



DataPets

| Apellidos, Nombres | Correo electrónico | Cedula | Rol |
|--------------------|-----------------------------|---------------|------------------------|
| Amador, Narda | narduk.a@gmail.com | 1.019.060.527 | Gestor bases de datos |
| Ferreira, Sharom | ferreiravela6@gmail.com | 1.090.523.999 | Tester |
| Mican, Paola | paomr92@gmail.com | 1.069.740.628 | Desarrollador Frontend |
| Peña, Joel | joelsmp07@gmail.com | 1.092.354.079 | Gestor de proyecto |
| Pidiache, Blanca | blancapidiache274@gmail.com | 1.056.436.081 | Desarrollador Backend |

Enlaces del proyecto

Enlace de Trello: <https://trello.com/b/YAReJSeK/datapet>

Enlace en la nube de su proyecto: <https://datapets.herokuapp.com/>

Enlace repo con todo el código: <https://github.com/blanca-marlen/DataPets.git>

Enlace video explicando las funcionalidades de la app: <https://www.youtube.com/watch?v=E6UKNqPXkew>

Especificación de Requerimientos

Descripción del Diseño

DATAPETS

| Apellidos, Nombres | Correo electrónico | Rol |
|--------------------|-----------------------------|------------------------|
| Amador, Narda | narda.amador@outlook.com | Gestor bases de datos |
| Ferreira, Sharom | ferreiravela6@gmail.com | Tester |
| Mican, Paola | paomr92@gmail.com | Desarrollador Frontend |
| Peña, Joel | joelsmp07@gmail.com | Gestor de proyecto |
| Pidiache, Blanca | blancapidiache274@gmail.com | Desarrollador Backend |

Fecha de presentación: 05/OCT/2022

Contenido

| | | |
|----------|---|-----------|
| 1 | INTRODUCCIÓN | 2 |
| 1.1 | PROPÓSITO | 2 |
| 1.2 | ALCANCE O ÁMBITO DEL SISTEMA | 2 |
| 1.3 | DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS | 3 |
| 1.3.1 | <i>Definiciones</i> | 3 |
| 1.3.2 | <i>Acrónimos</i> | 7 |
| 1.3.3 | <i>Abreviaturas</i> | 9 |
| 1.3.4 | <i>Referencias</i> | 11 |
| 1.4 | PERSPECTIVA GENERAL DEL DOCUMENTO | 11 |
| 2 | DESCRIPCIÓN GENERAL DE LA APLICACIÓN | 11 |
| 2.1 | PERSPECTIVA DE LA APLICACIÓN | 12 |
| 2.2 | FUNCIONES DE LA APLICACIÓN | 12 |
| 2.3 | CARACTERÍSTICAS DE LOS USUARIOS | 12 |
| 2.4 | RESTRICCIONES | 13 |
| 2.5 | SUPOSICIONES Y DEPENDENCIAS | 13 |
| 2.6 | REQUERIMIENTOS DIFERIDOS | 13 |
| 3 | REQUERIMIENTOS ESPECÍFICOS | 14 |
| 3.1 | REQUERIMIENTOS | 14 |
| 3.1.1 | <i>Product Backlog</i> | 14 |
| 3.1.2 | <i>Ciclo de Sprints del proyecto</i> | 14 |
| 3.1.3 | <i>Sprint Backlog</i> | 15 |
| 3.1.4 | <i>Historias de usuario (Tareas y Subtareas)</i> | 15 |
| 3.1.5 | <i>Mecánica de organización del grupo. (Reuniones, evidencias/artefactos)</i> | 20 |
| 4 | DESCRIPCIÓN DEL DISEÑO | 21 |
| 4.1 | INTERFAZ GRÁFICA (MOCKUPS) | 21 |
| 5 | GESTIÓN DE LA CONFIGURACIÓN | 30 |
| 6 | PRUEBAS | 35 |
| 6.1 | DESCRIPCIÓN DE PRUEBAS UNITARIAS | 35 |
| 6.2 | DESCRIPCIÓN DE PRUEBAS DE ACEPTACIÓN | 41 |
| 7 | GLOSARIO | 41 |

1 INTRODUCCIÓN

1.1 Propósito

Con el presente documento se pretende explicar de forma detallada la descripción del diseño de la página web llamada DataPets, la cual se elaboró con el fin de dar cumplimiento a los objetivos planteados en el ciclo 3 del programa de Desarrollo de software del ministerio TIC, MisiónTIC, impartido por la Universidad Tecnológica de Pereira.

1.2 Alcance o Ámbito del Sistema

La página web DataPets, es una página web creada para que las personas que poseen una mascota (perros o gatos) puedan conocer de antemano, cuáles establecimientos son pet friendly (permiten la entrada de dichos animales domésticos a sus establecimientos), así se evitan la problemática de no poder entrar a diferentes lugares o centros comerciales.

En la página web, los usuarios registrados tienen un número ilimitado de búsquedas mientras que los usuarios que no quieran registrarse (invitados) solo podrán realizar un máximo de 5 visitas. En la página no se podrán realizar reservas de ningún tipo, solo se usa como un medio para informarse sobre si un establecimiento acepta o no mascotas.

Dentro de los objetivos se encuentran:

Objetivo general:

Diseñar e implementar una página web que brinde información relevante sobre establecimientos que permitan el ingreso de personas y sus mascotas.

Objetivos específicos:

- Desarrollar una interfaz amigable para el usuario.
- Analizar el diseño o estructura del software a utilizar en este proyecto.
- Utilizar las herramientas necesarias obtenidas durante el proceso del curso.
- Realizar la ejecución de las pruebas necesarias para el correcto funcionamiento de la página.

1.3 Definiciones, Acrónimos y Abreviaturas

1.3.1 Definiciones

Adwords

Sistema de Google de anuncios publicitarios de texto o gráficos, que usualmente aparecen en el lado derecho de la pantalla de los resultados de búsqueda. También pueden aparecer en la red de sitios web que están afiliados a AdSense de Google.

Algoritmo de Google

El Algoritmo de Google es la forma que tiene el buscador de posicionar las páginas ante una búsqueda, es decir, es lo que decide si sales primero, segundo o en la segunda página. Este algoritmo cambia unas 500 veces al año y resulta difícil seguirle la pista. Por eso es preferible conocer bien cambios importantes como Panda y Penguin, cómo afectan al SEO y cómo recuperarnos.

Apache

Apache es un programa de servidor HTTP Web de código abierto (open source). Su desarrollo empezó en 1995 y actualmente es uno de los servidores web más utilizados en la red.

Avatar

Personalidad o identidad visual que se atribuyen algunos usuarios de Internet, ya sea en algún chat, juegos, etc. Un avatar es un facsímil gráfico que se puede utilizar en las habitaciones de discusión virtual o chat rooms y que permite jugar un rol e interactuar con gente en línea. El término se hizo popular en la novela Snow Crash de Neal Stephenson.

Backup

Copia de Respaldo o Seguridad. Acción de copiar archivos o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales. Esta sencilla acción evita numerosos, y a veces irremediables, problemas si se realiza de forma habitual y periódica.

Base de datos

Datos almacenados en una computadora que pueden ser accedidos y manipulados fácilmente; piense en un gran gabinete físico con muchas etiquetas con archivos de información relevante.

Caché

Es la habilidad de un navegador o de un servidor para guardar copias de páginas Web en un disco duro con la ventaja de que carga más rápido dado que la información no debe de ser "pedida y entregada" por un servidor si no que apunta a un lugar en específico mucho más cercano.

CDN

Una red de distribución de contenido no es más que un servicio que te permite alojar una versión de tu sitio Web en varios servidores para mejorar la velocidad de carga de tu página Web. De manera sencilla, se almacena la información en estos servidores y, dependiendo de dónde se haga la consulta, se redirige la petición al servidor más cercano, esto ahorra recursos en el servidor original, porque disminuye el número de solicitudes que se hacen en un mismo periodo de tiempo.

CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Desarrollador web (programador web)

Un desarrollador web es un creador de páginas web que enfoca su trabajo en el desarrollo interno del funcionamiento de una página web.

<html>

El elemento <html> es el primero de cualquier documento HTML. Envuelve el resto del conjunto. Si origina una página Web normal, el elemento <html> contiene dos ingredientes esenciales más: el elemento <head>, que define el título, los metadatos y las hojas de estilo vinculadas, y el elemento <body>, que alberga el contenido real.

HTTPS

Protocolo de Transferencia de Hipertexto sobre SSL (Secure Socket Layer). Es un protocolo TCP/IP utilizado por los servidores Web para transferir y mostrar contenido Web de forma segura. Los datos transmitidos son encriptados de manera que no se pueden leer por nadie a excepción del receptor. HTTPS es usado por cualquier sitio Web que esté recopilando información sensible de sus clientes como información bancaria o de compra. Si estás haciendo una transacción online, deberías asegurarte de que esta se lleva a cabo sobre HTTPS para que los datos permanezcan seguros y se transmitan de forma segura.

** (Imagen)**

El elemento apunta a un archivo de imagen que desea mostrar en la página. El atributo src lo identifica empleando un enlace relativo o absoluto.

<input>

El elemento <input> es el ingrediente más común de un formulario HTML (representado por el elemento <form>). Puede representar distintos artilugios en pantalla (llamados controles) que recogen la información de un visitante Web.

ID

Selector CSS que nos permite especificar propiedades de estilo para un elemento que no se repite en la página. En HTML se representa como id, en CSS se representa con un símbolo de numeral o gato "#"

Input

Etiqueta HTML que permite colocar un elemento en que el usuario introduce información, que puede ser en forma de texto, casilla de verificación, fecha, usuario y contraseña, email, URL entre otra

Página Web

Es una unidad de información individual a la que se accede a través de la WWW, en ella se presentan texto, imágenes, sonidos, vídeos y enlaces, y su extensión puede ser más larga que una pantalla de computador.

Pixel

Es la menor unidad de medición de los elementos gráficos mostrados en pantallas. Un píxel es equivalente a un punto de la pantalla, que es la división más pequeña de la resolución de la pantalla.

Servidor

Es el computador en el que se ejecutan los programas que realizan alguna tarea en beneficio de otras computadoras. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un sitio web, así como ejecutar sus aplicaciones asociadas, en beneficio directo del usuario final.

Sitio Web

Varias páginas web pueden estar agrupadas conformando un sitio Web, se trata de productos comunicacionales cuya característica básica es que potencian una "desestructuración comunicativa", es decir, que el sitio no hace explícito todas sus posibilidades de una sola vez, sino que para conocerlas se incita al usuario a explorar y a interactuar con los distintos elementos que aparecen en pantalla.

Sprites

Conjunto de técnicas de manejo de imágenes mediante CSS que permite al encargado del diseño web colocar diversas imágenes en un solo archivo de imagen y posteriormente llamarlas utilizando su posición y tamaño específicos, lo cual se traduce en menos espacio ocupado en el servidor, menos peticiones generadoras de tráfico innecesario al mismo, y por lo tanto una página que carga más rápido para el usuario

SSL

Son las siglas en inglés de Secure Socket Layer. Es un protocolo criptográfico empleado para realizar conexiones seguras entre un cliente y un servidor.

URL

Uniform Resource Locator. Sistema de direccionamiento estándar de archivos y funciones en Internet, especialmente en la WWW. El URL está formado por el protocolo de servicio (http: ftp: gopher: mailto), el nombre del servidor u ordenador que contiene el recurso, la ruta de acceso al recurso y el recurso buscado.

Visita

Una visita es una serie de peticiones de un visitante único en un período de tiempo determinado. Cada vez que realizamos clic en un enlace realizamos una vista de página (pageview) y podemos observar las rutas que los usuarios utilizan para navegar por el website.

Visitante

Una visita es una persona que visualiza el website. Un visitante puede visitar una o varias páginas por lo que es interesante tener este registro en base a herramientas propias o profesionales como Google Analytics. Para identificar los visitantes y extraer cierta información podemos obtener datos de ellos a través de su dirección IP, navegador, sistema operativo, idioma, etc. El número de visitas es una métrica que indica la popularidad de una web o website.

Vista de página (pageview)

Una vista de página es una petición para un ítem llamado página en analítica Web. Una página se define como todas las peticiones que son necesarias para construirla. Por ello una vista de página puede incluir muchos hits, dado que una página se construye con CSS, archivos e imágenes. Las vistas de páginas son una buena unidad de medida/métrica en las analíticas Web. El número de vistas de páginas o "pageviews" de un sitio web dan una medida de lo popular que es un sitio y de cuán atractivo será para los anunciantes. La mayoría de herramientas de analíticas Web te mostrarán las vistas de páginas (pageviews).

1.3.2 Acrónimos

- **AJAX** (Asynchronous JavaScript And XML)
- **API** (Application Program Interface)
- **APNG** (Animated Portable Network Graphics)
- **ARPA** (Advanced Research Projects Agency)
- **ASCII** (American Standard Code for Information Interchange)
- **ASP** (Active Server Pages)
- **BMP** (Bitmap)

- **BOM** (Browser Object Model)
- **BOM** (Byte Order Mark)
- **CERN** (Conseil Européenne pour le Recherche Nucléaire)
- **CGI** (Common Gateway Interface)
- **CSS** (Cascading Style Sheets)
- **DARPA** (Defense Advanced Research Projects Agency)
- **DHTML** (Dynamic HTML)
- **DNS** (Domain Name System)
- **DOM** (Document Object Model)
- **DTD** (Document Type Definition)
- **ECMA** (European Computer Manufacturers Association)
- **GIF** (Graphics Interchange Format)
- **HTML** (HyperText Markup Language)
- **HTTP** (HyperText Transfer Protocol)
- **HTTPS** (Hypertext Transfer Protocol Secure)
- **IANA** (Internet Assigned Numbers Authority)
- **IETF** (Internet Engineering Task Force)
- **INRIA** (Institut National de Recherche en Informatique et en Automatique)
- **ISO** (International Organization for Standards)
- **JPG/JPEG** (Joint Photographic Experts Group)
- **JS** (JavaScript)
- **JSF** (Java Server Faces)
- **JSP** (Java Server Pages)
- **LZW** (Lempel Ziv Welch)
- **MIME** (Multipurpose Internet Mail Extensions)

- **MIT** (Massachusetts Institute of Technology)
- **MNG** (Multiple-image Network Graphics)
- **PDF** (Portable Document Format)
- **PHP** (PHP Hypertext Pre-processor)
- **PNG** (Portable Network Graphics)
- **RFC** (Request for Comments)
- **RGB** (Red Green Blue)
- **RIA** (Rich Internet Application)
- **SEO** (Search Engine Optimization)
- **SGML** (Standard Generalized Markup Language)
- **SMIL** (Synchronized Multimedia Integration Language)
- **SSL** (Secure Socket Layer)
- **SVG** (Scalable Vector Graphics)
- **URL** (Uniform Resource Locator)
- **UTF** (Unicode Transformation Format)
- **W3C** (World Wide Web Consortium)
- **WAI** (Web Accessibility Initiative)
- **WCAG** (Web Content Accessibility Guidelines)
- **WWW** (World Wide Web)
- **XHTML** (eXtensible HyperText Markup Language)
- **XML** (eXtensible Markup Language)

