

# Innovadores Digitales: Diseño y Programación

## Responsable para Soluciones Reales

*Tecnología e Informática | Tecnología | Aprendizaje Basado en Proyectos*

### Descripción

En este plan de clase, estudiantes de secundaria explorarán el proceso completo de diseño de soluciones digitales, desde la identificación de problemas y necesidades reales de usuarios hasta la creación y validación de prototipos de interfaces digitales. Aprenderán a programar aplicaciones básicas y a utilizar la inteligencia artificial de manera responsable, enfocándose en la experiencia de usuario y ajustando sus diseños con base en retroalimentación real. El aprendizaje se desarrollará a través de un proyecto concreto que responde a una necesidad detectada en su propio colegio, lo que conecta directamente los contenidos con su entorno y vida cotidiana.

Este enfoque activo, basado en proyectos colaborativos, les permitirá desarrollar competencias de investigación, diseño, programación y validación, preparándolos para enfrentar desafíos tecnológicos con ética y creatividad. Además, la experiencia práctica con herramientas de prototipado y entrevistas a usuarios fortalecerá su comprensión del impacto social de las soluciones digitales y la importancia de usar la inteligencia artificial con responsabilidad.

### Objetivos de Aprendizaje

- Identificar necesidades reales de usuarios mediante entrevistas breves y observación directa.
- Definir requerimientos funcionales claros y específicos para una solución digital.
- Diseñar prototipos de interfaces digitales utilizando herramientas de wireframe o prototipado navegable.
- Validar la solución diseñada con usuarios reales y ajustar el diseño según retroalimentación recibida.
- Aplicar principios éticos en el uso responsable de la inteligencia artificial dentro del proyecto digital.

### Recursos Necesarios

- Computadoras o tablets con acceso a internet (1 por cada 2 estudiantes)
- Software de prototipado digital (por ejemplo, Figma o Adobe XD) accesible en versión gratuita
- Herramientas para programar aplicaciones básicas (Scratch, App Inventor o similar)
- Hojas impresas con guías para entrevistas a usuarios y formatos para definición de requerimientos
- Cuadernos o libretas para anotaciones
- Proyector para presentación y explicación de contenidos
- Materiales para notas adhesivas y rotafolio o pizarra blanca para organizar ideas
- Acceso a ejemplos breves y videos cortos sobre diseño UX y ética en IA

### Requisitos Previos

- Conocimientos básicos sobre uso de computadoras y navegadores web
- Experiencia previa mínima con conceptos básicos de programación visual o lógica (por ejemplo, bloques en Scratch)
- Habilidades comunicativas básicas para realizar entrevistas y trabajo en equipo
- Comprensión elemental sobre qué es un usuario y la importancia de diseñar pensando en él

## Actividades

### Sesión 1: Descubriendo necesidades y diseñando la solución

#### Fase de Inicio

**Tiempo estimado: 10 minutos**

##### Propósito de la sesión:

Presentar la importancia de identificar problemas reales y entender a los usuarios para diseñar soluciones digitales efectivas y responsables.

##### Activación de conocimientos previos:

- **Docente:** Pregunta inicial: “¿Alguna vez has usado una aplicación o programa que te haya resultado difícil o poco útil? ¿Qué cambiarías para mejorarla?”
- **Estudiantes:** Responden con ejemplos breves y comparten sus experiencias.

##### Motivación y enganche:

- **Docente:** Presenta un dato curioso: “¿Sabías que el 70% de las aplicaciones fallan porque no entendieron bien a sus usuarios? Hoy aprenderemos a evitar eso creando prototipos basados en necesidades reales.”
- **Estudiantes:** Escuchan y reflexionan sobre la importancia de diseñar con el usuario en mente.

##### Contextualización:

- **Docente:** Conecta el tema con su entorno: “Vamos a crear una solución digital para un problema real de nuestro colegio, usando herramientas fáciles y con un enfoque responsable.”
- **Estudiantes:** Se preparan para iniciar un proyecto que tendrá un impacto real en su comunidad escolar.

#### Fase de Desarrollo

**Tiempo estimado: 45 minutos**

##### Presentación del contenido:

Se introduce el proceso de diseño de soluciones digitales, desde la identificación del problema hasta el prototipado, destacando el rol del usuario y los requerimientos funcionales.

