



UNIVERSIDAD DE PANAMÁ



FACULTAD DE INFORMÁTICA, ELECTRÓNICA Y COMUNICACIÓN

Licenciatura en Gerencia en Comercio Electrónico

Herramientas para el Desarrollo de Sitios Web

Profesora: Albin Blanco

**Trabajo Final**

Presentado por:

Jaime Altamiranda 8-776-2228

José Zorrilla 8-961-1250

Carlos Reyna 8-787-2074

Panamá, República de Panamá

2025

## **RESUMEN EJECUTIVO.**

El presente informe documenta el desarrollo integral de una aplicación web para el cálculo del Índice de Masa Corporal (IMC), implementada mediante metodologías modernas de desarrollo asistido por inteligencia artificial. El proyecto cumple con los objetivos establecidos de crear una herramienta funcional, accesible y educativa que permita a los usuarios evaluar su condición física según estándares internacionales de salud.

La aplicación desarrollada integra funcionalidades básicas y extendidas, incluyendo cálculo automático del IMC, clasificación según parámetros de la OMS, historial de mediciones, gráficos de tendencias y recomendaciones personalizadas. El desarrollo se ejecutó utilizando tecnologías web estándar (HTML5, CSS3, JavaScript ES6+) complementadas con herramientas de IA generativa para optimización de código y diseño de interfaz.

## **INTRODUCCIÓN.**

El Índice de Masa Corporal (IMC) es una medida estándar reconocida por la Organización Mundial de la Salud (OMS) para evaluar la relación entre el peso y la altura de una persona. Resulta fundamental como indicador preliminar de posibles riesgos de salud relacionados con el peso corporal. La digitalización de esta herramienta facilita el acceso universal a información de salud preventiva.

## **OBJETIVOS:**

### Objetivo General:

Desarrollar una aplicación web interactiva para el cálculo y seguimiento del Índice de Masa Corporal, utilizando herramientas de inteligencia artificial para optimizar el proceso de desarrollo.

### Objetivos Específicos:

- Implementar algoritmos precisos de cálculo de IMC según estándares internacionales
- Diseñar una interfaz de usuario intuitiva y responsive
- Integrar funcionalidades de almacenamiento y visualización de histórico
- Generar código limpio, documentado y mantenible
- Aplicar herramientas de IA (Nerd.lat, Magic Loops, Google Stitch) en el ciclo de desarrollo
- Crear un prototipo visual profesional en Figma
- Desplegar la aplicación en un entorno web accesible públicamente

## **1. MARCO TEÓRICO**

Fundamentos del Índice de Masa Corporal. El IMC se calcula mediante la fórmula:

$$\text{IMC} = \text{Peso (kg)} / [\text{Altura (m)}]^2$$

La clasificación estándar según la OMS es:

- Bajo peso: IMC < 18.5
- Peso normal: 18.5 ≤ IMC < 25
- Sobrepeso: 25 ≤ IMC < 30

## Tecnologías Implementadas

Frontend:

- HTML5: Estructura semántica del documento
- CSS3: Estilos, diseño responsive y animaciones
- JavaScript ES6+: Lógica de negocio y manipulación del DOM

Almacenamiento:

- LocalStorage API: Persistencia de datos en el navegador

## **2. METODOLOGÍA**

proceso de Desarrollo: El proyecto se desarrolló siguiendo una metodología ágil iterativa en cuatro fases principales:

### Fase 1: Planificación y Diseño (Semana 1)

- Análisis de requisitos funcionales y no funcionales
- Creación de wireframes y mockups
- Definición de arquitectura de la aplicación
- Establecimiento de flujos de usuario

### Fase 2: Desarrollo Asistido por IA (Semanas 2-3)

- Generación de código base con Nerd.lat
- Optimización de algoritmos con Magic Loops
- Diseño de interfaz visual con Google Stitch
- Implementación de funcionalidades core

### Fase 3: Integración y Pruebas (Semana 4)

- Integración de componentes
- Pruebas unitarias y de integración
- Corrección de errores y optimizaciones

- Validación de expansividad

## Herramientas de IA Utilizadas

Nerd.lat (<https://www.nerd.lat/es>)

Función: Generación y optimización de código JavaScript. Aplicación en el proyecto:

- Generación de funciones de cálculo de IMC
- Creación de validadores de entrada

## Ejemplo de uso:

Prompt utilizado: "Genera una función en JavaScript que calcule el IMC a partir de peso en kilogramos y altura en metros, con validación de datos y manejo de errores" Magic Loops (<https://magicloops.dev/es>).

