

**Evidencia de producto: GA7-220501096-AA4-EV03 Componente frontend del
proyecto formativo y proyectos de clase (listas de chequeo)**

Presentado por: Jhonny Leonardo Chaparro Cetina
Análisis y Desarrollo de Software

Instructor: Área ADSO
Milton Iván Barbosa Gaona

Centro de la Tecnología del Diseño y de la Productividad Empresarial
Servicio Nacional de Aprendizaje SENA
Girardot – Cundinamarca

2025

Tabla de Contenido

Introducción	5
Justificación	6
Objetivos	7
Objetivo General.....	7
Objetivos Específicos.....	7
1. Codificación de Módulos del Software Según Requerimientos del Proyecto	8
2. Diagrama de Clases	8
3. Diagramas de Casos de Uso.....	9
4. Historias de Usuario	13
5. Diseño.....	16
6. Prototipo	17
6.1. Prototipo Página Inicio.....	18
6.2. Prototipo Página Registro	19
6.3. Prototipo Validación de Usuario y Credenciales	20
6.4. Prototipo Página Menu	20
6.5. Prototipo Restauración de Contraseña	21
6.6. Prototipo Planes de Entrenamiento	22
6.7. Prototipo Planes de Alimentación	23
6.8. Prototipo Conexión o Socializar	24
6.9. Prototipo Pagos	25
7. Informe Técnico de Plan de Trabajo para Construcción de Software	25
7.1. Perspectiva del Proyecto	26
7.2. Funciones del Proyecto.....	26
8. Sistemas de Control de Versiones	27
8.1. Sistemas de control de versiones locales.....	28
8.2. Sistemas de control de versiones Centralizadas	28
8.3. Sistemas de control de versiones Distribuidas.....	29
9. Sistema de Control de Versiones Git.....	29
9.1. Breve Historia de Git.....	29
9.2. Fundamentos de Git	30
9.3. Comandos básicos de Git	36
10. Git en Entornos Remotos	39

11. Plataformas para Implementar Integración Continua - Gitlab	41
Conclusiones	43
Bibliografía.....	44

Contenido de Imágenes

Tabla 1 Caso de Uso Registrar	9
Tabla 2 Caso de Uso Configurar	10
Tabla 3 Caso de Uso Planes de Ejercicio	10
Tabla 4 Caso de Uso Planes de Alimentación	11
Tabla 5 Caso de Uso Socialización	12
Imagen 1 Historia de Usuario - Registro Usuario	13
Imagen 2 Historia de Usuario - Configurar Perfil Usuario	14
Imagen 3 Historia de Usuario - Acceso Planes de Ejercicio.....	14
Imagen 4 Historia de Usuario - Acceso Planes de Alimentación	15
Imagen 5 Historia de Usuario - Participación Actividades Socialización	15
Imagen 6 Historia de Usuario - Efectuar Pago	16
Imagen 7 Diseño Plataforma Proyecto EFM	17
Ilustración 1 Prototipo Pagina Inicio	18
Ilustración 2 Prototipo Pagina de Registro.....	19
Ilustración 3 Prototipo Validación de Usuario y Credenciales.....	20
Ilustración 4 Prototipo Pagina Menu	21
Ilustración 5 Prototipo Restauración Contraseña	21
Ilustración 6 Prototipo Pagina Planes de Entrenamiento	22
Ilustración 7 Prototipo Pagina Planes de Alimentación	23
Ilustración 8 Prototipo Pagina Conexión.....	24
Ilustración 9 Prototipo Pagina de Pagos	25
Ilustración 1 Sistemas de Control de Versiones - Posibilidades	27
Ilustración 2 Los Tres Estados	34
Ilustración 3 Secciones Principales Git	35

Introducción

Hoy en día, la tecnología se ha convertido en una gran aliada para mejorar nuestra calidad de vida. A través de diferentes aplicaciones, es posible organizar rutinas de ejercicio, llevar un control de la alimentación y hasta compartir experiencias con otras personas que buscan objetivos similares. En este trabajo se presenta la codificación de los módulos de un software diseñado para apoyar a los usuarios en su proceso de bienestar físico y mental, integrando en un mismo espacio herramientas de entrenamiento, nutrición y socialización. Con ello, se busca ofrecer una solución práctica, accesible y segura que responda a las necesidades actuales de quienes quieren llevar un estilo de vida más saludable.

Justificación

Contar con una plataforma que reúna en un solo lugar actividades de ejercicio, planes de alimentación y espacios de interacción social es una necesidad creciente. Muchas veces, las personas utilizan varias aplicaciones para estos fines, lo que puede resultar confuso, costoso o poco motivador. La codificación modular del software permite que cada parte del sistema (ejemplo: rutinas, alimentación, comunidad) funcione de manera independiente pero conectada entre sí, asegurando una experiencia más organizada y sencilla para el usuario. De esta forma, se garantiza no solo el cumplimiento de los requerimientos del proyecto, sino también el aporte a la salud y bienestar de las personas.

Objetivos

Objetivo General

Desarrollar la codificación de los módulos del software de la plataforma integral de ejercicio, alimentación y socialización, con el fin de ofrecer una herramienta práctica y confiable que promueva hábitos saludables y mejore la calidad de vida de sus usuarios.

Objetivos Específicos

Identificar las funciones principales que debe cumplir cada módulo (ejercicio, alimentación, socialización, reportes).

Organizar el sistema en módulos que trabajen de manera clara, ordenada y fácil de mantener.

Programar cada módulo siguiendo buenas prácticas de desarrollo y cuidando la experiencia del usuario.

Asegurar que los módulos se integren de forma correcta mediante pruebas y ajustes.

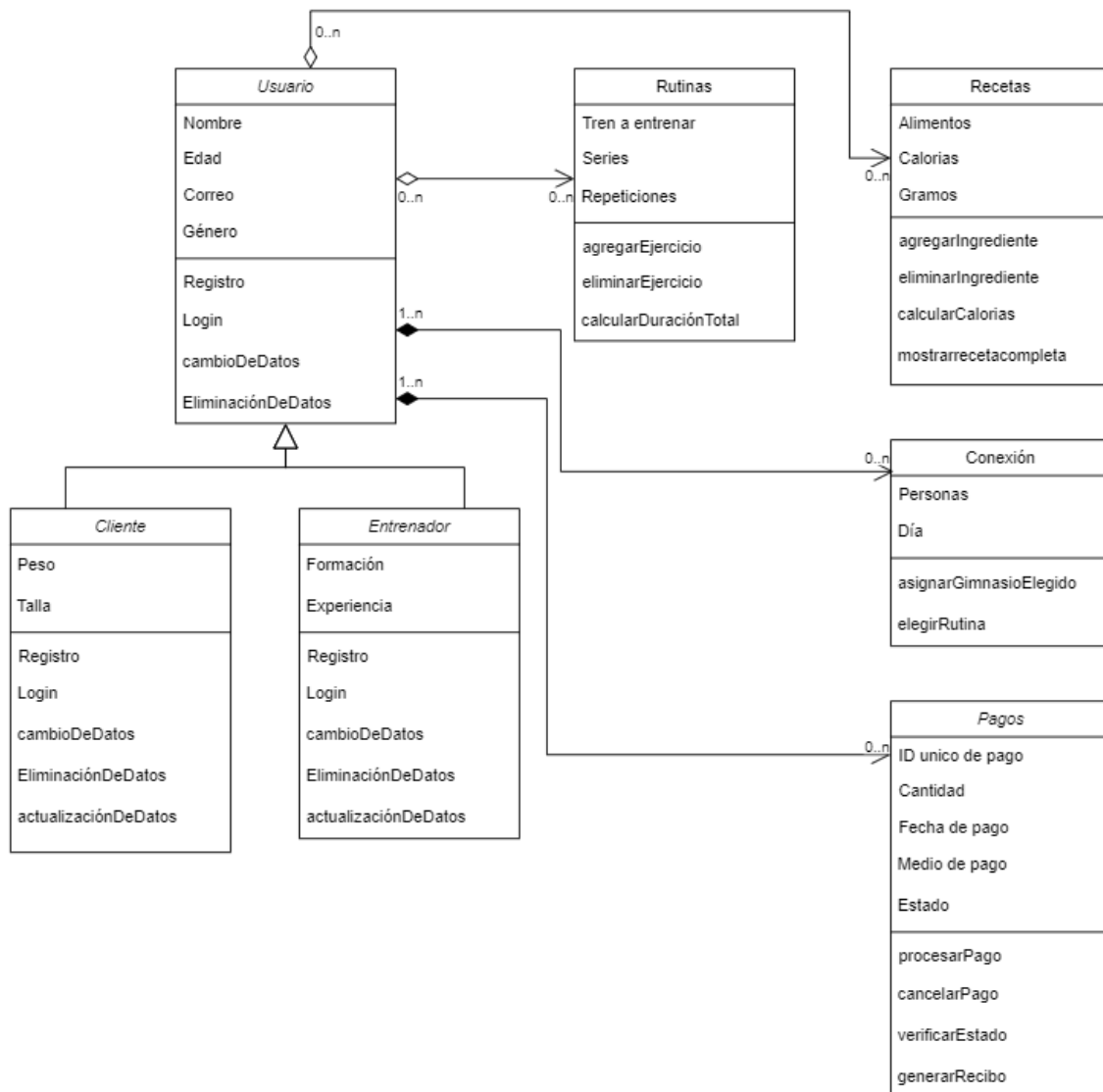
Documentar el proceso de codificación para facilitar su uso, mantenimiento y futuras mejoras.