## Actividad 1: Entrevista breve a usuarios para identificar necesidades

- **Objetivo:** Identificar necesidades reales y problemas de usuarios potenciales.
- **Instrucciones:**
  - **Docente:** Divide a los estudiantes en grupos de 3-4. Entrega una guía con preguntas para entrevistar a compañeros, profesores o personal del colegio sobre dificultades tecnológicas o necesidades no cubiertas.
  - Ejemplos de preguntas: “¿Qué problema tecnológico enfrentas en el colegio?”, “¿Qué te gustaría que una aplicación hiciera para facilitar tus actividades?”
  - **Estudiantes:** Realizan entrevistas rápidas (5 minutos), anotan respuestas y detectan al menos un problema o necesidad.
- **Organización:** Grupos de 3-4 estudiantes
- **Producto:** Lista breve de necesidades identificadas.
- **Tiempo:** 20 minutos
- **Rol docente:** Supervisa, orienta preguntas para profundizar y motiva a escribir respuestas claras.

## Actividad 2: Definición de requerimientos funcionales

- **Objetivo:** Definir requerimientos funcionales claros basados en necesidades detectadas.
- **Instrucciones:**
  - **Docente:** Explica con ejemplos qué es un requerimiento funcional (lo que la solución debe hacer). Ejemplo: “La aplicación debe permitir registrar asistencia rápidamente.”
  - **Estudiantes:** En grupo, transforman una necesidad en 3-5 requerimientos funcionales claros y específicos.
- **Organización:** Grupos de 3-4 estudiantes
- **Producto:** Documento con requerimientos funcionales para su proyecto.
- **Tiempo:** 15 minutos
- **Rol docente:** Ayuda a clarificar requerimientos y fomenta que sean medibles y alcanzables.

## Actividad 3: Diseño inicial de prototipo (wireframe simple)

- **Objetivo:** Crear un boceto o wireframe del prototipo de interfaz para la solución digital.
- **Instrucciones:**
  - **Docente:** Muestra una plantilla sencilla y ejemplos de wireframes. Explica la importancia de pensar en la experiencia del usuario.
  - **Estudiantes:** Usando papel o herramienta digital (según disponibilidad), dibujan la interfaz principal que responda a los requerimientos definidos.
- **Organización:** Grupos de 3-4 estudiantes
- **Producto:** Wireframe o boceto de la interfaz principal.
- **Tiempo:** 10 minutos

- **Rol docente:** Observa diseños, hace preguntas para fomentar claridad y facilidad de uso.

### **Diferenciación:**

- Para estudiantes que terminan antes: Proponer que agreguen detalles de interacción o flujo de pantallas adicionales.
- Para estudiantes que necesitan apoyo: Ofrecer ejemplos visuales y apoyo individual para definir requerimientos o dibujar wireframe.

### **Transición:**

**Docente:** Resume los avances y explica que en la siguiente sesión validarán sus prototipos y aprenderán sobre programación básica y uso responsable de IA.

## **Fase de Cierre**

### **Tiempo estimado: 5 minutos**

- **Síntesis:** Cada grupo comparte en 1-2 minutos su necesidad identificada y muestra su wireframe.
- **Reflexión metacognitiva (preguntas para estudiantes):**
  - ¿Qué aprendiste sobre la importancia de entender a los usuarios antes de diseñar?
  - ¿Cómo te ayudó la entrevista a definir mejor lo que debe hacer tu solución?
  - ¿Qué cambiarías en tu diseño inicial antes de programar la aplicación?
- **Retroalimentación:** Docente brinda comentarios positivos y sugerencias específicas para mejorar el enfoque y diseño.
- **Transferencia:** Anuncia que en la siguiente sesión construirán la solución digital y validarán con usuarios reales, aplicando principios de ética en el uso de IA.
- **Tarea/Reto:** Pensar en una función adicional que podría tener su aplicación para mejorar la experiencia del usuario, para compartir en la siguiente clase.

## **Sesión 2: Programando, validando y ajustando soluciones digitales con IA responsable**

### **Fase de Inicio**

### **Tiempo estimado: 10 minutos**

### **Propósito de la sesión:**

Revisar el diseño prototipado y preparar a los estudiantes para programar la aplicación y validar la experiencia del usuario, enfatizando el uso responsable de inteligencia artificial.

### **Activación de conocimientos previos:**

- **Docente:** Pregunta detonadora: “¿Cómo creen que podemos usar la inteligencia artificial para mejorar nuestra aplicación sin afectar la privacidad o seguridad de los usuarios?”
- **Estudiantes:** Responden en plenaria, compartiendo ideas y preocupaciones.

### **Motivación y enganche:**

- **Docente:** Muestra un video corto (2 min) sobre ejemplos responsables de IA en aplicaciones cotidianas y plantea el reto: “Vamos a crear una aplicación que use tecnología de forma ética y útil.”
- **Estudiantes:** Observan y se preparan para aplicar estos conceptos en su proyecto.

### **Contextualización:**

- **Docente:** Relaciona la programación y validación con el proyecto concreto del colegio para resolver un problema real.
- **Estudiantes:** Listos para desarrollar y mejorar su solución digital.