1.3.3 Abreviaturas

- **HTML:** (HyperText Markup Language). Lenguaje de descripción del contenido y la estructura de una página web. Es el lenguaje básico en el que están escritas las páginas web.
- **CSS:** (Cascade Style Sheet). Hojas de estilo en cascada. Es un lenguaje que permite definir el aspecto de los textos, bloques y otros contenidos de una página web. Por ejemplo, ¿el color, el tamaño, la posición en pantalla? Mientras el HTML define los contenidos y su estructura jerárquica, el CSS determina cómo se muestran en pantalla.
- **JS:** abreviatura de **JavaScript**, un lenguaje de programación que permite añadir interactividad a las páginas web (ver DHTML)
- **DHTML:** (HTML Dinámico): uso conjunto de HTML y JavaScript que permite aportar animación e interactividad a una página web. ¿Puede usarse para mostrar un pase de diapositivas, paneles desplegables?
- **XML:** (eXtended Markup Language). Es un lenguaje que permite describir los contenidos de una página web. Etiqueta cada ítem (p.ej. nombre, teléfono, descripción, código postal) para que la información esté correctamente estructurada y pueda ser empleada adecuadamente por una aplicación web. También se usa como contenedor de información alfanumérica en informática en general.
- **RSS:** (Really Simply Syndication). Es un formato especial de XML que permite distribuir información estructurada, como, por ejemplo, las publicaciones de un blog (¿título, fecha, autor, categoría, resumen o contenido completo?). Puede mostrarse la información de un RSS convenientemente formateada mediante un programa específico (llamado *agregador*) o un plugin (ampliación) de un navegador web.
- **AJAX:** (Asynchronous Javascript And XML). XML y Javascript asíncrono. Uso combinado de ambos lenguajes que permite descargar y mostrar nuevos contenidos en una página web sin necesidad de acceder a otra página web o descargarla completa de nuevo. Por ejemplo, las sugerencias (texto predictivo) que se muestran en la caja de búsqueda en Google.
- **PHP:** lenguaje de programación para páginas web que se ejecuta en el servidor (en el hosting). Permite utilizar información de una base de datos o añadirla a la misma, realizar cálculos, etc., para generar una *página web dinámica* que se genera en el momento en que se visita dicha web.
- **ASP:** otro lenguaje de programación web que se ejecuta en el servidor, con características similares al PHP, aunque es de propiedad de Microsoft, y de pago.
- **MySQL:** sistema de gestión de base de datos muy utilizado en programación de aplicaciones web, en combinación con el lenguaje PHP.

1.3.4 Referencias

Schwinger, W.; Koch, N. "Modeling Web Applications", Chapter 3 en: Kappel, G.; Pröll, B.; Reich, S.; Retschitzegger, W. (Editors) Web Engineering. The Discipline of Systematic Development of Web Applications, John Wiley & Sons Ltd., 2006.

Koch, N.; Knapp, A.; Zhang, G.; Baumeister, H. "UML-Based Web Engineering. An Approach Based on Standards", Chapter 7 en: Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. (Editors) Web Engineering: Modelling and Implementing Web Applications, Springer-Verlag London Limited, 2008.]

1.4 Perspectiva General del Documento

Luego de la introducción al proyecto, se realiza una descripción general de la aplicación, la cual incluye: las perspectivas y funciones de la aplicación, las características restricciones, supuestos, dependencias y requerimientos de los usuarios. La siguiente sección es la sección de requisitos, en la que se reflejan los Sprint, las historias de usuarios y las reuniones.

La siguiente sección es la interfaz gráfica, donde se capturan las maquetas que se desarrollaron a lo largo de la ejecución del proyecto. Otra sección del presente documento trata sobre la gestión de la configuración, donde se mencionan los lenguajes de programación, Frameworks y otras herramientas.

Finalmente, están las verificaciones, en este apartado se describen las pruebas unitarias y de aceptación del sitio web.

2 DESCRIPCIÓN GENERAL DE LA APLICACIÓN

DataPets busca llegar a las personas que tienen una mascota (perros o gatos) para que puedan conocer algunos establecimientos pet-friendly de la ciudad, esto hará más sencillo para dichos usuarios planear a qué lugares o centros comerciales irán con sus mascotas. Principalmente encontramos que les puede ayudar a dar un mejor uso del tiempo y ahorro del mismo.

El sitio web mostrará establecimientos alrededor de toda la ciudad con su respectiva información de contacto. La información será proporcionada por los establecimientos y se esperaría mantenerla constantemente actualizada dado que los propios establecimientos tienen acceso a modificar su información.

Para poder entender las funcionalidades del sitio, se realizan y desarrollan las historias de usuario. Estas llevaron a un mejor entendimiento y aprendizaje a lo largo de la ejecución del

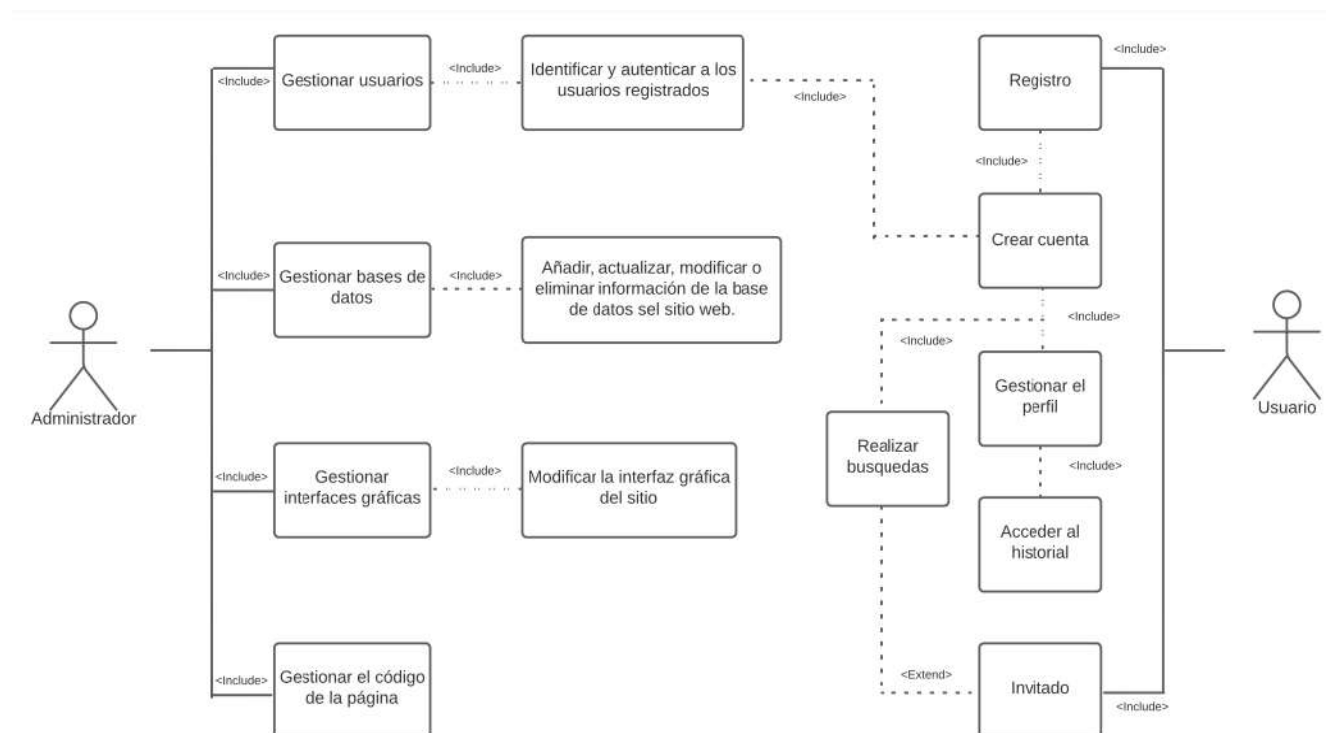
proyecto. De esta manera, se desarrolló una base de código con soporte para navegador web móvil y de escritorio que es visualmente atrayente y sencillo en términos de entendimiento para cualquiera de los tipos de usuario que tendrán acceso. El objetivo es proporcionar una interfaz fácil de usar, por lo que los elementos son altamente descriptivos.

2.1 Perspectiva de la Aplicación

El software funcionará en un entorno web lo que permitirá que se pueda acceder desde cualquier computador o dispositivo móvil con acceso a la red y conexión al servidor el cual va a estar corriendo la aplicación.

El sistema contará con una interfaz que permita a los usuarios utilizar la página sin estar registrado (invitado) o registrarse para acceder a más servicios y disfrutar de una mejor experiencia.

2.2 Funciones de la Aplicación



2.3 Características de los Usuarios

| | |
|---------------------------|--|
| Tipo de usuario | Administrador |
| Nivel de educación | Profesional/conocimientos en programación |
| Experiencia | Gestión de sistemas de información |
| Actividades | - Configurar y ajustar los parámetros de funcionamiento del software. - Administración de Usuarios. |

| | |
|---------------------------|--|
| Tipo de usuario | Usuario |
| Nivel de educación | N.A. |
| Experiencia | Manejo básico de sistemas de información |
| Actividades | - Acceder como invitado o usuario registrado. - Realizar consultas. - Gestionar el perfil del usuario. |

2.4 Restricciones

- El uso de la página requiere de conexión a internet.
- Interfaces de usuario intuitivas.
- Debe ser funcional en los navegadores más comunes.
- La página se ejecuta con servidores de Python.
- Debe usarse Django como Framework.
- El aplicativo debe contar con un sistema de validación de sesión.
- Utilizar un Modelo Vista Controlador (MVC).
- Los servidores deben ser capaces de atender consultas concurrentemente.

2.5 Suposiciones y Dependencias

- Se debe contar con un administrador que garantice el correcto funcionamiento de la página y sus servicios.
- Se debe contar con acceso a internet en los equipos donde sea utilizada la página.
- La base de datos de los establecimientos y sus restricciones deben ser actualizadas constantemente.

2.6 Requerimientos Diferidos

| | |
|--------------------|--|
| Código | RD1 |
| Nombre | Interfaz de usuario |
| Descripción | Desarrollar una interfaz de usuario intuitiva y minimalista que sea atractiva para los usuarios. |
| Prioridad | Media |

| | |
|--------------------|---|
| Código | RD2 |
| Nombre | Seguridad de la información |
| Descripción | Se deberá garantizar la seguridad de los datos almacenados tanto por los usuarios como por los administradores. |
| Prioridad | Media |

| | |
|--------------------|---|
| Código | RD3 |
| Nombre | Disponibilidad |
| Descripción | La página deberá garantizar un servicio continuo (24/7) a los usuarios. |
| Prioridad | Media |

| | |
|--------------------|---|
| Código | RD4 |
| Nombre | Portabilidad |
| Descripción | La página web deberá ser compatible con sitios web tanto de computadores como de equipos móviles. |
| Prioridad | Baja |

3 REQUERIMIENTOS ESPECÍFICOS

3.1 Requerimientos

3.1.1 Product Backlog

| ORDEN | REQUERIMIENTOS A DESARROLLAR | TAREAS | NOTAS |
|-------|---|---|--|
| 1 | Selección de proyecto según banco de retos. | <ul style="list-style-type: none">Asignar roles de los integrantes del equipoIdear cómo llevar a cabo el desarrolloCrear el repositorio en GITHUB | Observamos la necesidad de dividir tareas y la importancia de planear cuidadosamente qué debemos hacer y cómo. Debido a esto cada integrante tiene una labor principal distinta, sin embargo, una vez finalice su trabajo debe |





| | | | |
|---|--|--|---|
| | | | colaborar con sus compañeros con el fin de agilizar y mejorar el producto final. |
| 2 | Aclaración sobre la estructura del proyecto en Django, ramas en Git y GitHub Flow. | <ul style="list-style-type: none"> Definir ramas en GitHub para no dañar el trabajo principal. Apoyo a las tareas de los demás compañeros. Hacer seguimientos y retrospectivas. Utilizar prácticas estandarizadas de gestión de proyectos. | Se aprendió a usar las ramas de GitHub. |
| 3 | Diseño de la base de datos | <ul style="list-style-type: none"> Creación tablas Creación de DB en PostgreSQL mediante Django Modelado de DB | Para hacer un CRUD básico con Django no es necesario sentencias SQL. Se puede trabajar con una BD relacional utilizando sentencias SQL o sin ellas gracias a Django |
| 4 | Desarrollo del Backend | <ul style="list-style-type: none"> Crear la estructura del proyecto mediante el Framework Django. Crear el archivo "requirements.txt" con las librerías utilizadas <ul style="list-style-type: none"> Realizar las configuraciones necesarias básicas Crear carpeta para las plantillas y archivos HTML (templates) Crear carpeta para archivos estáticos (static) | El desarrollador backend escribe el código que permite que el botón obtenga los datos correctos de la base de datos. Esos datos se devuelven al frontend, donde el desarrollador del frontend determina cómo se muestran al visitante. |
| 5 | Desarrollo del Frontend | <ul style="list-style-type: none"> Se declaran los estilos CSS Bootstrap. Se declara localmente estilo .js Bootstrap <ul style="list-style-type: none"> Página principal <ul style="list-style-type: none"> Login Acceso al usuario Actualizar datos <ul style="list-style-type: none"> Lugares Acerca de nosotros | A través de lenguajes de programación HTML, CSS y Bootstrap que se utilizaron para la creación del sitio web, se logró incluir todos los elementos gráficos como colores, fondos o animaciones que conforman el diseño y estructura de la |

| | | <ul style="list-style-type: none"> • Dónde encontrarnos | página, es decir la parte visual. |
|---|-------------------|---|--|
| 6 | Análisis y Testeo | <ul style="list-style-type: none"> • Usabilidad • Tiempo de carga • Accesibilidad • Visualización | En el momento de lanzar el sitio web se requiere de probar la experiencia de usuario, la funcionalidad de la ruta de conversión, preparar copias de seguridad. |

3.1.2 Ciclo de Sprints del proyecto

1. Sprint 1: Entrega en primera clase semana 4 (3 semanas para su desarrollo).
 - i. Repositorio (GitHub) donde integrarán el trabajo.
 - ii. Gestión de configuración.
 - iii. Historias de usuario desarrolladas en el Sprint 1. (Trello)
 - iv. Presentación MVC
 1. Interfases gráficas de usuario
 2. Navegación entre interfases
 - v. Informe de retrospectiva
 - vi. Historias de usuario a desarrollar en el sprint 2. (Trello)
2. Sprint 2: Entrega en primera clase semana 6 (2 semanas para su desarrollo)
 - 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. (Trello)
3. Sprint 3: Entrega en última clase semana 6 (1 semana para su desarrollo)
 - i. Presentación MVC
 1. Aplicación con persistencia Relacional
 2. Pruebas unitarias de la lógica desarrollada
 - ii. Informe de retrospectiva
 - iii. Historias de usuario a desarrollar en el sprint 4. (Trello)
4. Sprint 4: Entrega en última clase semana 7 (1 semana para su desarrollo)
 - i. Presentación MVC
 1. Despliegue Back-End en la nube
 2. Despliegue base de datos en local/nube
 - ii. Informe de retrospectiva

3.1.3 Sprint Backlog

| SPRINT | OBJETIVO DEL SPRINT | FINALIZADO |
|--------|--|---|
| 1 | <ol style="list-style-type: none"> 1. Asignar roles de los integrantes del equipo 2. Idear cómo llevar a cabo el desarrollo 3. Crear el repositorio en GITHUB 4. Definir ramas en GitHub para no dañar el trabajo principal. 5. Apoyo a las tareas de los demás compañeros. 6. Hacer seguimientos y retrospectivas. 7. Utilizar prácticas estandarizadas de gestión de proyectos. |  |
| 2 | <ol style="list-style-type: none"> 1. Creación tablas 2. Creación de DB en PostgreSQL mediante Django. 3. Modelado de DB 4. Crear la estructura del proyecto mediante el Framework Django. |  |
| 3 | <ol style="list-style-type: none"> 1. Crear el archivo "requirements.txt" con las librerías utilizadas 2. Realizar las configuraciones necesarias básicas 3. Crear carpeta para las plantillas y archivos HTML (templates) 4. Crear carpeta para archivos estáticos (static) 5. Se declaran los estilos CSS Bootstrap. 6. Se declara localmente estilo .js Bootstrap 7. Página principal 8. Login 9. Acceso al usuario 10. Actualizar datos 11. Lugares 12. Acerca de nosotros 13. Dónde encontrarnos |  |
| 4 | <ol style="list-style-type: none"> 1. Usabilidad 2. Tiempo de carga 3. Accesibilidad 4. Visualización <p>PROYECTO APROBADO POR EL EQUIPO DE TRABAJO Y TERMINADO.</p> |  |

3.1.4 Historias de usuario (Tareas y Subtareas)

Durante el desarrollo del proyecto se desarrollaron las siguientes historias de usuario, distribuidas en las diferentes entregas de cada sprint.

