# Trabajo practico inicial

Materia:

* Laboratorio de construcción de software

Comisión:

* COM-01

Alumnos:

* Thomas Rossi
* Matias Meira
* Facundo Ibañez

Docentes:

* Juan Carlos Monteros
* Amim

Fechas de entrega:

* Primera entrega: Lunes 18/08
* Segunda entrega: Lunes 25/08
* Tercera entrega: Lunes 01/09
* Cuarta entrega: Lunes 08/09

## **Índice**

1. Introducción – pág. 3
2. Objetivo del trabajo práctico – pág. 3
3. Objetivos específicos de entregas – pág. 3  
   3.1. Primera entrega: Investigación y Preparación del proyecto – pág. 3  
   3.2. Segunda entrega: Preprocesamiento y Visualización de Datos – pág. 3  
   3.3. Tercera entrega: Prototipo de reconocimiento facial – pág. 3  
   3.4. Cuarta entrega: Integración y Publicación – pág. 3
4. Investigación: Procesos Productivos en PyMEs Alimentarias – pág. 4  
   4.1. Contexto de las PyMEs alimentarias – pág. 4  
   4.2. Etapas típicas de un proceso productivo en PyMEs alimentarias – pág. 4  
   - a) Recepción y almacenamiento de materias primas – pág. 4  
   - b) Preparación – pág. 4  
   - c) Producción / Procesamiento – pág. 4  
   - d) Control de calidad en proceso – pág. 4  
   - e) Envasado y etiquetado – pág. 5  
   - f) Almacenamiento y distribución – pág. 5  
   4.3. Variables clave para análisis productivo – pág. 5  
   4.4. Relación con el control de ingreso por reconocimiento facial – pág. 5  
   4.5. Conclusión – pág. 5
5. Investigación: Tecnologías de Reconocimiento Facial para Control de Ingreso en PyMEs Alimentarias – pág. 6  
   5.1. Contexto – pág. 6  
   5.2. Principales tecnologías disponibles – pág. 6  
   - a) OpenCV – pág. 6  
   - b) FaceAPI.js – pág. 6  
   - c) DeepFace – pág. 7  
   - d) Microsoft Face API (Azure) – pág. 7  
   5.3. Factores a considerar para el proyecto – pág. 7  
   5.4. Integración con el TP – pág. 7  
   5.5. Conclusión – pág. 8
6. Justificación de las Tecnologías a Utilizar – pág. 8
7. Conjuntos de datos simulados – pág. 10

**Link de Trello (tablero de User Storys) del grupo:**

<https://trello.com/invite/b/689ca238d2ce2dd49a5bc415/ATTId458a1d0ca986f893621b8461b9790ec46F47BB4/proyecto>

**Link de GitHub del grupo:**

<https://github.com/FacuIb01/prototipo_reconocimiento_facial.git>

## **Introducción**

El presente trabajo práctico tiene como objetivo el desarrollo de un sistema de control de ingresos basado en reconocimiento facial. La iniciativa surge como parte de las actividades académicas de la Universidad Nacional de General Sarmiento y busca aplicar conocimientos de programación, gestión de proyectos, investigación y gestión de datos para resolver un problema real: Controlar el ingreso/egreso del personal dentro de una PyME.

Mediante la implementación de herramientas de detección y comparación de rostros, el sistema pretende ofrecer una alternativa eficiente y confiable para el control de acceso.

# **Objetivo del trabajo practico**

El objetivo de este trabajo será el de introducirnos en el análisis de datos productivos de una PyME, investigar distintas herramientas de reconocimiento facial para implementarlas en nuestro trabajo y diseñar e implementar un sistema de control de ingreso/egreso de personal.

Objetivos específicos de entregas:

Primera entrega: Investigación y Preparación del proyecto

* Analizar procesos productivos en PyMEs alimenticias.
* Investigar herramientas y librerías de reconocimiento facial (OpenCV, FaceAPI, etc.).
* Crear un repositorio compartido para trabajo colaborativo.
* Definir un conjunto de datos simulados para representar la producción.
* Documentar formalmente los objetivos del trabajo práctico.

Segunda entrega: Pre procesamiento y Visualización de Datos

* Limpiar y estructurar datos productivos.
* Generar gráficos e indicadores (producción por tipo, desperdicio, eficiencia).
* Documentar hallazgos y conclusiones (insights).
* Preparar una presentación preliminar con resultados obtenidos.

Tercera entrega: Prototipo de reconocimiento facial.

* Diseñar una interfaz de usuario con captura de imagen en tiempo real.
* Implementar el reconocimiento facial utilizando OpenCV.
* Registrar ingresos y egresos con sello de tiempo (timestamp).
* Simular una base de datos de empleados.
* Documentar la arquitectura y diseño del prototipo.