## **Fase de Desarrollo**

### **Tiempo estimado: 45 minutos**

#### **Actividad 1: Programación básica de la aplicación**

- **Objetivo:** Construir una aplicación funcional básica que cumpla con los requerimientos definidos.
- **Instrucciones:**
  - **Docente:** Divide a los estudiantes en los mismos grupos. Introduce una plataforma de programación visual (Scratch, App Inventor o similar) y explica brevemente cómo crear pantallas y botones.
  - **Estudiantes:** Programan las funciones básicas de su aplicación siguiendo sus requerimientos y wireframes, trabajando colaborativamente.
- **Organización:** Grupos de 3-4 estudiantes
- **Producto:** Versión inicial funcional de la aplicación o prototipo navegable digital.
- **Tiempo:** 30 minutos
- **Rol docente:** Apoya con dudas técnicas, sugiere mejoras y fomenta el trabajo en equipo.

#### **Actividad 2: Validación y ajuste con usuarios reales**

- **Objetivo:** Validar la experiencia de usuario y ajustar el diseño o funcionalidades según retroalimentación.
- **Instrucciones:**
  - **Docente:** Organiza la presentación de la aplicación a otros estudiantes o usuarios voluntarios del colegio. Proporciona una guía simple para obtener comentarios (qué les gustó, qué fue confuso, qué falta).
  - **Estudiantes:** Muestran su aplicación, recogen comentarios y anotan sugerencias para mejorar.
- **Organización:** Grupos de 3-4 estudiantes con 1-2 usuarios por grupo

- **Producto:** Registro de retroalimentación recibida y lista de ajustes a realizar.
- **Tiempo:** 15 minutos
- **Rol docente:** Facilita la interacción, ayuda a interpretar comentarios y fomenta una actitud constructiva.

### Diferenciación:

- Para estudiantes que terminan antes: Proponer que integren una función simple de IA (ejemplo: respuestas automáticas básicas o recomendaciones) usando bloques de código o plugins disponibles, siempre con enfoque ético.
- Para estudiantes que necesitan apoyo: Ofrecer plantillas de código y apoyo directo para completar funciones básicas.

### Transición:

**Docente:** Resume el proceso: “Hemos programado, validado y ahora ajustaremos para mejorar nuestra solución digital.” Explica que cerrarán con reflexión y planificación final.

## Fase de Cierre

### Tiempo estimado: 5 minutos

- **Síntesis:** Cada grupo comparte un ajuste importante que harán basado en la retroalimentación y cómo planean usar la IA de forma responsable.
- **Reflexión metacognitiva (preguntas para estudiantes):**
  - ¿Cómo te ayudó la retroalimentación de los usuarios a mejorar tu solución?
  - ¿Qué aprendiste sobre la responsabilidad al usar inteligencia artificial en aplicaciones?
  - ¿Qué habilidades nuevas desarrollaste durante este proyecto?
- **Retroalimentación:** Docente ofrece retroalimentación positiva y orientadora, destacando el esfuerzo y el aprendizaje colaborativo.
- **Transferencia:** Anima a pensar en otras necesidades que podrían abordar con estas herramientas y principios en el futuro.
- **Tarea/Reto:** Invitar a documentar el proceso y preparar una breve presentación o video sobre su proyecto para compartir en la comunidad escolar.

## Evaluación

### Tipo de evaluación:

- **Diagnóstica:** Al inicio de la sesión 1, mediante la pregunta detonadora para conocer conocimientos previos y experiencias con aplicaciones.
- **Formativa:** Durante las actividades de entrevistas, definición de requerimientos, prototipado, programación y validación; con observación directa y retroalimentación continua.

- **Sumativa:** Al cierre de la sesión 2, con la presentación del prototipo ajustado y la reflexión final sobre el proceso y uso responsable de IA.

### **Criterios de evaluación:**

- Identifica correctamente necesidades reales de usuarios (Objetivo 1) – Evaluado en la entrevista y listado de necesidades.
- Define requerimientos funcionales claros y adecuados (Objetivo 2) – Evaluado en el documento de requerimientos.
- Diseña prototipos coherentes con los requerimientos y experiencia de usuario (Objetivo 3) – Evaluado en wireframe y prototipo digital.
- Valida la solución con usuarios y ajusta el diseño (Objetivo 4) – Evaluado en la documentación de retroalimentación y mejoras realizadas.
- Aplica principios éticos en el uso de IA dentro del proyecto (Objetivo 5) – Evaluado en la reflexión final y presentación del proyecto.

### **Instrumentos sugeridos:**

- Lista de cotejo para observación durante entrevistas, definición y prototipado.
- Rúbrica para evaluar prototipo y programación (funcionalidad, usabilidad, diseño).
- Portafolio digital o físico con evidencias de entrevistas, requerimientos, wireframes y prototipos.
- Autoevaluación y coevaluación con preguntas guía para reflexión individual y grupal.