Evidencia de producto: GA7-220501096-AA4-EV03 Componente frontend del proyecto formativo y proyectos de clase (listas de chequeo)

1. Componentes frontend del proyecto formativo y proyecto de clases (SENA, SENA, s.f.)

2. Diagrama de Clases

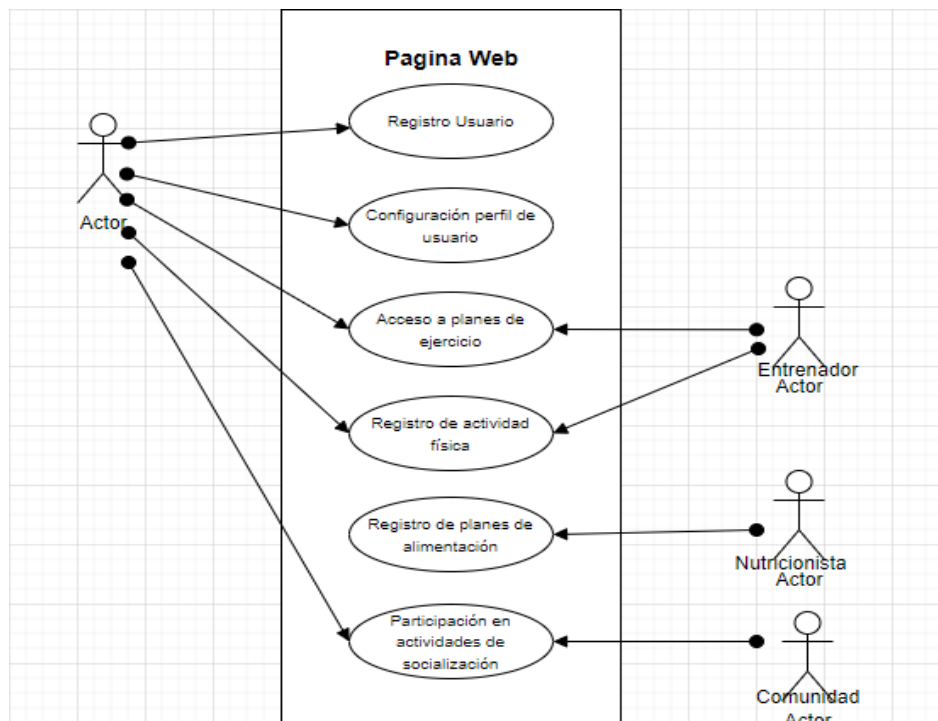
Diagrama 1 Clases Proyecto EFM



Nota Diagrama Creado por, Leonardo Chaparro

3. Diagramas de Casos de Uso

Diagrama 2 Casos de Uso Proyecto EFM



Nota Diagrama Creado por, Leonardo Chaparro

Tabla 1 Caso de Uso Registrar

Caso de Uso	Registrar
Descripción	Registro de usuario
Precondición	El usuario debe tener acceso a la plataforma (web o móvil).
Secuencia normal	Inicio del Registro
	Ingreso de Información Personal
	Confirmación de Contraseña
	Aceptación de Términos y Condiciones
	Verificación de Correo Electrónico
	Confirmación del Correo Electrónico
	Finalización del Registro
Postcondición	El usuario se encuentra registrado
Excepciones	Correo Electrónico ya Registrado

	Fallo en la Verificación del Correo
	Error en el Proceso de Registro

Nota Tabla Creada por, Leonardo Chaparro

Tabla 2 Caso de Uso Configurar

Caso de Uso	Configurar
Descripción	Configurar perfil del usuario
Precondición	El usuario debe estar registrado e iniciar sesión en la plataforma El usuario debe tener acceso a la plataforma (web o móvil).
Secuencia normal	Acceso a la Configuración del Perfil
	Ingreso de Información Personal
	Definición de Objetivos de Bienestar
	Preferencias de ejercicio
	Restricciones alimenticias
	Configuración de Recordatorios y Notificaciones
	Revisión y confirmación
	Guardado de la configuración
Postcondición	El usuario configuró su perfil
Excepciones	Información incompleta
	Error al guardar la configuración
	Inconsistencia en los Datos
	Sesión Vencida

Nota Tabla Creada por, Leonardo Chaparro

Tabla 3 Caso de Uso Planes de Ejercicio

Caso de Uso	Planes de ejercicio
Descripción	Acceso a Planes de Ejercicio
Precondición	El usuario debe estar registrado, haber iniciado sesión en la plataforma y tener su perfil.
	La plataforma debe tener planos de ejercicio disponibles y configurados según las necesidades del usuario
Secuencia normal	Acceso a la Sección de Planes de Ejercicio
	Visualización de Planos Disponibles
	Selección de un Plan de Ejercicio
	Visualización de Detalles del Plan
	Personalización del Plan
	Confirmación y Guardado del Plan
	Inicio del Plan de Ejercicio
Postcondición	El usuario configuró su perfil
Excepciones	No hay planes disponibles
	Error al cargar los planes

	Error al guardar el plan seleccionado
	Planes Premium Bloqueados

Tabla 4 Caso de Uso Planes de Alimentación

Nota Tabla Creada por; Leonardo Chaparro

Caso de Uso	Planes de alimentación
Descripción	Acceso a Planes de Alimentación
Precondi- ción	El usuario debe estar registrado y haber iniciado sesión en la plataforma.
	El usuario debe tener su perfil configurado, incluyendo preferencias dietéticas y objetivos de salud
	La plataforma debe tener planes de alimentación disponibles que se ajusten al perfil del usuario
Secuencia normal	Acceso a la Sección de Planes de Alimentación
	Visualización de Planes Disponibles
	Selección de un Plan de Alimentación
	Visualización de Detalles del Plan
	Confirmación del Plan
	Personalización del Plan
	Seguimiento del plan
Postcondi- ción	El usuario definió su plan de alimentación
Excepciones	No hay Planes Disponibles que Coincidan
	Error al cargar los planes
	Error al guardar el plan seleccionado
	Planes Premium Bloqueados

