Especificación de Requisitos de Software (SRS)

Calculadora de Nota Final

Versión: 1.0
Fecha: 29 de octubre de 2025
Preparado por: Equipo de Desarrollo

---

1. Introducción

1.1 Propósito

Este documento describe los requisitos funcionales y no funcionales de la Calculadora de Nota Final, una aplicación web diseñada para ayudar a estudiantes universitarios a calcular la nota necesaria en el tercer corte académico para aprobar una materia con una calificación mínima de 3.0.

1.2 Alcance

La aplicación "Calculadora de Nota Final" es una herramienta web educativa que:

- Calcula automáticamente la nota requerida en el tercer corte (34%) basándose en las notas de los dos primeros cortes (33% cada uno)
- Proporciona retroalimentación inmediata sobre la viabilidad de aprobar la materia
- Valida las entradas del usuario en tiempo real
- Ofrece una interfaz intuitiva y responsiva

1.3 Definiciones, Acrónimos y Abreviaturas

- **SRS:** Software Requirements Specification (Especificación de Requisitos de Software)
- **DOM:** Document Object Model
- **UX:** User Experience (Experiencia de Usuario)
- **UI:** User Interface (Interfaz de Usuario)
- **CSS:** Cascading Style Sheets
- **HTML:** HyperText Markup Language
- **JS:** JavaScript

1.4 Referencias

- Sistema de calificación académico colombiano (escala 0.0 - 5.0)
- Estándares de accesibilidad web WCAG 2.1
- Tailwind CSS v3.x Documentation

1.5 Visión General

Este documento está organizado en las siguientes secciones: descripción general del producto, requisitos funcionales específicos, requisitos no funcionales, y consideraciones técnicas de implementación.

---

2. Descripción General

2.1 Perspectiva del Producto

La Calculadora de Nota Final es una aplicación web standalone que funciona completamente en el navegador del cliente sin requerir backend o almacenamiento de datos en servidor. Utiliza tecnologías estándar web (HTML5, CSS3, JavaScript ES6+) y el framework CSS Tailwind.

2.2 Funciones del Producto

Las funciones principales incluyen:

- Entrada de notas para dos cortes académicos
- Validación en tiempo real de las entradas
- Cálculo automático de la nota necesaria en el tercer corte
- Visualización clara de resultados con diferentes estados (éxito, advertencia, error)
- Manejo de casos especiales (ya aprobado, imposible aprobar)

2.3 Características del Usuario

Usuario objetivo: Estudiantes universitarios en Colombia que:

- Tienen conocimiento básico de navegación web
- Están cursando materias con sistema de tres cortes
- Necesitan planificar su esfuerzo académico
- Utilizan dispositivos móviles o de escritorio

2.4 Restricciones

- La aplicación debe funcionar sin conexión a internet después de la carga inicial

- Debe ser compatible con navegadores modernos (Chrome, Firefox, Safari, Edge)
- El sistema de calificación está fijo a la escala colombiana (0.0 - 5.0)
- Los porcentajes de los cortes están predefinidos (33%, 33%, 34%)

2.5 Suposiciones y Dependencias

- El usuario tiene acceso a un navegador web moderno con JavaScript habilitado
- El usuario conoce sus notas de los primeros dos cortes
- La conexión a internet está disponible para la carga inicial de Tailwind CSS CDN
- El sistema de calificación utiliza 3.0 como nota mínima aprobatoria

---

3. Requisitos Funcionales

RF-001: Entrada de Notas

Prioridad: Alta
Descripción: El sistema debe permitir al usuario ingresar las notas de los dos primeros cortes académicos.

Criterios de aceptación:

- Debe haber dos campos de entrada claramente identificados
- Cada campo debe mostrar su porcentaje correspondiente (33%)
- Los campos deben aceptar números con punto o coma como separador decimal
- Los placeholders deben mostrar ejemplos de formato válido

RF-002: Validación en Tiempo Real

Prioridad: Alta
Descripción: El sistema debe validar las entradas del usuario mientras escribe, sin necesidad de enviar el formulario.

Criterios de aceptación:

- Los valores negativos deben convertirse automáticamente a 0
- Los valores mayores a 5.0 deben limitarse a 5.0
- Los caracteres no numéricos (excepto punto y coma) deben ser rechazados
- El guion solitario (-) debe permitirse temporalmente durante la escritura

- La validación debe ser no intrusiva y no interrumpir la experiencia de escritura

RF-003: Validación al Calcular

Prioridad: Alta

Descripción: El sistema debe validar completamente las entradas antes de realizar el cálculo.