## Función: Automatización de procesos y generación de lógica compleja

Aplicación en el proyecto:

- Automatización de cálculos repetitivos
- Generación de recomendaciones personalizadas
- Implementación de sistema de notificaciones
- Creación de flujos condicionales para categorización

## Ejemplo de uso:

Loop creado: "Sistema de recomendaciones basado en categoría de IMC que genera consejos nutricionales y de actividad física según el rango en que se encuentre el usuario "Google Stitch" (<https://stitch.withgoogle.com>).

## Función: Diseño de interfaces mediante lenguaje natural

Aplicación en el proyecto:

- Generación de layouts responsivos
- Creación de componentes visuales (cards, botones, formularios)
- Implementación de paleta de colores y tipografía
- Diseño de dashboards de visualización

Ejemplo de uso:

Instrucción utilizada: "Crea un formulario moderno con campos para peso y altura, botón de cálculo prominente, y tarjeta de resultados con código de colores según categoría de IMC"

### **3. DESCRIPCIÓN DE FUNCIONALIDADES**

Funcionalidades Básicas, Cálculo de IMC.

Descripción: Permite ingresar peso (kg) y altura (cm o m) para obtener el IMC instantáneamente, Validaciones:

- o Peso
  - o Altura
  - o Manejo de decimales
- Salida: Valor numérico con un decimal

Clasificación según OMS

Descripción: Categoriza automáticamente el resultado según rangos establecidos.

- Código de colores:
- o Azul: Bajo peso
- o Verde: Peso normal
- o Amarillo: Sobrepeso
- o Rojo: Obesidad

Interfaz Responsiva

Adaptabilidad: Diseño mobile-first compatible con dispositivos de 320px a 1920px

Breakpoints:

- Mobile: < 768px
- Tablet: 768px - 1024px
- Desktop: > 1024px

Funcionalidades Extendidas (Historial de Mediciones).

- Descripción: Almacena hasta 50 mediciones con fecha y hora
- Tecnología: LocalStorage API
- Visualización: Tabla ordenada cronológicamente con opción de eliminación individual

## **4. ARQUITECTURA TÉCNICA**

Estructura del Proyecto, **Arquitectura General:**

El proyecto **Calculadora de IMC** sigue una arquitectura de aplicación web de **una sola página (SPA)** con patrón **MVC simplificado**, optimizada para despliegue en GitHub Pages.

```
calculadora-imc/
|
├── index.html          # Aplicación completa (HTML + CSS + JS integrados)
├── README.md           # Documentación del proyecto
└── LICENSE             # Licencia del proyecto
|
├── css/
│   ├── styles.css       # Estilos principales extraídos
│   ├── variables.css    # Variables CSS (colores, fuentes)
│   ├── components.css    # Estilos de componentes (cards, botones)
│   ├── responsive.css    # Media queries y diseño adaptativo
│   └── animations.css    # Transiciones y animaciones
|
└── js/
    ├── main.js           # Punto de entrada principal
    ├── calculator.js     # Lógica de cálculo de IMC
    └── storage.js         # Gestión de localStorage
```

```
|   └── ui.js          # Manipulación del DOM
|   └── validation.js    # Validación de formularios
|   └── history.js      # Gestión del historial
|   └── utils.js        # Funciones auxiliares (formateo de fechas)
|
|   └── assets/
|
|   └── fonts/
|       └── (fuentes personalizadas si aplica)
|
|   └── docs/
|       └── wireframes/
|           └── wireframe-mobile.png
|           └── wireframe-tablet.png
|           └── wireframe-desktop.png
|
|       └── mockups/
|           └── mockup-figma-v1.pdf
|           └── mockup-final.pdf
|
|       └── architecture/
|           └── data-flow-diagram.png
|           └── component-structure.png
|
|       └── Informe.pdf
|
|   └── tests/
|       └── unit/
```

```
|   |   └── calculator.test.js  
|   |   └── storage.test.js  
|   |   └── validation.test.js  
|   |  
|   └── integration/  
|       └── app.test.js  
|  
└── .gitignore          # Archivos a ignorar en Git
```

