label>
14
15         </ion-label>
16       </ion-item>
17     </ion-item-sliding>
18 </ion-list>
19 </ion-content>

```

Ilustración 12 Aplicación de HTML5 para la interfaz gráfica.

Fuente: Autor

Realizado el código en HTML5, la conexión a base de datos y creados los métodos, tenemos un CRUD funcional el cual nos permite crear y mostrar los pacientes.

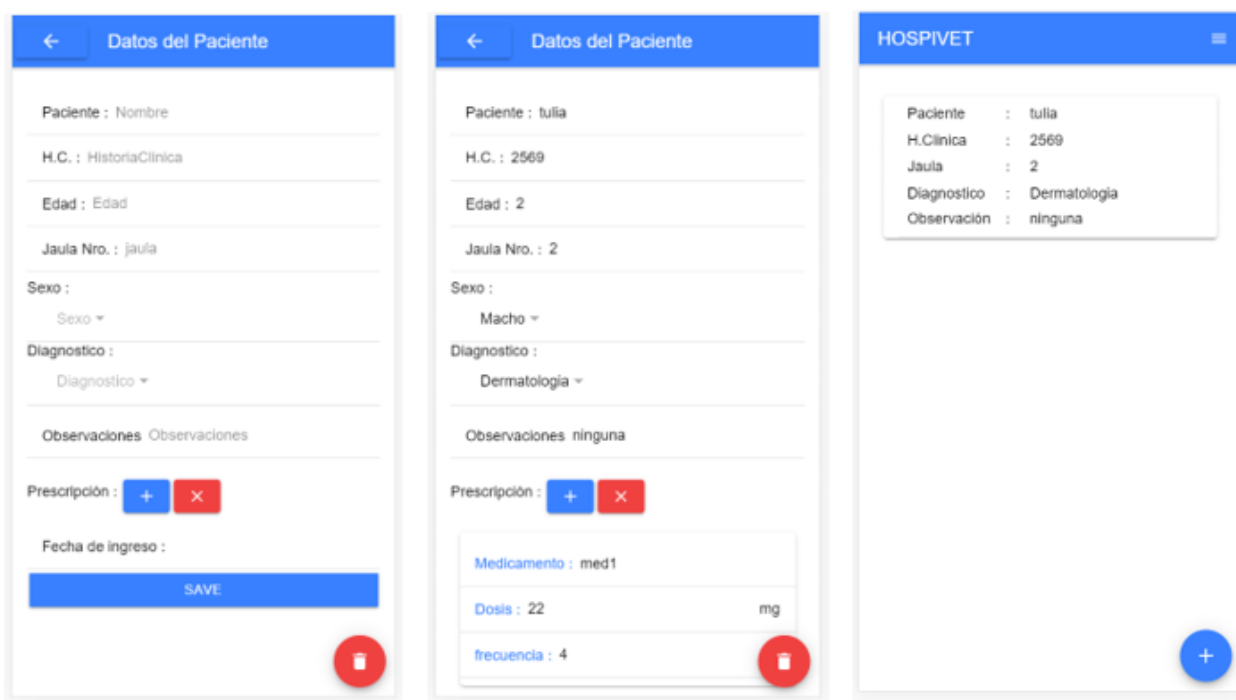


Ilustración 13 Interfaces ya creadas según diseños previos

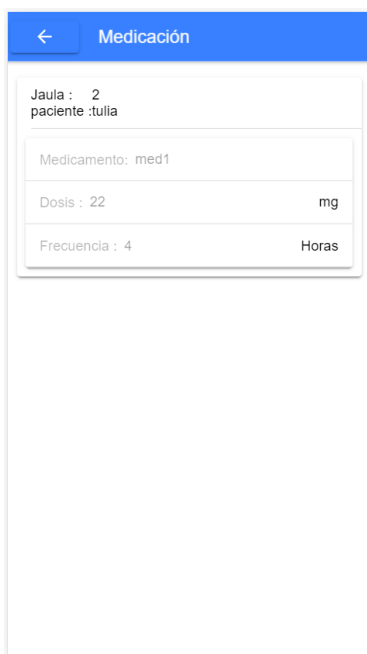
Fuente: Autor

Para mejorar el diseño se utilizó los componentes de ionic, los cuales varían en su diseño, también se implementaron algunos ionIcons, que son los iconos que ionic dispone para facilitar la mejora estética de la aplicación.

