Diagrama

Descripción generada automáticamente

**PLAN DE PROYECTO**

**JUSTIFICACIÓN**

**SISTEMA DE RECONOCIMIENTO FACIAL PARA REGISTRO DE ASISTENCIA ESCOLAR**

**Introducción**

En el presente documento se detalla el plan de desarrollo del proyecto de software cuyo objetivo es implementar un **sistema de reconocimiento facial con cámara de visión inteligente** para el registro de asistencia en una escuela primaria. Este sistema permitirá monitorear la asistencia diaria de los alumnos, generar reportes semanales y mensuales, y emitir alertas en caso de ausencias frecuentes o ingreso de personas no autorizadas al plantel.

**Importancia y Relevancia del Proyecto**

La educación es un pilar fundamental en la formación de la sociedad, y garantizar un control adecuado de la asistencia de los alumnos es crucial para su desarrollo académico. Tradicionalmente, el registro de asistencia se ha realizado de forma manual, lo que conlleva errores, pérdida de tiempo y posibles fraudes.

Este proyecto busca automatizar y optimizar este proceso mediante el uso de reconocimiento facial con visión inteligente, lo que proporciona múltiples beneficios:

* **Precisión y Eficiencia:** Se elimina el error humano en el registro manual de asistencia, asegurando una recopilación de datos precisa y en tiempo real.
* **Seguridad Escolar:** El sistema permite detectar accesos no autorizados, garantizando un ambiente más seguro dentro del plantel.
* **Monitoreo de la Asistencia:** Facilita la identificación de patrones de inasistencia y permite a los docentes y directivos tomar medidas preventivas.
* **Generación Automática de Reportes:** Reduce la carga administrativa de los docentes al automatizar la elaboración de reportes de asistencia.
* **Accesibilidad y Conveniencia:** Los reportes pueden consultarse en cualquier momento y desde distintos dispositivos.

Este proyecto se lleva a cabo porque responde a una necesidad real dentro del ámbito educativo, alineándose con la transformación digital y el aprovechamiento de la inteligencia artificial para mejorar los procesos escolares. Además, su implementación no solo beneficiará a la institución educativa en cuestión, sino que podría replicarse en otras escuelas y niveles educativos.

