****

**SERVICIO NACIONAL DE ADIESTRAMIENTO EN TRABAJO INDUSTRIAL**

**PLAN DE TRABAJO**

**DEL estudiantE**

1. **INFORMACIÓN GENERAL**

**001270288**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Apellidos y Nombres : | **SAAVEDRA SUCA JERSON** | ID : |  | |
| Dirección Zonal/CFP : |  | | | |
| Carrera : | **CFP -Villa el Salvador / Escuela ETI** | Semestre : | |  |
| Curso/ Mód. Formativo : | **Desarrollo de Aplicaciones Web**  **Desarrollo de Software**  **5** | | | |
| Tema del Trabajo : | **Framework Laravel Parte 2** | | | |

1. **PLANIFICACIÓN DEL TRABAJO**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N°** | **ACTIVIDADES/ ENTREGABLES** | **CRONOGRAMA/ FECHA DE ENTREGA** | | | | | | | | | |
|  |  | **26/10** | **27/10** | **28/10** | 30/10 |  |  |  |  |  |  |
|  | **Entregable 2** |  |  |  |  |  |  |  |  |  |  |
| 1 | Descargar las dependencias composer | X |  |  |  |  |  |  |  |  |  |
| 2 | Ejecutar el comando y crear el proyecto laravel, configurar .env | X |  |  |  |  |  |  |  |  |  |
| 3 | Crear y configurar las migraciones | X |  |  |  |  |  |  |  |  |  |
| 4 | Crear base de datos |  | X |  |  |  |  |  |  |  |  |
| 5 | Crear las rutas |  | X |  |  |  |  |  |  |  |  |
| 6 | Crear los modelos y controladores |  | X |  |  |  |  |  |  |  |  |
| 7 | Crear la vista para mostrar los datos |  |  | X |  |  |  |  |  |  |  |
| 8 | Probar el funcionamiento del Proyecto |  |  | X |  |  |  |  |  |  |  |
| **9** | **Entrega de entregable 2** |  |  |  | X |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

1. **ENTREGABLES:**

**Durante la investigación de estudio, deberán de dar solución a los planteamientos de cada entregable:**

|  |  |
| --- | --- |
| **Nº** | **ENTREGABLE 2** |
| **1** | **Entregable 2: VeterinaryDB - Laravel - CRUD**  En esta fase del proyecto VeterinaryDB, se implementó un sistema CRUD (Crear, Leer, Actualizar y Eliminar) para gestionar las entidades **Mascota**, **Propietario**, **Veterinario** y **Medicamento**. Esto permite a los usuarios realizar operaciones completas sobre los datos, mejorando la funcionalidad del sistema. Componentes Principales:  * **Migraciones: Se crearon para definir las tablas de la base de datos** * **Modelos**: Se crearon modelos para cada entidad, representando la estructura de los datos. * **Controladores**: Se ajustaron controladores para manejar las solicitudes y realizar operaciones CRUD. * **Vistas**: Se desarrollaron interfaces amigables con Blade para interactuar con los datos. * **Rutas**: Se configuraron rutas para direccionar las solicitudes a los controladores adecuados. * **Validación de datos**: Se implementó un sistema para asegurar que los datos ingresados cumplan con las reglas definidas. |