Es importante recalcar que los medios usados para realizar este proyecto son gratuitos, algunas herramientas como lo es firebase tienen un costo, pero eso es a partir de cierto número de registros, a partir de 30 000 registros ingresados, existe un costo, lo cual no implica un problema pues el número de registro esperados para la aplicación no supera los 5000 en caso de permanecer los datos pues los pacientes son datos de alta y los registros eliminados.

Siguiendo estos últimos procesos, se crearon diferentes ventanas:

Medicación: Utilizada para ver únicamente las prescripciones médicas de los pacientes



← Medicación	
Jaula : 2	
paciente :tulia	
Medicamento: med1	
Dosis : 22	mg
Frecuencia : 4	Horas

Ilustración 14 Ventana de Medicación general de los pacientes

Fuente: Autor

Lista de Medicación: Utilizada para generar la medicación diaria de los pacientes según su frecuencia.

← Lista de Medicacion

VER PROXIMAS MEDICACIONES

Paciente	: tulia
Hora	: 7:00
Medicamento	: med1
Dosis	: 22
Diagnostico	: Dermatologia
Observaciones	: ninguna

COMPLETADO

Paciente	: tulia
Hora	: 11:00
Medicamento	: med1
Dosis	: 22
Diagnostico	: Dermatologia
Observaciones	: ninguna

COMPLETADO

Paciente	: tulia
Hora	: 15:00
Medicamento	: med1
Dosis	: 22
Diagnostico	: Dermatologia

Ilustración 15 Ventana Lista medicación diaria de los pacientes

Fuente: Autor

Para controlar quien usa la aplicación también se creó un inicio de sesión el cual cuenta con un registro y un Login, al registrarse deben ingresar un Gmail y una contraseña alfanumérica.

En lo que respecta a seguridad Firebase posee los certificados ISO-27001 de seguridad y SOC1, SOC2 y SOC3, de esta forma se manejamos posibles amenazas.

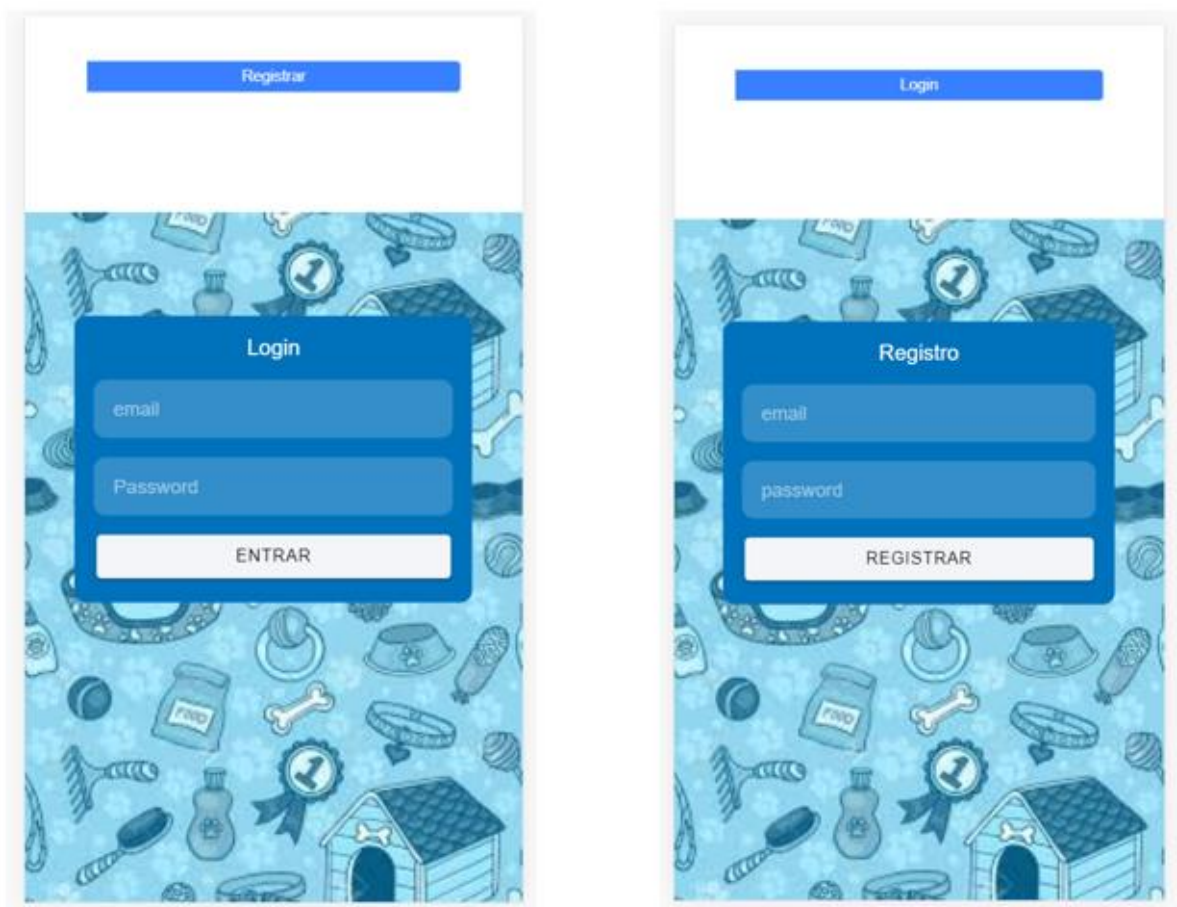


Ilustración 16 Ventanas de Login y Registro

Fuente: Autor

