# 

# **DOCUMENTACIÓN PROYECTO LARAVEL**

Anderson Camilo Chimbi Alarcón

Brajhand Darío Zarate

Yojhan Estiven Duran Lasso

Jorge Esteban Gómez Castañeda

Universidad De Cundinamarca Facultad De Ingeniería

Ingeniería De Sistemas Y Computación 2020

Fusagasugá

2023

# **DOCUMENTACIÓN PROYECTO LARAVEL**

* Índice Pág

1. Introducción ………………………………………………………………. 3
2. Diseño del Software ……………………………………………………... 3

2.1. Características del software …………..…………………………… 3

2.2. GUIs ………………………………………………………………….. 5

2.3. Bases de Datos …………………………………………………...… 5

2.4. Diagramas UML …………………………………………………….. 12

1. Implementación ……………………………………………….……..…… 14

3.1. Software Adopción de perros en PHP Html Css y MySQL.......... 14

1. Conclusiones ………………………………………………….…………... 16
2. Referencias ………………………………………………..……….……... 17
3. Introducción

En un mundo donde la conexión entre las personas y los animales es esencial, surge nuestra aplicación dedicada a facilitar la adopción de perros, creando un espacio digital donde los amantes de los animales pueden encontrar compañeros leales y dar a estos maravillosos seres una segunda oportunidad en la vida.

Nuestra aplicación, desarrollada con el poderoso framework PHP Laravel, combina una interfaz intuitiva y amigable con funcionalidades robustas para crear una experiencia única. Al utilizar Laravel, garantizamos un desarrollo eficiente, escalable y seguro, permitiendo a los usuarios explorar perfiles de perros disponibles para adopción o registrar a sus propios amigos peludos en busca de un nuevo hogar.

1. Diseño del Software

2.1. Características del software:

* Interfaz de Usuario (UI):

Proporciona una forma visual e interactiva para que los usuarios interactúen con el software.

Diseñada para ser intuitiva y fácil de usar.

* Funcionalidad:

Cumple con los requisitos y objetivos previstos.

Realiza tareas específicas o funciones para las cuales fue diseñado.

* Compatibilidad:

Puede funcionar en diferentes sistemas operativos y hardware.

Asegura la interoperabilidad con otros software y tecnologías.

* Eficiencia:

Utiliza recursos de manera eficiente (CPU, memoria, almacenamiento).

Ofrece un rendimiento rápido y sin problemas.

* Portabilidad:

Puede ser trasladado y ejecutado en diferentes entornos sin modificaciones significativas.

Adaptable a diversas configuraciones y dispositivos.

* Mantenibilidad:

Facilita la corrección de errores y la implementación de actualizaciones.

Código bien estructurado y documentado.

* Documentación:

Proporciona manuales, guías y documentación técnica para usuarios y desarrolladores.

Facilita la comprensión y el uso del software.

* Flexibilidad:

Puede adaptarse a cambios en los requisitos y necesidades del usuario.

Ofrece configuraciones personalizables.

* Escalabilidad:

Capaz de manejar un aumento en la carga de trabajo o en la cantidad de usuarios.

Escala de manera efectiva para soportar mayores demandas.

* Fiabilidad:

Proporciona resultados consistentes y precisos.

Minimiza los fallos y errores inesperados.

* Interconexión:

Puede integrarse con otros sistemas y aplicaciones.

Facilita la comunicación y el intercambio de datos.

* Facilidad de Aprendizaje:

Diseñado para que los usuarios aprendan a utilizarlo rápidamente.

Proporciona ayudas y tutoriales.

2.2. GUIs:

Código fuente: Se anexa el archivo o código fuente ejecutable en el rar.

2.3. Bases de Datos:

Se crea una base de datos llamada adopción y se importas en MySQL

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Servidor: 127.0.0.1

-- Tiempo de generación: 14-11-2023 a las 03:40:27

-- Versión del servidor: 10.4.28-MariaDB

-- Versión de PHP: 8.2.4

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Base de datos: `adopcion`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `failed\_jobs`

--

CREATE TABLE `failed\_jobs` (

`id` bigint(20) UNSIGNED NOT NULL,

`uuid` varchar(255) NOT NULL,

`connection` text NOT NULL,

`queue` text NOT NULL,

`payload` longtext NOT NULL,

`exception` longtext NOT NULL,

`failed\_at` timestamp NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `mascotas`