Cuarta entrega: Integración y Publicación

* Conectar las visualizaciones de datos con el prototipo de reconocimiento facial.
* Publicar una demo en la nube (Colab, GitHub Pages, Render, Vercel, etc.).
* Preparar la presentación final del proyecto.
* Reflexionar sobre cómo integrar lo aprendido en el Trabajo Práctico Principal.

# **Investigación: Procesos Productivos en PyMEs Alimentarias**

1. Contexto de las PyMEs alimentarias

Las pequeñas y medianas empresas (PyMEs) del sector alimentario en Argentina se caracterizan por:  
- Producción a menor escala que grandes industrias, pero con alto grado de flexibilidad y capacidad de personalización.  
- Enfoque en productos frescos o de elaboración artesanal/semindustrial.  
- Requerimientos estrictos en inocuidad, higiene y trazabilidad.  
- Alta dependencia de mano de obra calificada y procesos parcialmente manuales.  
- Necesidad de optimizar recursos para mejorar competitividad.  
  
En el marco del proyecto, comprender estos procesos es clave para:  
- Definir variables de control y registro de la producción.  
- Identificar puntos críticos donde el ingreso/egreso del personal impacta en la eficiencia.  
- Simular datos productivos para el análisis y visualización.

2. Etapas típicas de un proceso productivo en PyMEs alimentarias

Aunque varían según el tipo de producto (panificados, conservas, lácteos, cárnicos, etc.), en general se identifican estas fases:

a) Recepción y almacenamiento de materias primas:  
- Control de calidad inicial (temperatura, aspecto, etiquetado, fechas de vencimiento).  
- Registro de ingreso de insumos (manual o digital).  
- Almacenamiento según condiciones requeridas (cámara de frío, depósito seco).

b) Preparación:  
- Lavado y sanitización de materias primas.  
- Pesado y medición de ingredientes.  
- Pre procesado (corte, molienda, mezclado inicial).

c) Producción / Procesamiento:  
- Transformación física o química (cocción, fermentación, pasteurización, amasado, envasado).  
- Control de variables críticas (tiempo, temperatura, humedad, pH).  
- Uso de maquinaria semiautomática o manual.

d) Control de calidad en proceso:  
- Muestreo y verificación visual/organoléptica.  
- Registro de incidencias.  
- Ajuste de parámetros si es necesario.

e) Envasado y etiquetado:  
- Uso de materiales aptos para alimentos.  
- Etiquetado con información obligatoria (fecha, lote, ingredientes, datos del fabricante).

f) Almacenamiento y distribución:  
- Conservación según tipo de producto.  
- Registro de salida.  
- Entrega a clientes o puntos de venta.

3. Variables clave para análisis productivo

En línea con la US-01 y US-04 del TP, el registro y análisis de datos puede considerar:  
- Cantidad producida (unidades, kg).  
- Tiempo de producción por lote o turno.  
- Porcentaje de desperdicio.  
- Tiempo de inactividad por ausencia o retraso de personal.  
- Consumo de insumos y rendimiento.  
- Eficiencia global (output vs. capacidad instalada).  
  
Estas variables servirán para generar visualizaciones como las previstas en la US-07 (producción por tipo, desperdicio, eficiencia).

4. Relación con el control de ingreso por reconocimiento facial

La implementación del prototipo de control de ingreso/egreso del personal puede:  
- Asegurar puntualidad y registro exacto de horarios.  
- Relacionar asistencia con productividad, detectando correlaciones.  
- Reducir errores o fraudes en el fichaje manual.  
- Generar alertas automáticas en caso de ausencias en puestos críticos.  
  
Ejemplo de aplicación:  
Si un operario clave en la etapa de envasado no ingresa a tiempo, el sistema puede anticipar una demora y reflejarlo en el dashboard de producción.

5. Conclusión

La investigación sobre los procesos productivos en PyMEs alimentarias no solo permite entender el flujo de trabajo y sus puntos críticos, sino que es fundamental para:  
- Simular datos realistas que alimenten el análisis.  
- Diseñar un prototipo útil que aporte valor real al control de la producción.  
- Cumplir con las historias de usuario US-01, US-04, US-06, US-07 y US-15 del proyecto.

# **Investigación: Tecnologías de Reconocimiento Facial para Control de Ingreso en PyMEs Alimentarias**

1. Contexto

La tecnología de reconocimiento facial se propone como herramienta para:  
- Automatizar el registro de ingreso/egreso del personal.  
- Aumentar la precisión y evitar fraudes en el fichaje manual.  
- Integrar los datos de asistencia con métricas de producción para análisis y visualización.  
  