Quando un médico ya registrado ingresa a la aplicación, Firebase Registra su información y la fecha, controlando de esta manera el horario o a qué horas el personal ingresa a la aplicación.

CAPÍTULO V

4.1 Resultados

Socialización

El personal de la clínica veterinaria Fernández de Córdova contaba con conocimientos previos de la aplicación, ya que en el proceso de investigación se generó un conversatorio para analizar cuál sería el funcionamiento ideal para la aplicación.

Por lo tanto, la socialización nos sirvió para presentar la aplicación, verificando los requerimientos estipulados en un comienzo, explicando paso a paso que hace la aplicación.



Ilustración 17 Socialización de la aplicación

Fuente: Autor

Instalación

Culminada la socialización se procede a la instalación de la aplicación en los dispositivos del personal, para una mayor familiarización con su nuevo asistente hospitalario.

La aplicación no se encuentra en ninguna tienda de aplicaciones, el APK generado es enviado directamente al Doctor a cargo de la veterinaria mismo que se encarga de distribuirlo a los empleados que se encargan del área de hospitalización.

Descargada el APK solo se siguen las instrucciones predefinidas de instalación en sus dispositivos, también se brindó asesoría en este proceso para evitar cualquier tipo de molestia o inquietud.



Ilustración 18 Instalación de la aplicación en dispositivos del personal.

Fuente: Autor

Capacitación

En el momento que el personal ya cuenta con la aplicación en sus dispositivos se realiza la capacitación, misma que se efectúa en la clínica veterinaria.

La capacitación maneja los temas de:

1. Registro y Login.
2. Tipo de password.
3. Ingreso de pacientes.
4. Tipo de datos por campo.
5. Como Modificar datos de un paciente.
6. Como verificar prescripciones de los pacientes.
7. Generar medicación diaria de los pacientes.
8. Confirmación de cumplimiento de medicación diaria.
9. Alta de un paciente.



Ilustración 19 Capacitación.

