## Sistema de Alerta de Nivel de Agua

**Título del Proyecto:** (Título claro y conciso. Ejemplo: "Sistema de Monitoreo de Calidad del Aire")

**Integrantes:** (Nombres completos y números de identificación, indicando roles principales. Ejemplo: "Juan Pérez (Hardware), María Gómez (Software)")

**Fecha de Presentación:**

### 1. Introducción

* **Descripción General:**

Este proyecto consiste en el diseño de un sistema básico que permite monitorear el nivel de agua en un tanque o recipiente utilizando un sensor ultrasónico o un sensor de flotador conectado a un microcontrolador Arduino. Cuando el nivel de agua alcanza un umbral crítico (muy bajo o muy alto), el sistema activa una alerta local a través de un zumbador y muestra un mensaje en la consola. Además, registra los eventos en un archivo de texto. Esta versión simplificada es ideal para introducir a los estudiantes en conceptos de sensado físico, lógica de control y generación de alertas, con aplicaciones prácticas en prevención de desbordamientos o control de riego manual. Está dirigido a personas interesadas en automatización básica, educación técnica y soluciones de bajo costo para entornos rurales o domésticos.

* **Objetivo General:**

Diseñar un sistema simple de monitoreo de nivel de agua que alerte al usuario cuando se detecten niveles críticos mediante sensores de bajo costo.

**Objetivos Específicos:**

* Configurar un sensor ultrasónico o de flotador con Arduino para medir el nivel de agua.
* Activar una señal sonora o visual al superar un umbral predefinido.
* Mostrar mensajes de alerta en consola para notificar al usuario.
* Registrar los eventos de alerta en un archivo de texto.
* Validar el sistema en un entorno de prueba simulado (recipiente con agua).
* **Alcance:**

Este sistema se enfocará exclusivamente en la detección de nivel alto o bajo de agua. La salida será una alerta simple mediante zumbador o LED, y no se incluye control automático de bombas. El almacenamiento será local, en un archivo de texto simple, sin uso de bases de datos ni conexión remota. Tampoco se contempla visualización gráfica ni interfaz web. El enfoque es netamente educativo y demostrativo, permitiendo su uso en talleres escolares o como base para automatizaciones futuras.

### 2. Fundamentación

* **Justificación:**

El monitoreo del nivel de agua en tanques, sistemas de riego o depósitos es esencial para evitar desperdicios, prevenir inundaciones y asegurar el abastecimiento adecuado. Sin embargo, muchas soluciones comerciales requieren inversiones altas o conocimiento avanzado. Este proyecto ofrece una alternativa sencilla y funcional, ideal para comunidades, estudiantes o usuarios interesados en tecnologías accesibles. El sistema permite familiarizarse con el uso de sensores físicos, programación básica y respuesta automatizada ante condiciones críticas. Además, al estar alineado con los objetivos del curso, fomenta el aprendizaje práctico de la electrónica aplicada y la gestión de eventos físicos a través de software..

* **Estado del Arte:**

### Revisión de Literatura:

### Manuales técnicos de control de nivel de agua en sitios como PlataformaExtension.cl explican la importancia de monitoreo automatizado en entornos rurales y domésticos.

### Proyectos educativos en Arduino Project Hub muestran aplicaciones simples de sensores ultrasónicos para medir líquidos.

### Blogs de makers y educación técnica detallan cómo sistemas de alerta simples con zumbadores y LEDs pueden ser implementados en casa o escuelas.

### En entornos industriales, sensores ultrasónicos y de flotador son ampliamente utilizados, aunque requieren mayor inversión.

### Documentos académicos sugieren este tipo de proyectos como introducción ideal al diseño de sistemas embebidos en ingeniería electrónica.

### Análisis Comparativo:

| Proyecto / Fuente | Sensor utilizado | Complejidad | Comunicación | Valor agregado del presente proyecto |
| --- | --- | --- | --- | --- |
| Soluciones industriales | Sensor ultrasónico | Alta | IoT/PLC | Enfoque educativo, bajo costo |
| Proyectos académicos complejos | HC-SR04 + válvulas | Media | Inalámbrica | Eliminación de conectividad innecesaria |
| Presente proyecto | HC-SR04 o flotador | Baja | Ninguna | Fácil implementación, accesible y didáctico |

### Marco Teórico:

### Un sensor ultrasónico como el HC-SR04 mide distancias utilizando ondas de sonido de alta frecuencia. Al emitir un pulso y recibir su eco, el sensor calcula el tiempo de retorno y con ello la distancia al nivel del agua. Esta distancia permite inferir si el nivel está dentro del rango esperado. Alternativamente, un sensor de flotador puede detectar niveles altos o bajos mediante contacto físico. El microcontrolador Arduino interpreta estas señales y, si detecta un valor crítico, activa una alarma como un zumbador o un LED. Los eventos se registran en un archivo de texto en la computadora, a través de comunicación serial. Este sistema combina fundamentos de física (ultrasonido), electrónica digital y programación básica, siendo una excelente puerta de entrada al mundo de la automatización.

### 3. Marcos de Trabajo

Se detallarán metodologías, herramientas y cronograma.