--

CREATE TABLE `mascotas` (

`id` bigint(20) UNSIGNED NOT NULL,

`genero` varchar(255) NOT NULL,

`edad` varchar(255) NOT NULL,

`descripcion` varchar(255) NOT NULL,

`FotoMascota` varchar(2048) DEFAULT NULL,

`remember\_token` varchar(100) DEFAULT NULL,

`current\_team\_id` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `mascotas`

--

INSERT INTO `mascotas` (`id`, `genero`, `edad`, `descripcion`, `FotoMascota`, `remember\_token`, `current\_team\_id`, `created\_at`, `updated\_at`) VALUES

(1, 'Macho', '2 años', 'Gato pardo gordo encontrado en Silvania número de contacto 311 324 3409', 'img/FotoMascota/1699660652-jpg', NULL, NULL, '2023-11-11 04:57:32', '2023-11-11 04:57:32'),

(2, 'Macho', '3 años', 'Gato con mucho flow lo doy en adopcion porque no hace nada, teléfono de contacto: 310 243 2348', 'img/FotoMascota/1699661229-jpg', NULL, NULL, '2023-11-11 05:07:09', '2023-11-11 05:07:09');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `migrations`

--

CREATE TABLE `migrations` (

`id` int(10) UNSIGNED NOT NULL,

`migration` varchar(255) NOT NULL,

`batch` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `migrations`

--

INSERT INTO `migrations` (`id`, `migration`, `batch`) VALUES

(22, '2014\_10\_12\_000000\_create\_users\_table', 1),

(23, '2014\_10\_12\_100000\_create\_password\_reset\_tokens\_table', 1),

(24, '2014\_10\_12\_200000\_add\_two\_factor\_columns\_to\_users\_table', 1),

(25, '2019\_08\_19\_000000\_create\_failed\_jobs\_table', 1),

(26, '2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table', 1),

(27, '2023\_10\_22\_221106\_create\_sessions\_table', 1),

(28, '2023\_11\_07\_212358\_mascotas', 1);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `password\_reset\_tokens`

--

CREATE TABLE `password\_reset\_tokens` (

`email` varchar(255) NOT NULL,

`token` varchar(255) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `personal\_access\_tokens`

--

CREATE TABLE `personal\_access\_tokens` (

`id` bigint(20) UNSIGNED NOT NULL,

`tokenable\_type` varchar(255) NOT NULL,

`tokenable\_id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(255) NOT NULL,

`token` varchar(64) NOT NULL,

`abilities` text DEFAULT NULL,

`last\_used\_at` timestamp NULL DEFAULT NULL,

`expires\_at` timestamp NULL DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `sessions`

--

CREATE TABLE `sessions` (

`id` varchar(255) NOT NULL,

`user\_id` bigint(20) UNSIGNED DEFAULT NULL,

`ip\_address` varchar(45) DEFAULT NULL,

`user\_agent` text DEFAULT NULL,

`payload` longtext NOT NULL,

`last\_activity` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `sessions`

--

INSERT INTO `sessions` (`id`, `user\_id`, `ip\_address`, `user\_agent`, `payload`, `last\_activity`) VALUES

('J3pirlc7WP3Jbe3f0naVJnDCsTrfoKTdaF8jtTeg', NULL, '127.0.0.1', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0', 'YTozOntzOjY6Il90b2tlbiI7czo0MDoiQmNndG9MODNSa0lPdjhlTXlJZzIxc3VTYUxNZVdiMVlqY29WMzhzeiI7czo2OiJfZmxhc2giO2E6Mjp7czozOiJvbGQiO2E6MDp7fXM6MzoibmV3IjthOjA6e319czo5OiJfcHJldmlvdXMiO2E6MTp7czozOiJ1cmwiO3M6MjE6Imh0dHA6Ly8xMjcuMC4wLjE6ODAwMCI7fX0=', 1699664118);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `users`

--

CREATE TABLE `users` (

`id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(255) NOT NULL,

`email` varchar(255) NOT NULL,

`email\_verified\_at` timestamp NULL DEFAULT NULL,

`password` varchar(255) NOT NULL,

`two\_factor\_secret` text DEFAULT NULL,

`two\_factor\_recovery\_codes` text DEFAULT NULL,

`two\_factor\_confirmed\_at` timestamp NULL DEFAULT NULL,

`role\_id` smallint(5) UNSIGNED DEFAULT NULL,

`remember\_token` varchar(100) DEFAULT NULL,

`current\_team\_id` bigint(20) UNSIGNED DEFAULT NULL,

`profile\_photo\_path` varchar(2048) DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `users`