**Chat GPT**

|  |  |
| --- | --- |
| PLAN DE PROYECTO: SISTEMA DE RECONOCIMIENTO FACIAL PARA REGISTRO DE ASISTENCIA ESCOLAR | |
| INTRODUCCIÓN | En el presente documento se detalla el plan de desarrollo del proyecto de software cuyo objetivo es implementar un **sistema de reconocimiento facial con cámara de visión inteligente** para el registro de asistencia en una escuela primaria. Este sistema permitirá monitorear la asistencia diaria de los alumnos, generar reportes semanales y mensuales, y emitir alertas en caso de ausencias frecuentes o ingreso de personas no autorizadas al plantel. |
| OBJETIVOS | **Objetivo General**  Desarrollar un sistema de reconocimiento facial para el registro automatizado de asistencia escolar, optimizando la gestión de asistencia y seguridad dentro del plantel.  **Objetivos Específicos**   * Implementar un sistema de visión inteligente para el análisis de rostros. * Desarrollar una base de datos segura para almacenar los registros de asistencia. * Diseñar una interfaz que permita la generación de reportes automáticos. * Establecer mecanismos de alerta en caso de ausencias recurrentes o ingreso de personas ajenas. |
| ALCANCE DEL PROYECTO | * Registro automático de asistencia mediante reconocimiento facial. * Generación de reportes semanales y mensuales. * Notificaciones en tiempo real ante ausencias recurrentes. * Identificación de accesos no autorizados. * Acceso al sistema desde dispositivos móviles y computadoras. |
| METODOLOGÍA DE DESARROLLO | El equipo ha decidido emplear la metodología **Scrum**, debido a su enfoque iterativo e incremental, lo que permitirá desarrollar el sistema en sprints cortos, ajustando funcionalidades según el feedback recibido. Scrum nos proporcionará:   * Mayor flexibilidad ante cambios en los requerimientos. * Entregas parciales y funcionales del sistema en cada sprint. * Una mejor distribución de tareas entre los tres integrantes del equipo. * Evaluaciones periódicas para asegurar la calidad del producto final. |
| ANÁLISIS DE REQUISITOS | **Requisitos Funcionales**   * Registro de rostros de alumnos y personal autorizado. * Análisis de imágenes en tiempo real. * Almacenamiento de registros de asistencia. * Generación de reportes de asistencia. * Alertas automáticas para padres y docentes.   **Requisitos No Funcionales**   * Seguridad en el almacenamiento de datos. * Bajo tiempo de respuesta del sistema. * Compatibilidad con hardware de visión artificial. * Interfaz intuitiva y fácil de usar. |
| DISEÑO DEL SOFTWARE | * **Arquitectura:** Cliente-servidor con base de datos en la nube. * **Tecnologías:** Python (OpenCV, TensorFlow), Firebase para almacenamiento, Vue.js para la interfaz. * **Diagrama de flujo:** Se definirá en los primeros sprints. |
| PLAN DE DESARROLLO Y CRONOGRAMA | |  |  |  | | --- | --- | --- | | **SPRINT** | **DURACIÓN** | **OBJETIVOS** | | **1** | **2 semanas** | Definición de requisitos y configuración del entorno de desarrollo. | | **2** | **3 semanas** | Desarrollo del módulo de reconocimiento facial. | | **3** | **2 semanas** | Implementación del almacenamiento y base de datos. | | **4** | **2 semanas** | Desarrollo de la interfaz y generación de reportes. | | **5** | **2 semanas** | Pruebas, ajustes finales y despliegue. | |
| PRUEBAS Y VALIDACIÓN | * **Pruebas unitarias:** Evaluación de cada componente del software. * **Pruebas de integración:** Verificación del funcionamiento conjunto de los módulos. * **Pruebas de usuario:** Validación con docentes y personal de la escuela. |
| IMPLEMENTACIÓN Y DESPLIEGUE | * Implementación en una escuela piloto. * Configuración del hardware necesario. * Capacitación para el personal docente. |
| MANTENIMIENTO Y SOPORTE | * Corrección de errores post-implementación. * Actualización de funcionalidades según requerimientos de la escuela. * Monitoreo y optimización del rendimiento. |
| PRESUPUESTO Y RECURSOS | * **Hardware:** Cámaras de visión artificial. * **Software:** Licencias, servicios en la nube. * **Recursos Humanos:** Desarrollo, pruebas y soporte. |
| EVALUACIÓN Y DOCUMENTACIÓN FINAL | * Informe final del proyecto. * Manual de usuario y guía técnica. * Recomendaciones para futuras mejoras. |

