**Requerimientos -> Req**

**Criterios de aceptación -> Cr**

**Escenario de prueba -> Esc**

* Req-1: Se debe especificar un rango. (Constructor)
  + Cr-1-1: Se deben especificar dos números enteros como extremos (límites), separados por comas, sin espacios y encerrados por corchetes o paréntesis. Se debe interpretar que ‘parentesis’ implica ‘abierto’ (no incluye el extremo) y ‘corchetes’ implica ‘cerrado’(incluye el extremo).
    - Esc-1-1-1: range(“[1,2]”) -> limite inferior = 1 && limite superior = 2
    - Esc-1-1-2: range(“(1,3)”) -> limite inferior = 2 && limite superior = 2
    - Esc-1-1-3: range(“(1,2]”) -> limite inferior = 2 && limite superior = 2
    - Esc-1-1-4: range(“[1,2)”) -> limite inferior = 1 && limite superior = 1
  + Cr-1-2: Un rango vacío debe ser reconocido como tal.
    - Esc-1-2-1: range(“(1,2)”) -> “rango vacío”
* Req-2: Se debe determinar si un entero pertenece a un rango especificado.
  + Cr-2-1: Un entero N pertenece a un rango dado R, si N es mayor o igual al límite inferior de R y menor o igual al límite superior de R.
    - Esc-2-1-1: range(“(2,6)”).Contains(6) = False
    - Esc-2-1-2: range(“(2,6]”).Contains(6) = True
* Req-3: Se debe determinar si un entero no pertenece a un rango especificado.
  + Cr-3-1: Un entero N no pertenece a un rango dado R, si N es menor al límite inferior de R y mayor al límite superior de R.
    - Esc-3-1-1: range(“[2,9]”).notContains(2) = False
    - Esc-3-1-2: range(“(2,9]”).notContains(2) = True
* Req-4: Se debe determinar todos los números dentro de un rango especificado.
  + Cr-4-1: Solo se deben considerar números enteros contenidos en el rango especificado, es decir, todo entero N mayor o igual al límite inferior de R y menor o igual al límite superior de R.
    - Esc-4-1-1: range(“[1,4]”).getAllPoints() = “1,2,3,4”
    - Esc-4-1-2: range(“(1,4)”).getAllPoints() = “2,3”
    - Esc-4-1-3: range(“(1,3)”).getAllPoints() = ”2”
    - Esc-4-1-3: range(“(1,2)”).getAllPoints() -> “rango vacío”
* Req-5: Se debe determinar si un rango dado se encuentra dentro (y no solapa) de otro rango especificado.
  + Cr-5-1: Un rango R1 contiene a otro R2, si el límite inferior de R2 es mayor o igual al límite inferior de R1 y el límite superior de R2 es menor o igual al límite superior de R1.
    - Esc-5-1-1: range(“(0,9)”).ContainsRange(“[0,9]”) = False
  + Cr-5-2: Se debe respetar la propiedad de conjuntos que dicta *todo conjunto contiene al conjunto vacío.*
    - Esc-5-1-2: range(“[2,3]”).ContainsRange(“(2,3)”) = True
* Req-6: Se deben determinar los límites superior e inferior de un rango especificado.
  + Cr-6-1: Solo se deben considerar los límites superior e inferior del rango especificado.
    - Esc-6-1-1: [0,10).endPoints() = “0,9”
    - Esc-6-1-2: (0,10).endPoints() = “1,9”
* Req-7: Se debe determinar si un rango dado se solapa con otro rango especificado.
  + Cr-7-1: Dado un rango R1 y un rango R2, R1 se solapa con R2 si existe, por lo menos, un entero N que pertenece tanto a R1 como a R2, es decir, R1 contiene a N y R2 contiene a N.
    - Esc-7-1-1: range(“[2,5)”).overlapsRange(“[1,2)”) = False
    - Esc-7-1-2: range(“[2,5)”).overlapsRange(“[1,2]”) = True
* Req-8: Se debe determinar si un rango dado es igual a otro rango especificado.
  + Cr-8-1: Dado un rango R1 y un rango R2, R1 es igual a R2 si el límite inferior de R1 es igual al límite inferior de R2 y el límite superior de R1 es igual al límite superior de R2.
    - Esc-8-1-1: range(“(5,3)”).Equals(“(5,3)”) = True