Fuente: Autor

Este proceso tuvo la duración de una hora y media, fue dictada a 4 personas, el Dr. Fernando Fernández de Cordova, propietario, a dos Médicos Veterinarios que rotan turnos y a Alexandra Ambrosi, personal administrativo.

Se planifico un día y hora en la cual todos estuvieran en la veterinaria puesto que cumplen diferentes turnos entre horarios de la mañana, tarde y fines de semana.

Pruebas de Uso

En el momento que todo el personal participo y comprendió el funcionamiento de la aplicación, se procedió a realizar las primeras pruebas.

Ingresando un primer paciente al área de hospitalización la aplicación entro en uso.

El personal funciono muy bien con la aplicación, no tuvo complicaciones al usarla, además de generar el horario de medicación.

Al cambiar de turno uno de los principales problemas que se solía dar es el faltante de un registro del paciente en cuanto a la medicación, por ello el médico que llega a su turno consultaba al que esta de salida sobre la medicación, este problema se cubre gracias a la lista de medicación que genera la aplicación, por lo que ya no está faltando ese registro, al contrario, la aplicación muestra la próxima medicación o medicaciones pendientes a lo largo del día.

CAPÍTULO VI

5.1 Cronograma de actividades

Cronograma de actividades proceso de titulación octubre 2019 – febrero 2020.

ACTIVIDADES	Septiembre				Octubre				Noviembre				Diciembre				Enero			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Desarrollo y Aprobación de la Memoria (Proyecto Técnico).		X	X	X	X	X														
Investigación inicial				X	X	X	X	X												
Desarrollo de requerimientos							X	X	X											
Desarrollo del sistema									X	X	X	X	X	X	X	X	X	X	X	
Desarrollo y Aprobación del Proyecto Técnico										X	X	X	X	X	X	X	X	X	X	
Entrega del Informe Final del Proyecto Técnico																				X

5.2 Conclusiones

Desarrollar una aplicación híbrida en un nuevo lenguaje, requiere un gran esfuerzo, de igual manera tomar los requerimientos necesarios para el desarrollo de una aplicación, o un sistema en si debe ser claro y específico

Se debe tomar en cuenta cada detalle y, hacer las pruebas pertinentes en el transcurso del desarrollo para verificar el cumplimiento de los requerimientos.

La aplicación ayuda al personal de la clínica veterinaria, facilita su trabajo además de volver más eficiente al personal en el área de hospitalización mejorando su servicio y trato a las mascotas.

5.3 Recomendaciones

La aplicación, con el paso del tiempo debe crecer, abarcando más áreas y no solo la de hospitalización.

El personal debe ser actualizado en cuando a su funcionamiento, debe tener las capacitaciones necesarias para el completo entendimiento de la app.

Para agilizar el trabajo de desarrollo, es importante no solo tener la opinión de una persona en este caso de un médico veterinario, por el contrario, al tener la opinión de varios médicos se logra entender mejor el proceso de hospitalización.

BIBLIOGRAFÍA

- Altman, H. (2018). *Scrum: La Primera Metodología Agil Para Gestionar El Desarrollo de Productos Paso a Paso (Scrum in Spanish/ Scrum En Español)*. CreateSpace Independent Publishing Platform.
- Alzate, D. (28 de Noviembre de 2017). *medvetsite*. Obtenido de medvetsite: <https://medvetsite.com/especialidades-de-medicina-veterinaria/>
- Apache.Cordova. (s.f.). *cordova.apache.com*. Obtenido de cordova.apache.com: <https://cordova.apache.org/>
- Calder, A. (2017). *Nueve pasos para el éxito: Una visión de conjunto para la aplicación de la ISO 27001:2013*. Reino Unido: IT Governance Ltd.
- Consejo General de Colegios Oficiales de Farmacéuticos. (2016). *Guia Práctica, Dispensación y suministro de medicamentos de uso animal*. MADris: Redfarma.
