**Especificación de Requerimientos**

**Descripción del Diseño**

**Adopción de Mascotas**

**Aplicación Web para administración de organización dedicada a la entrega en adopción de animales**

|  |  |  |
| --- | --- | --- |
| Apellidos, Nombres | Correo electrónico | Rol |
| Karol Andrea Hormaza Chamorro |  |  |
| Guillermo Remires |  |  |
| Juan David Castro Marín |  |  |
| Luis Alberto Martínez Calderón |  |  |
| Santiago De Jesus Vega |  |  |

Fecha de presentación: 16/11/2021

Contenido

[**1**](#_heading=h.23ckvvd) **INTRODUCCIÓN 3**

[1.1](#_heading=h.1fob9te) Propósito 3

[1.2](#_heading=h.3znysh7) Alcance o Ámbito del Sistema 3

[1.3](#_heading=h.2et92p0) Definiciones, Acrónimos y Abreviaturas 3

[*1.3.1*](#_heading=h.tyjcwt) *Definiciones 3*

[*1.3.2*](#_heading=h.3dy6vkm) *Acrónimos 3*

[*1.3.3*](#_heading=h.1t3h5sf) *Abreviaturas 3*

[*1.3.4*](#_heading=h.4d34og8) *Referencias 4*

[1.4](#_heading=h.2s8eyo1) Perspectiva General del Documento 4

[**2**](#_heading=h.17dp8vu) **DESCRIPCIÓN GENERAL DE LA APLICACIÓN 4**

[2.1](#_heading=h.3rdcrjn) Perspectiva de la Aplicación 4

[2.2](#_heading=h.26in1rg) Funciones de la Aplicación 4

[2.3](#_heading=h.lnxbz9) Características de los Usuarios 5

[2.4](#_heading=h.35nkun2) Restricciones 5

[2.5](#_heading=h.1ksv4uv) Suposiciones y Dependencias 5

[2.6](#_heading=h.44sinio) Requerimientos Diferidos 5

[**3**](#_heading=h.2jxsxqh) **REQUERIMIENTOS ESPECÍFICOS 5**

[3.1](#_heading=h.z337ya) Requerimientos 6

[*3.1.1*](#_heading=h.3j2qqm3) *Product Backlog 6*

[*3.1.2*](#_heading=h.1y810tw) *Ciclo de Sprints del proyecto 6*

[*3.1.3*](#_heading=h.4i7ojhp) *Sprint Backlog 6*

[*3.1.4*](#_heading=h.2xcytpi) *Historias de usuario (Tareas y Subtareas) 6*

[*3.1.5*](#_heading=h.1ci93xb) *Mecánica de organización del grupo. (Reuniones, evidencias/artefactos) 6*

[3.2](#_heading=h.3whwml4) Modelo de Requerimientos 7

[*3.2.1*](#_heading=h.2bn6wsx) *Modelo de Casos de Uso 7*

[**4**](#_heading=h.qsh70q) **DESCRIPCIÓN DEL DISEÑO 9**

[4.1](#_heading=h.3as4poj) Interfaz gráfica (Mockups) 9

[**5**](#_heading=h.1pxezwc) **Gestión de la configuración 9**

[**6**](#_heading=h.ihv636) **PRUEBAS 9**

[6.1](#_heading=h.49x2ik5) Descripción de pruebas unitarias 9

[6.2](#_heading=h.2p2csry) Descripción de pruebas de aceptación 9

[**7**](#_heading=h.147n2zr) **GLOSARIO 10**

[**8**](#_heading=h.3o7alnk) **ANEXO(S) 10**

# Introducción

Con el presente documento damos a conocer nuestro proyecto con el cual pretendemos diseñar una aplicación web para la administración de organizaciones que se dedican a promocionar la adopción de mascotas.

## Propósito

Este documento pretende informar a los lectores sobre el funcionamiento de esta aplicación ya sean tutores, compañeros de estudio ó personas ajenas a la institución interesadas en conocer los alcances de estas campañas.

## Alcance o Ámbito del Sistema

La aplicación tiene nombre “Adopta”, la idea de hacerla se socializa en la reunión realizada por los integrantes del grupo E-7-19 en la apertura del ciclo 3, del curso Desarrollo de Software de la Universidad Tecnológica de Pereira. Adopta, es una aplicación que permite administrar una organización dedicada a la entrega en adopción de mascotas.