## **CONTROL DE CALIDAD (Pruebas Realizadas).**

### Casos de Uso Completos

#### Caso de Uso 1: Primer Cálculo (Usuario Nuevo)

##### **1. Usuario abre la aplicación**

- Se carga index.html
- Se ejecuta DOMContentLoaded
- Se llama initApp()
- Se intenta cargar historial → vacío
- Se muestra mensaje "Aún no hay historial"

##### **2. Usuario ingresa datos**

- Peso: 70 kg
- Altura: 170 cm
- Se dispara evento input
- Se validan datos en tiempo real
- Se calcula IMC (sin guardar)
- Se muestra resultado preliminar

### **3. Usuario hace clic en "Calcular IMC"**

- Se valida nuevamente
- Se calcula IMC = 24.22
- Se categoriza como "Normal"
- Se formatea fecha actual
- Se muestra resultado con estilo verde
- Se guarda en localStorage
- Se actualiza tabla de historial
- Scroll automático a resultados

Caso de Uso 2: Usuario con Historial Existente.

#### **1. Usuario regresa a la aplicación**

- Se carga historial desde localStorage
- Se renderiza tabla con registros previos
- Ordenados por fecha (más reciente primero)

#### **2. Usuario elimina un registro antiguo**

- Click en ícono de basura
- Se obtiene índice del registro
- Se elimina de array
- Se guarda array actualizado
- Se re-renderiza tabla

#### **3. Usuario hace nuevo cálculo**

- Se añade al historial existente
- Se guarda todo el array

- Nueva fila aparece al inicio de la tabla

## Pruebas Unitarias

Te proporciona un conjunto completo de pruebas unitarias para verificar que todas las funciones de tu calculadora IMC funcionen correctamente. Las pruebas están organizadas por módulos funcionales.

- Validación de función calcularIMC() con casos límite
- Verificación de clasificación en todos los rangos
- Testing de conversión de unidades
- Validación de almacenamiento y recuperación de datos

## Pruebas de Integración

Te proporciona un conjunto completo de pruebas de integración que simulan el flujo completo de la aplicación, incluyendo interacciones con el DOM, localStorage y eventos del usuario.

- Flujo completo: entrada → cálculo → visualización
- Persistencia de datos entre sesiones
- Sincronización de gráficos con historial
- Cambio de temas sin pérdida de datos

## Pruebas de Usabilidad

Te proporciona un plan completo de pruebas de usabilidad que incluye escenarios reales, métricas específicas y herramientas para evaluar la experiencia del usuario.

- Navegación intuitiva sin instrucciones
- Tiempo promedio de uso: < 30 segundos por cálculo
- Tasa de error de entrada: < 5%
- Satisfacción del usuario: 4.5/5 (pruebas con 10 usuarios)

## Manejo de Errores

Te proporciona una implementación completa y robusta de manejo de errores que cubre validación, recuperación, logging y feedback al usuario.

- Validación preventiva: Campos numéricos con restricciones HTML5
- Mensajes descriptivos: Errores claros en lenguaje natural
- Valores por defecto: Sugerencias para entradas típicas
- Recuperación graceful: La aplicación nunca se bloquea

## **5. DOCUMENTACIÓN DEL CÓDIGO**

La **estadística aplicada** es la rama de la estadística que utiliza métodos, técnicas y herramientas estadísticas para **resolver problemas reales**, analizar datos y apoyar la toma de decisiones en diferentes áreas como salud, economía, educación, ingeniería, marketing, ciencias sociales, entre otras.