**Deep seek**

|  |  |
| --- | --- |
| PLAN DE PROYECTO:  SISTEMA DE ASISTENCIA CON VISIÓN ARTIFICIAL | |
| INTRODUCCIÓN | El proyecto consiste en desarrollar un sistema de registro de asistencia para una escuela primaria mediante una cámara de visión inteligente. El software analizará los rostros de los alumnos para automatizar la asistencia diaria, generar reportes semanales/mensuales, alertar sobre inasistencias recurrentes y detectar personas ajenas al plantel.  **Problemática para resolver:**   * Eliminar el registro manual de asistencia, propenso a errores. * Mejorar la seguridad al identificar intrusos. * Proporcionar reportes en tiempo real para seguimiento académico y administrativo. |
| OBJETIVOS | **Objetivos**   * Desarrollar un sistema de registro de asistencia automatizado en 12 semanas. * Integrar un modelo de visión artificial con precisión mínima del 95% en reconocimiento facial. * Generar reportes personalizados para docentes y administradores. |
| ALCANCE DEL PROYECTO | * **Incluye:**   + Reconocimiento facial de alumnos registrados.   + Alertas en tiempo real por inasistencias recurrentes o intrusos.   + Dashboard con reportes semanales/mensuales. * **Excluye:**   + Integración con sistemas externos (ejemplo: plataformas de pagos).   + Desarrollo de hardware (se usará una cámara estándar).   **Stakeholders**   * **Cliente:** Escuela primaria (usuarios finales: docentes y administrativos). * **Equipo:** 3 integrantes (roles en Scrum). * **Patrocinadores:** Profesor de la asignatura (supervisión académica). |
| METODOLOGÍA DE DESARROLLO | **2. Metodología de Desarrollo Seleccionada: Scrum**  **Justificación de la Elección**  Scrum es la metodología idónea para este proyecto debido a:   1. **Equipo pequeño (3 integrantes):**    * Roles claros: *Product Owner* (gestión de requisitos), *Scrum Master* (facilitador), *Equipo de Desarrollo* (programación y pruebas).    * Favorece la comunicación constante y la colaboración cercana. 2. **Requisitos flexibles:**    * El alcance inicial podría ajustarse (ejemplo: priorizar funcionalidades como detección de intrusos antes que reportes mensuales). 3. **Entregas incrementales:**    * Permite mostrar avances funcionales en sprints cortos (ejemplo: MVP con registro básico de asistencia en 2 semanas). 4. **Adaptabilidad:**    * Ideal para proyectos innovadores con componentes tecnológicos complejos (visión artificial), donde pueden surgir desafíos técnicos imprevistos. |
| ANÁLISIS DE REQUISITOS | **Requisitos Funcionales**   1. Registro de rostros de alumnos en una base de datos. 2. Detección de rostros no registrados (intrusos). 3. Generación de reportes en PDF/Excel con filtros por fecha o alumno. 4. Envío de alertas por correo o SMS ante inasistencias recurrentes (>3 días).   **Requisitos No Funcionales**   1. **Seguridad:** Cifrado de datos biométricos (GDPR/LOPD). 2. **Rendimiento:** Tiempo de respuesta <2 segundos por reconocimiento. 3. **Usabilidad:** Interfaz intuitiva para usuarios no técnicos.   **Viabilidad**   * **Técnica:** Uso de librerías de visión artificial (OpenCV, Dlib) y frameworks ágiles (Python/Django). * **Económica:** Costos bajos (herramientas open-source, equipo ya cuenta con computadoras). * **Operativa:** La escuela ya tiene conexión a internet y cámaras básicas. |