El objetivo de la investigación es la selección de una de las tecnologías para desarrollar un prototipo funcional (Sprint 3 – US-10 a US-14) que use una librería o API de reconocimiento facial y pueda ejecutarse en entornos accesibles para una PyME (PC de escritorio, notebook, celular).

2. Principales tecnologías disponibles

a) OpenCV:

- Tipo: Librería de visión por computadora de código abierto.  
- Lenguajes: Python, C++, Java.  
- Ventajas:  
 • Gratuita y muy documentada.  
 • Permite detección y reconocimiento facial en tiempo real.  
 • Compatible con múltiples sistemas operativos.  
 • Se puede integrar con modelos como dlib o DeepFace para mejorar precisión.  
- Desventajas:  
 • Curva de aprendizaje media-alta.  
 • El reconocimiento facial puro requiere entrenamiento adicional o librerías complementarias.

b) FaceAPI.js:

- Tipo: Librería JavaScript para reconocimiento facial en el navegador.  
- Lenguajes: JavaScript.  
- Ventajas:  
 • Funciona directamente en el navegador, sin instalaciones por fuera de la web.  
 • Usa modelos preentrenados de TensorFlow.js.  
 • Ideal para prototipos web rápidos (GitHub Pages, Vercel).  
- Desventajas:  
 • Mayor consumo de recursos en cliente.  
 • Precisión menor en entornos con baja iluminación o cámaras de baja calidad.

c) DeepFace:

- Tipo: Framework Python de alto nivel para reconocimiento facial.  
- Lenguajes: Python.  
- Ventajas:  
 • Funciones listas para usar (verificación, análisis de atributos, búsqueda).  
 • Alta precisión con preentrenamiento.  
- Desventajas:  
 • Requiere más potencia de procesamiento.  
 • No está pensado para ejecutarse en el navegador por lo tanto requiere mayor esfuerzo integrarlo en distintos dispositivos.

d) Microsoft Face API (Azure):

- Tipo: API en la nube para detección y reconocimiento facial.  
- Lenguajes: Compatible con múltiples lenguajes vía REST API.  
- Ventajas:  
 • Alta precisión y escalabilidad.  
 • No requiere hardware potente localmente.  
 • Funciones avanzadas como identificación por grupos y detección de emociones.  
- Desventajas:  
 • Servicio de pago (aunque tiene cuota gratuita limitada).  
 • Depende de conexión a internet.

3. Factores a considerar para el proyecto

En el marco de una PyME alimentaria, la elección tecnológica debe tener en cuenta:  
- Costo: Preferencia por soluciones open source o gratuitas.  
- Requerimientos de hardware: Debe funcionar en PCs convencionales o notebooks.  
- Facilidad de integración con la base de datos simulada y dashboard de producción.  
- Velocidad de procesamiento para registrar fichajes sin demoras.  
- Robustez ante condiciones reales (cambio de iluminación, uso de cofias, barbijos).

4. Integración con el TP

Para el prototipo del Sprint 3:  
- Se puede usar OpenCV como base para detección y extracción de rostros.  
- Complementar con DeepFace o dlib para el reconocimiento (identificación).  
- En caso de querer una demo web ligera, FaceAPI.js es ideal para mostrar funcionalidad sin instalar software adicional.  
- La base de datos simulada (US-13) contendrá registros de empleados con ID y rostro asociado.  
- El sistema debe registrar timestamp y enviar la información para visualización junto con datos de producción (US-15).

5. Conclusión

Luego de analizar todas las opciones se decidió que se avanzará con la opción de “Face-Api.js” ya que brinda mayor flexibilidad para utilizarse en distintas plataformas al poder desarrollarse en una web (PC de escritorio, celulares y tablets). De esta manera, el costo de integración del sistema en una PyME se reduce solamente al pago de servidores y dispositivos con cámara para ejecutar la pagina web.

## **Justificación de las Tecnologías a Utilizar**

Para el desarrollo del presente trabajo práctico se seleccionaron tecnologías que permiten cumplir con los objetivos planteados de manera eficiente, asegurando accesibilidad, facilidad de implementación y bajo costo, aspectos clave en el contexto de una PyME. La elección se fundamenta tanto en los requerimientos del proyecto como en las competencias técnicas disponibles.

**JavaScript** se adoptó como lenguaje principal por su versatilidad y su capacidad de ser utilizado tanto en el frontend como en el backend. Esto posibilita un desarrollo unificado, reduciendo la complejidad del proyecto y permitiendo aprovechar conocimientos previos. Asimismo, JavaScript ofrece una amplia gama de librerías que facilitan la implementación de funcionalidades avanzadas, como el reconocimiento facial y la visualización de datos.