Funciones para administrador:

* Editar, agregar, eliminar registros (mascotas) en la aplicación de tal modo que el usuario pueda tener información real al hacer la consulta en la url.
* Tiene una bandeja de entrada de los mensajes que nos envian los usuarios a través de el formulario de la pagina

Funciones para usuarios:

* Consultar y ver las mascotas disponibles.
* Conocer información de cada una de las mascotas disponibles para escoger la que más se acomode a sus necesidades.
* Leer información de la organización
* Enviar mensajes de consulta al administrador de la página.
* Solicitar adoptar una mascota, enviar los datos que se requieren y esperar ser aceptado.

Funciones no disponibles.

* Administración contable de la organización.

El uso de esta aplicación permite que la información que se maneja en este tipo de organizaciones sea concentrada en una base de datos en la nube, por lo tanto dicha información va a estar disponible previa conexión desde cualquier dispositivo y va a estar segura.

Nuestra meta es que la aplicación sea de uso general, libre y con el tiempo vaya adquiriendo mas funcionalidades que en este corto tiempo no podemos lograr.

## Definiciones, Acrónimos y Abreviaturas.

**Framework:** Un entorno de trabajo​ (del inglés framework), o marco de trabajo​ es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio. *(Wikipedia)*

### Definiciones

### Acrónimos

### Abreviaturas

### Referencias

## Perspectiva General del Documento

En este documento se da a conocer nuestra la información de nuestro proyecto, en el siguiente orden, descripción general de la aplicación, requerimientos específicos, descripción del diseño, gestión de la configuración y pruebas.

# Descripción general de la aplicación

Nuestra web “Adopta”, busca facilitar el proceso de adopción de una mascota, ya que cuenta con una amplia base de datos de mascotas a nivel nacional, por lo tanto puedes buscar una mascota de forma sencilla sin importar la ciudad en la que te encuentres.

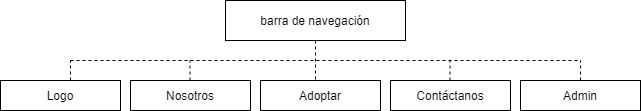
## Perspectiva de la Aplicación

Diseñamos esta web de forma autónoma entre 4 integrantes, haciendo uso de nuestros conocimientos en desarrollo web y bases de datos.

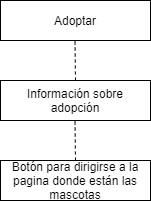
## Funciones de la Aplicación

Las funciones de nuestra página web son las siguientes:

* Por medio de una barra de navegación poder desplazarse por todos los apartados de la página web.



* Haciendo Click en el el logo sin importar en que parte de la pagina nos encontremos regresaremos a la página de inicio, en caso de ya estar en la página de inicio no sucederá nada.
* Haciendo click en adoptar se dirigirá a un apartado con una información muy importante antes de poder ver las mascotas para adoptar, debajo de esta información encontrará un enlace que lo llevará al apartado donde se encuentran las mascotas.



* En el apartado de contáctenos se dirigirá a un formulario que le solicitara sus datos y el motivo por el cual desea contactarse con nosotros, allí mismo encontrará el botón para enviar este formulario a nosotros.
* En la opción de Admin mostrará una nueva pagina que le pedirá su usuario y contraseña para ingresar a la página como administrador y así poder realizar cambios a la base de datos de las mascotas.

## Características de los Usuarios

* Personas interesadas en adoptar una mascota.
* Debe ser una persona mayor de edad para poder realizar la adopción.
* Contar con conocimientos básicos en el uso de herramientas informáticas

## Restricciones

* Requerimientos de lenguajes de alto nivel
* consideraciones de seguridad
* Tiempo de desarrollo

## Suposiciones y Dependencias

1. Nuestra aplicación estará disponible para ser utilizada en la mayoría de navegadores web.
2. Se tendrá una vista mas ordenada de todas las mascotas disponibles para la adopción.
3. Los datos enviados por los usuarios en el formulario de contacto se registran en una base de datos para tener mayor control.

## Requerimientos Diferidos

* Crear un login para los usuarios que no sean administradores del sitio y puedan guardar las mascotas que quieran adoptar, asi evitando que tengan que buscarla cada vez que consulten la página.
* Generar un mensaje de agradecimiento al diligenciar el formulario de contacto.
* Agregar una opción de donaciones para ayuda de las mascotas, la cual será opción del cliente si desea o no donar.

# Requerimientos específicos

* El diseño de la pagina debe ser usando Javascript, CSS, Base de datos en Mongo DB y Bootstrap 5.
* La página debe estar completa para el 7 de diciembre o antes.
* El diseño se realiza usando el editor de codigo Visual Studio Code.