- Córdova, D. F. (15 de Octubre de 2019). Área de hospitalización. (J. F. Ambrosi, Entrevistador)
- Firestore. (s.f.). *firebase*. Obtenido de firebase: <https://firebase.google.com/products/storage/>
- Freeman, A. (2019). *Essential TypeScript: From Beginner to Pro*. London, UK: Apress.
- IonicFramework. (9 de 12 de 2019). *ionicframework.com*. Obtenido de ionicframework.com: <https://ionicframework.com/docs/v1/guide/preface.html>
- Johnson, B. (2019). *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers*. John Wiley & Sons.
- Luna, F. (2016). *Desarrollo web para dispositivos móviles: Herramientas para diseñar y programar WebApps*. RedUSERS.

Luna, F., Peña Millahual, C., & Lacono, M. (2017). *PROGRAMACION WEB Full Stack 2 - HTML5: Desarrollo frontend y backend - Curso visual y práctico, Volumen 2 de PROGRAMACION WEB Full Stack - Curso visual y práctico* . RedUsers.

Luna, F., Peña Millahual, C., & Lacono, M. (2018). *PROGRAMACION WEB Full Stack 10 - CSS avanzado: Desarrollo frontend y backend - Curso visual y práctico*. RedUsers.

Nachimuthu, N. (2017). *Mastering Apache Cordova*. Dubai: Packt Publishing.

nodejs. (s.f.). *nodejs.org*. Obtenido de nodejs.org: <https://nodejs.org/es/about/>

OLLOVIER, S., & GURY, P.-A. (2016). *AngularJS: Desarrollo hoy las aplicaciones web de mañana*. Barcelona: Ediciones ENI.

ucm.es. (s.f.). Obtenido de umc.es: <https://www.ucm.es/data/cont/media/www/pag-92205/2.pdf>

5.5 Glosario

Multiplataforma

Aquel software que puede funcionar en diversas plataformas.

Software de código abierto

Aquel software cuyos derechos de autor son publicados bajo licencia de código abierto y pueden ser utilizados por cualquier tipo de persona sin costo alguno.

Depuración

Es el proceso en el cual se eliminan los errores encontrados en un programa.

Multilinguaje

Es cuando se manejan varios lenguajes de programación.

Gestor de Base de Datos

Es un conjunto de programas que administran y gestionan la información que posee una base de datos.

Servidor (Software)

Es un programa que brinda un servicio a otros programas llamados clientes.

Portabilidad

Es la capacidad de un programa de funcionar bajo diferentes sistemas.

Framework

Framework es un conjunto de herramientas, las cuales nos permiten trabajar de forma sencilla y la mayoría de veces podemos reutilizar en otros proyectos.

6.1 Anexos

Código

Home.page.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AngularFireAuth } from '@angular/fire/auth';
import { AuthService } from '../services/auth.service';
import { TaskI } from '../models/task.interface';
import { TodoService } from '../services/paciente.service';

import { PopoverController } from '@ionic/angular';
import { MenuComponent } from '../components/menu/menu.component';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage implements OnInit {
  todos: TaskI[];
  constructor(
    private authSvc: AuthService,
    private router: Router,
    private afAuth: AngularFireAuth,
    private todoService: TodoService,
    public popoverController: PopoverController) {}

  onLogout() {
    console.log('Logout');
    this.afAuth.auth.signOut();
    this.router.navigateByUrl('/login');
  }

  ngOnInit() {
    this.todoService.getTodos().subscribe(res => this.todos = res);
  }

  onRemoveall(idTodo: string) {
    this.todoService.removeTodo(idTodo);
  }

  onRemove(idTodo: string) {
```

```

    this.todoService.removeTodo(idTodo);
  }

  async openMenu(ev: any) {
    const popover = await this.popoverController.create({
      component: MenuComponent,
      event: ev,
      translucent: true
    });
