|  |
| --- |
| **NOMBRE DE LA ASIGNATURA: Computo en la nube** |
| **NOMBRE DEL PROFESOR: Justino Ramírez Ortegon** |
| **NOMBRE DEL ALUMNO: José Manuel Interian Chan** |
| **TEMA:** Proyecto de competencia enero 2026: Plataformas en la nube |

|  |
| --- |
| **RESUMEN:** |
| El presente proyecto consiste en el desarrollo e implementación de un sistema de punto de venta para un negocio de comida tradicional (mondongo kabic), apoyado por servicios basados en plataformas en la nube. El sistema permite la toma de pedidos, su gestión en cocina, el marcado de pedidos como completados y la generación de reportes de ventas, optimizando los procesos operativos del negocio.  Adicionalmente, se integran servicios en la nube bajo los modelos SaaS y PaaS, utilizando herramientas gratuitas como Google Sheets, Google Forms y GitHub Pages, permitiendo el acceso remoto, la centralización de la información y la escalabilidad del sistema sin costos adicionales. |

|  |
| --- |
| **MARCO TEÓRICO:** |
| Las plataformas en la nube permiten a las empresas acceder a recursos tecnológicos sin la necesidad de realizar grandes inversiones en infraestructura física. Estas plataformas se dividen en tres modelos principales: Software as a Service (SaaS), Platform as a Service (PaaS) e Infrastructure as a Service (IaaS).  El modelo SaaS proporciona aplicaciones listas para usarse a través de internet, eliminando la necesidad de instalación local. Ejemplos de este modelo son Google Sheets y Google Forms, los cuales permiten gestionar información, formularios y reportes de manera colaborativa y en tiempo real.  Por otro lado, el modelo PaaS ofrece una plataforma completa para el desarrollo, despliegue y alojamiento de aplicaciones, permitiendo a los desarrolladores concentrarse en la lógica del sistema sin preocuparse por la infraestructura. GitHub Pages es un ejemplo de PaaS que permite hospedar aplicaciones web de forma gratuita.  El uso de plataformas cloud en pequeñas y medianas empresas permite mejorar la eficiencia operativa, facilitar la comunicación entre áreas, garantizar la disponibilidad de la información y preparar al negocio para su crecimiento futuro. |

|  |
| --- |
| **EQUIPO Y SOFTWARE UTILIZACO:** |
| **Equipo de cómputo**   * Laptop con sistema operativo Windows 10 * Dispositivo móvil (celular) para pruebas de acceso remoto   **Software y plataformas**   * Navegador web (Google Chrome / Edge) * Google Sheets (SaaS) * Google Forms (SaaS) * GitHub y GitHub Pages (PaaS) * Editor de código (Visual Studio Code) * Lenguajes web: HTML, CSS y JavaScript |

|  |
| --- |
| **FUNCIONALIDAD DEL PROGRAMA:** |
| El sistema desarrollado cuenta con las siguientes funcionalidades principales:  **1. Toma de pedidos**  Permite registrar pedidos de clientes indicando los productos solicitados, cantidad y observaciones.      Este formulario desarrollado en HTML está conectado por medio de un forms para registrar la respuesta en Google forms como entrada de datos (Saas)    **2. Gestión de pedidos**  Los pedidos registrados se visualizan en una interfaz donde el personal de cocina puede dar seguimiento al estado de cada pedido.    Se conecta a Google sheet (Saas) siendo utilizado para manejar los datos y obtener resultados en tiempo real, fácil de acceder    **3. Marcado de pedidos como completados**  Una vez que el pedido está listo, se marca como completado, permitiendo llevar un control de pedidos atendidos.  Al igual que la implementacion de numero telefonico para confirmar pedidos    **4. Generación de reportes**  Los pedidos capturados mediante Google Forms se almacenan automáticamente en Google Sheets, donde se lleva un control centralizado de los pedidos registrados.  Una vez que los pedidos son marcados como *Completados*, estos se reflejan en la hoja de cálculo, permitiendo generar reportes diarios de ventas, conteo de pedidos y análisis básico del flujo de trabajo.  Google Sheets funciona como una base de datos en la nube y como herramienta de reportes, permitiendo el acceso en tiempo real desde cualquier dispositivo con conexión a Internet.  Con la siguiente formula =SUMA(D2:D)  Se obtiene el total de cantidades de piezas vendidas      **5. Página web del negocio**  El sistema y la página informativa del negocio se encuentran publicados en **GitHub Pages**, permitiendo su acceso desde cualquier dispositivo.        **PASO 1 – Creamos la “base de datos” (Google Sheets)**  Creamos un formulario llamado: **Registro de pedidos – Mondongo Kabic**  **Campos recomendados:**   * Nombre del cliente (Respuesta corta) * Pedido (Respuesta corta) * Cantidad (Número) * Estado (Respuesta corta → valor por defecto: *Pendiente*) * Fecha (Respuesta corta o automática)  1. En Forms:    * Ve a **Respuestas**    * Clic en el ícono verde → se crea el **Google Sheets**       **PASO 2 – Crear la aplicación web (HTML)**  Crea un archivo llamado **index.html**  **PASO 3 – Conectar HTML con Google Forms**  Este paso es CLAVE.  **🔍 Obtener la URL del Form**   1. En Google Forms → **Enviar** 2. Icono <> 3. Copia la URL del action <https://docs.google.com/forms/u/0/d/e/1FAIpQLSca03yAL_D97jPuF-M5c2TCuvXNMxZa0JfaAiPcuQ1abWcSJg/formResponse> 4. Form response   **Obtener los entry.xxxxx**   1. Clic derecho → *Inspeccionar* 2. Ve a cada campo 3. Copia el atributo name="entry.XXXX"   Se hace mediante los entry, y estos mismos se pueden buscar con control F, entry    **PASO 4 – Gestión en cocina (visual)**  Puedes usar:   * El **Google Sheets** abierto * Filtro por “Estado = Pendiente” * Cambiar a “Completado” manualmente   Cambio de pendiente a confirmado    **PASO 5 – Subir a GitHub Pages (PaaS)**   1. Crear repositorio: **punto-venta-mondongo** 2. Subir:    * index.html 3. Settings → Pages 4. Branch: main 5. Guardar     Se sube a Github Pages, se autoriza desde que carpeta    Y en cuestión de segundos se despliega  La conexión se realiza mediante servicios SaaS y PaaS totalmente gratuitos, utilizando Google Forms como interfaz de entrada y Google Sheets como almacenamiento en la nube, permitiendo escalabilidad sin infraestructura propia. |