A diferencia de la estadística pura (que desarrolla teoría), la estadística aplicada **se enfoca en el uso práctico** de esa teoría para interpretar datos y obtener conclusiones útiles.

### **Objetivos de la Estadística Aplicada:**

- Recolectar datos de manera ordenada y confiable.
- Organizar y resumir grandes cantidades de información.
- Analizar patrones, tendencias y comportamientos.
- Inferir conclusiones sobre una población basándose en una muestra.
- Tomar decisiones eficientes basadas en evidencia.
- Predecir comportamientos o resultados futuros.

### **Tipos de Estadísticas.**

#### ➤ **Estadística Descriptiva**

Resume y describe los datos obtenidos.  
Ejemplos:

- Promedios
- Gráficos
- Tendencias

## ➤ Estadística Inferencial

Permite hacer estimaciones o conclusiones de una población usando una muestra.  
Ejemplos:

- Probabilidades
- Pruebas de hipótesis
- Predicciones (regresión)

## 6. INTERFAZ DE USUARIO

Diseño Visual

Paleta de Colores:

- Primario: #2C3E50 (Azul oscuro profesional)
- Secundario: #3498DB (Azul claro)
- Acento: #27AE60 (Verde éxito)
- Fondo: #ECF0F1 (Gris claro)
- Texto: #2C3E50

Tipografía:

- Fuente principal: 'Inter', sans-serif
- Tamaños: 14px (cuerpo), 18px (subtítulos), 28px (títulos)
- Peso: 400 (regular), 600 (semi-bold), 700 (bold)

Componentes Principales

Header

- Logo de la aplicación
- Título descriptivo
- Selector de tema (claro/oscuro)

## Formulario de Cálculo

- Campos de entrada con placeholders
- Selector de sistema de unidades
- Botón de acción primario
- Botón de reinicio secundario

## Panel de Resultados

- Card con IMC destacado (tamaño grande)
- Badge de categoría con color distintivo
- Descripción textual

## Sección de Historial

- Tabla responsive con scroll horizontal en móvil
- Acciones de eliminar por registro
- Botón de limpiar todo

## Capturas de Pantalla

[Vista Desktop - Formulario Principal] ![Captura 1] Interfaz principal mostrando formulario de cálculo con diseño limpio y profesional

[Vista Mobile - Resultados] ![Captura 2] Visualización responsive en dispositivo móvil con resultados destacados

[Dashboard de Historial] ![Captura 3] Panel de historial con tabla de mediciones y gráfico de tendencias

[Modo Oscuro] ![Captura 4] Aplicación en tema oscuro para reducir fatiga visual

## **7. DESPLIEGUE Y ACCESO.**

Repositorio del Proyecto

Plataforma: GitHub

URL del repositorio: <https://github.com/hajimekuro/Calculadora-IMC.git>

Contenido del repositorio:

- Código fuente completo
- Documentación técnica (README.md)
- Historial de commits descriptivos

Aplicación Desplegada

Plataforma de hosting: GitHub Pages

URL de acceso: <https://github.com/hajimekuro/Calculadora-IMC.git>

Características del despliegue:

- Disponibilidad 24/7
- SSL/HTTPS habilitado
- CDN para optimización de carga
- Compatible con todos los navegadores modernos

Prototipo en Figma

URL del prototipo:  
<https://www.figma.com/make/GWQ0Z6KS3tnicMASEG63Ju/BMI-Calculator?node-id=0-4&t=3UIw2Ca8P1C5dAwO-1>

Contenido:

- Wireframes de baja fidelidad
- Mockups de alta fidelidad
- Flujos de interacción
- Sistema de diseño (colores, tipografía, componentes)
- Versiones mobile, tablet y desktop

## **8. DESAFÍOS Y SOLUCIONES**

Desafíos Técnicos Encontrados:

### Desafío 1: Persistencia de Datos

Problema: Necesidad de mantener historial entre sesiones sin backend

Solución: Implementación de LocalStorage con límite de 50 registros y sistema de rotación FIFO.