    return await popover.present();
  }
}

```

Home.page.html

```

<ion-header>
  <ion-toolbar color="primary">
    <ion-buttons slot="end" >

      <ion-button slot="only-icon" (click)="openMenu($event)">
        <ion-icon name="menu"></ion-icon>
      </ion-button>
    </ion-buttons>
  <ion-title>
    HOSPIVET
  </ion-title>

</ion-toolbar>
</ion-header>
<ion-content padding>

  <ion-list>

    <ion-item-sliding *ngFor="let todo of todos">

```

```
<ion-card>
<ion-item lines="inset" button [routerLink]="['/details', todo]">
  <ion-grid >
    <ion-row>
      <ion-col size="4">
        Paciente
      </ion-col>
      <ion-col size="1"></ion-col>
      <ion-col>
        {{ todo.nombre }}
      </ion-col>
    </ion-row>

    <ion-row>
      <ion-col size="4">
        H.Clinica
      </ion-col>
      <ion-col size="1"></ion-col>
      <ion-col>
        {{ todo.HistoriaClinica }}
      </ion-col>
    </ion-row>

    <ion-row>
      <ion-col size="4">
        Jaula
      </ion-col>
      <ion-col size="1"></ion-col>
      <ion-col>
        {{ todo.jaula }}
      </ion-col>
    </ion-row>

    <ion-row>
      <ion-col size="4">
        Diagnostico
      </ion-col>
      <ion-col size="1"></ion-col>
      <ion-col>
        {{ todo.diagnostico }}
      </ion-col>
    </ion-row>

    <ion-row>
      <ion-col size="4">
```

```

    Observación
  </ion-col>
  <ion-col size="1"></ion-col>
  <ion-col>
    {{ todo.task }}
  </ion-col>
</ion-row>
</ion-grid>

</ion-item>

</ion-card>
<ion-item-options size="end">
  <ion-item-option (click)="onRemoveall()" color="secondary">
    <ion-icon name="checkmark" slot="end"></ion-icon>
  </ion-item-option>
</ion-item-options>
</ion-item-sliding>

</ion-list>
<ion-fab vertical="bottom" horizontal="end" slot="fixed">
  <ion-fab-button routerLink="/details" routerDirection="forward">
    <ion-icon name="add"></ion-icon>
  </ion-fab-button>
</ion-fab>

</ion-content>

```

Task.Interface.ts

```

import { Time } from '@angular/common';

// import { SecureContextOptions } from 'tls';

export interface TaskI {
  id?: string;
  task: string;
  priority: number;
  nombre: string;
  HistoriaClinica: number;
  Sexo: string;
  diagnostico: string;
}

```

```

    jaula: number;
    fecha_in: Date;
    aux: number;
    prescripciones: Prescripcion[];
  }

  export interface Prescripcion {

    Medicamento: string;
    dosis: number;
    Frecuencia: number;
  }

```

Paciente.services.ts

```

import { Injectable } from '@angular/core';
import { AngularFire, AngularFireCollection } from 'angularfire2/firestore';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
import { TaskI } from '../models/task.interface';

@Injectable({
  providedIn: 'root'
})
export class TodoService {

  private todosCollection: AngularFireCollection<TaskI>;
  private todos: Observable<TaskI[]>;

  constructor(public db: AngularFire) {

  }

  getTodos() {
    this.todosCollection = this.db.collection<TaskI>('todos');
    this.todos = this.todosCollection.snapshotChanges().pipe(
      map(actions => {
        return actions.map(a => {
          const data = a.payload.doc.data();
          const id = a.payload.doc.id;
          return {id, ...data};
        });
      });
  }

```

```
    })
  );
  return this.todos;
}

getTodo(id: string) {
  return this.todosCollection.doc<TaskI>(id).valueChanges();
}

updateTodo(todo: TaskI, id: string) {
  return this.todosCollection.doc(id).update(todo);
}

addTodo(todo: TaskI) {
  return this.todosCollection.add(todo);
}

removeTodo(id: string) {
  return this.todosCollection.doc(id).delete();
}
}
```