****

**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 1** | | | |

1. **PLANIFICACIÓN DEL TRABAJO**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **N°** | **ACTIVIDADES/ ENTREGABLES** | **CRONOGRAMA/ FECHA DE ENTREGA** | | | | | | | | | |
|  |  | **26/10** | **27/10** | **28/10** | 30/10 |  |  |  |  |  |  |
|  | **Entregable 1** |  |  |  |  |  |  |  |  |  |  |
| 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 1** |  |  |  | X |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

1. **ENTREGABLES:**

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

|  |  |
| --- | --- |
| **Nº** | **ENTREGABLE 1** |
| **1** | **Entregable 1: WebVeterinary - Laravel**  En la primera fase, se desarrolló un sistema utilizando Laravel, donde se realizaron migraciones para definir la estructura de las tablas en la base de datos. Se crearon modelos que representaban las entidades del sistema: Mascota, Propietario, Veterinario y Medicamento, facilitando la interacción con los datos.  Se implementaron controladores, como MascotaController, PropietarioController, VeterinarioController y MedicamentoController, para gestionar las operaciones relacionadas con cada entidad. Las vistas se diseñaron utilizando Blade, lo que permitió crear interfaces de usuario dinámicas y fáciles de mantener.  Las rutas se configuraron en el archivo de rutas de Laravel, permitiendo definir las distintas URL del sistema y su relación con los controladores. Además, se configuró el archivo .env para establecer la conexión con la base de datos, asegurando que el sistema pudiera almacenar y recuperar datos de manera efectiva. |

|  |  |  |
| --- | --- | --- |
| **Desarrollo del Entregable N°1- 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.   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.  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 Pagina principal: Muestra una lista de los modulos.**      * **Vista de Mascotas**: Muestra una lista de todas las mascotas registradas y permite acceder a formularios.      * **Vista de Propietarios**: Presenta la información de los propietarios, permitiendo gestionar sus datos.      * **Vista de Veterinarios**: Permite visualizar y gestionar la información del personal veterinario.      * **Vista de Medicamentos**: Muestra los medicamentos disponibles y permite gestionar su información.     Extras para el funcionamiento de los datos y poder mostralos  Migraciones de Veterinario, Medicamento, Propietario y Mascota. | | |
|  | **[NOMBRE DEL TRABAJO] FRAMEWORK LARAVEL PARTE 1** | **[ESCALA]** |
| C:\Unidad_D\Nuevo Logo\SENATI_FF-01 Modificado.jpg | **[APELLIDOS Y NOMBRES] SAAVEDRA SUCA JERSON** | **1** |

**HOJA DE PLANIFICACIÓN (Entregable 1)**

**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 | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