Sprint 1

| Historia: registro de usuario | |
|--|--|
| Como | Usuario. |
| Quiero | Acceder a la opción de registro. |
| Para | Utilizar las funciones adicionales que ofrece la página web. |
| Prioridad | Alta |
| Validación: <ul style="list-style-type: none">● Acceder a los servicios totales de la página.● Guardar, editar y eliminar el historial de búsquedas. | |

| Historia: registro con correo electrónico | |
|--|---|
| Como | Usuario |
| Quiero | Registrarme con mi correo |
| Para | Acceder a las opciones de usuario registrado. |
| Prioridad | Alta |
| Validación: <ul style="list-style-type: none">● Opción para registrarse con el correo electrónico.● Opción para acceder a la cuenta a través del correo electrónico. | |

| Historia: eliminar cuenta | |
|--|---|
| Como | Usuario registrado. |
| Quiero | Tener la opción de eliminar mi cuenta. |
| Para | Dejar de acceder a los servicios ofrecidos por la página. |
| Prioridad | Alta |
| Validación: <ul style="list-style-type: none"> • Opción para eliminar la cuenta. • Confirmar la eliminación de la cuenta. | |

Sprint 2

| Historia: editar datos | |
|--|--|
| Como | Usuario registrado. |
| Quiero | Una opción de edición. |
| Para | Editar mis datos personales por si se presenta algún cambio. |
| Prioridad | Media |
| Validación: <ul style="list-style-type: none"> • Opción para editar datos personales. • Permitir que se guarden los cambios realizados. | |

| Historia: realizar comentarios | |
|--|--|
| Como | Usuario registrado |
| Quiero | Realizar comentarios. |
| Para | Comunicar la experiencia del sitio al que se visitó. |
| Prioridad | Media |
| Validación: <ul style="list-style-type: none"> • Opción para editar datos personales. • Permitir a un usuario registrado crear comentarios. | |

| Historia: invitado | |
|---|---|
| Como | Invitado |
| Quiero | Acceder a servicios limitados de la página. |
| Para | Evitar registrarme y realizar una búsqueda rápida |
| Prioridad | Media |
| Validación: <ul style="list-style-type: none"> • Acceder a los servicios de la página. • Realizar búsquedas limitadas. | |

| Historia: política de tratamiento de datos | |
|---|---|
| Como | Usuario |
| Quiero | Conocer la política de tratamiento de datos personales. |
| Para | Decidir si realizó o no el registro en la página. |
| Prioridad | Media |
| Validación: <ul style="list-style-type: none"> • Opción para acceder a la política de tratamiento de datos. • Opción para aceptar dicha política al finalizar el registro en el sistema. | |

Sprint 3

| Historia: contraseña | |
|--|--|
| Como | Usuario registrado |
| Quiero | Acceder a la opción de cambio o recuperación de contraseña. |
| Para | Acceder a mi cuenta si olvide la contraseña o deseo cambiarla por seguridad. |
| Prioridad | Media |
| Validación: <ul style="list-style-type: none">• Generar una opción de olvido de contraseña.• Crear una opción de cambio de contraseña dentro del perfil. | |

| Historia: historial | |
|--|---|
| Como | Usuario registrado |
| Quiero | Visualizar los sitios buscados recientemente. |
| Para | Tener un historial y saber de manera rápida los lugares frecuentes. |
| Prioridad | Media |
| Validación: <ul style="list-style-type: none">• Generar una opción de lugares visitados.• Editar o eliminar el historial de lugares visitados. | |

| Historia: comentarios y sugerencias | |
|---|--|
| Como | Usuario registrado |
| Quiero | Enviar algún comentario al soporte de la página. |
| Para | Realizar algún tipo de sugerencia, queja o acción de mejora. |
| Prioridad | Media |
| Validación: <p>Acceder a la opción de contacto.</p> <p>Formulario para el envío de comentarios al soporte.</p> | |

Sprint 4

| Historia: contraseña | |
|---|--|
| Como | Usuario registrado |
| Quiero | Calificar un establecimiento |
| Para | Saber de manera visual la percepción que tienen otros usuarios sobre un determinado lugar. |
| Prioridad | Alta |
| Validación: | |
| <ul style="list-style-type: none">○ Generar un método de calificación.○ Permitir que sea intuitivo y fácil de interpretar. | |

| Historia: historial | |
|--|--|
| Como | Administrador |
| Quiero | Bloquear o suspender una cuenta de usuario |
| Para | Evitar que se generen comentarios ofensivos o denigrantes hacia una determinada persona o lugar. |
| Prioridad | Alta |
| Validación: | |
| <ul style="list-style-type: none">○ Permitir al administrador suspender las cuentas de algún usuario registrado.○ Crear una opción para notificar al usuario en caso de presentarse una suspensión. | |

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

| Fecha | Asistencia | Temas |
|------------|--------------|---|
| 22/08/2022 | 5 | Selección de proyecto según banco de retos, roles de los integrantes del equipo, ideas sobre cómo llevar a cabo el desarrollo, creación del repositorio en GITHUB |
| 25/08/2022 | 4 y el tutor | Informe del proyecto elegido al tutor, roles de los integrantes del equipo e ideas sobre cómo llevar a cabo el desarrollo, aclaración de dudas sobre el proyecto y los sprints. |

| | | |
|------------|--------------|--|
| 26/08/2022 | 5 | División de los entregables para el sprint 1 entre los integrantes, planeación de método de trabajo, tareas a realizar. |
| 27/08/2022 | 4 | Revisión de avances sprint 1, retroalimentación, aclaración de algunos temas, deadline. |
| 30/08/2022 | 5 | Aclaración sobre la estructura del proyecto en Django, ramas en Git y GitHub Flow. |
| 1/09/2022 | 5 | Resolución de dudas sobre el proyecto y análisis técnico del mismo. |
| 3/09/2022 | 5 | División de tareas para el sprint 2. |
| 6/09/2022 | 5 | Análisis de la estructura del proyecto. |
| 8/09/2022 | 5 y el tutor | Explicación sobre los entregables de los siguientes sprint y revisión de la estructura del proyecto. |
| 10/09/2022 | 5 | Aclaración de los pasos a seguir para la ejecución local del proyecto con GitHub. |
| 13/09/2022 | 5 | Ejecución del proyecto en el servidor de Python y análisis de los errores presentados en la interfaz gráfica. |
| 15/09/2022 | 3 y el tutor | Análisis de los avances realizados en el diseño y la programación de la página web. |
| 17/09/2022 | 5 | Análisis de errores presentados en la página. |
| 20/09/2022 | 5 | Discusión sobre cambios en la estructura del proyecto. |
| 22/09/2022 | 3 y el tutor | Aclaración de dudas sobre aspectos técnicos del proyecto. |
| 24/09/2022 | 5 | Cambios en el diseño y mejoras generales de la página. |
| 27/09/2022 | 5 | Análisis de errores presentados en la página. |
| 29/09/2022 | 5 y el tutor | Revisión general de la página web hasta el momento y aclaración de dudas sobre el documento final y la sustentación del mismo. |
| 1/10/2022 | 5 | Retroalimentación del proyecto y del ciclo 3 en general. |

4 DESCRIPCIÓN DEL DISEÑO

4.1 Interfaz gráfica (Mockups)

Todo el proyecto usa código HTML, CSS y Bootstrap.

1. Se declaran los estilos CSS Bootstrap.

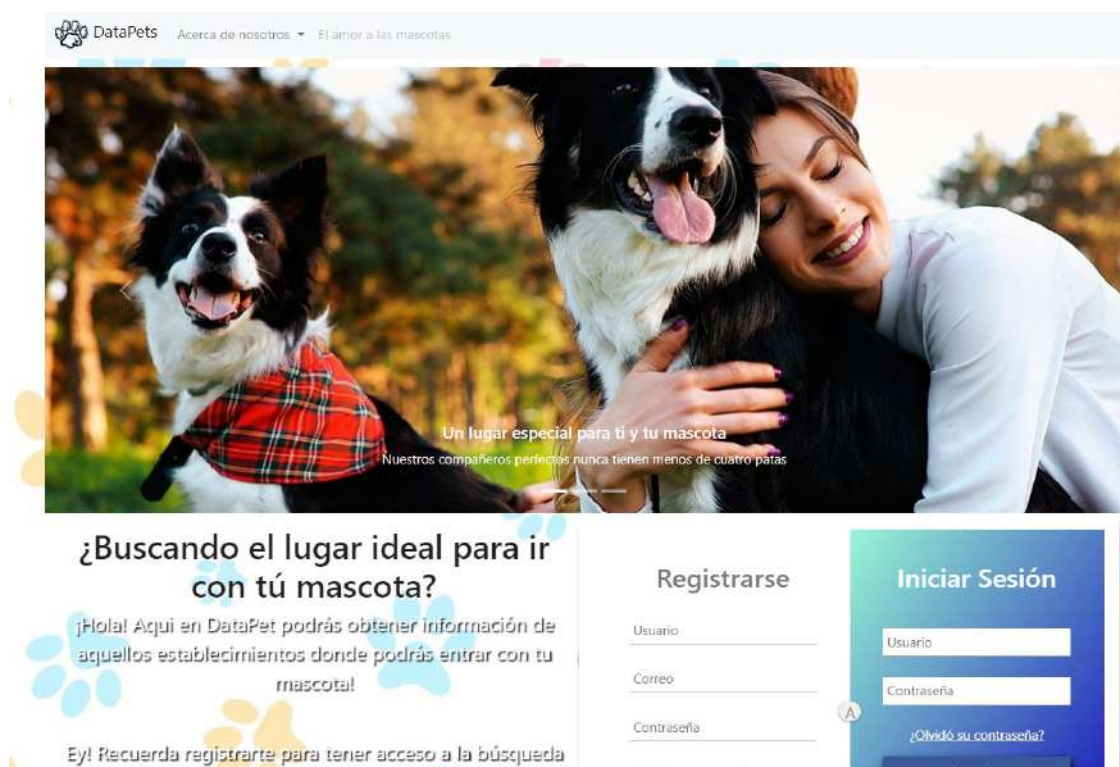
```
head.html X
templates > partials > head.html > link
1  {% load static %}
2  <meta charset="UTF-8">
3  <meta content="IE=edge">
4  <meta name="viewport" content="width=device-width, initial-scale=1.0">
5  <link rel="shortcut icon" href="{% static 'assets/img/huellal.png' %}">
6  <link rel="stylesheet" href="{% static 'assets/css/styles.css' %}">
7  <link rel="stylesheet" href="{% static 'assets/css/comments.css' %}">
8  <link rel="stylesheet" href="{% static 'assets/css/about.css' %}">
9  <link rel="stylesheet" href="{% static 'assets/css/user.css' %}">
10 <link rel="stylesheet" href="{% static 'assets/css/places.css' %}">
11 <link rel="stylesheet" href="{% static 'vendor/bootstrap/css/bootstrap.min.css' %}">
12 <!-- CSS only -->
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
```

2. Se declara localmente estilo .js Bootstrap.

```
layout.html X
templates > layout.html > html
4
5 <head>
6   {% include 'partials/head.html' %}
7   <title>{% block title %}{% endblock title %}</title>
8 </head>
9
10 <body class="container">
11
12   <header>
13     {% include 'partials/nav.html' %}
14   </header>
15
16   <main>
17     {% block main %}
18
19     {% endblock main %}
20   </main>
21
22   {% include 'partials/footer.html' %}
23
24   <!-- <script src="{% static 'assets/js/script.js' %}"></script> -->
25   <!-- la ruta de la carpeta static está predefinida, sólo es completar el resto de la dirección
26   <script src="{% static 'vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>
27
28 </body>
```

3. La página principal.

```
index.html X
templates > main > index.html > div.FC1.mx-1 > div.p-3 > p.text-start2
4
5
6 {% block main %}
7
8 {% include 'main/index/carousel.html' %}
9
10 <div class="FC1 mx-1">
11   <a href="#login_section"></a>
12   <div class="p-3">
13     <h1>¿Buscando el lugar ideal para ir con tú mascota?</h1>
14     <p class="text-start2">¡Hola! Aquí en DataPet podrás obtener información de aquellos establecim
15     entrar con tu mascota!</p><br>
16     <p class="text-start2">Ey! Recuerda registrarte para tener acceso a la búsqueda detallada para
17     especiales con tu mascota</p>
18   </div>
19
20   {% include 'user/login.html' %}
21
22 </div>
23
24 <h1 class="pt-5">Reseñas</h1><br>
25
26 {% include 'main/index/reviews.html' %}
27
28 {% endblock main %}
```

