

Fundación Universitaria de Popayán
Programa de Ingeniería de Sistemas
2025

Mag. Luis Vejarano
luis.vejarano@docente.fup.edu.co
Líder semillero de investigación
DEVENIAC
Área: Desarrollo Web Full Stack
www.deveniac.com

Mag. Jorge Aguirre
jorge.aguirre@docente.fup.edu.co
Líder semillero de investigación
SMARTER
Área: Desarrollo de Software

Mag. Franco Urbano
franco.urbano@docente.fup.edu.co
Líder semillero de investigación
SENPRO
Área: Desarrollo de Software

Reto Hackatón de Programación

Caso de estudio: Una empresa *retail* dedicada a la venta de ropa enfrenta un gran desafío relacionado con su proceso de expansión y consolidación en otras ciudades del país. En consecuencia ha tenido que aumentar su inventario, para dar una respuesta optima a la demanda real del mercado. Sus clientes reconocen en su ropa y accesorios: calidad, moda y diversidad, a precios justos. Cada mes la empresa maneja un stock de productos variado que incluye diferentes tipos de ropa y tallas, lo que genera grandes volúmenes de información que no siempre se analizan de manera eficiente. Actualmente, la empresa registra sus ventas y existencias de forma manual en hojas de cálculo, dificultando la identificación oportuna de los productos con mayor rotación y aquellos con baja salida, lo que impacta en sus costos operativos, decisiones de compra y rentabilidad.

La empresa maneja el siguiente portafolio por tipo de cliente:

Cliente	Tipo de ropa	Talla
Mujer	ABRIGO, BERMUDA, BUZOS, CAMISAS, FALDA, HOGAR, JEANS TERMINADOS, PANTALONES, PIJAMAS, ROPA INTERIOR, TERCERAS PIEZAS, TSHIRT, TERMINADAS, VESTIDOS.	XXS, XS, S, M, L, XL
Hombre	ABRIGO, BERMUDA, BUZO, CAMISAS, HOGAR, JEANS TERMINADOS, PANTALONES, POLOS, ROPA DE BAÑO, ROPA INTERIOR, TSHIRT TERMINADA.	XXS, XS, S, M, L, XL
Niño	BERMUDA, BUZO, CAMISAS, JEANS TERMINADOS, PANTALONES, POLOS, ROPA DE BAÑO, TSHIRT TERMINADA.	4, 6, 8, 10, 12, 14, 16
Niña	ABRIGO, BERMUDA, BUZO, CAMISAS, FALDA, JEANS TERMINADOS, PANTALONES, TERCERAS PIEZAS, TSHIRT TERMINADA,	4, 6, 8, 10, 12, 14, 16

VESTIDOS.

Objetivo del reto: Diseñar y construir un software tipo web que permita analizar los productos con mayor rotación mensual, para que la empresa pueda:

- Controlar su inventario de manera eficiente.
- Tomar decisiones basadas en datos sobre compras, promociones y reposición.
- Reducir pérdidas por sobre stock o desabastecimiento.
- Visualizar de forma clara las tendencias de ventas por categoría, género, talla y periodo.

Alcance del reto: Los equipos deberán.

1. Diseñar y construir un prototipo software funcional (MVP).
2. La solución debe responder claramente al reto planteado.
3. Incorporar visualizaciones gráficas que faciliten la toma de decisiones (*Dashboard*).
4. Permitir la carga y análisis de datos históricos de ventas o inventario (puede simularse).
5. Generar indicadores clave (KPI) como:
 - Productos más vendidos por categoría, mes y sucursal.
 - Tallas con mayor salida.
 - Comparativos entre géneros o grupos de edad.
 - Porcentaje de rotación mensual o promedio de ventas por tipo de prenda.
 - Otros que consideren.
6. Sugerir acciones automáticas o recomendaciones, como aumentar stock o aplicar descuentos a productos con baja rotación.

Aspectos técnicos sugeridos: Los equipos deben tener en cuenta para dar solución al reto. Las siguientes categorías: infraestructura, desarrollo, buenas prácticas, seguridad, trabajo colaborativo y presentación.

1. Enfoque del desarrollo
 - Se recomienda aplicar metodologías o marcos ágiles (Scrum o Design Thinking) para organizar el trabajo por fases (Idear → Prototipar → Testear → Presentar).
2. Entorno de desarrollo y tecnologías recomendadas
 - Los participantes pueden elegir libremente sus herramientas, pero deben garantizar compatibilidad, documentación y fácil demostración.

Recomendaciones:

- a) Frontend
 - HTML5, CSS3, JavaScript.
 - Frameworks recomendados: React, Vue.js, Angular.
 - Bibliotecas o librerías: Bootstrap o TailwindCSS.

- b) Backend
 - Lenguajes y frameworks sugeridos: Python (Flask/Django), Node.js (Express), C# (.NET Core), Java (Spring Boot).
 - Exponer servicios RESTful o GraphQL.
 - Separar claramente la lógica de negocio de la interfaz.
- c) Bases de datos
 - SQL: MySQL, PostgreSQL, Oracle.
 - NoSQL: MongoDB, Firebase, Redis.
- d) Infraestructura y despliegue
 - Se puede usar GitHub Pages, Netlify, Render, Vercel, Railway o Docker para mostrar el prototipo en línea.
 - Si se requiere nube, se puede usar AWS, Google Cloud o Azure (versiones gratuitas).
 - El proyecto debe poder ejecutarse localmente (documentar instalación y dependencias en un README.md).

3. Seguridad

- No incluir credenciales o llaves privadas en el repositorio público.
- Implementar medidas básicas de seguridad: validación de entradas, cifrado de contraseñas, uso de HTTPS (si aplica), JWT.

4. Colaboración y control de versiones

- Todo el código debe estar en un repositorio colaborativo (GitHub, Bitbucket o GitLab).
- Se recomienda uso del modelo gitflow.
- Incluir archivo README.md con:
 - Descripción del reto.
 - Instrucciones para ejecutar el proyecto.
 - Tecnologías utilizadas.
 - Roles del equipo y autores.
- Usar Jira o Trello para la gestión del proyecto.

5. Buenas prácticas de programación

- Código claro, comentado y estructurado.
- Nombres de variables y funciones significativos.
- Modularización y reutilización de componentes.
- Se valora el uso de patrones de diseño simples (MVC, Observer, Factory, etc.).

6. Reportes visuales para la toma de decisiones

- Incluir dashboards o reportes interactivos.
- Utilizar librerías o herramientas de visualización como:
 - Chart.js, D3.js, Recharts, Plotly, o Google Charts.

- Representar datos mediante:
 - Gráficas de barras, líneas, pastel, mapas de calor o indicadores KPI.
 - Tablas dinámicas o visualizaciones comparativas.

7. Innovación tecnológica

- Se valora la implementación de:
 - Inteligencia artificial (IA generativa, análisis predictivo o chatbots).
- La innovación debe tener un propósito claro, no debe ser un adorno tecnológico.

Presentación técnica del proyecto: El equipo debe demostrar el funcionamiento real del prototipo (demo funcional o video), y la presentación debe incluir:

- Problema detectado.
- Solución propuesta.
- Arquitectura o diagrama de componentes.
- Tecnologías usadas y justificación.
 - Limitaciones y posibles mejoras.
- Demostración del prototipo.

Nota:

- El jurado evaluará tanto la ejecución técnica como la claridad de la explicación.
- El jurado valorara si la exposición o las presentaciones se realizan en inglés.