--

INSERT INTO `users` (`id`, `name`, `email`, `email\_verified\_at`, `password`, `two\_factor\_secret`, `two\_factor\_recovery\_codes`, `two\_factor\_confirmed\_at`, `role\_id`, `remember\_token`, `current\_team\_id`, `profile\_photo\_path`, `created\_at`, `updated\_at`) VALUES

(1, 'Anderson', 'andersonchimbi79@gmail.com', NULL, '$2y$10$huvcUUzOEI5xksOGUpwYx.eYkW.JI4JICwEyUKiwScdjWgN7E2QOW', NULL, NULL, NULL, 2, NULL, NULL, 'profile-photos/Gsw3kiACckeha55HAgGICUYAtD9kXNGYEhWoBKuQ.jpg', '2023-11-11 04:55:20', '2023-11-11 05:03:00');

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `failed\_jobs`

--

ALTER TABLE `failed\_jobs`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `failed\_jobs\_uuid\_unique` (`uuid`);

--

-- Indices de la tabla `mascotas`

--

ALTER TABLE `mascotas`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `migrations`

--

ALTER TABLE `migrations`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `password\_reset\_tokens`

--

ALTER TABLE `password\_reset\_tokens`

ADD PRIMARY KEY (`email`);

--

-- Indices de la tabla `personal\_access\_tokens`

--

ALTER TABLE `personal\_access\_tokens`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `personal\_access\_tokens\_token\_unique` (`token`),

ADD KEY `personal\_access\_tokens\_tokenable\_type\_tokenable\_id\_index` (`tokenable\_type`,`tokenable\_id`);

--

-- Indices de la tabla `sessions`

--

ALTER TABLE `sessions`

ADD PRIMARY KEY (`id`),

ADD KEY `sessions\_user\_id\_index` (`user\_id`),

ADD KEY `sessions\_last\_activity\_index` (`last\_activity`);

--

-- Indices de la tabla `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `users\_email\_unique` (`email`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `failed\_jobs`

--

ALTER TABLE `failed\_jobs`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `mascotas`

--

ALTER TABLE `mascotas`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT de la tabla `migrations`

--

ALTER TABLE `migrations`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=29;

--

-- AUTO\_INCREMENT de la tabla `personal\_access\_tokens`

--

ALTER TABLE `personal\_access\_tokens`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `users`

--

ALTER TABLE `users`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

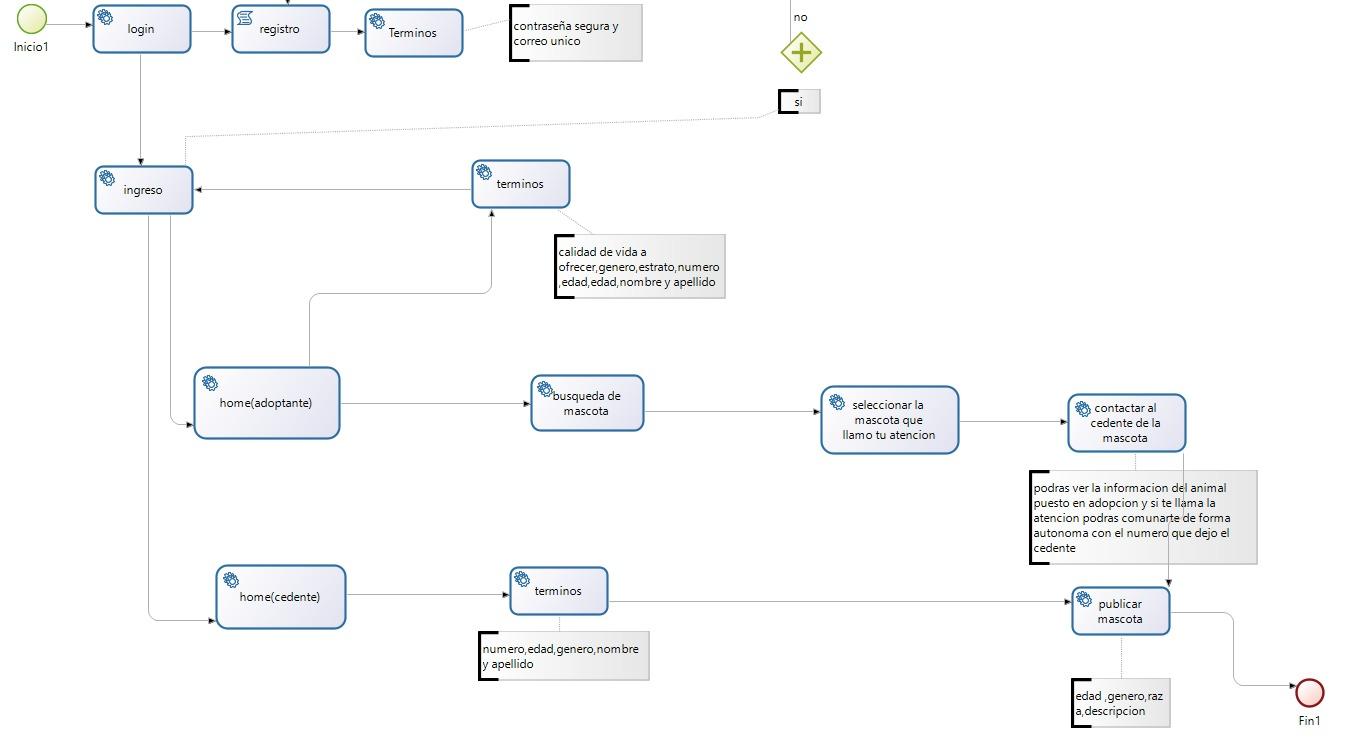
COMMIT;

