**Presentado por:**

**Jesus Alberto Santana Maza**

**5 semestre**

**Responsabilidades Claras**: Al dividir las clases según sus responsabilidades, aseguramos que cada una tenga un propósito claro y específico. Por ejemplo, en nuestro proyecto, podemos desarrollar clases distintas para administrar cuentas de usuario, inventarios de artículos y transacciones de intercambio. Esto no solo mejora la organización, sino que también facilita el manejo de cada componente del sistema de manera más eficiente.

**Tamaño y Mantenimiento**: Al mantener nuestras clases pequeñas y enfocadas en responsabilidades específicas, simplificamos considerablemente el proceso de prueba y depuración. Esto es especialmente crucial en un sistema de intercambio de artículos, donde las funcionalidades pueden evolucionar con el tiempo. Un código bien estructurado nos permitirá adaptarnos rápidamente a esos cambios, asegurando que nuestro sistema sea flexible y fácil de actualizar.

**Organización del Código**: La correcta organización de las clases es fundamental para la comprensión y mantenimiento del código. Si nuestro proyecto involucra diferentes tipos de artículos, usuarios y transacciones, una buena estructura nos permitirá gestionar estas entidades de manera más eficiente y con menos riesgo de cometer errores.