Nota Tabla Creada por, Leonardo Chaparro

Tabla 5 Caso de Uso Socialización

Caso de Uso	Socialización
Descripción	Participación en Actividades de Socialización
Precondición	El usuario debe estar registrado y haber iniciado sesión en la plataforma.
	Deben existir actividades de socialización disponibles en la plataforma
	El usuario debe tener un perfil configurado para poder participar en las actividades de socialización
Secuencia normal	Acceso a la Sección de Actividades de Socialización
	Exploración de actividades disponibles
	Selección de una actividad
	Visualización de detalles de la actividad
	Confirmación de Participación:
	Interacción Social
	Seguimiento de la Actividad
Postcondición	El usuario participa activamente en una actividad de socialización, lo que se registra en su perfil.
	El usuario puede interactuar con otros participantes, mejorar sus conexiones sociales y recibir retroalimentación
	La plataforma actualiza el historial de actividades sociales del usuario, que puede influir en futuras actividades
Excepciones	Actividad Lleno o Cerrada
	Error al Unirse a la Actividad
	Incompatibilidad del Perfil
	Cancelación o Cambio de Actividad

Nota Tabla Creada por, Leonardo Chaparro

Tabla 6 Caso de Uso Pagos

Caso de Uso	Pagos
Descripción	Efectuar Pago
Precondición	El usuario debe estar registrado y haber iniciado sesión en la plataforma.
	El usuario debe haber seleccionado un servicio o producto que requiere un pago
	El usuario debe tener acceso a un método de pago válido (Efectivo o digital).
Secuencia normal	Selección del Producto o Servicio
	Inicio del Proceso de Pago
	Ingreso de Información de Pago
	Revisión de la Compra
	Confirmación del Pago
	Procesamiento del Pago
	Notificación de Transacción Exitosa
	Recepción del Comprobante de Pago
Postcondición	El pago del usuario se procesa exitosamente, y el servicio o producto adquirido se activa o entrega
	El usuario recibe un comprobante de pago
	La cuenta del usuario se actualiza para reflejar los beneficios asociados con la compra

Excepciones	Pago Rechazado
	Error de Conexión
	Problemas con el Proveedor de Pago
	Transacción Duplicada
	Error en la Confirmación de Pago

Nota Tabla Creada por, Leonardo Chaparro

4. Historias de Usuario

REGISTRO DE USUARIO

Imagen 1 Historia de Usuario - Registro Usuario

Identificador (ID) de la Historia	Enunciado de la Historia				Criterios de Aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
HDU1	Como un usuario nuevo	Necesito registrarme en la plataforma	Para acceder a las funcionalidades de ejercicio físico, socialización y planes de alimentación personalizados que la plataforma ofrece.	1	El formulario de registro debe permitir al usuario ingresar su nombre, dirección de correo electrónico y crear una contraseña segura.	Un nuevo usuario visita la plataforma por primera vez y desea registrarse para utilizar los servicios ofrecidos	El usuario selecciona la opción "Registrarse", completa el formulario con la información requerida y envía el formulario	El sistema crea una cuenta para el usuario, envía un correo de confirmación, y tras la verificación de la cuenta, permite al usuario iniciar sesión y acceder a la configuración inicial de su perfil.
				2	El sistema debe validar los datos ingresados y enviar un correo electrónico de confirmación.			
				3	Tras la verificación del correo, el usuario debe poder iniciar sesión y acceder al perfil inicial de configuración.			

Nota Imagen Creada por, Leonardo Chaparro

CONFIGURAR PERFIL DE USUARIO

Imagen 2 Historia de Usuario - Configurar Perfil Usuario

HDU2	Como usuario recién registrado	Quiero configurar mi perfil personal en la plataforma	Para recibir recomendaciones personalizadas de ejercicio, socialización y alimentación que se adapten a mis objetivos y preferencias	1	El sistema debe permitir al usuario ingresar o modificar información personal relevante, como edad, género, peso, altura, nivel de actividad física, preferencias dietéticas y objetivos de salud.	Un usuario recién registrado inicia sesión en la plataforma por primera vez y se le solicita configurar su perfil para optimizar su experiencia.	El usuario accede a la sección de configuración de perfil, ingresa la información solicitada y confirma los datos.	El sistema guarda la configuración del perfil del usuario y adapta las recomendaciones y contenidos de la plataforma en función de la información proporcionada, mejorando la relevancia de los servicios ofrecidos.
				2	El usuario debe poder revisar y confirmar los datos ingresados antes de guardarlos.			
				3	Tras la configuración del perfil, la plataforma debe ofrecer recomendaciones personalizadas en función de la información proporcionada.			

Nota Imagen Creada por, Leonardo Chaparro

ACCESO A PLANES DE EJERCICIO

Imagen 3 Historia de Usuario - Acceso Planes de Ejercicio

HDU3	Como usuario	Quiero acceder a planes de ejercicio personalizados en la plataforma.	Para seguir un programa de ejercicio adaptado a mis objetivos y nivel de condición física que me ayude a mejorar mi bienestar físico.	1	El sistema debe mostrar una lista de planes de ejercicio disponibles que se ajusten al perfil del usuario (nivel de condición física, objetivos, preferencias).	Un usuario ha configurado su perfil y desea comenzar a realizar ejercicios utilizando la plataforma.	El usuario accede a la sección de "Planes de Ejercicio", explora las opciones disponibles, selecciona un plan que se ajusta a sus necesidades y lo activa.	El sistema asigna el plan de ejercicio seleccionado al perfil del usuario, permitiéndole seguir el programa, recibir recordatorios, y registrar su progreso diario.
				2	El usuario debe poder seleccionar y ver los detalles de cada plan de ejercicio, incluyendo la duración, intensidad, y objetivos específicos.			
				3	Una vez seleccionado un plan, el usuario debe poder iniciar el seguimiento del plan, recibir recordatorios y registrar su progreso.			

Nota Imagen Creada por, Leonardo Chaparro

ACCESO A PLANES DE ALIMENTACIÓN

Imagen 4 Historia de Usuario - Acceso Planes de Alimentación

HDU4	Como usuario	Quiero acceder a planes de alimentación personalizados en la plataforma.	Para seguir un plan de alimentación que se ajuste a mis objetivos de salud y preferencias dietéticas, mejorando mi bienestar general.	1	El sistema debe mostrar una lista de planes de alimentación disponibles que se adapten a las preferencias dietéticas y objetivos de salud del usuario (por ejemplo, pérdida de peso, ganancia muscular, dietas específicas).	Un usuario que ya ha configurado su perfil en la plataforma desea comenzar a seguir un plan de alimentación que lo ayude a alcanzar sus metas de salud.	El usuario navega a la sección de "Planes de Alimentación", explora las opciones disponibles, selecciona un plan que se ajuste a sus necesidades y lo activa.	El sistema asigna el plan de alimentación seleccionado al perfil del usuario, permitiéndole seguir las recomendaciones diarias, recibir recordatorios, y registrar su progreso en la plataforma.
				2	El usuario debe poder ver los detalles de cada plan, incluyendo recetas, valores nutricionales, y una lista de compras.			
				3	Una vez seleccionado un plan, el usuario debe poder activarlo y recibir recomendaciones y recordatorios diarios, así como registrar su cumplimiento.			