**Express.js**, framework minimalista para Node.js, se utilizará para el desarrollo del servidor y la creación de APIs REST. Esta herramienta permite gestionar el flujo de información entre el frontend y la base de datos, además de exponer servicios para el enrolamiento de empleados, registro de ingresos y egresos, y consulta de indicadores productivos. Su flexibilidad lo hace especialmente adecuado para proyectos académicos y prototipos.

**PostgreSQL** fue la base de datos seleccionada por su robustez, escalabilidad y carácter open source. Este motor de base de datos relacional garantiza integridad en los registros de empleados, fichajes y variables de producción. Asimismo, su soporte para tipos de datos avanzados (como arreglos y JSON) resulta útil para almacenar descriptores faciales generados por el sistema de reconocimiento.

En cuanto a las tecnologías específicas para el reconocimiento facial, tras la investigación de alternativas como OpenCV, DeepFace y Microsoft Face API, se optó por **Face-Api.js**. Esta librería, basada en JavaScript y TensorFlow.js, permite ejecutar detección y reconocimiento de rostros directamente en el navegador, sin necesidad de instalar software adicional ni disponer de hardware especializado. Esta característica reduce los costos de integración y facilita el despliegue en dispositivos comunes como PCs, notebooks o celulares con cámara, cumpliendo con la consigna del proyecto.

Para la **visualización de datos productivos**, se consideró el uso de librerías de gráficos en JavaScript, tales como **Chart.js**, debido a su simplicidad y capacidad de generar dashboards dinámicos a partir de la información almacenada en la base de datos. De este modo, se posibilita un análisis visual de métricas clave como producción, desperdicio y eficiencia.

Finalmente, para la **publicación de la demo en la nube**, se emplearán servicios accesibles y gratuitos como **Render**, **Vercel** o **GitHub Pages**, que permiten desplegar tanto el frontend como el backend sin requerir conocimientos avanzados de administración de servidores. Esto asegura la disponibilidad del prototipo y su fácil acceso para la evaluación académica.

En síntesis, la combinación de **JavaScript, Express.js, PostgreSQL, Face-Api.js y Chart.js**, junto con servicios de despliegue en la nube, constituye un stack tecnológico coherente con los objetivos del trabajo práctico. Esta selección garantiza el cumplimiento de los entregables, optimiza los recursos y asegura la viabilidad del proyecto en un entorno real de PyME alimentaria.

## **Conjuntos de datos simulados**

**1. maestro\_productos**

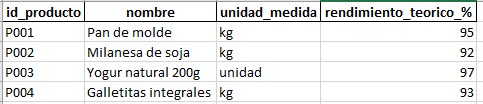
Es el **catálogo de productos** que fabrica la empresa.  
Sirve de **referencia** para vincular lotes de producción con un producto específico.

**Campos:**

* id\_producto → Identificador único del producto (ej.: "P001").
* nombre → Nombre del producto (ej.: "Pan de molde").
* unidad\_medida → En qué unidad se mide (kg, litros, unidades).
* rendimiento\_teorico\_% → Porcentaje de rendimiento esperado según receta (ej.: 95%).

**Uso:** Permite calcular desvíos de producción y de consumo de insumos.

**Ejemplo:**

****

**2. empleados**

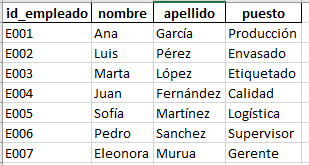
Es el **listado de empleados**.  
Sirve para cruzar datos de **asistencia** y **puestos críticos** con la producción.

**Campos:**

* id\_empleado → Identificador único del empleado.
* nombre, apellido → Datos personales.
* puesto → Rol principal (Producción, Envasado, Etiquetado, Calidad, Logística).
* descriptor\_faceid (vectores de la cara del empleado)

**Uso:** Relacionar ausencias/tardanzas con demoras en lotes o incidencias. Y reconocimiento facial a través de los vectores de la cara.

**Ejemplo:**



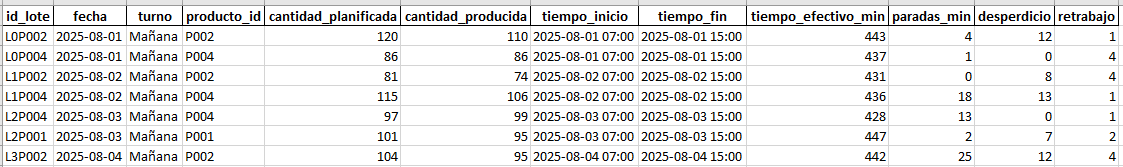
**3. lotes\_produccion**

Es el **registro de cada lote fabricado**.  
Contiene la información central del proceso productivo.

