## Sistema de Alarma de Seguridad Inteligente con Reconocimiento Facial

**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 plantea el diseño de un sistema básico de alarma de seguridad, basado en un sensor de movimiento PIR y una cámara conectada a una Raspberry Pi. En esta versión simplificada no se implementa reconocimiento facial en tiempo real, sino que se realiza una detección de rostros en imágenes estáticas capturadas al momento de detectar movimiento. El objetivo es registrar intrusiones mediante imágenes y emitir una alerta visual o sonora. El sistema también registra los eventos en un archivo de texto para su revisión posterior. Este enfoque permite introducirse en conceptos fundamentales de visión por computadora y seguridad electrónica sin la complejidad de sistemas de inteligencia artificial en tiempo real. Está dirigido a estudiantes, docentes o usuarios que deseen aprender sobre integración de sensores, cámaras y análisis básico de imágenes.

* **Objetivo General:**

Desarrollar un sistema básico de seguridad que detecte movimiento, capture una imagen del evento y realice una detección de rostro en imágenes estáticas.

**Objetivos Específicos:**

* Detectar movimiento utilizando un sensor PIR conectado a una Raspberry Pi.
* Capturar imágenes mediante un módulo de cámara al activarse la detección.
* Utilizar OpenCV para identificar rostros en las imágenes capturadas.
* Registrar los eventos y resultados en un archivo de texto.
* Emitir una alerta (sonido o LED) como respuesta al evento.
* **Alcance:**

Este proyecto no incluye reconocimiento facial completo ni análisis en tiempo real. Solo se detectan rostros en imágenes individuales capturadas al momento de una intrusión. La base de datos de rostros conocidos y la interfaz web de gestión serán excluidas en esta etapa. Tampoco se utilizarán servicios externos como Telegram ni se implementará transmisión de video en vivo. El almacenamiento será local, y las alertas serán visuales o auditivas simples. Es un sistema ideal para explorar fundamentos de seguridad electrónica y visión por computadora.

### 2. Fundamentación

* **Justificación:**

La necesidad de soluciones de seguridad accesibles y efectivas es creciente, especialmente en contextos domésticos o escolares donde se requiere un sistema básico que permita detectar y registrar intrusiones. Este proyecto ofrece una solución introductoria que combina hardware simple (sensor PIR y cámara) con software de análisis de imágenes, sin depender de conexiones remotas ni de modelos complejos de aprendizaje automático. Permite a los estudiantes comprender el flujo de detección, captura y análisis visual con herramientas de código abierto, fomentando el aprendizaje práctico en electrónica, Python y visión por computadora. Además, se alinea con los objetivos del curso al integrar sensores, programación y técnicas básicas de inteligencia artificial.

* **Estado del Arte:**

### Revisión de Literatura:

### ResearchGate (2018) analiza el uso de reconocimiento facial tradicional y profundo, destacando el uso de OpenCV como punto de partida.

### UCV Repositorio documenta el desarrollo de sistemas de vigilancia con sensores PIR y cámaras, aplicables en hogares.

### ArXiv.org presenta trabajos sobre detección de rostros con redes neuronales, aunque fuera del alcance de esta versión básica.

### Blogs especializados en visión por computadora recomiendan comenzar con detección en imágenes antes de avanzar a video en vivo.

### Proyectos de bajo costo documentados en línea muestran que una Raspberry Pi es suficiente para implementar este tipo de sistemas de forma accesible.

### Análisis Comparativo:

| Proyecto / Fuente | Reconocimiento facial | Tipo de imagen | Interfaz | Valor agregado del presente proyecto |
| --- | --- | --- | --- | --- |
| Sistemas comerciales | En tiempo real | Video streaming | Web/App | Costo muy bajo, enfoque educativo |
| Proyectos con IA avanzada | Aprendizaje profundo | Imágenes + video | Web compleja | Usabilidad, sin requerir entrenamiento complejo |
| Presente proyecto | Detección simple | Imágenes estáticas | Alerta local | Ideal para aprendizaje inicial y prototipos |

### Marco Teórico:

### El sensor PIR detecta movimiento basado en cambios de radiación infrarroja, permitiendo activar sistemas de seguridad de forma automática. En este proyecto, una Raspberry Pi controla la lógica y se conecta a una cámara para capturar imágenes en el momento de una detección. Luego, mediante la biblioteca OpenCV en Python, se realiza la detección de rostros en la imagen capturada utilizando clasificadores Haar o redes preentrenadas ligeras. Los resultados (rostro detectado o no) se registran en un archivo de texto. Aunque no se realiza reconocimiento facial (identificación), el sistema permite entender los fundamentos de captura de imágenes y procesamiento visual. Se utilizan herramientas básicas pero potentes, lo que convierte este proyecto en un excelente punto de partida para aplicaciones más avanzadas.

### 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:*

*Valverde, F. J. C. (2023). Sistema de seguridad con reconocimiento facial y detección de movimiento. Universidad César Vallejo.* [*https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/157736/B\_Valverde\_FJC-SD.pdf?sequence=1&isAllowed=y*](https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/157736/B_Valverde_FJC-SD.pdf?sequence=1&isAllowed=y)

*Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. arXiv.* [*https://arxiv.org/pdf/1503.03832*](https://arxiv.org/pdf/1503.03832)

*Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. ACM Computing Surveys (CSUR), 35(4), 399–458.* [*https://arxiv.org/pdf/1804.06655*](https://arxiv.org/pdf/1804.06655)

*Sultana, N. (s.f.). Deep learning for computer vision. Dirzon.* [*https://www.dirzon.com/file/telegram/somali%20educational%20books/deeplearningforcomputervision.pdf*](https://www.dirzon.com/file/telegram/somali%20educational%20books/deeplearningforcomputervision.pdf)

*Jafri, R., Arabnia, H. R., & Wei, Y. (2018). Face recognition: From traditional to deep learning methods. ResearchGate.* [*https://www.researchgate.net/publication/328685305\_Face\_Recognition\_From\_Traditional\_to\_Deep\_Learning\_Methods*](https://www.researchgate.net/publication/328685305_Face_Recognition_From_Traditional_to_Deep_Learning_Methods)

*HyperVerge. (s.f.). Face recognition algorithms: A complete guide.* [*https://hyperverge-co.translate.goog/blog/face-recognition-algorithm/?\_x\_tr\_sl=en&\_x\_tr\_tl=es&\_x\_tr\_hl=es&\_x\_tr\_pto=tc*](https://hyperverge-co.translate.goog/blog/face-recognition-algorithm/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

* 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.