Nota Imagen Creada por, Leonardo Chaparro

PARTICIPACIÓN EN ACTIVIDADES DE SOCIALIZACIÓN

Imagen 5 Historia de Usuario - Participación Actividades Socialización

HDU5	Como usuario	Quiero participar en actividades de socialización organizadas por la plataforma.	Para conectar con otras personas que tienen intereses similares, mejorando mi bienestar emocional y fortaleciendo mi red de apoyo social.	1	El sistema debe mostrar una lista de actividades de socialización disponibles, como grupos de interés, eventos comunitarios, o desafíos en grupo.	Un usuario, que ya ha configurado su perfil y está utilizando los servicios de la plataforma, desea participar en actividades sociales que complementen su proceso de mejoramiento físico y mental.	El usuario accede a la sección de "Actividades de Socialización", revisa las opciones disponibles, se inscribe en una actividad que le interese y confirma su participación.	El sistema registra la inscripción del usuario en la actividad seleccionada, le proporciona la información necesaria y facilita su participación activa antes y durante el evento, promoviendo la interacción y socialización entre los usuarios.
				2	El usuario debe poder inscribirse en una actividad, recibir detalles sobre la misma, y ver quién más participará.			
				3	Una vez inscrito, el usuario debe recibir recordatorios y actualizaciones sobre la actividad, y poder participar en discusiones o interacciones previas a la actividad a través de la plataforma.			

Nota Imagen Creada por, Leonardo Chaparro

EFFECTUAR PAGO

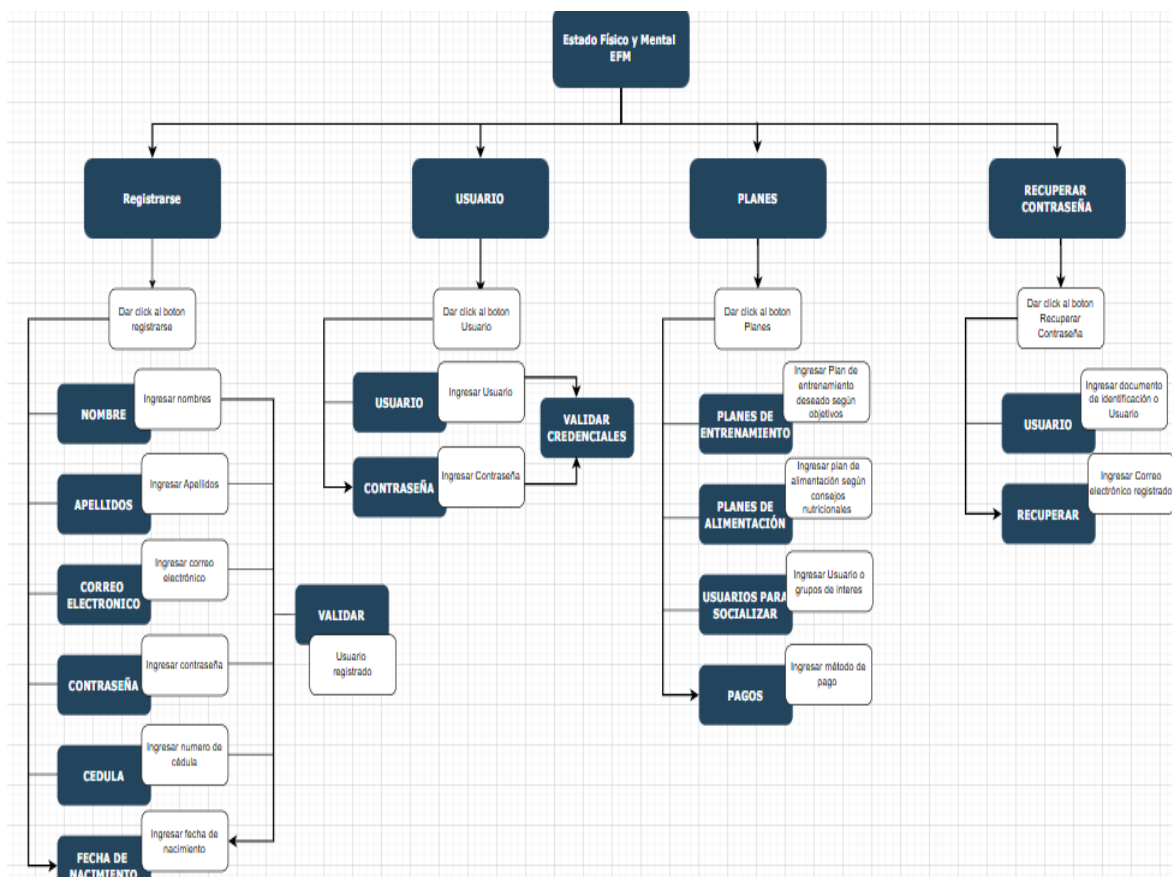
Imagen 6 Historia de Usuario - Efectuar Pago

H DU6	Como usuario	Quiero efectuar un pago en la plataforma para adquirir un servicio o producto.	Para desbloquear funcionalidades avanzadas, como planes de ejercicio o alimentación personalizados, y participar en eventos exclusivos.	1	El sistema debe permitir al usuario seleccionar un método de pago y realizar la transacción de manera segura.	Un usuario que ya utiliza la plataforma decide adquirir una suscripción premium, un plan de alimentación especializado o cualquier otro servicio que requiera un pago.	El usuario selecciona un servicio o producto en la plataforma, accede a la sección de pago, ingresa los datos requeridos y confirma la transacción.	El sistema procesa el pago, actualiza el perfil del usuario con el nuevo servicio adquirido, y envía una confirmación de la transacción junto con un recibo digital. El usuario puede inmediatamente acceder a las funciones o productos adquiridos.
				2	El pago debe ser procesado exitosamente, y el usuario debe recibir una confirmación de la transacción, junto con un recibo digital.			
				3	Tras el pago, el servicio o producto adquirido debe estar inmediatamente disponible para el usuario.			

Nota Imagen Creada por, Leonardo Chaparro

5. Diseño

Imagen 7 Diseño Plataforma Proyecto EFM



Nota Imagen Creada por, Leonardo Chaparro

6. Prototipo

En la presentación del Proyecto EFM realice los siguientes prototipos del software a trabajar así:

Página de inicio

Registro

Validación de Credenciales

Menu

Recuperar Contraseña

Planes de Entrenamiento

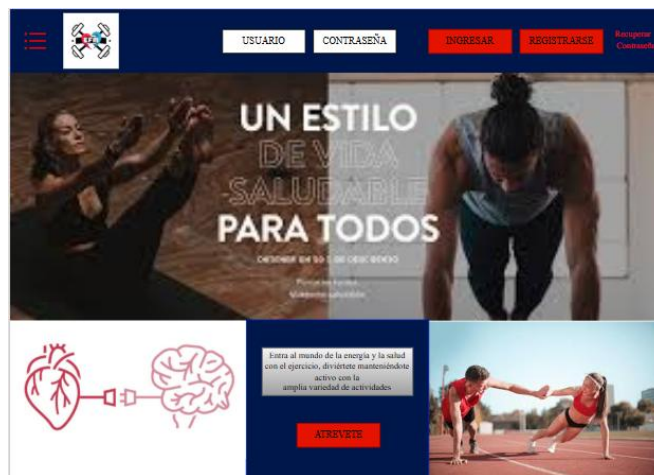
Planes de Alimentación

Conexión

Pagos

6.1. Prototipo Página Inicio

Ilustración 1 Prototipo Pagina Inicio



Nota Ilustración Creada por, Leonardo Chaparro

La interfaz principal de la plataforma, también conocida como pantalla de bienvenida, es el primer punto de contacto entre el usuario y el sistema. Su diseño está orientado a ser claro, intuitivo y motivador. Incluye los siguientes elementos:

Logo del proyecto: Ubicado en una zona visible parte superior.