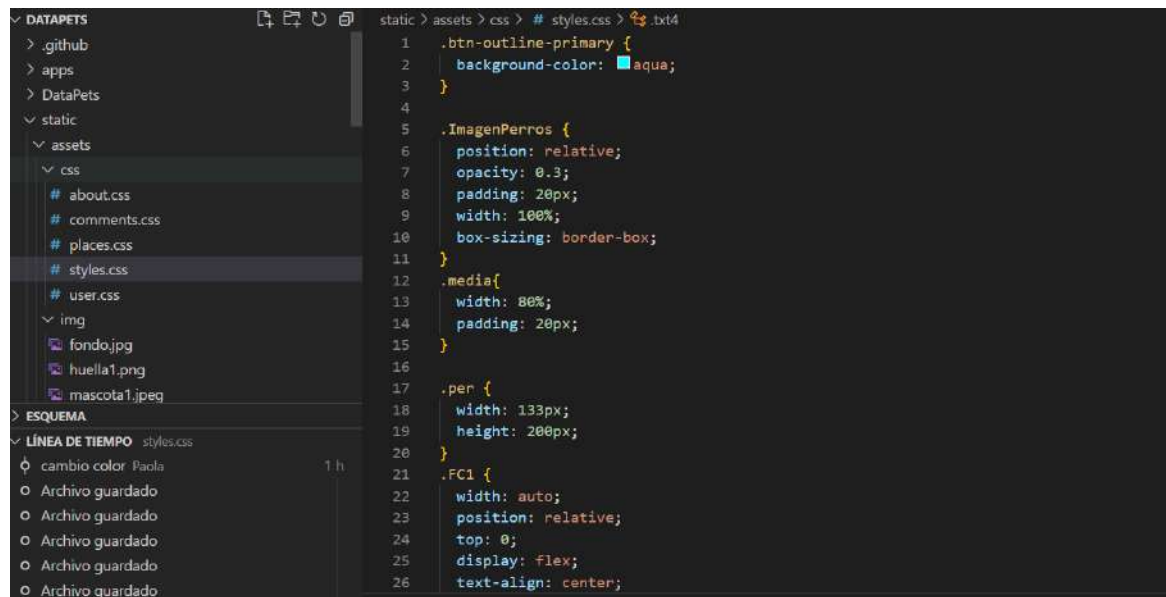


4. La parte del Login.

```
login.html X
templates > user > login.html > div > div.container1 > div.login-container > div.login
1  <div>
2      <div class="container1">
3          <div class="login-container">
4              <div class="register">
5                  <h2>Registrarse</h2>
6                  <form action="/sign-up" method="post">
7                      {% csrf_token %}
8                      <input name="username" type="text" placeholder="Usuario">
9                      <input name="email" type="text" placeholder="Correo">
10                     <input name="password1" type="password" placeholder="Contraseña">
11                     <input name="password2" type="password" placeholder="Confirma contraseña">
12                     <input type="submit" class="submit" value="REGISTRARSE">
13                 </form>
14             </div>
15             <div class="login">
16                 <h2>Iniciar Sesión</h2>
17                 <div class="login-items">
18                     <form action="/sign-in" method="post">
19                         {% csrf_token %}
20                         <input name="username" type="text" placeholder="Usuario">
21                         <input name="password" type="password" placeholder="Contraseña">
22                         <a href="password_reset/" class="text-sm-start">¿Olvidó su contraseña?</a>
23                         <div class="form-group">
24                             {% for message in messages %}
25                             <div style="color: red;">
26                                 {{message}}
```

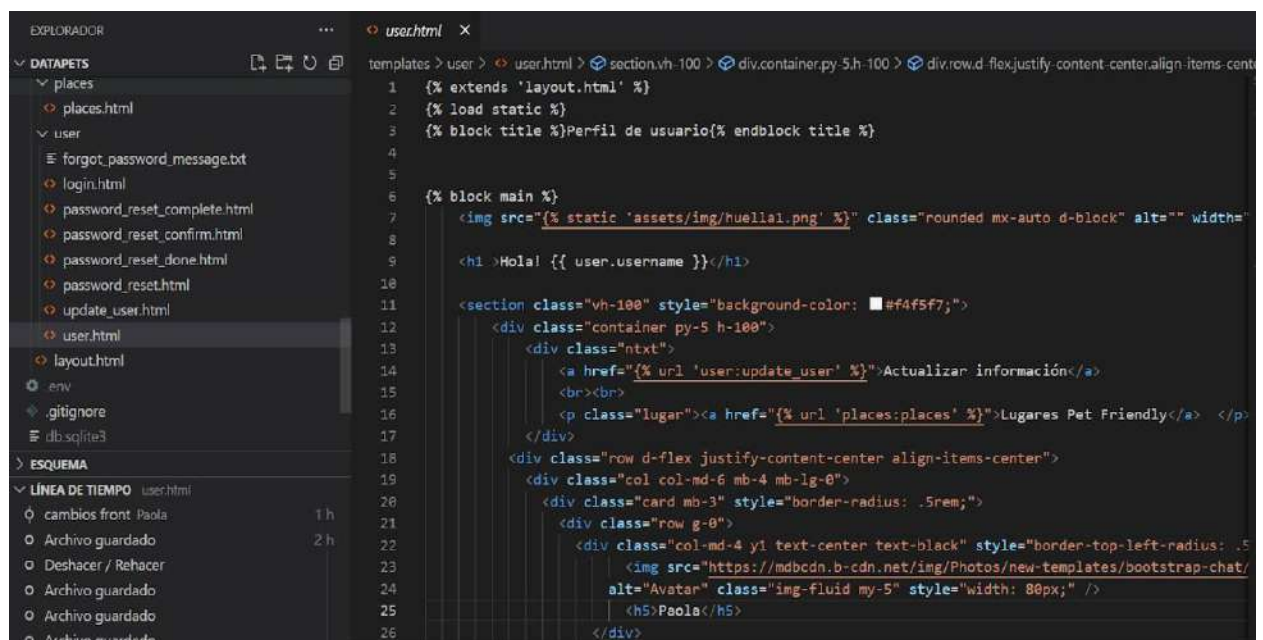
Se implementó un contenedor principal que tuviera 2 contenedores “Registrarse” e “Iniciar sesión”.

The image shows a web interface with two distinct login sections. On the left, the 'Registrarse' (Register) section features a white background with a light blue border. It contains four input fields: 'Usuario', 'Correo', 'Contraseña', and 'Confirma contraseña'. Below these fields is an orange button labeled 'REGISTRARSE'. On the right, the 'Iniciar Sesión' (Login) section has a blue gradient background. It includes input fields for 'Usuario' and 'Contraseña', a link that says '¿Olvidó su contraseña?', and a blue button labeled 'Acceder'. A small circular icon with the letter 'A' is positioned between the two forms.

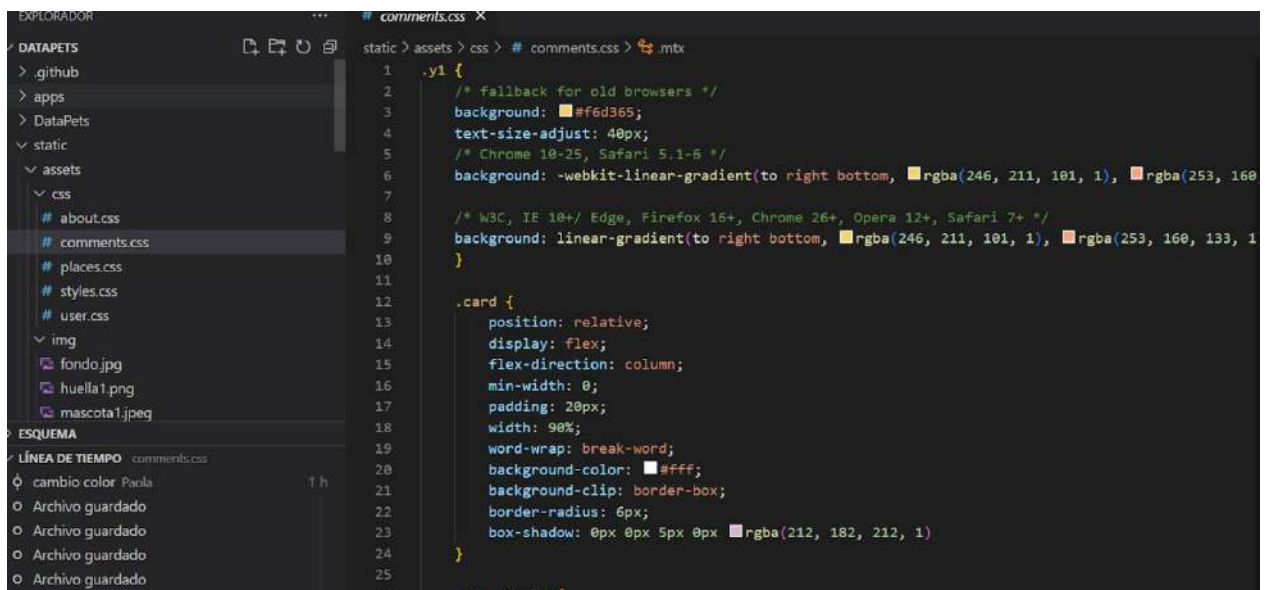
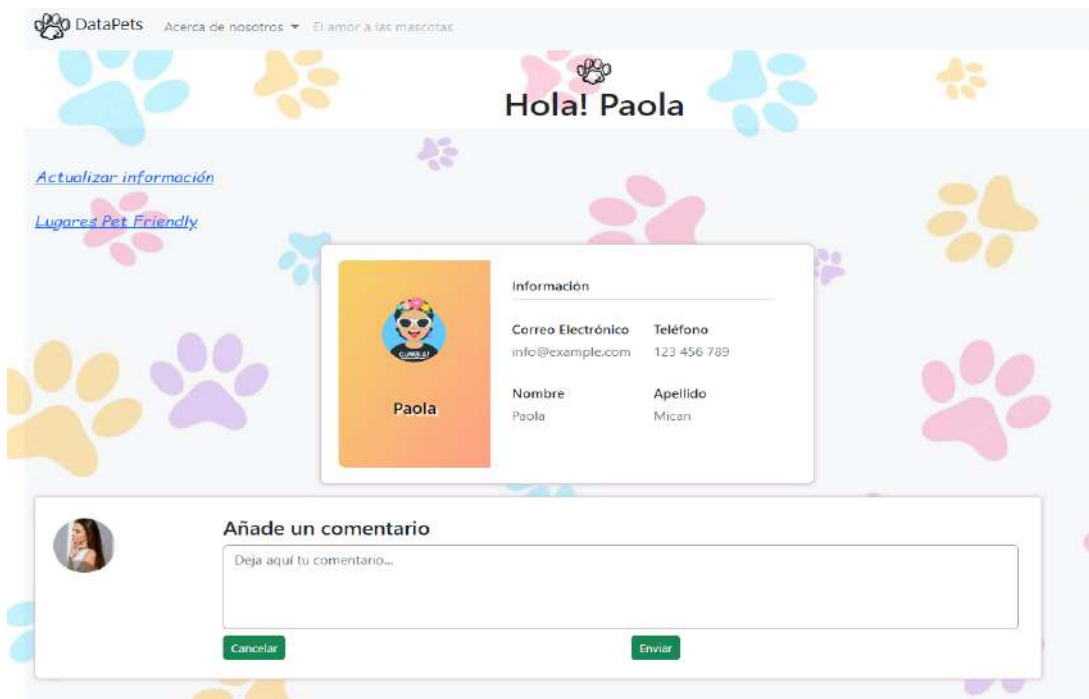


En la sección **styles.css** están los estilos usados para la página principal y el cuadro del login.

5. Acceso del usuario.



Se trabajaron 2 contenedores dentro de un contenedor principal *div class= "container"*.





En la parte de **comments.css** se dejó los estilos implementados en la sección de usuario.

6. Actualizar Datos.

```
update_user.html X
templates > user > update_user.html > ...
7      <p class="text-center txt4">Actualizar Datos</p>
8
9      <div class="container3">
10         <form method="post">
11             <div class="form-group">
12                 <label>Usuario</label>
13                 <input type="text" name="Usuario" placeholder="Nombre de Usuario"
14                     class="form-control" required>
15             </div>
16             <div class="form-group">
17                 <label>Nombres</label>
18                 <input type="text" name="Correo" placeholder="Nombres"
19                     class="form-control" required>
20             </div>
21             <div class="form-group">
22                 <label>Apellidos</label>
23                 <input type="text" name="Correo" placeholder="Apellidos"
24                     class="form-control" required>
25             </div>
26             <div class="form-group">
27                 <label>Teléfono</label>
28                 <input type="number" name="Correo" placeholder="Número de Teléfono"
29                     class="form-control" required>
30             </div>
31             <div class="form-group">
32                 <label>Contraseña</label>
```

Se crea el contenedor para la parte de modificación de datos por parte del usuario.


DataPets
Acerca de nosotros
El amor a las mascotas



Actualizar Datos

Usuario

Nombres

Apellidos

Teléfono

Contraseña

Guardar


© 2022 DataPets




7. Lugares Pet Friendly.

```

<> places.html x
templates > places > <> places.html > div.container2
13 <div class="container2">
14   <form method="post">
15     <div class="form-group">
16       <label>Lugar</label>
17       <input type="text" name="Usuario" placeholder="Nombre del Lugar"
18         class="form-control" required>
19     </div>
20     <div class="form-group">
21       <label>Dirección</label>
22       <input type="text" name="Correo" placeholder="Dirección"
23         class="form-control" required>
24     </div>
25     <div class="form-group">
26       <label>Horario de atención</label>
27       <input type="text" name="Correo" placeholder="Horario"
28         class="form-control" required>
29     </div>
30     <div class="form-group">
31       <label>Teléfono</label>
32       <input type="number" name="Correo" placeholder="Número de Teléfono"
33         class="form-control" required>
34     </div>
35
36     <div class="form-group">
37       <label>Fecha</label>
38       <input type="date" name="Fecha" placeholder="Fecha de inicio">
39     </div>
40   </form>
41 </div>

```


DataPets Acerca de nosotros El amor a las mascotas

Sitios que puedes visitar con tu peludito

Lugar

Dirección

Horario de atención

Teléfono

Fecha

Guardar

```

.container2 {
  display: block;
  padding: 20px;
  width: 100%;
  height: auto;
  background: #b0c4de;
  box-shadow: 6px 20px 10px #666666;
}

.container3 {
  display: block;
  padding: 20px;
  width: 100%;
  height: auto;
  background: #d9e1f2;
  box-shadow: 6px 20px 10px #666666;
}

.txt4 {
  font-style: italic;
  color: #000080;
  font-family: cursive;
  font-size: 50px;
  text-shadow: 4px 2px 5px #000000;
}

```

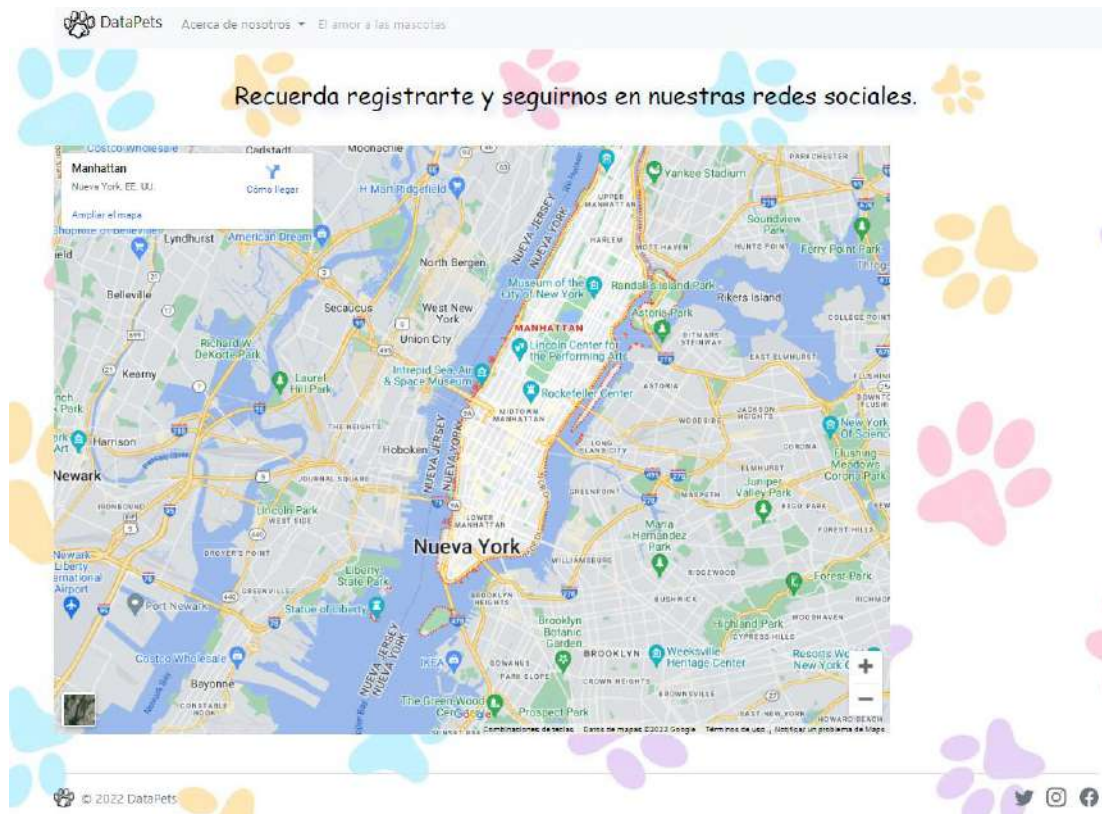
Los estilos utilizados para el diseño de Actualización de datos, Lugares Pet Friendly y texto de la página.