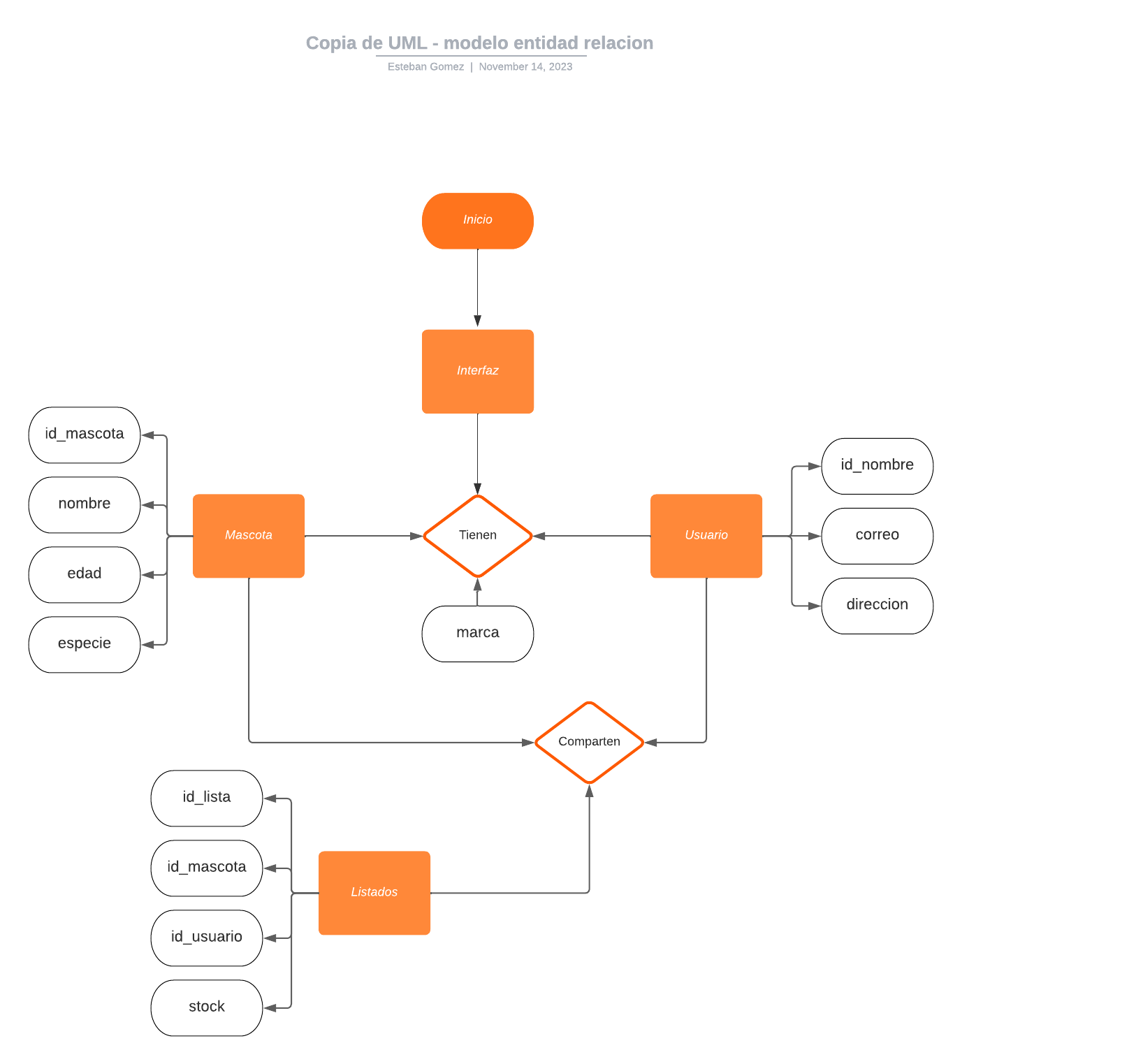
/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