Mensaje motivacional: Frase inspiradora que da la bienvenida al usuario y refuerza el propósito del proyecto ubicada en la parte inferior.

Menú: Permite acceder fácilmente a secciones clave del proyecto.

Campos de acceso: espacios para que el usuario introduzca su nombre de usuario y contraseña.

Botones de acción:

Iniciar sesión

Registrarse

Recuperar contraseña

Esta pantalla está diseñada no solo para permitir el acceso al sistema, sino también para motivar y generar confianza en los usuarios desde el primer momento.

6.2. Prototipo Página Registro

Ilustración 2 Prototipo Pagina de Registro

Ilustración de un prototipo de la página de registro. La interfaz tiene un encabezado azul oscuro con un logo de un corazón y un cerebro conectados por una línea, y un botón rojo "INICIO". El cuerpo principal es gris claro y contiene el título "Datos de registro" y seis campos de entrada organizados en una cuadrícula de 2x3: Nombres, Apellidos, Correo electrónico, Contraseña, Cédula y Fecha de Nacimiento. En la parte inferior, hay una barra azul oscura con un botón rojo "¡¡¡REGÍSTRATE!!!". A la izquierda de la barra, hay un icono de un corazón y un cerebro conectados por una línea. A la derecha, hay una imagen de dos corredores en una pista de atletismo.

Nota Ilustración Creada por, Leonardo Chaparro

- Se da clic en el ejecutable del programa y se diligencian los campos requeridos.
- Navegar a la página de registro.
- Ingresar datos válidos.

- Hacer clic en Registrarse.
- El sistema muestra un mensaje de éxito y el usuario es redirigido a la página de inicio.

6.3. Prototipo Validación de Usuario y Credenciales

Ilustración 3 Prototipo Validación de Usuario y Credenciales

Nota Ilustración Creada por, Leonardo Chaparro

- Se da clic en el ejecutable del programa y se diligencian los campos requeridos.
- El usuario está registrado en el sistema con el email y la contraseña proporcionados.
- El sistema muestra un mensaje de éxito y redirige al usuario a la página de bienvenida.

6.4. Prototipo Página Menu

Ilustración 4 Prototipo Pagina Menu

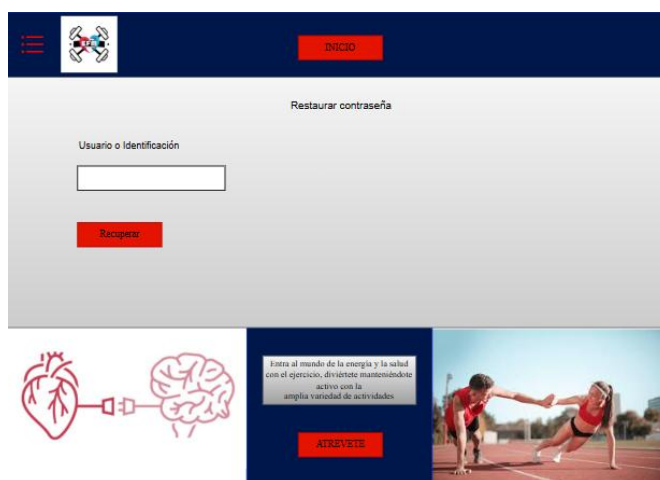


Nota Ilustración Creada por, Leonardo Chaparro

- El usuario ha iniciado sesión y está en la página de Menu búsqueda de planes de entrenamiento, planes de alimentación, conexión y pagos.
- Seleccionar tipo de plan según criterios de búsqueda.

6.5. Prototipo Restauración de Contraseña

Ilustración 5 Prototipo Restauración Contraseña



Nota Ilustración Creada por, Leonardo Chaparro

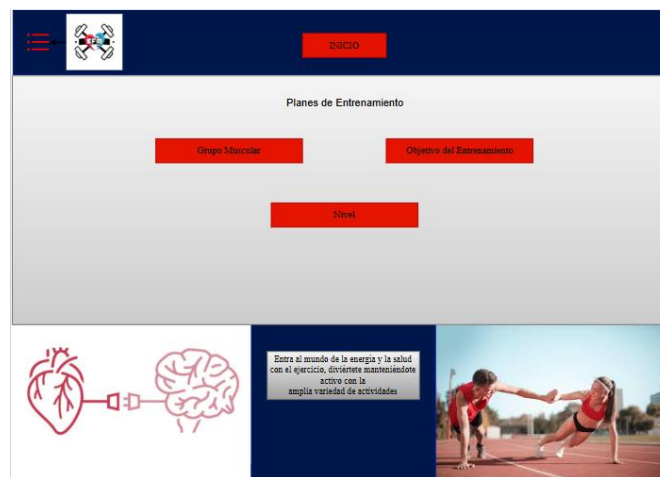
- Se da clic en el ejecutable del programa y se diligencian los campos requeridos.

- Verificar la restauración de contraseña con un email registrado.
- El email está registrado en el sistema.
- El sistema muestra un mensaje de éxito indicando que se ha enviado un enlace de restauración al email proporcionado.

- Navegar al enlace de restauración recibido en el email
- Ingresar la nueva contraseña
- Confirmar la nueva contraseña
- Hacer clic en Cambiar contraseña.
- El sistema muestra un mensaje de éxito indicando que la contraseña se ha cambiado correctamente.

6.6. Prototipo Planes de Entrenamiento

Ilustración 6 Prototipo Pagina Planes de Entrenamiento



Nota Ilustración Creada por, Leonardo Chaparro

- Navegar a la página de búsqueda de planes de entrenamiento.

- El usuario ha iniciado sesión y está en la página de búsqueda de planes de entrenamiento.
- Seleccionar tipo de ejercicio.
 - El sistema muestra una lista de planes de entrenamiento que coinciden con los criterios.

6.7. Prototipo Planes de Alimentación

Ilustración 7 Prototipo Pagina Planes de Alimentación

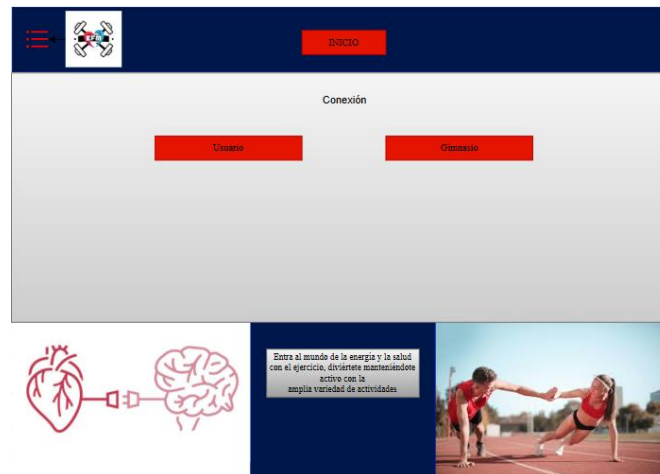


Nota Ilustración Creada por, Leonardo Chaparro

- Navegar a la página de búsqueda de planes de alimentación.
- El usuario ha iniciado sesión y está en la página de búsqueda de planes de alimentación.
- Seleccionar tipo de receta.
- El sistema muestra una lista de planes de alimentación que coinciden con los criterios.