8. Acerca de Nosotros



```
about_1.html X
templates > main > about_1.html > div.container.H-1
4
5
6 {% block main %}
7
8
9
10 <p class="text-center tcolor">Acerca de Nosotros!</p>
11
12 <div class="container H-1" >
13   <div class="flx2">
14     <br>
15     <!-- 
17     <p class="fw-semibold">¡Hola! DataPets es un espacio dedicado para aquellas personas que sc
18     <br>
19     <br>
20   </div>
21   <div class="embed-responsive embed-responsive-16by9 video">
22     <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/rsTLyukvxGU?start=
23   </div>
24 </div>
25
26
```

9. Donde encontrarnos.



```
> about_2.html X
templates > main > > about_2.html > > br
1  {% extends 'layout.html' %}
2  {% load static %}
3  {% block title %}¿Dónde encontrarnos?{% endblock title %}
4
5
6
7  {% block main %}
8  <br>
9  <p class="text-center ab2"> Recuerda registrarte y seguirmos en nuestras redes sociales. </p>
10 <br>
11 <div id="map-container-google-1" class="z-depth-1-half map-container" style="height: 500px">
12   <iframe src="https://maps.google.com/maps?q=manhatan&t=&z=13&ie=UTF8&iwloc=&output=embed"
13     style="border:0" allowfullscreen></iframe>
14   </div>
15
16   <br>
17
18 {% endblock main %}
```

La página es completamente responsive.

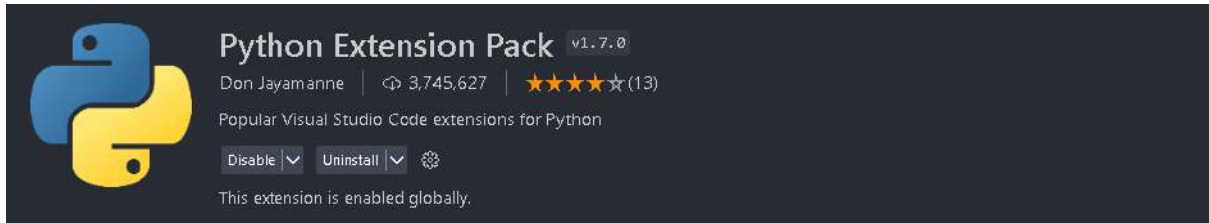
5 GESTIÓN DE LA CONFIGURACIÓN

1. Editor de código

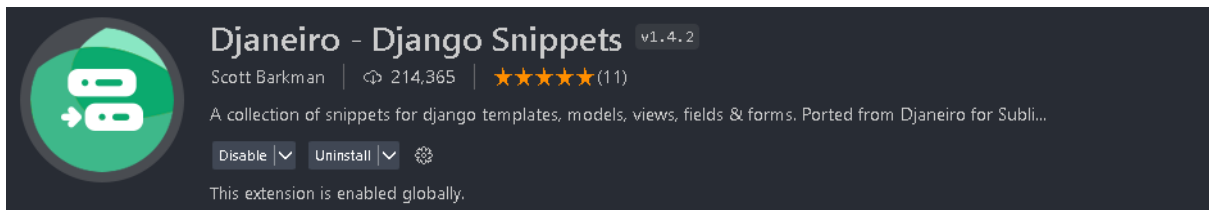
1.1. Visual Studio Code

1.1.1. Extensiones:

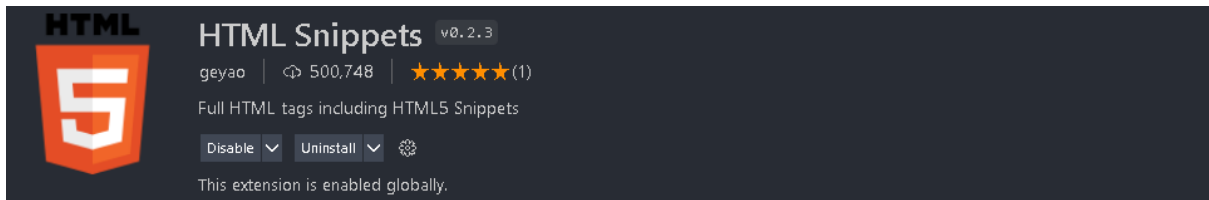
Python Extension Pack. Paquete de extensiones para facilitar la escritura de código con Python.



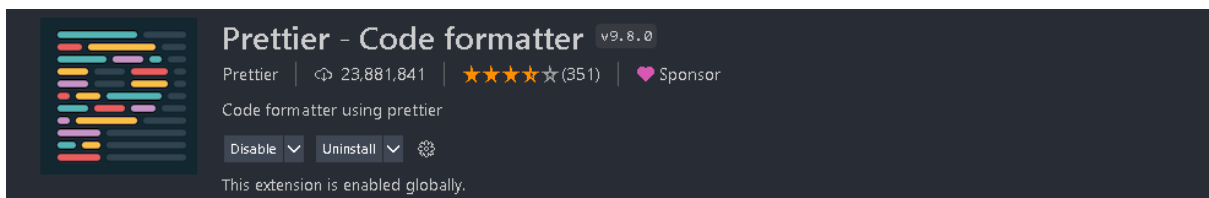
Djaneiro - Django Snippets. Extensión que permite utilizar *snippets* para *Django*.



HTML Snippets: autocompletar código HTML (*snippets*).

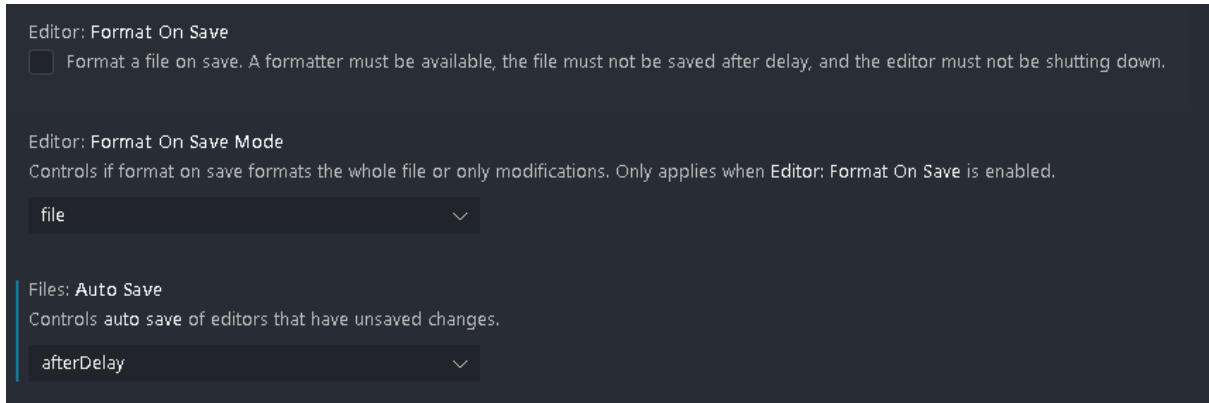


Prettier - Code formatter: formatear el código.



1.1.2. Configuración:

Los archivos se guardan automáticamente mientras se escribe, el código se formatea manualmente con click derecho sobre el archivo > formatear código, por medio de la extensión *Prettier*.



La terminal por defecto para ejecutar el servidor del proyecto Django es *Git Bash*.



0. Lenguajes

2.1. Python

2.1.1. Versión

Python 3.10

2.1.2. Bibliotecas (requirements.txt)

```
• asgiref==3.5.2
• Brotli==1.0.9
• dj-database-url==1.0.0
• Django==4.1
• django-crispy-forms==1.14.0
• gunicorn==20.1.0
• Pillow==9.2.0
• psycopg2==2.9.3
```

- `python-decouple==3.6`
- `sqlparse==0.4.2`
- `tzdata==2022.2`
- `whitenoise==6.2.0`

2.2. HTML

2.2.1. Versión HTML5

2.3. CSS

2.3.1. Versión CSS3

2.4. JavaScript

2.4.1. Versión EcmaScript2022

0. Frameworks

3.1. Bootstrap

3.1.1. Versión Bootstrap 5

0. Gestor de Base de Datos

4.1. PostgreSQL. Sistema gestor de bases de datos relacionales. Mediante modelos de Django se crean las diferentes tablas a utilizar.

0. Controlador de versiones

6.1. Git

6.1.1. Configuración

Se excluyen los siguientes archivos y carpetas del registro de versiones de git mediante el archivo “.gitignore” que contiene los siguiente:

```
/staticfiles  
*.sqlite3  
*.env  
*.pyc
```

Se establece el nombre de usuario y correo por medio de la terminal de *Git Bash* con los siguientes comandos:

```
$ git config user.name "<Nombre del integrante>"
$ git config user.email "<Correo del integrante>"
```

0. Alojamiento del repositorio

7.1. GitHub

7.1.1. Configuración local

El documento “.env” que no está siendo registrado por el controlador de versiones (Git) contiene información sensible del proyecto, tiene la siguiente estructura:

```
SECRET_KEY = <SECRET_KEY>

DEBUG = <DEBUG>

DATABASE_URL = <DATABASE_URL>

NAME = <NAME>
USER = <USER>
PASSWORD = <PASSWORD>
HOST = <HOST>
PORT = <PORT>

EMAIL_HOST_PASSWORD = <EMAIL_HOST_PASSWORD>
```

El proyecto accede a estos datos por medio de la biblioteca *python-decouple*.

7.1.2. Configuración remota

Se implementa *GitHub actions* con Django para ver el estado en línea del proyecto, el archivo de configuración que se encuentra en la ruta “.github/workflows/tests.yml” contiene lo siguiente:

```
name: Django CI

on:
  push:
    branches: [ "main", "frontend", "backend" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:
```



```

runs-on: ubuntu-latest
strategy:
  max-parallel: 4
  matrix:
    python-version: ['3.8', '3.9', '3.10']

steps:
- uses: actions/checkout@v3
- uses: actions/setup-python@v4
- name: Set up Python ${ matrix.python-version }
  uses: actions/setup-python@v3
  with:
    python-version: ${ matrix.python-version }
- name: Install Dependencies
  run: |
    python -m pip install --upgrade pip
    pip install -r requirements.txt
- name: Run Tests
  env:
    SECRET_KEY: ${ secrets.SECRET_KEY }
    DEBUG: ${ secrets.DEBUG }
    DATABASE_URL: ${ secrets.DATABASE_URL }
    NAME: ${ secrets.NAME }
    USER: ${ secrets.USER }
    PASSWORD: ${ secrets.PASSWORD }
    HOST: ${ secrets.HOST }
    PORT: ${ secrets.PORT }
    EMAIL_HOST_PASSWORD: ${ secrets.EMAIL_HOST_PASSWORD }
  run: |
    python manage.py test

```

Con esto se automatiza el proceso de verificación del proyecto en las ramas main, backend y frontend. También se prueba la compatibilidad con diferentes versiones de Python. Los datos sensibles almacenados en el documento “.env” se incluyen en los secretos del repositorio en GitHub para que la verificación de estado sea exitosa.

La rama “main” se encuentra protegida contra cambios, para agregar nuevo código se debe utilizar una *pull request*, la cual requiere la aprobación de 4 integrantes para realizar el respectivo merge, con esto se implementa el modelo de trabajo GitHub Workflow.

0. Hosting o despliegue (deploy)

Tanto el proyecto Django como la base de datos, están alojados en Heroku. Se establece la versión de Python 3.10.4 para el deploy en Heroku mediante el documento “runtime.txt” ubicado en la raíz del proyecto.

El archivo Procfile nos sirve para declarar explícitamente los tipos de proceso y los puntos de entrada de la aplicación, contiene lo siguiente:

```
web: gunicorn DataPets.wsgi:application --log-file -
```

Este documento depende de la biblioteca “gunicorn” que se encuentra en el documento requirements.txt.

El repositorio de GitHub está enlazado con Heroku para realizar el deploy cada vez que haya cambios nuevos en la rama principal “main”.

6 PRUEBAS

6.1 Descripción de pruebas unitarias

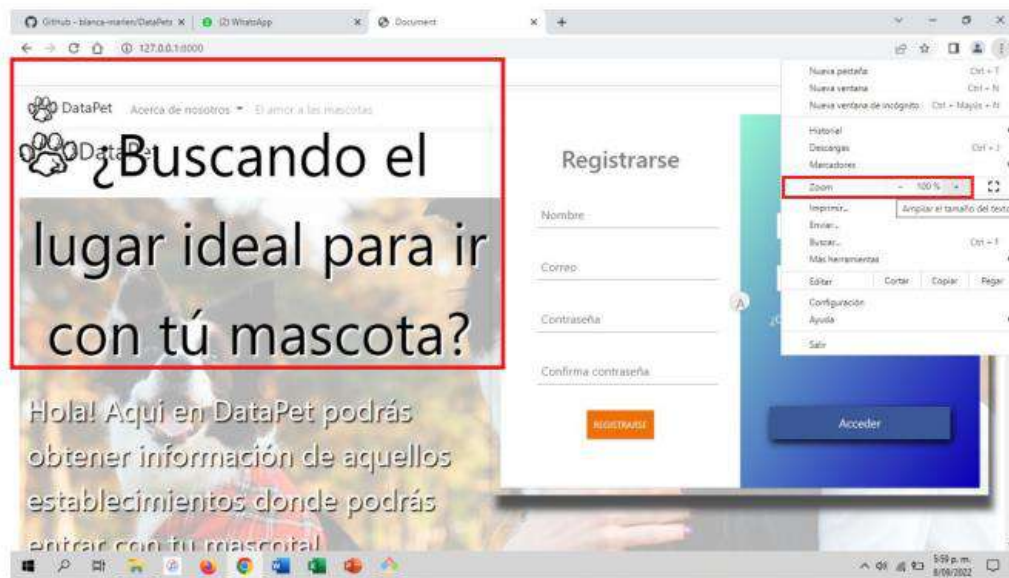
Pruebas en el Front-end:

- Violar las validaciones.
- Navegar a páginas con contenido no permitido (Rol).
- Cancelar la carga de la página durante procesos cruciales.
- Exigir la carga de la página con altas cantidades de controles dinámicos.
- Alterar peticiones http (modificar URL).

