**Tarea**

**Objetivo**: Desarrollar los métodos de la clase Avión que nos permitan implementar los requerimientos funcionales del caso de estudio.

Para cada uno de los problemas que se plantean a continuación, escriba el método que lo resuelve. No olvide identificar primero el patrón de algoritmo que se necesita y usar las guías que se dieron en secciones anteriores.

Calcular el número de sillas ejecutivas ocupadas en el avión:

**Def contarSillasEjecutivasOcupadas(self):**

Localizar la silla en la que se encuentra el pasajero identificado con la cédula que se entrega como parámetro. Si no hay ningún pasajero en clase ejecutiva con esa cédula, el método retorna null .

**Def buscarPasajeroEjecutivo: Silla (self, Cedula:String )**

Localizar una silla económica disponible, en una localización dada (ventana, centro o pasillo). Si no existe ninguna, el método retorna null :

**Def buscarSillaEconomicaLibre: Silla (self, ubicación: Ubicacion):**

Asignar al pasajero que se recibe como parámetro una silla en clase económica que esté libre (en la ubicación pedida). Si el proceso tiene éxito, el método retorna verdadero. En caso contrario, retorna falso:

**Def asignarSillaEconomica: boolean (self, ubicación: Ubicacion, pasajero: Pasajero ):**

Anular la reserva en clase ejecutiva que tenía el pasajero con la cédula dada. Retorna verdadero si el proceso tiene éxito:

**Def anularReservaEjecutivo: boolean (self, cedula: String )**

Contar el número de puestos disponibles en una ventana, en la zona económica del avión:

**Def contarVentanasEconomica: int (self )**

Informar si en la zona económica del avión hay dos personas que se llamen igual. Patrón de doble recorrido:

**Def hayDosHomonimosEconomica: boolean ( self )**