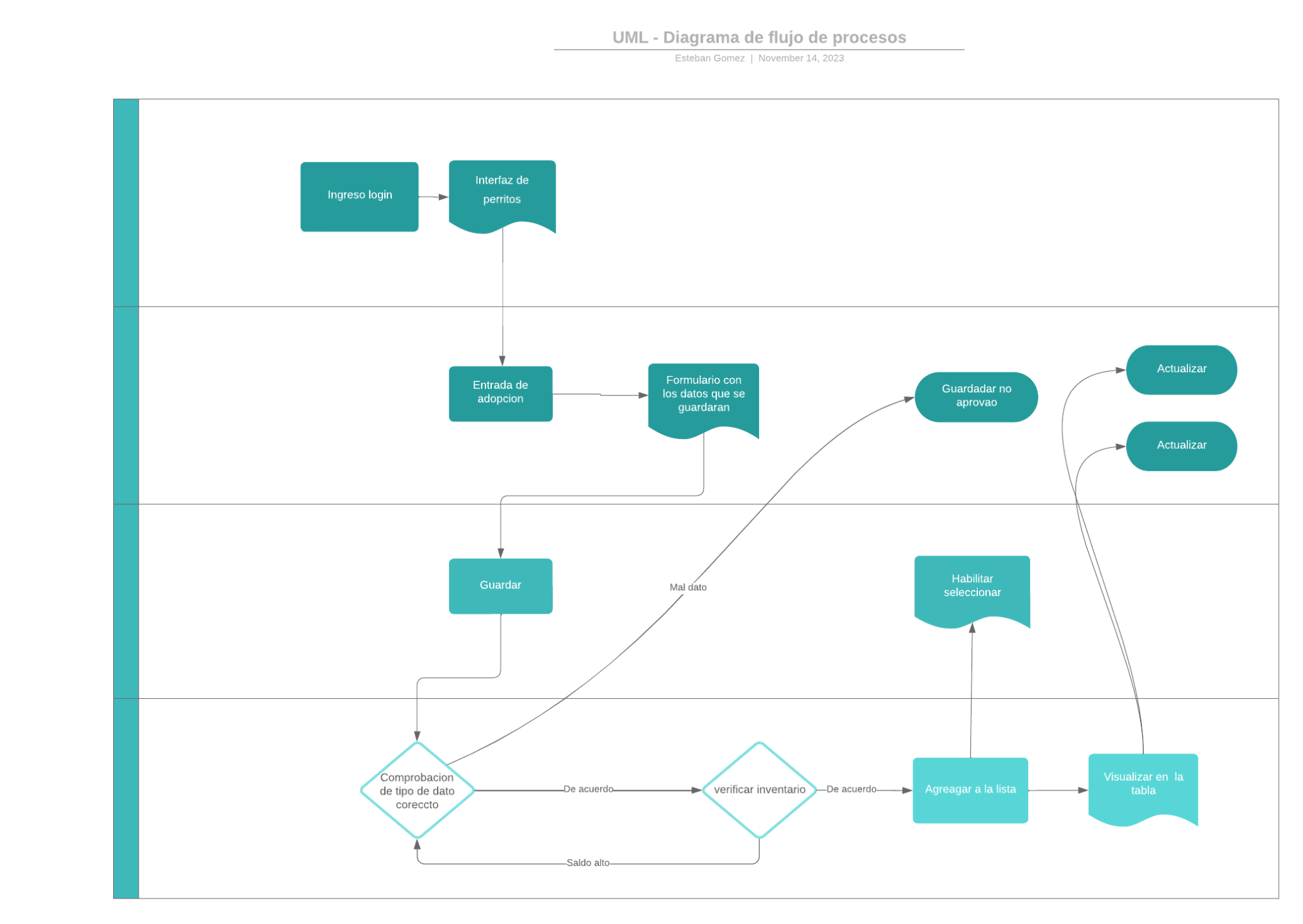
/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