6.8. Prototipo Conexión o Socializar

Ilustración 8 Prototipo Pagina Conexión

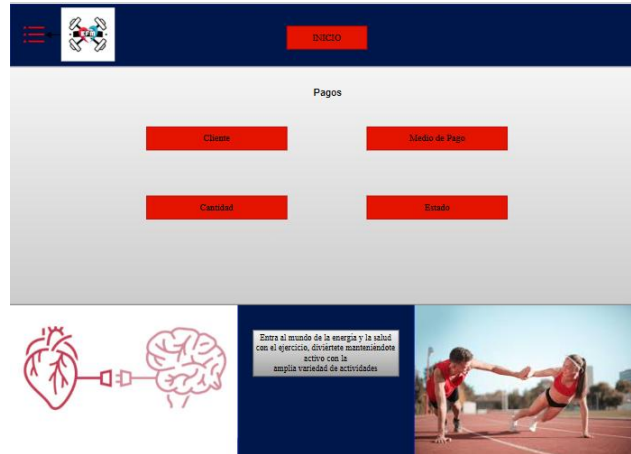


Nota Ilustración Creada por, Leonardo Chaparro

- Navegar a la página de búsqueda de Conexión.
- El usuario ha iniciado sesión y está en la página de búsqueda de planes de Conexión.
- Seleccionar usuario y/o gimnasio asignado.
- El sistema muestra una lista de conexiones que coinciden con los criterios de búsqueda.

6.9. Prototipo Pagos

Ilustración 9 Prototipo Pagina de Pagos



Nota Ilustración Creada por, Leonardo Chaparro

- Navegar a la página de Pagos.
- El usuario ha iniciado sesión y está en la página de Pagos.
- Seleccionar usuario, medio de pago y cantidad.
- El sistema muestra estado de pago.

7. Informe Técnico de Plan de Trabajo para Construcción de Software

PROYECTO ESTADO FÍSICO Y MENTAL EFM: Donde su objetivo principal del proyecto es apoyar el proceso de mejoramiento del estado físico y mental de los usuarios mediante una plataforma integral que promueva el ejercicio físico, la socialización y una alimentación saludable.

7.1. Perspectiva del Proyecto

El producto es una plataforma web cuyo objetivo principal es mejorar el bienestar físico y mental de los usuarios. Para lograrlo, combina tres áreas fundamentales: ejercicio físico, socialización y alimentación saludable. Esta herramienta busca ofrecer una experiencia accesible y práctica, adaptada a las necesidades individuales de cada usuario.

7.2. Funciones del Proyecto

El software incorpora funciones clave que garantizan su utilidad y efectividad. Estas funciones son:

- **Ejercicio físico:** Generar y sugerir rutinas de entrenamiento personalizadas, acompañado de videos explicativos. Permitir el registro del progreso del usuario y mostrar estadísticas claras.
- **Alimentación saludable:** Ofrecer planes alimenticios adaptados a los objetivos individuales, junto con recetas fáciles de seguir. Proporcionar una lista de compras automatizada basada en los planes seleccionados.
- **Socialización:** Facilitar la interacción entre los usuarios a través de foros, grupos de interés y eventos grupales. Permitir la conexión con otros usuarios con metas similares.
- **Accesibilidad:** Garantizar que todas las funciones sean inclusivas, como navegación por teclado, textos alternativos y diseño optimizado para diferentes dispositivos.

8. Sistemas de Control de Versiones

Es una herramienta que permite gestionar, registrar y controlar los cambios realizados en archivos a lo largo del tiempo. de tal forma que estos puedan ser recuperados con precisión de acuerdo a las necesidades de los usuarios. A este tipo de procesos se le puede denominar versionamiento y si bien este tipo de actividades son comunes en archivos o código fuente en la industria del desarrollo de software, este es un proceso aplicable a cualquier tipo de archivo.

Se utiliza principalmente en el desarrollo de software para permitir que varios desarrolladores colaboren sobre el mismo código sin perder historial ni provocar conflictos.

Ilustración 10 Sistemas de Control de Versiones - Posibilidades



Nota Material de Formacion SENA

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF29/index.html#/curso/tema1>

Respecto a los sistemas de control de versiones estos pueden ser categorizados según sus características en tres grupos: sistemas de control de

versiones locales, sistemas de control de versiones centralizados y sistemas de control de versiones distribuidos.

8.1. Sistemas de control de versiones locales

Una forma de llevar control de versiones locales es el crear copias manuales en un directorio exclusivo o de fácil reconocimiento donde llevamos registro de los cambios realizados sobre un documento particular, generalmente por medio de una numeración o incluso llevando registro de la fecha de realización de la versión, Lo anterior, aunque puede ser una práctica común y fácil de implementar, resultaría muy riesgosa, ya que la posibilidad de introducir errores, olvidar la ubicación del directorio, sobrescribir por accidente un archivo o nombrar de forma incorrecta un archivo es muy alta. (SENA, SENA, s.f.)

Para solucionar este problema aparecieron hace un tiempo los sistemas de control de versiones locales, los cuales contenían una base de datos simple donde se registraba información de todos los cambios realizados sobre los archivos

8.2. Sistemas de control de versiones Centralizadas

Es un sistema donde todos los archivos del proyecto y sus versiones se guardan en un único servidor central. Las personas que trabajan en el proyecto descargan una copia desde ese servidor, hacen cambios en ella y luego los suben nuevamente para compartirlos. Todo el equipo depende de ese mismo punto para trabajar, por lo que se necesita estar conectado al servidor para actualizar o colaborar. Ejemplos de sistemas centralizados son SVN (Subversion) y CVS (Concurrent Versions System). (Conservancy, s.f.)

8.3. Sistemas de control de versiones Distribuidas

Es un sistema donde cada persona tiene una copia completa del proyecto, incluyendo todo el historial de cambios. Esto le permite trabajar de manera local, sin necesidad de estar siempre conectado al servidor. Los cambios pueden compartirse y unirse con otras copias, lo que facilita la colaboración, incluso si alguien está desconectado o en una zona diferente. Ejemplos de sistemas distribuidos son Git y Mercurial. (Conservancy, s.f.)

9. Sistema de Control de Versiones Git

9.1. Breve Historia de Git

Como muchas de las grandes cosas en esta vida, Git comenzó con un poco de destrucción creativa y una gran polémica.

El kernel de Linux es un proyecto de software de código abierto con un alcance bastante amplio. Durante la mayor parte del mantenimiento del kernel de Linux (1991-2002), los cambios en el software se realizaban a través de parches y archivos. En el 2002, el proyecto del kernel de Linux empezó a usar un DVCS propietario llamado BitKeeper.

En el 2005, la relación entre la comunidad que desarrollaba el kernel de Linux y la compañía que desarrollaba BitKeeper se vino abajo y la herramienta dejó de ser ofrecida de manera gratuita. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que

aprendieron mientras usaban BitKeeper. Algunos de los objetivos del nuevo sistema fueron los siguientes:

- Velocidad
- Diseño sencillo
- Gran soporte para desarrollo no lineal (miles de ramas paralelas)
- Completamente distribuido
- Capaz de manejar grandes proyectos (como el kernel de Linux) eficientemente (velocidad y tamaño de los datos)

Desde su nacimiento en el 2005, Git ha evolucionado y madurado para ser fácil de usar y conservar sus características iniciales. Es tremendamente rápido, muy eficiente con grandes proyectos y tiene un increíble sistema de ramificación (branching) para desarrollo no lineal.