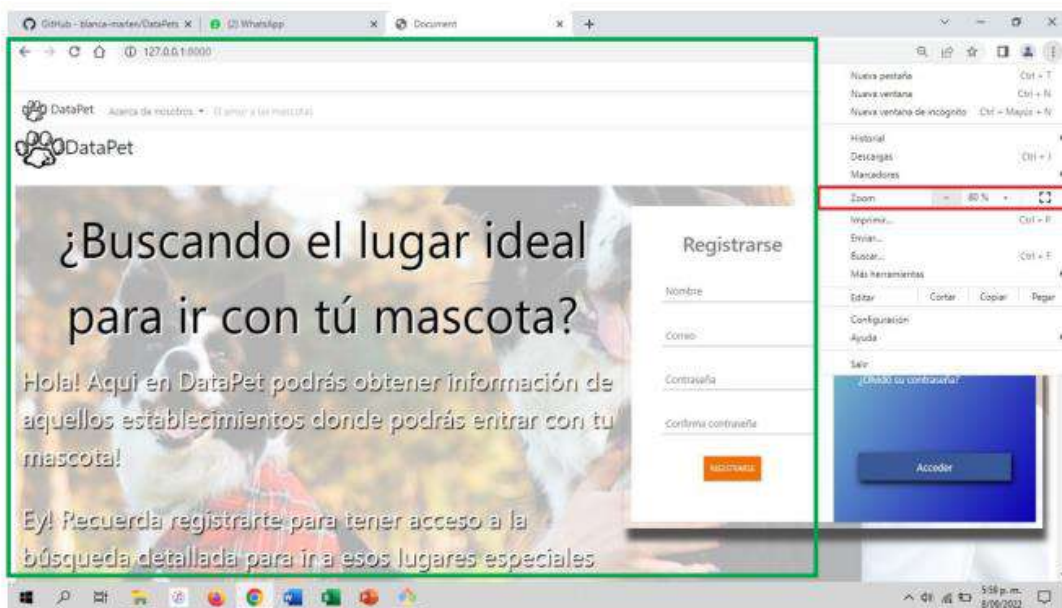
Pruebas en el Back-end:

- Reiterar llamadas a servicios.
- Enviar parámetros vacíos o fuera del rango.
- Enviar atributos nulos no permitidos (nativos).
- Negar acceso al API a usuarios no autorizados.

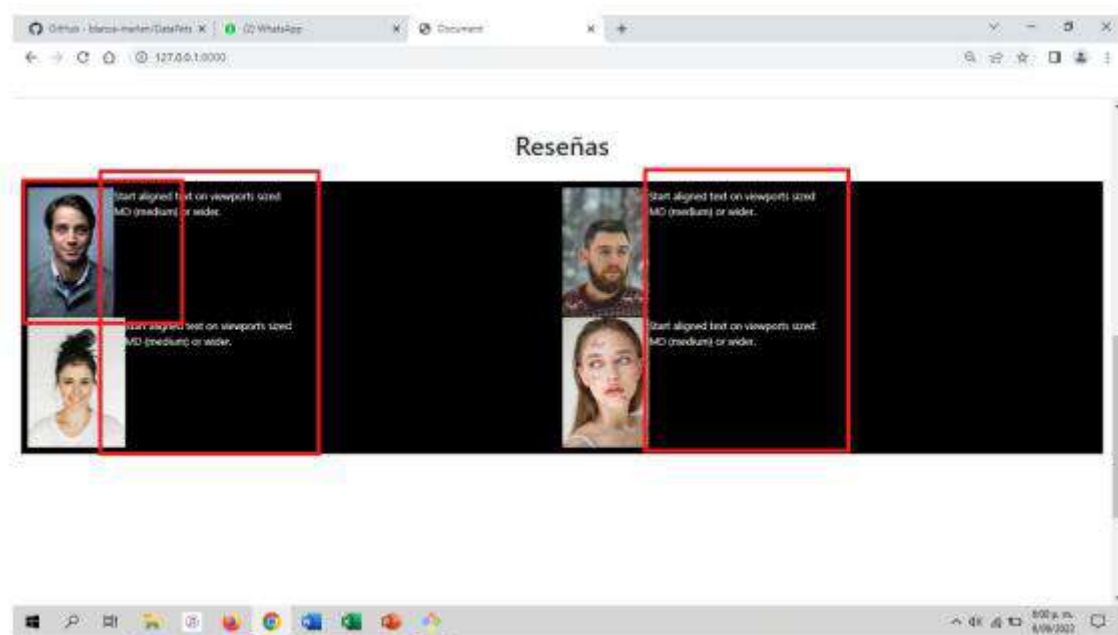
Pruebas realizadas a lo largo del desarrollo del proyecto



Al ingresar a la página, se evidencia que las letras se desubican de lugar, mostrándose aumentadas.

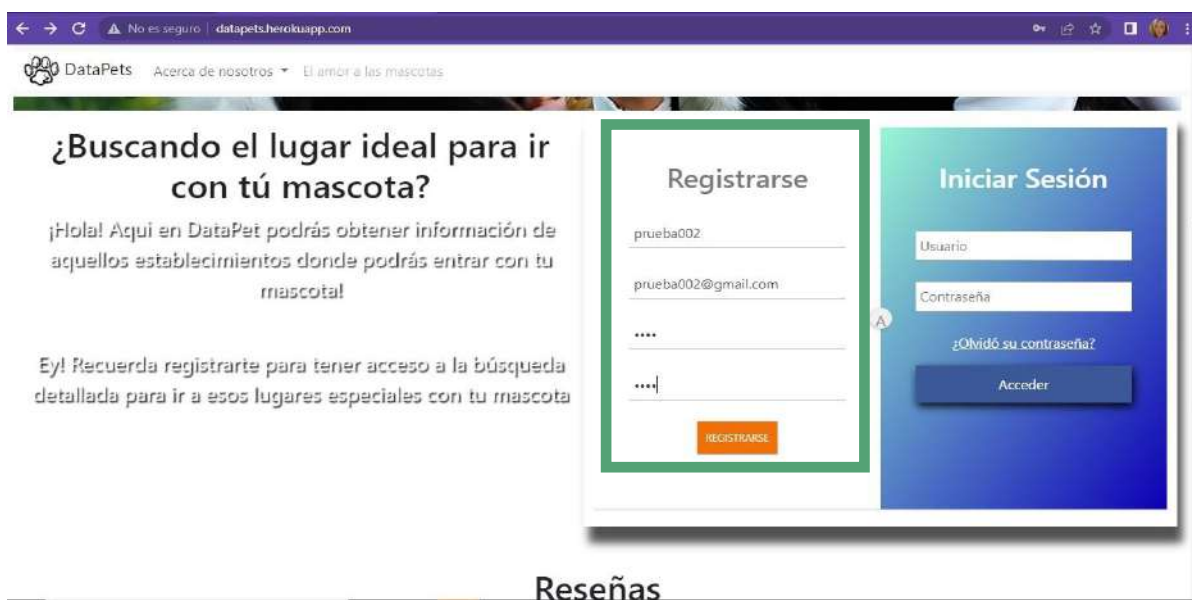


Cuando el zoom de la pantalla se coloca al 80 % ya se pueden observar las letras en su lugar.

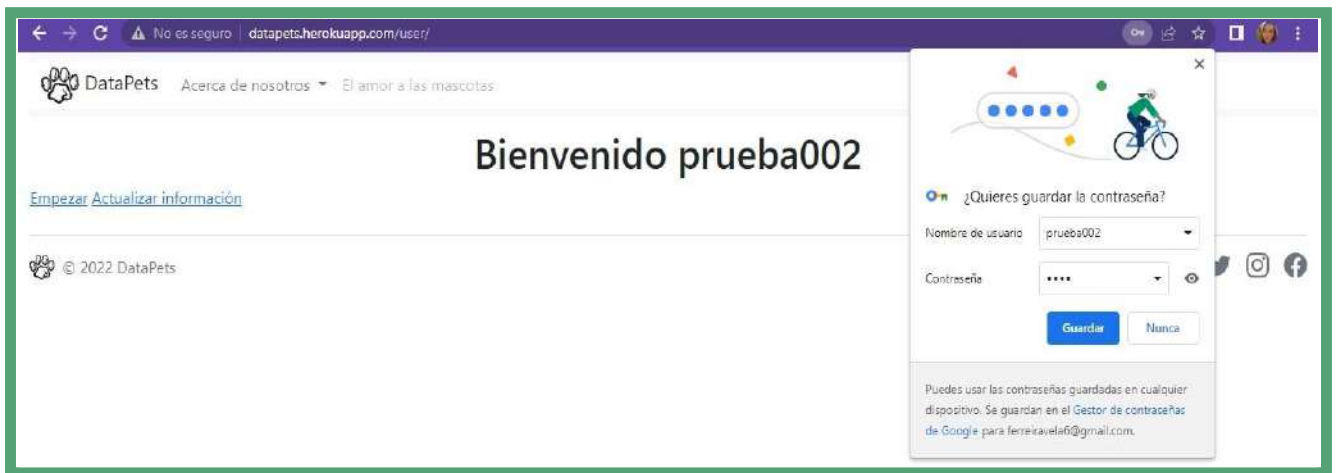


Sugerencia:

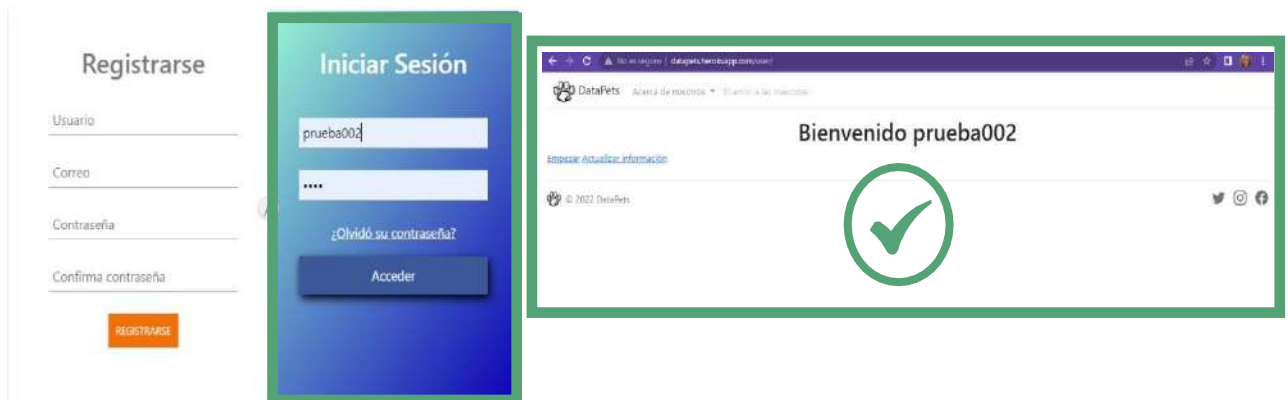
1. Que las imágenes sean del mismo tamaño.
2. Que el texto esté separado de las imágenes.



1. Se llenan todos los campos necesarios para crear un nuevo usuario.

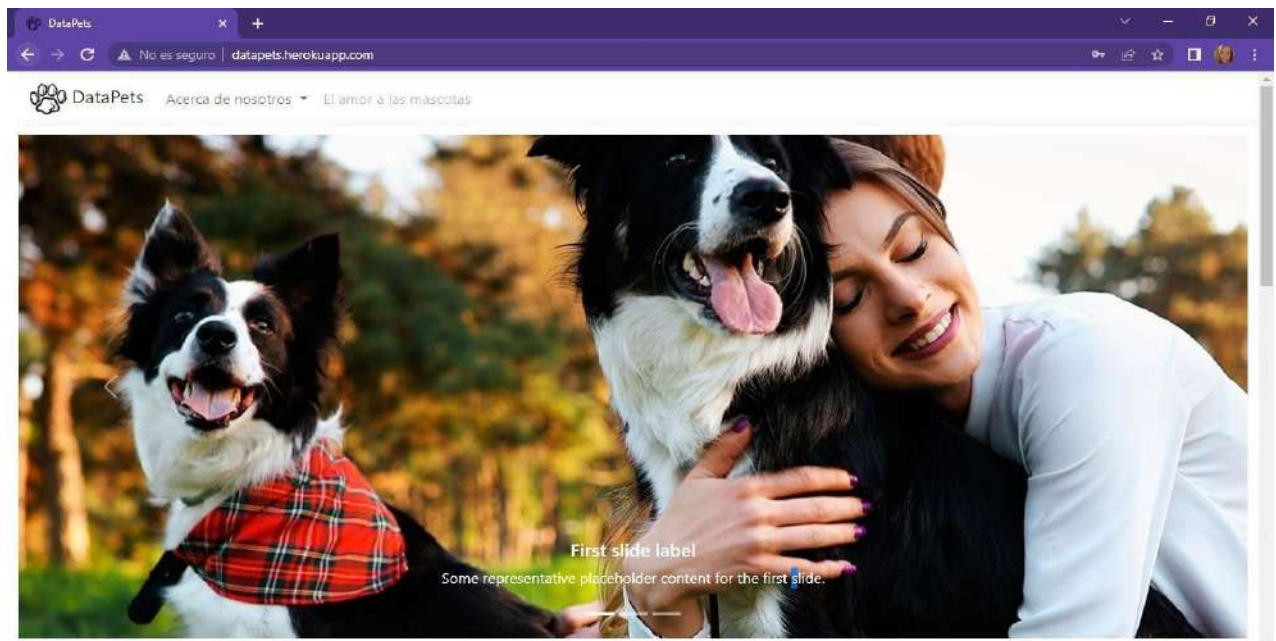


2. El usuario es creado exitosamente con los datos ingresados.



3. Se ingresa exitosamente al sistema con el usuario y la contraseña.

CAMINO FELIZ 



¿Buscando el lugar ideal para ir con tú mascota?

¡Hola! Aquí en DataPet podrás obtener información de aquellos establecimientos donde podrás entrar con tu mascota!

Ey! Recuerda registrarte para tener acceso a la búsqueda detallada para ir a esos lugares especiales con tu mascota

Registrarse

REGISTRARSE

Iniciar Sesión

[¿Olvidó su contraseña?](#)

Acceder

Reseñas



Start aligned text on viewports sized MD (medium) or wider.



Start aligned text on viewports sized MD (medium) or wider.



Start aligned text on viewports sized MD (medium) or wider.



Start aligned text on viewports sized MD (medium) or wider.

Los errores que se habían evidenciado en la versión anterior, fueron solucionados en esta nueva versión. La vista de inicio de la página cuenta con un diseño adecuado y funcional.

Fig. 1 shows two side-by-side login forms. The left form, titled 'Registrarse', has fields for 'Usuario', 'Correo', 'Contraseña', and 'Confirma contraseña', with an orange 'REGISTRAR' button at the bottom. A red circle with a cursor icon highlights the button. The right form, titled 'Iniciar Sesión', has fields for 'Usuario' and 'Contraseña', a link '¿Olvidó su contraseña?', and a blue 'Acceder' button.