| DISEÑO DEL SOFTWARE | **Arquitectura**   * **Frontend:** Interfaz web responsive (HTML/CSS/JavaScript). * **Backend:** Django (gestión de usuarios y reportes). * **Visión Artificial:** Modelo entrenado con imágenes de alumnos. * **Base de Datos:** PostgreSQL (almacenamiento de rostros y registros).   **Diagramas**   * **Flujo de datos:** Cámara → Procesamiento → Almacenamiento → Dashboard. * **Casos de uso:** Registro de asistencia, generación de reportes, detección de intrusos. |
| PLAN DE DESARROLLO Y CRONOGRAMA | **Estructura de Desglose del Trabajo (WBS)**   | **SPRINT** | **ENTREGABLE** | | --- | --- | | **SPRINT 1** | Prototipo de reconocimiento facial (MVP) | | **SPRINT 2** | Sistema de alertas y base de datos | | **SPRINT 3** | Dashboard de reportes | | **SPRINT 4** | Pruebas de seguridad y optimización |   **Cronograma**   * **Duración total:** 12 semanas (3 sprints de 3 semanas + 1 sprint para ajustes). * **Hitos clave:**   + Semana 3: MVP funcional.   + Semana 6: Integración con alertas.   + Semana 9: Dashboard completo.   + Infraestructura: Servidor local o cloud (AWS Free Tier). |
| ROLES DEL EQUIPO | 1. **Product Owner:** Define prioridades del backlog (ejemplo: priorizar detección de intrusos). 2. **Scrum Master:** Organiza reuniones diarias (*daily standups*) y elimina obstáculos. 3. **Equipo de Desarrollo:** Programa, prueba y documenta.   **Herramientas**   * **Gestión de tareas:** Trello/Jira (tablero Scrum con columnas *To Do*, *In Progress*, *Done*). * **Comunicación:** Slack/Teams para coordinación. * **Control de versiones:** GitHub. |
| JUSTIFICACIÓN FINAL DE SCRUM | * **Adaptación al equipo:** 3 integrantes permiten una comunicación ágil y roles definidos sin burocracia. * **Flexibilidad:** Permite ajustar el alcance si surgen problemas técnicos (ejemplo: mejorar precisión del reconocimiento facial en un sprint adicional). * **Entrega incremental:** Ideal para un proyecto académico con plazos ajustados, ya que el profesor puede evaluar avances parciales. |
| PRESUPUESTO Y RECURSOS | **Recursos**   * **Humanos:** 3 integrantes (roles Scrum). * **Tecnológicos:**   + Herramientas: Python, OpenCV, Django, PostgreSQL, Git.   **Presupuesto**   * **Costos:** $0 (uso de herramientas gratuitas). * **Contingencia:** 20% del tiempo para resolver problemas técnicos. |
| ANEXOS | **.**   * **Prototipo de interfaz:** Bocetos en Figma. * **Cronograma detallado:** Diagrama de Gantt. * **Repositorio GitHub:** Enlace al código fuente. |