9.2. Fundamentos de Git

Entonces, ¿qué es Git en pocas palabras? Es muy importante entender bien esta sección, porque si entiendes lo que es Git y los fundamentos de cómo funciona, probablemente te será mucho más fácil usar Git efectivamente. A medida que aprendas Git, intenta olvidar todo lo que posiblemente conoces acerca de otros VCS como Subversion y Perforce. Hacer esto te ayudará a evitar confusiones sutiles a la hora de utilizar la herramienta. Git almacena y maneja la información de forma muy diferente a esos otros sistemas, a pesar de que su interfaz de usuario es bastante similar. Comprender esas diferencias evitará que te confundas a la hora de usarlo.

Copias instantáneas, no diferencias

La principal diferencia entre Git y cualquier otro VCS (incluyendo Subversion y sus amigos) es la forma en la que manejan sus datos.

Conceptualmente, la mayoría de los otros sistemas almacenan la información como una lista de cambios en los archivos. Estos sistemas (CVS, Subversion, Perforce, Bazaar, etc.) manejan la información que almacenan como un conjunto de archivos y las modificaciones hechas a cada uno de ellos a través del tiempo.

Casi todas las operaciones son locales

La mayoría de las operaciones en Git sólo necesitan archivos y recursos locales para funcionar. Por lo general no se necesita información de ningún otro computador de tu red. Si estás acostumbrado a un CVCS donde la mayoría de las operaciones tienen el costo adicional del retardo de la red, este aspecto de Git te va a hacer pensar que los dioses de la velocidad han bendecido Git con poderes sobrenaturales. Debido a que tienes toda la historia del proyecto ahí mismo, en tu disco local, la mayoría de las operaciones parecen prácticamente inmediatas.

Por ejemplo, para navegar por la historia del proyecto, Git no necesita conectarse al servidor para obtener la historia y mostrártela - simplemente la lee directamente de tu base de datos local. Esto significa que ves la historia del proyecto casi instantáneamente. Si quieres ver los cambios introducidos en un archivo entre la versión actual y la de hace un mes, Git puede buscar el archivo de hace un mes y hacer un cálculo de diferencias localmente, en lugar de tener que

pedirle a un servidor remoto que lo haga, u obtener una versión antigua desde la red y hacerlo de manera local.

Esto también significa que hay muy poco que no puedes hacer si estás desconectado o sin VPN. Si te subes a un avión o a un tren y quieres trabajar un poco, puedes confirmar tus cambios felizmente hasta que consigas una conexión de red para subirlos. Si te vas a casa y no consigues que tu cliente VPN funcione correctamente, puedes seguir trabajando. En muchos otros sistemas, esto es imposible o muy engorroso. En Perforce, por ejemplo, no puedes hacer mucho cuando no estás conectado al servidor. En Subversion y CVS, puedes editar archivos, pero no puedes confirmar los cambios a tu base de datos (porque tu base de datos no tiene conexión). Esto puede no parecer gran cosa, pero te sorprendería la diferencia que puede suponer.

Git tiene integridad

Todo en Git es verificado mediante una suma de comprobación (checksum en inglés) antes de ser almacenado, y es identificado a partir de ese momento mediante dicha suma. Esto significa que es imposible cambiar los contenidos de cualquier archivo o directorio sin que Git lo sepa. Esta funcionalidad está integrada en Git al más bajo nivel y es parte integral de su filosofía. No puedes perder información durante su transmisión o sufrir corrupción de archivos sin que Git sea capaz de detectarlo.

El mecanismo que usa Git para generar esta suma de comprobación se conoce como hash SHA-1. Se trata de una cadena de 40 caracteres

hexadecimales (0-9 y a-f), y se calcula con base en los contenidos del archivo o estructura del directorio en Git. Un hash SHA-1 se ve de la siguiente forma:

```
24b9da6552252987aa493b52f8696cd6d3b00373
```

Verás estos valores hash por todos lados en Git, porque son usados con mucha frecuencia. De hecho, Git guarda todo no por nombre de archivo, sino por el valor hash de sus contenidos.

Git generalmente solo añade información

Cuando realizas acciones en Git, casi todas ellas sólo añaden información a la base de datos de Git. Es muy difícil conseguir que el sistema haga algo que no se pueda enmendar, o que de algún modo borre información. Como en cualquier VCS, puedes perder o estropear cambios que no has confirmado todavía. Pero después de confirmar una copia instantánea en Git es muy difícil perderla, especialmente si envías tu base de datos a otro repositorio con regularidad.

Esto hace que usar Git sea un placer, porque sabemos que podemos experimentar sin peligro de estropear gravemente las cosas.

Ilustración 11 Los Tres Estados



Nota Ilustración Material de Formación SENA

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF29/index.html#/curso/tema2>

Git tiene tres estados principales en los que se pueden encontrar tus archivos:

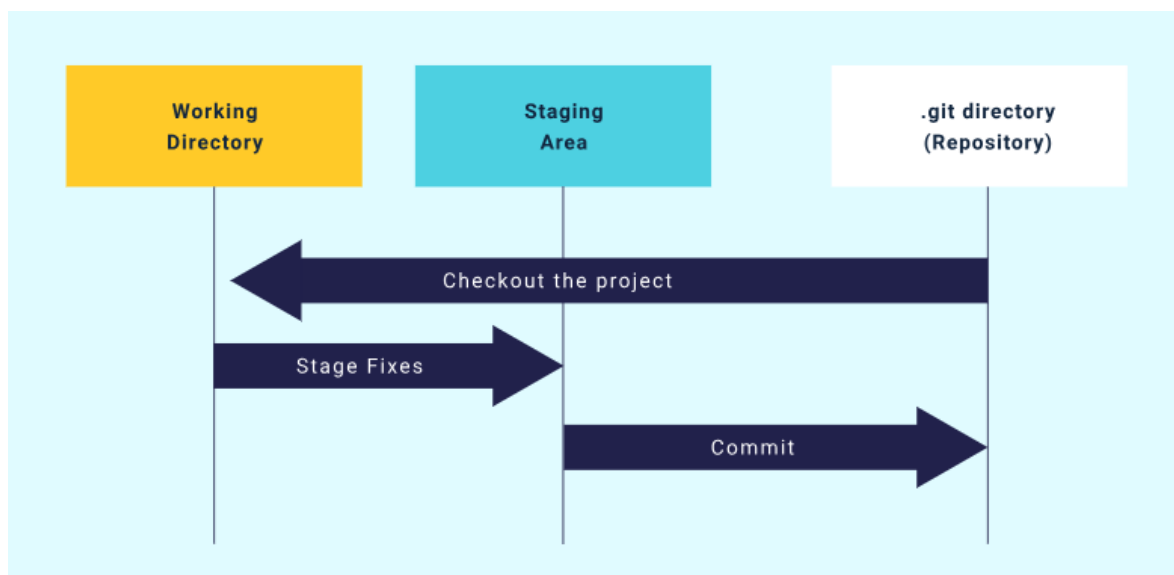
Confirmado (committed): Significa que los datos están almacenados de manera segura en tu base de datos local

Modificado (modified): Significa que has modificado el archivo, pero todavía no lo has confirmado a tu base de datos

Preparado (staged): Significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

Esto nos lleva a las tres secciones principales de un proyecto de Git: El directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area).

Ilustración 12 Secciones Principales Git



Nota Ilustración Material de Formación SENA

<https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF29/index.html#/curso/tema2>

El directorio de Git es donde se almacenan los metadatos y la base de datos de objetos para tu proyecto. Es la parte más importante de Git, y es lo que se copia cuando clonas un repositorio desde otra computadora.

El directorio de trabajo es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en disco para que los puedas usar o modificar.