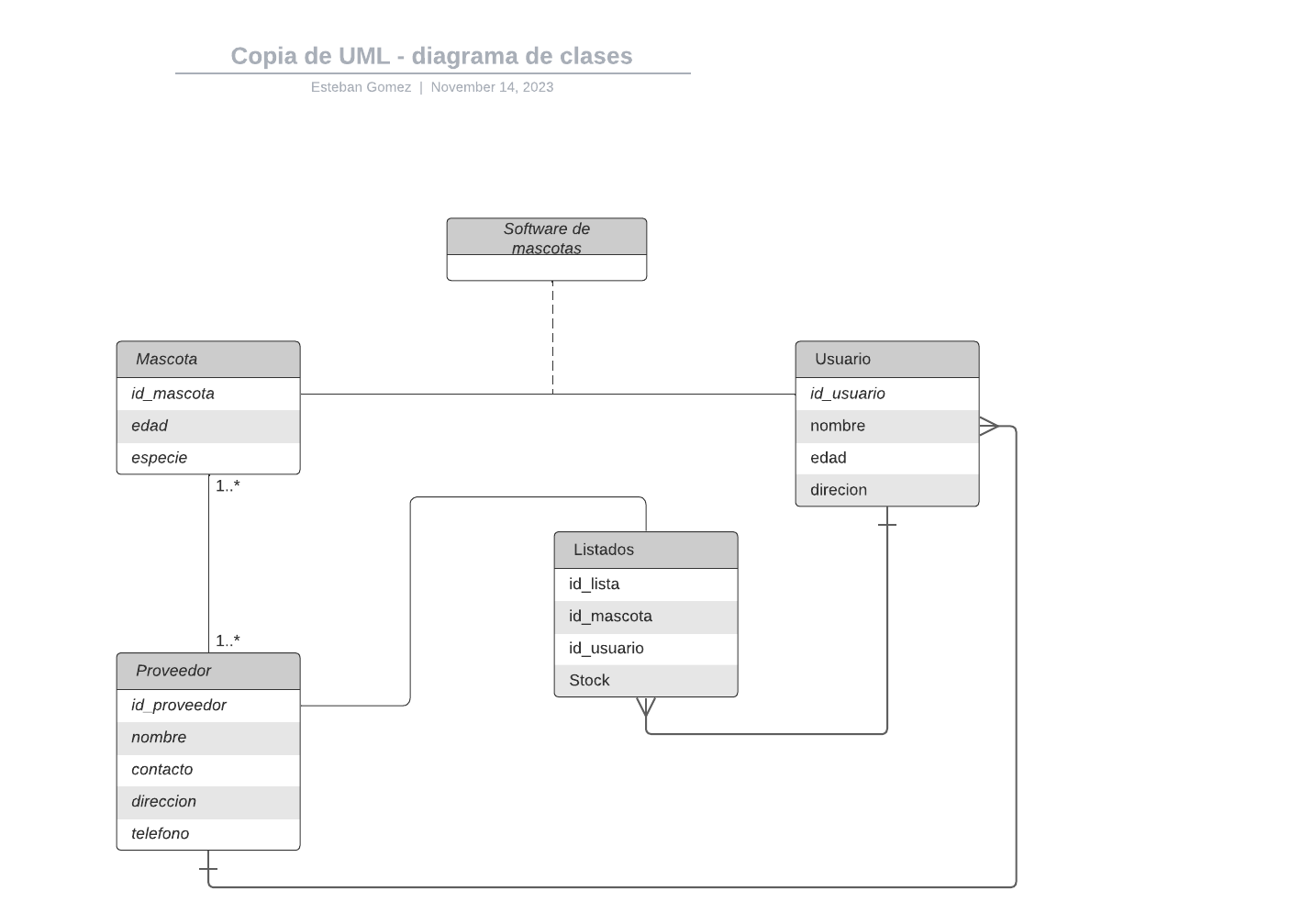
/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