**Gemini**

|  |  |
| --- | --- |
| PLAN DE PROYECTO: SISTEMA INTELIGENTE DE ASISTENCIA ESCOLAR | |
| INTRODUCCIÓN | Este proyecto busca desarrollar un sistema innovador para el control de asistencia en escuelas primarias, utilizando tecnología de visión por computadora para el reconocimiento facial de los alumnos. El sistema generará reportes automáticos y alertas en tiempo real para mejorar la gestión de la asistencia y la seguridad en el plantel. |
| OBJETIVOS | Diseñar y construir un sistema innovador de reconocimiento facial para la gestión de asistencia escolar, utilizando tecnologías de visión por computadora e inteligencia artificial, que permita optimizar el control de asistencia, generar información valiosa para la toma de decisiones y promover un entorno escolar más seguro. |
| ALCANCE DEL PROYECTO | **Funcionalidades**:   * + Reconocimiento facial de alumnos en la entrada del colegio.   + Registro automático de asistencia diaria.   + Generación de reportes semanales y mensuales de asistencia.   + Alertas en tiempo real sobre faltas recurrentes y acceso de personas no autorizadas.   + Interfaz web para consulta de reportes y gestión del sistema.   **Entregables**:   * + Software de reconocimiento facial.   + Base de datos de rostros de alumnos.   + Interfaz web para reportes y gestión.   + Documentación del sistema.   **Criterios de Aceptación**:   * + Precisión del reconocimiento facial superior al 95%.   + Generación de reportes sin errores.   + Funcionamiento estable del sistema en condiciones reales. |
| ANÁLISIS DE REQUISITOS | **Requisitos Funcionales**:   * + El sistema debe identificar y registrar la hora de ingreso de cada alumno.   + Debe generar reportes personalizables por alumno, grado y grupo.   + Debe enviar alertas por correo electrónico o SMS sobre faltas y accesos no autorizados.   **Requisitos No Funcionales**:\*   * + El sistema debe ser fácil de usar y mantener.   + Debe garantizar la privacidad y seguridad de los datos de los alumnos.   + Debe ser escalable para adaptarse a diferentes tamaños de escuela. |
| DISEÑO DE LA ARQUITECTURA DEL SOFTWARE | **Componentes**:   * + Cámara de visión inteligente con capacidad de procesamiento.   + Servidor central para almacenamiento y procesamiento de datos.   + Base de datos para rostros y registros de asistencia.   + Interfaz web para usuarios.   **Tecnologías**:   * + Lenguaje de programación: Python (con bibliotecas de visión por computadora como OpenCV).   + Base de datos: PostgreSQL (por su robustez y seguridad).   + Framework web: Django o Flask (para la interfaz web). |
| PLAN DE DESARROLLO | **Sprints**: Ciclos de desarrollo de 2 semanas.  **Roles**:   * + Product Owner: Define y prioriza los requisitos.   + Scrum Master: Facilita el proceso Scrum.   + Equipo de Desarrollo: 3 integrantes (desarrolladores).   **Ceremonias**:   * + Sprint Planning: Planificación del sprint.   + Daily Scrum: Reunión diaria de 15 minutos.   + Sprint Review: Revisión del trabajo realizado.   + Sprint Retrospective: Mejora continua del proceso. |
| PLAN DE PRUEBAS | **Tipos de Pruebas**:   * + Pruebas unitarias: Verificación de cada componente individual.   + Pruebas de integración: Verificación de la interacción entre componentes.   + Pruebas de aceptación: Verificación del cumplimiento de los requisitos por parte del usuario.   **Herramientas**:   * + Frameworks de testing de Python (pytest, unittest).   + Herramientas de simulación de escenarios. |
| PLAN DE IMPLEMENTACIÓN | **Fases**:   * + Configuración del hardware (cámara, servidor).   + Instalación y configuración del software.   + Carga de la base de datos de rostros.   + Pruebas en entorno real.   + Capacitación a usuarios.   **Puesta en Marcha**: Inicialmente en un grupo reducido de alumnos para ajustes y luego a toda la escuela. |
| PLAN DE GESTIÓN DE RIESGOS | **Riesgos**:   * + Fallo en el reconocimiento facial por condiciones de luz o ángulos.   + Sobrecarga del sistema por gran cantidad de alumnos.   + Vulnerabilidad de la base de datos.   **Mitigación**:   * + Entrenamiento del sistema con diversas condiciones.   + Optimización del código y uso de hardware escalable.   + Implementación de medidas de seguridad (encriptación, acceso restringido). |
| PRESUPUESTO Y CRONOGRAMA | **Presupuesto**:   * + Costos de hardware (cámara, servidor).   + Costos de software (licencias, desarrollo).   + Costos de personal (desarrolladores).   **Cronograma**:   * + Definir sprints y entregas parciales.   + Estimar la duración total del proyecto (ejemplo: 3 meses). |
| PLAN DE COMUNICACIÓN | **Canales**:   * + Reuniones diarias del equipo.   + Reuniones semanales con stakeholders.   + Correo electrónico para comunicaciones formales.   **Frecuencia**:   * + Diaria para el equipo de desarrollo.   + Semanal para stakeholders. |
| PLAN DE MANTENIMIENTO | **Fases**:\*   * + Corrección de errores.   + Actualizaciones de software.   + Soporte técnico a usuarios. |
| EQUIPO DEL PROYECTO | * + Product Owner   + Scrum Master   + Desarrolladores |