Fig. 1



Fig. 2

Cuando se da click en el botón “REGISTRAR” (Fig. 1.) sin haber ingresado algún campo, se redirige a la página que se observa en la (Fig.2), se sugiere que se validen cada uno de los campos y se muestre un mensaje más informativo para el usuario en caso de que algún campo esté erróneo.

Fig. 3 shows the same two forms as Fig. 1, but with error messages. The 'Registrarse' form has an orange 'REGISTRARSE' button. The 'Iniciar Sesión' form has a blue 'Acceder' button and a red error message: 'Nombre de usuario o contraseña inválido'. A red box highlights the error message.

Fig. 3

Cuando se da click en el botón “ACCEDER” (Fig. 3.) válida correctamente los campos y muestra un mensaje informativo.

Recomendable que las letras tengan un color legible.

6.2 Descripción de pruebas de aceptación

Dentro de las pruebas de aceptación se encuentran:

- Pruebas de respaldo y recuperación.
- Recuperación ante desastres.
- Gestión de cuentas de usuario y control de acceso al sistema.
- Tareas de mantenimiento.
- Tareas de carga y migración de datos.
- Revisiones periódicas de vulnerabilidades de seguridad.

7 GLOSARIO

- **Algoritmo:** es básicamente un conjunto de pasos para una determinada tarea.
- **Back-End:** es lo que el usuario no ve en un software, como servidores y base de datos. Es un procesamiento “detrás de escena”.
- **Bug:** problema en el código que hace que no ejecute correctamente su función.
- **Clase:** proyecto de todo objeto.
- **Código:** conjunto de palabras o símbolos que contienen instrucciones para la computadora.
- **Código fuente:** código de un programa. La computadora lo usa para generar una versión del programa que es capaz de procesar.
- **Código Máquina:** código que la máquina puede entender y ejecutar.
- **Código Binario:** código formado sólo por 0 y 1 que tiene como función codificar algo. Es la base de toda la lógica de la computadora. Todo se convierte, al final, en código binario.
- **Frameworks:** son como colecciones de herramientas, componentes y soluciones que puedes encontrar en un mismo paquete (el Framework) que facilitan la creación de aplicaciones complejas.
- **Front-End:** Es la parte de una aplicación web que la persona que la usa, ve e interactúa. Es ejecutada por el browser.
- **Full-stack:** unión de Front-End y Back-End.
- **Herencia:** capacidad de heredar las características de otra clase.

- **Instancia:** la ejecución de una clase.
- **Iteración:** iteración es el nombre de estructuras que repiten el mismo bloque de código por un número finito de veces, o por el tiempo que una condición es verdadera. Se puede decir que todos los lenguajes de programación de computadoras tienen al menos una estructura de iteración.
- **Lenguaje:** conjunto de definiciones (sintácticas y semánticas) utilizadas para hacer un texto que la computadora sea capaz de entender, es decir, un programa de computadora.
- **Miembro de Clase:** puede ser una función o una propiedad (variable).
- **Objeto:** la ejecución de una clase.
- **Paradigma de programación:** modelo de estructura utilizado por el lenguaje de programación. Hay varios paradigmas existentes y cada lenguaje utiliza uno o varios paradigmas.
- **Polimorfismo:** capacidad de tener funciones con el mismo nombre y firma, pero con diferentes comportamientos.
- **Propiedad:** característica del objeto y en programación, se representa mediante una variable.
- **Queries:** solicitud de información y/o datos para una base de datos.
- **Refactorización:** proceso de cambiar un programa para mejorar su estructura interna sin cambiar su funcionamiento. Estas mejoras pueden ser para el rendimiento o para facilitar futuros trabajos en el código, por ejemplo.
- **SSL (Secure Sockets Layer):** es un tipo de seguridad estándar para garantizar que todos los datos estén encriptados en tránsito entre un sitio y un navegador, por ejemplo.
- **SuperClase:** la clase padre; clase que se amplía.

Especificación de Requerimientos de Software

para

DataPets

Versión 1.0 aprobada

Preparada por Grupo 50 #6

**Amador, Narda
Ferreira, Sharom
Mican, Paola
Peña, Joel
Pidache, Blanca**

Octubre 2022

Tabla de Contenido

| | |
|--|----|
| 1. Introducción | 2 |
| 1.1. Propósito | 2 |
| 1.2. Público Objetivo Y Sugerencias De Lectura | 3 |
| 1.3. Alcance del Producto | 3 |
| 1.4. Referencias | 3 |
| 2. Descripción General | 4 |
| 2.1. Perspectiva del Producto | 4 |
| 2.2. Funciones del Producto | 4 |
| 2.3. Clases y Características de Usuario | 4 |
| 2.4. Entornos Operativos | 4 |
| 2.5. Restricciones de Diseño e Implementación | 5 |
| 2.6. Documentación de Usuario | 5 |
| 2.7. Suposiciones y Dependencias | 5 |
| 3. Requisitos Externos de la Interfaz | 5 |
| 3.1. Interfaz de Usuario | 5 |
| 3.2. Software Interfaces | 5 |
| 3.3. Interfaces de Comunicaciones | 7 |
| 4. Características del Sistema | 7 |
| 4.1. Registro de Usuario | 7 |
| 4.2. Actualizar Información de Usuario | 9 |
| 4.3. Recuperar Contraseña | 10 |
| 4.4. Agregar Establecimientos | 12 |
| 5. Otros requisitos no funcionales | 13 |
| 5.1. Requisitos de Desempeño | 13 |
| 5.2. Requisitos de Seguridad | 13 |
| 5.3. Atributos de Calidad del Software | 13 |
| 5.4. Usabilidad | 13 |

Historial de Revisiones

| Nombre | Fecha | Razón para cambios | Versión |
|--------|-------|--------------------|---------|
| | | | |
| | | | |

1. Introducción

1.1 Propósito

Con el presente documento se pretende explicar de forma detallada la descripción de la funcionalidad de la página web llamada DataPets, la cual se elaboró con el fin de dar

cumplimiento a los objetivos planteados en el ciclo 3 del programa de Desarrollo de software del ministerio TIC, MisiónTIC, impartido por la Universidad Tecnológica de Pereira.

1.2 Público Objetivo Y Sugerencias De Lectura

Este manual de usuario está destinado para la lectura por parte de testers y desarrolladores que busquen implementar mejoras futuras a nuevas versiones del sitio. Además, para aquellos usuarios que deseen saber más y mejor el funcionamiento del sitio.

1.3 Alcance del Producto

La página web DataPets, es una página web creada para que las personas que poseen una mascota (perros o gatos) puedan conocer de antemano, cuáles establecimientos son pet friendly (permiten la entrada de dichos animales domésticos a sus establecimientos), así se evitan la problemática de no poder entrar a diferentes lugares o centros comerciales.

En la página web, los usuarios registrados tienen un número ilimitado de búsquedas mientras que los usuarios que no quieran registrarse (invitados) solo podrán realizar un máximo de 5 visitas. En la página no se podrán realizar reservas de ningún tipo, solo se usa como un medio para informarse sobre si un establecimiento acepta o no mascotas.

1.4 Referencias

- Djaneiro - Django Snippets - Visual Studio Marketplace. Marketplace.visualstudio.com. (2022). Recuperado el 19 de agosto de 2022, de <https://marketplace.visualstudio.com/items?itemName=thebarkman.vscod-djaneiro>.
- HTML. Desarrolloweb.com. (2022). Recuperado el 19 de agosto de 2022, de <https://desarrolloweb.com/home/html>.
- Koch, N.; Knapp, A.; Zhang, G.; Baumeister, H. "UML-Based Web Engineering. An Approach Based on Standards", Chapter 7 en: Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. (Editors) Web Engineering: Modelling and Implementing Web Applications, Springer-Verlag London Limited, 2008.]
- Schwinger, W.; Koch, N. "Modeling Web Applications", Chapter 3 en: Kappel, G.; Pröll, B.; Reich, S.; Retschitzegger, W. (Editors) Web Engineering. The Discipline of Systematic Development of Web Applications, John Wiley & Sons Ltd., 2006.
- The web framework for perfectionists with deadlines | Django. Djangoproject.com. (2022). Recuperado el 19 de agosto de 2022, de <https://www.djangoproject.com/>.
- Visual Studio Code - Code Editing. Redefined. Code.visualstudio.com. (2022). Recuperado el 19 de agosto de 2022, de <https://code.visualstudio.com/>.
- Welcome to Python.org. Python.org. (2022). Recuperado el 19 de agosto de 2022, de <https://www.python.org/>.

2. Descripción General

2.1 Perspectiva del Producto

Se implementó un sitio web que permite agregar y actualizar datos sobre establecimientos Pet Friendly como son los horarios, fotos, dirección, entre otros. Además, permite registrar un usuario con nombre, correo, contraseña y foto de perfil; usuario que según sea el caso tendrá cierta cantidad de permisos sobre la actualización de la información de interés en el sitio.

2.2 Funciones del Producto

Los procesos o funciones que conforman el sitio web son los siguientes:

Registro de Usuario: agrega un nuevo usuario que tiene acceso al sitio web con permisos de agregar establecimientos con su respectiva información y a visualizar todos los establecimientos registrados en el sitio.

Agregar Establecimientos: el usuario registra, según sus permisos, establecimientos Pet Friendly con la información requerida.

Actualizar Establecimientos: el usuario modifica la información exclusivamente de los establecimientos agregados por este.

Dejar Comentarios: el usuario registra comentarios en el sitio que pueden ser vistos por cualquiera que ingrese al sitio.

2.3 Clases y Características de Usuario

El usuario realiza estos procesos el sitio:

| Función | Usuario | SuperUsuario | UsuarioEditor | Usuario |
|--|---------|--------------|---------------|---------|
| Registrar usuario | X | X | X | X |
| Actualizar información de usuario | X | X | X | X |
| Recuperar contraseña | X | X | X | X |
| Agregar establecimientos | X | X | X | |
| Actualizar información de establecimientos | X | X | X | |
| Eliminar establecimientos | X | X | X | |
| Registrar comentarios | X | X | X | |
| Editar comentarios | X | | | |
| Eliminar comentarios | X | | | |

2.4 Entornos Operativos

El sitio web puede ser visitado desde cualquier navegador de escritorio o web. Se desarrolló con HTML5, CSS, JavaScript, CSS, SQL y Python.

2.5 Restricciones de Diseño e Implementación

- El uso de la página requiere de conexión a internet.
- Interfaces de usuario intuitivas.
- Debe ser funcional en los navegadores más comunes.
- La página se ejecuta con servidores de Python.
- Debe usarse Django como framework.
- El aplicativo debe contar con un sistema de validación de sesión.
- Utilizar un Modelo Vista Controlador (MVC).
- Los servidores deben ser capaces de atender consultas concurrentemente.

2.6 Documentación de Usuario

- Este manual de usuario.
- Video de presentación del sitio: <https://youtu.be/E6UKNqPXkew>

2.7 Suposiciones y Dependencias

- Se debe contar con un administrador que garantice el correcto funcionamiento de la página y sus servicios.
- Se debe contar con acceso a internet en los equipos donde sea utilizada la página.
- La base de datos de los establecimientos y sus restricciones deben ser actualizadas constantemente.

3. Requisitos Externos de la Interfaz

3.1 Interfaz de Usuario

La Interfaz de Usuario es bastante intuitiva, consta del mismo tipo de botones e hipervínculos que se encuentra en la mayoría de las páginas web, esto con el fin de que el usuario se sienta familiarizado, ya que la mayoría de las personas hemos usado una página informativa con ingreso de usuario y contraseña.

No se desarrollará interfaz del hardware porque la aplicación no tiene relación con otros sistemas.

3.2 Software Interfaces

Selección de editor de código: **Visual Studio Code**.

Instalación de extensiones:

- *Auto Rename Tag*: renombrar etiquetas HTML/XML automáticamente.
- *Djaneiro - Django Snippets*: colección de snippets para plantillas, modelos, vistas, campos y formularios de Django.

- *Django*: mejorar la sintaxis de django
- *HTML Snippets*: autocompletar código HTML.
- *IntelliCode*: mostrar elementos de finalización recomendados para el contexto de código.
- *Live Server*: ver cambios en tiempo real de archivos estáticos.
- *Prettier - Code formatter*: formatear el código.
- *Python*: compatibilidad con Python.
- *Python snippets*: snippets para Python.
- *VScode-icons*: íconos para los diferentes archivos.

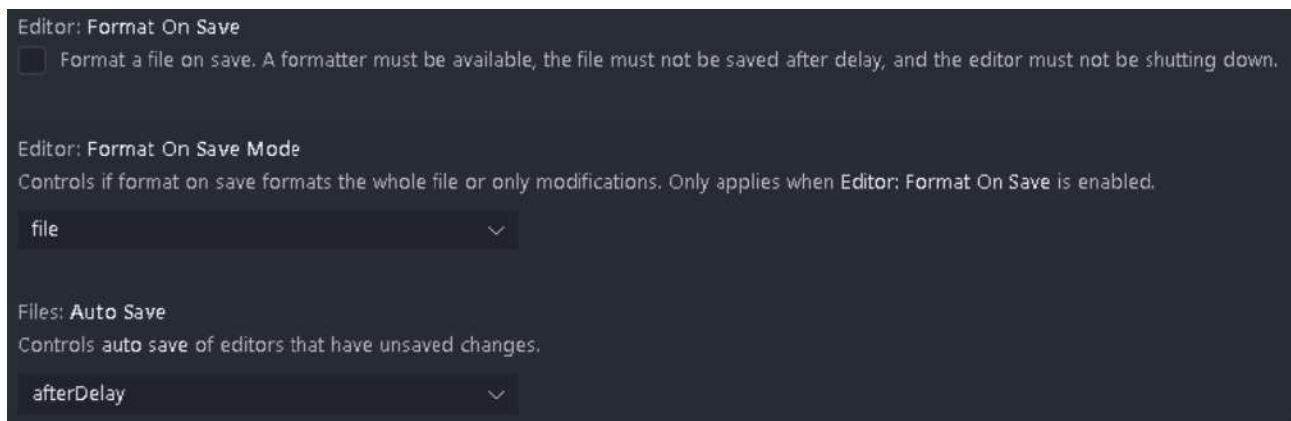
Configuración:

Los archivos se guardan automáticamente mientras se escribe, el código se formatea manualmente por medio de las extensiones por defecto de los diferentes archivos.