## Requerimientos

* Creación de una página web fácil de usar.
* Crear un apartado de administrador para llevar un control de las mascotas que se agregan o eliminan de la base de datos.
* Un diseño ligero, con colores amigables entre tonos verdes y blanco.
* Diseñar un formulario de contacto completo y fácil de llenar.

### Product Backlog

|  |  |  |
| --- | --- | --- |
| TAREA | ESTADO | % FINALIZADO |
| Creación del esqueleto de la pagina web | Proceso | 60% |
| Ingresar la información de a las pestañas nosotros, adoptar y contacto | Proceso | 40% |
| Creación del archivo CSS con los estilos y la carpeta con las imágenes para el frontend de la página | Proceso | 40% |
| Desarrollo del backend para la conexión de la base de datos y creación del código para el servidor | Proceso | 40% |
| Realizar el montaje de la pagina web en Heroku |  | 0% |

### Ciclo de Sprints del proyecto

Los entregables de cada sprint deben ser los siguientes:

1. Sprint 0: Entrega en última clase semana 2

i. Product Backlog priorizado

ii. Repositorio (GitHub) donde integrarán el trabajo

iii. Gestión de configuración.

iv. Historias de usuario a desarrollar en el Sprint 1.

v. DevOps (Scrum board y tares).

2. Sprint 1: Entrega en última clase semana 3

i. Presentación MVC

1. Interfaces gráficas de usuario

2. Navegación entre interfaces

ii. Informe de retrospectiva

3. Sprint 2: Entrega en última clase semana 4

i. Presentación MVC

1. Interfaces gráficas de usuario

2. Navegación entre interfaces

ii. Informe de retrospectiva

iii. Historias de usuario a desarrollar en el sprint 2

4. Sprint 3: Entrega en última clase semana 5

i. Presentación MVC

1. Implementación de la lógica de negocio

2. Pruebas unitarias de la lógica desarrollada

ii. Informe de retrospectiva

iii. Historias de usuario a desarrollar en el sprint 3

5. Sprint 4: Entrega en última clase semana 6

i. Presentación MVC

1. Aplicación con persistencia local en MongoDB

2. Pruebas unitarias de la lógica desarrollada

ii. Informe de retrospectiva

iii. Historias de usuario a desarrollar en el sprint 4

6. Sprint 5: Entrega en última clase semana 7

i. Presentación MVC

1. Despliegue Back-End en Heroku

2. Despliegue base de datos en MongoAtlas

ii. Informe de retrospectiva

iii. Historias de usuario a desarrollar en el sprint 5

### Sprint Backlog

### Historias de usuario (Tareas y Subtareas)

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **ID:** 01 | **Nombre:** Unión del crud con el index |
| **Prioridad del negocio:** Alta | **Interacción Asignada:** |
| **Rol:** Yo como desarrollador | |
| **Funcionalidad:** Necesito realizar la unión del crud con la parte index | |
| **Criterio de Aceptación**:  · La página index.ejs debe estar con el crud | |