|  |  |  |
| --- | --- | --- |
| **Desarrollo del Entregable N°2- DIBUJO / ESQUEMA / DIAGRAMA**  **(Adicionar páginas que sean necesarias)**  **Estructura del Proyecto:** Se organizaron los paquetes en controladores, modelos y vistas, rutas lo que promovió una arquitectura clara y modular.   Configuración del archivo .env El archivo **.env** en Laravel es un archivo de configuración donde se definen variables de entorno clave para la aplicación. En este proyecto, se utilizó para **configurar la conexión a la base de datos**, estableciendo detalles como:   * **DB\_CONNECTION**: tipo de base de datos (por ejemplo, mysql). * **DB\_HOST**: dirección del servidor de la base de datos. * **DB\_PORT**: puerto en el que escucha la base de datos. * **DB\_DATABASE**: nombre de la base de datos. * **DB\_USERNAME** y **DB\_PASSWORD**: credenciales de acceso.    Migraciones  1. **Migración de Propietarios**    * **Descripción**: Esta migración crea la tabla propietarios, que almacena la información básica de los propietarios de las mascotas, incluyendo su DNI, nombre, apellido, teléfono y correo. Se asegura de que cada propietario tenga un identificador único (propietario\_id) y un registro de tiempo para seguimiento. 2. **Migración de Veterinarios**    * **Descripción**: Crea la tabla veterinarios, que guarda los datos de los veterinarios, tales como su DNI, nombre, apellido, número de teléfono, especialidad y correo electrónico. También incluye una fecha de nacimiento que es opcional. Al igual que la tabla de propietarios, esta migración asegura un identificador único (veterinario\_id) para cada veterinario. 3. **Migración de Medicamentos**    * **Descripción**: Define la tabla medicamentos, donde se almacenan los detalles de los medicamentos disponibles, incluyendo el nombre, una descripción opcional, la cantidad disponible y el precio. Cada medicamento cuenta con un identificador único (medicamento\_id) para facilitar su gestión. 4. **Migración de Mascotas**    * **Descripción**: Crea la tabla mascotas, que almacena la información de las mascotas, como su nombre, especie, raza, género (masculino o femenino), edad y color. Esta migración incluye un identificador único (mascota\_id) para cada mascota, facilitando su referencia y gestión en el sistema.    Modelos1. **Mascota** Representa a los animales que son atendidos en la veterinaria. Contiene atributos como nombre, especie, raza, edad, y color. Este modelo permite gestionar la información básica sobre cada mascota y sus respectivas citas. 2. **Propietario** Representa a los dueños de las mascotas. Incluye información como nombre, apellido, DNI, correo y teléfono. Este modelo es crucial para vincular a cada mascota con su propietario, facilitando la gestión de la información de contacto y la programación de citas. 3. **Veterinario** Representa a los profesionales que atienden a las mascotas. Contiene datos como nombre, apellido, especialidad, y teléfono. Este modelo permite gestionar la información del personal veterinario, facilitando la asignación de veterinarios a las citas de las mascotas. 4. **Medicamento** Representa los medicamentos que pueden ser prescritos a las mascotas. Incluye atributos como nombre, descripcion, y precio. Este modelo es esencial para la gestión de tratamientos y recetas, permitiendo asociar medicamentos específicos con las mascotas y sus diagnósticos. Controladores Los controladores son componentes clave en la arquitectura MVC (Modelo-Vista-Controlador) de Laravel. Su función es gestionar la lógica de la aplicación, procesar las solicitudes del usuario y retornar las respuestas adecuadas. En WebVeterinary, se implementaron los siguientes controladores:   * **MascotaController**: Encargado de manejar las operaciones relacionadas con las mascotas, como la creación, actualización y eliminación de registros. También se ocupa de mostrar la lista de mascotas y los detalles de cada una.        * **PropietarioController**: Gestiona las acciones vinculadas a los propietarios de las mascotas, permitiendo agregar, editar y eliminar información de contacto, así como relacionar propietarios con sus mascotas.        * **VeterinarioController**: Administra los registros de veterinarios, permitiendo crear, editar y eliminar información sobre el personal veterinario, así como asignar veterinarios a citas.        * **MedicamentoController**: Se ocupa de la gestión de los medicamentos, incluyendo su creación, edición y eliminación, además de asociarlos a tratamientos específicos de las mascotas.      Rutas Las rutas en Laravel son responsables de definir la URL que se asigna a cada controlador. Permiten que el sistema responda a las solicitudes HTTP adecuadas. En WebVeterinary, se configuraron rutas para cada entidad y acción, tales como:   * **Rutas de Mascota**: Incluyen rutas para listar, crear, editar y eliminar mascotas, asignadas al MascotaController. * **Rutas de Propietario**: Definen las acciones para gestionar los propietarios a través del PropietarioController. * **Rutas de Veterinario**: Permiten acceder a las funcionalidades de gestión de veterinarios mediante el VeterinarioController. * **Rutas de Medicamento**: Configuran las operaciones relacionadas con los medicamentos, conectando con el MedicamentoController.    Manejo de errores y validación de datos. La validación de datos en Laravel se utiliza para asegurarse de que los datos enviados por los usuarios sean correctos y cumplan con ciertas reglas antes de ser procesados. Al utilizar el método validate() dentro de un controlador, se pueden definir reglas que especifican qué campos son obligatorios y qué formato deben tener. Por ejemplo, se puede requerir que el nombre y el apellido sean cadenas de texto no vacías y que el correo tenga un formato válido. Si los datos no cumplen con las reglas, Laravel generará automáticamente un mensaje de error, lo que mejora la experiencia del usuario al informarle sobre los campos que deben ser corregidos.  CreateMascotaRequest      CreateMedicamentoRequest    CreatePropietarioRequest    CreateVeterinarioRequest   Vistas Las vistas son archivos que definen cómo se presentan los datos al usuario. En Laravel, se utilizan las plantillas Blade para crear vistas. En WebVeterinary, se diseñaron las siguientes vistas:   * **Vista de la página principal: Muestra de una lista de módulos.**        * **Vista de Mascotas**: Muestra una lista de todas las mascotas registradas y permite acceder a formularios para crear o editar registros.     Registro Mascota          Editar Mascota            Eliminar Mascota          **Vista de Propietarios**: Presenta la información de los propietarios, permitiendo gestionar sus datos y vincularlos con sus respectivas mascotas.  Listado Propietario      Registrar Propietario            Editar Propietario            Eliminar Propietario          **Vista de Veterinarios**: Permite visualizar y gestionar la información del personal veterinario, así como asignar citas.  Listado Veterinario      Registrar Veterinario            Editar Veterinario            Eliminar Veterinario          **Vista de Medicamentos**: Muestra los medicamentos disponibles y permite gestionar su información.  Listado Medicamento      Registrar Medicamento            Editar Medicamento            Eliminar Medicamento | | |
|  | **[NOMBRE DEL TRABAJO] FRAMEWORK LARAVEL PARTE 2** | **[ESCALA]** |
| C:\Unidad_D\Nuevo Logo\SENATI_FF-01 Modificado.jpg | **[APELLIDOS Y NOMBRES] SAAVEDRA SUCA JERSON** | **1** |