Criterios de aceptación:

- Debe verificar que ambos campos contengan valores
- Debe rechazar el guion solitario (-) como valor final
- Debe validar que los valores sean números válidos (no NaN)
- Debe verificar que los valores estén en el rango 0.0 - 5.0
- Debe mostrar mensajes de error específicos para cada tipo de validación fallida

RF-004: Cálculo de Nota Necesaria

Prioridad: Alta

Descripción: El sistema debe calcular la nota necesaria en el tercer corte utilizando la fórmula: $Nota3 = (3.0 - (Corte1 \times 0.33) - (Corte2 \times 0.33)) / 0.34$

Criterios de aceptación:

- El cálculo debe realizarse al presionar el botón "Calcular Nota Necesaria"
- El resultado debe mostrarse con dos decimales de precisión
- El cálculo debe ser matemáticamente preciso
- Debe manejar todos los casos posibles (aprobado, por aprobar, imposible)

RF-005: Visualización de Resultados

Prioridad: Alta

Descripción: El sistema debe mostrar el resultado del cálculo con formato apropiado según el escenario.

Criterios de aceptación:

- **Caso 1 - Ya aprobado (nota ≤ 0):** Mostrar mensaje de felicitación con estilo verde
- **Caso 2 - Posible aprobar (0 < nota ≤ 5):** Mostrar nota necesaria con estilo azul informativo
- **Caso 3 - Imposible aprobar (nota > 5):** Mostrar mensaje de imposibilidad con estilo rojo
- Todos los mensajes deben incluir emojis apropiados para mejorar la experiencia

- La nota calculada debe mostrarse en negrita y destacada

RF-006: Manejo de Errores

Prioridad: Alta
Descripción: El sistema debe proporcionar retroalimentación clara cuando ocurran errores de entrada.

Criterios de aceptación:

- Error por campos vacíos: "Por favor, ingresa las notas de los dos primeros cortes."
- Error por guion solitario: "El guion solo no es un número válido..."
- Error por formato inválido: "Formato de número no válido..."
- Error por rango inválido: "Las notas deben estar entre 0 y 5."
- Todos los errores deben mostrarse en el área de resultado con estilo rojo

RF-007: Limpieza y Reset

Prioridad: Media
Descripción: El sistema debe permitir resetear las clases CSS del contenedor de resultado antes de mostrar nuevos mensajes.

Criterios de aceptación:

- Al calcular nuevamente, las clases previas deben eliminarse
- El contenedor debe prepararse para recibir nuevas clases de estado
- No debe haber residuos visuales de cálculos anteriores

---

4. Requisitos No Funcionales

RNF-001: Usabilidad

Descripción: La aplicación debe ser intuitiva y fácil de usar.

Criterios de aceptación:

- Tiempo de aprendizaje menor a 2 minutos para un usuario nuevo
- Interfaz en español claramente legible
- Mensajes de error comprensibles para usuarios no técnicos
- Flujo de trabajo lineal y simple (ingresar → calcular → resultado)

RNF-002: Rendimiento

Descripción: La aplicación debe responder rápidamente a las interacciones del usuario.

Criterios de aceptación:

- La validación en tiempo real debe ser instantánea (< 50ms)
- El cálculo debe completarse en menos de 100ms
- La animación de transiciones debe ser fluida (60fps)
- La carga inicial de la página debe ser menor a 2 segundos en conexiones 3G

RNF-003: Compatibilidad

Descripción: La aplicación debe funcionar en diferentes navegadores y dispositivos.

Criterios de aceptación:

- Compatible con Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- Responsive design funcional en pantallas desde 320px de ancho
- Funcionalidad completa en dispositivos móviles (iOS/Android)
- Degradación elegante en navegadores antiguos

RNF-004: Accesibilidad

Descripción: La aplicación debe ser accesible para usuarios con diferentes capacidades.

Criterios de aceptación:

- Etiquetas semánticas correctas en HTML
- Contraste de color adecuado (WCAG AA mínimo)
- Navegación completa por teclado
- Atributos ARIA donde sea apropiado
- Mensajes de error asociados correctamente con los campos

RNF-005: Mantenibilidad

Descripción: El código debe ser fácil de mantener y extender.

Criterios de aceptación:

- Código JavaScript organizado en funciones claras y reutilizables

- Comentarios descriptivos en secciones clave
- Separación de responsabilidades (HTML, CSS, JS en archivos separados)
- Nombres de variables y funciones descriptivos en español
- Estructura CSS organizada por componentes

RNF-006: Seguridad

Descripción: La aplicación debe manejar las entradas de forma segura.

