**RF10-REGISTRAR CAMPAÑA**

Se define los servicios de salud

Se define cantidad deseada para cada servicio de salud

Se cuenta la utilización de cada uno de los servicios de salud

Se compara la cantidad utilizada con la cantidad deseada de cada servicio de salud

Si la cantidad utilizada es mayor o igual a la cantidad deseada se selecciona el servicio de salud para participar en la campaña

Se define los horarios de atención para cada uno de los servicios de salud

Se realiza la reserva para cada uno de los servicios

**RF11-CANCELAR SERVICIOS DE LA CAMPAÑA.**

Se cuenta el número de personas que han reservado cada uno de los servicios de la campaña.

Se compara el número de reservas con el total dispuesto para la campaña, si el número de reservas es menor al total dispuesto para la campaña se libera cada uno de los servicios y la campaña.

**RF12-DESHABILITAR SERVICIOS DE SALUD**.

Se definen los servicios de salud para cada IPS

Se define el rango de fechas

Los servicios definidos se bloquean para que no puedan ser reservados en el rango de fechas definido

Para cada servicio de salud de cada IPS se encuentra servicio equivalente de la IPS

Se evalúa si los servicios de salud tienen citas programadas para el rango de fechas, de ser así se reasigna al servicio equivalente de la IPS

Se genera reporte de las reasignaciones al servicio equivalente

Se genera reporte de los servicios que no pudieron ser reasignados

**RF13-REGISTRAR LA REPERTURA DE SERVICIOS DE SALUD.**

Para cada uno de los servicios de salud que están reservados para la campaña se liberan y quedan disponibles para que los afiliados puedan solicitar citas

**PROPIEDADES ACID**

Para cumplir con las propiedades ACID, dentro del proyecto se ejecutó cada requerimiento con un bloque try catch, en donde se creó una transacción. Al principio, se usó el método begin() para iniciar la transacción. Si la ejecución presenta errores, en el bloque de catch se realiza un rollback. Si la ejecución es perfecta, se hace un commit.