**Instituto Nacional De Educación Nivel Diversificada INED Santa Cruz Naranjo Santa Rosa**

**Integrantes:** Luis Mario López Paniagua ID: 1701

 Estefany Fabiola Donis Herrera ID: 1694

 Marlen Aranza Estrada Suazo ID: 1721

 Fernanda Analy Gonzales Gonzales ID: 1704

 Reinita del Carmen Aguilar Yuman ID: 1715

 Deilin Adileidi Mejia Ramirez ID: 1713

**Tema:** Proyecto Fase1

**Catedratico:** Gustavo Blanco

**Catedra:** producción

**Fecha de Entrega:** 30/08/2025

Índice

Contenido

[Prototipo 3](#_Toc207443854)

[Identificar requisitos: 3](#_Toc207443855)

[ Recopilar retroalimentación 3](#_Toc207443856)

[ Mitigar riesgos: 3](#_Toc207443857)

[¿Cómo funciona? 3](#_Toc207443858)

[**1. Creación del prototipo:** 3](#_Toc207443859)

[ Prototipos de baja fidelidad: 4](#_Toc207443860)

# Prototipo

La investigación de software en prototipos se refiere a la creación de versiones preliminares o modelos funcionales del software para probar ideas, recopilar comentarios y refinar el diseño en las primeras etapas del desarrollo, permitiendo identificar requisitos, mitigar riesgos y asegurar que el producto final cumpla con las expectativas del cliente antes de un desarrollo completo y costoso. Se utilizan prototipos de diferentes niveles de fidelidad, desde bocetos en papel hasta versiones interactivas, que son probadas por las partes interesadas para obtener retroalimentación valiosa.

¿Qué es un prototipo de software?

Un prototipo es una versión simplificada y a menudo incompleta del software que demuestra su funcionalidad básica y apariencia. Su propósito es funcionar como un "borrador" para explorar conceptos, probar la viabilidad y obtener una visión clara del producto final.

¿Para qué se utiliza?

## Identificar requisitos:

Ayuda a aclarar y validar los requisitos de los usuarios y del proyecto en general.

 Recopilar retroalimentación**:**

Permite que los usuarios y las partes interesadas interactúen con el software y proporcionen comentarios cruciales.

 Refinar diseños:

Facilita la mejora iterativa del diseño de la interfaz y la experiencia del usuario.

##  Mitigar riesgos:

Detecta posibles problemas y desafíos de usabilidad antes de que se conviertan en problemas serios y costosos en etapas posteriores del desarrollo.

 Evaluar conceptos:

Es una herramienta efectiva para probar nuevas ideas y la viabilidad de un proyecto.

## ¿Cómo funciona?

## **1. Creación del prototipo:**

Se crea una versión simplificada del software, que puede variar desde bocetos en papel hasta prototipos digitales interactivos.

 **2**. Interacción y evaluación**:**

Las partes interesadas (usuarios, clientes, etc.) interactúan con el prototipo para probar su funcionalidad y diseño.

 **3.** Recopilación de comentarios**:**

Se recoge la opinión de los usuarios sobre la usabilidad, la navegación y las características del prototipo.

 **4.** Iteración y mejora:

El prototipo se refina y actualiza continuamente en función de la retroalimentación recibida.

 **5.** Ciclos repetidos**:**

Este proceso iterativo continúa hasta que el prototipo cumple con los requisitos del proyecto.

Tipos de prototipos:

* Prototipos de baja fidelidad: Simples y rápidos de crear, como bocetos en papel, que ayudan a explorar ideas sin mucho detalle.

 Prototipos de alta fidelidad**:** Se asemejan mucho al producto final en términos de apariencia y funcionalidad, siendo útiles para probar la usabilidad en detalle.

 Prototipos horizontales**:** Cubren una amplia gama de funcionalidades de manera superficial.

 Prototipos verticales: Profundizan en un conjunto específico y limitado de funcionalidades.

Diseños de base de datos



Diseño web



Diseño



DISEÑO

1. Uso de sesiones:

Se utilizan para que el sistema recuerde qué usuario ha iniciado sesión, incluso cuando cambia de página. Esto evita que el usuario tenga que escribir su contraseña en cada pantalla.

2. Conexión con la base de datos:

Se conecta con MySQL para comprobar si los datos que el usuario escribió existen realmente en la tabla de usuarios.

3. Consultas preparadas:

Se decidió usar consultas seguras en lugar de concatenar texto directamente. Esto protege contra ataques de inyección SQL, que son intentos de entrar manipulando los formularios.

4. Validación de usuario y contraseña:

El sistema comprueba si el usuario existe y si la contraseña coincide. Si es correcto, lo deja entrar; si no, muestra un error.

5. Redirección al panel:

Si los datos son válidos, el usuario es llevado directamente a la página del panel. Esto asegura un flujo natural: primero inicio de sesión, luego acceso al panel.

---

📋 En el panel de control

1. Protección con sesión:

Antes de mostrar el panel, se revisa si el usuario está logueado. Si no lo está, lo devuelve al login. Esto evita que alguien escriba la dirección del panel y entre sin permiso.

2. Diseño del menú:

Se colocan tres botones claros (productos, empleados y alumnos) para que el usuario entienda rápido qué opciones de gestión tiene.

3. Uso de un estilo sencillo:

Se aplicó un diseño básico directamente en el mismo archivo para que sea fácil de probar. En un sistema real, lo ideal es separar los estilos en archivos CSS.

4. Función de cerrar sesión (logout):

Se añadió la opción de salir del sistema de forma segura. Esto borra la sesión activa y evita que alguien más pueda usar el mismo navegador y entrar sin autenticarse.

---

📑 En las páginas de gestión (productos, empleados, alumnos)

1. Protección con sesión:

Igual que en el panel, estas páginas revisan que el usuario esté logueado antes de mostrar el contenido.

2. Estructura inicial:

Por ahora son páginas simples que solo muestran información básica y un enlace para volver al panel. Esto se dejó así para que luego puedan crecer a un sistema completo de agregar, editar y eliminar registros.