El área de preparación es un archivo, generalmente contenido en tu directorio de Git, que almacena información acerca de lo que va a ir en tu próxima confirmación. A veces se le denomina índice (“index”), pero se está convirtiendo en estándar el referirse a ella como el área de preparación.

El flujo de trabajo básico en Git es algo así:

- Modificas una serie de archivos en tu directorio de trabajo.
- Preparas los archivos, añadiéndolos a tu área de preparación.
- Confirmas los cambios, lo que toma los archivos tal y como están en el área de preparación y almacena esa copia instantánea de manera permanente en tu directorio de Git.

Si una versión concreta de un archivo está en el directorio de Git, se considera confirmada (committed). Si ha sufrido cambios desde que se obtuvo del repositorio, pero ha sido añadida al área de preparación, está preparada (staged). Y si ha sufrido cambios desde que se obtuvo del repositorio, pero no se ha preparado, está modificada (modified).

9.3. Comandos básicos de Git

Para trabajar con Git es importante conocer los diferentes comandos que posibilitan la creación de repositorios, a continuación, se podrá acceder a ellos a través del siguiente recurso:

- Para inicializar un repositorio desde un directorio existente se debe usar el comando:

```
$ git init
```

- Este comando creará un nuevo subdirectorio llamado .git, el cual contendrá todos los archivos necesarios para el repositorio. Si se

desea obtener una copia desde un repositorio existente se debe utilizar el comando de clonación.

```
$ git clone https://url_del_repositorio
```

- Este comando permite recibir una copia de casi todos los archivos existentes en el repositorio de origen, incluyendo cada versión de cada uno de los archivos. Así, si el servidor original llegara a dañarse se podrá realizar una restauración a partir de todos los clones generados.

Una vez se dispone de un espacio de trabajo inicializado, se debe tener en cuenta que todos los archivos van a tener dos posibles estados, los cuales son:

Rastreados (tracked files): son todos aquellos archivos que estaban en la última versión del proyecto independiente de si estos archivos están modificados, sin modificar o preparados. Cuando se clona un proyecto a partir de un repositorio existente, todos estos archivos pertenecen al estado rastreado y sin modificar.

Una vez un archivo es editado pasa al estado modificado, luego se deben pasar al estado de preparación para finalmente confirmarlos.

Sin rastrear: corresponden a todos aquellos archivos que no formaban parte de la última versión del proyecto y no están en el área de preparación (staging area).

- Para visualizar el estado actual de tus archivos indicando si están o no rastreados por Git se puede ejecutar el siguiente comando:

\$ git status

- Para todos los archivos nuevos que se desean ser rastreados por Git, se debe indicar con el siguiente comando:

\$ git add Nombre_archivo

- También es posible indicar a Git que haga rastreo de un directorio, lo cual implica que recursivamente se hace rastreo de todos los archivos en el interior del directorio.

\$ git add Directorio

- El comando git add además de servir para iniciar el rastreo de un archivo o directorio que no estaba en la última versión, también sirve para preparar archivos (enviarlos al staging area) y para marcar archivos como resueltos en los casos donde estos entran en conflictos por operaciones de combinación. Dicho de una manera más simplificada: el comando add indica que un archivo o directorio debe ser añadido en la próxima confirmación. Para confirmar cambios se ejecuta el comando:

\$ git commit

- Al ejecutar la confirmación el sistema desplegará un mecanismo para recibir un mensaje de confirmación que será asociado a esta operación de commit. También es posible agregar el comentario explícitamente en la ejecución de la confirmación usando la opción -m.

\$ git commit -m "En esta version se arreglo el archive W"

- También es posible ejecutar una operación de confirmación que salte el paso de preparación que se logra con la ejecución del comando add. Es decir, la operación de confirmación se encarga de preparar todos los archivos rastreados y luego confirmar. Esto es posible agregando la opción -a.

\$ git commit -a -m 'comentario de esta confirmación'

- Para visualizar el histórico de las confirmaciones desde la más reciente hasta la más antigua realizadas sobre un repositorio se ejecuta el comando:

\$ git log

10. Git en Entornos Remotos

Para poder colaborar con un equipo de trabajo e implementar el concepto de integración continua, es necesario hospedar los proyectos en repositorios remotos los cuales son accesibles por medio de internet o por un entorno de red.

En estos repositorios remotos se definen una serie de permisos que indican si se tiene la posibilidad de leer o de lectura y escritura. El colaborar implica la gestión de estos repositorios enviando y obteniendo información desde estos, teniendo en cuenta los estados que ya se han mencionado anteriormente.

Pasos a seguir para los entornos remotos:

- Para ver los repositorios remotos configurados se puede ejecutar el comando:

```
$ git remote
```

- Para definir un repositorio remoto y asociarlo a un nombre para su referenciación se utiliza el siguiente comando:

```
git remote add [nombre-remoto] [url]
```

- Donde nombre-remoto corresponde al nombre con que se referencia el repositorio y URL es la ubicación lógica del mismo en un entorno de red o en una dirección de internet. En el siguiente ejemplo se define un repositorio ubicado en un servidor de Github y cuyo nombre de referencia es ref.

```
$ git remote add ref https://github.com/paulboone/ticgit
```

- Una vez definido el remote se pueden extraer los datos utilizando el siguiente comando:

```
$ git fetch [nombre-remoto]
```

- El anterior comando se conecta al repositorio remoto y trae todos los datos de los cuales aún no se tiene copia en tu repositorio local. Para enviar información desde el repositorio local hacia el servidor remoto, se utiliza el siguiente comando:


```
git push [nombre-remoto] [nombre-rama]
```

- Recordemos que, si se ha clonado un repositorio desde alguna ubicación, Git asigna el nombre origin al servidor del que se ha realizado la clonación. Así, si por ejemplo queremos enviar nuestra rama master al servidor origin se debe ejecutar el comando de la siguiente forma:

```
$ git push origin master
```

11. Plataformas para Implementar Integración Continua - Gitlab

En la actualidad existe gran cantidad de plataformas con servicios en la nube en los que se pueden implementar procesos de integración continua, los cuales, generalmente, ofrecen la posibilidad de crear repositorios públicos y privados, se integran con sistemas de control de versiones, permiten el registro de grupos de trabajo y otras posibilidades más. Entre los más utilizados en la industria que se basan en sistemas de control de versiones en Git, se encuentran los descritos a continuación:

GitHub: Conoce GitHub un portal creado para alojar códigos de las aplicaciones.

GitLab: Conoce sobre GitLab, un servicio web que permite controlar las versiones y desarrollos de software colaborativo basado en Git.

BitBucket: Conoce a Bitbucket un servicio de alojamiento basado en web para los proyectos que utilizan sistema de control de versiones Mercurial y Git.

Conclusiones

La codificación modular es una forma eficaz de convertir las ideas y requerimientos del proyecto en un software útil y funcional. Al dividir el sistema en partes bien definidas, se logra un mayor orden, se reducen los errores y se facilita la mejora continua. En el caso de esta plataforma, cada módulo cumple un papel importante: motivar a las personas a ejercitarse, acompañarlas en su alimentación y brindarles un espacio para compartir con otros. De esta manera, no solo se construye un programa tecnológico, sino una herramienta que busca impactar de manera positiva en la vida diaria de los usuarios, ayudándoles a mantener un equilibrio entre cuerpo y mente.

Bibliografía

Conservancy, G. e. (s.f.). *Git fast version control*. Obtenido de <https://git-scm.com/book/es/v2>

SENA. (s.f.). *SENA*. Obtenido de <https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF29/index.html#/curso/tema1>

SENA. (s.f.). *SENA*. Obtenido de <https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF30/index.html#/>