## Sistema de Detección de Intrusiones (Ciberseguridad)

**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:**

El presente proyecto propone el diseño e implementación de un sistema básico de detección de intrusos basado en un sensor de movimiento PIR. Esta versión simplificada está pensada como una solución introductoria a los sistemas de seguridad electrónica, con énfasis en la detección de presencia y la generación de alertas locales. Al detectar movimiento, el sistema activa una alarma sonora o visual y registra el evento en un archivo de texto. El proyecto está diseñado para ambientes domésticos, educativos o pequeños espacios que requieren monitoreo sin altos costos. No requiere internet ni servidores, lo cual lo hace ideal para la enseñanza de fundamentos de ciberseguridad física y electrónica básica. Los beneficiarios son principalmente estudiantes, docentes y personas interesadas en implementar soluciones económicas de vigilancia.

* **Objetivo General:**

Diseñar un sistema sencillo para la detección de movimiento usando un sensor PIR, con alertas locales y registro de eventos.

**Objetivos Específicos:**

* Configurar un sensor PIR con Arduino para detectar presencia.
* Registrar los eventos en un archivo de texto de forma cronológica.
* Activar una alerta sonora (buzzer) o mensaje en consola al detectar movimiento.
* Validar el funcionamiento del sistema en un entorno controlado.
* Documentar el proceso de desarrollo y pruebas del sistema.
* **Alcance:**

El sistema detectará únicamente movimiento mediante un sensor PIR. Al detectar una intrusión, emitirá una señal sonora o mostrará un mensaje en pantalla, y registrará la hora del evento en un archivo de texto plano. No se utilizarán sensores adicionales, conectividad inalámbrica, almacenamiento en base de datos ni notificaciones remotas. Su funcionamiento se limitará al entorno local y a la ejecución desde consola o script simple. Esta versión no incluye interfaz web, ni reconocimiento de patrones o análisis avanzado de datos.

### 2. Fundamentación

* **Justificación:**

Los sistemas de seguridad son esenciales para la protección de espacios físicos, sin embargo, muchas soluciones comerciales son costosas o complejas para ser implementadas en entornos educativos o por personas sin experiencia técnica. Este proyecto busca introducir los conceptos básicos de detección de intrusos a través de una solución accesible, de bajo costo y fácil de implementar. Permite comprender cómo sensores de movimiento pueden ser integrados a un microcontrolador para generar respuestas automáticas ante eventos físicos. Además, permite explorar aspectos básicos de ciberseguridad física, incluyendo monitoreo, registro de eventos y activación de alarmas. Alineado con los objetivos del curso, se trabajan habilidades de programación, electrónica básica y documentación técnica, fomentando el aprendizaje práctico y aplicado.

* **Estado del Arte:**

**Revisión de Literatura:**

1. Invgate Blog (2023) presenta principios de seguridad perimetral, destacando la importancia de sistemas de detección como una primera línea de defensa.
2. KPMG (2023) discute la integración de seguridad física y digital en la protección de infraestructuras, resaltando sensores como parte fundamental.
3. LinkedIn Pulse aborda cómo sensores PIR son ampliamente utilizados en sistemas domésticos y escolares por su fiabilidad y bajo costo.
4. Guiasegura.com proporciona guías prácticas para implementar sistemas básicos de detección sin requerir tecnologías avanzadas.
5. Avigilon Blog analiza tendencias actuales en monitoreo físico, incluyendo sistemas híbridos que combinan hardware simple con lógica local.

**Análisis Comparativo:**

| **Proyecto / Fuente** | **Tecnología** | **Almacenamiento** | **Alerta** | **Valor agregado del presente proyecto** |
| --- | --- | --- | --- | --- |
| Sistemas comerciales | PIR + cámaras | Nube o servidor | App móvil o central | Bajo costo, ideal para prototipos y entornos educativos |
| Proyectos académicos avanzados | Varios sensores | Bases de datos | Notificaciones web | Simplicidad, enfocado en fundamentos |
| Presente proyecto | Sensor PIR | Archivo de texto | Buzzer o consola | Accesible, educativo, sin dependencia externa |

### Marco Teórico:

El sensor PIR (Passive Infrared) detecta cambios en la radiación infrarroja emitida por cuerpos calientes en movimiento, como personas. Estos sensores son ampliamente usados en sistemas de seguridad por su bajo costo y facilidad de implementación. El microcontrolador Arduino se encarga de leer la señal del sensor y activar una respuesta, como encender un LED, sonar un buzzer o registrar el evento. La comunicación entre Arduino y el computador puede hacerse por puerto serie, permitiendo que un script en Python o el propio entorno del Arduino IDE registre eventos. La información se almacena en un archivo de texto que sirve como bitácora. Este sistema es una introducción práctica a los principios de la seguridad electrónica, la automatización y el registro de eventos físicos.

### 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:*

InvGate. (2023). *Seguridad perimetral informática: qué es y cómo implementarla*.<https://blog.invgate.com/es/seguridad-perimetral-informatica>

KPMG. (2023). *Cyber security and physical security: working better together*.<https://kpmg.com/uk/en/blogs/home/posts/2023/03/cyber-security-and-physical-security.html>

Eibic. (s.f.). *La seguridad física como primera línea de defensa*. LinkedIn. [https://www.linkedin.com/pulse/la-seguridad-física-como-primera-línea-de-eibic/](https://www.linkedin.com/pulse/la-seguridad-f%C3%ADsica-como-primera-l%C3%ADnea-de-eibic/)

Guía Segura. (s.f.). *Detección de intrusos en seguridad física*.<https://guiasegura.com/seguridad-fisica/deteccion-intrusos/?utm_source=chatgpt.com>

Avigilon. (s.f.). *Guía de seguridad física: lo que debe saber*.<https://www.avigilon.com/es/blog/physical-security-guide?utm_source=chatgpt.com>

* 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.