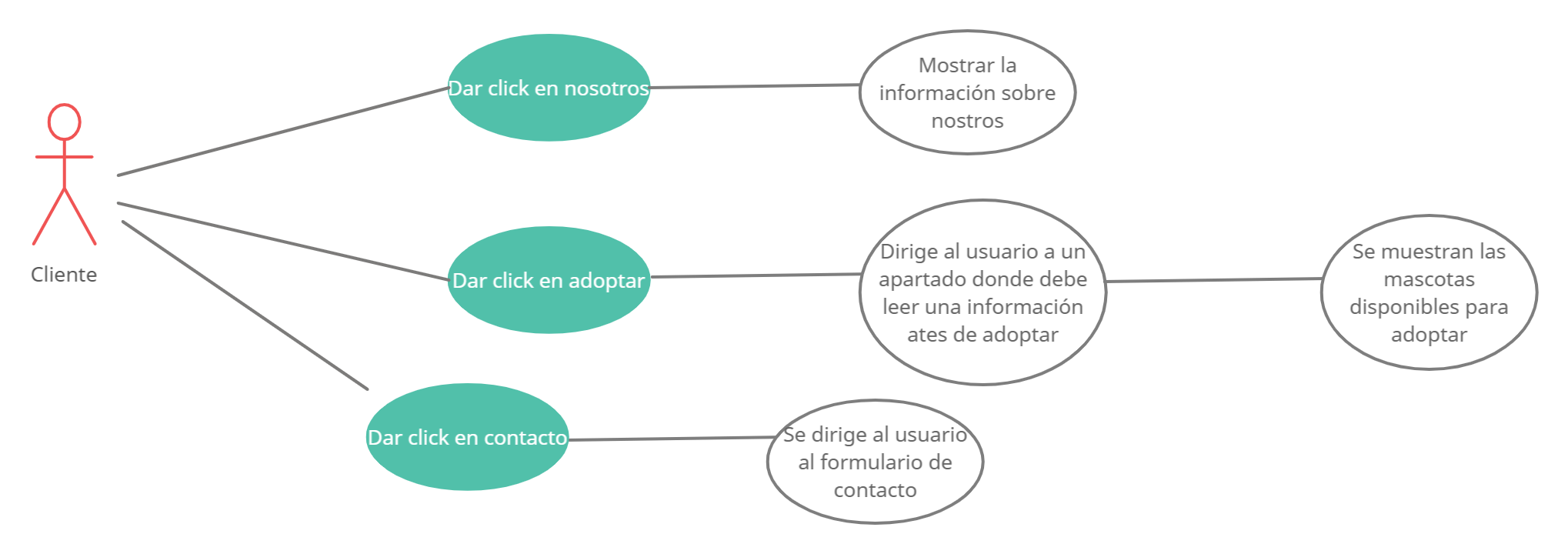
|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **ID:** 4 | **Nombre:** Uso de layaouts |
| **Prioridad del negocio:** Alta | **Interacción Asignada:** |
| **Rol:** Yo como usuario desarrollador | |
| **Funcionalidad:** debe integrar los layaouts en la página index.ejs | |
| **Criterio de Aceptación**:  · El archivo index.ejs debe tener los layaouts de los otros archivos para evitar tanto código. | |

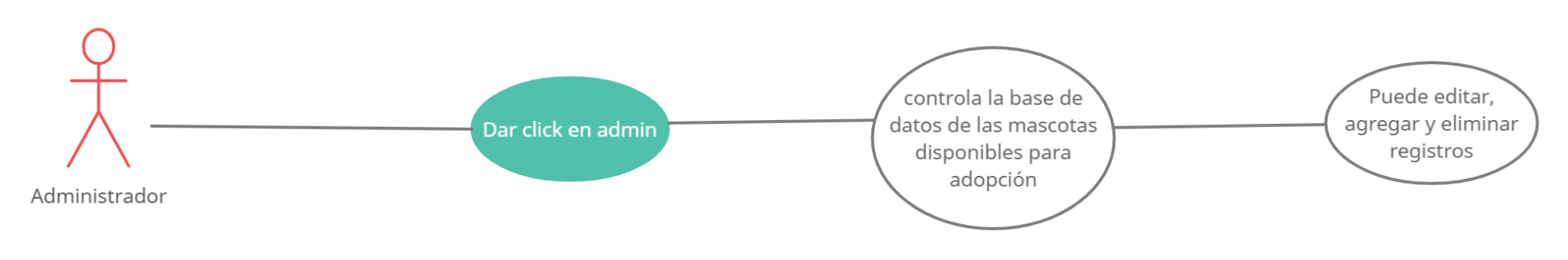
### Mecánica de organización del grupo. (Reuniones, evidencias/artefactos)

Para realizar un buen trabajo en equipo desde el momento que conformo el grupo realizamos reuniones por lo menos una vez a la semana para dividir el trabajo y saber que tenia que realizar cada uno, estas reuniones eran grabadas pro el tutor y se realizaban en la plataforma de google meet.

## Modelo de Requerimientos

### Modelo de Casos de Uso





**CU-01: "Conocer más sobre la página"**

|  |  |  |
| --- | --- | --- |
| Iniciador | Cliente | |
| Otros actores |  | |
| Precondiciones |  | |
| Flujo básico | | |
| **Actor** | | **Sistema** |
| 1.Dar click en nosotros | |  |
|  | | 2. El sistema lo dirigirá a la parte donde encontrara la información sobre nosotros. |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Flujo alternativo 1 | El usuario puede desplazarse con la barra de scroll a este apartado para lograr ver esta información | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