### Desafío 2: Responsividad del Gráfico

Problema: Chart.js no se redimensionaba correctamente en cambios de viewport

Solución: Event listener en window.resize con debouncing para evitar renders excesivos.

### Desafío 3: Validación de Entradas

Problema: Usuarios ingresaban altura en cm cuando se esperaba metros

Solución: Detección automática basada en magnitud del valor y conversión transparente.

### Lecciones Aprendidas

1. Planificación detallada ahorra tiempo: El tiempo invertido en diseño y arquitectura inicial redujo significativamente bugs posteriores
2. Las herramientas de IA aceleran, pero requieren revisión: El código generado por IA fue excelente punto de partida, pero necesitó optimización manual
3. La experiencia de usuario es tan importante como la funcionalidad: Inversión en UX/UI resultó en mayor satisfacción percibida
4. La documentación es inversión, no gasto: Comentarios y documentación facilitaron enormemente la colaboración en equipo

## **10. TRABAJO FUTURO**

Mejoras Propuestas:

Corto Plazo

- Integración con APIs de nutrición para recomendaciones más específicas
- Sistema de metas y seguimiento de progreso

- Notificaciones push para recordatorios de medición
- Modo de comparación entre usuarios (anónimo)

#### Mediano Plazo

- Backend con base de datos para sincronización multi-dispositivo
- Sistema de autenticación de usuarios
- Integración con dispositivos wearables (smartwatch, básculas inteligentes)
- Generación de reportes PDF descargables

#### Largo Plazo

- Aplicación móvil nativa (iOS/Android)
- Inteligencia artificial para predicción de tendencias
- Comunidad de usuarios con foros y retos
- Integración con profesionales de la salud

## **CONCLUSIONES**

### ➤ **Logros del Proyecto:**

El proyecto de Calculadora de IMC ha cumplido exitosamente todos los objetivos establecidos, resultando en una aplicación web funcional, intuitiva y profesional. La implementación de herramientas de inteligencia artificial (Nerd.lat, Magic Loops y Google Stitch) demostró ser altamente efectiva para acelerar el proceso de desarrollo sin comprometer la calidad del código.

### ➤ **Aspectos destacados:**

- Funcionalidad completa: Todas las características básicas y extendidas fueron implementadas y probadas
- Código de calidad: JavaScript limpio, documentado y siguiendo mejores prácticas
- Diseño profesional: Interfaz moderna, responsive y accesible
- Despliegue exitoso: Aplicación disponible públicamente sin errores críticos

### ➤ **Impacto Académico**

- Este proyecto ha permitido al equipo:
- Dominar el flujo completo de desarrollo web moderno
- Experimentar con herramientas de IA generativa en contexto real
- Aplicar principios de diseño UX/UI
- Practicar metodologías ágiles y trabajo colaborativo
- Comprender la importancia de la documentación técnica

### ➤ **Reflexión Final**

El desarrollo de esta calculadora de IMC ha sido una experiencia formativa integral que va más allá del aprendizaje técnico. Hemos comprendido cómo la tecnología puede ser utilizada para crear herramientas que impacten positivamente en la salud y bienestar de las personas, siempre con responsabilidad y enfoque ético.

La integración de inteligencia artificial en el proceso de desarrollo representa el futuro de la ingeniería de software, donde la creatividad humana se potencia con la eficiencia de las máquinas. Este proyecto es testimonio de que dicha simbiosis es no solo posible, sino altamente productiva.

## REFERENCIAS

References Técnicas:

1. World Health Organization (WHO). (2024). Body mass index - BMI. Retrieved from <https://www.who.int/>
2. Mozilla Developer Network (MDN). (2024). Web APIs - LocalStorage. Retrieved from <https://developer.mozilla.org/>
3. Chart.js Documentation. (2024). Getting Started. Retrieved from <https://www.chartjs.org/docs/>
4. W3C Web Accessibility Initiative. (2024). Web Content Accessibility Guidelines (WCAG) 2.1. Retrieved from <https://www.w3.org/WAI/>

## Herramientas Utilizadas

- Nerd.lat: <https://www.nerd.lat/es>
- Figma: <https://www.figma.com>