* **Metodología de Desarrollo:** Descripción del enfoque (100-150 palabras) (ej., prototipado rápido, desarrollo iterativo). Explicación de etapas, justificación de la elección y descripción de la gestión del proyecto.
* **Herramientas y Tecnologías:** Enumeración de recursos:
  + **Hardware:** Listado de componentes electrónicos con modelos y especificaciones. Tabla detallada: Componente, Modelo, Especificaciones, Cantidad, Costo (opcional), Imagen (opcional).
  + **Software:** Listado de lenguajes, librerías, frameworks y entornos. Lista detallada: Nombre, Versión, Función.
* **Cronograma:** Cronograma de actividades (diagrama de Gantt sugerido) desde la investigación hasta la presentación, incluyendo todas las etapas, fechas y dependencias (si aplica).

### 4. Diseño del Proyecto

Se presentarán diagramas y modelos de la arquitectura y funcionamiento.

* **Diagramas de Conexión de Circuitos:** Esquemas detallados (usar Fritzing, Tinkercad). Diagramas para cada etapa/módulo, facilidad de comprensión, indicación de función de cada componente y conexión, y lista de materiales (BOM) correspondiente.
* **Diagramas de Flujo:** Representación gráfica de la lógica del programa (símbolos estándar). Descripción clara del inicio/fin, todas las rutas de ejecución y comentarios explicativos.
* **Diagramas UML (Opcional, recomendado):** Si aplica POO, incluir diagramas relevantes (clases, secuencia, estados). Explicación del propósito y relación con el proyecto. Usar herramienta de modelado (draw.io, Lucidchart). Diagramas precisos, completos y comprensibles.
* **Diseño de la Interfaz (si aplica):** Maquetas/wireframes de la interfaz. Descripción de la experiencia de usuario (UX) y usabilidad, incluyendo esquemas, descripción de elementos, flujo de navegación, consideraciones de usabilidad y herramientas de diseño (Balsamiq, Figma).

### 5. Implementación

Se detallará la construcción del proyecto.

* **Descripción Detallada:** Explicación de la construcción (200-300 palabras):
  + **Hardware:** Descripción paso a paso del ensamblaje del circuito, conexión de componentes, resolución de problemas y fotografías.
  + **Software:** Estructura del programa, función de módulos, algoritmos, manejo de datos del sensor e implementación de funcionalidades.
* **Código Fuente:** Fragmentos relevantes con comentarios. Si es extenso, incluir como anexo o enlace a repositorio (GitHub, GitLab) público y organizado.
* **Pruebas y Resultados:** Descripción de pruebas, presentación de resultados (datos, gráficos), evaluación del cumplimiento de objetivos. Incluir métodos de prueba, datos, análisis, comparación con objetivos e identificación de errores/mejoras.
* **Dificultades Encontradas y Soluciones:** Documentación de problemas y soluciones, con descripción específica de la naturaleza del problema y los pasos seguidos.

### 6. Conclusiones

Se presentarán las reflexiones finales.

* **Resumen de Logros:** Recapitulación de los logros y el cumplimiento de objetivos (150-200 palabras). Destacar resultados y resumir el cumplimiento de objetivos.
* **Análisis Crítico:** Evaluación del proyecto (150-200 palabras). Evaluación objetiva, identificación de fortalezas/debilidades, y consideración del impacto y potencial.
* **Trabajo Futuro:** Proposición de mejoras, extensiones o aplicaciones (100-150 palabras). Sugerir ideas, proponer extensiones y explorar aplicaciones en otros contextos.

### 7. Referencias

*Fuentes recomendadas para comenzar:*

*Ramírez-Rivera, J., González-Estrada, M. G., & González-Hernández, R. (2024). Design and implementation of an automatic water level control system using microcontrollers. Results in Engineering, 21, 102829.* [*https://doi.org/10.1016/j.rineng.2024.102829*](https://doi.org/10.1016/j.rineng.2024.102829)

*Ali, F., Rehman, M. U., Yousaf, M. H., Nawaz, M., & Shabbir, M. S. (2024). IoT-based smart water tank monitoring and control system. Case Studies in Chemical and Environmental Engineering, 10, 100942.* [*https://doi.org/10.1016/j.cscee.2024.100942*](https://doi.org/10.1016/j.cscee.2024.100942)

*Patel, D., Gohil, R., & Patel, N. (2024). Comparative study of water level measurement techniques using ultrasonic, capacitive and radar sensors. Flow Measurement and Instrumentation, 92, 102777.* [*https://doi.org/10.1016/j.flowmeasinst.2024.102777*](https://doi.org/10.1016/j.flowmeasinst.2024.102777)

*Chandrakant, P., & Lokhande, P. (2023). Open-source water tank level monitoring system using NodeMCU. Open Hardware, 12, e00427.* [*https://doi.org/10.1016/j.ohx.2023.e00427*](https://doi.org/10.1016/j.ohx.2023.e00427)

*Anilkumar, R., & Harikrishna, B. S. (2023). Low-cost smart water level monitoring system using ESP32. Open Hardware, 11, e00414.* [*https://doi.org/10.1016/j.ohx.2023.e00414*](https://doi.org/10.1016/j.ohx.2023.e00414)

*Yu, C. L., Lin, C. T., & Chang, C. F. (2016). Design and implementation of water level measurement system based on pressure sensor. Measurement, 90, 525–531.* [*https://doi.org/10.1016/j.measurement.2016.04.019*](https://doi.org/10.1016/j.measurement.2016.04.019)

* Listar fuentes (libros, artículos, sitios web) con formato de citación consistente (APA, IEEE).
* Garantizar la inclusión de todas las fuentes citadas.
* Sugerir gestor de referencias (Zotero, Mendeley).

### 8. Anexos (Opcional)

* Incluir información adicional (código completo, datos de pruebas, manuales).
* Anexos claramente etiquetados y organizados.