**CU-02: "querer adoptar"**

|  |  |  |
| --- | --- | --- |
| Iniciador | Cliente | |
| Otros actores |  | |
| Precondiciones |  | |
| Flujo básico | | |
| **Actor** | | **Sistema** |
| 1.Dar click en Adoptar | |  |
|  | | 2. El sistema lo dirigirá a la parte donde encontrará la información que el usuario debe leer antes de adoptar una mascota. |
| 3. Dar click en el botón estoy listo | |  |
|  | | 4.Se dirigira al usuario a una tabla donde se encuentran las mascotas disponibles para la adopción |
|  | |  |
|  | |  |
| Flujo alternativo 1 | El usuario puede desplazarse con la barra de scroll a este apartado para lograr ver esta información, pero después deberá dar click en el el botón que lo dirigirá a la tabal con las mascotas. | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

**CU-03: "Contactar con la gente de la página"**

|  |  |  |
| --- | --- | --- |
| Iniciador | Cliente | |
| Otros actores |  | |
| Precondiciones |  | |
| Flujo básico | | |
| **Actor** | | **Sistema** |
| 1.Dar click en contactanos | |  |
|  | | 2. El sistema lo dirigirá a la parte donde encontrará un formulario para diligenciar y poder enviarnos su mensaje. |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Flujo alternativo 1 |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

**CU-03: "Manipular la base de datos "**

|  |  |  |
| --- | --- | --- |
| Iniciador | Administrador | |
| Otros actores |  | |
| Precondiciones |  | |
| Flujo básico | | |
| **Actor** | | **Sistema** |
| 1.Dar click en admin | |  |
|  | | 2. El sistema lo dirigirá a la parte donde encontrará la información que se está mostrando en adoptar y podrá manipular esta información |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Flujo alternativo 1 |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

# Descripción del diseño

## Interfaz gráfica (Mockups)

# Gestión de la configuración

El diseño lo llevamos a cabo usando las siguiente tecnología de desarrollo:

FRONTEND: Bootstrap 5, Javascript

BACKEND: Javascript como lenguaje y NodeJs como entorno de ejecución. Dentro de este último cabe mencionar el uso de los siguientes módulos: Mongoose para la conexión con la base de datos, Express para el servidor, Morgan y Nodemon

Hemos optado por usar archivos en extensión .ejs debido a que se clasifican como archivos de desarrollo y a la vez porque en nuestro proceso de formación necesitamos abrirnos a nuevas formas y probar otros motores de plantillas que con el tiempo y gracias a la experiencia que vayamos adquiriendo irán definiendo nuestra metodología.

La base de datos utilizada es MongoDB. Inicialmente hemos trabajado con base de datos locales y posteriormente hemos migrado a la conexión en la nube gracias a el servicio Atlas ofrecido por Mongodb.

La aplicación se aloja en el servidor gratuito suministrado por Heroku app (heroku.com).

Para el trabajo en equipo hemos subido nuestro repositorio a GitHub, donde los cinco integrantes hemos ido haciendo a medida del desarrollo los commits necesarios. Esto nos permite trabajar a todos desde nuestras casas y mantener la revisión en tiempo real del trabajo de los demás.

Hemos usado localmente el editor de código Visual Studio Code y el aplicativo local Git para procesar y mantener al día el repositorio.

# Pruebas

## Descripción de pruebas unitarias.

* Inicialmente hemos diseñado el frontend, hemos probado la navegación en las diferentes páginas. En equipo hemos examinado, probado y corregido los errores que van saliendo. Html y CSS
* Conectamos la aplicación a la base de datos en modo local y probamos que se esté realizando la conexión correcta a la base de datos, que no haya errores o conflictos en el puerto utilizado.
* Migramos el diseño que teníamos en html, los hemos llevado a las plantillas .ejs, hemos configurado nuestra carpeta public dónde alojamos los .css e imágenes de la plantilla. Creamos los layouts para hacer más fácil los cambios en los menús y el footer.
* Procedimos codificando el crud inicial que permite introducir datos a la base de datos; esto nos permitió alimentar una base de datos con la información de las mascotas y su estado.
* Se crea el formulario de contacto, permitiendo que los mensajes se almacenen en la base de datos y puedan ser leídos desde el administrador.
* Una vez que consideramos que el producto es funcional modificamos la conexión de la base de datos, esta vez se conecta a mongo atlas y la app sesube a heroku.com

## Descripción de pruebas de aceptación

-Todas las pruebas aceptadas una vez en equipo convenimos que la aplicación realiza las funciones para las que fue diseñada.

# Glosario

# Anexos

-https://github.com/Juandah88/MisionTic\_Ciclo4.git