**DESCRIPCION DE LOS MIEMBROS DEL EQUIPO 3**

|  |  |
| --- | --- |
| INTEGRANTES | CONOCIMIENTOS, HABILIDADES Y APTITUDES |
| ITZEL PAULINA ESQUIVEL URENDA | **Conocimientos**   * Educación primaria y metodologías de enseñanza. * Desarrollo de aplicaciones educativas (*MateLand*). * Evaluación formativa y estrategias pedagógicas de la Nueva Escuela Mexicana.   **Habilidades**   * Gestión de proyectos con metodologías ágiles (Scrum). * Desarrollo y aplicación de estrategias para mejorar el aprendizaje en matemáticas. * Pensamiento crítico y resolución de problemas. * Diseño de materiales didácticos digitales. * Trabajo en equipo y liderazgo educativo.   **Aptitudes**   * Innovación y creatividad en la enseñanza. * Compromiso con la mejora educativa a través de la tecnología. * Capacidad de adaptación a nuevos retos. * Orientación al logro de objetivos y resultados. |
| JUAN PABLO GONZÁLEZ RIVERA | **Conocimientos:**   * Conocimientos básicos en inteligencia artificial. * Lenguajes de programación: Python, C++, JavaScript, HTML y CSS. * Enfoque en desarrollo web. * Aprendiendo sobre el proyecto de reconocimiento facial.   **Habilidades:**   * Desarrollo de software: Capacidad para diseñar, programar y optimizar aplicaciones. * Pensamiento crítico: Análisis profundo de problemas para encontrar soluciones eficientes. * Trabajo en equipo: Colaboración efectiva en proyectos multidisciplinarios. * Resolución de problemas: Habilidad para identificar, analizar y solucionar errores en código o sistemas. * Lógica de programación: Capacidad para estructurar algoritmos y desarrollar software de manera eficiente. * Gestión del tiempo: Organización y priorización de tareas para cumplir con plazos establecidos. * Investigación y aprendizaje continuo: Habilidad para mantenerse actualizado en nuevas tecnologías y tendencias en programación e inteligencia artificial.   **Aptitudes:**   * **Autodidacta:** Capacidad para aprender de forma independiente y continuar con el desarrollo profesional. * **Proactividad:** Disposición para buscar soluciones y mejorar procesos sin esperar indicaciones. * **Creatividad:** Habilidad para innovar en el desarrollo de proyectos tecnológicos. * **Resolución de problemas:** Capacidad para analizar y enfrentar desafíos de manera efectiva. * **Adaptabilidad:** Facilidad para ajustarse a nuevas tecnologías y metodologías de trabajo. |
| LUIS FERNANDO LUPERCIO RAMIREZ | **Conocimientos**   * **Educación**: Profesor de primaria * **Programación**: Python * **Inteligencia Artificial**: Fundamentos de IA, chatbots con Rasa   **Habilidades**   * **Docencia**: Planeación y evaluación educativa, manejo de grupos. * **Desarrollo de software**: Creación de aplicaciones con Python Tkinter * **Redacción**: Elaboración de informes educativos, escritura técnica sobre redes. * **Gestión de proyectos**: Desarrollo de apps, creación de un chatbot.   **Aptitudes**   * **Autodidacta**: Aprende nuevos temas constantemente (IA, programación, redes). * **Paciencia y liderazgo**: Manejo de alumnos con diversas necesidades. * **Pensamiento crítico**: Análisis profundo en educación, tecnología e IA. * **Adaptabilidad**: Capacidad para cambiar y mejorar estrategias según el contexto. |

**REFERENCIAS BIBLIOGRAFICAS**

**CHAT GPT**

* Boehm, B. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61-72. https://doi.org/10.1109/2.59
* Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.
* Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org. <https://www.scrumguides.org/>
* Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2nd ed.). Springer.
* Van Rossum, G., & Drake, F. L. (2009). *The Python language reference manual*. Network Theory Ltd.

**DEEP SEEK**

* European Union. (2016). *Regulation (EU) 2016/679 (GDPR): General Data Protection Regulation*. <https://gdpr-info.eu/>
* McKinney, W. (2018). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O’Reilly Media.
* OpenCV. (n.d.). *OpenCV Documentation*. OpenCV. <https://docs.opencv.org/>
* Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner’s Approach* (9th ed.). McGraw-Hill Education.
* Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Scrum.org. <https://www.scrum.org/resources/scrum-guide>
* Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.

**GEMINI**

**Metodologías de Desarrollo de Software (Scrum y Ágiles)**

* Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org.
* Rubin, K. S. (2013). *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley Professional.
* Cockburn, A. (2002). *Agile software development*. Addison-Wesley.

**Tecnologías de Visión por Computadora y Reconocimiento Facial**

* Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with OpenCV*. O'Reilly Media.
* Rosebrock, A. (2015). *Practical Python and OpenCV: An introduction to computer vision*. PyImageSearch.

**Gestión de Proyectos de Software**

* PMI. (2021). *A guide to the project management body of knowledge (PMBOK guide) (7th ed.)*. Project Management Institute.