**Campos:**

* id\_lote → Identificador único del lote.
* fecha → Día de producción.
* turno → Turno en que se produjo (Mañana/Tarde).
* producto\_id → Vinculado a maestro\_productos.
* cantidad\_planificada → Lo que se esperaba producir.
* cantidad\_producida → Lo efectivamente producido.
* tiempo\_inicio, tiempo\_fin → Intervalo de producción.
* tiempo\_efectivo\_min → Tiempo productivo real (minutos).
* paradas\_min → Tiempo de paradas.
* desperdicio → Cantidad perdida o descartada.
* retrabajo → Cantidad que necesitó reprocesarse.

**Uso:** Base para KPIs de **eficiencia, desperdicio, cumplimiento de plan y OEE**.

**Ejemplo:**

**4. consumo\_insumos**

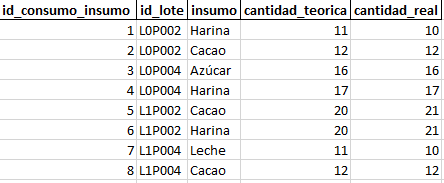
Registra cuántos insumos se usaron en cada lote.  
Sirve para medir el **rendimiento de materias primas**.

**Campos:**

* Id\_consumo\_insumo → Identificador único del consumo del insumo
* id\_lote → Referencia al lote.
* insumo → Nombre del insumo (ej.: Harina, Leche).
* cantidad\_teorica → Lo que debería usarse según receta.
* cantidad\_real → Lo que efectivamente se usó.

**Uso:** Permite detectar **sobrecostos o desperdicio de insumos**.

**Ejemplo:**

****

**5. control\_calidad**

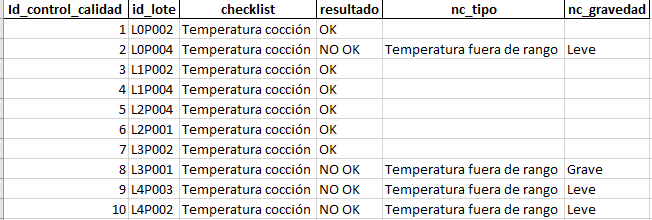
Registra los **controles de calidad** aplicados durante la producción.

**Campos:**

* Id\_control\_calidad → Identificador único del control de calidad
* id\_lote → Referencia al lote.
* checklist → Tipo de control (ej.: Temperatura cocción).
* resultado → OK / NO OK.
* nc\_tipo → Tipo de no conformidad (si hubo).
* nc\_gravedad → Severidad (Leve / Grave).

**Uso:** Permite calcular KPIs de **rechazo, defectos, cumplimiento de variables críticas**.

**Ejemplo:**

****

**6. paradas**

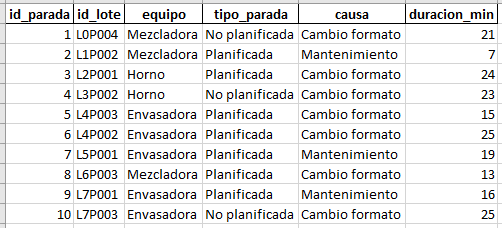
Registra las **paradas de equipos** que afectan la disponibilidad.

**Campos:**

* id\_parada → Identificador único de la parada
* id\_lote → Referencia al lote.
* equipo → Nombre del equipo (Horno, Envasadora).
* tipo\_parada → Planificada (ej.: limpieza) o no planificada (falla).
* causa → Motivo (Mantenimiento, Falla técnica, Cambio de formato).
* duracion\_min → Duración de la parada en minutos.

**Uso:** Base para **Disponibilidad y análisis de OEE**.

**Ejemplo:**

****

**7. ingresos\_salidas**

Es el registro de **ingresos/salidas de empleados**, pensado para integrarse con **reconocimiento facial**.

**Campos:**

* id\_ingreso\_salida → Identificador único del ingreso y salida del empleado
* id\_empleado → Referencia a empleados.
* fecha → Día de registro.
* timestamp\_entrada → Hora exacta de entrada.
* timestamp\_salida → Hora exacta de salida.
* tardanza\_min → Minutos de atraso en la entrada.
* ausencia → Sí / No.

**Uso:** Relacionar **puntualidad y ausentismo con la producción**, ver impacto de ausencias en puestos críticos.

**Ejemplo:**