|  |
| --- |
| **INSTRUCCIONES DE USO:** |
| 1. Acceder al sistema desde el navegador web.   https://jsinterian.github.io/punto-venta-mondongo/   1. Registrar un pedido ingresando los datos solicitados. 2. El pedido aparece automáticamente en el área de cocina. 3. Una vez preparado, el pedido se marca como completado. 4. El sistema registra el pedido en Google Sheets. 5. El administrador puede consultar el reporte diario de ventas.             Reporte final    conclusión:  Así logramos concluir con el desarrollo de un sistema de punto de venta para una empresa de venta de comida, más específicamente mondongo, entonces hicimos uso de servicios en la nube como Saas, en este caso usamos 2 software gratuitos, los cuales son Google forms y Google sheet. Desarrollamos una página web también y la subimos por medio de GitHub Pages, siendo usada como Paas, ya que la plataforma nos brinda el entorno para comenzar a desarrollar nuestros proyectos de software en línea, quedaron cubiertas las necesidades de la propuesta. |

|  |
| --- |
| **CODIGO RELEVANTE:** |
| <form        action="https://docs.google.com/forms/d/e/1FAIpQLSca03yAL\_D97jPuF-M5c2TCuvXNMxZa0JfaAiPcuQ1abWcSJg/formResponse"        method="POST"        target="\_self">        <div class="row g-3">          <!-- Cliente -->          <div class="col-md-4">            <label class="form-label">Nombre del Cliente:</label>            <input type="text" name="entry.1746796448" class="form-control">          </div>          <!-- Cantidad -->          <div class="col-md-2">            <label class="form-label">Cantidad:</label>            <input type="number" name="entry.428740919" class="form-control" min="1" value="1" required>          </div>          <!-- Platillo -->          <div class="col-md-4">            <label class="form-label">Platillo:</label>            <select name="entry.1748853043" class="form-select" onchange="toggleOtroPlatillo()">              <option value="Mondongo">Mondongo</option>              <option value="Mondongo Cabeza">Cabeza</option>              <option value="Otro">Otro</option>            </select>          </div>          <!-- Otro platillo -->          <div class="col-md-4" id="otroDiv" style="display:none;">            <label class="form-label">Especificar otro platillo:</label>            <input type="text" name="entry.9999999999" class="form-control">          </div>          <!-- Tipo pedido -->          <div class="col-md-4">            <label class="form-label">Tipo de pedido:</label>            <select name="entry.8888888888" class="form-select">              <option value="En sitio">En sitio</option>              <option value="Domicilio">Domicilio</option>            </select>          </div>          <!-- Estado -->          <input type="hidden" name="entry.1321358064" value="Pendiente">          <!-- Fecha -->          <input type="hidden" id="year" name="entry.577004131\_year">          <input type="hidden" id="month" name="entry.577004131\_month">          <input type="hidden" id="day" name="entry.577004131\_day">          <div class="col-12 text-center mt-4">            <button type="submit" class="btn btn-success btn-lg">✅ Agregar Pedido</button>          </div> |

|  |
| --- |
| **BIBLIOGRAFÍA:** |
|  Google Cloud. (2025). *Google Workspace Documentation*   GitHub. (2025). *GitHub Pages Documentation*   Sommerville, I. *Ingeniería de Software*   <https://docs.github.com>   https://www.google.com/forms   https://www.google.com/sheets |