Lenguaje de programación Python

Versión:

- 3.10 - <https://www.python.org/downloads/>



Librerías:

- Django - <https://www.djangoproject.com/>
- psycopg2 - <https://pypi.org/project/psycopg2/>
- python-decouple - <https://pypi.org/project/python-decouple/>
- sqlparse - <https://pypi.org/project/sqlparse/>

Base de datos:

- PostgreSQL - <https://www.postgresql.org>

Controlador de versiones:

- Git - <https://git-scm.com/>

Alojamiento de repositorio:

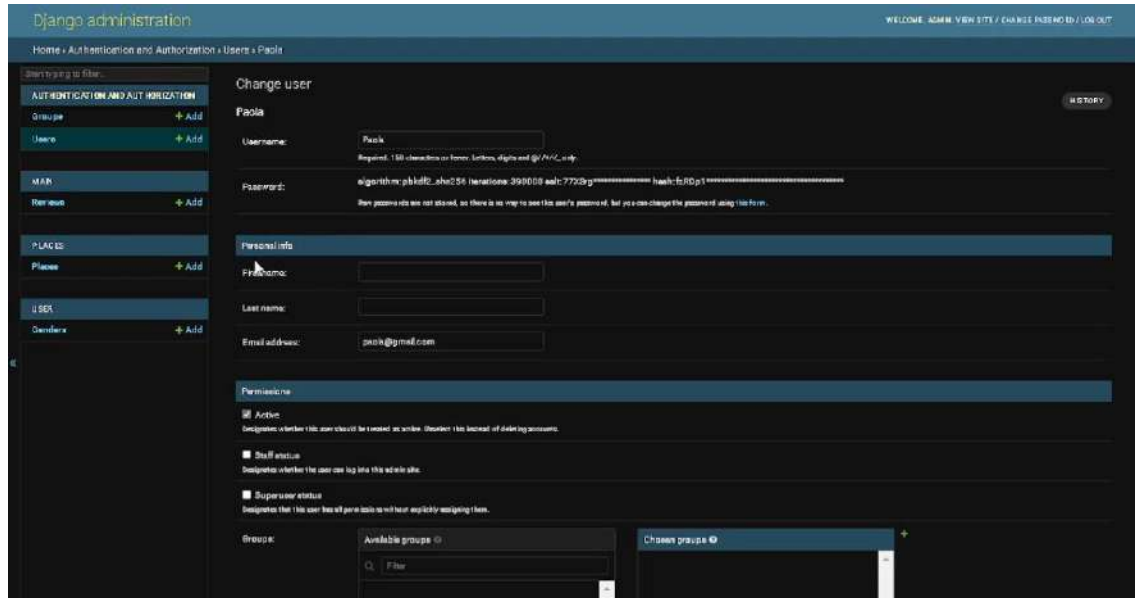
- GitHub - <https://github.com/blanca-marlen/DataPets>

Plataforma de publicación en la nube:

- Heroku - <https://datapets.herokuapp.com/>

3.3 Interfaces de Comunicaciones

Con el administrador de Django se hicieron la mayor parte de configuraciones de la página, entre ellos los grupos de usuarios para autenticación, permisos de edición, entre otros.

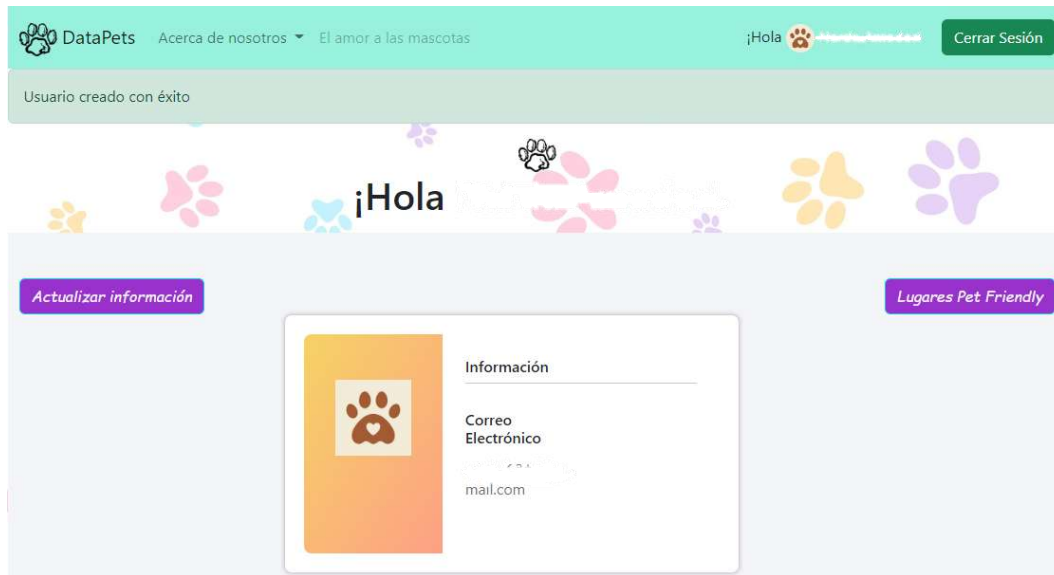


4. Características del Sistema

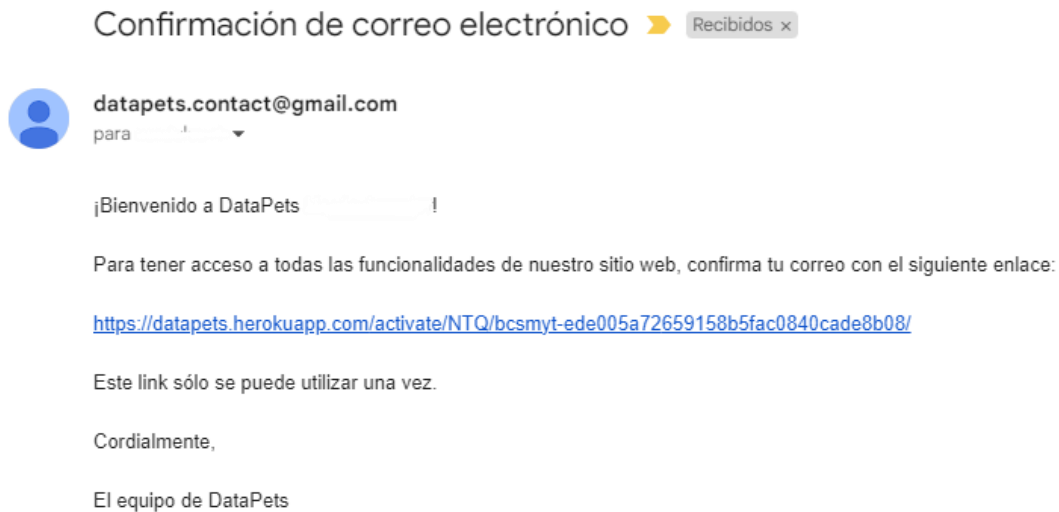
4.1 Registro de Usuario



- El usuario ingresa 4 campos: Usuario, correo, contraseña y confirma contraseña.
- Da clic en el botón Registrarse.



- La página muestra un mensaje: “usuario creado con éxito”.
- Solicita: “Revisa la bandeja de entrada de tu correo electrónico y actualiza tus datos para poder dejar una reseña”
- El usuario recibe en su correo un correo electrónico de verificación para empezar a usar el sitio.



- Al ingresar al link, aparece un mensaje que informa la activación de la cuenta:

4.2 Actualizar Información de Usuario

- Como usuario registrado, ingresa a su perfil.
- Da clic en el botón Actualizar información.

The screenshot shows the 'Actualizar información' page. At the top, there is a navigation bar with the DataPets logo, links to 'Acercas de nosotros' and 'El amor a las mascotas', and a user profile section with '¡Hola', a paw print icon, and a 'Cerrar Sesión' button. Below the navigation bar, the page has a light blue background with colorful paw prints. On the left, there is a purple button labeled 'Actualizar información'. On the right, there is a purple button labeled 'Lugares Pet Friendly'. In the center, there is a white box with an orange header and a paw print icon. Below the header, the text 'Información' is followed by a form field for 'Correo Electrónico' with the placeholder text 'mail.com'. At the bottom of the page, there is a text prompt: 'Actualiza tus datos para poder dejar una reseña'.

The screenshot shows the 'Actualizar Datos' page. At the top, there is a navigation bar with the DataPets logo, links to 'Acercas de nosotros' and 'El amor a las mascotas', and a user profile section with '¡Hola', a paw print icon, and a 'Cerrar Sesión' button. Below the navigation bar, the page has a light green background with colorful paw prints. On the left, there is a large paw print icon. To the right of the icon, there is a form with the following fields: 'Cambiar foto de perfil' (with a sub-label 'Seleccionar archivo' and 'Sin archivos seleccionados'), 'Usuario', 'Nombres', 'Apellidos', 'Género' (with a dropdown menu), 'Teléfono' (with a sub-label '(+57) Número de teléfono'), 'Contraseña', 'Confirmar Contraseña', and 'Confirma contraseña'. At the bottom right of the form, there is a green button labeled 'Guardar'. At the bottom left of the page, there is a copyright notice: '© 2022 DataPets'. At the bottom right, there are social media icons for Twitter, Instagram, and Facebook.

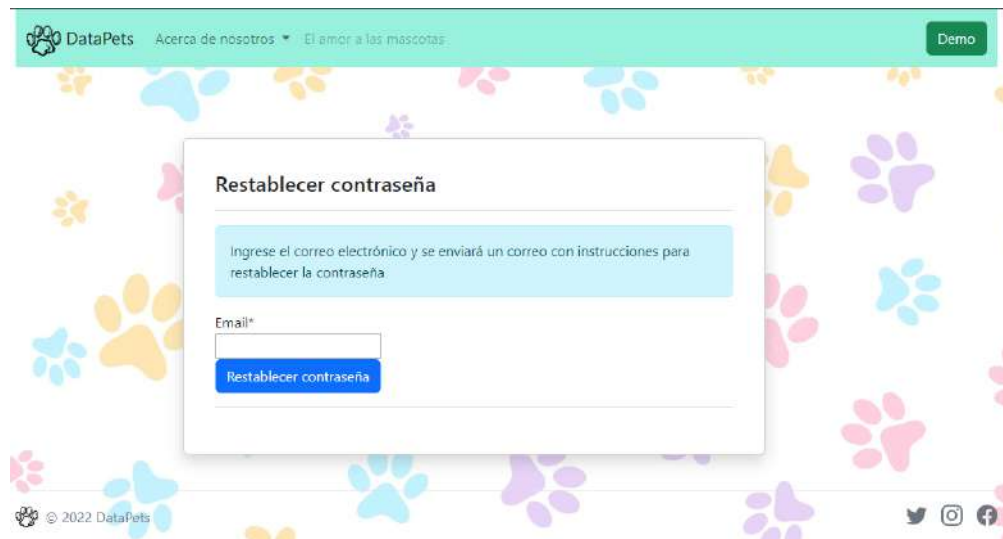
- Rellena el formulario y da clic en el botón Guardar.

4.3 Recuperar Contraseña

- Clic en ¿Olvidó su contraseña?

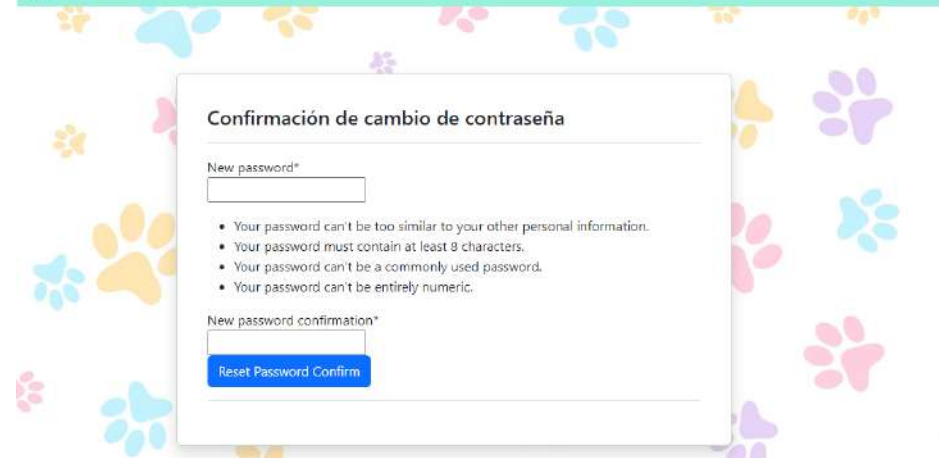
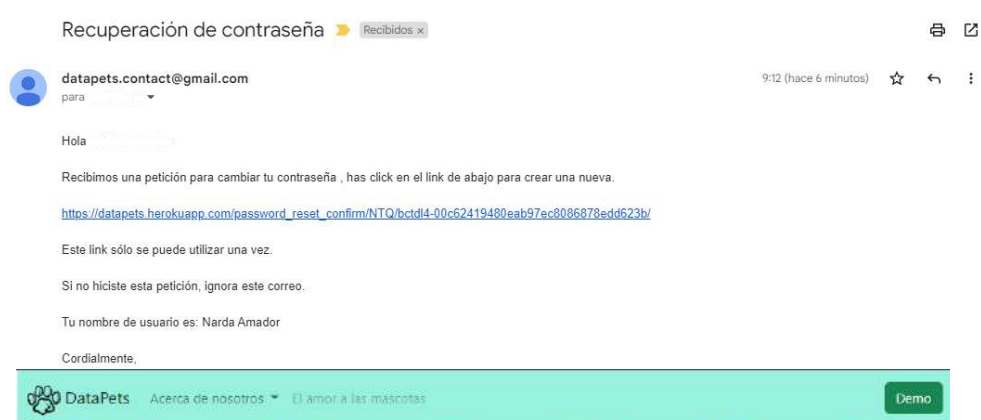


- Ingrese el correo electrónico y se enviará un correo con instrucciones para restablecer la contraseña.





- Llega un correo de recuperación
- Siguiendo el link se crea una nueva contraseña, inmediatamente se puede ingresar con la nueva contraseña.





4.4 Agregar Establecimientos

- Clic en Lugares Pet Friendly

A screenshot of the 'Agregar Establecimientos' (Add Establishments) form. The form is set against a background of colorful paw prints. It contains several input fields: 'Foto' (Photo) with a 'Seleccionar archivo' button and 'Sin archivos seleccionados' text; 'Logo' with a 'Seleccionar archivo' button and 'Sin archivos seleccionados' text; 'Lugar' (Location) with a text input; 'Dirección' (Address) with a text input; 'Teléfono' (Phone) with a text input and a placeholder '(+57) Número de teléfono'; 'Horario de atención' (Operating hours) with a text input and a placeholder 'Horario de atención - (días / horas)'; and 'Fecha de visita' (Visit date) with a text input and a placeholder 'YYYY-MM-DD'. At the bottom of the form, a button reads 'Actualiza tus datos para poder dejar una reseña' (Update your data to be able to leave a review).

- Llenar el formulario y clic en Agregar
- Aparecerá en la sección de establecimientos

5. Otros requisitos no funcionales

5.1 Requisitos de Desempeño

- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario con agilidad.
- El sistema debe ser capaz de operar adecuadamente con varios usuarios a la vez.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden.

5.2 Requisitos de Seguridad

- Los permisos de acceso al sistema podrán ser cambiados solamente por el Super Usuario.

5.3 Atributos de Calidad del Software

- *Disponibilidad:* Permanente gracias al estar en una plataforma de publicación en la nube.
- *Escalabilidad:* Se pretende aumentar el alcance y capacidad del sitio para versiones futuras.
- *Modificabilidad:* Al ser un sitio sencillo se puede modificar fácilmente.
- *Testeabilidad:* Se puede testear con el uso de un número mínimo de aplicaciones.
- *Distribución del desarrollo:* Ya que se encuentra el repositorio en GitHub, equipos separados a nivel global puedan trabajar en él.

5.4 Usabilidad

- El tiempo de aprendizaje del sistema por un usuario es menor a 1 hora.
- Se cuenta con manual de usuario.
- El sitio entrega mensajes de error que son informativos y orientados al usuario final.



- El sitio posee un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tablets y teléfonos inteligentes.
- El sitio posee interfaces gráficas bien formadas.