Criterios de aceptación:

- Validación de entrada en el cliente para prevenir valores inesperados
- No ejecución de código arbitrario
- Sanitización de entrada antes de mostrar en el DOM
- No almacenamiento de datos sensibles

RNF-007: Estética y Diseño

Descripción: La interfaz debe ser visualmente atractiva y profesional.

Criterios de aceptación:

- Paleta de colores coherente y agradable
- Espaciado consistente entre elementos
- Tipografía legible (familia Inter)
- Efectos visuales sutiles (sombras, transiciones, hover states)
- Diseño centrado y balanceado

---

5. Requisitos de Interfaz

5.1 Interfaz de Usuario

Estructura de la Página

- **Contenedor principal:** Centrado, con ancho máximo de 450px, fondo blanco, bordes redondeados y sombra
- **Título:** "Calculadora de Nota Final" en tamaño grande (3xl) y negrita

- **Descripción:** Texto explicativo en gris medio
- **Campos de entrada:** Dos inputs claramente etiquetados con sus porcentajes
- **Botón de acción:** Botón azul ancho y prominente
- **Área de resultado:** Contenedor dinámico que cambia de estilo según el resultado

Estados Visuales

1. **Estado Inicial:** Campos vacíos, sin resultado visible
2. **Estado de Entrada:** Campos con valores, validación en tiempo real
3. **Estado de Éxito:** Resultado verde para "ya aprobado"
4. **Estado Informativo:** Resultado azul para "nota necesaria"
5. **Estado de Error:** Resultado rojo para errores o "imposible aprobar"

Elementos Interactivos

- **Inputs:**
  - Placeholder con ejemplos
  - Borde azul en focus
  - Ring azul semi-transparente al enfocar
  - Padding cómodo para escritura
- **Botón:**
  - Color azul primario (#2563eb)
  - Hover: azul más oscuro + escala 101%
  - Focus: ring azul
  - Transición suave de 200ms

5.2 Interfaz de Hardware

No aplica. La aplicación es completamente basada en web y no requiere hardware especializado.

5.3 Interfaz de Software

- **Tailwind CSS CDN:** https://cdn.tailwindcss.com
- **Navegador Web:** Debe soportar ES6+ y API del DOM moderno

5.4 Interfaz de Comunicación

No aplica. La aplicación no realiza comunicación con servidores o APIs externas.

---

6. Modelo de Datos

6.1 Variables de Entrada

```
corte1: Number (0.0 - 5.0)
  - Nota del primer corte académico
  - Porcentaje: 33%

corte2: Number (0.0 - 5.0)
  - Nota del segundo corte académico
  - Porcentaje: 33%
```

6.2 Constantes del Sistema

```
notaMinima: 3.0
  - Nota mínima aprobatoria

porcentajeCorte1: 0.33
porcentajeCorte2: 0.33
porcentajeCorte3: 0.34
```

6.3 Variables de Salida

```
notaNecesaria: Number
  - Nota requerida en el tercer corte
  - Calculada con precisión de 2 decimales
  - Rango posible: -∞ a +∞ (aunque interpretado contextualmente)
```

6.4 Fórmula Matemática

```
notaNecesaria = (notaMinima - (corte1 × 0.33) - (corte2 × 0.33)) / 0.34
```

Donde:

- Si $notaNecesaria \leq 0$: El estudiante ya aprobó
- Si $0 < notaNecesaria \leq 5$: Es posible aprobar

- Si `notaNecesaria > 5`: Es imposible aprobar

---

7. Casos de Uso

CU-001: Calcular Nota para Aprobar

Actor: Estudiante

Precondiciones:

- El estudiante tiene acceso a la aplicación
- El estudiante conoce sus notas de los dos primeros cortes

Flujo Principal:

1. El estudiante abre la aplicación en su navegador
2. El sistema muestra la interfaz con dos campos vacíos
3. El estudiante ingresa la nota del primer corte (ej: 3.5)
4. El sistema valida la entrada en tiempo real
5. El estudiante ingresa la nota del segundo corte (ej: 2.8)
6. El sistema valida la entrada en tiempo real
7. El estudiante presiona el botón "Calcular Nota Necesaria"
8. El sistema realiza la validación final
9. El sistema calcula la nota necesaria
10. El sistema muestra el resultado con el estilo apropiado