**HOJA DE PLANIFICACIÓN (Entregable 2)**

**PROCESO DE EJECUCIÓN**

|  |  |
| --- | --- |
| **OPERACIONES / PASOS /SUBPASOS** | **SEGURIDAD / MEDIO AMBIENTE / NORMAS -ESTANDARES** |
| **1. Selección y Análisis del Caso Elegido** | **Se mantuvo un ambiente tranquilo y sin distracciones para la lectura detallada del caso.** |
| 1.1: Seleccionar el caso del Veterinario como proyecto a desarrollar. | **Fomentar un ambiente de colaboración para la generación de preguntas claras y pertinentes.** |
| 1.2: Leer y comprender en detalle los requerimientos del sistema de veterinario | **Proporcionar un espacio físico adecuado para reuniones productivas y participativas.** |
| 1.3: Identificar las posibles entidades. | **Acceso a bases de datos en línea y otras fuentes de información confiables.** |
| **2.Diseño de la Arquitectura del Sistema** | **Utilizar herramientas de software adecuadas para la recopilación y organización eficiente de la información.** |
| 2.1: Definir las los modelos |  |
| 2.2: Crear las migraciones |  |
| 2.3: Definir las rutas. |  |
| **3.Implementación de Clases de Veterinario** |  |
| 3.1: Crear las modelos que implementan la interfaz de las entidades de veterinario |  |
| 3.2: Hacer la lógica necesaria dentro de cada clase. |  |
| **4.Hacer los Controladores** |  |
| 4.1: Configurar los controladores de cada entidad |  |
| 4.2: Asegurarse de que el controlador pueda utilizar diferentes servicios sin necesidad de cambiar su código. |  |
| **5.Pruebas y Verificación** |  |
| 5.1: Realizar pruebas para verificar que cada entidad puede realizar los servicios en las páginas Blade. |  |
| 5.2: Revisar y ajustar el código según los resultados de las pruebas para garantizar el correcto funcionamiento. |  |
|  |  |
|  |  |

**INSTRUCCIONES:** debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.

**LISTA DE RECURSOS**

**INSTRUCCIONES: completa la lista de recursos necesarios para la ejecución del trabajo.**

|  |  |  |
| --- | --- | --- |
| 1. MÁQUINAS Y EQUIPOS | | |
|  | 1 laptop |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 2. HERRAMIENTAS E INSTRUMENTOS | | |
|  | Laravel |  |
|  | Visual Studio Code |  |
|  | Word |  |
|  |  |  |
| Abrir Visual Studio Code desde el terminal de Mac OS X - return(GiS);    3. MATERIALES E INSUMOS | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