2.4. Diagramas UML:









1. Implementación:

3.1. Software Adopción de perros en PHP Html Css y MySQL

Entiendo que estás buscando una descripción de la implementación de un software de adopción de perros utilizando PHP, HTML, CSS, MySQL y Laravel. Aquí está una descripción general de cómo estos elementos podrían interactuar en la implementación de una aplicación de adopción de perros:

PHP (Laravel):

Laravel, un framework de PHP, se utilizaría para la lógica del lado del servidor.

Se encargaría de manejar las rutas, controladores, modelos y vistas de la aplicación.

Gestiona las interacciones entre la base de datos y la interfaz de usuario.

HTML y CSS:

HTML se utiliza para estructurar el contenido y la información que se muestra en la aplicación web.

CSS se emplea para dar estilo y diseño al contenido HTML, mejorando la apariencia y la experiencia visual para los usuarios.

Ambos trabajan juntos para proporcionar una interfaz de usuario atractiva y funcional.

MySQL:

Se utiliza como sistema de gestión de base de datos para almacenar información sobre los perros disponibles para adopción, los usuarios y sus solicitudes.

Almacena datos como detalles de los perros (nombre, edad, raza, descripción, etc.), información de contacto de los adoptantes, registros de adopciones, entre otros.

Interacción entre Componentes:

Laravel se encarga de gestionar las solicitudes del cliente y las respuestas del servidor.

Los controladores de Laravel interactúan con la base de datos MySQL para recuperar, actualizar, insertar o eliminar datos según sea necesario.

La lógica de negocio, como el proceso de adopción, validación de formularios y gestión de usuarios, se implementa en la parte PHP de la aplicación.

Seguridad:

Se aplican medidas de seguridad a nivel de código (PHP) y en la base de datos para proteger la información confidencial y prevenir vulnerabilidades como inyecciones SQL, entre otros.

1. Conclusiones:

El proyecto de la aplicación de adopción de perros, basada en Laravel, aborda la necesidad social de conectar dueños de perros con personas interesadas en adoptar mascotas. Destaca por su enfoque en la experiencia del usuario a través de una interfaz intuitiva y eficiente, respaldada por el framework Laravel que garantiza seguridad, escalabilidad y eficacia en el desarrollo. La gestión de datos sensibles se considera fundamental para proteger la información de mascotas y adoptantes. Además, la simplificación del proceso de adopción mediante solicitudes en línea promueve un aumento en las adopciones exitosas. La creación de una comunidad de usuarios para compartir historias y experiencias y la flexibilidad para adaptarse a las necesidades cambiantes de los usuarios y tendencias en adopción de mascotas son aspectos clave para el continuo crecimiento y relevancia de la aplicación.

1. Referencias:

[1] "Clean Code: A Handbook of Agile Software Craftsmanship" por Robert C. Martin: Ofrece principios y prácticas para escribir código limpio y comprensible.

[2] "The Pragmatic Programmer: Your Journey To Mastery" por Andrew Hunt y David Thomas: Explora estrategias prácticas y principios para mejorar las habilidades de programación.

[3] "Design Patterns: Elements of Reusable Object-Oriented Software" por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides: Presenta patrones de diseño comunes para mejorar la reutilización y la flexibilidad del software.

[4] "Code Complete: A Practical Handbook of Software Construction" por Steve McConnell: Aborda los conceptos fundamentales de la construcción de software y ofrece pautas prácticas para mejorar la calidad del código.

[5] "Refactoring: Improving the Design of Existing Code" por Martin Fowler: Explica cómo mejorar el diseño del código existente mediante técnicas de refactorización.

[6] "Introduction to the Theory of Computation" por Michael Sipser: Un libro de referencia sobre teoría de la computación que cubre temas como autómatas, lenguajes formales y complejidad computacional.

[7] "Structure and Interpretation of Computer Programs" por Harold Abelson y Gerald Jay Sussman: Explora los fundamentos de la programación y la computación a través de un enfoque basado en el lenguaje Scheme.

[8] "Head First Design Patterns" por Eric Freeman, Bert Bates, Kathy Sierra, y Elisabeth Robson: Presenta patrones de diseño de software de una manera accesible y fácil de entender.