Flujo Alternativo A - Ya Aprobó:

- En el paso 9, si la nota necesaria es ≤ 0
- El sistema muestra mensaje de felicitación en verde
- Caso de uso finaliza

Flujo Alternativo B - Imposible Aprobar:

- En el paso 9, si la nota necesaria es > 5
- El sistema muestra mensaje de imposibilidad en rojo

- Caso de uso finaliza

Flujo de Excepción E1 - Campos Vacíos:

- En el paso 8, si algún campo está vacío
- El sistema muestra error solicitando completar ambos campos
- El caso de uso retorna al paso 3

Flujo de Excepción E2 - Formato Inválido:

- En el paso 8, si el formato no es numérico válido
- El sistema muestra error de formato
- El caso de uso retorna al paso 3

Postcondiciones:

- El estudiante conoce la nota que necesita en el tercer corte
- El resultado permanece visible en pantalla

CU-002: Corregir Entrada Inválida

Actor: Estudiante

Precondiciones:

- El estudiante está ingresando una nota

Flujo Principal:

1. El estudiante comienza a escribir en un campo
2. El estudiante intenta ingresar un valor > 5 (ej: 5.8)
3. El sistema automáticamente limita el valor a 5.0
4. El campo muestra 5.0

Flujo Alternativo A - Valor Negativo:

- En el paso 2, el estudiante intenta ingresar un valor negativo
- El sistema automáticamente lo convierte a 0
- El campo muestra 0

Flujo Alternativo B - Caracter Inválido:

- En el paso 2, el estudiante intenta ingresar una letra
- El sistema rechaza el caracter
- El campo mantiene su valor anterior

Postcondiciones:

- El campo contiene un valor válido
- El estudiante puede continuar con la entrada

---

8. Especificaciones de Diseño

8.1 Paleta de Colores

Colores Principales:

- Azul primario: #2563eb (botones, focus)
- Azul hover: #1d4ed8
- Azul claro: #dbeafe (fondo informativo)
- Azul texto: #1e40af

Colores de Estado:

- Verde éxito: #d1fae5 (fondo), #065f46 (texto)
- Rojo error: #fee2e2 (fondo), #991b1b (texto)
- Gris fondo: #f3f4f6 (body)
- Gris texto: #1f2937 (títulos), #6b7280 (subtítulos)

Colores de UI:

- Blanco: #ffffff (contenedor principal)
- Gris borde: #d1d5db (inputs)
- Azul ring: rgba(59, 130, 246, 0.3) (focus ring)

8.2 Tipografía

- **Familia:** 'Inter', sans-serif
- **Título principal:** 3xl (1.875rem), peso 700

- **Etiquetas:** sm (0.875rem), peso 600
- **Texto regular:** base (1rem), peso 400-500
- **Resultado destacado:** 1.25rem, peso 700

8.3 Espaciado

- **Padding contenedor:** 1.5rem (mobile) / 2.5rem (desktop)
- **Margen entre campos:** 1rem
- **Padding inputs:** 0.5rem vertical / 1rem horizontal
- **Margen superior botón:** 2rem
- **Margen superior resultado:** 1.5rem

8.4 Efectos Visuales

- **Sombra contenedor:** 0 10px 15px rgba(0,0,0,0.1)
- **Border radius:** 0.5rem (inputs/botón), 1rem (contenedor)
- **Transiciones:** 150ms-300ms ease-in-out
- **Hover botón:** scale(1.01)
- **Focus ring:** 4px con 50% opacidad

---

9. Restricciones Técnicas

9.1 Tecnologías Requeridas

- HTML5
- CSS3 (con Tailwind CSS v3.x via CDN)
- JavaScript ES6+
- Navegador con soporte para:
  - addEventListener
  - classList API
  - Template literals
  - Arrow functions
  - const/let

- parseFloat

9.2 Limitaciones del Sistema

- No hay persistencia de datos (no se guardan cálculos previos)
- No hay funcionalidad de exportar o compartir resultados
- El sistema de porcentajes está fijo y no es configurable
- La nota mínima aprobatoria (3.0) está hardcodeada
- Dependencia de CDN externo para Tailwind CSS

9.3 Consideraciones de Implementación

- Todos los cálculos se realizan en el cliente
- No se requiere servidor backend
- La aplicación puede funcionar offline después de la carga inicial (excepto Tailwind CDN)
- No se utilizan cookies ni almacenamiento local
- El código es puramente declarativo y funcional