### **Evidencias de aprendizaje:**

- Notas y registros de entrevistas con usuarios.
- Documento con requerimientos funcionales.
- Wireframes y prototipos digitales o bocetos.
- Aplicación o prototipo programado funcional.
- Registro de retroalimentación de usuarios y ajustes realizados.
- Respuestas a preguntas de reflexión y presentación final del proyecto.

## **Enriquecimientos**

### **Desarrollo - Ejemplos**

#### **Ejemplos Prácticos y Casos de Estudio para "Innovadores Digitales"**

Estos ejemplos y casos están diseñados para que los estudiantes de secundaria puedan conectar sus aprendizajes con situaciones reales y cotidianas, facilitando la aplicación práctica de los conceptos durante las dos sesiones de trabajo.

- **Ejemplo 1: Aplicación para Mejorar la Comunicación entre Estudiantes y Profesores**

- *Contexto:* En la escuela, los estudiantes sienten que no siempre pueden expresar dudas o recibir feedback rápido sobre tareas y proyectos.

- *Problema a resolver:* Falta de canales efectivos para consultas y seguimiento académico.
- *Usuario:* Estudiantes y profesores del colegio.
- *Requerimientos funcionales:*
  - Permitir enviar preguntas y recibir respuestas.
  - Notificaciones cuando hay respuestas nuevas.
  - Interfaz amigable y fácil de usar para adolescentes y docentes.
- *Actividad ABP:* Los estudiantes, en equipo, entrevistan a compañeros y profesores para identificar necesidades específicas, luego diseñan un prototipo simple (wireframe) de la aplicación con herramientas digitales o papel, validan el diseño mostrando el prototipo a otros usuarios para recibir retroalimentación y ajustan el diseño según comentarios.

#### • **Ejemplo 2: Programa para Gestionar el Uso Responsable de Dispositivos Electrónicos en la Escuela**

- *Contexto:* Muchos estudiantes usan sus dispositivos móviles en clase, lo que a veces afecta la concentración y el ambiente de aprendizaje.
- *Problema a resolver:* Necesidad de promover el uso responsable y controlado de dispositivos electrónicos.
- *Usuario:* Estudiantes del colegio.
- *Requerimientos funcionales:*
  - Registro del tiempo de uso permitido por día.
  - Alertas para tomar descansos o limitar el uso.
  - Opciones para que los estudiantes personalicen sus límites.
- *Actividad ABP:* Los alumnos realizan entrevistas breves con compañeros para conocer hábitos y problemas, definen requerimientos, crean prototipos de la interfaz de la aplicación que permita configurar límites y alertas, prueban el prototipo con usuarios reales y ajustan el diseño para facilitar la usabilidad.

#### • **Ejemplo 3: Chatbot Simple para la Biblioteca Escolar**

- *Contexto:* La biblioteca del colegio tiene muchas consultas frecuentes sobre horarios, disponibilidad de libros y normas, pero el personal no siempre está disponible para responder.
- *Problema a resolver:* Falta de respuestas rápidas para consultas comunes en la biblioteca.
- *Usuario:* Estudiantes y profesores que usan la biblioteca.
- *Requerimientos funcionales:*
  - Responder preguntas frecuentes automáticamente.
  - Guiar a los usuarios en el uso de los servicios de la biblioteca.
  - Interfaz sencilla que pueda integrarse en la web o app del colegio.
- *Actividad ABP:* Los estudiantes investigan las preguntas más comunes entrevistando a usuarios, diseñan un flujo de conversación simple mediante un prototipo navegable, validan con usuarios reales y mejoran el diseño para hacerlo más intuitivo y completo.

Conexión con Objetivos de Aprendizaje

| Objetivo                                              | Ejemplo(s) Relacionado(s)                      | Actividad ABP relevante                                              |
|-------------------------------------------------------|------------------------------------------------|----------------------------------------------------------------------|
| Identificar necesidades reales de usuarios            | Todos (Comunicación, Uso Responsable, Chatbot) | Entrevistas breves y observación directa para definir el problema    |
| Definir requerimientos funcionales                    | Todos                                          | Listar funcionalidades clave basadas en las entrevistas              |
| Diseñar prototipos de interfaz                        | Todos                                          | Crear wireframes o prototipos digitales sencillos                    |
| Validar una solución con usuarios y ajustar el diseño | Todos                                          | Presentar prototipos a usuarios reales, recopilar feedback y mejorar |

Estos ejemplos permiten a los estudiantes aplicar habilidades de diseño y programación responsable, considerando siempre la experiencia del usuario y el impacto social de sus soluciones, en un tiempo adecuado para las dos sesiones planificadas.