---

10. Pruebas y Validación

10.1 Casos de Prueba Funcionales

CP-001: Entrada Válida Normal

- **Entrada:** Corte1 = 3.5, Corte2 = 2.8
- **Resultado Esperado:** "Para aprobar, necesitas sacar 3.00 en el último corte"
- **Estado:** Info (azul)

CP-002: Ya Aprobado

- **Entrada:** Corte1 = 4.5, Corte2 = 4.5
- **Resultado Esperado:** "¡Felicidades! Ya has aprobado la materia. Necesitas un 0.00"
- **Estado:** Éxito (verde)

CP-003: Imposible Aprobar

- **Entrada:** Corte1 = 1.0, Corte2 = 1.0

- **Resultado Esperado:** "Necesitas sacar X.XX. ¡Es imposible aprobar!"
- **Estado:** Error (rojo)

CP-004: Campos Vacíos

- **Entrada:** Corte1 = "", Corte2 = ""
- **Resultado Esperado:** Error solicitando llenar ambos campos
- **Estado:** Error (rojo)

CP-005: Formato con Coma

- **Entrada:** Corte1 = "3,5", Corte2 = "4,2"
- **Resultado Esperado:** Acepta y calcula correctamente (convierte coma a punto)

CP-006: Validación de Rango Superior

- **Entrada:** Intentar escribir 6.0 en cualquier campo
- **Resultado Esperado:** Se limita automáticamente a 5.0

CP-007: Validación de Rango Inferior

- **Entrada:** Intentar escribir -1.0 en cualquier campo
- **Resultado Esperado:** Se convierte automáticamente a 0

CP-008: Guion Solitario

- **Entrada:** Corte1 = "-", Corte2 = "3.5"
- **Resultado Esperado:** Error indicando que el guion solo no es válido

CP-009: Valores Extremos Válidos

- **Entrada:** Corte1 = 0.0, Corte2 = 5.0
- **Resultado Esperado:** Calcula correctamente sin errores

CP-010: Formato Inválido

- **Entrada:** Corte1 = "3.4.5", Corte2 = "2.8"
- **Resultado Esperado:** Error de formato no válido

10.2 Casos de Prueba No Funcionales

CP-NF-001: Rendimiento de Validación

- **Prueba:** Escribir rápidamente en los campos
- **Criterio:** La validación debe responder sin lag perceptible

CP-NF-002: Compatibilidad Móvil

- **Prueba:** Abrir en dispositivo móvil (320px-480px)
- **Criterio:** Interfaz completamente funcional y legible

CP-NF-003: Accesibilidad de Teclado

- **Prueba:** Navegar usando solo Tab y Enter
- **Criterio:** Poder completar todo el flujo sin mouse

CP-NF-004: Múltiples Cálculos

- **Prueba:** Realizar 10 cálculos consecutivos con diferentes valores
- **Criterio:** Cada resultado debe mostrarse correctamente sin residuos previos

---

11. Apéndices

A. Glosario

- **Corte:** Período académico de evaluación (generalmente 5-6 semanas)
- **Nota aprobatoria:** Calificación mínima requerida para pasar una materia (3.0/5.0)
- **Validación en tiempo real:** Verificación de datos mientras el usuario escribe
- **NaN:** Not a Number, valor especial que indica un resultado matemático indefinido

B. Ejemplos de Cálculo

Ejemplo 1:

Corte 1: 3.5 (33%)
Corte 2: 3.0 (33%)
Nota necesaria = (3.0 - (3.5 × 0.33) - (3.0 × 0.33)) / 0.34
= (3.0 - 1.155 - 0.99) / 0.34
= 0.855 / 0.34
= 2.51

Ejemplo 2 - Ya Aprobado:

Corte 1: 4.5 (33%)
Corte 2: 4.0 (33%)
Nota necesaria = (3.0 - (4.5 × 0.33) - (4.0 × 0.33)) / 0.34
= (3.0 - 1.485 - 1.32) / 0.34
= 0.195 / 0.34
= -0.80 (ya aprobó)

Ejemplo 3 - Imposible:

Corte 1: 1.5 (33%)
Corte 2: 1.0 (33%)
Nota necesaria = (3.0 - (1.5 × 0.33) - (1.0 × 0.33)) / 0.34
= (3.0 - 0.495 - 0.33) / 0.34
= 2.175 / 0.34
= 6.40 (imposible, máximo es 5.0)

C. Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.0	29/10/2025	Equipo de Desarrollo	Versión inicial del documento SRS

Fin del Documento